## A WEB-BASED WAREHOUSE INVENTORY MANAGEMENT SYSTEM FOR STOCK ITEMS TRACKING

ELAINE LOW JING YI

UNIVERSITI TUNKU ABDUL RAHMAN

## A WEB-BASED WAREHOUSE INVENTORY MANAGEMENT SYSTEM FOR STOCK ITEMS TRACKING

ELAINE LOW JING YI

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering

Lee Kong Chian Faculty of Engineering and Science Universiti Tunku Abdul Rahman

September 2022

#### DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Λ	
	•
The	

Signature	:
Name	: Elaine Low Jing Yi
ID No.	: 18UEB03140
Date	: 08/09/2022

#### APPROVAL FOR SUBMISSION

I certify that this project report entitled **"A WEB-BASED WAREHOUSE INVENTORY MANAGEMENT SYSTEM FOR STOCK ITEMS TRACKING"** was prepared by **ELAINE LOW JING YI** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Hons) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature	:	Lai
Supervisor	:	Dr Lai Yen Lung
Date	:	
Signature	:	
Co-Supervisor	:	
Date	:	

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2022, Elaine Low Jing Yi. All right reserved.

#### **ACKNOWLEDGEMENTS**

I would like to thank everyone who contributed to the successful completion of this project and express my gratitude to my supervisor, Dr Lai Yen Lung for his guidance and enormous patience throughout the completion of this project as well as Mr Sor Kean Vee for his advice and suggestions to improve this project.

In addition, I would also like to express my gratitude to my loving family members and friends who had given me support and encouragement during my degree studies.

#### ABSTRACT

Warehouses act as an important role in the supply chain to store inventories. To keep up with the customer's demand on time, inventories need to be organized and tracked with proper inventory system control so that business operations are done efficiently. However, in the current competitive market environment, warehouses have to deal with discrepancies between the recorded inventory count and the actual item count. Besides, to maintain the accurate inventory count, inventory system control such as cycle counting need to be in place, in which extra manpower is needed to classify the stock keeping unit (SKUs) and schedule the cycle counting process. Moreover, the existing SKU classification methods have their limitations in classifying the SKU. Therefore, a web-based warehouse inventory management system was implemented to address the mentioned problems. The system was developed using the phased-development methodology and was divided into three phases to develop features including the automated cycle counting scheduler, real-time check-in/ out stock module, report management, and the management of warehouses, users, inventories and categories. The front end of the system was developed using Vue.js whereas the back end (APIs) was developed with Pusher Channels for the real-time inventory stock tracking module. ng module. Web service testing was carried out using Postman and the entire system was deployed to Heroku after all test cases of web service testing passed for user acceptance testing and usability testing. The average SUS score was 85.5, hence, it could be concluded that the user interface of implemented system is simple and easy to use. Besides, 100 iterations of check-in/out stock were done and the time differences between creating stock and receiving API message was observed and recorded. The average delay time for the real-time functionality was around 1 second. In a nutshell, the implemented system successfully addressed and achieved the main project objectives to keep track of inventory stocks in real-time, automate cycle counting scheduler and improve and implement the existing SKU classification with higher completeness and efficiency.

## **TABLE OF CONTENTS**

DECLARATION	iii
APPROVAL FOR SUBMISSION	iv
ACKNOWLEDGEMENTS	vi
ABSTRACT	vii
TABLE OF CONTENTS	viii
LIST OF TABLES	xiii
LIST OF FIGURES	xvii
LIST OF SYMBOLS / ABBREVIATIONS	xxiii
LIST OF APPENDICES	XXV

## CHAPTER

1	INTR	ODUCT	ION	1
	1.1	Introdu	iction	1
	1.2	Backgi	ound of the Problem	2
	1.3	Proble	m Statement	3
		1.3.1	Inaccuracy of inventory record	3
		1.3.2	Difficulties in scheduling cycle count	ting
		progres	ss in the most efficient way	3
		1.3.3	Limitation in SKU classification methods.	4
	1.4	Project	Objectives	5
	1.5	Project	Solution	6
	1.6	Project	Approach	6
	1.7	Project	Scope	8
		1.7.1	Inventory Stock Tracking	9
		1.7.2	Cycle Counting Auto Scheduler	9
		1.7.3	Report Management	9
2	LITE	RATURI	E REVIEW	11

2.1	Introduction	11

2.2	Similar	r Projects Review	11
	2.2.1	Challenges of existing system	11
	2.2.2	Features of inventory management system	12
	2.2.3	Significance and Limitation	12
2.3	Cycle	Counting	13
	2.3.1	Traditional ABC analysis	14
	2.3.2	ABC-VED	15
	2.3.3	Cycle Counting Scheduling	17
2.4	Softwa	re Development Methodologies	24
	2.4.1	Waterfall Method	24
	2.4.2	Rapid Application Development (RAD)	25
	2.4.3	Agile Development Methodology	27
	2.4.4	Choosing a methodology	28
2.5	Conclu	ision	29
MET	HODOL	OGY AND WORK PLAN	31
3.1	Introdu	iction	31
3.2	Phased	Development-Based Methodology	31
	3.2.1	Planning and Analysis	32
	3.2.2	Design	35
	3.2.3	Development and Testing	36
	3.2.4	Closing	37
3.3	Project	Plan	38
	3.3.1	WBS	38
	3.3.2	Gantt chart	42
3.4	Develo	opment Tools	44
	3.4.1	Visual Studio Code	44
	3.4.2	Axure RP 10	44
	3.4.3	Vue.js	44
	3.4.4	Laravel	45
	3.4.5	MySQL	45
	3.4.6	GitHub	45
	3.4.7	SourceTree	45
3.5	Conclu	ision	45

3

PRO.	JECT SP	ECIFICATION	47
4.1	Introdu	uction	47
4.2	Facts H	Finding	47
	4.2.1	Demographic Information	47
	4.2.2	General Information	48
	4.2.3	Inventory Accuracy	49
	4.2.4	Physical Count / Cycle Counting	51
	4.2.5	Summary of survey	54
4.3	Requir	rement Specification	55
	4.3.1	Functional Requirement	55
	4.3.2	Non-Functional Requirement	57
4.4	System	n Use Case	58
	4.4.1	Use Case Diagram	58
	4.4.2	Use Case Description	59
4.5	Entity	Relationship Diagram (ERD)	77
4.6	Prototype		78
	4.6.1	Inventory stock recording	78
	4.6.2	Cycle counting auto scheduler	81
	4.6.3	Report Management	85
4.7	Summ	ary	90
SYST	TEM DES	SIGN	91
51	Introdu	uction	91

х

5.1	Introdu	Introduction		
5.2	System	System Architecture Design		
5.3	Model	Modelling Diagram		
	5.3.1	Data Flow Diagram	93	
	5.3.2	Interface Flow Diagram	111	
5.4	User Interface Design			
	5.4.1	User (Admin, Manager and Staff)	113	
	5.4.2	Admin	113	
	5.4.3	Manager	124	
	5.4.4	Staff	129	
	5.4.5	Staff & Manager	132	
	5.4.6	Admin & Manager	134	

	5.5	Summary 1	36
6	SYST	TEM IMPLEMENTATION 1	.37
	6.1	Introduction 1	37
	6.2	Backend 1	37
		6.2.1 API List 1	37
		6.2.2 Secure the REST API 1	42
		6.2.3 Pusher Channel 1	42
	6.3	Frontend 1	44
		6.3.1 UI Libraries 1	44
		6.3.2 JS Libraries 1	46
	6.4	Deployment to Heroku 1	47
	6.5	Conclusion 1	50
7	SYST	FEM TESTING 1	51
	7.1	Introduction 1	51
	7.2	Web Service Testing 1	51
	7.3	Realtime Check In/Out Module Testing 1	84
	7.4	User Acceptance Testing and Usability Testing 1	87
		7.4.1 User Acceptance Testing Result 1	89
		7.4.2 Usability Testing Result 1	92
	7.5	Traceability between Use Cases, Function	nal
	Requi	irements and Test Cases 1	95
		7.5.1 Use Cases Table 1	95
		7.5.2 Functional Requirements Table 1	96
		7.5.3 Traceability Matrix 1	98
8	CON	CLUSIONS AND RECOMMENDATIONS 2	200
	8.1	Introduction 2	200
	8.2	Objectives Achievement 2	201
		·	201

- 8.2.2 Objective 2 201
- 8.2.3 Objective 3 201

xi

8.3	Project Limitations	207
8.4	Recommendations for future work	208
REFERENCES		210
APPENDICES		213

#### xiii

## LIST OF TABLES

Table 1:1: Roles and Responsibilities for Warehouse Inventory Managemen System	nt 8
Table 2:1: Summary of features developed by James (2016), Maharjan and Humagain (2016), and Tahir (2020)	12
Table 2:2: Counting Frequency Adopted by PT XYZ	15
Table 2:3: ABC-VED classification	16
Table 2:4: Analysis of cycle counting policies	16
Table 2:5: Daily counts for Group I, II, and III	18
Table 2:6: Necessary data for each input	22
Table 4:1: Summary of responses on physical counting or cycle counting	52
Table 4:2: Functional requirements by roles	55
Table 4:3: Use case description of login account	59
Table 4:4: Use case description of manage inventory	60
Table 4:5: Use case description of view all inventories	62
Table 4:6: Use case description of manage warehouse	63
Table 4:7: Use case description of manage user	65
Table 4:8: Use case description of view reports	67
Table 4:9: Use case description of manage cycle counting approval report	69
Table 4:10: Use case description of manage cycle counting	70
Table 4:11: Use case description of record inventory	72
Table 4:12: Use case description of perform cycle counting	73
Table 4:13: Use case description of view all staff	74
Table 4:14: Use case description of manage category	75
Table 6:1: List of API endpoints	138
Table 7:1: Summary of web service testing's test cases with status	151

Table 7:2: Test case of user login	154
Table 7:3: Test case of logging out	154
Table 7:4: Test case of viewing inventory list	155
Table 7:5: Test case of viewing inventory details	155
Table 7:6: Test case of adding new inventory with category assigne	d 156
Table 7:7: Test case of adding new inventory with no category assig	gned 157
Table 7:8: Test case of updating an inventory with different categor	ry 158
Table 7:9: Test case of updating an inventory with same or no categories of the same or no categories of the same	gory 159
Table 7:10: Test case of deleting an inventory	159
Table 7:11: Test case of viewing warehouse list	160
Table 7:12: Test case of viewing warehouse details	160
Table 7:13: Test case of multi-assigning storage bins of a warehous	e 161
Table 7:14: Test case of editing inventory of a warehouse storage bi	in 162
Table 7:15: Test case of adding new warehouse	163
Table 7:16: Test case of updating a warehouse	163
Table 7:17: Test case of deleting a warehouse	164
Table 7:18: Test case of viewing user list	164
Table 7:19: Test case of viewing only active staffs	165
Table 7:20: Test case of viewing user details	165
Table 7:21: Test case of adding new user	166
Table 7:22: Test case of updating a staff	167
Table 7:23: Test case of deleting a user	168
Table 7:24: Test case of resetting password	168
Table 7:25: Test case of viewing category list	169
Table 7:26: Test case of adding new category	169

Table 7:27: Test case of updating a category	170
Table 7:28: Test case of deleting a category	170
Table 7:29: Test case of viewing daily report list	171
Table 7:30: Test case of viewing cycle counting approval report list	171
Table 7:31: Test case of approving cycle counting approval report	172
Table 7:32: Test case of rejecting cycle counting approval report with recounting	173
Table 7:33: Test case of rejecting cycle counting approval report without recounting	174
Table 7:34: Test case of viewing cycle counting summary report list	175
Table 7:35: Test case of starting cycle counting	176
Table 7:36: Test case of viewing cycle counting	178
Table 7:37: Test case of reassigning staff for cycle count schedules	179
Table 7:38: Test case of viewing all staffs	180
Table 7:39: Test case of performing cycle counting	181
Table 7:40: Test case of checking in inventory stock	182
Table 7:41: Test case of checking out inventory stock	183
Table 7:42: Testing iterations of delay time	184
Table 7:43: Summary of User Acceptance Testing Result	190
Table 7:44: SUS score collected	193
Table 7:45: Responses on open-ended questions	194
Table 7:46: Use Cases Table	195
Table 7:47: Functional requirements table	196
Table 7:48: Traceability matrix between use cases, functional requirement web service testing and user acceptance testing	s, 198
Table 8:1: Pseudocode of ABC and ABC-VED algorithm	203
Table 8:2: Pseudocode of proposed classification algorithm	206

208

## LIST OF FIGURES

Figure 1.1: System overview	6
Figure 1.2: Phased Development Based Methodology	7
Figure 2.1: General concepts of ABC cycle counting method	15
Figure 2.2: Cycle counting Schedule UI	19
Figure 2.3: TOC Buffer zones and penetration rates of each zone	20
Figure 2.4: Illustration of buffer time management	20
Figure 2.5: Conceptual model of cycle counting scheduling support system	21
Figure 2.6: Problem solving schematics	22
Figure 2.7: Counting frequency for each class	23
Figure 2.8: Colour coding of counting urgency class	23
Figure 2.9: Final categorization results	23
Figure 2.10: Layout of monitoring dashboard	24
Figure 2.11: Waterfall development approach	25
Figure 2.12: RAD approach	26
Figure 2.13: Phased development-based approach	27
Figure 2.14: Agile development approach	28
Figure 2.15: Methodologies evaluation results	29
Figure 2.16: Criteria in selecting a development methodology	29
Figure 3.1: Phased development-based approach	32
Figure 3.2: System architecture	35
Figure 3.3: Planning and analysis phase timeline	42
Figure 3.4: Design phase timeline	42
Figure 3.5: Development and testing phase 1 timeline	42
Figure 3.6: Development and testing phase 2 timeline	43

Figure 3.7: Development and testing phase 3 timeline	43
Figure 3.8: Closing phase timeline	43
Figure 4.1: Age of 20 respondents	47
Figure 4.2: Gender of the respondents	48
Figure 4.3: Statistic of respondents to have warehouse management system a place in their warehouse or not	in 48
Figure 4.4: Distribution of respondents who has more than one warehouse from multiple locations or not	49
Figure 4.5: Methods of inventory recording of respondents	49
Figure 4.6: Statistic of whether any discrepancies occur between inventory record and actual count	50
Figure 4.7: Statistic of whether any inconsistency of inventory count occurs across multiple location	50
Figure 4.8: Statistic of whether inventory and data accuracy would affect the business operation	e 51
Figure 4.9: Possible reasons on why data accuracy would affect business operation	51
Figure 4.10: Distribution of inventory control methods implemented in respondents' warehouse(s)	51
Figure 4.11: Use case diagram	58
Figure 4.12: ERD diagram	77
Figure 4.13: Staff- record inventory page	78
Figure 4.14: Staff- check in inventory popup	79
Figure 4.15: Staff- check out inventory popup	79
Figure 4.16: Staff- show bin location after check in inventory	80
Figure 4.17: Warehouse manager- manage cycle counting page	81
Figure 4.18: Warehouse manager- start cycle counting	82
Figure 4.19: Warehouse manager- start cycle counting confirmation popup	83
Figure 4.20: Warehouse manager- view upcoming cycle counting	83

Figure 4.21: Staff- perform cycle counting	84
Figure 4.22: Staff- count SKU	84
Figure 4.23: Warehouse manager - view daily reports	85
Figure 4.24: Admin - view daily reports	86
Figure 4.25: Warehouse manager & admin- view daily report details	86
Figure 4.26: Warehouse manager- view cycle counting summary reports	87
Figure 4.27: Admin- view cycle counting summary reports	87
Figure 4.28: Warehouse manager & admin- view cycle counting summary report details	88
Figure 4.29: Warehouse manager - view cycle counting approval reports	89
Figure 4.30: Warehouse manager - approve cycle counting approval reports	; 89
Figure 4.31: Warehouse manager - reject cycle counting approval reports	89
Figure 5.1: System architecture design	92
Figure 5.2: Context diagram	94
Figure 5.3: Data flow diagram level-1	95
Figure 5.4: Data flow diagram level-2 (Manage inventory)	97
Figure 5.5: Data flow diagram level-2 (Manage user)	98
Figure 5.6: Data flow diagram level-2 (Manage warehouse)	99
Figure 5.7: Data flow diagram level-2 (Manage category)	100
Figure 5.8: Data flow diagram level-2 (Manage cycle counting)	101
Figure 5.9: Data flow diagram level-2 (Perform cycle counting)	102
Figure 5.10: Data flow diagram level-2 (Record inventory)	103
Figure 5.11: Data flow diagram level-2 (View reports)	104
Figure 5.12: Data flow diagram level-3 (View warehouse details)	105
Figure 5.13: Data flow diagram level-3 (View cycle counting approval reports)	106

Figure 5.14: Data flow diagram level-3 (Start cycle counting)	107
Figure 5.15: Data flow diagram level-3 (View cycle counting)	108
Figure 5.16: Data flow diagram level-4 (Approve cycle count approval re-	eport) 109

Figure 5.17: Data flow diagram level-4 (Reject cycle count approval report) 110

Figure 5.18:	Interface flow diagram for staff role	111
Figure 5.19:	Interface flow diagram for admin role	111
Figure 5.20:	Interface flow diagram for manager role	112
Figure 5.21:	User interface design for login account	113
Figure 5.22:	User interface design for manage inventory page	114
Figure 5.23:	User interface design for adding inventory	114
Figure 5.24:	User interface design for updating inventory	115
Figure 5.25:	User interface design for deleting inventory	115
Figure 5.26:	User interface design for manage warehouse page	116
Figure 5.27:	User interface design for view warehouse details & view staff	list 117
Figure 5.28:	User interface design for view warehouse details & view stora bins	ge 117
Figure 5.29:	User interface design for multi-assigning storage bins	118
Figure 5.30:	User interface design for editing storage bin's inventory	118
Figure 5.31:	User interface design for adding new warehouse	119
Figure 5.32:	User interface design for editing warehouse details	119
Figure 5.33:	User interface design for deleting warehouse	120
Figure 5.34:	User interface design for managing category page	120
Figure 5.35:	User interface design for adding new category	121
Figure 5.36:	User interface design for editing category	121

Figure 5.37: User interface design for deleting category	122
Figure 5.38: User interface design for manage user page	122
Figure 5.39: User interface design for viewing user details	123
Figure 5.40: User interface design for adding new user	123
Figure 5.41: User interface design for editing user	123
Figure 5.42: User interface design for deleting user	124
Figure 5.43: User interface design for viewing cycle count approval reports	s 124
Figure 5.44: User interface design for approving cycle count approval repo	ort 125
Figure 5.45: User interface design for rejecting cycle count approval report	t 125
Figure 5.46: User interface design for starting cycle counting	126
Figure 5.47: User interface design for view cycle counting page	127
Figure 5.48: User interface design for reassigning staff	127
Figure 5.49: User interface design for viewing cycle counting settings	128
Figure 5.50: User interface design for viewing assigned staff for cycle counting	128
Figure 5.51: User interface design for viewing SKUs for cycle counting	128
Figure 5.52: User interface design for viewing all staff	129
Figure 5.53: User interface design for cycle counting page	130
Figure 5.54: User interface design for counting SKU with no bin assigned	130
Figure 5.55: User interface design for counting SKU with bin assigned	131
Figure 5.56: User interface design for check in/out stock page	131
Figure 5.57: User interface design for checking in stock	132
Figure 5.58: User interface design for checking out stock	132
Figure 5.59: User interface design for view inventory page	133
Figure 5.60: User interface design for view inventory details	133

Figure 5.61: User interface design for viewing daily reports	134
Figure 5.62: User interface design for viewing daily report details	135
Figure 5.63: User interface design for viewing cycle count summary report	rts 135
Figure 5.64: User interface design for viewing cycle count summary repordetails	rt 135
Figure 6.1: Overall mechanism of JWT	142
Figure 6.2: Overview of Pusher Channels dashboard	143
Figure 6.3: Events created for channel subscription	143
Figure 6.4: API message broadcasted to all subscribers of the channel.	144
Figure 6.5: Success notice component	144
Figure 6.6: Modal with customized footer.	145
Figure 6.7: Login component	145
Figure 6.8: Navigation bar component	146
Figure 6.9: Filtering list of inventories by warehouse	147
Figure 6.10: Filtering a warehouse's storage bins using category	147
Figure 6.11: Config vars of the Heroku app	149
Figure 6.12: Overview of Heroku app	150
Figure 6.13: Deployed web-based inventory management system on Hero	ku 150
Figure 7.1: Graph of time differences against iteration number	187
Figure 7.2: System Usability Scale Scoring	193
Figure 8.1: Coding implementation of ABC-VED	202

## LIST OF SYMBOLS / ABBREVIATIONS

API	Application Programming Interface
AUP	Agile Unified Process
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
ERP	Enterprise Resource Planning
FDD	Feature-Driven Development
HTML	HyperText Markup Language
НТТР	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IMS	Inventory Management System
IRA	Inventory Record Accuracy
JAD	Joint Application Development
JSON	JavaScript Object Notation
JWT	JSON Web Token
KEGA	Key Economic Growth Activities
MVC	Model-View-Controller
PHP	Hypertext Preprocessor
RAD	Rapid Application Development
SDLC	Software Development Life Cycle

SKU	Stock Keeping Unit
TDD	Test-Driven Development
TOC	Theory of Constraints
UAT	User Acceptance Testing
UI	User Interface
WBS	Work Breakdown Structure
WMS	Warehouse Management System

## LIST OF APPENDICES

APPENDIX A: Questionnaire Form	213
APPENDIX B: User Acceptance Testing Form	222
APPENDIX C: User Acceptance Testing Results	234
APPENDIX D: User Satisfaction Survey Form	294

#### **CHAPTER 1**

#### **INTRODUCTION**

#### 1.1 Introduction

Manufacturing companies process raw materials into finished goods, which will then be sold into the market. To meet production needs, warehouses often act as an important role to store all the raw materials before being processed into final products. This is important in the supply chain to reduce transportation costs, achieve economies of scale in terms of producing or purchasing, provide value-added processes, and lastly shorten the response time (Harjono, 2019).

To keep up with the customer's demand on time, keeping track and managing inventories are mostly important for easier and smoother operational activity and operation cycle. By keeping the inventories organised and tracked, inefficient work such as halting business operations to recount all stocks in the warehouses yearly to make sure the inventory count in the system is the same as the real inventory count can be avoided. Besides, it is also necessary to have a proper inventory system control as it may affect the production process if inventories were not restocked in time and have an impact on the financial statements of the company (Harjono, 2019).

Shared Prosperity Vision 2030, which was announced by the Malaysian government on 5th October 2019, aims to provide a decent standard of living to Malaysians by 2030. In this Shared Prosperity Vision, it proposed seven strategic thrusts, one of which is the Key Economic Growth Activities (KEGA), which consists of activities which demonstrate the potential to achieve Malaysia's aspiration to reach high-level economic development. Among the total of 15 KEGAs, KEGA 14 which is the advanced and modern services stated that the services sector, as the largest sector in Malaysia, must be modernised and diversified to be in line with the current demand of the market and advancement of technology (Ministry of Economic Affairs, 2019). This project is able to fulfil this activity by digitising the inventory recording process and making it to be in real-time so that inventory data is consistent across multiple locations to support the business operations to be in line with current market demand. This warehouse inventory management system can also reduce the workload of scheduling for cycle counting by automating the analysis and scheduling process.

This project is initiated to analyse the issues faced in inventory management of warehouses and propose solutions regarding issues found. This chapter will be discussing the background of the problem, the problem statements, project objectives, proposed solution, proposed approach as well as the project scope.

#### **1.2 Background of the Problem**

As the world population rises from day to day, demand for items and foods increases. Hence, the supply chain and manufacturing companies have to keep pace with this situation. Warehouses, the "middleman" of supply chain and manufacturing operations, also became greatly important in today's competitive market environment.

Although there are many types of warehouses such as raw materials warehouses, distribution warehouses, etc, most of them have a similar pattern of material flow. The warehouses need to deal with daily operations and transactions such as receiving goods, putting away, order picking, accumulating, sorting, packing, cross-docking, shipping, and internal replenishment (Ramaa, Subramanya, Rangaswamy, 2012). The performance of a warehouse and the accuracy of every transaction data dramatically affect the efficiency and accuracy of the whole supply chain.

To maintain the inventory accuracy in the warehouses, physical inventory counts were performed, usually once or twice a year, to count all inventories in the warehouses before they compile their annual financial reports. This solution works for companies with only a minimal number of inventories. Large companies with thousands of items normally need to halt their operations for a week or more for a full physical inventory count. This will cause them to provide poor customer service and some discrepancies due to internal or temporary staff not recording some inventory correctly or classifying items incorrectly, which cannot be avoided (Jenkins, 2020).

On the other hand, cycle counting solves the issues raised on physical inventory counts as it counts small, preselected sections of inventory periodically. It can save a lot of time while improving inventory accuracy. Besides, by doing cycle counting, companies no longer need to shut down their operations for a few days to perform a full inventory count. Hence, cycle counting has become a popular inventory management strategy for companies across all industries due to its benefits that included higher accuracy on inventory assessments, fewer errors, zero costs for employee overtime, and early detection of thefts in a timelier manner (Schwarz, 2021). Nevertheless, the accuracy of the cycle counting is affected by the inventory counts since cycle counting needs to match the actual count with the inventory counts to calculate the inventory record accuracy (IRA). Besides, the schedule of cycle counting should also be managed wisely.

#### **1.3** Problem Statement

This section discusses the three problem statements noticed from the problem formulation phase, which include the inaccuracy of inventory records, difficulties in scheduling cycle counting progress in the most efficient way and limitations in the SKU classification method.

#### 1.3.1 Inaccuracy of inventory record

The inventory records in the warehouse system should be real-time and accurate, which can keep track of the items coming in and going out (Curley, 2015).

In the current competitive market environment, manufacturing businesses have to move quickly, which means that the inventory count should be updated in realtime, and the availability of inventory counts should be at all times. If failed to do so, there will be discrepancies between the inventory count in the ERP system and the actual item count (Wisys, n.d.). When the warehouses have to deal with multiple locations, outstanding transactions, and paperwork lags, the inventory count might not be updated in real-time, if manual recordings were used. As a result, false variances will occur, and the inventory count will be inaccurate (Schwarz, 2021). This will cause a lot of problems such as being unable to replenish stocks at the right time or carrying too many stocks which increase the warehouse' expenses. Inventory shortage can then lead to a loss of revenue due to the inability to fulfil orders (NC-Vision, 2021).

Hence, a warehouse system that is able to keep track of the inventory stocks in real time is needed. For instance, the system from different locations should be able to record the name of items, amounts, and required details when the items are received and shipped out on one end.

# **1.3.2** Difficulties in scheduling cycle counting progress in the most efficient way

Ideally, the cycle counting process should be able to cover all SKUs needed to be counted within the time interval specified.

However, extra manpower would be needed to classify the items into respective counting groups by following the specified documented process and arranging the cycle counting based on their category, counting frequency, and time interval if done manually.

An auto-scheduler should be used to schedule the cycle counting process based on settings entered by the warehouse manager.

#### **1.3.3** Limitation in SKU classification methods.

The SKU classification method implemented should be able to define the importance of each inventory in a wider paradigm, so that more classifiers are considered to classify the SKU.

There are several types of inventory management techniques such as ABC analysis and VED analysis, being the most common inventory control methods among all. Each technique has its limitation in classifying the SKU for cycle counting. For instance, ABC analysis only classifies the SKU based on their usage value or stock value (Drakeley, 2021). It classifies inventories into three categories:

- 1. Category A is the fastest-moving and most expensive product. The threshold can be determined by the organisation. The most common threshold is the top 20% of the inventory list by velocity.
- 2. Category C is usually the bottom 15% to 20% of the item list by velocity.
- 3. Other inventories that did not fall into categories A and C will be in category B (Curley, 2017).

Stock keeping units in A class will be counted more often, followed by B class and finally C class. For example, all SKUs in A class are counted daily, followed by B class to be counted weekly and C class to be counted monthly. The cycle counting frequency of each class can be defined by the organization.

However, some of the inventories may not hold to this paradigm. For example, critical medicine such as an inhaler or snake venom vaccine should be given higher priority when comes to cycle counting. This is to ensure that vital drugs that are lifesaving are always present in the pharmacy or hospital. VED analysis, on the other hand, fits this purpose in classifying the inventories into vital (V), essential (E) or desirable (D) based on their criticality, unfortunately, with the limitation of not

considering the stock value of the inventories. In other words, pure ABC analysis has the limitation of VED analysis's strength and vice versa.

In fact, there is a matrix analysis of ABC-VED, which can satisfy both purposes. Instead of categorising them into A, B, C or V, E, D, ABC-VED analysis cross-tabulates both together and come up with new classes namely I, II and III (Ceylan and Bulkan, 2022). Cross tabulation is meant to overcome the limitation of both ABC and VED analysis by considering all possible combinations of A, B, C and V, E, D, ie., AV, AE, AD, BV, BE, BD, CV, CE, and CD before further classify them into the new classes I, II and III. Nevertheless, ABC-VED analysis is not in which the classification of groups I, II and III only considered the nine cross-tabulated classes. This is mainly due to the classification of the inventories' criticality has only considered three classes: V, E, and D, which limited the resultant number of cross-tabulated classes for a wider range of criticality assessments. In practice, different inventories can be assigned with different criticality scores (more than three) where a change in an inventories' operational context might cause its criticality value to increase or decrease (English and Yunusa-Kaltungo, 2022).

According to ISO 55000, "a critical asset, can be safety-critical, environmentcritical, or performance-critical, is one that has the potential to significantly impact the achievement of the organization's objectives". As a result, it is appealing that a more quantitative way of classifying inventories that considers a broader range of inventories by their criticality (not restricted to only V, E, and D) be established.

#### **1.4 Project Objectives**

The objectives to be achieved in this project are

- 1. To develop an application to record all in-and-out of the warehouse inventories from different locations in real-time
- 2. To develop an automated scheduler which can categorise SKUs, create schedules, and allocate them to cycle counting personnel.
- To propose and implement an improvement over existing SKU classification to achieve a higher degree of completeness and efficiency.

#### **1.5 Project Solution**

To solve the problems addressed above, a web-based warehouse inventory management system was proposed. The targeted users will be the admin, warehouse manager, and warehouse staff. Besides, this system will be able to update inventory in real-time to avoid inconsistencies or false variances, with an auto-scheduler that schedules cycle counting based on the enhanced ABC-VED cycle counting method.

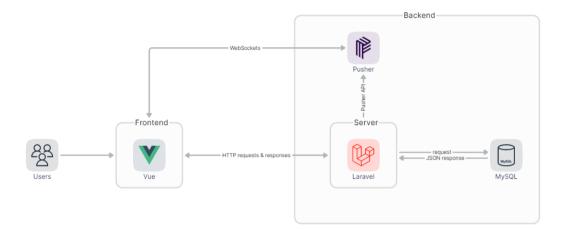


Figure 1.1: System overview

Figure 1.1 illustrates the high-level system overview for this project. The frontend development will be using Vue.js, whereas the backend development will be using Laravel with MySQL database. Once an HTTP request has been made, the server will send the event through Pusher API to the Pusher server and Pusher will push to clients via implementing WebSocket. This provides the ability of this warehouse inventory management system to update data in real time, which means that clients from different sites or locations will be able to view the updates of the inventory stocks without having to refresh the page.

#### 1.6 Project Approach

The proposed system development methodology for this project is the Phased-Development Based Rapid Application Development (RAD).

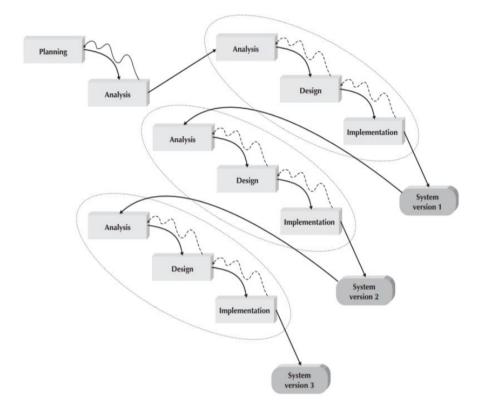


Figure 1.2: Phased Development Based Methodology (Dennis, Wixom and Tegarden, 2015)

This methodology divides the whole system into several versions of the system which will be developed sequentially. The overall system concept will be determined in the analysis phase, and the requirements can be categorised into several versions. The most important requirements can be prioritised in the first version and proceed to the design and implementation phase which only includes the requirements for the first version. After version 1 is implemented, another round of analysis, design and implementation can be performed for version 2. This process iterates until the fully functional system has been developed. As a result, a system with useful functions can be developed quickly (Dennis, Wixom, and Tegarden, 2015).

For this project, the system can be divided into three versions in which the first version would be the modules that are the most important. This involves the setting up of databases, preparing sample data and authentication as they are important for later development. Besides, the first module will also be included in this version, which is the real-time checking in and out stock module. The second version will include the auto-scheduler module as they are the main objectives of the project. The third version will be developing the report management which will include the daily report, cycle counting approval report and cycle counting summary report. All CRUD of users, inventories, warehouses and categories will also be done in this version.

## 1.7 Project Scope

This project develops a simple web-based warehouse management system, which focuses on inventory management. This web application can be accessed by the admin, warehouse manager, and staff. The responsibilities and actions for each role are listed in Table 1.1. This warehouse management system involves three main modules, which are inventory stock tracking, auto-scheduler, and report management.

Table 1:1: Roles and Responsibilities for Warehouse Inventory Management System

Roles	Actions/ Responsibilities
Admin	<ul> <li>Add, delete, update and view users, and set roles for them</li> <li>Add, delete, update and view inventories of all warehouses</li> <li>Add, delete, update and view a warehouse.</li> <li>Add, delete, update and view a category.</li> <li>View daily reports of in-and-out of inventories from all warehouses</li> <li>View the summary report for cycle counting of all warehouses</li> </ul>
Warehouse manager	<ul> <li>View all inventories in the system</li> <li>Start a cycle counting process</li> <li>View all upcoming cycle counting SKUs and reassign staff to replace the original staff responsible, if necessary</li> <li>View current cycle counting settings</li> <li>Approve or reject cycle count pending approval report</li> <li>View daily reports of in-and-out of inventories from own warehouse</li> <li>View the summary report for cycle counting for own warehouse</li> <li>View all staff in the warehouse</li> </ul>
Staff	<ul> <li>Check-in and out stock</li> <li>View the cycle counting schedule assigned to them</li> <li>Perform cycle counting and submit cycle count report</li> <li>View all inventories in the system</li> </ul>

#### 1.7.1 Inventory Stock Tracking

The first module is the inventory stock tracking functionality in which users can record the in-and-out operations. For example, when a distributor arrives and takes inventories from the warehouse, the staff in charge counts the inventories taken and updates the system by searching the inventory IDs or names and entering the amount taken. These changes will be updated in real-time to the database. Other staff in the warehouse will also have the newest updates in their system without the need to refresh the page.

#### 1.7.2 Cycle Counting Auto Scheduler

Secondly, this system provides an auto-scheduler which will be able to automatically generate schedules and assign them to staff based on settings entered by the manager. The classification of SKUs will be using the proposed classification method that improves the ABC-VED classification method with higher completeness and efficiency. Using the count frequency of each class and workday calendar of the warehouse entered by the warehouse manager, the auto-scheduler can determine which and how many inventories need to be counted daily and the auto-scheduler will auto-assign SKUs to staff that was assigned to cycle counting evenly. Then, the staff can select an SKU to start performing the cycle count and enter the actual count into the system. After the staff has performed cycle counting, the system will generate a cycle count pending approval report for the manager to approve and update the count to the system.

#### 1.7.3 Report Management

Thirdly, this system auto-compiles the entries entered by cycle counting staff and generates the cycle count pending approval report, which will calculate the variances with the recorded inventory count with the date counted. The warehouse manager can approve the adjustment of the actual inventory count or reject the count in which the staff will need to perform the cycle counting to that SKU again. If the manager approves the approval report, the cycle counting summary will be generated with the inventory record accuracy (IRA). Warehouse managers and admin can view the summary reports and sort the list of summary reports by ascending order of inventory record accuracy (IRA) to find out the lowest IRA and define the root cause of low

accuracy and try to solve the issue. The system also generates daily reports for warehouse managers and admin to view the transactions carried out each day.

#### **CHAPTER 2**

#### LITERATURE REVIEW

#### 2.1 Introduction

Warehouse inventory management systems can be complicated system as it involves a lot of processes and holds an important role in business operations. Hence, a literature review was conducted to better understand areas that are related to the proposed idea of this project. This chapter aims to:

- 1. Review similar projects on warehouse inventory management system
- 2. Understands cycle counting classification methods
- 3. Learns to perform cycle counting scheduling
- 4. Identify a suitable development methodology for this project

#### 2.2 Similar Projects Review

#### 2.2.1 Challenges of existing system

James (2016) stated that the old system used by Smart shoppers' in Masaska faced an inefficiency in record tracking because they mainly rely on paperwork to record the products and stock information. As a result, office spaces were required to compile those manual files, which also increases the cost of the business. A lot of time was also wasted retrieving and manipulating the data. Tahir (2020) also added that a lot of time was wasted using hard copy documentation in Walid Halal to search for particular records, and data duplication is common with an update and delete error. Besides, he also reminded that the manual ledger system adopted by Walid Halal also faced the challenges of wear and tear of ledger books which leads to loss of data.

# 2.2.2 Features of inventory management system

James (2016), Maharjan and Humagain (2016), and Tahir (2020) have developed an inventory management system to solve various problems that occur due to paperwork and manual files. Their projects were reviewed and for easier illustration, the features developed are tabulated in Table 2.1.

Table 2:1:Summary of features developed by James (2016), Maharjan and<br/>Humagain (2016), and Tahir (2020)

Sources	James (2016)	Maharjan and Humagain (2016)	Tahir (2020)
Features			
Register customer	$\checkmark$		
Create warehouse		$\checkmark$	
Stock CRUD	$\checkmark$		$\checkmark$
Category CRUD	$\checkmark$		
Inventory CRUD	$\checkmark$		$\checkmark$
Record sales to customer	$\checkmark$	$\checkmark$	
Record purchases from supplier		√	
Record sales and purchases return		✓	
View reports		$\checkmark$	
Backup and restore data		✓	

# 2.2.3 Significance and Limitation

The system developed benefits in a way that it increases the efficiency of business operations. James (2016) and Tahir (2020) save Smart shoppers' expenses on purchasing paper and ink to document products and stock data. Besides, they also reduced the fraud that may be caused by paper-based recording, by developing an inventory management system to keep track of inventories neatly. Customers of Smart

shoppers and Walid Halal Spices also benefit in a way that they are better served due to faster information access using Inventory Management System (IMS). Maharjan and Humaigan (2016), on the other hand, enables organisations with more than one warehouse to use the IMS for easier inventory management. His system also provides reports for daily sales and purchases which ease management decisions.

However, there were some limitations in the projects mentioned above, in which Maharjan and Humaigan (2016) stated that the application developed is not applicable for warehouses that have multiple levels of roles such as admin, warehouse manager and normal staff, meaning that no authorization was done to grant access of actions to specific roles in the IMS. Besides, the system developed is not applicable for businesses that have large quantities of inventories. This is due to the lack of searching features as mentioned by Tahir (2020).

# 2.3 Cycle Counting

The cycle counting method is a kind of inventory control method in a way that it fixes the mismatch of inventory counts recorded in IMS and the actual count of the stock keeping units (SKUs) in the warehouse. It can happen anytime depending on the business operations. The main advantage of cycle counting compared to the traditional physical inventory is that cycle counting will not need to shut down their business to count all items in warehouses (Shen, 2017). Fathoni, Ridwan and Santosa (2019) also listed some of the benefits of using cycle counting in which fewer experienced and knowledgeable people were needed for this task. This helps in a way that experienced and expert people can be allocated to other crucial parts of the business operations. Besides, they also confirmed that cycle counting improves the process by reducing inaccuracy of inventory records by tracking and fixing discrepancies between records and actual counts.

In cycle counting, a good and stable sample for cycle counting is critical as a sample that is too small cannot represent the entire population and a large sample may defeat the purpose of sampling. However, no matter how accurate the cycle counting was performed and how perfect the samples were, errors in the cycle counting process will always occur and these errors are important to define the root cause of the existing inventory problems instead of just simply correcting the inventory errors whenever errors occur as it will not help in a long run (Shen, 2017). Hence, the calculation of inventory record accuracy (IRA) is important to ease management decisions to correct

the existing problems that cause inventory accuracy. Efforts should always be done to achieve a high IRA.

### 2.3.1 Traditional ABC analysis

Shen (2017) researched the four most commonly used cycle counting techniques, which include control group cycle counting, random sample cycle counting, ABC cycle counting and process control cycle counting. Among all of the four techniques, the ABC cycle counting and process control cycle counting focus more on warehouses and distribution centres whereas the rest of the two techniques are more on store-level cycle counting.

Shen (2017) and Mahtamtama, Ridwan and Santosa (2018) stated that ABC analysis's major approach is to assign SKUs by classifying the SKUs based on the 80/20 rule, which means that 80% of the cumulative revenue or stock value in the warehouse is from 20% of the SKUs. SKUs will be classified into three classes, in which SKUs that fall into 80% of the revenue or stock value will be in class A, 85%-95% in class B and the rest of it will be in class C. The cycle counting frequency of SKUs in class A will be the highest, followed by class B. Class C will be counted least frequently. Figure 2.1 shows the general concepts of classification of ABC analysis.

Mahtamtama, Ridwan and Santosa (2018) researched the calculation steps for stock value to classify the SKU. Firstly, the stock value should be counted by multiplying the unit cost of each product or inventory with the quantity on hand. Secondly, in order to categorise the SKUs, the stock value needs to be converted to a range of 0 to 1. In order to do this, all stock values for all SKUs should be calculated first and the maximum and minimum stock value will be used to calculate the transformed stock value. The calculation formula is as follows:

$$= \frac{\text{SKU stock value} - \text{Min stock value}}{\text{Max stock value} - \text{Min stock value}}$$

After the transformed stock value has been calculated, the SKUs can be classified using the ratio mentioned above. The cycle counting frequency adopted by PT XYZ is in Table 2.2. However, the cycle counting frequency varies between organisations and it is decided by the management.

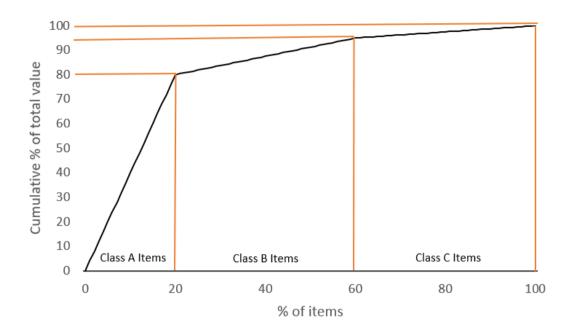


Figure 2.1: General concepts of ABC cycle counting method

Table 2:2: Counting Frequency Adopted by PT XYZ

(Mahtamtama, I	Ridwan	and S	Santosa,	2018)
----------------	--------	-------	----------	-------

ABC class	Cycle counting frequency
А	Weekly
В	Monthly
С	Quarterly

### 2.3.2 ABC-VED

In some cases, the ABC analysis alone is not sufficient especially when the business inventories have criticality. For example, drugs which are critical for life but not having high stock value as the disease may be not common but deadly if without medicine control.

Fathoni, Ridwan and Santosa (2019) developed a new stock-taking policy, which is the ABC- VED cycle counting, with the purpose of improving the record accuracy of a pharmaceutical company. VED analysis classifies the SKUs into three groups, which are vital (V), essential (E) and Desirable (D).

#### 2.3.2.1 Classification of SKUs

The classification is based on the criticality of the inventories. In ABC-VED analysis, the drugs are classified into 9 subclasses, each class was labelled by two alphabets: the first one indicates ABC analysis while the second one indicates VED analysis. The 9 subclasses will then be categorised into 3 groups: I, II, and III. The classes in ABC-VED analysis are in Table 2.3 (Fathoni, Ridwan and Santosa, 2019).

### Table 2:3: ABC-VED classification

(Fathoni, Ridwan and Santosa, 2019)

Groups	Subclasses
Ι	AV, AE, AD, BV, CV
II	BE, BD, CE
III	CD

To prove the effectiveness and efficiency of the new stocking policy, Fathoni, Ridwan and Santosa (2019) analysed the classification of using ABC analysis alone, VED analysis alone, and ABC-VED analysis, which are tabulated in Table 2.4.

# Table 2:4: Analysis of cycle counting policies

(Fathoni, Ridwan and Santosa, 2019)

Cycle counting policy	Classification Result	Limitation	Summary
ABC analysis	(79.97% of total sales) B: 146 items (2.72%	attention to the vital items that include: B: 146 items	given the highest priority as they are
VED analysis	V + E: 74.37% of total sales	Ignores 170 items in A class (17.97% of total sales)	-

		which is in the desirable (D) category	considered because they are in class D
ABC-VED analysis	I: 1064 items (33.1% of total sales) II: 1256 items (14.2% of total sales) III: 893 items (2.4% of total sales)	None or not mentioned	None or not mentioned

## 2.3.2.2 Strength and Limitation

In a nutshell, the ABC analysis has the strength of taking monetary aspects of the inventories into classification, whereas the VED analysis includes the criticality of the products to classify the SKUs. Hence, pure ABC analysis has the limitation of not taking the criticality of inventories into account and the same goes for pure VED analysis in which it is not taking the monetary value of the inventories such as sales value, stock value and so on into consideration. The ABC-VED analysis, on the other hand, solves both limitations of ABC and VED analysis by combining them and reclassifying them into three categories (I, II and III).

# 2.3.3 Cycle Counting Scheduling

Some research was done on cycle counting scheduling for the purpose of learning how to schedule the cycle counting. Scheduling using the ABC-VED classification approach and buffer-time management approach was covered in this section.

# 2.3.3.1 ABC-VED Scheduling

To ease the cycle counting process, Fathoni, Ridwan and Santosa (2019) included the generation of schedules of cycle counting in the inventory control system with the ABC-VED method. They calculated and tabulated the daily count for each group I, II and III in Table 2.5.

Group	Items	Frequency	Frequency (in a year)	Daily Count
Ι	1064	Monthly	12	35
II	1256	Every 2 months	6	21
III	893	Every 10 months	1.2	3
Total	3213		19.2	63

Table 2:5: Daily counts for Group I, II, and III

(Fathoni, Ridwan and Santosa, 2019)

The calculation of the daily count is done by dividing the total items in each category by the time frame given to count all items for that category. For example, by referring to Table 2.5, the group I has a total of 1064 items and all items need to be counted monthly (in 30 days). Hence, the calculation steps are as follows:

Daily count = 
$$\frac{\text{Total items}}{\text{frequency}}$$
, (2.2)  
Daily count =  $\frac{1064}{30 \text{ days}}$ ,  
Daily count = 35.4

(means 35 SKUs need to be counted daily for Group I)

After the daily count has been calculated, the cycle counting schedule can be generated based on the business operation. There are a few aspects that need to be considered in scheduling which are: how many shifts in a day, the working days of the warehouse and how many workers are assigned to perform cycle counting in each shift. For example, in a scenario of 3 shifts and 2 workers in each shift, the total daily count of 63 can be divided equally into 20 items per shift, and assign 10 items per worker in the shift. Figure 2.2 shows the UI of the cycle counting schedule developed.

Group		Group II	🗞 Group I					
Show :	o • entr	ries				Search:		
ID II	↓† Code	ltem ↓† Name	.∐ Quantity	∬ Unit	↓† Category	Storage 🗐 Number	Schedule	Action
123	A00342	Acetin 600 eff	1	pcs	RS	11	2018-01-05	Edit
124	A00353	Kawa Breast Pump 120 ml	1	pcs	ME	16	2018-01-09	Edit
125	A0037	Abate 1 gr	3	pcs	HE	26	2018-01-13	Edit
130	A00395	Puremed Thermometer digital	5	pcs	ME	15	2018-01-07	Edit

Figure 2.2: Cycle counting Schedule UI

(Fathoni, Ridwan and Santosa, 2019)

#### 2.3.3.2 Scheduling Using Buffer Time Management

There is a method to protect the schedule of cycle counting, which is by using the buffer time management. According to Reyes, Alvarez. and Vasquez (2016), the Theory of Constraints (TOC) is an approach to identify constraints or target levels to improve business performance. Mahtamtama, Ridwan and Santosa (2018) summarised that TOC is basically a philosophy for scheduling and inventory controlling which proposes interdependent links between departments, functions, and resources and weak links between them may cause the potential for better performances to be hidden. Hence, TOC aims to improve those weak links by finding bottlenecks in the system. Often, TOC buffers will be placed before the controlled constraints so that the constraints will never be in shortfall. It is stated by Reyes, Alvarez and Vasquez (2016) that TOC buffers are divided into 5 zones in which 3 of them are the main zones and the rest of the 2 are additional zones for further analysis. Figure 2.3 shows the buffer zones and penetration rates.



Figure 2.3: TOC Buffer zones and penetration rates of each zone (Reyes, Alvarez. and Vasquez, 2016)

Mahtamtama, Ridwan and Santosa (2018) adopted the buffer time management which defined buffer time as the time between the last date cycle counting was performed and the deadline for the next cycle count. Figure 2.4 is the illustration of the buffer time management. The formula for calculating the percentage of remaining buffer time is as follows:

% of remaining buffer time = 
$$\frac{\text{Buffer time left in days}}{\text{Total buffer time}} * 100\%$$
 (2.3)

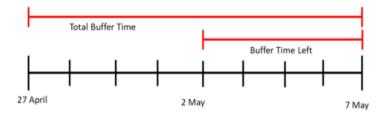


Figure 2.4: Illustration of buffer time management (Mahtamtama, Ridwan and Santosa, 2018)

Mahtamtama, Ridwan and Santosa (2018) developed the conceptual model of their research as in Figure 2.5, and the data inputs needed to conduct the research are tabulated in Table 2.6. Figure 2.5 shows the problem-solving schematics developed by Mahtamtama, Ridwan and Santosa (2018) which explains the process flow of cycle counting scheduling.

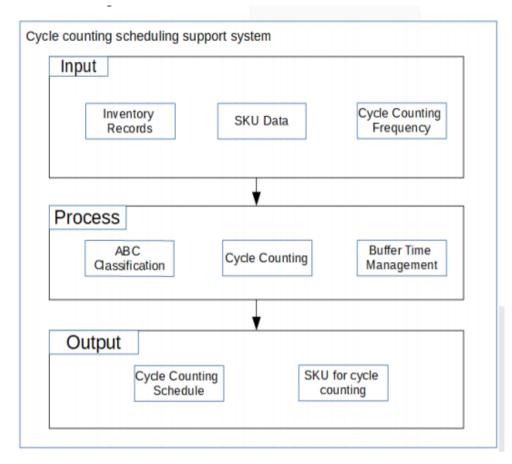
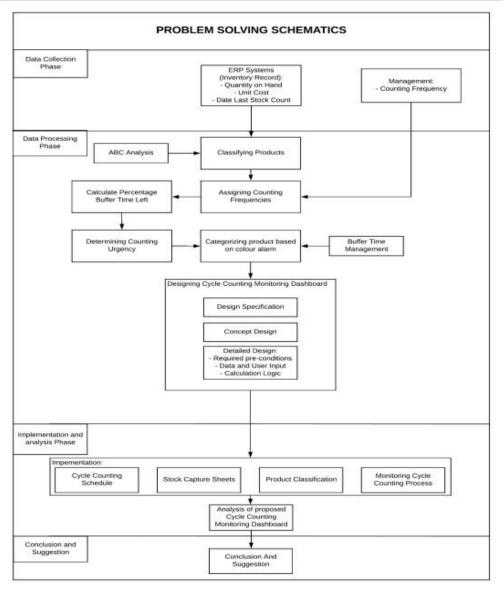


Figure 2.5: Conceptual model of cycle counting scheduling support system (Mahtamtama, Ridwan and Santosa, 2018)

## Table 2:6: Necessary data for each input

Inputs	Necessary Data
SKU data	Item category, item code and unit cost
Inventory records	Last stock count data and number of items
Cycle counting frequency	Determined by management

(Mahtamtama, Ridwan and Santosa, 2018)



## Figure 2.6: Problem solving schematics

(Mahtamtama, Ridwan and Santosa, 2018)

According to Figure 2.6, after the SKUs have been classified using ABC classification, counting frequencies for each class can be assigned like Figure 2.7 to

calculate the next stock count date. Then, the calculation of the percentage of buffer time left can be done and classify the percentage of buffer time left by counting urgency. Figure 2.8 shows the colour coding of each counting frequency class. Lastly, the final categorization result is shown in Figure 2.9. The information in Figure 2.9 provides information on which time to be counted first.

