INVENTORY MONITORING WEB PORTAL DEVELOPMENT USING LOW CODE PROGRAMMING

By

Tan Rui Zheng

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

DECLARATION OF ORIGINALITY

I declare that this report entitled "**INVENTORY MONITORING WEB PORTAL DEVELOPMENT USING LOW CODE PROGRAMMING**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : Tan Rui Zheng

Date : 13/9/2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: INVENTORY MONITORING WEB PORTAL DEVELOPMENT USING LOW CODE PROGRAMMING

Academic Session: ___Jun 2024_____

I _____TAN RUI ZHENG_____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ ___ _____

(Author's signature) (Supervisor's signature)

Address:

__17, Jalan Laksamana 10,_____

_Taman Desa Idaman_____ ____Goh Hock Guan_____

_41200 Klang, Selangor_____ Supervisor's name

Date: __13/9/2024_____ **Date**:____13/9/2024_____

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: ___13/9/2024_____

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ___Tan Rui Zheng_____ (ID No: __***20ACB04806***__) has completed this final year project/ dissertation/ thesis* entitled "INVENTORY MONITORING WEB PORTAL DEVELOPMENT USING LOW CODE PROGRAMMING" under the supervision of Mr Goh Hock Guan(Supervisor) from the Department of Information System, Faculty of Information and Communication Technology from the Department of Information System, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(*Tan Rui Zheng*)

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Goh Hock Guan who has given me guidance throughout this web portal development for my final year project. A million thanks to you.

Finally, I need to thank my dear family and friends for the suggestions and encouragement they have given me when I was developing my final year project. Their support carried me this far.

ABSTRACT

This project is about developing an Inventory Monitoring Web Portal using low code programming. The programming language that was used is Node-RED language. The reason Node-RED was used is because low-code programming languages have the advantages of reduced complexity, high scalability, and rapid prototyping.

In this project, the Inventory Monitoring System is a comprehensive software solution designed to streamline and optimize the management of laboratory equipment and resources. In an environment where accuracy and accountability are critical, this system encourages efficient monitoring of items to meet the needs of its users, which includes students, lab staff, and supervisors.

An effective inventory monitoring system is crucial for ensuring the availability of laboratory equipment while minimizing issues such as item misplacement or mismatching. Manual tracking methods are prone to errors and can be time-consuming, leading to inefficiencies and potential equipment loss. This project addresses the need for a modern inventory monitoring system that automates and simplifies the tracking and management of lab items.

The project begins with a thorough analysis of existing inventory systems, as well as research into journal articles on inventory management practices, to identify the strengths, limitations, and possible improvements for laboratory environments.

Table of Contents

INVENTORY MONITORING WEB PORTAL DEVELOPMENT USING LOW CODE PROGRAMMING I

DECLARATION OF ORIGINALITY II

ACKNOWLEDGEMENTS III

ABSTRACT VI

CHAPTER 1 INTRODUCTION 5

Project Background 5

1.1 Problem Statement and Motivation 6
Problem Statement 6
Motivation 6

1.2 Project Objectives 7
1.2.1 To design a simple, cost-effective system 7
1.2.2 To design a user-friendly system 7

1.3 Project Scope 8
1.3.1 Inventory set up and Data Entry 8
1.3.2 Real-time Inventory Monitoring 8
1.3.3 Scalability 8

1.4 Expected Contributions 9
1.4.1 Development of an Inventory Monitoring Web Portal with user-friendly interface 9
1.4.2 Accessibility and inclusivity 9
1.4.3 Efficiency and Cost Reduction 9

1.5 Organization of the Report 10

CHAPTER 2 LITERATURE REVIEW 11

2.1 Review of the Programming Language 11

2.2 Previous Works on Inventory Monitoring System 12
2.2.1 Web Portal and System for Management of Critical Reagent Inventory 12
2.2.2 Design and Implementation of an Online Inventory Monitoring System 15
2.2.3 The Design and Implementation of an Online Inventory Monitoring System 15
2.2.4 Review of Zoho Inventory System 16

2.3 Limitation of Previous Studies 23
2.3.1 Web Portal and System for Management of Critical Reagent Inventory 23
2.3.2 Design and Implementation of both Online Inventory Monitoring System 23

2.4 Summary of the existing systems 25

2.5 Proposed Solutions 26

CHAPTER 3 SYSTEM METHODOLOGY 27

3.1 System Development Models 27
3.1.1 Waterfall Model 27
3.1.2 Rapid Application Development Model (RAD) 29
3.1.3 Agile Model 31

3.1.4 Methodology selected 32

3.2 Functional Requirement 34

3.3 Project Milestone 35

CHAPTER 4 SYSTEM DESIGN 36

4.1 System Architecture 36

4.2 Functional Modules in the System 38

4.3 System Flow 52
4.3.1 Flowchart 52
4.3.2 Activity Diagrams 54

4.4 GUI Design 58

4.5 Concluding Remark 61

CHAPTER 5 SYSTEM IMPLEMENTATION 62
5.1 Hardware Setup 62
5.2 Software Setup 63
5.3 Setting and Configuration 63
5.4 System Operations 67
5.4.1 User Login Page 67
5.4.2 Sign Up Page 70
5.4.3 Student Interface 72
5.4.4 Supervisor Interface 75
5.4.5 Staff Interface 81

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION 93
6.1 System Testing and Performance Metrics 93

6.1.1 Sign Up Use Case 93
6.1.2 Log In Use Case 94
6.1.3 Log Out Use Case 95
6.1.4 Student Borrow Use Case 96
6.1.5 Staff Add Inventory Use Case 97
6.1.6 Staff Update Inventory Use Case 98
6.1.7 Staff Remove Inventory Use Case 99
6.1.8 Supervisor Sign Up Request Use Cases 99
6.1.9 Supervisor Borrow Request Use Cases 99
6.1.10 Staff Borrow Use Case 100
6.1.11 Staff Return Use Case 100
6.2 Testing Setup and Result 101
6.3 Project Challenges 104
6.4 Objectives Evaluation 106

CHAPTER 7 CONCLUSION AND RECOMMENDATIONS 107
7.1 Conclusion 107
7.2 Recommendations 108

REFERENCES 109

APPENDIX 110

CHAPTER 1

LIST OF FIGURES

FIGURE 1 IMPORTANT DESIGN FEATURES AND USER NEEDS 13
FIGURE 2 ZOHO MAIN PAGE 16
FIGURE 3 ZOHO DASHBOARD 17
FIGURE 4 ZOHO DASHBOARD (CONT) 18
FIGURE 5 ZOHO ITEMS 18
FIGURE 6 ZOHO ITEMS (CONT) 19
FIGURE 7 ZOHO SALES ORDER 19
FIGURE 8 ZOHO PACKAGES 20
FIGURE 9 ZOHO INTEGRATIONS 20
FIGURE 10 ZOHO REPORTS 21
FIGURE 11 ZOHO CONFIGURATIONS 21
FIGURE 12 ZOHO SETTINGS 22
FIGURE 13 WATERFALL MODEL 27
FIGURE 14 RAPID APPLICATION DEVELOPMENT (RAD) MODEL 29
FIGURE 15 AGILE METHODOLOGY 31
FIGURE 16 TIMELINE OF FINAL YEAR PROJECT 2 35
FIGURE 17 DESIGN BLOCK DIAGRAM OF INVENTORY MONITORING WEB PORTAL 36
FIGURE 18 USE CASE DIAGRAM OF THE INVENTORY MONITORING WEB PORTAL 38
FIGURE 19 FLOWCHART OF THE INVENTORY MONITORING WEB PORTAL 52
FIGURE 20 ACTIVITY DIAGRAM OF SIGN UP USE CASE 54
FIGURE 21 ACTIVITY DIAGRAM OF LOGIN USE CASE 55
FIGURE 22 ACTIVITY DIAGRAM OF STAFF INVENTORY MANAGEMENT USE CASES 56
FIGURE 23 ACTIVITY DIAGRAM OF BORROW AND RETURN USE CASES 57
FIGURE 24 LOGIN INTERFACE DESIGN 58
FIGURE 25 SIGN UP INTERFACE DESIGN 58
FIGURE 26 STAFF INTERFACE DESIGN 59
FIGURE 27 STUDENT INTERFACE DESIGN 59
FIGURE 28 SUPERVISOR INTERFACE DESIGN 60
FIGURE 29 INVENTORY.CSV FILE CONTENT 63
FIGURE 30 REQUEST.CSV FILE CONTENT 64
FIGURE 31 USER_INFO.CSV FILE CONTENT 64
FIGURE 32 EXECUTION OF NODE-RED ON COMMAND PROMPT 65
FIGURE 33 NODE-RED EDITOR 65
FIGURE 34 PALETTE NODE-RED-DASHBOARD 66
FIGURE 35 USER LOGIN PAGE 67
FIGURE 36 INVALID USERNAME 68
FIGURE 37 INVALID PASSWORD 68
FIGURE 38 SUCCESSFUL LOGIN 69
FIGURE 39 SIGN UP PAGE 70
FIGURE 40 EXISTING USERNAME 71
FIGURE 41 SUCCESSFUL SIGN UP 71
FIGURE 42 UPDATED REQUEST.CSV 71
FIGURE 43 STUDENT INTERFACE 72
FIGURE 44 SELECTED ROW, BORROW PROMPT 73
FIGURE 45 BORROW AMOUNT 73
FIGURE 46 BORROW REQUEST SENT 74
FIGURE 47 UPDATED REQUEST.CSV 74
FIGURE 48 BORROW AMOUNT EXCEEDING EXISTING QUANTITY 75
FIGURE 49 SUPERVISOR INTERFACE 76
FIGURE 50 SELECTED ROW, APPROVE PROMPT 76

FIGURE 51 APPROVED NOTIFICATION 77
FIGURE 52 DENIED REQUEST NOTIFICATION, REMOVAL OF SELECTED ROW 77
FIGURE 53 UPDATED STATUS IN REQUEST.CSV AND REMOVAL OF REQUEST 78
FIGURE 54 UPDATED STATUS IN USER_INFO.CSV 78
FIGURE 55 NEW SIGN UP REQUEST 78
FIGURE 56 SELECTED SIGN UP REQUEST, APPROVE PROMPT 79
FIGURE 57 APPROVED SIGN UP REQUEST NOTIFICATION 79
FIGURE 58 UPDATED USER_INFO.CSV 80
FIGURE 59 DENIED USER SIGN UP REQUEST 80
FIGURE 60 STAFF INTERFACE 81
FIGURE 61 ADD NEW INVENTORY, KEY IN ITEM POP UP 82
FIGURE 62 ADD ITEM NOTIFICATION 82
FIGURE 63 UPDATED INVENTORY.CSV, ADDED NEW ITEM 83
FIGURE 64 SELECT ROW, UPDATE ITEM POP UP 83
FIGURE 65 KEY IN ITEM PROMPT 84
FIGURE 66 UPDATED ITEM NOTIFICATION 84
FIGURE 67 UPDATED INVENTORY.CSV WITH UPDATED ITEM INFORMATION 85
FIGURE 68 SELECT ROW, REMOVE ITEM POP UP 85
FIGURE 69 REMOVED ITEM NOTIFICATION 86
FIGURE 70 UPDATED INVENTORY.CSV WITH REMOVED SELECTED ITEM 86
FIGURE 71 USER STATUS IN USER_INFO.CSV BEFORE STAFF BORROW 87
FIGURE 72 USER STATUS IN REQUEST.CSV BEFORE STAFF BORROW 87
FIGURE 73 INVENTORY.CSV BEFORE STAFF BORROW 88
FIGURE 74 BORROW REQUEST 88
FIGURE 75 ITEM BORROWED NOTIFICATION 89
FIGURE 76 UPDATED INVENTORY.CSV AFTER STAFF BORROW 89
FIGURE 77 UPDATED USER STATUS AND BORROW DATE AND TIME IN REQUEST.CSV AFTER STAFF BORROW 90
FIGURE 78 UPDATED STUDENT STATUS AND BORROWED AMOUNT IN USER_INFO.CSV 90
FIGURE 79 ITEM RETURNED NOTIFICATION 91
FIGURE 80 UPDATED ITEM QUANTITY IN INVENTORY.CSV AFTER STAFF RETURN 91
FIGURE 81 UPDATED REQUEST.CSV WITH RETURN DATE, TIME AND DURATION 92
FIGURE 82 UPDATED USER STATUS IN USER_INFO.CSV AFTER STAFF RETURN 92
FIGURE 83 SET UP INVENTORY.CSV CONTENT 101
FIGURE 84 SET UP REQUEST.CSV CONTENT 101
FIGURE 85 SET UP USER_INFO.CSV CONTENT 102
FIGURE 86 FINAL INVENTORY.CSV CONTENT 102
FIGURE 87 FINAL REQUEST.CSV CONTENT 103
FIGURE 88 FINAL USER_INFO.CSV CONTENT 103

LIST OF TABLES

TABLE 1 ISSUES AND STEPS TAKEN TO RESOLVE THE ISSUE 16
TABLE 2 SUMMARY OF EXISTING SYSTEMS 25
TABLE 3 SIGN UP USE CASE 39
TABLE 4 LOGIN USE CASE 40
TABLE 5 LOGOUT CASE 41
TABLE 6 STUDENT BORROW ITEM USE CASE 42
TABLE 7 SUPERVISOR APPROVE SIGN UP REQUEST USE CASE 43
TABLE 8 SUPERVISOR DENY SIGN UP REQUEST USE CASE 44
TABLE 9 SUPERVISOR APPROVE BORROW REQUEST USE CASE 45
TABLE 10 SUPERVISOR DENY BORROW REQUEST USE CASE 46
TABLE 11 STAFF GENERATE BORROW ITEM USE CASE 47
TABLE 12 STAFF GENERATE RETURN ITEM USE CASE 48
TABLE 13 STAFF ADD NEW INVENTORY USE CASE 49
TABLE 14 STAFF UPDATE INVENTORY USE CASE 50
TABLE 15 STAFF REMOVE INVENTORY USE CASE 51
TABLE 16 SPECIFICATIONS OF LAPTOP 62
TABLE 17 SYSTEM TESTING AND PERFORMANCE METRICS FOR SIGN UP 93
TABLE 18 SYSTEM TESTING AND PERFORMANCE METRICS FOR LOG IN 94
TABLE 19 SYSTEM TESTING AND PERFORMANCE METRICS FOR LOG OUT 95
TABLE 20 SYSTEM TESTING AND PERFORMANCE METRICS FOR STUDENT BORROW 96
TABLE 21 SYSTEM TESTING AND PERFORMANCE METRICS FOR STAFF ADD INVENTORY 97
TABLE 22 SYSTEM TESTING AND PERFORMANCE METRICS FOR STAFF UPDATE INVENTORY 98
TABLE 23 SYSTEM TESTING AND PERFORMANCE METRICS FOR STAFF REMOVE INVENTORY 99
TABLE 24 SYSTEM TESTING AND PERFORMANCE METRICS FOR SUPERVISOR SIGN UP REQUEST 99
TABLE 25 SYSTEM TESTING AND PERFORMANCE METRICS FOR SUPERVISOR BORROW REQUEST 99
TABLE 26 SYSTEM TESTING AND PERFORMANCE METRICS FOR STAFF BORROW 100
TABLE 27 SYSTEM TESTING AND PERFORMANCE METRICS FOR STAFF RETURN 100

LIST OF ABBREVIATIONS

API	Application Programming Interface
RAD	Rapid Application Development
SDLC	System Development Life Cycle
LC	Low-Code
GUI	Graphical User Interface

CHAPTER 1 Introduction

Project Background

Efficient inventory management is vital in ensuring smooth operations within academic laboratory environments. The ability to track and monitor the status of borrowed equipment is crucial for maintaining an organized system, preventing misplacements, and ensuring that resources are available when needed. Traditional manual methods of managing inventory, such as pen-and-paper logging or basic spreadsheets, are often prone to human error, lack real-time updates, and can result in inefficiencies.

Node-RED, a low-code programming tool, enables the development of applications through an intuitive clicking interface. By connecting nodes that represent various functions, it allows for streamlined automation and real-time data management without requiring extensive coding expertise.

This project leverages low-code programming, specifically Node-RED, to develop a web portal for inventory monitoring in university laboratories. The portal is designed to enable lab staff to easily track laboratory items borrowed and returned by students. The system aims to replace outdated manual processes with a more efficient, automated solution tailored to the unique needs of educational institutions, thereby reducing errors and improving accessibility for all users.

1.1 Problem Statement and Motivation

Problem Statement

The problem addressed by this project is the inefficiency and limitations of current inventory monitoring systems across industries. Existing systems are often outdated, labor-intensive, and prone to inaccuracies, leading to issues such as inaccurate tracking and item mismatch. Schools and institutions tend to struggle with acquiring and maintaining complex inventory management systems, hindering operational efficiency and decision-making. As for the larger institutions, the scale of their systems is enormous, and they face high maintenance costs for their advanced systems. These challenges cultivate the need for an efficient and accessible inventory monitoring solution.

Motivation

The main motivation of this project is to provide a user-friendly and efficient tool for inventory monitoring, specifically designed for educational institutions like universities. Effective inventory management is essential to ensure smooth operations in laboratory environments, minimizing the risk of item misplacement and optimizing the use of resources. Many labs still rely on manual tracking systems, which are prone to errors and inefficiencies.

By developing an Inventory Monitoring Web Portal using low-code programming, this project aims to create an accessible and straightforward solution tailored to the needs of students, lab staff, and supervisors. The project is motivated by the potential to improve operational efficiency by automating the tracking of laboratory items and simplifying the overall inventory management process.

1.2 Project Objectives

The aim of the thesis is to develop an Inventory Monitoring Web Portal that tackles the limitations of manual tracking systems commonly used in educational labs, while incorporating useful features and innovations. The main target user groups are university lab staff, students, and supervisors who need an easy-to-use system for monitoring lab equipment.

1.2.1 To design a simple, cost-effective system

A simple system with all the basic functionalities of an Inventory Monitoring System reduces the cost of development and maintenance, while abandoning the need to rely on external vendors for large scale development. The changes of system will be made according to the ongoing user requirements. This can remove any redundancy in the system and further reduce the cost of development and maintenance.

1.2.2 To design a user-friendly system

The system features a simple and straightforward user interface, with basic functionalities for inventory tracking. The simplicity of the design will make it easy for lab staff and students to use without extensive training.

A simple manual will be provided serving as a guide to ensure a smooth user experience.

This project DOES NOT cover the objectives of integration with other systems, security and compliances, as well as cross platform availability.

1.3 Project Scope

The main scope of this project is to develop an Inventory Monitoring Web Portal that caters to the needs of the university lab staff, students and supervisors, as well as resolving the issues in the existing systems in the lab environment. The scopes of the project can be broken down into a few parts.

1.3.1 Inventory set up and Data Entry

The project consists of the development of an intuitive inventory set up process within the web portal. Lab staffs will be able to register and set up their inventory effortlessly. This will include user-friendly registration forms and mechanisms for users to input essential item information such as item ID, item name, item quantity. The goal is to streamline the process of users entering their inventory data, ensuring a smooth and efficient user experience.

1.3.2 Real-time Inventory Monitoring

A core feature of the system is real-time monitoring of lab items. Users will be able to view current inventory levels, item locations, and status updates (e.g., borrowed/returned items) via a dashboard. This will help prevent missing equipment and enhance lab efficiency.

1.3.3 Scalability

Although the system is designed for a small-scale university lab environment, it is built with flexibility to accommodate potential future features or user feedback. This approach ensures that the web portal remains adaptable and responsive to changes in the inventory management landscape.

1.4 Expected Contributions

1.4.1 Development of an Inventory Monitoring Web Portal with user-friendly interface

A web-based portal will be designed, developed and deployed using Node-RED, which is a low-code programming technique to manage inventory in real time. The portal will feature a user-friendly interface for users to navigate, with the functionalities such as track inventory levels, borrow requests approval and many more.

1.4.2 Accessibility and inclusivity

By employing low-code programming, the solution aims to make inventory management accessible to users with different levels of technical knowledge. This inclusivity will enable a wider range of users to participate in the changing of the system, inventory monitoring and decision-making.

1.4.3 Efficiency and Cost Reduction

The inventory monitoring web portal will enhance efficiency by automating the inventory tracking process, reducing the risk of errors, and improving resource management. This will increase operational efficiency for lab staff and students, reducing the time spent on manual tasks and minimizing equipment losses.

1.5 Organization of the Report

Chapter 1 introduce the project background, problem statements, motivation, project objectives, project scope as well as the contributions from the project.

Chapter 2 consists of the literature review of the technologies, and the existing inventory monitoring web portals. In addition to that, this chapter also includes the limitations of the existing systems and the proposed solutions to these limitations.

Chapter 3 regards the system methodology. System development models chosen, system requirements, functional requirement, and project milestone were included in this chapter.

Chapter 4 contains the system design including the system architecture, system functional modules, system flow, and GUI design.

Chapter 5 includes the system implementation where the hardware setup, software setup, setting and configuration were recorded. The system operation was also included in this chapter.

Chapter 6 includes system evaluation and discussion about the system testing and performance metrics, testing setup and result, project challenges and objectives evaluation.

Chapter 7 concludes the summarization of the whole report of the project and the recommendations for future work.

CHAPTER 2 Literature Review

2.1 Review of the Programming Language

This project was developed using low-code programming language Node-RED. Literature review was conducted on Node-RED. According to [1], low-code (LC) software development is a new way to create applications that doesn't require a lot of manual coding. With low-code platforms, you use easy-to-understand visual tools to build software using pre-made parts that you can drag and drop into place. This method makes it possible for people with different levels of technical skill to create simple software quickly and easily.

Low-code greatly increase the accessibility of software development for individuals with limited coding experience. Secondly, low-code speeds up application development through rapid prototyping. Thirdly, it reduces development costs by minimizing the need for specialized developers and extensive training, making software creation more cost-effective. Lastly, low-code development promotes faster innovation by enabling swift experimentation, iteration, and adaptation to changing business requirements, ultimately driving continuous improvement and responsiveness in software development processes.

2.2 Previous Works on Inventory Monitoring System

This project primarily focuses on the development of Inventory Monitoring System on Web Portal. The journals regarding Inventory Monitoring System will be reviewed and determine the related existing system on the market for their strengths, weaknesses and how do we resolve the weaknesses mentioned. Evaluating the existing systems will help in the development of this project by applying the strengths and resolving the weaknesses of the existing systems.

2.2.1 Web Portal and System for Management of Critical Reagent Inventory

The authors of this report created and implemented an inventory management system called LCM+ to provide users with efficient and secured access through web portal to the critical reagent inventory tracking and information. As suggested by Maravadi [2], an inventory management system is very important to track the reagent inventory. However, most reagent inventory management systems are typically designed to exclusively support the management of internal reagent within the organization [3]. Therefore, these systems were unable to manage the critical reagent efficiently for external clients by providing others the accessibility and interaction ability with the software.

That is why the authors strived to find out the critical reagent inventory management system design to fit the needs of the users in the bioanalytical field. After thorough analysis, some of the requirements worth noting for system development were different levels of access for different users, dashboards, ability to perform the task with few clicks, increased operational efficiency and reduce system usage time, the users' ability to perform multiple functions.

To fulfil the requirements listed, low code/no-code solutions were chosen by the authors for rapid application development, to make the process of constructing the database to maximize the optimization, enabling more user input.

Table 1. Key design characteristics and user requirements: the features found in B2S's LCM+ system as compared with other commercial off-the-shelf or home-built systems.

Key design characteristics and user requirements	Off-the-shelf commercial software solution	Spreadsheets, collaboration sites and desktop-based database management	B2S LCM+ information technology platform
Password-authenticated login	Yes	Yes	Yes
Documentation of reagent characteristics	Yes	Yes	Yes
Inventory management tracking of critical reagents	Yes	Yes	Yes
Ability to access information anywhere at anytime	Sometimes	Yes	Yes
Configuration supports critical reagent attributes and processes	Yes	Yes	Yes
Third-party controlled access	No	No	Yes
Monitoring of critical reagent logistics during shipping	No	No	Yes
Regulatory compliance (21 Code of Federal Regulations part 21)	Sometimes	No	Yes

Figure 1 Important design features and user needs

The authors utilized a commercial laboratory information management system to develop an enhanced inventory management system. Furthermore, a customized web-based portal named the LCM+ Portal was developed to connect the authors' system, offering a regulated interface for users who are the clients to access from externally, request, and manage critical reagents.

By choosing the Rapid Application Development (RAD) model for building the foundation of the LCM+ Portal, the starting development was fast and the inputs from stakeholders can be incorporated earlier. This strategy also made the development process more refined, efficient and optimized.

In the LCM+ system, a relational structure is employed that associates the data tables together. Workflow functionalities and enhancements to the user interface are integrated with the database to ensure a smooth user experience. The LCM+ system also establishes internal and external teams to separate the permitted users to the specific part of the system. A user can be granted access to one or multiple projects. By assigning a project manager, they can authorize the external teams to access the related projects. An authorized user with project access can see all samples within the project. Data can be accessed and monitored in real-time through the portal. For the requisitions part, the movement of the inventory whether it is internally or externally can be generated. Authorized users can track the status of requisitions using the system. Additionally, the process of updating the courier details tracking is updated in the records during shipment.

The LCM+ system boasts numerous standout features, such as regulated user access, permissions tailored to specific user roles, and streamlined process workflows.

Changes in the system are tracked with a comprehensive audit log that users can access based on their permissions. Each record includes an audit trail showing previous and updated data, timestamp of change, and the user responsible. The system also features robust search functionality at both database and table levels, with users able to view search results based on their access permissions. Users can customize their experience by creating custom lists, applying filters, and designing personalized dashboards.

For effective maintenance of the portal system, a small, specialized team of proficient users was essential for the efficiency of the operation. These group of users possess in-depth knowledge of the way the portal system operates, although they do not necessarily need to be IT professionals since the system employs simpler technologies compared to older systems. The team size is determined by the system's implementation needs for training and troubleshooting. System changes can be implemented at both detailed and comprehensive levels. For example, adding new email notifications for system changes requires minimal adjustments, while larger tasks may necessitate structural reworking. Most system modifications involved layering new components onto existing layers. Major project developments often involve contracting external vendors. Routine backups are conducted using a third-party vendor.

2.2.2 Design and Implementation of an Online Inventory Monitoring System

In this project, a store inventory monitoring system was created, incorporating various subsystems including a web portal and an Android application for monitoring of inventory. The web portal manages stock and vendor records, conducts supply and sales operations, and stores data in a database. The Android app monitors inventory levels, displays sales forecasts and analytics, and accesses vendor records. Additionally, a RESTful API was developed to facilitate data exchange between subsystems and the database. The project utilized technologies like HTML, CSS, JavaScript, NodeJS, React-Native, and PostgreSQL.

The system in this project has very minimal hardware requirements, and the records can be easily retrieved.

2.2.3 The Design and Implementation of an Online Inventory Monitoring System

In this project, an Online Inventory Monitoring System was established for shops. Similar to the previous project, this system consists of various subsystems to achieve the goal. These subsystems include a web portal and an Android application, as well as a RESTful API to make it easier to exchange data between different systems on different platforms.

This project faces certain issues where the researcher was unable to control fully. The issues faced and the steps taken to resolve them by the researcher are:

Table 1 Issues And Steps Taken To Resolve The Issue

Issues faced	**Steps taken to resolve**
Limited access to store hardware.	A mock store setup on available local system.
Bugs or errors from third-party APIs.	Limit the use of third-party APIs to only stable APIs such as Google.
Insufficient hardware device to test the developed mobile application.	Use recommended hardware device for development.

The notable strength of this system is that the system is very flexible for future modifications. This system was designed such that provisions can be given for future enhancement and maintenance without affecting the current developed system.

2.2.4 Review of Zoho Inventory System

In this project, a review of an existing Inventory Management System was conducted, which is Zoho Inventory System. It is a web-based inventory management system that is packed with complete functionalities an inventory management system should have.

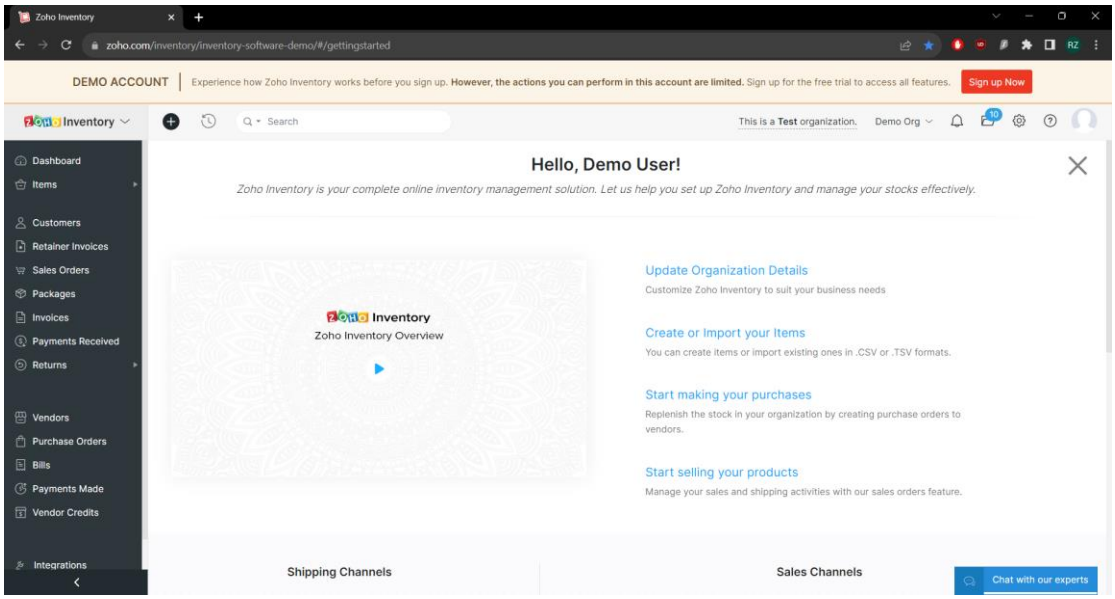


Figure 2 Zoho Main Page

Upon navigating to the Zoho website, there were a lot of functionalities in the main page. On the left side of the page is a panel with all the functions such as Dashboard, Items, Customers, Vendors and many more. On the top of the page, there were Quick Create icon, Recent Activities icon, a search bar, Notification icon, Settings icon, as well as profile.

