FORECASTING AND TRACKING OF PROJECT COMPLETION DATELINE USING REQUIREMENT TRACEABILITY MATRIX IN SCRUM

TAN ENG IAN

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

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

May 2024

DECLARATION

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

Signature :

Name : Tan Eng Ian

ID No. : 2003343

Date : 26/4/2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"FORECASTING AND TRACKING OF PROJECT COMPLETION DATELINE USING REQUIREMENT TRACEABILITY MATRIX IN SCRUM"** was prepared by **TAN ENG IAN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Honours) Software Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : D.Gunavathi

Supervisor : Gunavathi Duraisamy

Date : 17 May 2024

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor, Ms. Gunavathi a/p Duraisamy, for her guidance and support throughout the entire final year project period. Her supervision has provided me with valuable insights and helped clear my doubts while developing solutions for this project. I would also like to extend special thanks to Tee Wei Qi, a full-time UI/UX Designer whom I met during my internship. Tee Wei Qi's guidance on design principles and ideas greatly assisted me in designing the web application. Additionally, I am grateful to my friends who were willing to help test and provide feedback for my final year project.

ABSTRACT

Forecasting and tracking a project completion dateline plays a crucial role in project management and planning, as these are key factors in determining a project's success. SCRUM is an agile software development life cycle (SDLC) that is dynamic and flexible. In SCRUM methodology, the development life cycle is iterative, involving multiple iterations where each iteration focuses on a specific part of the system until its completion. This iterative nature accommodates dynamic requirements, as users are continuously involved in the software development process to ensure user satisfaction aligns with user requirements. After each iteration, or known as sprint, users evaluate the outcome, provide feedback, and necessary modifications are made to the requirements. As the requirements are changing over time, the problem of lack of formal documentations would also arise in the methodology itself. All these characteristics has made the process of forecasting and tracking the project completion dateline in SCRUM to be difficult.

To address the issue, this final year project involves developing a web application to assist project managers and SCRUM practitioners in managing their SCRUM projects. The web application will incorporate various tools and techniques to aid in tracking project completion timelines. Burndown and burnup charts will be used to visualize the pace of SCRUM projects. Additionally, the application will include a planning poker game to facilitate collaboration among team members in determining user story points for sprint backlog items. A Requirements Traceability Matrix (RTM) will be provided, summarizing the status of all SCRUM projects in a single table. Lastly, the project will implement an estimation model, an algorithm designed to predict the final completion date of SCRUM projects.

TABLE OF CONTENTS

DECLARATION **ii**

APPROVAL FOR SUBMISSION **iii**

ACKNOWLEDGEMENTS **v**

ABSTRACT **vi**

TABLE OF CONTENTS **vii**

LIST OF TABLES **xi**

LIST OF FIGURES **xiv**

LIST OF SYMBOLS/ABBREVIATIONS **xxvi**

LIST OF APPENDICES **xxvii**

CHAPTER

1 INTRODUCTION **1**

1.1 Introduction 1

1.2 Project Background 1

1.3 Problem Statement 4

1.3.1 Dynamic Requirements 4

1.3.2 Iterative Development 6

1.3.3 Lack of Formal Documentation 7

1.4 Objective 8

1.5 Project Scope 9

1.5.1 Modules Covered 9

1.5.2 Out of Scope 10

1.6 Project Solution 11

1.7 Project Approach 13

1.7.1 Requirements Gathering and Analysis 13

1.7.2 Quick Design 14

1.7.3 Building Prototype 14

1.7.4 User Evaluation 14

1.7.5 Refining Prototype 14

1.7.6 Implementation 15

1.8 Conclusion 15

2 LITERATURE REVIEW 16
2.1 Introduction 16
2.2 Software Development Methodology 16
2.2.1 Waterfall Methodology 17
2.2.2 SCRUM Methodology 21
2.2.3 Prototype Methodology 29
2.2.4 Comparison between SDLC Methodology 32
2.3 Existing Applications 34
2.3.1 Jira 34
2.3.2 Trello 42
2.3.3 Microsoft Excel and Google Sheets 48
2.3.5 SCRUMwise 53
2.3.6 Analysis of Existing Applications 61
2.4 Techniques and Approaches 64
2.4.1 Planning Poker 64
2.4.2 Requirement Traceability Matrix 69
2.5 Conclusion 71

3 METHODOLOGY AND WORK PLAN 72
3.0 Introduction 72
3.1 Prototype Methodology 72
3.1.1 Requirement Gathering and Analysis 72
3.1.2 Planning 73
3.1.3 Quick Design 74
3.1.4 Iteration 74
3.1.5 Implementation 75
3.2 Schedule 77
3.2.1 Schedule Table 77
3.2.2 Gantt Chart 80
3.3 Development Tools 81

4 PROJECT SPECIFICATION 84
4.1 Introduction 84
4.2 Functional Requirements 84
4.3 Non-Functional Requirements 86

4.4 Use Cases 87

4.4.1 Use Case Diagram 87

4.4.2 Use Case Description 89

5 SYSTEM DESIGN 114

5.1 Introduction 114

5.2 System Architecture and Design 114

5.2.1 System Architecture 114

5.2.2 Design Pattern 115

5.2.3 Combination of Architecture and Design Pattern 117

5.3 Database Design 118

5.3.1 Entity Relationship Diagram (ERD) 118

5.3.2 Database Dictionary 119

5.4 Prototype Design 126

6 SYSTEM IMPLEMENTATION 132

6.1 Introduction 132

6.2 File Structures 132

6.2.1 Vue.js 133

6.2.2 Laravel 137

6.3 Security Mechanism 141

6.3.1 Hashing 141

6.3.2 Cross-Origin Resource Sharing (CORS) 142

6.3.3 Cross-Site Request Forgery (CSRF) Token 144

6.3.5 JSON Web Token (JWT) 145

6.4 API Route 150

6.5 Modules 154

6.5.1 Login and Logout 154

6.5.2 Register 157

6.5.3 Project Management 159

6.5.4 Backlog 167

6.5.5 Requirement Traceability Matrix (RTM) 173

6.5.6 SCRUM Board 177

6.5.7 Sprint 182

6.5.8 Estimation 191

6.5.9 Project Completion Dateline Estimation 200

7 SYSTEM TESTING 206

7.1 Introduction 206

7.2 Test Objectives 206

7.3 Test Strategy 206

7.3.1 Entry Criteria 208

7.3.2 Exit Criteria 208

7.4 Unit Testing 209

7.5 Usability Testing 256

7.6 UAT Testing 260

8 CONCLUSION AND RECOMMENDATION 262

8.1 Conclusion 262

8.1.1 Limitation 262

8.2 Recommendation 263

REFERENCES 264

APPENDICES 268

LIST OF TABLES

Table 2.1: Comparison Matrix of SDLC

Table 2.2: Comparison Matrix between Existing Applications Reviewed

Table 3.1: Schedule of Final Year Project

Table 4.1: Functional Requirements

Table 4.2: Non-Functional Requirements

Table 4.3: Use Case Description of Login and Logout

Table 4.4: Use Case Description of Register

Table 4.5: Use Case Description of Create Project

Table 4.6: Use Case Description of Select and Manage Project

Table 4.7: Use Case Description of Add or Remove Project Member

Table 4.8: Use Case Description of Manage Project Member Role

Table 4.9: Use Case Description of Add or Remove Product Backlog Item

Table 4.10: Use Case Description of Edit Product Backlog Item

Table 4.11: Use Case Description of Add or Remove Sprint Backlog Item

Table 4.12: Use Case Description of Edit Sprint Backlog Item

Table 4.13: Use Case Description of Initiate and End Sprint

Table 4.14: Use Case Description of Customize SCRUM Board

Table 4.15: Use Case Description of Estimate Sprint Backlog Item

Table 5.1: Users Table

Table 5.2: Project Table

Table 5.3: ProjectStatus Table

Table 5.4: ProjectMember Table

Table 5.5: ProductBacklog Table

Table 5.6: SprintBacklog Table

Table 5.7: EndedSprintRecord Table

Table 5.8: Sprint Table

Table 5.9: PlanningPoker Table

Table 5.10: PlanningPokerSequence Table

Table 5.11: Note Table

Table 6.1: User Controller

Table 6.2: Project Controller

Table 6.3: Backlog Controller

Table 6.4: Planning Poker Controller

Table 7.1: Register Module Test Case

Table 7.2: Login and Logout Module Test Case

Table 7.3: Project Module

Table 7.4: Backlog Module Test Case

Table 7.5: RTM Module Test Case

Table 7.6: SCRUM Board Module Test Case

Table 7.7: Sprint Module Test Case

Table 7.8: Estimation Module Test Case

Table 7.9: Estimated Completion Date Test Case

Table 7.10: Usability Testing Scenario

Table 7.11: User Satisfaction Survey Result

Table 7.12: UAT Testing Summary

LIST OF FIGURES

Figure 1.1: Classification of Partial SDLC Methodology (Gurianov Myshenkov and Terakhov, 2023).

Figure 1.2: Overview of MVC Architecture (jaydeeosathwara272, 2022).

Figure 1.3: Prototyping Model SDLC Phases (Martin, 2023).

Figure 2.1: Overview of Waterfall Model (Bassil, 2012).

Figure 2.2: An overview of SCRUM Processes (Agile Arena, 2021).

Figure 2.3: Sample Physical SCRUM Board (Eby, 2016).

Figure 2.4: Prototyping Model SDLC Phases (Martin, 2023).

Figure 2.5: Jira Software Logo

Figure 2.6: Creation of Project

Figure 2.7: Project Selection Dashboard

Figure 2.8: Project Dashboard

Figure 2.9: Adding User

Figure 2.10: Successfully Adding User

Figure 2.11: List of User(s) in the Project

Figure 2.12: Backlogs

Figure 2.12: Assigning Backlogs

Figure 2.13: Story Point Estimation

Figure 2.14: Creation of Epic

Figure 2.15: Empty Sprint

Figure 2.16: Adding Backlogs to Sprint

Figure 2.16: Start Sprint

Figure 2.17: SCRUM Board

Figure 2.18: Timeline

Figure 2.19: Jira Report Option

Figure 2.20: Search Input

Figure 2.21: Search Result

Figure 2.22: Backlog Filter

Figure 2.23: SCRUM Board Filter

Figure 2.24: Trello Logo

Figure 2.25: Creating a Trello Board

Figure 2.26: Creating a Trello Board with Template

Figure 2.27: Empty Trello Board

Figure 2.28: SCRUM Template Board

Figure 2.29: Adding User

Figure 2.30: Creation of List

Figure 2.31: Creation of Card

Figure 2.32: Before Moving Task 1 to In Progress List

Figure 2.33: Moving Task 1 to In Progress List

Figure 2.34: Customization of Card

Figure 2.35: Filter Option

Figure 2.36: Power-Ups Interface

Figure 2.37: Microsoft Excel Logo

Figure 2.38: Google Sheets Logo

Figure 2.40: Default Excel Template

Figure 2.41: Default Google Sheets Template

Figure 2.42: Microsoft Excel Sample SCRUM Template

Figure 2.43: Google Sheets Sample SCRUM Template

Figure 2.44: Sharing of Microsoft Excel File

Figure 2.45: Sharing of Google Sheets

Figure 2.46: SCRUMWise Logo

Figure 2.47: Creation and Selection of Project

Figure 2.48: Creation of Team

Figure 2.49: Member Interfaces

Figure 2.50: Adding User

Figure 2.51: Successfully Adding User

Figure 2.51: List and Backlogs

Figure 2.52: Backlog Item Details

Figure 2.53: Creating a Sprint

Figure 2.54: Available Sprint Days

Figure 2.55: Backlog Interface

Figure 2.56: Adding Backlogs into Sprint

Figure 2.57: SCRUM Board

Figure 2.58: Burndown Chart

Figure 2.59: Metrics Available

Figure 2.60: Search Function

Figure 2.61: Filter Option

Figure 2.62: RE of Small Tasks without Prior Experience (Haugen, 2006).

Figure 2.63: RE of Large Tasks without Prior Experience (Haugen, 2006).

Figure 2.64: RE of Small Tasks with Prior Experience (Haugen, 2006).

Figure 2.65: RE of Large Tasks with Prior Experience (Haugen, 2006).

Figure 2.66: Requirement Traceability Matrix

Figure 3.1: Prototyping Model SDLC Phases (Martin, 2023).

Figure 4.1: SCRUM Web Application Use Case Diagram

Figure 5.1: Overview of MVC Architecture (jaydeeosathwara272, 2022).

Figure 5.2: System Architecture and Design of FYP Web Application

Figure 5.3: Entity Relationship Diagram

Figure 5.4: Login Page

Figure 5.5: Register Page

Figure 5.6: Project Page (No Project)

Figure 5.7: Project Page

Figure 5.8: User Management Page

Figure 5.9: Backlog Page

Figure 5.10: Planning Poker Page

Figure 5.11: Planning Poker Page (Revote)

Figure 5.12: Sprint Page (No Sprint)

Figure 5.13: Sprint Page

Figure 5.14: Sprint Page (Expanded)

Figure 5.15: RTM Page

Figure 5.16: Estimation Page

Figure 6.1: Component Folder

Figure 6.2: Layout Folder

Figure 6.3: View Folder

Figure 6.4: Store Folder

Figure 6.5: index.js File in Store Folder

Figure 6.6: Mixins Folder

Figure 6.7: projectState.js

Figure 6.8: Assets Folder

Figure 6.9: Snippet of API Routes of the Web Application

Figure 6.10: Controller Folder

Figure 6.11: Snippet of User Controller

Figure 6.12: Services Folder

Figure 6.13: Snippet of User Service

Figure 6.14: Contracts Folder

Figure 6.15: User Repository Contract

Figure 6.16: Repositories Folder

Figure 6.17: Snippet of User Repository

Figure 6.18: Repository Service Provider

Figure 6.19: Models Folder

Figure 6.20: Snippet of User Model

Figure 6.21: Hashing of Password

Figure 6.22: CORS Visualization (Kosaka, n.d.).

Figure 6.23: Middleware Registration

Figure 6.24: CORS Middleware

Figure 6.25: Axios CSRF Token Request

Figure 6.26: Laravel Environment Setup

Figure 6.27: Laravel Sanctum Configuration

Figure 6.28: Code Snippets of User Selecting Project

Figure 6.29: CheckJWTToken Middleware

Figure 6.30: Axios Interceptor

Figure 6.31: Check Creator Midddleware

Figure 6.32: Check Product Owner Middleware

Figure 6.33: Check SCRUM Master Middleware

Figure 6.34: Check SCRUM Master or Team Member Middleware

Figure 6.35: Login Page

Figure 6.36: Login Function (Vue.js)

Figure 6.37: Login Function (Laravel)

Figure 6.38: Empty Email or Password Field Validation

Figure 6.39: Project Page

Figure 6.40: Logout Function (Laravel)

Figure 6.41: Register Page

Figure 6.42: Empty Field Validation

Figure 6.43: Validation of Registration (Laravel)

Figure 6.44: Successful Registration Alert Message

Figure 6.45: Hashing of Password (Laravel)

Figure 6.46: Project Page (No Project Found)

Figure 6.47: Project Page

Figure 6.48: Create New Project Modal

Figure 6.49: Empty Invited User Field Validation

Figure 6.50: Successful Create Project Alert Message

Figure 6.51: Create Project Function (Laravel)

Figure 6.52: Project Page with New Project Created

Figure 6.53: Confirmation of Selecting Project

Figure 6.54: Backlog Page

Figure 6.55: Select Project Function

Figure 6.56: User Management Page

Figure 6.57: User Management Page (Non-Creator View)

Figure 6.58: SCRUM Master Filtering

Figure 6.59: Inviting New Project Member

Figure 6.60: Empty User List Validation

Figure 6.61: Successful Invite User to Project Alert Message

Figure 6.62: Confirmation of Removing Project Member

Figure 6.63: Edit Project Member Role

Figure 6.64: Empty Project Member Role Validation

Figure 6.65: Backlog Page (No Backlog)

Figure 6.66: Backlog Page

Figure 6.67: Add New Product Backlog

Figure 6.68: Empty Product Backlog Description Validation

Figure 6.69: Successful Create Product Backlog Alert Message

Figure 6.70: Confirmation of Delete Product Backlog

Figure 6.71: Edit Product Backlog

Figure 6.72: Product Backlog (Non-Product Owner View)

Figure 6.73: Product Backlog (Sprint Backlog involves in Active Sprint)

Figure 6.74: Sprint Backlog View

Figure 6.75: Add New Sprint Backlog

Figure 6.76: Empty Field Validation

Figure 6.77: Successful Create Sprint Backlog Alert Message

Figure 6.78: Confirmation of Delete Sprint Backlog

Figure 6.79: Edit Sprint Backlog

Figure 6.80: Sprint Backlog (Product Owner View)

Figure 6.81: Sprint Backlog (Sprint Backlog involves in Active Sprint)

Figure 6.82: RTM Page (No Backlog)

Figure 6.83: RTM Page

Figure 6.84: Product Backlog Search Result

Figure 6.85: Sprint Backlog Search Result

Figure 6.86: Get Backlogs Function (Vue.js)

Figure 6.87: Deep Clone RTM Function

Figure 6.88: Search Algorithm (Vue.js)

Figure 6.89: SCRUM Board Page (No Sprint Backlog)

Figure 6.90: SCRUM Board Page

Figure 6.91: SCRUM Board HTML Code (Vue.js)

Figure 6.92: SCRUM Board Drag and Drop Event (Vue.js)

Figure 6.93: Adding New SCRUM Board (New Status)

Figure 6.94: Empty SCRUM Board Validation

Figure 6.95: New SCRUM Board

Figure 6.96: Backlog Page

Figure 6.97: Sprint Backlog Status Selection

Figure 6.98: Customized SCRUM Board

Figure 6.99: Confirmation of Deleting SCRUM Board

Figure 6.100: Customized Status Sprint Backlog Set to To Do

Figure 6.101: Customize Board (Product Owner View)

Figure 6.102: Sprint Page (No Sprint)

Figure 6.103: Sprint Page

Figure 6.104: Active Sprint Exist

Figure 6.105: Completed Sprint

Figure 6.106: Active Sprint

Figure 6.107: Initiate Sprint

Figure 6.108: Empty Dates Validation

Figure 6.109: First Sprint

Figure 6.110: Second Sprint

Figure 6.101: Create Sprint Function (Laravel)

Figure 6.102: Completed Sprint

Figure 6.103: Active Sprint (Product Owner and Team Member View)

Figure 6.104: Active Sprint

Figure 6.105: Actual Effort

Figure 6.106: Actual Effort Validation

Figure 6.107: End Sprint Function (Vue.js)

Figure 6.108: End Sprint Function (Laravel)

Figure 6.109: Sprint Backlog

Figure 6.110: Sprint Backlog (Product Owner View)

Figure 6.111: Estimation Modal

Figure 6.112: Successful Update Estimation Alert Message

Figure 6.113: Estimation Effort Validation

Figure 6.114: Update Sprint Backlog Estimation Function (Laravel)

Figure 6.115: Planning Poker Page

Figure 6.116: Set Planning Poker Session (Vue.js)

Figure 6.117: Get Planning Poker Session (Laravel)

Figure 6.118: Notification Page

Figure 6.119: Notification Variable

Figure 6.120: Sticky Notes

Figure 6.121: Planning Poker Page

Figure 6.122: Planning Poker Session Ended

Figure 6.123: Revote Function (Laravel)

Figure 6.124: Different Estimation Chosen

Figure 6.125: Planning Poker Page

Figure 6.126: Change Sequence Number

Figure 6.127: Sequence Number Validation

Figure 6.128: Overall Burndown and Burnup Chart

Figure 6.129: Overall Burnup Chart

Figure 6.130: Summary Graphs

Figure 6.131: Sprint Summary

Figure 6.132: Get Sprint Summary Function (Vue.js)

Figure 6.133: Plot Overall Burnup Chart Function (Vue.js)

Figure 6.134: Plot Overall Burndown Chart Function (Vue.js)

Figure 7.1: Usability and UAT Testing with Tan Le Qie

Figure 7.2: Usability and UAT Testing with Ong Pei Kang

LIST OF SYMBOLS / ABBREVIATION

IDE	Integrated Development Environment
RTM	Requirement Traceability Matrix
SDLC	Software Development Life Cycle
WBS	Work Breakdown Structure

LIST OF APPENDICES

APPENDIX A: Usability Test Form 268

APPENDIX B: User Acceptance Test 276

CHAPTER 1

INTRODUCTION

1.1 Introduction

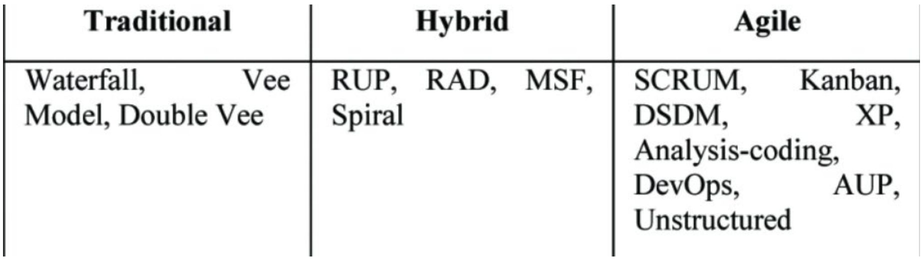
This project aims to tackle the challenge of tracking and forecasting project completion dateline in SCRUM by utilizing the usage of Requirement Traceability Matrix (RTM). This chapter offers an overview of the project, which includes the project background, the problem formulation, objectives, expected scope of the project outcome, the proposed solution and the proposed approach. Generally, the project aims to develop a web-based application using prototyping methodology that will allow the end-user to monitor their SCRUM projects progress.

1.2 Project Background

Forecasting and tracking a project completion dateline plays a crucial role in every project management and planning, as they are the key factors in determining a project's success. By defining the datelines, a well-planned schedule can be produced, which helps mitigate risks, plan resources, identify issues early on, and keep everyone involved updated to minimize the risk of failure (Indeed, 2022). Since projects can encounter unforeseen delays, tracking datelines becomes essential to ensure projects are completed within the desired time frame. Monitoring a project's progress provides a clear perception of how well the project is being executed and increases the accuracy of decision-making for the project processes (Gupta, 2023).

A Software Development Life Cycle or more commonly known by its abbreviation, SDLC, is an organized management system that incorporates specific deliverables at every stage, making it an integral part of every software development process in the current digital world. It is unavoidable for any development process not to adapt SDLC (Amazon, n.d.). Whether one is aware of SDLC or not, they will indirectly be utilizing it, even when working on their university projects. Over the years, more than 50 SDLC methodologies have been introduced. Gurianov, Myshenkov and Terakhov (2023) classified these methodologies into three different categories: Traditional, Hybrid, and Agile.

Each SDLC methodology possesses its own unique characteristics, along with specific advantages and disadvantages. For this project, our focus will be on the Agile methodology known as SCRUM.



Traditional	Hybrid	Agile
Waterfall, Vee Model, Double Vee	RUP, RAD, MSF, Spiral	SCRUM, Kanban, DSDM, XP, Analysis-coding, DevOps, AUP, Unstructured

Figure 1.1: Classification of Partial SDLC Methodology (Gurianov Myshenkov and Terakhov, 2023).

The main difference between the traditional methodology, often referred to as the Waterfall methodology, and agile methodology is their adaptability to changes. In the traditional approach, most elements are fixed and identified through a considerable amount of planning within the team. As it follows a sequential stage-by-stage process without having any parallel activities, a significant amount of time is allocated to establishing the requirements before proceeding to the next step (Bagiu, Avasilcăi and Rusu, 2022). This lack of dynamism and flexibility has led to a decline in its adoption. While the traditional waterfall model offers a higher level of assurance, it fails as an SDLC due to its requirement for almost perfect planning. In software development, uncertainties arise during the coding stage, making the necessity of stakeholder involvement being a must to address scope indecision. The inflexible nature of the waterfall model makes incorporating changes almost impossible after requirements are fixed in the planning stage, impacting schedules and budgets as the whole team must revert to the previous stages for replanning. Consequently, traditional models often do not meet stakeholder until the product is completed, leading to challenges in user acceptance sign-off (Lei et al., 2017).

With that being said, as a response to the downfall of traditional methodology, agile methodology, notably the SCRUM methodology has been introduced as a counterstrategy to fill in the void of traditional methodology.

However, it is important to note that there are many others more SDLC which are not agile but yet still succeed as its own. One of the greatest factors and the main topic of this project is the forecasting and tracking of project completion dateline.

SCRUM is one of the most flexible and dynamic methodologies currently available. In SCRUM, user involvement in ensuring the acceptability of the products is significantly evident compared to other SDLC methodology. After every sprint, which is also known as a period during which the development team produces a system based on the sprint backlog, the SCRUM methodology delivers a usable product to the stakeholders. This product is then reviewed during the sprint review session, enabling stakeholders to identify any uncertainties or problems within the products. Thus, the SCRUM team can redefine their product backlog to meet the user requirements. However, unlike other SDLC methodologies, which are often easier to forecast and track due to their detailed plans and signed requirements, the SCRUM methodology presents unique challenges of its own. As SCRUM typically focuses on 2 main artifacts for documentation, the product backlog and sprint backlog, the tracking of requirements becomes a bit challenging which makes the creation a detailed schedule to be difficult. Moreover, SCRUM's flexibility and dynamic characteristics comes from the continuous user involvement in updating and evolving product requirements, driven by stakeholder feedback after each sprint. Furthermore, SCRUM departs from fixed scopes and predefined phases found in traditional methodologies, opting instead for iterative development with time-boxed sprints (Schwaber and Sutherland, 2020). This approach makes it difficult to estimate the development progress as a whole. The dynamism of SCRUM makes creating a definitive schedule challenging as changes are always expected, rendering fixed schedules and Work Breakdown Structure (WBS) to be unnecessary.

Despite these challenges, SCRUM remains one of the most successful methodologies widely applied by organizations. According to a survey conducted by The State of Agile in 2017, a total of 56% of organizations worldwide have adopted the SCRUM methodology, while 27% have adopted a hybrid methodology involving SCRUM, which is known as SCRUMBUT (Petrova, n.d.). Its iterative nature offers numerous advantages in addressing various project challenges. However, the same characteristic has caused the precision of forecasting and tracking project completion dateline to be difficult. As challenges can be overcome, organizations are utilizing SCRUM while dealing with these obstacles. Hence, this final year project aims to overcome the problem of forecasting and tracking project completion timelines in SCRUM and enhance the visibility of timeline by utilizing the requirement traceability matrix.

1.3 Problem Statement

The SCRUM methodology is widely recognized as one of the most powerful and efficient SDLC methodologies available, being adopted by various organizations for its specialty in adapting to the dynamic pace of the software industry. However, while addressing the challenges faced by other SDLC approaches, SCRUM methodology has also introduced unique challenges of its own. In this part of the problem statement, the project will be focusing on the challenges of the SCRUM methodology in forecasting and tracking project completion dateline.

1.3.1 Dynamic Requirements

To fully grasp this problem, it is important to understand the concept of the sprint review in SCRUM. One of the key factors contributing to the success of the SCRUM methodology is the sprint review itself. At the conclusion of each sprint, the sprint team is responsible for delivering a workable product to the stakeholders, which is then reviewed during the sprint review. During the sprint review, the entire sprint team, which includes the SCRUM Master, Product Owner, and stakeholders, gather to observe the product demonstrations developed by the teams. The goal is to review the accomplishments of the sprint and assess whether if the sprint goal has been achieved. This process allows for uncertainties, dissatisfaction, or any required changes to be voiced out and discussed. The SCRUM Master plays a vital role in this process, as they will refine the product backlog based on the feedback received from stakeholders. This ensures that the product backlog is updated and ready to meet the expectations of the stakeholders in the next sprint (Schwaber and Sutherland, 2020).

This process stands out as a unique characteristic of SCRUM since it actively involves users throughout the development process. This high level of engagement with stakeholders contributes to their satisfaction with the final product which making most of the user acceptance sign-off to be achieved. However, the dynamic changing of requirement has let the monitoring of SCRUM project dateline to be hard as scope creep or refinement are being done all the time (Kumar, 2023).

In traditional project management methodologies, the scope is usually fixed, and monitoring processes are facilitated by utilizing various tools and techniques to create a well-defined schedule. Following this schedule allows for easy tracking and forecasting of the project completion dateline. However, in SCRUM, user involvement is significantly higher compared to traditional methodologies. The potentials of user involvement have led to the constantly changing of requirements, which sets SCRUM apart from traditional approaches. Consequently, fixing a schedule becomes challenging as the scope keep evolving, resulting in the addition of more tasks. Attempting to create a schedule based on evolving requirements is an unnecessary task due to the rapid pace of SCRUM, where documentation is not a primary focus, and excessive documentation can be perceived as a waste of time as changes to these documents may be needed all the time and it may potentially become outdated and useless after a sprint (Džanić, Džanić and Toroman, 2022).

Additionally, stakeholders typically become involved only at the end of each sprint, which marks the end of the development phase. Although this is a good practice as user involvement in almost every stage may disrupt the progress, the delayed involvement of stakeholders makes it uncertain what changes they might request, causing uncertainty in future schedules and workload. Planning and forecasting become more challenging when stakeholders' requests are not known until later stages. In a survey conducted by Lei and his teams in 2017, 66.7% of the respondent has agreed that SCRUM can adapt to changes quickly while only 57.2% agreed that the scope of SCRUM projects is well-defined.

1.3.2 Iterative Development

Iterative development and dynamic requirements are interrelated as flexibility of changing requirements after each sprint in SCRUM has makes it to have iterative characteristic to fulfill the changes. As stated above, at the end of every sprint, the SCRUM Master will be refining the product backlog to ensure the expectations of stakeholders. The dynamic nature of requirements in SCRUM has introduced uncertainty in project planning and the dateline forecasting, making the exact number of sprints required to complete the project to vary. As the requirements changes frequently, the development process necessitates additional sprints to address these requested modifications. These change requests may continue to emerge until stakeholders are satisfied with the final product. As a result, the number of required changes remains unpredictable, leading to an iterative development approach that continues until the project's completion.

While SCRUM allows users to be involved in changing product requirements based on their needs, the sprint goals themselves are usually fixed and not meant to be changed under any circumstances. This approach ensures that the team remains focused on the sprint goal and concentrates solely on the agreed product and sprint backlog. According to Ozkan et al. (2022), SCRUM demonstrates adaptability to frequent requirement changes at the overall product level. However, during the sprint itself, there may be a lack of flexibility and predictability due to the fixed sprint goals and timeboxes. This can lead to project estimations being time-consuming and often inaccurate. Consequently, if any changes to the product are needed during the sprint, they must wait until the sprint ends, and the development team can then focus on addressing these changes in upcoming sprints, increasing the length of feedback loops and sprints needed.

As stated by Ozkan and his team, the dynamic changes in requirements are not the only factors that challenge the monitoring of project timelines in

SCRUM. Another key factor contributing to this challenge is the concept of timeboxes used in SCRUM. Timeboxing in SCRUM means that every activity within the framework must be completed within a fixed length of time, which may vary based on the development team's capabilities and estimation. The rapid pace of timeboxing can lead to inaccurate estimations as some discussions may be rushed, lacking the depth of understanding and thinking required. The main drawback of timeboxing in SCRUM is the potential insufficiency of time (Visual Paradigm, n.d.).

Unlike traditional methodologies, which often include buffer time for unforeseen circumstances, SCRUM allows for less buffer time. If the Definition of Done for a sprint is not achieved, the team is forced to either reschedule another sprint or adjust their sprint backlog to complete the remaining tasks, which may increase the number of sprints needed to reach the project's goals. In a survey conducted by Lei and his teams in 2017, only 52.4% of the respondent has agreed that SCRUM projects can deliver their projects on time according to schedule. With that said, SCRUM enables clear and precise tracking of sprint timeline because the scope is fixed within each sprint. However, managing the product backlog's dynamic nature can be challenging for the overall project making the completion dateline to be unknown.

1.3.3 Lack of Formal Documentation

As agile is designed to be a fast-paced methodology to solve short project timelines, requests and changes are often collected informally and incrementally. Short meetings and face-to-face communications are preferred over formal documentation for any matter. However, the lack of formal documentation may lead to challenges in maintaining and upgrading the system, resulting in uncertainty for any future development affecting the future sprint as a whole (Džanić, Džanić, and Toroman, 2022). At the same time, the absence of formal documentation may also bring upon to various problems that may not be immediately apparent or obvious, affecting the forecasting and tracking of project completion dateline. This problem is further supported by a study conducted by Budiman, Raharjo, and Suhanto in 2022. The study highlighted the lack of developers' understanding of requirements, insufficient traceability between user stories and requirements, and the improper documentation of test results and quality requirements as the significant difficulties in SCRUM.

According to the findings of Shahzad et al. (2021), requirement engineering is one of the challenges that is paired with SCRUM methodology as the absence of formal documentation has led to the issues in the visibility and prioritization of requirements. Consequently, the development team may face difficulties in comprehending the full scope of the project, resulting in inaccuracies in estimations. Furthermore, the lack of well-defined requirements in SCRUM can lead to a situation where the development team has an incomplete understanding of the necessary tasks, potentially leading to delays and rework. This insufficient of requirement traceability may also cause oversight regarding the dependencies between various product backlogs or sprint backlogs, making the additional time allocated to address these matters when arise being necessary. Without a doubt, the aforementioned factors indirectly contribute to the lengthening of feedback loops within the development process, as more sprints are often required to address these unforeseen delays and ambiguities, making the project completion dateline to be uncertain.

1.4 Objective

1. To study on the existing tools and techniques used to forecast and tracking the completion dateline of a project and analyze how it can be applied in the SCRUM methodology with the usage of RTM.

2. To develop a web-based application which able:
   - to allow user to generate RTM.
   - To allow user to create product backlogs and sprint backlogs.
   - to generate a digital SCRUM board automatically.
   - to track and monitor the progress of the SCRUM projects through RTM with the help of visualization tools.
   - to forecast the completion dateline of SCRUM projects.

3. To evaluate the effectiveness of the proposed system and technique through usability testing by achieving a satisfactory rate of 80%

1.5 Project Scope

In this project, a web-based application that utilizes the RTM to enhance the visibility of tracking and forecasting project completion datelines in SCRUM will be developed. The main target users of this application will be SCRUM practitioners, which can be categorized into three groups: SCRUM team members, SCRUM Masters, and Product Owners. Project managers will also be included as they are responsible in helping to manage the projects. Each of them will have access to the same information in the application if shared, but with different levels of authority, as detailed later.

1.5.1 Modules Covered

In this section, the main modules of the project, which highlight the core functionalities of the application will be discussed. While providing an overview of these modules, the project will not delve into the extensive details of the functions of the application such as validations and navigations.

1.5.1.1 Creation of SCRUM Project

By default, all users are categorized as project manager, granting them the authority to create multiple SCRUM projects as desired. Users can add team members to their projects or remove team members from the project. Each team member is assigned as a SCRUM member by default. The project manager can change the role of team members to any of the three choices: SCRUM Team Member, SCRUM Master, or Product Owner. One team member can have more than 1 role if desired to.

1.5.1.2 Backlog Items Management with RTM

In a SCRUM project, the two main artifacts that is visible in documentations and to focus on are the product backlog and sprint backlog. In the system application, the Product Owner have the ability to add items into the product backlog in the form of user stories. The product backlog can be further detailed by the SCRUM Master and SCRUM team members by adding the sprint backlog items. These backlog items will be used to generate a RTM table. Any changes to the backlogs will also be reflected on the RTM. Alternatively, users can make changes to the RTM, and it will be reflected on the backlogs.

1.5.1.3 Sprint Initiation

After creating the product backlog items and sprint backlog items, the SCRUM Master has the authority to create sprints by selecting the desired product backlog and sprint backlog items to be worked on the sprint. The system application will automatically generate a SCRUM board based on the sprint information. The main elements of the SCRUM board will be organized based on the status of the sprint backlog items: "To Do," "In Progress," and "Done." This SCRUM board serves as an illustrative visualization of the sprint progress and tasks.

1.5.1.4 Planning Poker Estimation

The web application will allow SCRUM team members and SCRUM Master to use Planning Poker to estimate the sprint backlog effort. The Planning Poker default sequences will be provided for the team to choose. However, this sequence can also be customized if desired to. The estimation effort unit shall also be declared by the team. With the estimation effort identified, a forecast of the project completion dateline will be generated.

1.5.1.5 Monitoring of Project

The monitoring of the project dateline will be through RTM along with other visualization tools such as burndown and burnup charts. These charts serve to showcase the overall progress of the sprint. For each sprint, dedicated burndown and burnup charts will also be provided, offering a visual representation of the sprint's progress. The RTM will also summarize all the project elements in a table, allowing the require traceability to be easy.

1.5.2 Out of Scope

The project will solely focus on forecasting and tracking project dateline in SCRUM projects without considering other SDLC methodologies. All information and application design will be centered around SCRUM. The primary objective is to enhance the visibility of monitoring progress in SCRUM projects by utilizing the RTM. The application will dynamically provide a rough estimation of the project's progress and end date based on the available estimation effort and requirement changes in future. It is essential to clarify that the project does not aim to predict the final project dateline using any machine learning algorithm. Instead, the focus is on providing a clear requirement tracing along with its completion datelines to the user to assist project stakeholders in understanding the project status and potential project completion dates.

1.6 Project Solution

In this project, the goal is to design a web-based application to develop the features and modules outlined in the project scope. The primary objective is to enhance the visibility of the monitoring process for SCRUM projects and to forecast and track the project completion datelines using RTM. To achieve this, Microsoft Visual Code will be utilized as the Integrated Development Environment (IDE) for editing the codes. The key technologies to be employed includes HTML, CSS, JavaScript, and PHP. The functions of each of the technology are as follow:

1. HTML will serve as the foundational template to create the structure and content of the web pages.
2. CSS will be responsible for decorating and styling the HTML code, making the presentation dynamic and visually appealing.
3. JavaScript will handle the dynamic behavior of the web application, allowing for interactive and responsive features.
4. PHP will be used predominantly for backend tasks, enabling server-side scripting, especially for interactions with the database.

It's important to note that the combination of these four tools is essential for the proper functioning of the web application. Without any one of them, the application would not work as intended.

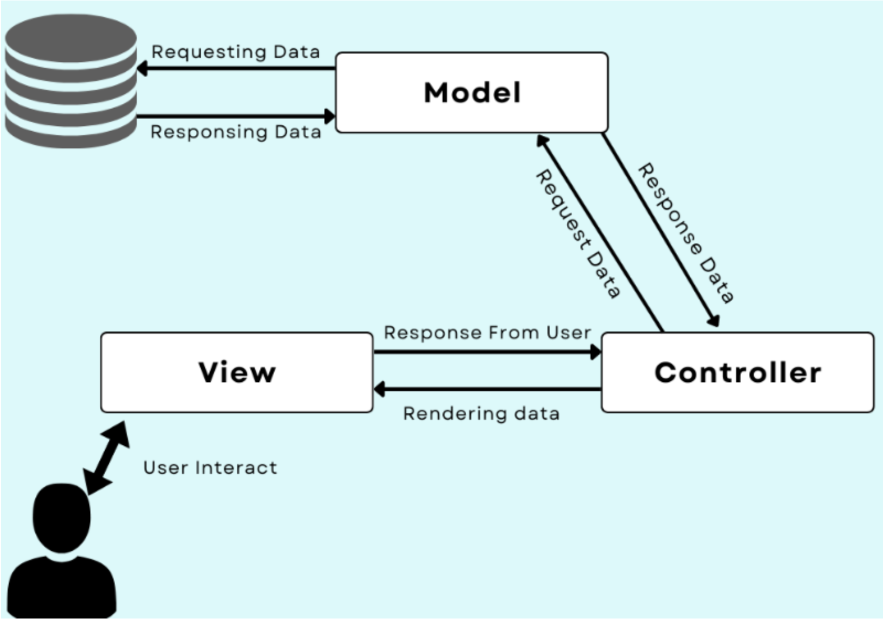


Figure 1.2: Overview of MVC Architecture (jaydeeosathwara272, 2022).

The system will be developed using the Laravel framework, which is an extension of PHP. Laravel implements the Model-View-Controller (MVC) architecture, providing higher flexibility in the development process and increasing efficiency due to its structured nature. Moreover, it enhances the system's security providing a more convincing system to the user. Additionally, Laravel's seamless API integration enables the creation of burndown and burnup charts with ease, making their implementation significantly easier reducing the complexity of development (jaydeepsathwara272, 2022).

In addition, Figma will be used to create the UI/UX design for the web application. Designing the UI/UX before commencing the development process enables us to grasp a comprehensive understanding of the system and minimizes the inconsistencies with the identified specifications and requirements. This approach helps to reduce the redundancy of the SDLC phases making changes to be minimum (Fox, n.d.).

By combining these tools and technologies, the project aims to create an efficient and comprehensive web application that facilitates the monitoring process of SCRUM projects, making it more accessible and visible to all stakeholders.

1.7 Project Approach

For this project, prototyping methodology has been chosen as the SDLC. According to tryQa (n.d.), the prototyping model is ideal when there are extensive user interactions involved. This will suit the project well as most of the features to be develop requires extensive user input. By leveraging the prototyping model, the project aims to gather end-user requirements through feedback obtained from users during the prototype evaluation phase. This approach will prioritize the user satisfaction and align the final product to the project goal. Generally, prototyping methodology consist of 6 SDLC phases: Requirements gathering and analysis, quick design, build prototype, user evaluation, refining prototype, implement and maintain (Martin, 2023). In this case, the maintain will not be applicable.

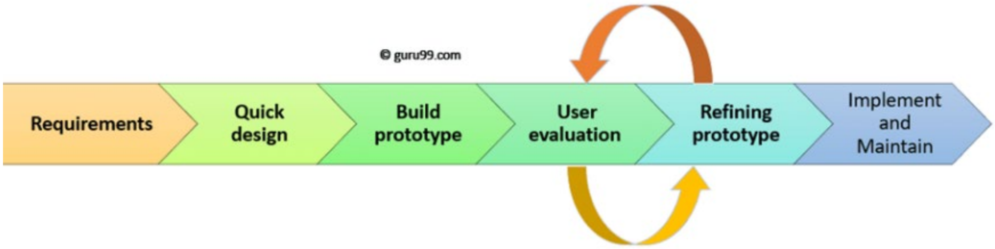


Figure 1.3: Prototyping Model SDLC Phases (Martin, 2023).

1.7.1 Requirements Gathering and Analysis

The first and most crucial phase will be requirement gathering and analysis, as the final product's success depends on the meeting user specifications. During this phase, several techniques will be employed to gather the requirements. Firstly, a meeting will be conducted with the FYP supervisor, Ms. Gunavathi, to initiate the requirements gathering process and establish a clear direction for the project. Once the problem has been identified and initial requirements have been drafted, the requirements will be further refined by referencing existing research or applications. This will help to enhance the project's scope and ensure alignment with established standards and best practices. To fully grasp the main idea of how the system functions, the project will utilize various formal documentation techniques, including use case diagrams, use case descriptions, and other relevant methods which will be cover in Chapter 4 of this project.

1.7.2 Quick Design

The second phase of the prototype model is quick design. Quick design involves creating a non-interactive prototype that showcases the overall UI/UX design of the web application to be developed. Based on the gathered requirements and analysis, Figma will be used to design the user interface. This process will provide a clear image and instructions to begin the coding process. Additionally, the design will help to identify the general features and functionality of the system, enabling us to understand how the front-end and back-end integration will work and providing a clear architectural view of the system.

1.7.3 Building Prototype

Based on the user interface designed in previous stage, a first version of prototype will be built. This initial prototype will provide a comprehensive overview of the proposed system, showcasing the general features and functions of the web application. While keeping validations to a minimum, the primary goal is to demonstrate the overall functionality of the system. The prototype will offer insights into the user interface and interactions, allowing user to visualize the intended user experience. As one of the characteristics of prototyping model is the iteration of refining prototyping based on user feedback, this first prototype will serve as a foundation for further development.

1.7.4 User Evaluation

The next phase of prototyping is user evaluation. During this stage, the built prototype will be presented to the end-users for review, and valuable feedback will be gathered to address any uncertainties regarding the web application's functionality. Notably, the FYP supervisor will serve as a key stakeholder in assessing the product's performance. However, the primary focus will be on evaluating the effectiveness of the technique used on monitoring the SCRUM project completion dateline using RTM.

1.7.5 Refining Prototype

After receiving feedback from the supervisor, enhancement will be made to the prototyping by refining the prototype model based on any changes in requirements. If the estimation technique used proves to be inappropriate or inaccurate, new techniques or refinements will be implemented to further enhance the accuracy of the monitoring process. The implementation of RTM will also be reviewed to verify if it is fully utilized. Subsequently, the revised version of prototype will be subjected to another round of user evaluation, allowing us to show the changes made and receive further feedbacks. This iterative process will continue until all uncertainties are addressed.

1.7.6 Implementation

The final stage of the prototyping phase will be the implementation phase. During this phase, the prototype will be finalized, transforming it into a proper web application ready for deployment. The primary focus will be on ensuring that the web application functions as intended and are consistent with the user requirements. In this phase, multiple tests will also be undertaken to assess the effectiveness of the designed software. Testing will be conducted to ensure that the web application will performs as intended and to identify any bugs or malfunctions in the software.

1.8 Conclusion

This final year project aims to forecast and track the project completion dateline using a requirement traceability matrix in the SCRUM framework. This is motivated by SCRUM's unique characteristics, notably dynamic requirements and iterative development, which result in fewer formal documentations, thereby complicating the process of monitoring project completion datelines. To address this challenge, the project proposes the development of a web-based application that enables SCRUM practitioners to utilize the requirement traceability matrix, enhancing the visibility of tracking and forecasting the project completion dateline. The project will employ prototyping development as its chosen SDLC approach.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, a literature review will be conducted on three different aspects: Software development methodologies, existing applications, and the techniques and approaches to be applied in this project. Each of these aspects aims to provide a better understanding of the references utilized to enhance and guide the project, while giving a better aid in gathering and understanding the requirements for developing the web application designed to address the complexities of the SCRUM methodology.

Section 2.2 will be focusing on reviewing the SDLC methodologies, a comparison between traditional and agile approaches. In Section 2.3, the focus will be on the review of existing applications. This will enable an exploration of the currently available applications, allowing the identification of their benefits and disadvantages, and how this project can these implement the existing features by introducing improvements. Finally, Section 2.4 will be reviewing on the techniques and approaches employed to solve the problem statement. This section will primarily focus on the requirement traceability matrix and Planning Poker methodologies.

2.2 Software Development Methodology

A software development methodology is a set of processes or phases implemented by developers that provides a systematic and organized approach to managing a software development process. These methodologies usually consist of phases known as the Software Development Life Cycle (SDLC), which typically describe the processes involved in developing software, starting from the requirement gathering phase and extending to the closure of the development phase. Each SDLC is unique, containing different variations of phases that cause them to behave differently from one another. These characteristics introduce their own advantages and disadvantages, resulting in none of them being perfect or the most suitable SDLC to implement. The decision of which methodology to implement heavily depends on characteristics of the development team, such as the time and resources allocated. Generally, SDLC can be categorized into three main types: Traditional, Hybrid, and Agile.

While both traditional and agile methodologies have their respective advantages and disadvantages, making a direct comparison between both at higher level can be overly broad. To provide a more specific and meaningful analysis, this literature review will focus on the specific SDLC methodologies from each approach. Specifically, the Waterfall methodology from the traditional approach and SCRUM methodology from the agile approach. The Prototype methodology will also be considered for comparison due to it arguably not belonging to any approaches. Given that this final year project centers on the SCRUM methodology, a deeper exploration into SCRUM will be undertaken.

2.2.1 Waterfall Methodology

The Waterfall methodology is one of the earliest SDLC model that was introduced into the software development process by Dr. Winston W. Royce back in 1970. The term "Waterfall" comes from the model's characteristic feature of sequential progression, where developers are required to complete previous phase before moving on to the next stage. This sequence flows downward, much like a waterfall. Due to its linear pattern, the Waterfall model makes it difficult to revert to the previous stage, as each phase requires flawless execution, making changes as a whole to be difficult. This has led to a recursive pattern in each phase, ensuring it is perfected before proceeding to the next step (Bassil, 2012).

2.2.1.1 Phases of Waterfall Models

Generally, the waterfall methodology consists of 5 different phases in a sequential pattern: Analysis, design, implementation, testing and maintenance.

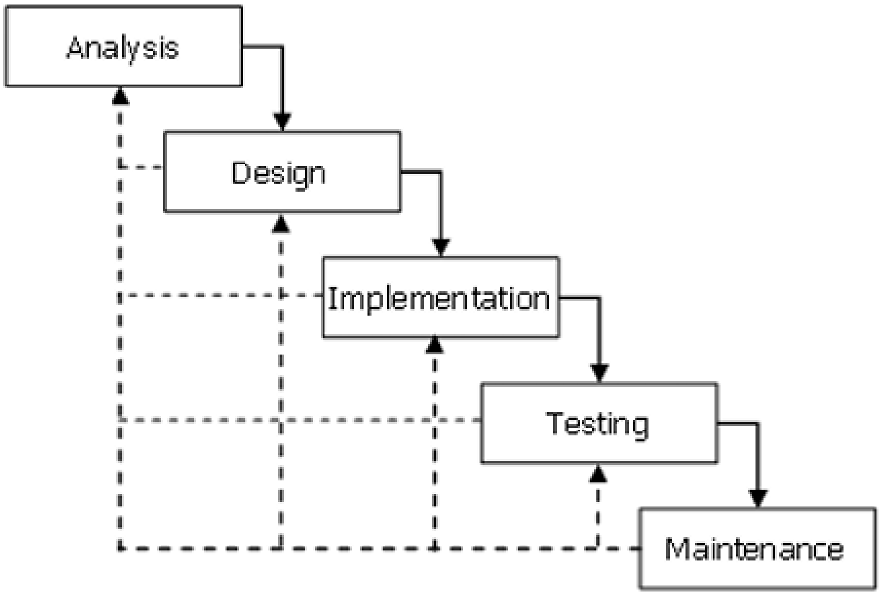


Figure 2.1: Overview of Waterfall Model (Bassil, 2012).

Analysis

Analysis or more commonly known as the requirement gathering phase is the initial phase of the Waterfall model in software development. In this phase, both functional and non-functional requirements for the system are gathered and analyzed. Various tools, including but not limited to use case diagrams, use case descriptions, sequence diagrams, data flow diagrams, are employed to understand the business logic and identify potential workflows for how the system should operate. The requirements identified through this analysis are then documented in a Software Requirements Specification (SRS). This document serves as a foundational reference for the entire development process, guiding developers to ensure that the final system aligns with the identified user requirements (Bassil, 2012).

Design

Referencing the previously defined SRS, the developers proceed to identify the technical requirements of the system. This phase can be described as the process of planning and problem-solving for a software solution. During this phase, the developers collaborate to create a plan for developing a solution for the system. The outcome of the design phase is a comprehensive design specification that includes both High-Level Design and Low-Level Design. This specification covers various aspects, including software architecture designs, database schema designs, algorithms to be employed, programming languages or frameworks to be used, and potentially graphical user interface design and data structure choices. Essentially, this design specification recorded the technical aspects derived from the requirements identified in the SRS (Bassil, 2012).

Implementation

Implementation is the phase where the actual coding of the system begins. Developers will create the system based on the requirements identified in the SRS and the design specification, transforming it into a functional system ready for deployment. This phase is typically the longest in the entire SDLC since it encompasses the crucial core of the project success — the creation of the actual system. It consists of stages like debugging, compiling, integration, and, most importantly, coding (Kramer, 2018).

Testing

This phase is referred to as the verification and validation stage of the developed system. Its purpose is to ensure that the system operates as intended and to identify any defects or bugs present. Additionally, this phase guarantees that the software aligns with the original requirements, specifications, and project goals. Testers and quality assurance engineers collaborate to formulate test cases, and the resulting test outcomes are documented. The objective of this phase is to explicitly test the system to uncover any bugs, defects, or glitches. Subsequently, the identified issues are addressed, and the system is refined accordingly (Kramer, 2018). This iterative cycle is repeated multiple times until no errors are detected by the testers, thus preparing the system for the next stage.

Maintenance

Depending on the nature of the project, the maintenance phase may be optional, but it often overlaps with the broader project life cycle. In this phase, the deployed system undergoes modifications to refine its functionality, fix bugs, and address any glitches that were not discovered during earlier testing. With the feedback collected after the system is deployed, the maintenance phase often includes upgrades designed to enhance the overall performance and quality of the system.

2.2.1.2 Advantages & Disadvantages of Waterfall Methodology

Advantages

1. The Waterfall methodology is straightforward to understand and implement. This simplicity is particularly beneficial for entry-level developers or inexperienced teams, enabling them to adapt quickly to the framework (Alshamrani & Bahattab, 2015).

2. The Waterfall methodology provides a clearly defined outcome due to its well-structured phases. Requirements gathering and analysis are well conducted the actual implementation. The sequential nature of the methodology ensures that each phase is completed one at a time, without parallel activities. This structure simplifies the entire SDLC and provides clear milestones (UKEssays, 2018).

3. Waterfall places a significant emphasis on documentation at each phase, leading to clear and precise requirements. This focus enhances the visibility of the project timeline and allows for more accurate and straightforward timeline predictions (Bagiu, Avasilcăi and Rusu, 2022).

Disadvantages

1. The Waterfall methodology generally only involve stakeholders in the requirements gathering phase and post-implementation review. This limited engagement can lead to a misalignment between the developed product and stakeholder expectations, as there's little opportunity for stakeholder feedback during the implementation phase itself (Alshamrani & Bahattab, 2015).

2. The Waterfall methodology is not well-suited for projects with frequently changing requirements. It demands almost a perfect planning for each stage and offers little room for alterations once the project is

phase is executed. As a result, any revisions can be extremely costly in terms of time and resources (Bagiu, Avasilcăi & Rusu, 2022).

3. In Waterfall, testing occurs only after the entire system has been implemented. This late-stage testing may potentially bring upon multiple issues, including functionalities that do not work as intended. Such late discoveries can necessitate rolling back to previous stages, significantly increasing the project's overall costs and timeframes (Bagiu, Avasilcăi & Rusu, 2022).

2.2.2 SCRUM Methodology

SCRUM is one of the most successful methodologies in the software industry, being adopted by 56% of the organization worldwide (The State of Agile, 2017). It's an Agile approach that focuses on short delivery timelines for projects, employing both iterative and incremental techniques to manage unforeseen risks. This allows development teams to adapt quickly to changing requirements. SCRUM is often used in projects where the requirements are either unclear or likely to change.

Originating in the early 1990s, the concept of SCRUM was first introduced by Takeuchi and Nonaka in their article back in 1986 titled "The New New Product Development Game". It was later implemented and formalized by Jeff Sutherland, John Scumniotales, and Jeff McKenna in 1993, drawing inspiration from that article. The term "SCRUM" comes from the sport of rugby. SCRUMmage, or simply known as SCRUM, refers to how players in rugby pack closely together in an attempt to gain possession of the ball. Similar to how players in a rugby SCRUM work together to gain possession, developers in SCRUM collaborates to move a project towards succession (Bultmann, n.d.).

In this section, SCRUM will be reviewed from it is applied in the industry, which includes how it works, the formation of SCRUM team, the artifacts of SCRUM, and its advantages and disadvantages.

2.2.2.1 How SCRUM Works?

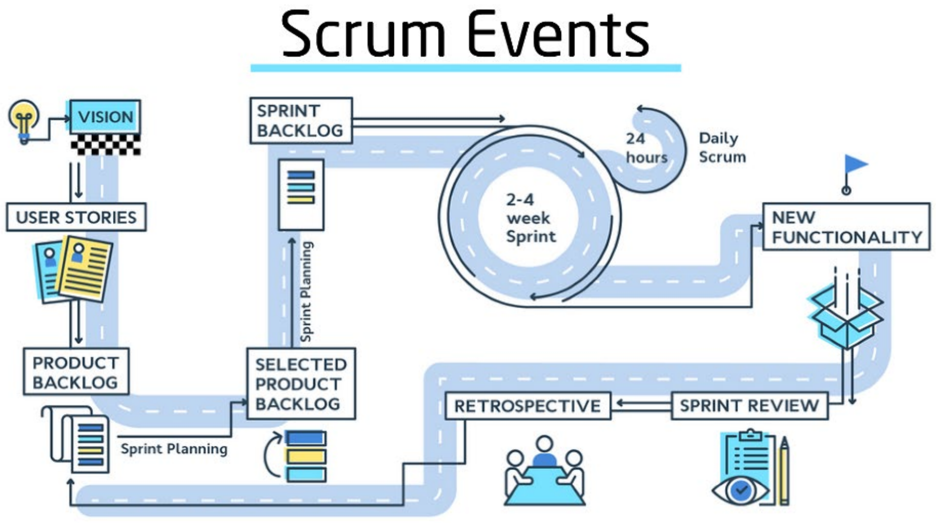


Figure 2.2: An overview of SCRUM Processes (Agile Arena, 2021).

In this section, all references to SCRUM processes will be sourced from The SCRUM Guide by Ken Schwaber & Jeff Sutherland (2020). Similar to many other SDLC approaches, the first phase in SCRUM also involves gathering user requirements. However, what sets SCRUM apart is that these requirements are captured in the form of user stories and recorded in the Product Backlog. Once the requirements have been identified and added to the product backlog, the SCRUM process proceeds through its main events, which consist of five main activities: Sprint Planning, the Sprint itself, Daily SCRUM, Sprint Review, and Sprint Retrospective. These events are iterative and continue until the entire product backlog has been developed. It's important to note that product backlog may be updated following the sprint review to reflect stakeholder feedback.

Sprint Planning

Sprint planning marks the start of each Sprint. During this event, the SCRUM Team comes together to determine the work that needs to be accomplished in the upcoming Sprint and how they will achieve it. The team begins by selecting a Product Backlog Item (PBI) for the upcoming sprint and identifying the specific features that need to be addressed in the product backlog itself. These features are then documented in the sprint backlog, outlining the tasks to be completed or developed during the sprint in alignment with the Definition of

Done. Additionally, the sprint planning session results in the creation of a sprint goal, a concise objective for the sprint and for the team to achieve. The duration of the sprint planning event is typically around 8 hours for a one-month-long sprint, although it may be shorter depending on the chosen product backlog.

The Sprint

The sprint is the core component of the SCRUM, and it is a time-boxed phase typically lasting from one week to one month. During the sprint, the team focuses on coding and testing, working to accomplish the sprint backlog and progress towards the Sprint goal (Praecipio, 2022). Additionally, the Sprint may address previously unfinished tasks and make adjustments to the product backlog items that have already been done based on stakeholder feedback.

While the sprint is in progress, refinement of the product backlog is allowed, but modifications to the sprint backlog require the agreement of the Product Owner. However, it is important to note that modification of sprint backlog is not encouraged as it may alter the team workflow. Each sprint takes the team one step closer to achieving the overall product goal, and it is essential to maintain an iterative approach until all items in the product backlog have been completed. To assess the progress of the sprint, various metrics such as burn-down or burn-up charts are implemented.

A SCRUM board is often used to effectively manage the progress of a sprint within the SCRUM framework. It functions as a visual representation of the ongoing sprint, comprising the sprint backlog items for the sprint. In a physical SCRUM board, it typically compromised of sticky notes, where these sticky notes represent the sprint backlog items derived from the overall product backlog. These sticky notes are categorized into three distinct columns: "To Do," "In Progress," and "Done." The "To Do" column signifies items that are yet to be initiated, the "In Progress" column denotes items currently undergoing development, while the "Done" column represents items that have been successfully developed (Crail, Bottorff and Watts, 2023). Alternatively, SCRUM boards can take on a digital form through platforms such as Jira, thereby replicating the same structure as the physical SCRUM board in a virtual environment. The primary purpose of using a SCRUM board is to ensure that everyone remains informed about the sprint's progress. Furthermore, it encourages collaboration among team members by providing a platform for discussions during the Daily SCRUM meetings.

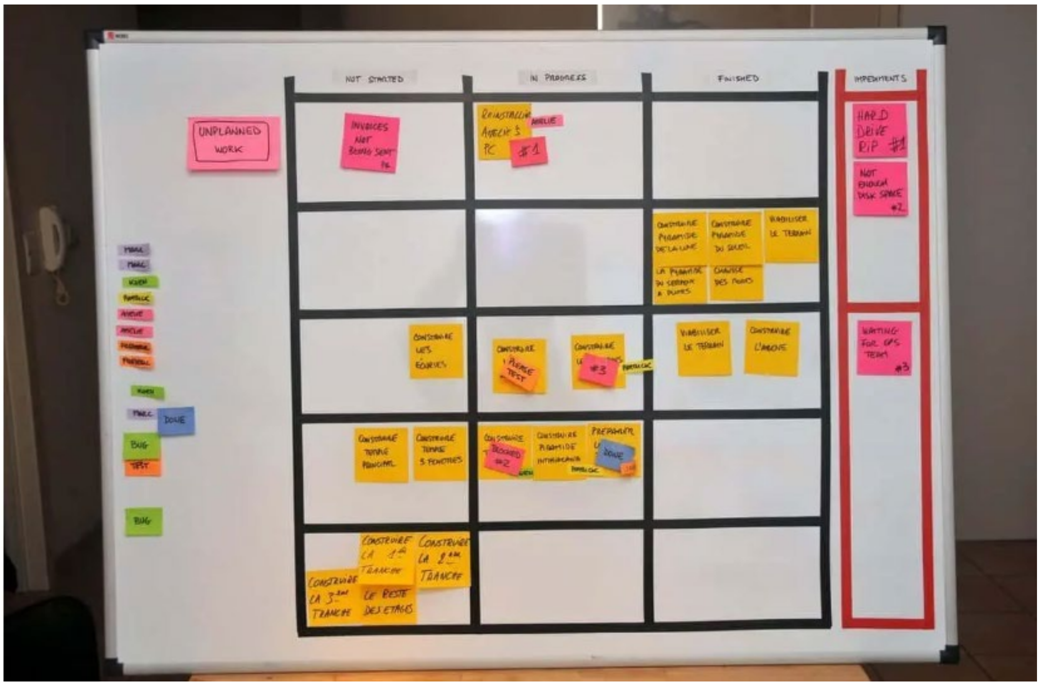


Figure 2.3: Sample Physical SCRUM Board (Eby, 2016).

Daily SCRUM

The Daily SCRUM is a daily meeting that occurs throughout the sprint in the SCRUM framework. Its primary purpose is to inspect the overall progress of the sprint toward achieving its goal and to inspect the development team's adherence to the schedule. This short meeting typically lasts around 15 minutes and is in informal matter where the entire SCRUM team reports their progress, collaborates, and assists each other in case of any issues or challenges. The daily SCRUM enhances team communication, facilitates rapid problem-solving and decision-making, and minimizes the need for formal meetings. It provides a dedicated time for the team to come together, discuss adjustments to their plan, and support each other in resolving any obstacles or problems that may arise during the sprint.

Sprint Review

At the conclusion of each sprint in SCRUM, the SCRUM team is responsible for delivering a workable product to the stakeholders. This product increment undergoes a thorough inspections during the sprint review. The primary purpose of the sprint review is to assess the outcomes of the Sprint and whether if the sprint goal has been achieved. During this meeting, the SCRUM team showcases their completed work to the stakeholders, who, in turn, provide feedback to allow any uncertainties, dissatisfaction or changes to be addressed. The sprint review serves as a valuable opportunity to collect stakeholder feedback, enabling the team to refine the product backlog accordingly. This refinement may involve making changes to existing requirements or considering new features for future sprints, ensuring that the final product meet the stakeholder's expectation. For a one-month sprint, the sprint review typically has a maximum duration of four hours.

Sprint Retrospective

The sprint retrospective is a collaborative session similar to a 'lessons learned' phase. During this meeting, the SCRUM team gathers to reflect on what went well and what didn't go as planned in sprint. The primary aim of the Sprint Retrospective is to identify spaces for improvement in teamwork, processes, and overall effectiveness, with the goal of enhancing the quality of work in upcoming sprints. The sprint retrospective focus on the continuous improvement within the team. For a one-month sprint, this meeting typically lasts for a maximum of three hours.

2.2.2.2 Formation of SCRUM Team

A SCRUM team is composed of three distinct roles: the SCRUM Team, often referred to as developers, the Product Owner, and the SCRUM Master. Typically, a SCRUM team comprises fewer than 10 individuals. According to Schwaber and Sutherland (2020), a smaller team size is preferred as the team size grows, collaboration and coordination become increasingly challenging. In contrast, smaller teams tend to communicate more effectively and are generally more productive.

SCRUM Team

A SCRUM Team is responsible for completing tasks within each Sprint. The team members typically don't have specific job titles, as they are all capable of taking on various roles within the team. The scope of work often encompasses responsibilities such as business analysis, system analysis, development, testing, and quality assurance. The primary objective is to deliver a functional product at the end of each sprint, and to continue this iterative and incremental process until the project is complete.

SCRUM Master

The SCRUM Master serves as a facilitator and coach for the entire SCRUM team and typically possesses an extensive understanding of the SCRUM framework. Their expertise enables them to guide the development team in using the framework effectively. They help to monitor the SCRUM events and the progress of a sprint, ensuring the team the team stay aligned with project goals and meet the Definition of Done.

Product Owner

The Product Owner plays an important role in the team, acting as the liaison between the SCRUM team and the external stakeholders. They are responsible for managing, refining and prioritizing Product Backlog, ensuring that the SCRUM team to have a clear understanding of what are the outcomes of the projects. The Product Owner collaborates closely with both the development team and stakeholders to collect feedback and make adjustments accordingly, while keeping the team aligned with the expectations of stakeholders.

2.2.2.3 SCRUM Artifacts

As stated by Duraisamy and Atan (2013), SCRUM primarily focuses on frequent deliverables and lightweight development processes, which often results in minimal documentation due to frequent changes in requirements. SCRUM has three main artifacts: the Product Backlog, the Sprint Backlog, and the Increment.

<u>Product Backlog</u>

A product backlog is a comprehensive list of tasks, features, and fixes that are needed for a project. It can be thought of as a "wish list" containing all the elements — from features to bug fixes that need to be developed or addressed in a SCRUM project. It serves as a foundation for the sprint planning in a SCRUM project. Although many items in the product backlog are often recorded in the form of user stories to facilitate a clearer understanding of system functionality, this is not a strict requirement. Using user stories can aid in more accurate estimation during sprint planning (Rehkopf, n.d.). Once all the product backlog items has been completed, the SCRUM project will be considered a complete.

<u>Sprint Backlog</u>

The sprint backlog is a subset of the product backlog items. It comprises a collection of tasks and requirements for a specific sprint, providing a detailed overview of the objectives to be achieved within that sprint. It functions as both a plan and a guide for developers throughout the sprint, aiding them in pursuing completion of all the items on the sprint backlog. A product backlog item is only considered as complete when all the sprint backlog items are completed.

<u>Increment</u>

The Increment is the sum of all product backlog items and sprint backlog items that have been accomplished in both the current sprint and the previous sprint. It is the compilation of all deliverable products after every sprint, with each one being considered complete and meeting the standards of the Definition of Done set by the SCRUM Team. This concept can be likened to assembling puzzle pieces, where each piece represents the output of a sprint, and the final Increment represents the fully assembled puzzle.

2.2.2.4 Advantages & Disadvantages of SCRUM

<u>Advantages</u>

1. The SCRUM methodology enhances communication within the SCRUM team, contributing to improved teamwork. Consequently, it produces a higher quality of work and greater productivity within the

team, ultimately resulting in increased customer satisfaction with the output (Srivastava, Bhardwaj, and Saraswat, 2017).

2. The SCRUM methodology consistently involves stakeholders at the end of each sprint to gather feedback and refine the system accordingly. This approach ensures that the final product is almost guaranteed to meet stakeholder expectations (Džanić, Džanić, and Toroman, 2022).

3. The SCRUM methodology is flexible to changes through its iterative and incremental approach. This adaptability allows SCRUM to swiftly respond to changes by addressing them in future Sprints, eliminating the need for reversion to previous stages (Bagiu, Avasilcăi & Rusu, 2022).

4. The SCRUM methodology is well-known for its efficient and fast-paced development process. This approach enables the team to work collaboratively and deliver valuable outcomes within short timeframes.

Disadvantages

1. One of the drawbacks of the SCRUM methodology is its limited documentation, which can lead to difficulties in tracking project completion datelines and schedule timelines. Moreover, the absence of thorough documentation complicates the process of maintaining and upgrading the system. In situations where a team member departs during an on-going sprint or project, their knowledge leaves with them, making knowledge transfer challenging (Džanić, Džanić, and Toroman, 2022).

2. While the dynamic nature of the SCRUM methodology encourages adaptability, it can also present challenges. The frequent changes in requirements can result in an uncertain project completion timeline, affecting various aspects of the project (Bagiu, Avasilcăi & Rusu, 2022).

3. Transitioning from a non-agile methodology to SCRUM can prove to be difficult, as it requires a significant adjustment. Additionally, effectively

utilizing SCRUM demands an experienced and skilled team (Srivastava, Bhardwaj, and Saraswat, 2017).

4. The SCRUM methodology employs the concept of timeboxing for all set timeframes. While this approach provides flexibility in scheduling, it can lead to overwork within the sprint itself and potentially lesser buffer time allocated. This concept introduces multiple uncertainty during the project itself (Ozkan et al, 2022).

2.2.3 Prototype Methodology

The prototype methodology doesn't neatly fit into the traditional, hybrid, or agile categories of SDLC approaches. Its classification largely depends on how developers implement it; it can belong to any category. However, the prototype methodology is most often iterative, aligning closely with the characteristics of Agile approaches. This makes the entire development process iterative until the prototype is finalized and becomes the final product. The term 'Prototype' captures the essence of this methodology, where prototypes are built, tested, and reworked until an acceptable final version is achieved (Martin 2023).

2.2.3.1 Phases of Prototype Methodology

There are many variations of the prototype methodology, but they all share common characteristics in their phases, specifically in the building and refining of prototypes after user evaluation. Generally, the prototype methodology consists of six distinct phases: Requirements Gathering and Analysis, Quick Design, Building Prototype, User Evaluation, Refining Prototype, and Implement and Maintain.

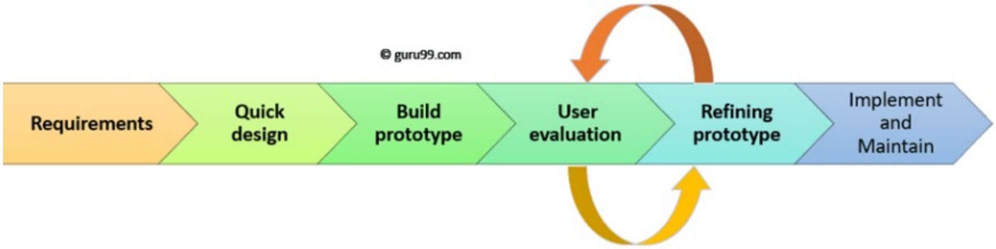


Figure 2.4: Prototyping Model SDLC Phases (Martin, 2023).

Requirements

Similar to all other SDLC approaches, the first phase of the prototype methodology involves gathering requirements. In this phase, all the necessary requirements are identified and analyzed. Interviews are commonly used to gather these requirements from stakeholders and end-users.

Quick Design

With the requirements identified, a preliminary design or more commonly known as a quick design is created for the system. While this design may not yet be interactive, it serves to present the general idea and layout of how the system will function. This preliminary design helps identify the system's key features and functionalities. It aids developers in understanding the integration between the front end and the back end, providing a clear concept for the development process.

Build Prototype

Based on the quick design established in the previous phase, an initial version of the prototype is built. This prototype offers a functional overview of the system, showcasing a scaled-down version that includes the most critical and significant features. This initial prototype serves as a foundational model for all subsequent versions, as it will go through an iterative process of refinement until a final prototype is approved. It also allows stakeholders to visualize how the main requirements are being implemented into the system.

User Evaluation

Once the initial prototype is built, it undergoes a user evaluation process. During this phase, stakeholders provide feedback on the presented prototype. This allows developers to identify both the strengths and weaknesses of the system, as well as any uncertainties, misimplementations of requirements, or missing features that need to be addressed. The feedback collected from stakeholders is then used as a reference in the subsequent refining phase. This evaluation process is iterative and works in tandem with the refining prototype phase, leading to an iterative cycle of improvement.

Refining prototype

With the feedback collected during the User Evaluation phase and the previously built prototype as a reference, developers refine the prototype accordingly. The revised prototype is then subjected to another round of user evaluation. This cycle of refinement and evaluation continues iteratively until all requirements and user needs are met, at which point this phase is deemed unnecessary and is omitted.

Implement and Maintain

The "Implement and Maintain" phase is the final stage of the prototype methodology. During this phase, the prototype is finalized and subjected to multiple testing to ensure that all features and system components are functioning as intended. Once verified, it is transformed into a full-fledged system ready for deployment. Additionally, the system may undergo maintenance as needed to refine and update it over time.

2.2.3.2 Advantages & Disadvantages of Prototype Methodology

The advantages and disadvantages of both SCRUM and prototype methodology can be very similar as they share the same characteristics.

Advantages

1. The prototype methodology inherently involves stakeholders in the user evaluation process, enhancing the prototype accordingly. This ensures that the finalized prototype or system aligns with stakeholders' expectations (Thakur, n.d.). Stakeholders also experience reduced waiting times, as they can quickly see how the system works and is implemented with the first prototype built (Neha, 2020).

2. Since the prototype methodology is an iterative approach used for refinement, the built prototype closely resembles the final system. This similarity allows developers to reduce their efforts in building the final system. Moreover, continuous user involvement leads to clearer and more precise specifications, ensuring that the final system meets

expectations and minimizing the chances of errors. Early detection of errors is also facilitated during the prototyping stage (Neha, 2020).

3. The prototype methodology offers high flexibility towards changes to requirements due to its iterative approach involving user evaluation and prototype refinement. If needed, prototypes can even be discarded to create new ones (Neha, 2020).

4. The prototype methodology is straightforward, making it easy to understand and implement. It doesn't require an experienced team, allowing those who are less experienced to effectively implement it (Martin, 2023).

Disadvantages

1. The iterative nature of user evaluation and refining prototypes can lead to uncertainty in project schedules and completion deadlines. Additionally, due to the frequent involvement in user evaluation, stakeholders may take advantages for granted and always be not satisfied with the prototype, increasing the length of iteration loops. (Neha, 2020).

2. Due to potential customer dissatisfaction with prototypes, there might be instances where prototypes need to be discarded and rebuilt. This can lead to increased resource allocation and additional time needed to revert back to previous stages, ultimately impacting the project budget (Neha, 2020).

3. Displaying prototypes to stakeholders can create a false impression of rapid progress. Stakeholders might develop unrealistic expectations about the speed at which the final product can be delivered (Thakur, n.d.).

2.2.4 Comparison between SDLC Methodology

In this section, a comparison matrix will be presented to highlight the key factors relevant to the final year project development. Factors such as cost and post-deployment maintenance will be omitted from consideration, as they do not directly impact the project. The matrix will provide a clear overview of the factors that are essential for evaluating the project's development process.

Table 2.1: Comparison Matrix of SDLC

Model / Features	Waterfall	SCRUM	Prototype
Flexibility	Low	High	High
User Involvement	Low	High	High
Iterative	No	Yes	Yes
Requirement Gathering	During the beginning phase	Frequently Change	Frequently Change
Risk Involved	High	Low	Low
Documentation	High	Low	Low
Time Frame	Long	Short	Short
Simplicity	Simple	Intermediate	Simple
Requirement of Team Members	No	Yes	No

The prototype methodology has been selected as the preferred SDLC for this final year project. Several factors need to be considered in this decision.

Firstly, the flexibility of the chosen SDLC is crucial. Given that the project relies on the guidance of the supervisor and that requirements might change over time, a high level of flexibility is required. This is necessary due to the need for continuous modifications in response to evolving requirements. User involvement is mandatory in this project, necessitating an iterative approach. Failing to adopt an iterative methodology would increase the project's risk, as the inability to address changes could be detrimental.

Furthermore, the final year project spans a mere 13 weeks for the development process, making the timeframe exceptionally short. While SCRUM and prototype methodologies might not prioritize extensive documentation, the nature of this final projects project mandates extensive documentation of the design solution. This requirement of documentation has overcome the disadvantages of both SDLC approaches.

A significant factor to consider is that this final year project is undertaken solo. Consequently, methodologies like SCRUM, which rely on team collaboration, are not recommended. Given limited experience, it is advisable to opt for SDLC that is easy to understand and implement.

In conclusion, the chosen prototype methodology aligns with the project's characteristic, which include rapid delivery time, flexibility, extensive documentation, adaptability for a solo developer and most importantly it is easy to implement. These factors greatly contribute to the success execution of the final year project.

2.3 Existing Applications

In this section of the literature review, five existing applications used in SCRUM management are reviewed and referenced for analysis for the final year project web application. The review will encompass their core functions, advantages, and drawbacks. An analysis will be conducted to identify the features to be included in our application and to explore how the final year project application can enhance these features. The five applications that will be reviewed are: Jira, Trello, Microsoft Excel, Google Sheet, and SCRUMwise. While each of the application may offer various tools and functions, this review will only solely focus on the SCRUM project management.