Category	Stock Code	ABC Class	Count Frequency
Cocoa Butter	DF 200_25	С	Quarterly
Cocoa Butter	DF 100	В	Monthly
Cocoa Powder	DF 700- 11M_600	A	Weekly
Cocoa Mass	DF 300_20	А	Weekly
Cocoa Mass	DF 3200	А	Weekly

Figure 2.7: Counting frequency for each class

(Mahtamtama, Ridwan and Santosa, 2018)

Counting Urgency	Colour Code
Low	Green
Medium	Yellow
High	Red
Critical	Black

Figure 2.8: Colour coding of counting urgency class

(Mahtamtama, Ridwan and Santosa, 2018)

Category	Stock Code	Count Frequency	Days Between Count	Date Last Stock Count	Stock Count Due Date	Days Left	%Buffer Time Left	Count Urgency
Cocoa Powder	DF 705-11	Weekly	7	7-Jan-18	14-Jan-18	7	100.00%	Low
Cocoa Butter	DF 102	Weekly	7	1-Jan-18	8-Jan-18	1	14.29%	High
Cocoa Mass	DF 304_20	Weekly	7	1-Jan-18	8-Jan-18	1	14.29%	High
Cocoa Powder	DF 700-11BR	Weekly	7	4-Jan-18	11-Jan-18	4	57.14%	Medium
Cocoa Powder	DF 700-11M_600	Weekly	7	7-Jan-18	14-Jan-18	7	100.00%	Low
Cocoa Mass	DF 300_20	Weekly	7	1-Jan-18	8-Jan-18	1	14.29%	High
Cocoa Mass	DF 3200	Weekly	7	4-Jan-18	11-Jan-18	4	57.14%	Medium
Cocoa Powder	DF 720-11	Weekly	7	1-Jan-18	8-Jan-18	1	14.29%	High

Figure 2.9: Final categorization results

(Mahtamtama, Ridwan and Santosa, 2018)

Date	Summarized	<b>Counting Frequ</b>	encies											
07/01/2018 5	Category	Stock Code	Location	Quantity On Hand	ABC Class	Date Last Stock Count	DaysLeft	Count Urgency	% Buller Time Left	Print	*	(° Op	C Strech	@ 2009.
2123	Cocoa Butter	DF 200_25	RA01801	442	C	January 7, 2018	91	Low	100.00%					
m ///S	Cocoa Butter	DF 100	RA01802	358	8	January 7, 2018	30	Low	100.00%					
100 100	Cocce Powder	DF 700-11M_600	RA01803	553	- A :	January 7, 2018	7	Low	100.00%					
1127	Cocoa Mass	DF 300_20	RA01804	119	A	January 1, 2018	1	High	14.29%					
	Cocoa Mass	DF 3200	RA01805	104	A	January 4, 2018	.4	Medium	37,1915					
al O	Cocce Powder	DF 720-11	R401806	332	A	January 1, 2018	1	High	14.29%					
1000	Cocca Powder	DF 705-11	RA01807	263	A	January 7, 2018	7	Low	100.00%					
Select Al	Cocoa Butter	DF 102	RB04401	46	A	January 1, 2018	1	High	14.29%					
IN Print	Cocoa Mass	DF 304_20	RB04402	59	A	January 1, 2018	1	High	14.29%					
	Cocoa Powder	DF 700-118R	RB04403	243	A	January 4, 2018	4	Medium	37.14%					
	Cocoa Mass	DF 3005 20	RB04404	45	A	January 4, 2018	4	Medium	57.14%					
	Cocua Mass	DF 3100LT_20	RB04405	45	A	January 1, 2018	1	High	14.29%					
	Cocce Powder	DF 680-118R	RB04406	174	A	January 4, 2018	4	Medium	57,1451					
	Cocca Powder	DF 700-11M 425	RB04407	174	A	January 7, 2018	7	Low	100.00%					
	Cocce Powder	DF 760-11	RB04408	174	A	January 1, 2018	1	High	14.29%					
	Cócoa Butter	DF 200_15	RB04409	32	A	January 4, 2018	4	Medium	57.54%					Summarized Counting Frequencies
	Cocce Powder	PD 4013	RK02342	168	A	January 1, 2018	1	High	14,7985					Summarized Counting Preducticles
	Cocoa Mass	C.MAS5	R802343	39	A	January 1, 2018	1	High	14,29%			208 -		
	Cocoa Mass	DF 400 20	RK02344	30	A	January 6, 2018	6	Low	85.71%			- 801 -		
	Cocoa Powder	DF 700-11MS 25	RK02345	113	A	January 7, 2018	7	Low	100.00%			1.00		
	Cocce Powder	DF 780-11	RK02346	94	A	January 1, 2018	1	High	14.29%			100		
	Cocce Powder	WM A-000-T	RK02347	84	- 4	January 7, 2018	7	Low	100.00%					1000-000
	Cocca Powder	DF	RX02348	41	A	January 6, 2018	6	Low	85.71%			400		_
	Cocce Powder	DF 580-11	RK02340	38	Å	January 4, 2018	4	Medium	107.000					
	Cocoa Butter	OF 200UT_25	PL05221	6	A	January 6, 2018	6	Low	85.71%			46		
	Cocce Powder	DF 506-11	PL05222	30	A.	January 4, 2018	4	Medium	57.14%					A second s
	Cocce Powder	DF 760-11BR 25	PL05223	25	A	January 7, 2018	7	Low	100.00%				100	Halas By Time
	Cocce Powder	DF 500-11	PL05224	22	A	latuary 4, 2018	4	Medium	57.1034	1				
	Cocos Powder	PD 301107_25	PL05225	17	A	January 1, 2018	1	High	14,29%					
	Cocos Powder	OF BGY-11 25TPF	PL05226	16	A.	January 1, 2018	1.1	High	14.29%					
	Cocce Powder	DF 700-11M 25	PL05227	15	A.	January 4, 2018	4	Medium	57.10%					
	Cocce Powder	WM N-000-T_25T	PC01365	12	A	January 1, 2018	1	High	14.20%					
	Cocca Powder	DF 740-11	PC01366	11	Â	January 7, 2018	2	Low	100.00%					
	Cocos Powder	PO 3011 UT	PC01367	10	A	January 7, 2018	7	Low	100.00%					
	Cocce Powder	PD 3011	PC01368	10	Å	January 7, 2018	7	Low	100.00%	-				
	Cocce Powder	DF 700-115UT 25	PC01368	3	- 2	January 1, 2018	1	High	14,29%					
	Corce Powder	DF 86Y-11	TA03312		- 2	January 6, 2018		Low	85.71%					
	Course Dounder	DF 031-11	TA00312			January 6, 2018		Low	1.1.1.100		-			

Figure 2.10: Layout of monitoring dashboard

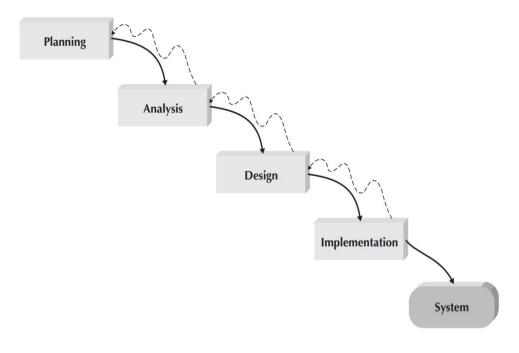
(Mahtamtama, Ridwan and Santosa, 2018)

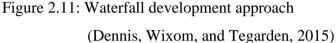
# 2.4 Software Development Methodologies

Software development methodologies are different and unique based on the order and focus of each SDLC phase. There are many categories of system development methodologies such as structure design, rapid application development (RAD) and agile development.

# 2.4.1 Waterfall Method

In the waterfall approach, the key deliverables are very long for each phase and approval from the project sponsor is needed to end the phase and move to the next phase. Although it is possible to go backwards in the SDLC, it is extremely difficult in the waterfall method. The advantage of the waterfall method is that it defines the system requirements long before the coding starts and changes in the requirements during the project progress are lowered. The disadvantage of this methodology is that the design of the system must be completely specified before coding starts and a lot of time is required to deliver the system from the completion of the analysis phase (Dennis, Wixom, and Tegarden, 2015).





# 2.4.2 Rapid Application Development (RAD)

Rapid application development (RAD) divides software designs into four phases, which are requirement planning, user design, construction phase and cutover phase. This methodology solves the disadvantages of the waterfall method mentioned above by adjusting the SDLC phases to deliver some part of the system faster to the project sponsor or client (Dennis, Wixom, and Tegarden, 2015). Sasmito, Wibowo and Dairoh (2020) proved this statement by developing a web-based geographic information system (GIS) using RAD and compared to the older existing system that was developed by the waterfall method. The results showed that the GIS was developed within 84 days using RAD, which is 5 weeks faster than the system developed using the waterfall approach, and with better quality. This is done by the nature of RAD which focuses on the development process and shortens the planning process. Shaydulin and Sybrandt (2017) added that RAD also ensures the incorporation of user feedback into the system earlier. However, they also noticed that RAD might cause poor design because developers seek short-term functionality without paying attention to the technical debt.

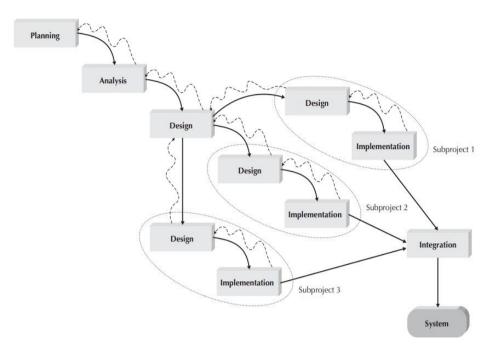


Figure 2.12: RAD approach (Dennis, Wixom, and Tegarden, 2015)

#### 2.4.2.1 Phased Development Approach

The phased development-based methodology is a kind of RAD, in which it breaks the whole system into several versions and develops it sequentially. The analysis phase will first identify the whole system, project team, users and system sponsors, then, the requirements will be divided into several versions. The most important and basic requirements will be included in the first version. This is supported by the case study of the phased implementation of the Industry 4.0 Learning Factory between 2015 to 2020 done by F Leal, Zancul and Schützer (2021), in which the development of Learning Factories was divided into three phases in which the first phase (2015-2017) is the concept design and pilot which covers the development of the limited demonstration space with students, graduate and the partner companies' support. In the second phase (2018-2019), the structure, equipment and physical space were expanded and the didactic application and research results were achieved. In the third phase (2020), the sustainable operational model was built with the purpose of strengthening relationships with the industry.

The advantage of this methodology is similar to RAD's in a way that it gets a useful system to the hands of users quicker (Dennis, Wixom, and Tegarden, 2015). F Leal, Zancul and Schützer (2021) found out that the phased implementation is beneficial to Learning Factories because its design and implementation are resourceintensive and complex. However, this methodology also has disadvantages in which the users will begin to use the system that is incomplete, hence, the selection of important requirements for each version especially the first version is crucial (Dennis, Wixom, and Tegarden, 2015).

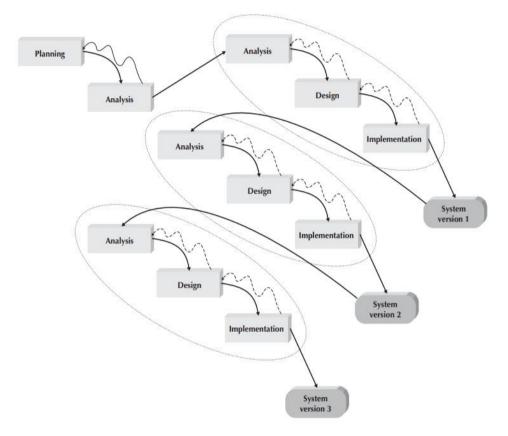
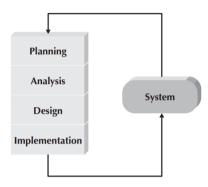


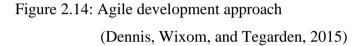
Figure 2.13: Phased development-based approach (Dennis, Wixom, and Tegarden, 2015)

# 2.4.3 Agile Development Methodology

Agile development streamlines the software development process by eliminating the modelling and documentation overhead as well as time spent on these tasks. Instead, it emphasises simple and interactive application development. There are a few criticisms of agile development in which it is unrealistic if the development team requires co-location. Secondly, the development process can devolve into a prototyping approach if the agile development is not carefully managed. Thirdly, lack of actual documentation also will cause auditability issues in the system in which the system development and the development process cannot be assured. The last criticism

of agile is based on whether agile development can deliver large systems that are mission-critical (Dennis, Wixom, and Tegarden, 2015).





# 2.4.4 Choosing a methodology

The survey conducted by Vijayasarathy, et al. (2016 cited in Shaydulin and Sybrandt, 2017) on 153 developers to describe their software development process listed the seven most commonly used methodology which includes waterfall, agile unified process (AUP), scrum, test-driven development (TDD), rapid application development (RAD), joint application development (JAD) and lastly feature-driven development (FDD). In order to evaluate these seven methodologies, Shaydulin and Sybrandt (2017) defined a set of criteria which evaluates on quality and agility of the methodology as shown in Figure 2.14. On the other hand, Dennis, Wixom, and Tegarden (2015) also tabulated the comparison of software development methodologies which includes structured design methodologies (waterfall and parallel), RAD (phased, prototyping, throwaway prototyping) and agile methodologies (XP and SCRUM) in Figure 2.15.

				Meth	odology			
		Waterfall	AUP	Scrum	TDD	RAD	JAD	FDD
	Requirements flexibility	No	Yes	Yes	Yes	Yes	Yes	No
g	Requirements fulfillment guarantee	Yes	Yes	Yes	No <sup>1</sup>	Yes	No	Yes
eri	Cost estimation	Yes	Yes	Yes	No	Yes	No	Yes
Criteria	Cost estimates refinement	No	Yes	Yes	No	Yes	No	Yes
	Validation	Yes	Yes <sup>2</sup>	Yes <sup>3</sup>	Yes	Yes	Yes	Yes
Quality	Quick validation	No	Yes <sup>2</sup>	Yes <sup>3</sup>	Yes	Yes	Yes	Yes
Jue	Focus on customer	No	Yes <sup>4</sup>	Yes	No	Yes	Yes	No
U.	Understandability guarantee	Yes $^{5}$	No	No	No	No	Yes <sup>6</sup>	No
	Technical debt control	Yes	No	No	Yes	No	No	No
6	Prioritizes added value	No	Yes	Yes	Yes	Yes	Yes	Yes
eri	Allows partial requirements	No	Yes	Yes	Yes	Yes	Yes	Yes
ţ.	Focuses on small teams	No	Yes <sup>7</sup>	Yes	Yes	Yes	Yes	Yes
õ	Develops minimal viable architecture	No	Yes	Yes	Yes	Yes	Yes	Yes
Agility Criteria	Produces minimal documentation	No	Yes	Yes	Yes	Yes	No	Yes
Agi	Relies heavily on customer feedback	No	Yes	Yes	No	Yes	Yes	No
ł	Susceptible to unforeseen risks	No	Yes	Yes	Yes	No	Yes	Yes

Figure 2.15: Methodologies evaluation results

(Shaydulin and Sybrandt, 2017)

		ctured odologies	RAD Meth	nodologies			gile dologies
Ability to Develop Systems	Waterfall	Parallel	Phased	Prototyping	Throwaway Prototyping	ХР	SCRUM
With Unclear User Requirements	Poor	Poor	Good	Excellent	Excellent	Excellent	Excellent
With Unfamiliar Technology	Poor	Poor	Good	Poor	Excellent	Good	Good
That Are Complex	Good	Good	Good	Poor	Excellent	Good	Good
That Are Reliable	Good	Good	Good	Poor	Excellent	Excellent	Excellent
With a Short Time Schedule	Poor	Good	Excellent	Excellent	Good	Excellent	Excellent
With Schedule Visibility	Poor	Poor	Excellent	Excellent	Good	Excellent	Excellent

Figure 2.16: Criteria in selecting a development methodology

(Dennis, Wixom, and Tegarden, 2015)

# 2.5 Conclusion

In conclusion, this literature review covers the similar project review which includes three similar systems on warehouse inventory management systems and findings on the reviewed projects' challenges, features, significance and limitations were done. It was found that paperwork causes inefficiency in record tracking and data duplication and adaptation of an automated inventory management system can reduce the fraud and expenses used in buying paper and ink. Nevertheless, there were some limitations in the project reviews in which the system is not applicable for warehouses with multiple roles and it is also not suitable for large quantities of inventories due to lacking of searching features.

A review on ABC analysis was done to understand the calculation steps for classifying SKUs and ABC-VED analysis was also included to deal with warehouses that have inventories with criticality. Besides, ABC-VED analysis was proven to be more effective and efficient compared to ABC and VED analysis alone and it covers more SKUs. Scheduling using ABC-VED is different compared to scheduling using buffer time management in a way that ABC-VED calculates the number of SKUs to be counted daily in each category and assigns them to workers who perform cycle counting evenly, whereas buffer time management colour-code the SKUs based on counting urgency and prioritise the counting for SKUs that have highest counting urgency. As the outcome of the literature review in this section, the ABC-VED analysis will be adopted in this project as it is proven to have higher efficiency compared to pure ABC and VED analysis. Additionally, this method will be further improved during implementation by using numerical values to replace the V, E and D classes so that the criticality is no longer restricted to three classes and the classification of SKU will not be limited to nine subclasses as illustrated in Table 2.3. The system can directly assign cycle counting schedules to workers which enables cycle counting to be performed in a more "scheduled manner" instead of following the counting urgency in buffer time management which can cause confusion if more than one worker was assigned to perform cycle counting. Besides, counting all SKUs within the counting frequency (eg. in one month) may not be achieved as the number of SKUs to be counted daily in each category is not fixed in buffer time management.

Last but not least, software development methodologies which include the waterfall model, RAD and agile methods were studied. Further study was done on RAD and phased development which led to a conclusion that the phased development-based approach is the most suitable development methodology for the project.

#### **CHAPTER 3**

### METHODOLOGY AND WORK PLAN

# 3.1 Introduction

This chapter will be discussing the methodology and work plan for this project. The adopted development methodology is a phased development-based methodology which will include three phases in which the details of each phase are outlined in this chapter. Besides, the work breakdown structure (WBS) and the Gantt chart are developed for project scheduling. Last but not least, the development tools are selected and described.

# 3.2 Phased Development-Based Methodology

The phased development-based methodology is chosen as the development methodology of this project. This methodology breaks the development of software into several sets of phases, each handling different aspects of software development. There are five phases in this methodology which are planning, analysis, design, implementation and testing. In this project, the whole system will be divided into three phases as shown in Figure 3.1 in which the first phase includes features with the highest priority. This is to make sure that the main functions of the system are developed first and leave out minor features first. The second phase of development and testing will only be able to start if the development and testing in the first phase have been completed. After the third phase has been done, the whole system has been completed and the closing phase will be carried out for documentation of the project.

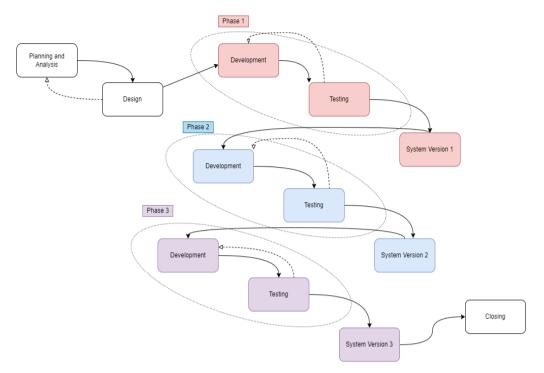


Figure 3.1: Phased development-based approach

# 3.2.1 Planning and Analysis

This project starts with the planning phase and is followed by the analysis phase. The planning phase will include the identification of problems, objectives and scope, whereas analysis will involve requirement gathering, similar project review and selection of appropriate development tools and development methodology. WBS, which is the key deliverable for project scheduling, will be developed in the planning phase after a sufficient understanding of problems, objectives and project scope.

# 3.2.1.1 Identify Problems, Project Objectives and Scope

The first task of this project proposal is to identify the problems regarding warehouse inventory management without a warehouse inventory management system. By reviewing the business operations of warehouses, some problems were identified:

- 1. Inaccuracy of inventory record
- 2. Difficulties in scheduling the cycle counting progress in the most efficient way
- 3. Limitation in SKU classification methods

The project objectives were then identified with the purpose of solving each of the problem statements identified above. The objectives are listed below:

- 1. To develop an application to record all in-and-out of the warehouse inventories from different locations in real-time.
- 2. To develop an automated scheduler which can categorise SKUs, create schedules, and allocate them to cycle counting personnel.
- 3. To propose and implement an improvement over existing SKU classification to achieve a higher degree of completeness and efficiency.

The objectives declared above will be the main goals to achieve for this project. With objectives in place, the project scope was also determined in which this system will have three main roles, which are admin, warehouse manager, and warehouse staff. Each role will have different actions to perform using the system. The project scope also covers basic info on what the system will be able to do in three aspects that are in line with the three objectives which include inventory tracking, cycle counting auto-scheduler, and report management.

# 3.2.1.2 Requirement Gathering and Elicitation

To gather requirements on the warehouse inventory management system, a questionnaire was designed and distributed to targeted users to collect opinions from them. The collected responses will be converted into the functional requirements of this system. Besides, similar projects were also reviewed to analyse their features and limitations so that they can be improved for this project.

#### 3.2.1.2.1 Questionnaire

The questionnaire was designed using a Google Form and distributed to 20 targeted users who work in a warehouse and understand the basic operations in a warehouse. The responses cover age from 21 to 60 years old and 70% of them is male respondents.

There are three sections after the demographic information collection. The first section collects some general information on the respondents' warehouse. The results show that the majority of the respondents already have a warehouse management system in place, and most of them have multiple warehouses at different locations.

The second section is to collect respondents' opinions on the inventory accuracy. Surprisingly, only half of the respondents use a warehouse management system, while the rest of them still implement Excel or paper recording. Besides, most of the respondents agreed that there are discrepancies between the inventory record and the actual count. Most of them also supported that the inventory count is inconsistent across multiple sites and the inventory data accuracy will affect their business operation in a way that extra time is consumed to investigate the discrepancies and correct the data.

The third section collects respondents' thoughts on the physical count and cycle counting. From the responses collected, there are several significant findings in which more than 90% of the respondents supported that SKU, cycle counting, and calculation of the inventory record accuracy are hard to be done manually. Besides, they also agreed that dealing with physical reports is troublesome.

#### **3.2.1.2.2 Similar Project Review**

3 projects on the warehouse inventory management system were reviewed and features developed in the projects reviewed were tabulated in Appendix A. There are no common features that all three systems acquire, however, there were common limitations on the three systems which the system developed did not contain cycle counting features. Besides, there were no searching features in place, which makes the system inconvenient to use if the quantity of inventories becomes large. Moreover, there was no authorization in place for different roles, meaning that no control of access to specific actions was in place. Hence, cycle counting management, the ability to search through inventories and authorization should be included as part of the functional requirements in this project.

Besides limitations, the projects reviewed also proved that implementation of a warehouse inventory system has increased the efficiency of business operations by reducing the frauds that are caused by the paper recording and by gaining faster information access. Besides, the expenses of business were also reduced by eliminating the use of paper and ink.

# 3.2.1.2.3 Project Scheduling

Project scheduling is also one of the key deliverables of the planning phase. For this phase, a work breakdown structure (WBS) will be developed using a top-down

approach, meaning that high-level tasks can be identified first, and broken down into smaller subtasks. A WBS will contain information such as the duration of the task and the dependency of the task. Task dependencies exist when one task cannot be started or another task has yet to complete. A WBS basically represents all of the small milestones that a project will need to achieve.

Gantt charts will also be developed in this project to present the tasks of WBS in a graphical way. Since this project will be developed using a phased development approach, the Gantt chart will also divide the tasks of development and testing into three phases. Gantt chart helps to illustrate the timeline of the project tasks so that the status of a task can be easily determined eg. ahead of schedule or behind schedule.

#### 3.2.2 Design

After the analysis of the scope has been done, the design phase starts by determining the software architecture as shown in Figure 3.2. A use case diagram will be created to model the system behaviour and identify the interactions between the system and its actors as this project consists of three different roles that can perform different actions. Next, an entity relationship diagram will also be included in the design phase to show relationships between entities of the system to support the set-up of the database in the implementation phase.

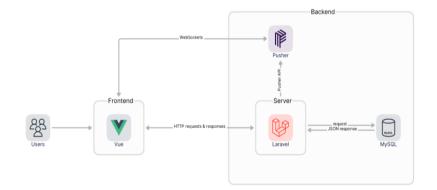


Figure 3.2: System architecture

A prototype is also developed to illustrate the approximate user interface design of the warehouse inventory management system. Throughout the development

of wireframes of the prototype, necessary data columns needed are better visualised. Besides, the developed wireframe also assists in the development and enhancement of the use case description as well. The wireframes will be used as guidance throughout the actual development of the system. All wireframes created for the modules will be included in Chapter 4.

#### 3.2.3 Development and Testing

The development phase starts after the design phase has been completed. In this phase, the development team will start to build the warehouse inventory management system with the support of diagrams provided in the design phase. After the development phase is done, unit testing will be carried out to test out the features developed. In this project, there will be three phases that consist of a development and testing process and each of the phases will contribute to different parts of the system. Upon completion of the three phases, several testing will be carried out to test out the final build.

#### 3.2.3.1 Phase 1

In phase 1, the repository will be set up with Laravel and Vue.js framework. The database will also be set up in which all tables will be created and some dummy data will be inserted into each table for easier testing purposes for later development. Besides, the authentication of users will also be done in this phase so that the authorization of actions can be implemented in later development. This phase will consist of the first project objective for this warehouse inventory management system which is to develop a module that allows checking in and out stock in real-time. Unit testing will be carried out, followed by integration testing to make sure that each module is working fine with the other.

#### 3.2.3.2 Phase 2

After the basic features mentioned in phase 1 have been completed and tested, this phase will include the auto-scheduler for cycle counting, which is the largest module of this system that covers the second and third project objectives. Unit testing and system integration testing will be carried out to make sure that features developed in this phase are working fine and cope well with those developed in phase 1.

### 3.2.3.3 Phase 3

When all of the features are completed, phase 3 will be completing the reports for checking in and out stock and cycle counting since both of the module has been done in the previous phase. All CRUDs will also be done in this phase. The unit testing and system integration tests must be passed before moving to the closing phase.

## 3.2.4 Closing

The closing phase will be the phase in which documentation will be done. The documentation will provide a detailed description of the developed system, with snapshots attached. Conclusion with the objectives achievement and limitations of this project will also be included in the documentation.

# 3.3 Project Plan

This section depicts the whole project plan with a detailed work breakdown structure (WBS) and a Gantt chart.

# 3.3.1 WBS

# 1.0 Planning and analysis

- 1.1 Identify problems
  - 1.1.1 Identify background of problems
  - 1.1.2 Develop problem statements
- 1.2 Identify project objectives
  - 1.2.1 Draft out project objectives
  - 1.2.2 Refine project objectives with SMART metrics
- 1.3 Propose solution
  - 1.3.1 Develop system architecture
- 1.4 Decide project approach
  - 1.4.1 Decide research approach
  - 1.4.2 Decide development approach
- 1.5 Develop project scope
  - 1.5.1 Identify target user
    - 1.5.1.1 Decide on user roles
    - 1.5.1.2 Define each roles' responsibilities
  - 1.5.2 Define system scope
    - 1.5.2.1 Define inventory tracking scope
    - 1.5.2.2 Define cycle counting auto scheduler scope
    - 1.5.2.3 Define report management scope
- 1.6 Literature review
  - 1.6.1 Similar project review
    - 1.6.1.1 Review related projects
    - 1.6.1.2 Tabulate features development in each project
    - 1.6.1.3 Identify significance and limitation
  - 1.6.2 Review on cycle counting
    - 1.6.2.1 Cycle counting classification review
      - 1.6.2.1.1 Traditional ABC analysis
      - 1.6.2.1.2 ABC-VED

1.6.2.1.3 Identify strengths and limitation

- 1.6.2.2 Cycle counting scheduling
  - 1.6.2.2.1 ABC-VED scheduling
  - 1.6.2.2.2 Buffer- time management scheduling
- 1.6.3 Software development methodologies review
  - 1.6.3.1 Waterfall method
  - 1.6.3.2 Rapid application development (RAD)
  - 1.6.3.3 Agile development methodology
  - 1.6.3.4 Comparison of methodologies
- 1.7 Requirement gathering and elicitation
  - 1.7.1 Questionnaire
    - 1.7.1.1 Structure questionnaire questions
    - 1.7.1.2 Gather responses
    - 1.7.1.3 Develop facts finding
      - 1.7.1.3.1 Analyse and interpret data collected
  - 1.7.2 Review related projects
    - 1.7.2.1 Identify common features
    - 1.7.2.2 Identify significance of project
    - 1.7.2.3 Identify limitations
  - 1.7.3 Develop requirements specification
    - 1.7.3.1 Develop functional requirements
    - 1.7.3.2 Develop non-functional requirements
- 1.8 Project planning
  - 1.8.1 Create WBS
  - 1.8.2 Create Gantt chart
- 1.9 Decide development tools

# 2.0 Design

- 2.1 Develop system architecture
- 2.2 Develop use case diagram
- 2.3 Develop use case diagram description
- 2.4 Develop ERD diagram
- 2.5 Design prototype

# 3.0 Development and Testing Phase 1

3.1 Setup project

3.1.2 Set up MySQL database

- 3.1.3 Create tables with dummy data
- 3.1.4 Connect application to database
- 3.2 Authentication and authorization of users

3.2.1 Define roles

- 3.2.2 Define guards, gates, and policies for each role
- 3.2.3 Develop login module
- 3.3 Develop inventory tracking module
  - 3.3.1 Develop user interface
  - 3.3.2 Develop algorithm to check in and out inventories
  - 3.3.3 Implement Pusher API
- 3.4 Testing
  - 3.4.1 Unit testing
  - 3.4.2 Integration testing

# 4.0 Development and Testing Phase 2

- 4.1 Develop cycle counting auto scheduler
  - 4.1.1 Develop user interface
  - 4.1.2 Develop scheduling algorithm
    - 4.1.2.1 Retrieve necessary information from database
    - 4.1.2.2 Classify SKUs
    - 4.1.2.3 Schedule SKUs
    - 4.1.2.4 Assign SKUs to responsible staff
- 4.2 Testing
  - 4.2.1 Unit testing
  - 4.2.2 Integration testing

### 5.0 Development and Testing Phase 3

- 5.1 Develop report generation
  - 5.1.1 Develop cycle count report
    - 5.1.1.1 Develop pending approval report
    - 5.1.1.2 Develop summary report
  - 5.1.2 Develop daily report
- 5.2 Develop CRUD of users
- 5.3 Develop CRUD of inventories

- 5.4 Develop CRUD of warehouses
- 5.5 Develop CRUD of categories
- 5.6 Testing
  - 5.6.1 Unit testing
  - 5.6.2 Integration testing

# 6.0 Closing

- 6.1 Conduct web service testing
- 6.2 Conduct real time check in/out module testing
- 6.3 Conduct user acceptance testing (UAT) & usability testing
  - 6.3.1 Prepare test cases
  - 6.3.2 Prepare user satisfaction survey
  - 6.3.2 Decide UAT participants
  - 6.3.3 Conduct UAT
  - 6.3.4 Collect UAT results
  - 6.3.5 Collect and calculate average SUS score
- 6.4 Design poster
- 6.5 Finalise project documentation
- 6.6 Prepare presentation slide

# 3.3.2 Gantt chart

#### 3.3.2.1 Planning and analysis

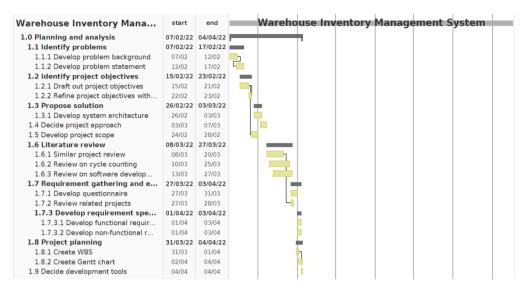


Figure 3.3: Planning and analysis phase timeline

### 3.3.2.2 Design

2.0 Design	04/04/22	21/04/22
2.1 Develop system architecture	04/04	04/04
2.2 Develop use case diagram	04/04	04/04
2.3 Develop use case diagram descri	05/04	06/04
2.4 Develop ERD diagram	07/04	08/04
2.5 Design prototype	04/04	21/04

Figure 3.4: Design phase timeline

# 3.3.2.3 Development and testing phase 1

3.0 Development and Testing Phas	22/04/22	24/05/22
3.1 Setup project	22/04/22	27/04/22
3.2 Authentication and authoriza	28/04/22	03/05/22
3.2.1 Define roles	28/04	29/04
3.2.2 Define guards, gates, and pol	29/04	30/04
3.2.3 Develop login module	01/05	03/05
3.3 Develop inventory tracking	04/05/22	19/05/22
3.3.1 Develop user interface	04/05	06/05
3.3.2 Develop algorithm to check in	06/05	12/05
3.3.3 Implement Pusher API	11/05	19/05
3.4 Testing	20/05/22	24/05/22
3.4.1 Unit testing	20/05	23/05
3.4.2 Integration testing	23/05	24/05

Figure 3.5: Development and testing phase 1 timeline

# 3.3.2.4 Development and testing phase 2

4.0 Development and Testing Phas	25/05/22	27/06/22
4.1 Develop cycle counting auto	25/05/22	22/06/22
4.1.1 Develop user interface	25/05	02/06
4.1.2 Develop scheduling algorithm	03/06	22/06
4.2 Testing	23/06/22	27/06/22
4.2.1 Unit testing	23/06	25/06
4.2.2 Integration testing	26/06	27/06

Figure 3.6: Development and testing phase 2 timeline

3.3.2.5 Development and testing phase 3

5.0 Development and Testing Phas	28/06/22	29/07/22
5.1 Develop report generation	28/06/22	09/07/22
5.1.1 Develop cycle count repo	28/06/22	05/07/22
5.1.1.1 Develop pending approv	28/06	01/07
5.1.1.2 Develop summary report	02/07	05/07
5.1.2 Develop daily report generati	06/07	09/07
5.2 Develop CRUD of users	10/07	12/07
5.3 Develop CRUD of inventories	13/07	15/07
5.4 Develop CRUD of warehouses	16/07	18/07
5.5 Develop CRUD of categories	18/07	20/07
5.6 Testing	22/07/22	29/07/22
5.6.1 Unit testing	22/07	25/07
5.6.2 Integration testing	26/07	29/07

Figure 3.7: Development and testing phase 3 timeline

3.3.2.6 Closing

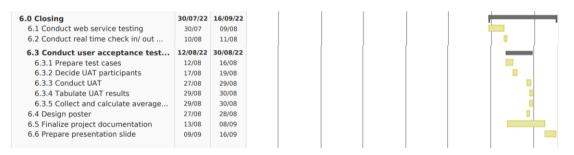


Figure 3.8: Closing phase timeline

# **3.4 Development Tools**

This section outlines all development tools required in developing the implemented system, which includes Visual Studio Code as the IDE tool, Axure RP 10 as the prototyping tool, Vue.js and Laravel for the frontend and backend framework, MySQL database to store data and lastly, GitHub and the SourceTree for the version control of the project.

#### 3.4.1 Visual Studio Code

Visual studio code is an IDE tool that will support the development of this project. This is because Visual Studio Code supports a great number of packages and extensions which will ease the coding process. For example, PHP Intelephense is an extension which is full of essential features that support productive PHP development. Besides, extensions for Vue.js and Laravel which support syntax highlighting features are also crucial to provide faster detection of syntax errors. Besides useful extensions, visual studio code also provides version control features in which conflicts can be solved efficiently and changes made in previous commits can be reverted easily. Last but not least, the file searching feature of visual studio code using Ctrl + P also saves a lot of time and provides convenience to locate a file instead of browsing through the repository for one file.

#### 3.4.2 Axure RP 10

In order to develop the prototype of the warehouse inventory management system, Axure RP 10 is selected to create the high-fidelity prototype as interaction events, conditional logic, working forms and multistate containers are available to be applied to the prototype. Besides, Axure RP is easy to use as the wireframes can be created by using drag-and-drop only.

#### 3.4.3 Vue.js

Vue.js is a progressive JavaScript framework that provides component-based programming which enables user interfaces to be developed efficiently. This framework is lightweight and easy to pick up because no comprehensive understanding of JSX or TypeScript is needed, instead, only basic knowledge of HTML, CSS and JavaScript will be needed to be able to pick up Vue.js. Besides, this framework works perfectly with Laravel.

#### 3.4.4 Laravel

For backend development, Laravel is selected as the PHP framework which uses an MVC design pattern. One of the reasons for choosing a PHP framework rather than the JavaScript framework like Node JS for backend is that the warehouse inventory management system is mainly on data retrieval and manipulation, hence, the PHP framework would be more suitable rather than the JavaScript framework or Pythonbased backend.

# 3.4.5 MySQL

MySQL is an open-source, free-to-use, relational database which facilitates the management of databases effectively. It has multiple advantages such as it is stable, and reliable in terms of data security. Besides, it is open source and compatible with a wide range of systems, database models and programming languages.

# 3.4.6 GitHub

GitHub will be used as the code repository hosting platform for version control of this project. Pushing changes or commits to GitHub periodically is important to account for accidents such as laptop breaking down, project file loss, or malware-attack. Besides, using GitHub, changes from old versions can be reverted if the latest version is buggy.

## 3.4.7 SourceTree

SourceTree is a free-to-use desktop client that simplifies the interactions with Git repositories. For example, pulling changes or pushing changes to the server can be done in only a few clicks. Besides, all codes in each commit on any branch can be visualised clearly to keep track of the changes. By using SourceTree, we can decide on which line, hunk or file to be committed to the server by staging or discarding it. Most importantly, by eliminating the use of git commands, we can focus more on the coding part.

# 3.5 Conclusion

In conclusion, the phased development methodology is chosen as the software development approach for this project and tasks for each phase are specified. In this

section, a work breakdown structure and Gantt chart are created to visualise the timeline of the whole project. The project is starting from 7th February 2022 to 16th September 2022, with a total duration of 221 days. Last but not least, Visual Studio Code, Axure RP9, Vue.js, Laravel, MySQL, GitHub and SourceTree are chosen as the development tools for this project.

## **CHAPTER 4**

### **PROJECT SPECIFICATION**

# 4.1 Introduction

After the requirement gathering and elicitation, the initial specification is determined in this chapter. First of all, facts findings on the questionnaire responses are developed. Then, a use case diagram and use case description are prepared to illustrate how actors interact with the warehouse inventory management system. An entity-relationship diagram (ERD) is also created to show the relationships between entities of the system. Lastly, a prototype is developed to draft the user interfaces of the system.

# 4.2 Facts Finding

A questionnaire was distributed to targeted users who work or understand the operations of warehouses to understand more about the software requirements and a total of 20 responses were collected. There are a total of 4 sections which include demographic information, general information, inventory accuracy and the physical count or cycle counting.

### 4.2.1 Demographic Information

The first section is to collect demographic information from the respondents. The responses cover ages ranging from 21 to 60 years old. Figure 4.1 shows that most of the respondents are in the range of 41 to 50 years old, which contributes to 40% of the total respondents, followed by 31 to 40 years old, which covers 35% of total respondents. The gender distribution of the respondents of the questionnaire is 70% of male respondents and 30% of female respondents, as shown in Figure 4.2.

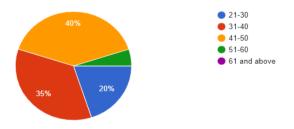


Figure 4.1: Age of 20 respondents

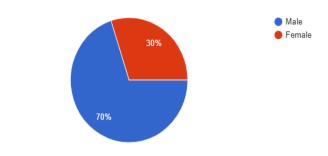


Figure 4.2: Gender of the respondents

#### 4.2.2 General Information

The second section of the questionnaire aims to collect some general information regarding the warehouse management of the respondents. The first question is to ask whether the respondents' warehouse(s) has a warehouse management system already in place. Figure 4.3 shows that most of the respondents (70%) are already using a warehouse management system to manage their warehouse. However, there are still a significant number (30%) of respondents who have yet to implement a warehouse management system in their warehouse.

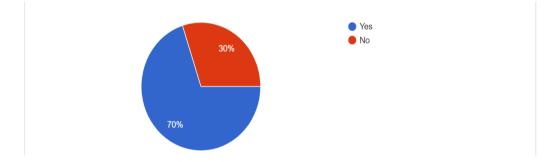


Figure 4.3: Statistic of respondents to have warehouse management system in place in their warehouse or not

The second question is asking whether the respondent has more than one warehouse from multiple locations and the result in Figure 4.4 shows that 65% of the respondents have multiple warehouses at different locations.

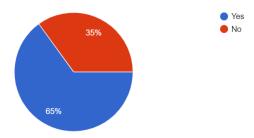


Figure 4.4: Distribution of respondents who has more than one warehouse from multiple locations or not

## 4.2.3 Inventory Accuracy

This section aims to collect information on the accuracy of inventories in the respondent's warehouse(s). The first question asks respondents about their method to record inventory and the results show that only half of the respondents use a warehouse management system or similar software to record their inventory, whereas another 40% of the respondents use Excel or online spreadsheets for inventory recordings and 10% of the respondents still have paper records, as shown in Figure 4.5.

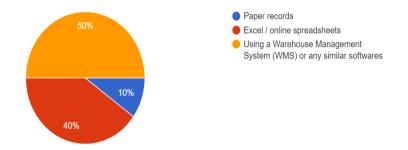


Figure 4.5: Methods of inventory recording of respondents

The second question asked whether there are any discrepancies between the inventory record and the actual count and the results are shown in Figure 4.6. Surprisingly, a high percentage of 75% of respondents responded that the inventory record does not match the actual count of the inventory.

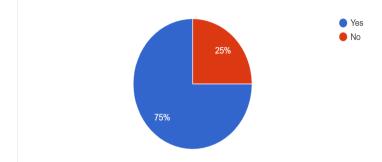


Figure 4.6: Statistic of whether any discrepancies occur between inventory record and actual count

The third question also asked if the inventory count is inconsistent across multiple locations. The result in Figure 4.7 shows that 80% of the respondents responded yes to this question, which means that most of the respondents are facing the issue of having the same inventory to have different counts recorded in the system from multiple sites.

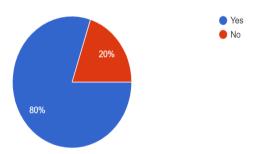


Figure 4.7: Statistic of whether any inconsistency of inventory count occurs across multiple location

The fourth question asked respondents their opinion on whether inventory data accuracy affects business operation. Figure 4.8 shows that 90% of the respondents agreed that data accuracy would affect their normal business operations. The fifth question further confirmed with the respondents why they think that it would affect them. 89.5% of the respondents agreed that inventory accuracy would affect business operation in a way that additional time would be needed to research the discrepancies and correct the data. The results are shown in Figure 4.9.

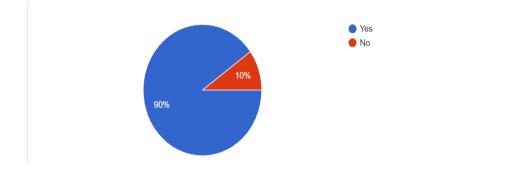


Figure 4.8: Statistic of whether inventory and data accuracy would affect the business operation

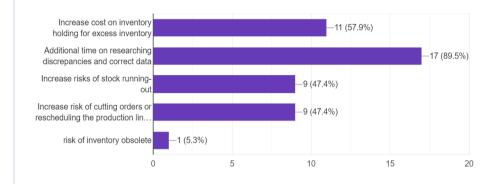


Figure 4.9: Possible reasons on why data accuracy would affect business operation

# 4.2.4 Physical Count / Cycle Counting

This is the last section of the questionnaire and it aims to collect the opinions of respondents on cycle counting and inventory controls. As the first question, the inventory control method implemented by respondents is collected. From Figure 4.10, it is shown that half of the respondents utilise both physical inventory count and cycle counting as their inventory control methods.

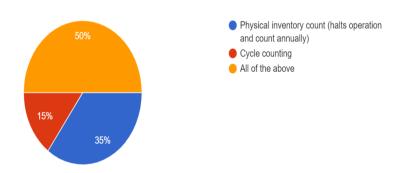


Figure 4.10: Distribution of inventory control methods implemented in respondents' warehouse(s)

From question 2 to question 9, respondents were given a statement in each question and were asked whether they agree or disagree on a scale of 1 to 5, in which 1 means strongly disagree and 5 is strongly agree. For easier illustration purposes, the responses for questions 2 to 9 are tabulated in Table 4.1.

No	Questions	1 (Strongl y disagre e)	2 (Strongly agree)	3 (Neutral)	4 (Agree)	5 (Strongly Agree)
1	Manual inventory recording/ counting without a WMS system is troublesome.	-	-	5%	40%	55%
2	It is difficult to classify SKUs manually.	-	-	10%	50%	40%
3	Extra manpower and resources will be needed to analyse and schedule for physical counting and cycle counting.	-	-	15%	30%	55%
4	It is difficult to keep track of the cycle counting progress manually.	-	-	5%	40%	55%
5	Continuous operations and incoming transactions while performing physical count/cycle count may cause false variances in inventory counts.	-	5%	-	35%	60%

Table 4:1: Summary of responses on physical counting or cycle counting

Table 4.1 (Continued)

6	There will be a lot of hassle if were to deal with all of the physical reports regarding physical counts/ cycle counting.	-	5%	5%	35%	55%
7	Adjusting inventory records should be in real-time to avoid inconsistencies.	-	-	5%	35%	60%
8	Extra effort will be needed to calculate inventory record accuracy (IRA) to keep track of the performance.	-	-	5%	45%	50%

From Table 4.1, the first question to the fourth question proved that most respondents agreed that doing inventory recording, classifying SKU, scheduling cycle counting, and keeping track of cycle counting progress manually is inefficient and troublesome. For the fifth question, 95% of the respondents agreed that continuous and incoming transaction while performing physical count or cycle counting will cause variances in inventory count. Question 6 showed that 90% of the respondents supported those physical reports management is inefficient. Question 7 evidenced that 95% of respondents want inventory records to be updated in real time. In the last question, 95% of respondents agreed that extra efforts are needed to calculate IRA.

#### 4.2.5 Summary of survey

From the survey, several important points on the requirements of the warehouse inventory management system were obtained from the respondents. Firstly, it is proven that most of the respondents are facing the issue of discrepancies between inventory records and actual count, as well as the discrepancies between inventory records across multiple locations and these issues affect their business operation in various ways as stated above. Hence, warehouse inventory management systems should be able to update the stock data in real time, and cycle counting features should be included. Secondly, respondents would like to have inventory recording, SKU classification, cycle counting scheduling, and cycle counting progress monitoring to be done automatically by the system. Thirdly, the warehouse inventory management system should be able to manage reports and calculate IRA automatically.

## 4.3 Requirement Specification

This section depicts the system requirement specification, which can be classified into two main categories which are functional requirements and non-functional requirements.

## 4.3.1 Functional Requirement

A system must adhere to the formalized functional requirements in order to suit the needs of the end user. All of these features must be included in the system as stipulated by the contract. These are depicted or stated as the system's required input, the action taken, and the desired result. System functional requirements are essentially the user-stated requirements that are evident in the finished product, as opposed to non-functional needs (Zhou, 2004). In this project, the functional requirements are collected from the similar projects review in the literature review sections and from the questionnaire responses for requirements elicitation in Section 4.2. The functional requirements are divided into three different roles: admin, warehouse manager and staff as shown in Table 4.2.

Roles	Functional Requirements
Admin	<ol> <li>The system should allow admin to view, add, update and delete the inventory of a warehouse.</li> <li>The system should allow admin to view, add, update, and delete a warehouse from the system.</li> <li>The system should allow admin to view, add, update and delete a user in the system.</li> <li>The system should allow admin to view, add, update and delete a category from the system.</li> <li>The system should allow admin to view daily reports of in- and-out inventories of all warehouses.</li> <li>The system should allow admin to view the summary report for cycle counting of all warehouses.</li> </ol>
Warehouse manager	<ol> <li>The system should allow managers to start a new cycle counting process.</li> <li>The system should allow managers to view all upcoming cycle counting SKUs.</li> <li>The system should allow managers to view current cycle counting settings.</li> </ol>

Table 4:2: Functional requirements by roles

	<ul> <li>10. The system should allow a manager to view daily reports of in-and-out inventories of his or her warehouse only.</li> <li>11. The system should allow managers to view summary reports for cycle counting of his or her warehouse only.</li> <li>12. The system should allow managers to view, approve or reject cycle count pending approval reports.</li> <li>13. The system should allow managers to view all staffs in the warehouse.</li> </ul>
Staff	<ul><li>14. The system should allow staff to check in-and-out of the inventory stock.</li><li>15. The system should allow staff to view cycle count schedules that are assigned to them.</li><li>16. The system should allow staff to perform cycle counting on the SKU assigned.</li></ul>
Manager and staff	17. The system should allow the manager and staff to view all inventories in the system.
Admin, manager and staff	18. The system should allow admin, manager and staff to login into their account.
System	<ul> <li>19. The system should calculate inventory record accuracy (IRA) of each SKU and generate summary reports for each cycle count input from staff.</li> <li>20. The system should classify SKUs, calculate the number of SKUs to be counted daily for each class, generate schedules and assign to staff available evenly.</li> </ul>

## 4.3.2 Non-Functional Requirement

The non-functional requirement describes a system performance feature. It encompasses all requirements that are not covered by functional requirements. They define criteria that determine the functionality of a system rather than specific behaviour (Chung, Nixon, Yu and Mylopoulos, 2012). The IEEE-Std 830 – 1993 (IEEE Computer Society, n.d.) lists 13 non-functional requirements to be included in a Software Requirements Document:

- 1. Performance requirements
- 2. Interface requirements
- 3. Operational requirements
- 4. Resource requirements
- 5. Verification requirements
- 6. Acceptance requirements
- 7. Documentation requirements
- 8. Security requirements
- 9. Portability requirements
- 10. Quality requirements
- 11. Reliability requirements
- 12. Maintainability requirements
- 13. Safety requirements

In this project, we focus on three main non-functional requirements, which are the performance (usability) requirement, security requirements and verification requirements.

#### 4.3.2.1 Performance-(usability) Requirement

1. The system should ask for confirmation for every destructive actions.

#### 4.3.2.2 Security Requirement

- The system should protect users' credentials in which passwords should be encrypted.
- 2. The system should not be accessed by an unauthorised third party.

## 4.3.2.3 Verification Requirement

1. The users of the warehouse inventory management system should authenticate themselves using username and password.

## 4.4 System Use Case

Use cases are a list of actions that define the interaction between the user (admin, warehouse manager and staff) and the implemented system. System use cases detailed specific processes inside the implemented system to reach users' goals. It is used in the analysis phase to identify, describe and clarify the functional requirements from the end users' perspective, including the dependencies among use cases as well. In this section, a use case diagram will be shown along with the use case description for each use case.

## 4.4.1 Use Case Diagram

Figure 4.11 shows the use case diagram of the warehouse inventory management system.

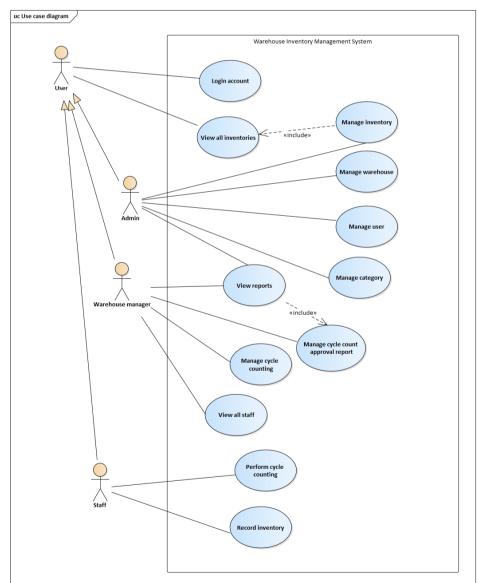


Figure 4.11: Use case diagram

# 4.4.2 Use Case Description

Table 4:3: Use case description of log	in account
ruble 1.5. Obe cuse description of log	in account

Use Case Name: Login account		ID: 1	Importance Level: High				
Primary Actor: User (admin, warehouse Use Case Type: Detail, essential manager, staff)							
Stakeholders and Interests: User - wants to login their account and access the system.							
Brief Description: This use case describes before accessing the system.	s how ı	isers logir	n their account				
Trigger: When user wants to access the syst	em.						
Relationships:         Association : User         Include : -         Extend : -         Generalization : -         Normal Flow of Events:         1. The user opens the application.         2. The user enters their username and p         3. The application verifies the username         4. The system displays the homepage of         Sub-flows:	e and pa	assword.					
<ul> <li>Alternate/Exceptional Flows:</li> <li>3a. If the username and password is incorrect prompt the user to re-enter again.</li> <li>4a. If users log in their account for the first to their password will be displayed.</li> <li>1. The system replaces the temporary pentered by the user.</li> </ul>	ime, a m	nessage pro	ompt for setting				

Use C	ase Nan	ne: Manage inventory		ID: 2	Importance Level: High				
Prima	Primary Actor: Admin Use Case Type: Detail, essential								
		and Interests: s to view, add, delete and upda	te the inv	entory of	a warehouse.				
	-	tion: This use case describes h ory of a warehouse.	ow admii	n view, a	dd, delete and				
		type of inventory arrives one s price changes or inventory i			•				
Assoc Includ Exten	le	: Admin : View inventories : -							
Norma 1. 2. 3. 4.	Admin Admin The sy of man	of Events: h login into the application. h chooses to manage inventory ystem executes view all invent hage inventory module. h wants to add, update or delet If admin wants to add ne warehouse The S-2: add inventory subfl If admin wants to update an in The S-3: update inventory su If admin wants to delete an in The S-4: delete inventory subfl	tories use e an inve ew inven ow is per inventory bflow is nventory	case as t ntory. tory to formed. performe	the homepage one or more d.				
1. 2. 3. 4. 5. S-3: U 1.	Add new Admin Syster Admin priorit Syster Syster Jpdate in Admin Syster	r inventory n wants to add new inventory. n displays a modal pop up to p n enters inventory name, starti y and warehouse name. n auto-generates id and adds th n shows inventory successfull nventory n selects an inventory to updat n displays update inventory m n edit the inventory name, cost	ng stock, he new inv y added r e from th odal pop-	cost per ventory to nessage. e invento up to pro	o the database. ry list. ompt input.				