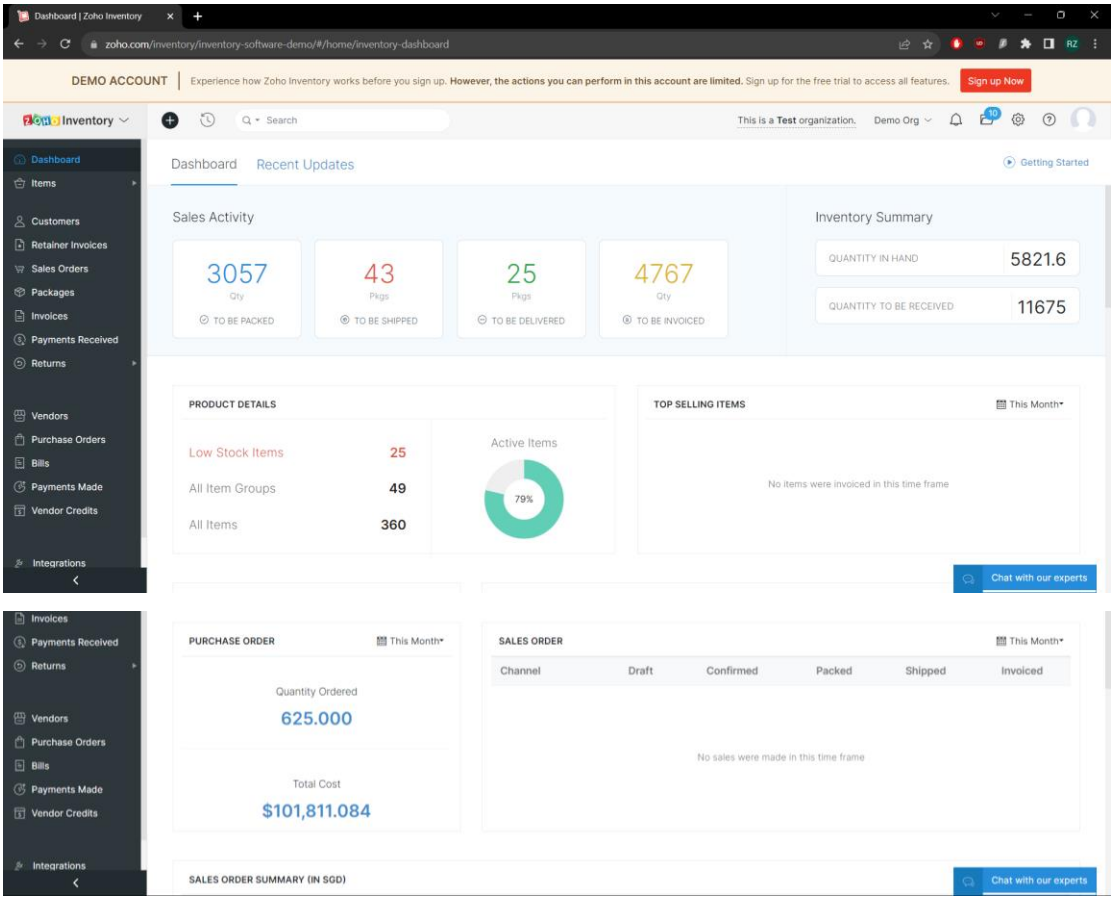


Figure 3 Zoho Dashboard

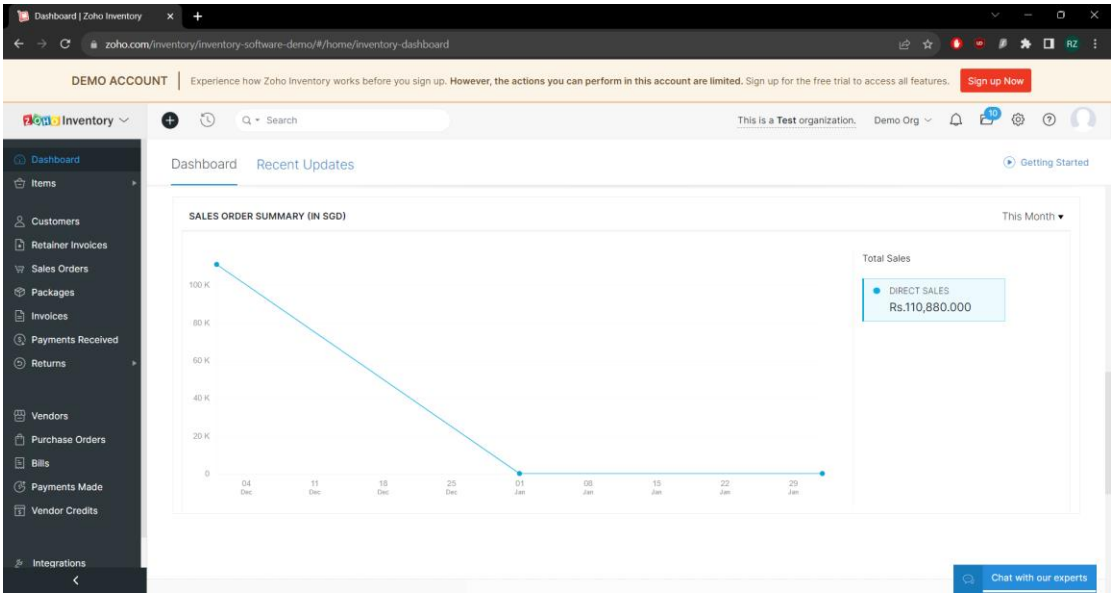


Figure 4 Zoho Dashboard (cont)

In the dashboard, there were many useful information to be shown such as Sales Activity which include quantity of items to be packed, shipped, delivered, and invoiced. It also shows the Inventory Summary for quantity of items in hand and quantity to be received. There are also Product Details, Top Selling Items, Purchase Order, Sales Order and Sales Order Summary. Overall, the dashboard contains many useful information for inventory management.

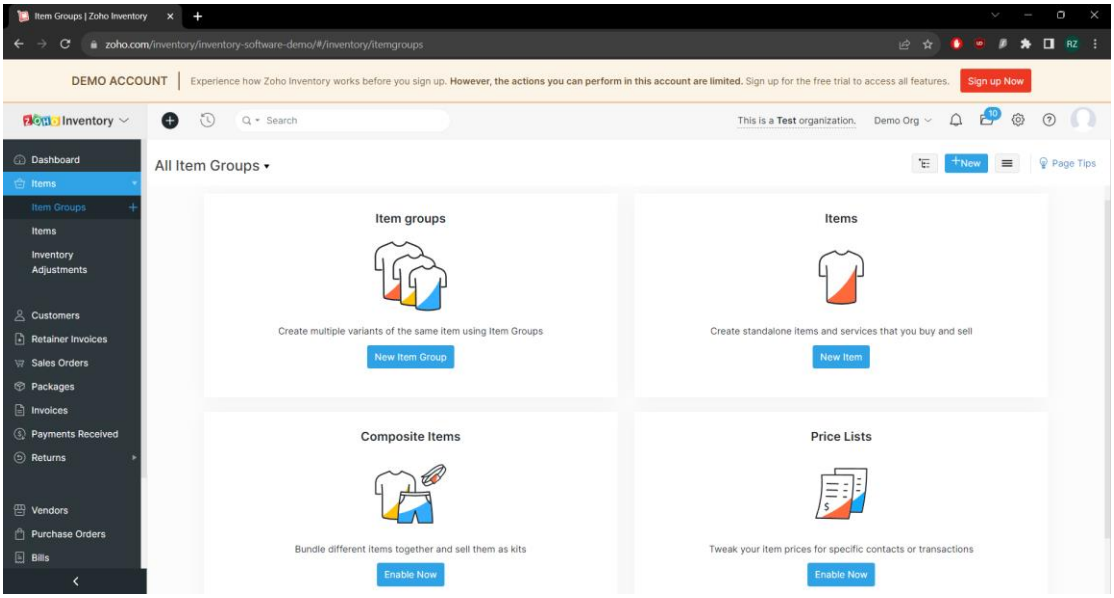


Figure 5 Zoho Items

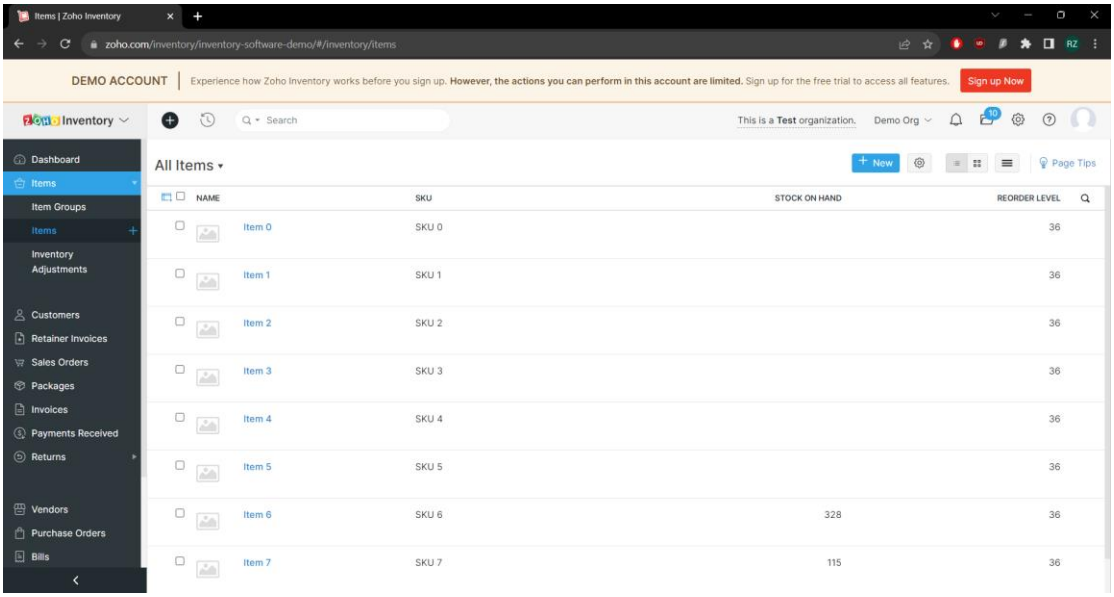


Figure 6 Zoho Items (cont)

In the Items, the users can create new item group and new item, as well as make inventory adjustments.

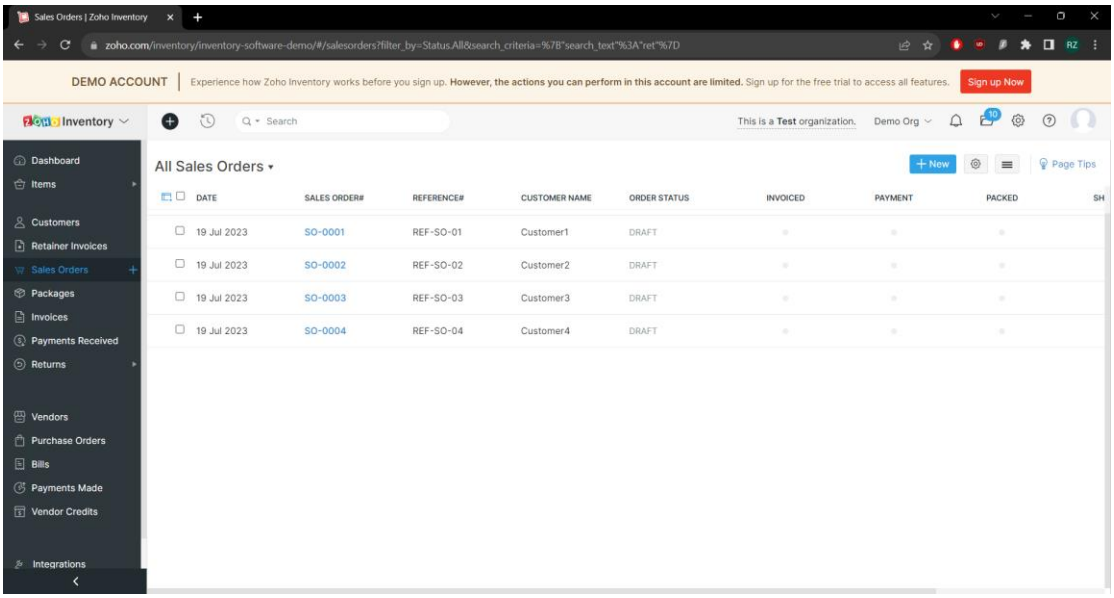


Figure 7 Zoho Sales Order

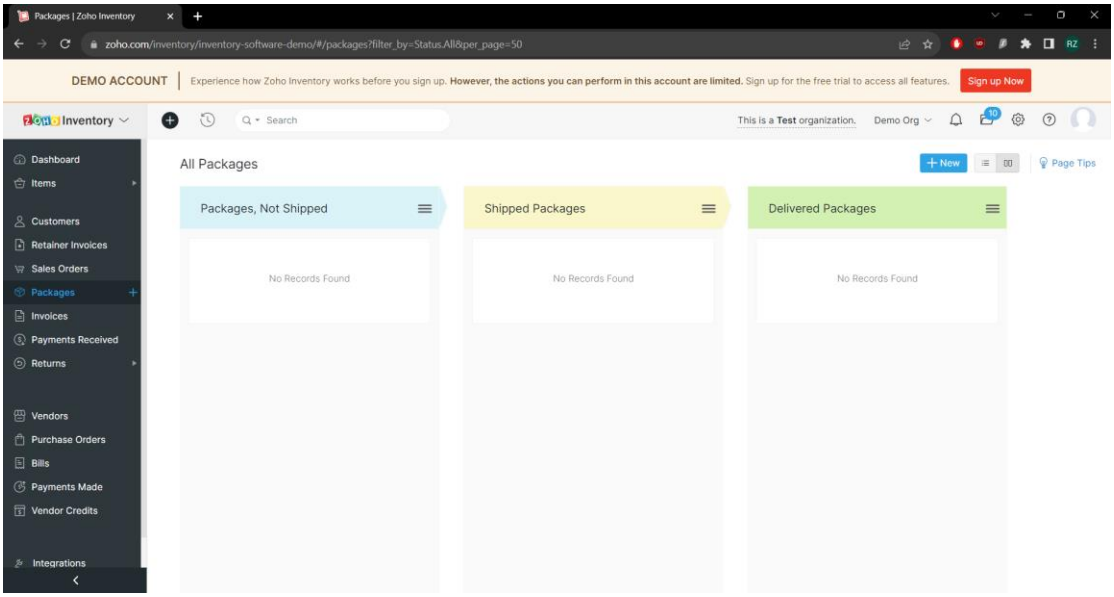


Figure 8 Zoho Packages

All the icons in the left panel allow the user to look into the details of the respective information. For example, in Sales Order, the information such as date, sales order number, reference number, customer name, order status and many more can be shown, sorted and monitored.

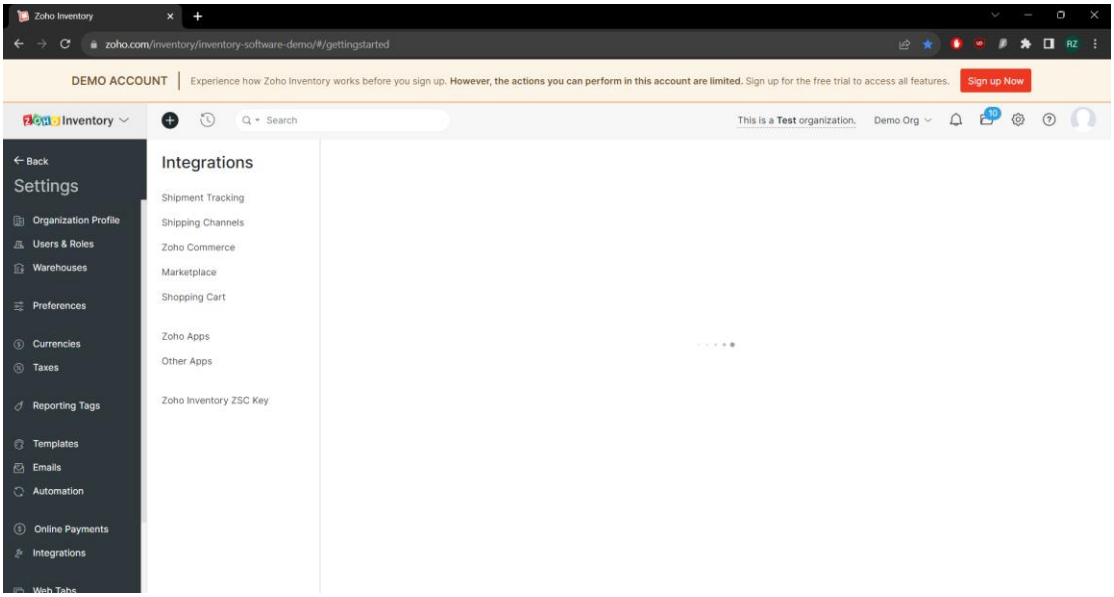


Figure 9 Zoho Integrations

In the Integrations page, the users can integrate functions such as Shipment Tracking, Marketplace into the existing inventory management.

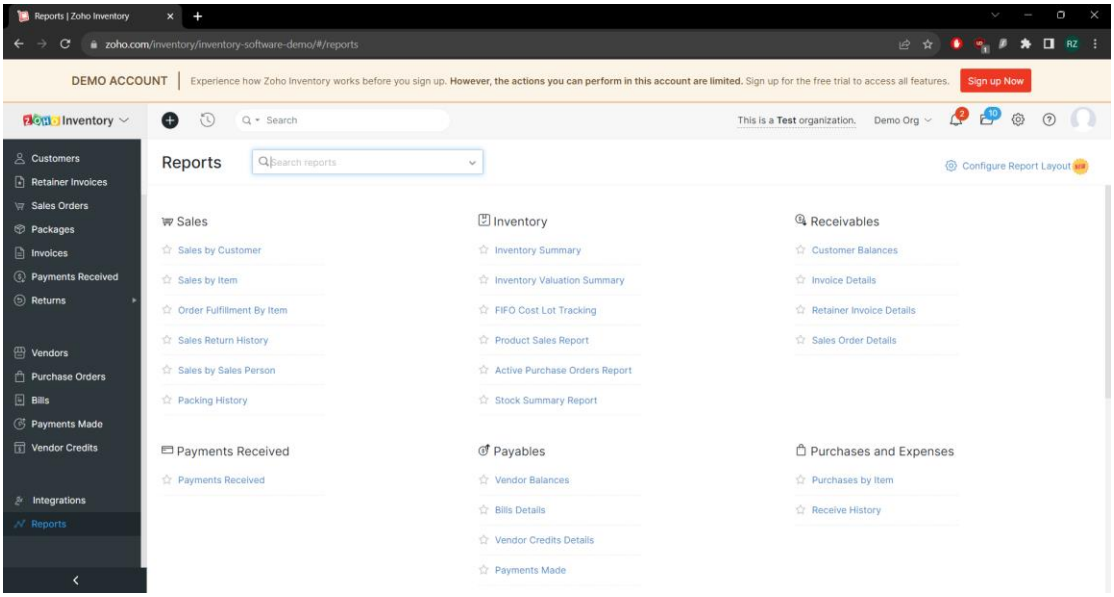


Figure 10 Zoho Reports

The reports are shown by categories in the Reports page.

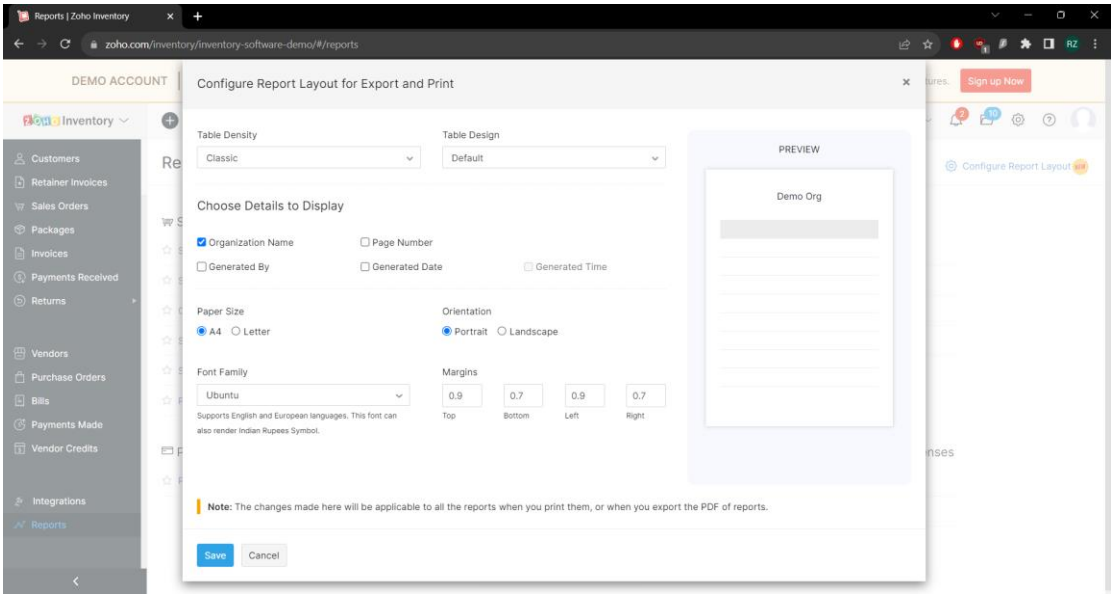


Figure 11 Zoho Configurations

The user can also configure the layout for the report page.

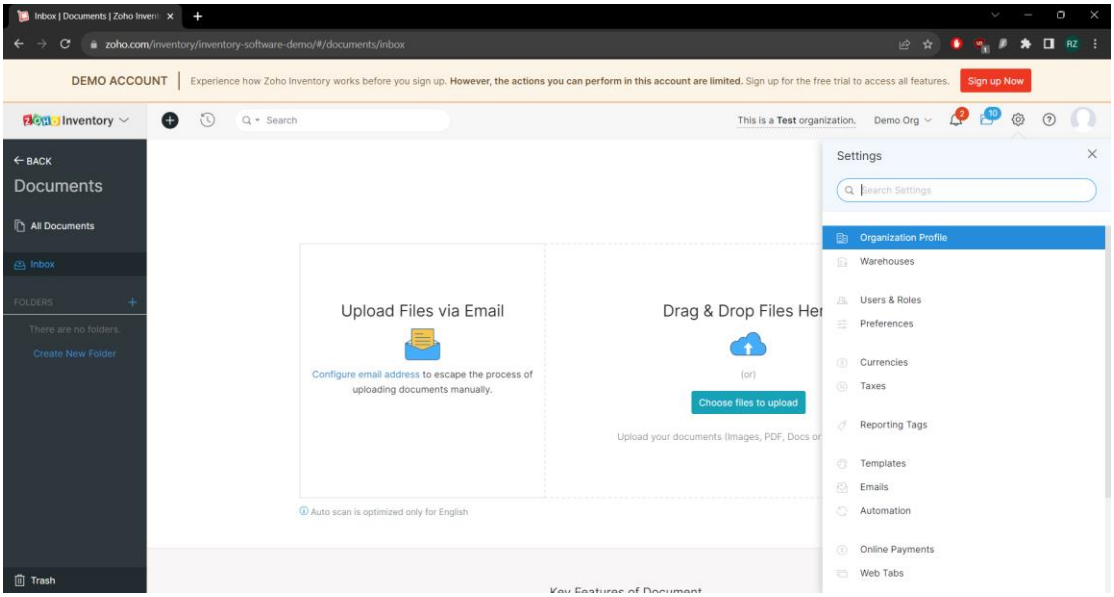


Figure 12 Zoho Settings

In the Settings, the user can configure the organisation profile, warehouses, preferences, change the currencies, monitor taxes, as well as back up the data.

The review of Zoho Inventory System was conducted using a free demo account. As shown as the figures above, the system is fully functional with complete functionality for a company to utilize it for inventory monitoring.

2.3 Limitation of Previous Studies

2.3.1 Web Portal and System for Management of Critical Reagent Inventory

The main limitations of this system are the implementation complexity, cost of the development and maintenance, data security, and dependency on third-party vendors.

The complexity of the implementation of the system is due to the fact that LCM+ is a customized software system, therefore there are many details and functional requirements that need meticulous design, implementation and maintenance. There is a need for a team of advanced users for the portal system maintenance and operational efficiency. These users should be equipped with proficient knowledge in the specific field as well as how the portal system works, instead of just IT professionals. Hiring those professionals from the fields also cost a lot of money. The development and maintenance of a customized inventory management system is also extremely expensive, particularly for smaller organisations or research institutions with very limited budget. Since external vendors need to be contracted for the major project developments, this makes the system reliant on third-party vendors. The need to contract external vendors for any major development further adds to the cost of the system.

2.3.2 Design and Implementation of both Online Inventory Monitoring System

The first system proposed by Dairo Ibukun Olatayo [4] contain the weaknesses of needing training to the staff due to reasons such as complexity of the inventory monitoring system. Providing training for the stuff members can be very resource-intensive in terms of time and cost. The system also incorporates new technology that may be unfamiliar to some users. Inventory monitoring systems often incorporate new technologies, such as specialized software or hardware. For example, a web portal and an android application were implemented in this system might be a technological barrier to staffs who are not familiar with technology. The staffs may also face a steep

learning curve when dealing with new technologies and result in reduced productivity during the initial stages of system implementation as they adapt to the new system.

The second system proposed by Sulihat Ibrahim-Imam [5] have the weaknesses of vulnerability to potential system crash. A system crash, whether due to software glitches, hardware failures or other issues can result in the loss of valuable inventory data. Besides, system crash can also cause operational disruption where the inventory management operations come to a halt. The disruption can have a significant impact on the business whether it be financial and productivity losses. The other weakness of the system is the potential loss of crucial business information due to accidents such as corrupted hard drive and power blackouts. These accidents can potentially wipe out the information of the business stored in the devices or causes abrupt shutdown of database servers' connection.

2.4 Summary of the existing systems

Existing system	Advantages	Limitations
2.3.1 Web Portal and System for Management of Critical Reagent Inventory	• Controlled user access • Optimized process workflows	• Implementation complexity • High development and maintenance cost • Dependency on third-party vendors
2.3.2 Design and Implementation of an Online Inventory Monitoring System	• Low hardware requirements • Records can be easily retrieved	• System complexity • Time-intensive • High cost
2.3.3 The Design and Implementation of an Online Inventory Monitoring System	• Flexible for future modifications	• Vulnerable to potential system crash
2.3.4 Review of Zoho Inventory System	• User-friendly interface, seamless integrations, real-time updates.	• Expensive maintenance • Complex

Table 2 Summary of existing systems

2.5 Proposed Solutions

This project aims to propose the solutions in order to resolve the aforementioned limitations found in the reviewed papers and system.

First, the Web Portal and System for Management of Critical Reagent Inventory has the weaknesses of requiring external vendors for major development due to system complexity and the cost of development and maintenance of the system. The proposed solution to overcome these weaknesses is to keep the system simple in terms of development and maintenance. The system should be developed by applying Agile methodology, where small developments are made periodically based on the requirements needed, instead of a big development at a time. By keeping the system simple, the developments can be done by the IT professionals in the organization instead of depending on external vendors for huge development.

The proposed solution addresses the weaknesses and limitations of current online inventory monitoring systems by focusing on simplicity and effectiveness. By building only essential functions and avoiding redundancy, it minimizes the need for staff training on unfamiliar technology.

CHAPTER 3 System Methodology

3.1 System Development Models

According to [6], the Software Development Life Cycle (SDLC) is a framework used in project management to outline the stages involved in developing an information system, from the initial feasibility study to maintaining the completed application. Various software development life cycle models define the specific stages and design followed during software development. These models, also known as "Software Development Process Models," each have their own set of phases tailored to their type to ensure successful software development steps. The development models reviewed in this project are waterfall model, Rapid Application Development (RAD) model, and Agile model.

3.1.1 Waterfall Model

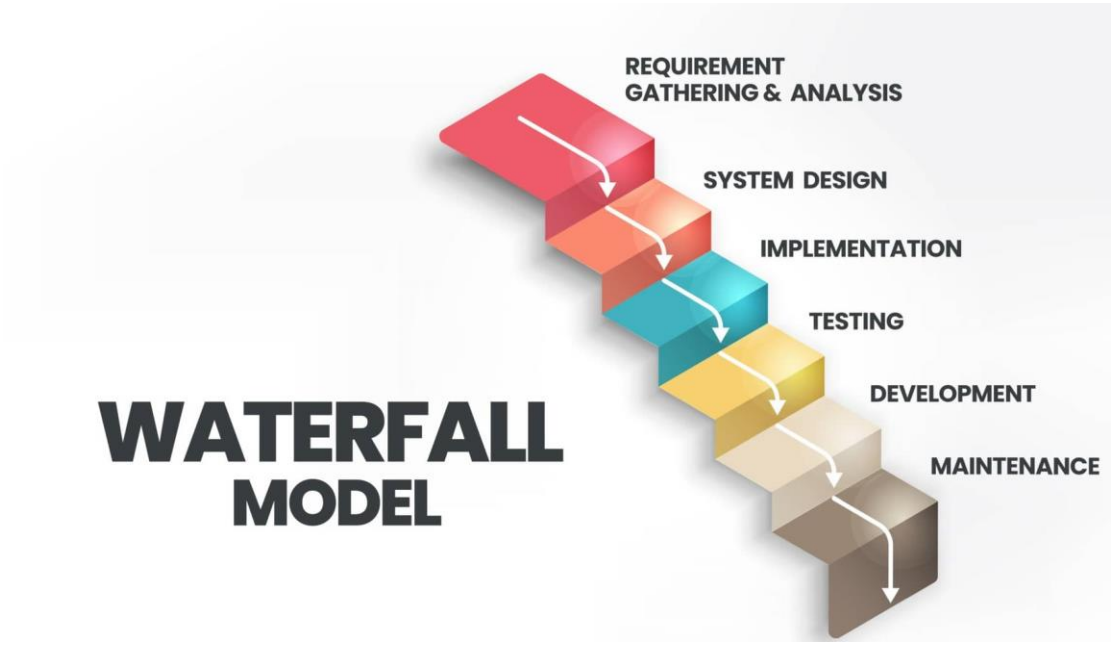


Figure 13 Waterfall Model

According to [7], waterfall methodology is a sequential approach to software development that follows a linear progression through defined phases, where each phase must be completed before moving to the next. The methodology consists of five distinct phases:

Requirements

This phase defines what the system should do and establishes project scope, team responsibilities, timeline, and process details.

Design

Designers create solutions that meet the established requirements, outlining project schedules, deliverables, and blueprints for software or physical products.

Implementation

Developers build the approved design, creating an implementation plan, allocating resources, and addressing any design challenges.

Verification

Quality assurance tests the developed software to ensure it meets all use case scenarios, identifies bugs, and establishes testing criteria.

Maintenance

After deployment, maintenance involves addressing customer feedback, bug fixes, and releasing updates.

Waterfall offers several benefits, including clear project structure, predictable costs and timelines, easier progress tracking, repeatable processes, comprehensive documentation, improved risk management, and enhanced accountability for team members.

However, Waterfall has limitations such as longer delivery times due to its rigid structure, limited flexibility for innovation or client feedback during development, potential for numerous feature requests post-launch, and delays caused by issues that require revisiting previous phases for resolution. These limitations make Waterfall less suitable for projects with unpredictable variables or evolving requirements.

3.1.2 Rapid Application Development Model (RAD)

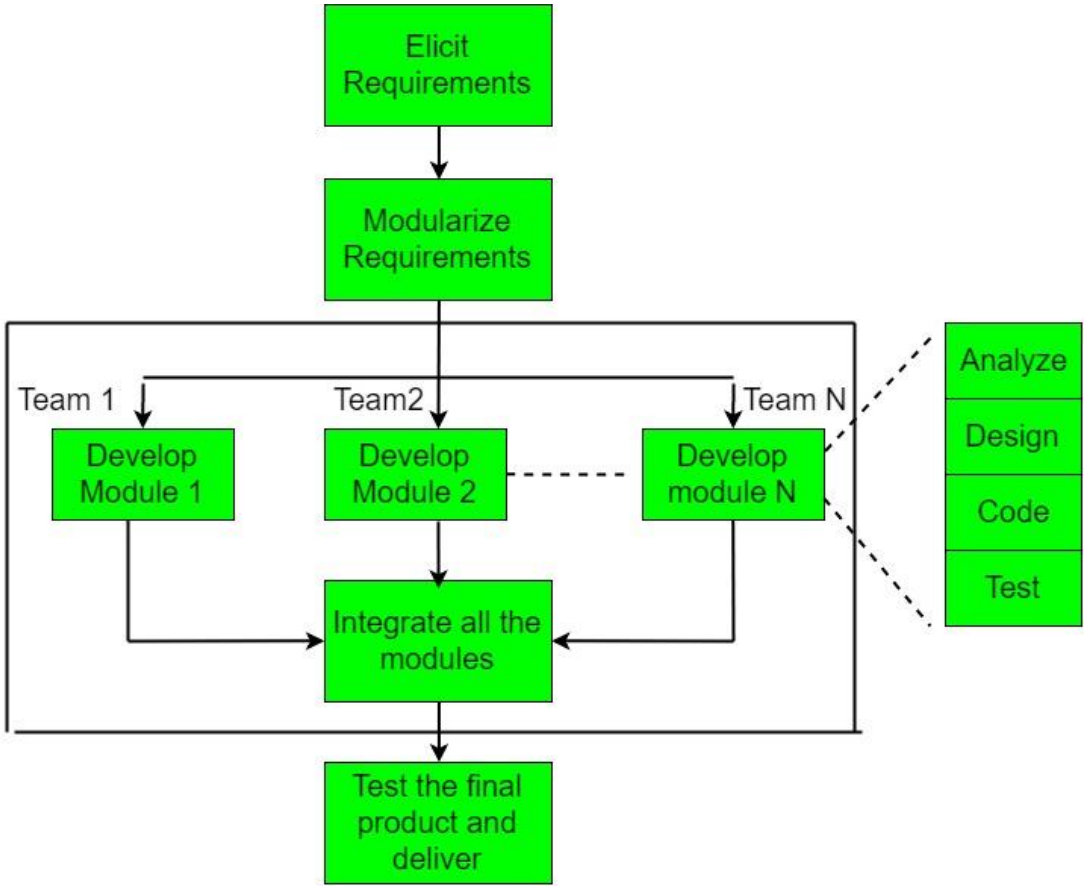


Figure 14 Rapid Application Development (RAD) Model

According to [8], RAD (Rapid Application Development) is an incremental process model designed for projects with well-understood requirements and a need for quick development cycles. It involves short development iterations and utilizes powerful development tools. The key phases of RAD include:

Requirements Gathering

Involves techniques like brainstorming and user scenarios to understand project scope and critical data.