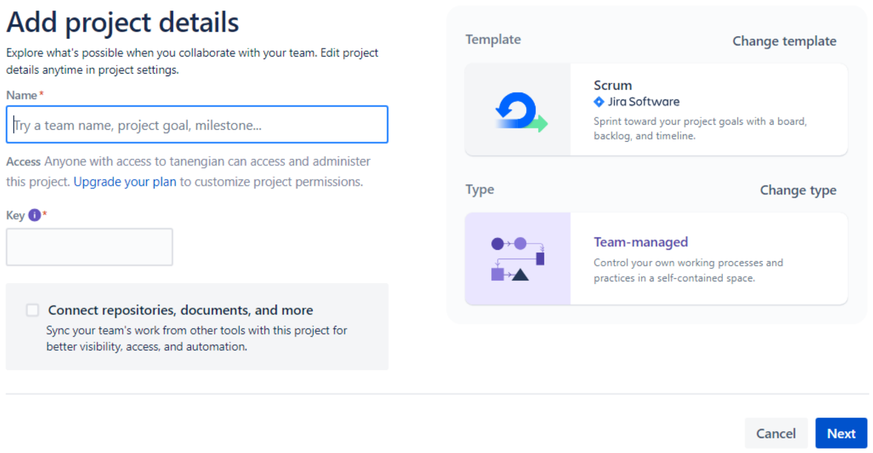
2.3.1 Jira

Figure 2.5: Jira Software Logo

Jira Software, commonly referred to as Jira, stands as one of the most widely used project management and issue tracking software developed by Atlassian. Jira allow teams and organizations to engage in comprehensive project management, which includes the planning, tracking, and the effective management of a project. This platform emphasize on the collaboration, streamlines workflows, and amplifies overall team productivity, allowing everyone involved to share awareness of project progress. With various features available, Jira accommodates various timeframes and project methodologies, notably including the SCRUM framework.

2.3.1.1 Features of Jira

Creation of Project



Add project details

Explore what's possible when you collaborate with your team. Edit project details anytime in project settings.

Name *

Try a team name, project goal, milestone...

Access Anyone with access to tanengian can access and administer this project. Upgrade your plan to customize project permissions.

Key *

□ Connect repositories, documents, and more
Sync your team's work from other tools with this project for better visibility, access, and automation.

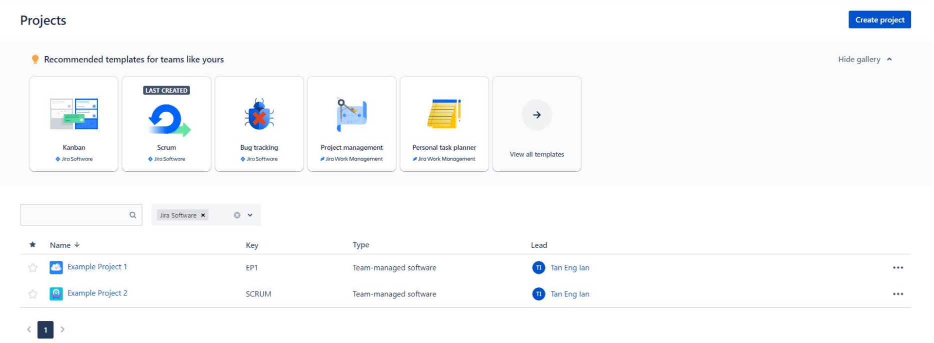
Template — Change template

Scrum
Jira Software
Sprint toward your project goals with a board, backlog, and timeline.

Type — Change type

Team-managed
Control your own working processes and practices in a self-contained space.

Figure 2.6: Creation of Project



Projects

Recommended templates for teams like yours

Name	Key	Type	Lead
Example Project 1	EP1	Team-managed software	Tan Eng Ian
Example Project 2	SCRUM	Team-managed software	Tan Eng Ian

Figure 2.7: Project Selection Dashboard

Jira allows user to create multiple SCRUM projects and choose their desired project to manage on the project dashboard.

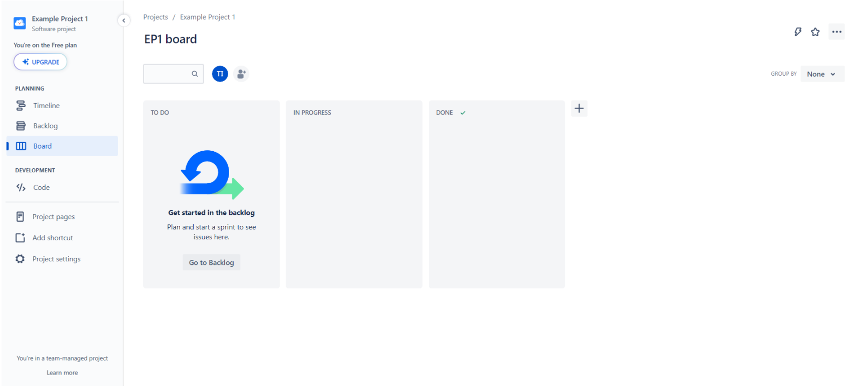
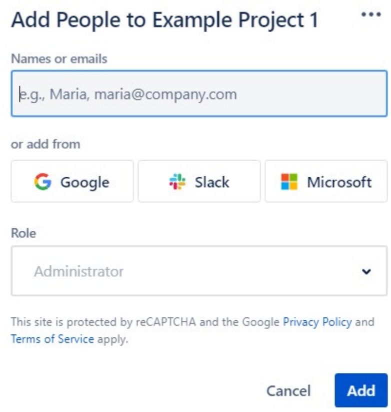


Figure 2.8: Project Dashboard

An empty template will be initiated for the SCRUM project created.

User Management



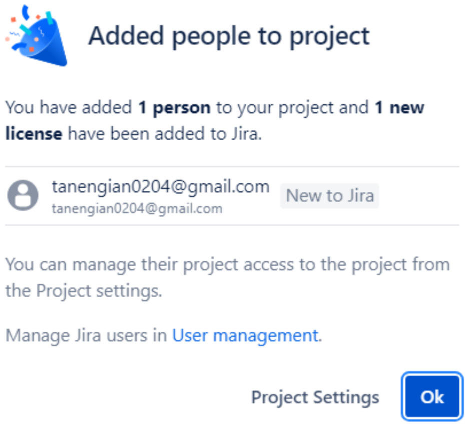
Add People to Example Project 1

Names or emails: e.g., Maria, maria@company.com

Role: Administrator

This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.

Figure 2.9: Adding User



Added people to project

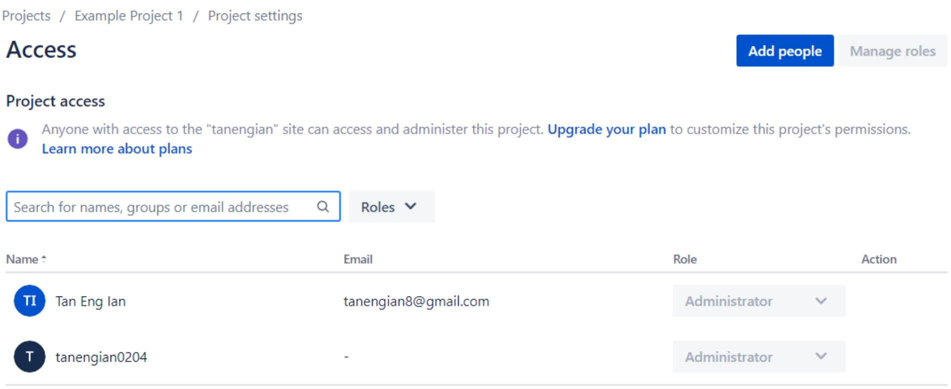
You have added **1 person** to your project and **1 new license** have been added to Jira.

tanengian0204@gmail.com
tanengian0204@gmail.com New to Jira

You can manage their project access to the project from the Project settings.

Manage Jira users in User management.

Figure 2.10: Successfully Adding User



Projects / Example Project 1 / Project settings

Access

Project access

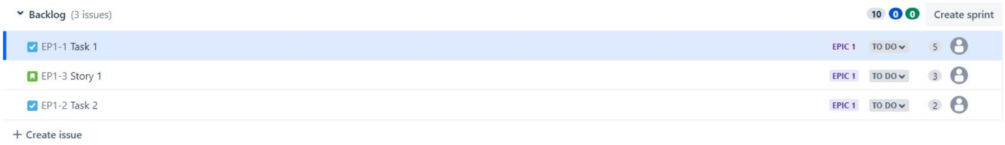
Anyone with access to the "tanengian" site can access and administer this project. Upgrade your plan to customize this project's permissions.
Learn more about plans

Name	Email	Role	Action
TI Tan Eng Ian	tanengian8@gmail.com	Administrator	
T tanengian0204	-	Administrator	

Figure 2.11: List of User(s) in the Project

Jira allows a project administrator to add users to their desired projects regardless of the user is a registered Jira user or not. The number of users is limited only up to 10 for free version. The role of the users can be edited only in the paid version. Each role will have different permission in editing the boards.

Backlog



Backlog (3 issues)

EP1-1 Task 1	EPIC 1	TO DO	5
EP1-3 Story 1	EPIC 1	TO DO	3
EP1-2 Task 2	EPIC 1	TO DO	2

Figure 2.12: Backlogs

Jira allows the user to create multiple backlogs, and these backlogs can be categorized according to user's customization such ask the type of backlog and the status.

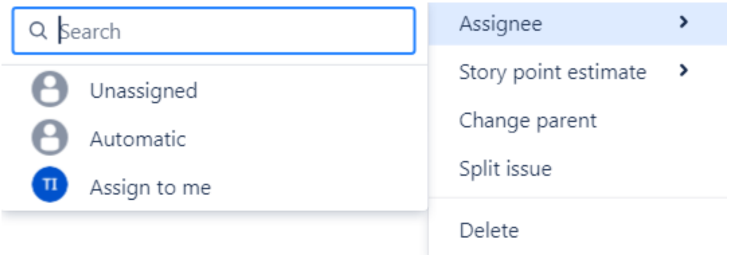


Figure 2.12: Assigning Backlogs

These backlogs can be assigned to any of the users within the project.

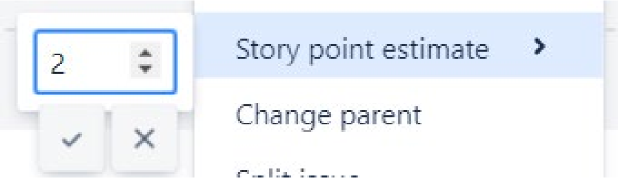
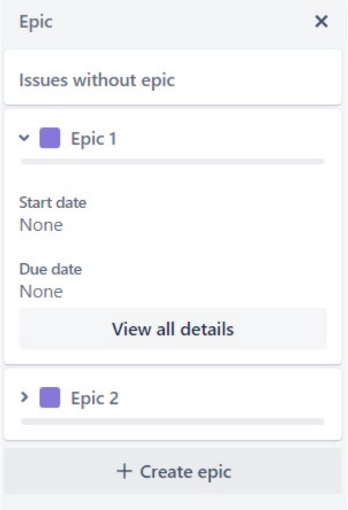


Figure 2.13: Story Point Estimation

Story point estimation can be added to all of the backlogs.


Epic

Issues without epic

Epic 1

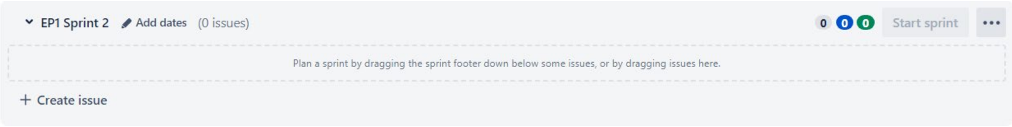
Start date
None

Due date
None

Figure 2.14: Creation of Epic

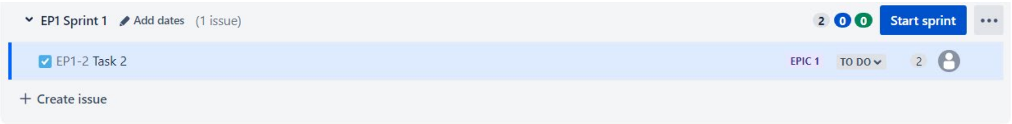
Jira does not categorize backlog into product backlog and sprint backlog. However, they have another feature called "Epic". Epic is similar to product backlog. It can consist of multiple backlog items to be done and have a start date and end date.

Sprint


EP1 Sprint 2 Add dates (0 issues)

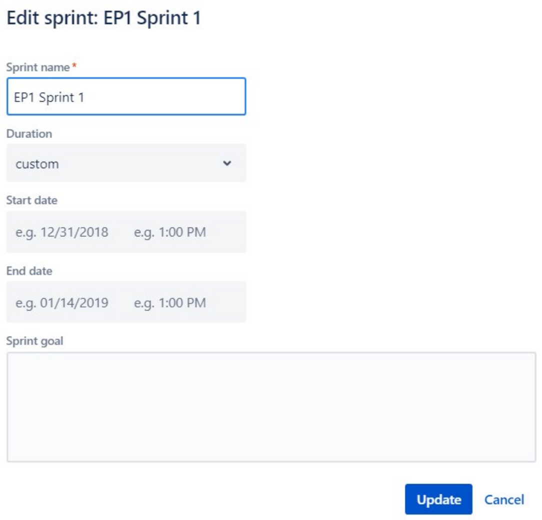
Plan a sprint by dragging the sprint footer down below some issues, or by dragging issues here.

Figure 2.15: Empty Sprint


EP1 Sprint 1 Add dates (1 issue)

☑ EP1-2 Task 2 EPIC 1 TO DO 2

Figure 2.16: Adding Backlogs to Sprint



Edit sprint: EP1 Sprint 1

Sprint name*

EP1 Sprint 1

Duration

custom

Start date

e.g. 12/31/2018 e.g. 1:00 PM

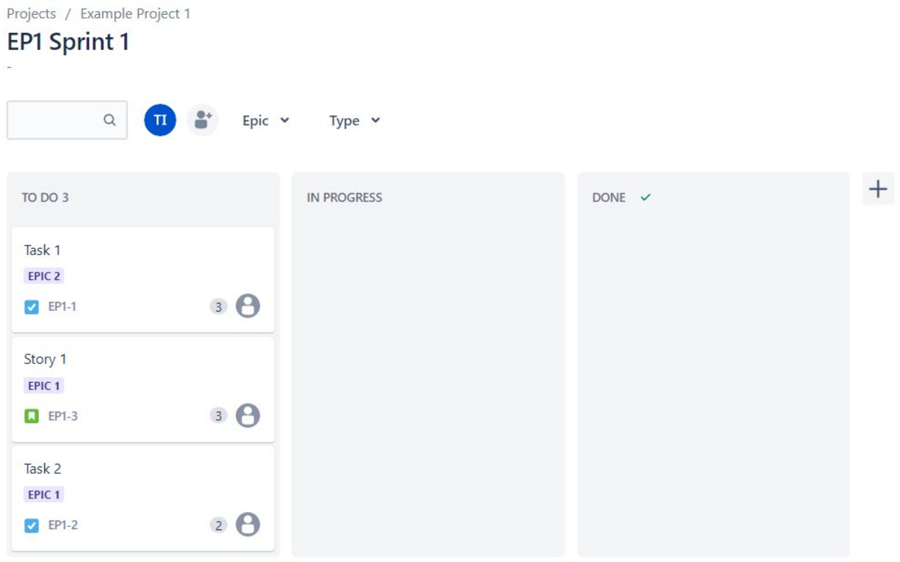
End date

e.g. 01/14/2019 e.g. 1:00 PM

Sprint goal

Figure 2.16: Start Sprint

Jira allow users to create multiple sprints. The backlog items that have been created previously can be added into the Sprint.



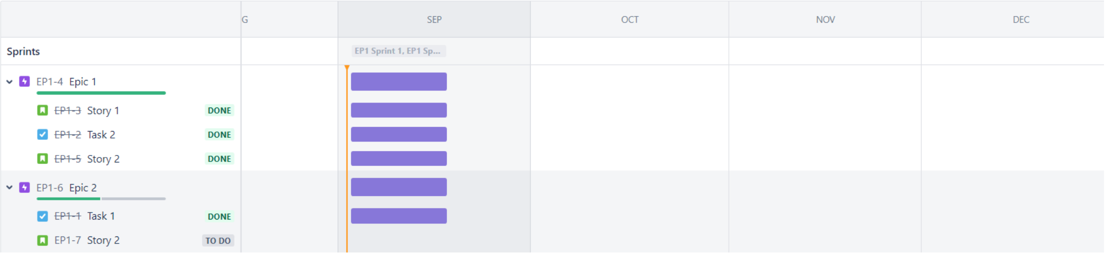
Projects / Example Project 1

EP1 Sprint 1

TO DO 3	IN PROGRESS	DONE ✓
Task 1 EPIC 2 ☑ EP1-1 3		
Story 1 EPIC 1 EP1-3 3		
Task 2 EPIC 1 ☑ EP1-2 2		

Figure 2.17: SCRUM Board

Once a sprint has started, the SCRUM board will be auto generated. The SCRUM board consist of 3 categories: To Do, In Progress, Done. The SCRUM board can be further extended to more category according to the user preferences.

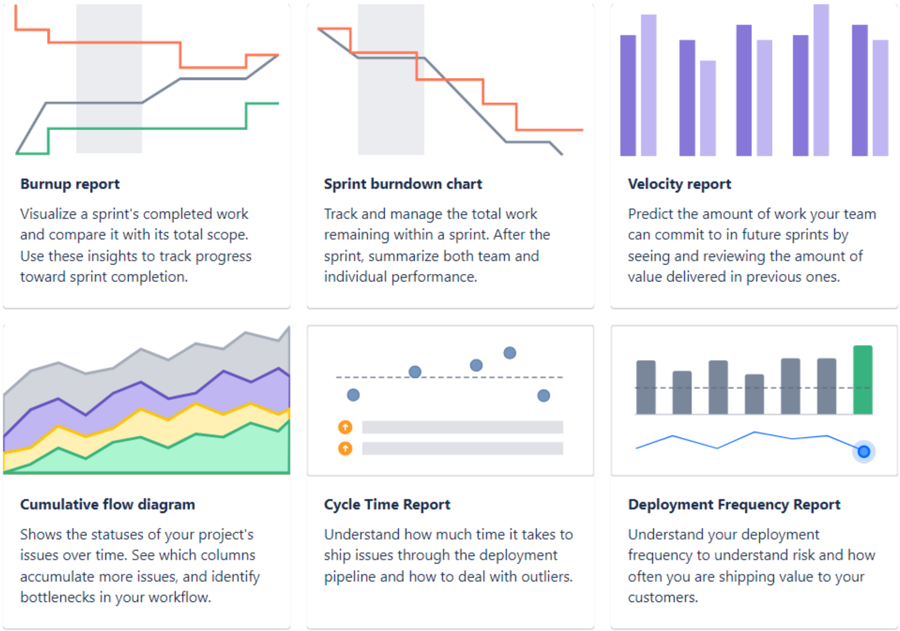


EP1-4 Epic 1
~~EP1-3~~ Story 1 DONE
~~EP1-2~~ Task 2 DONE
~~EP1-5~~ Story 2 DONE
EP1-6 Epic 2
~~EP1-1~~ Task 1 DONE
EP1-7 Story 2 TO DO

Figure 2.18: Timeline

A timeline will be auto generated based on the Epic created. This timeline will only be based on the Epic but not the Sprint itself.

Metrics



Burnup report

Visualize a sprint's completed work and compare it with its total scope. Use these insights to track progress toward sprint completion.

Sprint burndown chart

Track and manage the total work remaining within a sprint. After the sprint, summarize both team and individual performance.

Velocity report

Predict the amount of work your team can commit to in future sprints by seeing and reviewing the amount of value delivered in previous ones.

Cumulative flow diagram

Shows the statuses of your project's issues over time. See which columns accumulate more issues, and identify bottlenecks in your workflow.

Cycle Time Report

Understand how much time it takes to ship issues through the deployment pipeline and how to deal with outliers.

Deployment Frequency Report

Understand your deployment frequency to understand risk and how often you are shipping value to your customers.

Figure 2.19: Jira Report Option

Jira offers a wide range of autogenerated metrics for the users to choose from. These metrics are only applicable to Sprints but not the whole project.

Filtering

Backlog

story

Figure 2.20: Search Input

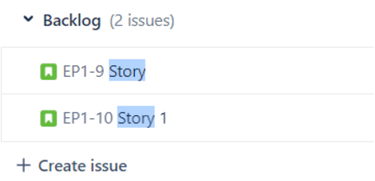


Figure 2.21: Search Result

Jira have search function in the timeline, backlog and SCRUM board. This search function will highlight the searched keywords.

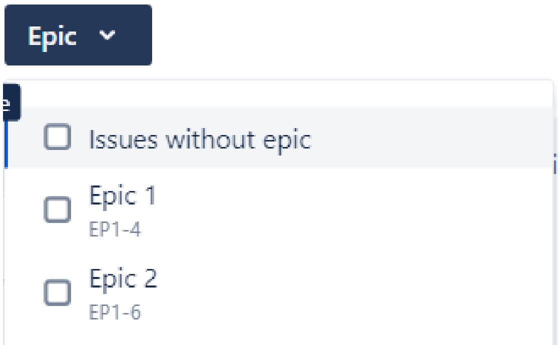


Figure 2.22: Backlog Filter

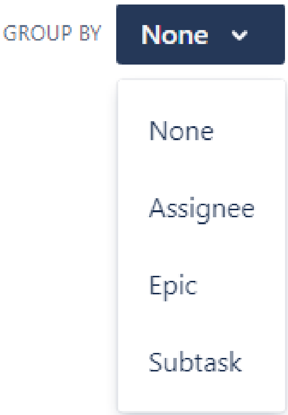


Figure 2.23: SCRUM Board Filter

Jira allows the user to filter the content of the pages depending on the choices of users.

2.3.1.2 Advantages & Drawbacks of Jira

Advantages

1. Jira is highly customizable, allowing third parties plugins to be added into the SCRUM project management.

2. Jira offers a wide range of metrics that is ready to be used such as burndown charts and burnup charts, enhancing the ability to track a Sprint timeline.

3. In Jira, most of the processes such as generating SCRUM boards, timeline and metrics are all automated.

Drawbacks

1. Once the backlog item is marked as done during the Sprint, the backlog item will be removed from the backlogs panel as the panel will only show those that are yet to be completed. Users are required to find the completed task manually from other part of the software. This will make the user hard to track what they have done. The search function will also only highlight the searched keyword, similar to the search function of a browser.

2. Jira requires a subscription fee to unlock majority of the features, making the customization to be limited. The most notable feature will be the role managing function. In the free version of Jira, role managing does not exist, making everyone invite can edit whatever they want in the project. This may cause unnecessary unauthorized changes, complicating the managing processes.

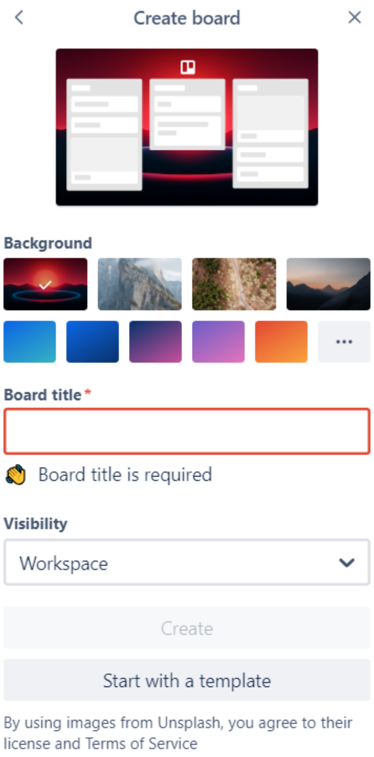
2.3.2 Trello

Figure 2.24: Trello Logo

Trello is a visual project management tool that facilitates collaboration and organization for both individuals and teams. Built around the concept of boards, lists, and cards, Trello offers a flexible and interactive way to manage tasks, track progress, and coordinate activities. Whether you're planning a personal project, managing a team's workflow, or overseeing a complex organizational operation, Trello provides an intuitive interface that simplifies the process. Its drag-and-drop functionality, real-time updates, and customizable features make it an adaptable tool for various type of SDLC project management. While it may not be a direct SCRUM management tools, it can be customized accordingly.

2.3.1.2 Features of Trello

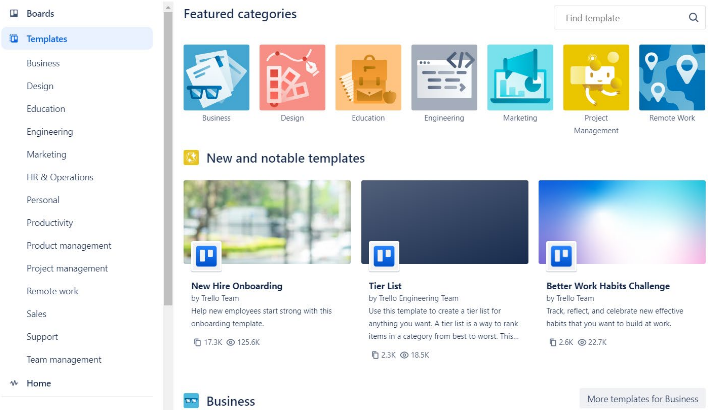
Creation of Board



Board title is required

By using images from Unsplash, you agree to their license and Terms of Service

Figure 2.25: Creating a Trello Board



New Hire Onboarding
by Trello Team
Help new employees start strong with this onboarding template.
17.3K 125.6K

Tier List
by Trello Engineering Team
Use this template to create a tier list for anything you want. A tier list is a way to rank items in a category from best to worst. This...
2.3K 18.5K

Better Work Habits Challenge
by Trello Team
Track, reflect, and celebrate new effective habits that you want to build at work.
2.6K 22.7K

Figure 2.26: Creating a Trello Board with Template

Differing from the usual creation of project, Trello allow the users to create boards. A board can be created with or without a template, where each board can indicate as a project.

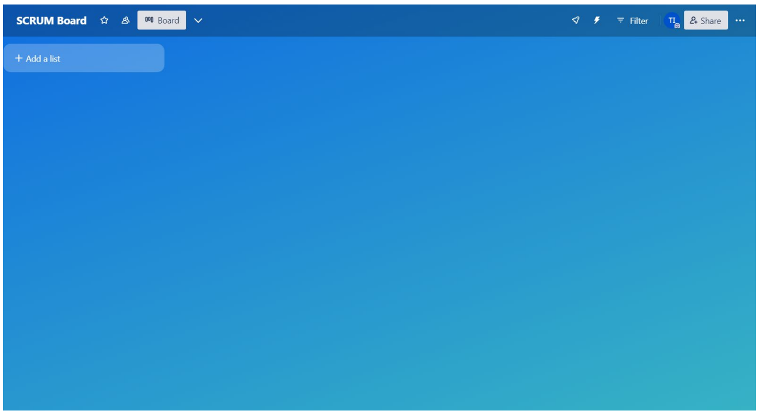
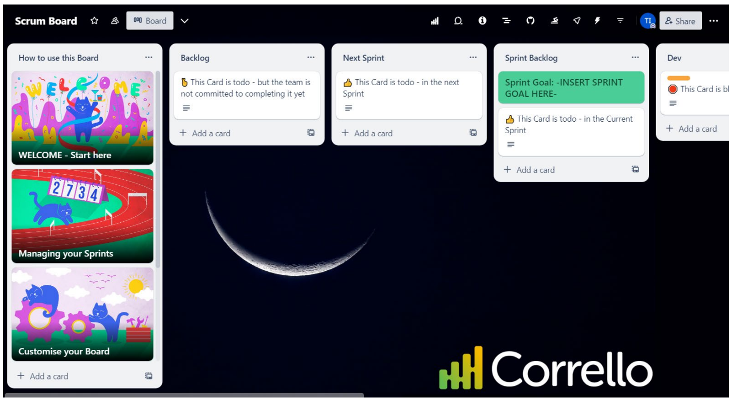


Figure 2.27: Empty Trello Board



How to use this Board

WELCOME - Start here

Managing your Sprints

Customise your Board

+ Add a card

Backlog

This Card is todo - but the team is not committed to completing it yet

+ Add a card

Next Sprint

This Card is todo - in the next Sprint

+ Add a card

Sprint Backlog

Sprint Goal: -INSERT SPRINT GOAL HERE-

This Card is todo - in the Current Sprint

+ Add a card

Dev

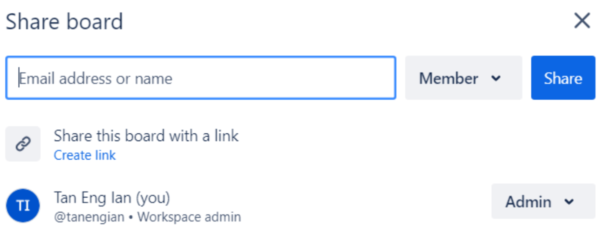
This Card is bl

+ Add a card

Figure 2.28: SCRUM Template Board

If the user creates without any template, the board will be empty. Alternatively, if the user chooses a template, all the elements created by the provider will be added automatically. This board is highly customizable according to the user preferences.

<u>User Management</u>



Share board ✕

Email address or name | Member ⌄ | Share

Share this board with a link
Create link

TI Tan Eng Ian (you)
@tanengian • Workspace admin | Admin ⌄

Figure 2.29: Adding User

Trello allows a board administrator to add users to their desired board regardless of the user is a registered Trello user or not. The role of the users can be edited only in the paid version. Each role will have different permission in editing the boards.

Board Customization

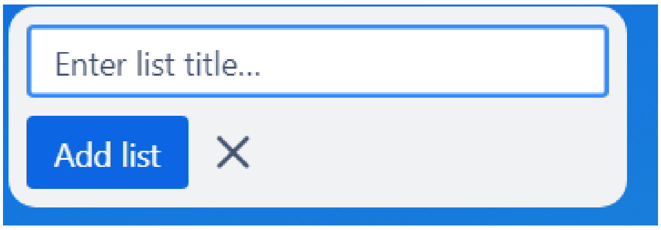


Figure 2.30: Creation of List

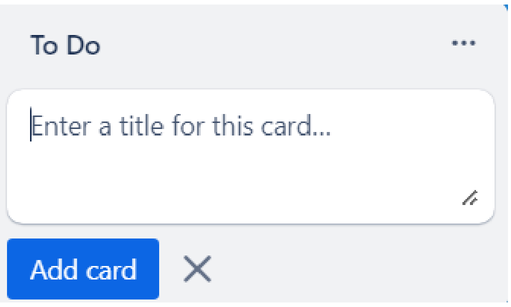


Figure 2.31: Creation of Card

The core feature of Trello is the creation of list and cards. It functions as a visualization tool to show case what are the task to be done similar to a SCRUM board.

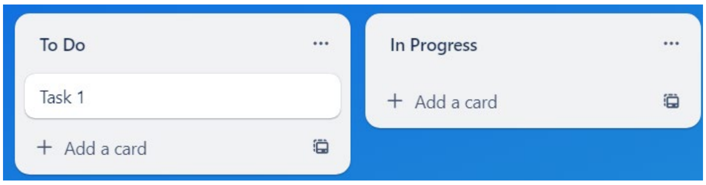


Figure 2.32: Before Moving Task 1 to In Progress List

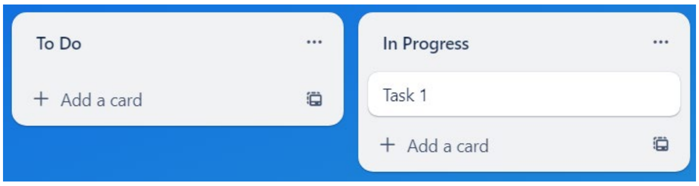
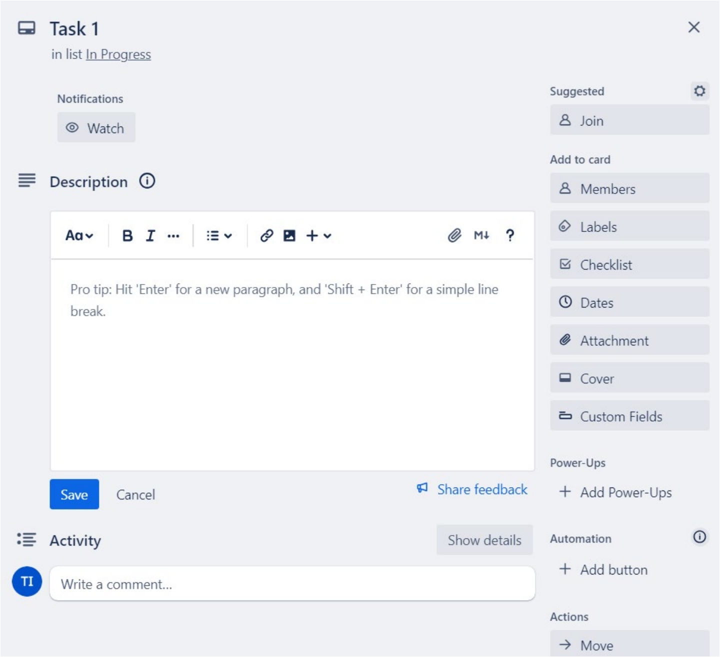


Figure 2.33: Moving Task 1 to In Progress List

The card in the list can be drag to another list, allowing the flexibility in moving the card category.



Task 1

in list In Progress

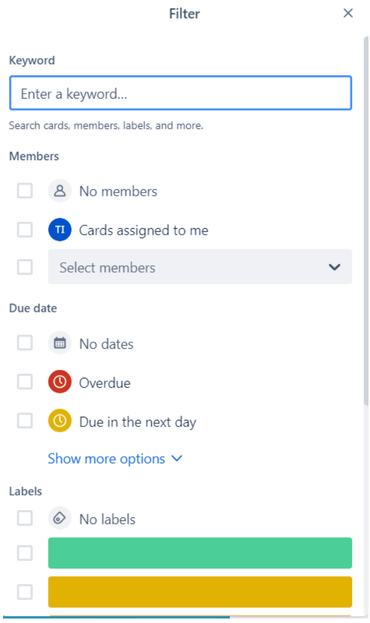
Pro tip: Hit 'Enter' for a new paragraph, and 'Shift + Enter' for a simple line break.

Write a comment...

Figure 2.34: Customization of Card

The card created can be further detailed by clicking into the list. The card can include things such as attachment, dates, checklist and members assigned which are all crucial to a sprint backlog.

Filtering



Filter

Keyword

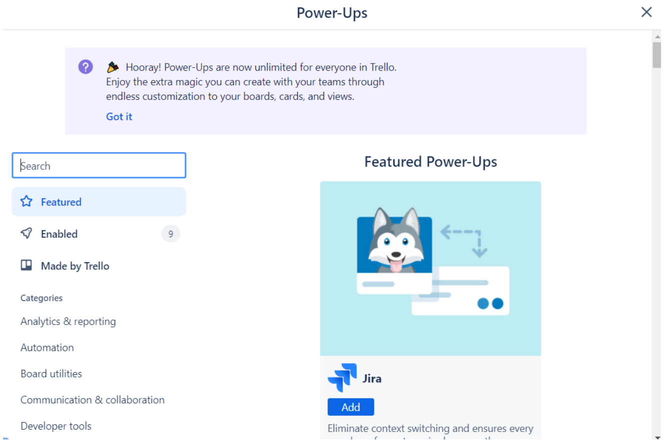
Enter a keyword...

Search cards, members, labels, and more.

Figure 2.35: Filter Option

Trello offers a wide range of filter option for the users to choose from and allow user to search. However, the search keyword does not include the card or list itself but rather the elements of both.

<u>Power-Ups</u>



Power-Ups

Hooray! Power-Ups are now unlimited for everyone in Trello. Enjoy the extra magic you can create with your teams through endless customization to your boards, cards, and views.

Got it

Featured Power-Ups

Jira

Eliminate context switching and ensures every

Figure 2.36: Power-Ups Interface

Trello features a section called "Power-Ups," which function as its name, powering up the overall functionality of a board. These Power-Ups act like extensions or plugins that can be installed directly onto a board, making it highly customizable. They range from simple utilities, such as a calendar view or voting on cards, to more complex integrations with other software tools for project management and time tracking.

2.3.2.2 Advantages & Drawbacks of Trello

<u>Advantages</u>

1. Trello is highly customizable especially with its feature Power-Ups, allowing for flexibility and utilizing various tools in effectively managing the SCRUM projects.

2. Without considering the Power-Ups, Trello is easy to understand and implement, making anyone to be able to create their own board.

Drawbacks

1. While Trello is highly customizable, Trello requires a lot of manuals set up especially with the Power-Ups. Tools like estimation, burn-down charts and report generation requires an extensive installation of the plug in. The integration of the plug ins would also require a lot of pre-condition, making it difficult to implement.

2. Trello requires a subscription fee to unlock majority of the features, making the customization to be limited. The most notable feature will be the role managing function. In the free version of Trello, role managing does not exist, making everyone invite can edit whatever they want in the board. This may cause unnecessary unauthorized changes, complicating the managing processes.

3. As Trello did everything manually, there are no actual features that users can implement to initiate a Sprint. This means that user is required to create their own SCRUM board manually and the Sprint may only be represented in list and cards.
4. Trello's primary function revolves around creating lists and cards, which can sometimes present difficulties in effectively tracking project timelines. While backlog items can be organized within lists and cards, longer project timelines may lead to many lists and cards, potentially resulting in visual complexity. Besides that, the search function is not included in the card and list but only on the element of both such as assignee and status.

2.3.3 Microsoft Excel and Google Sheets

Figure 2.37: Microsoft Excel Logo

Figure 2.38: Google Sheets Logo

Microsoft Excel and Google Sheets are two powerful spreadsheet software tools that are widely used for various tasks, ranging from simple calculations to complex data analysis and project management. Both applications offer a fundamental grid-based interface, facilitating the arrangement of data in rows and columns, thereby serving as a foundational structure for organizing project-related information. These applications allow users to perform calculations and manipulate data while offering a wide range of visualization tools that enable users to translate raw data into meaningful visual representations. Both of these applications can be described as a programming language; by default, they provide a simple empty interface. However, users with expertise can effectively "code" within these applications according to their preferences. The applications can be customized with various functions that can effectively manage various tasks.

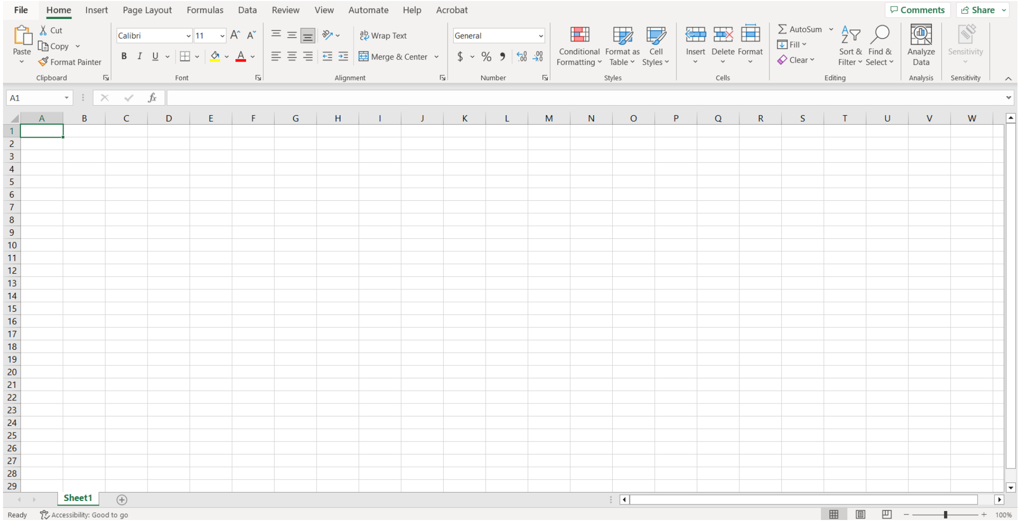


Figure 2.40: Default Excel Template

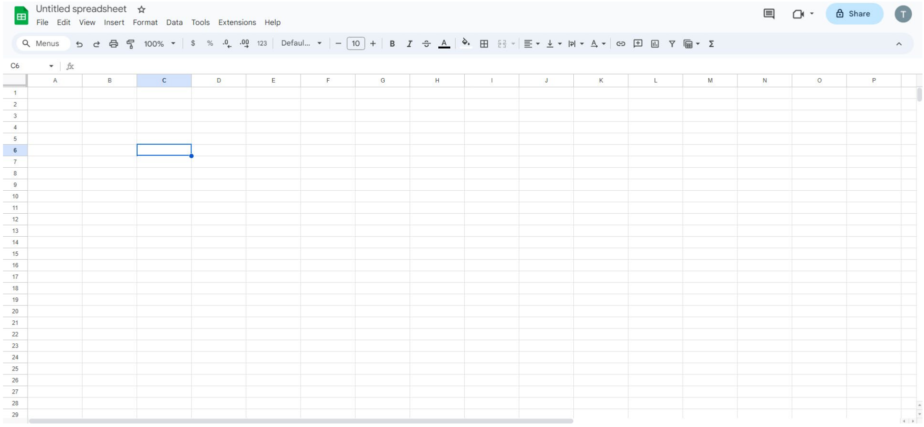
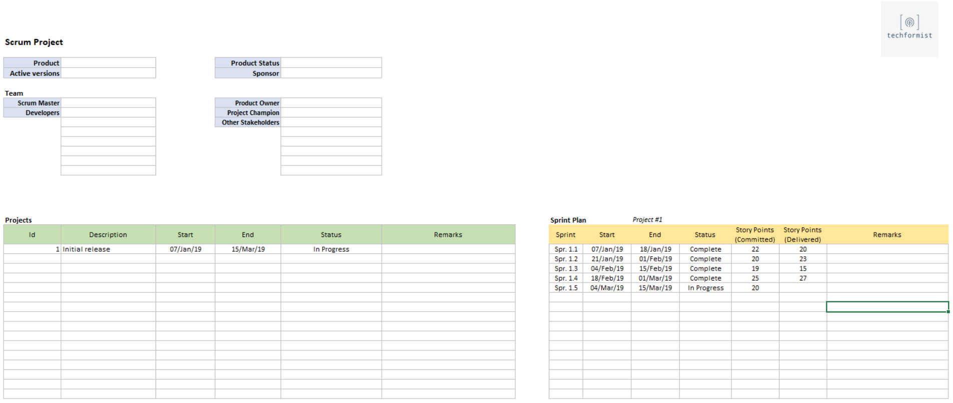


Figure 2.41: Default Google Sheets Template

By default, the spreadsheets are empty. However, user can customize the spreadsheets accordingly to their needs. However, this process can be very complicated and tedious. It requires a professional or user to undergo training in order to fully utilize the features. Alternatively, user can download template online.



Scrum Project

Product	
Active versions	

Product Status	
Sponsor	

Team

Scrum Master	
Developers	

Product Owner	
Project Champion	
Other Stakeholders	

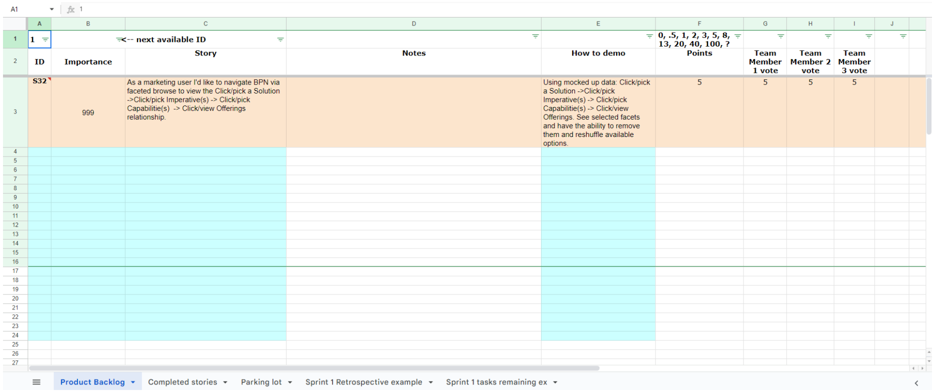
Projects

Id	Description	Start	End	Status	Remarks
1	Initial release	07/Jan/19	15/Mar/19	In Progress	

Sprint Plan *Project #1*

Sprint	Start	End	Status	Story Points (Committed)	Story Points (Delivered)	Remarks
Spr. 1.1	07/Jan/19	18/Jan/19	Complete	22	20	
Spr. 1.2	21/Jan/19	01/Feb/19	Complete	20	23	
Spr. 1.3	04/Feb/19	15/Feb/19	Complete	19	15	
Spr. 1.4	18/Feb/19	01/Mar/19	Complete	25	27	
Spr. 1.5	04/Mar/19	15/Mar/19	In Progress	20		

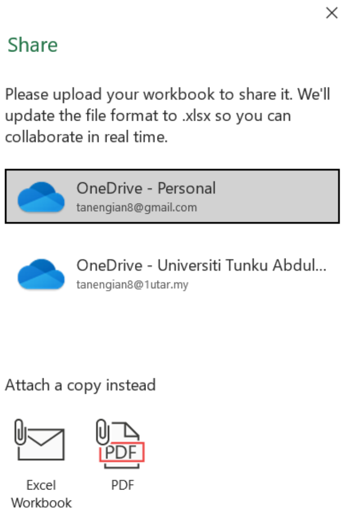
Figure 2.42: Microsoft Excel Sample SCRUM Template



A	B	C	D	E	F	G	H	I	J
1	<-- next available ID				0, .5, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?				
ID	Importance	Story	Notes	How to demo	Points	Team Member 1 vote	Team Member 2 vote	Team Member 3 vote	
S32	999	As a marketing user I'd like to navigate BPN via faceted browse to view the Click/pick a Solution ->Click/pick Imperative(s) -> Click/pick Capabilitie(s) -> Click/view Offerings relationship.		Using mocked up data: Click/pick a Solution ->Click/pick Imperative(s) -> Click/pick Capabilitie(s) -> Click/view Offerings. See selected facets and have the ability to remove them and reshuffle available options.	5	5	5	5	

Figure 2.43: Google Sheets Sample SCRUM Template

Microsoft Excel and Google Sheets can be considered excellent tools for SCRUM project management. They allow for tracking and forecasting SCRUM project completion timelines and offer an unlimited range of customization, depending on the user's creativity. Features like product backlogs, sprint backlogs, auto-generated SCRUM boards, sprint management, burndown and burnup charts are all possible to implement. As final year projects that utilize a Requirement Traceability Matrix to track and forecast completion datelines, these features can also be incorporated. Users can create a product backlog and auto-generate a Requirement Traceability Matrix as needed.



Share

Please upload your workbook to share it. We'll update the file format to .xlsx so you can collaborate in real time.

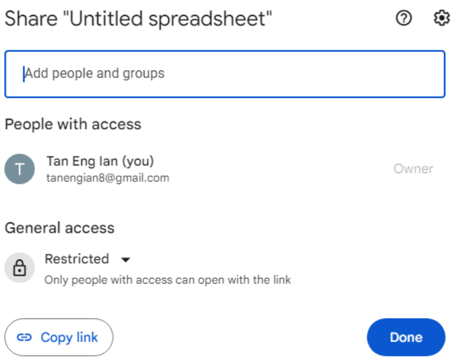
OneDrive - Personal
tanengian8@gmail.com

OneDrive - Universiti Tunku Abdul...
tanengian8@1utar.my

Attach a copy instead

Figure 2.44: Sharing of Microsoft Excel File

Microsoft Excel allow user to share their Excel file through any cloud drives. User can only choose whether if the user can or cannot edit the file. Alternatively, an Excel file can also be imported into Google Sheets to use.



Share "Untitled spreadsheet"

Add people and groups

People with access

Tan Eng Ian (you)
tanengian8@gmail.com
Owner

General access

Restricted
Only people with access can open with the link

Figure 2.45: Sharing of Google Sheets

Google Sheets allows users to share their sheets by entering an email address or by copying the link. Users with access to only the link can also edit the file if given permission making logging in to be optional. Google Sheets can also be downloaded and be edited offline through Microsoft Excels.

2.3.3.1 Advantages & Drawbacks of Microsoft Excels and Google Sheets

Advantages

Both Microsoft Excel and Google Sheets offer an extensive range of customization options for their spreadsheets, allowing almost any desired features to be implemented. It can be said that a fully customized Microsoft Excel or Google Sheets represents the spreadsheet version of the solution for the final year project. Advantages such as auto-generated SCRUM boards or customized metrics can also be tailored to suit specific needs.

Drawbacks

1. As Microsoft Excels and Google Sheets are not dedicated SCRUM project management tools, there are almost no user role management in the applications. They can only edit or not edit, unlike other applications, which have specific roles with specific authority.

2. While Google Sheets is free, Microsoft Excels is extremely expensive to purchase and does not have any free features like other available software. Users are required to purchase it to use all the features.

3. Microsoft Excel and Google Sheets are considerably more complex compared to many other existing applications. Microsoft Excel does not come with any pre-built functions specifically designed for SCRUM project management. Everything needs to be set up manually, and this process can be significantly more difficult compared to other software. This complexity is also influenced by their interfaces, which can be confusing for users, leading to uncertainty about how to effectively use them. Additionally, unlike other applications, Microsoft Excel and Google Sheets lack a wide range of tutorials on how to set up SCRUM project management.

4. As both applications did everything manually, there are no actual features that users can implement to initiate a Sprint but rather everything have to be changed manually. There are also no drag and drop

2.3.5 SCRUMwise

Figure 2.46: SCRUMWise Logo

SCRUMwise is a web-based project management tool developed for effective implementation of the SCRUM framework. Built to support agile methodologies, SCRUMwise assists teams in organizing and overseeing projects collaboratively. Its user-friendly interface simplifies tasks such as backlog management, sprint planning, and progress tracking. With features like visual SCRUM boards and optional sprint estimation, SCRUMwise enhances transparency and enables teams to adapt quickly to changing project requirements.

2.3.5.1 Features of SCRUMwise

Creation of Project

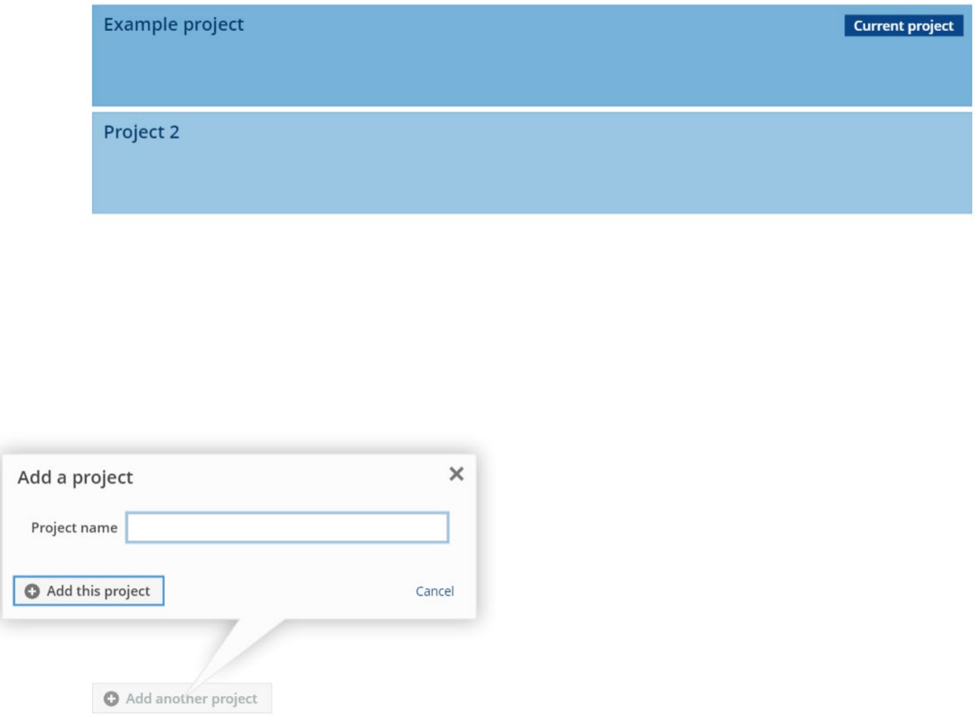
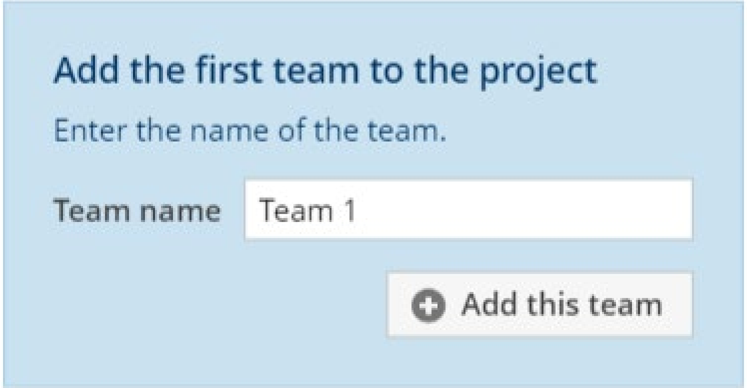


Figure 2.47: Creation and Selection of Project

SCRUMwise allow users to create multiple projects. The users can select the desired project to manage on the same page. However, differing from other application.

User Management



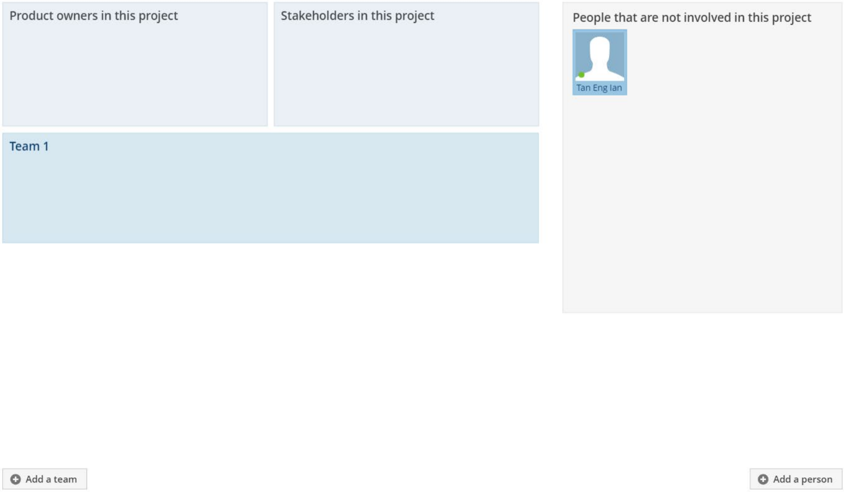
Add the first team to the project

Enter the name of the team.

Team name: Team 1

Figure 2.48: Creation of Team

To first allow users to be added into the project, SCRUMwise requires user to create a team.



Product owners in this project

Stakeholders in this project

Team 1

People that are not involved in this project

Tan Eng Ian

Figure 2.49: Member Interfaces

When a team is created, an interface will be shown to the user to allow them drag and drop the users into the specific columns. User can add more teams if desired.

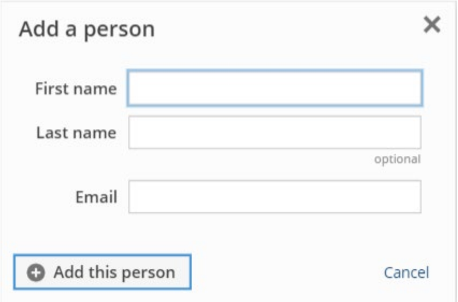


Figure 2.50: Adding User

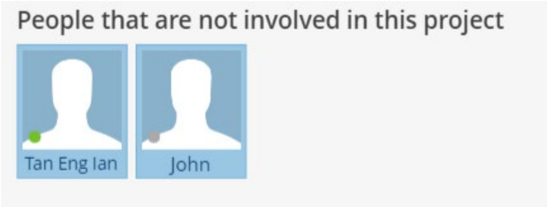


Figure 2.51: Successfully Adding User

SCRUMwise allows a project administrator to add users to their desired projects regardless of the user is a registered SCRUMwise user or not. There is no limit to the number of users that can be added. Upon successful addition, the user will be under "Not Involved”. In SCRUMwise, user roles are categorized as either administrator or normal member. The extra authority of an administrator is the add or removing of a team member.

Backlog

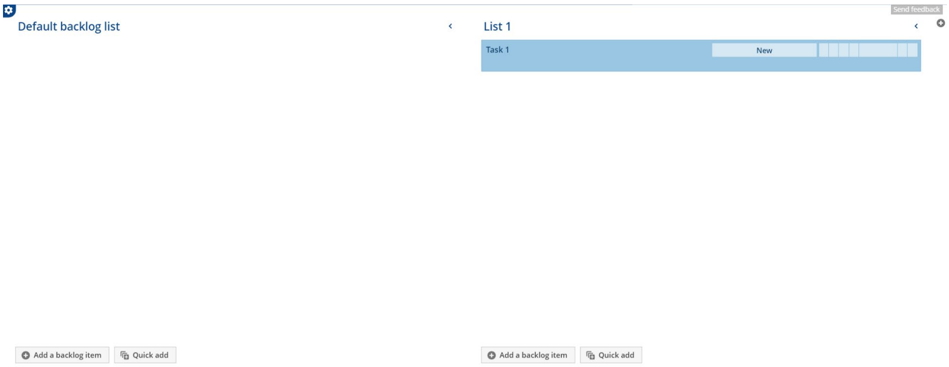
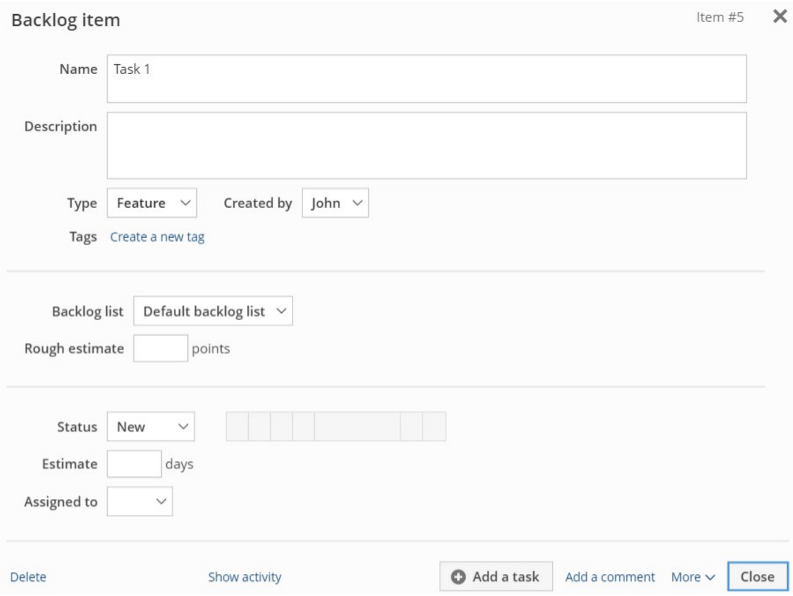


Figure 2.51: List and Backlogs

SCRUMwise does not categorize backlog into product backlog and sprint backlog. However, they allow users to create list which is similar to categorize backlog items which functions similar to a product backlog.



Backlog item

Item #5

Name: Task 1

Description:

Type: Feature

Created by: John

Tags: Create a new tag

Backlog list: Default backlog list

Rough estimate: points

Status: New

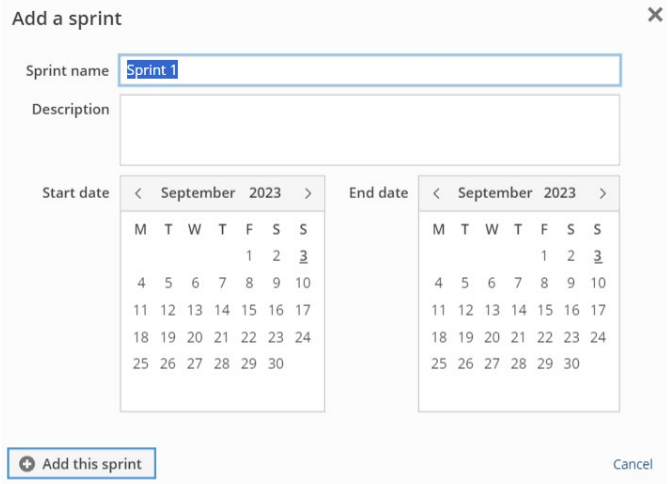
Estimate: days

Assigned to:

Figure 2.52: Backlog Item Details

The created backlogs can be further detailed with multiple type of elements such story point estimation, assigned to, status and estimation days.

Sprint



Add a sprint

Sprint name: Sprint 1

Description:

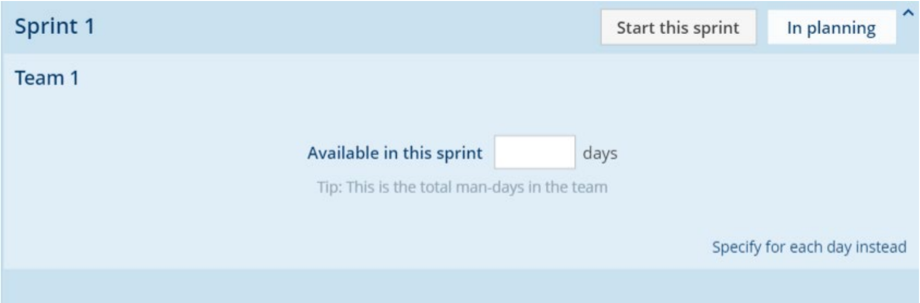
Start date: September 2023

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

End date: September 2023

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

Figure 2.53: Creating a Sprint



Sprint 1

Team 1

Available in this sprint days

Tip: This is the total man-days in the team

Figure 2.54: Available Sprint Days

SCRUMwise allow users to create multiple Sprint. After creating a sprint, the user is required to enter the number of days available in the Sprint period.

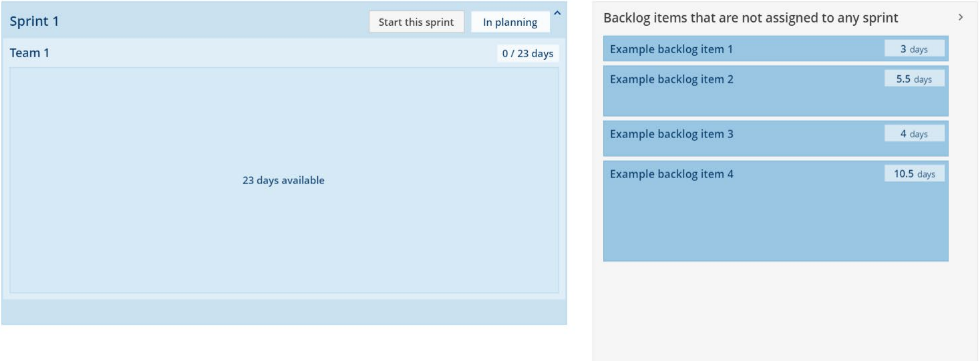


Figure 2.55: Backlog Interface

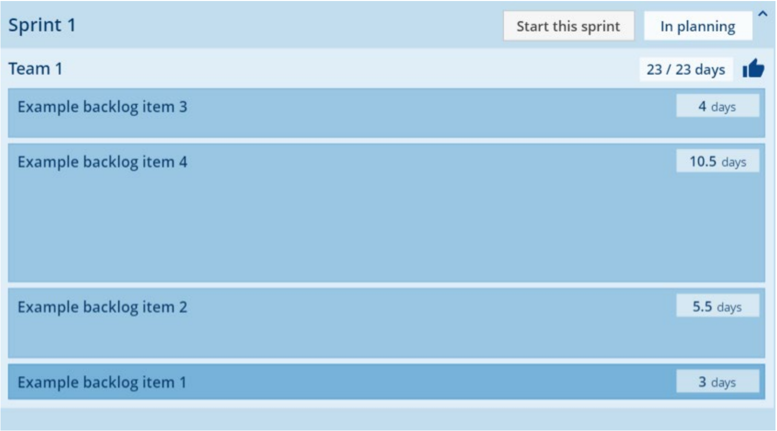


Figure 2.56: Adding Backlogs into Sprint

SCRUMwise requires all backlog items to undergo estimation in terms of days before they can be added into the Sprint. Users are only allowed to add backlog items equivalence or lesser than the available days.

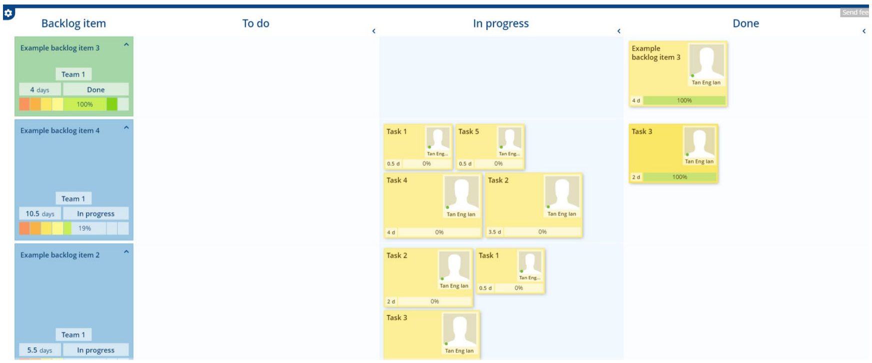


Figure 2.57: SCRUM Board

Once a sprint has started, the SCRUM board will be auto generated. The SCRUM board consist of 3 categories: To do, In Progress, Done. Extra column of category can be added according to the user preferences.

Metrics

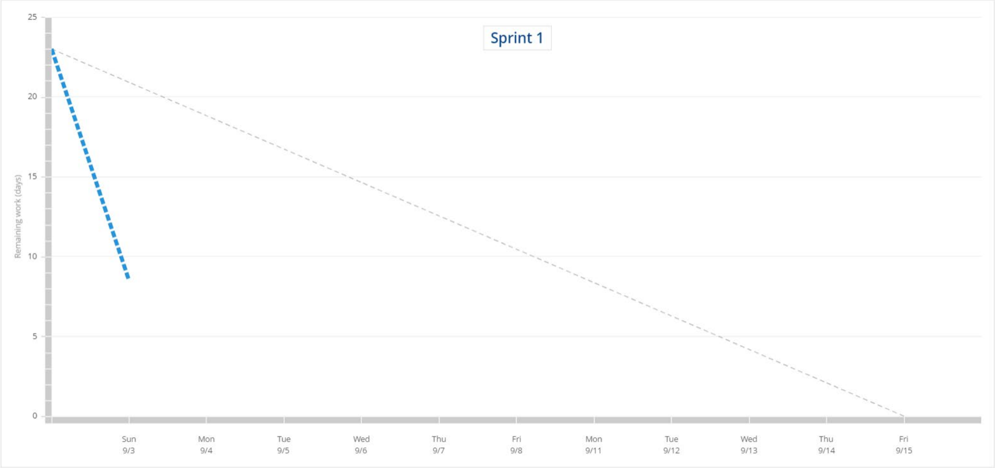
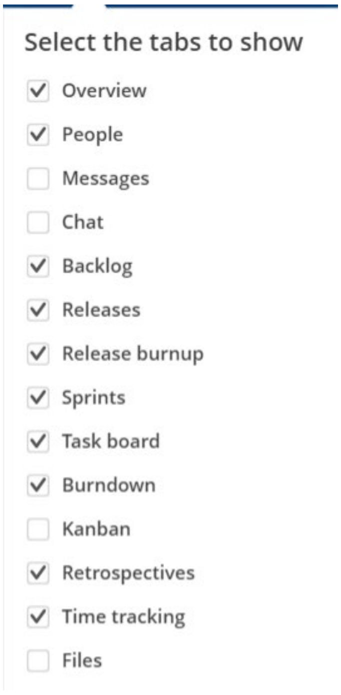


Figure 2.58: Burndown Chart



Select the tabs to show

☑ Overview
☑ People
□ Messages
□ Chat
☑ Backlog
☑ Releases
☑ Release burnup
☑ Sprints
☑ Task board
☑ Burndown
□ Kanban
☑ Retrospectives
☑ Time tracking
□ Files

Figure 2.59: Metrics Available

SCRUMwise offers a wide range of autogenerated metrics for the user to view. These metrics includes burndown chart and burnup chart, along with a timeline shown.

Filtering

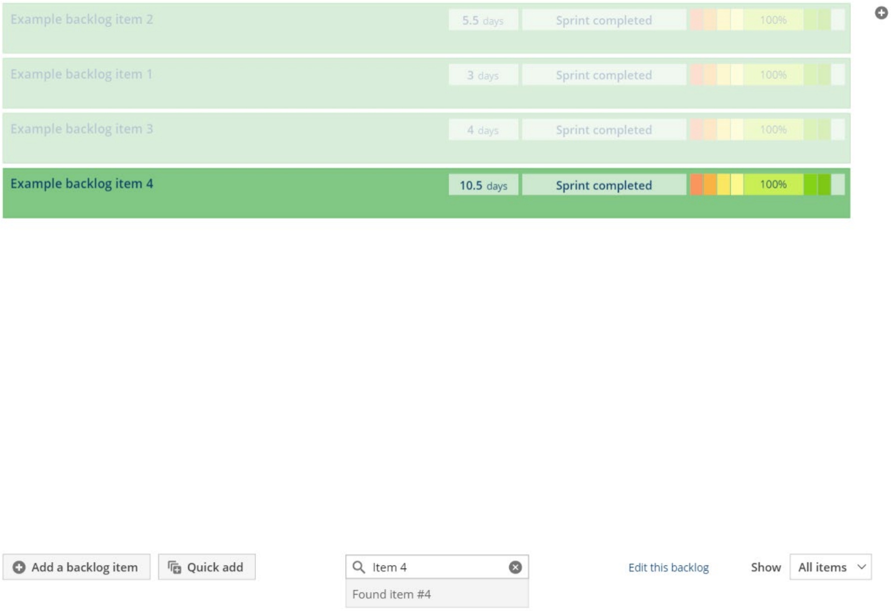


Figure 2.60: Search Function

SCRUMwise allow users to enter keyword to search and will show result accordingly.

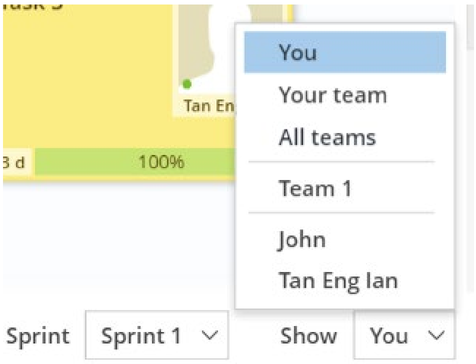


Figure 2.61: Filter Option

SCRUMwise allows the user to filter the content of the pages depending on the choices of users.

2.3.5.2 Advantages & Drawbacks of SCRUMwise

Advantages

1. SCRUMwise can be regarded as one of the most user-friendly applications for understanding and implementing SCRUM practices. Its

interface enables users to grasp its functionalities with ease. This user-friendly design minimizes the learning curve, eliminating the need for an extended period to become proficient in its usage.

2. SCRUMwise is customizable in various aspects such as SCRUM board, estimation, backlogs, etc., allowing for flexibility in managing SCRUM projects.

3. SCRUMwise offers a wide range of metrics that is ready to be used such as burndown charts and burnup charts, enhancing the ability to track a Sprint timeline.

4. In SCRUMwise, most of the processes such as generating SCRUM boards, timeline and metrics are all automated.

Drawbacks

1. There are bugs present within SCRUMwise. For example, if a user adds a non-registered SCRUMwise user, SCRUMwise will automatically create an account for them. Consequently, the newly added user is required to use the "Forgot Password" section to access their account, even though they haven't signed up themselves.

2. SCRUMwise uses days for estimating all efforts. While user story points are available, SCRUMwise restricts users from estimating their backlogs with days before assigning it to a sprint. This restriction limits user flexibility in sprint estimation.

3. SCRUMwise requires subscription fees to access all of its features. Unlike Jira and Trello, SCRUMwise is a pay-to-use application without any free features available.

4. SCRUMwise lacks robust user role and project management features. When a user adds another user to a project, SCRUMwise allows the added user to view and edit all other projects that the owner is working

on, leading to unnecessary invitations to unrelated projects. Additionally, SCRUMwise only offers two roles: Administrator and Team Member. However, the team member role possesses the same authority as the administrator in terms of project management, except for the ability to add and remove team members.

5. SCRUMwise forces and restrict a lot of user inputs, which does not suits the characteristic of SCRUM projects.

2.3.6 Analysis of Existing Applications

Table 2.2: Comparison Matrix between Existing Applications Reviewed

Application Features	**Jira**	**Trello**	**Microsoft Excels**	**Google Sheets**	**SCRUMwise**
Customizability	High	High	Intermediate	Intermediate	Medium
SCRUM Project Setup	Auto	Manual but have templates	Manual but have templates	Manual but have templates	Auto
Role Management	High (Paid Version) No (Free Version)	High (Paid Version) No (Free Version)	Low	Low	Low and concerning
Sprint Management	Can select backlogs and initiate sprints automatically	Manual setup	Manual setup	Manual setup	Can select backlogs and initiate sprints automatically
Estimation, Timeline, Burndown & Burnup Charts	story points, time estimates, and more with auto generated charts	Depends on plugins	Manual entry and calculation. Highly customizable but requires setup	Manual entry and calculation. Highly customizable but requires setup	Story points and time estimates and auto generated charts.
Backlogs Management	Available and highly customizable, but hard to track if backlog completed	Manual setup and highly customizable	Manual setup and highly customizable	Manual setup and highly customizable	Available and highly customizable

SCRUM Board Generation & Configuration	Auto generated and highly customizable	Manual setup and highly customizable	Manual setup and highly customizable	Manual setup and highly customizable	Auto generated and high customizable
Filtering	Available	Available	Available	Available	Available
Pricing	Free or $7.75 / user / month	Free or $10 / user / month	$6.99 per license / month	Free	$7.50 / user / month
Complexity	High	Low	Intermediate	Intermediate	Low
Usage of RTM	No	No	Can be done with setup	Can be done with setup	No

SCRUM Project Setup

Jira and SCRUMwise are dedicated to managing SCRUM projects, unlike Trello, Microsoft Excel, and Google Sheets. Consequently, Jira and SCRUMwise offer pre-built templates for SCRUM projects, while the other applications require manual setup. Given that the final year project is also focused on managing SCRUM projects, its setup should align with that of Jira and SCRUMwise.

Role Management

All five existing applications enable users to add new team members to their projects. However, SCRUM projects often require varying levels of authority, such as restricting team members from modifying product backlogs. To ensure effective role management, it's essential to implement clear levels of authority. Jira and Trello provide extensive role management features but requiring payment. On the other hand, the remaining three lack formal role management, allowing only limited editing authority. This approach lacks flexibility in defining specific editing restrictions. To address this, the final year project should emulate the role management features found in the paid versions of Jira and Trello.

Sprint Management

Jira and SCRUMwise allow users to select dedicated backlog items and add them to sprints. The status of backlogs can also be updated, reflecting changes throughout the software. However, Microsoft Excel, Google Sheets, and Trello do not visibly support sprint initiation due to their manual setup requirements. The final year project should emphasize a sprint initiation approach similar to that of Jira and SCRUMwise.

Estimation, Timeline, Burndown, and Burnup Chart Implementation

All five existing applications incorporate estimation, timeline, burndown, and burnup charts to provide users with insights into project progress. By employing these tools, the applications enhance project tracking. However, these tools currently focus solely on sprint timelines rather than overall project completion timelines. Improvements can be made by extending these features to encompass the entire project timeline, rather than solely focusing on sprints.

Backlogs Management

Each of the five applications allows users to create and customize backlogs, including assigning priority and estimation. These features can enhance the final year project's backlog management capabilities. However, even with provided templates, the applications lack a clear distinction between product and sprint backlogs items. While this approach enhances flexibility in sprint creation, it may lead to user confusion. To address this, the final year project should consider introducing a clear distinction between product and sprint backlogs items, while still maintaining flexibility in sprint creation.

SCRUM Board Management and Configuration

Jira and SCRUMwise facilitate the automatic creation of SCRUM boards when initiating a sprint. A similar feature may be achievable with Microsoft Excel and Google Sheets through proper setup. However, Trello requires manual setup. Furthermore, all applications offer highly customizable SCRUM board configurations. The final year project should adopt an approach similar to that of SCRUMwise and Jira for SCRUM board management and configuration.

Filtering

All five applications support filtering in various sections, which could be implemented by final year project for easier tracking of project progress.

Given the project's focus on SCRUM project, leveraging existing features as reference points for the web application's development is a strategic approach. This allow capitalize on the strengths of current solutions while addressing their limitations through improvements.

Nonetheless, the main objective of the final year project is the tracking and forecasting the project completion date in SCRUM using RTM. However, RTM is not being applied in any of the existing applications even with the Microsoft Excel or Google Sheet templates, leaving no existing feature for final year project to reference.

On the other hand, all five applications utilize tools like burndown charts, burnup charts, estimation, and timelines to assist users in monitoring sprint progress. The final year project can combine these existing features with RTM and expand its limitation to monitoring the whole project rather than just a sprint, allowing the tracking and forecasting completion dateline using RTM to be implemented.

2.4 Techniques and Approaches

In this section of the literature review, the focus is on existing techniques and approaches that are commonly employed in the software industry for tracking and forecasting project completion timelines. Specifically, this review will cover Planning Poker and the Requirement Traceability Matrix (RTM). These techniques and approaches will be referenced and used in the final year project web application.

2.4.1 Planning Poker

Planning Poker has emerged as a highly popular Agile estimation technique in the software industry in recent years. Inspired by the "Wideband Delphi" estimation method, James Grenning refined the technique, which was later popularized by Mike Cohn. Although it is not formally part of the official SCRUM Guide, Planning Poker is extensively utilized in both SCRUM and other Agile methodologies to estimate story points for backlog items. (Mahnic and Hovelja, 2012).

2.4.1.1 How Does Planning Poker implemented in Agile Methodology?

Planning Poker is an estimation technique commonly used by Agile teams to assess the story points of backlog items. The process unfolds as follows:

1. **Estimation Cards:** Each participant on the Agile team is equipped with a set of estimation cards displaying a sequence of numbers. This sequence can be anything, such as prime numbers or multiples. However, the Fibonacci sequence are the most commonly used sequence.

2. **User Story Selection:** A user story is selected from the backlog, which is then presented to the team. The Product Owner is present to provide clarifications and address any uncertainties regarding the chosen user story.

3. **Individual Estimations:** Each team member selects an estimation card that represents their estimation of the effort required to complete the user story presented. This card will not be revealed to anyone.

4. **Simultaneous Reveal:** Upon making their choices, team members simultaneously reveal their selected estimation cards.

5. **Consensus or Debate:** If everyone reveals the same card number, that number is chosen as the estimation for the story points. However, if there are differing estimations, team member justifies for their choices, leading to a debate or discussion.