Table 4:4: Use case description of manage inventory

4. System updates the inventory with new inputs by admin.

## S-4: Delete inventory

- 1. Admin selects an inventory to delete from the inventory list.
- 2. System displays delete inventory modal pop-up for confirmation.
- 3. Admin confirms to delete the inventory.
- 4. System soft-deletes the inventory and back to manage-inventory page.

Alternate/Exceptional Flows:

S-3, 4a. If the user changed the category of the inventory, the storage bin for the inventory will also be changed.

- 1. The system will check if the new category's storage bins are full or not.
- 2. If all bins for the new category is full, the system will ask the user to empty a bin for the new category first, else, the system will assign the inventory to the empty bin.
- 3. System will show an alert showing the new storage bin number.

Use Ca	ase Name: View all inventories		ID: 3	Importance Level: High				
	Primary Actor: User (Admin, warehouse Use Case Type: Detail, essential manager, staff)							
Stakeholders and Interests: User - wants to view all inventories of the warehouse.								
	Description: This use case describes aff view all inventories in the warehous		, wareho	use manager,				
Trigge	er: Admin, warehouse manager or sta	aff wants to	o view al	l inventories.				
Associ Includ Extend								
Norma	l Flow of Events:							
1.	The user logins into the application.							
2.	The user chooses to view all invento							
3.	System displays inventory list							
	3.1. If the user is admin, the syst warehouses.	tem display	s all inve	entories of all				
	3.2. If the user is a warehouse ma all inventories of their wareh		iff, the sy	stem displays				
4.	The user can select one of the inven		ew details	S.				
<ol> <li>The user can select one of the inventories to view details.</li> <li>The system returns inventory details which include the transaction history of that inventory.</li> </ol>								
Sub-fl	ows:							
	ate/Exceptional Flows:							
	Admin selects a particular warehouse		inventori	es.				
	Admin enters the desired warehouse							
2.	5	and returns	inventor	ies belonging				
	to the selected warehouse only.							

Table 4:5: Use case description of view all inventories

Use Ca	ase Nan	ne: Manage warehouse		ID: 4	Importance Level: High		
Primar	y Actor	: Admin	Use essenti	-	vpe: Detail,		
Stakeh	olders	and Interests:					
		s to add, view, update and dele	ete ware	house			
DriefI	Jacomine	ion. This was asso describes he	arri o dim	n view od	d undere and		
	wareho	ion: This use case describes he	ow adm	in view, ac	id, update and		
uelete	wateno	use.					
Trigge	r Adm	in wants to manage the warehouse	ouse				
111650		in wants to manage the waren					
Relatio	onships:						
Associ	ation	: Admin					
Includ	e	:-					
Extend	1	:-					
Genera	alization	n : -					
Norma	1 Flow	of Events:					
1.		lmin logins the application.					
1. 2.		lmin chooses to manage the w	arehous	e			
2. 3.		stem displays all warehouses.	archous	с.			
3. 4.	•	lmin can view, add, update or	delete a	warehous	0		
4.	4.1.	The admin wants to view a w					
	4.1.	S-1: View warehouse detail s					
	4.2.	The admin wants to add a ne		-	icu.		
	7.2.	S-2: Add new warehouse sub					
	4.3.	The admin wants to update th		-	•		
		S-3: Update a warehouse sub					
	4.4.	The admin wants to delete th		1	•		
		S-4: Delete a warehouse subf					
Sub-flo	ows:						
		rehouse detail					
		selects desired warehouse to	view fro	om the list			
		n displays the warehouse detai					
		warehouse					
1.	Admir	n wants to add a new warehous	se.				
2.	System	n displays a modal pop up to p	rompt i	nput.			
3.	•	n enters name, location, number	-	-	es.		
4.	System	n creates the storage bins and	add the	new warel	nouse into the		
	databa	se.					
S-3: U	pdate a	warehouse					
1.	Admir	n wants to update a warehouse					
2.	2. System displays a modal pop up to prompt input.						

Table 4:6: Use case description of manage warehouse

- 3. Admin edit the location or warehouse manager.
- 4. System updates the warehouse details.

S-4: Delete a warehouse

- 1. Admin wants to delete a warehouse.
- 2. System displays a confirmation modal to prompt for admin's confirmation.
- 3. Admin confirms to delete the warehouse.
- 4. System deletes the warehouse.

Alternate/Exceptional Flows:

Use Case Name: Manage user		ID: 5	Importance					
			Level: High					
	•		Level. High					
Primary Actor: Admin	Use Case Type: Detail,							
	essential	l						
Stakeholders and Interests:								
Admin - wants to view, add, update or delet	Admin - wants to view, add, update or delete a user.							
Brief Description: This use case shows ho	w admins	view a	dd undate or					
delete a user.		, view, a	uu, upuate of					
Trigger: Admin wants to view, add, update	or delete :	a user.						
inggen inalini wand to the w, add, apaale		<i>a asen</i>						
Relationships:								
Association : Admin								
Include : -								
Extend : -								
Generalization : -								
Normal Flow of Events:								
1. The admin logins the application.								
2. The admin rogins the appreation.	ser							
3. The system displays all users.								
4. The admin can view, add, update or	delete a u	lser.						
4.1. The admin wants to view a u								
S-1: View user detail subflo	w is perfor	rmed.						
4.2. The admin wants to add a ne	ew user.							
S-2: Add new user subflow	s perform	ed.						
4.3. The admin wants to update t	he user.							
S-3: Update user subflow is	-	1.						
4.4. The admin wants to delete the								
S-4: Delete user subflow is p	performed	•						
Sub-flows:								
S-1: View user detail subflow	<b>C</b> 1							
1. Admin selects the desired user to vi	ew from the	he list.						
2. System displays the user details.								
S-2: Add new user subflow 1. Admin wants to add a new user.								
	aromat in	ot						
<ol> <li>System displays a modal pop up to p</li> <li>Admin enters name, email, containing</li> </ol>		-	umber role					
warehouse, employed in, address an			iumber, ioie,					
4. System will use the phone number			password and					
create the user into the database.		Point 1	assing and					
S-3: Update user subflow								
1. Admin wants to update a user.								

Table 4:7: Use case description of manage user

- 2. System displays a modal pop up to prompt input.
- 3. Admin enters a new email, contact number, role, warehouse or address.
- 4. System updates the user details.
- S-4: Delete user subflow
  - 1. Admin wants to delete a user.
  - 2. System displays a confirmation modal to prompt for admin's confirmation.
  - 3. Admin confirms to delete the user.
  - 4. System change the user's status to inactive.

Alternate/Exceptional Flows:

Use Ca	ase Nar	ne: View	reports			ID: 6	Importance
0.50 0.0			reports			ID. 0	-
							Level: High
Primary Actor: Admin, warehouse Use Case Type: Detail, essential manager							
Stakeh	olders	and Inter	rests:				
	ı - want ehouse		w daily r	reports and cy	cle countir	ng summ	ary reports of
Wareh	ouse n	nanager	- wants	s to view da	ily reports	s and cy	cle counting
		-	s wareho		•		C
Brief I	Descrip	tion: Thi	s use cas	e describes ho	w admin a	nd wareh	ouse manager
view d	laily rep	ports and	l cycle co	ounting summ	ary reports	5.	_
				manager war	nts to view	daily rep	ports or cycle
		mary rep	oorts.				
	onships						
				louse manager			
Include		: Mana	ge cycle	counting app	roval repor	t	
Extend	ı alizatio	:-					
Genera	anzatio	n : -					
Norma		of Event					
1.				nanager logins			n.
2.				nanager wants		-	
	2.1.			house manage			ly reports
			•	report subflo	-		
	2.2.			0			nmary reports
	• •			mary report su	-		
	2.3.			-	to manag	e the cy	cle counting
		11	al report				
				cycle count	approval	report	use case is
0.1.0		perform	ned.				
Sub-flo		ly man a m	t aubfla				
		• •	t subflow	v nanager choos	as to view	doily ron	orta
1.				ints to view da		• 1	0118.
	a.			displays all da	v 1		arahousas
			•	nin wants to s	• •		
					-		orts for that
		111.	•	use only.	uispiays u	uny rep	ond not that
	h	If the v		se manager wa	ants to view	v dailv re	ports
	υ.	i.					f warehouse
			•	r's warehouse	•	-r-105 0	
2.	Admi	n or ward	U		•	f the rend	orts displayed
				in or stock ou		P	
3.		-				for the se	elected report.
	•		eport sub				*

Table 4:8: Use case description of view reports

- 4. Admin or warehouse manager chooses to view summary reports.
  - a. If the admin wants to view summary reports
    - i. System displays all summary reports of all warehouses
  - b. If the warehouse manager wants to view summary reports
    - i. System displays all cycle counting summary reports of warehouse manager's warehouse only
- 5. Admin or warehouse manager can select one of the SKU ID displayed in the listing for summary report details.

6. System displays a modal popup to display the summary report details. Alternate/Exceptional Flows:

Use Case Name: Manage cycle counting	approval	ID:	Importance			
report		7	Level: High			
Primary Actor: Warehouse manager Use Case Type: Detail essential						
Stakeholders and Interests: Warehouse manager - wants to manage cyc his warehouse.	cle counting	g appro	val reports of			
Brief Description: This use case describes and approve cycle counting approval reports		ouse m	anagers view			
Trigger: Admin or warehouse manager w approval reports.	ants to ma	nage c	ycle counting			
Relationships:Association: Warehouse managerInclude: -Extend: -Generalization : -						
<ol> <li>Normal Flow of Events:         <ol> <li>Warehouse manager logins into the second s</li></ol></li></ol>	ing approve port from the varehouse nection. th the varia	le coun al repor ne list to nanager nce.	rts. 5 approve.			
<ul> <li>Alternate/Exceptional Flows:</li> <li>4a: If the manager rejects the report <ol> <li>The system will display a popup mereassign the staff to recount the SKU</li> <li>If the manager confirms to recount the a schedule and reassign to the staff, status of the report as "rejected" vecount.</li> </ol> </li> </ul>	J. he SKU, the else, the sy	system stem w	will generate ill change the			

Table 4:9: Use case description of manage cycle counting approval report

	ID: 8	Importance
	ID. 8	Importance
		Level: High
e C entia	-	ype: Detail,
le coi	unting	
areh	ouse man	ager manages
he cy	cle count	ting
tart th is po to v	ycle cour he cycle c erformed view upc	counting
form ouse, o gnme cle co es th	to prom counting nt, inver ounting. e daily c	ting process. pt warehouse frequency for ntories to be ount for each
ots for cycl	r warehou e countii	ng process is unt personnel
	ots for cycl	cycle countin

Table 4:10: Use case description of manage cycle counting

- S-2: View upcoming cycle counting SKUs subflow is performed.
  - 1. Warehouse manager wants to view upcoming cycle counting SKUs.
  - 2. System displays the lists of upcoming SKUs by cycle counting classes.
  - 3. Warehouse manager wants to reassign staff or view cycle counting settings
    - a. If manager wants to reassign staff
      - i. System prompts manager to select the staff to replace and staff to assign
      - ii. System moves all schedules assigned for staff to replace and reassign to new staff selected.
    - b. If manager wants to view cycle counting settings
      - i. System shows a modal pop-up with all current cycle counting settings

Alternate/Exceptional Flows:

Use Case Name: Record inventory		ID: 9	Importance Level: High
Primary Actor: Staff	Use Case	Type: De	etail, essential
Stakeholders and Interests: Staff - wants to record inventory stock			
Brief Description: This use case describes h	ow staff red	cords inv	entory stock.
Trigger: Staff wants to check in/ out invento	ory stock		
Relationships:			
Association : Staff			
Include : -			
Extend :-			
Generalization : -	_	_	
Normal Flow of Events:			
1. Staff logins into the application.			
2. Staff chooses to record inventory.			
3. System displays the inventory list of		warehou	se.
4. Staff searches the inventory by ID or	r name.		
5. Staff checks in or checks out the des	ired invent	ory.	
6. System prompt inputs from staff.			
7. Staff enters quantity to check in or c		d the ren	narks.
8. System records the data into the data	ibase.		
Sub-flows:			
Alternate/Exceptional Flows:			

Table 4:11: Use case description of record inventory

Use Case Name: Perform cycle cour	nting	ID: 10	Importance Level: High
Primary Actor: Staff	Use C essentia	-	/pe: Detail,
Stakeholders and Interests: Staff - wants to perform cycle count	ting		
Brief Description: This use case desc	cribes how staff p	erforms c	ycle counting
Trigger: Staff wants to perform cycl	e counting		
Relationships: Association : Staff Include : - Extend : - Generalization : - Normal Flow of Events: 1. Staff logins into the applicati 2. System displays a list of upcon cycle counting SKUs. 3. Staff chooses one SKU from count. 4. System displays the bin locat	oming, pending a	ning SK	Us to perform
<ul> <li>a modal popup to prompt the</li> <li>5. Staff enters the actual count</li> <li>6. System records the actual count</li> <li>a cycle count approval report</li> <li>7. System moves the counted S</li> </ul>	e actual count of t into the modal p punt, calculates v t.	he SKU opup. ariances	from staff. and generates
Sub-flows:		ig approv	ai 115t.
Alternate/Exceptional Flows: 7a: If the cycle count approval report the SKU will be placed into the upco		e wareho	ouse manager,

Table 4:12: Use case description of perform cycle counting

Use Case Name: View all staff		ID: 11	Importance Level: High
Primary Actor: Warehouse manager	Use Case 7	Гуре: De	etail, essential
Stakeholders and Interests: Warehouse manager - wants to view all stat	ff in his war	rehouse.	
Brief Description: This use case describes all staff in his warehouse.	how a ware	house m	anager views
Trigger: Warehouse manager wants to view	staff inform	nation.	
Relationships: Association : Warehouse manager Include : - Extend : - Generalization : -			
<ol> <li>Normal Flow of Events:         <ol> <li>Warehouse manager logins to the sy</li> <li>Warehouse manager wants to view t</li> <li>System displays the details of the wa</li> <li>Warehouse manager chooses a staff to</li> <li>System displays the staff's details.</li> </ol> </li> </ol>	he staff in t arehouse an	d a staff	list.
Sub-flows:			
Alternate/Exceptional Flows:			

Table 4:13: Use case description of view all staff

Use C	ase Nai	me: Manage category		ID: 12	Importance Level: High
Prima	ry Acto	r: Admin	Use essenti	-	ype: Detail,
Stakel	nolders	and Interests:			
		ts to add, view, update and dele	ete categ	gory	
	Descrip catego	tion: This use case describes h ry.	ow adm	in view, ac	ld, update and
Trigge	er: Adn	nin wants to manage the catego	ory.		
Assoc Incluc Exten		: Admin : - : -			
Norm	al Flow	of Events:			
5.		dmin logins the application.			
6.		dmin chooses to manage the ca	ategory.		
7.		ystem displays all categories.	0 1		
8.		dmin can add, update or delete	a categ	ory.	
	8.1.	The admin wants to add a ne	-	•	
		S-1: Add new category subfl	0	•	
	8.2.	The admin wants to update t	-		
		S-2: Update a category subfl	-	•	
	8.3.	The admin wants to delete th	-		
		S-3: Delete a category subfle	ow is pe	rformed.	
Sub-fl	ows:	× ·	*		
S-1: A	dd new	category			
1.	Admi	n wants to add a new category	•		
2.	System	m displays a modal pop up to p	prompt i	nput.	
		n enters the category name.			
	•	m creates the category and sav	e into da	atabase.	
S-2: U	Jpdate a	acategory			
1.	Admi	n wants to update a category.			
	-	m displays a modal pop up to p	prompt i	nput.	
		n edit the category name			
	-	m updates the category.			
		category			
1.		n wants to delete a category.			
2.	•	m displays a confirmation 1	modal t	to prompt	for admin's
	confir	mation.			

Table 4:14: Use case description of manage category

- Admin confirms to delete the category.
   System deletes the category.

Alternate/Exceptional Flows:

# 4.5 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) illustrates all entities needed for the implemented system and how each entity is related to each other. Figure 4.12 shows the ERD diagram of the implemented warehouse inventory management system.

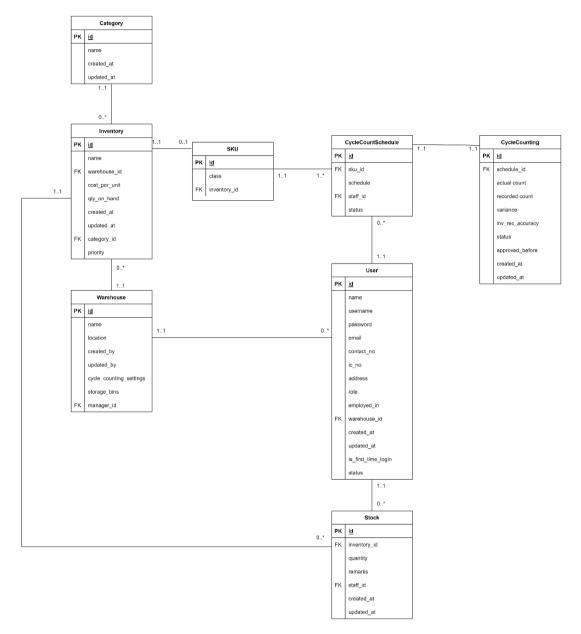


Figure 4.12: ERD diagram

## 4.6 Prototype

The prototype development was developed based on the three main modules specified in the project scope, which will include the module of inventory stock recording, cycle counting auto-scheduler and report management.

## 4.6.1 Inventory stock recording

The first module included in the prototype development is the inventory stock recording. Figure 4.13 shows the wireframe of staff viewing all inventories on the record inventory page. After the staff selects one of the inventories to check in / check out, an inventory modal pop-up will be displayed to prompt for the quantity being taken out or received, then, a confirmation modal is shown for staff's confirmation as shown in Figure 4.14 and Figure 4.15.

ord inventory	View inventory	Cycle (	Counting					
						Q Search		
Inventory ID	Name	Cost per unit	Quantity on hand	Storage Bin Number	Created by	Updated by	Action	
INV-001	SYRINGE -5ML XMVPGH20 100PCS	RM300.00	50	A23	2022-04-05 2:35 PM	2022-04-05 2:35 PM	CHECK IN/OUT	
INV-002	OXYMETER SFSDF422 1PC	RM150.00	120	C27	2022-04-05 4:35 PM	2022-04-05 4:35 PM	CHECK IN/OUT	
INV-003							CHECK IN/OUT	
INV-004							CHECK IN/OUT	
INV-005							CHECK IN/OUT	

Figure 4.13: Staff- record inventory page

Check in	n / out inve	ntory - INV-001	
Inventory Name	SUNLIGHT KLANG VA	LLEY SDN BHD	
Check in	Check out		
Check In			
Amount			
Remarks			
	Cancel	Confirm	

Are you	sure to check in:
Inventory ID	INV-001
Inventory Name	SUNLIGHT KLANG VALLEY SDN BHD
Amount	50
Remarks	Received from SUPPLIER
	Cancel Confirm

Figure 4.14: Staff- check in inventory popup

Check in	/ out inventory - INV-001	
Inventory Name	SUNLIGHT KLANG VALLEY SDN BHD	
Check in	Check out	
Check Out		
Amount		
Remarks		
	Cancel Confirm	

sure to check out:
INV-001
SUNLIGHT KLANG VALLEY SDN BHD
30
Shipping for ABC PHARMACY
Cancel Confirm

Figure 4.15: Staff- check out inventory popup



Figure 4.16: Staff- show bin location after check in inventory

## 4.6.2 Cycle counting auto scheduler

The second included module is the cycle counting auto-scheduler. The first wireframe for this module is the manage-cycle-counting feature which only warehouse managers can access. For the homepage of the manage-cycle-counting view, there are two actions available, which are starting a new cycle counting or viewing upcoming cycle counting SKUs as shown in Figure 4.17.

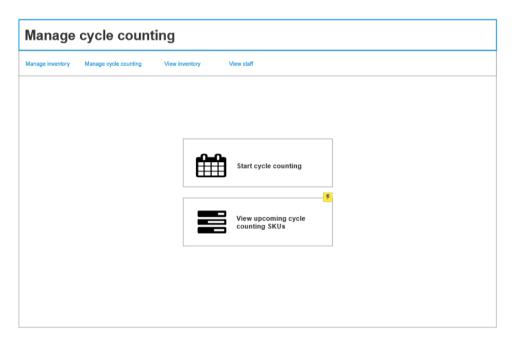


Figure 4.17: Warehouse manager- manage cycle counting page

For starting a new cycle counting, a few inputs will be required which include the workday of the warehouse, the counting frequency of each cycle counting class, staff to be assigned to cycle counting, SKUs to be included, and the start and end date of the cycle counting as shown in Figure 4.18.

Start cy	cle counting
Working day Monday 🗸	of warehouse: to Friday •
Counting free	ineuch.
	Every Month(s) V
Class B:	Every Month(s) V
Class C:	Every Month(s) ¥
Assign Staff:	Staff 2, Staff 3, Staff 5
Select Inventories:	Staff 1         If Staff 2         If Staff 3         Staff 4         If Staff 5         Staff 6
Start date:	
End date:	Cancel Confirm

Figure 4.18: Warehouse manager- start cycle counting

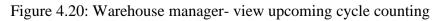
After the necessary details have been filled, the daily counts for each cycle counting class will be calculated and a confirmation dialog will pop up to display the daily counts and prompt for confirmation as shown in Figure 4.19. Upon confirmation, the system will auto-generate all schedules for cycle counting according to the start and end date entered and assign to staff evenly.

Assign Staff: No. of SKU:	Staff 2, Staff 3, 5 1043	Staff 5 🕘 2	2022-04-06 ~ 20	23-04-0
Group	Items	Frequency (per year)	Daily count	]
Class A	1064	12	35	
Class B	1256	6	21	]
Class C	893	1.2	3	]
Total	3213	19.2	63	1
It will replace the	existing cycle countin	ng (ignore if there is no	ongoing cycle cou	nting yet)

Figure 4.19: Warehouse manager- start cycle counting confirmation popup

Next, the warehouse manager is able to view all upcoming cycle counting schedules in this module as shown in Figure 4.20.

nage inventory	Manage cycle counting	View inventory	View staff				
Group A	Group B	Group C		2022-04-08 ~ 2023-04-08		Q Search	
SKU ID	Schedule ID	Inventory ID	Inventory name	Storage number	Schedule	Days due	Staff assigned
SKU-001	SCH-001	INV-001	SYRINGE -5ML XMVPGH20 100PCS	A-23	2022-06-01	1	Lee Chia Ning



Staff, on the other hand, will be able to view all SKUs assigned to them in Figure 4.21 and perform the counting before the deadline. When the staff chooses one SKU from the list to initiate the count, a popup will be displayed as shown in Figure 4.22 to show the bin location of that SKU and prompt for the actual count from the staff.

Record inventory	View inventory	Cycle Counting				
Upcoming	Pending Approval	Completed	e	2022-05-31 ~ 202	22-08-31 Q Sear	rch
SKU ID	Cycle counting ID	Inventory ID	Inventory name	Storage number	Schedule	Action
SKU-001	CC-001	INV-001	SYRINGE -5ML XMVPGH20 100PCS	A-23	2022-06-01	Count

Figure 4.21: Staff- perform cycle counting

Count SKU	
Bin location:	A23
System count	80
Enter actual count:	
ок	

Figure 4.22: Staff- count SKU

#### 4.6.3 Report Management

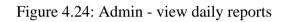
The third module is the report management. The report management can be accessed by the warehouse manager and the admin. For the warehouse manager, all information retrieved will be on his own warehouse only, whereas the admin will be able to view all reports from all warehouses.

The first report in this module is the daily reports, which contains information on all transactions of a warehouse on a specific date. The list of all reports is shown in Figure 4.23 for the warehouse manager and Figure 4.24 for the admin. The viewing of daily report details is in Figure 4.25.

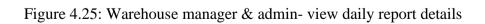
report  Perence Date Action	ports Manage cycle counting View inventory	View staff	
2 Q Search Prehouse ID Warehouse name Date Action	approval reports automary reports		
rehouse ID Warehouse name Date Action	/ report		
	022		Q Search
WH-001 SUNJIGHT KLANS VALLEY SDN BHD 2022-04-06 VIEW	Varehouse ID Warehouse name	Date	Action
	WH-001 SUNLIGHT KLANG VALLEY SDN BHD	2022-04-04	s view

Figure 4.23: Warehouse manager - view daily reports

View reports         Manage warehouse       Manage user       View reports         Daily reports         Daily reports         OBI04/2022       All warehouses       C       Search         Daily reports         OBI04/2022       All warehouses       O C       Search         View reports         View reports         OBI04/2022       All warehouses       O       C       Search         View reports       O         View reports       Image user       View reports       Image user       C       Search         View reports e ID       Warehouse name       Date       Action       View       View						
Mana	age inventory	Manage warehouse	Manage user View reports			
Da	aily reports	Summary reports				
Da	ily repor	t				
06/0	4/2022		All warehouses	¥	Q Search	
	Warehouse ID		Warehouse name	Date	Action	
	WH-001	SUNLIGHT KLANG VAL	LEY SDN BHD	2022-04-08	VIEW	
	WH-003	ABC WAREHOUSE		2022-04-06	VIEW	



		WH-001					
rehouse name		SUNLIGHT	KLANG VALLEY SDN B	BHD			
te of Report		2022-04-06					
Stock ID	Inventory ID	Inventory Name	Original count	Quantity in/out	Total count	Staff Responsible	Remarks
S-001	INV-001	SYRINGE -5ML XMVPGH20 100PCS	30	50	80	Lee Chia Ning	Recieved from SUPPLIER ABC
S-004	INV-001	SYRINGE -5ML XMVPGH20 100PCS	80	-30	50	Lee Chia Ning	Shipping for SUNLIGHT PHARMACY



The second report is the cycle counting summary report. Figure 4.26 and 4.27 shows the view of the warehouse manager and admin whereas Figure 4.28 shows the details of summary report.

							Q Search	
SKU ID	Cycle Count	Inventory name	Warehouse name	Staff name	Recorded count	Actual	Date counted	IRA (%)
sku 🥇	00.004	SYRINGE -5ML XMVPGH20 100PCS	SUNLIGHT KLANG VALLEY SDN BHD	LEE CHIA NING	376	370	2022-04-05 1:30 PM	98.4
		-		-				

Figure 4.26: Warehouse manager- view cycle counting summary reports

All warehous	es	¥					Q Search	
SKU ID	Cycle count ID	Inventory name	Warehouse name	Staff name	Recorded count	Actual count	Date counted	IRA (9
sкu- <mark>.</mark> .	CC-001	SYRINGE -5ML XMVPGH20 100PCS	SUNLIGHT KLANG VALLEY SDN BHD	LEE CHIA NING	376	370	2022-04-05 1:30 PM	98.4
SKU-002	CC-001	OXYMETER SCSHJAK720 1PC	ABC WAREHOUSE	LOW PEI LING	230	234	2022-04-06 7:00 AM	98.3

Figure 4.27: Admin- view cycle counting summary reports

Summ	ary	Re	port D	etails	;- \$	SKU-00	01
SKU ID		SKU-00	1				
Cycle count I	D	CC-001					
Inventory ID		INV-001					
Inventory nar	me	SYRING	GE -5ML XMVPGH	120 100PCS			
Warehouse I	D	WH-002	2				
Warehouse n	Warehouse name SU		HT KLANG VALLE	Y SDN BHD			
Satff ID		US-001					
Staff name		LEE CH	IIA NING				
Date counted	ł	2022-04	I-05 1:30 PM				
	System	n count	Actual count	Variance	•	Absolute variance	
	3	76	370	-6		6	
			IRA=	98.4%			
			Ok	c			

Figure 4.28: Warehouse manager & admin- view cycle counting summary report details

The third report is the cycle counting approval report, which can be accessed by the warehouse manager only as shown in Figure 4.29. The warehouse manager can select one of the reports and approve or reject it. By approving the report, the variance will be added on top of the inventory of that report, whereas by rejecting it, the cycle counting SKU will be returned to the warehouse staff for recounting. The confirmation modal of the approve and reject action is shown in Figure 4.30 and Figure 4.31.

KU ID         Cycle Count ID         Inventory name         Warehouse name         Staff name         Recorded count         Actual Count         Date counted         Variance         Action           U-001         CC-001         SYRINGE-5ML XMVPGH20 100PCS         SUNLIGHT KLANG VALLEY SDN BHD         LEE CHIA NING         376         370         2022-04-05 1:30 PM         -6         Approve         Rejet							C	λ Searc	:h	
	Cycle Count ID	Inventory name	Warehouse name	Staff name			Date counted	Variance	Actio	n
				LEE CHIA NING	376	370	2022-04-05 1:30 PM	-6	Approve	Reje
		ID	ID INVENIOUS NAME	ID INVENIOUS NAME WATEROUSE NAME OC. 001 SYRINGE -5ML SUNLIGHT KLANG VALLEY	ID INVENTION HAINE VALENT STAIL HAINE STAIL HAINE	ID INVENTION NAME VALUES NAME Statilitatile count	ID Inventory name variences ename Stain name count count	ID Inventory name vvarenouse name Stain name count count Date counted	ID Inventory name valenduse name Stain name count count Date counted valenduse name stain name count count Date counted valenduse name stain name count coun	ID Inventory name violationse name Stain name count Date counter violatione Action of SYRINGE -5ML SUNLIGHT KLANG VALLEY LEE CHILA NING 376 370 2023 04 05 120 DM 6 COUNTER VIOLATION OF COUNTER VIOLOTATION OF COUNTER VIOLATION OF COUNTER VIOLATION OF COUNTER VIOLOTATION OF COUNTER VIOLOTATION OF COUNTER

Figure 4.29: Warehouse manager - view cycle counting approval reports

Approve cycle count report	
Are you sure to approve the cycle count report	t?
This approval will modify the system count with the variances : -6	
Cancel Confirm	

Figure 4.30: Warehouse manager - approve cycle counting approval reports

Reject cycle count report
Are you sure to reject the cycle count report?
This reject will reassign the staff responsible to recount the SKU.
Cancel Confirm

Figure 4.31: Warehouse manager - reject cycle counting approval reports

### 4.7 Summary

This chapter discussed the facts-findings and functional and non-functional requirements. Besides, use case diagram and use case diagram description to visualise the interaction between admin, warehouse manager, staff and system are also included. An ERD diagram is designed to show the relationship of entities. Lastly, screenshots of the developed low-fidelity prototype are also included.

#### **CHAPTER 5**

#### SYSTEM DESIGN

#### 5.1 Introduction

In this chapter, the system architecture design and modelling diagrams which include data flow diagrams and user interface flow diagrams are described. For the data flow diagram, the processes are according to the use cases mentioned in earlier chapters (Figure 4.11). Context diagram, level-0 DFD, level-1 DFD, level-2 DFD and level-3 DFD are presented. The user interface flow diagrams are drawn based on three types of users: staff, admin and manager. Lastly, screenshots of the implemented web-based system's user interface design are provided. All screenshots are arranged based on use cases as well for easier reading and understanding purposes.

#### 5.2 System Architecture Design

The implemented system's architecture design is shown in Figure 5.1. The entire system is hosted on Heroku in which users or testers can access via <u>https://fyp-wims.herokuapp.com/</u>. The application is a web-based application developed in which its frontend is being developed using Vue.js whereas the RESTful APIs are built using the Laravel framework, which includes API authentication as well. The implemented system will send HTTP requests using the API endpoints in which Laravel will fetch data from the MySQL database and returns data to the frontend in JSON response format. For the check-in and check-out stock module, Laravel will create an event using Pusher API in which the Pusher will broadcast an API message to all clients that subscribed to the Pusher channel.

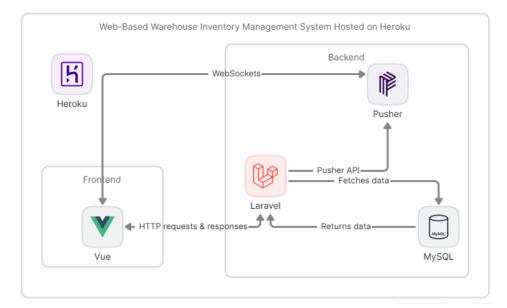


Figure 5.1: System architecture design

### 5.3 Modelling Diagram

This section shows modelling diagrams to visually describes the implemented system. The modelling diagrams will include data flow diagrams and interface flow diagrams.

### 5.3.1 Data Flow Diagram

A data flow diagram (DFD) maps the flow of data across the implemented system. Multi-level data flow diagrams show how data is handled from a lower perspective. In this section, a context diagram, DFD level-1, DFD level-2, DFD level-3 and DFD level-4 are drawn to show the data flow in the warehouse inventory management system.

# 5.3.1.1 Context Diagram

The context diagram is the basic overview of the data flow across the entire system as shown in Figure 5.2.

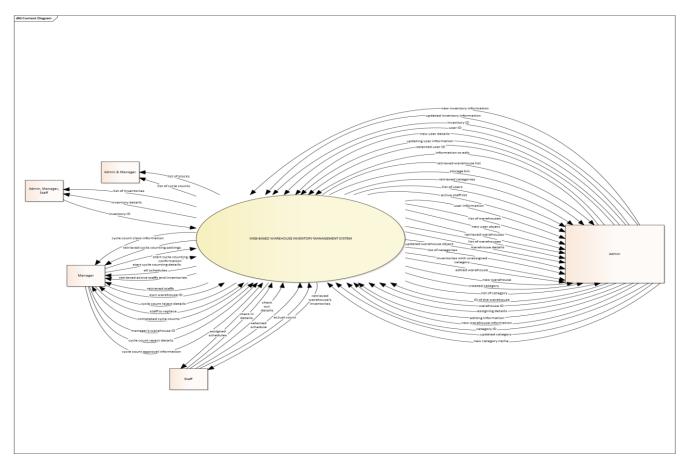


Figure 5.2: Context diagram

# 5.3.1.2 Data Flow Diagram Level-1

The DFD level-1 shows the detailed breakouts of the context diagram. At this level, all main processes of the system are highlighted and the data stores are also shown. Figure 5.3 shows the DFD level-1 of the implemented system in which the data flow between actors (users), processes and data stores are shown.

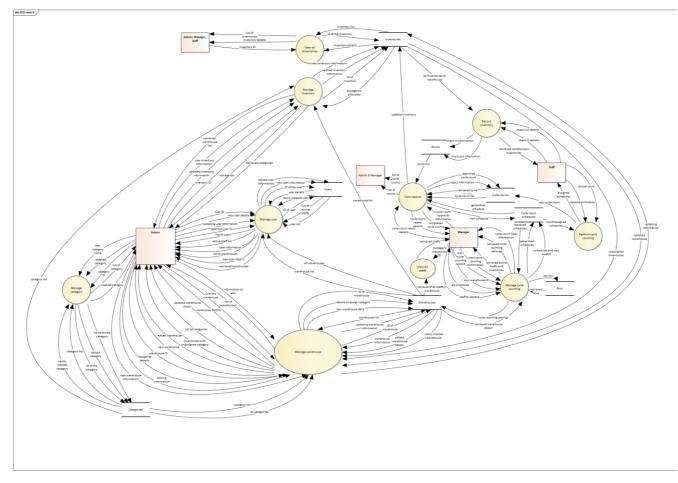


Figure 5.3: Data flow diagram level-1

## 5.3.1.3 Data Flow Diagram Level-2

DFD level-2 goes deeper into the processes of DFD level-1 which will show the subprocesses inside a process in DFD level-1. The processes in DFD level-1 that need to be detailed out are manage inventory process, manage user process, manage warehouse process, manage category process, manage cycle counting process, perform cycle counting process, record inventory process and view reports process.

### 5.3.1.3.1 Manage Inventory

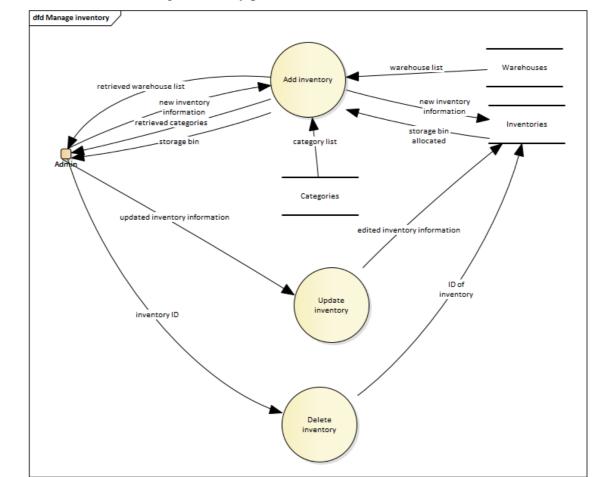
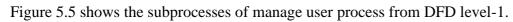


Figure 5.4 shows the DFD level-2 of the manage inventory process from DFD level-1.

Figure 5.4: Data flow diagram level-2 (Manage inventory)

# 5.3.1.3.2 Manage User



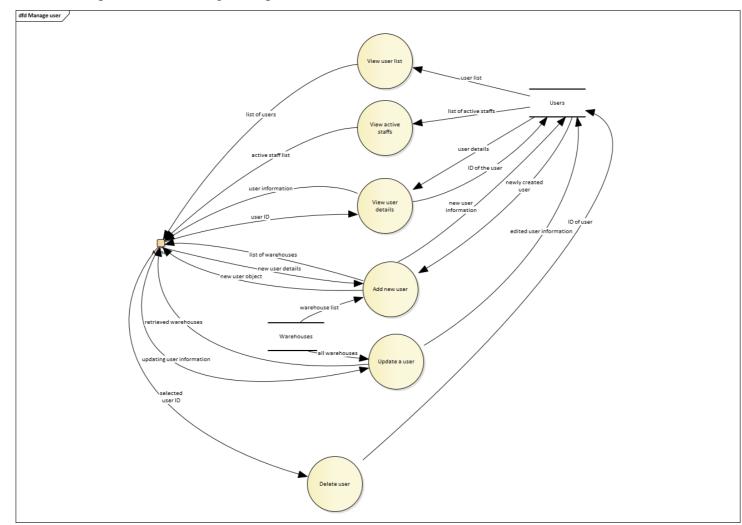


Figure 5.5: Data flow diagram level-2 (Manage user)

### 5.3.1.3.3 Manage Warehouse

Figure 5.6 shows the subprocesses of manage warehouse process from DFD level-1.

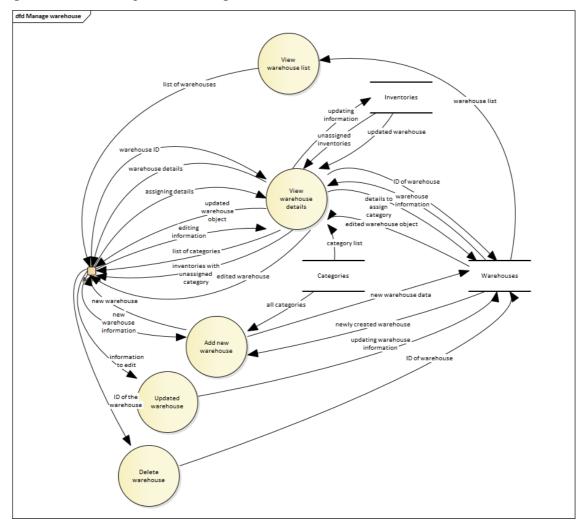


Figure 5.6: Data flow diagram level-2 (Manage warehouse)

## 5.3.1.3.4 Manage Category

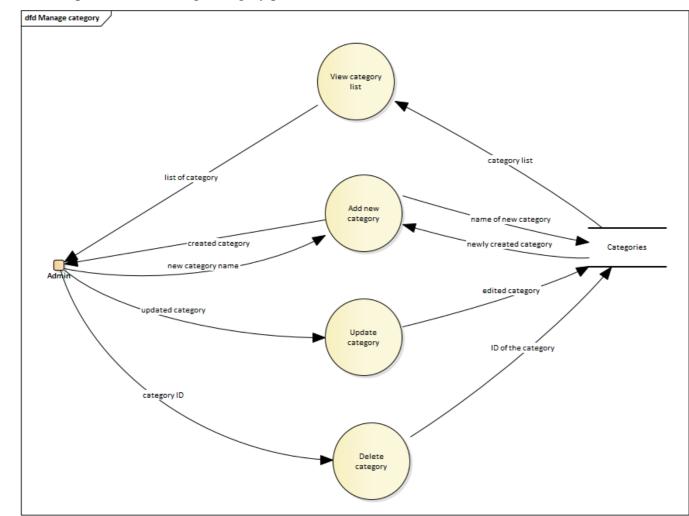


Figure 5.7 shows the subprocesses of manage category process from DFD level-1.

Figure 5.7: Data flow diagram level-2 (Manage category)

## 5.3.1.3.5 Manage Cycle Counting

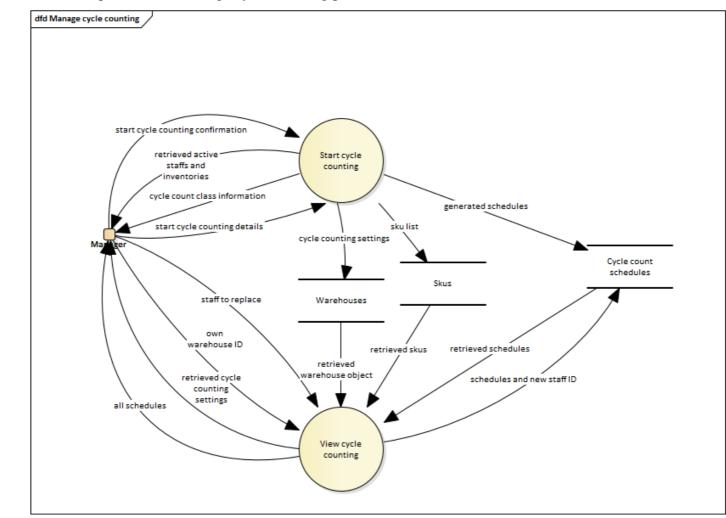
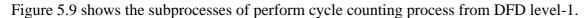


Figure 5.8 shows the subprocesses of manage cycle counting process from DFD level-1.

Figure 5.8: Data flow diagram level-2 (Manage cycle counting)

## 5.3.1.3.6 Perform Cycle Counting



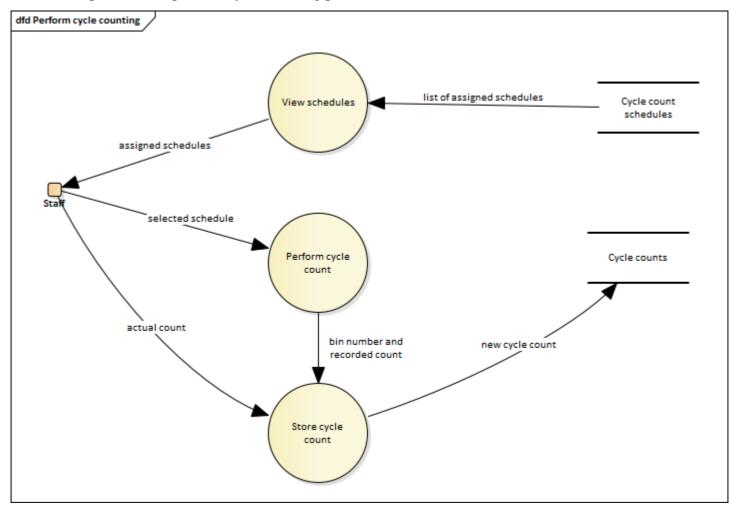


Figure 5.9: Data flow diagram level-2 (Perform cycle counting)

### 5.3.1.3.7 Record Inventory

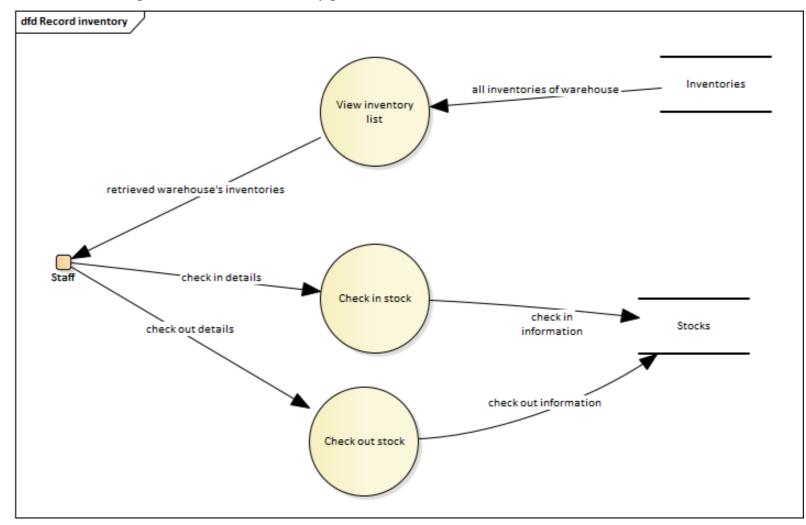


Figure 5.10 shows the subprocesses of record inventory process from DFD level-1.

Figure 5.10: Data flow diagram level-2 (Record inventory)

### 5.3.1.3.8 View reports

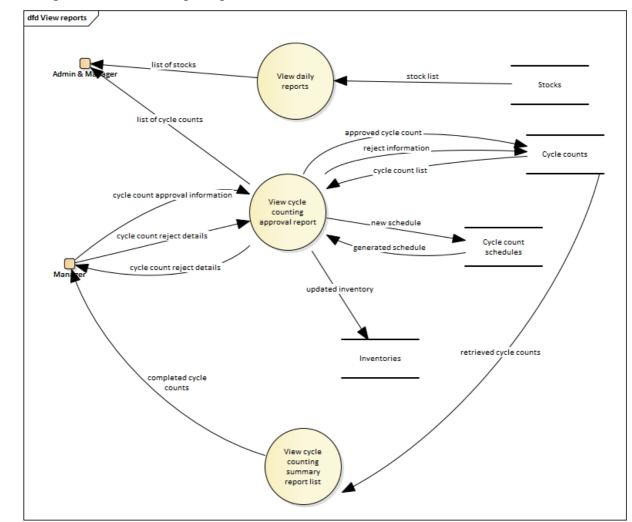


Figure 5.11 shows the subprocesses of view reports process from DFD level-1.

Figure 5.11: Data flow diagram level-2 (View reports)

#### 5.3.1.4 Data Flow Diagram Level-3

Similar to DFD level-2, the DFD level-3 goes deeper into processes in DFD level-2. This section includes the view warehouse details process, view cycle counting approval reports process, start cycle counting process and view cycle counting proves from DFD level-2.

#### **5.3.1.4.1** View Warehouse Details

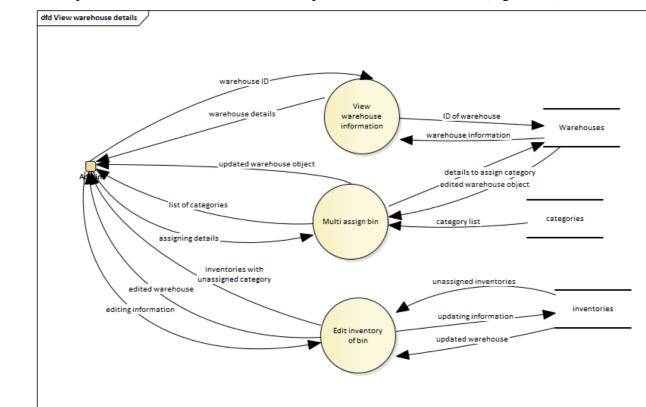


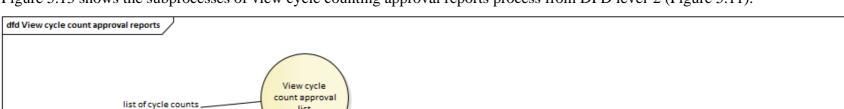
Figure 5.12 shows the subprocesses of view warehouse details process from DFD level-2 (Figure 5.6).

Figure 5.12: Data flow diagram level-3 (View warehouse details)

### 5.3.1.4.2 View Cycle Counting Approval Reports

dmin & Manager

Manage



cycle count list

approved cycle count

updated inventory

Cycle counts

Cycle count schedules

reject information

Inventories

Figure 5.13 shows the subprocesses of view cycle counting approval reports process from DFD level-2 (Figure 5.11).

list

Approve cycle

count approval

report

Reject cycle

count approval report

cycle count approval

information

reschedule information

cycle count reject details

Figure 5.13: Data flow diagram level-3 (View cycle counting approval reports)

new schedule

generated schedule

## 5.3.1.4.3 Start Cycle Counting

Figure 5.14 shows the subprocesses of start cycle counting process from DFD level-2 (Figure 5.8).

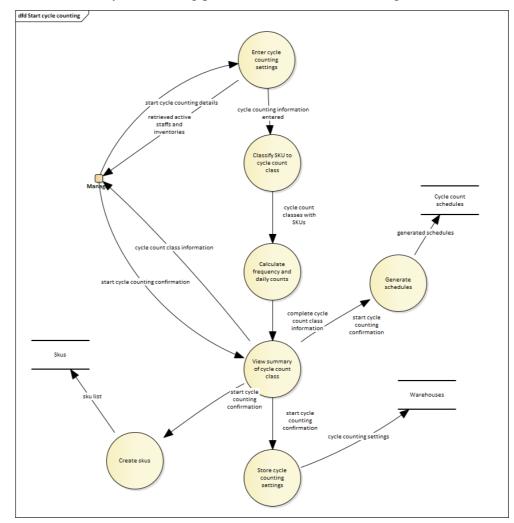


Figure 5.14: Data flow diagram level-3 (Start cycle counting)

### 5.3.1.4.4 View Cycle Counting

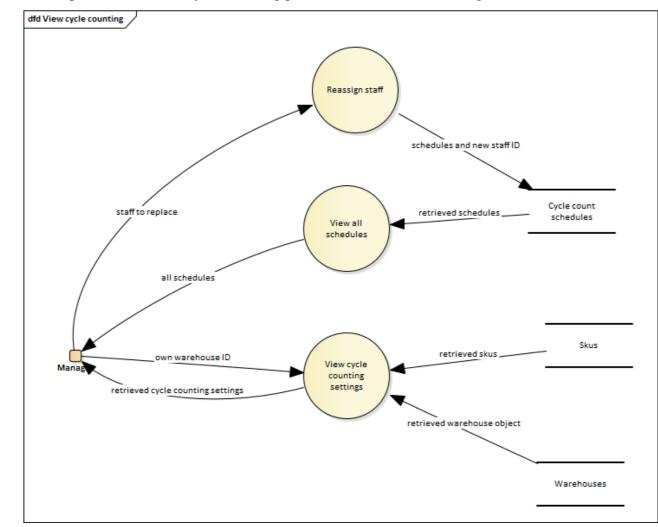


Figure 5.13 shows the subprocesses of view cycle counting process from DFD level-2 (Figure 5.8).

Figure 5.15: Data flow diagram level-3 (View cycle counting)

### 5.3.1.5 Data Flow Diagram Level-4

DFD level-4 shows subprocesses of processes in DFD level-3. This section includes the approve cycle count approval report process and the reject cycle count approval report process from DFD level-3.

### 5.3.1.5.1 Approve Cycle Count Approval Report

Figure 5.16 shows the subprocesses of approve cycle count approval report process from DFD level-3 (Figure 5.13).

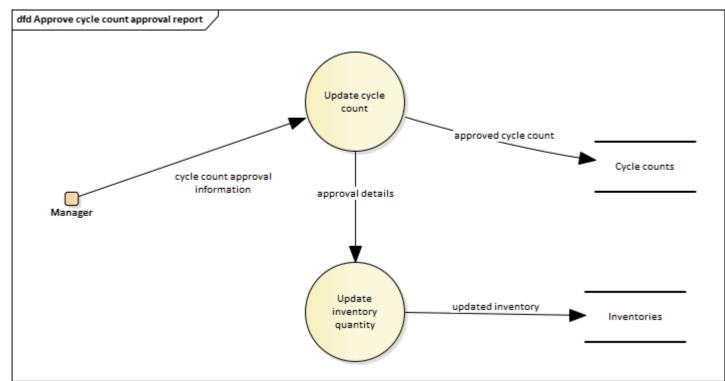


Figure 5.16: Data flow diagram level-4 (Approve cycle count approval report)

### 5.3.1.5.2 Reject Cycle Count Approval Report

Figure 5.17 shows the subprocesses of reject cycle count approval report process from DFD level-3 (Figure 5.13).