User Description

Captures user feedback and builds prototypes using developer tools, refining data attributes.

Construction

Refinement of prototypes and delivery using automated tools, incorporating modifications and enhancements.

Cutover

Testing interfaces between modules developed by separate teams, followed by user acceptance testing.

RAD is suitable for projects with stable and transparent requirements, time-sensitive deliverables, and moderate project size. It thrives with high user involvement, tasks requiring creativity and innovation, prototyping needs, and projects with straightforward technological specifications.

RAD accelerates development with rapid prototyping, promotes adaptability to changing requirements, enhances stakeholder participation and interaction, ensures quality through early prototyping, and aims for customer satisfaction with frequent feedback.

RAD requires highly skilled professionals for efficient tool usage, demands close collaboration between team leaders, developers, and customers, may struggle with non-modularizable projects, and requires heavy resource allocation, particularly for automated tools. Additionally, RAD is less suited for small-scale projects due to cost concerns and may encounter challenges adopting new technologies.

3.1.3 Agile Model

Figure 15 Agile Methodology

According to [9], Agile methodology is a software development approach that breaks projects into multiple phases known as sprints, forming iterative development and continuous improvement. Development teams review on each sprint's outcomes to adjust strategies for subsequent iterations.

Agile methods offer adaptability, allowing teams to quickly adjust strategies without disrupting project flow, unlike the sequential nature of traditional waterfall methods. This flexibility is crucial in the fast-paced and adaptable field of software development. Additionally, Agile methodologies prioritize customer needs by enabling close collaboration with customers and rapid feedback incorporation, ensuring features align with evolving customer requirements.

Agile methodologies face challenges with large-scale projects requiring extensive stakeholder involvement and may lack comprehensive documentation for compliance.

Adoption of agile methodology is related to project complexity, scope, and organizational requirements to ensure suitability and mitigate potential drawbacks.

3.1.4 Methodology selected

The software development applied in the development of this project was Agile methodology. The several factors that influenced the decision of the methodology chosen were adaptability to changing requirements, as well as rapid prototyping and experimentation. The user requirements and priorities in this project may evolve over time, and Agile's iterative approach allows for flexibility such that it can accommodate changes in functionalities and features based on the evolving requirements. As the development of this project went on, it required rapid prototyping and experimentation in order to enable quick validation of concepts and ideas. Agile's iterative approach was fitting for figuring out the most suitable design and refine the existing interface based on the requirement changes.

Plan

The first phase of the Agile methodology is conducting the initial planning of the project, including defining user stories, basic functional requirements, target users and the system environment. The target users of this project are students, staffs, and supervisor. An inventory monitoring web portal should contain the functions to add new inventory, update inventory, remove existing inventory, generate borrow request, and generate request approval.

Design

In the design phase, a system block diagram and a use case diagram were drawn to visualize the overall architecture of the inventory monitoring web portal. The system block diagram was drawn to visualize the flow of the functions and how these functions interact with each other. Besides, the use case diagram was drawn to define the user requirements.

Develop and Test

In the development and testing phases, the features and functionalities were implemented in Node-RED based on the defined requirements. The user stories defined in planning phase were broken down into development tasks and configurations of components. Then, the developed functions were tested to determine if it matches the functionalities as intended.

Deploy

After the determined functionalities were completed, the developed features were deployed. The performance and consistency of the functionalities were monitored during the deployment.

Review

The iteration was reviewed to gather insights and identify areas for improvement. The issues faced during the development and testing were recorded for future references to avoid making the same error.

3.2 Functional Requirement

Before starting the project, the basic functional requirements, target users and the scenario in which the system will be deployed and used are listed out first. In this project, the target users are students, staffs and supervisor. The user stories of the inventory monitoring web portal were: -

1) The staffs will be able to add new inventory to the system.
2) The staffs will be able to update existing inventory in the system.
3) The staffs will be able to remove an existing inventory from the system.
4) The staffs will be able to register the borrow of the item in the system.
5) The staffs will be able to register the return of the item in the system.
6) The students will be able to request to borrow an item.
7) The students will be able to sign up for a new account.
8) The supervisor will be able to approve the borrow request.
9) The supervisor will be able to deny the borrow request.
10) The supervisor will be able to approve the new student account registration.
11) The supervisor will be able to deny the new student account registration.
12) The students, staffs, and supervisors will be able to log into their account.
13) The students, staffs, and supervisors will be able to log out of their account.
14) The system will be able to generate a borrow request to the supervisor.
15) The system will be able to store the requests.
16) The system will be able to store the user information.
17) The system will be able to update the inventory.
18) The system will be able to validate the user information for login.
19) The system will be able to encrypt the password.
20) The system will be able to decrypt the password.

After the target users and the basic functional requirements are listed out, the scenario in which the system will be used is created. In this project, the scenario in which the inventory monitoring web portal is deployed is in a university laboratory.

3.3 Project Milestone

Task	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
Review and Revise Previous Work														
Revise the Functional Requirements														
Implement User Login Mechanism														
Revise and Improve the Whole System														
System testing and Error Checking														
Report Writing														
FYP2 Presentation														

Figure 16 Timeline of Final Year Project 2

The plan for Final Year Project 2 was to implement the User Login Mechanism. Before starting the implementation of the Login Mechanism, the previous work done was revised and reviewed and check for missing functional requirements. Roughly 5 weeks were spent on implementing the User Login Mechanism from scratch. Afterwards, the whole system was revised to ensure that it matches the functional requirements stated previously. Then, 2 weeks were spent to test the whole system and check for any errors or bugs to ensure that the system is functioning as intended. After making sure the system was functional, preparations were made to move on to report writing. The development process was on track with the timeline.

CHAPTER 4 System Design

4.1 System Architecture

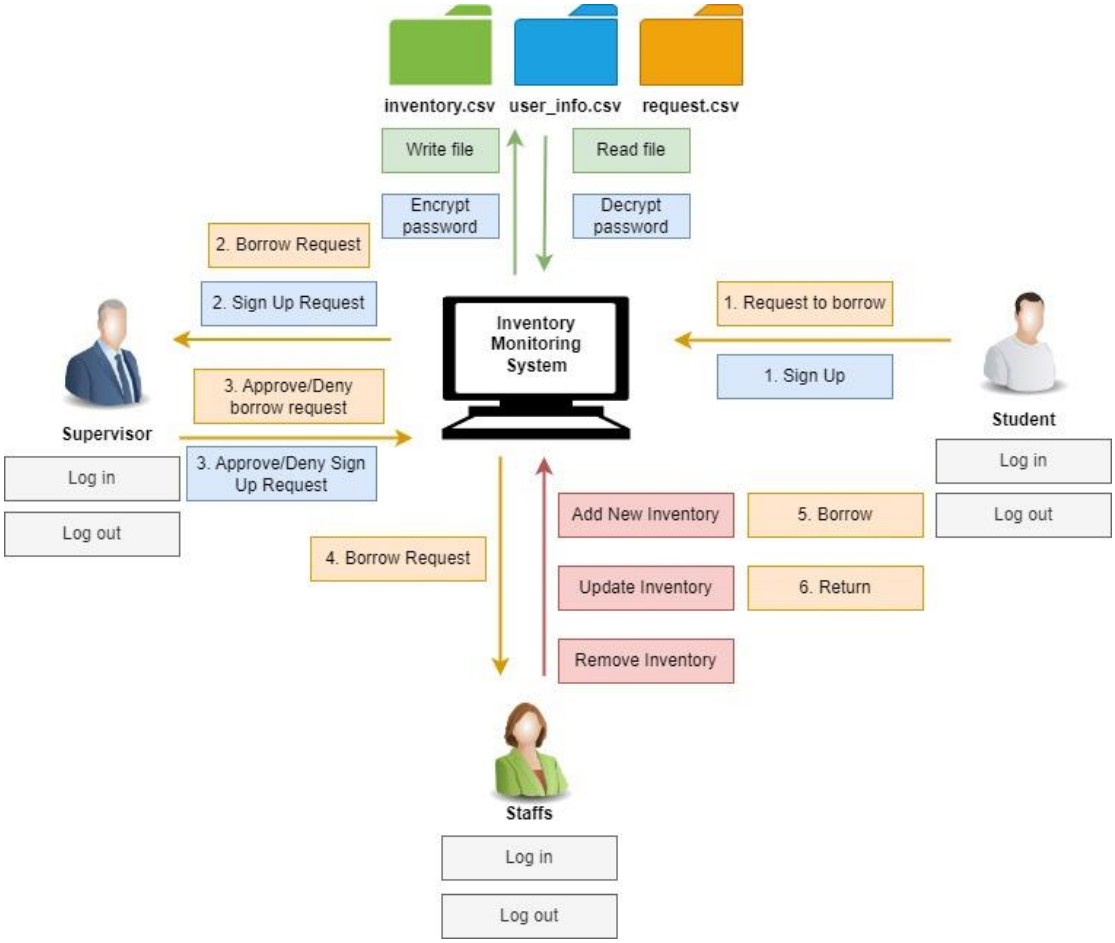


Figure 17 Design Block Diagram of Inventory Monitoring Web Portal

The block diagram above shows the complete flow of the inventory monitoring web portal. The orange blocks represent the flow of the inventory borrow and return. The green blocks represent the file handling, mainly reading and writing. The blue blocks represent the flow of student sign up and new account approval. The red blocks represent the flow of staffs handling the inventory directly. The grey blocks represent the login and logout of the users.

New students can sign up for new account by clicking the Sign Up button that leads to the sign up page, where the student is required to fill in the new username and password. The sign up request will be sent to all the supervisors, to which the respective

supervisor the student is under should approve the sign up request, then the system will automatically assign the supervisor name to the student in the file. The system will encrypt the password and store it in cipher text.

Before any of the users can access the inventory monitoring interface, the users must log in to their account before they were navigated to their respective interface. For example, staff log in leads to staff inventory monitoring page, while student log in leads to student's borrow page. Any of the user can not access the other users' interface. The users can also log out and the system resets back to the login page.

The student can request to borrow an item from the inventory list. The borrow request will be sent to the respective supervisor for approval. The supervisor can either approve or deny the borrow request. If the borrow request was approved, the lab staffs will be updated then they will borrow the item requested to the students. Once the student returned the item, the lab staff can also register the item as returned in the system, in which the system will update the inventory count automatically.

The staffs can add new inventory, update existing inventory and remove an inventory from the inventory list. The respective supervisor will only view the requests made by the students they are supervising. For example, Dr Lee can only view requests made by James and Jun Jie, both whom are under the supervision of Dr Lee.

Whenever any of the flows are initiated, the system first reads the file from the respective file. For instance, before a student can request to borrow an item on the system, the system first reads the user_info.csv, inventory.csv to validate the student's account and display the inventory information to the student. By the end of each flow, the system will write to the file.

4.2 Functional Modules in the System

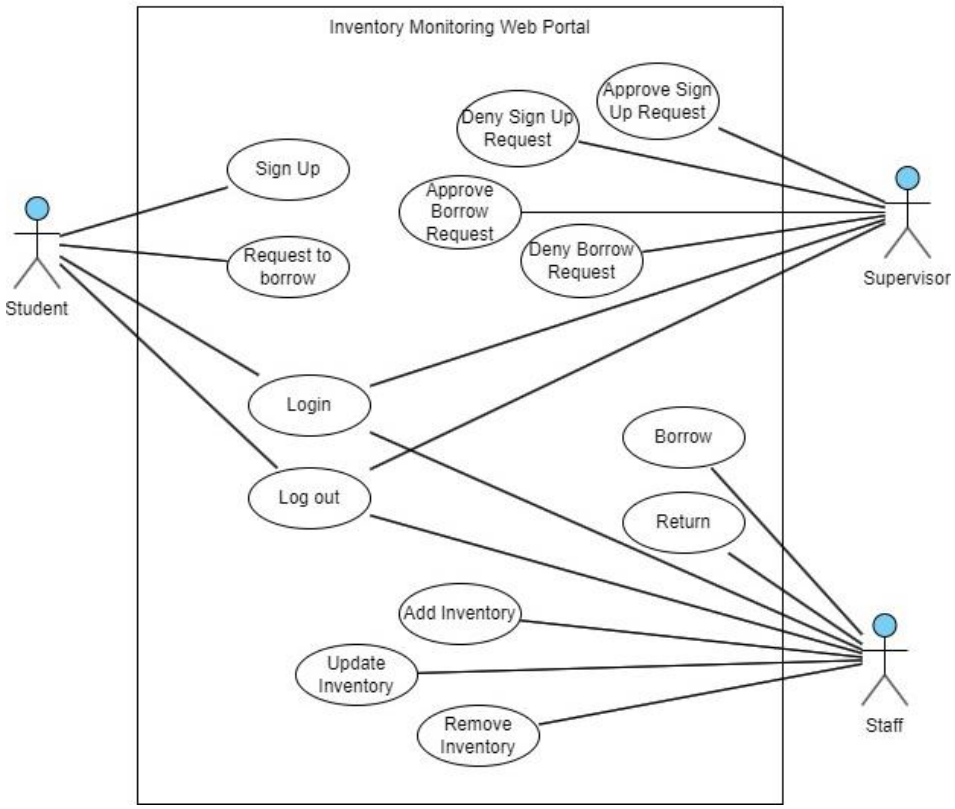


Figure 18 Use Case Diagram of the Inventory Monitoring Web Portal

The use case diagram [10] above shows the use cases for each of the actors in the test case. The actors in this use case are students, staffs, and supervisor. Student can sign up for a new account. Furthermore, student can request to borrow an item from the inventory. Supervisor can approve or deny sign up requests and borrow request generated by students. Next, staff can add new inventory, update existing inventory, remove an inventory. Other than that, staff can register the borrow and return of an item in the system. All users can log in and log out from the system.

Table 3 Sign Up Use Case

Use Case ID	UC001	Version	1.0
Use Case Name		Sign Up	
Purpose		To create a new account to log into the inventory monitoring web portal	
Actor		Student	
Trigger		Student clicks on “Sign up”	
Precondition		Student at login page	
Scenario Name	Step	Action	
Main Flow	1	Student clicks on “Sign Up”.	
	2	System prompts student to key in new username and password	
	3	Student keys in username and password	
	4	Student clicks on “Sign Up” button to create account.	
	5	System validates the user input.	
	6	System retrieves the data and append to user_info.csv	
Alternate Flow (Existing username)	5a.1	Student enters an existing username	
	5a.2	System validates the user input.	
	5a.3	System displays error message.	
Rules		-	
Author		Tan Rui Zheng	

Table 4 Login Use Case

Use Case ID	UC002	Version	1.0
Use Case Name		Login	
Purpose		To validate username and password and redirect user to their respective page based on their role	
Actor		Student, Staff, Supervisor (All user)	
Trigger		User starts the system	
Precondition		The system has been started successfully	
Scenario Name	Step	Action	
Main Flow	1	User starts the system.	
	2	System prompts user to key in their username and password	
	3	User keys in username and password	
	4	User clicks on "Login" button to create account.	
	5	System validates the username and password.	
	6	Login successful, system redirects user to their respective page.	
Alternate Flow (Invalid username or password)	5a.1	User enters an invalid username or password.	
	5a.2	System validates the username and password.	
	5a.3	System displays error message.	
Rules		-	
Author		Tan Rui Zheng	

Table 5 Logout Case

Use Case ID	UC003	Version	1.0
Use Case Name		Logout	
Purpose		To allow user to exit the page	
Actor		Student, Staff, Supervisor (All user)	
Trigger		User clicks “Log out” button	
Precondition		User at user page	
Scenario Name	Step	Action	
Main Flow	1	User clicks “Log Out”	
	2	System redirects user back to the Login Page	
Rules		-	
Author		Tan Rui Zheng	

Table 6 Student Borrow Item Use Case

Use Case ID	UC004	Version	1.0
Use Case Name		Borrow Item	
Purpose		To generate a request to borrow the item	
Actor		Student	
Trigger		Student selects an item	
Precondition		Student at student interface	
Scenario Name	Step	Action	
Main Flow	1	Student clicks on an item in the inventory list.	
	2	System prompts student to key in borrow amount.	
	3	Student keys in borrow amount.	
	4	Student clicks on “Borrow” button to generate borrow request.	
	5	System validates the borrow amount.	
	6	System retrieves the data and append to request.csv	
Alternate Flow (Existing username)	5a.1	Student enters a borrow amount exceeding existing inventory quantity.	
	5a.2	System validates the borrow amount.	
	5a.3	System displays error message.	
Rules		-	
Author		Tan Rui Zheng	

Table 7 Supervisor Approve Sign Up Request Use Case

Use Case ID	UC005	Version	1.0
Use Case Name		Approve Sign Up Request	
Purpose		To approve sign up request.	
Actor		Supervisor	
Trigger		Supervisor selects a sign up request.	
Precondition		Supervisor at supervisor page.	
Scenario Name	Step	Action	
Main Flow	1	Supervisor clicks on a sign up request in the request list.	
	2	System prompts supervisor to click “Approve” or “Deny” button.	
	3	Supervisor clicks “Approve” button	
	4	System assigns role, supervisor and appends the username, password, role, supervisor in user_info.csv	
Rules		-	
Author		Tan Rui Zheng	

Table 8 Supervisor Deny Sign Up Request Use Case

Use Case ID	UC006	Version	1.0
Use Case Name		Deny Sign Up Request	
Purpose		To deny sign up request.	
Actor		Supervisor	
Trigger		Supervisor selects a sign up request.	
Precondition		Supervisor at supervisor page.	
Scenario Name	Step	Action	
Main Flow	1	Supervisor clicks on a sign up request in the request list.	
	2	System prompts supervisor to click “Approve” or “Deny” button.	
	3	Supervisor clicks “Deny” button	
	4	System removes the sign up request from request.csv	
Rules		-	
Author		Tan Rui Zheng	

Table 9 Supervisor Approve Borrow Request Use Case

Use Case ID	UC007	Version	1.0
Use Case Name		Approve Borrow Request	
Purpose		To approve borrow request generated by student.	
Actor		Supervisor	
Trigger		Supervisor selects a borrow request.	
Precondition		Supervisor at supervisor page.	
Scenario Name	Step	Action	
Main Flow	1	Supervisor clicks on a borrow request in the request list.	
	2	System prompts supervisor to click “Approve” or “Deny” button.	
	3	Supervisor clicks “Approve” button	
	4	System updates the borrow request to “approved” and generate approve timestamp in request.csv	
	5	System updates the student’s user status to approved in user_info.csv.	
	6	System passes the updated borrow request to staff.	
Rules		-	
Author		Tan Rui Zheng	

Table 10 Supervisor Deny Borrow Request Use Case

Use Case ID	UC008	Version	1.0
Use Case Name		Deny Borrow Request	
Purpose		To deny borrow request generated by student.	
Actor		Supervisor	
Trigger		Supervisor selects a borrow request.	
Precondition		Supervisor at supervisor page.	
Scenario Name	Step	Action	
Main Flow	1	Supervisor clicks on a borrow request in the request list.	
	2	System prompts supervisor to click “Approve” or “Deny” button.	
	3	Supervisor clicks “Deny” button	
	4	System removes the borrow request from request.csv.	
	5	System updates the student’s user status to “none” in user_info.csv	
Rules		-	
Author		Tan Rui Zheng	

Table 11 Staff Generate Borrow Item Use Case

Use Case ID	UC009	Version	1.0
Use Case Name		Generate Borrow Item	
Purpose		To generate the borrow of an item in the system.	
Actor		Staff	
Trigger		Staff selects a borrow request.	
Precondition		Staff at staff page.	
Scenario Name	Step	Action	
Main Flow	1	Staff clicks on a borrow request in the borrow request list.	
	2	Staff clicks “Borrow” button	
	3	System removes the borrow request from the borrow list in the staff interface.	
	4	System updates the borrow request status to “borrowing” and update the borrow timestamp in the request.csv	
	5	System updates the student’s user status to “borrowing” and adds one to the borrowed column in the user_info.csv	
	6	System updates the inventory quantity by subtracting the borrow amount.	
	7	System appends the borrow request to the return list in the staff interface.	
Rules		-	
Author		Tan Rui Zheng	

Table 12 Staff Generate Return Item Use Case

Use Case ID	UC010	Version	1.0
Use Case Name		Return Item	
Purpose		To generate the return of an item in the system.	
Actor		Staff	
Trigger		Staff selects a return request.	
Precondition		Staff at staff page.	
Scenario Name	Step	Action	
Main Flow	1	Staff clicks on a return request in the borrow request list.	
	2	Staff clicks “Return” button	
	3	System removes the return request from the return list in the staff interface.	
	4	System updates the borrow request status to “returned” and update the return timestamp and duration in the request.csv	
	5	System subtracts the borrowed column value by 1 in the user_info.csv	
	6	System updates the inventory quantity by adding the borrow amount.	
Alternate Flow (borrowed column is 0 after subtraction)	5a.1	System subtracts the borrowed column value by 1 in the user_info.csv.	
	5a.2	The borrowed column value equals to 0.	
	5a.3	System updates the student’s user status to “none”	
Rules		-	
Author		Tan Rui Zheng	

Table 13 Staff Add New Inventory Use Case

Use Case ID	UC011	Version	1.0
Use Case Name		Add New Inventory	
Purpose		To add new item into the inventory list	
Actor		Staff	
Trigger		Staff clicks on the "Add New Inventory" button.	
Precondition		Staff at staff page.	
Scenario Name	Step	Action	
Main Flow	1	Staff clicks on the "Add New Inventory" button	
	2	System prompts staff to key in the item information.	
	3	Staff keys in the item name and quantity.	
	4	Staff clicks "Enter" button.	
	5	System validates the input item name and quantity.	
	6	System appends the new item name and quantity into inventory.csv	
	7	System displays the new item on the staff interface inventory list.	
Alternate Flow (Item already exist)	5a.1	System validates the input item name and quantity.	
	5a.2	The input item name already exist in the inventory list.	
	5a.3	System displays error message.	
Rules		-	
Author		Tan Rui Zheng	

Table 14 Staff Update Inventory Use Case

Use Case ID	UC012	Version	1.0
Use Case Name		Update Inventory	
Purpose		To update an item in the inventory list	
Actor		Staff	
Trigger		Staff clicks on the “Update Item” button.	
Precondition		Staff selects an item from the inventory list.	
Scenario Name	Step	Action	
Main Flow	1	Staff clicks on the “Update Item” button.	
	2	System prompts staff to key in the item information.	
	3	Staff keys in the item name and quantity.	
	4	Staff clicks “Enter” button.	
	5	System updates the item name and quantity in inventory.csv	
	6	System displays the updated item on the staff interface inventory list.	
Rules		-	
Author		Tan Rui Zheng	

Table 15 Staff Remove Inventory Use Case

Use Case ID	UC013	Version	1.0
Use Case Name		Remove Inventory	
Purpose		To remove an item from the inventory list	
Actor		Staff	
Trigger		Staff clicks on the “Remove Item” button.	
Precondition		Staff selects an item from the inventory list.	
Scenario Name	Step	Action	
Main Flow	1	Staff clicks on the “Delete Inventory” button.	
	2	System prompts “Remove Confirmation” and “Cancel” button.	
	3	Staff clicks on the “Confirm” button.	
	4	System removes the selected item from the inventory list.	
	5	System removes the selected item from the displayed inventory list.	
Alternate Flow (Staff cancel the deletion)	2a.1	System prompts “Remove Confirmation” and “Cancel” button.	
	2a.2	Staff clicks on the “Cancel” button.	
Rules		-	
Author		Tan Rui Zheng	

4.3 System Flow

4.3.1 Flowchart

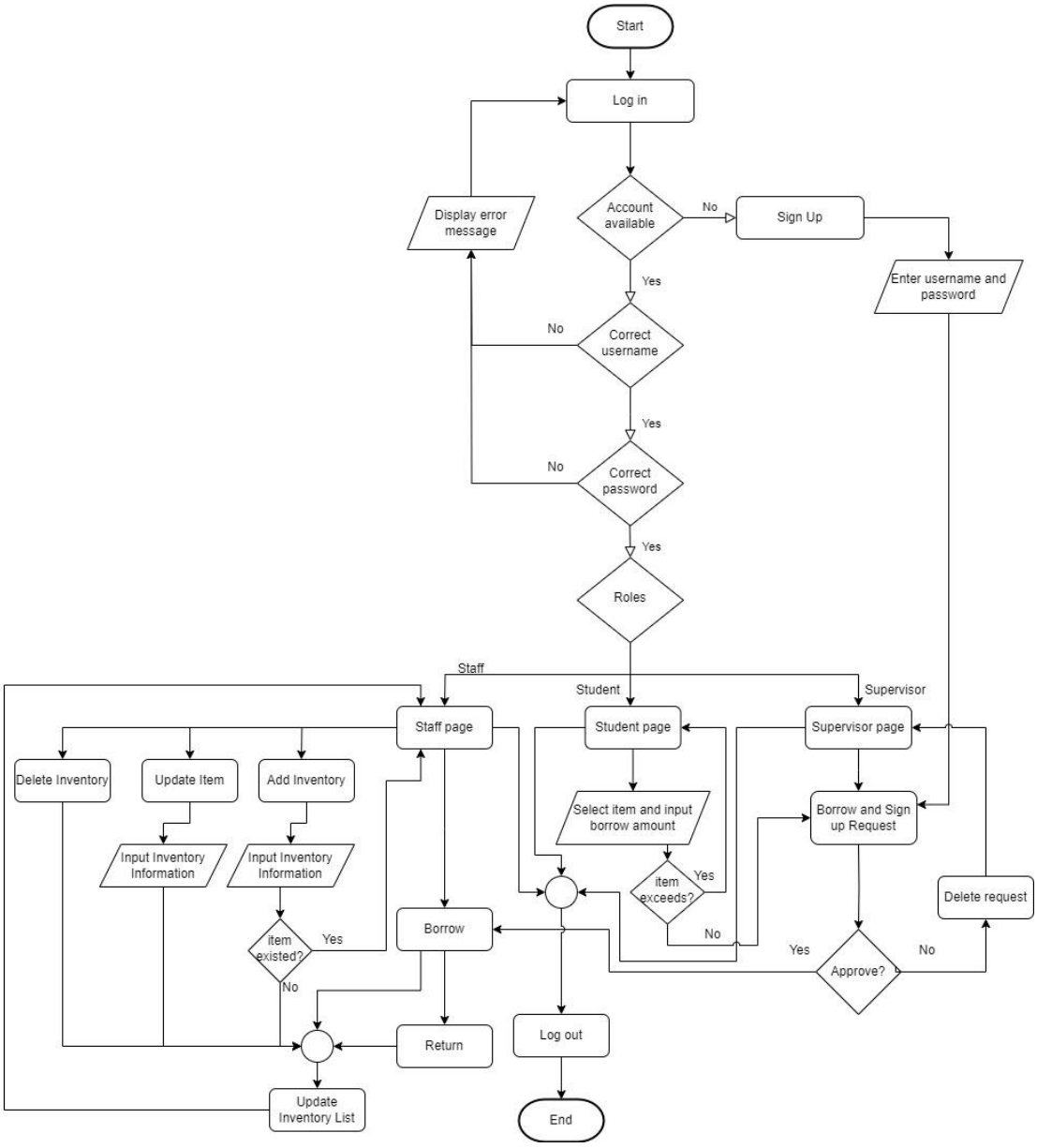


Figure 19 Flowchart of the inventory monitoring web portal

When the user starts the system, the system redirects the user to the login page. If the user does not have an account, the user can choose to sign up for an account, which the user will be required to enter username and password for their account. The sign up request will be sent to the supervisor for approval. If the user has an existing account upon login, they can key in their username and password. Then, the system

will authenticate the user's username and password. If either the username does not exist, or the password is incorrect or both, the system will display error message and refresh the login page. If the username exists and the password is correct, the system will read the role of the user is student, staff or supervisor, where the user will be redirected to their respective page.

The staff can choose to add inventory or update inventory, the system prompts the user to enter the item name and quantity. If the staff adds inventory, the system will check if the item already exists in the inventory list. If the item already exists in the inventory, the system will redirect the staff back to the staff page. If the staff updates inventory, delete inventory or the added item did not exist in the inventory, the system will update the inventory list and redirect back to the staff page.

Student can select an item and borrow, which the system will prompts the student to input borrow amount. The system will validate if the borrow amount exceeds the existing quantity. If true, then the student will be redirected back to the student page. If borrow amount did not exceed existing quantity, the system will proceed to generate the borrow request to the supervisor.

Supervisor can approve or deny borrow request, as well as approve or deny sign up request generated by students. If denied, the request will be deleted. If approved, the borrow request will be sent to the staff. The staff can register the requested item as borrowed, which the inventory list will be updated. After borrowing, a return request will be generated, and the staff can register the item as returned.

All the users can log out in any part of the system.

4.3.2 Activity Diagrams

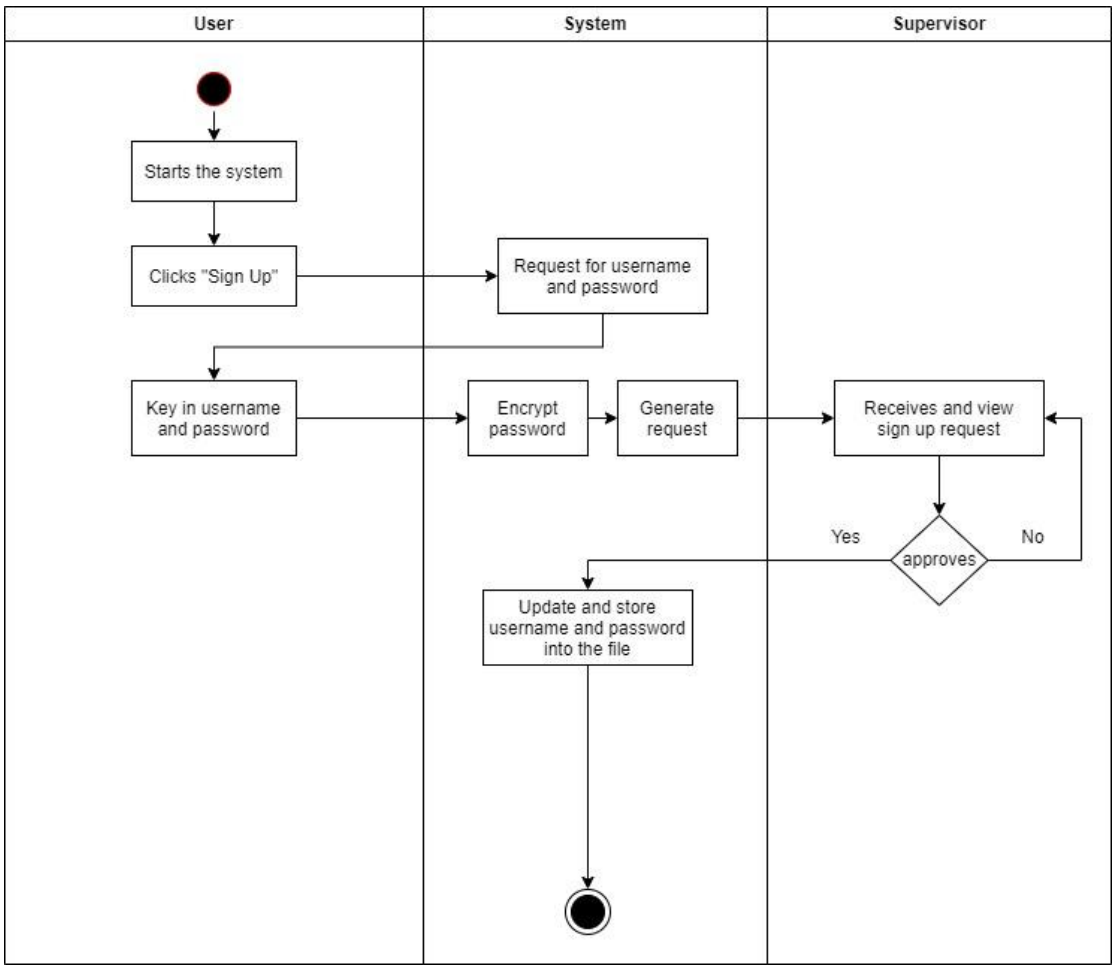


Figure 20 Activity Diagram of Sign Up Use Case

The figure above shows the activity diagram for the Sign Up use case. The user first starts the system and clicks “Sign Up”, the system will request for username and password. After the user key in the username and password, the system will encrypt the password and generate sign up request to the supervisor. The supervisor approves or denies the sign up request. If approved, the system updates and store the username and password registered into user_info.csv.