6. **Re-estimation:** Following the discussion, a new round of estimation takes place for the same user story.

7. **Iteration:** This process repeats until all team members converge on the same estimation number for the user story.

This process will be repeated until all the user story has been estimated (Hartman, 2009).

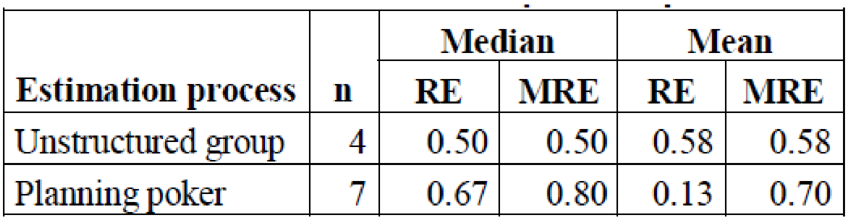
2.4.1.2 Advantages and Drawbacks of Planning Poker

Planning Poker has been proven by many studies to play a significant role in estimating within the Agile framework. It helps increase the accuracy of the overall user story estimation process due to its collaborative nature, involving the entire team in discussions. This collaborative approach leads to a consensus estimation of user story points, as opposed to relying on a single person to summarize the entire estimation process (Hartman, 2009).

In a study conducted by Moløkken-Østvold, Haugen, and Benestad in 2008, they found that while Planning Poker does require more time for estimation compared to a statistical combination of individual estimations with means of 6.3 hours and 7.1 hours, respectively, the group consensus estimation has been proven to be more accurate than the statistical combination of individual expert estimations. This is evident from a Balance Relative Error (BRE) of 0.82 and 0.94. To put it simply, a higher BRE value indicates lower accuracy. Furthermore, in a study by Mahnic and Hovelja in 2012, Planning Poker estimates provided by experienced professionals exhibit less pessimism, with a median BRE bias of -0.2231 and greater accuracy, with a median BRE of 0.3260, when compared to the statistical combination of professional individual estimations with a median BRE bias of -2.445 and a median BRE of 0.3835.

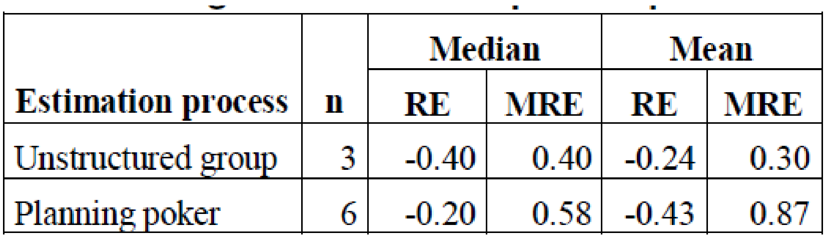
With the group consensus concept of Planning Poker, it compels the group to engage in an iterative process of discussion until a consensus estimation is reached. This approach has proven to be effective in identifying sub-tasks and challenges within the user stories, thanks to the mandatory discussions within the team, allowing the estimation of efforts to be more accurate (Moløkken-Østvold, Haugen, and Benestad, 2008).

However, Planning Poker necessitates the involvement of professionals or individuals with experience to fully utilize its potential. In cases where a group lacks experience, Planning Poker proves to be less accurate. This assertion finds support in a research paper by Haugen in 2006. The study compares the accuracy of effort estimation using Unstructured Group methods and Planning Poker within groups with prior experience and those lacking experience using Relative Error (RE).



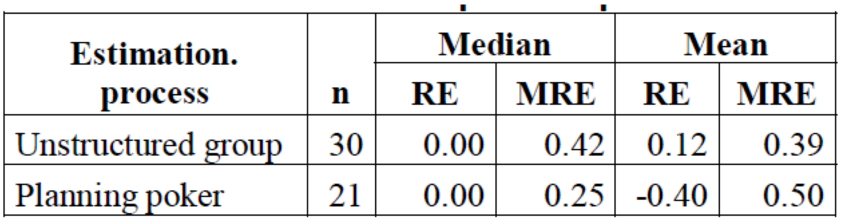
Estimation process	n	Median		Mean	
		RE	MRE	RE	MRE
Unstructured group	4	0.50	0.50	0.58	0.58
Planning poker	7	0.67	0.80	0.13	0.70

Figure 2.62: RE of Small Tasks without Prior Experience (Haugen, 2006).



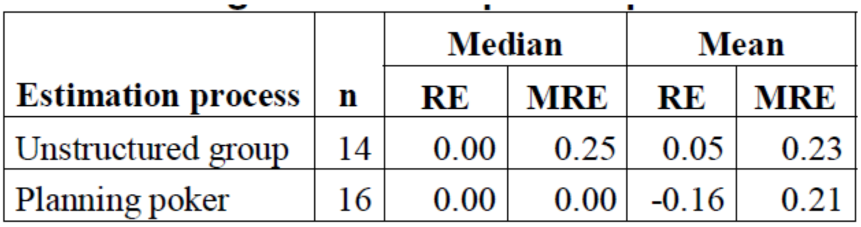
Estimation process	n	Median		Mean	
		RE	MRE	RE	MRE
Unstructured group	3	-0.40	0.40	-0.24	0.30
Planning poker	6	-0.20	0.58	-0.43	0.87

Figure 2.63: RE of Large Tasks without Prior Experience (Haugen, 2006).



Estimation. process	n	Median		Mean	
		RE	MRE	RE	MRE
Unstructured group	30	0.00	0.42	0.12	0.39
Planning poker	21	0.00	0.25	-0.40	0.50

Figure 2.64: RE of Small Tasks with Prior Experience (Haugen, 2006).



Estimation process	n	Median		Mean	
		RE	MRE	RE	MRE
Unstructured group	14	0.00	0.25	0.05	0.23
Planning poker	16	0.00	0.00	-0.16	0.21

Figure 2.65: RE of Large Tasks with Prior Experience (Haugen, 2006).

Similar to the BRE mentioned earlier, smaller values of RE and MRE indicate higher accuracy. Analyzing the four figures presented above leads to the conclusion that there is a significant disparity between groups with experience and those without experience when employing both Planning Poker and the Unstructured Group method, regardless of the task size. Individuals lacking experience demonstrate better estimation capabilities using the Unstructured Group Method, whereas those with experience exhibit better estimation accuracy when using Planning Poker. This divergence can be attributed to the fact that experienced individuals engaged in Planning Poker sessions conduct more in-depth analyses of user stories and share their insights, leveraging their expertise to enhance the estimation process. Each participant's distinct technical background and extensive domain knowledge contribute to superior estimation performance (Haugen, 2006).

Furthermore, as mentioned earlier and referencing the research paper by Mahnic and Hovelja in 2012, it is worth noting that university students exhibit lower accuracy when performing estimations using Planning Poker compared to the statistical combination of their individual estimates. This is evident through a median BRE of 0.6667 in Planning Poker, as opposed to a median BRE of 0.6 in the statistical combination method. The reduced accuracy observed with Planning Poker in this context could potentially be attributed to the fact that the group of students shares a similar background and knowledge base. This similarity limits their ability to explore a diverse perspective during group discussions, ultimately affecting the effectiveness of Planning Poker (Mahnic and Hovelja, 2012).

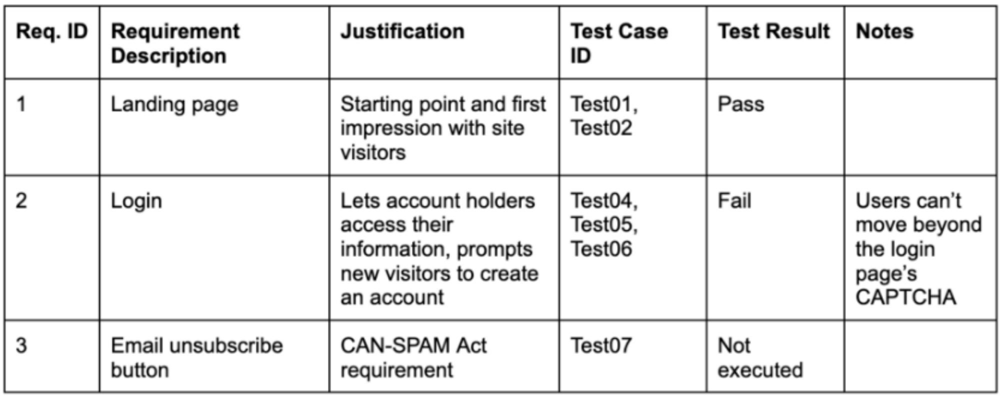
2.4.1.3 Why Estimation Matters

Estimating story points in SCRUM matters for several reasons. First, it fosters a shared understanding among team members about the scope and complexity of work for each user story or backlog items, enhancing collaboration and setting realistic expectations. Story point estimation also enables better sprint planning, as it allows the team to gauge how much work can be feasibly committed to in a given sprint. This helps in managing stakeholder expectations by providing more accurate forecasts of what will be delivered and when. Additionally, story points act as a metric that facilitates continuous improvement within the SCRUM team. By comparing estimated story points to actual effort over time, teams can identify areas where they might be overcommitting or underperforming and make the necessary adjustments for future sprints. Overall, story point estimation contributes to more effective planning, improved team communication, and a more predictable, sustainable pace of development.

In this final year project, story point estimation plays a foundational role in assessing the scope and workload for all SCRUM activities. These estimates serve as the basis for generating two key performance metrics: Sprint Burndown and Sprint Burnup charts. A sprint burndown chart displays the amount of work remaining in the sprint backlog over time, providing a quick visual representation of the team's progress and how much work is left to be completed. On the other hand, a sprint burnup chart shows the work completed over time against the total scope of work, offering insights into scope changes and the project’s rate of progress. Specialized software platforms, including Jira and SCRUMwise, depend on accurate estimation efforts for the creation of these insightful metrics. High accuracy in story point estimates directly contributes to more reliable and transparent project completion forecasts, meeting the project's primary goal of forecasting and tracking project completion datelines more precisely.

2.4.2 Requirement Traceability Matrix

The Requirement Traceability Matrix (RTM) is a comprehensive document that establishes a clear linkage between user and system requirements along with their attributes It acts as a bridge between the stakeholder requirements and the project objective, ensuring the final expectation is met and allowing the easy tracking of project progress. By providing an overview of the project requirements, the RTM offers immediate insight and understanding of the entire project elements. It fulfills a range of functions, encompassing scope validation, verification of requirement fulfillment, and the support of quality assurance and control processes. By providing a clear visualization of the relationships between different project elements, the RTM helps in identifying dependencies, managing changes, and ensuring that the project meets the defined requirements and quality standards.



Req. ID	Requirement Description	Justification	Test Case ID	Test Result	Notes
1	Landing page	Starting point and first impression with site visitors	Test01, Test02	Pass	
2	Login	Lets account holders access their information, prompts new visitors to create an account	Test04, Test05, Test06	Fail	Users can't move beyond the login page's CAPTCHA
3	Email unsubscribe button	CAN-SPAM Act requirement	Test07	Not executed	

Figure 2.66: Requirement Traceability Matrix

The Requirement Traceability Matrix (RTM) is a simple yet powerful tool. It isn't bounded by any guidelines, but rather, it can be adapted to users' needs, allowing customization. Presenting a straightforward table format, it significantly enhances requirement traceability by offering an overview of the essential details of the project. Surprisingly, despite its potential benefits, RTM hasn't found widespread adoption, as evidenced by its infrequent utilization within existing applications of SCRUM management reviewed.

In the SCRUM framework, requirements are primarily described through sprint backlogs and product backlogs, omitting formal documentation due to the dynamic nature of requirement changes. This dynamic nature is useful yet present challenges. As per Budiman, Raharjo, and Suhanto (2020), the SCRUM methodology faces challenges in establishing sufficient traceability between user stories and requirements, leading to less documentation of test outcomes and quality requisites. Shahzad et al. (2021) also found that requirement engineering poses challenges in SCRUM due to the absence of formal documentation, which impacts requirement visibility and prioritization.

Amid these persistent challenges, RTM presents a potential solution. Jeong, Cho, and Lee (2018) suggest an innovative approach: auto-generating RTM to manage Agile methodology requirement traceability. Given the frequent requirement changes in SCRUM, RTM's adaptability allows for automatic updates, eliminating the need for manual revisions. This addresses the documentation issue by making the process automatic and effortless, aligning well with the dynamic nature of SCRUM. While not a complete solution, implementing RTM can potentially mitigate these challenges and streamline requirement traceability in SCRUM projects, allowing the forecasting and tracking of project completion dateline to be clearer.

2.5 Conclusion

This literature review has reviewed three different topics. Firstly, it discusses the software development process and includes a review of three different SDLC methodologies: Waterfall, SCRUM, and Prototype. An analysis has been conducted to determine the most suitable SDLC for the final year project.

The second topic focuses on existing applications, namely Jira, Trello, Microsoft Excel, Google Sheets, and SCRUMwise. Each application's core functions in SCRUM project management have been reviewed and referenced for potential application in the web application developed for the final year project.

Lastly, the review delves into techniques and approaches to address challenges in the final year project. Specifically, it explores the utilization of Planning Poker and requirement traceability matrix.

CHAPTER 3

METHODOLOGY AND WORK PLAN

3.0 Introduction

This chapter will discuss about the methodology and work plan set forth for the final year project. Section 3.1 highlights the phases of the SDLC that have been adopted. Section 3.2 will present the detailed work plan for the final year project, focusing specifically on the Gantt Chart and Work Breakdown Structure (WBS). Meanwhile, Section 3.3 will elaborate on the tools chosen for the development of the final year project.

3.1 Prototype Methodology

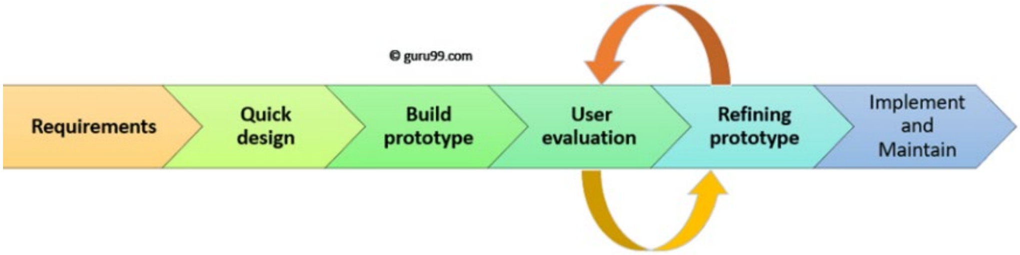


Figure 3.1: Prototyping Model SDLC Phases (Martin, 2023).

Based on the analysis conducted in the literature review, the prototype methodology has been selected as the SDLC for the development of the final year project's web application. In general, it comprises six phases: Requirements, Quick Design, Prototype Building, User Evaluation, Prototype Refinement, and Implementation and Maintenance. However, this is just a general guideline which are not strict to follow. The SDLC methodology will be refined according to the needs of the final year project.

3.1.1 Requirement Gathering and Analysis

The initial phase of the final year project focuses on requirement gathering and analysis. During this stage, key components such as the project title, background, problem statement, scope, approach, solution, and objective are identified. This establishes a foundational understanding of the project, ensuring that the project's intended outcome is clearly expressed and offering insight into the tasks ahead.

A literature review will be conducted on three distinct topics: SDLC Methodology, Existing Applications, and Tools and Techniques. The SDLC Methodology segment will offer insights into three different methodologies: Waterfall, Prototype, and SCRUM. Each methodology will be assessed based on its phases as well as its respective advantages and disadvantages. An analysis will be carried out to determine the most suitable SDLC methodology for the final year project, considering various factors such as being a lone developer, limited timelines, and resource constraints.

Besides that, five existing applications, namely Jira, Trello, Google Sheets, Microsoft Excel, and SCRUMwise, will also be examined. Each application will be reviewed for its benefits and limitations. These reviews will serve as a reference during the requirement gathering phase, as the final year project aims to develop an enhanced version of these existing applications, particularly in terms of tracking and forecasting project completion timelines.

Lastly, two techniques and approaches, Planning Poker and Requirement Traceability Matrix, will be explored. These techniques and approaches are crucial for enhancing the tracking and forecasting of project completion dates in a SCRUM environment. Finally, a discussion session will be arranged with the final year project supervisor to pinpoint the requirements for the web application.

3.1.2 Planning

Based on the requirements analyzed and identified in the previous phase, a detailed schedule and work breakdown structure will be constructed to improve the visibility of the final year project's timeline. Additionally, the development tools to be utilized will be identified, allowing for preliminary exploration before the commencement of the actual development process. Although the prototype methodology is iterative and doesn't typically adhere to a fixed schedule, the final year project's development phase is constrained to 12 weeks. Therefore, the number of iterations can be predetermined allowing the schedule creation.

3.1.3 Quick Design

With the requirements identified, this phase will produce a quick design of the web application, serving as a foundational reference for building the prototype. While this design won't be interactive, it offers an architectural overview, illustrating the integration between the front-end and back-end. Additionally, it will showcase the UI/UX design. Once created, the supervisor will evaluate the design. Feedback from the supervisor will be considered in refining the design, ensuring the web application is user-friendly for SCRUM practitioners and that all essential functions are included.

3.1.4 Iteration

The final year project will set the prototype methodology to include three iterations. Each iteration will concentrate on distinct tasks, adding more features and functionality to the final prototype.

3.1.4.1 Iteration 1: Basic Functionality

In the first iteration, an initial prototype will be developed based on the quick design. This prototype will feature basic functionalities of the web application, including the login, logout, registration, project creation, and team member addition and remove functions. The web application's database will also be structured during this phase. A navigation system, connecting all features, will be set up. Pages that are not developed will be leave blank. Following its construction, both the prototype and database will be evaluated in a session with the supervisor. Based on feedback, necessary refinements will be made to the prototype. This cycle of evaluation and refinement will continue until the supervisor is content with the web application's functionality and UI.

3.1.4.2 Iteration 2: User Management, Backlog Management and Sprint Management

In the second iteration, building upon the first prototype, the emphasis will shift to the user management. This involves user role management, adjusting and defining user access rights within the web application. With user roles defined, the backlog management will be developed to ensure that backlog items can be added, with only authorized users having the ability to add and modify them. Additionally, the capability of the web application to auto-generate a RTM and customized it will also be implemented. Following the backlog setup, sprint management will become the focal point. Features such as the auto-generation and customization of the SCRUM board will be developed. Furthermore, the database will be integrated with the application's UI to validate and verify the logical flow of the web application. This iteration is to ensure all the core functions of the web application are developed. The iteration will be subjected to another iterative round of user evaluation and prototype refinement.

3.1.4.2 Iteration 3: Estimations

The third iteration represents the final stage of the development process. Building on the foundation of the second prototype, this iteration will primarily concentrate on the estimation of the project completion dateline. Features such as the planning poker game, and the generation of burndown and burnup charts, will be developed during this phase. The mechanism for estimating the project's completion date will also be introduced. The core objective of this iteration is to validate the accuracy and appropriateness of the estimation techniques and algorithms employed in the backend. Once developed, the prototype will be submitted for user evaluation by the supervisor to assess the efficiency of the chosen estimation techniques. Based on feedback, any necessary refinements will be made.

3.1.5 Implementation

The implementation phase marks the end of the final year project. During this phase, the prototype from the last iteration is polished and transformed into a ready-to-deploy application. A crucial part of this phase is testing, which ensures that every functionality operates as expected and helps in detecting any potential defects that could affect the performance of the web application. The web application will undergo three distinct types of testing:

Unit Testing

Unit testing is a method where individual units or components of the application are tested individually. This ensures that every function performs its intended task effectively. The unit testing will be performed on the web application as a whole, where the backend and frontend integration are developed and integrated together.

Usability Testing

Usability testing plays a significant role. It records user satisfaction levels concerning the system. Testers can be provided with a checklist to ensure comprehensive assessment of every function, and the feedback can be collected, offering insights into the system's efficiency and effectiveness.

User Acceptance Testing

User acceptance testing acts as the final barrier before the web application's release. It confirms that the web application aligns with the defined requirements and meet the stakeholders’ expectation.

3.2 Schedule

3.2.1 Schedule Table

Table 3.1: Schedule of Final Year Project

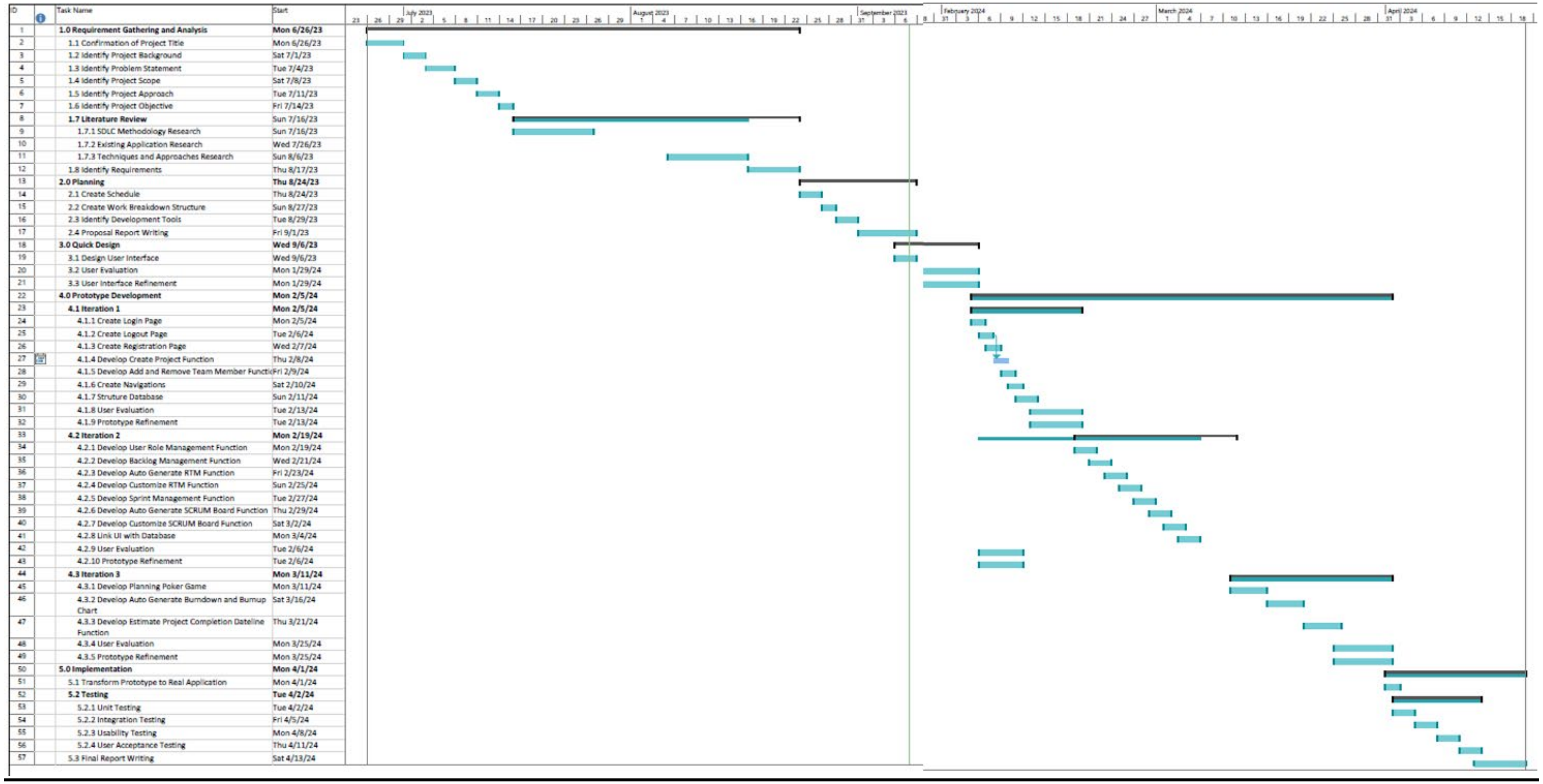
Task Name	Start	Finish
1.0 Requirement Gathering and Analysis	**Mon 6/26/23**	**Wed 8/23/23**
1.1 Confirmation of Project Title	Mon 6/26/23	Fri 6/30/23
1.2 Identify Project Background	Sat 7/1/23	Mon 7/3/23
1.3 Identify Problem Statement	Tue 7/4/23	Fri 7/7/23
1.4 Identify Project Scope	Sat 7/8/23	Mon 7/10/23
1.5 Identify Project Approach	Tue 7/11/23	Thu 7/13/23
1.6 Identify Project Objective	Fri 7/14/23	Sat 7/15/23
1.7 Literature Review	Sun 7/16/23	Wed 8/23/23
1.7.1 SDLC Methodology Research	Sun 7/16/23	Wed 7/26/23
1.7.2 Existing Application Research	Wed 7/26/23	Sun 8/6/23
1.7.3 Techniques and Approaches Research	Sun 8/6/23	Wed 8/16/23
1.8 Identify Requirements	Thu 8/17/23	Wed 8/23/23
2.0 Planning	**Thu 8/24/23**	**Fri 9/8/23**
2.1 Create Schedule	Thu 8/24/23	Sat 8/26/23
2.2 Create Work Breakdown Structure	Sun 8/27/23	Mon 8/28/23
2.3 Identify Development Tools	Tue 8/29/23	Thu 8/31/23
2.4 Proposal Report Writing	Fri 9/1/23	Fri 9/8/23
3.0 Quick Design	**Wed 9/6/23**	**Mon 2/5/24**
3.1 Design User Interface	Wed 9/6/23	Fri 9/8/23
3.2 User Evaluation	Mon 1/29/24	Mon 2/5/24
3.3 User Interface Refinement	Mon 1/29/24	Mon 2/5/24
4.0 Prototype Development	**Mon 2/5/24**	**Mon 4/1/24**
4.1 Iteration 1	**Mon 2/5/24**	**Mon 2/19/24**
4.1.1 Create Login Page	Mon 2/5/24	Tue 2/6/24
4.1.2 Create Logout Page	Tue 2/6/24	Wed 2/7/24
4.1.3 Create Registration Page	Wed 2/7/24	Thu 2/8/24
4.1.4 Develop Create Project Function	Thu 2/8/24	Fri 2/9/24
4.1.5 Develop Add and Remove Team Member Function	Fri 2/9/24	Sat 2/10/24
4.1.6 Create Navigations	Sat 2/10/24	Sun 2/11/24
4.1.7 Structure Database	Sun 2/11/24	Tue 2/13/24
4.1.8 User Evaluation	Tue 2/13/24	Mon 2/19/24
4.1.9 Prototype Refinement	Tue 2/13/24	Mon 2/19/24
4.2 Iteration 2	**Mon 2/19/24**	**Mon 3/11/24**
4.2.1 Develop User Role Management Function	Mon 2/19/24	Wed 2/21/24
4.2.2 Develop Backlog Management Function	Wed 2/21/24	Fri 2/23/24
4.2.3 Develop Auto Generate RTM Function	Fri 2/23/24	Sun 2/25/24

4.2.4 Develop Customize RTM Function	Sun 2/25/24	Tue 2/27/24
4.2.5 Develop Sprint Management Function	Tue 2/27/24	Thu 2/29/24
4.2.6 Develop Auto Generate SCRUM Board Function	Thu 2/29/24	Sat 3/2/24
4.2.7 Develop Customize SCRUM Board Function	Sat 3/2/24	Mon 3/4/24
4.2.8 Link UI with Database	Mon 3/4/24	Wed 3/6/24
4.2.9 User Evaluation	Tue 2/6/24	Sun 2/11/24
4.2.10 Prototype Refinement	Tue 2/6/24	Sun 2/11/24
4.3 Iteration 3	**Mon 3/11/24**	**Mon 4/1/24**
4.3.1 Develop Planning Poker Game	Mon 3/11/24	Fri 3/15/24
4.3.2 Develop Auto Generate Burndown and Burnup Chart	Sat 3/16/24	Wed 3/20/24
4.3.3 Develop Estimate Project Completion Dateline Function	Thu 3/21/24	Mon 3/25/24
4.3.4 User Evaluation	Mon 3/25/24	Mon 4/1/24
4.3.5 Prototype Refinement	Mon 3/25/24	Mon 4/1/24
5.0 Implementation	**Mon 4/1/24**	**Fri 4/19/24**
5.1 Transform Prototype to Real Application	Mon 4/1/24	Tue 4/2/24
5.2 Testing	**Tue 4/2/24**	**Sat 4/13/24**
5.2.1 Unit Testing	Tue 4/2/24	Thu 4/4/24
5.2.2 Integration Testing	Fri 4/5/24	Sun 4/7/24
5.2.3 Usability Testing	Mon 4/8/24	Wed 4/10/24
5.2.4 User Acceptance Testing	Thu 4/11/24	Sat 4/13/24
5.3 Final Report Writing	Sat 4/13/24	Fri 4/19/24

In the provided schedule, it can be seen that quick design phase spans from September 2023 to February 2024. This extended duration is because the interface design will be created during FYP1, while the evaluation will be conducted in FYP2. For the first iteration, it's allocated only 2 weeks times, due to the functionalities for this phase are straightforward and easy to implement. The subsequent iterations receive 3 weeks of time, reflecting their increased complexity. While the third iteration might seem to have fewer tasks, its focus is on the estimation techniques, which will likely demand extensive refinement to enhance the efficiency and accuracy. Therefore, more time is allocated ensuring more attention are given to estimations, even if the task list appears shorter. It is also important to note that the date given is note equivalent to the effort available.

From schedule 1.0 to 3.0, it aligns with objective 1 of the final year project, wherein research on the tools and techniques for forecasting and tracking project completion datelines will be executed and analyzed to determine how they can be applied in the development of the final year project's web application. During the development phase, as stated in 4.0, objective 2 of the final year project will be achieved, as the full web application will be developed. Objective 3 will be achieved in schedule 5.0, where the web application testing will undergo multiple tests to ensure that the project achieves a satisfactory rate of at least 80%.

3.2.2 Gantt Chart



ID	ℹ	Task Name	Start
1		**1.0 Requirement Gathering and Analysis**	**Mon 6/26/23**
2		1.1 Confirmation of Project Title	Mon 6/26/23
3		1.2 Identify Project Background	Sat 7/1/23
4		1.3 Identify Problem Statement	Tue 7/4/23
5		1.4 Identify Project Scope	Sat 7/8/23
6		1.5 Identify Project Approach	Tue 7/11/23
7		1.6 Identify Project Objective	Fri 7/14/23
8		**1.7 Literature Review**	Sun 7/16/23
9		1.7.1 SDLC Methodology Research	Sun 7/16/23
10		1.7.2 Existing Application Research	Wed 7/26/23
11		1.7.3 Techniques and Approaches Research	Sun 8/6/23
12		1.8 Identify Requirements	Thu 8/17/23
13		**2.0 Planning**	**Thu 8/24/23**
14		2.1 Create Schedule	Thu 8/24/23
15		2.2 Create Work Breakdown Structure	Sun 8/27/23
16		2.3 Identify Development Tools	Tue 8/29/23
17		2.4 Proposal Report Writing	Fri 9/1/23
18		**3.0 Quick Design**	**Wed 9/6/23**
19		3.1 Design User Interface	Wed 9/6/23
20		3.2 User Evaluation	Mon 1/29/24
21		3.3 User Interface Refinement	Mon 1/29/24
22		**4.0 Prototype Development**	**Mon 2/5/24**
23		**4.1 Iteration 1**	**Mon 2/5/24**
24		4.1.1 Create Login Page	Mon 2/5/24
25		4.1.2 Create Logout Page	Tue 2/6/24
26		4.1.3 Create Registration Page	Wed 2/7/24
27	▦	4.1.4 Develop Create Project Function	Thu 2/8/24
28		4.1.5 Develop Add and Remove Team Member Functic	Fri 2/9/24
29		4.1.6 Create Navigations	Sat 2/10/24
30		4.1.7 Struture Database	Sun 2/11/24
31		4.1.8 User Evaluation	Tue 2/13/24
32		4.1.9 Prototype Refinement	Tue 2/13/24
33		**4.2 Iteration 2**	**Mon 2/19/24**
34		4.2.1 Develop User Role Management Function	Mon 2/19/24
35		4.2.2 Develop Backlog Management Function	Wed 2/21/24
36		4.2.3 Develop Auto Generate RTM Function	Fri 2/23/24
37		4.2.4 Develop Customize RTM Function	Sun 2/25/24
38		4.2.5 Develop Sprint Management Function	Tue 2/27/24
39		4.2.6 Develop Auto Generate SCRUM Board Function	Thu 2/29/24
40		4.2.7 Develop Customize SCRUM Board Function	Sat 3/2/24
41		4.2.8 Link UI with Database	Mon 3/4/24
42		4.2.9 User Evaluation	Tue 2/6/24
43		4.2.10 Prototype Refinement	Tue 2/6/24
44		**4.3 Iteration 3**	**Mon 3/11/24**
45		4.3.1 Develop Planning Poker Game	Mon 3/11/24
46		4.3.2 Develop Auto Generate Burndown and Burnup Chart	Sat 3/16/24
47		4.3.3 Develop Estimate Project Completion Dateline Function	Thu 3/21/24
48		4.3.4 User Evaluation	Mon 3/25/24
49		4.3.5 Prototype Refinement	Mon 3/25/24
50		**5.0 Implementation**	**Mon 4/1/24**
51		5.1 Transform Prototype to Real Application	Mon 4/1/24
52		**5.2 Testing**	**Tue 4/2/24**
53		5.2.1 Unit Testing	Tue 4/2/24
54		5.2.2 Integration Testing	Fri 4/5/24
55		5.2.3 Usability Testing	Mon 4/8/24
56		5.2.4 User Acceptance Testing	Thu 4/11/24
57		5.3 Final Report Writing	Sat 4/13/24

3.3 Development Tools

Vue.js is used as the front-end application, a framework that utilizes JavaScript, html and css, allows users to code in OOP to facilitate the reusability of code implementation. Vuex, as part of Vue.js, will be utilized to store the frontend session state. This is to enable efficient management of application state and data flow within the frontend components.

Laravel is an extension framework to PHP, known for its MVC architecture, serves as the backend language for the web application. By adhering to the MVC pattern, Laravel simplifies the development process, making modifications and maintenance more manageable over time. Additionally, Laravel comes with built-in authentication and authorization mechanisms. These features enhance the security of the web application by providing robust user authentication and access control functionalities out of the box. By leveraging Laravel's authentication system, developers can implement secure login and user management features with ease, reducing the risk of security vulnerabilities and ensuring a safer user experience.

JWT Token is used for API token authentications communication between the frontend and backend. The frontend receives this JWT and includes it in subsequent requests to the backend. The backend verifies the JWT signature to ensure its authenticity and extracts the user information from the payload.

MariaDB serves as the database for this application. MariaDB presents numerous advantages that set it apart from MySQL and other open-source RDBMS. For example, MariaDB places a strong emphasis on security by addressing critical security concerns and incorporating encryption functionalities to safeguard databases. Additionally, it boasts a dynamic thread pool feature that optimizes server resources to enhance speed, improve replication, and quicker updates (Alexandrea, 2021).

Xampp is used to provide the service of MariaDB. It sets up an environment on the local machine that simulates a web server environment.

It also simplifies the process of setting up a local web server environment and make it easier for developers to work on their projects locally.

Microsoft Visual Studio is used as the IDE for web application development. Visual Studio is preferred over other IDEs because it allows easy integration with extension packages such as CoPilot, which incorporates AI assistance in the coding process, increasing the efficiency of coding the web application. Besides that, Visual Studio provides a formatter, which automatically organizes all the code in a neat way.

CHAPTER 4

PROJECT SPECIFICATION

4.1 Introduction

This chapter focuses on the project specifications for the web application design. It begins with an analysis conducted through research on existing applications and in-depth discussions with the final year project supervisor to identify system requirements. Section 4.2 will discuss the functional requirements, while Section 4.3 covers the non-functional requirements. Section 4.4 will showcase the use case diagram and provide use case descriptions based on the functional requirements. Finally, Section 4.5 presents the initial prototype screen designs for the system's main features.

4.2 Functional Requirements

The functional and non-functional requirements are being collected through a thorough analysis with the supervisor of the final year project, a certified SCRUM Master, who provides insights into real-life industry SCRUM practices. The functional requirements have been identified for four main user roles: Project Manager, SCRUM Master, SCRUM Team Member, and Product Owner. The functional requirements are as below:

Table 4.1: Functional Requirements

ID	**Requirement**
FR01	The web application shall allow user to login and logout.
FR02	The web application shall allow user to register account.
FR03	The web application shall allow project manager to create multiple SCRUM project to manage.
FR04	The web application shall allow user to select a project to manage.
FR05	The web application shall allow project manager (creator of project) to have superadmin access throughout the projects (not role restricted).
FR06	The web application shall allow project manager to add or remove project members from the project.

FR07	The web application shall allow project manager to manage team member role.
FR08	The web application shall allow users to view all pages if invited.
FR09	The web application shall allow Product Owner to add or remove product backlog item into a project.
FR10	The web application shall allow Product Owner to edit product backlog item.
FR11	The web application shall allow SCRUM Master and SCRUM team member to add or remove sprint backlog item into product backlog item.
FR12	The web application shall allow SCRUM Master and SCRUM team member to edit sprint backlog item.
FR13	The web application shall auto generate Requirement Traceability Matrix based on the backlog items available.
FR14	The web application shall allow SCRUM Master to initiate and end sprint.
FR15	The web application shall auto generate SCRUM board based on the sprint backlog items available.
FR16	The web application shall allow SCRUM Master and SCRUM team member to customize the SCRUM board.
FR17	The web application shall allow SCRUM Master and SCRUM team member to drag and drop the sprint backlog items within the SCRUM board (changing status of sprint backlog items).
FR18	The web application shall allow SCRUM Master and SCRUM team member to estimate sprint backlog items.
FR19	The web application shall allow SCRUM Master or SCRUM team member to initiate a Planning Poker session for a sprint backlog item.
FR20	The web application shall allow project members to estimate sprint backlog items with Planning Poker.
FR21	The web application shall notify all project members when a Planning Poker session is initiated.

FR22	The web application shall auto generate burndown and burnup chart when a sprint is ended.
FR23	The web application shall estimate the project completion dateline based on available estimation effort after the first sprint ended.

4.3 Non-Functional Requirements

The non-functional requirement of web application is as follow:

Table 4.2: Non-Functional Requirements

ID	**Requirement**	**Category**
NFR01	The web application shall implement authentication and authorization mechanisms to ensure only authorized users can access specific functionalities and data.	Security
NFR02	The web application shall implement password policies to enhance user account security.	Security
NFR03	The web application shall use encryption protocol for passwords.	Security
NFR04	The web application shall be available 24/7 to provide the services to the users.	Availability
NFR05	The web application shall be compatible with the latest versions of popular browsers.	Availability
NFR06	The web application shall backup all data to prevent loss due to accidental deletion or system failures.	Integrity
NFR07	The web application shall have a user-friendly interface with clear navigation paths and well-organized contents.	Usability
NFR08	The web application shall validate the input of users to assist users in submitting accurate data or information.	Usability

4.4 Use Cases

4.4.1 Use Case Diagram

A use case diagram is used to describe the requirements and the functionality of the system. Derived from the functional requirements, use case diagram often shows the relationship between the requirements and the stakeholders. The use case diagram of the web application is as follow:

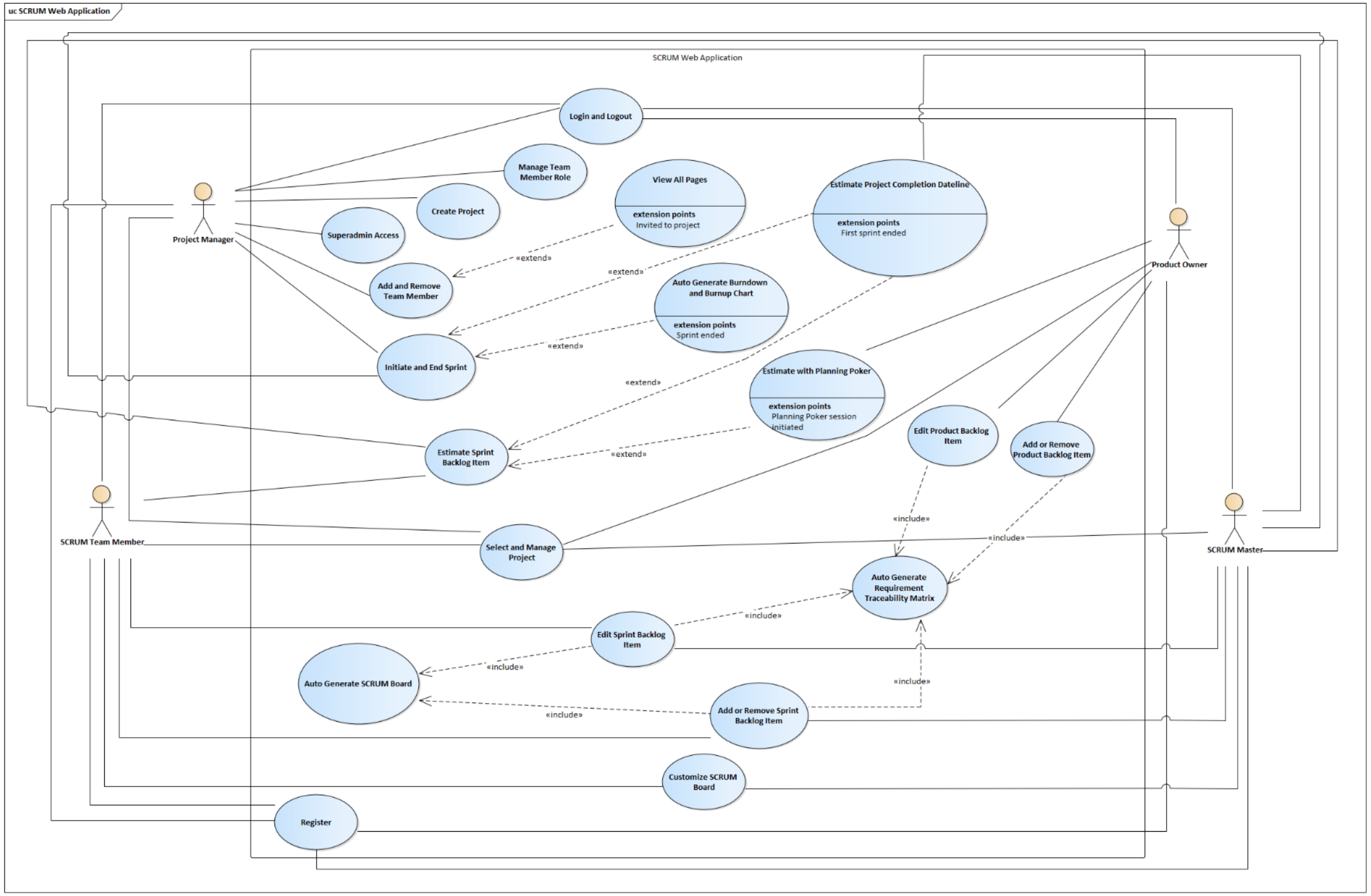


Figure 4.1: SCRUM Web Application Use Case Diagram

4.4.2 Use Case Description

In the use case description, users will be referring to Project Manager, Product Owner, SCRUM Master, SCRUM Team Member to reduce repetitive terms.

4.4.2.1 Login and Logout

Table 4.3: Use Case Description of Login and Logout

Use Case Name: Login and Logout	**ID:** UC01	**Importance Level:** High
Primary Actor: Project Manager, Product Owner, SCRUM Master, SCRUM Team Member	**Use Case Type:** Detail, Real	
Stakeholders and Interests: Users – To login and logout from the web application.		
Brief Description: This use case describes how users can login into the web application by entering the correct email and password and logout from the system by clicking the logout button.		
Trigger: Users selected the login option or the logout option in the web application.		
Relationships: Association : Project Manager, Product Owner, SCRUM Master, SCRUM Team Member Include : - Extend : - Generalization: -		
Precondition: **<u>Logout</u>** User has logged into the web application.		

Normal Flow of Events: **Login** 1. The web application prompts the user to enter email and password to login into the web application. 2. The user clicks on the ‘Login’ button. 3. If the email or password field is empty, V-1 will be executed. 4. If the combination of email and password entered are incorrect or could not found in the database, V-2 will be executed. 5. If the combination of email and password matches with the database, the web application redirects the user to the project list page. **Logout** 1. The user clicks on the “Logout” button. 2. The web application redirects the user to the login page.
Sub-flows: -
Alternate/Exceptional Flows: **V-1: Empty Email or Password** 1. The web application reminds the user that both the password field and email field are required to be filled. **V-2: Invalid Combination of Email and Password** 1. The web application reminds the user that the combination of email and password is invalid.

4.4.2.2 Register

Table 4.4: Use Case Description of Register

Use Case Name: Register	**ID:** UC02	**Importance Level:** High

Primary Actor: Project Manager, Product Owner, SCRUM Master, SCRUM Team Member	**Use Case Type:** Detail, Real
Stakeholders and Interests: User – To register an account to use the web application.	
Brief Description: This use case describes how users can register a valid account to login into the web application.	
Trigger: Users selected the register option in the login page.	
Relationships: Association : Project Manager, Product Owner, SCRUM Master, SCRUM Team Member Include : - Extend : - Generalization: -	
Precondition: -	
Normal Flow of Events: 1. The web application prompts the user to enter name, username, email, password, confirm password, address and phone to register under account. 2. The user clicks on the "Register" button. 3. If the username entered already exist in the system database, V-1 will be executed. 4. If the email entered is not a valid email, V-2 will be executed. 5. If the email entered already exist in the system database, V-3 will be executed.	

6. If the password entered does not match with the confirm password, V-4 will be executed.
7. If the password is not 8-20 characters with uppercase, lowercase, number and special character, V-5 will be executed.
8. If the phone number format entered is not a valid phone number, V-6 will be executed.
9. If the phone number entered already exist in the system database, V-7 will be executed.
10. If any of the field entered is empty, V-8 will be executed.
11. The web application notifies the user that the account has been created successfully.
12. The web application redirects the user to the login page.

Sub-flows: -

Alternate/Exceptional Flows:

V-1: Repeated Username

1. The web application reminds the user that the username entered already existed in the system database.

V-2: Invalid Email

1. The web application reminds the user that the email format entered is invalid.

V-3: Repeated Email

1. The web application reminds the user that the email entered already existed in the system database.

V-4: Confirm Password Mismatch

1. The web application reminds the user that the confirm password entered does not match with the password entered.

V-5: Invalid Password Format

1. The web application reminds the user that the password format entered is invalid. 2. The web application hints the user on the correct format of password. **V-6: Invalid Phone Number** 1. The web application reminds the user that the phone number format entered is invalid. **V-7: Repeated Phone Number** 1. The web application reminds the user that the phone number entered already existed in the system database. **V-8: Empty Field** 1. The web application reminds the user that all fields are required to be filled.

4.4.2.3 Create Project

Table 4.5: Use Case Description of Create Project

Use Case Name: Create Project	**ID:** UC03	**Importance Level:** High
Primary Actor: Project Manager		**Use Case Type:** Detail, Real
Stakeholders and Interests: Project Manager – create new SCRUM project to manage		
Brief Description: This use case describes how project manager can create a new SCRUM project to manage and invite team members by entering their email.		
Trigger: Project manager selects the create new project option in the web application.		

Relationships:

Association : Project Manager
Include : -
Extend : -
Generalization: -

Precondition:

The project manager has logged into the web application.

Normal Flow of Events:

1. The web application prompts the project manager to enter the title of project to be created.
2. The web application prompts the project manager to enter the email of the project member to be added into the project (Optional).
3. The web application will show a list of users to be invited into the project with all being assigned with the role “Team Member”.
4. The web application will allow the project manager to switch the project member role between “Team Member”, “SCRUM Master” and “Product Owner”,
5. The project manager clicks on the “Confirm” button.
6. If the project title is empty, V-1 will be executed.
7. If the email entered is not a valid email, V-2 will be executed.
8. If the project member invited is not registered in the system, V-3 will be executed.
9. If the project owner invite himself as the team member, V-4 will be executed.
10. If the email entered is blank, V-5 will be executed.
11. The web application notifies the user that the project has been created successfully.

Sub-flows: -

Alternate/Exceptional Flows: -

V-1: Empty Project Title

1. The web application reminds the project manager that the project title is required to be filled. **V-2: Invalid Email** 1. The web application reminds the project manager that the email entered is not a valid format. **V-3: Member Not Registered** 1. The web application reminds the project manager that the member invited is not a registered member in the system. **V-4: Cannot Invite Yourself** 1. The web application reminds the project manager that he cannot invite himself to the project. **V-5: Empty Email** 1. The web application reminds the project manager that the email field is required to be filled.

4.4.2.4 Select and Manage Project

Table 4.6: Use Case Description of Select and Manage Project

Use Case Name: Select and Manage Project	**ID:** UC04	**Importance Level:** High
Primary Actor: Project Manager, Product Owner, SCRUM Master, SCRUM Team Member	**Use Case Type:** Detail, Real	
Stakeholders and Interests: User – select a SCRUM project to be manage.		
Brief Description:		

This use case describes how user can select a SCRUM project to manage.
Trigger: User has logged into the web application.
Relationships: Association : Project Manager, Product Owner, SCRUM Master, SCRUM Team Member Include : - Extend : - Generalization: -
Precondition: The user has logged into the web application.
Normal Flow of Events: 1. The web application shows a list of projects that the user is involved in. 2. The web application prompts the user to select a project to manage. 3. The web application redirects the user to the backlog page.
Sub-flows: -
Alternate/Exceptional Flows: -

4.4.2.5 Add or Remove Project Member

Table 4.7: Use Case Description of Add or Remove Project Member

Use Case Name: Add or Remove Project Member	**ID:** UC05	**Importance Level:** High
Primary Actor: Project Manager	**Use Case Type:** Detail, Real	

Stakeholders and Interests:

Project Manager – Add or remove project member from the chosen project.

Brief Description:

This use case describes how project manager can add or remove project member from the chosen project by entering their email or choosing the remove option.

Trigger:

Project manager selects the add project member or remove project member option in the web application.

Relationships:

Association : Project Manager
Include : -
Extend : View All Pages
Generalization: -

Precondition:

The project manager has logged into the web application and chosen a project to manage.

Normal Flow of Events:

<u>Add Project Member</u>

1. The web application prompts the project manager to enter the email of the project member to be added into the project.
2. If the email entered is not a valid email, V-1 will be executed.
3. If the project member invited is not registered in the system, V-2 will be executed.
4. If the project owner invite himself as the team member, V-3 will be executed.
5. If the project member invited is already in the project, V-4 will be executed.

6. If the email entered is blank, V-5 will be executed.
7. The web application will show a list of users to be invited into the project with all being assigned with the role “Team Member”.
8. The web application will allow the project manager to switch the project member role between “Team Member”, “SCRUM Master” and “Product Owner”,
9. The project manager clicks on the “Confirm” button.
10. The web application will add all the users into the project and notifies the project manager.
11. Extend use case E-1 will be executed.

Remove Project Member

12. The web application prompts the project manager to confirm the removing of the project member.
13. The web application removes the project member from the project.

Sub-flows:

E-1: View All Pages

1. The web application will allow the invited project member to view every page of the selected project.

Alternate/Exceptional Flows:

V-1: Invalid Email

1. The web application reminds the project manager that the email entered is not a valid format.

V-2: Member Not Registered

1. The web application reminds the project manager that the member invited is not a registered member in the system.

V-3: Cannot Invite Yourself

1. The web application reminds the project manager that he cannot invite himself to the project.

V-4: Project Member Already Invited 1. The web application will remind the project manager that the member invited is already invited to the project. **V-5: Empty Email** The web application reminds the project manager that the email field is required to be filled.

4.4.2.6 Manage Project Member Role

Table 4.8: Use Case Description of Manage Project Member Role

Use Case Name: Manage Project Member Role	**ID:** UC06	**Importance Level:** High
Primary Actor: Project Manager		**Use Case Type:** Detail, Real
Stakeholders and Interests: Project manager – change project member role.		
Brief Description: This use case describes how project manager can edit the project member role to one or more of the three roles: SCRUM Master, Team Member and Product Owner.		
Trigger: Project Manager selects the edit project member role option in the web application.		
Relationships: Association : Project Manager Include : - Extend : - Generalization: -		

Precondition: The project manager has log into the web application and chosen a project to manage.
Normal Flow of Events: 1. The web application prompts the project manager to choose the project member to be managed for their role. 2. The web application will show 3 options for the project manager to select (can select more than 1): - Team Member - SCRUM Master - Product Owner 3. The web application prompts the project manager to confirm. 4. If the project manager did not choose any of the 3 roles for a project member, V-1 will be executed. 5. The web application updates the role of project member accordingly.
Sub-flows: -
Alternate/Exceptional Flows: - **V-1: Role Required for Project Member** 1. The web application reminds the project manager that a minimum of 1 role is required to be assigned to a project member.

4.4.2.7 Add or Remove Product Backlog Item

Table 4.9: Use Case Description of Add or Remove Product Backlog Item

Use Case Name: Add or Remove Product Backlog Item	**ID:** UC07	**Importance Level:** High
Primary Actor: Product Owner	**Use Case Type:** Detail, Real	

Stakeholders and Interests: Product Owner – Add or remove product backlog item
Brief Description: This use case describes how a Product Owner can add or remove product backlog item of a project.
Trigger: Product Owner choose the add or remove product backlog item option in the web application.
Relationships: Association : Product Owner Include : Auto Generate Requirement Traceability Matrix Extend : - Generalization: -
Precondition: The Product Owner log into the web application and chosen a project that he is invited to.
Normal Flow of Events: **<u>Add Product Backlog Item</u>** 1. The web application prompts the Product Owner for the product backlog item description and priority to be added. 2. The Product Owner clicks on the “Confirm” button. 3. If the product backlog item description is empty, V-1 will be executed. 4. The web application adds the product backlog item into the project with the status “To Do”. 5. Include use case U-1 is executed. **<u>Remove Product Backlog Item</u>**

1. The web application prompts the Product Owner to confirm the removing of product backlog item. 2. The web application removes product backlog item from the project. 3. Include use case U-1 is executed.
Sub-flows: **U-1: Auto Generate Requirement Traceability Matrix** 1. The web application auto generates or update the requirement traceability matrix based on modification of Product Owner.
Alternate/Exceptional Flows: **V-1: Empty Product Backlog Description** 1. The web application reminds the Product Owner that the product backlog description is required to be filled.

4.4.2.8 Edit Product Backlog Item

Table 4.10: Use Case Description of Edit Product Backlog Item

Use Case Name: Edit Product Backlog Item	**ID:** UC08	**Importance Level:** High
Primary Actor: Product Owner	**Use Case Type:** Detail, Real	
Stakeholders and Interests: Product Owner – edit existing product backlog item		
Brief Description: This use case describes how Product Owner can edit the existing product backlog item in a project.		
Trigger: Product Owner selects the edit product backlog item option in the web application.		

Relationships:

Association : Product Owner
Include : Auto Generate Requirement Traceability Matrix
Extend : -
Generalization: -

Precondition:

The Product Owner log into the web application, chose a project that he is invited to and there is existing product backlog item.

Normal Flow of Events:

1. The web application prompts the Product Owner to select a product backlog item to be edited.
2. The web application prompts the Product Owner to input the new value of the product backlog item, which are the description, status and the priority.
3. If the product backlog item description is empty, V-1 will be executed.
4. The Product Owner clicks on the “Confirm” button.
5. The web application will update the product backlog item accordingly.
6. Include use case U-1 is executed.

Sub-flows:

U-1: Auto Generate Requirement Traceability Matrix

1. The web application updates the requirement traceability matrix based on the modification of the Product Owner.

Alternate/Exceptional Flows:

V-1: Empty Product Backlog Description

1. The web application reminds the Product Owner that the product backlog description is required to be filled.

4.4.2.9 Add or Remove Sprint Backlog Item

Table 4.11: Use Case Description of Add or Remove Sprint Backlog Item

Use Case Name: Add or Remove Sprint Backlog Item	**ID:** UC09	**Importance Level:** High
Primary Actor: SCRUM Master, SCRUM Team Member	**Use Case Type:** Detail, Real	
Stakeholders and Interests: SCRUM Master and SCRUM Team Member – Add or remove sprint backlog item in a product backlog item		
Brief Description: This use case describes how a SCRUM Master and SCRUM Team Member can add or remove sprint backlog item in a product backlog item of a project.		
Trigger: SCRUM Master or SCRUM Team Member choose the add sprint backlog item option in the web application.		
Relationships: Association : SCRUM Master, SCRUM Team Member Include : Auto Generate Requirement Traceability Matrix Extend : - Generalization: -		
Precondition: The SCRUM Master or SCRUM Team Member log into the web application, chose a project that he is invited to, and there is existing product backlog item.		

Normal Flow of Events: 1. The web application prompts the SCRUM Master or SCRUM Team Member for the sprint backlog item description, priority, assigned to, and category to be added. 2. The SCRUM Master or SCRUM Team Member clicks on the “Confirm” button. 3. If any of the field entered is empty, V-1 will be executed. 4. The web application adds the sprint backlog item into the product backlog item, set the status as “To Do” and estimation effort as “-”. 5. Include use case U-1 is executed.
Sub-flows: **U-1: Auto Generate Requirement Traceability Matrix** 1. The Web application auto update the requirement traceability matrix based on the input of the SCRUM Master or SCRUM Team Member
Alternate/Exceptional Flows: **V-1: Empty Field** 1. The web application reminds the user that all fields are required to be filled.

4.4.2.10 Edit Sprint Backlog Item

Table 4.12: Use Case Description of Edit Sprint Backlog Item

Use Case Name: Edit Sprint Backlog Item	**ID:** UC10	**Importance Level:** High
Primary Actor: SCRUM Master, SCRUM Team Member	**Use Case Type:** Detail, Real	
Stakeholders and Interests: SCRUM Master and SCRUM Team Member – Edit sprint backlog item in a product backlog item		

Brief Description:

This use case describes how a SCRUM Master and SCRUM Team Member can edit an existing sprint backlog item in a product backlog item of a project.

Trigger:

SCRUM Master or SCRUM Team Member choose the edit sprint backlog item option in the web application.

Relationships:

Association : SCRUM Master, SCRUM Team Member
Include : Auto Generate Requirement Traceability Matrix
Extend : -
Generalization: -

Precondition:

The SCRUM Master or SCRUM Team Member log into the web application, chose a project that he is invited to, and there is existing product backlog item with sprint backlog item.

Normal Flow of Events:

1. The web application prompts the SCRUM Master or SCRUM Team Member for the sprint backlog item to be edited.
2. The web application prompts the SCRUM Master or SCRUM Team Member to input the new value of the sprint backlog item, which are the description, category, status, priority and assigned to.
3. The SCRUM Master or SCRUM Team Member clicks on the "Confirm" button.
4. The web application updates the sprint backlog item in the product backlog item.
5. Include use case U-1 is executed.

Sub-flows:

U-1: Auto Generate Requirement Traceability Matrix

1. The web application updates the requirement traceability matrix based on modification of the SCRUM Master or SCRUM Team Member.
Alternate/Exceptional Flows:

4.4.2.11 Initiate and End Sprint

Table 4.13: Use Case Description of Initiate and End Sprint

Use Case Name: Initiate and End Sprint	**ID:** UC11	**Importance Level:** High
Primary Actor: SCRUM Master	**Use Case Type:** Detail, Real	
Stakeholders and Interests: SCRUM Master – initiate or end sprint by choosing backlog items.		
Brief Description: This use case describes how SCRUM Master can initiate a sprint in a project by selecting the backlog items to be done in the sprint and end a sprint if desired to.		
Trigger: SCRUM Master selects the start sprint or end sprint option in the web application.		
Relationships: Association : SCRUM Master Include : Auto Generate SCRUM Board, Auto Generate Burndown Chart and Burnup Chart Extend : Estimate Project Completion Dateline Generalization: -		

Precondition:

<u>Start Sprint</u>

The SCRUM Master log into the web application and chosen a project that he is invited to, and there is existing product backlog items and sprint backlog items.

<u>End Sprint</u>

The SCRUM Master log into the web application and chosen a project that he is invited to, and there is existing sprint.

Normal Flow of Events:

<u>Initiate Sprint</u>

1. The web application prompts the SCRUM Master to enter a date range for the sprint.
2. If the date range is invalid, V-1 will be executed.
3. The web application prompts the SCRUM Master to choose the sprint backlog items to be added into the sprint. A mix of sprint backlog items from different product backlogs item are allowed.
4. If the sprint initiated is not the first, an estimated completion date will be calculated.
5. Include use case U-1 is executed.

<u>End Sprint</u>

1. The web application prompts the SCRUM Master for the actual effort used to complete the sprint.
2. The SCRUM Master clicks on the "Confirm" button.
3. The web application ends the sprint.
4. Include use case U-1 is executed.
5. Include use case U-2 is executed.
6. Extend use case E-1 will be executed.

Sub-flows:

U-1: Auto Generate Requirement Traceability Matrix

1. The web application updates the requirement traceability matrix based on sprint information. **U-2: Auto Generate Burndown Chart and Burnup Chart** 1. The web application auto generates a burndown chart and burnup chart based on the estimation effort of the sprint backlog items. **E-1: Estimate Project Completion Dateline** 1. The web application will update the estimated project completion dateline based on the available estimation efforts and accumulative user pace in completing the sprint.
Alternate/Exceptional Flows: **V-1: Invalid Date Range** 1. The web application reminds the SCRUM Master that the date range entered is invalid. 2. The web application prompts the SCRUM Master to enter the date range again.

4.4.2.12 Customize SCRUM Board

Table 4.14: Use Case Description of Customize SCRUM Board

Use Case Name: Customize SCRUM Board	**ID:** UC11	**Importance Level:** High
Primary Actor: SCRUM Master, SCRUM Team Member		**Use Case Type:** Detail, Real
Stakeholders and Interests: SCRUM Master and SCRUM Team Member – Customize the generated SCRUM Board		
Brief Description:		

This use case describes how users can customize the SCRUM board in the web application.

Trigger:

Users selected customize SCRUM Board option.

Relationships:

Association : SCRUM Master, SCRUM Team Member
Include : -
Extend : -
Generalization: -

Precondition:

The SCRUM Master and SCRUM Team Member log into the web application, chose a project that he is invited to, and there is existing sprint backlog item.

Normal Flow of Events:

1. The web application prompts the SCRUM Master or SCRUM Team Member to enter the description of the new column for the SCRUM board.
2. The web application prompts the SCRUM Master or SCRUM Team Member to confirm.
3. If the description entered is empty, V-1 will be executed.
4. The web application will generate a new column with the description entered.
5. The web application allows SCRUM Master and SCRUM Team Member to drag and drop the sprint backlog item to different column.
6. The status of the sprint backlog item will be reflected on the other pages of the web application according to the column changes.

Sub-flows: -

Alternate/Exceptional Flows: -

V-1: Empty Description

1. The web application reminds the SCRUM Master or SCRUM Team Member that the description is required to be filled.

4.4.2.13 Estimate Sprint Backlog Item

Table 4.15: Use Case Description of Estimate Sprint Backlog Item

Use Case Name: Estimate Sprint Backlog Item	**ID:** UC13	**Importance Level:** High
Primary Actor: SCRUM Master, SCRUM Team Member	**Use Case Type:** Detail, Real	
Stakeholders and Interests: SCRUM Master, SCRUM Team Member – To estimate the sprint backlog item.		
Brief Description: This use case describes how SCRUM Master and SCRUM Team Member can estimate the sprint backlog item.		
Trigger: SCRUM Master and SCRUM Team Member choose the estimation option in sprint backlog item.		
Relationships: Association : SCRUM Master, SCRUM Team Member Include : Extend : Estimate Project Completion Dateline, Estimate with Planning Poker Generalization: -		
Precondition:		

The SCRUM Master and SCRUM Team Member log into the web application, chose a project that he is invited to, and there is existing sprint backlog item.

Normal Flow of Events:

1. The web application prompt the SCRUM Master or Team Member to choose the sprint backlog item to be estimated.
2. The web application prompts the SCRUM Master or Team Member for the estimation effort and unit for the sprint backlog item.
3. If the SCRUM Master or SCRUM Team Member wishes to estimate with planning poker, extend use case E-1 will be executed.
4. The SCRUM Master or SCRUM Team Member clicks on the "Confirm" button.
5. If the estimation effort is invalid, V-1 will be executed.
6. If any of the field entered is empty, V-1 will be executed.
7. The estimation effort will be updated accordingly for the sprint backlog item.
8. Extend use case E-2 will be executed.

Sub-flows:

E-1: Estimate with Planning Poker

1. SCRUM Master or SCRUM Team Member clicks on the "Planning Poker Estimation" to initiate a Planning Poker session for the sprint backlog item.
2. All of the project members will be notified about the created session.
3. The web application prompts the initiator to select a sequence number to be used in the Planning Poker game.
4. The web application allows all the project members to interact with each other with a sticky note on the page.
5. The web application prompts all the project members to choose a number based on the sequence chosen as the estimation effort for the sprint backlog item.
6. If all the project members chose the same estimation effort, the web application goes back to Normal Flow Step 7.

7. If any of the estimation effort chosen is different from others, the web application notifies everyone regarding the result and goes back to Step 5.
8. If SCRUM Master or SCRUM Team Member wishes to revote even after a success attempt, the web application goes back to Step 5.

E-2: Estimate Project Completion Dateline

2. The web application will update the estimated project completion dateline based on the available estimation efforts and accumulative user pace in completing the sprint.

Alternate/Exceptional Flows:

V-1: Invalid Estimation Effort

1. The web application reminds the SCRUM Master or SCRUM Team Member that the estimation effort entered is invalid.

V-2: Empty Field

1. The web application reminds the SCRUM Master or SCRUM Team Member all the fields are required to be filled.

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

This chapter will provide an in-depth discussion of the system design for the SCRUM project management web application. Section 5.2 will cover the system architecture and design, focusing on the flow of the backend operations and the choices made regarding architecture and design. Section 5.3 will delve into the database design, including the data dictionary along with an ERD diagram. Finally, Section 5.4 will showcase the prototype of the web application designed using Figma. For the whole final year project, camel case naming will be used as the standard for naming the variables.

5.2 System Architecture and Design

5.2.1 System Architecture

The chosen system architecture for this final year project is the Model-View-Controller (MVC) architecture, which aligns with the main characteristics of the development tools used, Laravel. The MVC architecture comprises three main components: Model, View, and Controller.

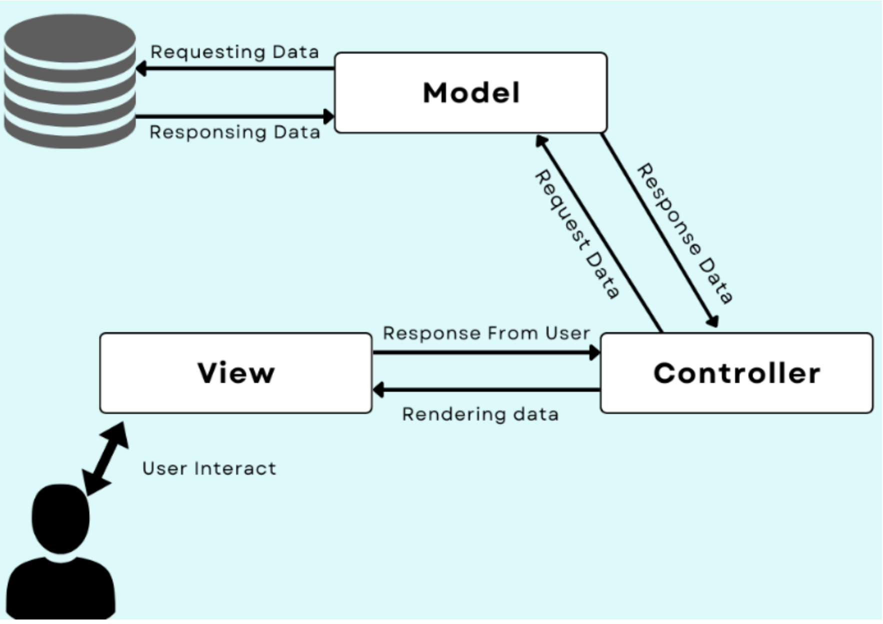


Figure 5.1: Overview of MVC Architecture (jaydeeosathwara272, 2022).

In MVC architecture, the Model serves as the foundational layer. It is responsible for fetching data from the database and returning the requested query to the controller. Additionally, it defines all input for the database tables and helps establish relationships between them.

The View represents the user interface of the web application. It manages communication between the controller and user input or requests.

The Controller acts as the middleman in the MVC architecture. It handles requests from the view, processes required logic, and interacts with the database through the model. Subsequently, the output is provided in reverse order: the model returns data to the controller, which then supplies the views.

One of the greatest benefits of MVC that led to its selection is its ability to provide clean code separation. Since the final year project web application will be a large-scale program, MVC helps organize all the code in a logical manner of separation. This organization enables quicker searching of code in a large-scale program, thus enhancing maintainability. Furthermore, MVC allows the program to be easily modifiable. Adding new features or updating existing ones does not affect the user-interface, and the logic between each component does not overlap. This increases flexibility in coding and allows future developers to understand the code more easily and contribute effectively (verma_anushka, 2023).

5.2.2 Design Pattern

In the final year project, 2 different design patterns have been chosen: Component Design Pattern and Service-Repository Pattern.

5.2.2.1 Component Design Pattern

Component design pattern is one of the characteristics of object-oriented programming. It allows developers to organize their code into reusable components, enabling them to reuse code without repetitive typing. This results in minimum coding and much cleaner code (Java Design Pattern, n.d.).

Imagine wanting to save costs: in a scenario where you need to cook 100 dishes, instead of buying 100 different stoves, you can use just one stove to cook all 100 dishes. Similarly, when designing a front-end web application, reducing the amount of code is essential. Many components can be reused, such as the navigation bar and search bar.

The component design pattern is similar to the composite pattern, differing only in the hierarchical structure. However, it inherits the benefits of the Open/Close Principle of the composite pattern. When introducing new components into a web application, the existing components remain unaffected, ensuring that the original code continues to function without breaking (Refactoring Guru, n.d.).

5.2.2.2 Service-Repository Pattern

The Service-Repository Pattern is typically used as an abstraction layer over the MVC architecture. It extends the characteristics of MVC architecture by introducing two new layers: Service and Repository.

The service layer is responsible for filtering data before passing it into the repository pattern. It also helps to combine different types of requests or logics into one (Ankitpal, 2022).

The repository layer centralizes the handling of all data requests. Here, it receives requests from the service layer, which have already been filtered. Hence, this layer is solely responsible for interacting with the database. The repository layer also introduces interfaces, which provide benefits such as code reusability.

The Service-Repository design pattern adheres to the SOLID principles, separating concerns between business logic and database logic, ensuring that each layer has only one responsibility. It centralizes all methods, ensuring better maintainability (Bergman, 2017).

5.2.3 Combination of Architecture and Design Pattern

By combining the MVC architecture with the component design pattern and service-repository pattern, the web application inherits all of their advantages.

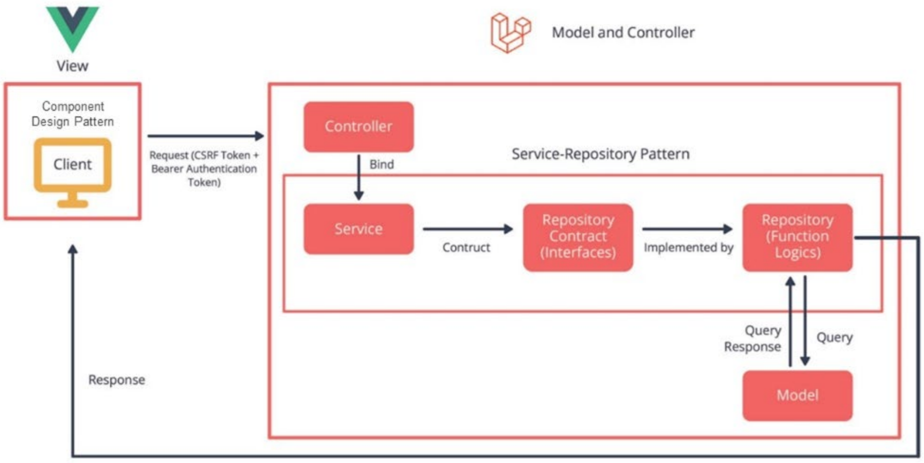

View
Component Design Pattern
Client
Request (CSRF Token + Bearer Authentication Token)
Model and Controller
Controller
Bind
Service-Repository Pattern
Service
Contruct
Repository Contract (Interfaces)
Implemented by
Repository (Function Logics)
Query Response
Query
Model
Response

Figure 5.2: System Architecture and Design of FYP Web Application

The frontend, representing the View, is developed with Vue.js, implementing the component design pattern. Instead of writing repetitive code, Vue.js are utilized with the reuse of components, reducing code duplication. Each page is similar to building a puzzle diagram, with each puzzle representing a component.

Additionally, for communication between the frontend and backend, bearer authentication tokens and CSRF tokens are required to ensure secure communication. By implementing the service-repository pattern in the model and controller, the web application can separate all logic and filtering, ensuring each component has its own responsibility. The controller ensures that only authorized users can access the API and redirects the requests to services. Services filter all data requests and send them to repositories. Repositories implement repository contracts, which are interfaces, ensuring that only data logic is executed. The repository then interacts with the model for CRUD queries.

While MVC presents a clean code concept, the implementation of service-repository pattern further enhances the organization of code.

5.3 Database Design

5.3.1 Entity Relationship Diagram (ERD)

The ERD illustrates the relationships between each of the tables in the web application. There are a total of 11 tables, all interconnected through primary keys and foreign keys. With Laravel's support for relationships, utilizing these primary and foreign keys becomes straightforward.

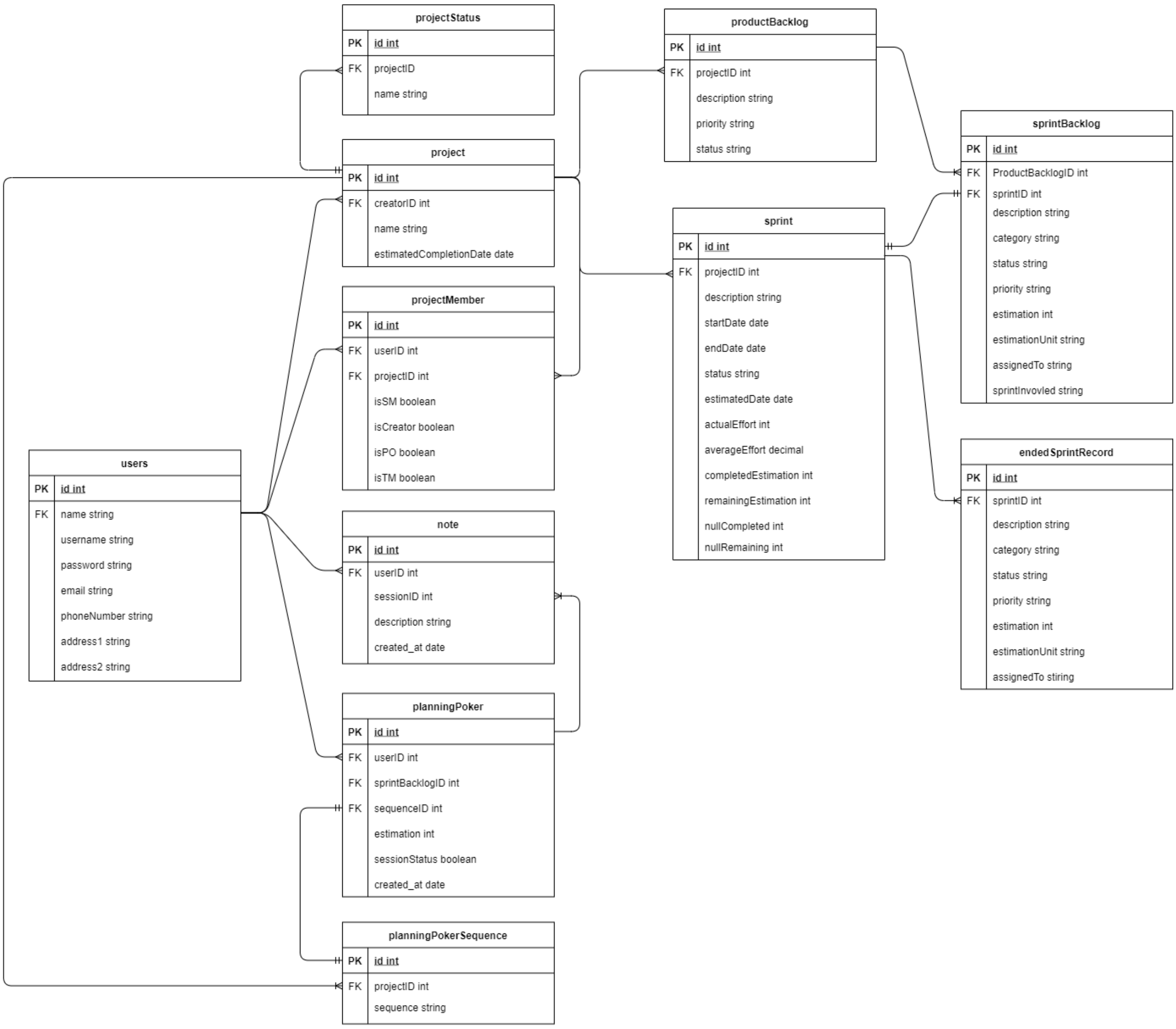


Figure 5.3: Entity Relationship Diagram

5.3.2 Database Dictionary

5.3.2.1 User Table

The users table is used store all the information of the registered user in which includes their account and password.

Table 5.1: Users Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the user	PK	int	Auto Increment
name	The name of the user	-	string	-
username	The username of the user	-	string	-
email	The email of user	-		
password	The password of the user	-	string	-
phoneNumber	The phone number of the user	-	string	-
address1	Address line 1 of the user	-	string	-
Address2	Address line 2 of the user incase line 1 is not sufficient.	-	String	null

5.3.2.2 Project Table

The project table is used to store the information of a project created by the users.