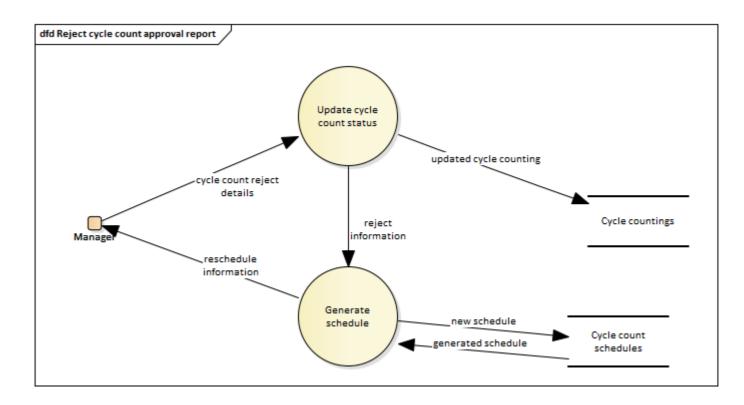


Figure 5.17: Data flow diagram level-4 (Reject cycle count approval report)

#### 5.3.2 Interface Flow Diagram

In this section, interface flow diagrams were drawn to illustrate the flow of the warehouse inventory management system. A total of three diagrams were drawn to show the user interfaces flow for three user roles covered in this project: staff (Figure 5.18), admin (Figure 5.19) and manager (Figure 5.20).

#### 5.3.2.1 Staff

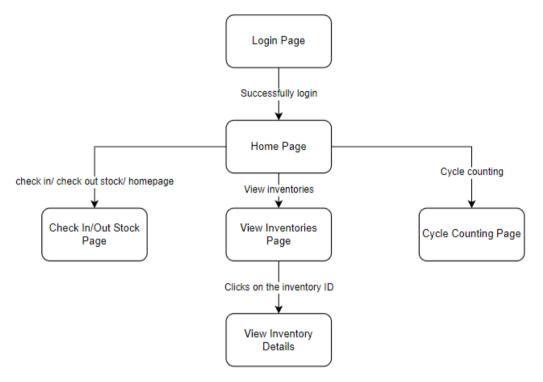
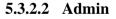


Figure 5.18: Interface flow diagram for staff role



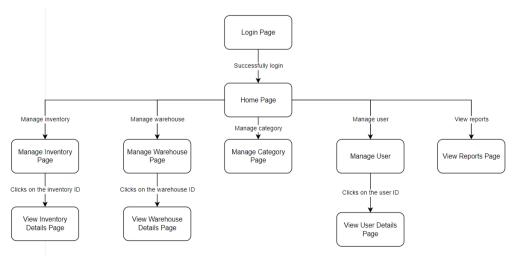


Figure 5.19: Interface flow diagram for admin role

#### 5.3.2.3 Manager

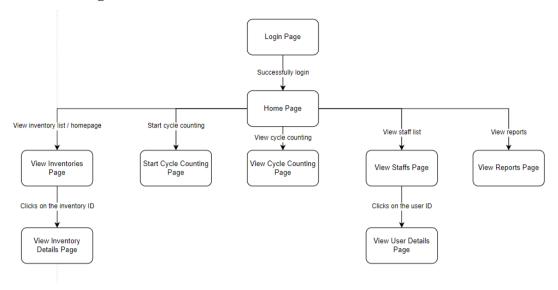


Figure 5.20: Interface flow diagram for manager role

### 5.4 User Interface Design

This section shows all user interface designs implemented in the system and is divided into subsections based on user roles: admin, manager and staff.

#### 5.4.1 User (Admin, Manager and Staff)

This section shows the module that can be used by all users, which is the login module so that the user can log in to the application. After login, they will be presented with different modules based on their roles and will be covered in the following sections.

### 5.4.1.1 Login Account

The Figure 5.21 shows the user interface design for the login account page.

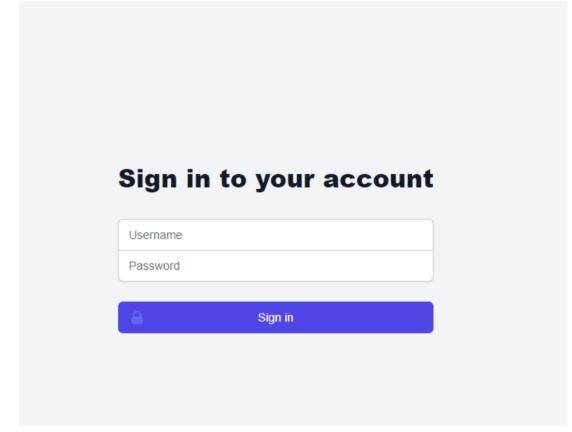


Figure 5.21: User interface design for login account

#### 5.4.2 Admin

This section will list all modules for user role admin, which covers the manage inventory, manage warehouse, manage category, and manage user modules.

#### 5.4.2.1 Manage Inventory

Figure 5.22 shows the user interface design for the manage inventory page and the Figure 5.23 shows the add new inventory modal that shows up after the user clicked on the "Add" button on the manage inventory page. Figure 5.24 is the update inventory modal displayed if the user clicked on the "Edit" button for one of the inventories

displayed on the manage inventory page, and the Figure 5.25 shows the confirmation modal of the inventory deletion.

Manage I	nventory Manage Warehouse Manage Category Mana	age User View reports							Hello, admii
lanage	Inventory								
⊕ Add				~					
ow 10 entrie	is						Search:		
nventory D	Name	Warehouse	Cost Per Unit	Quantity On Hand	Storage Bin Number	Created By	Updated By	Actions	
	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	Hapi Fresh Sdn Bhd	3.5	567	C7		-	Edit	Delete
	Starbucks Frappuccino Chilled Mocha Drink 281 ml	Hapi Fresh Sdn Bhd	13	89	C8		-	Edit	Delete
	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g	Hapi Fresh Sdn Bhd	13.2	230	-	-	-	Edit	Delete
	Grante 100% Pomegranate, Grape And Apple Juice 250 ml	Hapi Fresh Sdn Bhd	6.6	40	A4	-	-	Edit	Delete
	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets) 25 g	Hapi Fresh Sdn Bhd	14.15	89	A5	-	-	Edit	Delete
	Ma Ma Mi Roasted Robusta Ground Coffee 250 g	Hapi Fresh Sdn Bhd	24.6	103	A6	-	-	Edit	Delete
	Coffee Cherry Roasted Robusta Coffee Beans 250 g	Hapi Fresh Sdn Bhd	24.6	70	A7			Edit	Delete
	Delica De Mountain Black Coffee Blend (10 Pieces) 10 g	Hapi Fresh Sdn Bhd	12.6	15	A8		-	Edit	Delete

### 5.4.2.1.1 Manage inventory page

Figure 5.22: User interface design for manage inventory page

### 5.4.2.1.2 Add inventory

Manage I	nventory Manage Warehouse Manage Category	Manage Us	er View report	S							Hello, admin
Manage	Inventory	Add new in	ventory			×					
⊕ Add	© Add		Enter inventory na	me							
Show 10 entrie	'S	Warehouse	Select			~			Search:		
Inventory ID	Name	Quantity On Hand	0				ige Bin ber	Created By	Updated By	C Actions	
	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	Cost per unit	0							Edit	Delete
	Starbucks Frappuccino Chilled Mocha Drink 281 ml	Category				~				Edit	Delete
	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 4									Edit	Delete
4	Grante 100% Pomegranate, Grape And Apple Juice 250 ml	Priority	1			~				Edit	Delete
	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets)		Can	cel Cont	īrm					Edit	Delete
	Ma Ma Mi Roasted Robusta Ground Coffee 250 g	па	pi Fiesii Sun Dilu	24.0	103	Ab				Edit	Delete
	Coffee Cherry Roasted Robusta Coffee Beans 250 g	На		24.6		A7				Edit	Delete
	Delica De Mountain Black Coffee Blend (10 Pieces) 10 g	Ha	pi Fresh Sdn Bhd	12.6		A8				Edit	Delete

Figure 5.23: User interface design for adding inventory

# 5.4.2.1.3 Update inventory

Manag	e Inventory Manage Warehouse Manage Category	Manage Us	ser View reports						Hello, admin
Manag	e Inventory	Edit invent	ory	×					
⊕ Add		ID Name	1 Authentic Tea House Da Hong Pao Colong Tea 300 ml						
Show 10 ent	ries Name		Hapi Fresh Sdn Bhd		ige Bin	Created	Search:	Actions	
1D	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	Quantity On Hand			ber	Ву	By -	Edit	Delete
	Starbucks Frappuccino Chilled Mocha Drink 281 ml	Cost per unit	3.5					Edit	Delete
3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 4 Grante 100% Pomegranate, Grape And Apple Juice 250 ml	Category	Dry & Canned Goods	~				Edit	Delete
	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets)	Priority	5	~				Edit	Delete
	Ma Ma Mi Roasted Robusta Ground Coffee 250 g		Cancel Confirm					Edit	Delete
	Coffee Cherry Roasted Robusta Coffee Beans 250 g							Edit	Delete
	Delica De Mountain Black Coffee Blend (10 Pieces) 10 g		api Fresh Sdn Bhd 12.6 15	AS				Edit	Delete

Figure 5.24: User interface design for updating inventory

## 5.4.2.1.4 Delete inventory

Manage I	nventory Manage Warehouse Manage Category	Manage User View repor	ts					Hello, admin
Manage	Inventory	Delete inventory			×			
Add     Show 10 entrie		Are you sure to delete inventory This action is irrev All data related to thi	versible.				Search:	
Inventory ID	Name	Ca	ancel Del	ete	ige Bin ber	Created By	Updated By	Actions 0
	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	Hapl Fresh Sdn Bhd						Edit Delete
	Starbucks Frappuccino Chilled Mocha Drink 281 ml	Hapi Fresh Sdn Bhd		89	C8			Edit Delete
	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40	g Hapi Fresh Sdn Bhd						Edit Delete
4	Grante 100% Pomegranate, Grape And Apple Juice 250 ml	Hapi Fresh Sdn Bhd	6.6	40	A4			Edit Delete
	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets) 25	g Hapi Fresh Sdn Bhd	14.15	89	A5			Edit Delete
	Ma Ma Mi Roasted Robusta Ground Coffee 250 g	Hapi Fresh Sdn Bhd	24.6		A6			Edit Delete
	Coffee Cherry Roasted Robusta Coffee Beans 250 g	Hapi Fresh Sdn Bhd			A7			Edit Delete
	Delica De Mountain Black Coffee Blend (10 Pieces) 10 g	Hapi Fresh Sdn Bhd	12.6		A8			Edit Delete

Figure 5.25: User interface design for deleting inventory

#### 5.4.2.2 Manage Warehouse

Figure 5.26 shows the user interface design for the manage warehouse page. Figure 5.27 and Figure 5.28 are the user interface designs that display the warehouse details once the user clicked on the warehouse ID on the manage warehouse page. On the warehouse details page, the staff list is shown in Figure 5.27 whereas the list of storage bins of the warehouse is shown in Figure 5.28. The multi-assigning category to storage bins and the editing inventory of storage bin modules are shown in Figure 5.29 and 5.30 respectively, whereas the addition, update and deletion of a warehouse are in Figure 5.31, 5.32 and 5.33 respectively.

#### 5.4.2.2.1 Manage warehouse page

Manage Inv	entory Manage Warehouse	Manage Category Manage User View reports				Hello, admin
Manage	Warehouse					
⊕ Add						
Show 10 entries					Search:	
Warehouse 1D	Name 0	Location	Warehouse Manager	Created at	Updated at A	Actions
1	Hapi Fresh Sdn Bhd	48, jalan ss2/24 47300 petaling jaya selangor D.E, 47300 Petaling Jaya, Selangor	-	-	2022-08- 24T17:22:42.000000Z	Edit Delete
2	GROCERY FRESH WAREHOUSE	Seksyen 4., 34, Jalan Kemajuan, Seksyen 12, 46200 Petaling Jaya, Selangor	GyDci74shp	-	-	Edit Delete
3	ALI GROCERY SDN BHD	77, Jalan 12/17, Seksyen 12, 46200 Petaling Jaya, Selangor	0vp1k0oa4O	-	-	Edit Delete
4	PJ GROCERY WAREHOUSE	No: 4, Jalan 51A/241, Seksyen 51a, 46100 Petaling Jaya, Selangor	xy7jTUNgpb	-		Edit Delete
5	KUALA LUMPUR WAREHOUSE SDN BHD	2, Jalan Telawi 1, Bangsar, 59100 Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur	IZdinJ958H		1997 - E	Edit Delete
14	testadd	asdfsdfsdfsdsf	-	2022-07- 24T00:24:10.000000Z	2022-08- 11T23:06:40.000000Z	Edit Delete
15	test123	sfgsdfgsdfg	test manager	2022-07- 24T08:09:05.000000Z	2022-07- 27T11:20:21.000000Z	Edit Delete

Figure 5.26: User interface design for manage warehouse page

5.4.2.2.2 View warehouse detail	s &	z view	staff list
---------------------------------	-----	--------	------------

Manage Inve	ntory I	Manage Warehouse	Manage Categor	y Manage Use	r Vie	w reports					Hello,	admin
/iew War	ehou	se										
Warehouse	Info											
ID	1						Name	Hapi Fresh	Sdn Bhd			
Location	48, j	alan ss2/24 47300 petal	ing jaya selangor D	D.E, 47300 Petaling	Jaya, Selar	ngor	Warehouse Manager	-				
Created by	-						Updated by	2022-08-24	T17:22:42.000000Z			
Staffs Storag	e Bins											
ow 10 entries										Se	arch:	
Jser ID 🙏 Nar	me ĉ	Email	Co	ntact No. 👘 R	ole 0	Employed in	i ŝ	Status	Created at		Updated at	
stat	г	staff@wims.com	123	3456789 S	taff	Jun-20		ACTIVE			2022-08-02T03:11:13.000000Z	
adn	nin	admin@wims.com	123	3456789 A	dmin	Jun-20		ACTIVE	-		2022-08-19T08:41:26.000000Z	
mai	nager	manager@wims.con	123	3456789 N	lanager	Jun-20		ACTIVE				
5 test	tinf	testing123@wims.co	m 234	423423424 S	taff	2022-08-097	16:00:00.000Z	ACTIVE	2022-08-10T07:33:48.000000Z		2022-08-11T23:39:57.000000Z	
6 test	tstaff	sd@wims.com	123	34567890 S	taff	2022-08-107	16:00:00.000Z	INACTIVE	2022-08-10T08:32:05.000000Z		2022-08-12T00:25:25.000000Z	
9 gwe	erdfasdf	postman@wims.com	012	23433331 S	taff	2022-08-117	16:00:00 0007	INACTIVE	2022-08-11T17-13-13-0000007		2022-08-19T08:38:18 0000007	

Figure 5.27: User interface design for view warehouse details & view staff list

### 5.4.2.2.3 View warehouse details & view storage bins

Manage	e Inventory	Manage Warehouse	Manage Category	Manage User	View reports				Hello, admin
View W	/areho	use							
Wareho	ouse Info								
ID		1				Name	Hapi Fresh Sdn Bhd		
Location		48, jalan ss2/24 47300 peta	aling jaya selangor D.E,	47300 Petaling Jaya	, Selangor	Warehouse Manage	r -		
Created by		-				Updated by	2022-08-24T17:22:42.000000Z		
	itorage Bins								
① Multi Assig	in Bin						~		
Show 10 entr	ries							Search:	
Bin ID 🙏	Bin Numb	er Category I	D â	Inventory ID					Action 0
1	A1	-		-					Edit Inventory
2	A2	Drinks, Cot	ffee & Tea	test add2					Edit Inventory
3	A3	Dry & Canr	ned Goods	-					Edit Inventory
4	A.4	Drinks Col	Tee & Tea	Grante 100% Pomer	ranate Grane And	Apple Juice 250 ml			Edit Inventory

Figure 5.28: User interface design for view warehouse details & view storage bins

# 5.4.2.2.4 Multi-assign storage bins

Manage Inventory Ma	nage Warehouse Manage Category	Manage User View reports	Hello, admin
View Warehous	e	Multi Assign Bin	×
		Category Select ~	
Warehouse Info		Storage Select All Bins	Bhd
Location 48, jaia Created by -	an ss2/24 47300 petaling jaya selangor D.E	A1     A2     A3     A4     A5     A6     A7     A8       A9     A10     A11     A12     A13     A14     A15       A16     A17     A18     A19     A20     B1     B2	
Staffs Storage Bins		B3 B4 B5 B6 B7 B8 B9 B1 B11 B12 B13 B14 B15 B16 B17	
Multi Assign Bin		B18 B19 B20 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11	
Show 10 entries Bin ID t Bin Number		C12 C13 C14 C15 C16 C17 C1 C19 C20 D1 D2 D3 D4 D5	18 Search:
1 A1	Category ID	D6       D7       D8       D9       D10       D11       D12         D13       D14       D15       D16       D17       D18       D1	19 Edit Inventory
2 A2	Drinks, Coffee & Tea	D20       E1       E2       E3       E4       E5       E6       E         E8       E9       E10       E11       E12       E13       E14	Edit Inventory
3 A3	Dry & Canned Goods	E15 E16 E17 E18 E19 E20	Edit Inventory

Figure 5.29: User interface design for multi-assigning storage bins

### 5.4.2.2.5 Edit storage bin's inventory

View Warehouse	Edit inventory ×		
	Inventory test add2 ~		
Warehouse Info			
ID 1	Cancel Edit Inventory Bhd		
Location 48, jalan ss2/24 47300 petaling jaya selangor D.E	47300 Petaling Jaya, Selangor Warehouse Manager -		
Created by -	Updated by 2022-08-24T17:22:42.000000Z		
Staffs Storage Bins	×		
Show 10 entries		Search:	
Bin ID 🕆 Bin Number 🔅 Category ID 🔅	Inventory ID	C Action	
1 A1 -		Edit Inver	itory
2 A2 Drinks, Coffee & Tea	test add2	Edit Inver	itory
3 A3 Dry & Canned Goods		Edit Inver	itory

Figure 5.30: User interface design for editing storage bin's inventory

### 5.4.2.2.6 Add new warehouse

Manage Inv	ventory Manage Warehouse	Manage Category	Manage Us	ser View reports						Hello, admin
Manage	Warehouse		Add new w	varehouse		×				
⊙ Add			Name	Enter warehouse name						
Show 10 entries Warehouse	Name :	Location	<ul> <li>Number of Bins</li> </ul>	0				Search:	C Actions	
	Hapi Fresh Sdn Bhd	48, jalan ss2/24 473 Petaling Jaya, Selar	• Zones	Enter all zones separated	by comma(eg. A,B,C,D,E)			2022-08- 24T17:22:42.000000Z	Edit	Delete
	GROCERY FRESH WAREHOUSE	Seksyen 4,, 34, Jala Jaya, Selangor							Edit	Delete
	ALI GROCERY SDN BHD	77, Jalan 12/17, Seł		Cancel	Confirm				Edit	Delete
4		No: 4, Jalan 51A/24 Selangor	1, Seksyen 51a	a, 46100 Petaling Jaya,	xy7jTUNgpb				Edit	Delete
	KUALA LUMPUR WAREHOUSE SDN BHD	2, Jalan Telawi 1, Ba Persekutuan Kuala I		Kuala Lumpur, Wilayah					Edit	Delete
	testadd	asdfsdfsdfsdsf				2022-07- 24T00:24:1		2022-08- 11T23:06:40.000000Z	Edit	Delete
	test123				test manager	2022-07- 24T08:09:0	5.00000Z	2022-07- 27T11:20:21.000000Z	Edit	Delete

Figure 5.31: User interface design for adding new warehouse

### 5.4.2.2.7 Edit warehouse details

Manage Inv	ventory Manage Warehouse	Manage Category	Manage Us	er View reports						Hello, admin
Manage	Warehouse		Edit Wareh	iouse		×				
① Add			ID							
Show 10 entries			Name					Search:		
Warehouse . ID	Name	Location	Location	48, jalan ss2/24 47300 pe	taling jaya selangor D.E,	47300 Petal		Updated at	a Actions	
	Hapi Fresh Sdn Bhd	48, jalan ss2/24 47 Petaling Jaya, Sela	Warehouse manager	Select		~		2022-08- 24T17:22:42.000000Z	Edit	Delete
	GROCERY FRESH WAREHOUSE	Seksyen 4,, 34, Jal Jaya, Selangor	3	Cancel	Confirm				Edit	Delete
	ALI GROCERY SDN BHD	77, Jalan 12/17, Se	ksyen 12, 4620	0 Petaling Jaya, Selangor	0vp1k0ca4O	-			Edit	Delete
4		No: 4, Jalan 51A/24 Selangor	1, Seksyen 51a	, 46100 Petaling Jaya,	xy7jTUNgpb				Edit	Delete
	KUALA LUMPUR WAREHOUSE SDN BHD	2, Jalan Telawi 1, B Persekutuan Kuala		Kuala Lumpur, Wilayah					Edit	Delete
14	testadd	asdfsdfsdfsdsf				2022-07- 24T00:24:10.		2022-08- 11T23:06:40.000000Z	Edit	Delete
	test123	sfgsdfgsdfg			test manager	2022-07- 24T08:09:05.	000000Z	2022-07- 27T11:20:21.000000Z	Edit	Delete

Figure 5.32: User interface design for editing warehouse details

#### 5.4.2.2.8 Delete warehouse

vianage	Warehouse		Delete warehouse	×			
Add how 10 entries	Name	Location	Are you sure to delete warehouse Hapi Fri This action is irreversible. All data related to this wareh includes staff, inventory, sku staff to other warehouse if n	ouse will be deleted a , reports, and stocks.		Search	Actions
	Hapi Fresh Sdn Bhd	48, jalan ss2/24 473 Petaling Jaya, Selar	Cancel	Delete			Edit Delete
		Seksyen 4., 34, Jala Jaya, Selangor	i Kemajuan, Seksyen 12, 46200 Petaling	GyDci74shp			Edit Delete
	ALI GROCERY SDN BHD	77, Jalan 12/17, Sek	syen 12, 46200 Petaling Jaya, Selangor	0vp1k0oa4O			Edit Delete
		No: 4, Jalan 51A/24 Selangor	, Seksyen 51a, 46100 Petaling Jaya,	xy7jTUNgpb			Edit Delete
	KUALA LUMPUR WAREHOUSE SDN BHD	2, Jalan Telawi 1, Ba Persekutuan Kuala I	ngsar, 59100 Kuala Lumpur, Wilayah umpur				Edit Delete
	testadd	asdisdisdisdst			2022-07- 24T00.24:10.0000002	2022-08- 11T23-06-40.0000002	Edit Delete
				test manager			Edit Delete

Figure 5.33: User interface design for deleting warehouse

# 5.4.2.3 Manage Category

Figure 5.34 shows the user interface design for the manage category page and the add, update and delete of category is shown in Figure 5.35, 5.36 and 5.37 respectively.

### 5.4.2.3.1 Manage category page

	Manage Inventor	ry Manage Warehouse	Manage Category	Manage User	View reports				Hello, admin
Ma	nage Ca	itegory							
⊕ A	44								
_	10 entries							Search:	
ID	ੈ Ca	ategory Name		Created At		Updated At	Action		
1	Dri	inks, Coffee & Tea		-		-		Edit	Delete
2	Dry	y & Canned Goods		-		-		Edit	Delete
3	Sna	lacks		-		-		Edit	Delete
4	Ch	nilled & Frozen		-		-		Edit	Delete
11	tes	st category		2022-07-29		2022-07-29		Edit	Delete
13	nev	w category		2022-08-12		2022-08-12		Edit	Delete
16	pos	stman		2022-08-20		2022-08-20		Edit	Delete
17	pos	stman		2022-08-20		2022-08-20		Edit	Delete
18	pos	stman		2022-08-20		2022-08-20		Edit	Delete

Figure 5.34: User interface design for managing category page

Manage Inv	entory Manage Warehouse Manage Category	Manage User View reports	\$			Hello, admin
Manage	Category	Add new category		×		
⊖ Add		Name Enter category nam	ne			
Show 10 entries		Cano	cel Confirm		Search:	
ID û	Category Name	Created At	Updated At	-	Action	
1	Drinks, Coffee & Tea				Edit	Delete
	Dry & Canned Goods				Edit	Delete
	Snacks				Edit	Delete
4	Chilled & Frozen				Edit	Delete
	test category				Edit	Delete
	new category	2022-08-12	2022-08-12		Edit	Delete
	postman				Edit	Delete
	postman	2022-08-20	2022-08-20		Edit	Delete
	nostman	2022-08-20	2022-08-20		Edit	Delete

Figure 5.35: User interface design for adding new category

# 5.4.2.3.3 Edit category

Manage	Inventory Manage Warehouse Manage Cate	gory Manage User View re	ports			Hello, admin
Manage	e Category	Edit Category		×		
Add     Show 10 entri	es	ID 1 Name Drinks, Coffee	& Tea		Search	
ID	Category Name	_	Cancel Confirm		Action	
1	Drinks, Coffee & Tea			_	Edit	Delete
2	Dry & Canned Goods				Edit	Delete
	Snacks				Edit	Delete
4	Chilled & Frozen				Edit	Delete
	test category				Edit	Delete
	new category	2022-08-12	2022-08-12		Edit	Delete
16	postman				Edit	Delete
17	postman		2022-08-20		Edit	Delete
18	postman	2022-08-20	2022-08-20		Edit	Delete

Figure 5.36: User interface design for editing category

Manage Inv	rentory Manage Warehouse Manage Category	Manage User Viev	/ reports		Hello, admin	
Manage	Category	Delete category	×			
		Are you sure to delete car	egory "Drinks, Coffee & Tea"?			
Add     Show 10 entries			Cancel Delete			
	Category Name	Created At	Updated At	a Action		
	Drinks, Coffee & Tea			Edit	Delete	
2	Dry & Canned Goods			Edit	Delete	
	Snacks			Edit	Delete	
4	Chilled & Frozen			Edit	Delete	
	test category	2022-07-29		Edit	Delete	
	new category	2022-08-12	2022-08-12	Edit	Delete	
	postman			Edit	Delete	
	postman	2022-08-20	2022-08-20	Edit	Delete	
18	postman	2022-08-20	2022-08-20	Edit	Delete	

Figure 5.37: User interface design for deleting category

#### 5.4.2.4 Manage User

The manage user page, user details page, add new user, update existing user and delete user interface design are shown in Figure 5.38, Figure 5.39, Figure 5.40, Figure 5.41 and Figure 5.42 respectively.

#### 5.4.2.4.1 Manage user page

N	lanage Inventory	Manage Warehouse	Manage Catego	ny Mana	ige User	View reports						Hello, admin
Man	age Use	r										
① Add												
Show 10	entries									Search:		
User ID	Name 0	Email	: C	ontact o.	Role	Warehouse	Employed in	Status	Created at	Updated at	Actions	
2	staff	staff@wims.com	12	23456789	Staff	Hapi Fresh Sdn Bhd	Jun-20	ACTIVE	-	2022-08- 02T03:11:13.000000Z	Edit	Delete
3	admin	admin@wims.com	12	23456789	Admin	Hapi Fresh Sdn Bhd	Jun-20	ACTIVE	-	2022-08- 19T08:41:26.000000Z	Edit	Delete
4	manager	manager@wims.com	12	23456789	Manager	Hapi Fresh Sdn Bhd	Jun-20	ACTIVE	-	-	Edit	Delete
5	GyDci74shp	f9EdyJfk2S@wims.com	12	23456789	Manager	GROCERY FRESH WAREHOUSE	Jun-20	ACTIVE	-	-	Edit	Delete
6	0vp1k0oa4O	tL3cbYmdY9@wims.com	12	23456789	Manager	ALI GROCERY SDN BHD	Jun-20	ACTIVE			Edit	Delete
7	xy7jTUNgpb	PbFib2Y6Ng@wims.com	01	123456789	Manager	PJ GROCERY WAREHOUSE	June 2020	ACTIVE	-	-	Edit	Delete
8	IZdinJ958H	cRMvZLWAWY@wims.com	n 01	123456789	Manager	KUALA LUMPUR WAREHOUSE	June 2020	ACTIVE		-	Edit	Delete

Figure 5.38: User interface design for manage user page

# 5.4.2.4.2 View user details

Manage Inventory	Manage Warehouse	Manage Category	Manage User	View reports			Hello, admin
View user							
ID Name Email Contact No IC No Role Warehouse Employed In Address Created At Updated At			2 staff staff@wims.com 123456789 123456789 123456787 Staff Hapi Fresh Sdn Bhd Jun-20 asdfsd - 2022-08-02T03:11:1	3.000000Z	Delete		

Figure 5.39: User interface design for viewing user details

5.4.2.4.3	Add	new	user
-----------	-----	-----	------

M				Manage L	ser View reports					Hello, admin
Man	age Use	r		Add new u	ser	×				
Inalia	age use	1		Name	Enter user name					
• Add				Email	Enter email (eg.abc@wims.com)					
Show 10				Contact No	Enter contact no. (eg.0123456789)			Search:		
User ID	Name 3	Email	Con No.	IC No	Enter IC no. (eg. 123456-12-1234)		Created at	Updated at	Actions	
	staff			Role	Select	~			Edit	Delete
	admin	admin@wims.com	1234	Warehouse	Select	~		2022-08- 19T08:41:26.000000Z	Edit	Delete
	manager	manager@wims.com		Employed in	Select date				Edit	Delete
	GyDci74shp	f9EdyJfk2S@wims.com	1234	Address	Enter address				Edit	Delete
	0vp1k0oa4O		1234			h			Edit	Delete
	xy7jTUNgpb	PbFib2Y6Ng@wims.com		Username	Enter username				Edit	Delete
					Cancel				Edit	Delete

Figure 5.40: User interface design for adding new user

# 5.4.2.4.4 Edit user

Ma	inage Inventory	Manage Warehouse N	lanage Category	Manage U	ser View reports				Hello, admin
Mana	age Use	r		Edit User		×			
				ID					
⊙ Add	1			Name					
Show 10	entries			Email	staff@wims.com			Search:	
User ID	Name 0	Email	Cont No.	Contact	123456789		Created at	Updated at	Actions 0
	staff							2022-08- 02T03:11:13.000000Z	Edit Delete
	admin	admin@wims.com	1234	Role		~		2022-08- 19T08:41:26.000000Z	Edit Delete
	manager	manager@wims.com	1234		Hapi Fresh Sdn Bhd	*			Edit Delete
	GyDci74shp	19EdyJfk2S@wims.com	1234	Address					Edit Delete
	0vp1k0oa4O			Address		h			Edit Delete
	xy7jTUNgpb			Username	staff				Edit Delete
8	IZdinJ958H	cRMvZLWAWY@wims.com	0123		Cancel Confirm				Edit Delete

Figure 5.41: User interface design for editing user

### 5.4.2.4.5 Delete user

Man	age Use	r	Delete	user			>	<			
	_		Are you	u sure to de	lete user 2 from Hapi F	resh Sdn Bhd ?					
⊕ Add					Cancel	Delete					
User ID	Name	Email	contact Io.	Role 3	Warehouse :	Employed in	Status 0	Created at	Search:	Actions	
	staff			Staff	Hapi Fresh Sdn Bhd		ACTIVE		2022-08- 02T03:11:13.000000Z	Edit	Delete
	admin	admin@wims.com		Admin	Hapi Fresh Sdn Bhd	Jun-20	ACTIVE		2022-08- 19T08:41:26.000000Z	Edit	Delete
	manager	manager@wims.com		Manager	Hapi Fresh Sdn Bhd		ACTIVE			Edit	Delete
		f9EdyJfk2S@wims.com		Manager			ACTIVE			Edit	Delete
	0vp1k0oa4O			Manager	ALI GROCERY SDN BHD		ACTIVE			Edit	Delete
	xy7jTUNgpb	PbFib2Y6Ng@wims.com	123456789	Manager		June 2020	ACTIVE			Edit	Delete
				Manager	KUALA LUMPUR WAREHOUSE	June 2020	ACTIVE			Edit	Delete

Figure 5.42: User interface design for deleting user

# 5.4.3 Manager

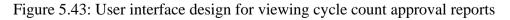
This section will show all modules that are available for user role manager, which includes the manage cycle count approval report module, manage cycle counting and view all staff modules.

# 5.4.3.1 Manage Cycle Count Approval Report

The Figure 5.43 displays the user interface design for viewing the list of cycle counting approval reports. Users can approve or reject the approval report in which the confirmation modal designs are shown in Figure 5.44 and Figure 5.45 respectively.

5.4.3.1.1	View reports-	cycle count	approval	reports
-----------	---------------	-------------	----------	---------

/iew Rej	port										
Daily Report	Cycle Cou	Inting Approval Report Cycle Count Summ	nary Report								
now 10 entries									Search:		
D Cycle Count	SKU ID	Inventory Name	Warehouse Name	Staff Name	Recorded Count	Actual Count	Date Counted	Approve Before	Variance 0	Action	
.1	1068	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	Hapi Fresh Sdn Bhd	staff	567	560	2022-08-27	2022-08-24	-7	Approve	Reject
1	1069	Starbucks Frappuccino Chilled Mocha Drink 281 ml	Hapi Fresh Sdn Bhd	staff	89	80	2022-08-27	2022-08-22	-9	Approve	Reject



View Re	port		Approve cycle	count repor	t		×			
Daily Report		nting Approval Report Cycle Count Summ	na 🥂 This			le count report? m count of Inventory I	D(1)		Search:	
Cycle Count	SKU ID	Inventory Name		c	Cancel Confi	m	ted	Approve Before	Variance	Action
41		Authentic Tea House Da Hong Pao Oolong Tea 300 ml	Hapi Fresh Sdn Bhd	staff	567	560	2022-08-27	2022-08-24		Approve Reject
	1069	Starbucks Frappuccino Chilled Mocha Drink 281 ml	Hapi Fresh Sdn Bhd	staff	89	80			-9	Approve Reject

Figure 5.44: User interface design for approving cycle count approval report

# 5.4.3.1.3 Reject cycle count approval report

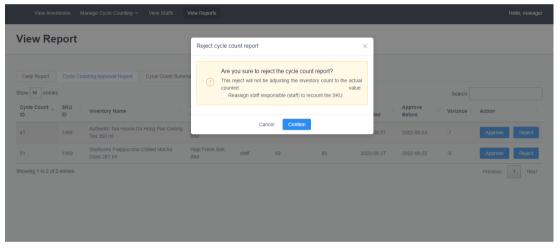


Figure 5.45: User interface design for rejecting cycle count approval report

# 5.4.3.2 Manage Cycle Counting

Managers can also manage the cycle counting by either starting a new cycle counting or viewing the current cycle counting schedules.

#### 5.4.3.2.1 Start cycle counting

The start cycle counting page interface design is shown in Figure 5.46. The manager can submit the form after filling in all of the information required and the system will automatically generate cycle counting schedules based on the inputs.

View Inventories Manage Cycle Counting View Staffs	View Reports	Hello, manager
Start Cycle Counting		
Working day of warehouse: Select Workday Start	to Select Workday End 🗸	
Counting frequency: Class A: Every 0	Solicet Workday	
Class B: Every 0	Select Workday •	
Class C: Every 0 Assign Staff(s):	Soliet Window •	
Assign staff		ĥ
Select Inventories(s): Select Inventories		
Select inventories		lê
Start and End date	Careel Submit	

Figure 5.46: User interface design for starting cycle counting

# 5.4.3.2.2 View cycle counting

Figure 5.47 shows the view cycle counting page that displays all cycle counting schedules generated. The warehouse manager can reassign all schedules assigned to one staff to another staff if the original staff is leaving his or her position as shown in Figure 5.48. Additionally, the manager can also view the current cycle counting settings and the assigned staff as well as selected SKUs as shown in Figure 5.49, Figure 5.50 and Figure 5.51 respectively.

View	Inventories Manage Cycle	counting ✓ View Staffs View Reports				Hello, manage
View C	Cycle Countir	ng				
Reassign st	taff					View cycle counting settings
Class A	Class B Class C					
Show 10 er	ntries				Search:	
SKU ID	Inventory ID	Inventory Name	Storage Number	Schedule	Days due	Staff assigned
1069	2	Starbucks Frappuccino Chilled Mocha Drink 281 ml	C8	2022-08-23	-3	testinf
1070	3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g	-	2022-08-24	-2	staff
1068	1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	C7	2022-08-25	-1	testinf
1070	3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g		2022-08-29	3	testinf
1068	1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	C7	2022-08-30	4	staff
1069	2	Starbucks Frappuccino Chilled Mocha Drink 281 ml	C8	2022-08-31	5	testinf
1070	3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g		2022-09-01	6	staff
1068	1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	C7	2022-09-02	7	testinf
1069	2	Starbucks Frappuccino Chilled Mocha Drink 281 ml	C8	2022-09-05	10	staff
1070	3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g		2022-09-06	11	testinf

# Figure 5.47: User interface design for view cycle counting page

View II	nventories Manage Cycle	e Counting Y View Staffs	/iew Reports					Hello, manager
View C	Sycle Countir	ng	Reassign staff		×			
Reassign st	alf Class B Class C		Select Select staff to replace Select		~			View cycle counting settings
Show 10 en		Inventory Name	staft Select		<u> </u>	Schedule	Search:	Staff assigned
1069	2		Cancel	Reassign	- 1	2022-08-23	-3	testinf
								staff
		Authentic Tea House Da Hong Pa	ao Oolong Tea 300 ml					testinf
		Yit Foh Instant 3 In 1 Premix Milk	Tea Teh Tarik (12 Sticks) 40 g					testinf
	1	Authentic Tea House Da Hong Pa	ao Oolong Tea 300 ml				4	staff
1069		Starbucks Frappuccino Chilled M	ocha Drink 281 ml	C8				testinf
		Yit Foh Instant 3 In 1 Premix Milk	Tea Teh Tarik (12 Sticks) 40 g				6	staff
1068		Authentic Tea House Da Hong Pa	ao Oolong Tea 300 ml					testinf
		Starbucks Frappuccino Chilled M	ocha Drink 281 ml	C8				staff
1070	3	Yit Foh Instant 3 In 1 Premix Milk	Tea Teh Tarik (12 Sticks) 40 g			2022-09-06	11	testinf

Figure 5.48: User interface design for reassigning staff

View I	Inventories Manage Cycl	e Counting 👻 View Staffs 🛛 Vie	ew Reports						Hello, managi
View C	Cycle Countii	ng	Current cycle co	unting settings		×			
	_		Period Working day	2022-08-21 - 202 monday to friday	2-09-10				
Reassign st	taff		Counting frequency	Class A	Class B	Class C			View cycle counting settings
Class A	Class B Class C			Every 3 day (s)	Every 3 week (s)	Every 3 mont h(s)			
now 10 er	ntries		Warehouse	Hapi Fresh Sdn B	hd			Search:	
SKU ID	Inventory ID	Inventory Name	Assigned Staff No. of SKU	2 3			Schedule	Days due	* Staff assigned
		Starbucks Frappuccino Chilled Mc	Group	Items	Frequency (within date	Daily Count			testinf
		Yit Foh Instant 3 In 1 Premix Milk			range)		2022-08-24	-2	staff
	1	Authentic Tea House Da Hong Pa	А	3	5	1.00			testinf
		Yit Foh Instant 3 In 1 Premix Milk	в	0	1	0.00			testinf
1068	1	Authentic Tea House Da Hong Pa	C	0	0.33333333	• • • •		4	staff
1069		Starbucks Frappuccino Chilled Mc	Total	3	5	1			testinf
		Yit Foh Instant 3 In 1 Premix Milk		_					staff
068	1	Authentic Tea House Da Hong Pa			ок			7	testinf
		Starbucks Frappuccino Chilled Mo	sha Drink 281 ml		CE	1			staff
		Yit Foh Instant 3 In 1 Premix Milk T							

Figure 5.49: User interface design for viewing cycle counting settings

View I	nventories Manage C	ycle Counting - Mew Statts - Me	w Reports						Hello, manager
View C	Cycle Count	ting	View Staff A	ssigned			×		
			Show 10 er Staff ID	ntries	Search: Name				
			2		staff				
			25		testinf				
		Inventory Name	Showing 1 to 2	2 of 2 entries	Pre	vious 1	Next		
		Statbucks Proposcono Chiled Ma			OK				
		Vit Fait Instant 5 In † Přenik Miller							
		Authoritic Tea House Da Hong Pa	A	3	5		i.		
		VIL Publicistant 3 In Y Premis Lask	в		1				
		Automic Tea House Dia Hong Fia					-		
		Starbucks Frappuccess Chilled Mc	Total						
		YE FOR Instant 3 In T. Preven Mak							
		Authenite Tes House Da Hong Pa			OK				
				N .		12.			

Figure 5.50: User interface design for viewing assigned staff for cycle counting

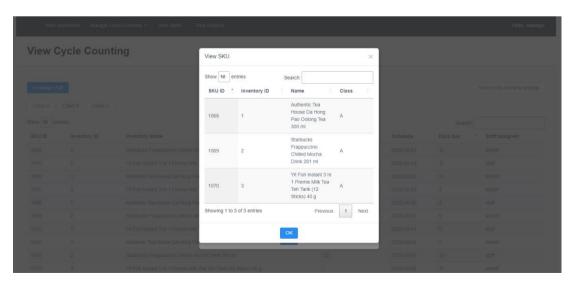


Figure 5.51: User interface design for viewing SKUs for cycle counting

#### 5.4.3.3 View All Staff

The Figure 5.52 shows the interface design of the view staff module, in which the manager's warehouse details and all staff working in the manager's warehouse are displayed.

View In	nventories Mana	ge Cycle Counting ~ View	w Staffs	View Reports						He	ilo, manage
/iew S	staffs										
Wareho	ouse Info										
ID	1					Name	Hapi	Fresh Sdn Bhd			
Location						Warehouse M	anager -				
Location	48, jal	an ss2/24 47300 petaling jaya	a selangor (	D.E, 47300 Petaling Ja	iya, Selangor	warehouse w					
Created by		an ss2/24 47300 petaling jaya	a selangor l	D.E, 47300 Petaling Ja	aya, Selangor	Updated by	-	-08-24T17:22:42.000000Z			
		an ss2/24 47300 petaling jaya	a selangor l	D.E, 47300 Petaling Ja	iya, Selangor		-	-08-24T17:22:42.000000Z			
Created by		an ss2/24 47300 petaling jayı	a selangor (	D.E, 47300 Petaling Ja	iya, Selangor		-	-08-24T17:22:42.000000Z	:	Search:	
Created by	y -	email	a selangor l	Contact No.	Employed in		-	08-24T17:22.42.000000Z Created at		Search:	
Created by	w -		a selangor l			Updated by	2022-		:		÷
Created by	ry - tries Name	Email	a selangor l	Contact No.	Employed in	Updated by	2022- Status	Created at	:	Updated at	
Created by	ry - tries Name staff	Email staff@wims.com	a selangor I	Contact No. : 123456789	Employed in Jun-20	Updated by	2022- Status : ACTIVE	Created at	: 1	Updated at 2022-08-02T03:11:13.000000	DZ
Created by how 10 ent User ID 1 2 25	tries staff testinf	Email staff@wims.com testing123@wims.com	a selangor I	Contact No. 123456789 23423423424	Employed in Jun-20 2022-08-09T16.00.0	Updated by 0.000Z	2022- Status : ACTIVE ACTIVE	Created at - 2022-08-10107:33:48.000000Z		Updated at 2022-08-02T03:11:13.000000 2022-08-11T23:39:57.000000	DZ DZ
Created by how 10 ent User ID = 2 25 26	v tries Name Staff testinf testiaff	Email staff@wims.com sd@wims.com	a selangor l	Contact No. 123456789 23423423424 1234567890	Employed in Jun-20 2022-08-09T16:00:0 2022-08-10T16:00:0	Updated by 0.000Z 0.000Z 0.000Z	2022- Status ACTIVE ACTIVE INACTIVE	Created at - 2022-08-10707-33-48.000000Z 2022-08-10708-32-05.000000Z		Updated at 2022-08-02T03:11:13.000000 2022-08-11T23:39:57.000000 2022-08-12T00:25:25.000000	DZ DZ
Created by how 10 ent User ID 1 2 25 26 29	w - Anne	Email staff@wims.com testing123@wims.com sd@wims.com postman@wims.com	a selangor l	Contact No.         I           123456789         I           23423423424         I           1234567890         I           0123433331         I	Employed In Jun-20 2022-08-09T16 00.0 2022-08-10T16 00.0 2022-08-11T16 00.0	Updated by 0.000Z 0.000Z 0.000Z	2022- Status 2 ACTIVE 2 INACTIVE 1 INACTIVE 2	Created at		Updated at 2022-06-02103 11: 13.000000 2022-08-11723 39.57.000000 2022-08-12100 25: 25.000000 2022-08-19108 38: 18.000000	DZ DZ DZ DZ

Figure 5.52: User interface design for viewing all staff

# 5.4.4 Staff

This section will list out all modules for the user role staff, which are the perform cycle counting and record inventory modules.

#### 5.4.4.1 Perform Cycle Counting

The Figure 5.53 shows the list of cycle counting schedules assigned to the staff. The staff can click the "Count" button on one of the cycle-counting schedules to perform cycle counting as shown in Figure 5.54 and Figure 5.55.

# 5.4.4.1.1 Cycle counting page

Cycle	Counting					
Upcoming	Pending Approval	Completed				
how 10 en	tries				Search:	
SKU ID	Inventory ID	Inventory Name	Storage Number	Schedule	Days due A	ction 0
1070	3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g	-	2022-08-24	-2	Count
1068	1	Authentic Tea House Da Hong Pao Colong Tea 300 ml	C7	2022-08-30	4	Count
1070	3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g	-	2022-09-01	6	Count
1069	2	Starbucks Frappuccino Chilled Mocha Drink 281 ml	C8	2022-09-05	10	Count
1068	1	Authentic Tea House Da Hong Pao Colong Tea 300 ml	C7	2022-09-07	12	Count
1070	3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g		2022-09-09	14	Count

Figure 5.53: User interface design for cycle counting page

# 5.4.4.1.2 Count SKU (no bin assigned)

ycie	Counting	Count	t SKU		×			
	Pending Approval	Completed	approach	signed! Kindly the manager. 230				
how 10 er	itries		Enter actual count				Search:	
SKU ID	Inventory ID	0 Inventory Name				Schedule	Days due	Action 3
		Yit Foh Instant 3 In 1 Premix Mill	Cancel	e		2022-08-24		Count
1068	1	Authentic Tea House Da Hong Pao Oolon	ig Tea 300 ml	C7			4	Count
		Yit Foh Instant 3 In 1 Premix Milk Tea Teh	Tarik (12 Sticks) 40 g				6	Count
1069		Starbucks Frappuccino Chilied Mocha Dri	ink 281 ml	C8				Count
		Authentic Tea House Da Hong Pao Oolon	g Tea 300 mi					Count
		Yit Foh Instant 3 In 1 Premix Milk Tea Teh	Tarik (12 Sticks) 40 g				14	Count

Figure 5.54: User interface design for counting SKU with no bin assigned

5.4.4.1.3 Count SKU	(with bin	n assigned)
---------------------	-----------	-------------

ycie	Counting		Count SKU		×			
Upcoming how 10 e		Completed	Bin Location: C System Count: 56 • Enter actual count				Search:	
SKU ID	Inventory ID	Inventory Name				Schedule	Days due	Action
		Yit Foh Instant 3 In 1 Premix Mil	Cancel Create		_	2022-08-24		Count
1068	1	Authentic Tea House Da Hong Pao	Oolong Tea 300 ml				4	Count
		Yit Foh Instant 3 In 1 Premix Milk To	ea Teh Tarik (12 Sticks) 40 g				6	Count
069	2	Starbucks Frappuccino Chilled Mod	ha Drink 281 ml	C8				Count
	1	Authentic Tea House Da Hong Pao	Colong Tea 300 ml					Count
		Yit Foh Instant 3 In 1 Premix Milk To					14	Count

Figure 5.55: User interface design for counting SKU with bin assigned

#### 5.4.4.2 Record Inventory

On the check-in/ out stock page, the list of inventories in the staff's warehouse is displayed in Figure 5.56. The staff can choose one of the inventories to check in or check out some stocks. The checking in and checking out stock modal is as shown in Figure 5.57 and Figure 5.58 respectively.

### 5.4.4.2.1 Check in/out stock page

Check In/Out Stock	View Inventory Cycle Counting							Hello,
Show 10 entrie Inventory ID	s Name :	Cost per Unit	Quantity On Hand	Storage Bin Number	Category ID	Created By	Search: Updated By	· Action ·
1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	3.5	560	C7	Dry & Canned Goods			Check In/Out Stock
2	Starbucks Frappuccino Chilled Mocha Drink 281 ml	13	89	C8	Dry & Canned Goods			Check In/Out Stock
3	Yil Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g	13.2	230	-	-			Check In/Out Stock
4	Grante 100% Pomegranate, Grape And Apple Juice 250 ml	6.6	40	A4	Drinks, Coffee & Tea			Check In/Out Stock
5	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets) 25 g	14.15	89	A5	Drinks, Coffee & Tea			Check In/Out Stock
6	Ma Ma Mi Roasted Robusta Ground Coffee 250 g	24.6	103	A6	Drinks, Coffee & Tea			Check In/Out Stock
7	Coffee Cherry Roasted Robusta Coffee Beans 250 g	24.6	70	A7	Drinks, Coffee & Tea			Check In/Out Stock
8	Delica De Mountain Black Coffee Blend (10 Pieces) 10 g	12.6	15	A8	Drinks, Coffee &			Check In/Out Stock

Figure 5.56: User interface design for check in/out stock page

### 5.4.4.2.2 Check in stock

Check In/ Out Stock     Check In/ Out Inventory -Authentic Tea House Da Hong Pao Oolo.x     Inventory ID:     1   Guantity On Hand:   50     Name     1   Quantity O   1   Authentic Tea House Da Hong Pao Oolon X     1   2   Starbucks Frappuccino Chilled Mocha Dink 281 ml     2   3   Yil Foh Instant 3 in 1 Premix Maik: Tea Teh Tarik (12 Stick)   4   Grante 100%: Pomegranate, Grape And Apple Julice 250 ml   4   5   Delica 2 in 1 Instant Premix (b) White Cotee (16 Packets) 25 g)   14   5	
Brow 10 entries     Cuantity On Hand: 560     Check Out     y ID     Created By     Updated By       1     Authentic Tea House Da Hong Pao Oolong Tea 300 ml     Check In     Check Out     undity     Inned       2     Starbucks Frappuccino Chilled Mocha Drink 281 ml     Cancel     Continue     Inned     Inned       3     Yil Poh Instant 3 in 1 Premix Mik Tea Teh Tarik (12 Sticks     Cancel     Continue     Context     Inned       4     Grante 100%, Pomegranate, Grape And Apple Juce 250 m     6.5     40     r.4     Tea       5     Dallea 3 in 1 Internet with Mage College (5 Bardeta 26 n)     14 15     19     45     Dinks, Coffee &	
Inventor     Name       1     Authentic Tea House Da Hong Pao Colong Tea 300 ml       2     Starbucks Frappuccino Chilled Mocha Drink 281 ml       3     Yill Foh Instant 3 in 1 Premix Milk Tea Teh Tarik (12 Sticks       4     Grante 100% Pomegranate, Grape And Apple Juce 250 ml       5     Dalleg 2 in 1 Instant Stin 1 Streme holds Mocha Chilled (5 Backets) 26.0	
Authentic Tea House Da Hong Pao Colong Tea 300 ml     Authentic Tea House Da Hong Pao Colong Tea 300 ml     Remarks     Enter remarks     Origin     Starbucks Frappuccino Chilled Mocha Dirik 281 ml     Origin     Cancel     Contim     Cancel     Contim     Contee &     Contee &     Derice 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     16 16      Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 In 1 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 Internet Premix Keit Mothe Coffee (16 Particity 36 m     Dates 3 Internet Premix Keit Mothe Coffee (1	C Action C
2     Starbucks Frappuccino Chilled Mocha Drink 281 ml     0191g     anned       3     Yil Foh Instant 3 in 1 Premix Milk Tea Teh Tarik (12 Slicks     Cancel     Contime       4     Grante 100% Pomegranate, Grape And Apple Juce 250 ml     6.6     40     r.4       5     Dallea 3 in 1 Instant On Multice College (15 Backston 26 ml     14 15     89     45	Check In/Out Stock
Cancel         Control         Control           4         Grante 100% Pomegranate, Grape And Apple Juce 250 hr         6.6         40         A4         Tea           5         Dates 2 in 1 tested Dommin tonb Mittle Ordina //5 Rackath 26.0         14.15         80         A5         Drinks, Coffee &	Check In/Out Stock
4         Grante 100% Pomegranate, Grape And Apple Juice 250 mil         6.6         40         A4         Tea           5         Dalka 2 in 4 Instant Domini koh Witte Octlea (45 Darketh 25 c         14.15         89         A5         Drinks, Coffee &	Check In/Out Stock
	Check In/Out Stock
	Check In/Out Stock
6 Ma Ma Mi Roasted Robusta Ground Coffee 250 g 24.6 103 A6 Tea	Check In/Out Stock
7 Coffee Chemy Roasted Robusta Coffee Beans 250 g 24.6 70 A7 Tea	Check In/Out Stock

Figure 5.57: User interface design for checking in stock

### 5.4.4.2.3 Check out stock

Check In/Out Stoc	k View Inventory Cycle Counting						Hello, staff
Check In	/ Out Stock	Check In/ Out Inv	entory - Authentic	Tea House Da Hong Pa	ao Oolo.×		
Show 10 entrie		Inventory ID: 1 Quantity On Hand: 560 Check In Ch	eck Out		ny ID	Search: Created Updated By By	Action
1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	Quantity 0			anned	бу бу	Check In/Out Stock
2	Starbucks Frappuccino Chilled Mocha Drink 281 ml	Remarks Enter	remarks	0/191 <sub>6</sub>	anned		Check In/Out Stock
	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks		Cancel	Confirm			Check In/Out Stock
4	Grante 100% Pomegranate, Grape And Apple Juice 250 m	ni 6.6	40	A4	Coffee & Tea		Check In/Out Stock
5	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets	s) 25 g 14.15	89	A5	Drinks, Coffee & Tea		Check In/Out Stock
6	Ma Ma Mi Roasted Robusta Ground Coffee 250 g	24.6		A6	Drinks, Coffee & Tea		Check In/Out Stock
7	Coffee Cherry Roasted Robusta Coffee Beans 250 g	24.6		A7	Drinks, Coffee & Tea		Check In/Out Stock
8	Delica De Mountain Black Coffee Blend (10 Pieces) 10 g	12.6	15	A8	Drinks, Coffee &		Check In/Out Stock

Figure 5.58: User interface design for checking out stock

# 5.4.5 Staff & Manager

This section outlines the view all inventories module that is available for staff and manager.