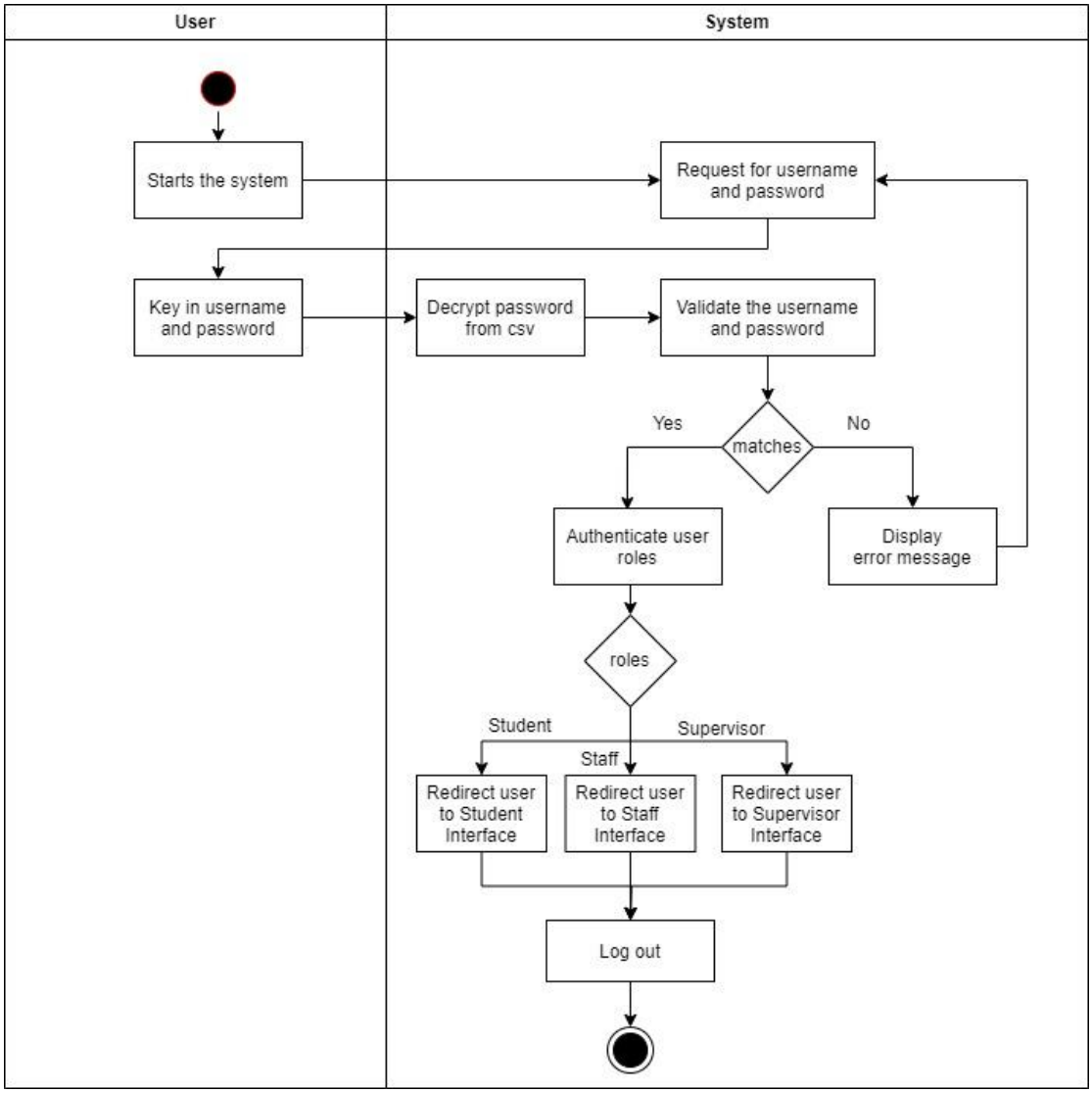


Figure 21 Activity Diagram of Login Use Case

The figure above shows the activity diagram for the Login use case. The user starts the system and the system requests the user to key in the username and password. After the user inputs the information, the system decrypts the password from user_info.csv and validate the username and password. If it does not match, the system displays an error message and redirects back to request user input. If it matches, the system will authenticate the role of the user. If student, redirect user to student interface. If staff, redirect staff to staff interface. If supervisor, redirect user to supervisor interface. The user can log out in the interface.

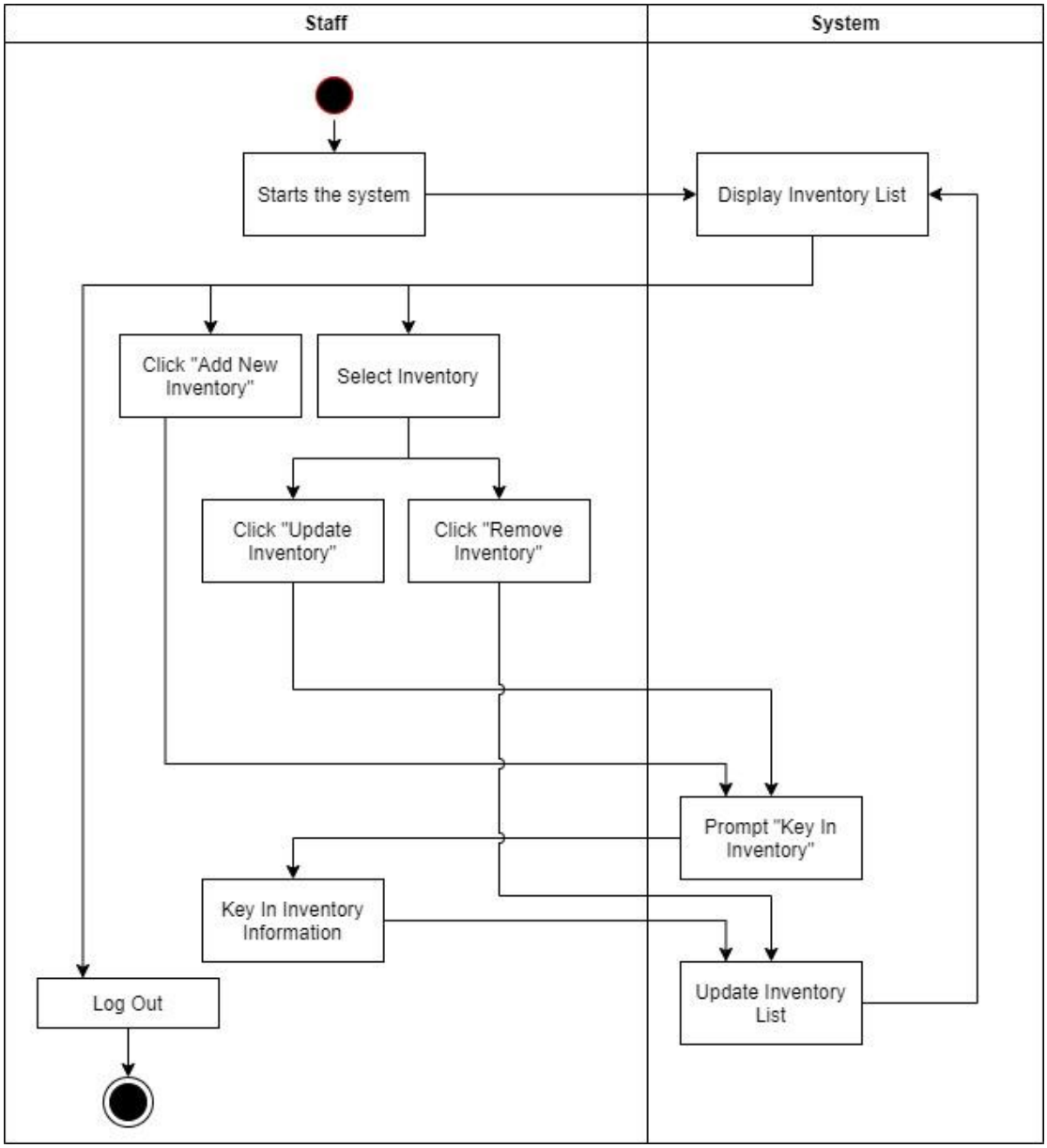


Figure 22 Activity Diagram of Staff Inventory Management Use Cases

The figure above shows the activity diagram for the Staff Inventory Management use cases. The system displays the inventory list. The staff can either add new inventory, or select inventory to be updated or removed. Add and update inventory will prompt the user to key in the inventory to be added or updated. After the user key in the item name and quantity, the system will update the inventory list. Staff removing the inventory also updates the inventory immediately. After updating the inventory list, the system redirects staff to the inventory list. The staff can log out at any point in the flow.

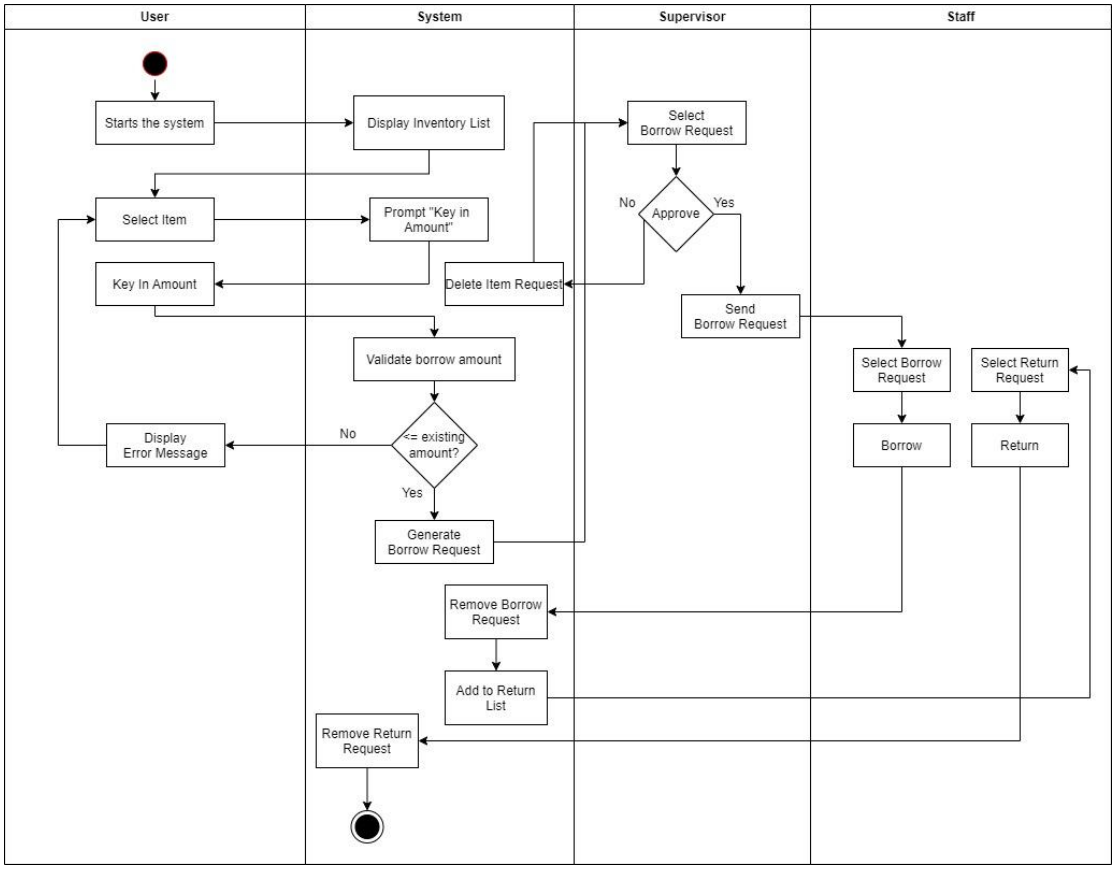


Figure 23 Activity Diagram of Borrow and Return Use Cases

The figure above shows the activity diagram for the Borrow and Return use cases. First, the system displays inventory list to the student, then the student select the item to borrow. The system prompts the student to key in the borrow amount and validates the borrow amount. If the borrow amount exceeds existing item quantity, the system will display error message and redirect back to the student interface. If the borrow amount is less than existing item quantity, then the system generates borrow request to the supervisor for approval. If denied, the borrow request will be deleted and redirect back to the supervisor interface. If approved, the borrow request will be sent to staff. The staff can register the item as borrowed, then the borrow request will be removed, and added to the return list. Then the staff can register the item as returned, then remove the return request.

4.4 GUI Design

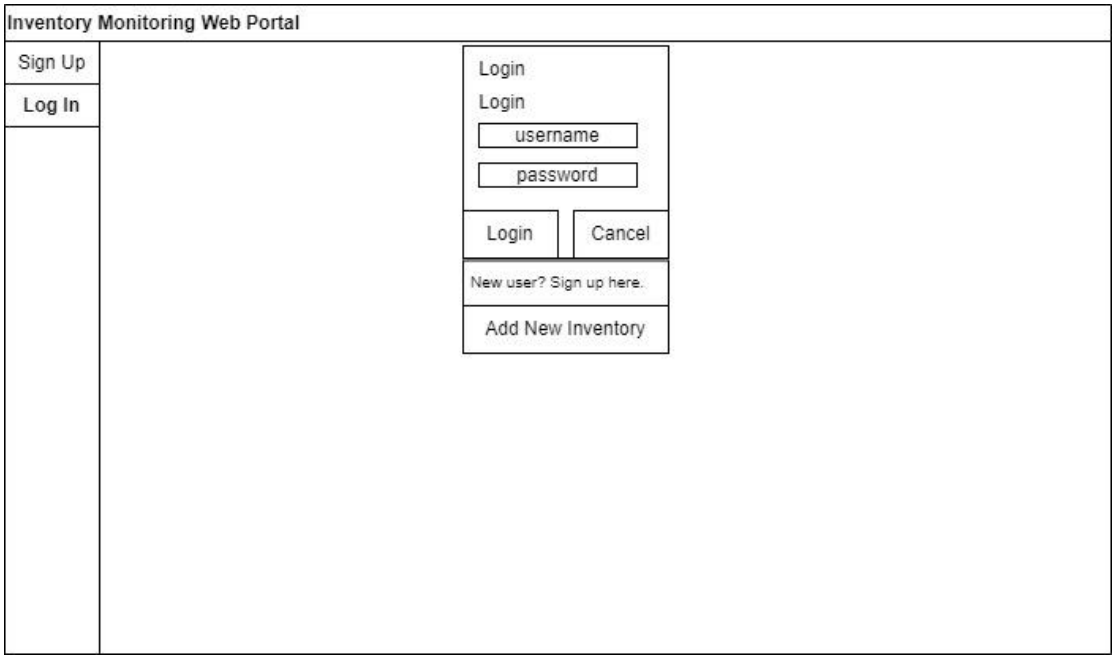


Figure 24 Login Interface Design

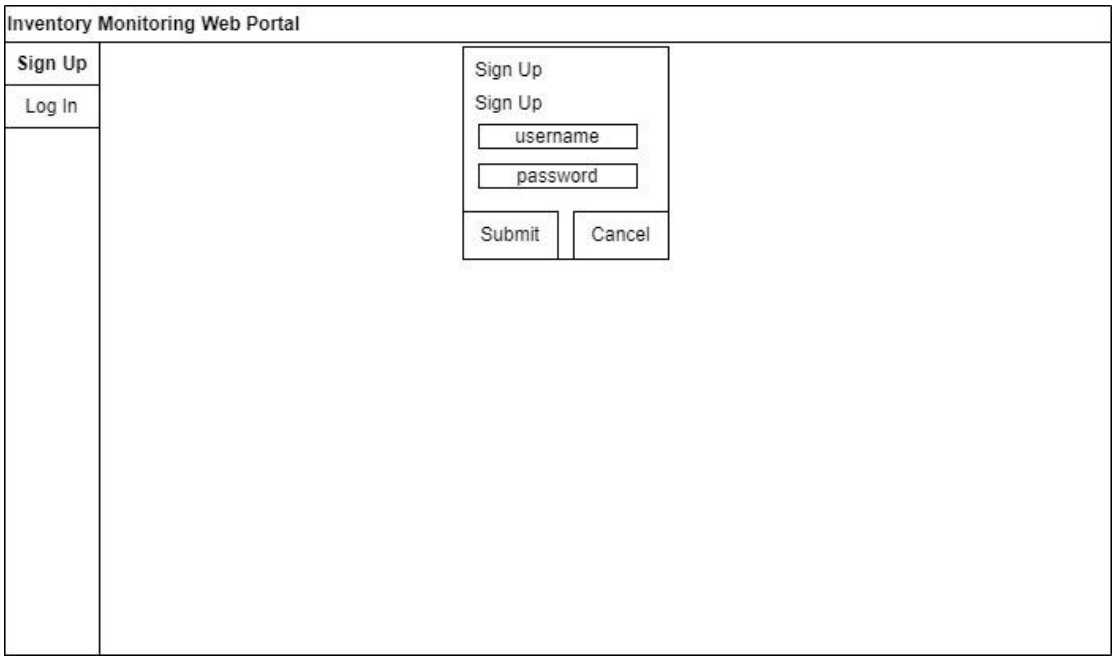


Figure 25 Sign Up Interface Design

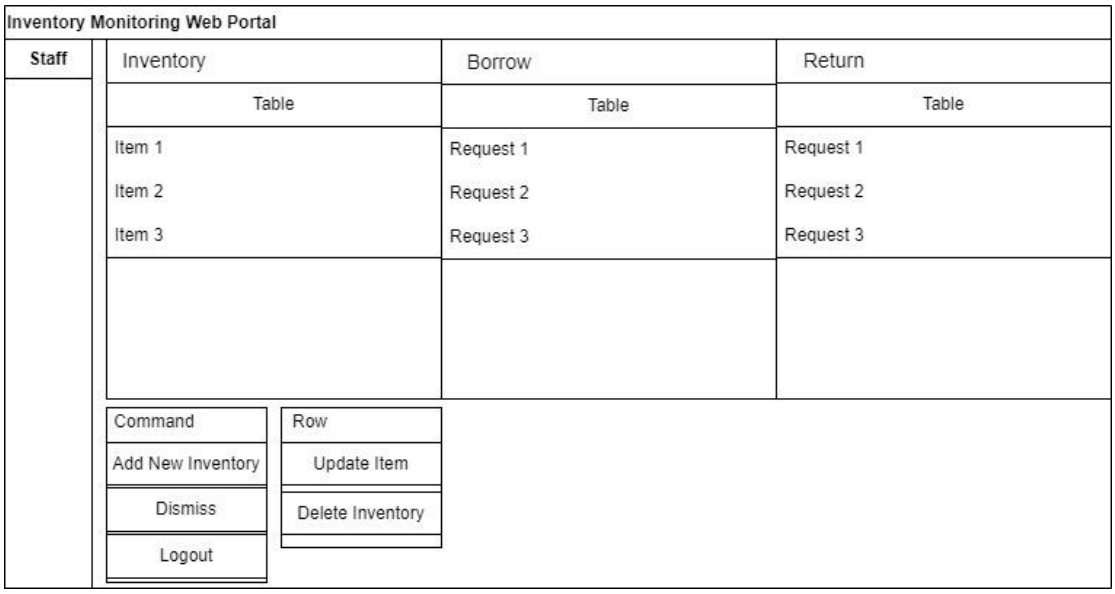


Figure 26 Staff Interface Design

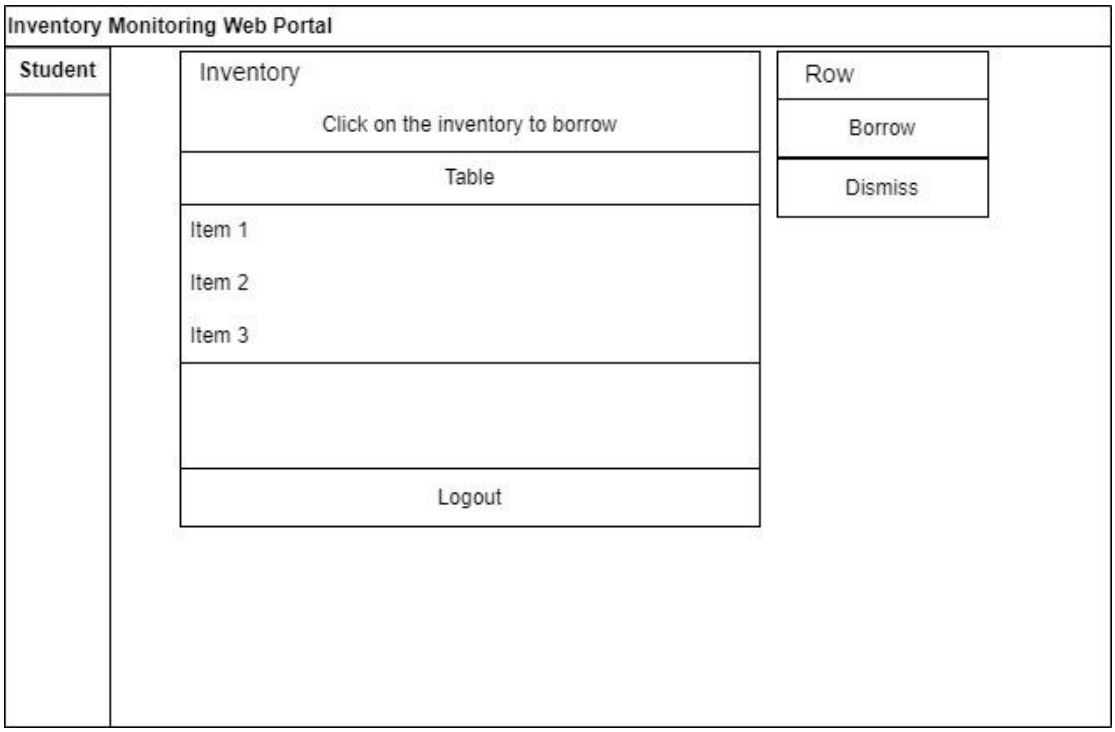


Figure 27 Student Interface Design

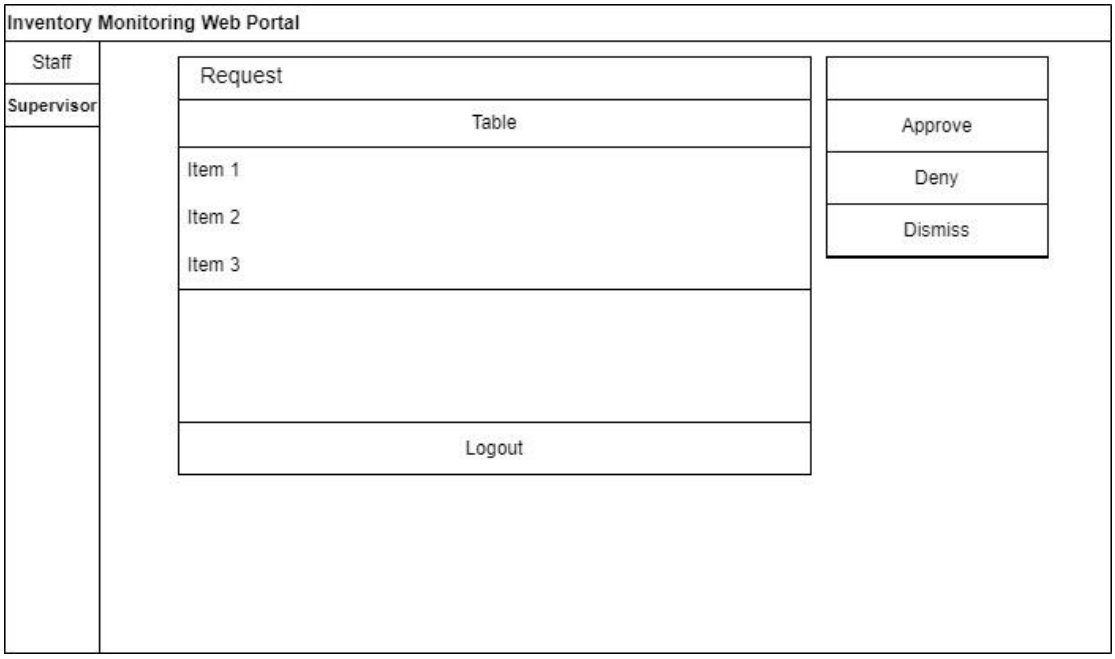


Figure 28 Supervisor Interface Design

4.5 Concluding Remark

In this chapter, the system architecture diagram was illustrated. Next, the functional modules in the systems were illustrated as a use case diagram, along with all the 13 use case descriptions. In addition to that, to demonstrate the system flow, a flowchart of the whole system was drawn with explanations of each of the flows. Besides, four activity diagrams representing the Sign Up use case, Login use case, Staff Inventory Management use cases, Borrow and Return Item use cases were drawn. Five of the design interfaces which includes Login, Sign Up, Student, Staff, Supervisor were shown.

Chapter 5 System Implementation

5.1 Hardware Setup

The hardware involved in this project was computer device. Computer is used for developing the web portal and the testing of the web portal.

Table 16 Specifications of laptop

Description	**Specifications**
Model	Acer Nitro 5 series
Processor	AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz
Operating System	Windows 11
Graphic	NVIDIA GeForce RTX3050ti 4GB DDR4
Memory	16GB DDR4 RAM
Storage	Kingston 512GB SSD

5.2 Software Setup

The programming tool used for development in this project is Node-RED. Node-RED provides browser-based editor that simplifies the process of development by wiring together the flows using nodes and palette that can be deployed to runtime in one click. For the database storage, only csv files were used to perform simple read and write files.

Node-RED operates on Node.js, which serves as the runtime environment. For this project, Node.js version 20.11.1was installed. Node.js is required to run the Node-RED server and manage the installed palettes and flows.

5.3 Setting and Configuration

The inventory.csv with the inventory list was set up and stored in the same directory. Node.js was installed and ran. Then, Node-RED was downloaded and executed. The required modules such as node-red-dashboard was also installed in the Node-RED page.

	A	B	C	D	E	F	G	H
1	id	name	quantity					
2	1	Raspberry	2					
3	2	LCD projec	3					
4	3	24 port sw	1					
5	4	HDMI cabl	6					
6	5	LED monit	7					
7	6	Pump	3					
8	7	Resistors	11					
9	8	Capacitors	13					
10	9	Transistors	14					
11	10	Diodes	8					
12								
13								
14								
15								
16								

Figure 29 inventory.csv file content

	A	B	C	D	E	F	G	H	I	J	K
1	username	id	item	amount	status	supervisor	start date	end date	start time	end time	duration
2	bodhi	10	Diodes	3	In Progress	drlim	6/9/2024		12:31:07		
3	bodhi	11	Router	1	In Progress	drlim	6/9/2024		12:31:50		
4	brandon	7	Resistor	4	In Progress	drtan	7/9/2024		14:07:24		
5	james	8	Capacitor	3	In Progress	drlee	7/9/2024		15:52:30		
6	james	7	Resistor	2	In Progress	drlee	8/9/2024		9:01:06		
7	junjie	9	Transistor	2	In Progress	drlee	8/9/2024		12:30:30		
8											
9											
10											

Figure 30 request.csv file content

	A	B	C	D	E	F
1	username	password	role	status	borrowed	supervisor
2	staff	U2FsdGVkX1/AM	staff			
3	drlee	U2FsdGVkX19UiF	supervisor			
4	james	U2FsdGVkX19qzj	student	requested	0	drlee
5	brandon	U2FsdGVkX1/R1a	student	requested	0	drtan
6	staff2	U2FsdGVkX18sZZ	staff			
7	bodhi	U2FsdGVkX19FjC	student	requested	0	drlim
8	drtan	U2FsdGVkX1+3+I	supervisor			
9	drlim	U2FsdGVkX18uF(	supervisor			
10	junjie	U2FsdGVkX19H9	student	requested	0	drlee
11	jiale	U2FsdGVkX19eb2	student	none	0	drlim
12						
13						

Figure 31 user_info.csv file content

Figure 5.3.1, 5.3.2 and 5.3.3 above show the inventory.csv, request.csv and user_info.csv. The inventory.csv was used to store the inventory list that contains the item id, item name and item quantity. The request.csv file was used to store borrow request, return request and sign up request. For item borrow and return requests, the format is username, item id, item name, borrow amount, user status, supervisor, the starting date, ending date, starting time, ending time and the total duration of the request. As for the sign up request stored request.csv were in the format of username, role, password and status. The user_info.csv stores username, encrypted password, user role, user status (none, requested, borrowing), borrowed amount and assigned supervisor.