Table 5.2: Project Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the project	PK	int	Auto Increment
creatorID	The creator ID of the project reference from Users table.	FK	int	-
name	The name of the project created	-	string	-

estimated Completion Date	The overall estimated completion date of the project.	-	date	null

5.3.2.3 Project Status Table

The projectStatus table is used to store the customized status of the SCRUM Board identified by SCRUM Master or SCRUM Team Member.

Table 5.3: ProjectStatus Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the project status	PK	int	Auto Increment
projectID	The project ID referenced from Project table	FK	int	-
name	The name of the customized project status	-	string	-

5.3.2.4 Project Member Table

The projectMember table is used to store all the invited project member information for a project.

Table 5.4: ProjectMember Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the project members	PK	int	Auto Increment
userID	The user ID of the project reference from Users table	FK	int	-
projectID	The project ID that the user is involved in reference from the Project table	FK	int	-
isSM	Whether if the invited project member is a SCRUM Master	-	boolean	false

isPO	Whether if the invited project member is a Product Owner	-	boolean	false
isTM	Whether if the invited project member is a Team Member	-	boolean	false
isCreator	Whether if the invited project member is a Creator	-	boolean	false

5.3.2.5 Product Backlog Table

The ProductBacklog table is used to store all the product backlog items in a project.

Table 5.5: ProductBacklog Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the product backlog	PK	int	Auto Increment
projectID	The project ID of the product backlog reference from the Project table.	FK	int	-
description	The description of the product backlog	-	string	-
priority	The priority of the product backlog	-	string	-
status	The status of the product backlog	-	string	To Do

5.3.2.6 Sprint Backlog Table

The sprintBacklog table is used to store all the sprint backlog item of a product backlog in a project.

Table 5.6: SprintBacklog Table

Data Name	Data Description	FK/PK	Data Type	Default Value

id	The unique ID of the sprint backlog	PK	int	Auto Increment
productBacklogID	The product backlog ID that the current sprint backlog belongs to reference from the productBacklog table	FK	int	-
sprintID	The sprint ID that the current sprint backlog is involved in reference from the Sprint table	FK	int	null
description	The description of the sprint backlog	-	string	-
category	The category of the sprint backlog	-	string	-
status	The status of the sprint backlog	-	string	To Do
priority	The priority of the sprint backlog	-	string	-
estimation	The estimation effort of the sprint backlog	-	int	null
estimationUnit	The estimation unit of the sprint backlog	-	string	day(s)
assignedTo	The project member who are responsible for this sprint backlog	-	string	null
sprintInvovled	All the sprint history that the sprint backlog has involved in	-	string	null

5.3.2.7 Ended Sprint Record Table

The endedSprintRecord table stores all the sprint backlog item information at the time of the sprint ended. The active sprint of a project is directly reference from the live sprint backlog item, hence requiring another table to store for ended sprints.

Table 5.7: EndedSprintRecord Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the sprint backlog	PK	int	Auto Increment
sprintID	The sprint ID that the current sprint backlog is involved in reference from the Sprint table	FK	int	null
description	The description of the sprint backlog	-	string	-
category	The category of the sprint backlog	-	string	-
status	The status of the sprint backlog	-	string	To Do
priority	The priority of the sprint backlog	-	string	-
estimation	The estimation effort of the sprint backlog	-	int	null
estimationUnit	The estimation unit of the sprint backlog	-	string	day(s)
assignedTo	The project member who are responsible for this sprint backlog	-	string	null

5.3.2.8 Sprint Table

The sprint table is used to store all the sprint information in which it includes active sprints and ended sprints.

Table 5.8: Sprint Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the sprint backlog	PK	int	Auto Increment
projectID	The project ID of the product backlog reference from the Project table.involved in reference from the Sprint table	FK	int	-
description	The description of the sprint	-	string	-
startDate	The start date of the sprint	-	date	-
endDate	The end date of the sprint	-	date	-
status	The status of the sprint	-	string	To Do
estimatedDate	The estimation finish date of the sprint	-	date	null
actualEffort	The actual effort of the sprint	-	int	null
averageEffort	The average effort used to finish the sprint	-	decimal	null
completed Estimation	The completed estimation effort when the sprint end	-	int	null
remaining Estimation	The remaining estimation effort when the sprint end	-	int	null
nullCompleted	The sprint backlog that has been completed with null estimation effort.	-	int	null
nullRemaining	The sprint backlog that is incomplete with null estimation effort.	-	int	null

5.3.2.9 Planning Poker Table

The planningPoker table is used to store all the planning poker session information in a project.

Table 5.9: PlanningPoker Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the planning poker	PK	int	Auto Increment
userID	The user ID that is involved in the planning poker reference from the User table.	FK	int	-
sprint Backlog ID	The sprint backlog ID that is currently involved in the planning poker estimation reference from sprintBacklog table.	FK	int	-
sequenceID	The sequence number of the estimation cards that is used in the planning poker session reference from the planningPokerSequence table	FK	int	null
estimation	The estimation effort made by users through planning poker	-	int	null
sessionStatus	The activeness of the planning poker session	-	boolean	false
created_at	The planning poker session initiated time	-	date	-

5.3.2.10 Planning Poker Sequence Table

The planningPokerSequence table is used to store the planning poker sequence number used in the estimation cards for the planning poker sessions.

Table 5.10: PlanningPokerSequence Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the planning poker sequence number	PK	int	Auto Increment
proejctID	The projectID that the sequence number belongs to reference from the Project table	FK	int	-
sequence	The sequence number used in the estimation cards.	-	string	-

5.3.2.11 Note Table

The note table is used to store all the sticky notes that has been written by the project members in a planning poker session.

Table 5.11: Note Table

Data Name	Data Description	FK/PK	Data Type	Default Value
id	The unique ID of the note	PK	int	Auto Increment
userID	The ID of the writer of the note reference from the User table	FK	int	-
description	The description of the note written.	-	string	-
created_at	The created time of the note	-	date	-

5.4 Prototype Design

The prototype, created using Figma, is visual representation of the final year project's user interface. It provides an overview of the idea of the web application and serves as a reference for developing the actual code of the web application.

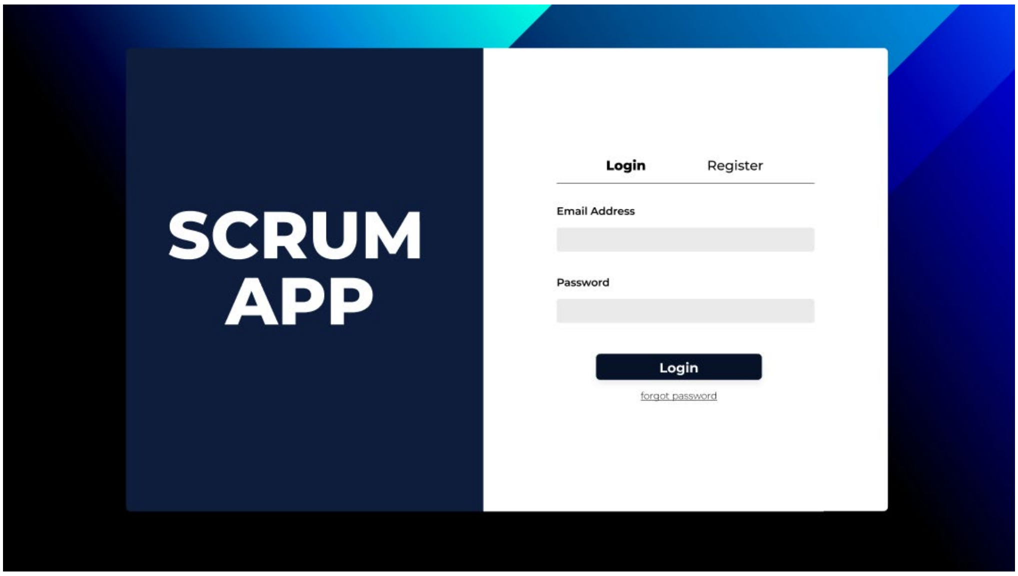


Figure 5.4: Login Page

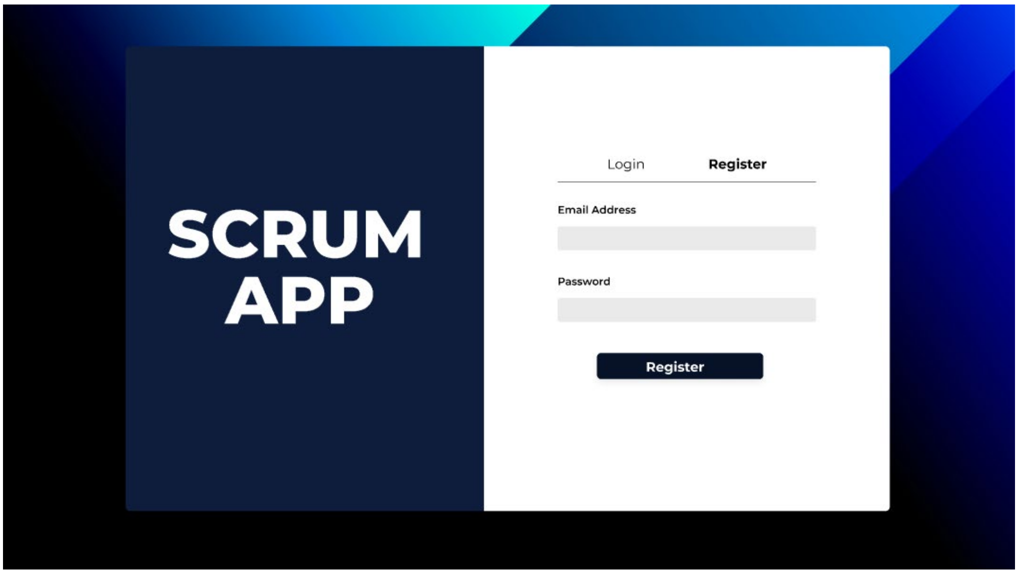
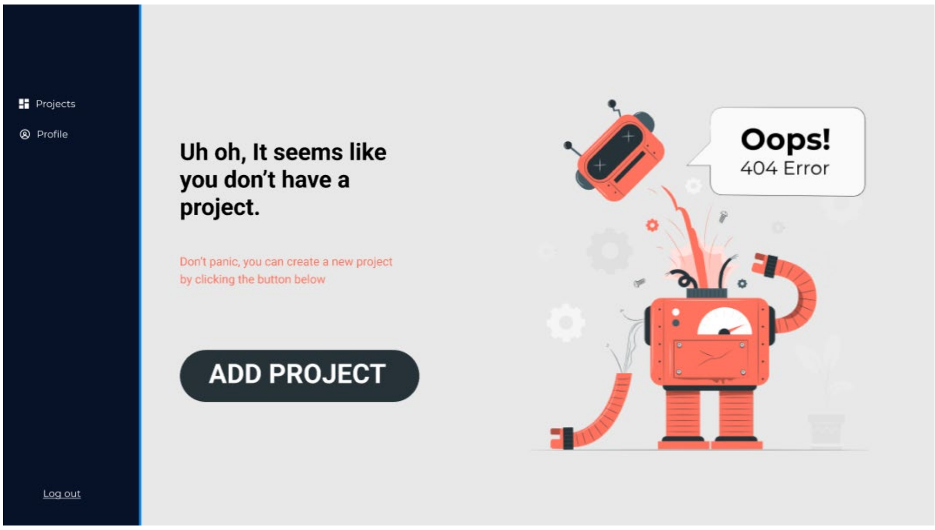


Figure 5.5: Register Page



Uh oh, It seems like you don't have a project.

Don't panic, you can create a new project by clicking the button below

Figure 5.6: Project Page (No Project)

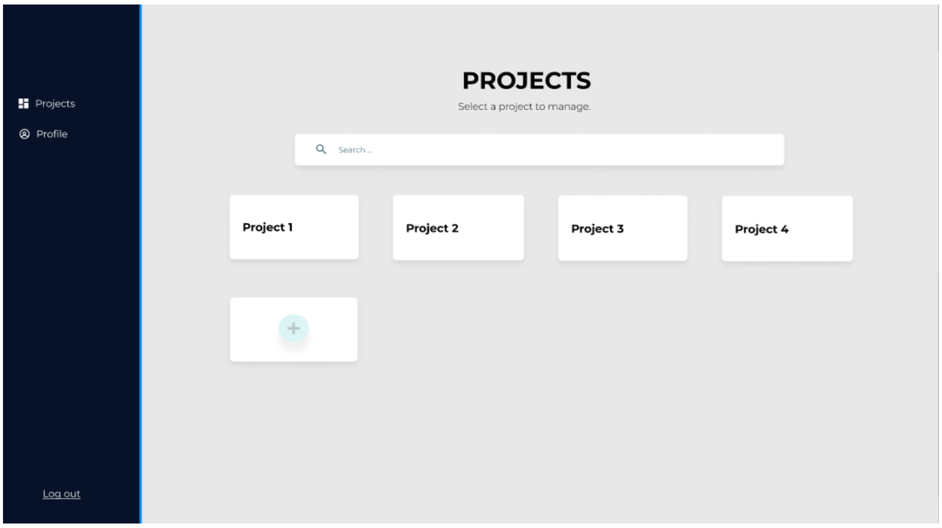
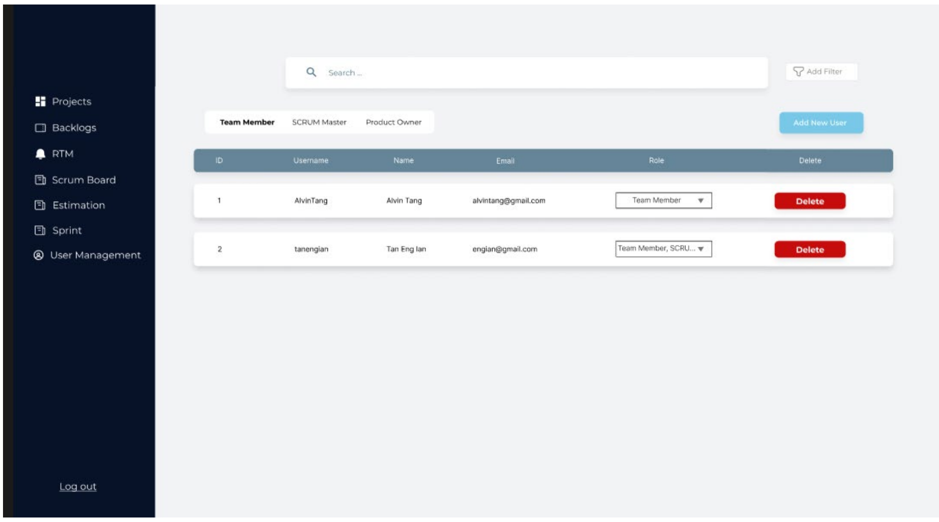
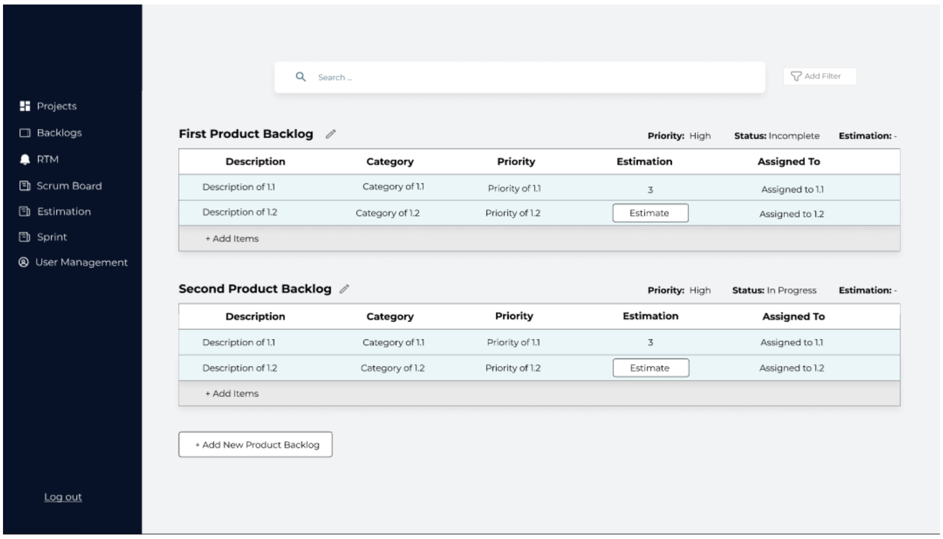


Figure 5.7: Project Page



ID	Username	Name	Email	Role	Delete
1	AlvinTang	Alvin Tang	alvintang@gmail.com	Team Member ▼	Delete
2	tanengian	Tan Eng Ian	engian@gmail.com	Team Member, SCRU... ▼	Delete

Figure 5.8: User Management Page



First Product Backlog

Priority: High **Status:** Incomplete **Estimation:** -

Description	Category	Priority	Estimation	Assigned To
Description of 1.1	Category of 1.1	Priority of 1.1	3	Assigned to 1.1
Description of 1.2	Category of 1.2	Priority of 1.2	Estimate	Assigned to 1.2
+ Add Items				

Second Product Backlog

Priority: High **Status:** In Progress **Estimation:** -

Description	Category	Priority	Estimation	Assigned To
Description of 1.1	Category of 1.1	Priority of 1.1	3	Assigned to 1.1
Description of 1.2	Category of 1.2	Priority of 1.2	Estimate	Assigned to 1.2
+ Add Items				

Figure 5.9: Backlog Page

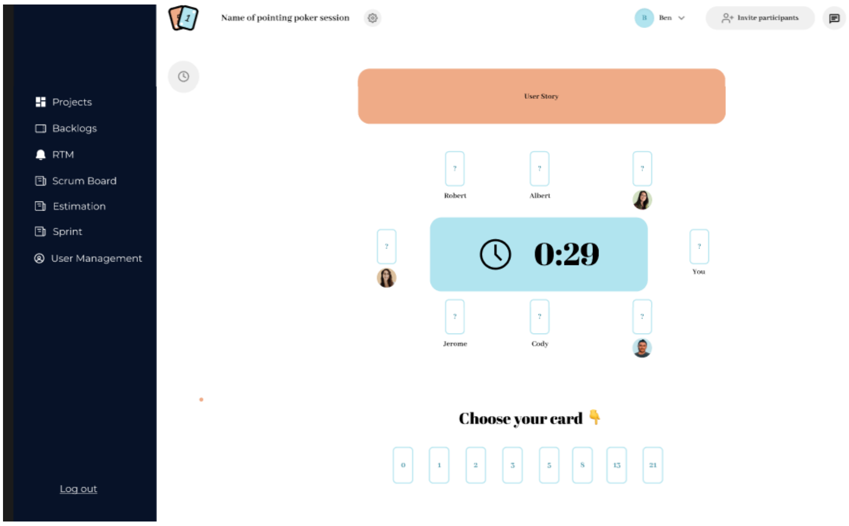


Figure 5.10: Planning Poker Page

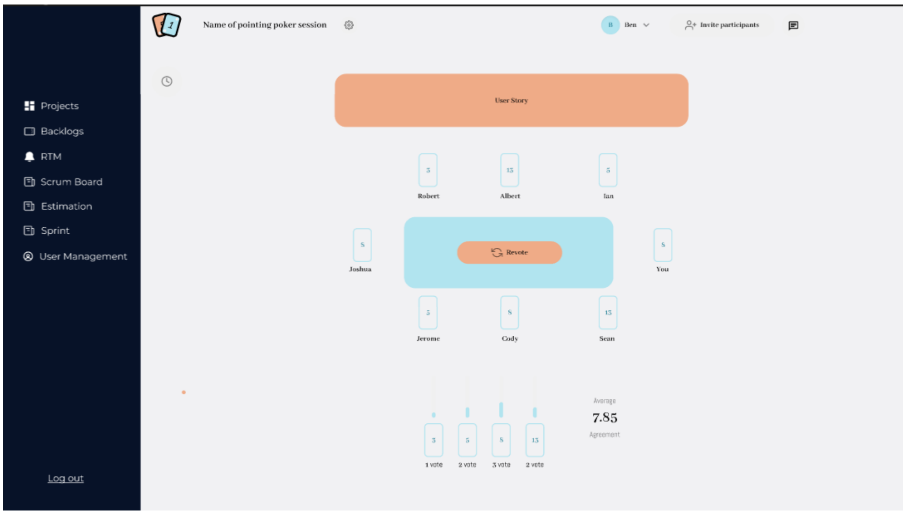
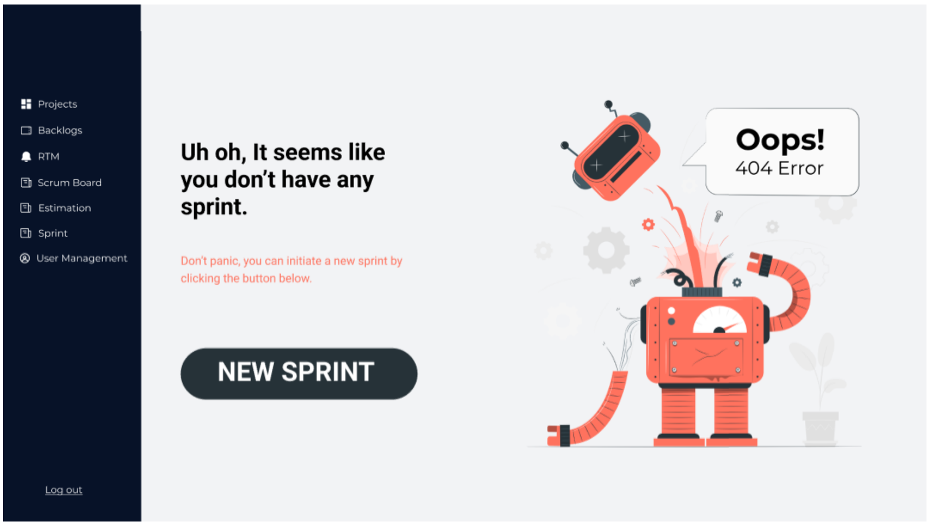


Figure 5.11: Planning Poker Page (Revote)



Uh oh, It seems like you don't have any sprint.

Don't panic, you can initiate a new sprint by clicking the button below.

Figure 5.12: Sprint Page (No Sprint)

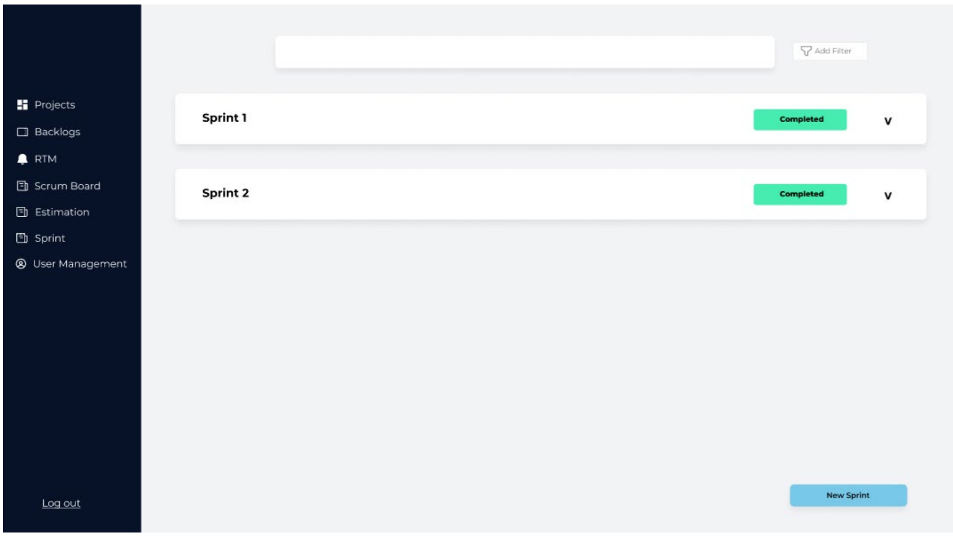
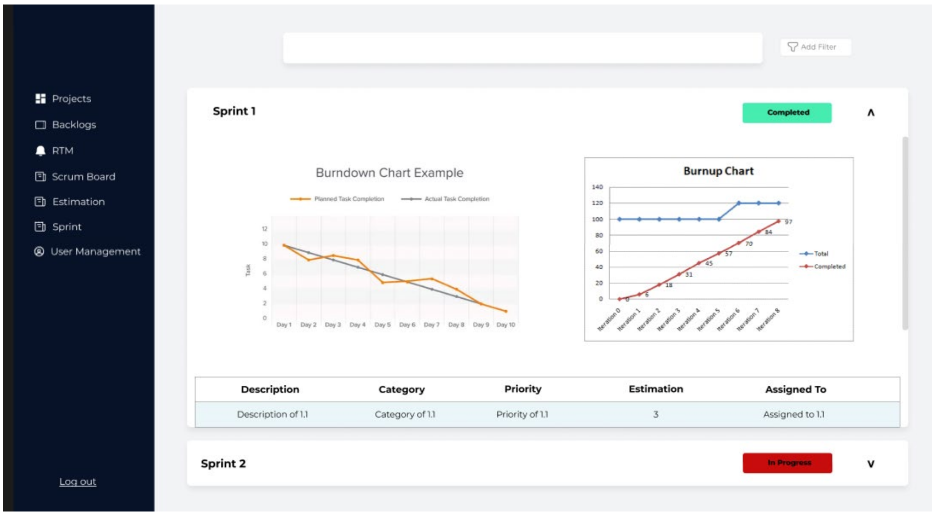
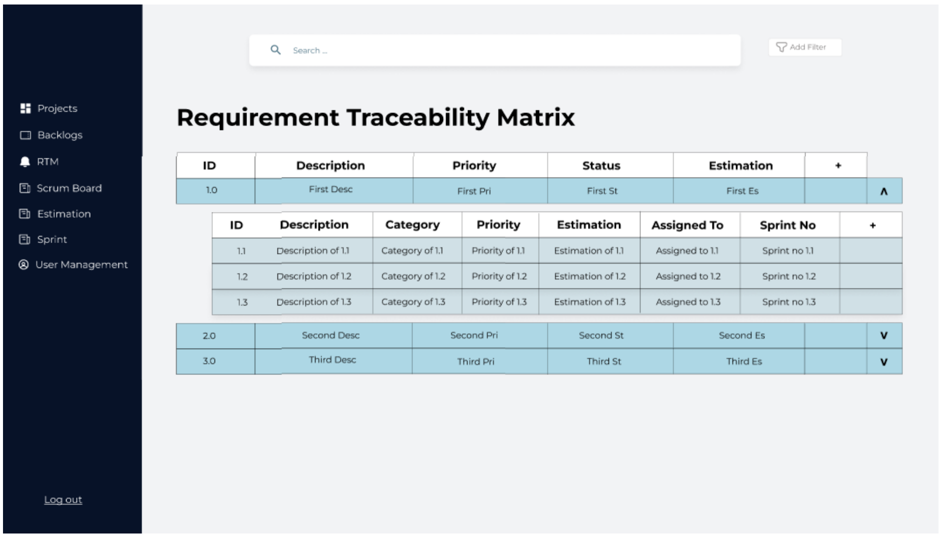


Figure 5.13: Sprint Page



Description	Category	Priority	Estimation	Assigned To
Description of 1.1	Category of 1.1	Priority of 1.1	3	Assigned to 1.1

Figure 5.14: Sprint Page (Expanded)



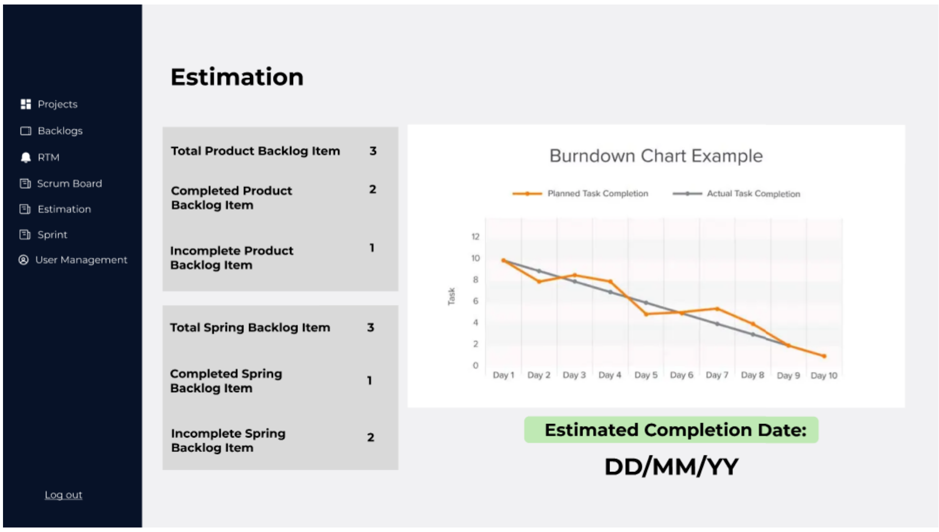
Requirement Traceability Matrix

ID	Description	Priority	Status	Estimation	+	
1.0	First Desc	First Pri	First St	First Es		^

ID	Description	Category	Priority	Estimation	Assigned To	Sprint No	+
1.1	Description of 1.1	Category of 1.1	Priority of 1.1	Estimation of 1.1	Assigned to 1.1	Sprint no 1.1	
1.2	Description of 1.2	Category of 1.2	Priority of 1.2	Estimation of 1.2	Assigned to 1.2	Sprint no 1.2	
1.3	Description of 1.3	Category of 1.3	Priority of 1.3	Estimation of 1.3	Assigned to 1.3	Sprint no 1.3	

2.0	Second Desc	Second Pri	Second St	Second Es		V
3.0	Third Desc	Third Pri	Third St	Third Es		V

Figure 5.15: RTM Page



Estimation

Total Product Backlog Item	3
Completed Product Backlog Item	2
Incomplete Product Backlog Item	1
Total Spring Backlog Item	3
Completed Spring Backlog Item	1
Incomplete Spring Backlog Item	2

Estimated Completion Date:

DD/MM/YY

Figure 5.16: Estimation Page

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Introduction

In this chapter, the project delves into the implementation details of the solutions designed to address the problems. The developed web application aims to enhance visibility of the SCRUM project timeline and forecast project completion dateline. Section 6.2 discusses the file structures of the web application, comprising both the frontend and backend, detailing the setup of structures to achieve the intended software architecture and design pattern. Section 6.3 covers the implemented security mechanisms, ensuring that only authenticated and authorized users can access the application. In Section 6.4, the project explores the API routes developed for the web application, detailing the endpoints and functionalities provided. Lastly, Section 6.6 delves into the modules developed for the web application, showcasing sample screens along with partial code snippets.

6.2 File Structures

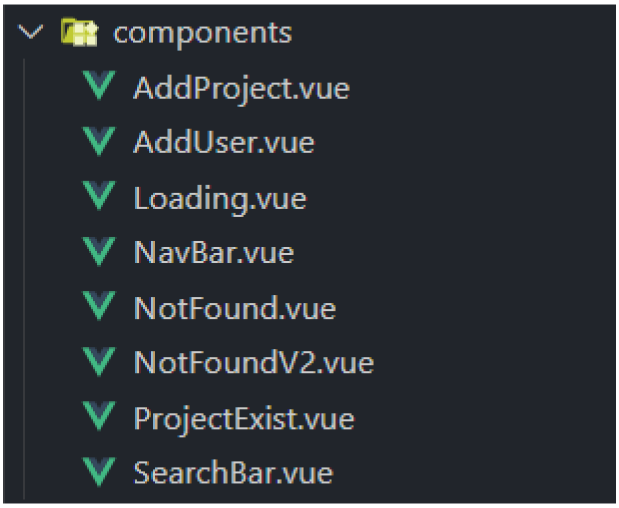
The file structures of a web application play an important role in defining its design pattern and overall architecture. In the web application, the MVC architecture has been adopted, with Vue.js serving as the View layer and Laravel handling the Model and Controller layers.

Vue.js implements the component design pattern, allowing for the creation of reusable UI components. On the other hand, Laravel acts as the Model and Controller layers of the MVC architecture. To enhance its functionality, service-repository pattern has been integrated, which separates business logic from data access operations. This promotes a cleaner and more maintainable codebase.

To allow the communication between Vue.js and Laravel, Axios package will be utilized, in which it allows Vue.js to send HTTP requests to the API endpoints of Laravel. These API endpoints contain the necessary logic implemented in Laravel to process the requests and interact with the database.

6.2.1 Vue.js

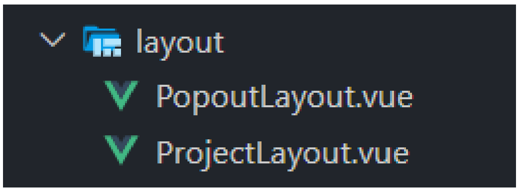
As mentioned earlier, Vue.js implements component design pattern, allowing code reuse to minimize redundancy in the project. The Vue.js project structure consists of three main folders: Components, Layout, and Views.



```
components
    AddProject.vue
    AddUser.vue
    Loading.vue
    NavBar.vue
    NotFound.vue
    NotFoundV2.vue
    ProjectExist.vue
    SearchBar.vue
```

Figure 6.1: Component Folder

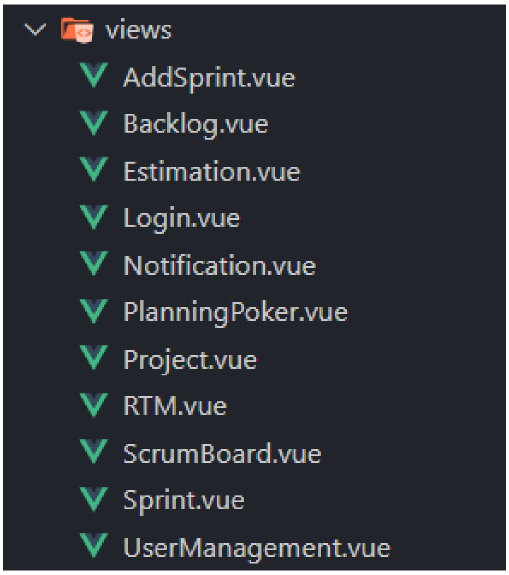
The Component folder contains various reusable components that can be utilized across multiple pages.



```
layout
    PopoutLayout.vue
    ProjectLayout.vue
```

Figure 6.2: Layout Folder

The Layout folder defines the overall layout structure of the project. It comprises components and logic specific to each layout. Different pages may have different layouts, but these layouts are designed to be reusable, reducing the amount of repetitive layout code in each view.



```
views
    AddSprint.vue
    Backlog.vue
    Estimation.vue
    Login.vue
    Notification.vue
    PlanningPoker.vue
    Project.vue
    RTM.vue
    ScrumBoard.vue
    Sprint.vue
    UserManagement.vue
```

Figure 6.3: View Folder

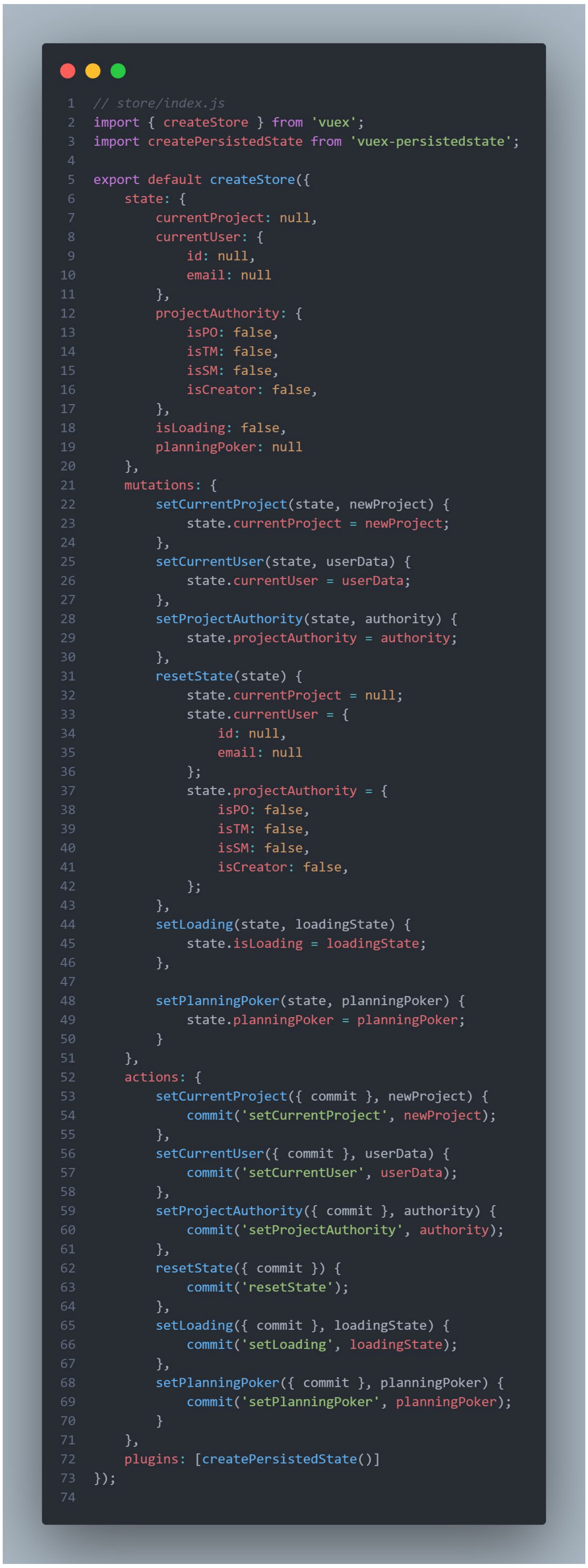
The View folder contains the main pages displayed in the web application. Each view has its own layout and may include multiple reusable components. Additionally, each view may have unique code logic specific to its functionality. This unique logic is contained within the individual pages, as it is not needed by other pages, thus promoting code encapsulation and ensuring that only necessary code is included in each view.

Besides that, there are also subfiles which plays a crucial role in the web application.

```
store
    index.js
```

Figure 6.4: Store Folder

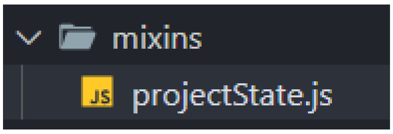
Additionally, the Store folder contains logic codes utilized by the Vuex package. Vuex is a frontend state management library that allows for the initiation and management of frontend sessions.



```javascript
// store/index.js
import { createStore } from 'vuex';
import createPersistedState from 'vuex-persistedstate';

export default createStore({
    state: {
        currentProject: null,
        currentUser: {
            id: null,
            email: null
        },
        projectAuthority: {
            isPO: false,
            isTM: false,
            isSM: false,
            isCreator: false,
        },
        isLoading: false,
        planningPoker: null
    },
    mutations: {
        setCurrentProject(state, newProject) {
            state.currentProject = newProject;
        },
        setCurrentUser(state, userData) {
            state.currentUser = userData;
        },
        setProjectAuthority(state, authority) {
            state.projectAuthority = authority;
        },
        resetState(state) {
            state.currentProject = null;
            state.currentUser = {
                id: null,
                email: null
            };
            state.projectAuthority = {
                isPO: false,
                isTM: false,
                isSM: false,
                isCreator: false,
            };
        },
        setLoading(state, loadingState) {
            state.isLoading = loadingState;
        },

        setPlanningPoker(state, planningPoker) {
            state.planningPoker = planningPoker;
        }
    },
    actions: {
        setCurrentProject({ commit }, newProject) {
            commit('setCurrentProject', newProject);
        },
        setCurrentUser({ commit }, userData) {
            commit('setCurrentUser', userData);
        },
        setProjectAuthority({ commit }, authority) {
            commit('setProjectAuthority', authority);
        },
        resetState({ commit }) {
            commit('resetState');
        },
        setLoading({ commit }, loadingState) {
            commit('setLoading', loadingState);
        },
        setPlanningPoker({ commit }, planningPoker) {
            commit('setPlanningPoker', planningPoker);
        }
    },
    plugins: [createPersistedState()]
});

```

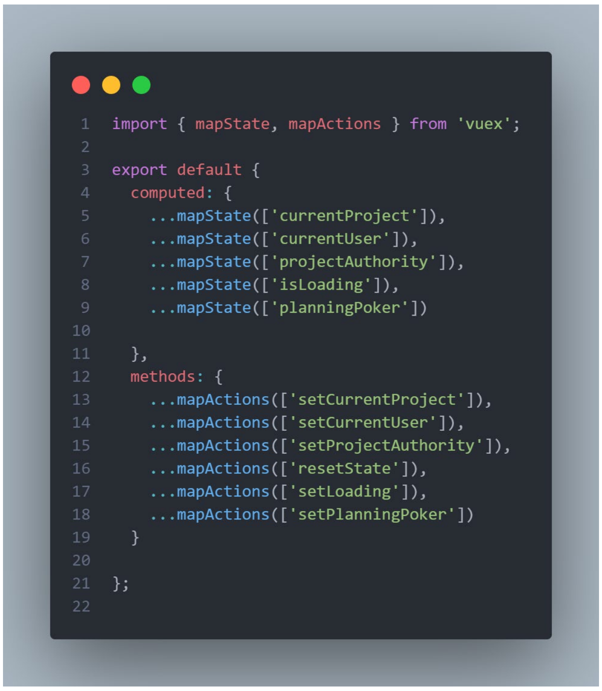
Figure 6.5: index.js File in Store Folder

The Vuex session consists of several elements, including the currently selected project, the currently authorized user, project authority information, planning poker session data, and loading indicators.



```
mixins
  projectState.js
```

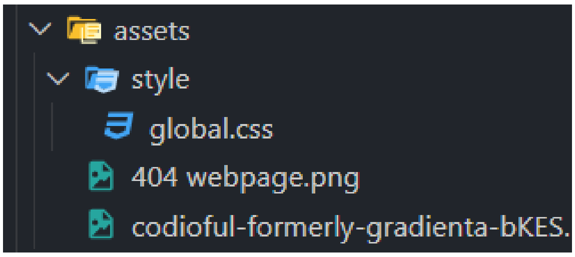
Figure 6.6: Mixins Folder



```
import { mapState, mapActions } from 'vuex';

export default {
  computed: {
    ...mapState(['currentProject']),
    ...mapState(['currentUser']),
    ...mapState(['projectAuthority']),
    ...mapState(['isLoading']),
    ...mapState(['planningPoker'])

  },
  methods: {
    ...mapActions(['setCurrentProject']),
    ...mapActions(['setCurrentUser']),
    ...mapActions(['setProjectAuthority']),
    ...mapActions(['resetState']),
    ...mapActions(['setLoading']),
    ...mapActions(['setPlanningPoker'])
  }

};
```

Figure 6.7: projectState.js

Mixins can be considered as global functions accessible throughout the entire frontend of the application. Therefore, all functions used to modify the Vuex session mentioned earlier will be implemented here. This ensures that every page has access to modify the sessions when necessary, reducing the repetition of codes writing.



```
assets
  style
    global.css
  404 webpage.png
  codioful-formerly-gradienta-bKES.
```

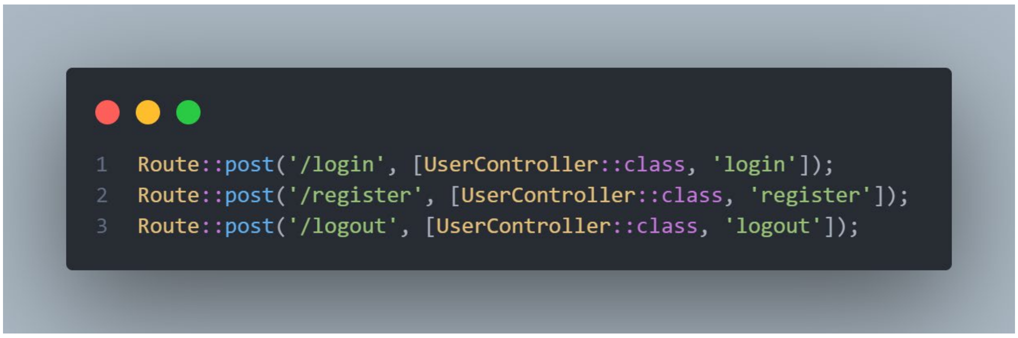
Figure 6.8: Assets Folder

A global.css file has been created to reuse CSS styles across all pages of the application.

With the organized file structures, component design pattern will be implemented, ensuring a clean and minimal code structures.

6.2.2 Laravel

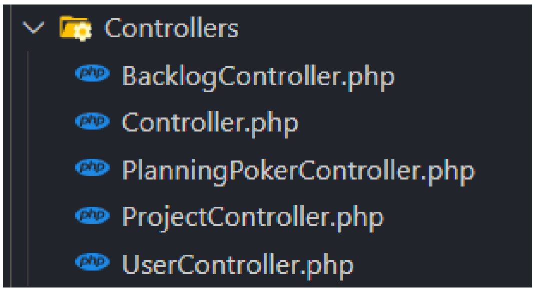
To implement service-repository pattern and MVC architecture in Laravel, 5 different folders are required: Controller, Services, Contracts, Repositories and Models. Given the following scenario:



```
Route::post('/login', [UserController::class, 'login']);
Route::post('/register', [UserController::class, 'register']);
Route::post('/logout', [UserController::class, 'logout']);
```

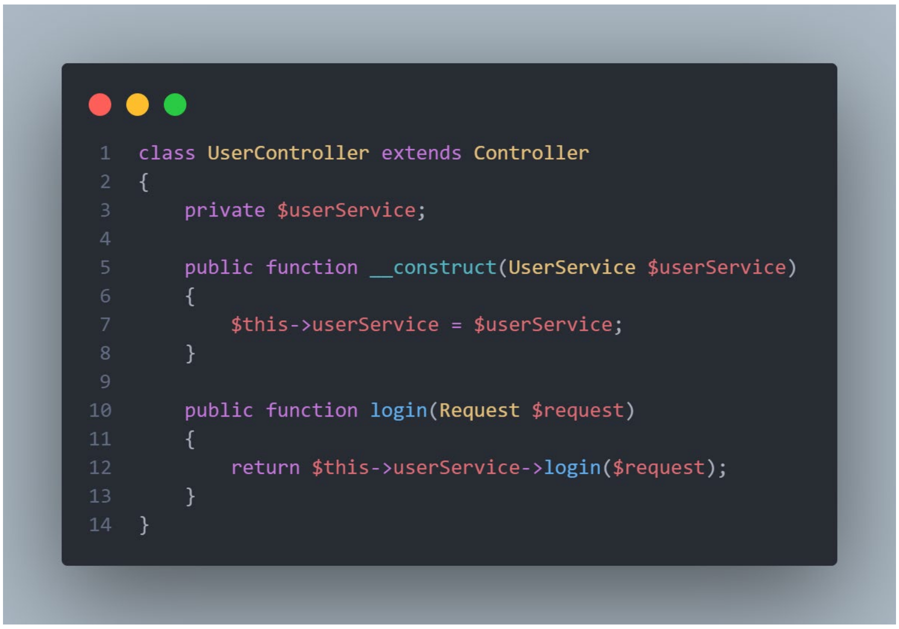
Figure 6.9: Snippet of API Routes of the Web Application

When a user makes a login request from the frontend, it will be redirected to the User Controller, which is in the controller folder.



```
Controllers
    BacklogController.php
    Controller.php
    PlanningPokerController.php
    ProjectController.php
    UserController.php
```

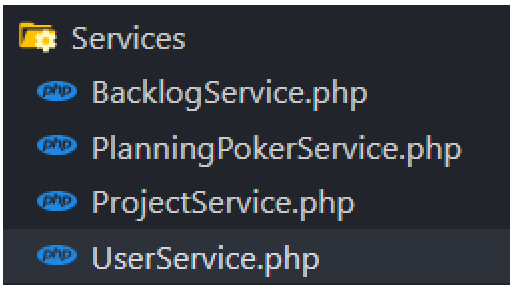
Figure 6.10: Controller Folder



```php
class UserController extends Controller
{
    private $userService;

    public function __construct(UserService $userService)
    {
        $this->userService = $userService;
    }

    public function login(Request $request)
    {
        return $this->userService->login($request);
    }
}
```

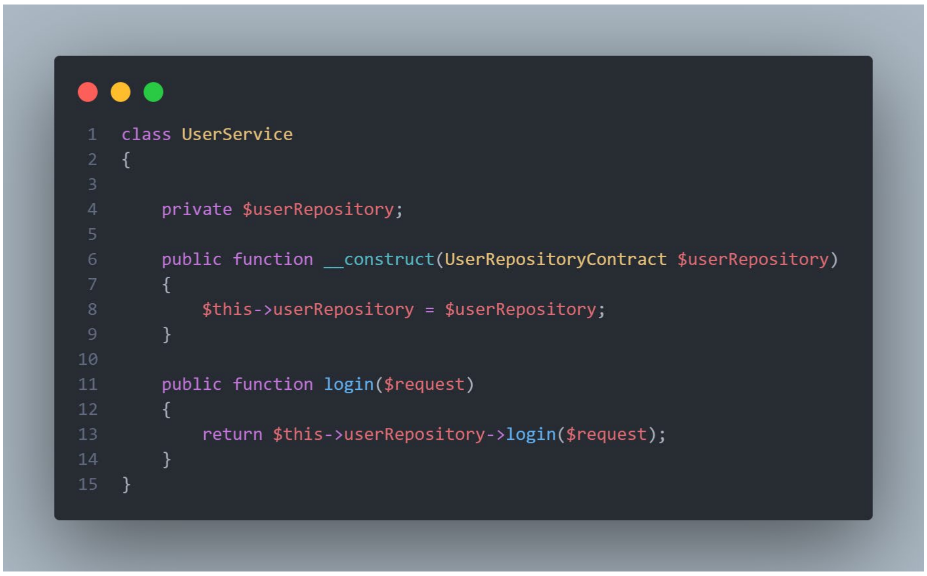
Figure 6.11: Snippet of User Controller

The controller will construct a service from the Service folder, and the login request will utilize the service login function.



```
Services
    BacklogService.php
    PlanningPokerService.php
    ProjectService.php
    UserService.php
```

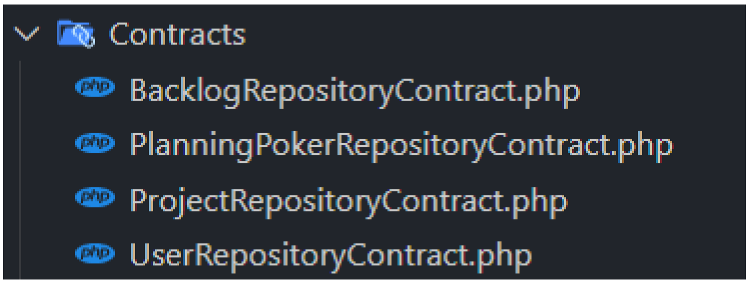
Figure 6.12: Services Folder



```php
class UserService
{

    private $userRepository;

    public function __construct(UserRepositoryContract $userRepository)
    {
        $this->userRepository = $userRepository;
    }

    public function login($request)
    {
        return $this->userRepository->login($request);
    }
}
```

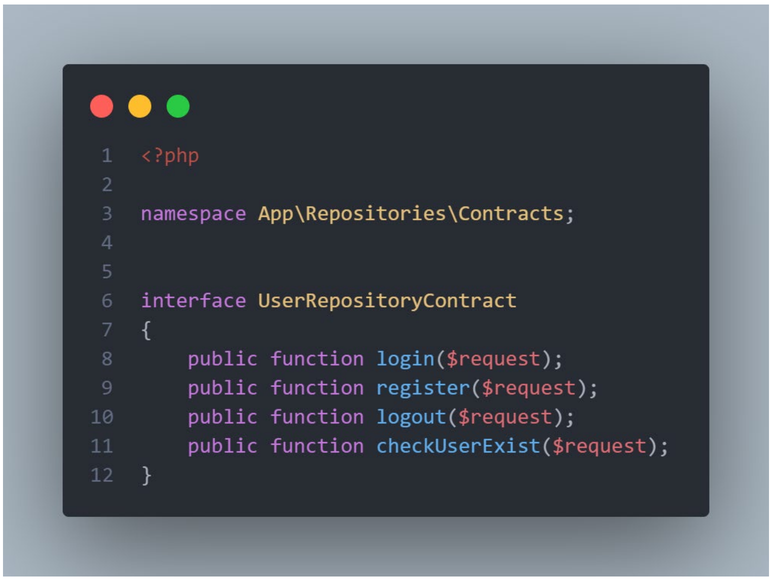
Figure 6.13: Snippet of User Service

The user service then constructs a repository contract from the Contracts folder. While filtering of data requests is typically required in the service, since login does not necessitate any filtering, the web application would just redirect the request to the repository.



```
Contracts
    BacklogRepositoryContract.php
    PlanningPokerRepositoryContract.php
    ProjectRepositoryContract.php
    UserRepositoryContract.php
```

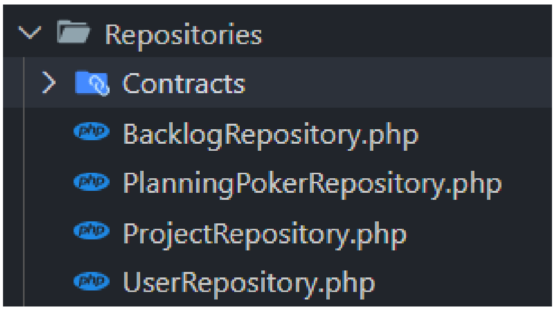
Figure 6.14: Contracts Folder



```php
<?php

namespace App\Repositories\Contracts;


interface UserRepositoryContract
{
    public function login($request);
    public function register($request);
    public function logout($request);
    public function checkUserExist($request);
}
```

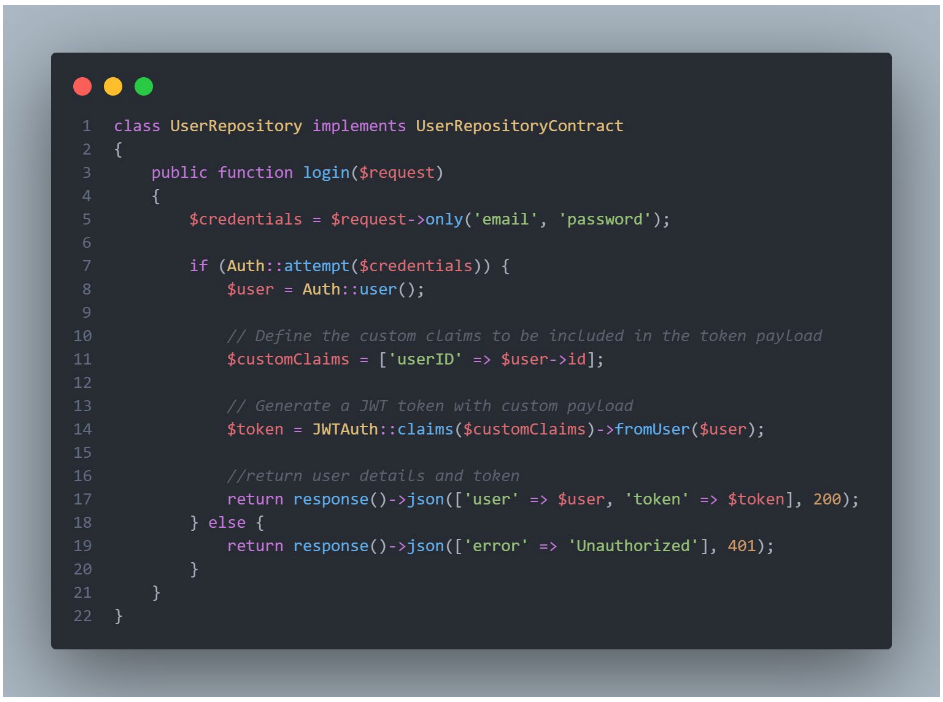
Figure 6.15: User Repository Contract

The repository contract consists of interfaces that the repository will implement, in which it will define all the necessary functions.



```
Repositories
  > Contracts
    BacklogRepository.php
    PlanningPokerRepository.php
    ProjectRepository.php
    UserRepository.php
```

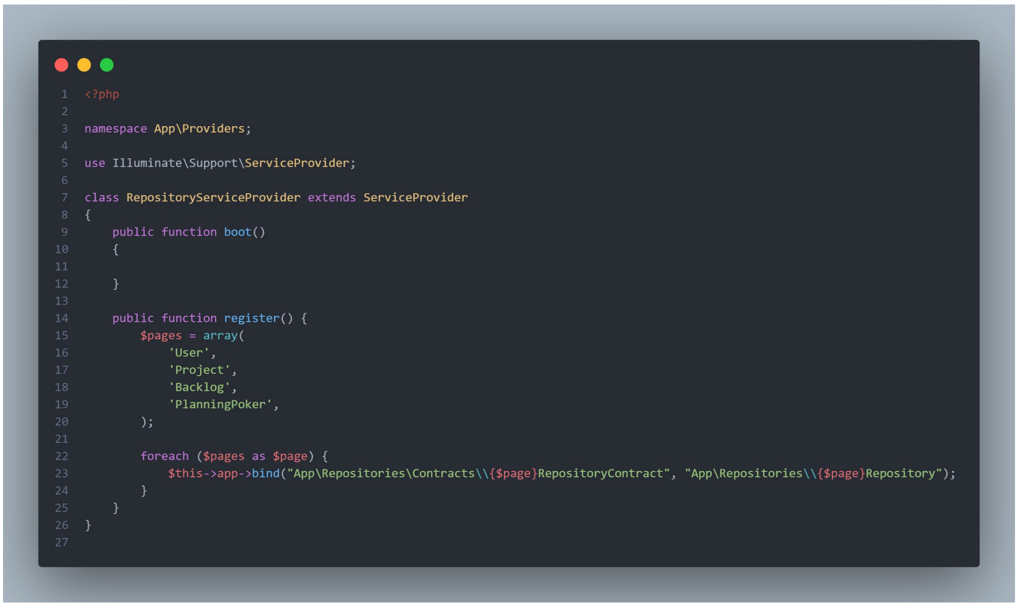
Figure 6.16: Repositories Folder



```php
class UserRepository implements UserRepositoryContract
{
    public function login($request)
    {
        $credentials = $request->only('email', 'password');

        if (Auth::attempt($credentials)) {
            $user = Auth::user();

            // Define the custom claims to be included in the token payload
            $customClaims = ['userID' => $user->id];

            // Generate a JWT token with custom payload
            $token = JWTAuth::claims($customClaims)->fromUser($user);

            //return user details and token
            return response()->json(['user' => $user, 'token' => $token], 200);
        } else {
            return response()->json(['error' => 'Unauthorized'], 401);
        }
    }
}
```

Figure 6.17: Snippet of User Repository

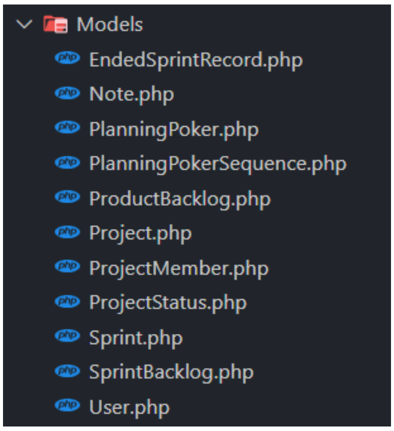
The user repository will implement the repository contract, executing the necessary functions and queries with the database. It will send requests to the Model and receive responses from it.



```php
<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;

class RepositoryServiceProvider extends ServiceProvider
{
    public function boot()
    {

    }

    public function register() {
        $pages = array(
            'User',
            'Project',
            'Backlog',
            'PlanningPoker',
        );

        foreach ($pages as $page) {
            $this->app->bind("App\Repositories\Contracts\\{$page}RepositoryContract", "App\Repositories\\{$page}Repository");
        }
    }
}

```

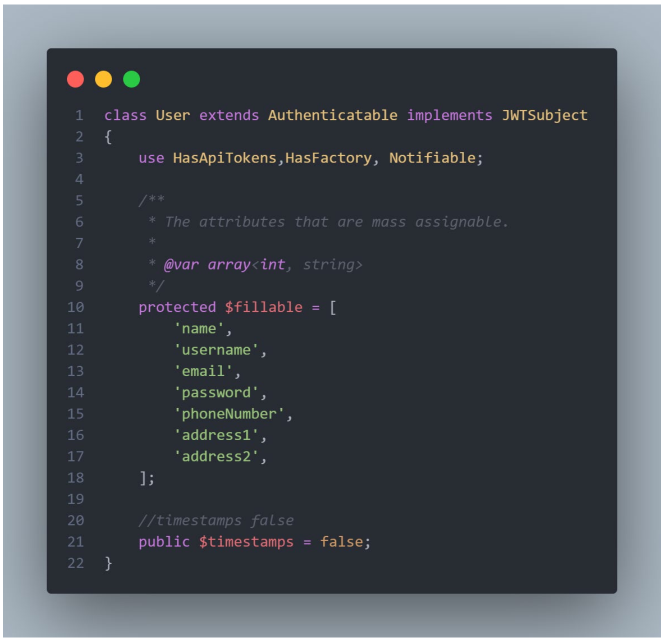
Figure 6.18: Repository Service Provider

To ensure that the repository contracts are bound to the repositories, registration of providers is required.



- Models
  - EndedSprintRecord.php
  - Note.php
  - PlanningPoker.php
  - PlanningPokerSequence.php
  - ProductBacklog.php
  - Project.php
  - ProjectMember.php
  - ProjectStatus.php
  - Sprint.php
  - SprintBacklog.php
  - User.php

Figure 6.19: Models Folder



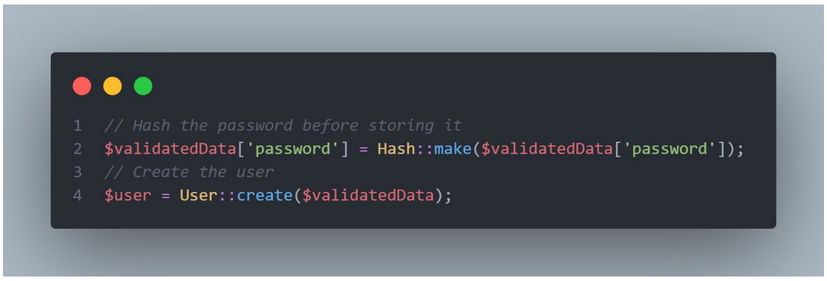
```php
class User extends Authenticatable implements JWTSubject
{
    use HasApiTokens,HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'username',
        'email',
        'password',
        'phoneNumber',
        'address1',
        'address2',
    ];

    //timestamps false
    public $timestamps = false;
}
```

Figure 6.20: Snippet of User Model

The Models folder will consist of multiple models, where each models represent a table in the database.

6.3 Security Mechanism

6.3.1 Hashing



```php
// Hash the password before storing it
$validatedData['password'] = Hash::make($validatedData['password']);
// Create the user
$user = User::create($validatedData);
```

Figure 6.21: Hashing of Password

The web application has implemented hashing encryption for all registered passwords in the database. This measure aims to prevent unauthorized access to the database and retrieval of unencrypted passwords, reducing the risk of abuse. Additionally, password hashing aligns with the identified Non-Functional Requirements.

6.3.2 Cross-Origin Resource Sharing (CORS)

Cross-Origin Resource Sharing, or more commonly known as CORS, is a mechanism that defines which domains are allowed to interact with a server. This mechanism plays a crucial role in preventing Cross-Site Request Forgery (CSRF) attacks.

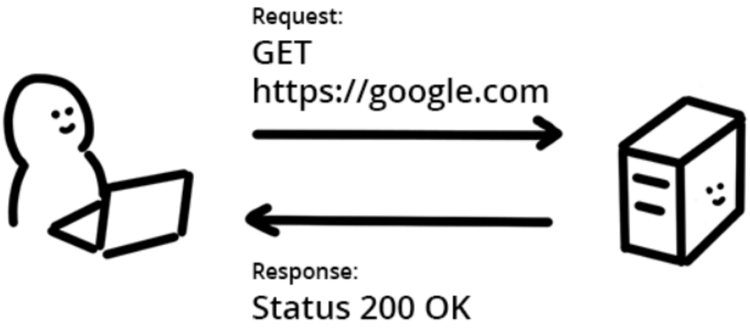
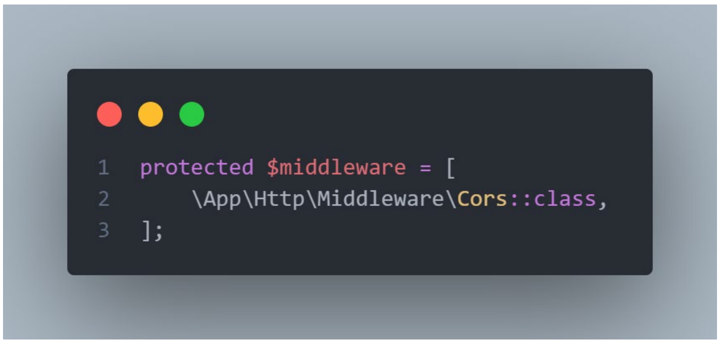


Figure 6.22: CORS Visualization (Kosaka, n.d.).

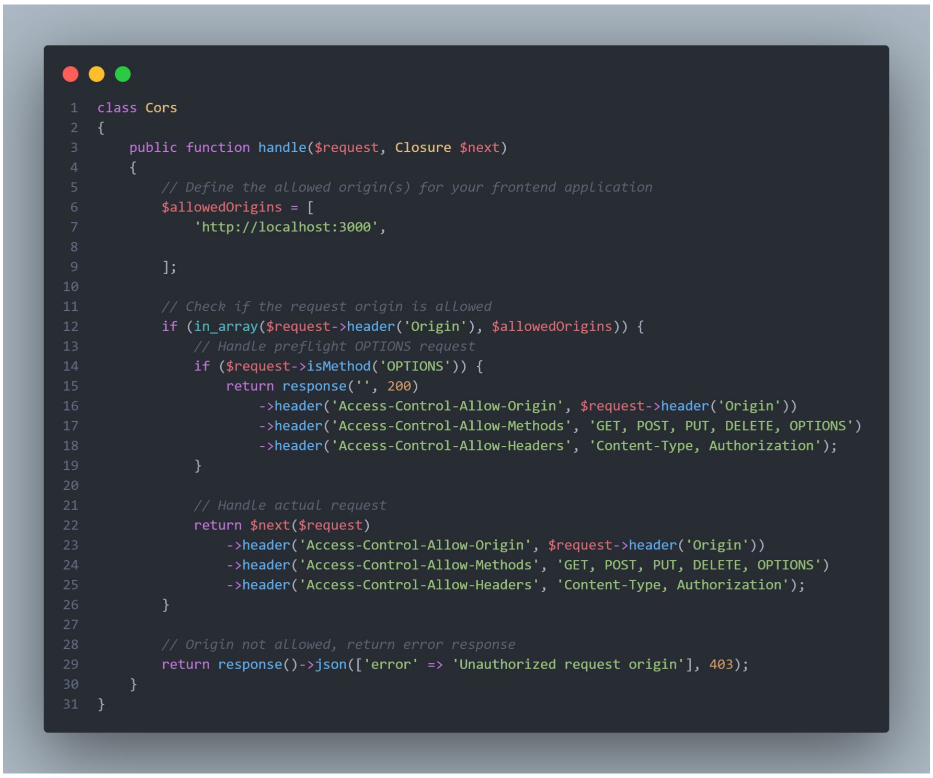
When a user sends a request to the server, CORS checks for the 'Origin' header in the request. This header contains the hosts and ports from which the request originated. If the CORS mechanism verifies that the request's Origin header is 'whitelisted' on the server, it allows the request to proceed, allowing the interactions between the user and the server. However, if the Origin header is not allowed, the server returns a response of 403 Unauthorized, preventing the user from accessing the server (Kosaka, n.d.).

To implement CORS in Laravel, the usage of middleware is required, and this middleware needs to be registered as a global middleware.



```php
protected $middleware = [
    \App\Http\Middleware\Cors::class,
];
```

Figure 6.23: Middleware Registration



```php
class Cors
{
    public function handle($request, Closure $next)
    {
        // Define the allowed origin(s) for your frontend application
        $allowedOrigins = [
            'http://localhost:3000',

        ];

        // Check if the request origin is allowed
        if (in_array($request->header('Origin'), $allowedOrigins)) {
            // Handle preflight OPTIONS request
            if ($request->isMethod('OPTIONS')) {
                return response('', 200)
                    ->header('Access-Control-Allow-Origin', $request->header('Origin'))
                    ->header('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE, OPTIONS')
                    ->header('Access-Control-Allow-Headers', 'Content-Type, Authorization');
            }

            // Handle actual request
            return $next($request)
                ->header('Access-Control-Allow-Origin', $request->header('Origin'))
                ->header('Access-Control-Allow-Methods', 'GET, POST, PUT, DELETE, OPTIONS')
                ->header('Access-Control-Allow-Headers', 'Content-Type, Authorization');
        }

        // Origin not allowed, return error response
        return response()->json(['error' => 'Unauthorized request origin'], 403);
    }
}
```

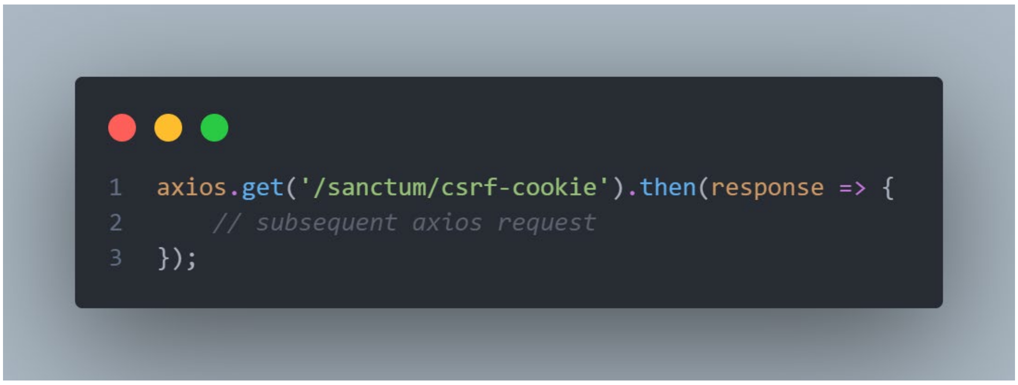
Figure 6.24: CORS Middleware

The allowed origin of the web application is set to http://localhost:3000, which is the development URL used with Vue.js. Any request that includes GET, POST, PUT, DELETE, and OPTIONS methods will be checked with CORS before being allowed to interact. If any other domain attempts to access the backend API, it will result in a status of 403 Unauthorized. This ensures that only requests originating from the specified domain are permitted, helping to prevent unauthorized access to the server.

6.3.3 Cross-Site Request Forgery (CSRF) Token

CSRF attacks are a type of attack that bypasses the authentication mechanism of a web application, allowing unauthorized interaction and potentially forcing unwanted actions. Laravel offers strong protection against CSRF attacks by requiring CSRF verification for any form submitted through Laravel views (KirstenS, n.d.).

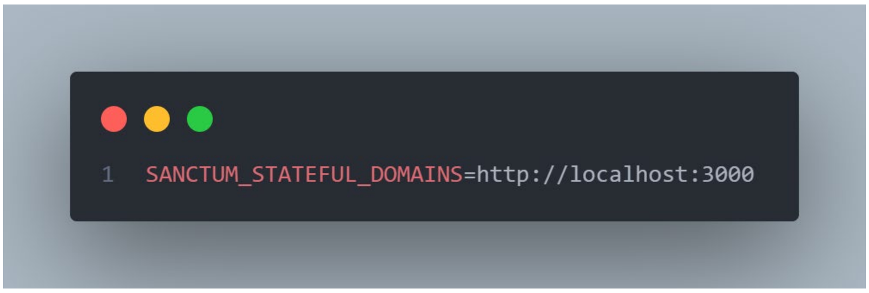
However, since the web application's view is built with Vue.js, the default CSRF protection mechanisms are not automatically applied. To address this, the web application utilizes the Laravel Sanctum package, which automatically sets up CSRF protection for requests sent to the Laravel backend. Consequently, the frontend requires a CSRF token to interact with Laravel endpoints, typically for any method other than GET.



```
axios.get('/sanctum/csrf-cookie').then(response => {
    // subsequent axios request
});
```

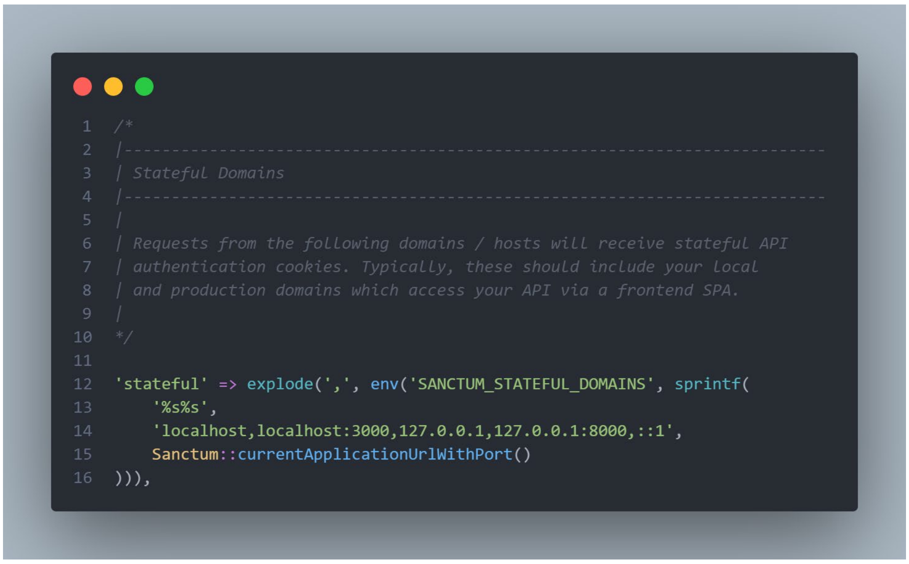
Figure 6.25: Axios CSRF Token Request

To retrieve the CSRF token and access the Laravel API, Vue.js must make an Axios request as shown in Figure 6.23. This request automatically sets the CSRF token for subsequent requests, authorizing access to the API (Laravel, n.d.).



```
SANCTUM_STATEFUL_DOMAINS=http://localhost:3000
```

Figure 6.26: Laravel Environment Setup



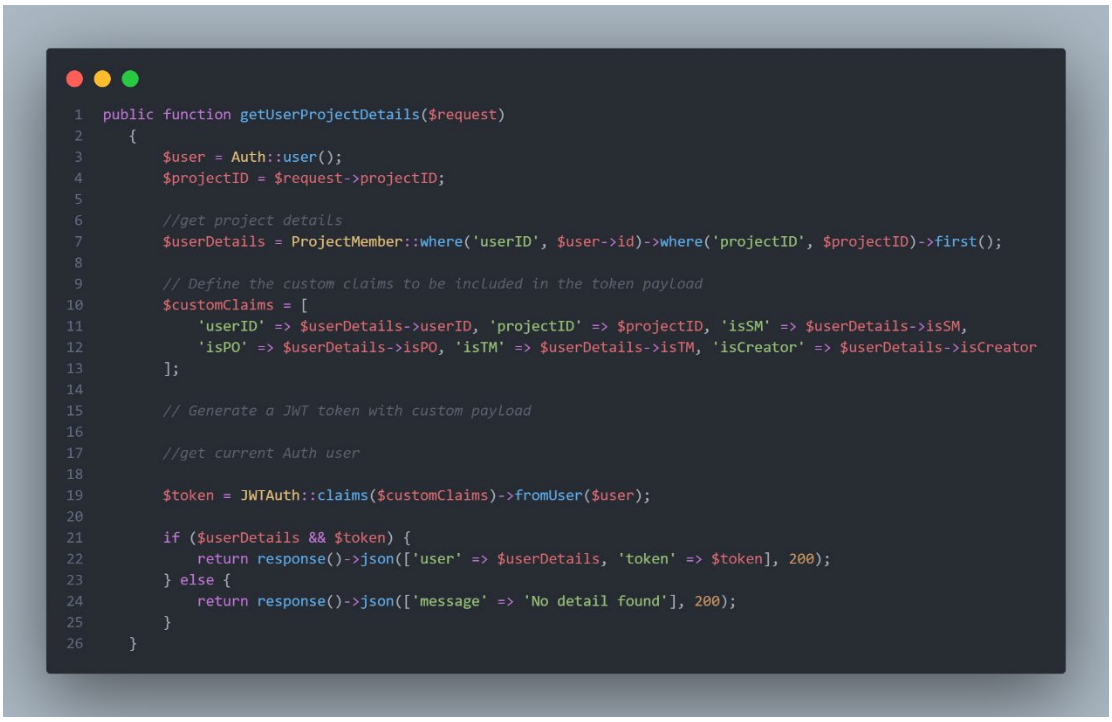
```
/*
|--------------------------------------------------------------------------
| Stateful Domains
|--------------------------------------------------------------------------
|
| Requests from the following domains / hosts will receive stateful API
| authentication cookies. Typically, these should include your local
| and production domains which access your API via a frontend SPA.
|
*/

'stateful' => explode(',', env('SANCTUM_STATEFUL_DOMAINS', sprintf(
    '%s%s',
    'localhost,localhost:3000,127.0.0.1,127.0.0.1:8000,::1',
    Sanctum::currentApplicationUrlWithPort()
))),
```

Figure 6.27: Laravel Sanctum Configuration

Alternatively, a setup using `LARAVEL_SANCTUM_DOMAINS` will allow the frontend application to be exempted from CSRF token authentication. This may not be as secure as the previous option since it exempts specific domains. If any attacker were to set their Origin header similar to the exempted domains, attacks are still possible.

6.3.5 JSON Web Token (JWT)

While Laravel Sanctum provides built-in token authentication, it lacks the high level of customization offered by JWT (JSON Web Tokens). JWT allows tokens to be generated with a payload, which represents customized data that the web application can send to the user. By leveraging this advantage, the web application can generate a customized token for the user. These payloads enhance communication security between the frontend and backend, as the web application relies on the token input instead of request parameters from the frontend. Furthermore, JWT introduces difficulty to modify parameters, as tokens are encrypted.