# 5.4.5.1 View All Inventories

The Figure 5.59 shows the user interface design for the view inventory page, whereas the Figure 5.60 shows the inventory details page when the user clicked on the inventory ID on the view inventory page.

# 5.4.5.1.1 View inventory page

iew Inv	entory					
	-					
ow 10 entries					Search:	
nventory ID 🙏	Name	Cost Per Unit	Quantity On Hand	Storage Bin Number	Created By	Updated By
	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	3.5	567	C7	-	-
	Starbucks Frappuccino Chilled Mocha Drink 281 ml	13	89	C8		-
	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g	13.2	230	-	-	-
	Grante 100% Pomegranate, Grape And Apple Juice 250 ml	6.6	40	A4		-
	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets) 25 g	14.15	89	A5		-
	Ma Ma Mi Roasted Robusta Ground Coffee 250 g	24.6	103	A6	-	-
	Coffee Cherry Roasted Robusta Coffee Beans 250 g	24.6	70	A7	-	-
	Delica De Mountain Black Coffee Blend (10 Pieces) 10 g	12.6	15	A8		-
	Nutrigold 3 In 1 Instant Premix Coffee Rich (25 Sticks) 20 x 20	11.7	98	A9	-	-
0	F&N Seasons Ice Peach Tea Drink 1 L	3.95	142	A10	-	-

Figure 5.59: User interface design for view inventory page

# 5.4.5.1.2 View inventory details

View	Inventories Man	age Cycle Counting V View Staffs View Reports							Hello, man	nager
Inventory	Info									
ID	1			Quantity on hand	567					
Name	Authentic	Tea House Da Hong Pao Oolong Tea 300 ml		Storage bin	C7					
Warehouse	Hapi Fres	h Sdn Bhd		Created at	-					
Cost per unit	3.5			Updated at	-					
Show 10 entries	5						Search:			
Stock ID 1	Inventory ID	Inventory Name	÷ (	Quantity in/out	Date		Staff Responsible		Remarks	
3	1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml		1	2022-06-05T01:01:51.000000Z		staff		1	į
4	1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml		4	2022-06-05T01:13:54.000000Z		staff		1	
5	1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml		4	2022-06-05T01:13:57.000000Z		staff		1	
6	1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml		1	2022-06-05T01:22:39.000000Z		staff		1	
7	1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	0	D	2022-06-05T01:25:21.000000Z		staff		1	
Showing 1 to 10 of	f 187 entries		Edit	Delete		Previous	1 2 3 4	5	19	Next

Figure 5.60: User interface design for view inventory details

#### 5.4.6 Admin & Manager

This section lists the modules made available for both user roles: admin and manager, which is the view reports module that allow them to view the daily reports and the cycle counting summary reports.

#### 5.4.6.1 View Reports

Figure 5.61 shows the list of daily reports on the view report page. When the user clicks the "View" button, the selected daily report details will be displayed as shown in Figure 5.62. Figure 5.63 and Figure 5.64, on the other hand, show the list of cycle counting summary reports and the selected summary report details respectively.

#### **5.4.6.1.1** View reports- daily reports

View Inventories Manage Cycle	Counting v View Staffs View Reports		Hello, manager
View Report			
Daily Report Cycle Counting Approv	val Report Cycle Count Summary Report		Search:
Warehouse ID	â Warehouse Name	Date	Action 0
1	Hapl Fresh Sdn Bhd	2022-07-26	View
1	Hapi Fresh Sdn Bhd	2022-06-05	View
1	Hapi Fresh Sdn Bhd	2022-06-06	View
1	Hapi Fresh Sdn Bhd	2022-06-09	View
1	Hapi Fresh Sdn Bhd	2022-06-13	View
1	Hapi Fresh Sdn Bhd	2022-06-22	View
1	Hapi Fresh Sdn Bhd	2022-07-02	View
1	Hapi Fresh Sdn Bhd	2022-07-29	View
1	Hapi Fresh Sdn Bhd	2022-08-01	View

Figure 5.61: User interface design for viewing daily reports

# 5.4.6.1.2 View daily report details

View Inventories. Manage Cycle Counting ~	View Staffs	View Repo	orts					Hello, manager
View Report	Daily Rep	ort						1
Daily Report Cycle Counting Approval Report Show 19 entries Warehouse ID	Warehouse Warehouse Date of Rep Show 10 Stock	Name ort entries Inventory	202 Inventory	i Fresh Sdn Bho 2-07-26 Quantity	Searc	Staff	Remarks	Search
	1D	3	Name Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tanik (12 Sticks) 40 g	iniout 100	2022-07- 26T08:32:57.000000Z	staff	sdfasdfasdf	Ver
	Showing 11	o 1 of 1 entries				Previous	1 Next	View
				Q	к			View
								View
								View
		api Fresh Sdn Bh						View
								View

Figure 5.62: User interface design for viewing daily report details

# **5.4.6.1.3** View reports- cycle count summary reports

View Inventories	Manage Cycle Cour	nting 🗸 View Staffs	View Reports				Hello, manager
View Report							
Daily Report Cycle C	ounting Approval Re	port Cycle Count Sur	nmary Report				
Show 10 entries						Search:	
Cycle Count ID	SKU ID	Inventory Name		Warehouse Name	Staff Name	Date Counted	î IRA(%) î
41	1068	Authentic Tea House Da	I Hong Pao Oolong Tea 300 ml	Hapi Fresh Sdn Bhd	staff	2022-08-27	0.988
Showing 1 to 1 of 1 entries							Previous 1 Next

Figure 5.63: User interface design for viewing cycle count summary reports

View Inventories Manage Cycle Counting ~ View Staffs Vie	w Reports	Hello, manager
View Report	W Reports Summary Report Details × × SKU ID 1068 Cycle Count ID 41 Inventiony ID 1 Inventiony ID 1 Warehouse ID 1 Warehouse ID 1 Warehouse Name Hapi Fresh Stn Bhd Staff ID 2 Staff Name staff Date Counted 2022-08-27 Inventory Record Accuracy	Helo, manager
	Recorded Count 567 Actual Count 560 Variance -7 Absolute Variance 7 IRA = 0.988 %	

5.4.6.1.4 View cycle count summary report details

Figure 5.64: User interface design for viewing cycle count summary report details

### 5.5 Summary

In short, this chapter discussed the system architecture diagram, which detailed out the communication between the frontend and the backend as well as the role of Pusher in the system architecture on the real-time feature by broadcasting API messages to all clients once an event is created from the backend. Modelling diagrams that include data flow diagrams and interface flow diagrams, as well as the screenshots of implemented system's user interface design are also outlined and arranged according to the user roles for easier understanding purposes.

#### **CHAPTER 6**

#### SYSTEM IMPLEMENTATION

#### 6.1 Introduction

This chapter documents the implementation details of the proposed system. The chapter divides into three parts. The first part is the backend part which includes the list of REST APIs used in the implemented system, the JWT authentication which secures the REST APIs and the Pusher Channel implemented for the real-time functionality. The second part covers all third-party user interface (UI) and JavaScript libraries used in frontend and the third part is the deployment of the implemented system to Heroku so that it can be accessed by user acceptance and usability testers later on.

# 6.2 Backend

This section will cover the backend setup of the implemented system, which will include the API List, JWT authentication to secure the REST APIs and the Pusher Channels setup for the real-time functionality.

#### 6.2.1 API List

For the communication between the implemented system and the database, REST APIs are used extensively for every request. The implemented system will call an API to view, create, update or delete data from the database. All API endpoints used in the implemented system are listed in Table 6.1.

Table 6:1: List of API endpoints

Endpoints	Method	Description	Parameters required
/api/login?username={username}&password	POST	To login into the system	1. Username
={password}			2. Password
/api/reset-	POST	To reset the password	1. New password
password?new_password={new_password}			2. Confirm password
&confirm_password={confirm_password}"			
/api/logout	POST	To logout from the system	
/api/inventories	GET	To retrieve the list of inventories	
/api/inventory/{id}	GET	To retrieve an inventory's details	Inventory ID
/api/getStocksByInventory/{id}	GET	To retrieve all stocks of an inventory	Inventory ID
/api/inventories-unassigned-category	GET	To retrieve all inventories with no	
		category	
/api/inventory?name={name}&warehouse_id	POST	To add a new inventory	1. Name
={warehouse_id}			2. Warehouse ID
&qty_on_hand={qty_on_hand}			3. Quantity on hand
&cost_per_unit={cost_per_unit}			4. Cost per unit
&priority={priority}			5. Priority
/api/inventory/{inventory_id}?name={name}	PUT	To update an inventory	1. Inventory ID
&category_id={category_id}			2. Name
&cost_per_unit={cost_per_unit}			3. Category
&priority={priority}			4. Cost per Unit
			5. Priority
/api/inventory/{id}	DELETE	To delete an inventory	1. Inventory ID
/api/warehouses	GET	To retrieve the list of warehouses	
/api/warehouse/{id}	GET	To retrieve a warehouse's details	1. Warehouse ID
/api/assign-category-to-bin/{id}	POST	To assign category to a warehouse's bin	1. Category ID
1 0 0 0 0 0 0 0 0 0			2. List of storage bin numbers
			to be assigned

/api/storage-bin-edit- inventory/{warehouse_id}?bin_id={bin_id} &inventory_id={inventory_id} &category_id={category_id}	POST	To edit the inventory in a storage bin	<ol> <li>Warehouse ID</li> <li>Storage bin ID</li> <li>Category ID</li> <li>Inventory ID</li> </ol>
/api/warehouse	POST	To add a new warehouse	1. Warehouse name2. Warehouse location3. Storage bins
/api/warehouse/{id}?location={location}&ma nager_id={manager_id}	PUT	To update a warehouse	<ol> <li>Warehouse ID</li> <li>New warehouse location</li> <li>New manager ID</li> </ol>
/api/warehouse/{id}	DELETE	To delete a warehouse	1. Warehouse ID
/api/users	GET	To retrieve a list of users	
/api/active-staffs	GET	To retrieve only active staffs	
/api/user/{id}	GET	To retrieve a user detail	1. User ID
<pre>/api/user?name={name}&amp;email={email}&amp;co ntact_no={contact_no} ⁣_no={ic_no} &amp;role={role}&amp;warehouse_id={warehouse_id} } &amp;employed_in={employed_in} &amp;address={address} &amp;username={username}</pre>	POST	To add a new user	<ol> <li>User's name</li> <li>User's email</li> <li>User's Contact number</li> <li>User's IC number</li> <li>User's role</li> <li>User's varehouse ID</li> <li>User's employed date</li> <li>User's address</li> <li>User's username</li> </ol>
/api/user/{id}?email={email} &contact_no={contact_no}&role={role}&wa rehouse_id={warehouse_id}&address={addre ss}	PUT	To update a user	<ol> <li>User ID</li> <li>User's email</li> <li>User's Contact number</li> <li>User's role</li> <li>User's warehouse ID</li> </ol>

			6. User's address
/api/user/{id}	DELETE	To delete an existing user	
/api/categories	GET	To retrieve a list of categories	
/api/category?name={name}	POST	To add a new category	1. New category's name
/api/category/{category_id}?name={name}	PUT	To update a new category	<ol> <li>Desired category's ID</li> <li>New category name</li> </ol>
/api/category/{category_id}	DELETE	To delete a category	1. Desired category's ID
/api/stocks	GET	To retrieve all inventory stocks	
/api/cycle-counts	GET	To retrieve all cycle count records	
<pre>/api/approve-cycle- count?cycle_counting_id={cycle_counting_id } &amp;ira={ira}&amp;inventory_id={inventory_id} &amp;variance={variance}</pre>	POST	To approve a cycle count	<ol> <li>Cycle counting ID</li> <li>IRA</li> <li>Inventory ID</li> <li>Variance</li> </ol>
<pre>/api/reject-cycle- count?cycle_counting_id={cycle_counting_id } &amp;recount={recount} &amp;schedule_date={schedule_date}</pre>	POST	To reject a cycle count	<ol> <li>Cycle counting ID report</li> <li>Recount</li> <li>Schedule date</li> </ol>
<pre>/api/storeCycleCountingSettings/{warehouseI d}?working_day_start={working_day_start} &amp;working_day_end={working_day_end} &amp;cycle_count_class={cycle_count_class} &amp;warehouse_id={warehouse_id} &amp;staff_ids={staff_ids} &amp;inventory_ids={inventory_ids} &amp;start_end_date={start_end_date}</pre>	PUT	To update the cycle counting settings of a warehouse	<ol> <li>Warehouse ID</li> <li>Starting of working day</li> <li>Ending of working day</li> <li>Cycle count classes information</li> <li>warehouse ID</li> <li>staff ids</li> <li>inventory ids</li> <li>start and end date</li> </ol>
/api/sku	POST	To store all SKUs into the database	Array of SKUs

/api/schedule	POST	To store all cycle count schedules into the database	Array of schedules
/api/skus	GET	To retrieve a list of SKUs	
/api/schedules	GET	To retrieve a list of cycle count schedules	
/api/reassignStaff	POST	To reassign all schedules allocated from one staff to another staff	<ol> <li>Array of schedule ids</li> <li>New staff ID</li> </ol>
<pre>/api/cycle-count?schedule_id={schedule_id} &amp;actual_count={actual_count} &amp;recorded_count={recorded_count} &amp;variance={variance}&amp;inv_rec_accuracy={i nv_rec_accuracy} &amp;status={status}</pre>	POST	To create a new cycle count record	<ol> <li>Cycle Count Schedule ID</li> <li>Actual count</li> <li>Recorded count</li> <li>Variance</li> <li>Inventory record accuracy calculated</li> <li>Status</li> </ol>
/api/stock?inventory_id={inventory_id} &mode={mode} &quantity={quantity}&remarks={remarks}	POST	To create a stock record (Check in/ check out)	<ol> <li>Inventory ID</li> <li>Check In/Out Mode</li> <li>Quantity</li> <li>Remarks</li> </ol>

#### 6.2.2 Secure the REST API

JWT Web tokens are an open way to represent users' identities during a two-party communication. For example, when two systems are exchanging data, the JWT token can be used to identify the user without having to send private data for every communication. For REST API context, the user can login to the system and once the user credentials are verified, the JWT Token will be created and returned to the web application (Doglio, 2021). Then, the web application can use the JWT Token on every request so that only authenticated users can access the APIs. In this way, the APIs can be secured while not having to send private data on every request to API. The overall mechanism of JWT is shown in Figure 6.1.

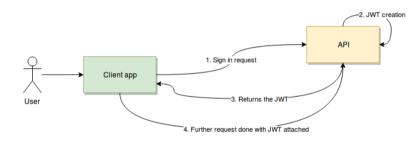


Figure 6.1: Overall mechanism of JWT (Doglio, 2021)

#### 6.2.3 Pusher Channel

Pusher Channels facilitates the communications between the server, app and devices. When an event happens on the app, the app will notify all other clients using the app about the changes. In this project, when a staff checks in or checks out an inventory stock, the Pusher Channel will broadcast the event to other staff that are using the system. This means that other staff can immediately see the changes without refreshing the page. Figure 6.2 shows the overview of the Pusher Channel dashboard.

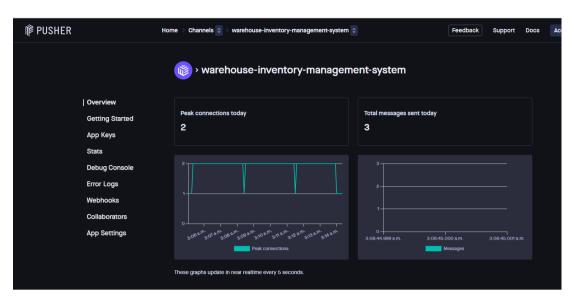


Figure 6.2: Overview of Pusher Channels dashboard

When two devices are using the system, both of them will be subscribed to the channel "check-in-out-stock.<warehouse\_id>. The debug console in the Pusher Channel dashboard will show the events of the channel subscription in Figure 6.3. If one of the users is checking in or out stock, an event will be created which will be broadcasted by the pusher as an API message with the updated inventory object as shown in Figure 6.4 and an event will be created. In this case, this API message will be sent to both users that are subscribed to the channel.

EVENT	DETAILS	TIME
Subscribed	Channel: check-in-out-stock.1 Socket ID: 17351.49365224	19:17:19
Connection	Origin: https://fyp-wims.herokuapp.com Socket ID: 17351.49365224	19:17:19
Occupied	Channel: check-in-out-stock.1	19:17:07
Subscribed	Channel: check-in-out-stock.1 Socket ID: 17302.49373716	19:17:07
Connection	Origin: http://fyp-wims.herokuapp.com Socket ID: 17302.49373716	19:17:07

Figure 6.3: Events created for channel subscription

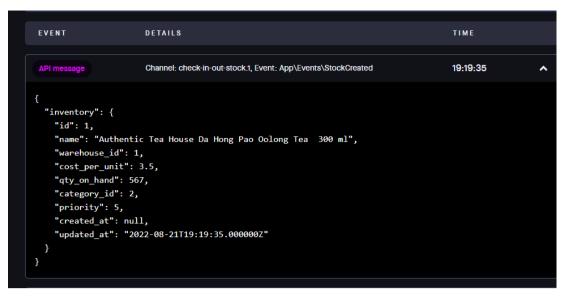


Figure 6.4: API message broadcasted to all subscribers of the channel.

# 6.3 Frontend

This section will include the user interface libraries and JavaScript libraries used in the frontend development of the implemented system.

# 6.3.1 UI Libraries

IView UI library is used in this project for most of the UI components. This library provides more than 80 base components that are mainly used for enterprise-level middle and background systems. In this project, all buttons, forms, inputs, modals, alerts, notices and tabs are integrated from this library. An example of a notice component utilized in the web application is shown in Figure 6.5. Besides, this library also allows the developer to customize the components to suit their needs. For example, some modals' footer in the web-based warehouse inventory system is customized so that only one button is displayed as shown in Figure 6.6.

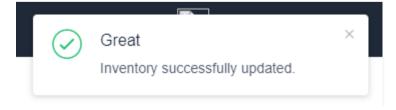


Figure 6.5: Success notice component

π									
	Daily Report								
le Count (	Warehouse ID Warehouse Nam Date of Report Show 10 entri		1 Hapi Fresh Sdn Bhd 2022-07-26		Search:				
	Stock ID	Quantity in/out	Date	÷	Staff Res	sponsible 🔅	Remar	rks 🌐	Actio
	86	100	2022-07-26T08:32:57.000000Z		staff		sdfasd	fasdf	
	Showing 1 to 1 o	f 1 entries				Previous	1	Next	Vie
									Vie
			ок						Vie

Figure 6.6: Modal with customized footer.

Another UI library used in this project is the TailwindUI. TailwindUI is also a UI library that provides beautifully designed components and templates. The login page and the top navigation bar components from the TailwindUI are used in this project as shown in Figure 6.7 and Figure 6.8. Besides, it also provides utilities that include layouts, flexbox and grid, spacing, sizing, typography and many more which are utilized in the project.

	<b>.</b>
Sign in to	your account
Username	
Password	
Remember me	Forgot your password?
<b>A</b>	Sign in

Figure 6.7: Login component

	lanage Inventory Manage Warehouse	Manage Category	Manage U	ser View rep	orts				
Manag	e Inventory								
Add     Show 18 ent	tries				~		Search:		
Inventory ID	Name	Warehouse	Cost Per 0 Unit	Quantity On Hand	Storage Bin Number	Created By	Updated By	Actions	
1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	Hapi Fresh Sdn Bhd	3.5	560	C7	-	-	Edit	Delete
2	Starbucks Frappuccino Chilled Mocha Drink 281 ml	Hapi Fresh Sdn Bhd	13	89	C8	-	-	Edit	Delete
3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g	Hapi Fresh Sdn Bhd	13.2	230	-	-	-	Edit	Delete
4	Grante 100% Pomegranate, Grape And Apple Juice 250 ml	Hapi Fresh Sdn Bhd	6.6	40	A4	-		Edit	Delete
5	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets) 25 g	Hapi Fresh Sdn Bhd	14.15	89	A5	-		Edit	Delete
6	Ma Ma Mi Roasted Robusta Ground Coffee 250 g	Hapi Fresh Sdn Bhd	24.6	103	A6	-	-	Edit	Delete
7	Coffee Cherry Roasted Robusta Coffee	Hapi Fresh Sdn	24.6	70	A7			Edit	Delete

Figure 6.8: Navigation bar component

#### 6.3.2 JS Libraries

In this web application, there are a lot of tables that are used to display data retrieved from the database. This is done by the DataTables, which is a plug-in for the jQuery JavaScript library. It is a highly flexible tool which adds advanced features to basic HTML tables. For example, DataTables allows pagination, instant search, multi-column ordering and many more just in several lines of code. There is also a more advanced feature provided by DataTables that is utilized in this project such as filtering the rows by column. This is used in filtering the list of inventories by warehouse and filtering the warehouse's storage bins by category as shown in Figure 6.9 and Figure 6.10.

			Hapi Fresh	Sdn Bhd	/				
now 10 entri	es						Search:		
nventory D	Name 0	Warehouse 🗍	Cost Per : Unit	Quantity On Hand	Storage Bin Number	Created By	Updated By	Actions	
1	Authentic Tea House Da Hong Pao Oolong Tea 300 ml	Hapi Fresh Sdn Bhd	3.5	560	C7	-	-	Edit	Delete
2	Starbucks Frappuccino Chilled Mocha Drink 281 ml	Hapi Fresh Sdn Bhd	13	89	C8	-	-	Edit	Delete
3	Yit Foh Instant 3 In 1 Premix Milk Tea Teh Tarik (12 Sticks) 40 g	Hapi Fresh Sdn Bhd	13.2	230	-	-	-	Edit	Delete
1	Grante 100% Pomegranate, Grape And Apple Juice 250 ml	Hapi Fresh Sdn Bhd	6.6	40	A4	-	-	Edit	Delete
5	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets) 25 g	Hapi Fresh Sdn Bhd	14.15	89	A5	-	-	Edit	Delete
5	Ma Ma Mi Roasted Robusta Ground Coffee 250 g	Hapi Fresh Sdn Bhd	24.6	103	A6	-	-	Edit	Delete
7	Coffee Cherry Roasted Robusta Coffee Beans 250 g	Hapi Fresh Sdn Bhd	24.6	70	A7	-	-	Edit	Delete

Staffs	Storage Bins							
Multi Ass	⊙ Multi Assign Bin Drinks, Coffee & Tea ∨							
Show 10 er	ntries			Search:				
Bin ID 🗍	Bin Number	Category ID	Inventory ID		C Action C			
2	A2	Drinks, Coffee & Tea	test add2		Edit Inventory			
4	A4	Drinks, Coffee & Tea	Grante 100% Pomegranate, Grape And Apple Juice 250 ml		Edit Inventory			
5	A5	Drinks, Coffee & Tea	Delica 2 In 1 Instant Premix Ipoh White Coffee (15 Packets) 25 g		Edit Inventory			
6	A6	Drinks, Coffee & Tea	Ma Ma Mi Roasted Robusta Ground Coffee 250 g		Edit Inventory			
7	A7	Drinks, Coffee & Tea	Coffee Cherry Roasted Robusta Coffee Beans 250 g		Edit Inventory			
8	A8	Drinks, Coffee & Tea	Delica De Mountain Black Coffee Blend (10 Pieces) 10 g		Edit Inventory			
9	A9	Drinks, Coffee & Tea	Nutrigold 3 In 1 Instant Premix Coffee Rich (25 Sticks) 20 x 20		Edit Inventory			
10	A10	Drinks, Coffee & Tea	F&N Seasons Ice Peach Tea Drink 1 L		Edit Inventory			
11	A11	Drinks, Coffee & Tea	Nutrigold 3 In 1 Regulart Instant Premix Coffee (30 Sticks) 20 g		Edit Inventory			
12	A12	Drinks, Coffee & Tea	F&N Seasons Ice Lemon Tea Drink 1 L		Edit Inventory			

#### Figure 6.9: Filtering list of inventories by warehouse

Figure 6.10: Filtering a warehouse's storage bins using category

Lodash, on the other hand, is used in this project for working with arrays, numbers, objects, strings, etc. It removes the hassle of dealing with them with plain JavaScript. For instance, there is a map function, which allows us to modify each element's value in an array with any value easily. Besides, the chain function can process any elements passed to it with several functions in one go. This can effectively reduce the lines of codes which can also improve the readability.

Moreover, momentJS is also used to parse, validate, manipulate and display the date in JavaScript. It is convenient to use momentJS to format the date from the database with UTC format to a fixed format or a more readable format before displaying out in the user interface. For instance, there is a global function created in the project, which uses momentJS to formate the date object before displaying it in the table cell.

### 6.4 Deployment to Heroku

To deploy the web application, Heroku is used to manage and automate the deployment. Heroku is a container-based cloud platform as a service (PaaS) owned by Salesforce that helps developers to deploy, manage and scale modern apps without being distracted by maintaining the servers, hardware or infrastructures. Heroku allows developers to create software up to a certain size and publish non-commercial apps such as proof of concepts, personal projects, etc for free.

Before deploying the developed web application to Heroku, the Git and Heroku CLI are installed beforehand and a Heroku account is created. Then, a new app can be created on Heroku and connected to the GitHub repository for the deployment method. By doing this, the changes made on GitHub will also automatically initiate Heroku to restart the server according to the necessary changes and republish it. The automatic deployment is also enabled so that every push to the master branch will be automatically deployed to the server. A file name Procfile is created to specify the public directory as the document root directory since this web application is built on the Laravel framework. Both PHP and node.js buildpack are added to compile both PHP and Vue.js.

Then, the ClearDB MySQL add-on provided by Heroku is used as the resource of the Heroku app to create the database instance. To connect with the MySQL database created, MySQL Workbench is installed. MySQL Workbench is a unified visual tool for data modelling, SQL development and an administration tool to configure servers, backup and many more. The connection can be added using MySQL Workbench to connect with the database instance created by using the connection parameters provided by the database instance. After the database is successfully connected, the complete database with tables and records done in PHPMyAdmin is exported out in SQL file format and imported into the database connected in MySQL Workbench. Lastly, the connection parameters and other necessary parameters such as app key, app URL and JWT secrets as shown in Figure 6.11 are added to the config vars of the Heroku app. Figure 6.12 shows the overview of the Heroku app with the ClearDB MySQL add-on installed. The deployed web application is shown in Figure 6.13.

Co	g٧	/a	rs

APP_DEBUG
APP_ENV
APP_KEY
APP_NAME
APP_URL
CLEARDB_DATABASE_URL
DB_CONNECTION
DB_DATABASE
DB_HOST
DB_PASSWORD
DB_PORT
DB_USERNAME
JWT_SECRET
PUSHER_APP_ID
PUSHER_APP_KEY
PUSHER_APP_SECRET
PUSHER_APP_CLUSTER

Figure 6.11: Config vars of the Heroku app

H HEROKU	Jump to Favor	ites, Apps, Pipelines, Spaces	
	Personal 0 > (0) fyp-wims     GitHub Q elainelize/vides      Immin	☆ Open app ) More ◊	
	Overview Resources Deploy Metrics Activity Access S4	ettings	
	Get a complete visualization of your app in a team-based continuous deli	very environment with (6; Heroku Pipelines, Hide Create a Heroku Pipeline	
	Installed add-ons (20.00/month) Configure	Additions      Latest activity      AllActivity	
	ClearDB MySQL (2) Ignite cleardb-silhouetted-37505	Compared and the second	
	Dyno formation (\$0.00/ments) Configu	te Denos 🐵 🛞 🌚 elainetowijngyt00@gmail.com: Deployed 7aff793af Today at 12:06 AM-v25: <u>Compare dff</u>	
	This app is using free dynos	lelainelowjingyi00@gmail.com: Build succeeded	
	web heroku-php-apache2 public/	ON	
	Collaborator activity	elaineloonjingyl00@gmail.com: Sot CLEARDE_DATABASE_URL Config var Vesterday at 11:53 PM - v24	
	G <sup>2</sup> elainelowijngyi00@gmail.com	deploys elainetowijnegy@dogmal.com: Detach CLEARDB_COPPER (gmf.cleardb- ympronice330) Yesterday at 11.07 PM -v23	
		<ul> <li>elainelowijngvi00@gmail.com: @ref.cleardb-aerodynamic.84316 completed provisioning.setting CLEAR08_COPPER_UBL. Yesterday at 11:03 PM. v22</li> </ul>	
		elainelowjingyl00@gmail.com: Attach CLEARDB_COPPER (@ref-cleardb-	

Figure 6.12: Overview of Heroku app

⊗ Laravel × +		~ - a ×
← → C ☆ 🔒 fyp-wims.herokuapp.com/login		년 ☆ ¥ @ ¥ @ 券 팩 🛛 😫 :
🗙 How To Create a Si 📁 Free Icons   Font A 🧿 【汪神说】CSS学习笔 🛞 🍝 Downloads	- ISTQ8 🛞 IP Conversion   Sub 📭 IP Calculator / IP Su 🛅 (木筏/Raft)各种岛.	🗙 keyboard-shortcuts 🔞 The Python Mega C 🛓 [Raft capture] Tang »
	<b>.</b>	
	Sign in to your account	
	Username	
	Password	
	Remember me Forgot your password?	
	Sign in	

Figure 6.13: Deployed web-based inventory management system on Heroku

# 6.5 Conclusion

This chapter covers the implementation details for the intended system from the aspect of frontend and backend. For the frontend, all third-party libraries are covered whereas for the backend, the API list, JWT authentication and Pusher Channel are explained. Finally, after the development has been done, the implemented system is deployed to Heroku and the process of deployment is demonstrated in the last section.

#### **CHAPTER 7**

#### SYSTEM TESTING

# 7.1 Introduction

This chapter includes testing done after all development phases ended. The testing involved includes web service testing, real-time check-in/out module testing, user acceptance testing and usability testing. Besides, a traceability matrix is done between use cases, functional requirements, web service test cases and user acceptance test cases to validate that all use cases and functional requirements are tested during the system testing phase.

#### 7.2 Web Service Testing

Web service testing is software testing that is used to validate the functionality, reliability security and performance of application program interface (APIs). It is comparable to unit testing as it tests specific codes instead of user interface objects. In this project, web service testing is performed with the help of Postman to validate the correctness of each API in returning the correct results when specified inputs are provided.

Test Case ID	Test Case Title	Status
WST001	Test Case of User Login	PASS
WST002	Test Case of Logging Out	PASS
WST003	Test Case of Viewing Inventory List	PASS
WST004	Test Case of Viewing Inventory Details	PASS
WST005	Test Case of Adding New Inventory with Category Assigned	PASS
WST006	Test Case of Adding New Inventory with No Category Assigned	PASS
WST007	Test Case of Updating an Inventory with Different Category	PASS
WST008	Test Case of Updating an Inventory with Same or No Category	PASS

Table 7:1: Summary of web service testing's test cases with status

WST009	Test Case of Deleting an Inventory	PASS
WST010	Test Case of Viewing Warehouse List	PASS
WST011	Test Case of Viewing Warehouse Details	PASS
WST012	Test Case of Multi-Assigning Storage Bins of a	PASS
	Warehouse	
WST013	Test Case of Editing Inventory of a Warehouse Storage	PASS
	Bin	
WST014	Test Case of Adding New Warehouse	PASS
WST015	Test Case of Updating a Warehouse	PASS
WST016	Test Case of Deleting a Warehouse	PASS
WST017	Test Case of Viewing User List	PASS
WST018	Test Case of Viewing Only Active Staffs	PASS
WST019	Test Case of Viewing User Details	PASS
WST020	Test Case of Adding New User	PASS
WST021	Test Case of Updating a Staff	PASS
WST022	Test Case of Deleting a User	PASS
WST023	Test Case of Resetting Password	PASS
WST024	Test Case of Viewing Category List	PASS
WST025	Test Case of Adding New Category	PASS
WST026	Test Case of Updating a Category	PASS
WST027	Test Case of Deleting a Category	PASS
WST028	Test Case of Viewing Daily Report List	PASS
WST029	Test Case of Viewing Cycle Counting Approval Report	PASS
	List	
WST030	Test Case of Approving Cycle Counting Approval	PASS
	Report	
WST031	Test Case of Rejecting Cycle Counting Approval	PASS
	Report with Recounting	
WST032	Test Case of Rejecting Cycle Counting Approval	PASS
	Report without Recounting	
WST033	Test Case of Viewing Cycle Counting Summary Report	PASS
	List	
WST034	Test Case of Starting Cycle Counting	PASS

WST035	Test Case of Viewing Cycle Counting	PASS
WST036	Test Case of Reassigning Staff for Cycle Count	PASS
	Schedules	
WST037	Test Case of Viewing All Staffs	PASS
WST038	Test Case of Performing Cycle Counting	PASS
WST039	Test Case of Checking in Inventory Stock	PASS
WST040	Test Case of Checking out Inventory Stock	PASS

Table 7:2: Test case	of user login	
----------------------	---------------	--

Test Case ID	WST001		Status		PASS	
Test Case Title	Test Case of User Login					
Test Case Description	To verify the successful login status if the username and password provided by user exists in the					
	database					
Endpoints Involved	"http://127.0.0.1:8000/api/login?username={username}&password={password}"					
Test Steps	Test Data	Expected Result Ac		Actual 1	Actual Result	
1. The user enters the username	1. Username	JSON res	ponse with status	The use	er logged in successfully.	
and password to the endpoint	2. Password	code 200	which consists of the			
"http://127.0.0.1:8000/api/login?u			ken, token type, token			
sername={username}&password=		expiry tin	ne and the logged in			
{password}" to login into the		user obje	et			
system.						

Table 7:3: Test case of logging out

Test Case ID	WST002	Status	PASS	
Test Case Title	Test Case of Logging Out			
Test Case Description	To verify if user can be logged out from the system			
Endpoints Involved	"http://127.0.0.1:8000/api/logout"			
Test Steps	Test Data	Expected Result	Actual Result	
1. The user log out from the		JSON response with status	The user is logged out successfully.	
system using endpoint		code 200.		
"http://127.0.0.1:8000/api/logout"				

Table 7:4: Test case	of viewing	inventory list
	0	

Test Case ID	WST003	Status	PASS		
Test Case Title	Test Case of Viewing Inventory List				
Test Case Description	To verify if user can view the list of inventories				
Endpoints Involved	"http://127.0.0.1:8000/api/inventories"				
Test Steps	Test Data	Expected Result	Actual Result		
1. The user retrieves the list of		JSON response with an array	The list of inventories is retrieved		
inventories from endpoint		of inventories and status code	successfully.		
"http://127.0.0.1:8000/api/inventories"		200.			

# Table 7:5: Test case of viewing inventory details

Test Case ID	WST004	Status	PASS	
Test Case Title	Test Case of Viewing Inventory Details			
Test Case Description	To verify if user can view the inventory details			
Endpoints Involved	"http://127.0.0.1:8000/api/inventory/{id}",			
	"http://127.0.0.1:8000/	/api/getStocksByInventory/	/{id}"	
Test Steps	Test Data	Expected Result	Actual Result	
1. The user retrieves the inventory details from endpoint "http://127.0.0.1:8000/api/inventory/{id}","	1. Inventory ID	Inventory data object in JSON format with status code 200	The inventory data with all stocks related to it is retrieved successfully	
2. The user retrieves all stock records of the inventory from the endpoint "http://127.0.0.1:8000/api/getStocksByInventory/{id}"	1. Inventory ID	JSON object with array of stocks of given inventory and status code 200		

Test Case ID	WST005	Status	PASS		
Test Case Title	Test Case of Adding New Inventory with Category Assigned				
Test Case Description	To verify if user ca	To verify if user can add a new inventory with a category assigned to			
	it to the database				
Endpoints Involved	"http://127.0.0.1:8	8000/api/warehouses",			
	"http://127.0.0.1:8	8000/api/categories",			
	"http://127.0.0.1:8	8000/api/inventory?name	={name}&warehouse_id=		
	{warehouse_id}&	qty_on_hand={qty_on_h	nand}&cost_per_unit={co		
	st_per_unit}&cate	gory_id={category_id}&	<priority={priority}"< pre=""></priority={priority}"<>		
Test Steps	Test Data	Expected Result	Actual Result		
1. The user retrieves all warehouses from the database using		JSON object with	The new inventory is		
endpoint "http://127.0.0.1:8000/api/warehouses"		array of warehouses	created and added to the		
	and status code 200 database.				
2. The user retrieves all categories from the database using		JSON object with			
endpoint "http://127.0.0.1:8000/api/categories"	array of categories				
	and status code 200				
3. The user enters name, quantity on hand, cost per unit,	1. Name	JSON object with			
priority and selects the warehouse and category from the	2. Warehouse status code 200 and				
retrieved list to endpoint	3. Quantity on the storage bin object				
http://127.0.0.1:8000/api/inventory?name={name}&warehous					
e_id={warehouse_id}&qty_on_hand={qty_on_hand}&cost_p	4. Cost per Unit inventory is placed in				
er_unit={cost_per_unit}&category_id={category_id}&priorit	5. Category				
y={priority}	6. Priority				

Table 7:6: Test case of adding new inventory with category assigned

Test Case ID	WST006	Status	PASS		
Test Case Title	Test Case of Adding New Inventory with No Category Assigned				
Test Case Description	To verify if user car	n add a new inventory w	vith no category assigned		
	to it to the database				
Endpoints Involved	-	00/api/warehouses",			
	"http://127.0.0.1:80	1 0 /			
	-	1 V	={name}&warehouse_id		
			_hand}&cost_per_unit={c		
	ost_per_unit}&prior				
Test Steps	Test Data	Expected Result	Actual Result		
1. The user retrieves all warehouses from the database using		A JSON object with	The new inventory is		
endpoint "http://127.0.0.1:8000/api/warehouses",		array of warehouses	created and added to the		
		and status code 200	database.		
2. The user retrieves all categories from the database using	A JSON object with				
endpoint "http://127.0.0.1:8000/api/categories"		array of categories			
		and status code 200			
3. The user enters name, quantity on hand, cost per unit,	1. Name	A JSON object with			
priority and selects the warehouse from the retrieved list to	2. Warehouse	status code 201 and			
endpoint	3. Quantity on	the created inventory			
"http://127.0.0.1:8000/api/inventory?name={name}&warehou	Hand object				
se_id={warehouse_id}&qty_on_hand={qty_on_hand}&cost_p	-				
er_unit={cost_per_unit}&priority={priority}"	5. Priority				

Table 7:7: Test case of adding new inventory with no category assigned

Test Case ID	WST007	Status	PASS		
Test Case Title	Test Case of Upda	Test Case of Updating an Inventory with Different Category			
Test Case Description	To verify if user ca	in update an inventory v	with different category		
	compared to the or	iginal data			
Endpoints Involved			entory_id}?name={name}		
	&category_id={category_id}&cost_per_unit={cost_per_unit}&prio				
	rity={priority}"				
Test Steps	Test Data	Expected Result	Actual Result		
1. The user updates the name, category, cost per unit and	1. Inventory ID	A JSON object with	The inventory has been		
priority of a selected inventory using endpoint	2. Name storage bin data that updated with new				
"http://127.0.0.1:8000/api/inventory/{inventory_id}?name={na	a 3. Category is newly assigned to information entered by				
me}&category_id={category_id}&cost_per_unit={cost_per_u	4. Cost per Unit the inventory user.				
nit}&priority={priority}"	5. Priority				

Table 7:8: Test case of updating an inventory with different category

Test Case ID	WST008	Status	PASS		
Test Case Title	Test Case of Upd	Test Case of Updating an Inventory with Same or No Category			
Test Case Description	To verify if user	can update an inventory	with the no category or		
	same category as	the original data.			
Endpoints Involved	"http://127.0.0.1:8000/api/inventory/{inventory_id}?name={name}				
	&cost_per_unit={cost_per_unit}&priority={priority}"				
Test Steps	Test DataExpected ResultActual Result				
1. The user updates the name, cost per unit and priority of a	1. Inventory ID	A JSON response	The inventory has been		
selected inventory using endpoint	2. Name with status code 200 updated with		updated with new		
"http://127.0.0.1:8000/api/inventory/{inventory_id}?name={na	a 3. Cost per Unit and value 1 information entered				
me}&cost_per_unit={cost_per_unit}&priority={priority}"	4. Priority		user.		

Table 7:9: Test case of updating an inventory with same or no category

Table 7:10: Test case of deleting an inventory

Test Case ID	WST009	Status	PASS			
Test Case Title	Test Case of Deleting	Test Case of Deleting an Inventory				
Test Case Description	To verify if user can d	To verify if user can delete an inventory from the database				
Endpoints Involved	"http://127.0.0.1:8000/api/inventory/{id}"					
Test Steps	Test Data Expected Result Actual Result					
1. The user deletes an inventory using endpoint "http://127.0.0.1:8000/api/inventory/{id}".	1. Inventory ID					

Test Case ID	WST010	Status			PASS
Test Case Title	Test Case of Viewing	Test Case of Viewing Warehouse List			
Test Case Description	To verify if user can v	To verify if user can view the list of warehouses			
Endpoints Involved	"http://127.0.0.1:8000/api/warehouses"				
Test Steps	Test Data	Test Data Expected Result			al Result
1. The user retrieves the warehouses		A JSON	response with array	The li	ist of warehouses is retrieved
from endpoint	of warehouses and status code successfully			ssfully	
"http://127.0.0.1:8000/api/warehouses".		200			

Table 7:12: Test case of viewing warehouse details

Test Case ID	WST011	Status		PASS	
Test Case Title	Test Case of Viewing Warehouse Details				
Test Case Description	To verify if user can vi	ew the warehouse	details		
Endpoints Involved	"http://127.0.0.1:8000/api/warehouse/{id}"				
Test Steps	Test Data	Expected Result Actual Result			
1. The user retrieves the warehouse data	1. warehouse ID	A JSON object with status The warehouse data with			
from endpoint	code 200 and the warehouse storage bins and staffs is				
"http://127.0.0.1:8000/api/warehouse/{id}"	e				

Test Case ID	WST012	Status	PASS			
Test Case Title	Test Case of Multi-Ass	Test Case of Multi-Assigning Storage Bins of a Warehouse				
Test Case Description	To verify if user can as	sign a category to multiple bins.				
Endpoints Involved	"http://127.0.0.1:8000/a	api/categories", "http://127.0.0.1:	8000/api/assign-category-to-bin/{id}"			
Test Steps	Test Data	Expected Result Actual Result				
1. The user retrieves the list of		A JSON object with array of	The warehouse storage bins are			
categories from endpoint		categories and status code 200 updated and the category and				
"http://127.0.0.1:8000/api/categories"		storage bin data of those				
2. The user assigns multiple storage	1. Category ID	A JSON object with the	inventories that was assign to one			
bins with one category using	2. List of storage bin	updated warehouse object and	the selected bins were set back to			
endpoint	numbers to be	status code 200 null				
"http://127.0.0.1:8000/api/assign-	assigned					
category-to-bin/{id}"						

Table 7:13: Test case of multi-assigning storage bins of a warehouse

Test Case ID	WST013	Status	PASS	
Test Case Title	Test Case of Editing Inventory of a Warehouse Storage Bin			
Test Case Description	To verify if user can edit the inventory inside a warehouse's			
	storage bin			
Endpoints Involved		0/api/inventories-unassign	ned-category",	
		0/api/storage-bin-edit-		
	•	e_id}?bin_id={bin_id}&i	inventory_1d={1nvent	
Test Stans	ory_id}&category_id		Actual Result	
Test Steps	Test Data	Expected Result		
1. The user retrieves the list of inventories that are not assigned		A JSON object with	The warehouse	
with a category from endpoint		array of inventories	storage bin has	
"http://127.0.0.1:8000/api/inventories-unassigned-category"		that have no category	been updated with	
		and status code 200	the given	
2. The user edit the inventory in a selected storage bin of a	1. Warehouse ID	A JSON object with	inventory. The	
warehouse using endpoint "http://127.0.0.1:8000/api/storage-	2. Storage bin ID	the updated warehouse	selected inventory	
bin-edit-	3. Category ID	object and status code	is also updated with	
inventory/{warehouse_id}?bin_id={bin_id}&inventory_id={in	4. Inventory ID	200	the assigned bin's	
ventory_id}&category_id={category_id}"			category. If the	
			inventory of the	
			storage bin is	
			cleared, then the	
			category and	
			storage bin	
			information of the	
			original inventory	
			in the bin will also	
			be removed.	

Table 7:14: Test case of editing inventory of a warehouse storage bin

Test Case ID	WST014		Status	PASS			
Test Case Title	Test Case of Adding New Wa	Test Case of Adding New Warehouse					
Test Case Description	To verify if user can add a new	w warehou	ise				
Endpoints Involved	"http://127.0.0.1:8000/api/cate	egories", '	"http://127.0.0.1:8000/a	pi/warehouse"			
Test Steps	Test Data	Expected	d Result	Actual Result			
1. The user retrieves a list of		A JSON	object with array of	The new warehouse is			
categories from endpoint	categories and status code 200 successfully created into the						
"http://127.0.0.1:8000/api/categories"		database.					
2. The user sends information which	1. New warehouse name	A JSON	object with the new				
includes name, location and storage	2. New warehouse location	warehou	se object and status				
bins for the new warehouse to	3. Storage bins code 201						
endpoint							
"http://127.0.0.1:8000/api/warehouse"							

Table 7:16: Test case of updating a warehouse

Test Case ID	WST015		Status	PASS	
Test Case Title	Test Case of Updating a Warehouse				
Test Case Description	To verify if user can upo	date an	existing warehou	ise	
Endpoints Involved	"http://127.0.0.1:8000/a	pi/ware	house/{id}?locat	tion={location}&m	
	anager_id={manager_id}"				
Test Steps	Test Data	Expec	ted Result	Actual Result	
1. The user updates location and warehouse manager of a	1. Warehouse ID	A JSC	N object with	The warehouse's	
warehouse using endpoint	2. New warehouse	status	code 200 and	location is	
"http://127.0.0.1:8000/api/warehouse/{id}?location={location}	location	value	1	updated. Manager	
&manager_id={manager_id}"	3. New manager ID			of the warehouse	
				will only be	
				updated if new	
				manager is given.	

Test Case ID	WST016	S	Status	PASS
Test Case Title	Test Case of Deleting a Ware	ehouse		
Test Case Description	To verify if user can delete an	n existing w	varehouse	
Endpoints Involved	"http://127.0.0.1:8000/api/wa	arehouse/{ic	d}"	
Test Steps	Test Data	Expected	Result	Actual Result
1. The user deletes a selected warehouse	1. Warehouse ID		object with status	The selected warehouse is
using endpoint		code 200 a	and value 1	deleted from the database.
"http://127.0.0.1:8000/api/warehouse/{id}"				All staffs, inventories, cycle
				countings and stocks
				related to this warehouse
				are deleted.