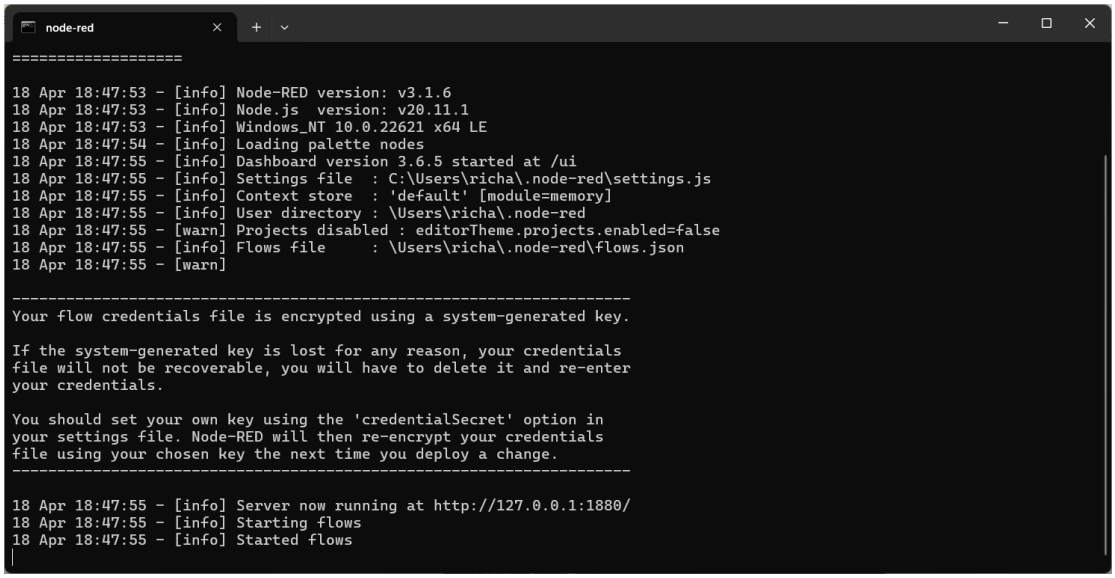


Figure 32 Execution of Node-RED on command prompt

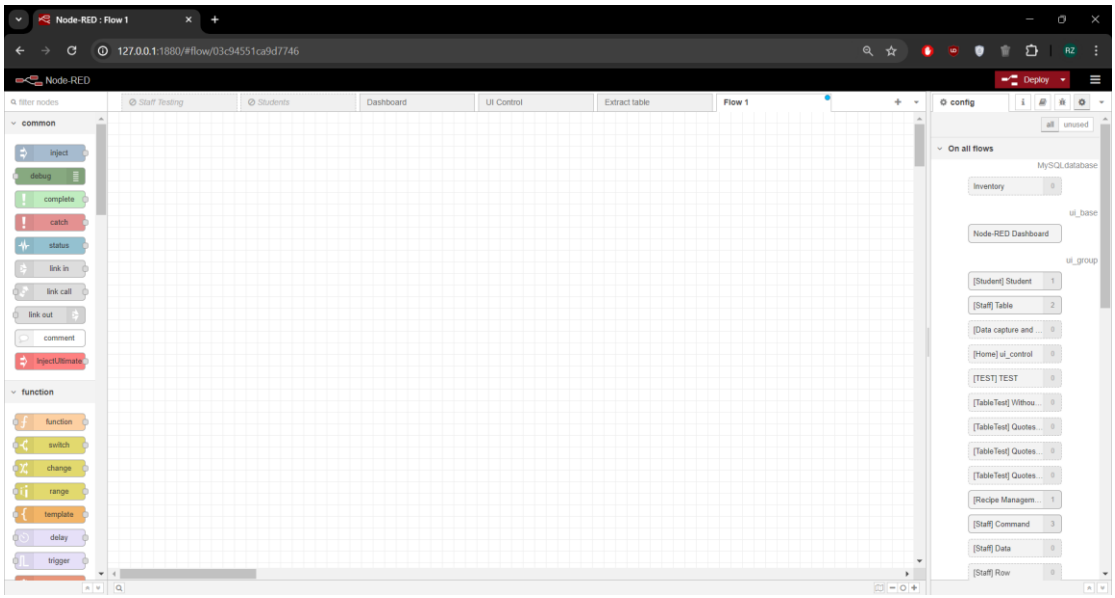


Figure 33 Node-RED editor

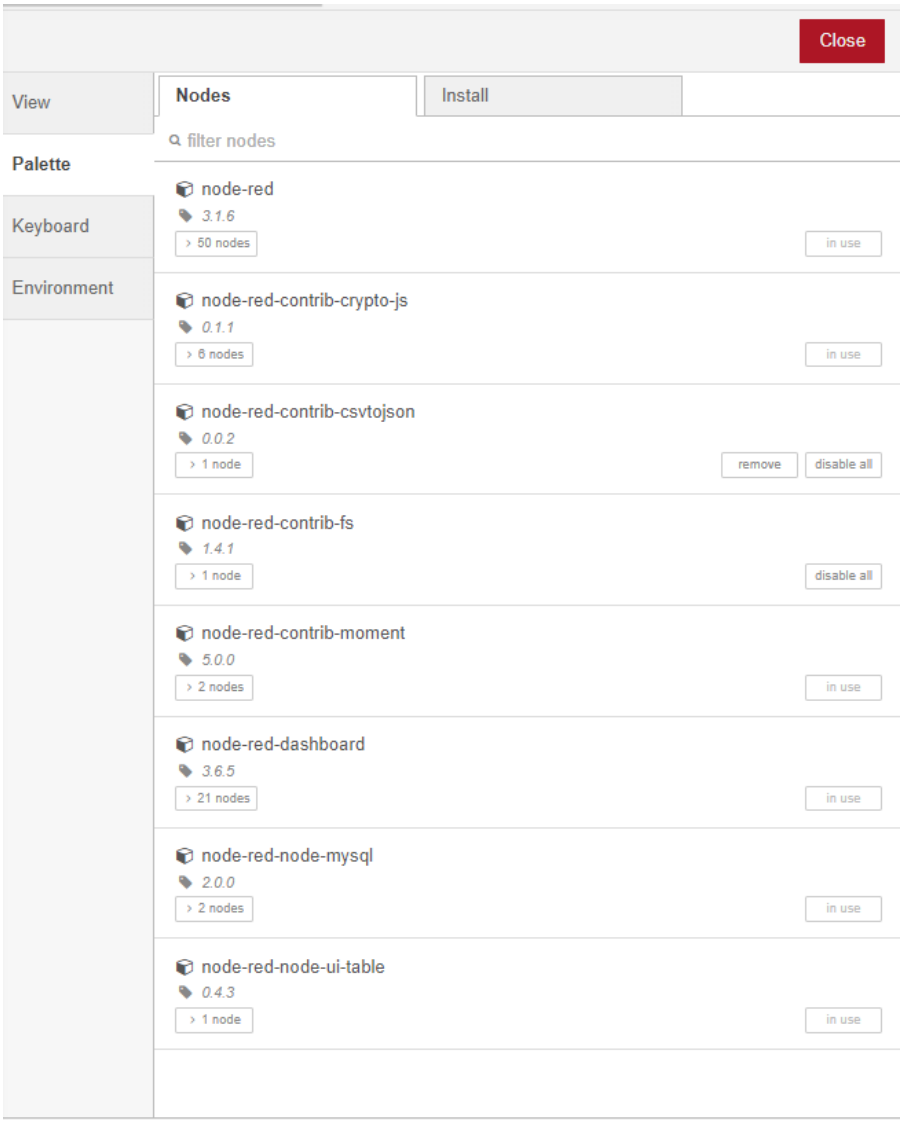


Figure 34 Palette node-red-dashboard

There were several palettes that were installed additionally and configured to be utilized for the project. The first one is node-red-crypto-js which was used for encryption and decryption of the password. Node-red-contrib-moment is a palette that helps extracting the current timestamp in an easy to read format. Node-red-dashboard and node-red-node-ui-table are the nodes that were used to display the inventory list and request on the dashboard, to better visualize the data.

5.4 System Operations

5.4.1 User Login Page

Figure 5.4.1.1 shows the user login page of the inventory monitoring web portal. The system requires user to input username and password in order to log into the system. Figure 5.4.1.2 shows when the user input invalid username, the system will prompt "Username not found, please try again!". Figure 5.4.1.3 shows if the user key in invalid username or wrong password, causing the system to prompt "Login failed. Please try again.". Figure 5.4.1.4 shows if the user successfully key in correct username and password, the system redirect user to the respective page with a "Successfully Login!" notification.

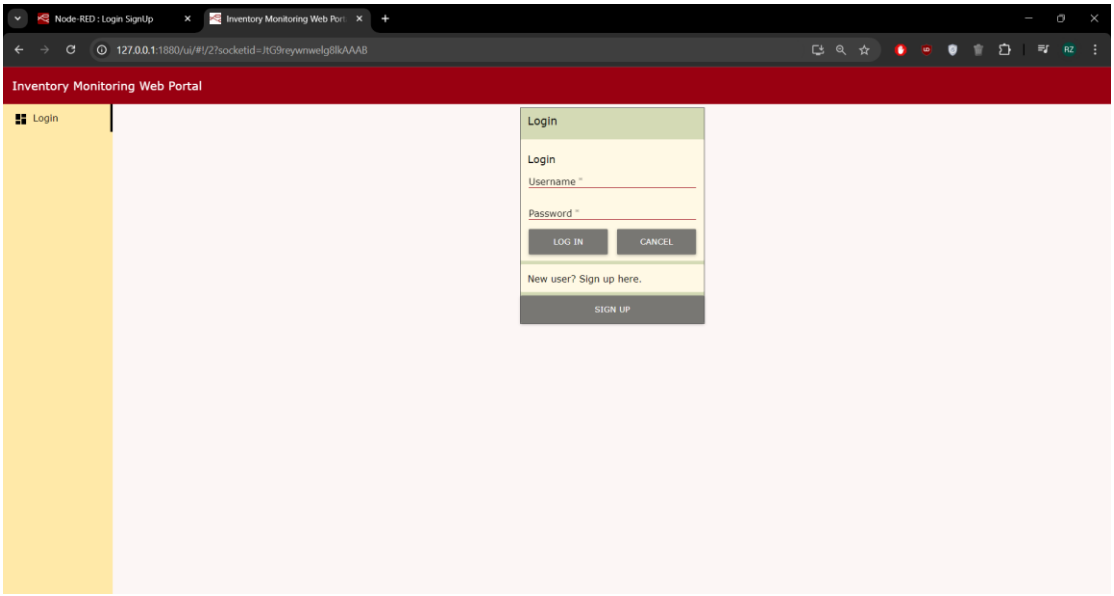


Figure 35 User Login Page

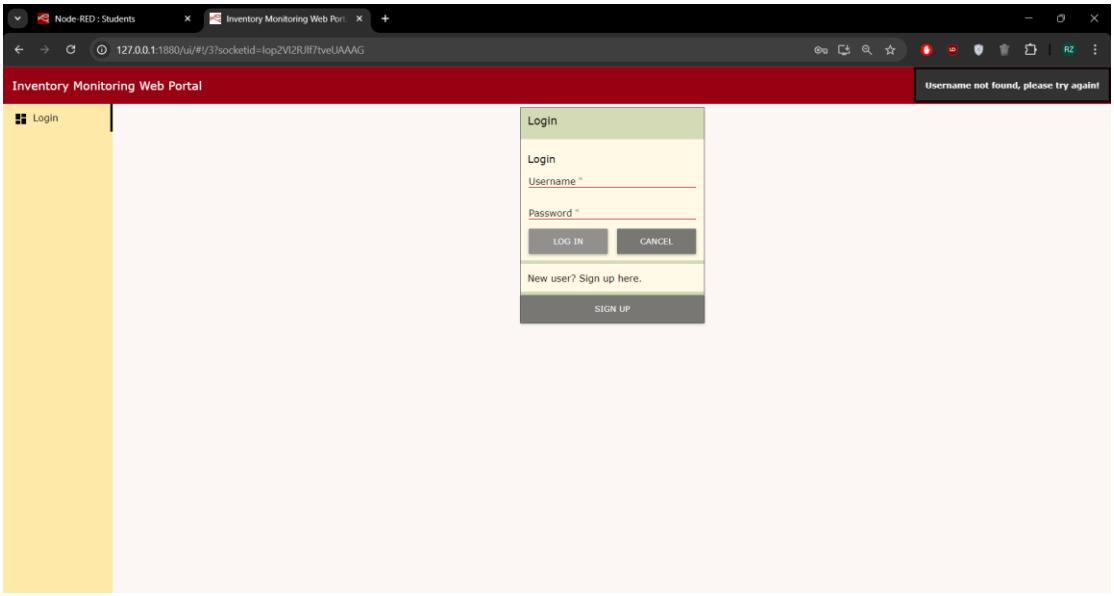


Figure 36 Invalid username

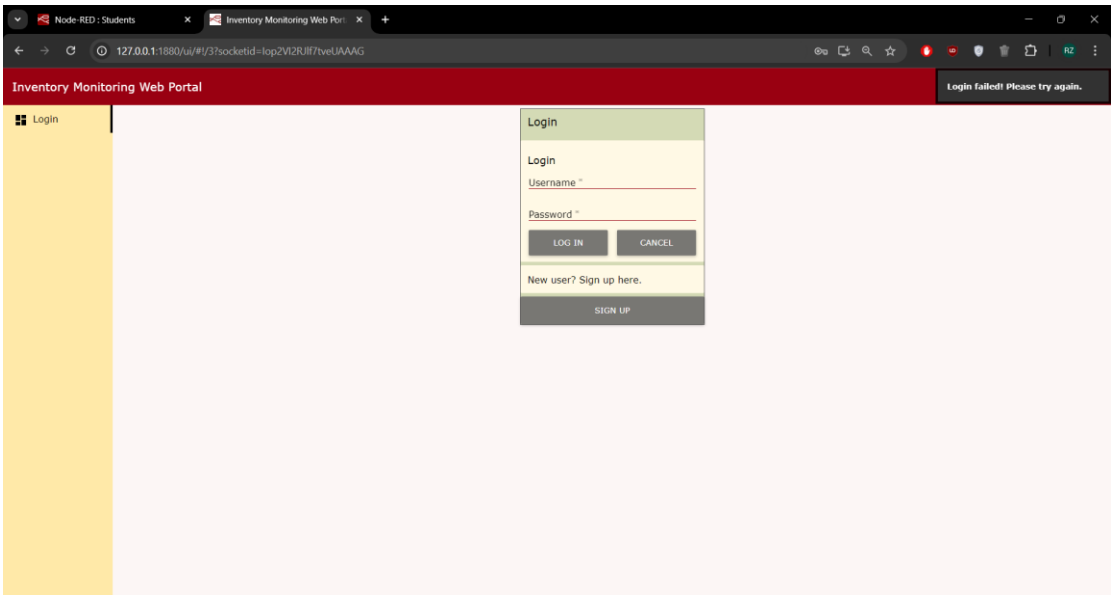


Figure 37 Invalid password

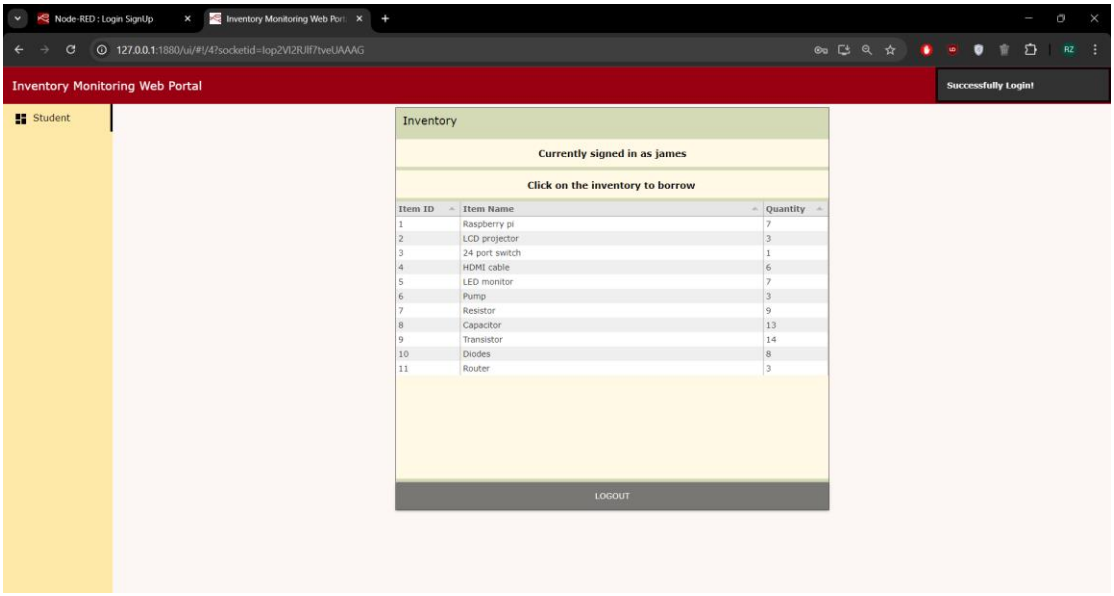


Figure 38 Successful Login

5.4.2 Sign Up Page

Figure 5.4.2.1 shows the user sign up page. Figure 5.4.2.2 shows the output if the user key in existing username. In this example, the username ‘james’ has already been taken. The system prompts “The username ‘james’ has already been taken.” to notify the user that the username is taken. Figure 5.4.2.3 shows if the user entered a new username, the system will prompt the username is successfully registered, waiting for approval. In the figure 5.4.2.4, the new sign up request is added to the request.csv file, containing the username, role, encrypted password, new user status.

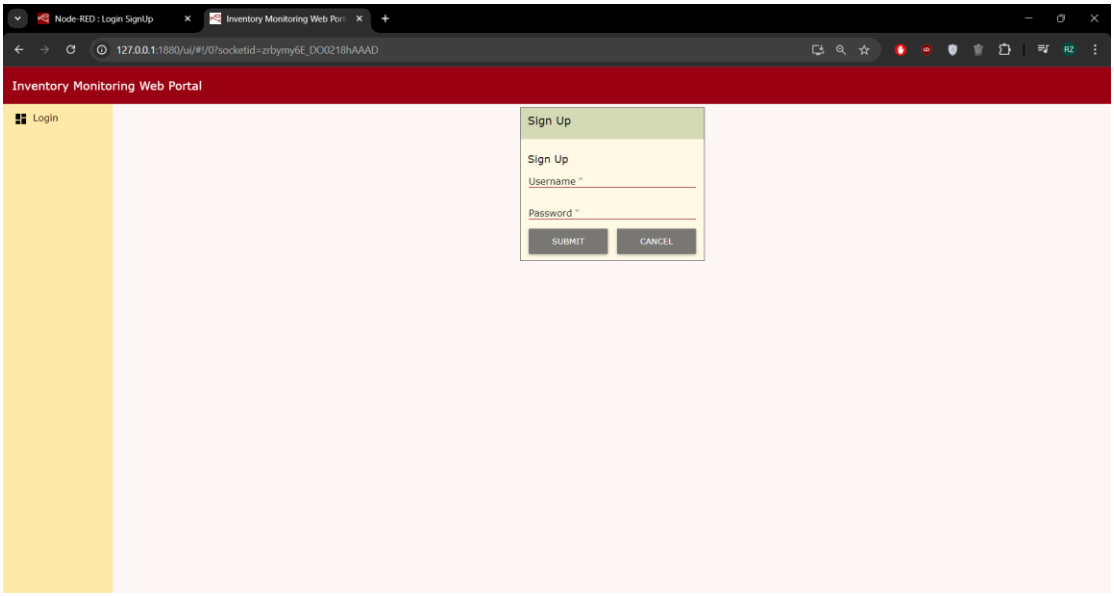


Figure 39 Sign Up Page

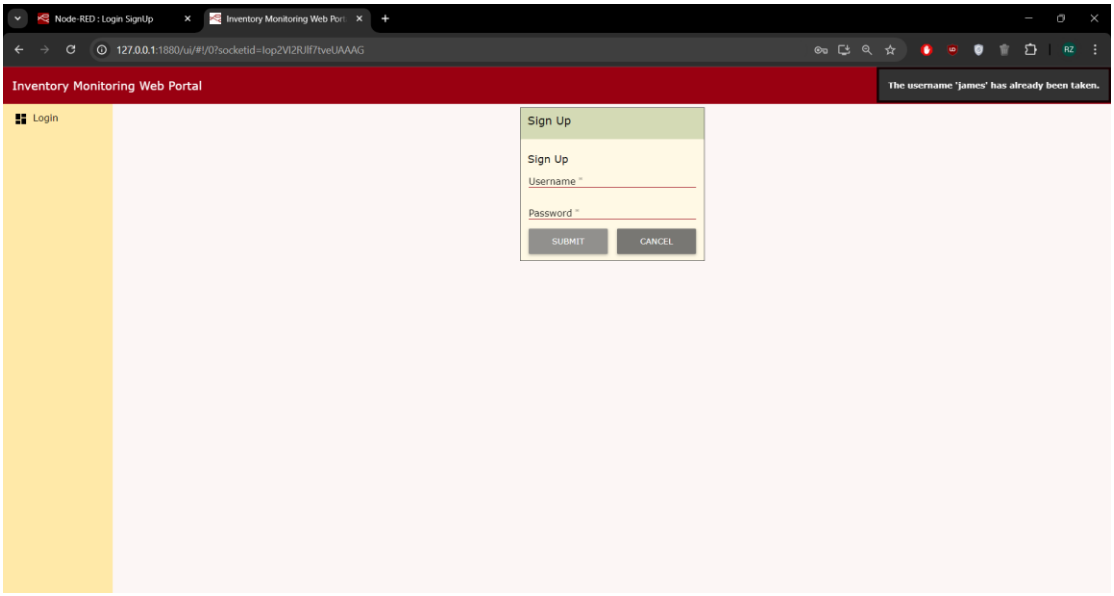


Figure 40 Existing username

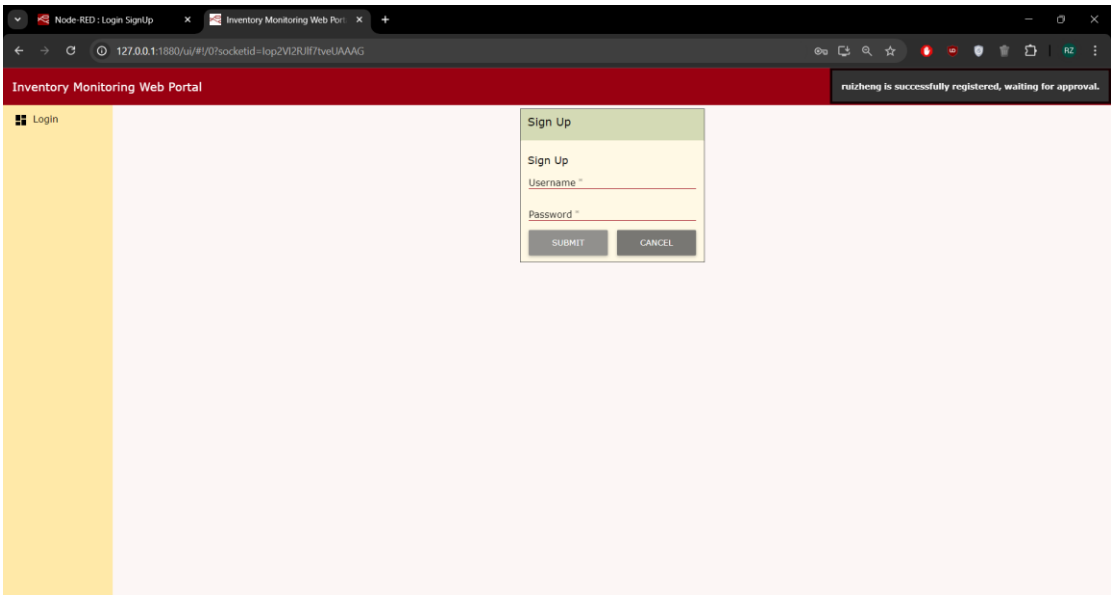


Figure 41 Successful sign up

	A	B	C	D	E	F	G	H	I	J	K
1	username	id	item	amount	status	supervisor	start date	enddate	start time	endtime	duration
2	bodhi	10	Diodes	3	In Progress	drlim	6/9/2024		12:31:07		
3	bodhi	11	Router	1	approved	drlim	########		6:12:09		
4	brandon	7	Resistor	4	approved	drtan	########		6:12:17		
5	james	8	Capacitor	3	In Progress	drlee	7/9/2024		15:52:30		
6	james	7	Resistor	2	borrowed	drlee	########		6:12:43		
7	junjie	9	Transistor	2	In Progress	drlee	8/9/2024		12:30:30		
8	ruizheng	role	U2FsdGVkX1+kUfAFu		new user						
9											

Figure 42 Updated request.csv

Figure 5.4.2.4

5.4.3 Student Interface

Figure 5.4.3.1 shows the student interface with the inventory displayed. Figure 5.4.3.2 shows if the student clicks on an inventory, the row is selected and the “Borrow” button pops up. Figure 5.4.3.3 shows the system prompts the user to key in borrow amount after clicking the “Borrow” button. If the borrow amount is lower than the existing item quantity, the system will notify “Borrow request sent!” as shown in figure 5.4.3.4 and adds the borrow request to the request.csv as shown in Figure 5.4.3.5. If the borrow amount input by the student exceeds the existing item quantity, the system will prompt “The borrow amount must be less than or equals to the available quantity.” as shown in figure 5.4.3.6

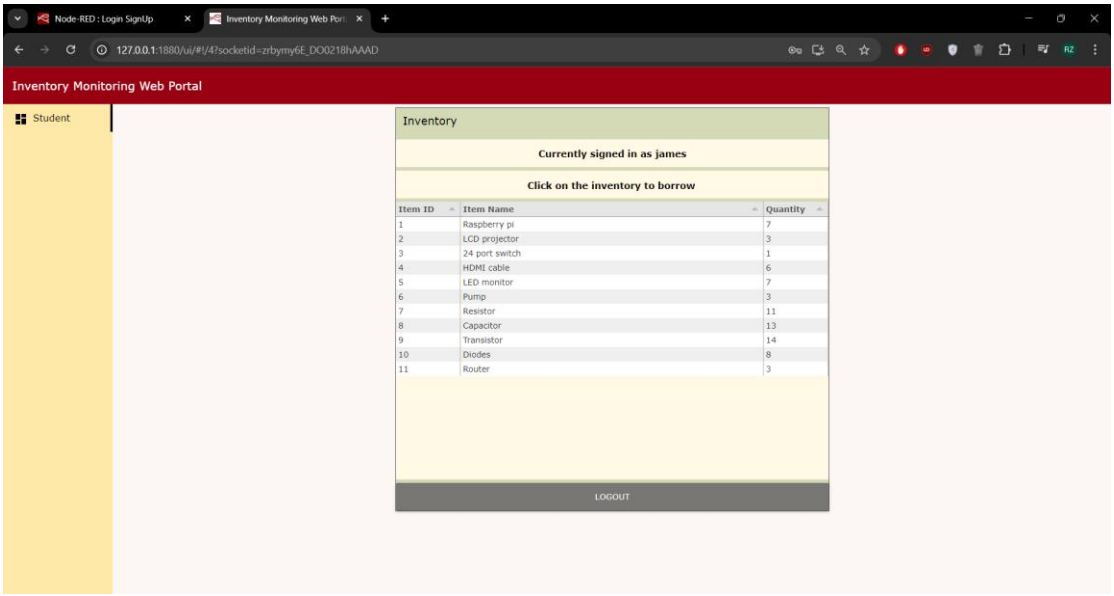


Figure 43 Student Interface

Figure 5.4.3.1

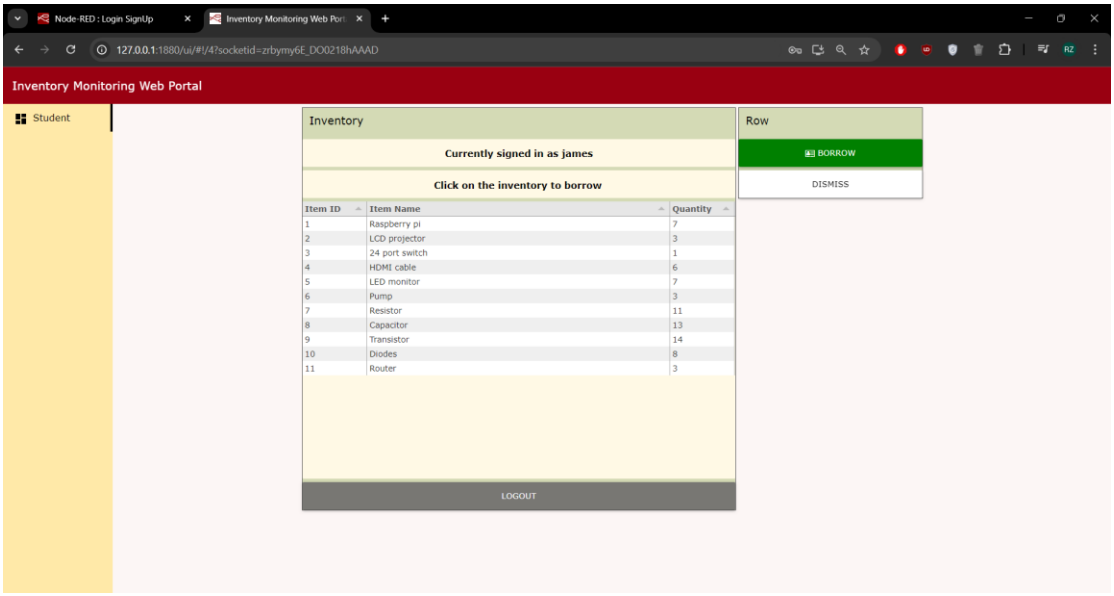


Figure 44 Selected row, Borrow prompt

Figure 5.4.3.2

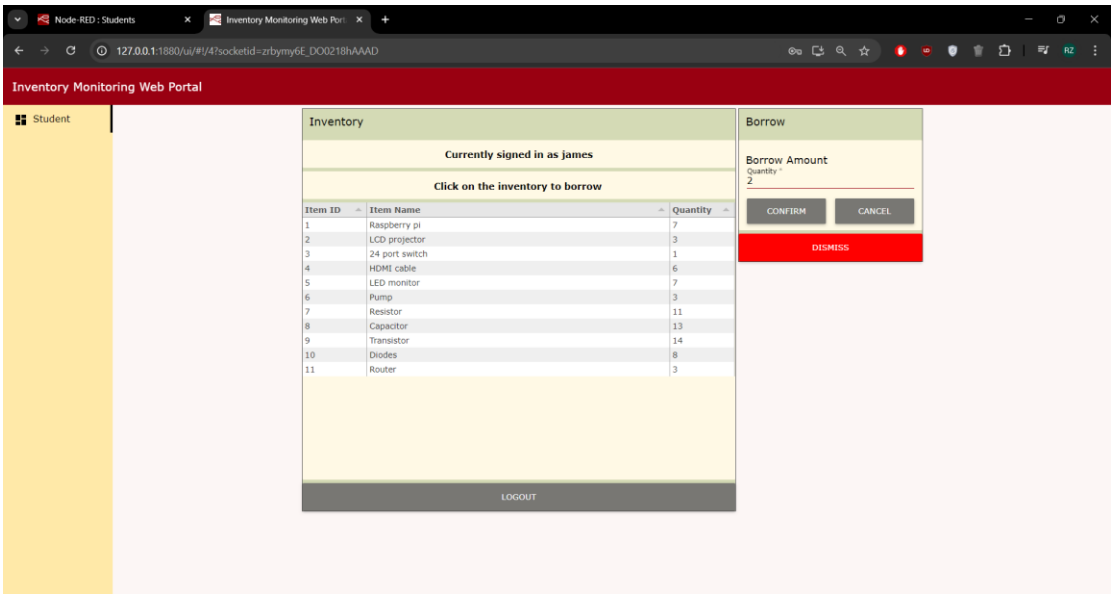


Figure 45 Borrow amount

Figure 5.4.3.3

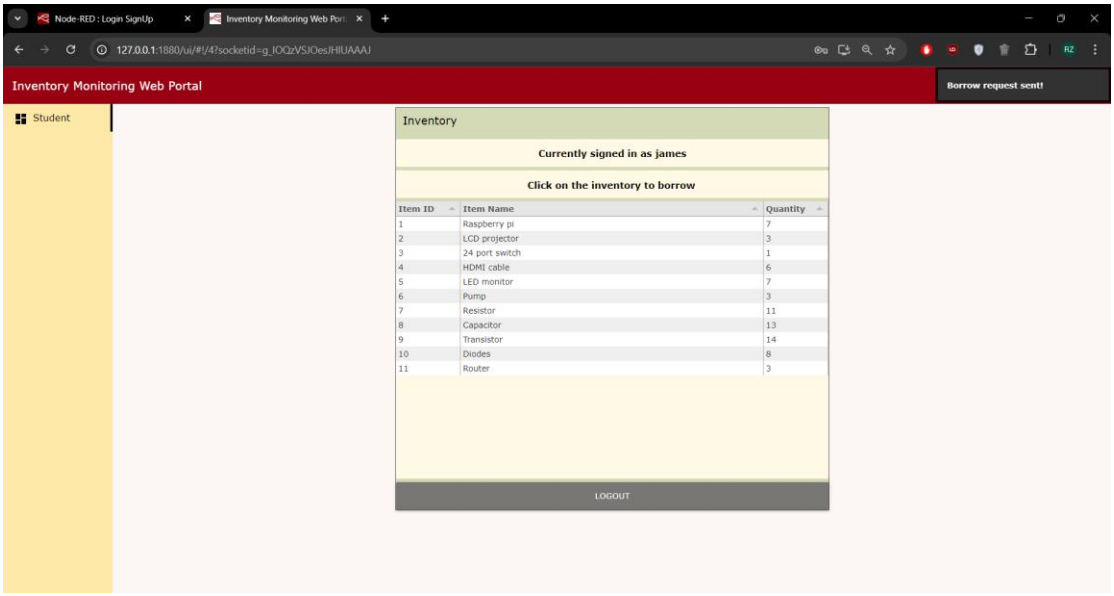


Figure 46 Borrow request sent

Figure 5.4.3.4

	A	B	C	D	E	F	G	H	I	J	K
1	username	id	item	amount	status	supervisor	start date	enddate	start time	endtime	duration
2	bodhi	10	Diodes	3	In Progress	drlim	6/9/2024		12:31:07		
3	bodhi	11	Router	1	In Progress	drlim	6/9/2024		12:31:50		
4	brandon	7	Resistor	4	In Progress	drtan	7/9/2024		14:07:24		
5	james	8	Capacitor	3	In Progress	drlee	7/9/2024		15:52:30		
6	james	7	Resistor	2	In Progress	drlee	8/9/2024		9:01:06		
7	junjie	9	Transistor	2	In Progress	drlee	8/9/2024		12:30:30		
8	james	4	HDMI cabl	2	In Progress	drlee	12/9/2024		7:23:00		
9											

Figure 47 Updated request.csv

Figure 5.4.3.5

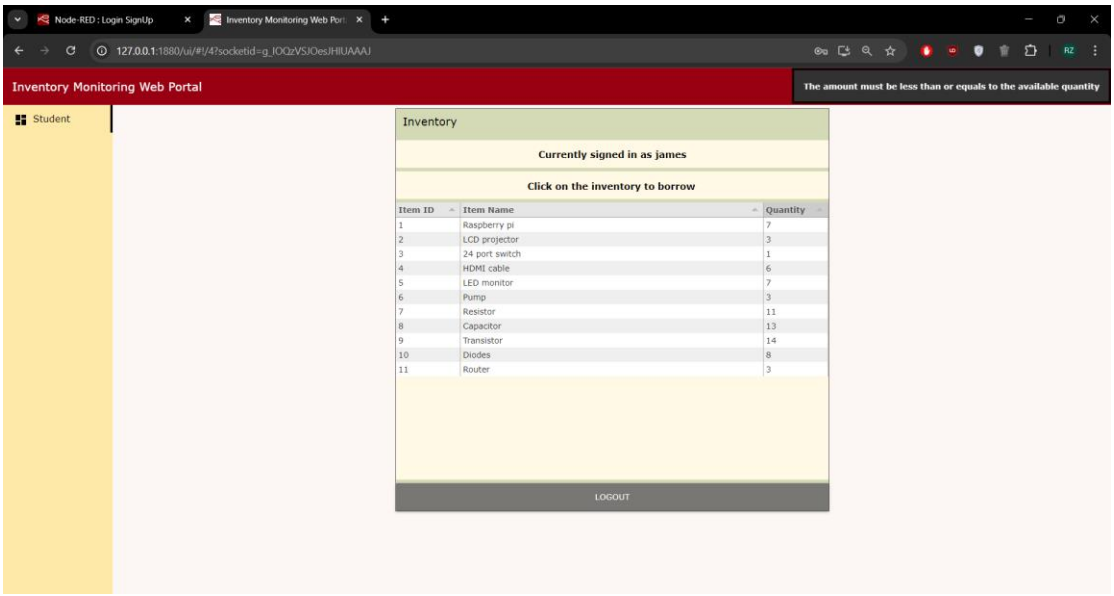


Figure 48 Borrow amount exceeding existing quantity

Figure 5.4.3.6

5.4.4 Supervisor Interface

Figure 5.4.4.1 shows the supervisor interface with the borrow requests and sign up requests of students under the supervisor shown. For example, the users 'james' and 'junjie' are students under the supervision of 'drlee'. Only the requests made by the users 'james' and 'junjie' will be displayed for the user 'drlee'. Figure 5.4.4.2 shows if the supervisor select a row of request, the "Approve", "Deny" and "Dismiss" buttons will be prompted. If the supervisor clicks on the "Approve" button, the system will prompt "Approved!" to the user as shown in figure 5.4.4.3. If the supervisor clicks on "Deny" button, the system will show "Removed request from the inventory" notification and remove the selected row from the supervisor interface as shown in figure 5.4.4.4. Figure 5.4.4.5 shows the updated status in the request.csv to "approved" and the removal of the denied request. The user status is also updated to "approved" in user_info.csv as shown in figure 5.4.4.6. The figure 5.4.4.7 shows the new sign up request generated shown in the example figure 5.4.2.3. When the supervisor selects the sign up request, the "Approve", "Deny" and "Dismiss" buttons will pop up as shown in figure 5.4.4.8. Once the supervisor clicks the "Approve" button, the notification "Approved!" will be displayed as shown in figure 5.4.4.9. Figure 5.4.4.10 shows the approved sign up request was updated in user_info.csv. Figure 5.4.4.11 shows the notification of denied sign up request.