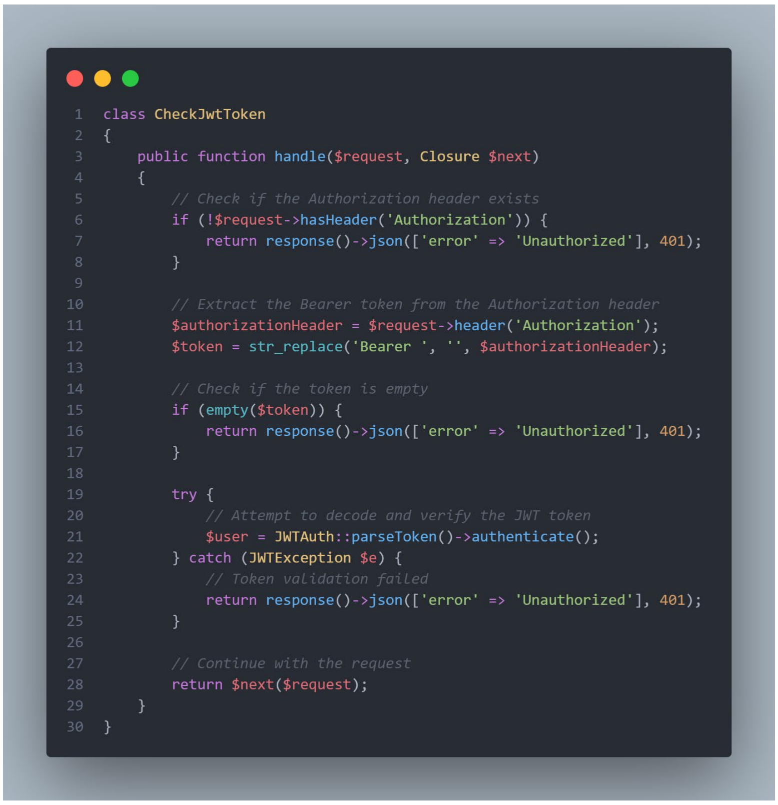


```php
public function getUserProjectDetails($request)
   {
       $user = Auth::user();
       $projectID = $request->projectID;

       //get project details
       $userDetails = ProjectMember::where('userID', $user->id)->where('projectID', $projectID)->first();

       // Define the custom claims to be included in the token payload
       $customClaims = [
           'userID' => $userDetails->userID, 'projectID' => $projectID, 'isSM' => $userDetails->isSM,
           'isPO' => $userDetails->isPO, 'isTM' => $userDetails->isTM, 'isCreator' => $userDetails->isCreator
       ];

       // Generate a JWT token with custom payload

       //get current Auth user

       $token = JWTAuth::claims($customClaims)->fromUser($user);

       if ($userDetails && $token) {
           return response()->json(['user' => $userDetails, 'token' => $token], 200);
       } else {
           return response()->json(['message' => 'No detail found'], 200);
       }
   }
```

Figure 6.28: Code Snippets of User Selecting Project

When a user selects a project, a customized token with the payload containing information such as the user ID, project ID, and project role (e.g., Project Manager/Creator, SCRUM Master, Team Member, or Product Owner) will be generated and sent back to the user.

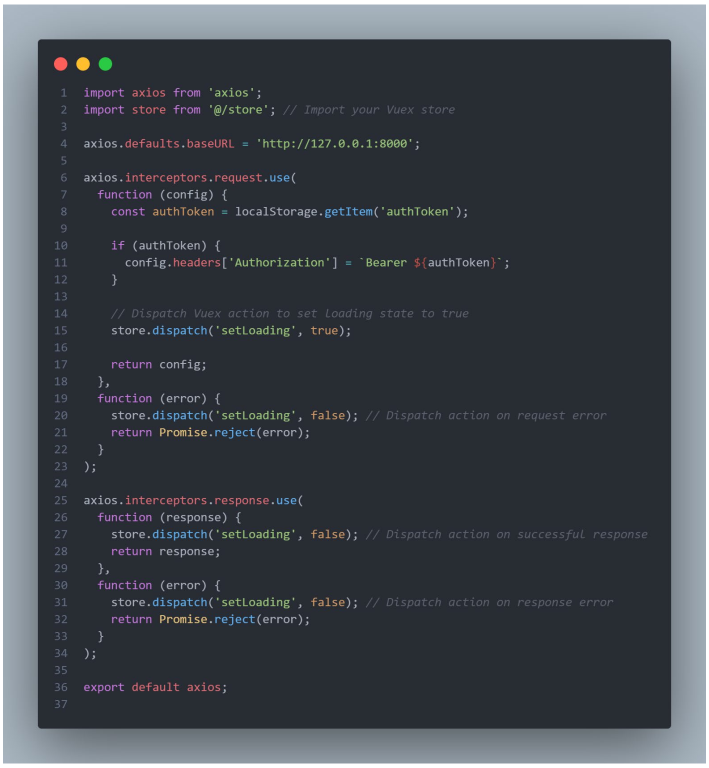
5 different middleware will be used to verify whether if user is authenticated or authorized to use the API of the web applications.



```php
class CheckJwtToken
{
    public function handle($request, Closure $next)
    {
        // Check if the Authorization header exists
        if (!$request->hasHeader('Authorization')) {
            return response()->json(['error' => 'Unauthorized'], 401);
        }

        // Extract the Bearer token from the Authorization header
        $authorizationHeader = $request->header('Authorization');
        $token = str_replace('Bearer ', '', $authorizationHeader);

        // Check if the token is empty
        if (empty($token)) {
            return response()->json(['error' => 'Unauthorized'], 401);
        }

        try {
            // Attempt to decode and verify the JWT token
            $user = JWTAuth::parseToken()->authenticate();
        } catch (JWTException $e) {
            // Token validation failed
            return response()->json(['error' => 'Unauthorized'], 401);
        }

        // Continue with the request
        return $next($request);
    }
}
```

Figure 6.29: CheckJWTToken Middleware

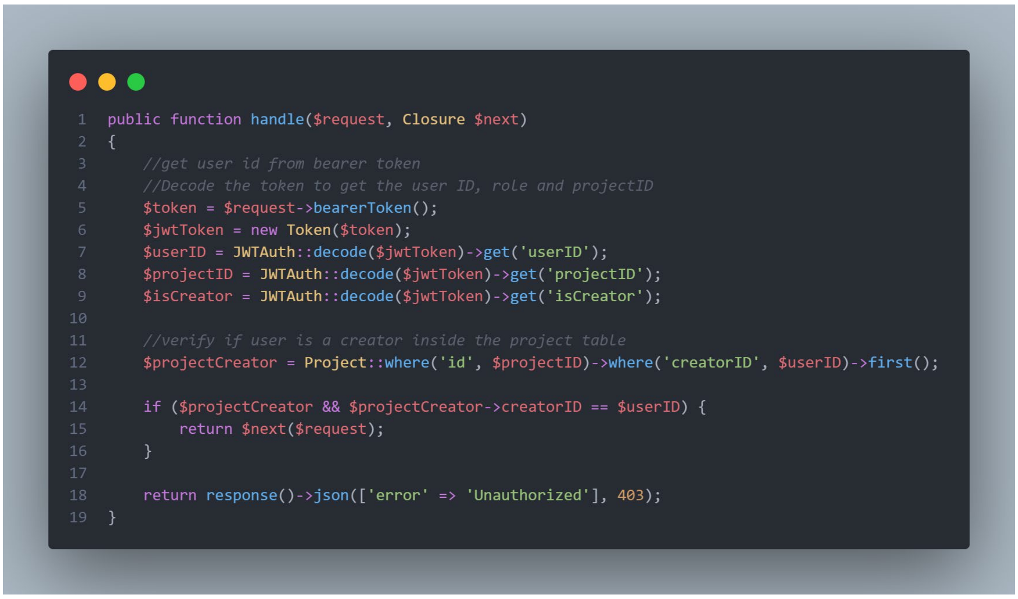
The first middleware would verify if user had a JWT token in their request header. If not, user is not authorized to use the web application.



```javascript
import axios from 'axios';
import store from '@/store'; // Import your Vuex store

axios.defaults.baseURL = 'http://127.0.0.1:8000';

axios.interceptors.request.use(
  function (config) {
    const authToken = localStorage.getItem('authToken');

    if (authToken) {
      config.headers['Authorization'] = `Bearer ${authToken}`;
    }

    // Dispatch Vuex action to set loading state to true
    store.dispatch('setLoading', true);

    return config;
  },
  function (error) {
    store.dispatch('setLoading', false); // Dispatch action on request error
    return Promise.reject(error);
  }
);

axios.interceptors.response.use(
  function (response) {
    store.dispatch('setLoading', false); // Dispatch action on successful response
    return response;
  },
  function (error) {
    store.dispatch('setLoading', false); // Dispatch action on response error
    return Promise.reject(error);
  }
);

export default axios;

```

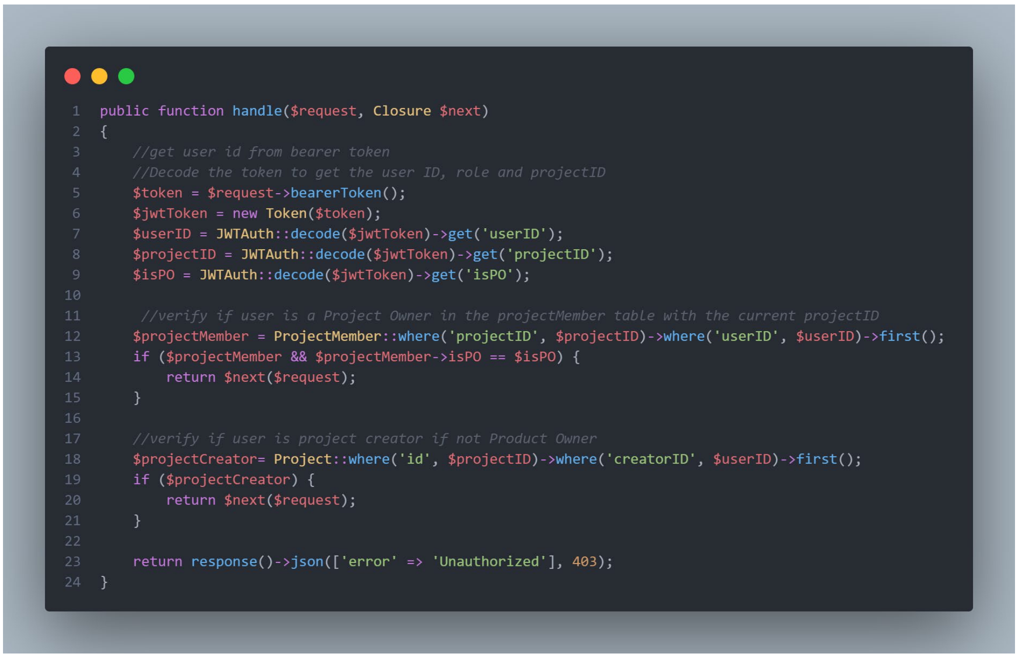
Figure 6.30: Axios Interceptor

An Axios interceptor is a piece of code that intercepts every Axios request sent to Laravel. When intercepted, the interceptor inserts a bearer authorization token, which is the JWT token generated by Laravel. This ensures that every request sent includes the JWT token without requiring explicit code in each request.



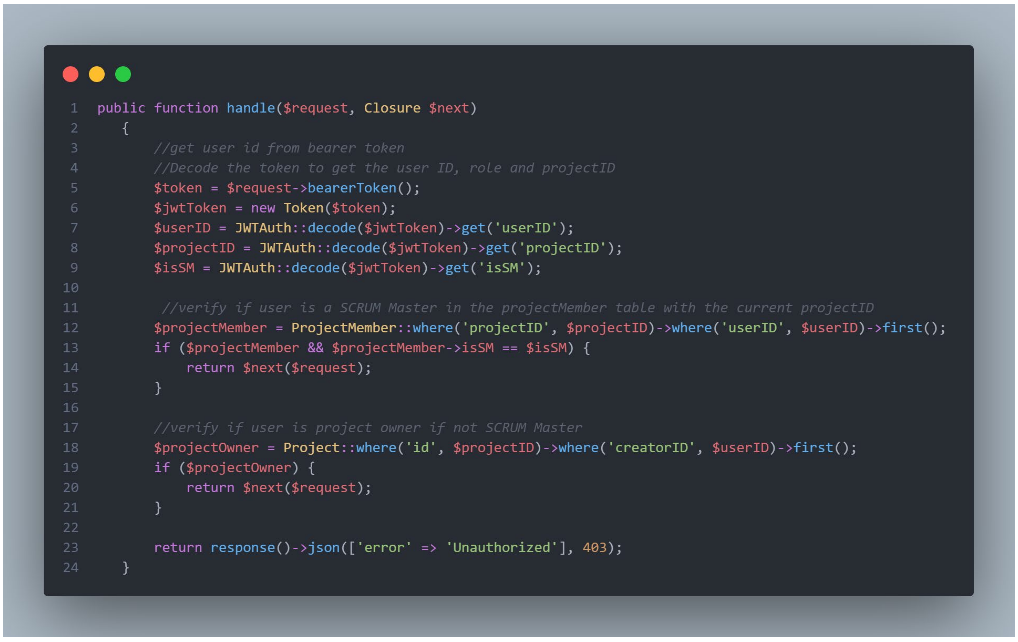
```php
public function handle($request, Closure $next)
{
    //get user id from bearer token
    //Decode the token to get the user ID, role and projectID
    $token = $request->bearerToken();
    $jwtToken = new Token($token);
    $userID = JWTAuth::decode($jwtToken)->get('userID');
    $projectID = JWTAuth::decode($jwtToken)->get('projectID');
    $isCreator = JWTAuth::decode($jwtToken)->get('isCreator');

    //verify if user is a creator inside the project table
    $projectCreator = Project::where('id', $projectID)->where('creatorID', $userID)->first();

    if ($projectCreator && $projectCreator->creatorID == $userID) {
        return $next($request);
    }

    return response()->json(['error' => 'Unauthorized'], 403);
}
```

Figure 6.31: Check Creator Midddleware



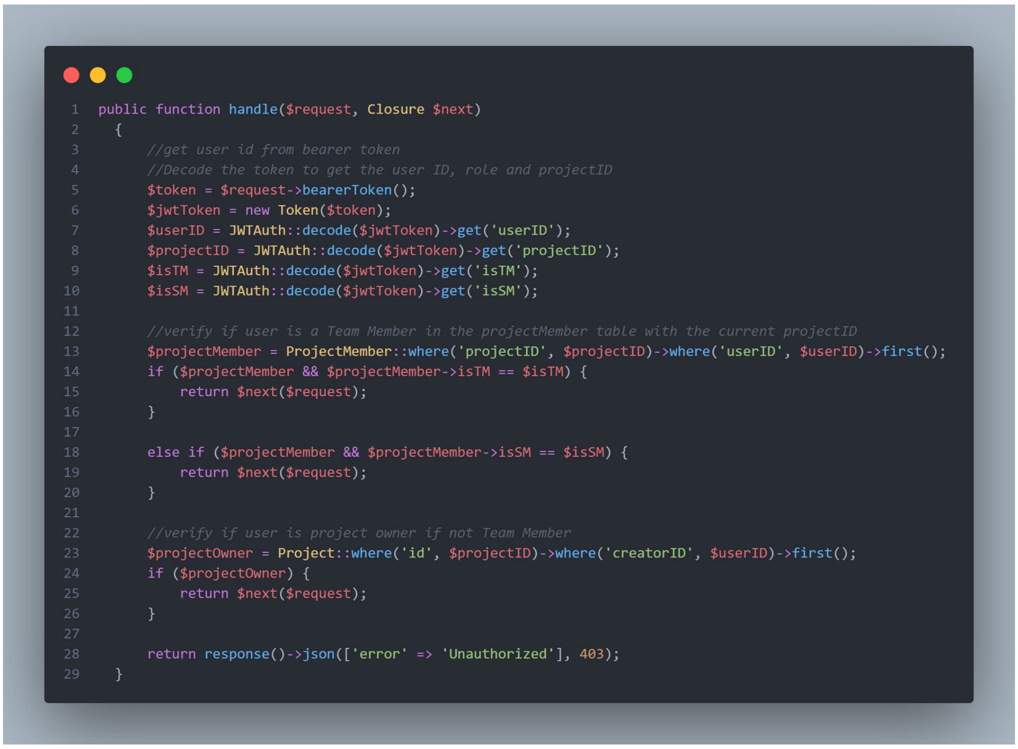
```php
public function handle($request, Closure $next)
{
    //get user id from bearer token
    //Decode the token to get the user ID, role and projectID
    $token = $request->bearerToken();
    $jwtToken = new Token($token);
    $userID = JWTAuth::decode($jwtToken)->get('userID');
    $projectID = JWTAuth::decode($jwtToken)->get('projectID');
    $isPO = JWTAuth::decode($jwtToken)->get('isPO');

     //verify if user is a Project Owner in the projectMember table with the current projectID
    $projectMember = ProjectMember::where('projectID', $projectID)->where('userID', $userID)->first();
    if ($projectMember && $projectMember->isPO == $isPO) {
        return $next($request);
    }

    //verify if user is project creator if not Product Owner
    $projectCreator= Project::where('id', $projectID)->where('creatorID', $userID)->first();
    if ($projectCreator) {
        return $next($request);
    }

    return response()->json(['error' => 'Unauthorized'], 403);
}
```

Figure 6.32: Check Product Owner Middleware



```php
public function handle($request, Closure $next)
    {
        //get user id from bearer token
        //Decode the token to get the user ID, role and projectID
        $token = $request->bearerToken();
        $jwtToken = new Token($token);
        $userID = JWTAuth::decode($jwtToken)->get('userID');
        $projectID = JWTAuth::decode($jwtToken)->get('projectID');
        $isSM = JWTAuth::decode($jwtToken)->get('isSM');

         //verify if user is a SCRUM Master in the projectMember table with the current projectID
        $projectMember = ProjectMember::where('projectID', $projectID)->where('userID', $userID)->first();
        if ($projectMember && $projectMember->isSM == $isSM) {
            return $next($request);
        }

        //verify if user is project owner if not SCRUM Master
        $projectOwner = Project::where('id', $projectID)->where('creatorID', $userID)->first();
        if ($projectOwner) {
            return $next($request);
        }

        return response()->json(['error' => 'Unauthorized'], 403);
    }
```

Figure 6.33: Check SCRUM Master Middleware



```php
public function handle($request, Closure $next)
  {
      //get user id from bearer token
      //Decode the token to get the user ID, role and projectID
      $token = $request->bearerToken();
      $jwtToken = new Token($token);
      $userID = JWTAuth::decode($jwtToken)->get('userID');
      $projectID = JWTAuth::decode($jwtToken)->get('projectID');
      $isTM = JWTAuth::decode($jwtToken)->get('isTM');
      $isSM = JWTAuth::decode($jwtToken)->get('isSM');

      //verify if user is a Team Member in the projectMember table with the current projectID
      $projectMember = ProjectMember::where('projectID', $projectID)->where('userID', $userID)->first();
      if ($projectMember && $projectMember->isTM == $isTM) {
          return $next($request);
      }

      else if ($projectMember && $projectMember->isSM == $isSM) {
          return $next($request);
      }

      //verify if user is project owner if not Team Member
      $projectOwner = Project::where('id', $projectID)->where('creatorID', $userID)->first();
      if ($projectOwner) {
          return $next($request);
      }

      return response()->json(['error' => 'Unauthorized'], 403);
  }
```

Figure 6.34: Check SCRUM Master or Team Member Middleware

The remaining four middleware functions similarly; they are used to verify the roles of the user. Instead of verifying parameters sent by users, the middleware decodes the JWT token, extracts the user's role from the payload, and checks whether the user is authorized to access based on the extracted role and compared it with the database. The middleware checks for roles such as Creator (or Project Owner in the use case description), Product Owner, SCRUM Master, and SCRUM Master or Team Member (since SCRUM Master has the authority of a Team Member). Since the Creator would have superadmin access, all the middleware would also verify if the user is a Creator if the authorized role is not found.

6.4 API Route

The middleware will be represented by abbreviation:

1. **CJWT** – Verify if user has JWT Token
2. **PO** – Verify if user is a Product Owner
3. **SM** – Verify if user is a SCRUM Master
4. **SMTM** – Verify if user is a Team Member or SCRUM Master
5. **CR** – Verify if user is a Creator

Table 6.1: User Controller

Method	**Route**	**Description**	**Middleware**
POST	/login	Login request	-
POST	/register	Register request	-
POST	/logout	To log the user out from the web application and expire the token created	CJWT
GET	/checkUserExist	Verify if a user is registered in the system	CJWT

Table 6.2: Project Controller

Method	**Route**	**Description**	**Middleware**
POST	/createProject	To create a new project in the system	CJWT
GET	/checkUserExistProject	To check if a user exists before inviting	CJWT

GET	/getProjectList	To get the list of projects the user is involved in	CJWT
GET	/getUserProjectDetails	To get the project details of the project selected	CJWT
GET	/getProjectMembers	To get the project member details of the project selected	CJWT
GET	/getProjectStatus	To get the customized project status (SCRUM Board)	CJWT
GET	/getCompletionDate	To get the estimated completion date of the project	CJWT
POST	/addUserToProject	To invite a user into a project	CJWT, CR
DELETE	/deleteProjectMember	To remove a user from a project	CJWT, CR
PUT	/updateProjectMemberRole	To update a user role in a project	CJWT, CR
POST	/addProjectStatus	To add a customize status in a project (SCRUM Board)	CJWT, SM
DELETE	/deleteProjectStatus	To delete a customized status in a project (SCRUM Board)	CJWT, SM

Table 6.3: Backlog Controller

Method	**Route**	**Description**	**Middleware**

GET	/getProductBacklog	To get all the backlogs created of a project	CJWT
GET	/getSprint	To get all the sprint details of a project	CJWT
POST	/createProductBacklog	To create a new product backlog	CJWT, CR
PUT	/updateProductBacklog	To update an existing product backlog	CJWT, CR
DELETE	/deleteProductBacklog	To delete an existing product backlog	CJWT, CR
POST	/createSprint	To initiate a new sprint	CJWT, SM
POST	/endSprint	To end an active sprint	CJWT, SM
PUT	/updateSprintBacklog AssignedTo	To update the assignation of a sprint backlog item	CJWT, SMTM
POST	/createSprintBacklog	To create a new sprint backlog item for a product backlog	CJWT, SMTM
PUT	/updateSprintBacklog	To update an existing sprint backlog item	CJWT, SMTM
PUT	/updateSprintBacklog Estimation	To update the estimation of a sprint backlog item	CJWT, SMTM
PUT	/updateSprintBacklog Status	To update the status of a sprint backlog item	CJWT, SMTM
DELETE	/deleteSprintBacklog	To delete an existing sprint backlog item	CJWT, SMTM

Table 6.4: Planning Poker Controller

Method	**Route**	**Description**	**Middleware**
GET	/getNotification	To get the notification of existing planning poker sessions	CJWT

POST	/getPlanningPokerSession	To get the planning poker sessions, POST request as because if no session found, create new one	CJWT
POST	/getSessionDetails	To get the planning poker session details	CJWT
GET	/getNotes	To get the notes associated with the planning poker	CJWT
POST	/addNotes	To add notes associated with the planning poker	CJWT
PUT	/updatePlanningPokerEstiamtion	To update the planning poker estimation effort	CJWT
POST	/revote	To initiate a revote on an ended planning poker session	CJWT, SMTM
GET	/getSequenceNumber	To get the customized sequence number of planning poker session	CJWT, SMTM
POST	/addSequenceNumber	To add new customized sequence number of planning poker session	CJWT, SMTM

6.5 Modules

6.5.1 Login and Logout

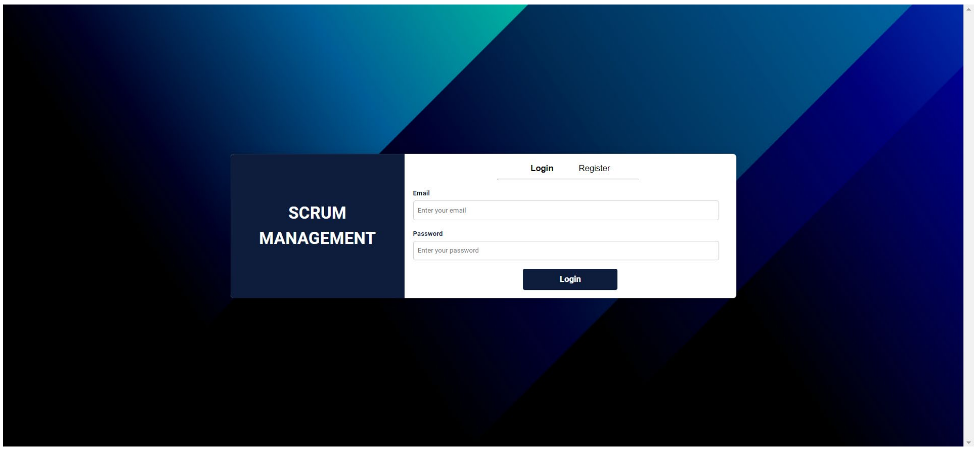
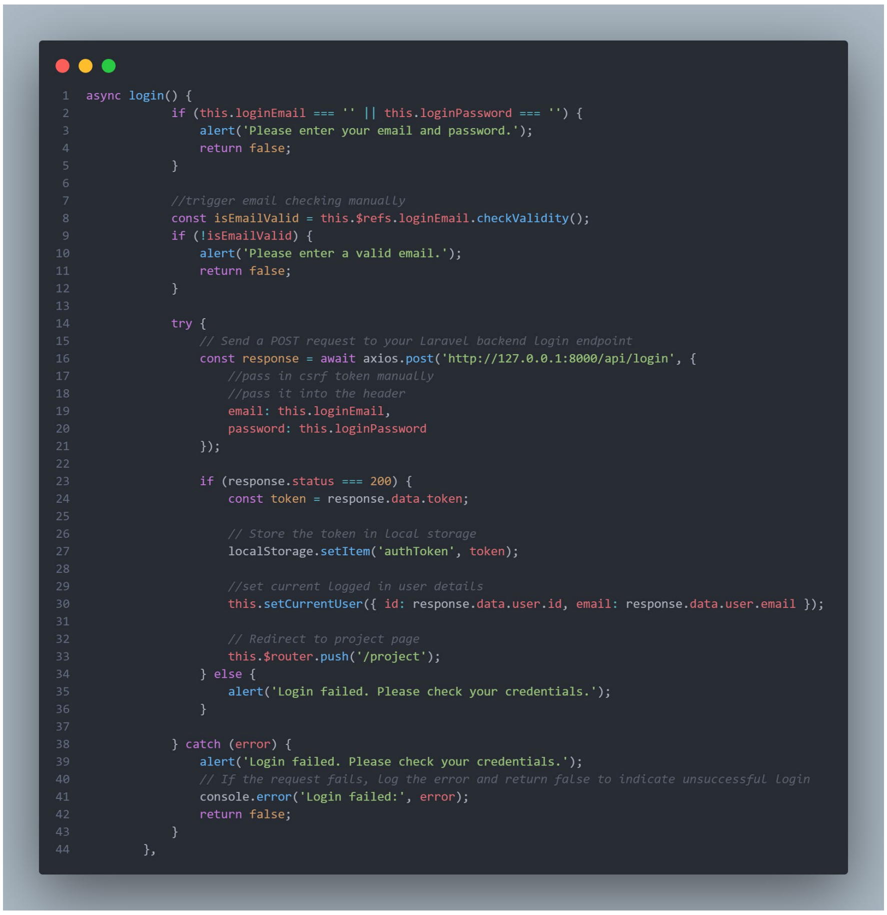


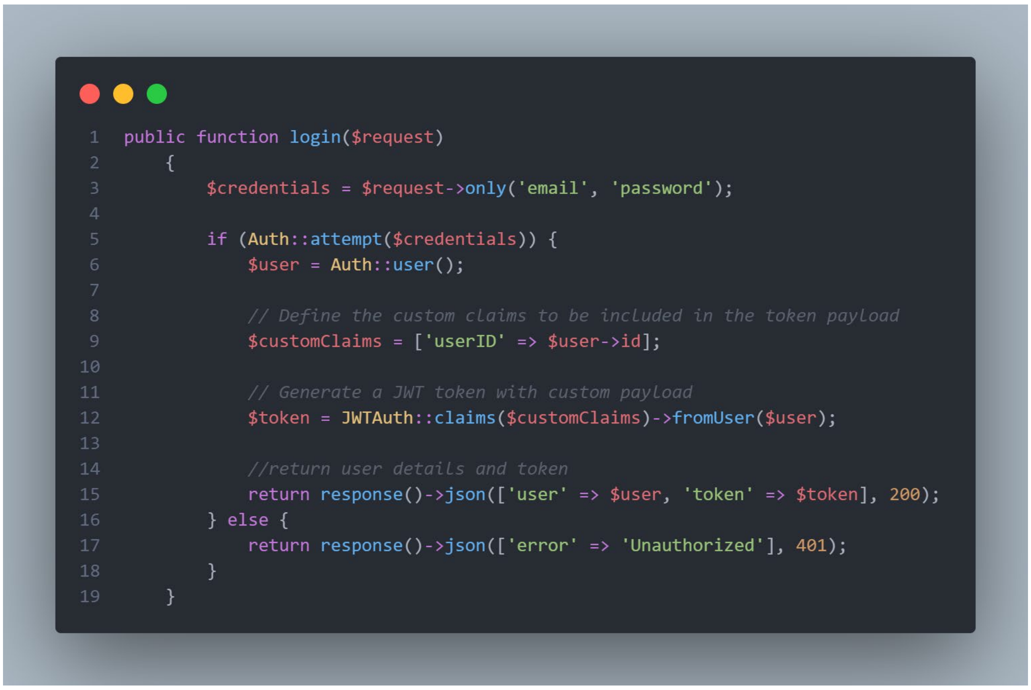
Figure 6.35: Login Page

When the web application first launched, the user will be prompt to login. The user is required to fill in the email field and the password field and clicks on the “Login” button.



```
async login() {
            if (this.loginEmail === '' || this.loginPassword === '') {
                alert('Please enter your email and password.');
                return false;
            }

            //trigger email checking manually
            const isEmailValid = this.$refs.loginEmail.checkValidity();
            if (!isEmailValid) {
                alert('Please enter a valid email.');
                return false;
            }

            try {
                // Send a POST request to your Laravel backend login endpoint
                const response = await axios.post('http://127.0.0.1:8000/api/login', {
                    //pass in csrf token manually
                    //pass it into the header
                    email: this.loginEmail,
                    password: this.loginPassword
                });

                if (response.status === 200) {
                    const token = response.data.token;

                    // Store the token in local storage
                    localStorage.setItem('authToken', token);

                    //set current logged in user details
                    this.setCurrentUser({ id: response.data.user.id, email: response.data.user.email });

                    // Redirect to project page
                    this.$router.push('/project');
                } else {
                    alert('Login failed. Please check your credentials.');
                }

            } catch (error) {
                alert('Login failed. Please check your credentials.');
                // If the request fails, log the error and return false to indicate unsuccessful login
                console.error('Login failed:', error);
                return false;
            }
        },
```

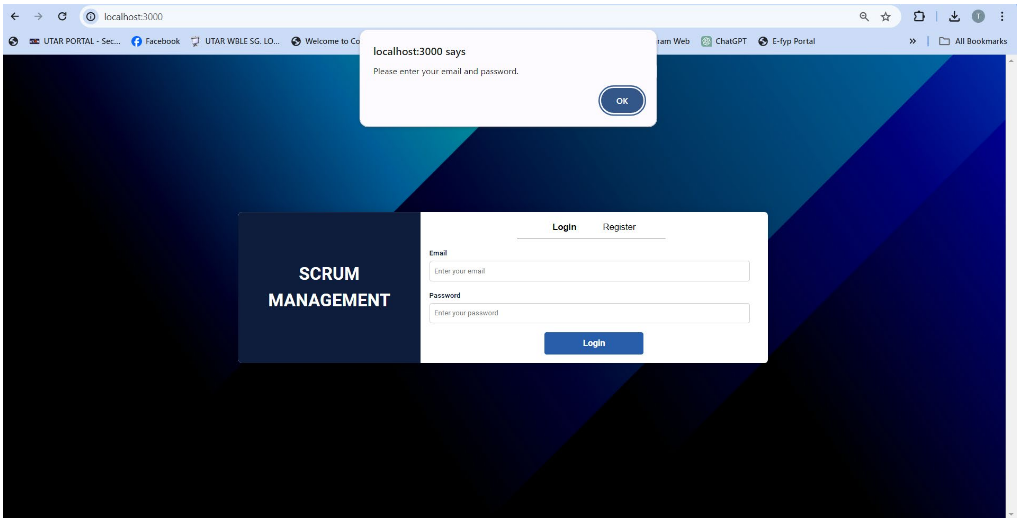
Figure 6.36: Login Function (Vue.js)



```php
public function login($request)
    {
        $credentials = $request->only('email', 'password');

        if (Auth::attempt($credentials)) {
            $user = Auth::user();

            // Define the custom claims to be included in the token payload
            $customClaims = ['userID' => $user->id];

            // Generate a JWT token with custom payload
            $token = JWTAuth::claims($customClaims)->fromUser($user);

            //return user details and token
            return response()->json(['user' => $user, 'token' => $token], 200);
        } else {
            return response()->json(['error' => 'Unauthorized'], 401);
        }
    }
```

Figure 6.37: Login Function (Laravel)

If the credential matches with the database of the web application, a JWT Token will be generated for the user to store in the frontend.



localhost:3000 says

Please enter your email and password.

Figure 6.38: Empty Email or Password Field Validation

If any of the user input has triggered the validation, an alert message will be shown to user. The following scenario has been validated:

1. Empty email or password field
2. Invalid email format

3. Mismatch of credentials

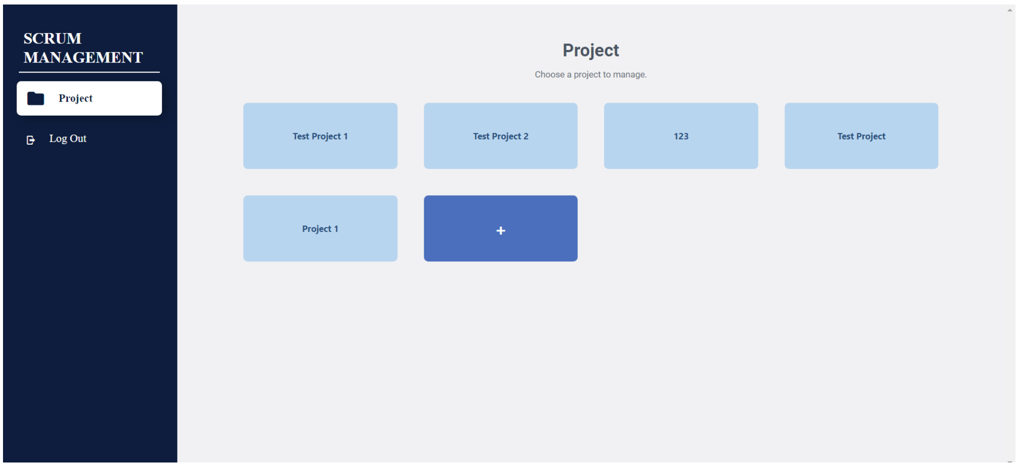
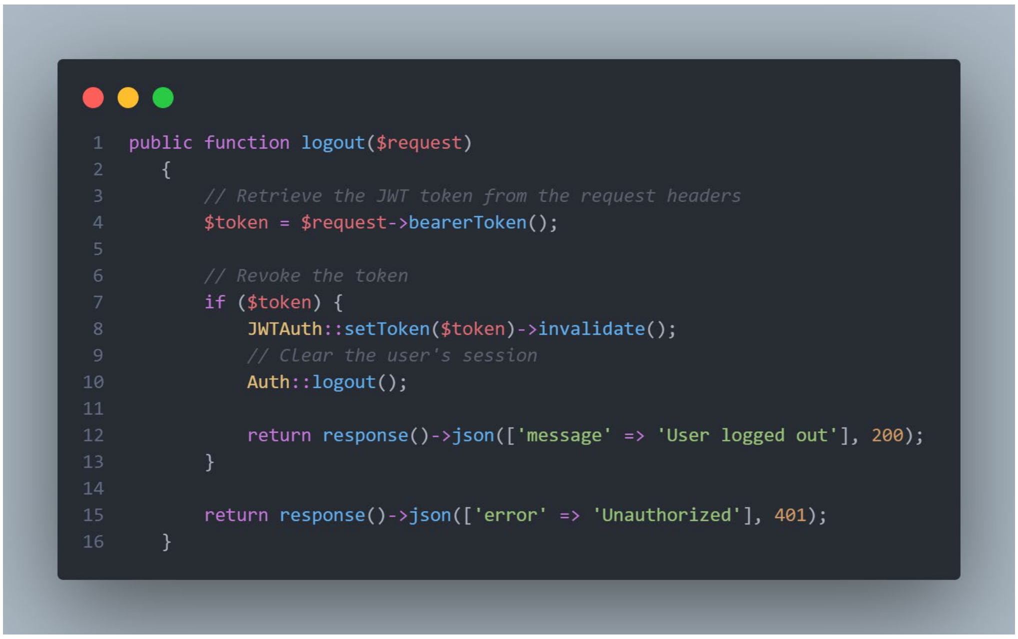


Figure 6.39: Project Page

When successfully login, user will be redirected to the project page.



```php
public function logout($request)
    {
        // Retrieve the JWT token from the request headers
        $token = $request->bearerToken();

        // Revoke the token
        if ($token) {
            JWTAuth::setToken($token)->invalidate();
            // Clear the user's session
            Auth::logout();

            return response()->json(['message' => 'User logged out'], 200);
        }

        return response()->json(['error' => 'Unauthorized'], 401);
    }
```

Figure 6.40: Logout Function (Laravel)

If clicks on the log out button, user will be redirected to login page. On the backend of the web application, the user JWT Token will be invalidated, so that the web application ensures that only authorized user has token access.

6.5.2 Register

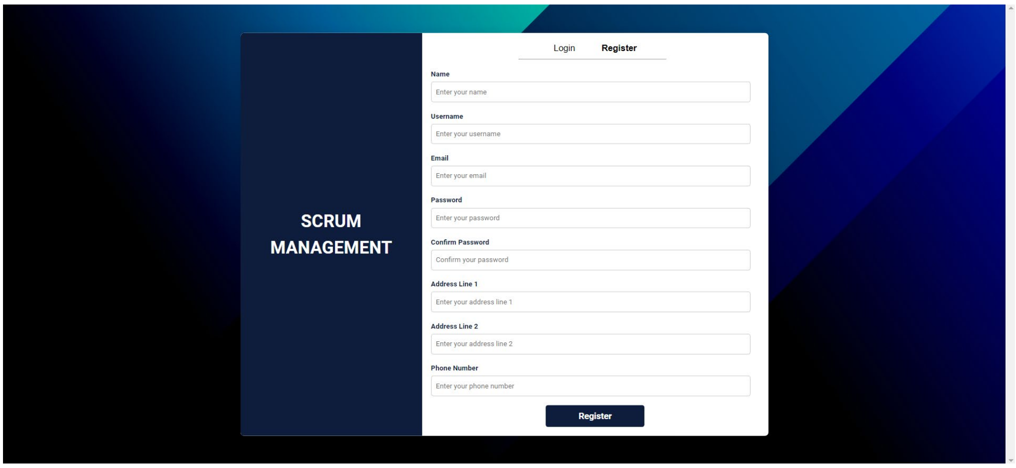
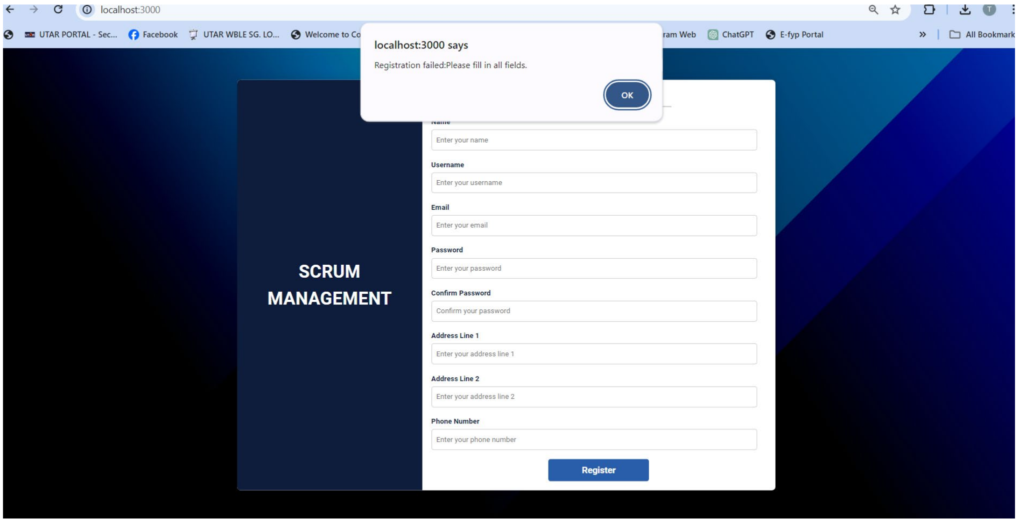


Figure 6.41: Register Page

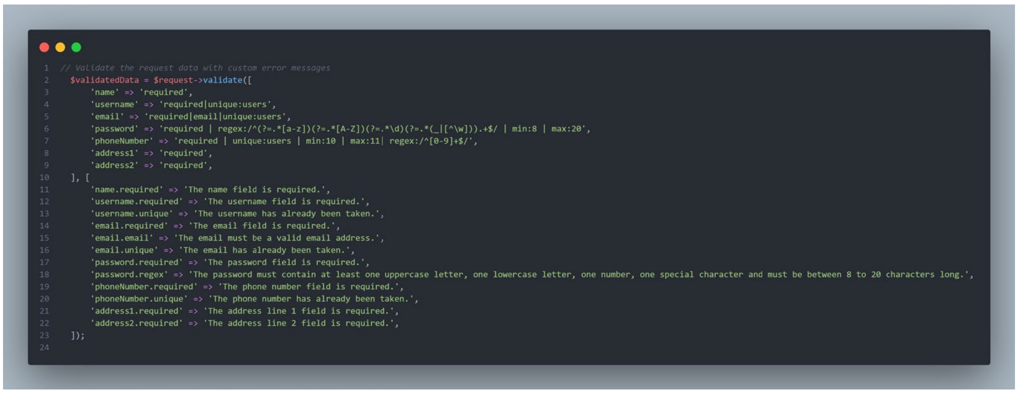
User can register an account on the registration page. The web application will prompt the user to enter all necessary details.



localhost:3000 says

Registration failed:Please fill in all fields.

Figure 6.42: Empty Field Validation



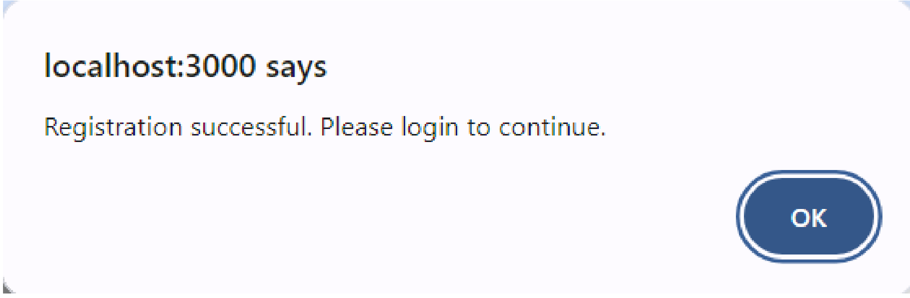
```php
// Validate the request data with custom error messages
  $validatedData = $request->validate([
      'name' => 'required',
      'username' => 'required|unique:users',
      'email' => 'required|email|unique:users',
      'password' => 'required | regex:/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*(_|[^\w])).+$/ | min:8 | max:20',
      'phoneNumber' => 'required | unique:users | min:10 | max:11| regex:/^[0-9]+$/',
      'address1' => 'required',
      'address2' => 'required',
  ], [
      'name.required' => 'The name field is required.',
      'username.required' => 'The username field is required.',
      'username.unique' => 'The username has already been taken.',
      'email.required' => 'The email field is required.',
      'email.email' => 'The email must be a valid email address.',
      'email.unique' => 'The email has already been taken.',
      'password.required' => 'The password field is required.',
      'password.regex' => 'The password must contain at least one uppercase letter, one lowercase letter, one number, one special character and must be between 8 to 20 characters long.',
      'phoneNumber.required' => 'The phone number field is required.',
      'phoneNumber.unique' => 'The phone number has already been taken.',
      'address1.required' => 'The address line 1 field is required.',
      'address2.required' => 'The address line 2 field is required.',
  ]);
```

Figure 6.43: Validation of Registration (Laravel)

If any of the user input has triggered the validation, an alert message will be shown to user. The following scenario has been validated:

1. Empty fields
2. Invalid email format
3. Invalid phone number format (Malaysia Phone Number Format)
4. Invalid password format (at least 1 uppercase, 1 lowercase, 1 number and 1 special character)
5. Email already registered
6. Phone number already registered
7. Username already registered
8. Confirm password does not match with password entered

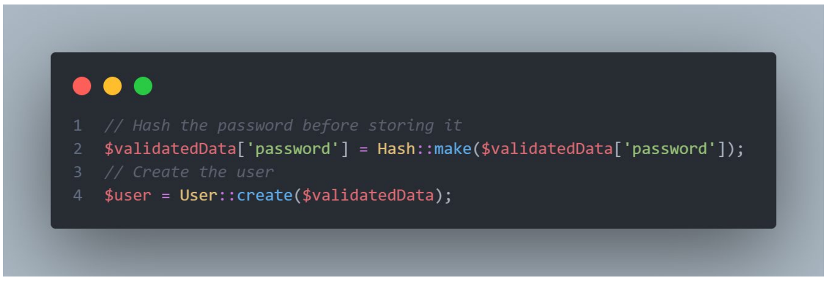
The validation of password formats aligns with the Non-Functional Requirements identified.



localhost:3000 says

Registration successful. Please login to continue.

Figure 6.44: Successful Registration Alert Message



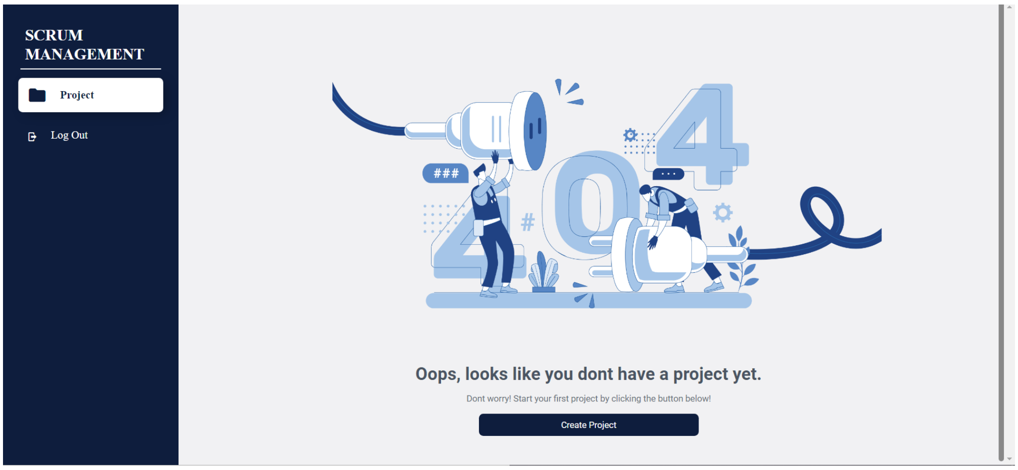
```
// Hash the password before storing it
$validatedData['password'] = Hash::make($validatedData['password']);
// Create the user
$user = User::create($validatedData);
```

Figure 6.45: Hashing of Password (Laravel)

Upon successful registration, the web application would add the registered user details into the database, with the password being hashed. An alert message will also be shown to user to remind them that they have successfully registered an account. The user will be redirected to the login page.

6.5.3 Project Management

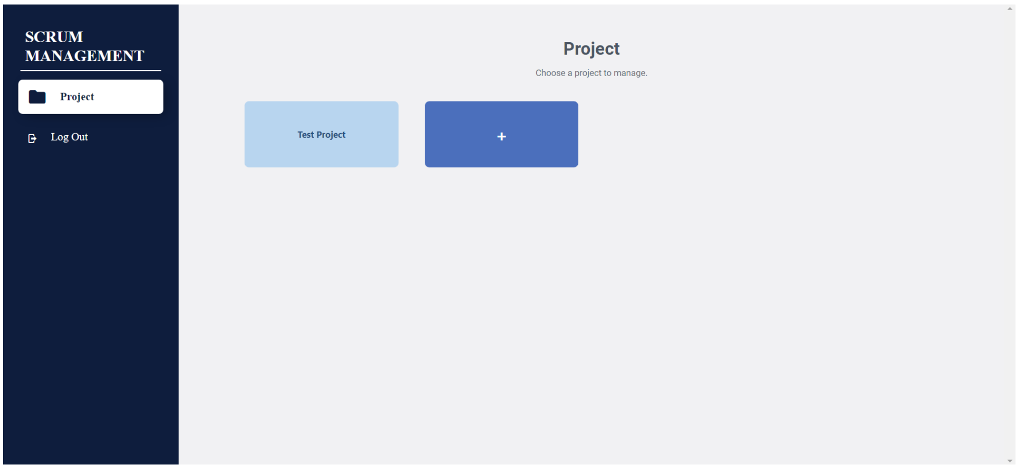
6.5.3.1 Creation and Selection of Project



Oops, looks like you dont have a project yet.

Dont worry! Start your first project by clicking the button below!

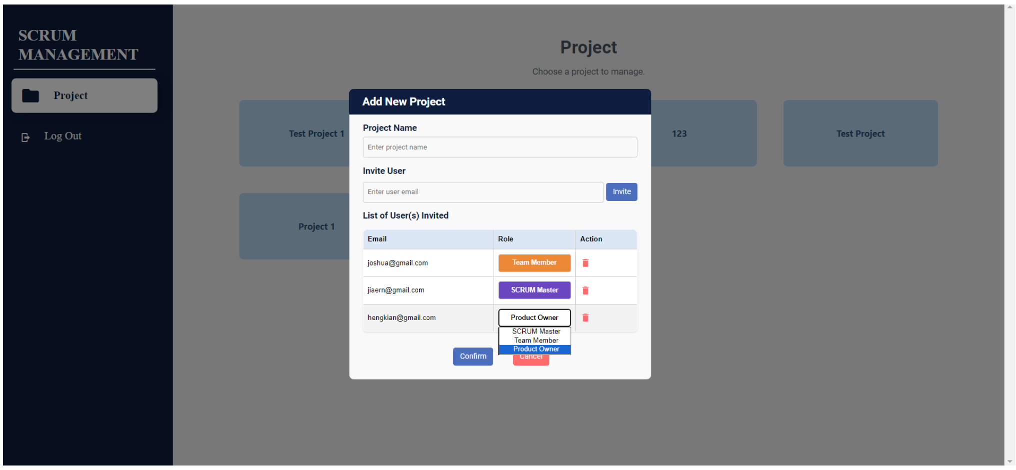
Figure 6.46: Project Page (No Project Found)



Project

Choose a project to manage.

Figure 6.47: Project Page



Project

Choose a project to manage.

Add New Project

Project Name

Enter project name

Invite User

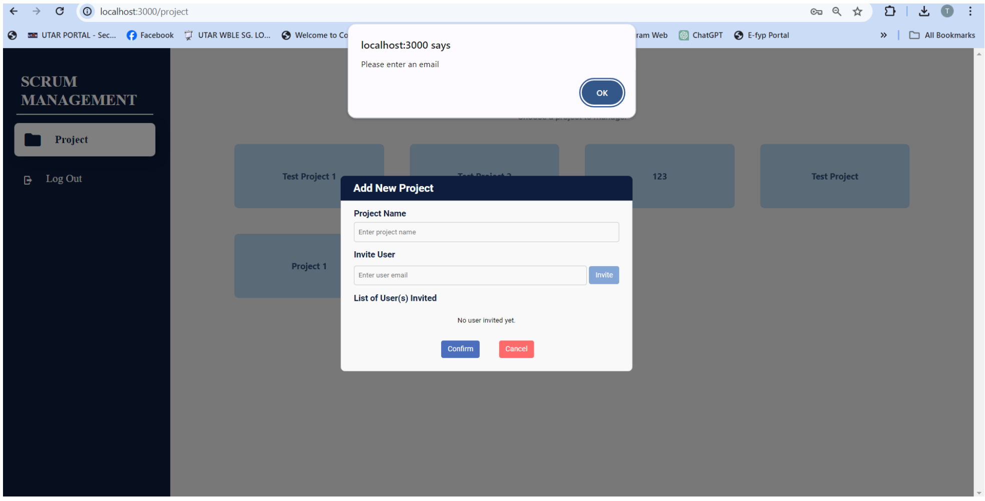
Enter user email

List of User(s) Invited

Email	Role	Action
joshua@gmail.com	Team Member	🗑
jiaern@gmail.com	SCRUM Master	🗑
hengkian@gmail.com	Product Owner SCRUM Master Team Member Product Owner	🗑

Figure 6.48: Create New Project Modal

The project page will display all the projects that the user is involved in. If no projects are found, the web application will show a "Not Found" page and prompt the user to create a new project. In both scenarios, clicking the "Create Project" button or the "+" button will trigger the same pop-up window to add a new project. Users can invite project members into the project and manage their roles before creating the project. The list of invited project members is displayed as shown in the figure above, with users able to adjust the roles between Team Member, Product Owner, and SCRUM Master. However, the inviting of project members is optional.


localhost:3000 says

Please enter an email

Add New Project

Project Name

Invite User

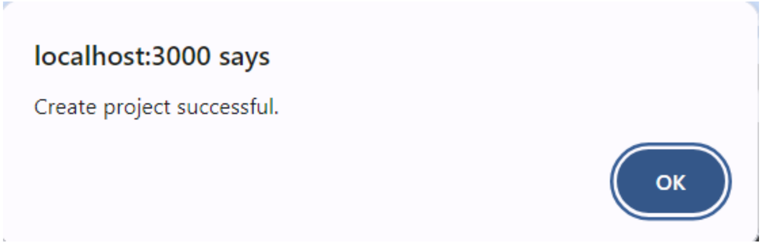
List of User(s) Invited

No user invited yet.

Figure 6.49: Empty Invited User Field Validation

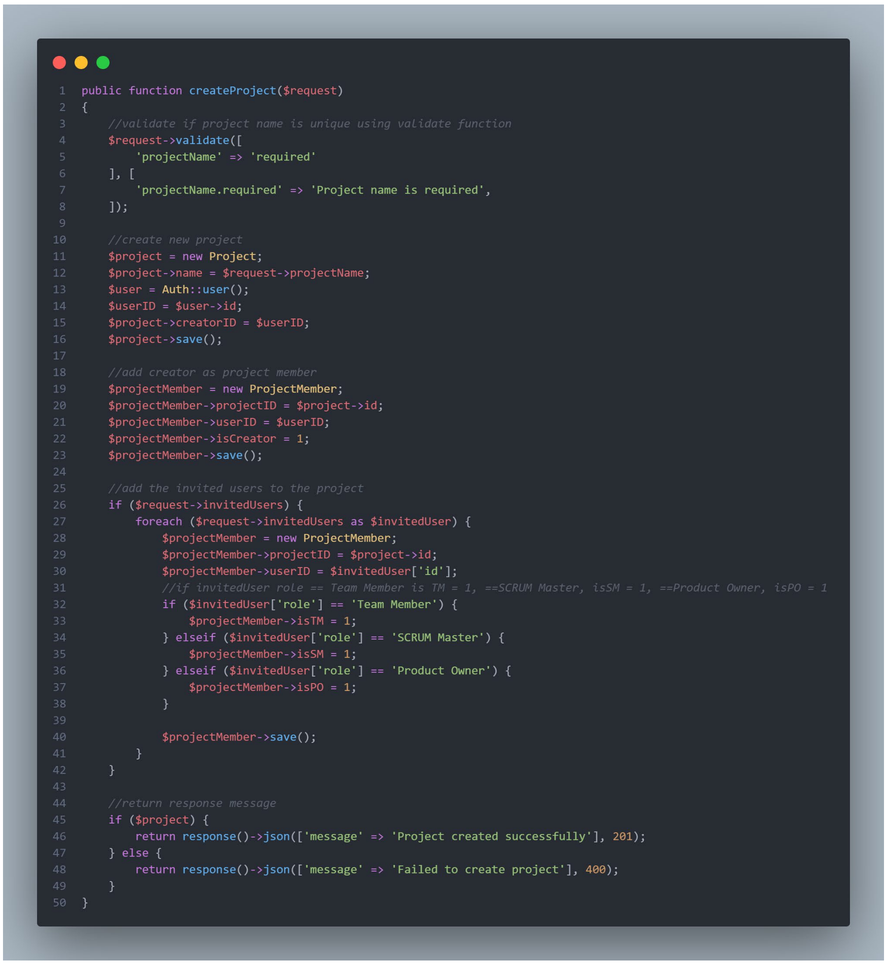
If any of the user input has triggered the validation, an alert message will be shown to user. The following scenario has been validated:

1. Empty project name
2. Empty invited user field
3. Invalid invited user email
4. User is not a registered user
5. User tries to invite himself


localhost:3000 says

Create project successful.

Figure 6.50: Successful Create Project Alert Message



```php
public function createProject($request)
{
    //validate if project name is unique using validate function
    $request->validate([
        'projectName' => 'required'
    ], [
        'projectName.required' => 'Project name is required',
    ]);

    //create new project
    $project = new Project;
    $project->name = $request->projectName;
    $user = Auth::user();
    $userID = $user->id;
    $project->creatorID = $userID;
    $project->save();

    //add creator as project member
    $projectMember = new ProjectMember;
    $projectMember->projectID = $project->id;
    $projectMember->userID = $userID;
    $projectMember->isCreator = 1;
    $projectMember->save();

    //add the invited users to the project
    if ($request->invitedUsers) {
        foreach ($request->invitedUsers as $invitedUser) {
            $projectMember = new ProjectMember;
            $projectMember->projectID = $project->id;
            $projectMember->userID = $invitedUser['id'];
            //if invitedUser role == Team Member is TM = 1, ==SCRUM Master, isSM = 1, ==Product Owner, isPO = 1
            if ($invitedUser['role'] == 'Team Member') {
                $projectMember->isTM = 1;
            } elseif ($invitedUser['role'] == 'SCRUM Master') {
                $projectMember->isSM = 1;
            } elseif ($invitedUser['role'] == 'Product Owner') {
                $projectMember->isPO = 1;
            }

            $projectMember->save();
        }
    }

    //return response message
    if ($project) {
        return response()->json(['message' => 'Project created successfully'], 201);
    } else {
        return response()->json(['message' => 'Failed to create project'], 400);
    }
}
```

Figure 6.51: Create Project Function (Laravel)

Upon successful creation of project, an alert message will also be shown to user to remind them that they have successfully created a project. In the Laravel backend, it will create a new project record and add all invited users into the project.

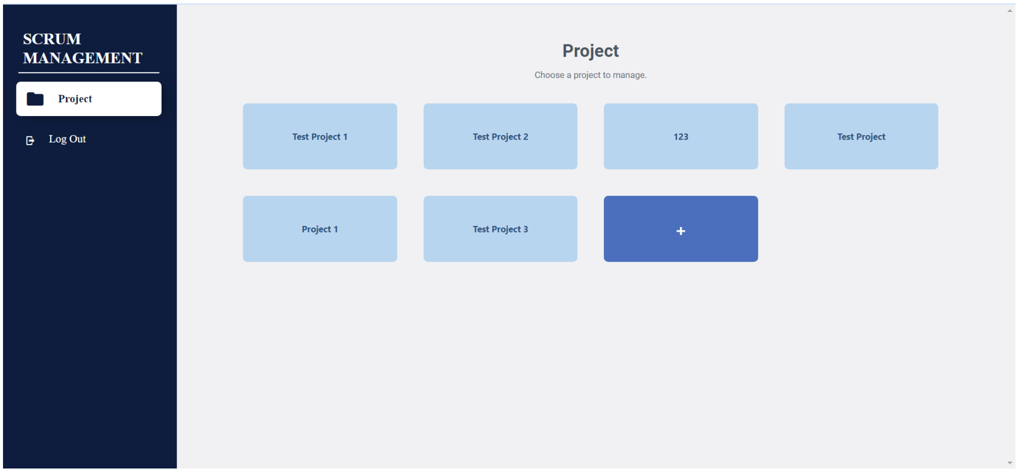
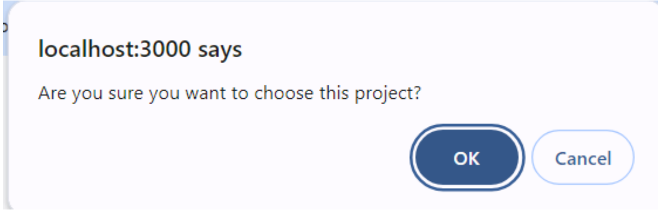


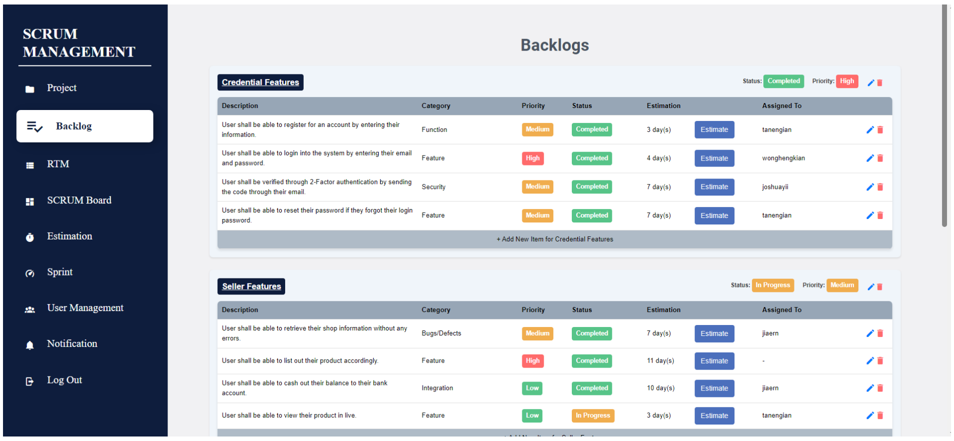
Figure 6.52: Project Page with New Project Created



localhost:3000 says

Are you sure you want to choose this project?

Figure 6.53: Confirmation of Selecting Project



Backlogs

Credential Features — Status: Completed Priority: High

Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to register for an account by entering their information.	Function	Medium	Completed	3 day(s)	Estimate	tanengian
User shall be able to login into the system by entering their email and password.	Feature	High	Completed	4 day(s)	Estimate	wonghengkian
User shall be verified through 2-Factor authentication by sending the code through their email.	Security	Medium	Completed	7 day(s)	Estimate	joshuayii
User shall be able to reset their password if they forgot their login password.	Feature	Medium	Completed	7 day(s)	Estimate	tanengian

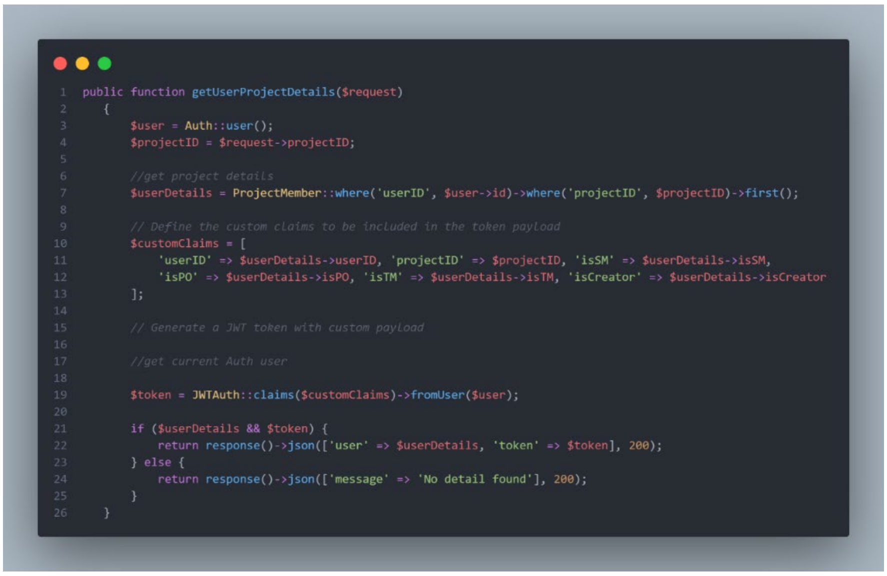
+ Add New Item for Credential Features

Seller Features — Status: In Progress Priority: Medium

Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	7 day(s)	Estimate	jiaern
User shall be able to list out their product accordingly.	Feature	High	Completed	11 day(s)	Estimate	-
User shall be able to cash out their balance to their bank account.	Integration	Low	Completed	10 day(s)	Estimate	jiaern
User shall be able to view their product in live.	Feature	Low	In Progress	3 day(s)	Estimate	tanengian

Figure 6.54: Backlog Page

After creating a new project, the web application will reload to display the newly created project. Users can select a project to manage by clicking on any of the projects. The web application will then prompt the user to confirm their selection. If the user confirms the selection, they will be redirected to the backlog page.

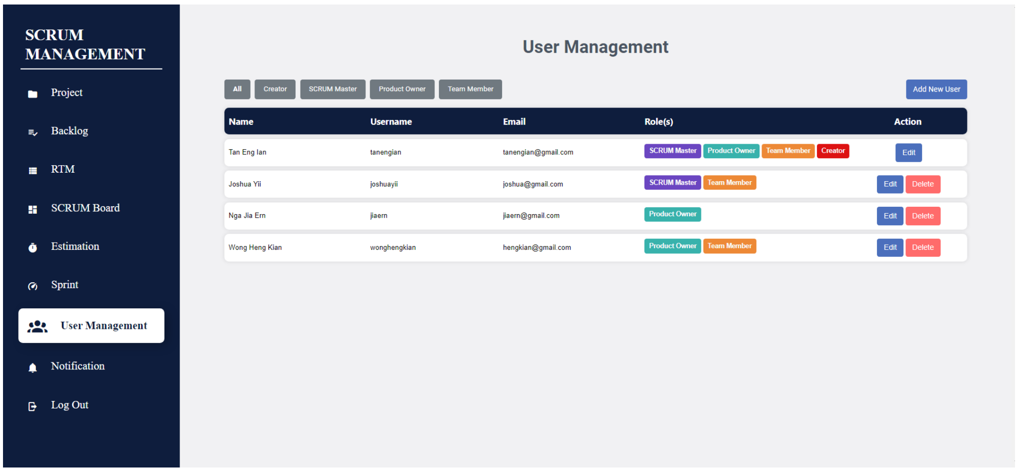


```
public function getUserProjectDetails($request)
    {
        $user = Auth::user();
        $projectID = $request->projectID;

        //get project details
        $userDetails = ProjectMember::where('userID', $user->id)->where('projectID', $projectID)->first();

        // Define the custom claims to be included in the token payload
        $customClaims = [
            'userID' => $userDetails->userID, 'projectID' => $projectID, 'isSM' => $userDetails->isSM,
            'isPO' => $userDetails->isPO, 'isTM' => $userDetails->isTM, 'isCreator' => $userDetails->isCreator
        ];

        // Generate a JWT token with custom payload

        //get current Auth user

        $token = JWTAuth::claims($customClaims)->fromUser($user);

        if ($userDetails && $token) {
            return response()->json(['user' => $userDetails, 'token' => $token], 200);
        } else {
            return response()->json(['message' => 'No detail found'], 200);
        }
    }
```

Figure 6.55: Select Project Function

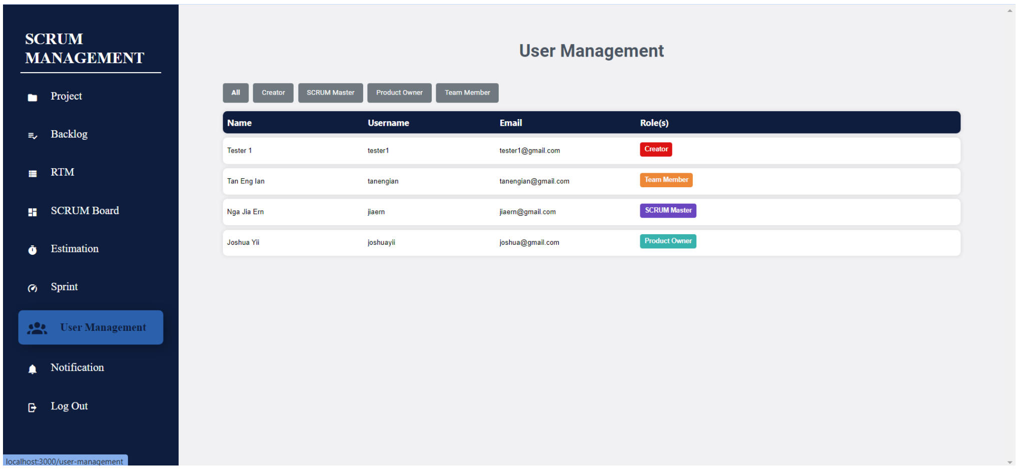
After successfully selecting a project to manage, the web application will generate a new token with custom payload information, such as the user's role, project ID, and user ID. This token will be sent back to the frontend to store securely. The token is then used to verify the authorization of the user accessing an API, ensuring security and proper access control. The navigation bar upon selecting a project to manage would be different, allowing the access to more pages.

6.5.3.2 User Management



Name	Username	Email	Role(s)	Action
Tan Eng Ian	tanengian	tanengian@gmail.com	SCRUM Master Product Owner Team Member Creator	Edit
Joshua Yii	joshuayii	joshua@gmail.com	SCRUM Master Team Member	Edit Delete
Nga Jia Ern	jiaern	jiaern@gmail.com	Product Owner	Edit Delete
Wong Heng Kian	wonghengkian	hengkian@gmail.com	Product Owner Team Member	Edit Delete

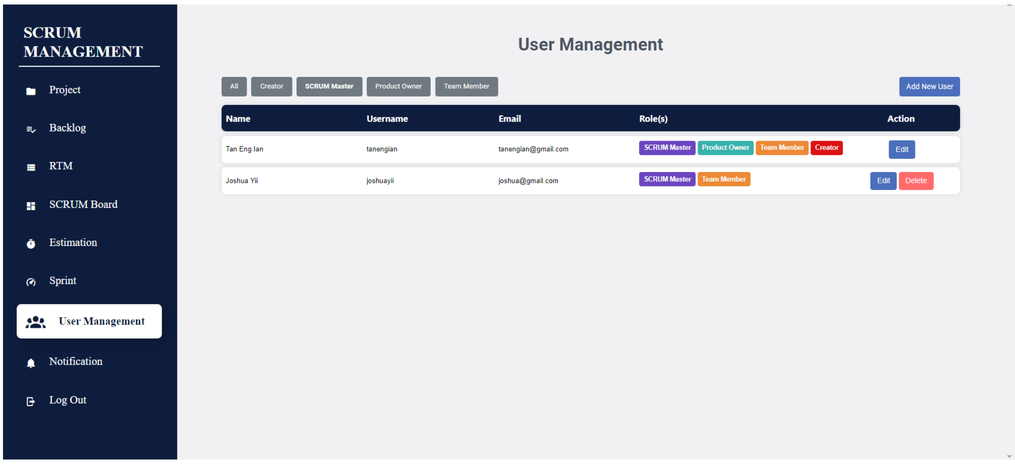
Figure 6.56: User Management Page



Name	Username	Email	Role(s)
Tester 1	tester1	tester1@gmail.com	Creator
Tan Eng Ian	tanengian	tanengian@gmail.com	Team Member
Nga Jia Ern	jiaern	jiaern@gmail.com	SCRUM Master
Joshua Yii	joshuayii	joshua@gmail.com	Product Owner

Figure 6.57: User Management Page (Non-Creator View)

In the user management page, user will be shown with a list of users involved in the project. The creator of the project is allowed to add or remove project members or invite new project members into the project. It is important to note that if user is not a creator, the modification buttons will not be shown. The creator is also not allowed to remove themselves from the project.

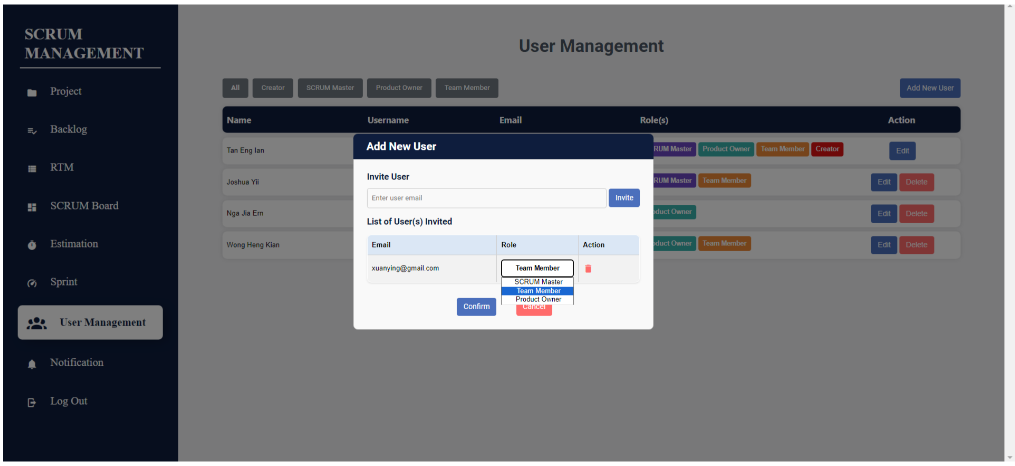


User Management

Name	Username	Email	Role(s)	Action
Tan Eng Ian	tanengian	tanengian@gmail.com	SCRUM Master Product Owner Team Member Creator	Edit
Joshua Yii	joshuayii	joshua@gmail.com	SCRUM Master Team Member	Edit Delete

Figure 6.58: SCRUM Master Filtering

User is also allowed to filter the page by roles.



User Management

Add New User

Invite User

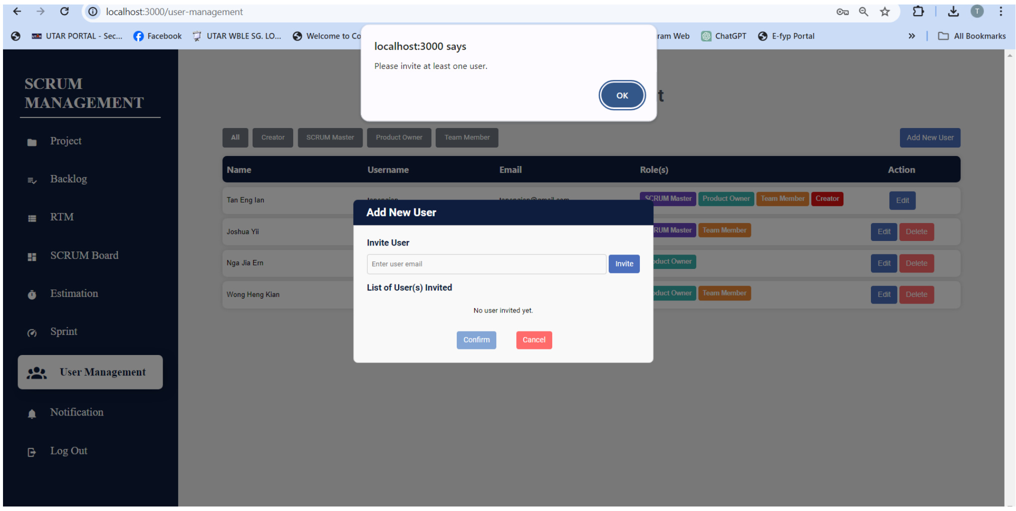
Enter user email

List of User(s) Invited

Email	Role	Action
xuanying@gmail.com	Team Member	

Figure 6.59: Inviting New Project Member

Creator can invite project members into the project and manage their roles before confirmation of invitation. The list of invited project members is displayed as shown in the figure above, with creator being able to adjust the roles between Team Member, Product Owner, and SCRUM Master.



localhost:3000 says

Please invite at least one user.

Add New User

Invite User

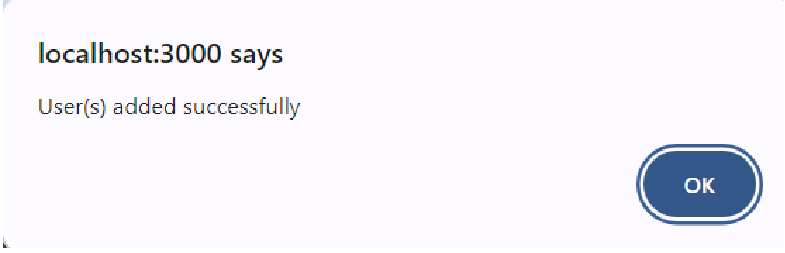
List of User(s) Invited

No user invited yet.

Figure 6.60: Empty User List Validation

If any of the user input has triggered the validation, an alert message will be shown to user. The following scenario has been validated:

1. No project member invited but press on “Confirm” button
2. Empty invited user field
3. Invalid invited user email
4. User is not a registered user
5. User tries to invite himself
6. User already invited to the project

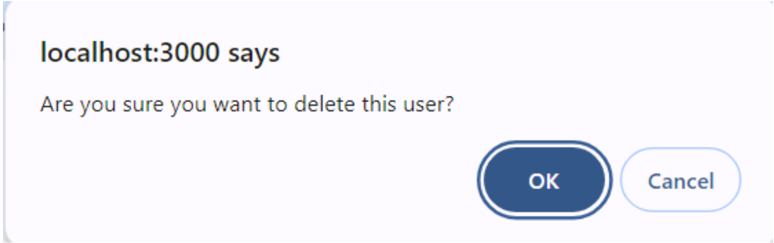


localhost:3000 says

User(s) added successfully

Figure 6.61: Successful Invite User to Project Alert Message

Upon successfully inviting new project members into the project, an alert message will be displayed to remind user that the new user has been invited.