Table 7:18: Test case of viewing user list

Test Case ID	WST017	Status	PASS		
Test Case Title	Test Case of Viewing User List				
Test Case Description	To verify if user can view the list of users.				
Endpoints Involved	"http://127.0.0.1:8000/api/users"				
Test Steps	Test Data	Expected Result	Actual Result		
1. The user retrieves the list of		A JSON object with an array of	The list of users that includes		
users from endpoint		both active and inactive users is			
"http://127.0.0.1:8000/api/users"			retrieved		

Table 7:19: Test case	of	viewing	only	active staffs
-----------------------	----	---------	------	---------------

Test Case ID	WST018		Status	PASS		
Test Case Title	Test Case of Viewing Only Ac	Test Case of Viewing Only Active Staffs				
Test Case Description	To verify if user can view only active staffs. Inactive staffs, managers and admin will not be					
	retrieved.					
Endpoints Involved	"http://127.0.0.1:8000/api/active-staffs"					
Test Steps	Test Data	Expected	Result	Actual Res	sult	
1. The user retrieves the list of		A JSON o	bject with an array of	The list of	active staffs is	
active staffs from endpoint	active staffs and status code 200 retrieved succes		uccessfully			
"http://127.0.0.1:8000/api/active-						
staffs"						

## Table 7:20: Test case of viewing user details

Test Case ID	WST019	Status	PASS			
Test Case Title	Test Case of Viewing User De	Test Case of Viewing User Details				
Test Case Description	To verify if user can view the selected user's details.					
Endpoints Involved	"http://127.0.0.1:8000/api/user/{id}"					
Test Steps	Test Data	Expected Result	Actual Result			
1. The user retrieves the user details	1. User ID	A JSON object with the	The selected user object is			
f a selected user using endpoint		retrieved user object and status	successfully retrieved.			
"http://127.0.0.1:8000/api/user/{id}"		code 200				

Test Case ID	WST020	Status	PASS		
Test Case Title	Test Case of Adding New User				
Test Case Description	To verify if user can add a new us	er to a warehouse			
Endpoints Involved	"http://127.0.0.1:8000/api/wareho	uses",			
	"http://127.0.0.1:8000/api/user?na	me={name}&email={email}&contac	t_no={contact_no}⁣_no={ic_no}		
	&role={role}&warehouse_id={warehouse_id=	arehouse_id}&employed_in={employ	ved_in}&address={address}		
	&username={username}"				
Test Steps	Test Data	Expected Result	Actual Result		
1. The user retrieves		A JSON object with an array of	A temporary password is created		
all warehouses from		warehouses and status code 200	using the new user's contact		
the endpoint			number and the new user is		
"http://127.0.0.1:8000			successfully created into the		
/api/warehouses"			database.		
2. The user sends all	1. User's name	A JSON object with the newly			
information needed	2. User's email	created user object and status code			
for a new warehouse	3. User's Contact number	201			
to the endpoint	4. User's IC number				
"http://127.0.0.1:8000	5. User's role				
/api/user?name={nam	6. User's warehouse ID				
e}&email={email}&c	7. User's employed date				
ontact_no={contact_n	8. User's address				
o}⁣_no={ic_no}&r	9. User's username				
ole={role}&warehous					
e_id={warehouse_id}					
&employed_in={empl					
oyed_in}&address={a					
ddress}					
&username={usernam					
e}"					

Table 7:21: Test case of adding new user

Test Case ID	WST021	Status	PASS		
Test Case Title	Test Case of Updating a Staff				
Test Case Description	To verify if user can update the er	nail, contact number, role, warehouse	and address of an existing user		
Endpoints Involved	"http://127.0.0.1:8000/api/wareho	uses",			
	"http://127.0.0.1:8000/api/user/{id	d}?email={email}			
	&contact_no={contact_no}&role	={role}&warehouse_id={warehouse_i	d}&address={address}"		
Test Steps	Test Data	Expected Result	Actual Result		
1. The user retrieves		A JSON object with an array of	The user is updated with the given		
all warehouses from		warehouses and status code 200	information.		
the endpoint					
"http://127.0.0.1:8000					
/api/warehouses"					
2. The user sends all	1. User ID	A JSON object with status code 200			
information needed	2. User's email	and value 1			
for a new warehouse	3. User's Contact number				
to the endpoint	4. User's role				
"http://127.0.0.1:8000	5. User's warehouse ID				
/api/user/{id}?email=	6. User's address				
{email}					
&contact_no={contact					
_no}&role={role}&w					
arehouse_id={wareho					
use_id}&address={ad					
dress}"					

Table 7:23: Test case of deleting a user

Test Case ID	WST022	Status	PASS			
Test Case Title	Test Case of Deleting a User					
Test Case Description	To verify if user can delete an existing user					
Endpoints Involved	"http://127.0.0.1:8000/api/user/{id	d}"				
Test Steps	Test Data	Expected Result	Actual Result			
1. User deletes a user	1. User ID	An empty JSON object with status	The selected user is set as inactive			
using endpoint		code 200	user			
"http://127.0.0.1:8000						
/api/user/{id}"						

# Table 7:24: Test case of resetting password

Test Case ID	WST023	VST023 Status		PA	ASS		
Test Case Title	Test Case of Resetting Password						
Test Case Description	To verify if user can reset the pass	sword on firs	t time login				
Endpoints Involved	"http://127.0.0.1:8000/api/reset-						
	password?new_password={new_j	password}&c	confirm_password={confir	m_passw	vord}"		
Test Steps	Test Data	Expected R	esult	Actual I	Result		
1. User sends the new	1. New password	A JSON ob	ject with the updated	The log	ged in user's password is		
password to the	2. Confirm password	user object	and status code 200	updated			
endpoint							
"http://127.0.0.1:8000							
/api/reset-							
password?new_passw							
ord={new_password}							
&confirm_password=							
{confirm_password}"							

# Table 7:25: Test case of viewing category list

Test Case ID	WST024		Status		PASS		
Test Case Title	Test Case of Viewing Category L	Test Case of Viewing Category List					
Test Case Description	To verify if user can view the list	To verify if user can view the list of categories					
<b>Endpoints Involved</b>	"http://127.0.0.1:8000/api/categories"						
Test Steps	Test Data	Expected Result			Actual Result		
1. User retrieves the category list from the endpoint "http://127.0.0.1:8000 /api/categories"		•	categories in the format status code 200		list of categories is retrieved essfully		

## Table 7:26: Test case of adding new category

Test Case ID	WST025 Status		Status		PASS		
Test Case Title	Test Case of Adding New Catego	Test Case of Adding New Category					
Test Case Description	To verify if user can add a new ca	tegory for a	ll warehouses				
<b>Endpoints Involved</b>	"http://127.0.0.1:8000/api/category?name={name}"						
Test Steps	Test Data	Expected Result		Actual Result			
1. User sends the new	1. New category's name	The newly created category object			new category is created into		
category name to the			at of JSON and status	the c	latabase successfully		
endpoint		code 201					
"http://127.0.0.1:8000							
/api/category?name={							
name}"							

Test Case ID	WST026		Status		PASS		
Test Case Title	Test Case of Updating a Category	Test Case of Updating a Category					
Test Case Description	To verify if user can update an ex	isting category	's name				
<b>Endpoints Involved</b>	"http://127.0.0.1:8000/api/categor	"http://127.0.0.1:8000/api/category/{category_id}?name={name}"					
Test Steps	Test Data	Expected Result		Actual Result			
1. User sends the new	1. Desired category's ID	A JSON obje	ect status code 200 and	The	selected category's name is		
category name to the	2. New category name	value 1		upda	ited.		
endpoint							
"http://127.0.0.1:8000							
/api/category/{categor							
y_id}?name={name}"							

Table 7:27: Test case of updating a category

Table 7:28: Test case of deleting a category

Test Case ID	WST027	Status		PASS			
Test Case Title	Test Case of Deleting a Category	fest Case of Deleting a Category					
Test Case Description	To verify if user can delete an exi	To verify if user can delete an existing category's name					
Endpoints Involved	"http://127.0.0.1:8000/api/category/{category_id}"						
Test Steps	Test Data	Expected Result	Act	Actual Result			
1. User deletes the	1. Desired category's ID	A JSON object status code 200 and	The	selected category's is			
selected category		value 1	suce	cessfully deleted from the			
using the endpoint			data	base			
"http://127.0.0.1:8000							
/api/category/{categor							
y_id}"							

Test Case ID	WST028	Status	PASS				
Test Case Title	Test Case of Viewing Daily Repo	Test Case of Viewing Daily Report List					
Test Case Description	To verify if user can view the list	To verify if user can view the list of daily reports.					
<b>Endpoints Involved</b>	"http://127.0.0.1:8000/api/stocks"						
Test Steps	Test Data	Expected Result	Actual Result				
1. User retrieved the		An array of inventory stock records	The list of daily reports is retrieved				
daily reports from the		in JSON format and status code 200	) from the database successfully.				
endpoint							
"http://127.0.0.1:8000							
/api/stocks"							

Table 7:30: Test case	of viewing cycle	counting approv	al report list
1 abic 7.30. 10st case	of viewing cycle	counting approv	ai report fist

Test Case ID	WST029		Status		PASS		
Test Case Title	Test Case of Viewing Cycle Court	nting Approv	val Report List				
Test Case Description	To verify if user can view the list	of cycle cou	inting approval reports				
Endpoints Involved	"http://127.0.0.1:8000/api/cycle-counts"						
Test Steps	Test Data	Expected Result			Actual Result		
1. User retrieves the		An array of cycle counting records			list of cycle counting approval		
cycle count approval		in JSON fo	ormat and status code 200	repo	orts is retrieved from the		
reports from the				data	base successfully.		
endpoint							
"http://127.0.0.1:8000							
/api/cycle-counts"							

Test Case ID	WST030	Status	PASS					
Test Case Title	Test Case of Approving Cycle Co	unting Approval Report						
Test Case Description	To verify if user can approve one	of the cycle-counting approval report	from the list					
Endpoints Involved	"http://127.0.0.1:8000/api/approve	e-cycle-						
	count?cycle_counting_id={cycle_	_counting_id}&ira={ira}&inventory_	id={inventory_id}&variance={varian					
	ce}"							
Test Steps	Test Data	Expected Result	Actual Result					
1. User approves one	1. Cycle counting ID of the	A JSON object with status code 200	The selected cycle counting					
of the cycle-counting	cycle counting approval report	and value 1	approval report is approved with					
approval reports from	2. IRA calculated from variance the calculated IRA updated to the							
the list by sending the	3. Inventory ID of the cycle		database. The status of the cycle					
cycle counting id,	counting approval report		counting changed from					
IRA, inventory ID and	4. Variance between the		"PENDING" to "COMPLETED".					
variance to the	recorded count and the actual		The recorded count of the involved					
endpoint	count		inventory is modified with the					
"http://127.0.0.1:8000			variance.					
/api/approve-cycle-								
count?cycle_counting								
_id={cycle_counting_								
id}&ira={ira}&invent								
ory_id={inventory_id								
}&variance={variance								
Y"								

Table 7:31: Test case of approving cycle counting approval report

Test Case ID	WST031	Status	PASS				
Test Case Title	Test Case of Rejecting Cycle Cou	nting Approval Report with Recountin	g				
Test Case Description	To verify if user can reject one of	the cycle-counting approval report from	n the list. Schedule should be				
	created for recounting.						
Endpoints Involved	"http://127.0.0.1:8000/api/reject-c	cycle-count?cycle_counting_id={cycle_	_counting_id}&recount={recount}				
	&schedule_date={schedule_date}	"					
Test Steps	Test Data	Expected Result	Actual Result				
1. User rejects one of	1. Cycle counting ID of the	A JSON object with the newly	The selected cycle counting				
the cycle-counting	cycle counting approval report	created schedule for recounting and	approval report is rejected and a				
approval reports from	2. Recount (true or false)	status code 201. new schedule for recounting the					
the list by sending the	3. Schedule date for recounting		sku is generated. The status of the				
cycle counting id,	the cycle counting		cycle counting changed from				
recount and schedule			"PENDING" to "REJECTED".				
date to the endpoint							
"http://127.0.0.1:8000							
/api/reject-cycle-							
count?cycle_counting							
_id={cycle_counting_							
id}&recount={recount							
}							
&schedule_date={sch							
edule_date}"							

Table 7:32: Test case of rejecting cycle counting approval report with recounting

Test Case ID	WST032	Status	PASS
Test Case Title	Test Case of Rejecting Cycle Cou	nting Approval Report without Recour	nting
Test Case Description	To verify if user can reject one of	the cycle-counting approval report from	m the list with no recounting.
Endpoints Involved	"http://127.0.0.1:8000/api/reject-c	eycle-	
	count?cycle_counting_id={cycle_	<pre>_counting_id}&amp;recount={recount}&amp;scl</pre>	hedule_date= {schedule_date}"
Test Steps	Test Data	Expected Result	Actual Result
1. User rejects one of	1. Cycle counting ID of the	A JSON object with status code 200	The selected cycle counting
the cycle-counting	cycle counting approval report	and value 1.	approval report is rejected. The
approval reports from	2. Recount (true or false)		status of the cycle counting
the list by sending the	3. Schedule date for recounting		changed from "PENDING" to
cycle counting id,	the cycle counting		"REJECTED".
recount and schedule			
date to the endpoint			
"http://127.0.0.1:8000			
/api/reject-cycle-			
count?cycle_counting			
_id={cycle_counting_			
id}&recount={recount			
}&schedule_date=			
{schedule_date}"			

Table 7:33: Test case of rejecting cycle counting approval report without recounting

Test Case ID	WST033 Status			PASS		
Test Case Title	Test Case of Viewing Cycle Court	nting Summa	ry Report List			
Test Case Description	To verify if user can view the list	of cycle cour	nting summary reports			
<b>Endpoints Involved</b>	"http://127.0.0.1:8000/api/cycle-counts"					
Test Steps	Test Data	Expected Result			Actual Result	
1. User retrieves the		An array of cycle counting records			list of cycle counting summary	
cycle count summary		in JSON fo	rmat and status code 200	repo	rts is retrieved from the	
reports from the				data	base successfully.	
endpoint						
"http://127.0.0.1:8000						
/api/cycle-counts"						

Table 7:34: Test case of viewing cycle counting summary report list

Test Case ID	WST034	Status	S	PASS
Test Case Title	Test Case of Starting Cycle Counting			
Test Case Description	To verify if user can start or restart the cycle counting by entering settings			
Endpoints Involved	"http://127.0.0.1:8000/api/active-staffs",			
	"http://127.0.0.1:8000/api/invento			
				vorking_day_start={working_day_st
				e_count_class}&warehouse_id={wa
	rehouse_id}&staff_ids={staff_ids	s}&inventory_ids=	{inventory_ids}&sta	art_end_date={start_end_date}",
	"http://127.0.0.1:8000/api/sku",			
	"http://127.0.0.1:8000/api/schedu			
Test Steps	Test Data	Expected Result		Actual Result
1. User retrieves the			ith an array of user	The list of active staffs of user's
active staffs from the		objects that are ad	ctive and status	warehouse is retrieved
endpoint		code 200		successfully.
"http://127.0.0.1:8000				
/api/active-staffs"				
2. User retrieves all		A JSON object w		The list of inventories of user's
inventories of his		inventory objects		warehouse is retrieved
warehouse from the		warehouse and sta	atus code 200	successfully.
endpoint				
"http://127.0.0.1:8000 /api/inventories"				
3. User sends the	1. Warehouse's ID	A warehouse obje	at with the	User's warehouse's cycle counting
workday start,	2. Starting of working day		inting settings and	settings is updated with the
workday end, cycle	3. Ending of working day	status code 200	and settings and	information given by user.
count classes,	4. Cycle count classes	status code 200		information given by user.
warehouse ID, staff	information which includes			
ids, inventory ids, and	class name (A, B or C), daily			
start end date to the	count for each class, frequency			
endpoint	to count all skus for each class,			
"http://127.0.0.1:8000	number of skus per class, type			

Table 7:35: Test case of starting cycle counting

/api/storeCycleCounti	of counting frequency (daily,		
ngSettings/{warehous	weekly, monthly or yearly), and		
eId}?working_day_sta	counting frequency per class		
rt={working_day_star	(eg. $3 = \text{every } 3$		
t}&working_day_end	days/weeks/months/year)		
={working_day_end}	5. warehouse ID to save the		
&cycle_count_class={	settings to		
cycle_count_class}&	6. staff ids to be assigned as		
warehouse_id={wareh	cycle counting personnels		
ouse_id}&staff_ids={	7. inventory ids to be assigned		
staff_ids}&inventory_	as sku for cycle counting		
ids={inventory_ids}&	8. start and end date of the cycle		
start_end_date={start_	counting period		
end date}",to save the			
settings.			
4. User sends the	1. Array of SKUs which	A JSON object with status code 200	The SKUs are inserted into the
array of SKUs to the	includes the inventory id and	and value 1	database successfully. All previous
endpoint	their corresponding cycle count		cycle counting's SKUs for user's
"http://127.0.0.1:8000	class (A, B, or C)		warehouse are removed from the
/api/sku"			database.
5. User sends the	1. Array of schedules for cycle	A JSON object with status code 200	The schedules are inserted into the
array of cycle	counting that includes the	and value 1	database successfully. All previous
counting schedules to	assigned staff ID, inventory ID		cycle counting's schedules for
the endpoint	and schedule date.		user's warehouse are removed
"http://127.0.0.1:8000			from the database.
/api/schedule"			

Test Case ID	WST035	Status	PASS	
Test Case Title	Test Case of Viewing Cycle Counting			
Test Case Description	To verify if user can view current cycle counting of his or her warehouse. The user should be able to view the list of cycle counting schedules and the current cycle counting settings with the list of staffs involved and SKUs.			
Endpoints Involved	"http://127.0.0.1:8000/api/war "http://127.0.0.1:8000/api/skus "http://127.0.0.1:8000/api/sche	s",		
Test Steps	Test Data	Expected Result	Actual Result	
1. The user retrieves his warehouse's details from endpoint "http://127.0.0.1:8000 /api/warehouse/{ware houseId}",	1. User's warehouse ID	A JSON object with the retrieved warehouse object and status code 200	The warehouse details which include the cycle counting settings are retrieved from the database.	
2. The user retrieves the SKUs for the cycle counting of his warehouse from endpoint "http://127.0.0.1:8000 /api/skus"		A JSON object with an array of retrieved SKUs and status code 200	All SKUs included in user warehouse's cycle counting are retrieved from the database.	
3. The user retrieves all cycle countig schedules of his warehouse from endpoint "http://127.0.0.1:8000 /api/schedules"		A JSON object with an array of retrieved cycle counting schedules and status code 200	All cycle counting schedules with status "OPEN" are retrieved successfully.	

Table 7:36: Test case of viewing cycle counting

Test Case ID	WST036	Status	PASS		
Test Case Title	Test Case of Reassigning Staff for	Test Case of Reassigning Staff for Cycle Count Schedules			
Test Case Description	To verify if user can reassign all c	cycle counting schedules of a selected s	taff to a new staff.		
Endpoints Involved	"http://127.0.0.1:8000/api/active-s	staffs",			
	"http://127.0.0.1:8000/api/reassig	nStaff"			
Test Steps	Test Data	Expected Result	Actual Result		
1. The user retrieves		A JSON object with an array of	The list of active staffs of user's		
all active staffs in his		retrieved active staffs and status	warehouse is retrieved		
warehouse from	code 200. successfully.				
endpoint					
"http://127.0.0.1:8000					
/api/active-staffs"					
2. The user sends the	1. Array of schedule ids to be	A JSON object with status code 200	All schedules of the original		
schedules and new	assigned to new staff.	and value 1	assigned staff have been		
staff to assigned to the	2. New staff ID		successfully assigned to the new		
endpoint	staff.				
"http://127.0.0.1:8000					
/api/reassignStaff"					

Table 7:37: Test case of reassigning staff for cycle count schedules

Test Case ID	WST037		Status		PASS
Test Case Title	Test Case of Viewing All Staffs	Test Case of Viewing All Staffs			
Test Case Description	To verify if user can view all staff	fs in a wareh	ouse.		
Endpoints Involved	"http://127.0.0.1:8000/api/wareho	"http://127.0.0.1:8000/api/warehouse/{warehouseId}"			
Test Steps	Test Data	Expected Result Actual Res		ual Result	
1. The user retrieves	1. Warehouse ID	A JSON of	oject with the retrieved	The	warehouse details that
the warehouse details		warehouse	object and status code	inch	udes all staffs related to the
from the endpoint		200 warehouse are retrieved		ehouse are retrieved	
"http://127.0.0.1:8000		successfully.		essfully.	
/api/warehouse/{ware					
houseId}"					

Test Case ID	WST038	Status	PASS	
Test Case Title	Test Case of Performing Cycle Counting			
Test Case Description	To verify if user can perform cycl	e counting on one of the upcoming cyc	cle-counting schedules	
Endpoints Involved	"http://127.0.0.1:8000/api/schedu	les",		
	"http://127.0.0.1:8000/api/cycle-			
		&actual_count={actual_count}&recond		
		y={inv_rec_accuracy}&status={status		
Test Steps	Test Data	Expected Result	Actual Result	
1. The user retrieves		A JSON object with an array of	All cycle counting schedules	
all upcoming cycle		retrieved cycle counting schedules	assigned to the user are retrieved.	
counting schedules		and status code 200		
from endpoint				
"http://127.0.0.1:8000				
/api/schedules"				
2. The user sends the	1. Cycle Count Schedule ID	The newly created cycle counting	A cycle counting data is created	
cycle count information to the	2. Actual count	object and status 201.	with the IRA calculated into the database. The counted schedule's	
	<ol> <li>Recorded count</li> <li>Variance between actual and</li> </ol>			
endpoint "http://127.0.0.1:8000	recorded count		status changed from "OPEN" to "CLOSED"	
/api/cycle-	5. Inventory record accuracy		CLOSED	
count?schedule_id={s	calculated			
chedule_id}&actual_c	6. Status			
ount={actual_count}	o. Status			
&recorded_count={re				
corded_count}&varia				
nce={variance}&inv_				
rec_accuracy={inv_re				
c_accuracy}&status={				
status}"				

Test Case ID	WST039	Status	PASS	
Test Case Title	Test Case of Checking in Inventory Stock			
Test Case Description	To verify if user can check in the	inventory stock		
Endpoints Involved	"http://127.0.0.1:8000/api/invento	pries",		
	"http://127.0.0.1:8000/api/stock?i	nventory_id={inventory_id}&mode={	mode}	
	&quantity={quantity}&remarks=	{remarks}"		
Test Steps	Test Data	Expected Result	Actual Result	
1. The user retrieves		A JSON object with an array of		
all inventories of his		retrieved inventories and status code		
warehouse from		200		
endpoint				
"http://127.0.0.1:8000				
/api/inventories"				
2. The user sends the	1. Inventory ID	The newly created stock object and	The inventory stock is created into	
inventory id, check	2. Check In/Out Mode – "check-	status 201.	the database. The inventory	
in/out mode, quantity	in"		quantity is updated with the check	
and remarks to the	3. Quantity		in quantity	
endpoint	4. Remarks			
"http://127.0.0.1:8000				
/api/stock?inventory_i				
d={inventory_id}&m				
ode={mode}				
&quantity={quantity}				
&remarks={remarks}				
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,				

Table 7:40: Test case of checking in inventory stock

Test Case ID	WST040	Status	PASS
Test Case Title	Test Case of Checking out Invento	ory Stock	
Test Case Description	To verify if user can check out the	e inventory stock	
Endpoints Involved	"http://127.0.0.1:8000/api/invento	ries",	
	"http://127.0.0.1:8000/api/stock?in	nventory_id={inventory_id}&mode={	mode}
	&quantity={quantity}&remarks=	{remarks}"	
Test Steps	Test Data	Expected Result	Actual Result
1. The user retrieves		A JSON object with an array of	
all inventories of his		retrieved inventories and status code	
warehouse from		200	
endpoint			
"http://127.0.0.1:8000			
/api/inventories"			
2. The user sends the	1. Inventory ID	The newly created stock object and	The inventory stock is created into
inventory id, check	2. Check In/Out Mode –	status 201.	the database. The inventory
in/out mode, quantity	"checkout"		quantity is updated with the
and remarks to the	3. Quantity		checkout quantity
endpoint	4. Remarks		
"http://127.0.0.1:8000			
/api/stock?inventory_i			
d={inventory_id}&m			
ode={mode}			
&quantity={quantity}			
&remarks={remarks}			

Table 7:41: Test case of checking out inventory stock

This testing is done manually by utilizing the Console time() function from JavaScript to keep track of the time differences between creating new stock and receiving the API message broadcasted by Pusher. To ensure accuracy, 100 stocks will be created manually and the time differences are recorded in Table 7.42 and converted into graphical representation for easier analysis as shown in Figure 7.1.

Iteration No.	Time Difference (Delay Time) (ms)
1	839.364013671875
2	804.43798828125
3	1300.865966796875
4	1040.69482421875
5	826.22607421875
6	813.696044921875
7	833.98486328125
8	806.329833984375
9	829.08984375
10	835.852783203125
11	814.08203125
12	811.736083984375
13	956.326904296875
14	1076.707763671875
15	803.6591796875
16	819.56201171875
17	803.190185546875
18	797.010986328125
19	1417.396728515625
20	976.5830078125
21	1073.643798828125
22	825.501953125
23	828.118896484375
24	801.23486328125

Table 7:42: Testing iterations of delay time

25	891.909912109375
26	873.7431640625
27	824.130859375
28	810.4287109375
29	847.43310546875
30	856.81005859375
31	880.105224609375
32	1290.8330078125
33	852.7109375
34	834.806884765625
35	907.81591796875
36	1421.77783203125
37	872.830810546875
38	849.484130859375
39	842.723876953125
40	1359.113037109375
41	860.568115234375
42	885.2978515625
43	1392.31103515625
44	834.68505859375
45	1260.85595703125
46	826.39697265625
47	1379.18798828125
48	1038.342041015625
49	1028.7900390625
50	1225.70068359375
51	804.845947265625
52	1143.73095703125
53	1284.787109375
54	819.579833984375
55	878.468017578125
56	854.807861328125

57	808.14794921875	
58	818.85498046875	
59	983.770263671875	
60	1056.886962890625	
61	1383.447998046875	
62	1034.9091796875	
63	808.204833984375	
64	859.890869140625	
65	1287.89501953125	
66	873.10986328125	
67	805.27587890625	
68	1076.5810546875	
69	913.40478515625	
70	817.7109375	
71	907.080810546875	
72	829.441650390625	
73	1143.14111328125	
74	846.197265625	
75	802.955078125	
76	1255.902099609375	
77	826.59912109375	
78	823.554931640625	
79	913.60986328125	
80	831.35498046875	
81	842.8876953125	
82	833.81494140625	
83	805.175048828125	
84	810.923095703125	
85	816.511962890625	
86	810.6708984375	
87	1487.77197265625	
88	897.81396484375	

89	1099.4169921875
90	1356.26318359375
91	804.204833984375
92	816.962890625
93	1052.086181640625
94	1066.708251953125
95	795.73193359375
96	822.212158203125
97	1201.85791015625
98	812.48193359375
99	1317.908935546875
100	1062.9638671875

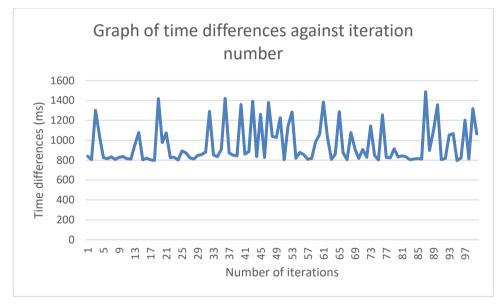


Figure 7.1: Graph of time differences against iteration number

From the graph in Figure 7.1, the maximum time difference is 1487.771973ms whereas the minimum time difference is 795.7319336ms. The average time difference calculated is 956.9260571ms, which is approximately equals to 1 second.

### 7.4 User Acceptance Testing and Usability Testing

User acceptance testing is carried out to determine whether the warehouse inventory management system can be accepted or not, whereas the usability testing is done to identify any usability issues that can cause confusion or difficulties to the user while completing their task. There will be five targeted users in this section. Two of them are warehouse workers who are familiar with the business requirements in the warehouse (expert users) and another three of them are non-warehouse workers (novice users) to make sure that the system developed can also be adopted by users that are not familiar with the warehouse operations, which can prove the usability of the system. This testing is performed after web service testing is completed and all issues have been fixed. Both user acceptance testing and usability testing will be combined using the same set of test cases and will cover all important modules in this system, which adhere to the three project objectives mentioned above. All tests were done anonymously to protect the testers' privacy.

The testing will be done online, with a test form given to the user which includes all modules needed to be tested with detailed steps to perform the actions. can access the deployed application with the URL https://fyp-Thev wims.herokuapp.com/ and the needed credentials were given to them for testing purposes. The developer kept the communication during the online session to a minimum level to give testers time to figure out their way to complete all of the test cases and observed the results obtained by the user for each test case. The developer was required to fill in the status and comments after the tester performed each test case. Status "PASS" indicates that the user has successfully performed the action with the observed result similar to the expected result whereas "FAIL" will be given if the user got a result other than the expected result or failed to complete the action due to any reason such as error occurred, confusions caused by the user interface, etc. The developer will be always on standby while the testers are performing the user acceptance testing to answer any question or provide a temporary solution to solve errors faced to ensure that the testing is done smoothly and all test cases are covered successfully.

After user acceptance testing has been done, the usability form will be given to the user to evaluate the usability of the system and a SUS score will be calculated afterwards for analysis. The results of the user acceptance testing are attached in Appendix C. There is only one failed test case and several comments given by the testers that require attention as shown in Table 7.43.

Test Case ID	Status	Error occurred/ Comments	Action Taken
TC013	Fail	Edited the phone number, the new	After investigation and testing, it was found out that the tester
		phone number cannot be used as the	edited the phone number of the created user. However, the default
		password, have to use the phone	password still remained as the previous phone number. Therefore,
		number that is created upon adding the	the action taken was to add an if-condition when updating a user
		new user	to make sure that the default password is updated accordingly as
			well.
TC001	Pass	'Opps' error message	After testing, it was found out that the error notification only show
			'Opps' as the title without any proper message which makes tester
			to be confused. This is caused by the login module not
			implementing the handleApiError global methods written. Hence,
			the action taken to fix this was to add condition for unauthorized
			error in the global method and implement it in the module.
TC003	Pass	Cancel button and close button not	After testing, the cancel button of the edit category modal does not
		working	close the modal. Hence, the action taken is to add the closing
			modal action to the cancel button.
TC020	Pass	No notification but got update	After testing, the approve cycle count approval action did not
			produce any success message which caused confusion to the tester.
			Hence, action to add a success notification is taken.

 Table 7:43: Summary of User Acceptance Testing Result

TC021	Pass	No notification but got update	After testing, the reject cycle count approval action did not
			produce any success message which caused confusion to the tester.
			Hence, action to add a success notification is taken.

## 7.4.2 Usability Testing Result

The System Usability Scale (SUS) was invented by John Brooke in 1986 as a 'quick and dirty' way to measure a system's usability. It considers and measures the combination of a system's efficiency, intuitiveness, ease of use and user satisfaction (Chinn, 2022). One notable study of finding usability problems in a system is conducted by Neilson and Launder (1993). With the assumption of the finding of usability problems are independent of whether they have been found before and independent of each other, Neilson and Launder found out that the detection of usability problems (as a function of the number of users tested or heuristic evaluator used) can be modelled as a *Poisson* process. Particularly, the number of usability problems found, denoted as  $F(i, N, \lambda)$ , in a usability test with *i* users can be described as:

$$F(i, N, \lambda) = N(1 - (1 - \lambda)^{i}), \qquad (7.1)$$

where *N* is the total number of usability problems in the system design and  $\lambda$  is the probability of discovering the average number of usability problems while testing with a single user. Substitute the typical value for  $\lambda = 0.31$  (Neilson and Launder, 1993), it is easy to verify that  $F(i = 5, N, \lambda = 0.31) \approx 0.85N$ , which means that more than 80% of the total number of usability problems can be discovered by only involving five users for testing. This is due to the fact that the more users we include in the test, the less information we can get (a usability issue), since we started to notice the same things over and over again.

A Google Form on the user satisfaction survey was given to the tester after they have performed the user acceptance testing as shown in Appendix D. The results were summarized in Table 7.44 with the SUS score calculated. The average SUS score is 85.5, which is considered excellent according to the scoring shown in Figure 7.2.

Tester	Sco	Score for each item							Total		
	1	2	3	4	5	6	7	8	9	10	-
Tester #1	4	2	4	4	5	1	4	2	5	2	77.5
Tester #2	4	4	4	1	3	1	4	2	5	4	70
Tester #3	5	2	5	3	5	1	5	1	4	3	85
Tester #4	5	1	5	1	4	1	5	1	5	2	95
Tester #5	5	1	5	1	5	1	5	1	5	1	100
Average SUS Score						85.5					

Table 7:44: SUS score collected

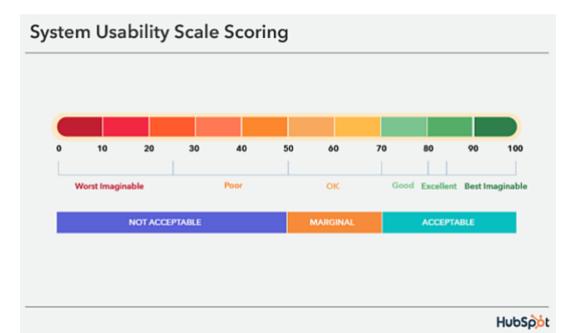


Figure 7.2: System Usability Scale Scoring (Chinn, 2022)

Aside from the 10 questions asked to calculate the SUS score, there are also four open-ended questions were given to the testers to collect their opinions on the system. The responses were tabulated in Table 7.45. It could be summarized that the user likes the simple user interfaces of the system and what they like least about the system is that the table does not update after every action and requires them to refresh the page to show updates. The second part that they do not like about the system is that the data loading is very slow. Hence, these two parts that the testers mentioned will be taken into consideration for recommendations on future works.

Question	Responses
What do you like best	simple and clean interface
about the system	Simple but useful design
	easy to use, simple UI
What do you like least	Table does not update, sometimes login tokens expired
about the system	and does not redirect to login
	need wait for the data to load
	Some pages require refresh
	takes time to load data
	need keep refresh
If you were to describe	Easy to use and implement
this site to a colleague in	a web app that can replace most of the paper work in
a sentence or two, what	warehouse
would you say?	
Do you have any other	very nice
final comments or	
questions?	

Table 7:45: Responses on open-ended questions

# 7.5 Traceability between Use Cases, Functional Requirements and Test Cases

This section will show the traceability between the use cases, functional requirements, web service test cases and user acceptance test cases. The use cases and the functional requirements are tabulated with specific ID in Section 7.5.1 and Section 7.5.2 respectively. Lastly, the traceability matrix is shown in Section 7.5.3.

# 7.5.1 Use Cases Table

Table 7.46 lists the use cases with use case ID and name.

Use Case ID	Use Case Name
UC001	Login account
UC002	Manage inventory
UC003	View all inventories
UC004	Manage warehouse
UC005	Manage user
UC006	Manage category
UC007	View reports
UC008	Manage cycle count approval report
UC009	Manage cycle counting
UC010	View all staff
UC011	Perform cycle counting
UC012	Record inventory

Table 7:46: Use Cases Table

Table 7.47 shows the functional requirements table with functional requirements ID and functional requirements statements.

Functional Requirement ID	Functional Requirement Statement
FR001	The system should allow admin to add, update and delete the inventory of a warehouse.
FR002	The system should allow admin to view, add, update, and delete a warehouse from the system.
FR003	The system should allow admin to view, add, update and delete a user in the system.
FR004	The system should allow admin to view, add,
	update and delete a category from the system
FR005	The system should allow admin to view daily reports of in-and-out inventories of all warehouses.
FR006	The system should allow admin to view the summary report for cycle counting of all warehouses.
FR007	The system should allow managers to start a new cycle counting process.
FR008	The system should allow managers to view all upcoming cycle counting SKUs.
FR009	The system should allow managers to view current cycle counting settings.
FR010	The system should allow a manager to view daily reports of in-and-out inventories of his or her warehouse only.
FR011	The system should allow managers to view summary reports for cycle counting of his or her warehouse only.

Table 7:47: Functional requirements table

FR012	The system should allow managers to view, approve or reject cycle count pending approval reports.
FR013	The system should allow managers to view all staffs in the warehouse.
FR014	The system should allow staff to check in-and-out of the inventory stock.
FR015	The system should allow staff to view cycle count schedules that are assigned to them.
FR016	The system should allow staff to perform cycle counting on the SKU assigned.
FR017	The system should allow admin, manager and staff to login into their account.
FR018	The system should allow the admin, manager and staff to view all inventories in the system.
FR019	The system should calculate inventory record accuracy (IRA) of each SKU and generate summary reports for each cycle count input from staff.
FR020	The system should classify SKUs, calculate the number of SKUs to be counted daily for each class, generate schedules and assign to staff available evenly.

# 7.5.3 Traceability Matrix

Traceability follows the requirements' life if it is done well. It started from the time requirements originated and continued on to fulfil the requirement. Hence, traceability ensures that the requirements fulfil the intended goals. Besides, traceability simplifies decision-making by allowing project teams to understand how the software design is impacted by the requirements. For instance, if there is a change in requirements, the impact of the changes across development can be easily analysed.

The traceability matrix maps the use cases, functional requirements, web service test cases and UAT test cases using their ID as shown in Table 7.48 to make sure that the web service testing and the UAT testing covered all use cases and functional requirements. The use case ID was formalized in Table 7.46 under Section 7.5.1, whereas the functional requirement ID was formalized under Section 7.5.2, Table 7.47. Web service testing IDs and their title were described under Section 7.2, Table 7.1. The test cases of the user acceptance test were covered in Appendix B.

Use Case ID	Functional	Web Service Testing	User Acceptance
	Requirement ID	ID	Test Case ID
UC001	FR017	WST001, WST002,	TC001, TC013
		WST023	
UC002	FR001	WST005, WST006,	TC008, TC009,
		WST007, WST008,	TC024
		WST009	
UC003	FR018	WST003, WST004	TC010
UC004	FR002	WST010, WST011,	ТС004, ТС007,
		WST012, WST013,	TC027, TC011,
		WST014, WST015,	TC012
		WST016	
UC005	FR003	WST017, WST018,	ТС005, ТС006,
		WST019, WST020,	TC025
		WST021, WST022	

 Table 7:48:
 Traceability matrix between use cases, functional requirements, web service testing and user acceptance testing

UC006	FR004	WST024, WST025,	TC002, TC003,
		WST026, WST027	TC026
UC007	FR005, FR006,	WST028, WST033	TC023, TC022
	FR010, FR011		
UC008	FR012, FR019	WST029, WST030,	TC020, TC021
		WST031, WST032,	
UC009	FR007, FR008,	WST034, WST035,	TC014, TC015
	FR009, FR020	WST036	
UC010	FR013	WST037	TC016
UC011	FR015, FR016	WST038	TC017
UC012	FR014	WST039, WST040	TC018, TC019

## **CHAPTER 8**

# CONCLUSIONS AND RECOMMENDATIONS

### 8.1 Introduction

From the planning phase to the closing phase, this project took approximately 7 months to complete. This project aims to develop a web-based warehouse inventory management system for stock item tracking. In the planning and analysis phase, problems were researched and formulated to come out with the project objectives, proposed solution, project approach and scope. Literature reviews were also done which researched similar projects, different cycle counting method and their application and lastly the software development methodology. A questionnaire was also distributed to warehouse users to gather requirements and develop functional and non-functional requirements. To plan out the schedule and keep track of the project progress, a web breakdown structure (WBS) and Gantt chart were used. In the design phase, several diagrams were drawn to illustrate the design of the system. Involve diagram and interface flow diagram.

Moving to the development phase, the whole system was broken down into three phases, following the phased-development methodology. The first phase focuses on the setting up of frameworks, preparing data, authentication of users, and the development of the first objective, which is the real-time check-in/out module. The second phase focused on the auto-scheduler for the cycle counting and the third phase covered the rest of the modules, which includes the report management, and all CRUDs. During each development phase, testing will be done to ensure that there were no errors before proceeding to the next phase.

For the closing phase, web service testing, user acceptance testing, usability testing, and testing for the real-time check-in/out module were done. An average SUS score of 85.5 was calculated, which is considered excellent in terms of the usability of the system. For the user acceptance testing, most test cases passed except one of the test cases (TC013), which was fixed after the testing. There were some comments received on passed test cases that were also fixed according to their feedback.

Responses from both user acceptance testing and usability testing will be considered for future works.

In later sections of this chapter, the objectives achievement, project limitations and recommendations for future works will be covered.

# 8.2 **Objectives Achievement**

In general, all objectives were achieved and delivered successfully and the details will be explained in this section.

# 8.2.1 Objective 1

Objective 1 aimed to develop an application to record all in-and-out of the warehouse inventories from different locations in real-time. This is achieved as the check-in/out module is developed in the implemented web-based warehouse inventory management system. The real-time functionality was also implemented using the Pusher Channels, as described in Chapter 6.2.3. This was tested in the Chapter 7.3 in which the check-in/out action was performed 100 times and the time difference between the time creating a new inventory stock and receiving the API message from Pusher Channels was recorded and tabulated in Table 7.42. The average time difference is around 1 second, which achieves the objective 1 with a delay time of less than 2 seconds.

# 8.2.2 Objective 2

Objective 2 is to develop an automated scheduler that can categorise SKUs, create schedules, and allocate them to cycle counting. The automated scheduler was successfully implemented in the system which can automatically classify the inventories selected for cycle counting to their respective cycle count class by calculating their stock value, create schedules according to workdays given by the manager and assign them to the allocated personnel evenly once the manager starts a new cycle counting process.

# 8.2.3 Objective 3

Objective 3 aims to propose and implement an improvement over existing SKU classification to achieve a higher degree of completeness and efficiency. Objective 3 was achieved by implementing the improved classification method by replacing the V,

E, and D classes from the ABC-VED method with priorities 1, 2, 3, 4, 5 and so on. As a result, the classification now is more complete as it is not only limited to the nine subclasses, instead, it can be more numbers of subclasses based on the maximum priority allowed in the system.

Aside from improvement for a more complete SKU classification from Objective 3, the efficiency was also ensured. The conventional ABC-VED classification method is to classify the selected inventories using the ABC classification method and VED classification method separately, which means that each inventory will hold two alphabets, one is A or B or C and another is V or E or D. Combining the two classes of each inventory will form the 9 subclasses AV, AE, AD, BV, BE, BD, CV, CE, and CE. Finally, they will be further classified into three classes I, II, and III as shown in Chapter 2 Table 2.3. To implement the conventional ABC-VED, this can be done by simply hardcoding the if-else statement for the classification of subclasses to the group I, II and III as shown in Figure 8.1. However, the efficiency of this algorithm is very dependent on the number of subclasses. If it was being extended in which there will be more classes instead of only V, E and D classes for the criticality of the inventories, then, there will be more subclasses to be classified into the final three classes (I, II and III) for cycle counting. For instance, if there are 1000 subclasses, then it is not possible to hardcode the if-else statement.

```
if (subclass=="AV"||subclass=="AD"||subclass=="BV"||subclass=="CV"){
    sku.class = "I"
    }else if (subclass=="BE"||subclass=="BD"||subclass=="CE"){
    sku.class = "II"
    }else{
    sku.class = "III"
}
```

Figure 8.1: Coding implementation of ABC-VED

# 8.2.3.1 SKU Classification Complexity

We will examine the complexity of these SKU classifications by their algorithm in terms of pseudo-code in order to have a more in-depth discussion.

# 8.2.3.1.1ABC and ABC-VED Classification

Table 8:1: Pseudocode of ABC and	ABC-VED algorithm
Table 5.1. I seudocode of ADC and	ADC-VED algorithm

ABC approach:	ABC-VED approach:		
Input: N number of SKUs, thresholds	Input: N number of SKUs with quantity		
$t_A = 0.2, t_C = 0.5$	in hand, criticality classes {V, E, D},		
	$t_A = 0.2, t_C = 0.5$		
1. For each SKU in SKUs: Compute SKU stock value; SV = unit cost of SKU × Quantity in hand; Output all computed SKU stock value as $[SV_1, SV_2,, SV_N]$ ; 2. Let Max stock value = $max([SV_1, SV_2,, SV_N])$ ; 3. Let Min stock value = $min([SV_1, SV_2,, SV_N])$ ; 4. For each $SV$ in $[SV_1, SV_2,, SV_N]$ : Compute the transform value $TV =$ $\frac{SV - Min \operatorname{stock value}}{\operatorname{Max stock value} - Min \operatorname{stock value}}$ ; where $TV \in [0,1]$ ; If $TV \leq t_A$ : categorize TV in class A Elseif $TV \geq t_C$ :	•		
categorize <i>TV</i> in class C Else: categorize <i>TV</i> in class B	categorize <i>TV</i> in class C Else: categorize <i>TV</i> in class B		
Record the class where TV is categorized in	Record the class where TV is categorized in		

<ol> <li>Let Z be the all the classes for each TV is categorized in.</li> </ol>	<ol> <li>Let X∈{A, B, C} be the class where <i>TV</i> is categorized in STEP 4, and Y∈{V,E,D};</li> </ol>
	6. Categorize <i>TV</i> in XY sub-class based on the item criticality;
	<ul> <li>7. Compute all possible pair-wise combination (sub-class) of X and Y, i.e., output all possible sub-classes as S = [AV, AE, AD, BV, BE, BD, CV, CE, CD]</li> </ul>
	8. Using S to construct the reference classes $C1= \{AV, AE, AD, BV, CV\}$ , $C2=\{BE, BD, CE\}, C3=\{CD\}$
	<ol> <li>For each sub-class in S: If sub-class =XY∈ C1, categorize TV in group I</li> </ol>
	<b>Elseif</b> sub-class =XY∈ C2, categorize TV in group II
	<b>Else</b> categorize TV in group III
	Record the group where TV is categorized in
	10. Let Z be the all the groups for each TV is categorized in.
Output: Z	Output: Z

By looking at the comparison of pseudocode, the ABC-VED algorithms involved additional steps to classify all SKUs into their respective cycle count class. For the ABC algorithm, there are two for-loops (step-1 and step-4) involved, hence, the complexity of the ABC algorithm is O(N) + O(N), which results to O(N). On the other hand, the ABC-VED algorithm has additional steps to compute the cycle count class of the SKUs as compared to the ABC algorithm. In step-7 of the ABC-VED pseudocode, it will loop for 9 times following the number of subclasses (combination of ABC classes with VED classes). Since the number of subclasses is fixed in the conventional ABC-VED algorithm, the complexity of this step is O(1). Therefore, the complexity of this algorithm is O(N)(step1) + O(N)(step4) + O(1)(step7) + O(N)(step9), which results to O(N) as the final complexity. In this case, the complexity of both the ABC and ABC-VED algorithm is the same.

This conventional classification limits the criticality of the inventories, in which the inventories can only be either vital(V), essential(E), or desirable(D). Therefore, the completeness of this algorithm is improved in Objective 3 by allowing the criticality classes to have more than three classes. However, if the number of criticality classes is not fixed, the complexity in step-7 of the ABC-VED pseudocode will be O(M), with M as the number of criticality class. This is because the complexity now will be O(3M) as the for-loop will now loop for  $3 \times M$  times to get all combinations of subclasses and the complexity of the algorithm is increased to be O(N + M). This means that the complexity depends on N (number of SKUs) and M (number of criticality classes).

# 8.2.3.1.2 Proposed Classification Method

Table 8:2: Pseudocode of proposed classification algorithm

**Proposed approach** Input: N number of SKUs with quantity in hand, criticality classes {1,2, ... M},  $t_A = 0.2, t_C = 0.5$ 1. For each SKU in SKUs: Based on the SKU criticality, assign a value of  $P \in \{1, 2, ..., M\}$  to it, i.e, higher value of P means higher criticality; stock value SV' = unit cost of SKU × Compute new SKU Quantity in hand  $\times P$ ; Set all computed SKU stock values as  $[SV'_1, SV'_2, ..., SV'_N]$ ; 2. Let Max stock value = max( $[SV'_1, SV'_2, ..., SV'_N]$ ); 3. Let Min stock value = min( $[SV'_1, SV'_2, ..., SV'_N]$ ); 4. For each SV' in  $[SV'_1, SV'_2, ..., SV'_N]$ : Compute the new transform value  $TV' = \frac{SV' - \text{Min stock value}}{\text{Max stock value} - \text{Min stock value}}$ , where  $TV' \in [0,1]$ ; If  $TV' \leq t_A$ : categorize TV' in class A **Elseif**  $TV' \ge t_C$ : categorize TV' in class C **Else**: categorize TV' in class B Record the class where TV is categorized in 5. Let Z be the all the classes where each TV is categorized in. Output: Z

In the proposed classification method, the criticality class is no longer limited to only 3 classes (V, E and D) but now is numerically represented (priority: from 1 to M). By doing this, the priority can be combined with the calculation of stock value in the ABC classification method. This direct computation reduced the complexity by eliminating the necessity of computing all possible pair-wise combinations of ABC classes and VED classes into the final cycle count class (step-6 to step-9 in ABC-VED pseudocode). With appropriated threshold  $t_A$  and  $t_C$ , the categorization of the transform stock value simply follows the ABC categorization method, hence the proposed improvement method enjoys both ABC and VED SKU classification benefits with an extension of a higher degree of criticality classes, up to arbitrary M criticality classes, and preserved the system efficiency with complexity order of O(N).

# 8.3 **Project Limitations**

There are some limitations noticed while using the system. The first limitation is that the loading time for displaying the rows of the table is usually 2 to 3 seconds. This is because the API endpoints needed some time for fetching the data from the database, especially when the data is large.

The second limitation is the storage bins creation. This is because the categories are shared among warehouses, hence, the storage bins will be created evenly for each of the categories in the database. However, this is not realistic as some warehouses may not store inventory for certain categories. Besides, warehouses might not have the same number of bins for each zone as well.

The third limitation is that the application will log the user out after 1 hour. This will cause the user to keep logging into the system after the login token expires, which is very inconvenient in day-to-day operations.

The fourth limitation is that only the check-in/out module is updated in realtime. The implemented system only creates events for check-in/out action that will broadcast the API message to other ends. This is because the check-in/out module is very crucial to keep track of the inventory stock accurately to prevent a situation like finding out that the actual quantity on hand in the warehouse is actually lesser than the quantity on hand stated in the system due to the reason that some stocks have been checked out from the other end and the user did not refresh the page. However, other modules should also be updated in real-time to eliminate the hassle of keep on refreshing the page to get the updated information.

# 8.4 **Recommendations for future work**

This system should be continuously enhanced and improved so that it can be more stable and provide a better user experience in the future. Hence, this section proposes some recommendations for future work (but not limited to) in Table 8.3.

Recommendations	Description
Enhance the flexibility of storage	The implemented system should be able to allow
bins creation while creating new	users to choose number of bins for each zone
warehouse	and assign category to each bins according to
	their preference. This gives convenience to
	warehouses that have inconsistent number of
	bins for each zone.
Add create storage bin feature	The implemented system should allow users to
	add extra storage bin after the warehouse has
	been created and has the flexibility to add it in a
	specified location in the list of storage bins. This
	feature is important as sometime warehouses
	may be expanded or renovated to have extra bins
	to be added to the system.
Add remember me token for	The implemented system should allow user to
authentication	keep on accessing the system if the user chose to
	remember his or her login while logging in. This
	provides better user experience as the user will
	not need to keep logging in to the system after
	the login token expired.
Implement real-time feature to all	The implemented system should be able to
modules of the system	update all data in real time. This feature will
	allow users to always keep track of the latest
	information and updates made from other ends.
	This is crucial especially for actions such as
	deleting action because the user might happen to
	perform some updates on an item which was

Table 8:3: Recommendations for future work

deleted by other user on other end. Then, faulty
data will be created which will cause an error to
occur.

## REFERENCES

Ceylan, Z. and Bulkan, S., 2022. Drug Inventory Management of a Pharmacy using ABC and VED Analysis. *Eurasian Journal of Health Technology Assessment*, 2(1), p.15.

Chinn, A., 2022. What's the System Usability Scale (SUS) & How Can You Use It?. [online] HubSpot. Available at: <a href="https://blog.hubspot.com/service/system-usability-scale-sus">https://blog.hubspot.com/service/system-usability-scale-sus</a> [Accessed 31 August 2022].

Chung, L., Nixon, B.A., Yu, E. and Mylopoulos, J., 2012. *Non-functional requirements in software engineering* (Vol. 5). Springer Science & Business Media.

Curley, R., 2015. *4 Best Practices For Cycle Count Procedures* [online]. Available at <<u>https://info.archon-interactive.com/blog/4-best-practices-for-cycle-count-procedures</u> > [Accessed on 26 February, 2022].

Curley, R., 2016. Paper-based processes vs. warehouse management systems [online]. Available at <<u>https://info.archon-interactive.com/blog/paper-based-processes-vs.-</u> warehouse-management-systems > [Accessed 8 April 2022].

Dennis, A., Wixom, B. and Tegarden, D., 2015. Systems analysis and design: An object-oriented approach with UML. John wiley & sons.

Doglio, F., 2022. How to secure a REST API using JWT. [online] LogRocket. Available at: <a href="https://blog.logrocket.com/how-to-secure-a-rest-api-using-jwt-7efd83e71432/">https://blog.logrocket.com/how-to-secure-a-rest-api-using-jwt-7efd83e71432/</a>> [Accessed 23 August 2022].

Drakeley, P., 2021. *How to Calculate ABC Classification - A Working Example* [online]. Available at: <<u>https://www.eazystock.com/uk/blog-uk/abc-classification-calculation-inventory-management/</u> > [Accessed on 9 March, 2022].

English, L. and Yunusa-Kaltungo, A., 2022. A practical application of methodologies

to determine asset criticality and work order prioritization. Engineering Reports,

p.e12555.

Fathoni, F.A., Ridwan, A.Y. and Santosa, B., 2019, March. Development of inventory control application for pharmaceutical product using abc-ved cycle counting method to increase inventory record accuracy. In 2018 International Conference on Industrial Enterprise and System Engineering (ICoIESE 2018) (pp. 266-271). Atlantis Press.

F Leal, L., Zancul, E. and Schützer, K., 2021. Industry 4.0 Learning Factory Phased Development. *Eduardo and Schützer, Klaus, Industry*, 4.

Harjono, H., 2019. Warehouse management system and its impact on data accuracy.[online]PresidentUniversity.Availableat<<u>https://budi.rahardjo.id/files/courses/2020/DL-023201905028-Report-Heru-<br/>Rev1.pdf</u> > [Accessed 3 March 2022].Example 100 - 100

Hodgson, R., 2019. Paper-based warehousing costs you more than money [online]. Available at <<u>https://www.theaccessgroup.com/en-gb/blog/paper-based-warehousing-costs-you-more-than-money/</u>>[Accessed on 8 April 2022]

IEEE Computer Society. Software Engineering Standards Committee and IEEE-SA Standards Board, 1998. *IEEE recommended practice for software requirements specifications* (Vol. 830, No. 1998). IEEE.

International Standards Organisation. ISO 55000:2014 Asset Management - Overview, Principles and Terminology. ISO; 2014.

James, K., 2016. Design and implementation of an automated inventory management system case study: Smart Shoppers Masaka. Bachelor. Kampala International University. Available at: < <a href="https://ir.kiu.ac.ug/bitstream/20.500.12306/2297/1/DESIGN%20AND%20IMPLEM">https://ir.kiu.ac.ug/bitstream/20.500.12306/2297/1/DESIGN%20AND%20IMPLEM</a> ENTATION%20OF%20AN%20AUTOMATED%20INVENTORY%20MANAGE MENT%20SYSTEM.pdf > [Accessed 20 March 2022].

Jenkins, A., 2022. 20 Inventory Management Challenges And Solutions For 2022 And Beyond [online]. Available at <<u>https://www.netsuite.com/portal/resource/articles/inventory-</u> management/inventory-management-challenges.shtml > [Accessed on 26 February 2022].

Maharjan, A. S. and Humagain, M., 2016. Inventory Management System. Bachelor.PokharaUniversity.Availableat:<https://www.academia.edu/26003928/Final\_Year\_Project\_On\_Inventory\_Managem</td>ent\_System\_Submitted\_By > [Accessed 20 March 2022].

Mahtamtama, E., Ridwan, A.Y. and Santosa, B., 2018, October. Development Of Cycle Counting Monitoring Dashboard With Buffer Time Management For Cocoa Company. In 2018 12th International Conference on Telecommunication Systems, Services, and Applications (TSSA) (pp. 1-5). IEEE.

Ministry of Economic Affairs, 2019. Shared Prosperity Vision 2030. [pdf] Putrajaya:MinistryofEconomicAffairs.Availableat:<<u>https://www.epu.gov.my/sites/default/files/2020-02/Shared%20Prosperity%20Vision%202030.pdf>[Accessed 11 March 2022].</u>

NC-Vision, 2021, *Top 3 Warehouse Management Challenges to Watch Out for In 2021* [online]. Available at: <<u>https://nc-vision.com/top-3-warehouse-management-</u> <u>challenges-to-watch-out-for-in-2021and-how-to-overcome-them/</u> > [Accessed on 27 February, 2022].

Nielsen, J., & Landauer, T.K. (1993). A mathematical model of the finding of usability problems. Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems.

Ramaa, A., Subramanya, K.N. and Rangaswamy, T.M., 2012. Impact of warehouse management system in a supply chain. *International Journal of Computer Applications*, 54(1).

Reyes, J., Alvarez, K. and Vasquez, R., 2016. Dynamic buffer management for raw material supply in the footwear industry. *Journal of Industrial and Intelligent Information Vol*, 4(1).

Sasmito, G.W., Wibowo, D.S. and Dairoh, D., 2020. Implementation of Rapid Application Development Method in the Development of Geographic Information Systems of Industrial Centers. *Journal of information and communication convergence engineering*, *18*(3), pp.194-200.

Schwarz, L., 2021.. *Inventory Cycle Counting 101: Best Practices & Benefits* [online]. Available at <<u>https://www.netsuite.com/portal/resource/articles/inventory-management/using-inventory-control-software-for-cycle-counting.shtml#:~:text=What%20Is%20Cycle%20Counting%3F,have%20counted% 20all%20their%20goods > [Accessed on 26 February 2022].</u>

Shaydulin, R. and Sybrandt, J., 2017. To agile, or not to agile: A comparison of software development methodologies. *arXiv preprint arXiv:1704.07469*.