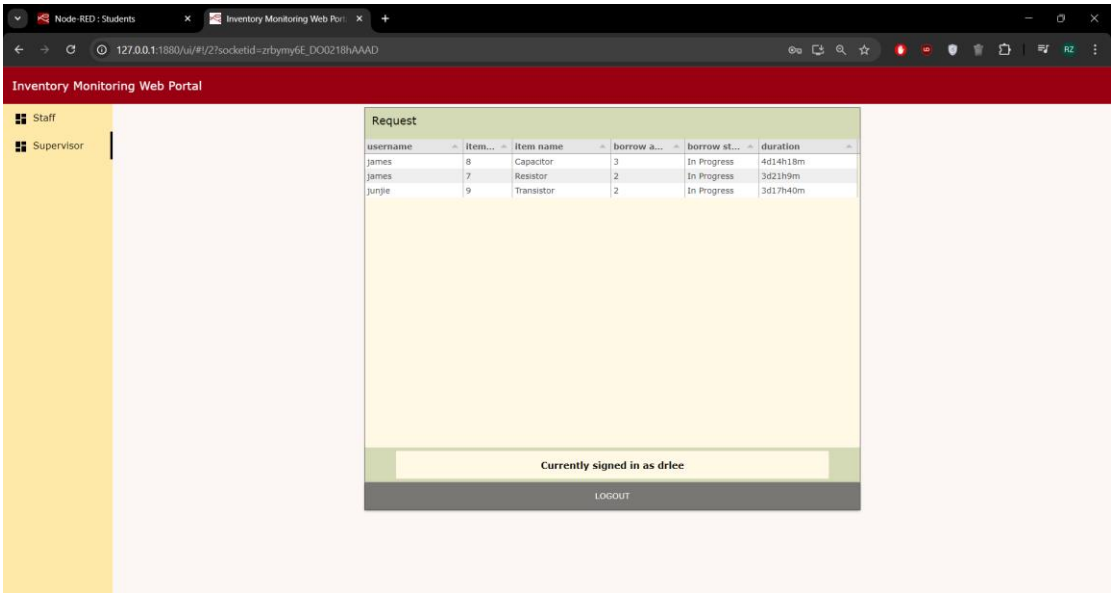


Figure 49 Supervisor Interface

Figure 5.4.4.1

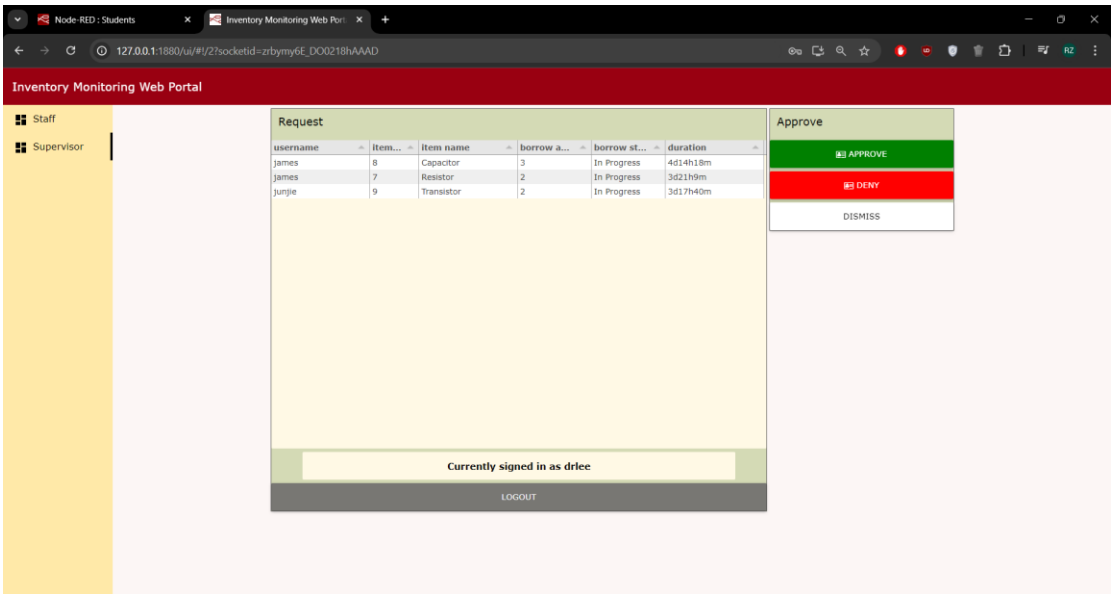


Figure 50 Selected row, Approve prompt

Figure 5.4.4.2

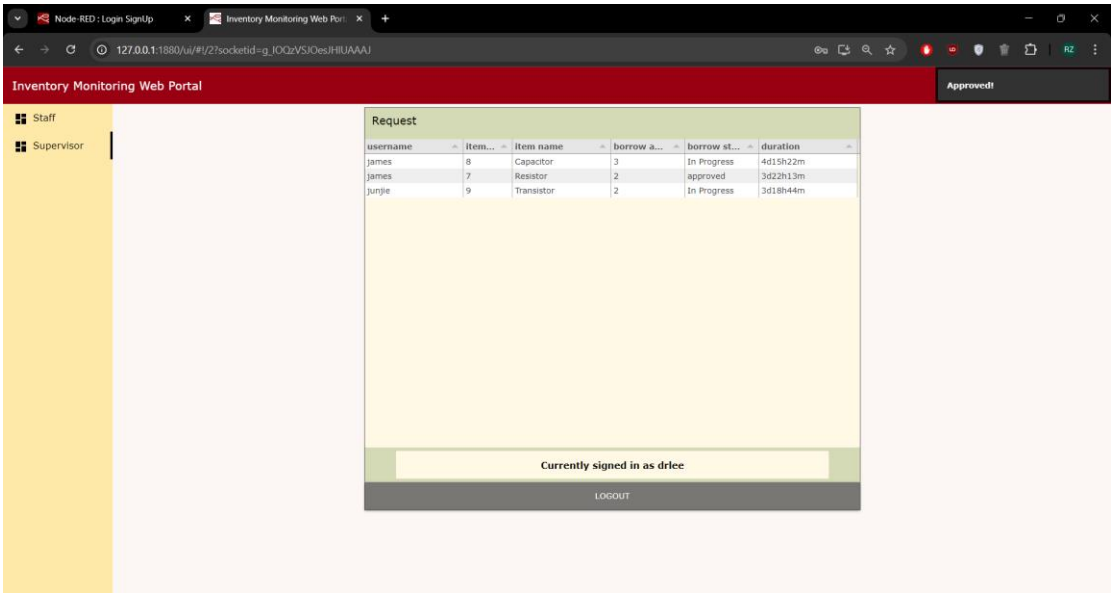


Figure 51 Approved notification

Figure 5.4.4.3

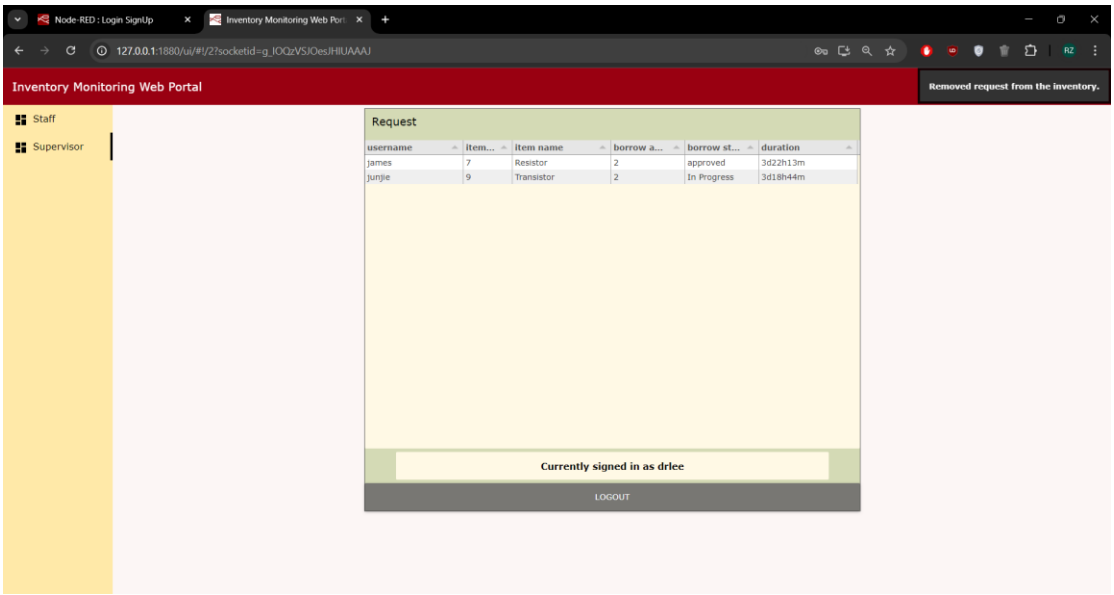


Figure 52 Denied request notification, removal of selected row

Figure 5.4.4.4

	A	B	C	D	E	F	G	H	I	J	K
1	username	id	item	amount	status	supervisor	start date	enddate	start time	endtime	duration
2	bodhi	10	Diodes	3	In Progress	drlim	6/9/2024		12:31:07		
3	bodhi	11	Router	1	In Progress	drlim	6/9/2024		12:31:50		
4	brandon	7	Resistor	4	In Progress	drtan	7/9/2024		14:07:24		
5	james	8	Capacitor	3	In Progress	drlee	7/9/2024		15:52:30		
6	james	7	Resistor	2	approved	drlee	12/9/2024		23:32:40		
7											
8											

Figure 53 Updated status in request.csv and removal of request

Figure 5.4.4.5

	A	B	C	D	E	F
1	username	password	role	status	borrowed	supervisor
2	staff	U2FsdGVk	staff			
3	drlee	U2FsdGVk	supervisor			
4	james	U2FsdGVk	student	approved	0	drlee
5	brandon	U2FsdGVk	student	requested	0	drtan
6	staff2	U2FsdGVk	staff			
7	bodhi	U2FsdGVk	student	requested	0	drlim
8	drtan	U2FsdGVk	supervisor			
9	drlim	U2FsdGVk	supervisor			
10	junjie	U2FsdGVk	student	none	0	drlee
11	jiale	U2FsdGVk	student	none	0	drlim
12						

Figure 54 Updated status in user_info.csv

Figure 5.4.4.6

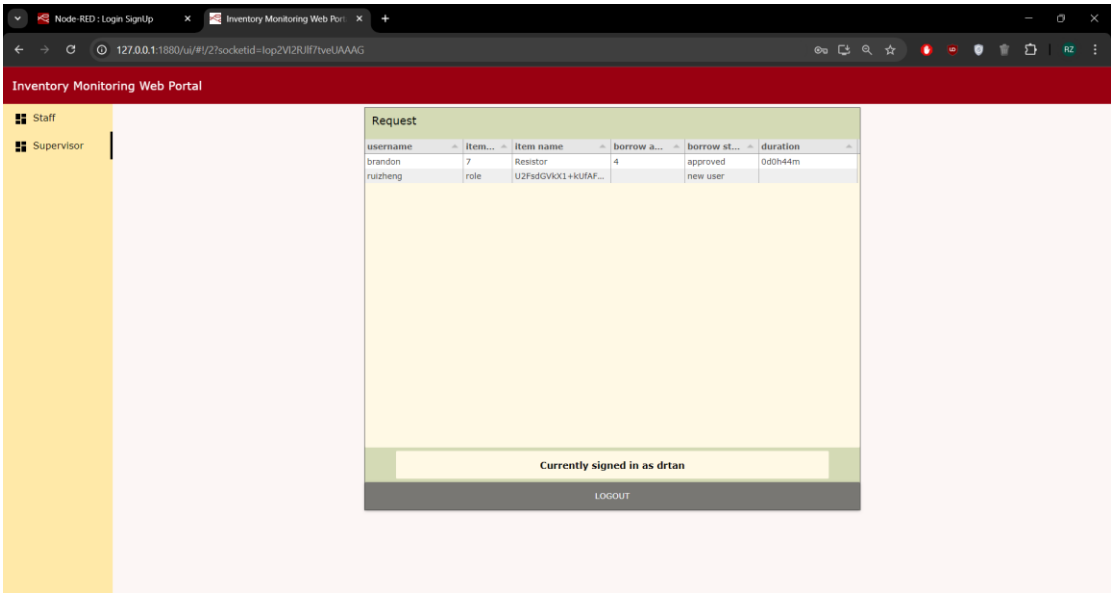


Figure 55 New sign up request

Figure 5.4.4.7

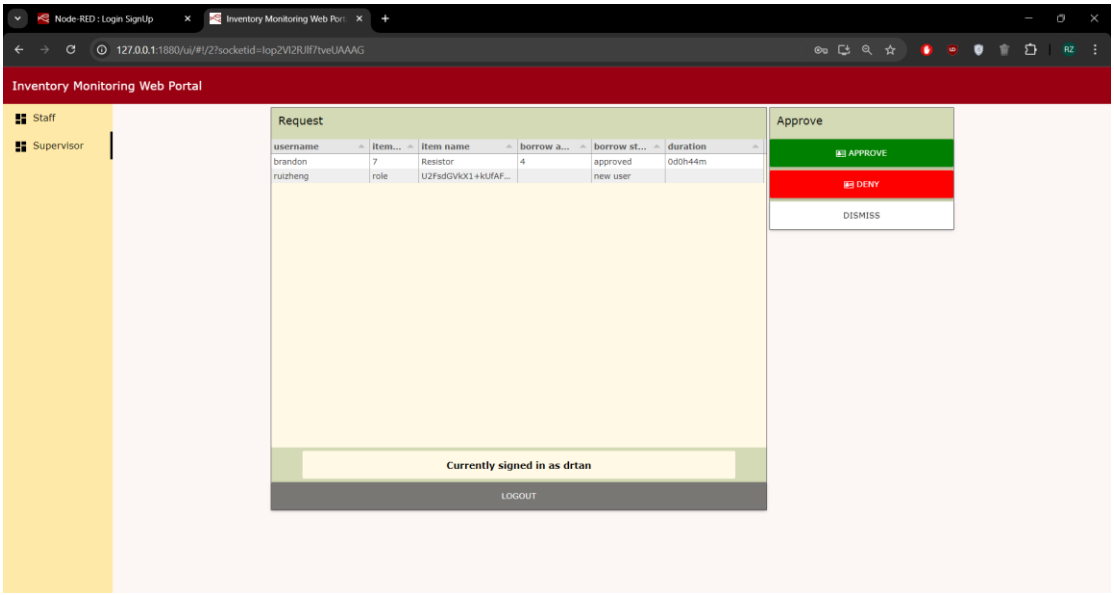


Figure 56 Selected sign up request, Approve prompt

Figure 5.4.4.8

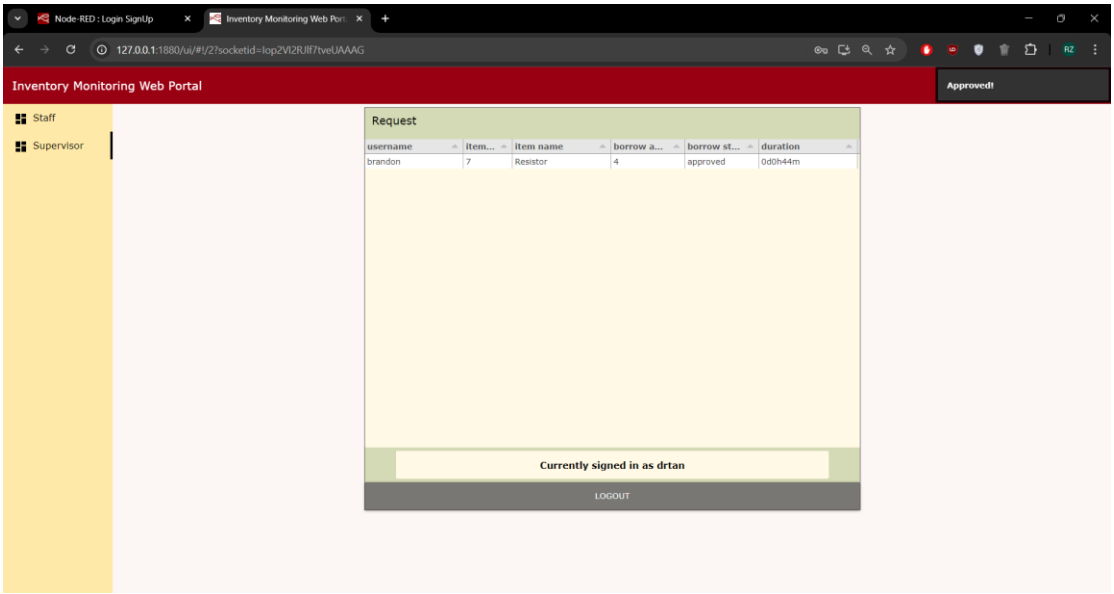


Figure 57 Approved sign up request notification

Figure 5.4.4.9

	A	B	C	D	E	F
1	username	password	role	status	borrowed	supervisor
2	staff	U2FsdGVk	staff			
3	drlee	U2FsdGVk	supervisor			
4	james	U2FsdGVk	student	borrowing	1	drlee
5	brandon	U2FsdGVk	student	approved	0	drtan
6	staff2	U2FsdGVk	staff			
7	bodhi	U2FsdGVk	student	approved	0	drlim
8	drtan	U2FsdGVk	supervisor			
9	drlim	U2FsdGVk	supervisor			
10	junjie	U2FsdGVk	student	requested	0	drlee
11	jiale	U2FsdGVk	student	none	0	drlim
12	ruizheng	U2FsdGVk	student	none	0	drtan
13						

Figure 58 Updated user_info.csv

Figure 5.4.4.10

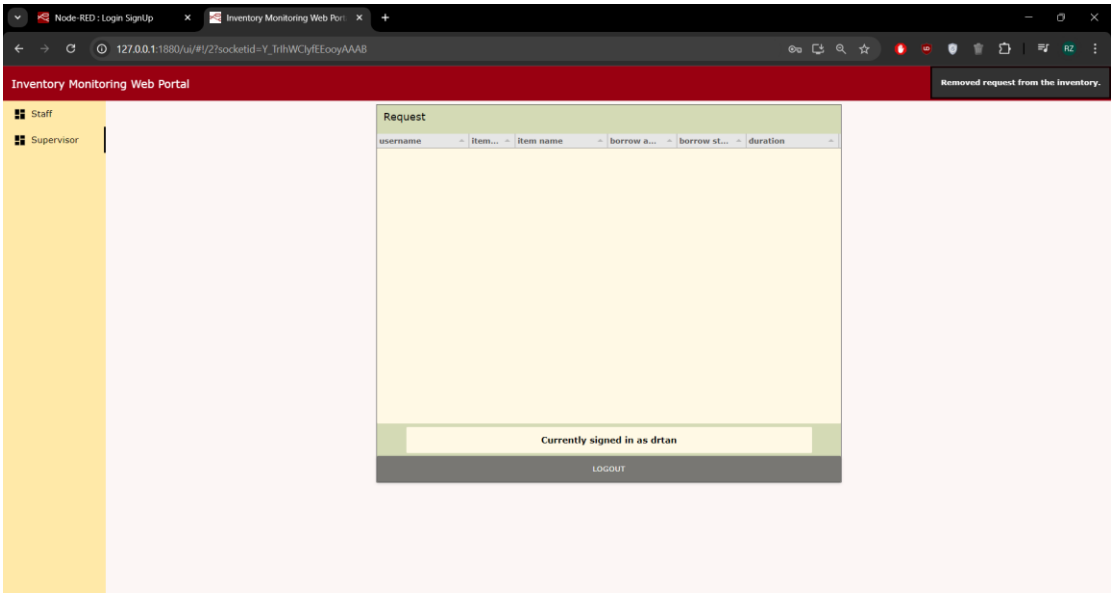


Figure 59 Denied user sign up request

Figure 5.4.4.11

5.4.5 Staff Interface

Figure 5.4.5.1 shows the staff interface with the inventory list, borrow request list and return request list on display. The command field with "Add New Inventory", "Dismiss" and "Logout" buttons were also displayed on default. Figure 5.4.5.2 and figure 5.4.5.6 show when "Add New Inventory" or "Update Inventory" button was clicked, the key in item field will pop up to prompt the staff to key in the item name and quantity. In figure 5.4.5.3, after the staff entered the item information, the add item notification will be shown. In this case, the notification is "Successfully added Switch" to the inventory. Figure 5.4.5.4 shows the updated inventory.csv with the newly added item. In figure 5.4.5.5, once a row of inventory is selected, the "Update Item" and "Delete Inventory" buttons will pop up. After the staff clicks "Update Item" and key in the item information, an update item notification will be shown. In figure 5.4.5.7, the notification "Successfully updated Switches" was shown. Figure 5.4.5.8 shows the updated item information in the inventory.csv. In figure 5.4.5.9, if the staff clicks on the "Delete Inventory" button, another prompt with the buttons "Confirm" and "Cancel" will be shown. Figure 5.4.5.10 shows the removed item notification as well as the inventory list with the selected item removed. Figure 5.4.5.11 shows the updated inventory list with the item removed in inventory.csv.

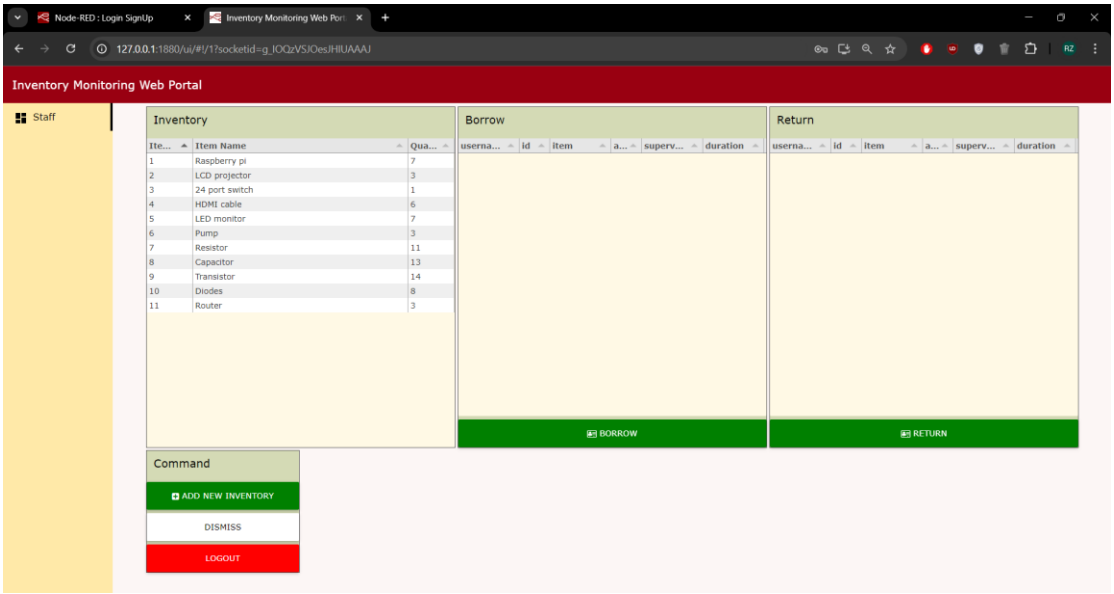


Figure 60 Staff Interface

Figure 5.4.5.1

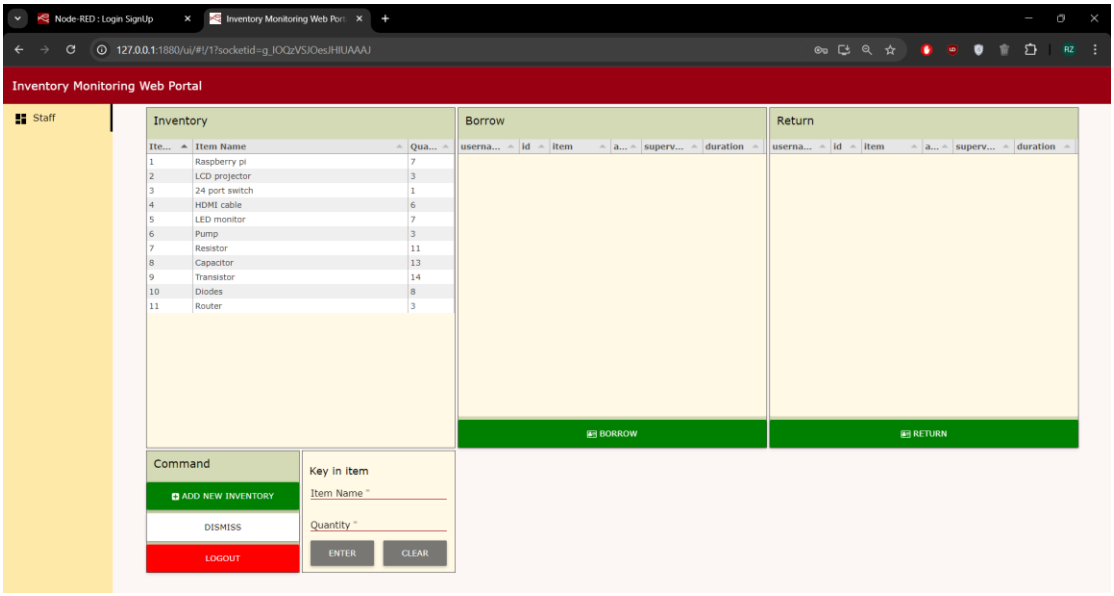


Figure 61 Add New Inventory, Key In Item Pop Up

Figure 5.4.5.2

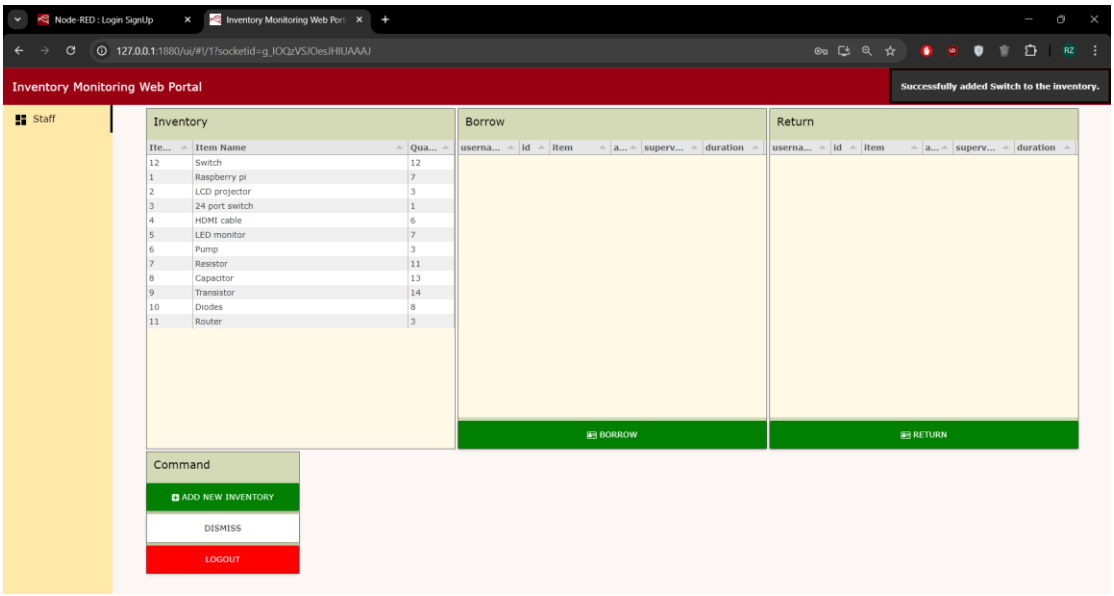


Figure 62 Add item notification

Figure 5.4.5.3

	A	B	C
1	id	name	quantity
2	1	Raspberry	7
3	2	LCD projec	3
4	3	24 port swi	1
5	4	HDMI cabl	6
6	5	LED monit	7
7	6	Pump	3
8	7	Resistor	11
9	8	Capacitor	13
10	9	Transistor	14
11	10	Diodes	8
12	11	Router	3
13	12	Switch	12
14			