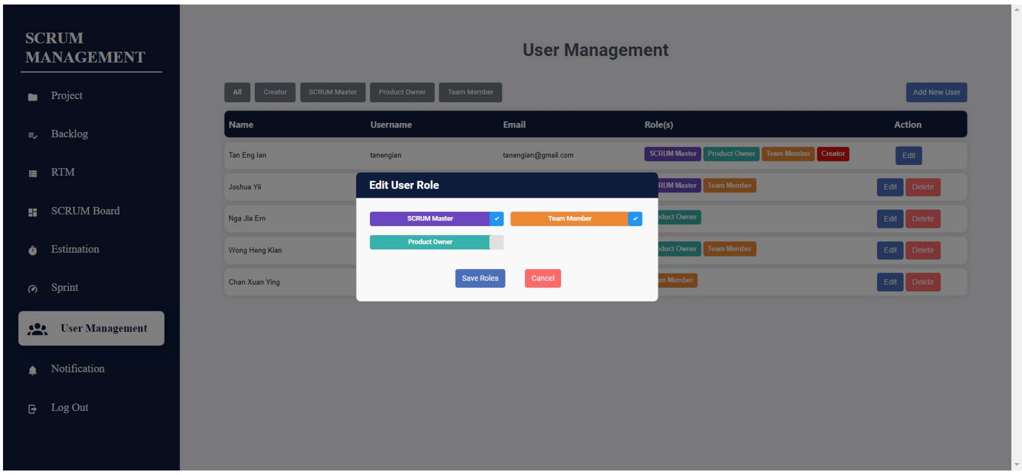


localhost:3000 says

Are you sure you want to delete this user?

Figure 6.62: Confirmation of Removing Project Member

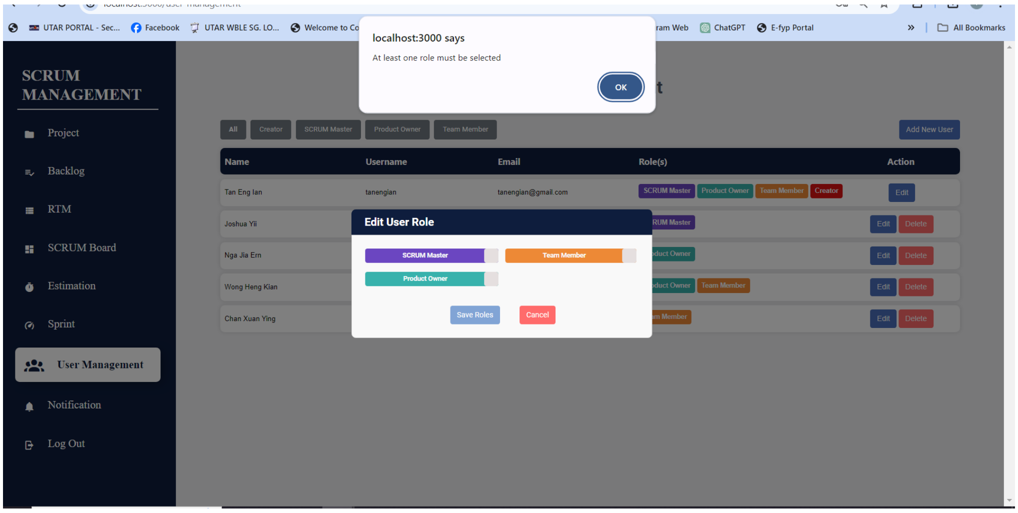
User can also remove the project member by clicking on the “Delete” button. The web application will prompt the creator to confirm the deletion. Upon confirmation, the user selected will be removed from the project.



Name	Username	Email	Role(s)	Action
Tan Eng Ian	tanengian	tanengian@gmail.com	SCRUM Master Product Owner Team Member Creator	Edit
Joshua Yii			RUM Master Team Member	Edit Delete
Nga Jia Ern			duct Owner	Edit Delete
Wong Heng Kian			duct Owner Team Member	Edit Delete
Chan Xuan Ying			m Member	Edit Delete

Figure 6.63: Edit Project Member Role

User is allowed to edit the project member role, in which 1 or more roles can be assigned to a project member.



localhost:3000 says

At least one role must be selected

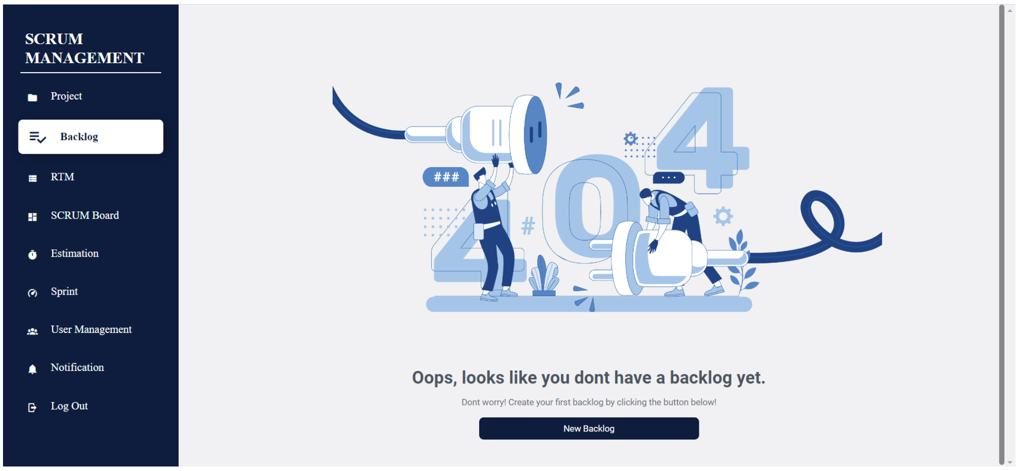
Name	Username	Email	Role(s)	Action
Tan Eng Ian	tanengian	tanengian@gmail.com	SCRUM Master Product Owner Team Member Creator	Edit
Joshua Yii			RUM Master	Edit Delete
Nga Jia Ern			duct Owner	Edit Delete
Wong Heng Kian			duct Owner Team Member	Edit Delete
Chan Xuan Ying			m Member	Edit Delete

Figure 6.64: Empty Project Member Role Validation

However, a minimum of 1 project role must be assigned to a project member.

6.5.4 Backlog

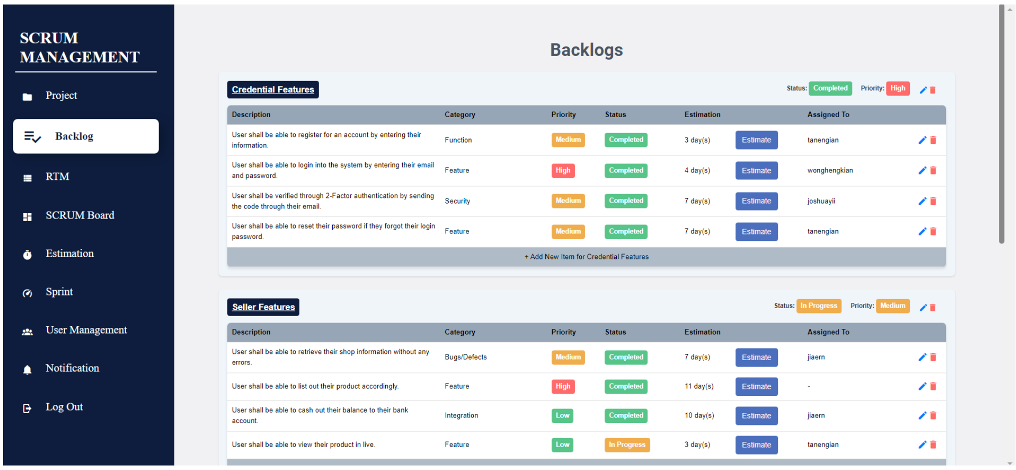
6.5.4.1 Product Backlog



Oops, looks like you dont have a backlog yet.

Dont worry! Create your first backlog by clicking the button below!

Figure 6.65: Backlog Page (No Backlog)



Backlogs

Credential Features — Status: Completed Priority: High

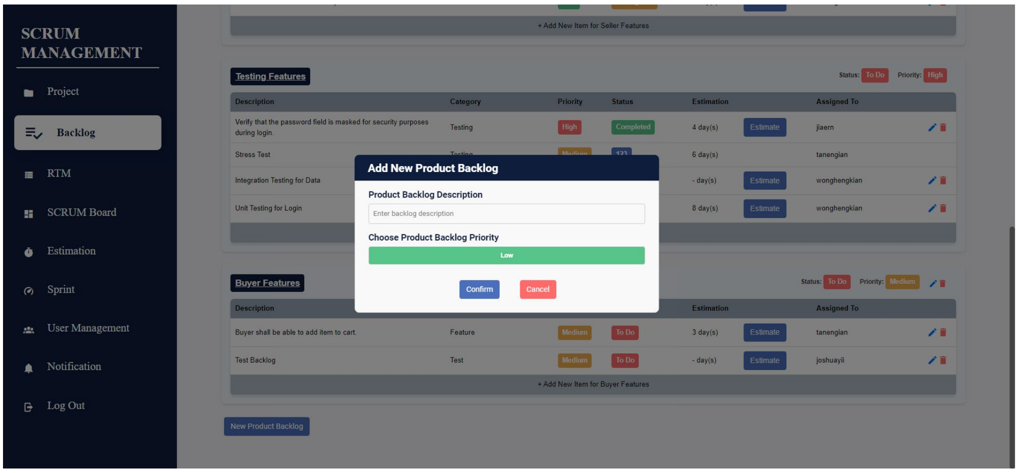
Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to register for an account by entering their information.	Function	Medium	Completed	3 day(s)	Estimate	tanengian
User shall be able to login into the system by entering their email and password.	Feature	High	Completed	4 day(s)	Estimate	wonghengkian
User shall be verified through 2-Factor authentication by sending the code through their email.	Security	Medium	Completed	7 day(s)	Estimate	joshuayii
User shall be able to reset their password if they forgot their login password.	Feature	Medium	Completed	7 day(s)	Estimate	tanengian

+ Add New Item for Credential Features

Seller Features — Status: In Progress Priority: Medium

Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	7 day(s)	Estimate	jiaern
User shall be able to list out their product accordingly.	Feature	High	Completed	11 day(s)	Estimate	-
User shall be able to cash out their balance to their bank account.	Integration	Low	Completed	10 day(s)	Estimate	jiaern
User shall be able to view their product in live.	Feature	Low	In Progress	3 day(s)	Estimate	tanengian

Figure 6.66: Backlog Page



+ Add New Item for Seller Features

Testing Features — Status: To Do Priority: High

Description	Category	Priority	Status	Estimation		Assigned To
Verify that the password field is masked for security purposes during login.	Testing	High	Completed	4 day(s)	Estimate	jiaern
Stress Test	Testing	Medium	123	6 day(s)		tanengian
Integration Testing for Data				- day(s)	Estimate	wonghengkian
Unit Testing for Login				8 day(s)	Estimate	wonghengkian

Add New Product Backlog

Product Backlog Description: Enter backlog description

Choose Product Backlog Priority: Low

Confirm | Cancel

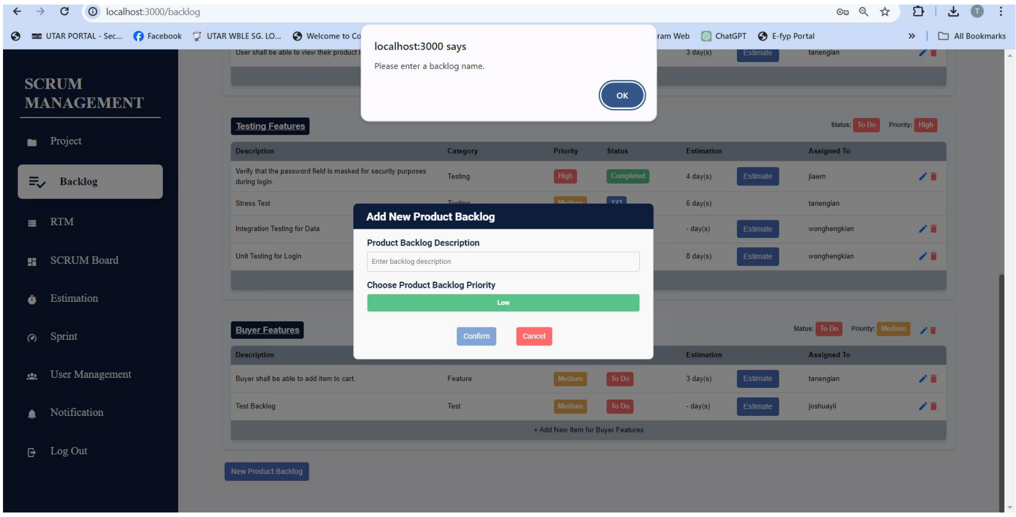
Buyer Features — Status: To Do Priority: Medium

Description	Category	Priority	Status	Estimation		Assigned To
Buyer shall be able to add item to cart.	Feature	Medium	To Do	3 day(s)	Estimate	tanengian
Test Backlog	Test	Medium	To Do	- day(s)	Estimate	joshuayii

+ Add New Item for Buyer Features

Figure 6.67: Add New Product Backlog

The backlog page will display all the backlogs has been created. If no backlogs are found, the web application will show a “Not Found” page and prompt the user to create a product backlog. In both scenarios, clicking the "New Backlog" button will trigger the same pop-up window to add a product backlog. It is important to note that only Product Onwer or creator can manage product backlog.



localhost:3000 says

Please enter a backlog name.

OK

User shall be able to view their product i | 3 day(s) | Estimate | tanengian

Testing Features — Status: To Do — Priority: High

Description	Category	Priority	Status	Estimation		Assigned To
Verify that the password field is masked for security purposes during login.	Testing	High	Completed	4 day(s)	Estimate	jiaern
Stress Test	Testing	Medium	123	6 day(s)		tanengian
Integration Testing for Data				- day(s)	Estimate	wonghengkian
Unit Testing for Login				8 day(s)	Estimate	wonghengkian

Add New Product Backlog

Product Backlog Description

Enter backlog description

Choose Product Backlog Priority

Low

Confirm Cancel

Buyer Features — Status: To Do — Priority: Medium

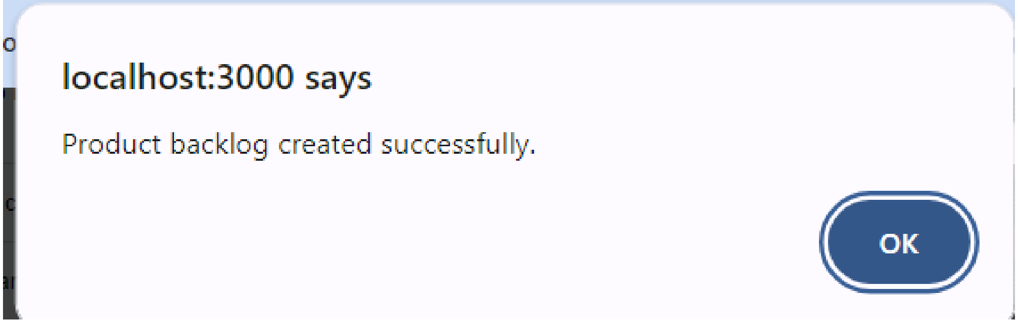
Description	Category	Priority	Status	Estimation		Assigned To
Buyer shall be able to add item to cart.	Feature	Medium	To Do	3 day(s)	Estimate	tanengian
Test Backlog	Test	Medium	To Do	- day(s)	Estimate	joshuayii

+ Add New Item for Buyer Features

New Product Backlog

Figure 6.68: Empty Product Backlog Description Validation

If the product backlog description entered is empty, the web application will remind the user with an alert message.



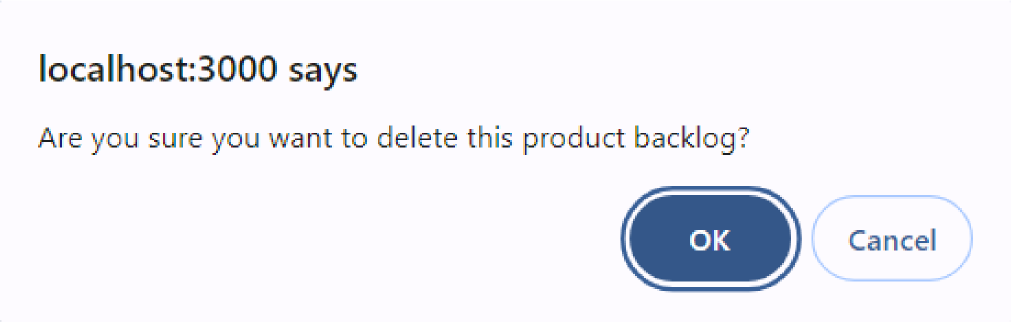
localhost:3000 says

Product backlog created successfully.

OK

Figure 6.69: Successful Create Product Backlog Alert Message

Upon successfully creating a new product backlog, the web application will remind the user. The default status of the new product backlog will be “To Do”.

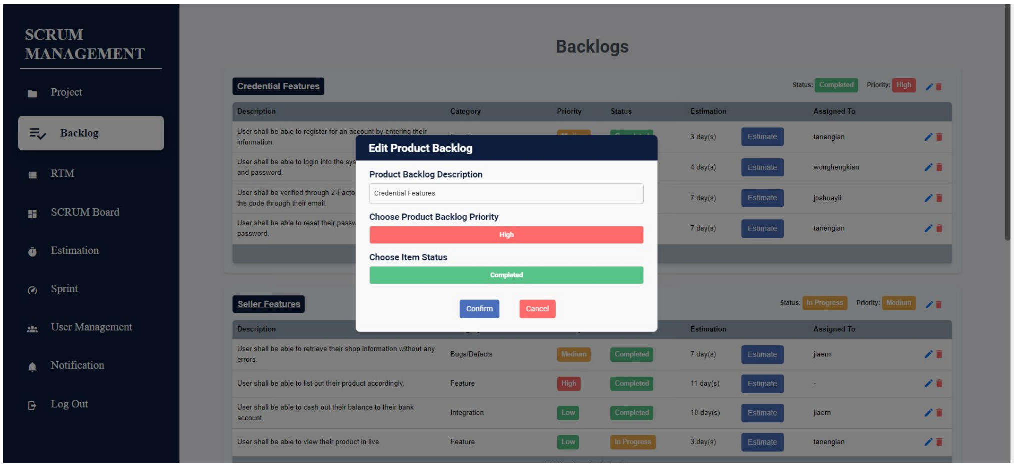


localhost:3000 says

Are you sure you want to delete this product backlog?

Figure 6.70: Confirmation of Delete Product Backlog

If the user clicks on the delete icon of a product backlog, the web application will prompt the user to confirm. Upon confirmation, the selected product backlog will be deleted along with its sprint backlog item (if any).



Credential Features

Status: Completed Priority: High

Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to register for an account by entering their information.				3 day(s)	Estimate	tanengian
User shall be able to login into the sys and password.				4 day(s)	Estimate	wonghengkian
User shall be verified through 2-Facto the code through their email.				7 day(s)	Estimate	joshuayii
User shall be able to reset their passw password.				7 day(s)	Estimate	tanengian

Edit Product Backlog

Product Backlog Description: Credential Features

Choose Product Backlog Priority: High

Choose Item Status: Completed

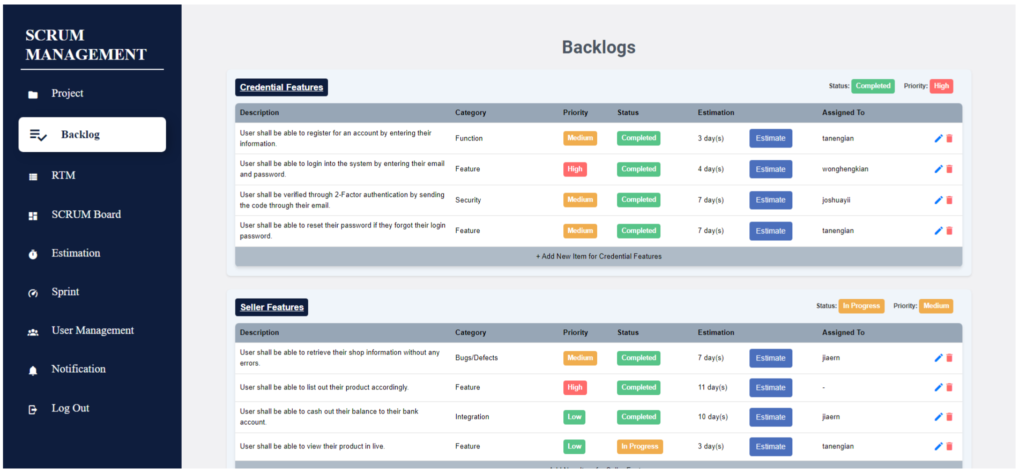
Seller Features

Status: In Progress Priority: Medium

Description				Estimation		Assigned To
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	7 day(s)	Estimate	jiaern
User shall be able to list out their product accordingly.	Feature	High	Completed	11 day(s)	Estimate	-
User shall be able to cash out their balance to their bank account.	Integration	Low	Completed	10 day(s)	Estimate	jiaern
User shall be able to view their product in live.	Feature	Low	In Progress	3 day(s)	Estimate	tanengian

Figure 6.71: Edit Product Backlog

If the user clicks on the edit icon of a product backlog, the web application will show a pop out of elements to be edited. Similar to add new product backlog, validation has been made if user enters an empty product backlog description.



Backlogs

Credential Features Status: Completed Priority: High

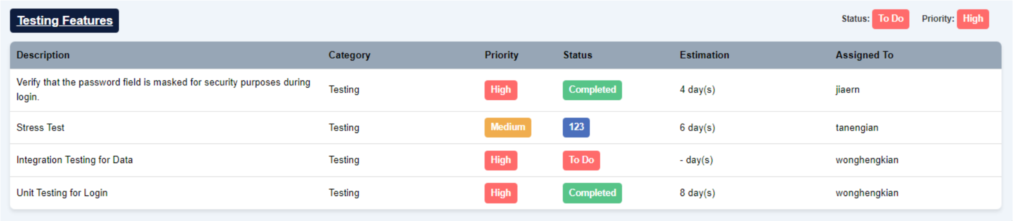
Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to register for an account by entering their information.	Function	Medium	Completed	3 day(s)	Estimate	tanengian
User shall be able to login into the system by entering their email and password.	Feature	High	Completed	4 day(s)	Estimate	wonghengkian
User shall be verified through 2-Factor authentication by sending the code through their email.	Security	Medium	Completed	7 day(s)	Estimate	joshuayii
User shall be able to reset their password if they forgot their login password.	Feature	Medium	Completed	7 day(s)	Estimate	tanengian

+ Add New Item for Credential Features

Seller Features Status: In Progress Priority: Medium

Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	7 day(s)	Estimate	jiaern
User shall be able to list out their product accordingly.	Feature	High	Completed	11 day(s)	Estimate	-
User shall be able to cash out their balance to their bank account.	Integration	Low	Completed	10 day(s)	Estimate	jiaern
User shall be able to view their product in live.	Feature	Low	In Progress	3 day(s)	Estimate	tanengian

Figure 6.72: Product Backlog (Non-Product Owner View)



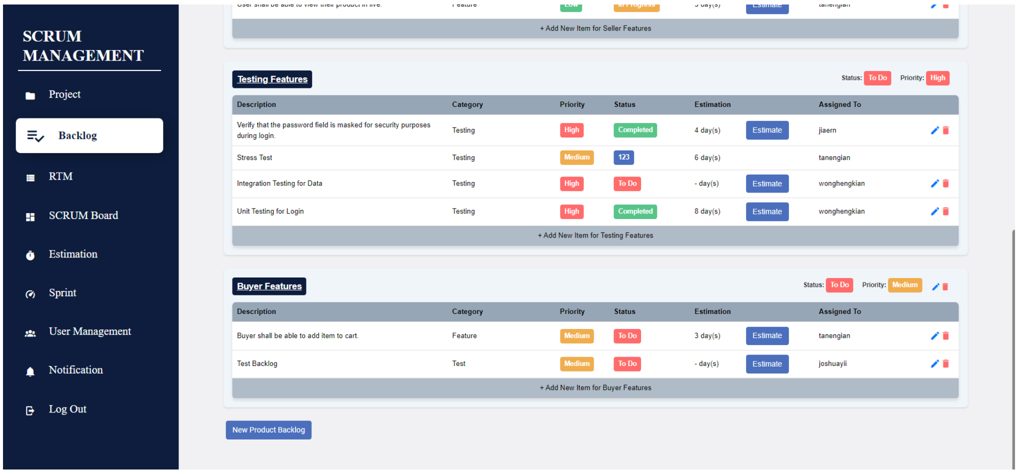
Testing Features Status: To Do Priority: High

Description	Category	Priority	Status	Estimation	Assigned To
Verify that the password field is masked for security purposes during login.	Testing	High	Completed	4 day(s)	jiaern
Stress Test	Testing	Medium	123	6 day(s)	tanengian
Integration Testing for Data	Testing	High	To Do	- day(s)	wonghengkian
Unit Testing for Login	Testing	High	Completed	8 day(s)	wonghengkian

Figure 6.73: Product Backlog (Sprint Backlog involves in Active Sprint)

If the user is not a Product Owner or a creator, the delete and edit icon will not be shown for the product backlog item. Besides that, if a sprint backlog item of a product backlog is currently involved in an active sprint, the product backlog item will also not be allowed to edit.

6.5.4.2 Sprint Backlog



+ Add New Item for Seller Features

Testing Features Status: To Do Priority: High

Description	Category	Priority	Status	Estimation		Assigned To
Verify that the password field is masked for security purposes during login.	Testing	High	Completed	4 day(s)	Estimate	jiaern
Stress Test	Testing	Medium	123	6 day(s)		tanengian
Integration Testing for Data	Testing	High	To Do	- day(s)	Estimate	wonghengkian
Unit Testing for Login	Testing	High	Completed	8 day(s)	Estimate	wonghengkian

+ Add New Item for Testing Features

Buyer Features Status: To Do Priority: Medium

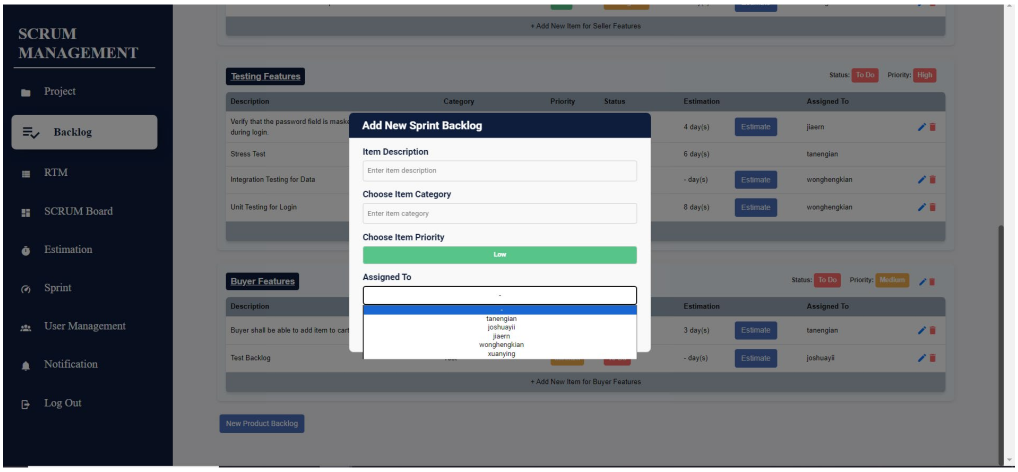
Description	Category	Priority	Status	Estimation		Assigned To
Buyer shall be able to add item to cart.	Feature	Medium	To Do	3 day(s)	Estimate	tanengian
Test Backlog	Test	Medium	To Do	- day(s)	Estimate	joshuayii

+ Add New Item for Buyer Features

New Product Backlog

Figure 6.74: Sprint Backlog View

The sprint backlog will be displayed according to their product backlog.



Add New Sprint Backlog

Item Description: Enter item description

Choose Item Category: Enter item category

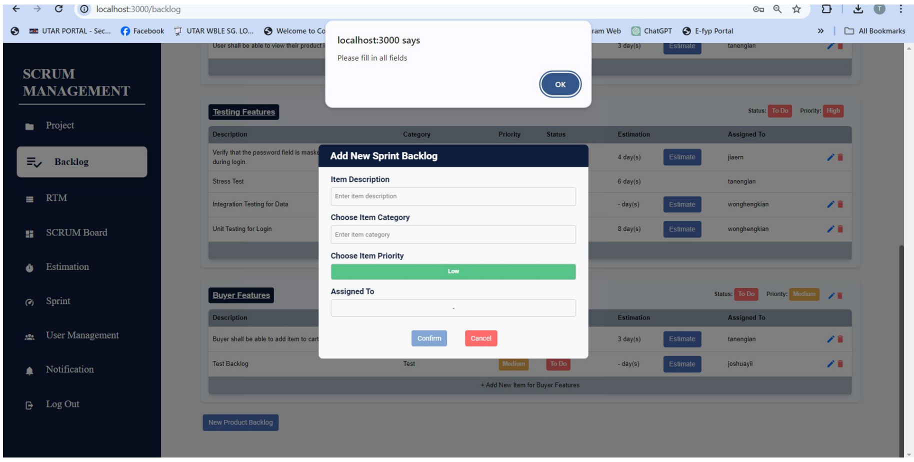
Choose Item Priority: Low

Assigned To: -

- -
- tanengian
- joshuayii
- jiaern
- wonghengkian
- xuanying

Figure 6.75: Add New Sprint Backlog

The web application will enable users to add new sprint backlog items into the product backlog. However, this functionality is only available to SCRUM Master, Team Member, and creator. When selecting the "Assigned To" option from the dropdown menu, users will see a list of project members who have been invited to the project.



localhost:3000 says

Please fill in all fields

OK

Add New Sprint Backlog

Item Description: Enter item description

Choose Item Category: Enter item category

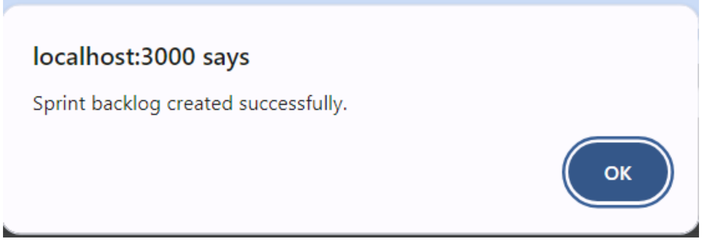
Choose Item Priority: Low

Assigned To: -

Confirm Cancel

Figure 6.76: Empty Field Validation

If any of the field entered is empty, the web application will remind the user that all field are required to be filled.



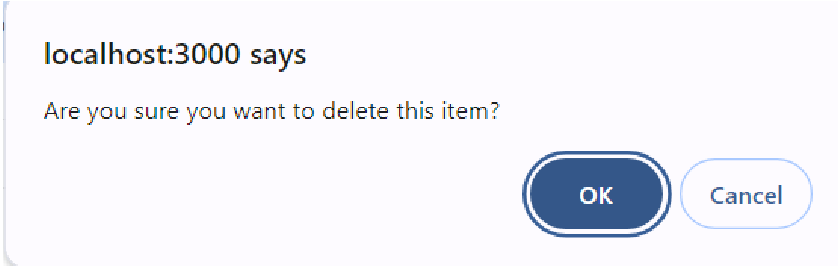
localhost:3000 says

Sprint backlog created successfully.

OK

Figure 6.77: Successful Create Sprint Backlog Alert Message

Upon successfully creating a new sprint backlog, the web application will remind the user. The default status of the new sprint backlog will be "To Do".

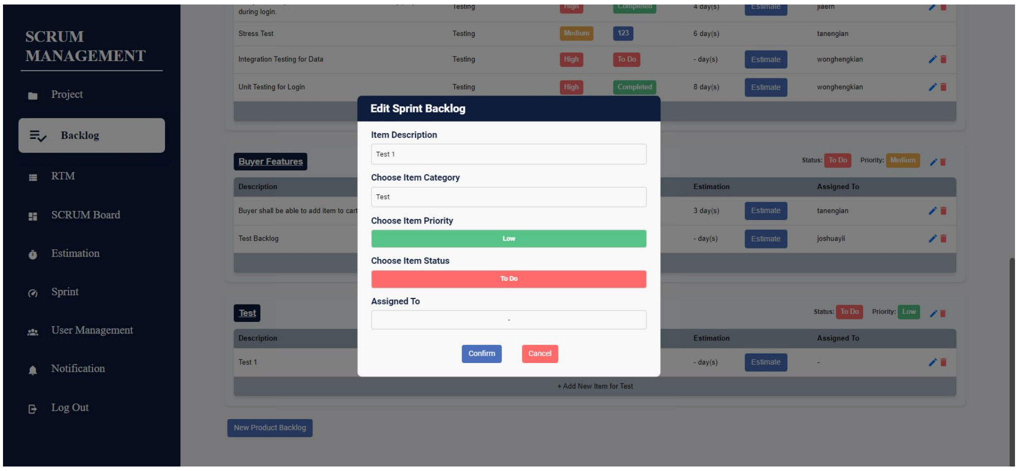


localhost:3000 says

Are you sure you want to delete this item?

Figure 6.78: Confirmation of Delete Sprint Backlog

If the user clicks on the delete icon of a sprint backlog, the web application will prompt the user to confirm. Upon confirmation, the selected sprint backlog item will be deleted.



Edit Sprint Backlog

Item Description: Test 1

Choose Item Category: Test

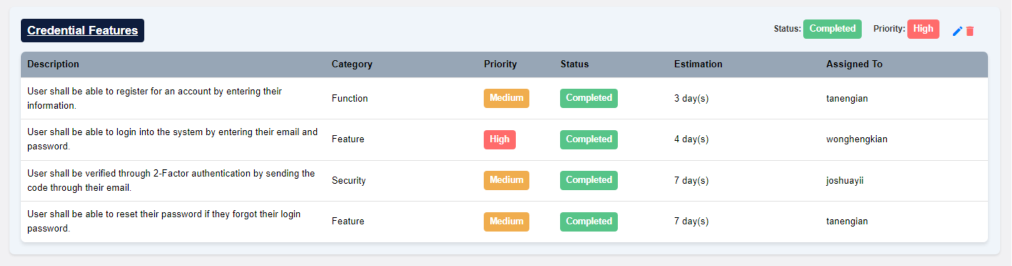
Choose Item Priority: Low

Choose Item Status: To Do

Assigned To: -

Figure 6.79: Edit Sprint Backlog

If the user clicks on the edit icon of a sprint backlog, the web application will show a pop out of elements to be edited. Similar to add new sprint backlog, validation has been made if user enters any empty field.

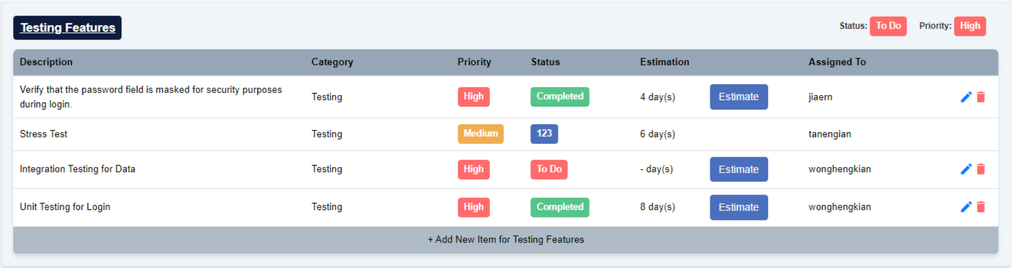


Credential Features

Status: Completed Priority: High

Description	Category	Priority	Status	Estimation	Assigned To
User shall be able to register for an account by entering their information.	Function	Medium	Completed	3 day(s)	tanengian
User shall be able to login into the system by entering their email and password.	Feature	High	Completed	4 day(s)	wonghengkian
User shall be verified through 2-Factor authentication by sending the code through their email.	Security	Medium	Completed	7 day(s)	joshuayii
User shall be able to reset their password if they forgot their login password.	Feature	Medium	Completed	7 day(s)	tanengian

Figure 6.80: Sprint Backlog (Product Owner View)



Testing Features

Status: To Do Priority: High

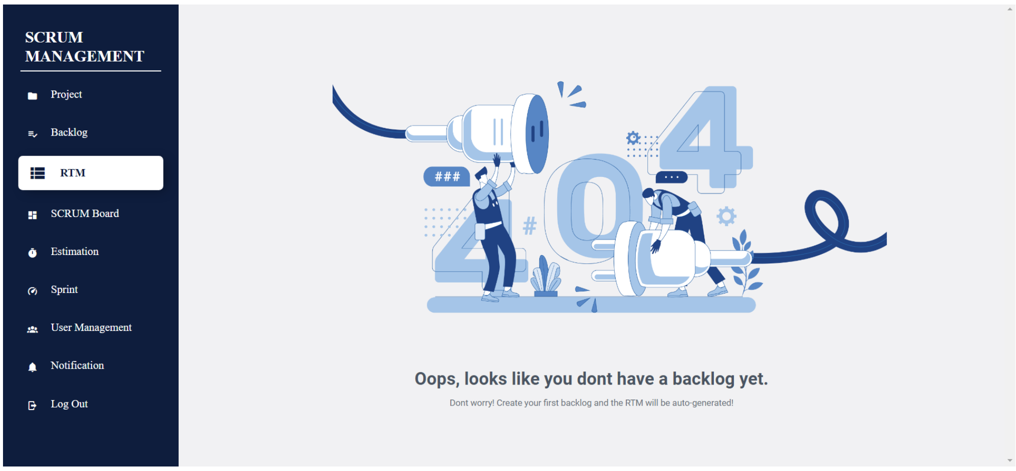
Description	Category	Priority	Status	Estimation		Assigned To	
Verify that the password field is masked for security purposes during login.	Testing	High	Completed	4 day(s)	Estimate	jiaern	✎ 🗑
Stress Test	Testing	Medium	123	6 day(s)		tanengian	
Integration Testing for Data	Testing	High	To Do	- day(s)	Estimate	wonghengkian	✎ 🗑
Unit Testing for Login	Testing	High	Completed	8 day(s)	Estimate	wonghengkian	✎ 🗑

+ Add New Item for Testing Features

Figure 6.81: Sprint Backlog (Sprint Backlog involves in Active Sprint)

If the user is not a SCRUM Master, Team Member or a creator, the delete and edit icon will not be shown for the sprint backlog item. Besides that, if a sprint backlog item is currently involved in an active sprint, the sprint backlog item will not be allowed to be edited.

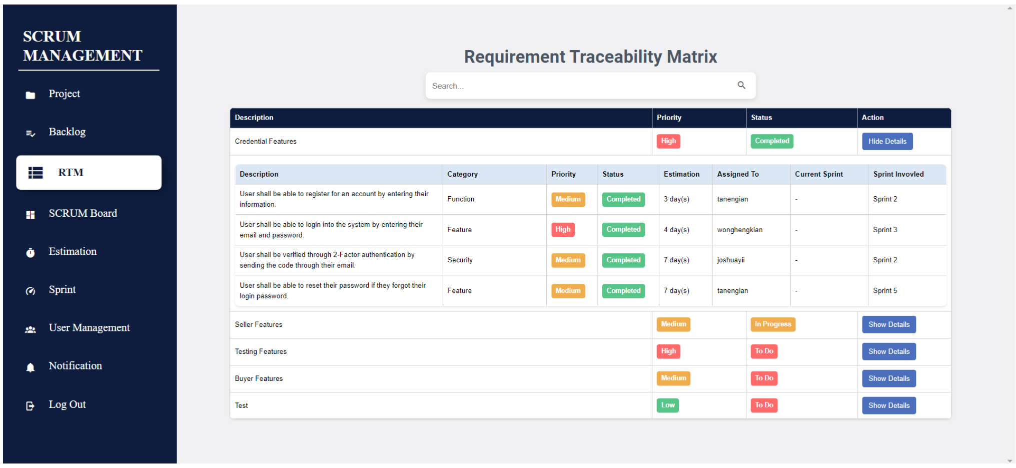
6.5.5 Requirement Traceability Matrix (RTM)



Oops, looks like you dont have a backlog yet.

Dont worry! Create your first backlog and the RTM will be auto-generated!

Figure 6.82: RTM Page (No Backlog)



Requirement Traceability Matrix

Search...

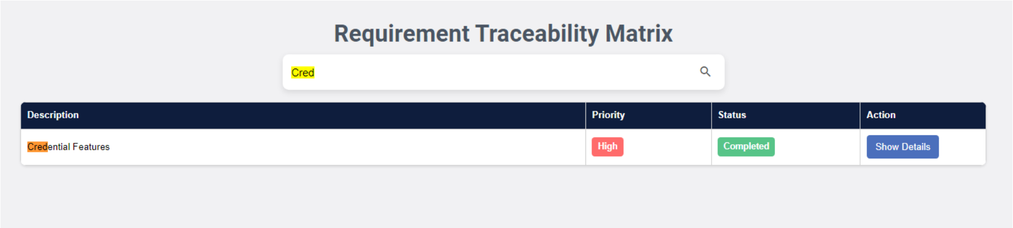
Description	Priority	Status	Action
Credential Features	High	Completed	Hide Details

Description	Category	Priority	Status	Estimation	Assigned To	Current Sprint	Sprint Invovled
User shall be able to register for an account by entering their information.	Function	Medium	Completed	3 day(s)	tanengian	-	Sprint 2
User shall be able to login into the system by entering their email and password.	Feature	High	Completed	4 day(s)	wonghengkian	-	Sprint 3
User shall be verified through 2-Factor authentication by sending the code through their email.	Security	Medium	Completed	7 day(s)	joshuayii	-	Sprint 2
User shall be able to reset their password if they forgot their login password.	Feature	Medium	Completed	7 day(s)	tanengian	-	Sprint 5

Description	Priority	Status	Action
Seller Features	Medium	In Progress	Show Details
Testing Features	High	To Do	Show Details
Buyer Features	Medium	To Do	Show Details
Test	Low	To Do	Show Details

Figure 6.83: RTM Page

The Requirement Traceability Matrix (RTM) is a central focus of the final year project, summarizing all backlog information into a single table and enhancing the visibility of the project timeline. On the RTM page, users can view all product backlog items created within the project. Clicking on the "Show Details" button expands the RTM, displaying associated sprint backlog items. If no backlogs are found, the web application will display a “Not Found” page.

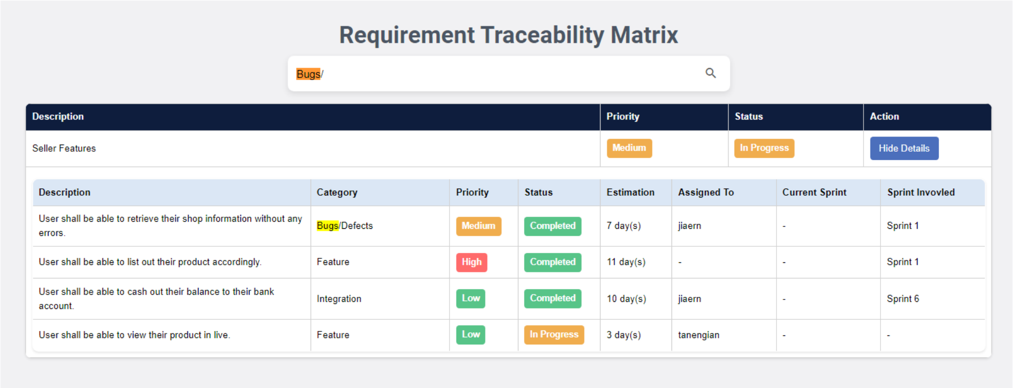


Requirement Traceability Matrix

Cred

Description	Priority	Status	Action
Credential Features	High	Completed	Show Details

Figure 6.84: Product Backlog Search Result



Requirement Traceability Matrix

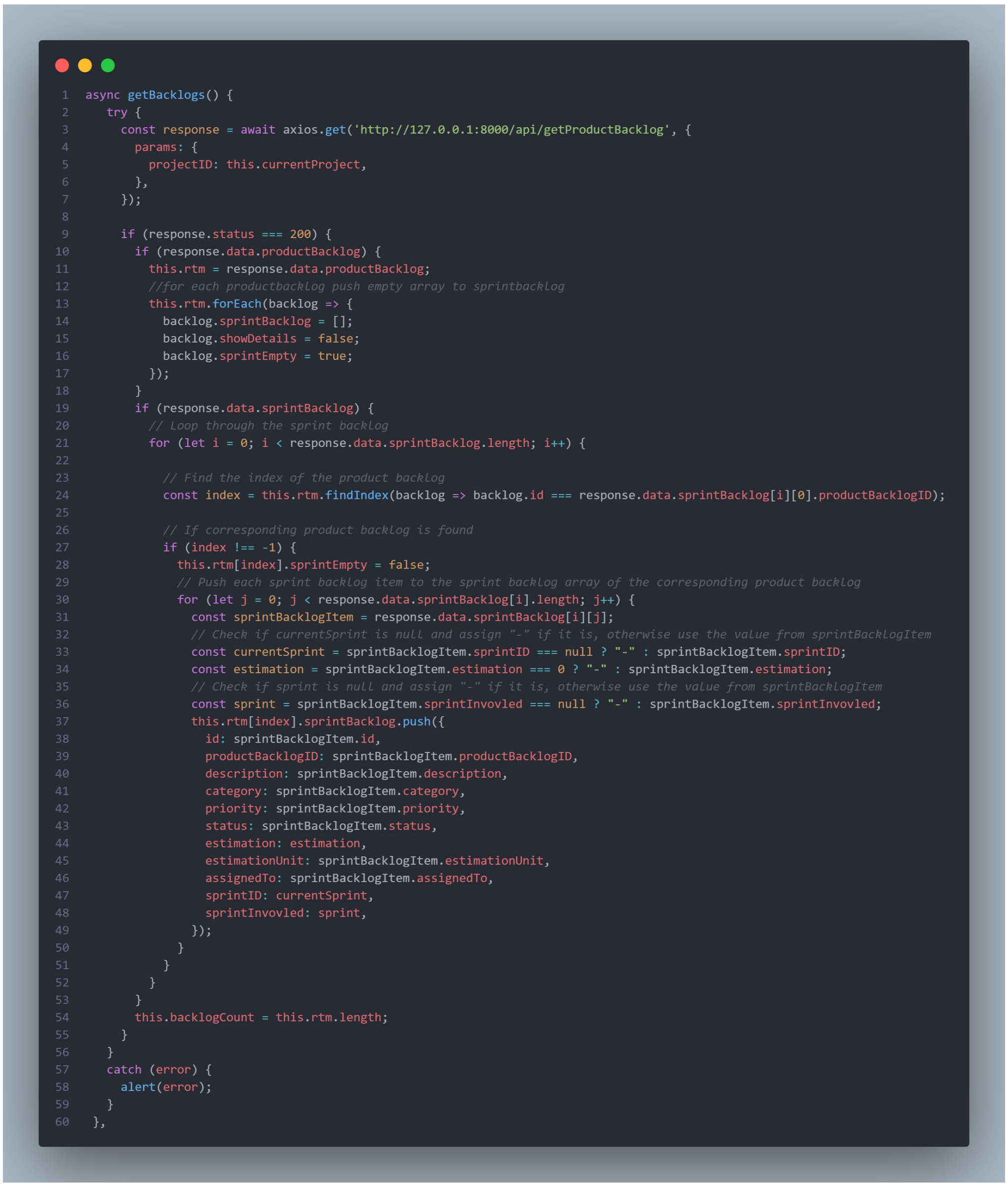
Bugs/

Description	Priority	Status	Action
Seller Features	Medium	In Progress	Hide Details

Description	Category	Priority	Status	Estimation	Assigned To	Current Sprint	Sprint Invovled
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	7 day(s)	jiaern	-	Sprint 1
User shall be able to list out their product accordingly.	Feature	High	Completed	11 day(s)	-	-	Sprint 1
User shall be able to cash out their balance to their bank account.	Integration	Low	Completed	10 day(s)	jiaern	-	Sprint 6
User shall be able to view their product in live.	Feature	Low	In Progress	3 day(s)	tanengian	-	-

Figure 6.85: Sprint Backlog Search Result

The RTM features a search function that allows users to filter the table. By further incorporating a search algorithm within the RTM, user can extract the information they needed quickly without navigating through multiple pages. The search algorithm is reactive, updating the page view whenever a single character is typed. If the searched keyword is found in the product backlog, the RTM will display the filtered results along with associated sprint backlog item. However, if the keyword is found in the sprint backlog, the "Show Details" section will automatically expand to display the relevant sprint backlog item.

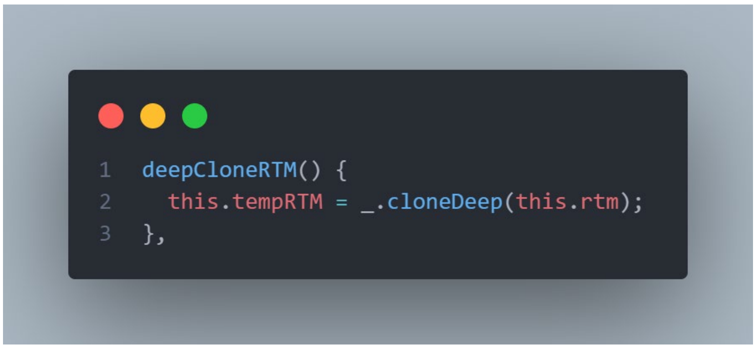


```javascript
async getBacklogs() {
   try {
     const response = await axios.get('http://127.0.0.1:8000/api/getProductBacklog', {
       params: {
         projectID: this.currentProject,
       },
     });

     if (response.status === 200) {
       if (response.data.productBacklog) {
         this.rtm = response.data.productBacklog;
         //for each productbacklog push empty array to sprintbacklog
         this.rtm.forEach(backlog => {
           backlog.sprintBacklog = [];
           backlog.showDetails = false;
           backlog.sprintEmpty = true;
         });
       }
       if (response.data.sprintBacklog) {
         // Loop through the sprint backlog
         for (let i = 0; i < response.data.sprintBacklog.length; i++) {

           // Find the index of the product backlog
           const index = this.rtm.findIndex(backlog => backlog.id === response.data.sprintBacklog[i][0].productBacklogID);

           // If corresponding product backlog is found
           if (index !== -1) {
             this.rtm[index].sprintEmpty = false;
             // Push each sprint backlog item to the sprint backlog array of the corresponding product backlog
             for (let j = 0; j < response.data.sprintBacklog[i].length; j++) {
               const sprintBacklogItem = response.data.sprintBacklog[i][j];
               // Check if currentSprint is null and assign "-" if it is, otherwise use the value from sprintBacklogItem
               const currentSprint = sprintBacklogItem.sprintID === null ? "-" : sprintBacklogItem.sprintID;
               const estimation = sprintBacklogItem.estimation === 0 ? "-" : sprintBacklogItem.estimation;
               // Check if sprint is null and assign "-" if it is, otherwise use the value from sprintBacklogItem
               const sprint = sprintBacklogItem.sprintInvovled === null ? "-" : sprintBacklogItem.sprintInvovled;
               this.rtm[index].sprintBacklog.push({
                 id: sprintBacklogItem.id,
                 productBacklogID: sprintBacklogItem.productBacklogID,
                 description: sprintBacklogItem.description,
                 category: sprintBacklogItem.category,
                 priority: sprintBacklogItem.priority,
                 status: sprintBacklogItem.status,
                 estimation: estimation,
                 estimationUnit: sprintBacklogItem.estimationUnit,
                 assignedTo: sprintBacklogItem.assignedTo,
                 sprintID: currentSprint,
                 sprintInvovled: sprint,
               });
             }
           }
         }
       }
       this.backlogCount = this.rtm.length;
     }
   }
   catch (error) {
     alert(error);
   }
 },
```

Figure 6.86: Get Backlogs Function (Vue.js)

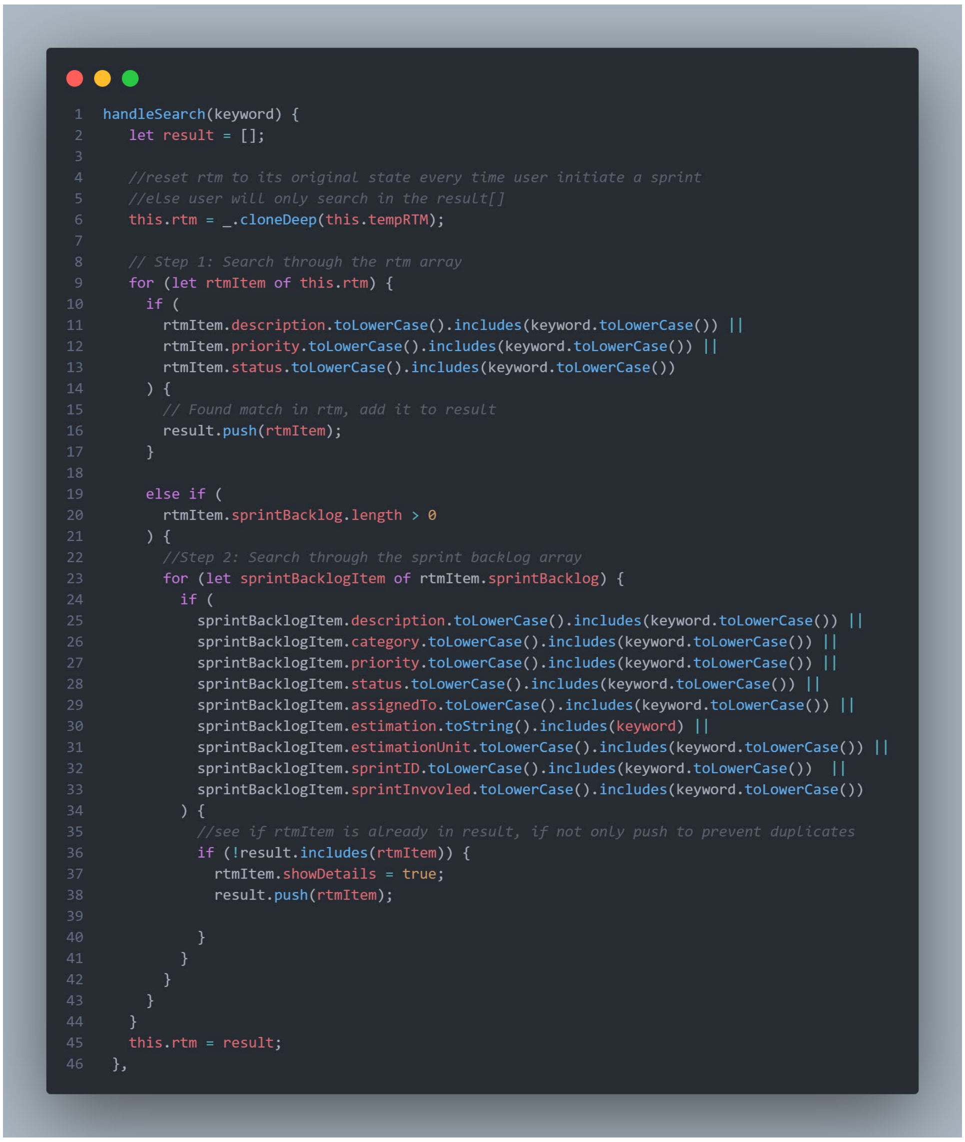
The search algorithm is implemented in Vue.js to minimize waiting time for users. Sending requests to the backend in Laravel would result in a delay before results are returned. By implementing the algorithm on the frontend, this delay is reduced, resulting in faster search result.

The algorithm first combines product backlog items with their associated sprint backlog items. This approach offers advantages, such as allowing the web application to navigate to the relevant sprint backlog items without having to go through the entire product backlog first. While the efficiency may be higher when the relevant sprint backlog item is among the first occurrences, it may decrease if it's among the last occurrences. However, the combined approach is generally more efficient. The combined backlogs will then replace the RTM variable.



```
deepCloneRTM() {
  this.tempRTM = _.cloneDeep(this.rtm);
},
```

Figure 6.87: Deep Clone RTM Function



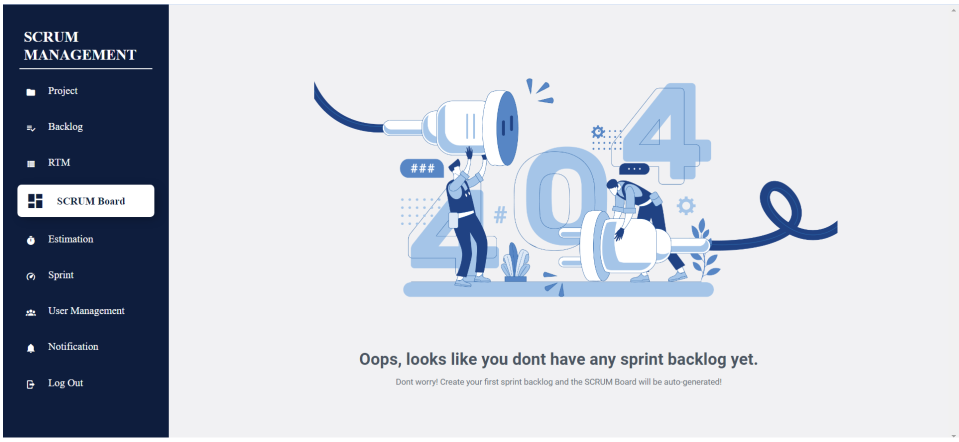
```
handleSearch(keyword) {
   let result = [];

   //reset rtm to its original state every time user initiate a sprint
   //else user will only search in the result[]
   this.rtm = _.cloneDeep(this.tempRTM);

   // Step 1: Search through the rtm array
   for (let rtmItem of this.rtm) {
     if (
       rtmItem.description.toLowerCase().includes(keyword.toLowerCase()) ||
       rtmItem.priority.toLowerCase().includes(keyword.toLowerCase()) ||
       rtmItem.status.toLowerCase().includes(keyword.toLowerCase())
     ) {
       // Found match in rtm, add it to result
       result.push(rtmItem);
     }

     else if (
       rtmItem.sprintBacklog.length > 0
     ) {
       //Step 2: Search through the sprint backlog array
       for (let sprintBacklogItem of rtmItem.sprintBacklog) {
         if (
           sprintBacklogItem.description.toLowerCase().includes(keyword.toLowerCase()) ||
           sprintBacklogItem.category.toLowerCase().includes(keyword.toLowerCase()) ||
           sprintBacklogItem.priority.toLowerCase().includes(keyword.toLowerCase()) ||
           sprintBacklogItem.status.toLowerCase().includes(keyword.toLowerCase()) ||
           sprintBacklogItem.assignedTo.toLowerCase().includes(keyword.toLowerCase()) ||
           sprintBacklogItem.estimation.toString().includes(keyword) ||
           sprintBacklogItem.estimationUnit.toLowerCase().includes(keyword.toLowerCase()) ||
           sprintBacklogItem.sprintID.toLowerCase().includes(keyword.toLowerCase())  ||
           sprintBacklogItem.sprintInvovled.toLowerCase().includes(keyword.toLowerCase())
         ) {
           //see if rtmItem is already in result, if not only push to prevent duplicates
           if (!result.includes(rtmItem)) {
             rtmItem.showDetails = true;
             result.push(rtmItem);

           }
         }
       }
     }
   }
   this.rtm = result;
 },
```

Figure 6.88: Search Algorithm (Vue.js)

The RTM variable is cloned into a variable named tempRTM to reset the RTM state after each search request. If the user were to search without resetting the RTM, subsequent search requests would be based on the filtered version of RTM. tempRTM restores it to its original state, enabling the search algorithm to be used repeatedly.

The search algorithm first checks the product backlog of the RTM. If any keyword matches, it returns the product backlog item along with its associated sprint backlog item. If no match is found, it then searches the sprint backlog item of the product backlog item. If a match is found, both the product backlog item and sprint backlog item are returned, and the "Show Details" flag is set to true to automatically display the sprint backlog. This process continues until a matching keyword is found or the end of the RTM is reached.

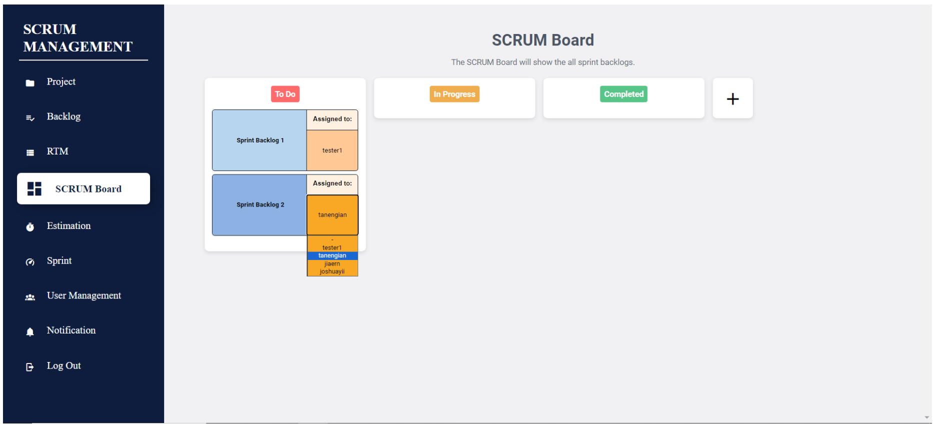
6.5.6 SCRUM Board



Oops, looks like you dont have any sprint backlog yet.

Dont worry! Create your first sprint backlog and the SCRUM Board will be auto-generated!

Figure 6.89: SCRUM Board Page (No Sprint Backlog)

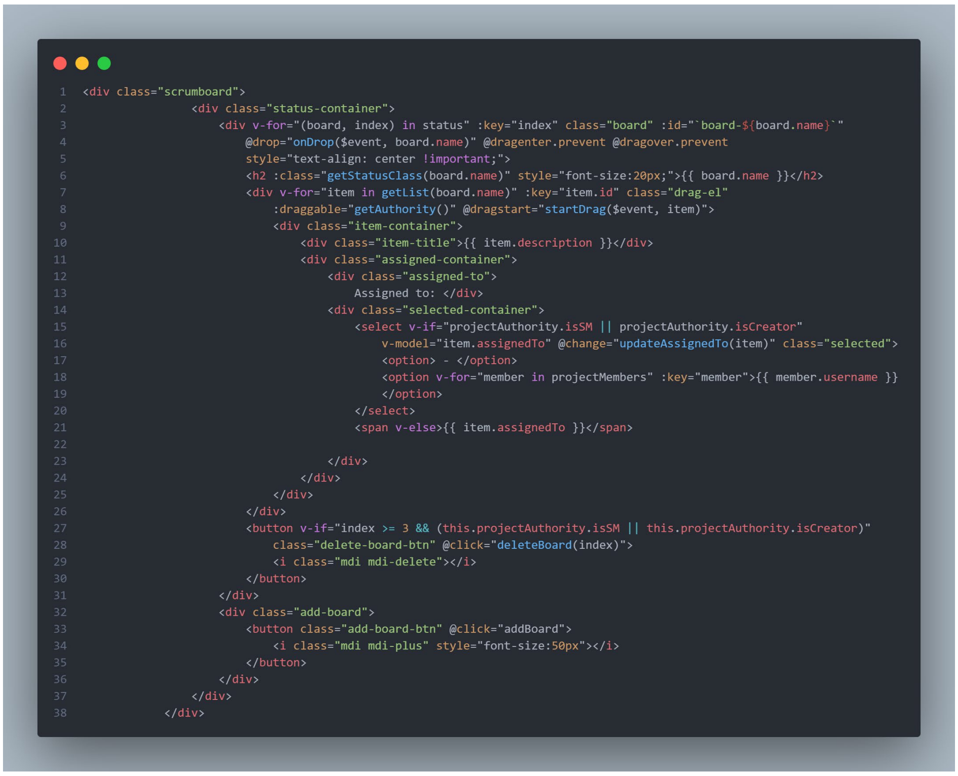


SCRUM Board

The SCRUM Board will show the all sprint backlogs.

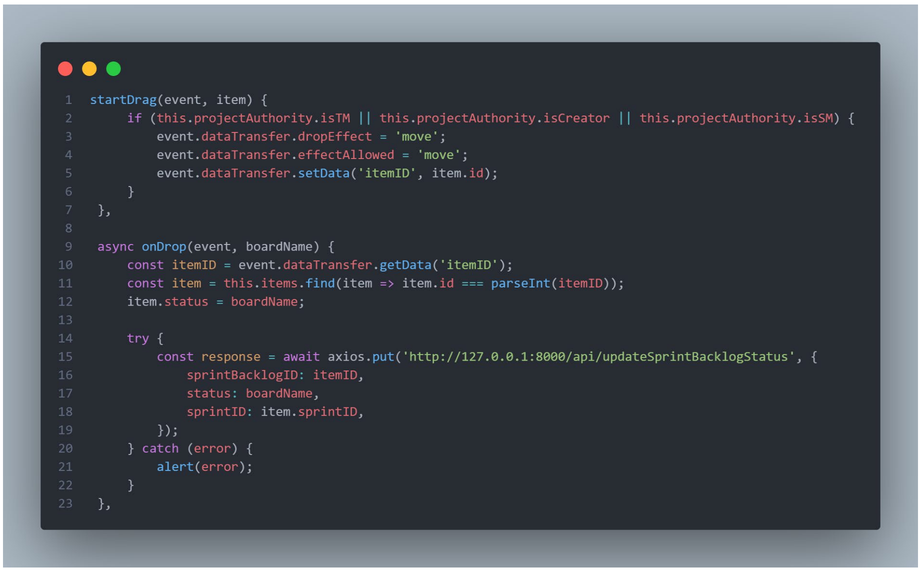
Figure 6.90: SCRUM Board Page

The SCRUM Board page displays all sprint backlog items for the current project. If no sprint backlog is found, a “Not Found” page will be displayed. SCRUM Master, Team Member, and Creator can drag and drop sprint backlog items between boards and modify the "Assigned To" for each item. Each board is representing a status in the project.



```html
<div class="scrumboard">
                <div class="status-container">
                    <div v-for="(board, index) in status" :key="index" class="board" :id="`board-${board.name}`"
                        @drop="onDrop($event, board.name)" @dragenter.prevent @dragover.prevent
                        style="text-align: center !important;">
                        <h2 :class="getStatusClass(board.name)" style="font-size:20px;">{{ board.name }}</h2>
                        <div v-for="item in getList(board.name)" :key="item.id" class="drag-el"
                            :draggable="getAuthority()" @dragstart="startDrag($event, item)">
                            <div class="item-container">
                                <div class="item-title">{{ item.description }}</div>
                                <div class="assigned-container">
                                    <div class="assigned-to">
                                        Assigned to: </div>
                                    <div class="selected-container">
                                        <select v-if="projectAuthority.isSM || projectAuthority.isCreator"
                                            v-model="item.assignedTo" @change="updateAssignedTo(item)" class="selected">
                                            <option> - </option>
                                            <option v-for="member in projectMembers" :key="member">{{ member.username }}
                                            </option>
                                        </select>
                                        <span v-else>{{ item.assignedTo }}</span>

                                    </div>
                                </div>
                            </div>
                        </div>
                        <button v-if="index >= 3 && (this.projectAuthority.isSM || this.projectAuthority.isCreator)"
                            class="delete-board-btn" @click="deleteBoard(index)">
                            <i class="mdi mdi-delete"></i>
                        </button>
                    </div>
                    <div class="add-board">
                        <button class="add-board-btn" @click="addBoard">
                            <i class="mdi mdi-plus" style="font-size:50px"></i>
                        </button>
                    </div>
                </div>
            </div>
```

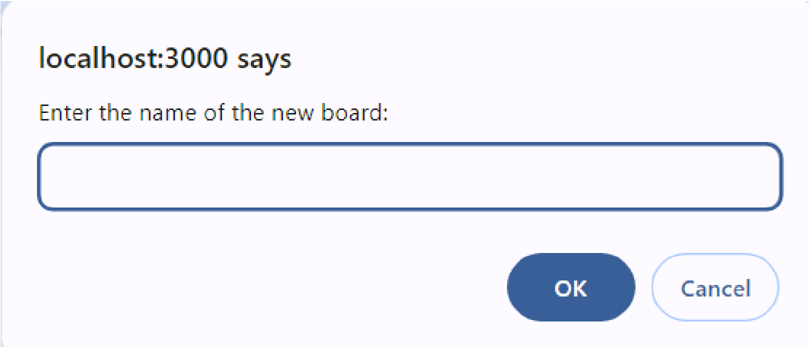
Figure 6.91: SCRUM Board HTML Code (Vue.js)



```javascript
startDrag(event, item) {
     if (this.projectAuthority.isTM || this.projectAuthority.isCreator || this.projectAuthority.isSM) {
         event.dataTransfer.dropEffect = 'move';
         event.dataTransfer.effectAllowed = 'move';
         event.dataTransfer.setData('itemID', item.id);
     }
 },

 async onDrop(event, boardName) {
     const itemID = event.dataTransfer.getData('itemID');
     const item = this.items.find(item => item.id === parseInt(itemID));
     item.status = boardName;

     try {
         const response = await axios.put('http://127.0.0.1:8000/api/updateSprintBacklogStatus', {
             sprintBacklogID: itemID,
             status: boardName,
             sprintID: item.sprintID,
         });
     } catch (error) {
         alert(error);
     }
 },
```

Figure 6.92: SCRUM Board Drag and Drop Event (Vue.js)

The drag-and-drop functionality of the SCRUM Board is implemented as an event. When a user initiates a drag action on a sprint backlog item, the current item's ID is set as the data being moved. Upon dropping the item, the onDrop function determines the target location and updates the status accordingly. Additionally, the onDrop function sends an Axios request to Laravel to ensure that the sprint backlog is updated across all pages. This request includes the sprint backlog ID, the new status, and the associated sprint ID.

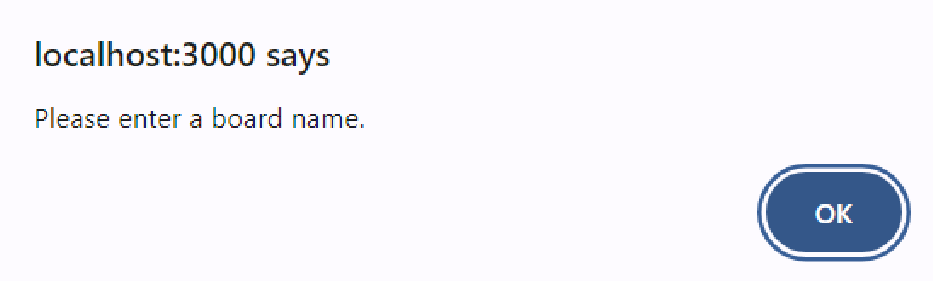


localhost:3000 says

Enter the name of the new board:

Figure 6.93: Adding New SCRUM Board (New Status)

User can add new add new SCRUM Board by clicking on the "+" button at the end of the board.



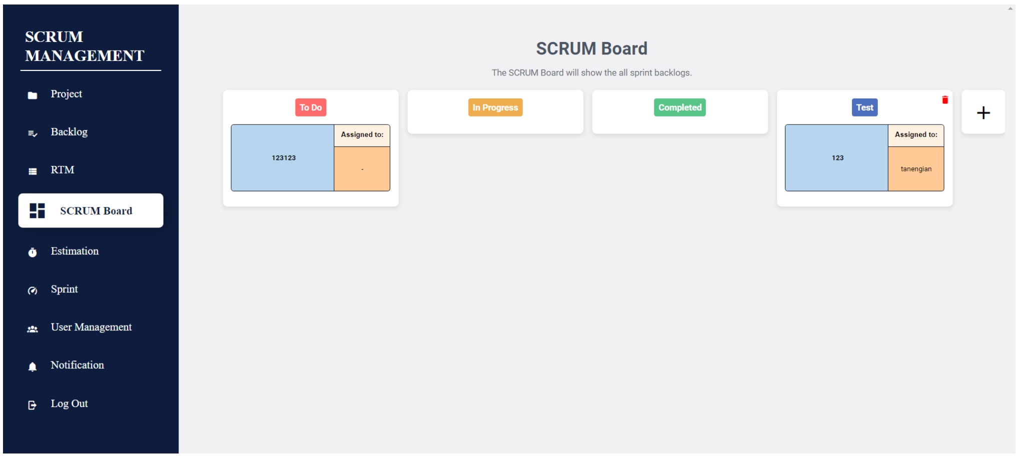
localhost:3000 says

Please enter a board name.

Figure 6.94: Empty SCRUM Board Validation

If any of the user input has triggered the validation, an alert message will be shown to user. The following scenario has been validated:

1. Empty SCRUM Board name
2. SCRUM Board already exist in the project



SCRUM Board

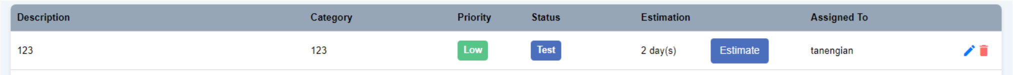
The SCRUM Board will show the all sprint backlogs.

	Assigned to:
123123	-

	Assigned to:
123	tanengian

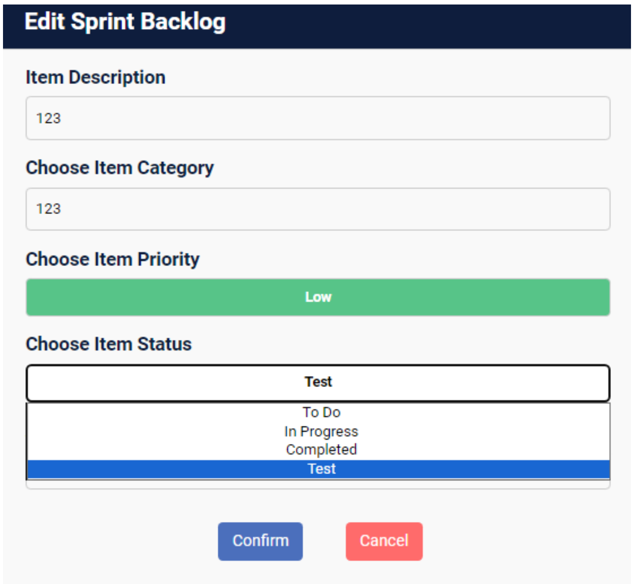
Figure 6.95: New SCRUM Board

The new SCRUM Board will be shown in the SCRUM Board page, in which user can drag and drop the sprint backlog item onto the new board.



Description	Category	Priority	Status	Estimation		Assigned To	
123	123	Low	Test	2 day(s)	Estimate	tanengian	

Figure 6.96: Backlog Page



Edit Sprint Backlog

Item Description

123

Choose Item Category

123

Choose Item Priority

Low

Choose Item Status

Test

To Do
In Progress
Completed
Test

Figure 6.97: Sprint Backlog Status Selection

The other pages of the web application will also reflect the updated status of the sprint backlog item, and the dropdown selection for the item's status will display the newly customized status.

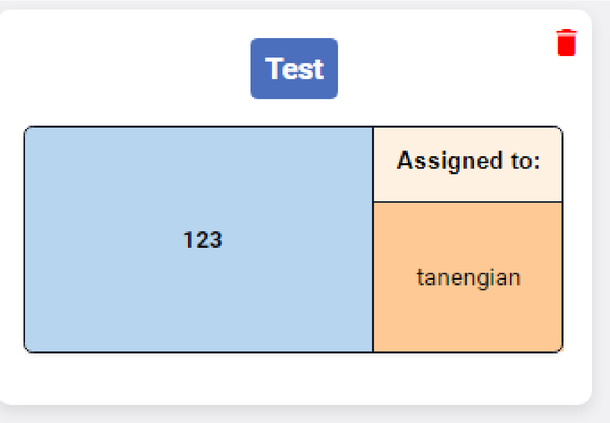
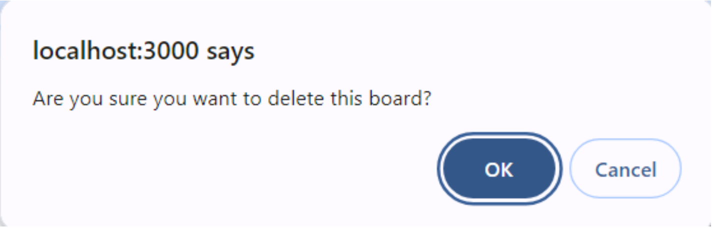


Figure 6.98: Customized SCRUM Board



localhost:3000 says

Are you sure you want to delete this board?

Figure 6.99: Confirmation of Deleting SCRUM Board

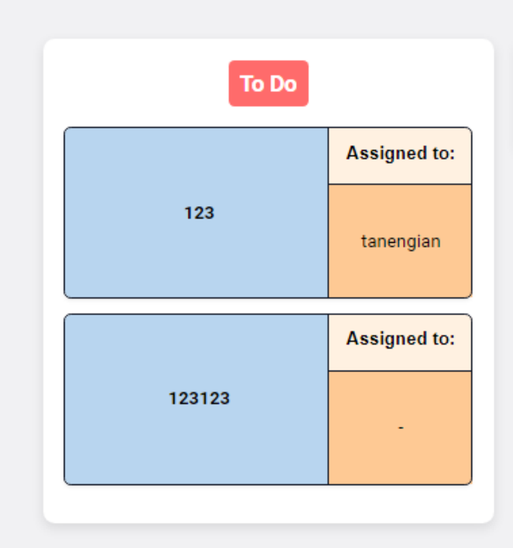
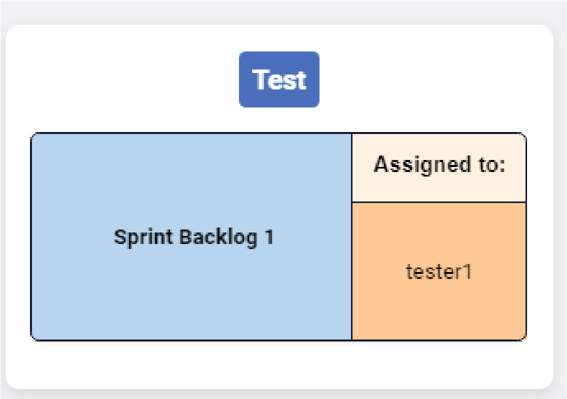


Figure 6.100: Customized Status Sprint Backlog Set to To Do

For customized boards, a delete icon will appear at the top right corner of the board, allowing users to delete it. Upon confirmation of deletion, all sprint backlog items within the board will be updated to the status "To Do".