Shen, Y., 2017. *Applied Cycle Counting Method for Grocery Retail Industry* (Doctoral dissertation, Pennsylvania State University).

Tahir, D., 2020. *Design and implementation of an inventory management system for Walid Halal Spices*. Bachelor. Baze University. Available at: < https://portal.bazeuniversity.edu.ng/student/assets/thesis/20210215004507841505869 .pdf > [Accessed 20 March 2022].

WiSys, n.d, *3 Common Inventory Management Challenges* [online]. Available at: <<u>https://www.wisys.com/3-common-inventory-management-challenges/</u> > [Accessed on 26 February, 2022].

Zhou, J.Y., 2004. Functional requirements and non-functional requirements: a survey (Doctoral dissertation, Concordia University).

# **APPENDICES**

## APPENDIX A: Questionnaire Form

9/8/22, 3:15 PM

A WEB-BASED WAREHOUSE INVENTORY MANAGEMENT SYSTEM FOR STOCK ITEMS TRACKING

# A WEB-BASED WAREHOUSE INVENTORY MANAGEMENT SYSTEM FOR STOCK ITEMS TRACKING

Dear respondents,

I am Elaine Low Jing Yi, a year 3 undergraduate student of Bachelor of Science (HONS) Software Engineering from Lee Kong Chian Faculty of Engineering & Science (LKCFES), Universiti Tunku Abdul Rahman (UTAR), Sungai Long Campus.

The aim of this questionnaire is to gather opinions on impact of data accuracy of warehouse inventories and issue regarding physical inventory counts and cycle counting. This will contributes to my final year project (FYP), which is to develop a web-based warehouse inventory management system particularly for stock item tracking and cycle counting.

#### Respondents must fulfill the criteria listed below:

- own or working in a warehouse
- Understand basic operations in the warehouse
- ability to understand English

#### This questionnaire will consists of four sections:

- Section A: Demographic Information
- Section B: General Information
- Section C: Inventory Accuracy
- Section D: Physical counting/ Cycle counting

The approximate time to complete the questionnaire is around **5 to 10 minutes**. All of the responses will stay anonymous and the data collected will only be used in requirements gathering for my final year project.

If you are willing to participate in this questionnaire, please answer all of the questions that you feel relatable and leave out those that do not apply to your scenario.

If you have any questions on this questionnaire, kindly contact me via <u>elainelowjingyi00@1utar.my</u>. Thank you for your time in answering this questionnaire. Your responses is highly appreciated as it greatly assists me in my final year project.

\*Required

1. I have read the information above. By clicking on the "Agree", I am consenting to \* participate in this survey/questionnaire project.

Mark only one oval.

O Agree

Section A: Demographic Information

2. Name

3. Gender \*

Mark only one oval.

Male Female

4. Age \*

Mark only one oval.

C	21-30
C	31-40
C	41-50
C	51-60
C	61 and above

https://docs.google.com/forms/d/1amQ\_5wvZmrx8x1jzv9rly-tBgePyK4A6zMhbX5p\_yYA/edit

2/9

9/8/22.	3.15	PM
8/0/22,	3.15	1.141

A WEB-BASED WAREHOUSE INVENTORY MANAGEMENT SYSTEM FOR STOCK ITEMS TRACKING

5. Race \*

Mark only one oval.

Malay	
Chinese	
Indian	
Other:	

6. Email

Section B: General Information

7. Do you operate your own warehouse?\*

Mark only one oval.

Yes

8. Do you operate a warehouse for others? \*

Mark only one oval.

C	) Yes	
C	No	

9. Is there any warehouse management system in use? \*

Mark only one oval.

C	$\supset$	Yes
C		No

https://docs.google.com/forms/d/1amQ\_5wvZmrx8x1jzv9rly-tBgePyK4A6zMhbX5p\_yYA/edit

3/9

9/8/22.	3:15	PM

10. More than one warehouse from different location/sites? \*

Mark only one oval.

C	Yes
C	No

11. Warehouse Name/Location (insert multiple if more than one)

12. What kind of commodities are stored in the warehouse(s)?\*

13. What is your role/position in warehouse management?\*

Section C: Inventory accuracy

https://docs.google.com/forms/d/1amQ\_5wvZmrx8x1jzv9rly-tBgePyK4A8zMhbX5p\_yYA/edit

#### 9/8/22, 3:15 PM

14. Methods of inventory recording \*

Mark only one oval.

Paper records
Excel / online spreadsheets
Using a Warehouse Management System (WMS) or any similar softwares
Other:

15. Is there any discrepancies between inventory count recorded and the actual \* count?

Mark only one oval.

C	Yes
$\subset$	No

16. Is there any inconsistency of inventory count across multiple location/sites?\*

Mark only one oval.

C	Yes	
C	No	

17. Would inventory and data accuracy affect business operation? \*

Mark only one oval.

C	) Yes	
C	No	

https://docs.google.com/forms/d/1amQ\_5wvZmrx8x1jzv9rly-tBgePyK4A6zMhbX5p\_yYA/edit

5/9

9/8/22,	3:15	PM

18. If yes, why do you think inventory accuracy would affect business operation?

Tick all that apply.

Increase cost on inventory holding for excess inventory

Additional time on researching discrepancies and correct data

Increase risks of stock running-out

Increase risk of cutting orders or rescheduling the production line, which impacts the service to customers

19. In your opinion, what is the main cause of inaccuracy of inventory record? \*

### Section D: Physical counting/ Cycle counting

20. For inventory controls, which of the following method is implemented in your \* warehouse?

Mark only one oval.

Physical inventory count (halts operation and count annually)

Cycle counting

All of the above

Other:

Instruction: Please answer all questions by choosing the most appropriate answer. 1 = Strongly Disagree 2 = Disagree 3 = Neutral 4 = Agree 5 = Strongly Agree

https://docs.google.com/forms/d/1amQ\_5wvZmrx8x1jzv9rly-tBgePyK4A6zMhbX5p\_yYA/edit

6/9

#### 9/8/22, 3:15 PM

21. Manual inventory recording/ counting without WMS system is troublesome. \*

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly agree

22. It is difficult to classify SKUs manually. \*

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly agree

23. Extra manpower and resources will be needed to analyze and schedule for \* physical counting and cycle counting.

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly agree

24. It is difficult to keep track of the cycle counting progress manually. \*

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly agree

https://docs.google.com/forms/d/1amQ\_5wvZmrx8x1jzv9rly-tBgePyK4A8zMhbX5p\_yYA/edit

25. Continuous operations and incoming transactions while performing physical \* count/cycle count may cause false variances in inventory counts.

Mark only one oval.						
	1	2	3	4	5	
Strongly disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly agree

26. There will be a lot of hassle if were to deal with all of the physical reports \* regarding physical counts/ cycle counting.

Mark only one oval.						
	1	2	3	4	5	
Strongly disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly agree

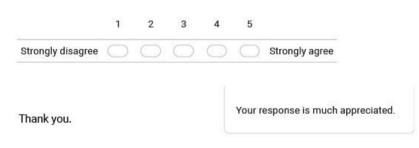
27. Adjusting inventory record should be in real-time to avoid inconsistencies. \*

Mark only one oval.

	1	2	3	4	5	
Strongly disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly agree

28. Extra effort will be needed to calculate inventory record accuracy (IRA) to keep \* track of the performance.

Mark only one oval.



https://docs.google.com/forms/d/1amQ\_5wvZmrx8x1jzv9rly-tBgePyK4A6zMhbX5p\_yYA/edit

8/9

9/8/22, 3:15 PM

A WEB-BASED WAREHOUSE INVENTORY MANAGEMENT SYSTEM FOR STOCK ITEMS TRACKING



This content is neither created nor endorsed by Google.

Google Forms

https://docs.google.com/forms/d/1amQ\_5wvZmrx8x1jzv9rly-tBgePyK4A8zMhbX5p\_yYA/edit

			User Acceptance Testing	Form		
Tester's Name			¥	Testing Start Date/ Time Testing End Date/ Time		
Test Case ID	Module	Test Case Title	Test Steps	Expected Results	Status (Pass/Fail)	Comments
TC001	Login account	Login an account	<ol> <li>User enters username and password of ADMIN account in the login page.</li> <li>User clicks on the login button.</li> </ol>	User will successfully logged-in into the system.		
TC002	Manage category	Add a new category	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> <li>User clicks on the "Add" button on the top right of the Manage category screen.</li> <li>User enters the name for the new category.</li> <li>User clicks on "confirm" button.</li> </ol>	The new category will be added successfully with a success notification.		
TC003	Manage category	Update a category name	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> <li>User selects the newly created category from the displayed list and clicks on the "Edit" button.</li> </ol>	The name of the category will be updated successfully with a success notification.		

APPENDIX B: User Acceptance Testing Form

		category from the modal popup			
		4. After editing, user clicks on the			
		"Confirm" button.			
Manage	Add a new	1. User logins into the system	The warehouse is added		
warehouse	warehouse	using <b>ADMIN</b> account.	successfully with a		
		2. User clicks on the "Manage	success notification.		
		warehouse" tab from the top			
		navigation bar.			
		3. User clicks on the "Add"			
		button on the top right of the			
		Manage warehouse screen.			
		4. User enters the name, location,			
		number of bins (enter 100) and			
		zones (enter "A,B,C") for the			
		new warehouse.			
		5. User clicks on "confirm"			
		button after all information has			
		been entered.			
Manage	Add a new user	1. User logins into the system	Two new users will be		
user		using <b>ADMIN</b> account.	added successfully with		
		2. User clicks on the "Manage	a success notification.		
		user" tab from the top navigation			
		bar.			
		3. User clicks on the "Add"			
		button on the top right of the			
		1 0			
		4. User enters the name, email,			
		contact number, IC number, role			
		(select "STAFF"), employed in,			
		address and username for the new			
		user.			
	Warehouse	warehousewarehouseManageAdd a new user	Manage warehouseAdd a new warehouse1. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location, number of bins (enter 100) and zones (enter "A,B,C") for the new warehouse. 5. User clicks on "confirm" button after all information has been entered.Manage userAdd a new user 1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user and a new userManage userAdd a new user 1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage user screen. 4. User enters the name, email, contact number, IC number, role (select "STAFF"), employed in, address and username for the new	Manage warehouseAdd a new warehouse1. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location, number of bins (enter 100) and zones (enter "A,B,C") for the new warehouse. 5. User clicks on the "Manage user "button after all information has been entered.The warehouse is added successfully with a success notification.Manage userAdd a new user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar. 3. User clicks on the "Manage user" tab from the top navigation bar. 3. User clicks on the "Manage user" tab from the top navigation bar. 3. User clicks on the "Manage user" tab from the top navigation bar. 3. User clicks on the "Manage user" tab from the top navigation bar. 3. User clicks on the "Manage user" tab from the top navigation bar. 3. User clicks on the "Manage user" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage user screen. 4. User enters the name, email, contact number, IC number, role (select "STAFF"), employed in, address and username for the newTwo new users	Manage warehouseAdd a new warehouseI. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location, number of bins (enter 100) and zones (enter "A,B,C") for the new warehouse. 5. User clicks on the "Manage user" tab from the top system using ADMIN account.The warehouse is added successfully with a success notification.Manage userAdd a new userI. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.Two new users will be added successfully with a success notification.Manage userAdd a new userI. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage user screen. 4. User enters the name, email, contact number, role (select "STAFF"), employed in, address and username for the newTwo new users will be address and username for the new

1	1				1	
			5. For the warehouse input, user			
			selects the newly created			
			warehouse.			
			6. User clicks on "confirm"			
			button after all information has			
			been entered.			
			7. Repeat step 3 to 6 to create			
			another user with STAFF role.			
TC006	Manage	Update a user's	1. User logins into the system	The details of the		
	user	detail	using <b>ADMIN</b> account.	selected user will be		
			2. User clicks on the "Manage	updated successfully		
			user" tab from the top navigation	with a success		
			bar.	notification.		
			3. User selects a newly created			
			user from the displayed list and			
			clicks on the "Edit" button.			
			4. User edits some information of			
			the user (except the warehouse			
			and role) on the modal pop-up			
			5. After editing, user clicks on the			
			"Confirm" button.			
TC007	Manage	Update a	1. User logins into the system	The warehouse details		
	warehouse	warehouse	using <b>ADMIN</b> account.	are updated		
			2. User clicks on the "Manage	successfully with a		
			warehouse" tab from the top	success notification.		
			navigation bar.			
			3. User selects the <b>newly created</b>			
			warehouse from the displayed			
			list and clicks on the "Edit"			
			button.			
			4. User edits the information by			
			adding a newly created user in			

				the warehouse manager field on		
				the modal pop-up		
				5. After editing, user clicks on the		
				"Confirm" button.		
TC008	Manage	Add	new	1. User logins into the system	Inventories are added	
	inventory	inventory		using <b>ADMIN</b> account.	successfully and a	
				2. User clicks on the "Manage	notification of the bin	
				inventory" tab from the top	number assigned to	
				navigation bar.	newly added inventory	
				3. User clicks on the "Add"	is shown.	
				button on the top right of the		
				Manage inventory screen.		
				4. For warehouse and category		
				fields, selects the warehouse		
				and category created earlier.		
				4. User enters the name, quantity		
				on hand, cost per unit and priority		
				for the inventory.		
				5. User clicks on "confirm"		
				button after all information has		
				been entered.		
				6. Repeat step 3 to 5 to create one		
				more inventory.		
TC009	Manage	Update	an	1. User logins into the system	The selected inventory	
	inventory	inventory		using <b>ADMIN</b> account.	details are updated	
				2. User clicks on the "Manage	according to user's	
				inventory" tab from the top	input.	
				navigation bar.	*	
				3. User selects a newly created		
				<b>inventory</b> from the displayed list		
				and clicks on the "Edit" button.		

		the inventory on the modal pop-			
		up			
		5. After editing, user clicks on the			
		"Confirm" button.			
View all	View inventory	1. User logins into the system	User will successfully		
inventories	and its details	using <b>ADMIN</b> account.	view the inventory list		
		2. User clicks on the "Manage	and will be redirected to		
		inventory" tab from the top	the inventory details		
		navigation bar.	page once an inventory		
		3. User view the list of	id is clicked.		
		inventories displayed.			
		4. User clicks on one of the			
		inventory ID to view the details.			
Manage	Edit inventory	1. User logins into the system	The selected inventory		
warehouse	to storage bin	using <b>ADMIN</b> account.	will be added to the		
	_	2. User clicks on the "Manage	selected storage bin of		
		warehouse" tab from the top	the warehouse.		
		navigation bar.			
		3. User clicks on <b>newly created</b>			
		warehouse's ID to view the			
		details.			
		4. In the warehouse details page,			
		user selects the "storage bin" tab			
		to view the list of storage bins in			
		the warehouse			
		5. User selects the bin with the			
		newly created inventory (can use			
		the search to search for the			
		inventory) and clicks the "Edit			
		inventory" button.			
	inventories	inventories and its details Manage Edit inventory	View all inventoriesView inventory and its details5. After editing, user clicks on the "Confirm" button.View all 	View all inventoriesView inventory and its detailsthe inventory on the modal pop- up 5. After editing, user clicks on the "Confirm" button.User will successfully view the system using ADMIN account. 2. User clicks on the "Manage inventory" tab from the top navigation bar. 3. User view the list of inventories displayed. 4. User clicks on one of the inventory ID to view the details.User logins into the system using ADMIN account. 2. User clicks on one of the inventory ID to view the details.Manage warehouseEdit inventory to storage bin1. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on newly created warehouse is ID to view the details.The selected inventory will be added to the selected storage bin of the warehouse.Manage warehouseEdit inventory to storage bin3. User clicks on newly created warehouse's ID to view the details.The selected inventory will be added to the selected storage bin of the warehouse.Manage warehouse5. User selects the "storage bin" tab to view the list of storage bins in the warehouse 5. User selects the bin with the newly created inventory (can use the search to search for the inventory) and clicks the "Edit	View all inventoriesView inventory and its detailsthe inventory on the modal pop- up 5. After editing, user clicks on the "Confirm" button.User will successfully view the inventory list and will be redirected to the inventory details page once an inventory id is clicked.Wiew all inventoriesView inventory and its details1. User logins into the system using ADMIN account. 2. User clicks on the "Manage inventories displayed. 4. User clicks on one of the inventory ID to view the details.User view the list of inventory ID to view the details.Manage warehouseEdit inventory to storage bin1. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on newly created warehouse" tab from the top navigation bar. 3. User clicks on newly created warehouse" tab from the top navigation bar. 3. User clicks on newly created warehouse is ID to view the details. 4. In the warehouse details page, user selects the 'storage bins in the warehouse 5. User selects the bin with the newly created inventory (can use the search to search for the inventory) and clicks the "EditImage inventory is clicks on the inventory is can use the search to search for the inventory) and clicks the "Edit

	1				
			6. User clears the current		
			inventory assigned on the modal		
			popup and click on the "Edit		
			inventory" button.		
TC012	Manage	Assign	1. User logins into the system	The category of the	
	warehouse	category to a	using <b>ADMIN</b> account.	selected bins is changed	
		storage bin	2. User clicks on the "Manage	to the selected category.	
			warehouse" tab from the top		
			navigation bar.		
			3. User clicks on <b>newly created</b>		
			warehouse's ID to view the		
			details.		
			4. In the warehouse details page,		
			user selects the "storage bin" tab		
			to view the list of storage bins in		
			the warehouse		
			5. User clicks on the "Multi		
			assign bin" to assign category to		
			bins.		
			6. User selects a category and		
			checks desired bin to assign.		
			7. User clicks on the "Assign"		
			button.		
TC013	Login	Reset password	1. User logins into the system	A success notification	
	account	-	using <b>ADMIN</b> account.	will be shown.	
			2. User clicks on the "Manage		
			User" tab and finds the newly		
			created user with "MANAGER"		
			role.		
			3. User takes note on the		
			username and contact number.		

· · · · · · · · · · · · · · · · · · ·					
			4. User logouts and logins into		
			the system using the username		
			and contact number as the		
			password.		
			5. User resets the password		
TC014	Manage	Start a new	1. User logins into the system	A success notification	
	cycle	cycle counting	using the created manager-role	will be shown stating	
	counting		user credentials (skip this step	that the cycle counting	
	_		if you completed TC013)	is started. Schedules	
			2. Note that there are <b>TWO</b>	will be created for all	
			inventories listed in the View	SKUs and assigned to	
			Inventory page, if no, kindly go	the selected staffs	
			back to TC008 to create an	evenly.	
			inventory.	-	
			3. User clicks on the "Start Cycle		
			counting" from the "Manage		
			Cycle counting" tab dropdown on		
			the top navigation bar.		
			3. User fills in all of the		
			information in the form		
			displayed.		
			4. User selects all inventories in		
			the "Select Inventories" field.		
			4. User clicks the "Submit"		
			button after completing the form.		
			5. User clicks the "Create" button		
			in the summary modal popup.		
TC015	Manage	View all	1. User clicks on the "View Cycle	The list of schedules for	
	cycle	upcoming cycle	counting" from the "Manage	the cycle counting and	
	counting	counting SKUs	Cycle counting" tab dropdown on	the settings will be	
	_	_	the top navigation bar.	displayed accordingly.	

	r				
			2. User clicks on the "View cycle		
			count settings" button on the top		
			right of the displayed table.		
TC016	View all	View all staff in	1. User clicks on the "View Staff"	The list of staffs and	
	staff	a warehouse	tab on the top navigation bar.	their details will be	
			2. User clicks on one of the staff	displayed accordingly.	
			ID from the list to view the staff		
			details.		
TC017	Perform	Perform cycle	1. Repeat TC013 to login and	The cycle counting	
	cycle	counting for an	reset password for the created	record is created with	
	counting	inventory	STAFF-role user.	success notification.	
			2. User clicks on the "Cycle		
			Counting" tab from the top		
			navigation bar.		
			3. User selects the "Upcoming"		
			tab and views the list of		
			upcoming cycle counting		
			schedules		
			3. User chooses the first one in		
			the list and clicks the "Count"		
			button.		
			4. User enters the actual count on		
			the modal popup		
			5. User clicks the "Create" button		
			6. User refresh the page.		
			7. Repeat step 3 to 6 to count		
			another one.		
TC018	Record	Check in stock	1. User clicks on the "Check	The selected	
	inventory	for an	In/Out Stock" tab from the top	inventory's quantity on	
		inventory	navigation bar.	hand will be updated in	
			2. User views the list of	real time without	
			inventories displayed.	refreshing the page.	

TC019	Record inventory	Check out stock for an inventory	navigation bar.	The selected inventory's quantity on hand will be updated in	
			<ol> <li>User views the list of inventories displayed.</li> <li>User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>User selects the "Check Out" tab in the modal popup.</li> <li>User enters the quantity and remarks and clicks the "Confirm" button.</li> </ol>	real time without refreshing the page.	
TC020	Manage cycle count approval report	Approve cycle count approval report	<ol> <li>User logins into the system using the created manager-role user credentials.</li> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Approval Report" tab and views the list of approval reports listed.</li> </ol>	The summary report for the selected cycle counting will be generated with IRA calculated.	

	1	1	1		1
			4. User clicks the "Approve" button for one of the approval		
			reports.		
			5. User clicks the "Confirm"		
			button in the confirmation modal.		
			6. User refresh the page.		
TC021	Manage	Reject cycle		A success message will	
10021	cycle	count approval		be shown to indicate the	
	count	report	navigation bar.	success rescheduling.	
	approval	report	2. User selects the "Cycle Count	success reseneduting.	
			Approval Report" tab and views		
	report		the list of approval reports listed.		
			3. User clicks the "Reject" button		
			for one of the approval reports.		
			4. User checks the checkbox for		
			reassigning staff to recount the		
			SKU.		
			5. User clicks the "Confirm"		
<b>T</b> C022			button in the confirmation modal.		
TC022	View	View cycle	1. User clicks on the "View	The list of summary	
	reports	counting	reports" tab from the top	reports and their details	
		summary report	0	will be displayed	
			2. User selects the "Cycle Count	accordingly.	
			Summary Report" tab and views		
			the list of summary reports listed.		
			3. User clicks on the cycle count		
			ID for one of the summary reports		
TC023	View	View daily	1. User clicks on the "View	The list of daily reports	
	reports	reports	reports" tab from the top	and their details and	
			navigation bar.	stocks will be displayed	
				accordingly.	

TC024       Manage inventory       Delete an inventory       1. User clicks on the "View" button for one of the daily reports       The selected inventory is successfully deleted from the top navigation bar.         3. User clicks on the "Manage inventory" tab from the top navigation bar.       3. User finds one of the newly created inventories from the displayed list (can utilize the successfully deleted inventor).       The status of the selected user user         TC025       Manage user       Delete a user       1. User logins into the system using ADMIN account.       The status of the selected user will be set to "INACTIVE".         TC025       Manage user       Delete a user       1. User logins into the system using ADMIN account.       The status of the selected user will be set to "INACTIVE".         TC025       Manage user       Delete a user       1. User logins into the organization bar.       The status of the selected user will be set to "INACTIVE".						
TC024       Manage inventory       Delete inventory       an inventory       1. User logins into the system using ADMIN account provided.       The selected inventory is successfully deleted from the top navigation bar.         3. User finds one of the newly created inventories from the displayed list (can utilize the filter or search function) and clicks on the "Delete" button on the confirmation modal.       The status of the selected user will be set to "INACTIVE".         TC025       Manage user       Delete a user       1. User clicks on the "Manage user" tab from the top navigation bar.       The status of the selected user will be set to "INACTIVE".				5 1		
TC024Manage inventoryDelete an inventory1. User clicks on the "View" button for one of the daily reportsThe selected inventoryTC024Manage inventoryDelete inventoryan inventory1. User logins into the system using ADMIN account provided. 2. User clicks on the "Manage inventory" tab from the top navigation bar. 3. User finds one of the newly created inventories from the displayed list (can utilize the filter or search function) and clicks on the "Delete" button on the confirmation modal.The status of the selected user will be set to "INACTIVE".TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				tab and views the list of daily		
TC024Manage inventoryDelete inventoryan inventory1. User logins into the system using ADMIN account provided. 2. User clicks on the "Manage inventory" tab from the top navigation bar. 3. User finds one of the newly created inventories from the displayed list (can utilize the filter or search function) and clicks on the "Delete" button. 4. User clicks on the "Delete" button. 4. User clicks on the "Delete" button. 4. User clicks on the "Delete" button modal.The status of the selected user will be set using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				1		
TC024Manage inventoryDelete inventoryan inventory1. User logins into the system using ADMIN account provided. 2. User clicks on the "Manage inventory" tab from the top navigation bar. 3. User finds one of the newly created inventories from the displayed list (can utilize the filter or search function) and clicks on the "Delete" button. 4. User clicks on the "Delete" button on the confirmation modal.The status of the selected user will be set to "INACTIVE".TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				3. User clicks on the "View"		
inventoryinventoryusing ADMIN account provided. 2. User clicks on the "Manage inventory" tab from the top navigation bar. 3. User finds one of the newly created inventories from the displayed list (can utilize the filter or search function) and clicks on the "Delete" button. 4. User clicks on the "Delete" button on the confirmation modal.is successfully deleted from the warehouse.TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				button for one of the daily reports		
2. User clicks on the "Manage inventory" tab from the top navigation bar.from the warehouse.3. User finds one of the newly created inventories from the displayed list (can utilize the filter or search function) and clicks on the "Delete" button.from the warehouse.TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".	TC024	Manage	Delete an	1. User logins into the system	The selected inventory	
Image		inventory	inventory	using <b>ADMIN</b> account provided.	is successfully deleted	
navigation bar. 3. User finds one of the newly created inventories from the displayed list (can utilize the filter or search function) and clicks on the "Delete" button. 4. User clicks on the "Delete" button on the confirmation modal.TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".		-	-	2. User clicks on the "Manage	from the warehouse.	
3. User finds one of the newly created inventories from the displayed list (can utilize the filter or search function) and clicks on the "Delete" button. 4. User clicks on the "Delete" button on the confirmation modal.TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				inventory" tab from the top		
TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				navigation bar.		
displayed list (can utilize the filter or search function) and clicks on the "Delete" button. 4. User clicks on the "Delete" button on the confirmation modal.The status of the selected user will be set to "INACTIVE".TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				3. User finds one of the newly		
filter or search function) and clicks on the "Delete" button. 4. User clicks on the "Delete" button on the confirmation modal.TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				created inventories from the		
Clicks on the "Delete" button. 4. User clicks on the "Delete" button on the confirmation modal.TC025Manage userDelete a user1. User logins into the system using <b>ADMIN</b> account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				displayed list (can utilize the		
4. User clicks on the "Delete" button on the confirmation modal.4. User clicks on the "Delete" button on the confirmation modal.TC025Manage userDelete a user1. User logins into the system using <b>ADMIN</b> account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				filter or search function) and		
Imagebutton on the confirmation modal.ImageImageImageTC025Manage userDelete a user1. User logins into the system using <b>ADMIN</b> account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				clicks on the "Delete" button.		
TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				4. User clicks on the "Delete"		
TC025Manage userDelete a user1. User logins into the system using ADMIN account. 2. User clicks on the "Manage user" tab from the top navigation bar.The status of the selected user will be set to "INACTIVE".				button on the confirmation		
user       using ADMIN account.       selected user will be set         2. User clicks on the "Manage       to "INACTIVE".         user" tab from the top navigation       bar.				modal.		
2. User clicks on the "Manage to "INACTIVE". user" tab from the top navigation bar.	TC025	Manage	Delete a user	1. User logins into the system	The status of the	
user" tab from the top navigation bar.		user		using <b>ADMIN</b> account.	selected user will be set	
bar.				2. User clicks on the "Manage	to "INACTIVE".	
				user" tab from the top navigation		
3. User selects one of the created				bar.		
				3. User selects <b>one of the created</b>		
<b>users</b> from the displayed list and				<b>users</b> from the displayed list and		
clicks on the "Delete" button.				clicks on the "Delete" button.		
4. User clicks on the "Delete"				4. User clicks on the "Delete"		
button on the confirmation				button on the confirmation		
modal.				modal.		
TC026 Manage Delete a 1. User logins into the system The selected category	TC026	Manage	Delete a	1. User logins into the system	The selected category	
category category using <b>ADMIN</b> account. will be deleted from the		U	category		will be deleted from the	
list.				_	list.	

				2. User clicks on the "Manage		
				category" tab from the top		
				navigation bar.		
				3. User selects the newly created		
				<b>category</b> from the displayed list		
				and clicks on the "Delete" button.		
				4. User clicks on the "Delete"		
				button on the confirmation		
				modal.		
TC027	Manage	Delete	а	1. User logins into the system	The selected warehouse	
	warehouse	warehouse		using <b>ADMIN</b> account.	is successfully deleted	
				2. User clicks on the "Manage	with a success message.	
				warehouse" tab from the top	C	
				navigation bar.		
				3. User selects the newly created		
				warehouse from the displayed		
				list and clicks on the "Delete"		
				button.		
				4. User clicks on the "Delete"		
				button on the confirmation		
				modal.		

## APPENDIX C: User Acceptance Testing Results

## Expert testers

			User Acceptance Testing	Form			
Tester's Name	Teh Khai M	ling	· · · · · · · · · · · · · · · · · · ·	TestingStartDate/27/8/2022 10:07pmTime27/8/2022 10:07pm			
		1		Testing End Date/ Time			
Test Case ID	Module	Test Case Title	Test Steps	Expected Results	Status (Pass/Fail)	Comments	
TC001	Login account	Login an account	<ol> <li>User enters username and password of ADMIN account in the login page.</li> <li>User clicks on the login button.</li> </ol>	User will successfully logged-in into the system.	Pass		
TC002	Manage category	Add a new category	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> <li>User clicks on the "Add" button on the top right of the Manage category screen.</li> <li>User enters the name for the new category.</li> <li>User clicks on "confirm" button.</li> </ol>	The new category will be added successfully with a success notification.	Pass	Added 'khai ming'	
TC003	Manage category	Update a category name	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> </ol>	The name of the categoryof the beupdatedsuccessfullywithasuccessnotification.	Pass	Edit to 'not khai ming'	

			2. User selects the newly created			
			category from the displayed list			
			and clicks on the "Edit" button.			
			3. User edits the name of the			
			category from the modal popup			
			4. After editing, user clicks on the			
			"Confirm" button.			
TC004	Manage	Add a new	1. User logins into the system	The warehouse is added	Pass	data (khai ming,
	warehouse	warehouse	using <b>ADMIN</b> account.	successfully with a		kota Kinabalu,
			2. User clicks on the "Manage	success notification.		100, (A,B,C))
			warehouse" tab from the top			
			navigation bar.			
			3. User clicks on the "Add"			
			button on the top right of the			
			Manage warehouse screen.			
			4. User enters the name, location,			
l			number of bins (enter 100) and			
			zones (enter "A,B,C") for the			
			new warehouse.			
			5. User clicks on "confirm"			
			button after all information has			
			been entered.			
TC005	Manage	Add a new user	1. User logins into the system	Two new users will be	pass	
	user		using ADMIN account.	added successfully with		
			2. User clicks on the "Manage	a success notification.		
			user" tab from the top navigation			
l			bar.			
			3. User clicks on the "Add"			
			button on the top right of the			
			Manage user screen.			
			4. User enters the name, email,			
			contact number, IC number, role			

			(select "STAFF"), employed in, address and username for the new			
			user.			
			5. For the warehouse input, user			
			selects the newly created			
			warehouse.			
			6. User clicks on "confirm"			
			button after all information has			
			been entered.			
			7. Repeat step 3 to 6 to create			
			another user with STAFF role.			
TC006	Manage	Update a user's	e ;	The details of the	Pass	
	user	detail	using <b>ADMIN</b> account.	selected user will be		
			2. User clicks on the "Manage	updated successfully		
			user" tab from the top navigation	with a success		
			bar.	notification.		
			3. User selects a newly created			
			<b>user</b> from the displayed list and			
			clicks on the "Edit" button.			
			4. User edits some information of			
			the user (except the warehouse			
			and role) on the modal pop-up			
			5. After editing, user clicks on the "Confirm" button.			
TC007	Manage	Update a	1. User logins into the system	The warehouse details	Pass	
	warehouse	warehouse	using ADMIN account.	are updated		
			2. User clicks on the "Manage	successfully with a		
			warehouse" tab from the top	success notification.		
			navigation bar.			
			3. User selects the <b>newly created</b>			
			warehouse from the displayed			

				list and clicks on the "Edit"			
				button.			
				4. User edits the information by			
				adding a newly created user in			
				the warehouse manager field on			
				the modal pop-up			
				5. After editing, user clicks on the			
				"Confirm" button.			
TC008	Manage	Add	new	1. User logins into the system	Inventories are added	Pass	
	inventory	inventory		using <b>ADMIN</b> account.	successfully and a		
				2. User clicks on the "Manage	notification of the bin		
				inventory" tab from the top	number assigned to		
				navigation bar.	newly added inventory		
				3. User clicks on the "Add"	is shown.		
				button on the top right of the			
				Manage inventory screen.			
				4. For warehouse and category			
				fields, selects the warehouse			
				and category created earlier.			
				4. User enters the name, quantity			
				on hand, cost per unit and priority			
				for the inventory.			
				5. User clicks on "confirm"			
				button after all information has			
				been entered.			
				6. Repeat step 3 to 5 to create one			
				more inventory.			
TC009	Manage	Update	an	1. User logins into the system	The selected inventory	Pass	
	inventory	inventory		using <b>ADMIN</b> account.	details are updated		
				2. User clicks on the "Manage	according to user's		
				inventory" tab from the top	input.		
				navigation bar.			

			3. User selects a newly created			
			<b>inventory</b> from the displayed list			
			and clicks on the "Edit" button.			
			4. User edits the information of			
			the inventory on the modal pop-			
			up			
			5. After editing, user clicks on the			
			"Confirm" button.			
TC010	View all	View inventory	1. User logins into the system	User will successfully	Pass	
	inventories	and its details	using <b>ADMIN</b> account.	view the inventory list		
			2. User clicks on the "Manage	and will be redirected to		
			inventory" tab from the top	the inventory details		
			navigation bar.	page once an inventory		
			3. User view the list of	id is clicked.		
			inventories displayed.			
			4. User clicks on one of the			
			inventory ID to view the details.			
TC011	Manage	Edit inventory	1. User logins into the system	The selected inventory	Pass	
	warehouse	to storage bin	using <b>ADMIN</b> account.	will be added to the		
		_	2. User clicks on the "Manage	selected storage bin of		
			warehouse" tab from the top	the warehouse.		
			navigation bar.			
			3. User clicks on <b>newly created</b>			
			warehouse's ID to view the			
			details.			
			4. In the warehouse details page,			
			user selects the "storage bin" tab			
			to view the list of storage bins in			
			the warehouse			
			5. User selects the bin with the			
			newly created inventory (can use			
			the search to search for the			

		-			-	
			inventory) and clicks the "Edit			
			inventory" button.			
			6. User clears the current			
			inventory assigned on the modal			
			popup and click on the "Edit			
			inventory" button.			
TC012	Manage	Assign	1. User logins into the system	The category of the	Pass	
	warehouse	category to a	0	selected bins is changed		
		storage bin	2. User clicks on the "Manage	to the selected category.		
			warehouse" tab from the top			
			navigation bar.			
			3. User clicks on <b>newly created</b>			
			warehouse's ID to view the			
			details.			
			4. In the warehouse details page,			
			user selects the "storage bin" tab			
			to view the list of storage bins in			
			the warehouse			
			5. User clicks on the "Multi			
			assign bin" to assign category to			
			bins.			
			6. User selects a category and			
			checks desired bin to assign.			
			7. User clicks on the "Assign"			
			button.			
TC013	Login	Reset password	1. User logins into the system	A success notification	Fail	Edited the phone
	account		using ADMIN account.	will be shown.		number, the new
			2. User clicks on the "Manage			phone number
			User" tab and finds the newly			cannot be used as
			created user with "MANAGER"			the password,
			role.			have to use the
						phone number

TC014       Manage cycle counting       Start a new cycle counting       1. User logins into the system using the username and contact number as the password.       A success notification will be shown stating that the system using the created manager-role will be shown stating that the cycle counting       Pass         TC014       Manage cycle counting       Start a new cycle counting       I. User logins into the system using the created manager-role will be shown stating that the cycle counting is started. Schedules       Pass         J. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" from the "Manage Cycle counting is the created staffs       Suser clicks on the "Start Cycle counting" tab dropdown on the top navigation bar.       Suser clicks all inventories in the "Start Cycle counting" tab dropdown on the "Select Inventories in the "Select Inventories" field.       4. User clicks the "Create" button in the summary modal popup.         TC015       Manage cycle       View all usumary modal popup.       1. User clicks on the "View Cycle counting" from the "Manage counting XUs       The list of schedules for the cycle counting and counting and counting and counting and counting and counting and counting from the "Manage cycle counting" from the "Manage counting and counting from the "Manage counting from the "Manage counting and counting and counting from the "Manage counting and counting from the "Manage counting from the "Mana				3. User takes note on the			that is created
TC014       Manage cycle counting       Start a new cycle counting       1. User logouts and logins into the system using the username and contact number as the password.       A success notification will be shown stating that the cycle counting if you completed TC013)       Pass         TC014       Manage cycle counting       Start a new cycle counting using the created manager-role using the created manager-role using the created manager-role using the created TC013)       A success notification will be shown stating that the cycle counting is started. Schedules will be created for all SKUs and assigned to the selected staffs evenly.         3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" from the "Manage Cycle counting" the oravigation bar.       3. User fills in all of the information in the form displayed.         4. User selects all inventories in the "Select Inventories field.       4. User clicks the "Create" button after completing the form. 5. User clicks the "Create" button after completing the form. 5. User clicks the "Create" button after completing the form.       Pass         TC015       Manage cycle       View all upcoming cycle       1. User clicks on the "Wew Cycle to the cycle counting" from the "Nanage       The list of schedules for the cycle counting the summary modal popup.							
TC014       Manage cycle counting       Start a new cycle counting       1. User logins into the system using the username and contact number as the password.       A success notification will be shown stating that the cycle counting is started. Schedules will be shown stating that the cycle counting is started. Schedules will be shown stating that the cycle counting is started. Schedules will be shown stating that the cycle counting is started. Schedules will be created for all SKUs and assigned to Inventors listed in the View Inventors listed in the View Cycle counting if from the "Manage Cycle counting" from the "Manage Cycle counting it ab dropdown on the top navigation bar.       Skus and assigned to the selected staffs evenly.         TC015       Manage cycle       View all user clicks on the "Stert Cycle ounting" from the "Manage cycle counting the form. 5. User clicks the "Create" button in the summary modal popup.       TC015       Manage View all user clicks on the "Start Cycle counting" from the "Manage cycle curve counting in the summary modal popup.       The list of schedules for pass       Pass							
TC014Manage cycle countingStart a new cycle countingStart a new cycle countingStart a new i. User logins into the system using the created manager-role user credentials (skip this step if you completed TC013) 2. Note that there are TWO inventories listed in the View Inventory page, if no, kindly go back to TC008 to create an inventory. 3. User fills in all of the information bar. 3. User fills in all of the information bar. 3. User selects all inventories" field. 4. User selects all inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the submary modal popup.PassTC015Manage cycleView upcoming cycleView all upcoming cycleI. User clicks on the "View Cycle counting" from the "Manage cycle counting from the "Manage cycle counting from the "ManagePass				0			new user
TC014Manage cycle countingStart a new cycle counting1. User logins into the system using the created manager-role user credentials (skip this step if you completed TC013) 2. Note that there are TWO inventory page, if no, kindly go back to TC008 to create an inventory. 3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" from the form displayed.A success notification will be shown stating that the cycle counting is started. Schedules Will be created for all SKUs and assigned to the selected staffs evenly.TC015Manage cycleView upcoming cycle1. User clicks on the "View Cycle counting" from the "Manage cycle counting the top navigation bar. 3. User fills in all of the information in the form displayed.PassTC015Manage cycleView upcoming cycle1. User clicks on the "View Cycle counting" from the "Manage cycle counting the top navigation bar. 3. User fills in all of the information in the form displayed.PassTC015Manage cycleView upcoming cycle1. User clicks on the "View Cycle counting" from the "Manage tuton after completing the form. 5. User clicks the "Submit" button after completing the form. 5. User clicks on the "Wiew Cycle counting" from the "Manage tyce counting model popup.Pass							
TC014Manage cycle countingStart a new cycle countingI. User logins into the system using the created manager-role user credentials (skip this step if you completed TC013) 2. Note that there are TWO inventories listed in the View Inventory page, if no, kindly go back to TC008 to create an inventory.A success notification will be shown stating that the cycle counting is started. Schedules will be created for all SKUs and assigned to the selected staffs evenly.TC015Manage cycleView all upcoming cycleView all 1. User clicks on the "View Cycle counting" from the "Manage cycle counting the top navigation bar. 3. User selects all inventories in the "Select Inventories" field. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks on the "Start Cycle counting" from the "Submit" button after completing the form. 5. User clicks on the "Submit"The list of schedules for PassTC015Manage cycleView all1. User clicks on the "Wanage counting" from the "ManageThe list of schedules for the cycle counting and the cycle counting and							
TC014Manage cycle countingStart a new cycle counting1. User logins into the system using the created manager-role user credentials (skip this step if you completed TC013) 2. Note that there are TWO inventories listed in the View Inventory page, if no, kindly go back to TC008 to create an inventory. 3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting in the "Select Inventories" field. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks on the "Create" button in the summary modal popup.A success notification will be shown stating that the cycle counting is started. Schedules will be created for all SKUs and assigned to the selected staffs evenly.TC015Manage cycleView all upcoming cycleI. User clicks on the "Yiew Cycle counting" from the "Manage the "Select Inventories" field. 4. User clicks on the "Submit" button after completing the form. 5. User clicks on the "Yiew Cycle counting" from the "ManageThe list of schedules for the cycle counting and Pass				1			
cycle countingcycle countingusing the created manager-role user credentials (skip this step if you completed TC013) 2. Note that there are TWO inventories listed in the View Inventory page, if no, kindly go back to TC008 to create an inventory. 3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" tab dropdown on the top navigation bar. 3. User fills in all of the information in the form displayed. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks on the "View Cycle counting" from the "Managewill be shown stating that the cycle counting is started. Schedules will be created for all SKUs and assigned to the selected staffs evenly.TC015Manage cycleView all upcoming cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting and	TC014	24	<u> </u>		A	D	
countinguser credentials (skip this step if you completed TC013) 2. Note that there are TWO inventories listed in the View Dack to TC008 to create an inventory. 3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" from the form displayed. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Create" button in the summary modal popup.that the cycle counting and the cycle counting and the cycle counting" from the "Manage Cycle counting" from the form displayed.that the cycle counting is started. Schedules will be created for all SKUs and assigned to the selected staffs evenly.TC015Manage cycleView all upcoming cycle1. User clicks on the "Yiew Cycle counting" from the "ManageThe list of schedules for the cycle counting the cycle counting in the summary modal popup.	1C014	0		•		Pass	
if you completed TC013)is started. Schedules2. Note that there are TWO inventories listed in the View Inventory page, if no, kindly go back to TC008 to create an inventory.is started. Schedules will be created for all SKUs and assigned to the selected staffs evenly.3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" tab dropdown on the top navigation bar.S. User fills in all of the information in the form displayed.4. User selects all inventories in the "Select Inventories" field.4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.The list of schedules for the cycle counting and passTC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "Manage the select counting andPass		•	cycle counting	0	C C		
Image2. Note that there are TWO inventories listed in the View Inventory page, if no, kindly go back to TC008 to create an inventory. 3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" tab dropdown on the top navigation bar. 3. User fills in all of the information in the form displayed. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.The list of schedules for the cycle counting and passTC015Manage cycleView upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting and		counting					
ImageInventorieslisted in the View Inventory page, if no, kindly go back to <b>TC008</b> to create an inventory.SKUs and assigned to the selected staffs evenly.3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" tab dropdown on the top navigation bar. 3. User fills in all of the information in the form displayed. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Create" button in the summary modal popup.SKUs and assigned to the selected staffs evenly.TC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "Manage to curate" button and popup.The list of schedules for the cycle counting and							
Inventory page, if no, kindly go back to <b>TC008</b> to create an inventory.the selected staffs evenly.3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" tab dropdown on the top navigation bar.evenly.3. User fills in all of the information in the form displayed User selects all inventories in the "Select Inventories" field.4. User selects all inventories in the "Select Inventories" field User clicks the "Submit" button after completing the form. 5. User clicks on the "View Cycle counting" from the "ManageTC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting and							
back to TC008 to create an inventory.evenly.3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" tab dropdown on the top navigation bar.evenly.3. User fills in all of the information in the form displayed4. User selects all inventories in the "Select Inventories" field4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popupTC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting and					-		
inventory. 3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" tab dropdown on the top navigation bar. 3. User fills in all of the information in the form displayed. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.The list of schedules for the cycle counting andTC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting and							
3. User clicks on the "Start Cycle counting" from the "Manage Cycle counting" tab dropdown on the top navigation bar. 3. User fills in all of the information in the form displayed. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.TC015Manage versionView all upcoming cycle1. User clicks on the "View Cycle counting" from the "Manage the cycle counting andPass					evenly.		
TC015Manage cycleView upcoming cycle1. User clicks on the "View Cycle counting" from the "Nanage cycle counting" from the "Manage cycle counting" from the "Manage				5			
Cycle counting" tab dropdown on the top navigation bar. 3. User fills in all of the information in the form displayed. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.TC015Manage cycleView upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting andPass				•			
TC015Manage cycleView upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting andPass				counting" from the "Manage			
3. User fills in all of the information in the form displayed.4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.TC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting and				Cycle counting" tab dropdown on			
TC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting andPass				the top navigation bar.			
Imagedisplayed. 4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.ImageImageTC015Manage cycleView upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting andPass				3. User fills in all of the			
4. User selects all inventories in the "Select Inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.TC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting and				information in the form			
the "Select Inventories" field. 4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.Image and a submit and a				displayed.			
4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.4. User clicks the "Submit" button after completing the form. 5. User clicks the "Create" button in the summary modal popup.TC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "Manage the cycle counting andPass				4. User selects all inventories in			
button after completing the form. 5. User clicks the "Create" button in the summary modal popup.button after completing the form. 5. User clicks the "Create" button in the summary modal popup.TC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "Manage the cycle counting andPass				the "Select Inventories" field.			
button after completing the form. 5. User clicks the "Create" button in the summary modal popup.button after completing the form. 5. User clicks the "Create" button in the summary modal popup.TC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "Manage the cycle counting andPass							
TC015       Manage cycle       View all upcoming cycle       1. User clicks on the "View Cycle the cycle counting" from the "Manage the cycle counting and       Pass							
in the summary modal popup.TC015Manage cycleView all upcoming cycle1. User clicks on the "View Cycle counting" from the "ManageThe list of schedules for the cycle counting andPass				1 0			
TC015Manage cycleView upcoming cycle1. User clicks on the "View CycleThe list of schedules for the cycle counting andPass							
cycle upcoming cycle counting" from the "Manage the cycle counting and	TC015	Manage	View all		The list of schedules for	Pass	
		0		5			
		•	1 0 0				

TC016	View all staff	View all staff in a warehouse	<ul> <li>Cycle counting" tab dropdown on the top navigation bar.</li> <li>2. User clicks on the "View cycle count settings" button on the top right of the displayed table.</li> <li>1. User clicks on the "View Staff" tab on the top navigation bar.</li> <li>2. User clicks on one of the staff ID from the list to view the staff details.</li> </ul>	the settings will be displayed accordingly. The list of staffs and their details will be displayed accordingly.	Pass	
TC017	Perform cycle counting	Perform cycle counting for an inventory		The cycle counting record is created with success notification.	Pass	
TC018	Record	Check in stock	1. User clicks on the "Check	The selected	Pass	
	inventory	for an inventory	In/Out Stock" tab from the top navigation bar.	inventory's quantity on hand will be updated in		

			<ol> <li>User views the list of inventories displayed.</li> <li>User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>User selects the "Check In" tab in the modal popup.</li> <li>User enters the quantity and remarks and clicks the "Confirm" button.</li> </ol>	real time without refreshing the page.		
TC019	Record inventory	Check out stock for an inventory	<ol> <li>User clicks on the "Check In/Out Stock" tab from the top navigation bar.</li> <li>User views the list of inventories displayed.</li> <li>User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>User selects the "Check Out" tab in the modal popup.</li> <li>User enters the quantity and remarks and clicks the "Confirm" button.</li> </ol>	The selected inventory's quantity on hand will be updated in real time without refreshing the page.	Pass	
TC020	Manage cycle count approval report	Approve cycle count approval report	<ol> <li>User logins into the system using the created manager-role user credentials.</li> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Approval Report" tab and views the list of approval reports listed.</li> </ol>	The summary report for the selected cycle counting will be generated with IRA calculated.	Pass	

TC021	Manage cycle count approval report	Reject cycle count approval report	<ul> <li>4. User clicks the "Approve" button for one of the approval reports.</li> <li>5. User clicks the "Confirm" button in the confirmation modal.</li> <li>6. User refresh the page.</li> <li>1. User clicks on the "View reports" tab from the top navigation bar.</li> <li>2. User selects the "Cycle Count Approval Report" tab and views the list of approval reports listed.</li> <li>3. User clicks the "Reject" button for one of the approval reports.</li> <li>4. User checks the checkbox for reassigning staff to recount the SKU.</li> <li>5. User clicks the "Confirm" button in the confirmation modal.</li> </ul>	A success message will be shown to indicate the success rescheduling.	Pass	
TC022	View reports	View cycle counting summary report	<ol> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Summary Report" tab and views the list of summary reports listed.</li> <li>User clicks on the cycle count ID for one of the summary reports</li> </ol>	The list of summary reports and their details will be displayed accordingly.	Pass	
TC023	View reports	View daily reports	1. User clicks on the "View reports" tab from the top navigation bar.	The list of daily reports and their details and stocks will be displayed accordingly.	Pass	

					1	
			2. User selects the "Daily report"			
			tab and views the list of daily			
			reports listed.			
			3. User clicks on the "View"			
			button for one of the daily reports			
TC024	Manage	Delete an	1. User logins into the system	The selected inventory	Pass	
	inventory	inventory	using <b>ADMIN</b> account provided.	is successfully deleted		
	-		2. User clicks on the "Manage	from the warehouse.		
			inventory" tab from the top			
			navigation bar.			
			3. User finds one of the newly			
			created inventories from the			
			displayed list (can utilize the			
			filter or search function) and			
			clicks on the "Delete" button.			
			4. User clicks on the "Delete"			
			button on the confirmation			
			modal.			
TC025	Manage	Delete a user	1. User logins into the system	The status of the	Pass	
10025	Ũ	Delete a user	using <b>ADMIN</b> account.	selected user will be set	r ass	
	user		2. User clicks on the "Manage	to "INACTIVE".		
			e	IO INACTIVE .		
			user" tab from the top navigation			
			bar.			
			3. User selects one of the created			
			users from the displayed list and			
			clicks on the "Delete" button.			
			4. User clicks on the "Delete"			
			button on the confirmation			
			modal.			
TC026	Manage	Delete a	1. User logins into the system	The selected category	Pass	
	category	category	using ADMIN account.	will be deleted from the		
				list.		

				2. User clicks on the "Manage			
				category" tab from the top			
				navigation bar.			
				3. User selects the newly created			
				category from the displayed list			
				and clicks on the "Delete" button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
				modal.			
TC027	Manage	Delete	а	1. User logins into the system	The selected warehouse	Pass	
	warehouse	warehouse		using <b>ADMIN</b> account.	is successfully deleted		
				2. User clicks on the "Manage	with a success message.		
				warehouse" tab from the top	C		
				navigation bar.			
				3. User selects the newly created			
				warehouse from the displayed			
				list and clicks on the "Delete"			
				button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
				modal.			

			User Acceptance Testing	Form		
Tester's Name				Testing Start Date/ Time Testing End Date/ Time		
Test Case ID	Module	Test Case Title	Test Steps	Expected Results	Status (Pass/Fail)	Comments
TC001	Login account	Login an account	<ol> <li>User enters username and password of ADMIN account in the login page.</li> <li>User clicks on the login button.</li> </ol>	User will successfully logged-in into the system.	Pass	'Opps' error message
TC002	Manage category	Add a new category	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> <li>User clicks on the "Add" button on the top right of the Manage category screen.</li> <li>User enters the name for the new category.</li> <li>User clicks on "confirm" button.</li> </ol>	The new category will be added successfully with a success notification.	Pass	
TC003	Manage category	Update a category name	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> <li>User selects the newly created category from the displayed list and clicks on the "Edit" button.</li> <li>User edits the name of the category from the modal popup</li> </ol>	The name of the category will be updated successfully with a success notification.	Pass	Cancel button and close button not working

			4. After editing, user clicks on the			
			"Confirm" button.			
TC004	Manage	Add a new	1. User logins into the system	The warehouse is added	Pass	
	warehouse	warehouse	using <b>ADMIN</b> account.	successfully with a		
			2. User clicks on the "Manage	success notification.		
			warehouse" tab from the top			
			navigation bar.			
			3. User clicks on the "Add"			
			button on the top right of the			
			Manage warehouse screen.			
			4. User enters the name, location,			
			number of bins (enter 100) and			
			zones (enter "A,B,C") for the			
			new warehouse.			
			5. User clicks on "confirm"			
			button after all information has			
			been entered.			
TC005	Manage	Add a new user	1. User logins into the system	Two new users will be	Pass	
	user		using <b>ADMIN</b> account.	added successfully with		
			2. User clicks on the "Manage	a success notification.		
			user" tab from the top navigation			
			bar.			
			3. User clicks on the "Add"			
			button on the top right of the			
			Manage user screen.			
			4. User enters the name, email,			
			contact number, IC number, role			
			(select "STAFF"), employed in,			
			address and username for the new			
			user.			