Figure 63 Updated inventory.csv, added new item

Figure 5.4.5.4

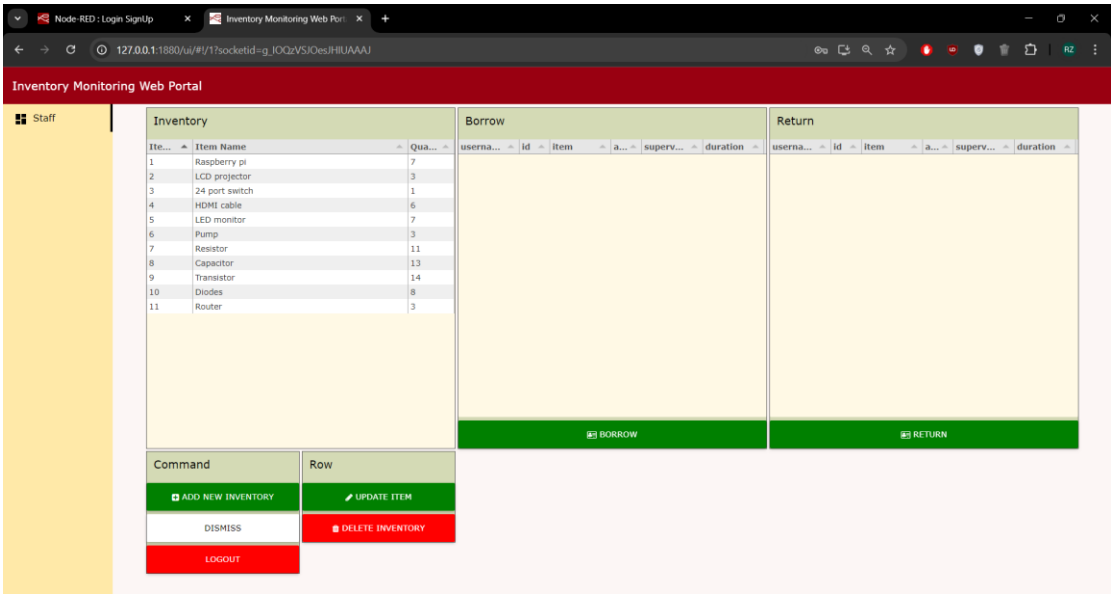


Figure 64 Select row, Update item pop up

Figure 5.4.5.5

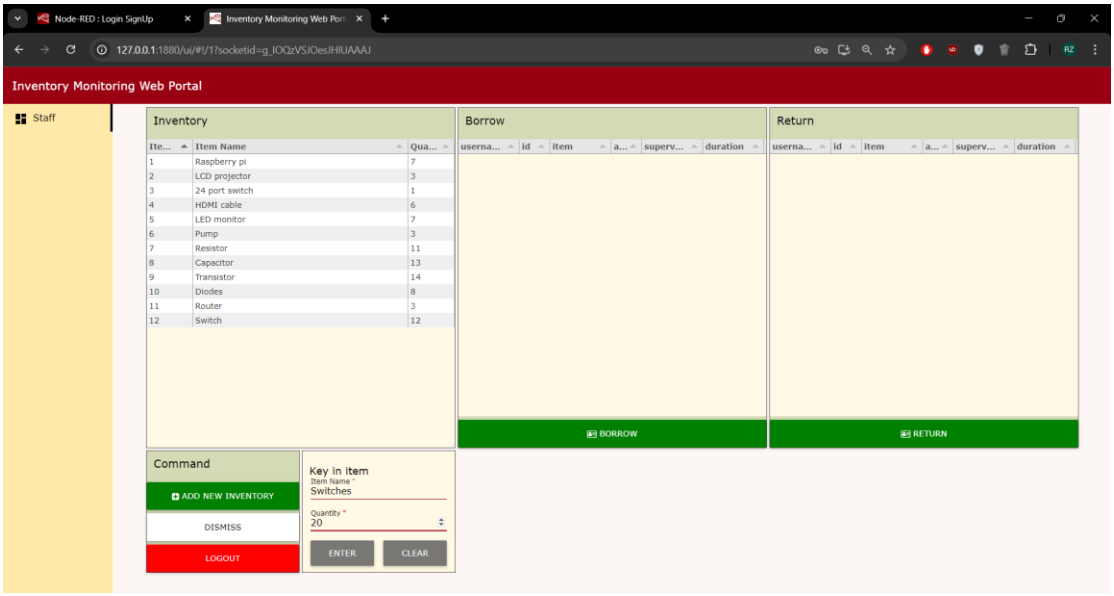


Figure 65 Key in item prompt

Figure 5.4.5.6

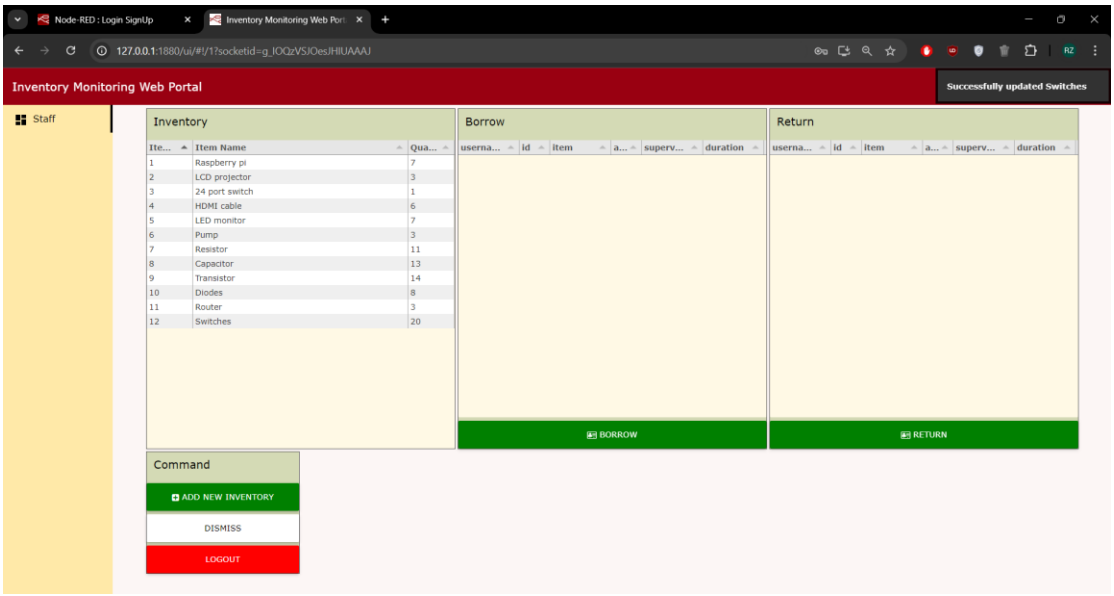


Figure 66 Updated item notification

Figure 5.4.5.7

	A	B	C
1	id	name	quantity
2	1	Raspberry	7
3	2	LCD projec	3
4	3	24 port swi	1
5	4	HDMI cabl	6
6	5	LED monit	7
7	6	Pump	3
8	7	Resistor	11
9	8	Capacitor	13
10	9	Transistor	14
11	10	Diodes	8
12	11	Router	3
13	12	Switches	20
14			

Figure 67 Updated inventory.csv with updated item information

Figure 5.4.5.8

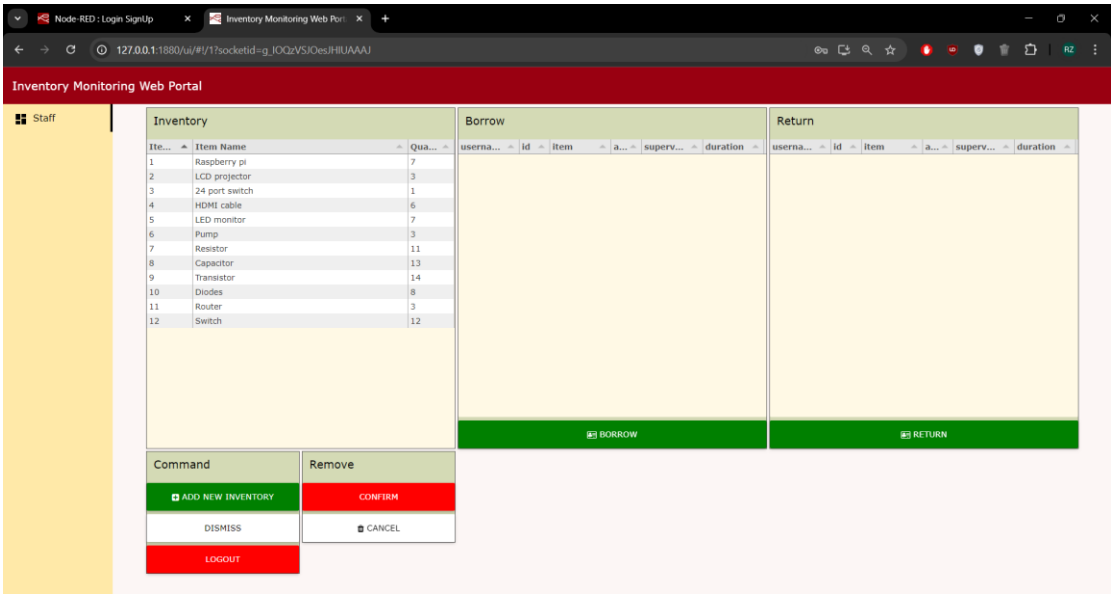


Figure 68 Select row, Remove item pop up

Figure 5.4.5.9

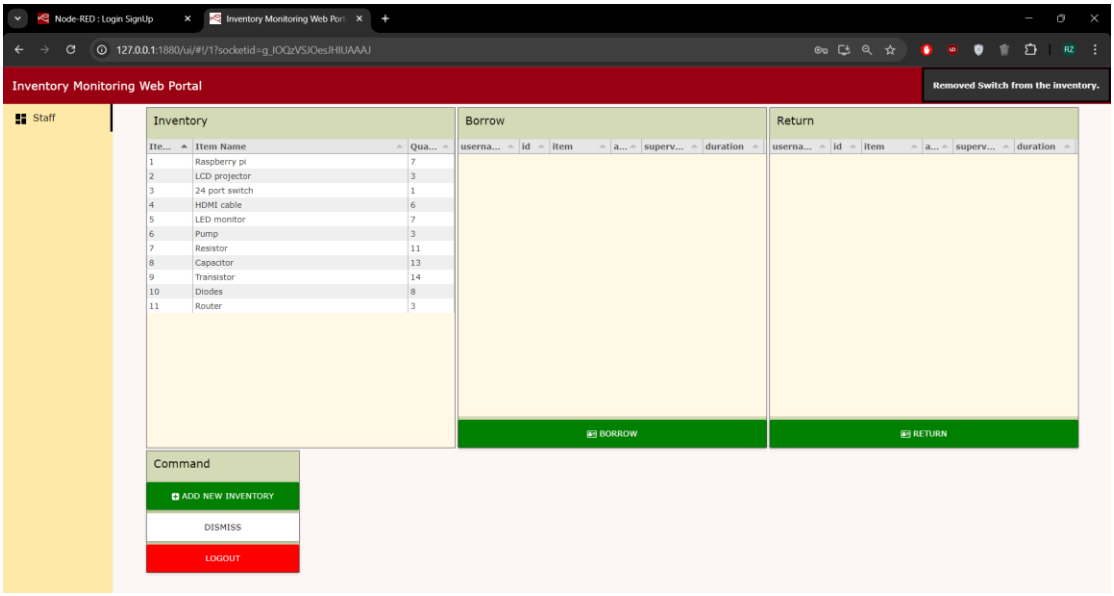


Figure 69 Removed item notification

Figure 5.4.5.10

	A	B	C
1	id	name	quantity
2	1	Raspberry	7
3	2	LCD projec	3
4	3	24 port swi	1
5	4	HDMI cabl	6
6	5	LED monit	7
7	6	Pump	3
8	7	Resistor	11
9	8	Capacitor	13
10	9	Transistor	14
11	10	Diodes	8
12	11	Router	3
13			

Figure 70 Updated inventory.csv with removed selected item

Figure 5.4.5.11

Figure 5.4.5.12 onwards show the borrow request and return request operations. Figures 5.4.5.12, 5.4.5.13 and 5.4.5.14 show the user_info.csv, request.csv and inventory.csv before the borrow and return operations were performed.

	A	B	C	D	E	F
1	username	password	role	status	borrowed	supervisor
2	staff	U2FsdGVk	staff			
3	drlee	U2FsdGVk	supervisor			
4	james	U2FsdGVk	student	approved	0	drlee
5	brandon	U2FsdGVk	student	approved	0	drtan
6	staff2	U2FsdGVk	staff			
7	bodhi	U2FsdGVk	student	approved	0	drlim
8	drtan	U2FsdGVk	supervisor			
9	drlim	U2FsdGVk	supervisor			
10	junjie	U2FsdGVk	student	none	0	drlee
11	jiale	U2FsdGVk	student	none	0	drlim
12						

Figure 71 User status in user_info.csv before staff borrow

Figure 5.4.5.12

	A	B	C	D	E	F	G	H	I	J	K
1	username	id	item	amount	status	supervisor	start date	enddate	start time	endtime	duration
2	bodhi	10	Diodes	3	approved	drlim	12/9/2024		23:46:54		
3	bodhi	11	Router	1	In Progress	drlim	6/9/2024		12:31:50		
4	brandon	7	Resistor	4	approved	drtan	12/9/2024		23:47:13		
5	james	8	Capacitor	3	In Progress	drlee	7/9/2024		15:52:30		
6	james	7	Resistor	2	approved	drlee	12/9/2024		23:32:40		
7											

Figure 72 User status in request.csv before staff borrow

Figure 5.4.5.13

	A	B	C
1	id	name	quantity
2	1	Raspberry	7
3	2	LCD projec	3
4	3	24 port swi	1
5	4	HDMI cabl	6
6	5	LED monit	7
7	6	Pump	3
8	7	Resistor	11
9	8	Capacitor	13
10	9	Transistor	14
11	10	Diodes	8
12	11	Router	3
13			
14			
15			

Figure 73 inventory.csv before staff borrow

Figure 5.4.5.14

Figure 5.4.5.15 shows the staff interface with new borrow requests that were approved by supervisors. Figure 5.4.5.16 shows the “Item has been borrowed successfully” notification after the staff clicks a request and clicks “Borrow” button underneath. The borrow request was also removed from the Borrow field and appended in Return field. Figure 5.4.5.17, 5.4.5.18, 5.4.5.19 show the updated contents of inventory.csv, request.csv and user_info.csv after the staff performed the borrow operation.

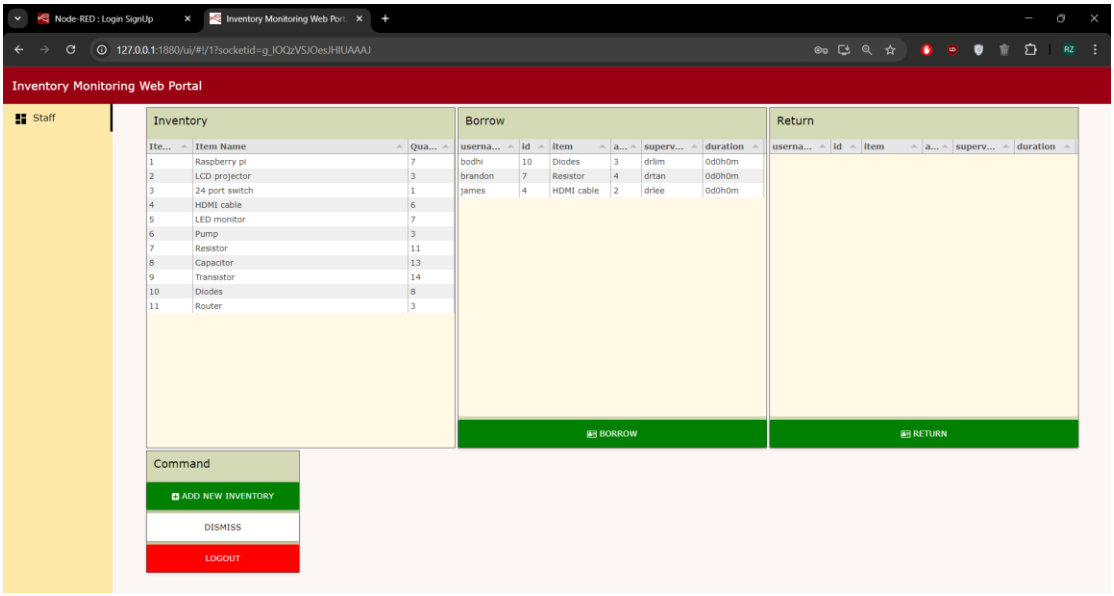


Figure 74 Borrow request

Figure 5.4.5.15

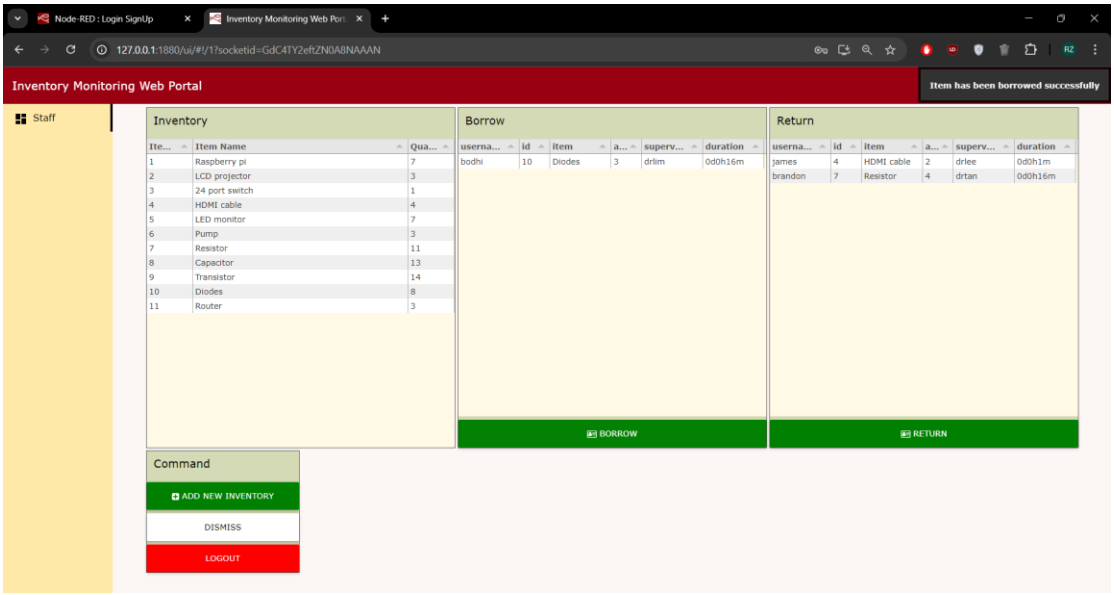


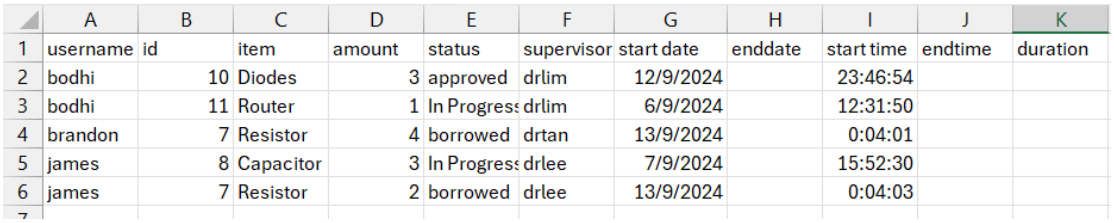
Figure 75 Item borrowed notification

Figure 5.4.5.16

	A	B	C
1	id	name	quantity
2	1	Raspberry	7
3	2	LCD projec	3
4	3	24 port swi	1
5	4	HDMI cabl	6
6	5	LED monit	7
7	6	Pump	3
8	7	Resistor	5
9	8	Capacitor	13
10	9	Transistor	14
11	10	Diodes	8
12	11	Router	3
13			

Figure 76 Updated inventory.csv after staff borrow

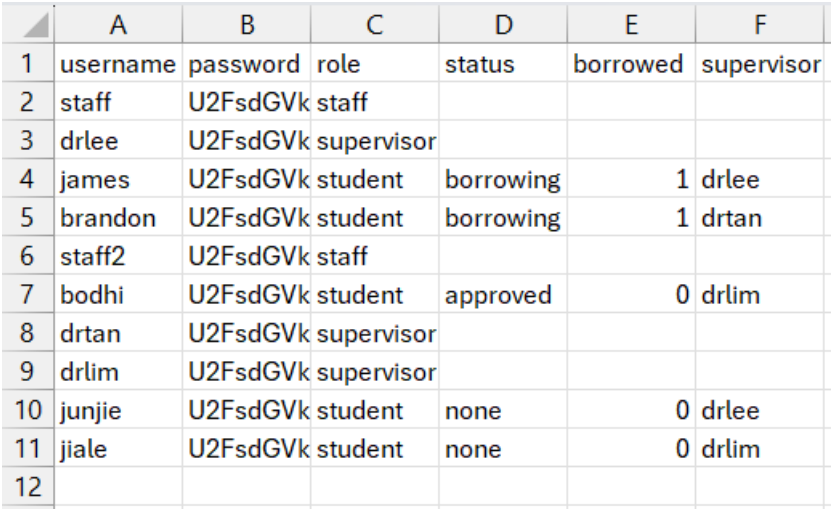
Figure 5.4.5.17



	A	B	C	D	E	F	G	H	I	J	K
1	username	id	item	amount	status	supervisor	start date	enddate	start time	endtime	duration
2	bodhi	10	Diodes	3	approved	drlim	12/9/2024		23:46:54		
3	bodhi	11	Router	1	In Progress	drlim	6/9/2024		12:31:50		
4	brandon	7	Resistor	4	borrowed	drtan	13/9/2024		0:04:01		
5	james	8	Capacitor	3	In Progress	drlee	7/9/2024		15:52:30		
6	james	7	Resistor	2	borrowed	drlee	13/9/2024		0:04:03		

Figure 77 Updated user status and borrow date and time in request.csv after staff borrow

Figure 5.4.5.18



	A	B	C	D	E	F
1	username	password	role	status	borrowed	supervisor
2	staff	U2FsdGVk	staff			
3	drlee	U2FsdGVk	supervisor			
4	james	U2FsdGVk	student	borrowing	1	drlee
5	brandon	U2FsdGVk	student	borrowing	1	drtan
6	staff2	U2FsdGVk	staff			
7	bodhi	U2FsdGVk	student	approved	0	drlim
8	drtan	U2FsdGVk	supervisor			
9	drlim	U2FsdGVk	supervisor			
10	junjie	U2FsdGVk	student	none	0	drlee
11	jiale	U2FsdGVk	student	none	0	drlim
12						

Figure 78 Updated student status and borrowed amount in user_info.csv

Figure 5.4.5.19

Figure 5.4.5.20 shows the “Item has been returned successfully” which will be prompted after the staff selects a return request and clicks “Return” button. Figure 5.4.5.21, 5.4.5.22, 5.4.5.23 show the updated contents in inventory.csv, request.csv and user_info.csv after the staff performed return operation. The item quantity of the returned item will be added back in inventory.csv, the request.csv will be updated with the user status to “returned” and the return date, time and borrow duration generated.

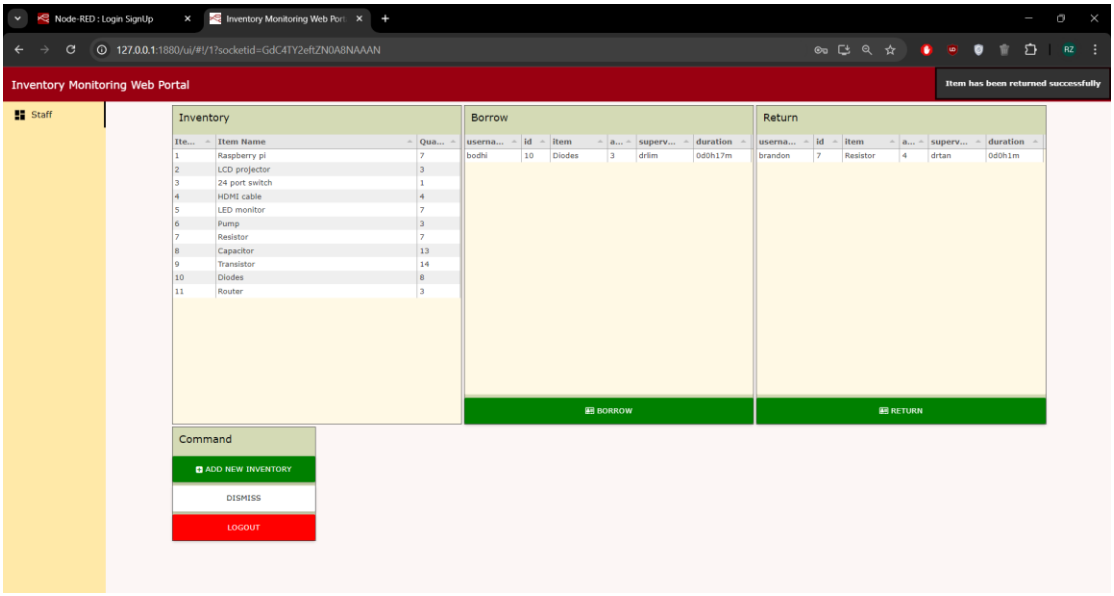


Figure 79 Item returned notification

Figure 5.4.5.20

	A	B	C
1	id	name	quantity
2	1	Raspberry	7
3	2	LCD projec	3
4	3	24 port swi	1
5	4	HDMI cabl	6
6	5	LED monit	7
7	6	Pump	3
8	7	Resistor	7
9	8	Capacitor	13
10	9	Transistor	14
11	10	Diodes	8
12	11	Router	3
13			

Figure 80 Updated item quantity in inventory.csv after staff return

Figure 5.4.5.21

	A	B	C	D	E	F	G	H	I	J	K
1	username	id	item	amount	status	supervisor	start date	enddate	start time	endtime	duration
2	bodhi	10	Diodes	3	approved	drlim	12/9/2024		7:34:24		
3	bodhi	11	Router	1	In Progress	drlim	6/9/2024		12:31:50		
4	brandon	7	Resistor	4	borrowed	drtan	12/9/2024		7:51:11		
5	james	8	Capacitor	3	In Progress	drlee	7/9/2024		15:52:30		
6	james	7	Resistor	2	In Progress	drlee	8/9/2024		9:01:06		
7	junjie	9	Transistor	2	In Progress	drlee	8/9/2024		12:30:30		
8	james	4	HDMI cabl	2	returned	drlee	12/9/2024	12/9/2024	7:49:18	7:52:23	0d0h3m
9											

Figure 81 Updated request.csv with return date, time and duration

Figure 5.4.5.22

	A	B	C	D	E	F
1	username	password	role	status	borrowed	supervisor
2	staff	U2FsdGVk	staff			
3	drlee	U2FsdGVk	supervisor			
4	james	U2FsdGVk	student	none	0	drlee
5	brandon	U2FsdGVk	student	borrowing	1	drtan
6	staff2	U2FsdGVk	staff			
7	bodhi	U2FsdGVk	student	approved	0	drlim
8	drtan	U2FsdGVk	supervisor			
9	drlim	U2FsdGVk	supervisor			
10	junjie	U2FsdGVk	student	none	0	drlee
11	jiale	U2FsdGVk	student	none	0	drlim
12						

Figure 82 Updated user status in user_info.csv after staff return

Figure 5.4.5.23

Chapter 6 System Evaluation and Discussion

6.1 System Testing and Performance Metrics

6.1.1 Sign Up Use Case

Table 6.1.1

Table 17 system testing and performance metrics for Sign Up

Test Case	Expected Result	Actual Result	Pass/Fail
• Username already in user_info.csv	Display error message	Display error message	Pass
• Username not in user_info.csv	Sign-up successful, request saved to request.csv	Sign-up successful, request saved to request.csv	Pass
• Signed up username • Not approved by supervisor • Log In	Display error message	Display error message	Pass
• Signed up username • Sign up same username again	Display error message	Display error message	Pass
• Supervisor approved sign up request • Log In	Successful, direct user to the student page	Successful, direct user to the student page	Pass
• Supervisor approved sign up request • Sign up same username again	Display error message	Display error message	Pass

6.1.2 Log In Use Case

Table 6.1.2

Table 18 system testing and performance metrics for Log In

Test Case	Expected Result	Actual Result	Pass/Fail
• Username not found in user_info.csv • Password found in csv	Display Error Message	Display Error Message	Pass
• Username found in user_info.csv • Password not found in csv	Display Error Message	Display Error Message	Pass
• Username not found in user_info.csv • Password not found in csv	Display Error Message	Display Error Message	Pass
• Username found in user_info.csv • Password found in csv • Student role	Redirect to Student Page	Redirect to Student Page	Pass
• Username found in user_info.csv • Password found in csv • Staff role	Redirect to Staff Page	Redirect to Staff Page	Pass
• Username found in user_info.csv • Password found in csv • Supervisor role	Redirect to Supervisor Page	Redirect to Supervisor Page	Pass

6.1.3 Log Out Use Case

Table 6.1.3

Table 19 system testing and performance metrics for Log Out

Test Case	Expected Result	Actual Result	Pass/Fail
Student clicks “Logout” button	Redirect to Login page	Redirect to Login page	Pass
Staff clicks “Logout” button	Redirect to Login page	Redirect to Login page	Pass
Supervisor clicks “Logout” button	Redirect to Login page	Redirect to Login page	Pass

6.1.4 Student Borrow Use Case

Table 6.1.4

Table 20 system testing and performance metrics for Student Borrow

Test Case	Expected Result	Actual Result	Pass/Fail
Student borrow amount > existing item quantity	Display Error Message	Display Error Message	Pass
Student borrow amount == item quantity	Borrow Requested	Borrow Requested	Pass
0 < Student borrow amount <= existing item quantity	Borrow Requested	Borrow Requested	Pass
Student borrow amount = 0	Display Error Message	Display Error Message	Pass
Student borrow same item twice	Display Error Message	Display Error Message	Pass

6.1.5 Staff Add Inventory Use Case

Table 6.1.5

Table 21 system testing and performance metrics for Staff Add Inventory

Test Case	Expected Result	Actual Result	Pass/Fail
• Add existing item name • Input exact quantity	Display Error Message	Display Error Message	Pass
• Add existing item name • Input different quantity	Display Error Message	Display Error Message	Pass
• Add new item name	Successfully Added	Successfully Added	Pass
• Add new item name in full caps (ITEM)	Successfully Added	Successfully Added	Pass
• Add same item name but in lower caps (item) • (Test Case Sensitivity)	Display Error Message	Display Error Message	Pass

6.1.6 Staff Update Inventory Use Case

Table 6.1.6

Table 22 system testing and performance metrics for Staff Update Inventory

Test Case	Expected Result	Actual Result	Pass/Fail
• Update existing inventory • Different item name • Different quantity	Successfully Updated	Successfully Updated	Pass
• Update existing inventory • Same item name • Different quantity	Successfully Updated	Successfully Updated	Pass
• Update existing inventory • Different item name • Same quantity	Successfully Updated	Successfully Updated	Pass
• Update existing inventory • Same item name • Same quantity	Successfully Updated	Successfully Updated	Pass
• Update existing inventory • Same item name, full caps (ITEM)	Successfully Updated	Successfully Updated	Pass
• Select inventory • Input another existing inventory name	Display Error Message	Display Error Message	Pass

6.1.7 Staff Remove Inventory Use Case

Table 6.1.7

Table 23 system testing and performance metrics for Staff Remove Inventory

Test Case	Expected Result	Actual Result	Pass/Fail
• Select inventory • Click "Remove Inventory" • Click "Confirm	Successfully Removed	Successfully Removed	Pass
• Remove inventory • Add new inventory	Item ID does not substitute removed item's ID	Item ID does not substitute removed item's ID	Pass