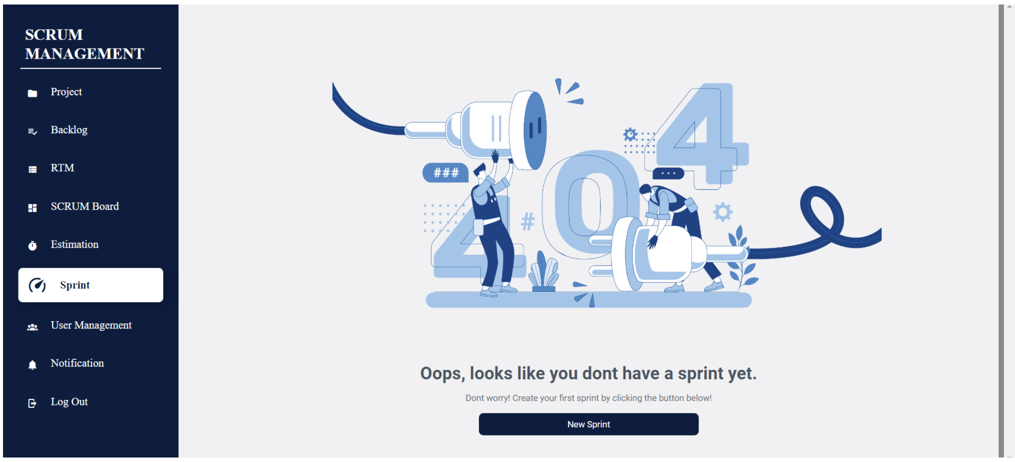
Test

Sprint Backlog 1	Assigned to:
	tester1

Figure 6.101: Customize Board (Product Owner View)

If the user is a Product Owner, the sprint backlog items will not be draggable, the "Assigned To" field will not be modifiable, and the delete icon will not appear on customized boards.

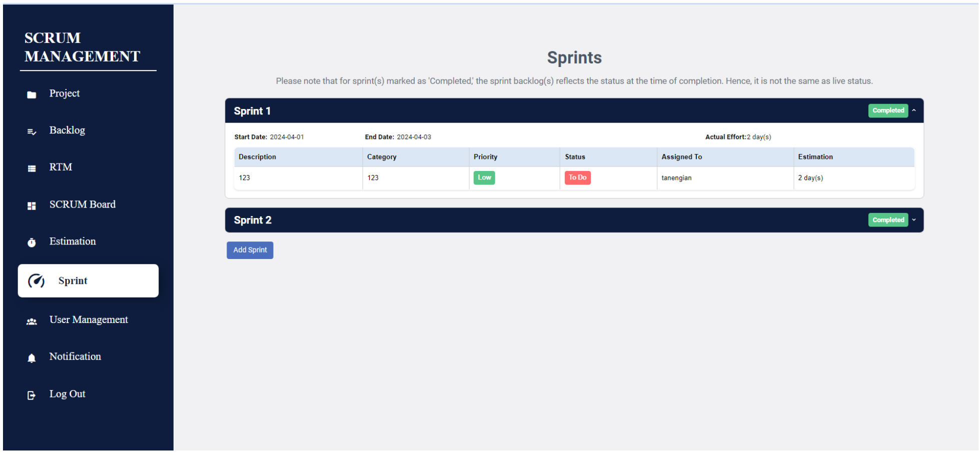
6.5.7 Sprint



Oops, looks like you dont have a sprint yet.

Dont worry! Create your first sprint by clicking the button below!

Figure 6.102: Sprint Page (No Sprint)



Sprints

Please note that for sprint(s) marked as 'Completed,' the sprint backlog(s) reflects the status at the time of completion. Hence, it is not the same as live status.

Sprint 1

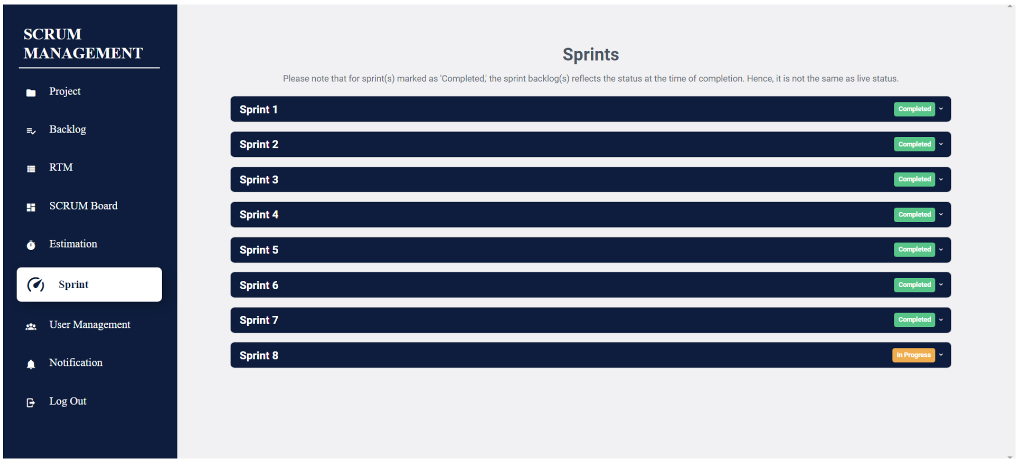
Start Date: 2024-04-01 End Date: 2024-04-03 Actual Effort:2 day(s)

Description	Category	Priority	Status	Assigned To	Estimation
123	123	Low	To Do	tanengian	2 day(s)

Sprint 2

Figure 6.103: Sprint Page

The sprint page will display all the sprints that the SCRUM Master has initiated. If no sprints are found, the web application will show a “Not Found” page and prompt the user to initiate a sprint. In both scenarios, clicking the "New Sprint" button will redirects the user to the add sprint page.



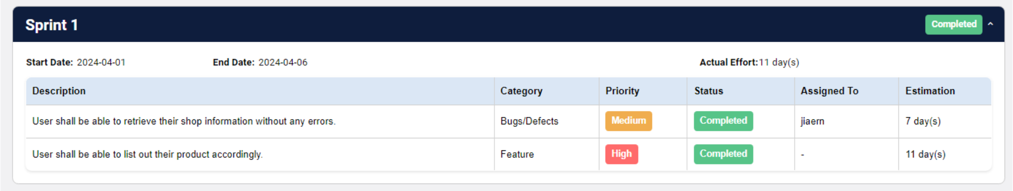
Sprints

Please note that for sprint(s) marked as 'Completed,' the sprint backlog(s) reflects the status at the time of completion. Hence, it is not the same as live status.

Sprint 1	Completed
Sprint 2	Completed
Sprint 3	Completed
Sprint 4	Completed
Sprint 5	Completed
Sprint 6	Completed
Sprint 7	Completed
Sprint 8	In Progress

Figure 6.104: Active Sprint Exist

The "New Sprint" button will only appear if there is currently no active sprint and only if the user is a SCRUM Master or creator.

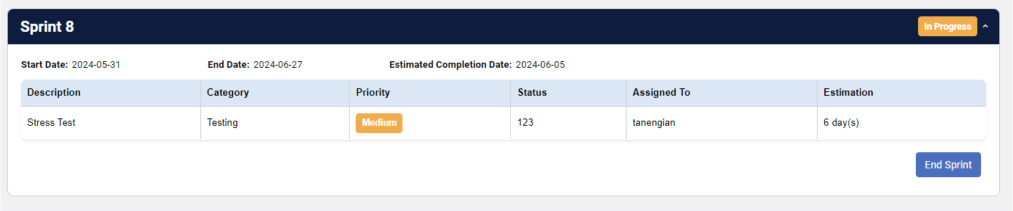


Sprint 1 — Completed

Start Date: 2024-04-01 **End Date:** 2024-04-06 **Actual Effort:** 11 day(s)

Description	Category	Priority	Status	Assigned To	Estimation
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	jiaern	7 day(s)
User shall be able to list out their product accordingly.	Feature	High	Completed	-	11 day(s)

Figure 6.105: Completed Sprint



Sprint 8 — In Progress

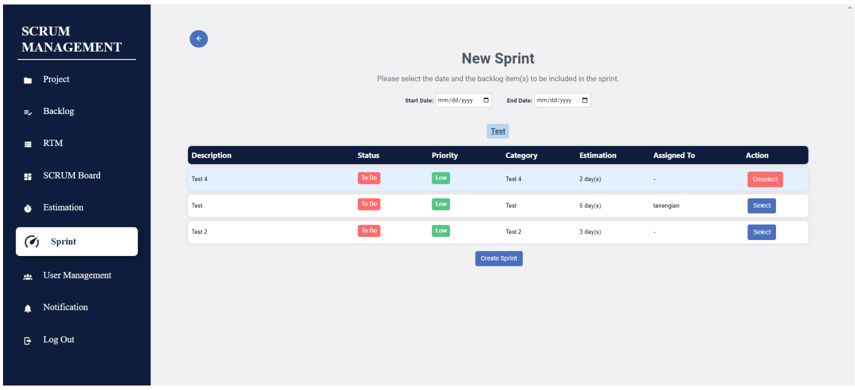
Start Date: 2024-05-31 **End Date:** 2024-06-27 **Estimated Completion Date:** 2024-06-05

Description	Category	Priority	Status	Assigned To	Estimation
Stress Test	Testing	Medium	123	tanengian	6 day(s)

Figure 6.106: Active Sprint

The sprint backlog items within a completed sprint will be reflected based on their status at the time the user completed the sprint. However, for active sprints, the sprint backlog items will reflect their current status.

6.5.7.1 Initiate Sprint



New Sprint

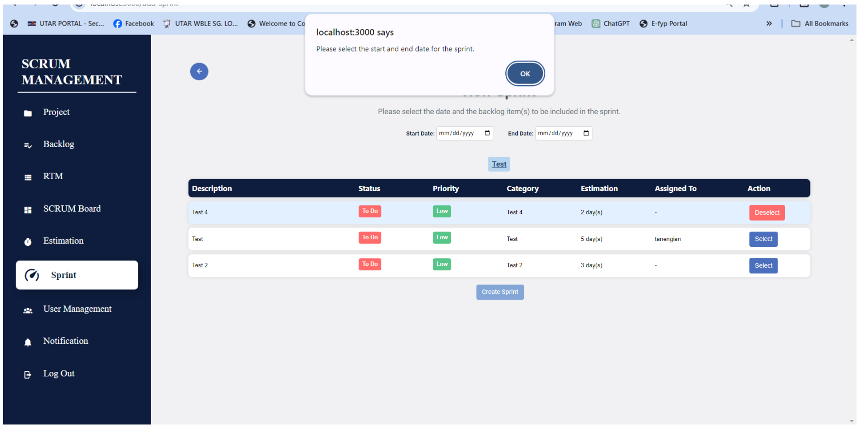
Please select the date and the backlog item(s) to be included in the sprint.

Start Date: mm/dd/yyyy End Date: mm/dd/yyyy

Description	Status	Priority	Category	Estimation	Assigned To	Action
Test 4	To Do	Low	Test 4	2 day(s)	-	Deselect
Test	To Do	Low	Test	5 day(s)	tanengian	Select
Test 2	To Do	Low	Test 2	3 day(s)	-	Select

Figure 6.107: Initiate Sprint

The add sprint page will display a list of sprints that are not currently marked as "Completed." Users will be able to select from these sprints to include them as part of the upcoming sprint.



localhost:3000 says

Please select the start and end date for the sprint.

Please select the date and the backlog item(s) to be included in the sprint.

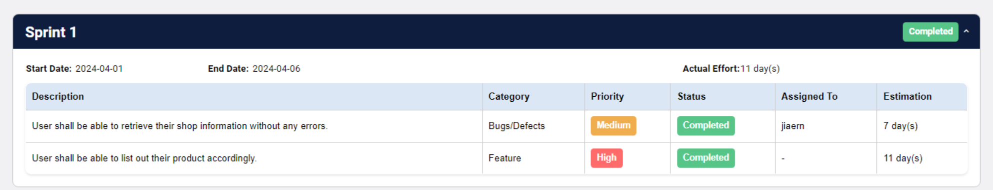
Start Date: mm/dd/yyyy End Date: mm/dd/yyyy

Description	Status	Priority	Category	Estimation	Assigned To	Action
Test 4	To Do	Low	Test 4	2 day(s)	-	Deselect
Test	To Do	Low	Test	5 day(s)	tanengian	Select
Test 2	To Do	Low	Test 2	3 day(s)	-	Select

Figure 6.108: Empty Dates Validation

If any of the user input has triggered the validation, an alert message will be shown to user. The following scenario has been validated:

1. Empty start date
2. Empty end date
3. Start Date is greater than end date
4. No sprint backlog item selected

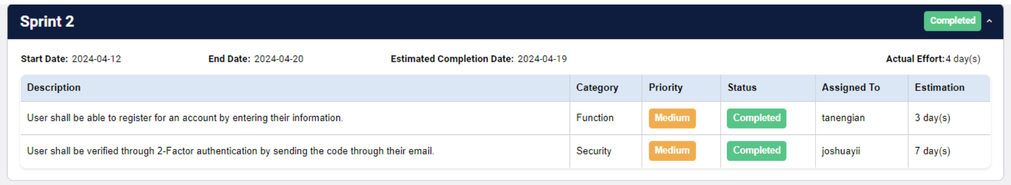


Sprint 1

Start Date: 2024-04-01 End Date: 2024-04-06 Actual Effort: 11 day(s)

Description	Category	Priority	Status	Assigned To	Estimation
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	jiaern	7 day(s)
User shall be able to list out their product accordingly.	Feature	High	Completed	-	11 day(s)

Figure 6.109: First Sprint

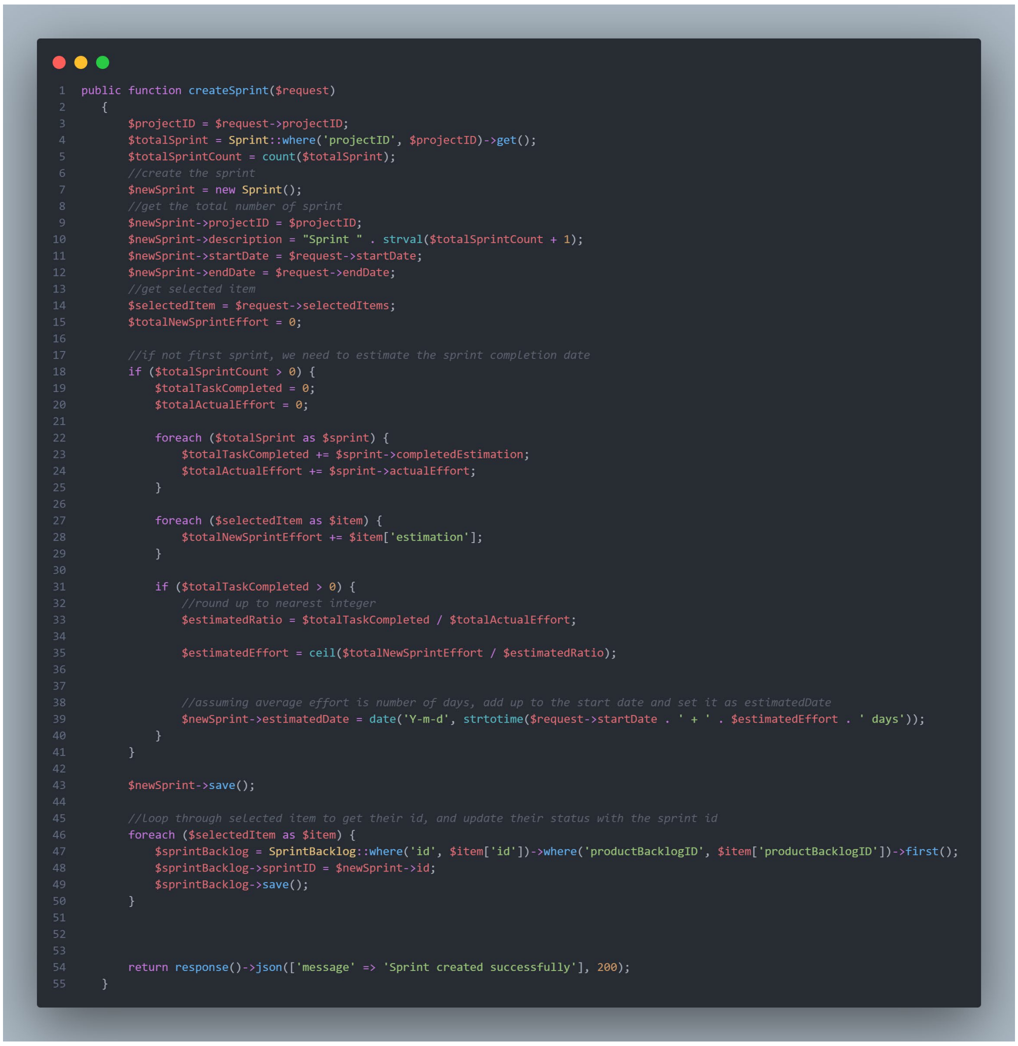


Sprint 2 Completed

Start Date: 2024-04-12 End Date: 2024-04-20 Estimated Completion Date: 2024-04-19 Actual Effort: 4 day(s)

Description	Category	Priority	Status	Assigned To	Estimation
User shall be able to register for an account by entering their information.	Function	Medium	Completed	tanengian	3 day(s)
User shall be verified through 2-Factor authentication by sending the code through their email.	Security	Medium	Completed	joshuayii	7 day(s)

Figure 6.110: Second Sprint

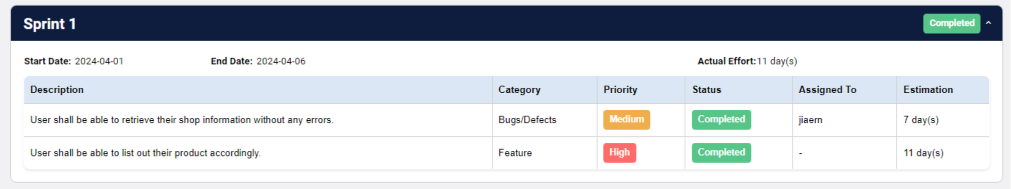


```php
public function createSprint($request)
    {
        $projectID = $request->projectID;
        $totalSprint = Sprint::where('projectID', $projectID)->get();
        $totalSprintCount = count($totalSprint);
        //create the sprint
        $newSprint = new Sprint();
        //get the total number of sprint
        $newSprint->projectID = $projectID;
        $newSprint->description = "Sprint " . strval($totalSprintCount + 1);
        $newSprint->startDate = $request->startDate;
        $newSprint->endDate = $request->endDate;
        //get selected item
        $selectedItem = $request->selectedItems;
        $totalNewSprintEffort = 0;

        //if not first sprint, we need to estimate the sprint completion date
        if ($totalSprintCount > 0) {
            $totalTaskCompleted = 0;
            $totalActualEffort = 0;

            foreach ($totalSprint as $sprint) {
                $totalTaskCompleted += $sprint->completedEstimation;
                $totalActualEffort += $sprint->actualEffort;
            }

            foreach ($selectedItem as $item) {
                $totalNewSprintEffort += $item['estimation'];
            }

            if ($totalTaskCompleted > 0) {
                //round up to nearest integer
                $estimatedRatio = $totalTaskCompleted / $totalActualEffort;

                $estimatedEffort = ceil($totalNewSprintEffort / $estimatedRatio);


                //assuming average effort is number of days, add up to the start date and set it as estimatedDate
                $newSprint->estimatedDate = date('Y-m-d', strtotime($request->startDate . ' + ' . $estimatedEffort . ' days'));
            }
        }

        $newSprint->save();

        //loop through selected item to get their id, and update their status with the sprint id
        foreach ($selectedItem as $item) {
            $sprintBacklog = SprintBacklog::where('id', $item['id'])->where('productBacklogID', $item['productBacklogID'])->first();
            $sprintBacklog->sprintID = $newSprint->id;
            $sprintBacklog->save();
        }



        return response()->json(['message' => 'Sprint created successfully'], 200);
    }
```

Figure 6.101: Create Sprint Function (Laravel)

If the initiated sprint is not the first sprint, users will be provided with an estimated completion date. The algorithm first calculates the total available estimation effort chosen for the current sprint. Then, it determines the user's accumulative pace by dividing the total estimation effort completed in the project through sprints by the total actual effort used to complete these efforts. This yields the average estimation effort completed per day. Next, the estimation effort of the current sprint is divided by this average, resulting in an estimate of how many days the user needs to complete the sprint. Since days cannot have decimals, this result is rounded up. Finally, this number of days is added to the start date of the sprint, yielding an estimated completion date. This approach provides users with a rough idea of when the sprint is expected to be finished based on their pace.

6.5.7.2 End Sprint

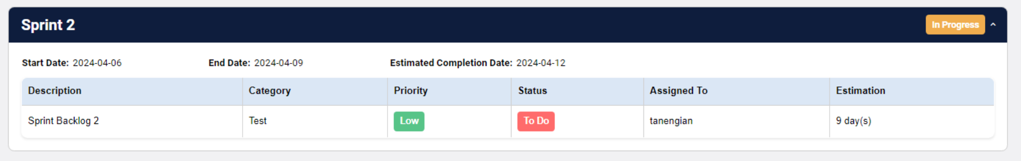


Sprint 1

Start Date: 2024-04-01 **End Date:** 2024-04-06 **Actual Effort:** 11 day(s)

Description	Category	Priority	Status	Assigned To	Estimation
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	jiaern	7 day(s)
User shall be able to list out their product accordingly.	Feature	High	Completed	-	11 day(s)

Figure 6.102: Completed Sprint

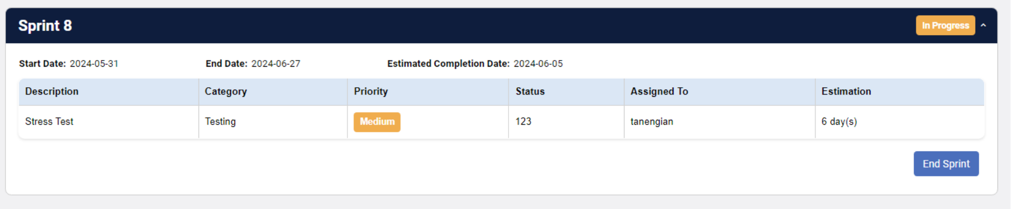


Sprint 2

Start Date: 2024-04-06 **End Date:** 2024-04-09 **Estimated Completion Date:** 2024-04-12

Description	Category	Priority	Status	Assigned To	Estimation
Sprint Backlog 2	Test	Low	To Do	tanengian	9 day(s)

Figure 6.103: Active Sprint (Product Owner and Team Member View)



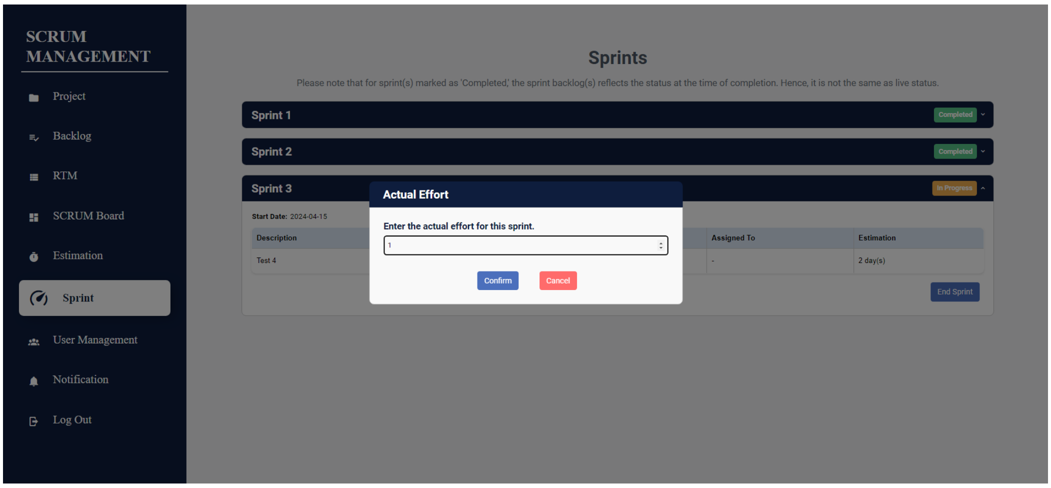
Sprint 8

Start Date: 2024-05-31 **End Date:** 2024-06-27 **Estimated Completion Date:** 2024-06-05

Description	Category	Priority	Status	Assigned To	Estimation
Stress Test	Testing	Medium	123	tanengian	6 day(s)

Figure 6.104: Active Sprint

The option to end a sprint will only be available to the SCRUM Master or creator and only when the sprint is active.



Sprints

Please note that for sprint(s) marked as 'Completed,' the sprint backlog(s) reflects the status at the time of completion. Hence, it is not the same as live status.

Sprint 1 Completed

Sprint 2 Completed

Sprint 3 In Progress

Start Date: 2024-04-15

Description	Assigned To	Estimation
Test 4	-	2 day(s)

End Sprint

Actual Effort

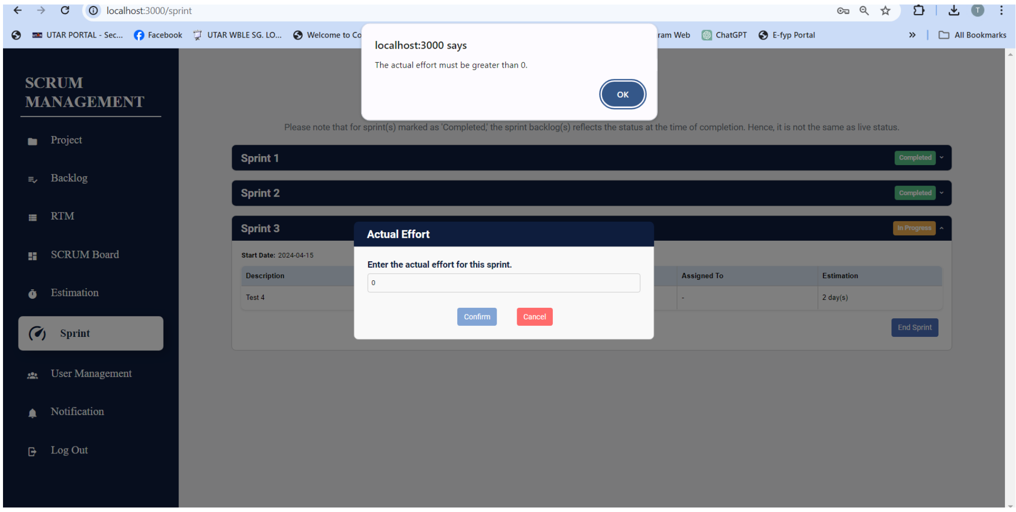
Enter the actual effort for this sprint.

1

Confirm Cancel

Figure 6.105: Actual Effort

When a user ends a sprint, the web application will prompt the user to enter the actual effort used to complete the sprint.



localhost:3000 says

The actual effort must be greater than 0.

OK

Please note that for sprint(s) marked as 'Completed,' the sprint backlog(s) reflects the status at the time of completion. Hence, it is not the same as live status.

Sprint 1 Completed

Sprint 2 Completed

Sprint 3 In Progress

Start Date: 2024-04-15

Description	Assigned To	Estimation
Test 4	-	2 day(s)

End Sprint

Actual Effort

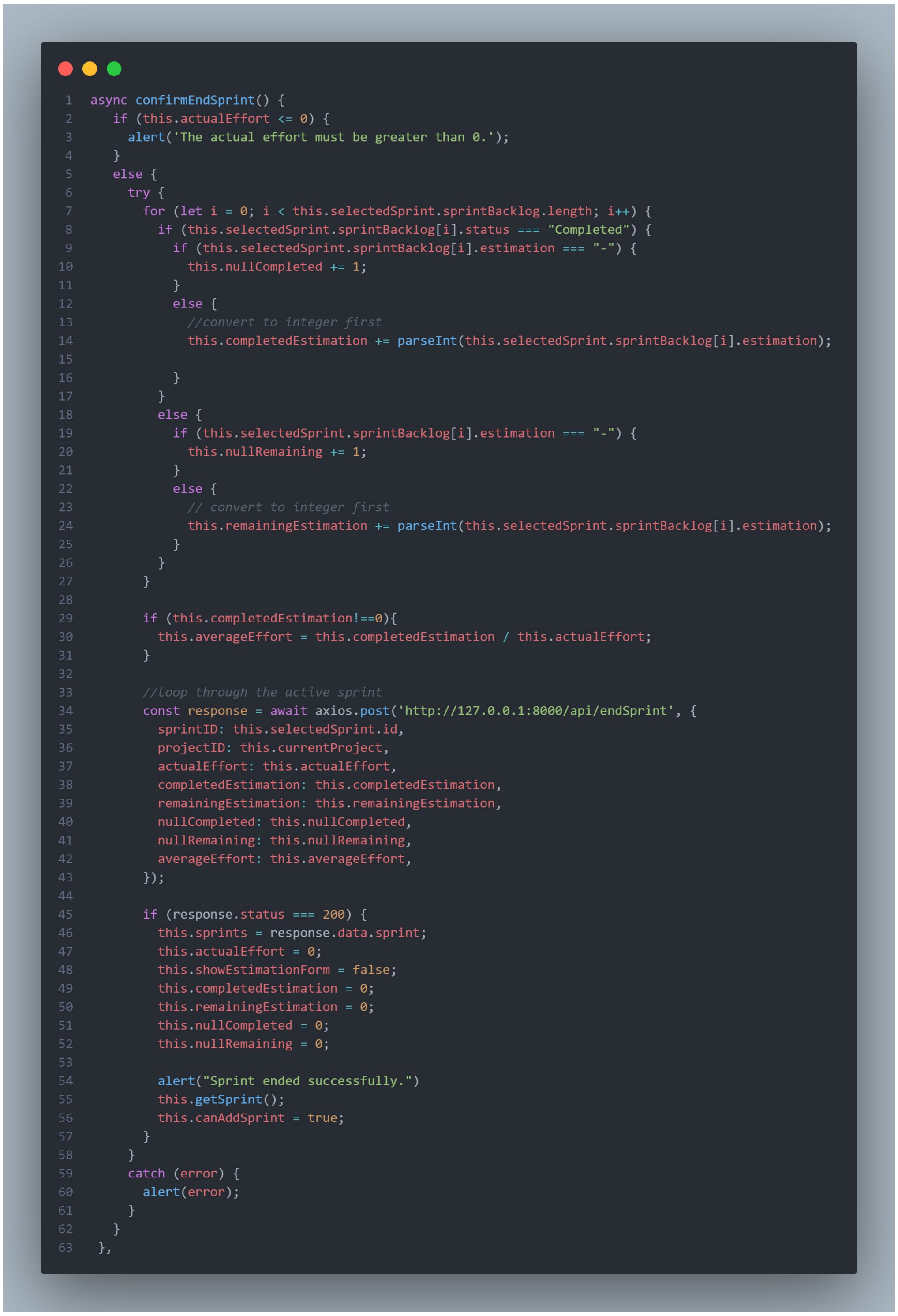
Enter the actual effort for this sprint.

0

Confirm Cancel

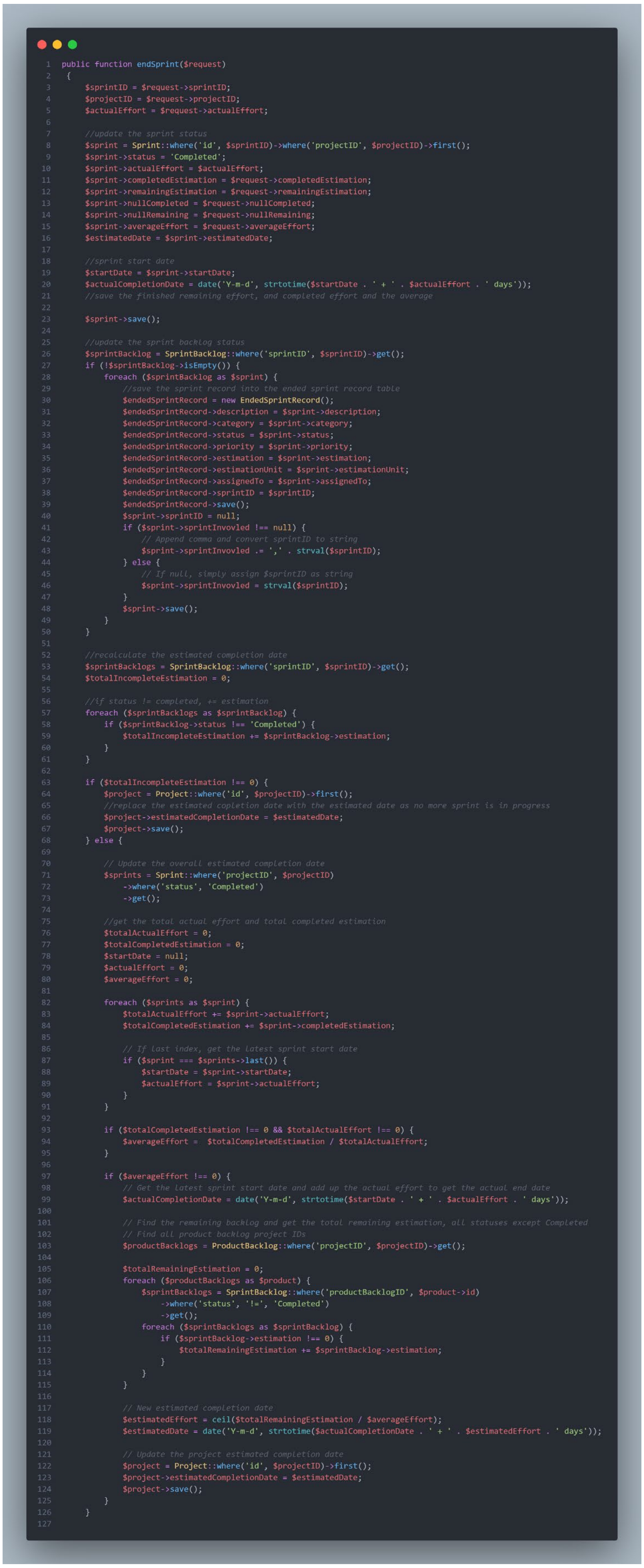
Figure 6.106: Actual Effort Validation

If the entered actual effort is less than or equal to zero, the web application will remind the user with an alert message.



```
async confirmEndSprint() {
   if (this.actualEffort <= 0) {
     alert('The actual effort must be greater than 0.');
   }
   else {
     try {
       for (let i = 0; i < this.selectedSprint.sprintBacklog.length; i++) {
         if (this.selectedSprint.sprintBacklog[i].status === "Completed") {
           if (this.selectedSprint.sprintBacklog[i].estimation === "-") {
             this.nullCompleted += 1;
           }
           else {
             //convert to integer first
             this.completedEstimation += parseInt(this.selectedSprint.sprintBacklog[i].estimation);

           }
         }
         else {
           if (this.selectedSprint.sprintBacklog[i].estimation === "-") {
             this.nullRemaining += 1;
           }
           else {
             // convert to integer first
             this.remainingEstimation += parseInt(this.selectedSprint.sprintBacklog[i].estimation);
           }
         }
       }

       if (this.completedEstimation!==0){
         this.averageEffort = this.completedEstimation / this.actualEffort;
       }

       //Loop through the active sprint
       const response = await axios.post('http://127.0.0.1:8000/api/endSprint', {
         sprintID: this.selectedSprint.id,
         projectID: this.currentProject,
         actualEffort: this.actualEffort,
         completedEstimation: this.completedEstimation,
         remainingEstimation: this.remainingEstimation,
         nullCompleted: this.nullCompleted,
         nullRemaining: this.nullRemaining,
         averageEffort: this.averageEffort,
       });

       if (response.status === 200) {
         this.sprints = response.data.sprint;
         this.actualEffort = 0;
         this.showEstimationForm = false;
         this.completedEstimation = 0;
         this.remainingEstimation = 0;
         this.nullCompleted = 0;
         this.nullRemaining = 0;

         alert("Sprint ended successfully.")
         this.getSprint();
         this.canAddSprint = true;
       }
     }
     catch (error) {
       alert(error);
     }
   }
 },
```

Figure 6.107: End Sprint Function (Vue.js)



```php
public function endSprint($request)
 {
     $sprintID = $request->sprintID;
     $projectID = $request->projectID;
     $actualEffort = $request->actualEffort;

     //update the sprint status
     $sprint = Sprint::where('id', $sprintID)->where('projectID', $projectID)->first();
     $sprint->status = 'Completed';
     $sprint->actualEffort = $actualEffort;
     $sprint->completedEstimation = $request->completedEstimation;
     $sprint->remainingEstimation = $request->remainingEstimation;
     $sprint->nullCompleted = $request->nullCompleted;
     $sprint->nullRemaining = $request->nullRemaining;
     $sprint->averageEffort = $request->averageEffort;
     $estimatedDate = $sprint->estimatedDate;

     //sprint start date
     $startDate = $sprint->startDate;
     $actualCompletionDate = date('Y-m-d', strtotime($startDate . ' + ' . $actualEffort . ' days'));
     //save the finished remaining effort, and completed effort and the average

     $sprint->save();

     //update the sprint backlog status
     $sprintBacklog = SprintBacklog::where('sprintID', $sprintID)->get();
     if (!$sprintBacklog->isEmpty()) {
         foreach ($sprintBacklog as $sprint) {
             //save the sprint record into the ended sprint record table
             $endedSprintRecord = new EndedSprintRecord();
             $endedSprintRecord->description = $sprint->description;
             $endedSprintRecord->category = $sprint->category;
             $endedSprintRecord->status = $sprint->status;
             $endedSprintRecord->priority = $sprint->priority;
             $endedSprintRecord->estimation = $sprint->estimation;
             $endedSprintRecord->estimationUnit = $sprint->estimationUnit;
             $endedSprintRecord->assignedTo = $sprint->assignedTo;
             $endedSprintRecord->sprintID = $sprintID;
             $endedSprintRecord->save();
             $sprint->sprintID = null;
             if ($sprint->sprintInvovled !== null) {
                 // Append comma and convert sprintID to string
                 $sprint->sprintInvovled .= ',' . strval($sprintID);
             } else {
                 // If null, simply assign $sprintID as string
                 $sprint->sprintInvovled = strval($sprintID);
             }
             $sprint->save();
         }
     }

     //recalculate the estimated completion date
     $sprintBacklogs = SprintBacklog::where('sprintID', $sprintID)->get();
     $totalIncompleteEstimation = 0;

     //if status != completed, += estimation
     foreach ($sprintBacklogs as $sprintBacklog) {
         if ($sprintBacklog->status !== 'Completed') {
             $totalIncompleteEstimation += $sprintBacklog->estimation;
         }
     }

     if ($totalIncompleteEstimation !== 0) {
         $project = Project::where('id', $projectID)->first();
         //replace the estimated copletion date with the estimated date as no more sprint is in progress
         $project->estimatedCompletionDate = $estimatedDate;
         $project->save();
     } else {

         // Update the overall estimated completion date
         $sprints = Sprint::where('projectID', $projectID)
             ->where('status', 'Completed')
             ->get();

         //get the total actual effort and total completed estimation
         $totalActualEffort = 0;
         $totalCompletedEstimation = 0;
         $startDate = null;
         $actualEffort = 0;
         $averageEffort = 0;

         foreach ($sprints as $sprint) {
             $totalActualEffort += $sprint->actualEffort;
             $totalCompletedEstimation += $sprint->completedEstimation;

             // If last index, get the latest sprint start date
             if ($sprint === $sprints->last()) {
                 $startDate = $sprint->startDate;
                 $actualEffort = $sprint->actualEffort;
             }
         }

         if ($totalCompletedEstimation !== 0 && $totalActualEffort !== 0) {
             $averageEffort =  $totalCompletedEstimation / $totalActualEffort;
         }

         if ($averageEffort !== 0) {
             // Get the latest sprint start date and add up the actual effort to get the actual end date
             $actualCompletionDate = date('Y-m-d', strtotime($startDate . ' + ' . $actualEffort . ' days'));

             // Find the remaining backlog and get the total remaining estimation, all statuses except Completed
             // Find all product backlog project IDs
             $productBacklogs = ProductBacklog::where('projectID', $projectID)->get();

             $totalRemainingEstimation = 0;
             foreach ($productBacklogs as $product) {
                 $sprintBacklogs = SprintBacklog::where('productBacklogID', $product->id)
                     ->where('status', '!=', 'Completed')
                     ->get();
                 foreach ($sprintBacklogs as $sprintBacklog) {
                     if ($sprintBacklog->estimation !== 0) {
                         $totalRemainingEstimation += $sprintBacklog->estimation;
                     }
                 }
             }

             // New estimated completion date
             $estimatedEffort = ceil($totalRemainingEstimation / $averageEffort);
             $estimatedDate = date('Y-m-d', strtotime($actualCompletionDate . ' + ' . $estimatedEffort . ' days'));

             // Update the project estimated completion date
             $project = Project::where('id', $projectID)->first();
             $project->estimatedCompletionDate = $estimatedDate;
             $project->save();
         }
     }

```

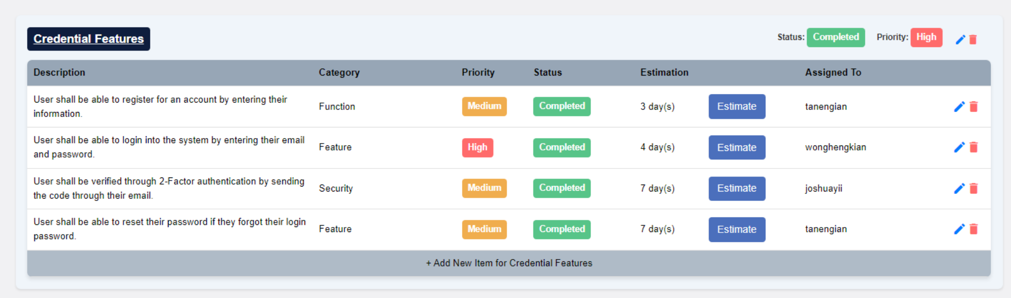
Figure 6.108: End Sprint Function (Laravel)

In Vue.js, when the sprint has ended, the web application will calculate all necessary information. This includes the total number of null value sprint backlog items completed, the total number of null value sprint backlog items remaining, the total number of estimation efforts completed, the total number of estimation efforts remaining, and the average effort used to complete the sprint. All of this information is essential for use in the estimation page, helping users visualize the timeline and their pace in the SCRUM project. After calculating this information, Vue.js will send an Axios request to the backend, along with the necessary information, to request the end of the sprint.

In Laravel, the web application recalculates the estimated completion date based on the newly ended sprint. With the end of a sprint, the average estimation effort finished per day may have changed, requiring recalculation. The algorithm first checks if there are any estimation efforts that are not marked as "Completed." If there are, recalculation begins; otherwise, the estimated completion date for the just-ended sprint becomes the new estimated completion date.

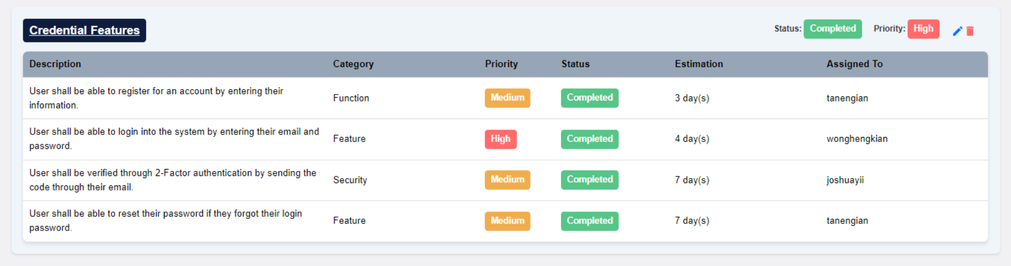
The recalculation process is similar to the creation of sprint. It starts by computing the total available estimation effort across the project that is not marked as "Completed." Then, it determines the user's new accumulative pace by dividing the total estimation effort completed in the project, including the newly ended sprint, by the total actual effort used to complete these efforts. This calculation yields the average estimation effort completed per day. Next, the estimation effort of the whole projects is divided by this average, providing an estimate of how many days are needed to complete the all the sprint backlogs. Since days cannot have decimals, this result is rounded up. Finally, this number of days is added to the actual completion date of the just-ended sprint, which is calculated by start date plus the actual effort in days, yielding the estimated completion date.

6.5.8 Estimation



Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to register for an account by entering their information.	Function	Medium	Completed	3 day(s)	Estimate	tanengian
User shall be able to login into the system by entering their email and password.	Feature	High	Completed	4 day(s)	Estimate	wonghengkian
User shall be verified through 2-Factor authentication by sending the code through their email.	Security	Medium	Completed	7 day(s)	Estimate	joshuayii
User shall be able to reset their password if they forgot their login password.	Feature	Medium	Completed	7 day(s)	Estimate	tanengian

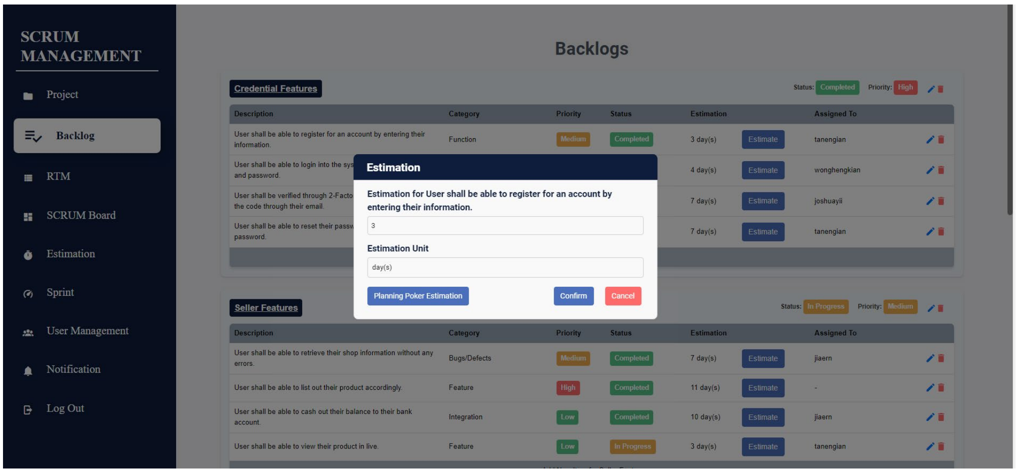
Figure 6.109: Sprint Backlog



Description	Category	Priority	Status	Estimation	Assigned To
User shall be able to register for an account by entering their information.	Function	Medium	Completed	3 day(s)	tanengian
User shall be able to login into the system by entering their email and password.	Feature	High	Completed	4 day(s)	wonghengkian
User shall be verified through 2-Factor authentication by sending the code through their email.	Security	Medium	Completed	7 day(s)	joshuayii
User shall be able to reset their password if they forgot their login password.	Feature	Medium	Completed	7 day(s)	tanengian

Figure 6.110: Sprint Backlog (Product Owner View)

The estimation effort of the sprint backlog items can be estimated by clicking on the "Estimation" button on the Backlog page. However, this button will only be available to SCRUM Master, Team Member and Creator.



Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to register for an account by entering their information.	Function	Medium	Completed	3 day(s)	Estimate	tanengian
User shall be able to login into the sys and password.				4 day(s)	Estimate	wonghengkian
User shall be verified through 2-Facto the code through their email.				7 day(s)	Estimate	joshuayii
User shall be able to reset their passw password.				7 day(s)	Estimate	tanengian

Estimation

Estimation for User shall be able to register for an account by entering their information.

3

Estimation Unit

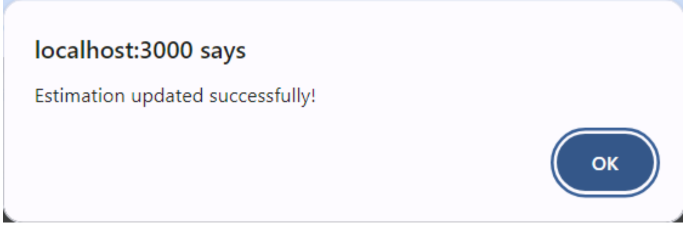
day(s)

Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	7 day(s)	Estimate	jiaern
User shall be able to list out their product accordingly.	Feature	High	Completed	11 day(s)	Estimate	-
User shall be able to cash out their balance to their bank account.	Integration	Low	Completed	10 day(s)	Estimate	jiaern
User shall be able to view their product in live.	Feature	Low	In Progress	3 day(s)	Estimate	tanengian

Figure 6.111: Estimation Modal

The web application offers users two options: Planning Poker Estimation or the default estimation.

For the default estimation, the process is straightforward. Users simply input their estimation effort and estimation unit, then click the "Confirm" button. The web application updates the estimation effort accordingly.

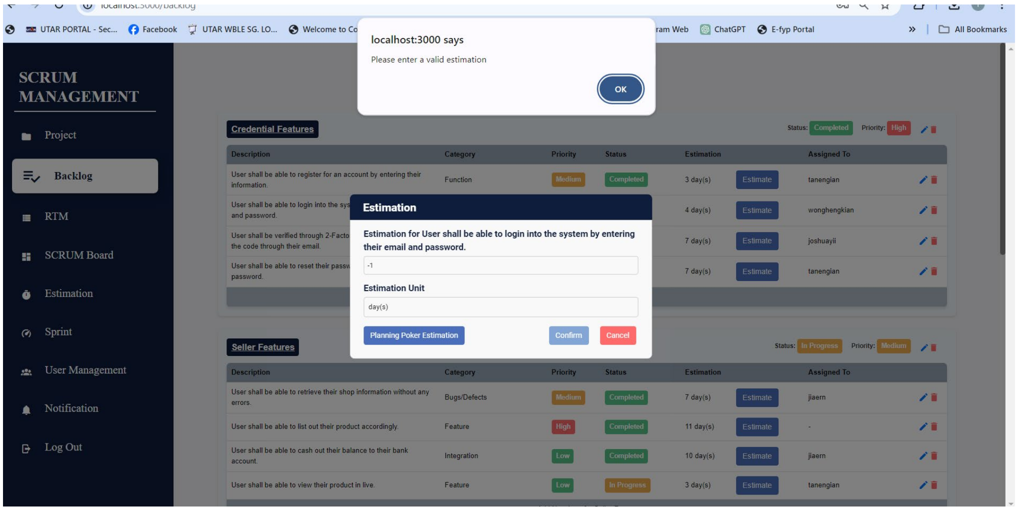


localhost:3000 says

Estimation updated successfully!

Figure 6.112: Successful Update Estimation Alert Message

Upon successfully updating the estimation effort, the web application will remind the user.



localhost:3000 says

Please enter a valid estimation

Estimation

Estimation for User shall be able to login into the system by entering their email and password.

-1

Estimation Unit

day(s)

Credential Features — Status: Completed Priority: High

Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to register for an account by entering their information.	Function	Medium	Completed	3 day(s)	Estimate	tanengian
User shall be able to login into the sys and password.				4 day(s)	Estimate	wonghengkian
User shall be verified through 2-Facto the code through their email.				7 day(s)	Estimate	joshuayii
User shall be able to reset their passw password.				7 day(s)	Estimate	tanengian

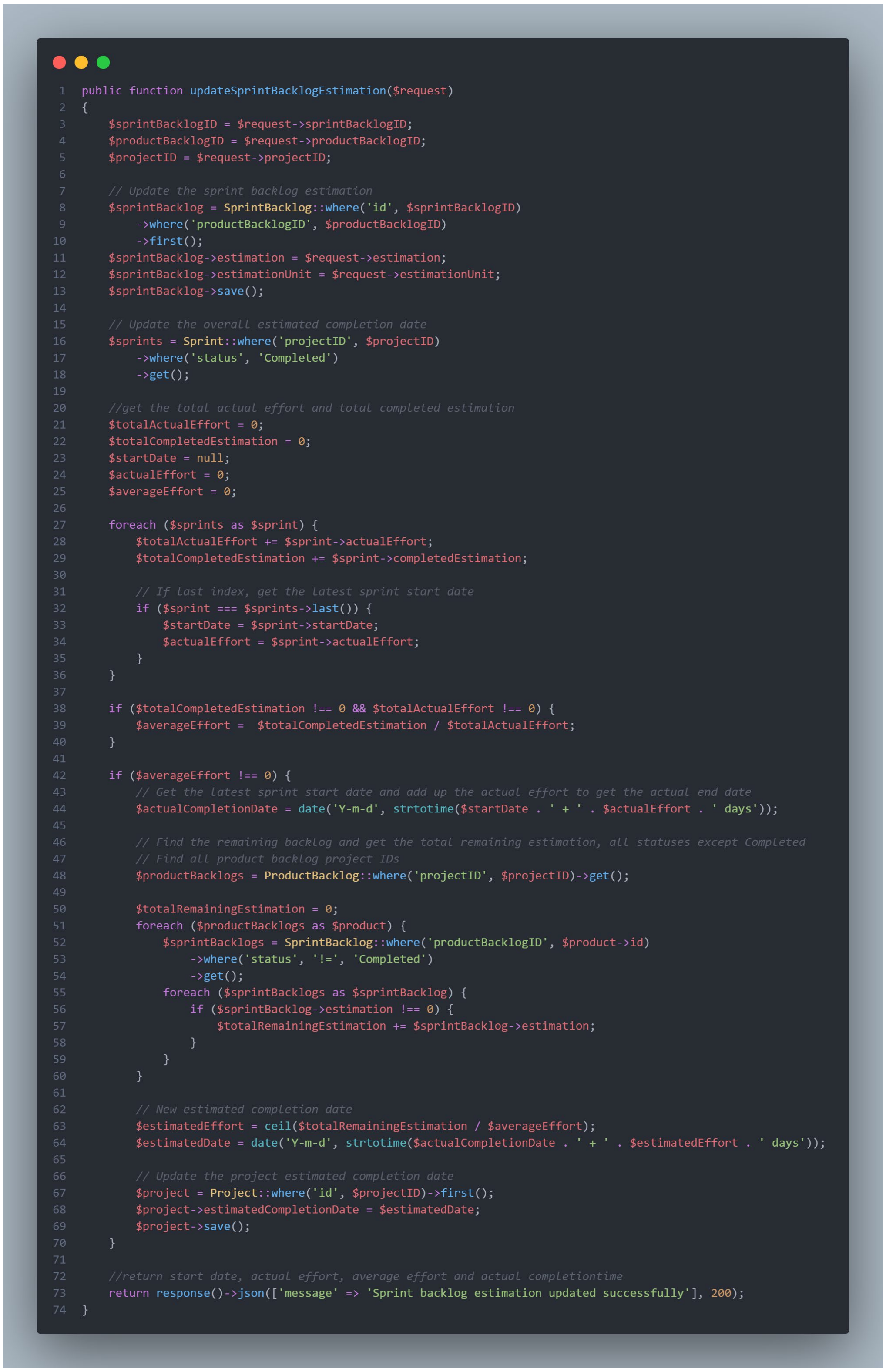
Seller Features — Status: In Progress Priority: Medium

Description	Category	Priority	Status	Estimation		Assigned To
User shall be able to retrieve their shop information without any errors.	Bugs/Defects	Medium	Completed	7 day(s)	Estimate	jiaern
User shall be able to list out their product accordingly.	Feature	High	Completed	11 day(s)	Estimate	-
User shall be able to cash out their balance to their bank account.	Integration	Low	Completed	10 day(s)	Estimate	jiaern
User shall be able to view their product in live.	Feature	Low	In Progress	3 day(s)	Estimate	tanengian

Figure 6.113: Estimation Effort Validation

If any of the user input has triggered the validation, an alert message will be shown to user. The following scenario has been validated:

1. Estimation Effort is less than 0
2. Estimation Unit is empty



```php
public function updateSprintBacklogEstimation($request)
{
    $sprintBacklogID = $request->sprintBacklogID;
    $productBacklogID = $request->productBacklogID;
    $projectID = $request->projectID;

    // Update the sprint backlog estimation
    $sprintBacklog = SprintBacklog::where('id', $sprintBacklogID)
        ->where('productBacklogID', $productBacklogID)
        ->first();
    $sprintBacklog->estimation = $request->estimation;
    $sprintBacklog->estimationUnit = $request->estimationUnit;
    $sprintBacklog->save();

    // Update the overall estimated completion date
    $sprints = Sprint::where('projectID', $projectID)
        ->where('status', 'Completed')
        ->get();

    //get the total actual effort and total completed estimation
    $totalActualEffort = 0;
    $totalCompletedEstimation = 0;
    $startDate = null;
    $actualEffort = 0;
    $averageEffort = 0;

    foreach ($sprints as $sprint) {
        $totalActualEffort += $sprint->actualEffort;
        $totalCompletedEstimation += $sprint->completedEstimation;

        // If last index, get the latest sprint start date
        if ($sprint === $sprints->last()) {
            $startDate = $sprint->startDate;
            $actualEffort = $sprint->actualEffort;
        }
    }

    if ($totalCompletedEstimation !== 0 && $totalActualEffort !== 0) {
        $averageEffort =  $totalCompletedEstimation / $totalActualEffort;
    }

    if ($averageEffort !== 0) {
        // Get the latest sprint start date and add up the actual effort to get the actual end date
        $actualCompletionDate = date('Y-m-d', strtotime($startDate . ' + ' . $actualEffort . ' days'));

        // Find the remaining backlog and get the total remaining estimation, all statuses except Completed
        // Find all product backlog project IDs
        $productBacklogs = ProductBacklog::where('projectID', $projectID)->get();

        $totalRemainingEstimation = 0;
        foreach ($productBacklogs as $product) {
            $sprintBacklogs = SprintBacklog::where('productBacklogID', $product->id)
                ->where('status', '!=', 'Completed')
                ->get();
            foreach ($sprintBacklogs as $sprintBacklog) {
                if ($sprintBacklog->estimation !== 0) {
                    $totalRemainingEstimation += $sprintBacklog->estimation;
                }
            }
        }

        // New estimated completion date
        $estimatedEffort = ceil($totalRemainingEstimation / $averageEffort);
        $estimatedDate = date('Y-m-d', strtotime($actualCompletionDate . ' + ' . $estimatedEffort . ' days'));

        // Update the project estimated completion date
        $project = Project::where('id', $projectID)->first();
        $project->estimatedCompletionDate = $estimatedDate;
        $project->save();
    }

    //return start date, actual effort, average effort and actual completiontime
    return response()->json(['message' => 'Sprint backlog estimation updated successfully'], 200);
}
```

Figure 6.114: Update Sprint Backlog Estimation Function (Laravel)

Upon updating the estimation effort, a recalculation of the estimated completion dateline is required since the total estimation effort has been modified. The algorithm used for this recalculation is the similar as the one used to end a sprint.

The algorithm first checks if there are any estimation efforts that are not marked as "Completed." If there are, recalculation begins; otherwise, the estimated completion date for the latest ended sprint becomes the new estimated completion date.

The algorithm starts by computing the total available estimation effort across the project that is not marked as "Completed." Then, it determines the user's accumulative pace by dividing the total estimation effort completed in the project by the total actual effort used to complete these efforts. This calculation yields the average estimation effort completed per day. Next, the estimation effort of the whole projects is divided by this average, providing an estimate of how many days are needed to complete the all the sprint backlogs. Since days cannot have decimals, this result is rounded up. Finally, this number of days is added to the actual completion date of the latest-ended sprint, which is calculated by start date plus the actual effort in days, yielding the estimated completion date.

6.5.8.1 Planning Poker

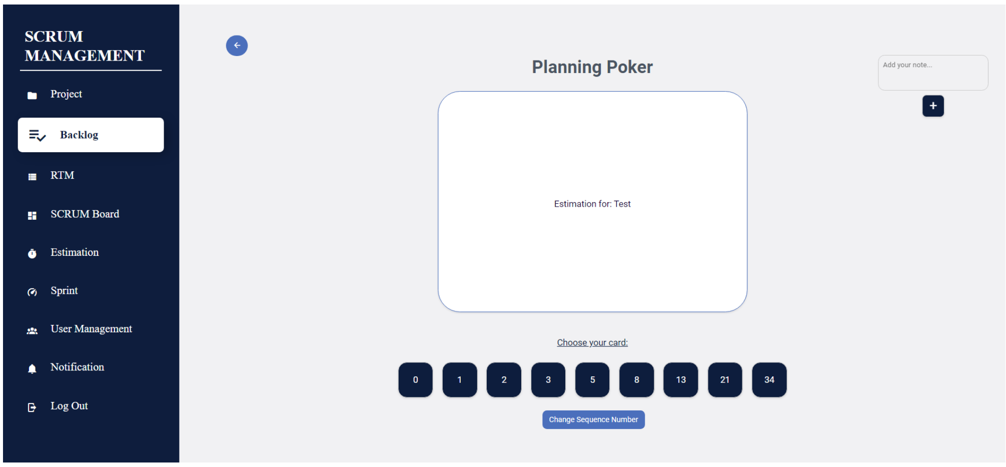
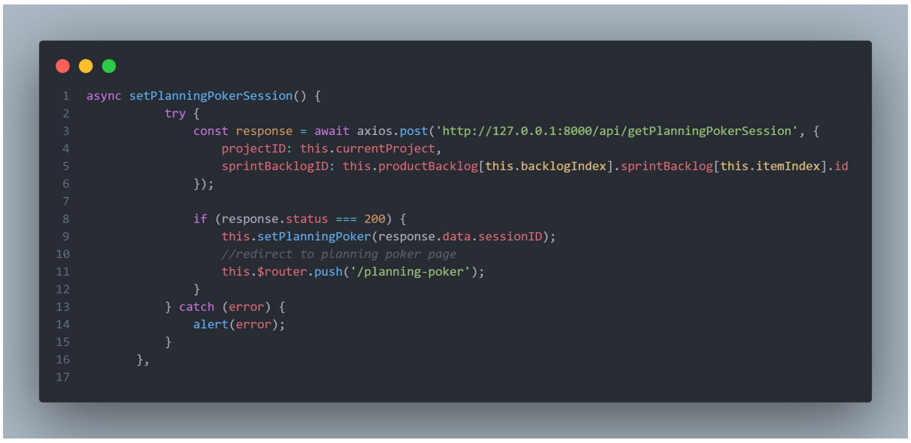


Figure 6.115: Planning Poker Page

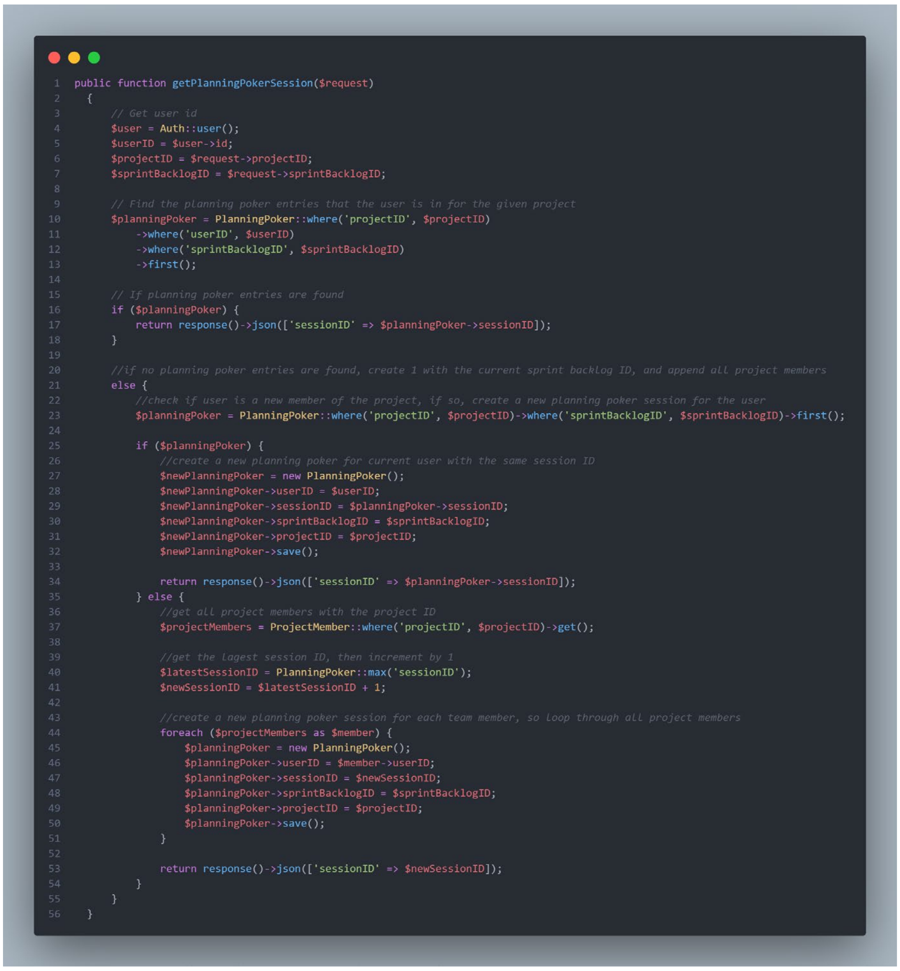
Planning Poker is a collaborative game designed to encourage discussion among project members, allowing them to estimate the effort required for specific tasks together, such as sprint backlog. Since the estimation model heavily rely on user input, collaborative games like Planning Poker can enhance the accuracy of estimation efforts, thereby improving the overall accuracy of the estimated completion date.

In this web application, Planning Poker sessions will only be available for SCRUM Masters, Team Members, and creators to initiate. However, all project members are invited to participate in the game, contributing to more accurate estimations.



```
async setPlanningPokerSession() {
           try {
               const response = await axios.post('http://127.0.0.1:8000/api/getPlanningPokerSession', {
                   projectID: this.currentProject,
                   sprintBacklogID: this.productBacklog[this.backlogIndex].sprintBacklog[this.itemIndex].id
               });

               if (response.status === 200) {
                   this.setPlanningPoker(response.data.sessionID);
                   //redirect to planning poker page
                   this.$router.push('/planning-poker');
               }
           } catch (error) {
               alert(error);
           }
       },
```

Figure 6.116: Set Planning Poker Session (Vue.js)

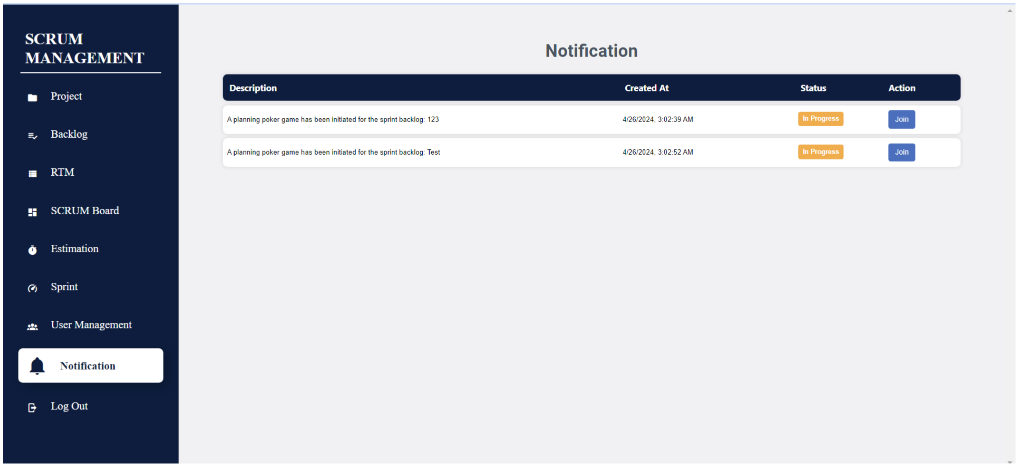


```
public function getPlanningPokerSession($request)
  {
      // Get user id
      $user = Auth::user();
      $userID = $user->id;
      $projectID = $request->projectID;
      $sprintBacklogID = $request->sprintBacklogID;

      // Find the planning poker entries that the user is in for the given project
      $planningPoker = PlanningPoker::where('projectID', $projectID)
          ->where('userID', $userID)
          ->where('sprintBacklogID', $sprintBacklogID)
          ->first();

      // If planning poker entries are found
      if ($planningPoker) {
          return response()->json(['sessionID' => $planningPoker->sessionID]);
      }

      //if no planning poker entries are found, create 1 with the current sprint backlog ID, and append all project members
      else {
          //check if user is a new member of the project, if so, create a new planning poker session for the user
          $planningPoker = PlanningPoker::where('projectID', $projectID)->where('sprintBacklogID', $sprintBacklogID)->first();

          if ($planningPoker) {
              //create a new planning poker for current user with the same session ID
              $newPlanningPoker = new PlanningPoker();
              $newPlanningPoker->userID = $userID;
              $newPlanningPoker->sessionID = $planningPoker->sessionID;
              $newPlanningPoker->sprintBacklogID = $sprintBacklogID;
              $newPlanningPoker->projectID = $projectID;
              $newPlanningPoker->save();

              return response()->json(['sessionID' => $planningPoker->sessionID]);
          } else {
              //get all project members with the project ID
              $projectMembers = ProjectMember::where('projectID', $projectID)->get();

              //get the lagest session ID, then increment by 1
              $latestSessionID = PlanningPoker::max('sessionID');
              $newSessionID = $latestSessionID + 1;

              //create a new planning poker session for each team member, so loop through all project members
              foreach ($projectMembers as $member) {
                  $planningPoker = new PlanningPoker();
                  $planningPoker->userID = $member->userID;
                  $planningPoker->sessionID = $newSessionID;
                  $planningPoker->sprintBacklogID = $sprintBacklogID;
                  $planningPoker->projectID = $projectID;
                  $planningPoker->save();
              }

              return response()->json(['sessionID' => $newSessionID]);
          }
      }
  }
```

Figure 6.117: Get Planning Poker Session (Laravel)

When a user selects the Planning Poker Estimation for a particular sprint backlog item, Vue.js will set the Planning Poker session using Vuex and redirect the user to the Planning Poker page.

In Laravel, the application first checks if a Planning Poker session ID exists for the initiated sprint backlog item. If found, the session ID will be returned. If not found, it creates a new session and invites all members. However, there are scenarios where newly added members are invited after the Planning Poker session has been initiated. In such cases, the new member will be added to the existing session.

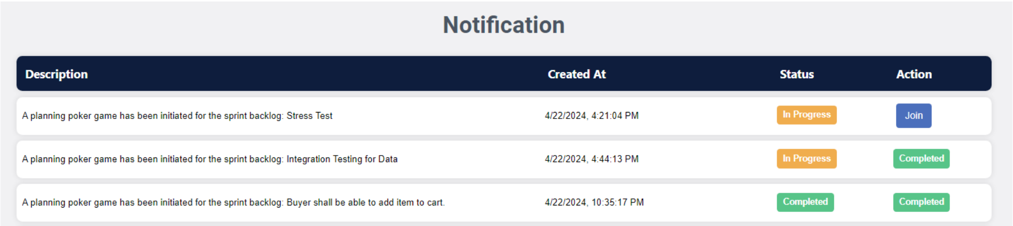


Notification

Description	Created At	Status	Action
A planning poker game has been initiated for the sprint backlog: 123	4/26/2024, 3:02:39 AM	In Progress	Join
A planning poker game has been initiated for the sprint backlog: Test	4/26/2024, 3:02:52 AM	In Progress	Join

Figure 6.118: Notification Page

All users involved in the project will receive a notification for the Planning Poker session, allowing them to join directly through the notification.



Notification

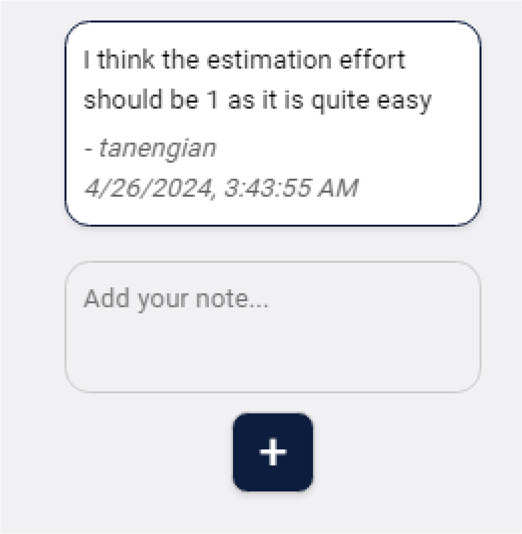
Description	Created At	Status	Action
A planning poker game has been initiated for the sprint backlog: Stress Test	4/22/2024, 4:21:04 PM	In Progress	Join
A planning poker game has been initiated for the sprint backlog: Integration Testing for Data	4/22/2024, 4:44:13 PM	In Progress	Completed
A planning poker game has been initiated for the sprint backlog: Buyer shall be able to add item to cart.	4/22/2024, 10:35:17 PM	Completed	Completed

Figure 6.119: Notification Variable

The notifications will have three different types:

1. Status In Progress Action Join: This indicates that the user has not yet participated in the session and can join.
2. Status In Progress Action Completed: This indicates that the user has already chosen their desired estimation effort for the sprint backlog.

3. Status Completed Action Completed: This indicates that the Planning Poker session has ended.



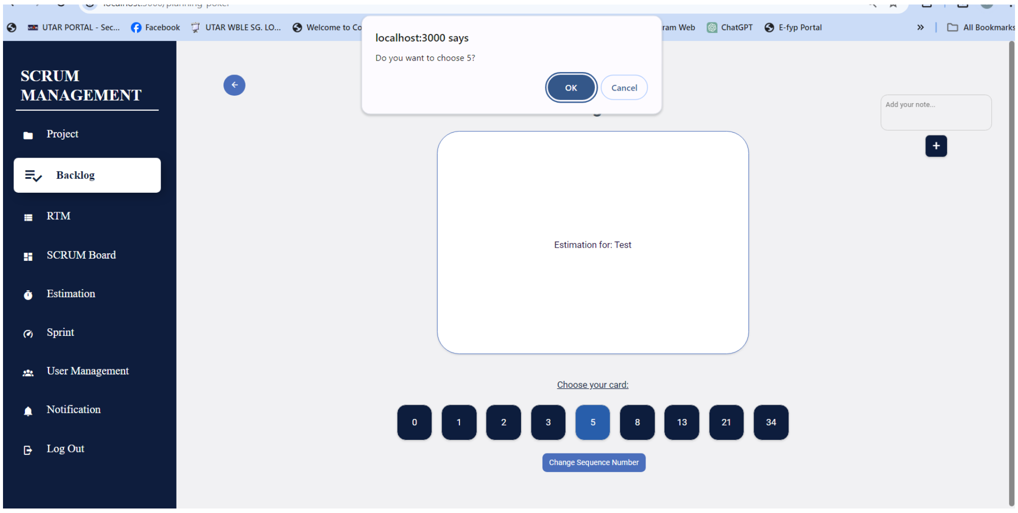
I think the estimation effort should be 1 as it is quite easy

- tanengian

4/26/2024, 3:43:55 AM

Figure 6.120: Sticky Notes

On the side of the Planning Poker Page, the web application will allow users to communicate with each other with sticky notes.

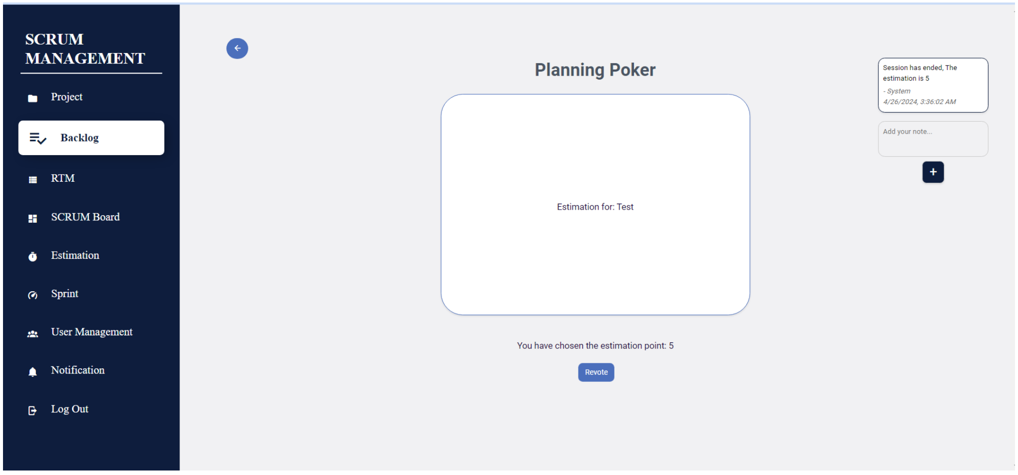


localhost:3000 says

Do you want to choose 5?

Figure 6.121: Planning Poker Page

Each user of the project is required to choose an estimation effort from the sequence cards provided. The web application will prompt the user to confirm their choice. The default sequence card will be Fibonacci Numbers, which are commonly used in Planning Poker.