	1	1	1		1	
			5. For the warehouse input, user			
			selects the newly created			
			warehouse.			
			6. User clicks on "confirm"			
			button after all information has			
			been entered.			
			7. Repeat step 3 to 6 to create			
			another user with STAFF role.			
TC006	Manage	Update a user's	1. User logins into the system	The details of the	Pass	
	user	detail	using <b>ADMIN</b> account.	selected user will be		
			2. User clicks on the "Manage	updated successfully		
			user" tab from the top navigation	with a success		
			bar.	notification.		
			3. User selects a newly created			
			user from the displayed list and			
			clicks on the "Edit" button.			
			4. User edits some information of			
			the user (except the warehouse			
			and role) on the modal pop-up			
			5. After editing, user clicks on the			
			"Confirm" button.			
TC007	Manage	Update a	1. User logins into the system	The warehouse details	Pass	
	warehouse	warehouse	using <b>ADMIN</b> account.	are updated		
			2. User clicks on the "Manage	successfully with a		
			warehouse" tab from the top	success notification.		
			navigation bar.			
			3. User selects the <b>newly created</b>			
			warehouse from the displayed			
			list and clicks on the "Edit"			
			button.			
			4. User edits the information by			
			adding a newly created user in			

				<ul><li>the warehouse manager field on the modal pop-up</li><li>5. After editing, user clicks on the "Confirm" button.</li></ul>			
TC008	Manage inventory	Add inventory	new	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage inventory" tab from the top navigation bar.</li> <li>User clicks on the "Add" button on the top right of the Manage inventory screen.</li> <li>For warehouse and category fields, selects the warehouse and category created earlier.</li> <li>User enters the name, quantity on hand, cost per unit and priority for the inventory.</li> <li>User clicks on "confirm" button after all information has been entered.</li> <li>Repeat step 3 to 5 to create one more inventory.</li> </ol>	Inventories are added successfully and a notification of the bin number assigned to newly added inventory is shown.		
TC009	Manage inventory	Update inventory	an	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage inventory" tab from the top navigation bar.</li> <li>User selects a newly created inventory from the displayed list and clicks on the "Edit" button.</li> </ol>	The selected inventory details are updated according to user's input.	Pass	

			4. User edits the information of			
			the inventory on the modal pop-			
			up			
			5. After editing, user clicks on the			
			"Confirm" button.			
TC010	View all	View inventory	1. User logins into the system	User will successfully	Pass	
	inventories	and its details	using <b>ADMIN</b> account.	view the inventory list		
			2. User clicks on the "Manage	and will be redirected to		
			inventory" tab from the top	the inventory details		
			navigation bar.	page once an inventory		
			3. User view the list of	id is clicked.		
			inventories displayed.			
			4. User clicks on one of the			
			inventory ID to view the details.			
TC011	Manage	Edit inventory	1. User logins into the system	The selected inventory	Pass	
	warehouse	to storage bin	using <b>ADMIN</b> account.	will be added to the		
			2. User clicks on the "Manage	selected storage bin of		
			warehouse" tab from the top	the warehouse.		
			navigation bar.			
			3. User clicks on <b>newly created</b>			
			warehouse's ID to view the			
			details.			
			4. In the warehouse details page,			
			user selects the "storage bin" tab			
			to view the list of storage bins in			
			the warehouse			
			5. User selects the bin with the			
			newly created inventory (can use			
			the search to search for the			
			inventory) and clicks the "Edit			
			inventory" button.			

[						
			6. User clears the current			
			inventory assigned on the modal			
			popup and click on the "Edit			
			inventory" button.			
TC012	Manage	Assign	1. User logins into the system	The category of the	Pass	
	warehouse	category to a	using <b>ADMIN</b> account.	selected bins is changed		
		storage bin	2. User clicks on the "Manage	to the selected category.		
			warehouse" tab from the top			
			navigation bar.			
			3. User clicks on <b>newly created</b>			
			warehouse's ID to view the			
			details.			
			4. In the warehouse details page,			
			user selects the "storage bin" tab			
			to view the list of storage bins in			
			the warehouse			
			5. User clicks on the "Multi			
			assign bin" to assign category to			
			bins.			
			6. User selects a category and			
			checks desired bin to assign.			
			7. User clicks on the "Assign"			
			button.			
TC013	Login	Reset password	1. User logins into the system	A success notification	Pass	
	account	_	using <b>ADMIN</b> account.	will be shown.		
			2. User clicks on the "Manage			
			User" tab and finds the newly			
			created user with "MANAGER"			
			role.			
			3. User takes note on the			
			username and contact number.			

	1					
			4. User logouts and logins into			
			the system using the username			
			and contact number as the			
			password.			
			5. User resets the password			
TC014	Manage	Start a new	1. User logins into the system	A success notification	Pass	
	cycle	cycle counting	using the created manager-role	will be shown stating		
	counting		user credentials (skip this step	that the cycle counting		
	_		if you completed TC013)	is started. Schedules		
			2. Note that there are <b>TWO</b>	will be created for all		
			inventories listed in the View	SKUs and assigned to		
			Inventory page, if no, kindly go	the selected staffs		
			back to TC008 to create an	evenly.		
			inventory.	-		
			3. User clicks on the "Start Cycle			
			counting" from the "Manage			
			Cycle counting" tab dropdown on			
			the top navigation bar.			
			3. User fills in all of the			
			information in the form			
			displayed.			
			4. User selects all inventories in			
			the "Select Inventories" field.			
			4. User clicks the "Submit"			
			button after completing the form.			
			5. User clicks the "Create" button			
			in the summary modal popup.			
TC015	Manage	View all	1. User clicks on the "View Cycle	The list of schedules for	Pass	
	cycle	upcoming cycle	counting" from the "Manage	the cycle counting and		
	counting	counting SKUs	Cycle counting" tab dropdown on	the settings will be		
	_	_	the top navigation bar.	displayed accordingly.		

			2 Ilean aliabe on the Wiener and			
			2. User clicks on the "View cycle			
			count settings" button on the top			
			right of the displayed table.			
TC016	View all	View all staff in	1. User clicks on the "View Staff"	The list of staffs and	Pass	
	staff	a warehouse	tab on the top navigation bar.	their details will be		
			2. User clicks on one of the staff	displayed accordingly.		
			ID from the list to view the staff			
			details.			
TC017	Perform	Perform cycle	1. Repeat TC013 to login and	The cycle counting	Pass	
	cycle	counting for an	reset password for the created	record is created with		
	counting	inventory	STAFF-role user.	success notification.		
	_	-	2. User clicks on the "Cycle			
			Counting" tab from the top			
			navigation bar.			
			3. User selects the "Upcoming"			
			tab and views the list of			
			upcoming cycle counting			
			schedules			
			3. User chooses the first one in			
			the list and clicks the "Count"			
			button.			
			4. User enters the actual count on			
			the modal popup			
			5. User clicks the "Create" button			
			6. User refresh the page.			
			7. Repeat step 3 to 6 to count			
			another one.			
TC018	Record	Check in stock	1. User clicks on the "Check	The selected	Pass	
	inventory	for an	In/Out Stock" tab from the top	inventory's quantity on		
	5	inventory	navigation bar.	hand will be updated in		
		· - ·	2. User views the list of	real time without		
			inventories displayed.	refreshing the page.		

			<ol> <li>User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>User selects the "Check In" tab in the modal popup.</li> <li>User enters the quantity and remarks and clicks the "Confirm" button.</li> </ol>			
TC019	Record inventory	Check out stock for an inventory	<ol> <li>User clicks on the "Check In/Out Stock" tab from the top navigation bar.</li> <li>User views the list of inventories displayed.</li> <li>User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>User selects the "Check Out" tab in the modal popup.</li> <li>User enters the quantity and remarks and clicks the "Confirm" button.</li> </ol>	The selected inventory's quantity on hand will be updated in real time without refreshing the page.	Pass	
TC020	Manage cycle count approval report	Approve cycle count approval report	<ol> <li>User logins into the system using the created manager-role user credentials.</li> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Approval Report" tab and views the list of approval reports listed.</li> </ol>	The summary report for the selected cycle counting will be generated with IRA calculated.	Pass	No notification but got update

			<ul> <li>4. User clicks the "Approve" button for one of the approval reports.</li> <li>5. User clicks the "Confirm" button in the confirmation modal.</li> <li>6. User refresh the page.</li> </ul>			
TC021	Manage cycle count approval report	Reject cycle count approval report	1. User clicks on the "View	A success message will be shown to indicate the success rescheduling.	Pass	No notification but got update
TC022	View reports	View cycle counting summary report	reports" tab from the top	The list of summary reports and their details will be displayed accordingly.	Pass	
TC023	View reports	View daily reports	1. User clicks on the "View reports" tab from the top navigation bar.	The list of daily reports and their details and stocks will be displayed accordingly.	Pass	

			2. User selects the "Daily report"			
			tab and views the list of daily			
			reports listed.			
			3. User clicks on the "View"			
			button for one of the daily reports			
TC024	Manage	Delete an	1. User logins into the system	The selected inventory	Pass	
	inventory	inventory	using <b>ADMIN</b> account provided.	is successfully deleted		
			2. User clicks on the "Manage	from the warehouse.		
			inventory" tab from the top			
			navigation bar.			
			3. User finds one of the newly			
			created inventories from the			
			displayed list (can utilize the			
			filter or search function) and			
			clicks on the "Delete" button.			
			4. User clicks on the "Delete"			
			button on the confirmation			
			modal.			
TC025	Manage	Delete a user	1. User logins into the system	The status of the	Pass	
	user		using <b>ADMIN</b> account.	selected user will be set		
			2. User clicks on the "Manage	to "INACTIVE".		
			user" tab from the top navigation			
			bar.			
			3. User selects one of the created			
			<b>users</b> from the displayed list and			
			clicks on the "Delete" button.			
			4. User clicks on the "Delete"			
			button on the confirmation			
			modal.			
TC026	Manage	Delete a		The selected category	Pass	
10020	category	category	using <b>ADMIN</b> account.	will be deleted from the	- 400	
				list.		
				1150.		

				2. User clicks on the "Manage			
				category" tab from the top			
				navigation bar.			
				3. User selects the newly created			
				<b>category</b> from the displayed list			
				and clicks on the "Delete" button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
				modal.			
TC027	Manage	Delete	a	1. User logins into the system	The selected warehouse	Pass	
	warehouse	warehouse		using <b>ADMIN</b> account.	is successfully deleted		
				2. User clicks on the "Manage	with a success message.		
				warehouse" tab from the top			
				navigation bar.			
				3. User selects the newly created			
				warehouse from the displayed			
				list and clicks on the "Delete"			
				button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
				modal.			

## Novice testers

			User Acceptance Testing	Form		
Tester's Name	Low Ee Ly	ne		Testing Start Date/ Time		
				Testing End Date/ Time		
Test Case ID	Module	Test Case Title	Test Steps	Expected Results	Status (Pass/Fail)	Comments
TC001	Login account	Login an account	<ol> <li>User enters username and password of ADMIN account in the login page.</li> <li>User clicks on the login button.</li> </ol>	User will successfully logged-in into the system.	Pass	
TC002	Manage category	Add a new category	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> <li>User clicks on the "Add" button on the top right of the Manage category screen.</li> <li>User enters the name for the new category.</li> <li>User clicks on "confirm" button.</li> </ol>	The new category will be added successfully with a success notification.	Pass	
TC003	Manage category	Update a category name	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> </ol>	The name of the categoryof the beupdatedsuccessfullywithasuccessnotification.	Pass	

Image: Constraint of the system warehouse2. User selects the newly created category from the displayed list and clicks on the "Edit" button. 3. User edits the name of the category from the modal popup 4. After editing, user clicks on the "Confirm" button.TC004Manage warehouseAdd a new using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,The warehouse is added success notification.
TC004Manage warehouseAdd a new warehouse1. User edits the name of the category from the modal popup 4. After editing, user clicks on the "Confirm" button.The warehouse is added successfully with a success notification.PassTC004Manage warehouseAdd a new warehouse1. User logins into the system using <b>ADMIN</b> account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,The warehouse is added success notification.Pass
3. User edits the name of the category from the modal popup 4. After editing, user clicks on the "Confirm" button.The warehouse is added successfully with a success notification.TC004Manage warehouseAdd a new warehouse1. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,The warehouse is added success notification.Pass
TC004Manage warehouseAdd a new warehouse1. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,The warehouse is added successfully with a success notification.
TC004Manage warehouseAdd a new warehouse1. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,The warehouse is added successfully with a success notification.Pass
TC004Manage warehouseAdd a new warehouse1. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,The warehouse is added successfully with a success notification.Pass
TC004Manage warehouseAdd a new warehouse1. User logins into the system using ADMIN account. 2. User clicks on the "Manage warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,The warehouse is added successfully with a success notification.
warehousewarehouseusing ADMIN account.successfully with a2. User clicks on the "Manage warehouse" tab from the top navigation bar.success notification.3. User clicks on the "Add" button on the top right of the Manage warehouse screen.success notification.4. User enters the name, location,user clicks
2. User clicks on the "Manage warehouse" tab from the top navigation bar.success notification.3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,success notification.
warehouse" tab from the top navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,
warehouse" tab from the top navigation bar.         3. User clicks on the "Add" button on the top right of the Manage warehouse screen.         4. User enters the name, location,
navigation bar. 3. User clicks on the "Add" button on the top right of the Manage warehouse screen. 4. User enters the name, location,
<ul> <li>3. User clicks on the "Add" button on the top right of the Manage warehouse screen.</li> <li>4. User enters the name, location,</li> </ul>
button on the top right of the Manage warehouse screen. 4. User enters the name, location,
Manage warehouse screen. 4. User enters the name, location,
4. User enters the name, location,
number of bins (enter 100) and
zones (enter "A,B,C") for the
new warehouse.
5. User clicks on "confirm"
button after all information has
been entered.
TC005 Manage Add a new user 1. User logins into the system Two new users will be pass
user using ADMIN account. added successfully with
2. User clicks on the "Manage a success notification.
user" tab from the top navigation
bar.
3. User clicks on the "Add"
button on the top right of the
Manage user screen.
4. User enters the name, email,
contact number, IC number, role

			<ul><li>(select "STAFF"), employed in, address and username for the new user.</li><li>5. For the warehouse input, user selects the newly created</li></ul>			
			<ul> <li>warehouse.</li> <li>User clicks on "confirm"</li> <li>button after all information has been entered.</li> <li>Repeat step 3 to 6 to create another user with STAFF role.</li> </ul>			
TC006	Manage user	Update a user's detail		The details of the selected user will be updated successfully with a success notification.	Pass	
TC007	Manage warehouse	Update a warehouse	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage warehouse" tab from the top navigation bar.</li> <li>User selects the newly created warehouse from the displayed</li> </ol>	The warehouse details are updated successfully with a success notification.	Pass	

				list and clicks on the "Edit"			
				button.			
				4. User edits the information by			
				adding a newly created user in			
				the warehouse manager field on			
				the modal pop-up			
				5. After editing, user clicks on the			
				"Confirm" button.			
TC008	Manage	Add	new	1. User logins into the system	Inventories are added	Pass	
	inventory	inventory		using <b>ADMIN</b> account.	successfully and a		
				2. User clicks on the "Manage	notification of the bin		
				inventory" tab from the top	number assigned to		
				navigation bar.	newly added inventory		
				3. User clicks on the "Add"	is shown.		
				button on the top right of the			
				Manage inventory screen.			
				4. For warehouse and category			
				fields, selects the warehouse			
				and category created earlier.			
				4. User enters the name, quantity			
				on hand, cost per unit and priority			
				for the inventory.			
				5. User clicks on "confirm"			
				button after all information has			
				been entered.			
				6. Repeat step 3 to 5 to create one			
magaaa				more inventory.			
TC009	Manage	Update	an	<b>e</b> .	The selected inventory	Pass	
	inventory	inventory		using <b>ADMIN</b> account.	details are updated		
				2. User clicks on the "Manage	according to user's		
				inventory" tab from the top	input.		
				navigation bar.			

	1				1	, , , , , , , , , , , , , , , , , , , ,
			3. User selects a newly created			
			<b>inventory</b> from the displayed list			
			and clicks on the "Edit" button.			
			4. User edits the information of			
			the inventory on the modal pop-			
			up			
			5. After editing, user clicks on the			
			"Confirm" button.			
TC010	View all	View inventory	1. User logins into the system	User will successfully	Pass	
	inventories	and its details	using <b>ADMIN</b> account.	view the inventory list		
			2. User clicks on the "Manage	and will be redirected to		
			inventory" tab from the top	the inventory details		
			navigation bar.	page once an inventory		
			3. User view the list of	id is clicked.		
			inventories displayed.			
			4. User clicks on one of the			
			inventory ID to view the details.			
TC011	Manage	Edit inventory	1. User logins into the system	The selected inventory	Pass	
	warehouse	to storage bin	using <b>ADMIN</b> account.	will be added to the		
		_	2. User clicks on the "Manage	selected storage bin of		
			warehouse" tab from the top	the warehouse.		
			navigation bar.			
			3. User clicks on <b>newly created</b>			
			warehouse's ID to view the			
			details.			
			4. In the warehouse details page,			
			user selects the "storage bin" tab			
			to view the list of storage bins in			
			the warehouse			
			5. User selects the bin with the			
			newly created inventory (can use			
			the search to search for the			

					1	
			inventory) and clicks the "Edit			
			inventory" button.			
			6. User clears the current			
			inventory assigned on the modal			
			popup and click on the "Edit			
			inventory" button.			
TC012	Manage	Assign	1. User logins into the system	The category of the	Pass	
	warehouse	category to a	Ç .	selected bins is changed		
		storage bin	2. User clicks on the "Manage	to the selected category.		
		C	warehouse" tab from the top			
			navigation bar.			
			3. User clicks on <b>newly created</b>			
			warehouse's ID to view the			
			details.			
			4. In the warehouse details page,			
			user selects the "storage bin" tab			
			to view the list of storage bins in			
			the warehouse			
			5. User clicks on the "Multi			
			assign bin" to assign category to			
			bins.			
			6. User selects a category and			
			checks desired bin to assign.			
			7. User clicks on the "Assign"			
			button.			
TC013	Login	Reset password	1. User logins into the system	A success notification	Pass	
10015	account	reset pussion	using <b>ADMIN</b> account.	will be shown.	1 400	
	account		2. User clicks on the "Manage			
			User" tab and finds the newly			
			created user with "MANAGER"			
			role.			
			1010.			

	1	1	1			1
			3. User takes note on the			
			username and contact number.			
			4. User logouts and logins into			
			the system using the username			
			and contact number as the			
			password.			
			5. User resets the password			
TC014	Manage	Start a new	1. User logins into the system	A success notification	Pass	
	cycle	cycle counting	using the created manager-role	will be shown stating		
	counting		user credentials (skip this step	that the cycle counting		
	8		if you completed TC013)	is started. Schedules		
			2. Note that there are <b>TWO</b>	will be created for all		
			inventories listed in the View	SKUs and assigned to		
			Inventory page, if no, kindly go	the selected staffs		
			back to <b>TC008</b> to create an	evenly.		
			inventory.			
			3. User clicks on the "Start Cycle			
			counting" from the "Manage			
			Cycle counting" tab dropdown on			
			the top navigation bar.			
			3. User fills in all of the			
			information in the form			
			displayed.			
			4. User selects all inventories in			
			the "Select Inventories" field.			
			4. User clicks the "Submit"			
			button after completing the form.			
			5. User clicks the "Create" button			
			in the summary modal popup.			
TC015	Manage	View all	1. User clicks on the "View Cycle	The list of schedules for	Pass	
	cycle	upcoming cycle	5	the cycle counting and	- 400	
	counting	counting SKUs	round non the manage	the eyere counting and		
	counting	counting 51105				l

TC016	View all staff	View all staff in a warehouse	<ul> <li>Cycle counting" tab dropdown on the top navigation bar.</li> <li>2. User clicks on the "View cycle count settings" button on the top right of the displayed table.</li> <li>1. User clicks on the "View Staff" tab on the top navigation bar.</li> <li>2. User clicks on one of the staff ID from the list to view the staff details.</li> </ul>	the settings will be displayed accordingly. The list of staffs and their details will be displayed accordingly.	Pass	
TC017	Perform cycle counting	Perform cycle counting for an inventory	1. Repeat TC013 to login and	The cycle counting record is created with success notification.	Pass	
TC018	Record	Check in stock	1. User clicks on the "Check	The selected	Pass	
	inventory	for an inventory	In/Out Stock" tab from the top navigation bar.	inventory's quantity on hand will be updated in		

			<ol> <li>User views the list of inventories displayed.</li> <li>User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>User selects the "Check In" tab in the modal popup.</li> <li>User enters the quantity and remarks and clicks the "Confirm" button.</li> </ol>	real time without refreshing the page.		
TC019	Record inventory	Check out stock for an inventory	<ol> <li>User clicks on the "Check In/Out Stock" tab from the top navigation bar.</li> <li>User views the list of inventories displayed.</li> <li>User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>User selects the "Check Out" tab in the modal popup.</li> <li>User enters the quantity and remarks and clicks the "Confirm" button.</li> </ol>	The selected inventory's quantity on hand will be updated in real time without refreshing the page.	Pass	
TC020	Manage cycle count approval report	Approve cycle count approval report	<ol> <li>User logins into the system using the created manager-role user credentials.</li> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Approval Report" tab and views the list of approval reports listed.</li> </ol>	The summary report for the selected cycle counting will be generated with IRA calculated.	Pass	

TC021	Manage cycle count approval report	Reject cycle count approval report	<ul> <li>4. User clicks the "Approve" button for one of the approval reports.</li> <li>5. User clicks the "Confirm" button in the confirmation modal.</li> <li>6. User refresh the page.</li> <li>1. User clicks on the "View reports" tab from the top navigation bar.</li> <li>2. User selects the "Cycle Count Approval Report" tab and views the list of approval reports listed.</li> <li>3. User clicks the "Reject" button for one of the approval reports.</li> <li>4. User checks the checkbox for reassigning staff to recount the SKU.</li> <li>5. User clicks the "Confirm" button in the confirmation modal.</li> </ul>	A success message will be shown to indicate the success rescheduling.	Pass	
TC022	View reports	View cycle counting summary report	<ol> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Summary Report" tab and views the list of summary reports listed.</li> <li>User clicks on the cycle count ID for one of the summary reports</li> </ol>	The list of summary reports and their details will be displayed accordingly.	Pass	
TC023	View reports	View daily reports	1. User clicks on the "View reports" tab from the top navigation bar.	The list of daily reports and their details and stocks will be displayed accordingly.	Pass	

			2. User selects the "Daily report"			
			tab and views the list of daily			
			reports listed.			
			3. User clicks on the "View"			
			button for one of the daily reports			
TC024	Manage	Delete an	1. User logins into the system	The selected inventory	Pass	
	inventory	inventory	using <b>ADMIN</b> account provided.	is successfully deleted		
	-		2. User clicks on the "Manage	from the warehouse.		
			inventory" tab from the top			
			navigation bar.			
			3. User finds one of the newly			
			created inventories from the			
			displayed list (can utilize the			
			filter or search function) and			
			clicks on the "Delete" button.			
			4. User clicks on the "Delete"			
			button on the confirmation			
			modal.			
TC025	Manage	Delete a user	1. User logins into the system	The status of the	Pass	
10025	user		using <b>ADMIN</b> account.	selected user will be set	1 455	
	user		2. User clicks on the "Manage	to "INACTIVE".		
			user" tab from the top navigation			
			bar.			
			3. User selects one of the created			
			<b>users</b> from the displayed list and			
			clicks on the "Delete" button.			
			4. User clicks on the "Delete"			
			button on the confirmation			
			modal.			
TC026	Manage	Delete a		The selected category	Pass	
10020	U		using <b>ADMIN</b> account.	will be deleted from the	1 455	
	category	category		list.		
				1151.		

				2. User clicks on the "Manage			
				category" tab from the top			
				navigation bar.			
				3. User selects the newly created			
				category from the displayed list			
				and clicks on the "Delete" button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
				modal.			
TC027	Manage	Delete	a	1. User logins into the system	The selected warehouse	Pass	
	warehouse	warehouse		using <b>ADMIN</b> account.	is successfully deleted		
				2. User clicks on the "Manage	with a success message.		
				warehouse" tab from the top	C		
				navigation bar.			
				3. User selects the newly created			
				warehouse from the displayed			
				list and clicks on the "Delete"			
				button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
				modal.			

			User Acceptance Testing	Form		
Tester's Name	Low Jia We	ei		Testing Start Date/ Time		
				Testing End Date/ Time		
Test Case ID	Module	Test Case Title	Test Steps	Expected Results	Status (Pass/Fail)	Comments
TC001	Login account	Login an account	<ol> <li>User enters username and password of ADMIN account in the login page.</li> <li>User clicks on the login button.</li> </ol>	User will successfully logged-in into the system.	Pass	
TC002	Manage category	Add a new category	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> <li>User clicks on the "Add" button on the top right of the Manage category screen.</li> <li>User enters the name for the new category.</li> <li>User clicks on "confirm" button.</li> </ol>	The new category will be added successfully with a success notification.	Pass	
TC003	Manage category	Update a category name		The name of the category will be updated successfully with a success notification.	Pass	

			4. After editing, user clicks on the			
			"Confirm" button.			
TC004	Manage	Add a new	1. User logins into the system	The warehouse is added	Pass	
	warehouse	warehouse	using <b>ADMIN</b> account.	successfully with a		
			2. User clicks on the "Manage	success notification.		
			warehouse" tab from the top			
			navigation bar.			
			3. User clicks on the "Add"			
			button on the top right of the			
			Manage warehouse screen.			
			4. User enters the name, location,			
			number of bins (enter 100) and			
			zones (enter "A,B,C") for the			
			new warehouse.			
			5. User clicks on "confirm"			
			button after all information has			
			been entered.			
TC005	Manage	Add a new user	1. User logins into the system	Two new users will be	pass	
	user		using <b>ADMIN</b> account.	added successfully with	-	
			2. User clicks on the "Manage	a success notification.		
			user" tab from the top navigation			
			bar.			
			3. User clicks on the "Add"			
			button on the top right of the			
			Manage user screen.			
			4. User enters the name, email,			
			contact number, IC number, role			
			(select "STAFF"), employed in,			
			address and username for the new			
			user.			

		1	I		1	
			5. For the warehouse input, user			
			selects the newly created			
			warehouse.			
			6. User clicks on "confirm"			
			button after all information has			
			been entered.			
			7. Repeat step 3 to 6 to create			
			another user with STAFF role.			
TC006	Manage	Update a user's	1. User logins into the system	The details of the	Pass	
	user	detail	using <b>ADMIN</b> account.	selected user will be		
			2. User clicks on the "Manage	updated successfully		
			user" tab from the top navigation	with a success		
			bar.	notification.		
			3. User selects a newly created			
			user from the displayed list and			
			clicks on the "Edit" button.			
			4. User edits some information of			
			the user (except the warehouse			
			and role) on the modal pop-up			
			5. After editing, user clicks on the			
			"Confirm" button.			
TC007	Manage	Update a	1. User logins into the system	The warehouse details	Pass	
	warehouse	warehouse	using <b>ADMIN</b> account.	are updated		
			2. User clicks on the "Manage	successfully with a		
			warehouse" tab from the top	success notification.		
			navigation bar.			
			3. User selects the newly created			
			warehouse from the displayed			
			list and clicks on the "Edit"			
			button.			
			4. User edits the information by			
			adding a newly created user in			

				<ul><li>the warehouse manager field on the modal pop-up</li><li>5. After editing, user clicks on the "Confirm" button.</li></ul>			
TC008	Manage inventory	Add inventory	new	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage inventory" tab from the top navigation bar.</li> <li>User clicks on the "Add" button on the top right of the Manage inventory screen.</li> <li>For warehouse and category fields, selects the warehouse and category created earlier.</li> <li>User enters the name, quantity on hand, cost per unit and priority for the inventory.</li> <li>User clicks on "confirm" button after all information has been entered.</li> <li>Repeat step 3 to 5 to create one more inventory.</li> </ol>	Inventories are added successfully and a notification of the bin number assigned to newly added inventory is shown.		
TC009	Manage inventory	Update inventory	an	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage inventory" tab from the top navigation bar.</li> <li>User selects a newly created inventory from the displayed list and clicks on the "Edit" button.</li> </ol>	The selected inventory details are updated according to user's input.	Pass	

_

	1	[	[			[
			6. User clears the current			
			inventory assigned on the modal			
			popup and click on the "Edit			
			inventory" button.			
TC012	Manage	Assign	1. User logins into the system	The category of the	Pass	
	warehouse	category to a	using <b>ADMIN</b> account.	selected bins is changed		
		storage bin	2. User clicks on the "Manage	to the selected category.		
			warehouse" tab from the top			
			navigation bar.			
			3. User clicks on <b>newly created</b>			
			warehouse's ID to view the			
			details.			
			4. In the warehouse details page,			
			user selects the "storage bin" tab			
			to view the list of storage bins in			
			the warehouse			
			5. User clicks on the "Multi			
			assign bin" to assign category to			
			bins.			
			6. User selects a category and			
			checks desired bin to assign.			
			7. User clicks on the "Assign"			
			button.			
TC013	Login	Reset password	1. User logins into the system	A success notification	Pass	
	account	-	using <b>ADMIN</b> account.	will be shown.		
			2. User clicks on the "Manage			
			User" tab and finds the newly			
			created user with "MANAGER"			
			role.			
			3. User takes note on the			
			username and contact number.			

	1				1	
			4. User logouts and logins into			
			the system using the username			
			and contact number as the			
			password.			
			5. User resets the password			
TC014	Manage	Start a new	1. User logins into the system	A success notification	Pass	
	cycle	cycle counting	using the created manager-role	will be shown stating		
	counting		user credentials (skip this step	that the cycle counting		
	-		if you completed TC013)	is started. Schedules		
			2. Note that there are <b>TWO</b>	will be created for all		
			inventories listed in the View	SKUs and assigned to		
			Inventory page, if no, kindly go	the selected staffs		
			back to TC008 to create an	evenly.		
			inventory.	-		
			3. User clicks on the "Start Cycle			
			counting" from the "Manage			
			Cycle counting" tab dropdown on			
			the top navigation bar.			
			3. User fills in all of the			
			information in the form			
			displayed.			
			4. User selects all inventories in			
			the "Select Inventories" field.			
			4. User clicks the "Submit"			
			button after completing the form.			
			5. User clicks the "Create" button			
			in the summary modal popup.			
TC015	Manage	View all	1. User clicks on the "View Cycle	The list of schedules for	Pass	
	cycle	upcoming cycle	counting" from the "Manage	the cycle counting and		
	counting	counting SKUs	Cycle counting" tab dropdown on	the settings will be		
	-	_	the top navigation bar.	displayed accordingly.		

			2. User clicks on the "View cycle			
			count settings" button on the top			
<b>TC</b> 016	x 7' 11	<b>X</b> <i>I</i> <b>11</b> (CC)	right of the displayed table.		<b>D</b>	
TC016	View all	View all staff in	1. User clicks on the "View Staff"	The list of staffs and	Pass	
	staff	a warehouse	tab on the top navigation bar.	their details will be		
			2. User clicks on one of the staff	displayed accordingly.		
			ID from the list to view the staff			
			details.			
TC017	Perform	Perform cycle	1. Repeat TC013 to login and	The cycle counting	Pass	
	cycle	counting for an	reset password for the created	record is created with		
	counting	inventory	STAFF-role user.	success notification.		
			2. User clicks on the "Cycle			
			Counting" tab from the top			
			navigation bar.			
			3. User selects the "Upcoming"			
			tab and views the list of			
			upcoming cycle counting			
			schedules			
			3. User chooses the first one in			
			the list and clicks the "Count"			
			button.			
			4. User enters the actual count on			
			the modal popup			
			5. User clicks the "Create" button			
			6. User refresh the page.			
			7. Repeat step 3 to 6 to count			
			another one.			
TC018	Record	Check in stock	1. User clicks on the "Check	The selected	Pass	
	inventory	for an	In/Out Stock" tab from the top	inventory's quantity on		
		inventory	navigation bar.	hand will be updated in		
		-	2. User views the list of	real time without		
			inventories displayed.	refreshing the page.		

TC019	Record	Check out	<ul> <li>3. User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>4. User selects the "Check In" tab in the modal popup.</li> <li>5. User enters the quantity and remarks and clicks the "Confirm" button.</li> <li>1. User clicks on the "Check In" tab in the modal population.</li> </ul>	The selected	Pass	
	inventory	stock for an inventory	<ul> <li>In/Out Stock" tab from the top navigation bar.</li> <li>2. User views the list of inventories displayed.</li> <li>3. User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>4. User selects the "Check Out" tab in the modal popup.</li> <li>5. User enters the quantity and remarks and clicks the "Confirm" button.</li> </ul>	inventory's quantity on hand will be updated in real time without refreshing the page.		
TC020	Manage cycle count approval report	Approve cycle count approval report	<ol> <li>User logins into the system using the created manager-role user credentials.</li> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Approval Report" tab and views the list of approval reports listed.</li> </ol>	The summary report for the selected cycle counting will be generated with IRA calculated.	Pass	

TC021	Manage cycle count approval report	Reject cycle count approval report	<ul> <li>4. User clicks the "Approve" button for one of the approval reports.</li> <li>5. User clicks the "Confirm" button in the confirmation modal.</li> <li>6. User refresh the page.</li> <li>1. User clicks on the "View reports" tab from the top navigation bar.</li> <li>2. User selects the "Cycle Count Approval Report" tab and views the list of approval reports listed.</li> <li>3. User clicks the "Reject" button for one of the approval reports.</li> <li>4. User checks the checkbox for reassigning staff to recount the SKU.</li> <li>5. User clicks the "Confirm" button in the confirmation modal.</li> </ul>	A success message will be shown to indicate the success rescheduling.	Pass	
TC022	View reports	View cycle counting summary report	<ol> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Summary Report" tab and views the list of summary reports listed.</li> <li>User clicks on the cycle count ID for one of the summary reports</li> </ol>	The list of summary reports and their details will be displayed accordingly.	Pass	
TC023	View reports	View daily reports	1. User clicks on the "View reports" tab from the top navigation bar.	The list of daily reports and their details and stocks will be displayed accordingly.	Pass	

					1	
			2. User selects the "Daily report"			
			tab and views the list of daily			
			reports listed.			
			3. User clicks on the "View"			
			button for one of the daily reports			
TC024	Manage	Delete an	1. User logins into the system	The selected inventory	Pass	
	inventory	inventory	using <b>ADMIN</b> account provided.	is successfully deleted		
	-		2. User clicks on the "Manage	from the warehouse.		
			inventory" tab from the top			
			navigation bar.			
			3. User finds one of the newly			
			created inventories from the			
			displayed list (can utilize the			
			filter or search function) and			
			clicks on the "Delete" button.			
			4. User clicks on the "Delete"			
			button on the confirmation			
			modal.			
TC025	Manage	Delete a user	1. User logins into the system	The status of the	Pass	
10025	U	Delete a user	using <b>ADMIN</b> account.	selected user will be set	r ass	
	user		2. User clicks on the "Manage	to "INACTIVE".		
			e	IO INACTIVE .		
			user" tab from the top navigation			
			bar.			
			3. User selects one of the created			
			users from the displayed list and			
			clicks on the "Delete" button.			
			4. User clicks on the "Delete"			
			button on the confirmation			
			modal.			
TC026	Manage	Delete a	1. User logins into the system	The selected category	Pass	
	category	category	using ADMIN account.	will be deleted from the		
				list.		

				2. User clicks on the "Manage			
				category" tab from the top navigation bar.			
				3. User selects the newly created			
				category from the displayed list			
				and clicks on the "Delete" button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
TICOL	2.6	<b>D</b> 1		modal.		2	
TC027	Manage	Delete	a			Pass	
	warehouse	warehouse		using <b>ADMIN</b> account.	is successfully deleted		
				2. User clicks on the "Manage	with a success message.		
				warehouse" tab from the top			
				navigation bar.			
				3. User selects the newly created			
				warehouse from the displayed			
				list and clicks on the "Delete"			
				button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
				modal.			

			User Acceptance Testing	Form		
Tester's Name				Testing Start Date/ Time		
				Testing End Date/ Time		1
Test Case ID	Module	Test Case Title	Test Steps	Expected Results	Status (Pass/Fail)	Comments
TC001	Login account	Login an account	<ol> <li>User enters username and password of ADMIN account in the login page.</li> <li>User clicks on the login button.</li> </ol>	User will successfully logged-in into the system.	Pass	
TC002	Manage category	Add a new category	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> <li>User clicks on the "Add" button on the top right of the Manage category screen.</li> <li>User enters the name for the new category.</li> <li>User clicks on "confirm" button.</li> </ol>	The new category will be added successfully with a success notification.	Pass	
TC003	Manage category	Update a category name	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage category" tab from the top navigation bar.</li> <li>User selects the newly created category from the displayed list and clicks on the "Edit" button.</li> <li>User edits the name of the category from the modal popup</li> </ol>	The name of the category will be updated successfully with a success notification.	Pass	

			4. After editing, user clicks on the			
			"Confirm" button.			
TC004	Manage	Add a new	1. User logins into the system	The warehouse is added	Pass	
	warehouse	warehouse	using <b>ADMIN</b> account.	successfully with a		
			2. User clicks on the "Manage	success notification.		
			warehouse" tab from the top			
			navigation bar.			
			3. User clicks on the "Add"			
			button on the top right of the			
			Manage warehouse screen.			
			4. User enters the name, location,			
			number of bins (enter 100) and			
			zones (enter "A,B,C") for the			
			new warehouse.			
			5. User clicks on "confirm"			
			button after all information has			
			been entered.			
TC005	Manage	Add a new user	1. User logins into the system	Two new users will be	pass	
	user		using <b>ADMIN</b> account.	added successfully with	-	
			2. User clicks on the "Manage	a success notification.		
			user" tab from the top navigation			
			bar.			
			3. User clicks on the "Add"			
			button on the top right of the			
			Manage user screen.			
			4. User enters the name, email,			
			contact number, IC number, role			
			(select "STAFF"), employed in,			
			address and username for the new			
			user.			

		1	1		1	
			5. For the warehouse input, user			
			selects the newly created			
			warehouse.			
			6. User clicks on "confirm"			
			button after all information has			
			been entered.			
			7. Repeat step 3 to 6 to create			
			another user with STAFF role.			
TC006	Manage	Update a user's	1. User logins into the system	The details of the	Pass	
	user	detail	using <b>ADMIN</b> account.	selected user will be		
			2. User clicks on the "Manage	updated successfully		
			user" tab from the top navigation	with a success		
			bar.	notification.		
			3. User selects a newly created			
			user from the displayed list and			
			clicks on the "Edit" button.			
			4. User edits some information of			
			the user (except the warehouse			
			and role) on the modal pop-up			
			5. After editing, user clicks on the			
			"Confirm" button.			
TC007	Manage	Update a	1. User logins into the system	The warehouse details	Pass	
	warehouse	warehouse	using <b>ADMIN</b> account.	are updated		
			2. User clicks on the "Manage	successfully with a		
			warehouse" tab from the top	success notification.		
			navigation bar.			
			3. User selects the <b>newly created</b>			
			warehouse from the displayed			
			list and clicks on the "Edit"			
			button.			
			4. User edits the information by			
			adding a newly created user in			

				<ul><li>the warehouse manager field on the modal pop-up</li><li>5. After editing, user clicks on the "Confirm" button.</li></ul>			
TC008	Manage inventory	Add inventory	new	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage inventory" tab from the top navigation bar.</li> <li>User clicks on the "Add" button on the top right of the Manage inventory screen.</li> <li>For warehouse and category fields, selects the warehouse and category created earlier.</li> <li>User enters the name, quantity on hand, cost per unit and priority for the inventory.</li> <li>User clicks on "confirm" button after all information has been entered.</li> <li>Repeat step 3 to 5 to create one more inventory.</li> </ol>	Inventories are added successfully and a notification of the bin number assigned to newly added inventory is shown.		
TC009	Manage inventory	Update inventory	an	<ol> <li>User logins into the system using ADMIN account.</li> <li>User clicks on the "Manage inventory" tab from the top navigation bar.</li> <li>User selects a newly created inventory from the displayed list and clicks on the "Edit" button.</li> </ol>	The selected inventory details are updated according to user's input.	Pass	

T					
		the inventory on the modal pop-			
		up			
		5. After editing, user clicks on the			
		"Confirm" button.			
View all	View inventory	1. User logins into the system	User will successfully	Pass	
inventories	and its details	using <b>ADMIN</b> account.	view the inventory list		
		2. User clicks on the "Manage	and will be redirected to		
		inventory" tab from the top	the inventory details		
		navigation bar.	page once an inventory		
		3. User view the list of	id is clicked.		
		inventories displayed.			
		4. User clicks on one of the			
		inventory ID to view the details.			
Manage	Edit inventory	1. User logins into the system	The selected inventory	Pass	
warehouse	to storage bin	using <b>ADMIN</b> account.	will be added to the		
	_	2. User clicks on the "Manage	selected storage bin of		
		warehouse" tab from the top	the warehouse.		
		navigation bar.			
		3. User clicks on <b>newly created</b>			
		warehouse's ID to view the			
		details.			
		4. In the warehouse details page,			
		user selects the "storage bin" tab			
		to view the list of storage bins in			
		the warehouse			
		5. User selects the bin with the			
		newly created inventory (can use			
		the search to search for the			
		inventory) and clicks the "Edit			
		inventory" button.			
	inventories	inventories and its details Manage Edit inventory	View all inventoriesView inventory and its details5. After editing, user clicks on the "Confirm" button.View all 	View all inventoriesView inventory and its detailsthe inventory on the modal pop- up 5. After editing, user clicks on the "Confirm" button.User will successfully view the system using ADMIN account. 2. User clicks on the "Manage inventory" tab from the top navigation bar. 3. User view the list of inventories displayed. 4. User clicks on one of the inventory ID to view the details.User logins into the system using ADMIN account.Manage warehouseEdit inventory to storage bin a S. User clicks on the "Manage using ADMIN account. 2. User clicks on one of the inventory ID to view the details.The selected inventory will be added to the selected storage bin of the warehouse" tab from the top navigation bar. 3. User clicks on newly created warehouse" ID to view the details.Manage warehouseI. In the warehouse details page, user selects the "storage bin" tab to view the list of storage bins in the warehouse 5. User selects the bin with the newly created inventory (can use the search to search for the inventory) and clicks the "Edit	View all inventoriesView inventorythe inventory on the modal pop- up 5. After editing, user clicks on the "Confirm" button.User will successfully view the inventory list and its detailsPassView all inventoriesView inventory and its details1. User logins into the system using ADMIN account.User will successfully view the inventory list and will be redirected to the inventory details page once an inventory id is clicked.PassManage warehouseEdit inventory to storage bin1. User logins into the system using ADMIN account.The selected inventory will be added to the selected storage bin of the warehouse" tab from the top navigation bar.The selected inventory will be added to the selected storage bin of the warehouse" tab from the top navigation bar.PassManage warehouseEdit inventory to storage bin1. User logins into the system using ADMIN account.The selected inventory will be added to the selected storage bin of the warehouse.PassManage warehouseEdit inventory to storage bin1. User clicks on newly created warehouse is ID to view the details.The selected inventory the warehousePassManage warehouse5. User selects the bin with the newly created inventory (can use the search to search for the inventory) and clicks the "EditStorage binHerein the warehouse

	1	[	[			[
			6. User clears the current			
			inventory assigned on the modal			
			popup and click on the "Edit			
			inventory" button.			
TC012	Manage	Assign	1. User logins into the system	The category of the	Pass	
	warehouse	category to a	using <b>ADMIN</b> account.	selected bins is changed		
		storage bin	2. User clicks on the "Manage	to the selected category.		
			warehouse" tab from the top			
			navigation bar.			
			3. User clicks on <b>newly created</b>			
			warehouse's ID to view the			
			details.			
			4. In the warehouse details page,			
			user selects the "storage bin" tab			
			to view the list of storage bins in			
			the warehouse			
			5. User clicks on the "Multi			
			assign bin" to assign category to			
			bins.			
			6. User selects a category and			
			checks desired bin to assign.			
			7. User clicks on the "Assign"			
			button.			
TC013	Login	Reset password	1. User logins into the system	A success notification	Pass	
	account	-	using <b>ADMIN</b> account.	will be shown.		
			2. User clicks on the "Manage			
			User" tab and finds the newly			
			created user with "MANAGER"			
			role.			
			3. User takes note on the			
			username and contact number.			

	1				1	
			4. User logouts and logins into			
			the system using the username			
			and contact number as the			
			password.			
			5. User resets the password			
TC014	Manage	Start a new	1. User logins into the system	A success notification	Pass	
	cycle	cycle counting	using the created manager-role	will be shown stating		
	counting		user credentials (skip this step	that the cycle counting		
			if you completed TC013)	is started. Schedules		
			2. Note that there are <b>TWO</b>	will be created for all		
			inventories listed in the View	SKUs and assigned to		
			Inventory page, if no, kindly go	the selected staffs		
			back to TC008 to create an	evenly.		
			inventory.	-		
			3. User clicks on the "Start Cycle			
			counting" from the "Manage			
			Cycle counting" tab dropdown on			
			the top navigation bar.			
			3. User fills in all of the			
			information in the form			
			displayed.			
			4. User selects all inventories in			
			the "Select Inventories" field.			
			4. User clicks the "Submit"			
			button after completing the form.			
			5. User clicks the "Create" button			
			in the summary modal popup.			
TC015	Manage	View all	1. User clicks on the "View Cycle	The list of schedules for	Pass	
	cycle	upcoming cycle	counting" from the "Manage	the cycle counting and		
	counting	counting SKUs	Cycle counting" tab dropdown on	the settings will be		
	-	_	the top navigation bar.	displayed accordingly.		

	1					
			2. User clicks on the "View cycle			
			count settings" button on the top			
			right of the displayed table.			
TC016	View all	View all staff in	1. User clicks on the "View Staff"	The list of staffs and	Pass	
	staff	a warehouse	tab on the top navigation bar.	their details will be		
			2. User clicks on one of the staff	displayed accordingly.		
			ID from the list to view the staff			
			details.			
TC017	Perform	Perform cycle	1. Repeat TC013 to login and	The cycle counting	Pass	
	cycle	counting for an	reset password for the created	record is created with		
	counting	inventory	STAFF-role user.	success notification.		
	_	-	2. User clicks on the "Cycle			
			Counting" tab from the top			
			navigation bar.			
			3. User selects the "Upcoming"			
			tab and views the list of			
			upcoming cycle counting			
			schedules			
			3. User chooses the first one in			
			the list and clicks the "Count"			
			button.			
			4. User enters the actual count on			
			the modal popup			
			5. User clicks the "Create" button			
			6. User refresh the page.			
			7. Repeat step 3 to 6 to count			
			another one.			
TC018	Record	Check in stock	1. User clicks on the "Check	The selected	Pass	
	inventory	for an	In/Out Stock" tab from the top	inventory's quantity on		
		inventory	navigation bar.	hand will be updated in		
		-	2. User views the list of	real time without		
			inventories displayed.	refreshing the page.		

TC019	Record	Check out	<ol> <li>User chooses one of the inventories and clicks on the "Check In/Out Stock" button.</li> <li>User selects the "Check In" tab in the modal popup.</li> <li>User enters the quantity and remarks and clicks the "Confirm" button.</li> <li>User clicks on the "Check</li> </ol>	The selected	Pass	
10019	inventory	Check out stock for an inventory		The selected inventory's quantity on hand will be updated in real time without refreshing the page.	Pass	
TC020	Manage cycle count approval report	Approve cycle count approval report	-	The summary report for the selected cycle counting will be generated with IRA calculated.	Pass	

					1	
			<ul> <li>4. User clicks the "Approve" button for one of the approval reports.</li> <li>5. User clicks the "Confirm" button in the confirmation modal.</li> </ul>			
			6. User refresh the page.			
TC021	Manage cycle count approval report	Reject cycle count approval report	<ol> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Approval Report" tab and views the list of approval reports listed.</li> <li>User clicks the "Reject" button for one of the approval reports.</li> <li>User checks the checkbox for reassigning staff to recount the SKU.</li> <li>User clicks the "Confirm" button in the confirmation modal.</li> </ol>	A success message will be shown to indicate the success rescheduling.	Pass	
TC022	View reports	View cycle counting summary report	<ol> <li>User clicks on the "View reports" tab from the top navigation bar.</li> <li>User selects the "Cycle Count Summary Report" tab and views the list of summary reports listed.</li> <li>User clicks on the cycle count ID for one of the summary reports</li> </ol>	The list of summary reports and their details will be displayed accordingly.	Pass	
TC023	View reports	View daily reports	1. User clicks on the "View reports" tab from the top navigation bar.	The list of daily reports and their details and stocks will be displayed accordingly.	Pass	

			1		1
			2. User selects the "Daily report"		
			tab and views the list of daily		
			reports listed.		
			3. User clicks on the "View"		
			button for one of the daily reports		
TC024	Manage	Delete an	1. User logins into the system	The selected inventory	Pass
	inventory	inventory	using <b>ADMIN</b> account provided.	is successfully deleted	
	-		2. User clicks on the "Manage	from the warehouse.	
			inventory" tab from the top		
			navigation bar.		
			3. User finds one of the newly		
			created inventories from the		
			displayed list (can utilize the		
			filter or search function) and		
			clicks on the "Delete" button.		
			4. User clicks on the "Delete"		
			button on the confirmation		
			modal.		
TC025	Manage	Delete a user	1. User logins into the system	The status of the	Pass
	user		using <b>ADMIN</b> account.	selected user will be set	
			2. User clicks on the "Manage	to "INACTIVE".	
			user" tab from the top navigation		
			bar.		
			3. User selects one of the created		
			<b>users</b> from the displayed list and		
			clicks on the "Delete" button.		
			4. User clicks on the "Delete"		
			button on the confirmation		
			modal.		
TC026	Manage	Delete a	1. User logins into the system	The selected category	Pass
	category	category	using <b>ADMIN</b> account.	will be deleted from the	
				list.	
		1		1151.	

				2. User clicks on the "Manage			
				category" tab from the top			
				navigation bar.			
				3. User selects the newly created			
				<b>category</b> from the displayed list			
				and clicks on the "Delete" button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
				modal.			
TC027	Managa	Delete	0		The selected warehouse	Pass	
10027	Manage		a			rass	
	warehouse	warehouse		using <b>ADMIN</b> account.			
				2. User clicks on the "Manage	with a success message.		
				warehouse" tab from the top			
				navigation bar.			
				3. User selects the newly created			
				warehouse from the displayed			
				list and clicks on the "Delete"			
				button.			
				4. User clicks on the "Delete"			
				button on the confirmation			
				modal.			

# APPENDIX D: User Satisfaction Survey Form

9/8/22, 3:22 PM

User Satisfaction Survey

# User Satisfaction Survey

1. I think that I would like to use this system/ website.

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree

## 2. I found the system/website unnecessarily complex.

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree

#### 3. I thought the system/website was easy to use

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree

 I think that I would need the support of a technical person to be able to use this system/website.

Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree
	1	2	3	4	5	
Mark only one oval.						

https://docs.google.com/forms/d/1brf6MBUvIUYEg8bU9DDznVGSZ0lkb4JnXxxJQVan1Xs/edit

#### 9/8/22, 3:22 PM

#### User Satisfaction Survey

5. I found this system/website was easily moved through without a lot of backtracking or data re-entry.

Mark only one oval.						
	1	2	3	4	5	
Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree

6. I thought there was too much inconsistency in this system/website.

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree

7. I would imagine that most people would learn to use this website very quickly.

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree

## 8. I found the system/website very awkward to use.

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree

9/8/22.	3:22	PM

User Satisfaction Survey

9. I felt very confident using the system/website.

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree

10. I needed to learn a lot of things before I could get going with this system/website.

Mark only one oval.

	1	2	3	4	5	
Strongly Disagree	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	$\bigcirc$	Strongly Agree

11. What do you like best about the system

12. What do you like least about the system

https://docs.google.com/forms/d/1brf0MBUvIUYEg8bU9DDznVGSZ0lkb4JnXxxJQVan1Xs/edit

User Satisfaction Survey

13. If you were to describe this site to a colleague in a sentence or two, what would you say?

14. Do you have any other final comments or questions?

This content is neither created nor endorsed by Google.

Google Forms

https://docs.google.com/forms/d/1brf0MBUvIUYEg8bU9DDznVGSZ0lkb4JnXxxJQVan1Xs/edit