6.1.8 Supervisor Sign Up Request Use Cases

Table 6.1.8

Table 24 system testing and performance metrics for Supervisor Sign Up Request

Test Case	Expected Result	Actual Result	Pass/Fail
• Supervisor approve sign up request	Approved user able to login	Approved user able to login	Pass
• Supervisor deny sign up request	Denied user invalid username	Denied user invalid username	Pass

6.1.9 Supervisor Borrow Request Use Cases

Table 6.1.9

Table 25 system testing and performance metrics for Supervisor Borrow Request

Test Case	Expected Result	Actual Result	Pass/Fail
• Supervisor approve borrow request	Successfully update dashboard and request.csv, user_info.csv	Successfully update dashboard and request.csv, user_info.csv	Pass

• Supervisor deny borrow request	Successfully update dashboard and request.csv, user_info.csv	Successfully update dashboard and request.csv, user_info.csv	Pass

6.1.10 Staff Borrow Use Case

Table 6.1.10

Table 26 system testing and performance metrics for Staff Borrow

Test Case	Expected Result	Actual Result	Pass/Fail
• Staff selects request • Click “Borrow” button	Selected request removed from Borrow field, added on Return field. Updated status in request.csv and user_info.csv	Selected request removed from Borrow field, added on Return field. Updated status in request.csv and user_info.csv	Pass

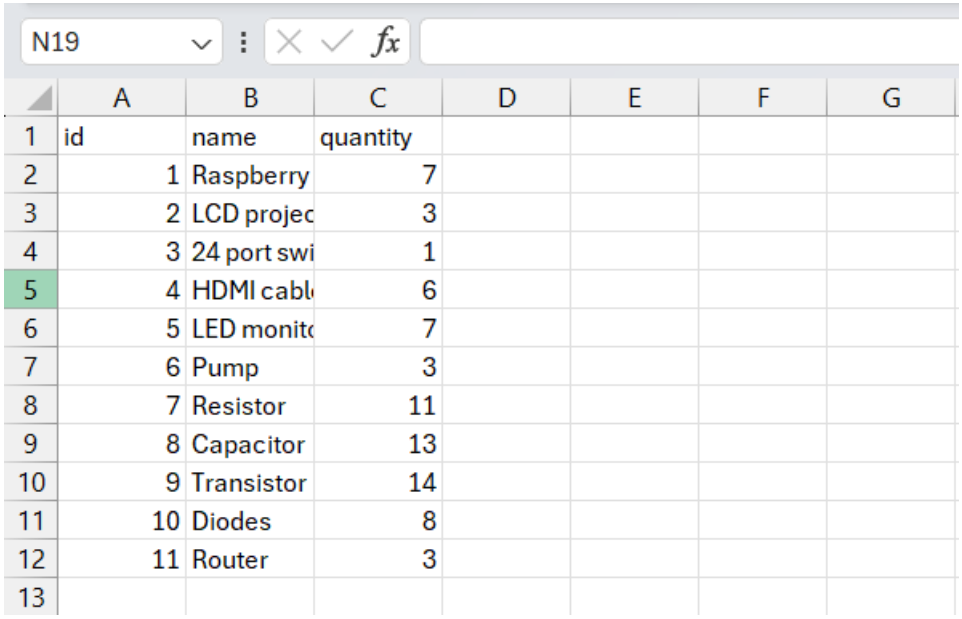
6.1.11 Staff Return Use Case

Table 6.1.11

Table 27 system testing and performance metrics for Staff Return

Test Case	Expected Result	Actual Result	Pass/Fail
• Staff selects request • Click “Return” button	Selected request removed from Return field. Updated status in request.csv and user_info.csv	Selected request removed from Return field. Updated status in request.csv and user_info.csv	Pass

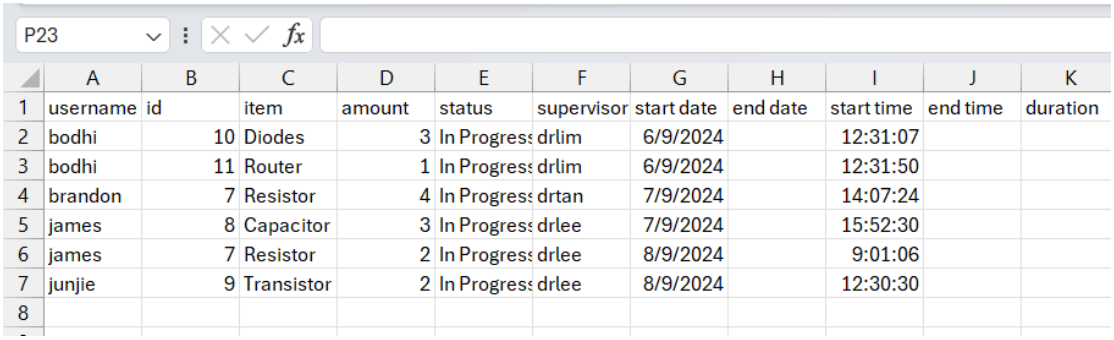
6.2 Testing Setup and Result



	A	B	C	D	E	F	G
1	id	name	quantity				
2	1	Raspberry	7				
3	2	LCD projec	3				
4	3	24 port swi	1				
5	4	HDMI cabl	6				
6	5	LED monit	7				
7	6	Pump	3				
8	7	Resistor	11				
9	8	Capacitor	13				
10	9	Transistor	14				
11	10	Diodes	8				
12	11	Router	3				
13							

Figure 83 set up inventory.csv content

Figure 6.2.1



	A	B	C	D	E	F	G	H	I	J	K
1	username	id	item	amount	status	supervisor	start date	end date	start time	end time	duration
2	bodhi	10	Diodes	3	In Progress	drlim	6/9/2024		12:31:07		
3	bodhi	11	Router	1	In Progress	drlim	6/9/2024		12:31:50		
4	brandon	7	Resistor	4	In Progress	drtan	7/9/2024		14:07:24		
5	james	8	Capacitor	3	In Progress	drlee	7/9/2024		15:52:30		
6	james	7	Resistor	2	In Progress	drlee	8/9/2024		9:01:06		
7	junjie	9	Transistor	2	In Progress	drlee	8/9/2024		12:30:30		
8											

Figure 84 set up request.csv content

Figure 6.2.2

N19

	A	B	C	D	E	F	G
1	username	password	role	status	borrowed	supervisor	
2	staff	U2FsdGVk	staff				
3	drlee	U2FsdGVk	supervisor				
4	james	U2FsdGVk	student	requested	0	drlee	
5	brandon	U2FsdGVk	student	requested	0	drtan	
6	staff2	U2FsdGVk	staff				
7	bodhi	U2FsdGVk	student	requested	0	drlim	
8	drtan	U2FsdGVk	supervisor				
9	drlim	U2FsdGVk	supervisor				
10	junjie	U2FsdGVk	student	requested	0	drlee	
11	jiale	U2FsdGVk	student	none	0	drlim	
12							

Figure 85 set up user_info.csv content

Figure 6.2.3

Three of the csv files were generated in Node-RED before starting the test cases. The test cases were also derived from the functional requirements stated in the previous chapters. The contents of the csv files were inspected before the starting of each test cases, and inspected right after each of the test cases was conducted.

	A	B	C
1	id	name	quantity
2	1	Raspberry	7
3	2	LCD projec	3
4	3	24 port swi	1
5	4	HDMI cabl	6
6	5	LED monit	7
7	6	Pump	3
8	7	Resistor	7
9	8	Capacitor	13
10	9	Transistor	14
11	10	Diodes	8
12	11	Router	3
13			

Figure 86 final inventory.csv content

Figure 6.2.4

The inventory.csv did not have any changes after all the test cases were done.

	A	B	C	D	E	F	G	H	I	J	K
1	username	id	item	amount	status	supervisor	start date	enddate	start time	endtime	duration
2	bodhi	10	Diodes	3	approved	drlim	12/9/2024		7:34:24		
3	bodhi	11	Router	1	In Progress	drlim	6/9/2024		12:31:50		
4	brandon	7	Resistor	4	borrowed	drtan	12/9/2024		7:51:11		
5	james	8	Capacitor	3	In Progress	drlee	7/9/2024		15:52:30		
6	james	7	Resistor	2	In Progress	drlee	8/9/2024		9:01:06		
7	junjie	9	Transistor	2	In Progress	drlee	8/9/2024		12:30:30		
8	james	4	HDMI cabl	2	returned	drlee	12/9/2024	12/9/2024	7:49:18	7:52:23	0d0h3m
9											

Figure 87 final request.csv content

Figure 6.2.5

The final request.csv content has some changes, mainly the user status and the ending date and time, with the calculated duration of the borrow.

	A	B	C	D	E	F
1	username	password	role	status	borrowed	supervisor
2	staff	U2FsdGVk	staff			
3	drlee	U2FsdGVk	supervisor			
4	james	U2FsdGVk	student	none	0	drlee
5	brandon	U2FsdGVk	student	borrowing	1	drtan
6	staff2	U2FsdGVk	staff			
7	bodhi	U2FsdGVk	student	approved	0	drlim
8	drtan	U2FsdGVk	supervisor			
9	drlim	U2FsdGVk	supervisor			
10	junjie	U2FsdGVk	student	none	0	drlee
11	jiale	U2FsdGVk	student	none	0	drlim
12						

Figure 88 final user_info.csv content

Figure 6.2.6

The final user_info.csv content also did not have much changes because the ending user state should be returning back to “none”.

Throughout the testing, the changes in the respective csv files were accurate and matched the expectation. It shows that the results were promising.

6.3 Project Challenges

In the implementation process of using Node-RED, one of the primary challenges was understanding the unique context and workflow of this visual programming tool. While the nodes and flows in Node-RED were visually represented and interconnected, the underlying processes differ significantly from traditional programming. Understanding how data flows between nodes, configuring node properties, and troubleshooting interactions within a visual environment required different mindset compared to conventional text-based coding. This adjustment involved learning new concepts specific to Node-RED, such as message passing, event-driven programming, and utilizing pre-built nodes effectively to achieve desired functionalities in the inventory monitoring web portal. The difficulties faced can be overcame by spending more time exploring and testing.

During the development of the Inventory Monitoring Web Portal using Node-RED, several challenges emerged that are distinct from traditional programming approaches.

One significant challenge was finding the correct palette to implement specific features, such as password encryption feature. Node-RED relies on pre-built palettes (node packages) to handle various functions, but with a vast library of palettes available, identifying the most suitable one often involved trial and error. Not all palettes are well-documented or optimized for specific tasks, which made it difficult to seamlessly integrate them into the system.

Another challenge was adjusting to flow-based logic and the way variable contexts are handled in Node-RED. Unlike traditional linear programming, where the flow of execution is more predictable, Node-RED uses a visual flow-based environment, which dictates the logic based on how nodes are connected. Managing variables was particularly tricky, as Node-RED allows variables to be scoped to the current message, flow, or global context. This required a shift in thinking compared to traditional programming, where variable scope is more explicit and easier to control. Inefficient use of these contexts could lead to performance bottlenecks or logical errors.

Scalability maintenance was also an issue. While it is easy to upscale the system, the maintenance poses as an issue to be solved. Node-RED is well-suited for smaller applications and prototypes, but as the system expands by incorporating more nodes, flows, and data sources, it becomes harder to manage. The visual nature of the flows can quickly become cluttered, making it troublesome to maintain and optimize.

A further difficulty was the limited debugging tools available in Node-RED. While it provides basic debugging nodes to trace data flow, these are not as robust or comprehensive as the debugging features available in traditional coding environments. Tracing errors in a visual flow environment is more time-consuming, especially as the system grows in complexity.

Customization limitations further complicated the development process. Node-RED heavily relies on existing nodes and palettes, and while these are useful for general purposes, they can restrict the ability to create more specialized or complex functionalities. In cases where no palette exists for a particular need, developers are forced to either create custom nodes or rely on external scripts, which can be more difficult than working with traditional programming languages that allow greater flexibility.

In summary, while Node-RED provides an accessible, low-code solution for the development Inventory Monitoring Web Portal, various challenges exist during the development process. These include identifying the correct palettes, managing flow-based logic, handling scalability issues, dealing with limited debugging tools, customizing functionalities, all of which required additional effort compared to a more traditional coding approach.

6.4 Objectives Evaluation

The objectives of designing a simple, cost-effective and user-friendly system were achieved. The main functional requirements of the main users which are student, staff and supervisor were also successfully achieved. The students were able to sign up for a new account, as well as request to borrow an item from laboratory in the system. The staffs were able to perform basic inventory monitoring operations such as adding new inventory, update existing inventory, remove existing inventory in the system. The staffs were also able to register the borrow and return of the items accurately in the system. The supervisors were able to approve and deny the borrow request and the new student account registration. All of the users were able to log in and log out of their account. Other than that, all the 11 use cases have passed the test cases conducted.

Chapter 7 Conclusion and Recommendations

7.1 Conclusion

The development of the Inventory Monitoring Web Portal using Node-RED has proven to be a great solution for addressing inefficiencies in traditional inventory management systems within academic laboratories. The project successfully met its objectives of designing a simple, cost-effective, and user-friendly system, tailored to the specific needs of university lab staff, students, and supervisors. The system enables students to sign up for new accounts and request to borrow items from the laboratory, while staff can manage inventory operations such as adding, updating, and removing items. Additionally, staff can register the borrowing and return of items, and supervisors have the authority to approve or deny borrow requests and student registrations. The project addressed 11 key use cases, all of which passed the necessary test cases, demonstrating the system's reliability and functionality.

Challenges encountered during development included understanding Node-RED's visual programming workflow, identifying the correct palettes for specific features, and managing flow-based logic and variable context. Scalability and maintenance also posed challenges, as Node-RED's visual nature made managing large, complex systems more difficult. Despite these obstacles, the system was implemented successfully by overcoming the learning curve, spending time testing, and optimizing flows. Additionally, customization limitations and basic UI components posed challenges, as Node-RED's pre-built UI nodes lacked the flexibility needed for fine-tuning the design, layout, and responsiveness.

7.2 Recommendations

The recommendations for future work include integrating the system with RFID (Radio Frequency Identification) technology that could significantly improve the efficiency of inventory tracking. By incorporating RFID readers and tags, lab items could be automatically scanned and tracked in real-time, reducing the need for manual input and minimizing the risk of errors. This would not only streamline the borrowing and returning process but also provide greater accuracy in monitoring inventory levels. Overall, while the project has successfully met its objectives, incorporating such technologies could further enhance the system's performance and usability.

The efficiency of the flows of the nodes and the JavaScript codes in the function could also be maximized by removing redundancy. The current nodes and flows were configured in such a way to make the system functions, the overall system optimizations were not prioritised since it is a simple system as of the completion of this project.

REFERENCES

[1] Rashmi P. Dagde, Vaishali Bhave, and Prajwal Mankar, “Review On Low Code/ No Code Software Development,” vol. 10, no. 10, Oct. 2023.

[2] L. Amaravadi, N. Palackal, and F. Garofolo, “Current and effective strategies for critical reagent characterization, storage, stability, retesting and life cycle management for ligand-binding assays and flow cytometry,” *Bioanalysis*, vol. 13, no. 10, pp. 737–740, May 2021, doi: 10.4155/bio-2021-0086.

[3] J. A. Feauto *et al.*, “Critical reagent inventory management system and web portal specifically optimized for supporting external clients,” *Bioanalysis*, vol. 14, no. 12, pp. 869–879, Jun. 2022, doi: 10.4155/bio-2022-0096.

[4] T. O. Oladele, R. O. Ogundokun, A. A. Adegun, E. A. Adeniyi, and A. T. Ajanaku, “Development of an inventory management system using association rule,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 3, p. 1868, Mar. 2021, doi: 10.11591/ijeecs.v21.i3.pp1868-1876.

[5] Sulihat Ibrahim-Imam, “The Design And Implementation Of An Online Inventory Monitoring System,” Jan. 2021.

[6] “SDLC Models.” https://www.javatpoint.com/software-engineering-sdlc-models

[7] Atlassian, “Waterfall Methodology: A Comprehensive Guide.”

[8] “Rapid application development model (RAD) – Software Engineering.”

[9] Sarah Laoyan, “What is Agile methodology? (A beginner’s guide).” https://asana.com/resources/agile-methodology

[10] Nishadha, “Use Case Diagram Relationships Explained with Examples.” https://creately.com/blog/diagrams/use-case-diagram-relationships

Appendix

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 2**
Student Name & ID: Tan Rui Zheng 20ACB04806	
Supervisor: Mr. Goh Hock Guan	
Project Title: Inventory Monitoring Web Portal Development Using Low Code Programming	

1. WORK DONE

Review and Revise Previous Work

2. WORK TO BE DONE

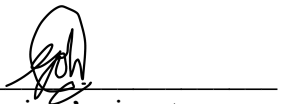
Review and revise FYP1 progress, define functional requirements, define user stories

3. PROBLEMS ENCOUNTERED

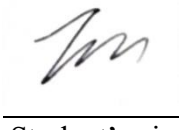
No problem

4. SELF EVALUATION OF THE PROGRESS

Doing well



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 4**
Student Name & ID: Tan Rui Zheng 20ACB04806	
Supervisor: Mr. Goh Hock Guan	
Project Title: Inventory Monitoring Web Portal Development Using Low Code Programming	

1. WORK DONE

Implement User Login Mechanism

2. WORK TO BE DONE

Implement User Login Mechanism

3. PROBLEMS ENCOUNTERED

No problem

4. SELF EVALUATION OF THE PROGRESS

Falling behind a little

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 6**
Student Name & ID: Tan Rui Zheng 20ACB04806	
Supervisor: Mr. Goh Hock Guan	
Project Title: Inventory Monitoring Web Portal Development Using Low Code Programming	

1. WORK DONE

Implement User Login Mechanism

2. WORK TO BE DONE

Implement User Login Mechanism

3. PROBLEMS ENCOUNTERED

No problem so far

4. SELF EVALUATION OF THE PROGRESS

Going well

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 8**
Student Name & ID: Tan Rui Zheng 20ACB04806	
Supervisor: Mr. Goh Hock Guan	
Project Title: Inventory Monitoring Web Portal Development Using Low Code Programming	

1. WORK DONE

Revise and Improve the Whole System

2. WORK TO BE DONE

Revise and Improve the Whole System

3. PROBLEMS ENCOUNTERED

Exist some logic error. Making sure no fatal errors causing incorrect output.

4. SELF EVALUATION OF THE PROGRESS

Not very good

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 10**
Student Name & ID: Tan Rui Zheng 20ACB04806	
Supervisor: Mr. Goh Hock Guan	
Project Title: Inventory Monitoring Web Portal Development Using Low Code Programming	

1. WORK DONE

System testing and Error Checking

2. WORK TO BE DONE

System testing and Error Checking

3. PROBLEMS ENCOUNTERED

Still a lot of issues faced during the testing process

4. SELF EVALUATION OF THE PROGRESS

Bad

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.: 12**
Student Name & ID: Tan Rui Zheng 20ACB04806	
Supervisor: Mr. Goh Hock Guan	
Project Title: Inventory Monitoring Web Portal Development Using Low Code Programming	

1. WORK DONE

Report Writing

2. WORK TO BE DONE

Report Writing

3. PROBLEMS ENCOUNTERED

No problem so far

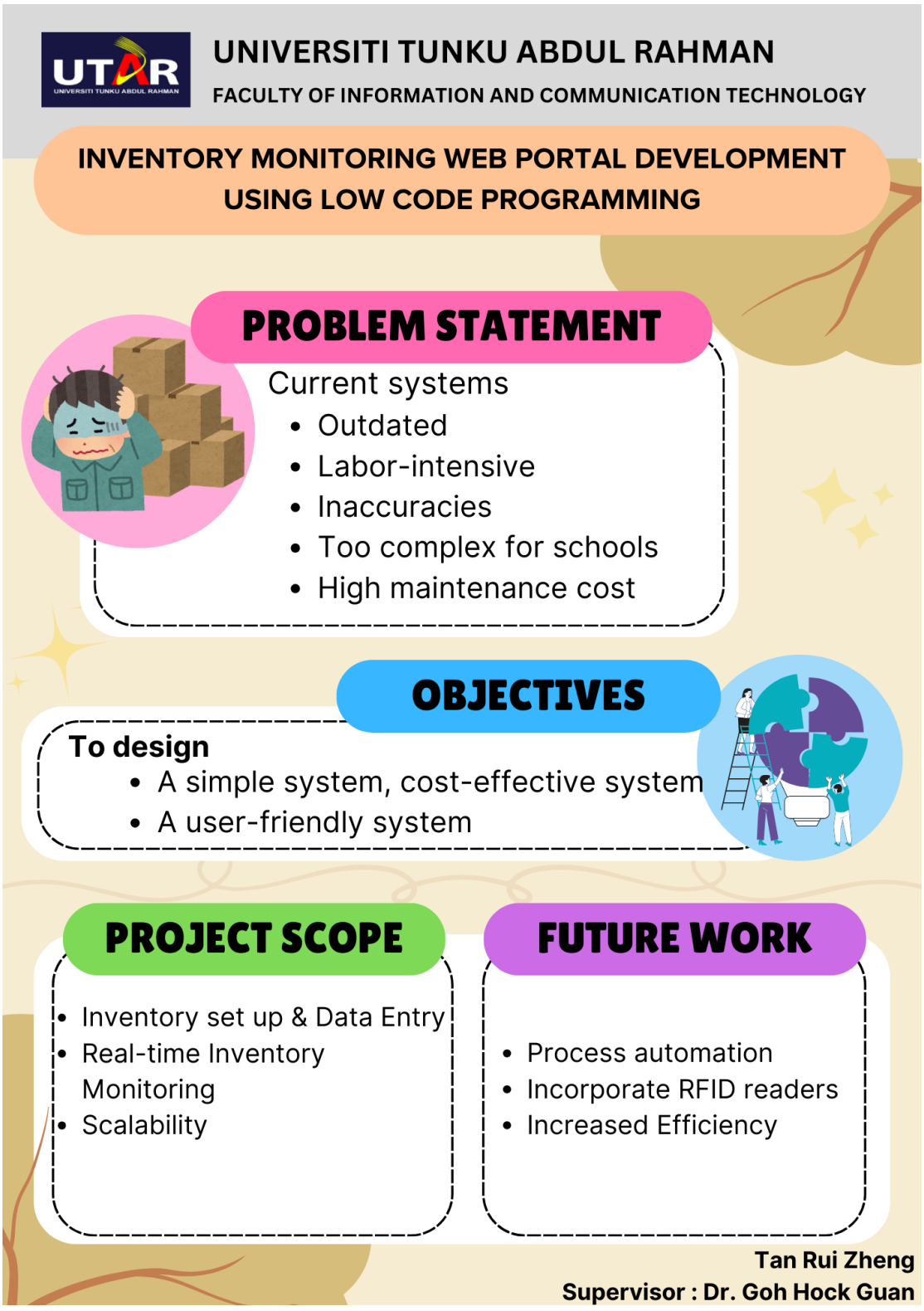
4. SELF EVALUATION OF THE PROGRESS

Doing well

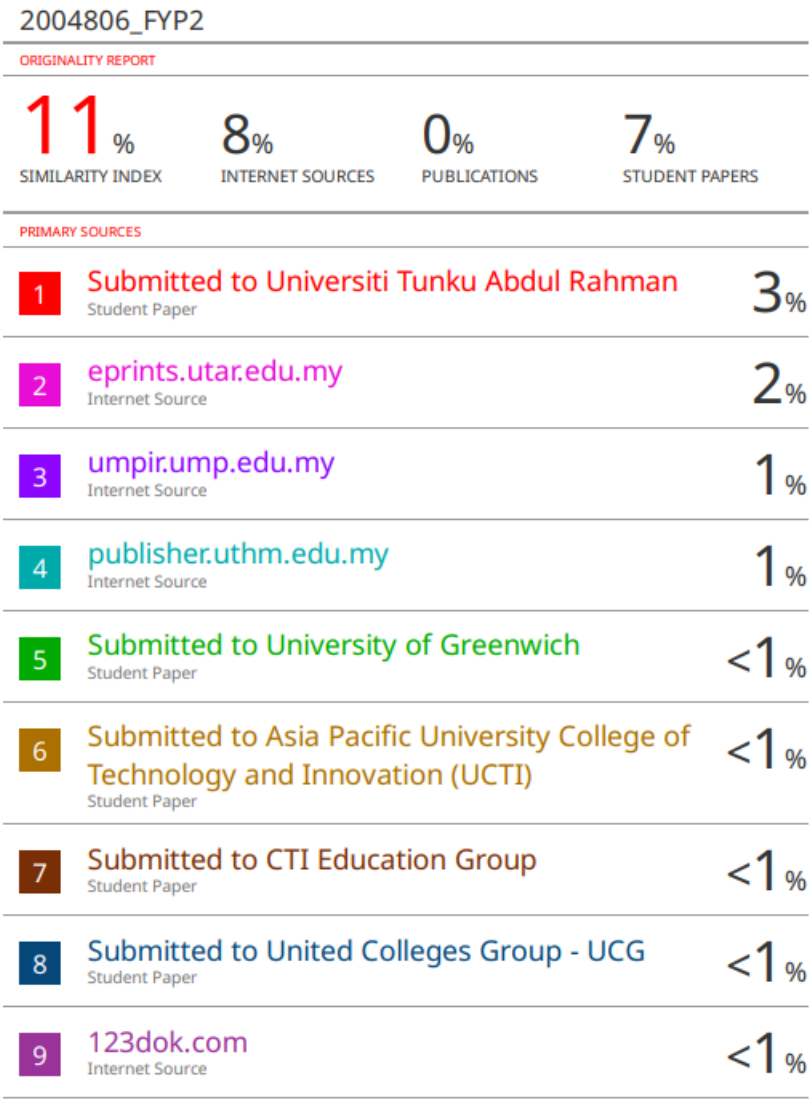
Supervisor's signature

Student's signature

Poster


UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
INVENTORY MONITORING WEB PORTAL DEVELOPMENT USING LOW CODE PROGRAMMING
PROBLEM STATEMENT
Current systems
Outdated
Labor-intensive
Inaccuracies
Too complex for schools
High maintenance cost
OBJECTIVES
To design
A simple system, cost-effective system
A user-friendly system
PROJECT SCOPE
Inventory set up & Data Entry
Real-time Inventory Monitoring
Scalability
FUTURE WORK
Process automation
Incorporate RFID readers
Increased Efficiency
Tan Rui Zheng
Supervisor : Dr. Goh Hock Guan

Plagiarism Check Result



2004806_FYP2

ORIGINALITY REPORT

11%	8%	0%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

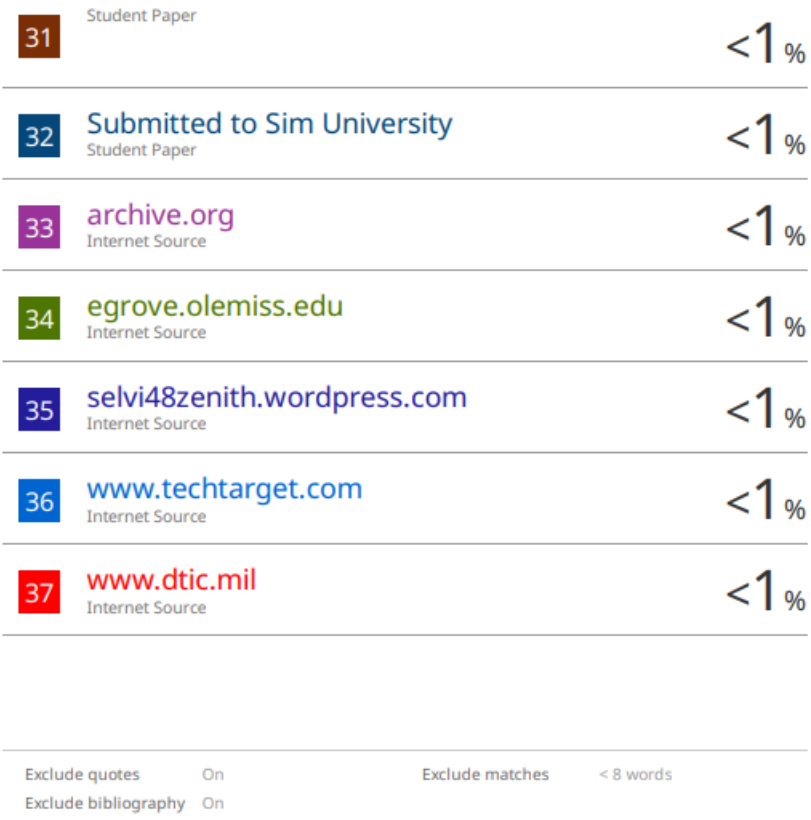
1	Submitted to Universiti Tunku Abdul Rahman Student Paper	3%
2	eprints.utar.edu.my Internet Source	2%
3	umpir.ump.edu.my Internet Source	1%
4	publisher.uthm.edu.my Internet Source	1%
5	Submitted to University of Greenwich Student Paper	<1%
6	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1%
7	Submitted to CTI Education Group Student Paper	<1%
8	Submitted to United Colleges Group - UCG Student Paper	<1%
9	123dok.com Internet Source	<1%

Appendix

10	Submitted to Bahrain Polytechnic Student Paper	<1 %
11	Submitted to Asia Pacific Instutute of Information Technology Student Paper	<1 %
12	Submitted to UNITEC Institute of Technology Student Paper	<1 %
13	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
14	Submitted to Eiffel Corporation Student Paper	<1 %
15	Submitted to Trident University International Student Paper	<1 %
16	Submitted to Universiti Tenaga Nasional Student Paper	<1 %
17	openjs.lfprojects.linuxfoundation.org Internet Source	<1 %
18	Submitted to Westcliff University Student Paper	<1 %
19	Submitted to University of East London Student Paper	<1 %
20	ir.unimas.my Internet Source	<1 %

Appendix

21	Submitted to Issaquah School District Student Paper	<1 %
22	Submitted to UOW Malaysia KDU University College Sdn. Bhd Student Paper	<1 %
23	Submitted to Dudley College Student Paper	<1 %
24	Submitted to King's Own Institute Student Paper	<1 %
25	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1 %
26	Submitted to Sunway Education Group Student Paper	<1 %
27	atrium.lib.uoguelph.ca Internet Source	<1 %
28	www.zoho.com Internet Source	<1 %
29	Submitted to KDU College Sdn Bhd Student Paper	<1 %
30	Ambily K K. "Azure DevOps for Web Developers", Springer Science and Business Media LLC, 2020 Publication	<1 %
	Submitted to MCAST	



31	Student Paper	<1 %
32	Submitted to Sim University Student Paper	<1 %
33	archive.org Internet Source	<1 %
34	egrove.olemiss.edu Internet Source	<1 %
35	selvi48zenith.wordpress.com Internet Source	<1 %
36	www.techtarget.com Internet Source	<1 %
37	www.dtic.mil Internet Source	<1 %

Exclude quotes On
Exclude bibliography On
Exclude matches < 8 words

Universiti Tunku Abdul Rahman			
Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Tan Rui Zheng
ID Number(s)	2004806
Programme / Course	CS
Title of Final Year Project	Inventory Monitoring Web Portal Development Using Low Code Programming

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 11 **%** **Similarity by source** Internet Sources: 8 % Publications: 0 % Student Papers: 7 %	
Number of individual sources listed of more than 3% similarity: 1	Template matching.
Parameters of originality required, and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Goh Hock Guan

Date: 13/9/2024

Signature of Co-Supervisor

Name: _____

Date: _____

Appendix

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student ID	20ACB04806
Student Name	Tan Rui Zheng
Supervisor Name	Dr. Goh Hock Guan

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
/	Title Page
/	Signed Report Status Declaration Form
/	Signed FYP Thesis Submission Form
/	Signed form of the Declaration of Originality
/	Acknowledgement
/	Abstract
/	Table of Contents
/	List of Figures (if applicable)
/	List of Tables (if applicable)
	List of Symbols (if applicable)
/	List of Abbreviations (if applicable)
/	Chapters / Content
/	Bibliography (or References)
/	All references in bibliography are cited in the thesis, especially in the chapter of literature review
/	Appendices (if applicable)
/	Weekly Log
/	Poster
/	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
/	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)
Date:13/9/2024