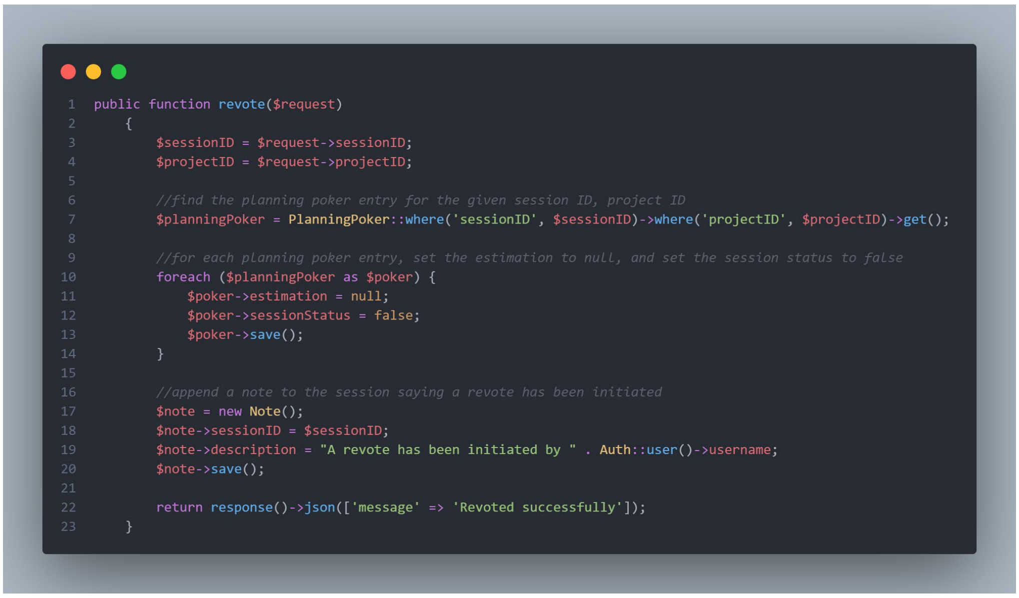
Session has ended, The estimation is 5
- System
4/26/2024, 3:36:02 AM

Estimation for: Test

You have chosen the estimation point: 5

Figure 6.122: Planning Poker Session Ended

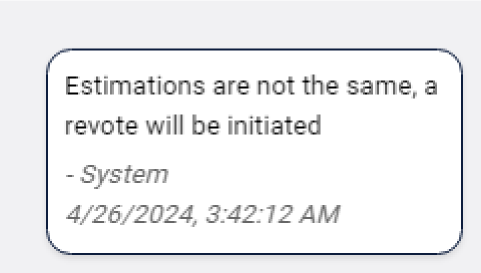
When all users within the project select the same estimation effort, the sprint backlog item will be updated with the agreed-upon estimation effort. The estimated completion date will also be updated accordingly, using the same API to update the estimation effort as the default estimation. A revote button will also be presented to the SCRUM Master, Team Member and creator to reinitiate the session.



```php
public function revote($request)
    {
        $sessionID = $request->sessionID;
        $projectID = $request->projectID;

        //find the planning poker entry for the given session ID, project ID
        $planningPoker = PlanningPoker::where('sessionID', $sessionID)->where('projectID', $projectID)->get();

        //for each planning poker entry, set the estimation to null, and set the session status to false
        foreach ($planningPoker as $poker) {
            $poker->estimation = null;
            $poker->sessionStatus = false;
            $poker->save();
        }

        //append a note to the session saying a revote has been initiated
        $note = new Note();
        $note->sessionID = $sessionID;
        $note->description = "A revote has been initiated by " . Auth::user()->username;
        $note->save();

        return response()->json(['message' => 'Revoted successfully']);
    }
```

Figure 6.123: Revote Function (Laravel)

The revote function will reset all session details to their default state, as if it were a new session. However, the sticky notes discussions will remain unchanged.



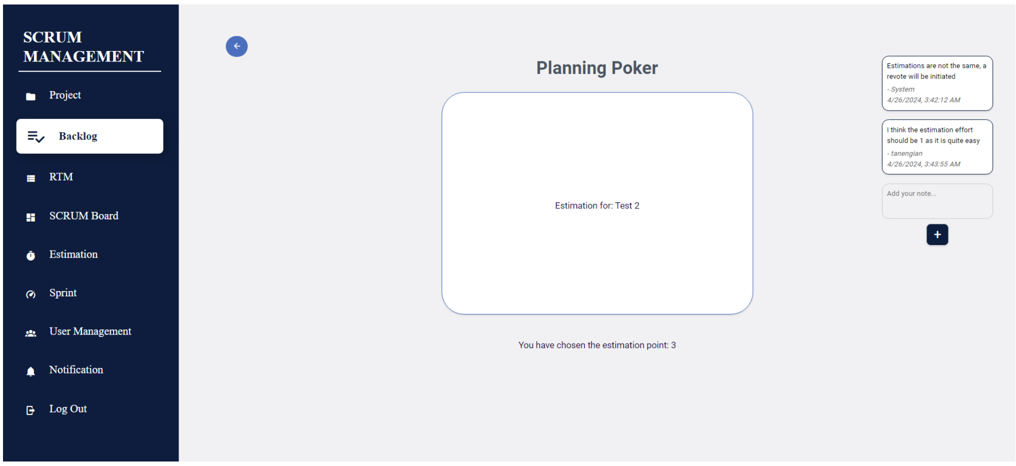
Estimations are not the same, a revote will be initiated

- System

4/26/2024, 3:42:12 AM

Figure 6.124: Different Estimation Chosen

If any of the project members choose a different estimation effort, the system will auto initiate a revote.



Estimation for: Test 2

You have chosen the estimation point: 3

Estimations are not the same, a revote will be initiated

- System

4/26/2024, 3:42:12 AM

I think the estimation effort should be 1 as it is quite easy

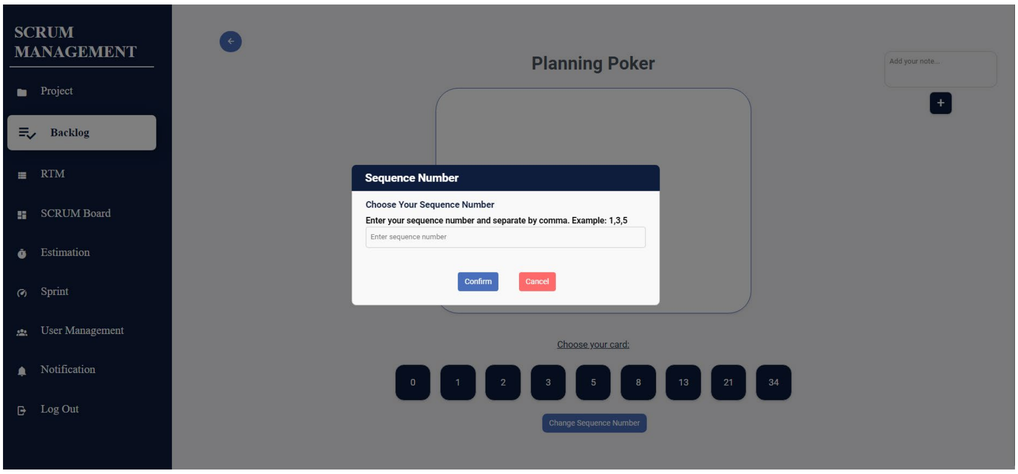
- tanengian

4/26/2024, 3:43:55 AM

Add your note...

Figure 6.125: Planning Poker Page

If a user has already finished choosing their estimation effort but the session has not ended, they will be restricted from making further changes and actions, except for participating in the sticky note discussion.



Sequence Number

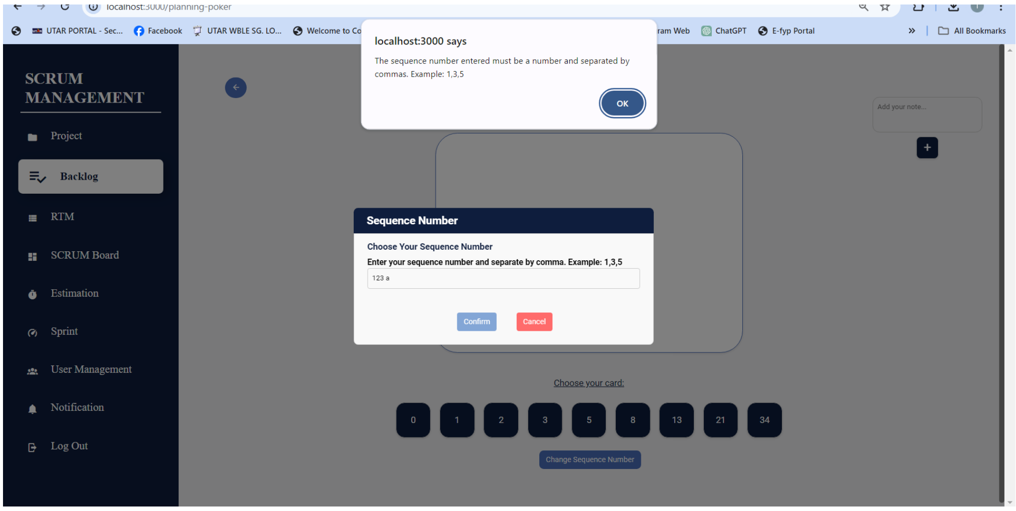
Choose Your Sequence Number

Enter your sequence number and separate by comma. Example: 1,3,5

Enter sequence number

Figure 6.126: Change Sequence Number

Alternatively, SCRUM Master, Team Member and creator can customize the sequence number by clicking on the “Change Sequence Number” button.



localhost:3000 says

The sequence number entered must be a number and separated by commas. Example: 1,3,5

OK

Sequence Number

Choose Your Sequence Number

Enter your sequence number and separate by comma. Example: 1,3,5

123 a

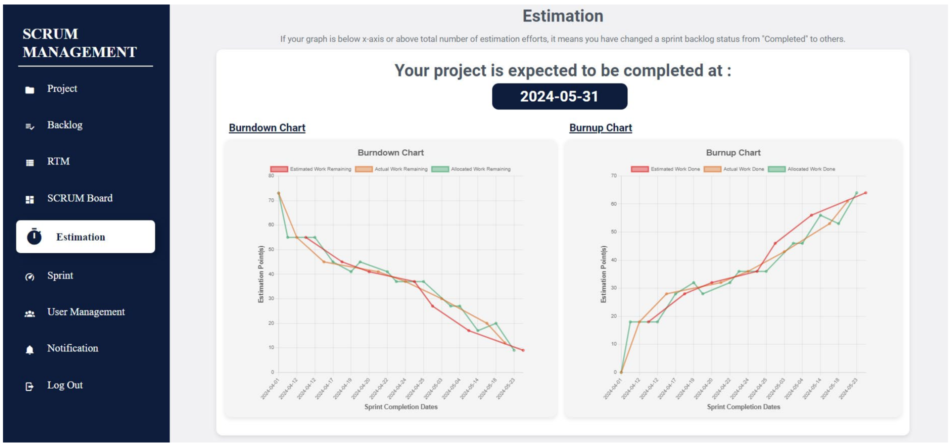
Confirm Cancel

Figure 6.127: Sequence Number Validation

If any of the user input has triggered the validation, an alert message will be shown to user. The following scenario has been validated:

1. Sequence number entered is not a number
2. Sequence number is not separated by comma
3. Sequence number entered is empty

6.5.9 Project Completion Dateline Estimation



Estimation

If your graph is below x-axis or above total number of estimation efforts, it means you have changed a sprint backlog status from "Completed" to others.

Your project is expected to be completed at :

2024-05-31

Figure 6.128: Overall Burndown and Burnup Chart

The web application will provide users with the estimated completion dateline of their project in the estimation page, which will be updated in two different scenarios as previously mentioned:

1. When a sprint has been ended.
2. When an estimation effort has been modified.

Additionally, the web application will offer users an overall burndown chart and burnup chart. These charts will provide users with visualizations of the timeline of their SCRUM projects.

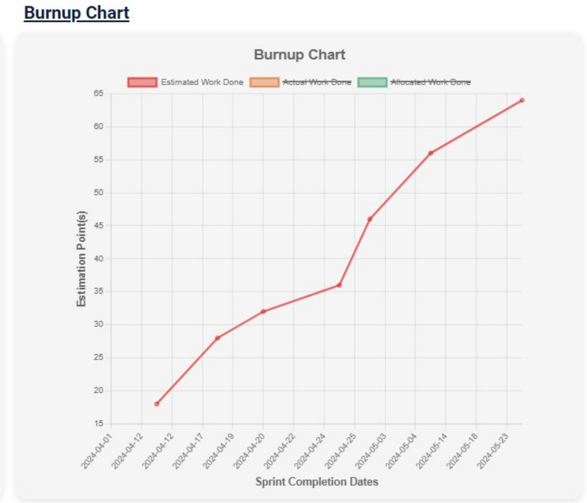


Figure 6.129: Overall Burnup Chart

For each graph plotted on the Estimation page, users can choose to hide specific lines to focus on other lines more easily.

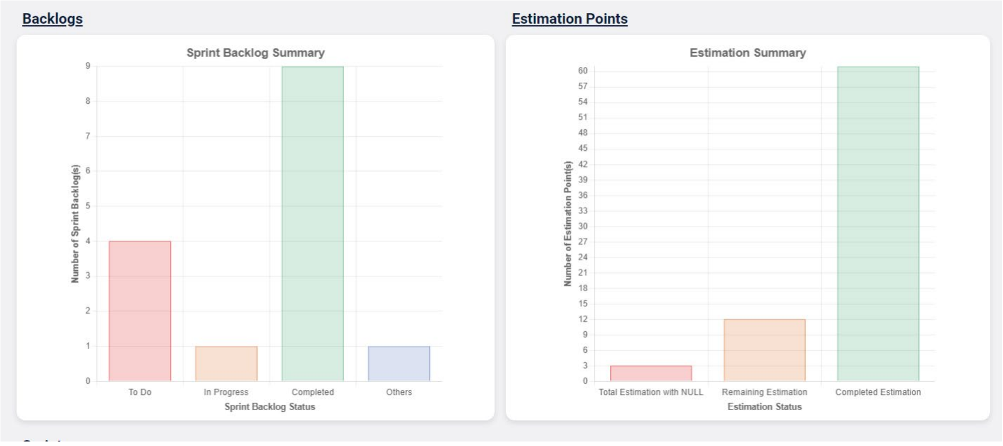
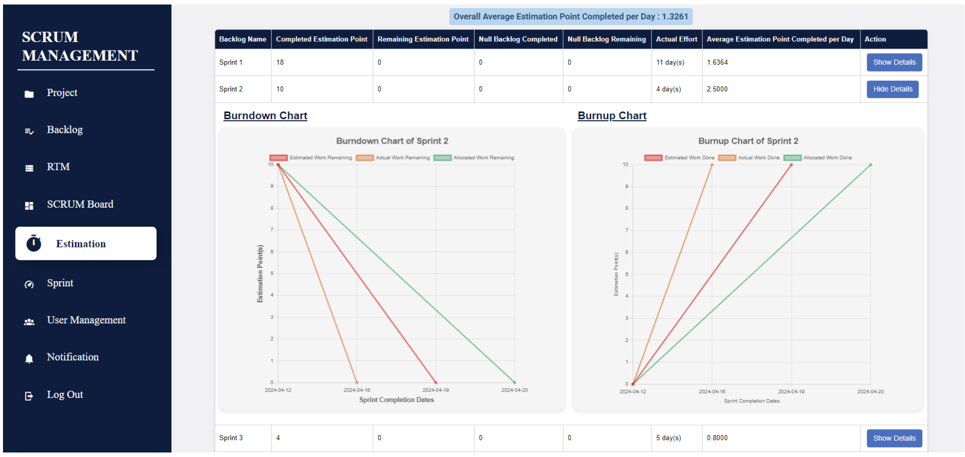


Figure 6.130: Summary Graphs

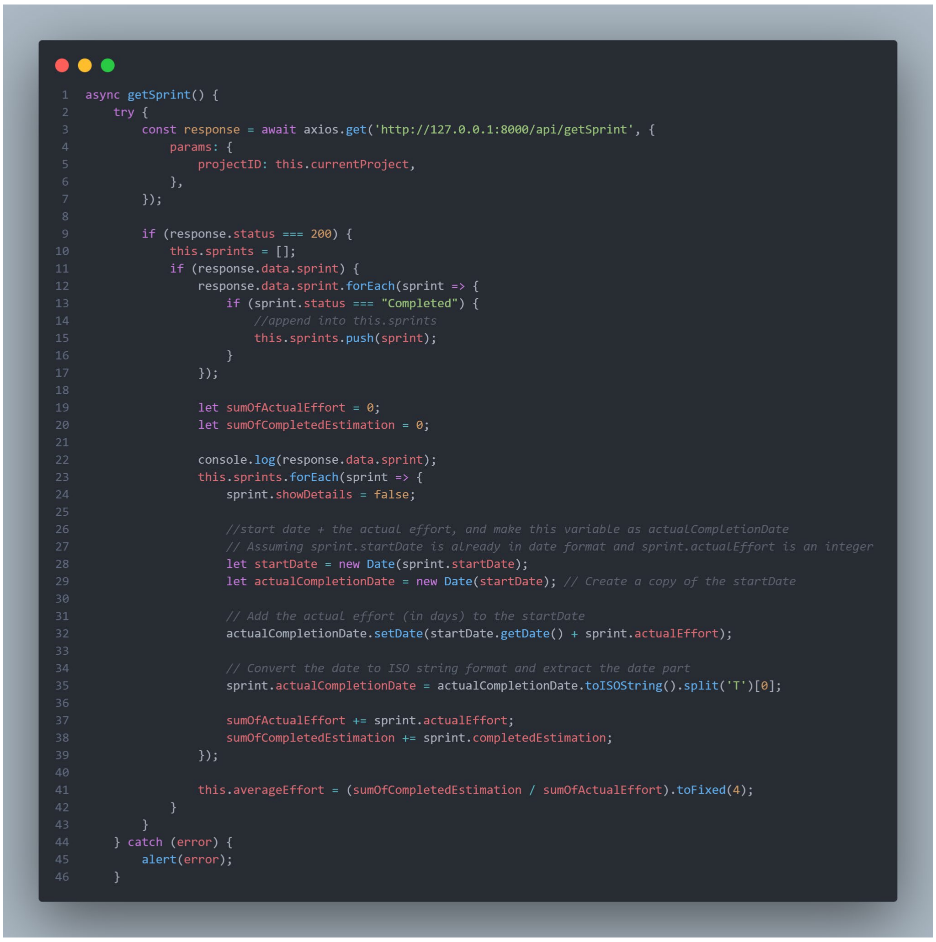
The web application will further enhance the visibility of the project timeline by incorporating summary graphs. These graphs will summarize all the sprint backlog items and their estimation effort statuses, indicating how many have been completed, how many are still pending and how many are in null.



Overall Average Estimation Point Completed per Day : 1.3261

Backlog Name	Completed Estimation Point	Remaining Estimation Point	Null Backlog Completed	Null Backlog Remaining	Actual Effort	Average Estimation Point Completed per Day	Action
Sprint 1	18	0	0	0	11 day(s)	1.6364	Show Details
Sprint 2	10	0	0	0	4 day(s)	2.5000	Hide Details
Sprint 3	4	0	0	0	5 day(s)	0.8000	Show Details

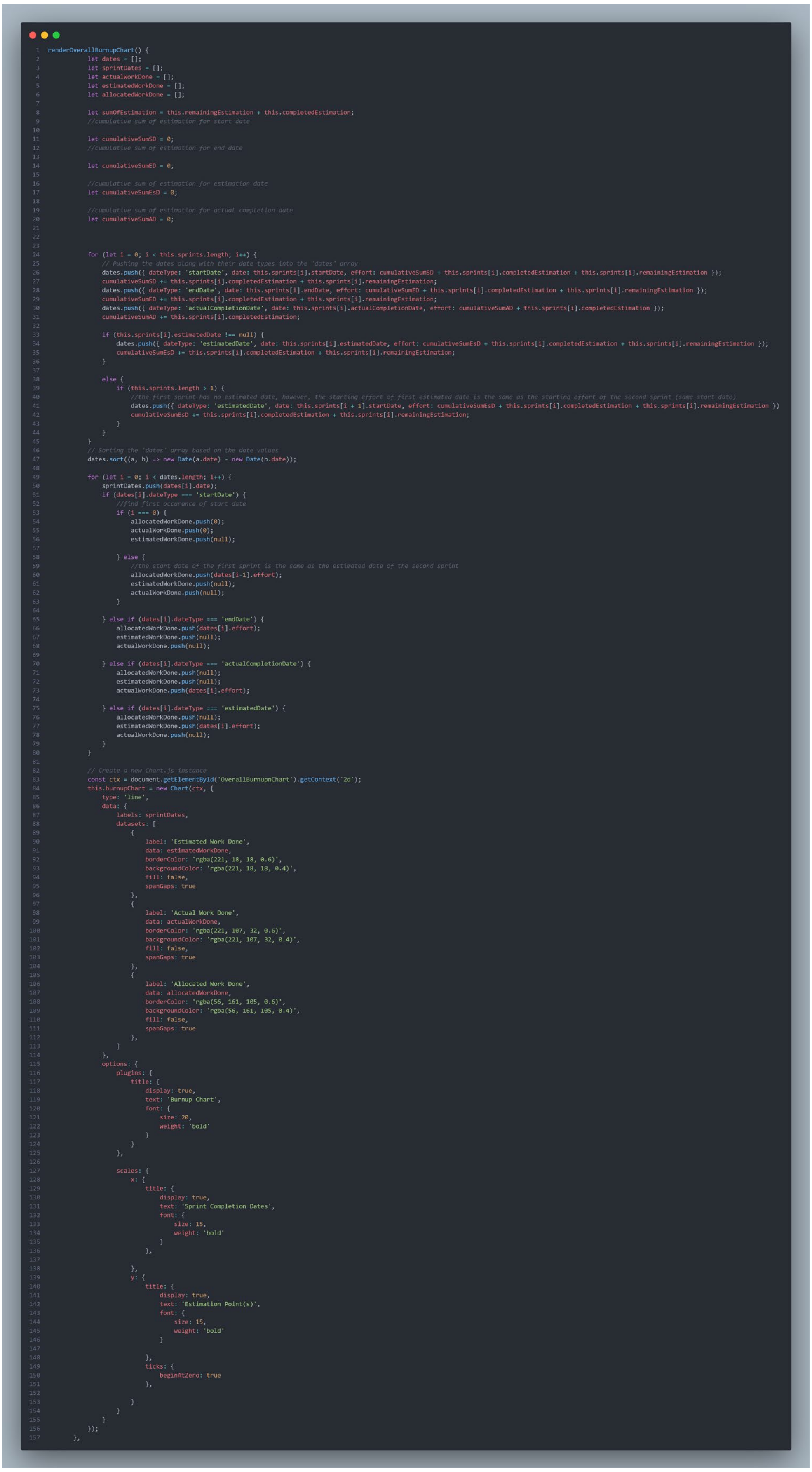
Figure 6.131: Sprint Summary



```
async getSprint() {
    try {
        const response = await axios.get('http://127.0.0.1:8000/api/getSprint', {
            params: {
                projectID: this.currentProject,
            },
        });

        if (response.status === 200) {
            this.sprints = [];
            if (response.data.sprint) {
                response.data.sprint.forEach(sprint => {
                    if (sprint.status === "Completed") {
                        //append into this.sprints
                        this.sprints.push(sprint);
                    }
                });

                let sumOfActualEffort = 0;
                let sumOfCompletedEstimation = 0;

                console.log(response.data.sprint);
                this.sprints.forEach(sprint => {
                    sprint.showDetails = false;

                    //start date + the actual effort, and make this variable as actualCompletionDate
                    // Assuming sprint.startDate is already in date format and sprint.actualEffort is an integer
                    let startDate = new Date(sprint.startDate);
                    let actualCompletionDate = new Date(startDate); // Create a copy of the startDate

                    // Add the actual effort (in days) to the startDate
                    actualCompletionDate.setDate(startDate.getDate() + sprint.actualEffort);

                    // Convert the date to ISO string format and extract the date part
                    sprint.actualCompletionDate = actualCompletionDate.toISOString().split('T')[0];

                    sumOfActualEffort += sprint.actualEffort;
                    sumOfCompletedEstimation += sprint.completedEstimation;
                });

                this.averageEffort = (sumOfCompletedEstimation / sumOfActualEffort).toFixed(4);
            }
        }
    } catch (error) {
        alert(error);
    }
```

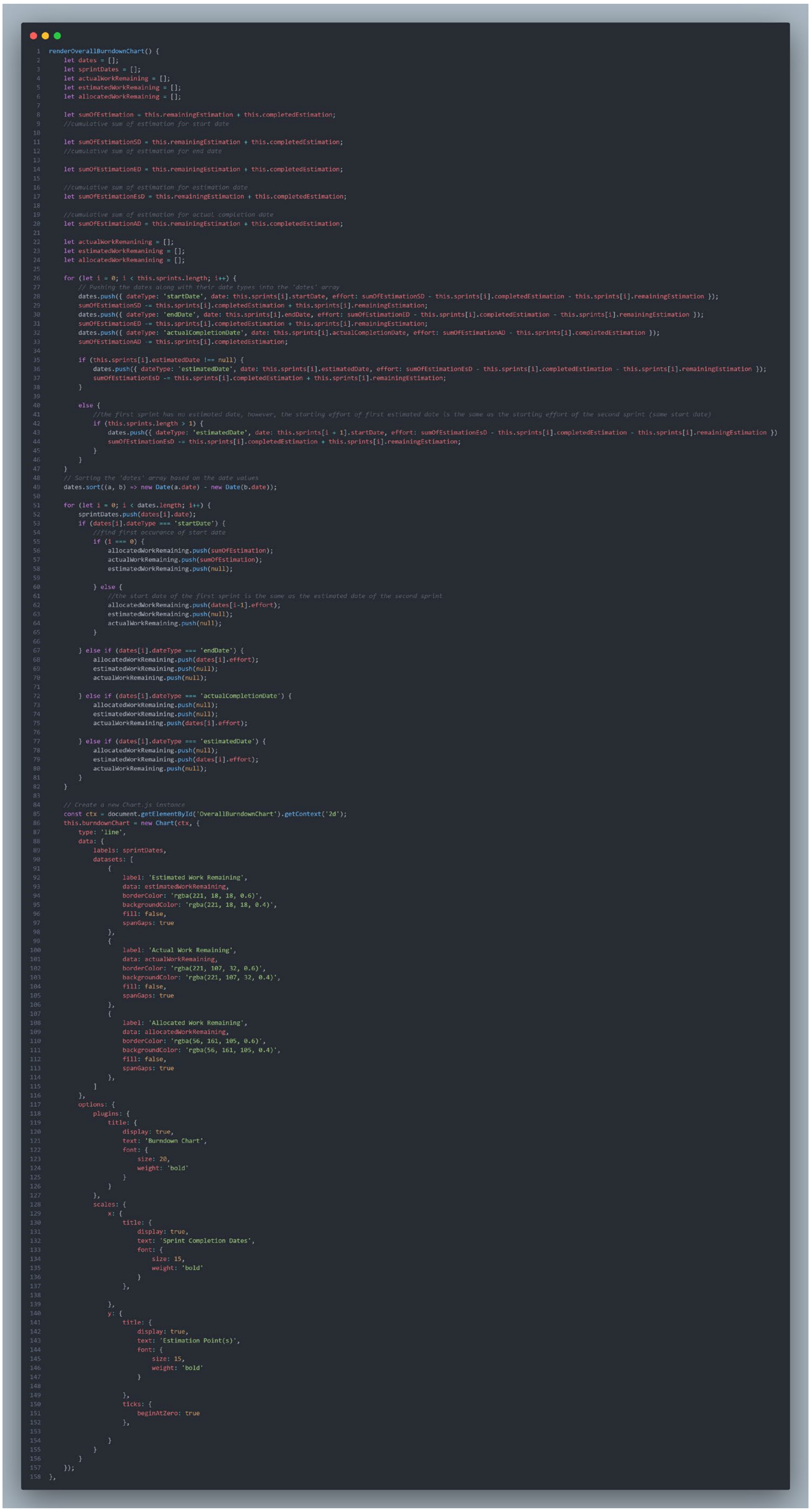
Figure 6.132: Get Sprint Summary Function (Vue.js)

Finally, a summary of sprints will be provided to the user, including individual burndown and burnup charts for each sprint. The page will also inform the user about the pace of the project, which is calculated by dividing the total estimation effort completed by the total actual effort expended.



```
renderOverallBurnupChart() {
            let dates = [];
            let sprintDates = [];
            let actualWorkDone = [];
            let estimatedWorkDone = [];
            let allocatedWorkDone = [];

            let sumOfEstimation = this.remainingEstimation + this.completedEstimation;
            //cumulative sum of estimation for start date

            let cumulativeSumSD = 0;
            //cumulative sum of estimation for end date

            let cumulativeSumED = 0;

            //cumulative sum of estimation for estimation date
            let cumulativeSumEsD = 0;

            //cumulative sum of estimation for actual completion date
            let cumulativeSumAD = 0;



            for (let i = 0; i < this.sprints.length; i++) {
                // Pushing the dates along with their date types into the 'dates' array
                dates.push({ dateType: 'startDate', date: this.sprints[i].startDate, effort: cumulativeSumSD + this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation });
                cumulativeSumSD += this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation;
                dates.push({ dateType: 'endDate', date: this.sprints[i].endDate, effort: cumulativeSumED + this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation });
                cumulativeSumED += this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation;
                dates.push({ dateType: 'actualCompletionDate', date: this.sprints[i].actualCompletionDate, effort: cumulativeSumAD + this.sprints[i].completedEstimation });
                cumulativeSumAD += this.sprints[i].completedEstimation;

                if (this.sprints[i].estimatedDate !== null) {
                    dates.push({ dateType: 'estimatedDate', date: this.sprints[i].estimatedDate, effort: cumulativeSumEsD + this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation });
                    cumulativeSumEsD += this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation;
                }

                else {
                    if (this.sprints.length > 1) {
                        //the first sprint has no estimated date, however, the starting effort of first estimated date is the same as the starting effort of the second sprint (same start date)
                        dates.push({ dateType: 'estimatedDate', date: this.sprints[i + 1].startDate, effort: cumulativeSumEsD + this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation })
                        cumulativeSumEsD += this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation;
                    }
                }
            }
            // Sorting the 'dates' array based on the date values
            dates.sort((a, b) => new Date(a.date) - new Date(b.date));

            for (let i = 0; i < dates.length; i++) {
                sprintDates.push(dates[i].date);
                if (dates[i].dateType === 'startDate') {
                    //find first occurance of start date
                    if (i === 0) {
                        allocatedWorkDone.push(0);
                        actualWorkDone.push(0);
                        estimatedWorkDone.push(null);

                    } else {
                        //the start date of the first sprint is the same as the estimated date of the second sprint
                        allocatedWorkDone.push(dates[i-1].effort);
                        estimatedWorkDone.push(null);
                        actualWorkDone.push(null);
                    }

                } else if (dates[i].dateType === 'endDate') {
                    allocatedWorkDone.push(dates[i].effort);
                    estimatedWorkDone.push(null);
                    actualWorkDone.push(null);

                } else if (dates[i].dateType === 'actualCompletionDate') {
                    allocatedWorkDone.push(null);
                    estimatedWorkDone.push(null);
                    actualWorkDone.push(dates[i].effort);

                } else if (dates[i].dateType === 'estimatedDate') {
                    allocatedWorkDone.push(null);
                    estimatedWorkDone.push(dates[i].effort);
                    actualWorkDone.push(null);
                }
            }

            // Create a new Chart.js instance
            const ctx = document.getElementById('OverallBurnupnChart').getContext('2d');
            this.burnupChart = new Chart(ctx, {
                type: 'line',
                data: {
                    labels: sprintDates,
                    datasets: [
                        {
                            label: 'Estimated Work Done',
                            data: estimatedWorkDone,
                            borderColor: 'rgba(221, 18, 18, 0.6)',
                            backgroundColor: 'rgba(221, 18, 18, 0.4)',
                            fill: false,
                            spanGaps: true
                        },
                        {
                            label: 'Actual Work Done',
                            data: actualWorkDone,
                            borderColor: 'rgba(221, 107, 32, 0.6)',
                            backgroundColor: 'rgba(221, 107, 32, 0.4)',
                            fill: false,
                            spanGaps: true
                        },
                        {
                            label: 'Allocated Work Done',
                            data: allocatedWorkDone,
                            borderColor: 'rgba(56, 161, 105, 0.6)',
                            backgroundColor: 'rgba(56, 161, 105, 0.4)',
                            fill: false,
                            spanGaps: true
                        },
                    ]
                },
                options: {
                    plugins: {
                        title: {
                            display: true,
                            text: 'Burnup Chart',
                            font: {
                                size: 20,
                                weight: 'bold'
                            }
                        }
                    },

                    scales: {
                        x: {
                            title: {
                                display: true,
                                text: 'Sprint Completion Dates',
                                font: {
                                    size: 15,
                                    weight: 'bold'
                                }
                            },

                        },
                        y: {
                            title: {
                                display: true,
                                text: 'Estimation Point(s)',
                                font: {
                                    size: 15,
                                    weight: 'bold'
                                }

                            },
                            ticks: {
                                beginAtZero: true
                            },

                        }
                    }
                }
            });
        },
```

Figure 6.133: Plot Overall Burnup Chart Function (Vue.js)



```
renderOverallBurndownChart() {
    let dates = [];
    let sprintDates = [];
    let actualWorkRemaining = [];
    let estimatedWorkRemaining = [];
    let allocatedWorkRemaining = [];

    let sumOfEstimation = this.remainingEstimation + this.completedEstimation;
    //cumulative sum of estimation for start date

    let sumOfEstimationSD = this.remainingEstimation + this.completedEstimation;
    //cumulative sum of estimation for end date

    let sumOfEstimationED = this.remainingEstimation + this.completedEstimation;

    //cumulative sum of estimation for estimation date
    let sumOfEstimationEsD = this.remainingEstimation + this.completedEstimation;

    //cumulative sum of estimation for actual completion date
    let sumOfEstimationAD = this.remainingEstimation + this.completedEstimation;

    let actualWorkRemanining = [];
    let estimatedWorkRemanining = [];
    let allocatedWorkRemanining = [];

    for (let i = 0; i < this.sprints.length; i++) {
        // Pushing the dates along with their date types into the 'dates' array
        dates.push({ dateType: 'startDate', date: this.sprints[i].startDate, effort: sumOfEstimationSD - this.sprints[i].completedEstimation - this.sprints[i].remainingEstimation });
        sumOfEstimationSD -= this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation;
        dates.push({ dateType: 'endDate', date: this.sprints[i].endDate, effort: sumOfEstimationED - this.sprints[i].completedEstimation - this.sprints[i].remainingEstimation });
        sumOfEstimationED -= this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation;
        dates.push({ dateType: 'actualCompletionDate', date: this.sprints[i].actualCompletionDate, effort: sumOfEstimationAD - this.sprints[i].completedEstimation });
        sumOfEstimationAD -= this.sprints[i].completedEstimation;

        if (this.sprints[i].estimatedDate !== null) {
            dates.push({ dateType: 'estimatedDate', date: this.sprints[i].estimatedDate, effort: sumOfEstimationEsD - this.sprints[i].completedEstimation - this.sprints[i].remainingEstimation });
            sumOfEstimationEsD -= this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation;
        }

        else {
            //the first sprint has no estimated date, however, the starting effort of first estimated date is the same as the starting effort of the second sprint (same start date)
            if (this.sprints.length > 1) {
                dates.push({ dateType: 'estimatedDate', date: this.sprints[i + 1].startDate, effort: sumOfEstimationEsD - this.sprints[i].completedEstimation - this.sprints[i].remainingEstimation })
                sumOfEstimationEsD -= this.sprints[i].completedEstimation + this.sprints[i].remainingEstimation;
            }
        }
    }
    // Sorting the 'dates' array based on the date values
    dates.sort((a, b) => new Date(a.date) - new Date(b.date));

    for (let i = 0; i < dates.length; i++) {
        sprintDates.push(dates[i].date);
        if (dates[i].dateType === 'startDate') {
            //find first occurance of start date
            if (i === 0) {
                allocatedWorkRemaining.push(sumOfEstimation);
                actualWorkRemaining.push(sumOfEstimation);
                estimatedWorkRemaining.push(null);

            } else {
                //the start date of the first sprint is the same as the estimated date of the second sprint
                allocatedWorkRemaining.push(dates[i-1].effort);
                estimatedWorkRemaining.push(null);
                actualWorkRemaining.push(null);
            }

        } else if (dates[i].dateType === 'endDate') {
            allocatedWorkRemaining.push(dates[i].effort);
            estimatedWorkRemaining.push(null);
            actualWorkRemaining.push(null);

        } else if (dates[i].dateType === 'actualCompletionDate') {
            allocatedWorkRemaining.push(null);
            estimatedWorkRemaining.push(null);
            actualWorkRemaining.push(dates[i].effort);

        } else if (dates[i].dateType === 'estimatedDate') {
            allocatedWorkRemaining.push(null);
            estimatedWorkRemaining.push(dates[i].effort);
            actualWorkRemaining.push(null);
        }
    }

    // Create a new Chart.js instance
    const ctx = document.getElementById('OverallBurndownChart').getContext('2d');
    this.burndownChart = new Chart(ctx, {
        type: 'line',
        data: {
            labels: sprintDates,
            datasets: [
                {
                    label: 'Estimated Work Remaining',
                    data: estimatedWorkRemaining,
                    borderColor: 'rgba(221, 18, 18, 0.6)',
                    backgroundColor: 'rgba(221, 18, 18, 0.4)',
                    fill: false,
                    spanGaps: true
                },
                {
                    label: 'Actual Work Remaining',
                    data: actualWorkRemaining,
                    borderColor: 'rgba(221, 107, 32, 0.6)',
                    backgroundColor: 'rgba(221, 107, 32, 0.4)',
                    fill: false,
                    spanGaps: true
                },
                {
                    label: 'Allocated Work Remaining',
                    data: allocatedWorkRemaining,
                    borderColor: 'rgba(56, 161, 105, 0.6)',
                    backgroundColor: 'rgba(56, 161, 105, 0.4)',
                    fill: false,
                    spanGaps: true
                },
            ]
        },
        options: {
            plugins: {
                title: {
                    display: true,
                    text: 'Burndown Chart',
                    font: {
                        size: 20,
                        weight: 'bold'
                    }
                }
            },
            scales: {
                x: {
                    title: {
                        display: true,
                        text: 'Sprint Completion Dates',
                        font: {
                            size: 15,
                            weight: 'bold'
                        }
                    },

                },
                y: {
                    title: {
                        display: true,
                        text: 'Estimation Point(s)',
                        font: {
                            size: 15,
                            weight: 'bold'
                        }

                    },
                    ticks: {
                        beginAtZero: true
                    },

                }
            }
        }
    });
},
```

Figure 6.134: Plot Overall Burndown Chart Function (Vue.js)

For the overall burndown chart and burnup chart, custom mathematical equations are implemented within the code to accurately plot the graphs, considering the complexity involved.

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

Testing is an essential component of the software development life cycle as it helps to detect bugs, defects, and areas for improvement in the developed software. It ensures that the quality of the system is maintained. This chapter will focus on the testing conducted for the final year project. Section 7.2 will outline the test objectives to be achieved throughout the testing period, section 7.3 will discuss the test strategy, 7.4 will focus on unit testing, where each module's functions are tested individually, and 7.5 will focus on usability testing, where the user satisfaction rate will be determined. Finally, 7.6 will focus on UAT testing, which is conducted alongside usability testing to ensure user acceptance of the software.

7.2 Test Objectives

While many functions have been developed, the test objectives will focus on the core features of the web application, which are aimed at solving the stated problems. The test objectives are as follows:

1. Ensure users can create and manage their SCRUM projects.
2. Ensure role-based controls are applied in the web application.
3. Ensure users can generate a RTM of their SCRUM projects.
4. Ensure users can apply search filters on their RTM.
5. Ensure users can visualize their SCRUM projects through graphs.
6. Ensure users can estimate their project's estimated completion date.
7. Ensure users can estimate user estimation effort through planning poker.

7.3 Test Strategy

Since the web application has already been fully developed with full stack features, it will be tested as a whole rather than separating it into backend and frontend components. Unit testing will be conducted manually, with test cases written based on all the validations implemented throughout the web application. For usability testing, the SUS (System Usability Scale) approach will be used to calculate the user satisfaction rate. Finally, for UAT (User Acceptance Testing), scenarios will be given to users to follow the test cases identified in unit testing.

Usability testing and UAT testing will be conducted by visiting the participants' homes. An explanation of the web application will be provided, after which the participants will be asked to test the product and provide their honest review. This approach allows for real-time interaction with the participants, gaining an insight of their experience with the web applications. However, for participants who are not available, Microsoft Teams will be used.

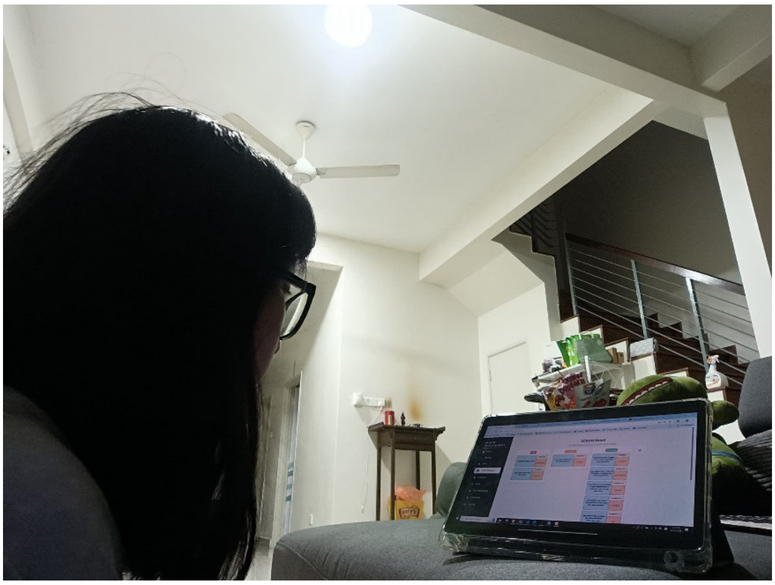


Figure 7.1: Usability and UAT Testing with Tan Le Qie

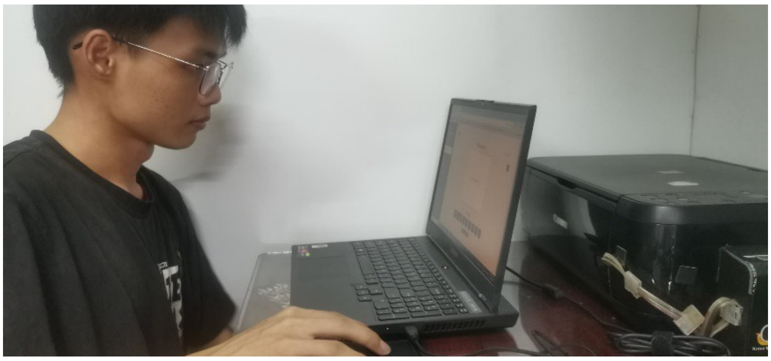


Figure 7.2: Usability and UAT Testing with Ong Pei Kang

7.3.1 Entry Criteria

Testing of the web application will only commence when:

1. All web application functions have been fully developed.
2. All test cases have been identified.

7.3.2 Exit Criteria

The web application will only be ended when:

1. All test cases have been executed.
2. The unit testing has achieved at least a 95% pass rate.
3. The user satisfaction rate is at least 80%.

The user satisfaction rate must achieve at least an 80% in alignment with the third objective of the project.

7.4 Unit Testing

Table 7.1: Register Module Test Case

Test Case #	2	**Test Case Name**	Login and Logout Module		
Test Case Summary	To test if the user is able to register an account in the web application.				
Pre-Conditions	-				
Prepared By	Tan Eng Ian				
Executed By	Tan Eng Ian				
Test Summary	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass / Fail)**
Register account with valid information	1. Enter name. 2. Enter username. 3. Enter email. 4. Enter password.	1. Valid name 2. Valid username 3. Valid email 4. Valid password	Alert message is shown to user that the account has been successfully	Alert message is shown to user that the account has been successfully	PASS

	5. Enter confirm password. 6. Enter address. 7. Enter phone number. 8. Click on "Register" button.	5. Valid confirm password 6. Valid address 7. Valid phone number	registered and redirects to login page.	registered and redirects to login page.	
Register account with invalid email format	1. Enter name. 2. Enter username. 3. Enter email. 4. Enter password. 5. Enter confirm password. 6. Enter address. 7. Enter phone number.	1. Valid name 2. Valid username 3. Invalid email 4. Valid password 5. Valid confirm password 6. Valid address 7. Valid phone number	Alert message is shown reminding user that email format entered is incorrect.	Alert message is shown reminding user that email format entered is incorrect	PASS

	8. Click on "Register" button.				
Register account with invalid password format	1. Enter name. 2. Enter username. 3. Enter email. 4. Enter password. 5. Enter confirm password. 6. Enter address. 7. Enter phone number. 8. Click on "Register" button.	1. Valid name 2. Valid username 3. Valid email 4. Invalid password 5. Valid confirm password 6. Valid address 7. Valid phone number	Alert message is shown reminding user that password field requires at least 1 upper case, 1 lower case, 1 number and 1 special character.	Alert message is shown reminding user that password field requires at least 1 upper case, 1 lower case, 1 number and 1 special character.	PASS
Register account with invalid confirm password	1. Enter name. 2. Enter username. 3. Enter email.	1. Valid name 2. Valid username 3. Valid email	Alert message is shown reminding user that confirm	Alert message is shown reminding user that confirm	PASS

	4. Enter password. 5. Enter confirm password. 6. Enter address. 7. Enter phone number. 8. Click on "Register" button.	4. Valid password 5. Invalid confirm password 6. Valid address 7. Valid phone number	password does not match with the password entered.	password does not match with the password entered.	
Register account with invalid phone number format	1. Enter name. 2. Enter username. 3. Enter email. 4. Enter password. 5. Enter confirm password. 6. Enter address. 7. Enter phone number.	1. Valid name 2. Valid username 3. Valid email 4. Valid password 5. Valid confirm password 6. Valid address 7. Invalid phone number	Alert message is shown reminding user that phone number must begin with 01 and range from 10 to 11 numbers.	Alert message is shown reminding user that phone number must begin with 01 and range from 10 to 11 numbers.	PASS

	8. Click on “Register” button.				
Register account with repeated username in database	1. Enter name. 2. Enter username. 3. Enter email. 4. Enter password. 5. Enter confirm password. 6. Enter address. 7. Enter phone number. 8. Click on “Register” button.	1. Valid name 2. Username: tanengian 3. Valid email 4. Valid password 5. Invalid confirm password 6. Valid address 7. Valid phone number	Alert message is shown reminding user that username is already registered in the system.	Alert message is shown reminding user that username is already registered in the system.	PASS
Register account with repeated email in database	1. Enter name 2. Enter username 3. Enter email	1. Valid name 2. Valid username	Alert message is shown reminding user that email is already	Alert message is shown reminding user that email is already	PASS

	4. Enter password 5. Enter confirm password 6. Enter address 7. Enter phone number 8. Click on "Register" button.	3. Email: tanengian@gmail.com 4. Valid password 5. Valid confirm password 6. Valid address 7. Valid phone number	registered in the system.	registered in the system.	
Register account with repeated phone number in database	1. Enter name. 2. Enter username. 3. Enter email. 4. Enter password. 5. Enter confirm password. 6. Enter address. 7. Enter phone number.	1. Valid name 2. Valid username 3. Valid email 4. Valid password 5. Valid confirm password 6. Valid address 7. Phone Number: 0123456789	Alert message is shown reminding user that phone number is already registered in the system.	Alert message is shown reminding user that phone number is already registered in the system.	PASS

	8. Click on “Register” button.				
Register account with blank fields	1. Enter name. 2. Enter username. 3. Enter email. 4. Enter password. 5. Enter confirm password. 6. Enter address. 7. Enter phone number. 8. Click on “Register” button.	All possible combination of leaving the fields blank (from 1 empty to all empty)	Alert message is shown reminding user that all of the field must be entered.	Alert message is shown reminding user that all of the field must be entered.	PASS

Table 7.2: Login and Logout Module Test Case

Test Case #	2	**Test Case Name**	Login and Logout Module		
Test Case Summary	To test if the user is able to login and logout from the web application.				
Pre-Conditions	User has successfully registered an account in the web application.				
Prepared By	Tan Eng Ian				
Executed By	Tan Eng Ian				
Test Summary	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass / Fail)**
Login account with valid email and password	1. Enter email. 2. Enter password. 3. Click "Login" button.	1. Valid email 2. Valid password	Successful login and redirect to project page	Successful login and redirect to project page	PASS

Login account with invalid email and password	1. Enter email. 2. Enter password. 3. Click "Login" button.	1. Invalid email 2. Invalid password	Alert message is shown reminding user that login credential mismatch.	Alert message is shown reminding user that login credential mismatch.	PASS
Login account with invalid email format	1. Enter email. 2. Enter password. 3. Click "Login" button.	1. Invalid email format 2. Valid password	Alert message is shown reminding user email format is incorrect.	Alert message is shown reminding user email format is incorrect.	PASS
Login account with blank email and password	1. Enter email. 2. Enter password. 3. Click "Login" button.	3 combinations: 1. Blank email 2. Valid password 1. Valid email 2. Blank password 1. Blank password 2. Blank email	Alert message is shown reminding user that all field must be entered.	Alert message is shown reminding user that all field must be entered.	PASS

Logout account from the web application	1. Click Logout button.	-	Successful Logout and redirects to Login page.	Successful Logout and redirects to Login page.	PASS

Table 7.3: Project Module

Test Case #	3	**Test Case Name**	Project Module		
Test Case Summary	To test if the user is able to create project, select a project to manage and manage users in the project.				
Pre-Conditions	User has successfully registered an account in the web application and logged in to it.				
Prepared By	Tan Eng Ian				
Executed By	Tan Eng Ian				
Test Summary	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass / Fail)**
Create new project	1. Click on "+" button or "Create Project" button.	1. Valid project name	Alert message is shown to user that the	Alert message is shown to user that the	PASS

	2. Enter project name 3. Click on “Create button”.		project has been successfully created.	project has been successfully created.	
Create new project with inviting project members	1. Click on “+” button or “Create Project” button. 2. Enter project name. 3. Enter project member email to be invited. 4. Click on the “Invite” button. 5. Click on “Create button”.	1. Valid project name 2. Invited user email: tanengian@gmail.com	Alert message is shown to user that the project has been successfully created.	Alert message is shown to user that the project has been successfully created.	PASS
Invite non-registered user	1. Click on “+” button or “Create Project” button. 2. Enter project name.	1. Valid project name 2. Invited user email: abc@gmail.com	Alert message is shown reminding user that the user invited is not registered.	Alert message is shown reminding user that the user invited is not registered.	PASS

	3. Enter project member email to be invited. 4. Click on the "Invite" button. 5. Click on "Create button".				
Invite invalid email format	1. Click on "+" button or "Create Project" button. 2. Enter project name. 3. Enter project member email to be invited. 4. Click on the "Invite" button. 5. Click on "Create button".	1. Valid project name 2. Invalid email format	Alert message is shown reminding user that the email format is incorrect.	Alert message is shown reminding user that the email format is incorrect.	PASS

Invite blank field	1. Click on "+" button or "Create Project" button. 2. Enter project name. 3. Enter project member email to be invited. 4. Click on the "Invite" button. 5. Click on "Create button".	1. Valid project name 2. Blank invited user email	Alert message is shown reminding user that the invited user email field must be entered.	Alert message is shown reminding user that the invited user email field must be entered.	PASS
Invite yourself	1. Click on "+" button or "Create Project" button. 2. Enter project name. 3. Enter project member email to be invited.	1. Valid project name 2. Current authenticated user email	Alert message is shown reminding user that the user cannot invite himself to the project.	Alert message is shown reminding user that the user cannot invite himself to the project.	PASS

	4. Click on the "Invite" button. 5. Click on "Create button".				
Delete pending invited project members	1. Click on "+" button or "Create Project" button. 2. Enter project name. 3. Enter project member email to be invited. 4. Click on the "Invite" button. 5. Click on delete icon. 6. Confirm delete.	1. Valid project name 2. Invited user email: tanengian@gmail.com	Alert message is shown reminding user the user has been successfully removed.	Alert message is shown reminding user the user has been successfully removed.	PASS

Invite new user to project	1. Select an existing project to manage. 2. Confirm selection of project. 3. Navigate through user management page. 4. Click on "Add New User" button. 5. Enter invited user email. 6. Click on "Invite" button. 7. Click on "Confirm" button.	1. Invited user email: joshua@gmail.com	Successful Logout and redirects to Login page.	Successful Logout and redirects to Login page.	PASS
Invite new user to project with invalid email format	1. Select an existing project to manage.	1. Invalid invited user email format	Alert message is shown reminding user	Alert message is shown reminding user	PASS

	2. Confirm selection of project. 3. Navigate through user management page. 4. Click on "Add New User" button. 5. Enter invited user email. 6. Click on "Invite" button. 7. Click on "Confirm" button.		that the email format is incorrect.	that the email format is incorrect.	
Invite new user to project with non-registered user	1. Select an existing project to manage. 2. Confirm selection of project.	1. Invited user email: abc@gmail.com	Alert message is shown reminding user that the user invited is not registered.	Alert message is shown reminding user that the user invited is not registered.	PASS

	3. Navigate through user management page. 4. Click on "Add New User" button. 5. Enter invited user email. 6. Click on "Invite" button. 7. Click on "Confirm" button.				
Invite new user to project that already invited to the project	1. Select an existing project to manage. 2. Confirm selection of project. 3. Navigate through user management page.	1. Ensures that tanengian@gmail.com is already invited to project. 2. Invited user email: tanengian@gmail.com	Alert message is shown reminding user that the user cannot an existing project member to the project	Alert message is shown reminding user that the user cannot an existing project member to the project	PASS

	4. Click on "Add New User" button. 5. Enter invited user email. 6. Click on "Invite" button. 7. Click on "Confirm" button.				
Invite yourself	1. Select an existing project to manage. 2. Confirm selection of project. 3. Navigate through user management page. 4. Click on "Add New User" button.	1. Invited user email: current authenticated user email	Alert message is shown reminding user that the user cannot invite himself to the project	Alert message is shown reminding user that the user cannot invite himself to the project	PASS

	5. Enter invited user email. 6. Click on "Invite" button. 7. Click on "Confirm" button.				
Invite new user to project with blank field.	1. Select an existing project to manage. 2. Confirm selection of project. 3. Navigate through user management page. 4. Click on "Add New User" button. 5. Enter invited user email.	1. Blank invited user email	Alert message is shown reminding user that the invited user email field must be entered.	Alert message is shown reminding user that the invited user email field must be entered.	PASS

	6. Click on “Invite” button. 7. Click on “Confirm” button.				
Delete pending invited project members	1. Select an existing project to manage. 2. Confirm selection of project. 3. Navigate through user management page. 4. Click on “Add New User” button. 5. Enter invited user email. 6. Click on “Invite” button.		Alert message is shown reminding user the user has been successfully removed.	Alert message is shown reminding user the user has been successfully removed.	PASS

	7. Click on delete icon. 8. Confirm delete				
Update user role	1. Select an existing project to manage. 2. Confirm selection of project. 3. Navigate through user management page. 4. Click on "Edit" button on a member. 5. Update role by checking the role 6. Click on "Confirm" button.	Role combination: SCRUM Master Product Onwer Team Member and 2-3 combination of these 3 roles as user can be assigned with more than 1 role.	Alert message is shown reminding user that the role has been successfully updated.	Alert message is shown reminding user that the role has been successfully updated	PASS

Update user role with 0 role	1. Select an existing project to manage. 2. Confirm selection of project. 3. Navigate through user management page. 4. Click on "Edit" button on a member. 5. Update role by checking the role 6. Click on "Confirm" button.	1. Role selected: none	Alert message is shown reminding user that a minimum of 1 role is required.	Alert message is shown reminding user that a minimum of 1 role is required.	PASS
Delete user from project	1. Select an existing project to manage. 2. Confirm selection of project.	Ensures that at least 1 project member other than the user is invited.	Alert message is shown reminding user that the project member has been	Alert message is shown reminding user that the project member has been	PASS

	3. Navigate through user management page. 4. Click on "Delete" button on a member. 5. Confirm delete		successfully removed.	successfully removed.	

Table 7.4: Backlog Module Test Case

Test Case #	4	**Test Case Name**	Backlog Module
Test Case Summary	To test if the user is able to manage their backlog items.		
Pre-Conditions	User has successfully registered an account in the web application, logged into it and selected a project to manage.		
Prepared By	Tan Eng Ian		
Executed By	Tan Eng Ian		

Test Summary	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass / Fail)**
Create product backlog	1. Click on “New Product Backlog” button. 2. Enter backlog description. 3. Choose backlog priority. 4. Click on “Confirm” button.	1. Valid description 2. Priority: Low, Medium or High	Alert message is shown reminding user that the product backlog has been successfully created.	Alert message is shown reminding user that the product backlog has been successfully created.	PASS
Create product backlog with empty description	1. Click on “New Product Backlog” button. 2. Enter backlog description. 3. Choose backlog status.	1. Blank description 2. Priority: Low, Medium or High	Alert message is shown reminding user that the product backlog description field is required to be entered.	Alert message is shown reminding user that the product backlog description field is required to be entered.	PASS

	4. Click on "Confirm" button.				
Delete product backlog	1. Click on the delete icon of the product backlog. 2. Confirm delete.		Alert message is shown that the product backlog has been successfully deleted.	Alert message is shown that the product backlog has been successfully deleted.	PASS
Update product backlog	1. Click on the edit icon of the product backlog. 2. Enter new value of the product backlog (new status, priority or description). 3. Click on "Confirm" button.	1. Valid description 2. Priority: Low, Medium or High 3. Status: To Do, In Progress or Completed.	Alert message is shown that the product backlog has been successfully updated.	Alert message is shown that the product backlog has been successfully updated.	PASS

Update product backlog with empty description	1. Click on the edit icon of the product backlog. 2. Enter new value of the product backlog (new status, priority or description). 3. Click on "Confirm" button.	1. Blank description 2. Priority: Low, Medium or High 3. Status: To Do, In Progress or Completed.	Alert message is shown reminding user that the product backlog description field is required to be entered.	Alert message is shown reminding user that the product backlog description field is required to be entered.	PASS
Create sprint backlog	1. Click on the add new item for product backlog. 2. Enter sprint backlog description. 3. Enter sprint backlog category. 4. Choose sprint backlog priority.	1. Valid description 2. Valid category 3. Priority: Low, Medium or High 4. Assigned To: anyone within the project	Alert message is shown reminding user that the sprint backlog has been successfully created.	Alert message is shown reminding user that the sprint backlog has been successfully created.	PASS

	5. Choose sprint backlog assignee.				
Create sprint backlog with empty fields	1. Click on the add new item for product backlog. 2. Enter sprint backlog description. 3. Enter sprint backlog category. 4. Choose sprint backlog priority. 5. Choose sprint backlog assignee.	3 combinations: 1. Blank description 2. Blank category 1. Valid description 2. Blank category 1. Blank description 2. Valid category 3. Priority: Low, Medium or High 4. Assigned To: anyone within the project	Alert message is shown reminding user that the sprint backlog fields are required to be entered.	Alert message is shown reminding user that the sprint backlog fields are required to be entered.	PASS

Delete sprint backlog	1. Click on the delete icon of the sprint backlog. 2. Confirm delete.		Alert message is shown that the sprint backlog has been successfully deleted.	Alert message is shown that the sprint backlog has been successfully deleted.	PASS
Update sprint backlog	1. Click on the edit icon of the sprint backlog. 2. Enter new value of the sprint backlog (new status, priority, description or assignee). 3. Click on "Confirm" button.	1. Valid description 2. Valid category 3. Priority: Low, Medium or High 4. Assigned To: anyone within the project 6. Status: To Do, In Progress or Completed	Alert message is shown that the sprint backlog has been successfully updated.	Alert message is shown that the sprint backlog has been successfully updated.	PASS

Update sprint backlog with empty field	1. Click on the edit icon of the product backlog. 2. Enter new value of the product backlog (new status, priority or description). 3. Click on “Confirm” button.	3 combinations: 1. Blank description 2. Blank category 1. Valid description 2. Blank category 1. Blank description 2. Valid category 3. Priority: Low, Medium or High 4. Assigned To: anyone within the project	Alert message is shown reminding user that the sprint backlog fields are required to be entered.	Alert message is shown reminding user that the sprint backlog fields are required to be entered.	PASS

		6. Status: To Do, In Progress or Completed			

Table 7.5: RTM Module Test Case

Test Case #	5	**Test Case Name**	RTM
Test Case Summary	To test if the user is able to view and search through the RTM.		
Pre-Conditions	User has successfully registered an account in the web application, logged into the it, selected a project and has existing product backlog and sprint backlog. Dummy Data: Product Backlog – Credential Features, Security Feature Sprint Backlog – Login Feature (Under Credential)		
Prepared By	Tan Eng Ian		
Executed By	Tan Eng Ian		

Test Summary	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass / Fail)
Expand Details	1. User clicks on the show details of the product backlog.		The sprint backlog of the product backlog will be shown.	The sprint backlog of the product backlog will be shown.	PASS
Search product backlog	1. Enter search keyword	1. Keyword: Cre	The RTM only shows the Credential Feature product backlog.	The RTM only shows the Credential Feature product backlog.	PASS
Search sprint backlog	1. Enter search keyword	1. Keyword: Log	The RTM only shows the Credential Feature product backlog and auto expand to show the Login Feature sprint backlog.	The RTM only shows the Credential Feature product backlog and auto expand to show the Login Feature sprint backlog.	PASS

Table 7.6: SCRUM Board Module Test Case

Test Case #	6	**Test Case Name**	SCRUM Board Module

Test Case Summary	To test if the user is able to drag and drop sprint backlog item within the SCRUM Board, update the assignee and create customized board (status).				
Pre-Conditions	User has successfully registered an account in the web application, logged into the it, selected a project and has existing product backlog and sprint backlog. Dummy Data: Sprint Backlog – Login Feature, Register Feature, Logout Feature				
Prepared By	Tan Eng Ian				
Executed By	Tan Eng Ian				
Test Summary	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass / Fail)**
Drag and drop sprint backlog item	1. Drag the Login Feature from To Do Board to In Progress Board.	-	The status of the sprint backlog item is updated accordingly in every page.	The status of the sprint backlog item is updated accordingly in every page.	PASS

	2. Drag the Login Feature from In Progress Board to Completed Board				
Changing assignee	1. Change the assignee of the sprint backlog item to other project members.		The assignee of the sprint backlog item has been updated accordingly in every page.	The assignee of the sprint backlog item has been updated accordingly in every page.	PASS
Create customized board	1. Click on the "+" button. 2. Enter customized board name. 3. Confirm creation.	1. Valid board name	An alert message is shown reminding user that the customized board has been successfully created.	An alert message is shown reminding user that the customized board has been successfully created.	PASS
Create customized board with empty board name	1. Click on the "+" button. 2. Enter customized board name.	1. Blank board name	An alert message is shown reminding user that the board name is required to be entered	An alert message is shown reminding user that the board name is required to be entered.	

	3. Confirm creation.				
Drag and drop item to customized board	1. Drag the Login Feature from Completed Board to the newly created customized board.		The customized status of the sprint backlog item is updated accordingly in every page.	The customized status of the sprint backlog item is updated accordingly in every page.	PASS
Delete customized board	1. Click on the delete icon on the customized board. 2. Confirm delete.		The sprint backlog items in the customized status are assigned with the status “To Do”.	The sprint backlog items in the customized status are assigned with the status “To Do”.	PASS

Table 7.7: Sprint Module Test Case

Test Case #	7	**Test Case Name**	Sprint Module
Test Case Summary	To test if the user is able to initiate and end a sprint.		
Pre-Conditions	User has successfully registered an account in the web application, logged into the it, selected a project and has existing product backlog and sprint backlog that is not completed.		

	Dummy Data: Sprint Backlog – Login Feature, Register Feature, Logout Feature All the sprint backlog data will be “To Do” status.				
Prepared By	Tan Eng Ian				
Executed By	Tan Eng Ian				
Test Summary	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass / Fail)**
Initiate sprint	1. Click on “New Sprint” button. 2. Select a valid start date and end date. 3. Select sprint backlog item.	1. Valid start date and end date. 2. Select any of the sprint backlog available in the dummy data.	An alert message is shown reminding the user that the sprint has been successfully created.	An alert message is shown reminding the user that the sprint has been successfully created.	PASS

	4. Click on "Create Sprint" button.				
Initiate sprint with empty dates	1. Click on "New Sprint" button. 2. Select a valid start date and end date. 3. Select sprint backlog item. 4. Click on "Create Sprint" button.	3 Combinations: 1. Empty start date 2. Empty end date 1. Empty start date 2. Valid end date 1. Valid start date 2. Empty end date 3. Select any of the sprint backlog available in the dummy data.	An alert message is shown reminding user that both of the dates are required to be entered.	An alert message is shown reminding user that both of the dates are required to be entered.	PASS

Initiate sprint with start date greater than end date	1. Click on "New Sprint" button. 2. Select a valid start date and end date. 3. Select sprint backlog item. 4. Click on "Create Sprint" button.	1.Start date: 5/12/2024 2. End date: 5/8/2024 3. Select any of the sprint backlog available in the dummy data.	An alert message is shown reminding user that start date cannot be greater than end date.	An alert message is shown reminding user that start date cannot be greater than end date.	PASS
Initiate sprint with no sprint backlog selected	1. Click on "New Sprint" button. 2. Select a valid start date and end date. 3. Select sprint backlog item. 4. Click on "Create Sprint" button.	1. Valid start date and end date 2. No sprint backlog selected	An alert message is shown reminding user that the sprint cannot be initiated when no sprint backlog item is selected.	An alert message is shown reminding user that the sprint cannot be initiated when no sprint backlog item is selected.	PASS
End sprint	1. Click on the "End Sprint" button.	1. valid actual effort	An alert message is shown reminding user	An alert message is shown reminding user	PASS

	2. Enter actual effort. 3. Click on "Confirm" button.		that the sprint has been successfully ended.	that the sprint has been successfully ended.	
End sprint with empty effort	1. Click on "New Sprint" button. 2. Select a valid start date and end date. 3. Select sprint backlog item. 4. Click on "Create Sprint" button.	1. Empty actual effort	An alert message is shown reminding user that the actual effort field is required to be entered.	An alert message is shown reminding user that the actual effort field is required to be entered.	PASS

Table 7.8: Estimation Module Test Case

Test Case #	8	**Test Case Name**	Estimation Module
Test Case Summary	To test if the user is able to estimate a sprint backlog item estimation effort through default option or planning poker session.		

Pre-Conditions	User has successfully registered an account in the web application, logged into the it, selected a project and has existing product backlog and sprint backlog can be estimate. Dummy Data: Sprint Backlog – Login Feature				
Prepared By	Tan Eng Ian				
Executed By	Tan Eng Ian				
Test Summary	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass / Fail)**
Update estimation	1. Select Login Feature sprint backlog. 2, Click on the "Estimation" button.	1. Valid estimation effort 2. Valid estimation unit	An alert message is shown reminding user that the estimation has been successfully updated.	An alert message is shown reminding user that the estimation has been successfully updated.	PASS

	3. Enter estimation effort. 4. Enter estimation unit. 5. Click on “Confirm” button.				
Update estimation with empty estimation unit	1. Select Login Feature sprint backlog. 2, Click on the “Estimation” button. 3. Enter estimation effort. 4. Enter estimation unit. 5. Click on “Confirm” button.	1. Valid estimation effort 2. Blank estimation unit	An alert message is shown reminding the user that the estimation unit field are required to be entered.	An alert message is shown reminding the user that the estimation unit field are required to be entered.	PASS

Update estimation with negative estimation effort	1. Select Login Feature sprint backlog. 2, Click on the "Estimation" button. 3. Enter estimation effort. 4. Enter estimation unit. 5. Click on "Confirm" button.	1. Negative estimation effort 2. Valid estimation unit	An alert message is shown reminding user that the estimation effort entered cannot be lesser than 0.	An alert message is shown reminding user that the estimation effort entered cannot be lesser than 0.	PASS
Planning poker estimation (same effort)	1. Select Login Feature sprint backlog. 2, Click on the "Estimation" button.	1. First account estimation effort: 3 2. Second account estimation effort: 3	The planning poker session ended, with the Login Feature having an estimation effort of 3.	The planning poker session ended, with the Login Feature having an estimation effort of 3.	PASS

	3. Click on the “Planning Poker Estimation” button. 4. Alternatively, planning poker session can be join through notification if other members initiated. 5. Enter estimation effort. 6. Step 1-5 is repeated using another account that is within the same project.				
Planning poker estimation (different effort)	1. Select Login Feature sprint backlog.	1. First account estimation effort: 3	The planning poker initiate a revote session.	The planning poker initiate a revote session.	PASS

	2, Click on the "Estimation" button. 3. Click on the "Planning Poker Estimation" button. 4. Alternatively, planning poker session can be join through notification if other members initiated. 5. Enter estimation effort. 6. Step 1-5 is repeated using another account that is within the same project. 7. Select an estimation effort of 5.	2. Second account estimation effort: 5			

Planning poker revote (only applicable to completed session)	1. Click on the "Revote" button. 2. Confirm revote.		The planning poker initiate a revote session.	The planning poker initiate a revote session.	PASS
Planning poker change sequence number	1. Click on the "Change Sequence Number" button. 2. Enter sequence number. 3. Click on the "Confirm" button.	1. Valid sequence number	The planning poker session sequence number is changed according to user input.	The planning poker session sequence number is changed according to user input.	PASS
Planning poker change sequence number to empty	1. Click on the "Change Sequence Number" button. 2. Enter sequence number. 3. Click on the "Confirm" button.	2. Invalid sequence number	An alert message is shown to remind user that the sequence number is required to be entered.	An alert message is shown to remind user that the sequence number is required to be entered.	PASS

Planning poker change to invalid sequence number	1. Click on the “Change Sequence Number” button. 2. Enter sequence number. 3. Click on the “Confirm” button.	3. Blank sequence number	An alert message is shown to remind user that the sequence number entered is invalid.	An alert message is shown to remind user that the sequence number entered is invalid.	PASS
Adding planning poker sticky note	1. Enter note message. 2. Click on the “+” button.	1. Valid note message	The note entered is updated on the side of the page.	The note entered is updated on the side of the page.	PASS
Adding empty planning poker sticky note	1. Enter note message. 2. Click on the “+” button.	2. Blank note message	An alert message is shown to remind user that the note is required to be entered.	An alert message is shown to remind user that the note is required to be entered.	PASS

Table 7.9: Estimated Completion Date Test Case

Test Case #	9	**Test Case Name**	Estimated Completion Date		
Test Case Summary	To test if the web application is able to estimate the completion date correctly.				
Pre-Conditions	User has successfully registered an account in the web application, logged into the it, selected a project and has existing product backlog and sprint backlog. Sample remaining effort: 15				
Prepared By	Tan Eng Ian				
Executed By	Tan Eng Ian				
Test Summary	**Test Steps**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass / Fail)**
End Sprint	1. User initiate a sprint. 2. User change the status of sprint	1. Start date: 1/1/2024 2. End date: 1/4/2024	The final completion dateline for remaining effort is 1/16/2024.	The final completion dateline for remaining effort is 1/16/2024.	PASS

	backlog item selected to Completed. 2. User end the sprint.	3. Sprint backlog estimation effort selected: 5 4. Actual effort: 5 5. Overall remaining effort: 10			
Update Estimation	1. Change the remaining effort.	1. Over remaining effort: 12	The final completion dateline for remaining effort is 1/18/2024.	The final completion dateline for remaining effort is 1/18/2024.	PASS

Since all of the written test cases have passed with a 100% success rate, the unit testing is considered successful. However, it's important to note that the written test cases may not cover all modules, so the results may not be entirely accurate.

7.5 Usability Testing

Usability testing evaluates the satisfaction level of users with the final product of the web application. In this final year project, five participants from software backgrounds will be invited to participate. They will be asked to perform tasks based on provided scenarios and then rate the web application using a set of questions. The System Usability Scale (SUS) approach, which is widely used in usability testing, will be employed. SUS is a questionnaire-based method used to assess the usability of a system or product. It consists of a series of ten statements that participants rate on a scale from 1 to 5, ranging from "Strongly Disagree" to "Strongly Agree". The SUS score is then calculated based on participants' responses. The following scenarios has been given:

Table 7.10: Usability Testing Scenario

Task Scenario	**Task Description**
Register an account	A project manager wanted to use the web application. However, he or she has yet to register an account.
Login and Logout	The project manager wishes log in to the web application using their credentials. Upon completing their tasks, the project manager wants to log out from the web application to ensure the security of their account and data.
Create project	The project manager wants to create a SCRUM project to manage and invite all of his or her team members into the project.
Manage user role	The project manager would like to set distinct roles for each of the project members, allowing them to have a limited level of authorities.
Manage backlog items	The Product Owner adds a new module feature to the product backlog. However, realizing a mistake in the priority assigned, the Product Owner updates the priority accordingly. As time progresses, the feature is abandoned, the Product Owner then delete it from the product backlog.

	The SCRUM Master and Team Members add sprint backlog items based on the product backlog. These sprint backlog items are regularly updated to reflect real-time data. If a feature is aborted, the corresponding sprint backlog items are also deleted to maintain accuracy and relevance.
Initiate and end sprint	The SCRUM Master would like to initiate a sprint based on the available sprint backlog item. The SCRUM Master would like to end a sprint after the sprint session has ended in real-time.
Sprint Backlog Estimation	The SCRUM Master initiates a planning poker session, inviting all project members to estimate the sprint backlog's estimation effort. Additionally, based on past experience, the SCRUM Master directly estimates the effort for backlog items that he considered as easy.
Estimated Completion Date	The teams are having difficulty in estimating their project completion dateline and tracking the project progress.

To calculate the average SUS score, a questionnaire is prepared. The odd-numbered questions are positive, while the even-numbered ones are negative. The score for odd-numbered questions is subtracted by 1, and for even-numbered questions, it's subtracted by 5. Then, these scores are multiplied by 2.5 to get the 100% score, where higher scores indicate higher satisfaction rate.

Table 7.11: User Satisfaction Survey Result

Questions	**Participant**					**Average**
	1	**2**	**3**	**4**	**5**	
1. I think that I would like to use the web application	5	5	5	5	5	

to manage my SCRUM project						
2. I found the web application to be unnecessarily complex.	2	2	1	1	1	
3. I thought the web application was easy to use.	4	4	4	5	5	
4. I think that I would need the support of a technical person to be able to use the web application.	1	1	1	1	1	
5. I found the role-based authorization of the web application are useful.	5	5	5	5	5	
6. I thought there was too much inconsistency in the web application.	1	1	1	1	1	
7. I would imagine that most people would learn to use the web application very quickly.	5	5	5	4	5	
8. I found the web application to be very confusing to use.	1	1	1	1	1	
9. I felt very confident using the web application.	5	5	5	5	5	
10. I needed to learn a lot of things before I could get going with the web application.	1	2	1	1	1	
SUS Raw Score	38	37	39	39	40	

SUS Final Score	95	92.5	97.5	97.5	100	96.5

The overall feedback that has been received is as follow:

Positive Comments

- The system has a high simplicity, with most pages featuring only two-color scheme, enhances readability and usability.
- Easy navigation
- The system offers high functionality, providing SCRUM management functions like adding a new project, drag-and-drop functionality for SCRUM Board and planning poker session which enhances the user experience.
- Clear visualization, such as color-coded progress tracking with graphs and a requirement traceability matrix, helps users easily understand project status and progress.
- Allowing the estimated completion date to be calculate, in which many SCRUM web applications doesn’t have.

Negative Comments

- Lack of instructions may pose challenges for beginners in SCRUM management when creating projects.
- Limited HCI components result in a less interactive interface for project creation and management.
- Task complexity, covering multiple perspectives, may lead to confusion or difficulty in understanding requirements.

Recommendations

- Implement simple questions or a survey to gauge users' knowledge levels in SCRUM management software.
- Add step-by-step guidelines to assist different user groups with navigation.

Through usability testing, the third objective of the project has been achieved, as the usability testing has obtained a SUS score of 96.5%.

7.6 UAT Testing

User acceptance testing ensures that the developed features in the web application meet expectations and satisfy users. It involves giving users simple tasks based on scenarios, and if they successfully complete the task, the UAT for that module is considered successful.

Table 7.12: UAT Testing Summary

Participant / **Test Modules (Pass/Fail)**	**1**	**2**	**3**	**4**	**5**
Register Module	Pass	Pass	Pass	Pass	Pass
Login and Logout Module	Pass	Pass	Pass	Pass	Pass
Project Module	Pass	Pass	Pass	Pass	Pass
Backlog Module	Pass	Pass	Pass	Pass	Pass
RTM Module	Pass	Pass	Pass	Pass	Pass
Sprint Module	Pass	Pass	Pass	Pass	Pass
Estimation Module	Pass	Pass	Pass	Pass	Pass

The User Acceptance Testing (UAT) has successfully passed all tests, covering the basic functionalities of the web application. However, it's important to acknowledge that due to time constraints, not all possible validation cases were addressed during the testing process. Besides that, the testers are only testing from the Project Manager perspective, limiting the role testing. As feedback has already been collected in the Usability Test process, UAT will not be collecting any comments from tester.

While the UAT results are positive, it's essential to recognize that the testing was conducted with only five users who possess minimal knowledge of SCRUM methodology. Despite passing the UAT, there may still be undiscovered limitations that require further testing with a larger user base.

During the UAT process, users faced challenges while using the estimation module, requiring a need for hints to assist in task completion. Therefore, there is an opportunity for improvement in the UI of the estimation module to enhance user understanding and usability.

CHAPTER 8

CONCLUSION AND RECOMMENDATION

8.1 Conclusion

In this project, several challenges within the SCRUM methodology have been identified, including the dynamic nature of SCRUM, iterative development, dynamic requirements, and lack of formal documentation. These characteristics make forecasting and tracking project completion timelines difficult. However, by enhancing visibility and allowing forecasting of project completion deadlines, these challenges can be overcome.

The project has three main objectives: to study existing tools and techniques, develop a SCRUM web application, and achieve a user satisfaction rate of at least 80% through usability testing.

The first objective is achieved through studying burnup charts, burndown charts, RTM, and planning poker, all of which are core functionalities of the web application. Additionally, the second objective is achieved upon fully developing the web application. The third objective is achieved during the testing phase, with a user satisfaction rate of 96.5%.

8.1.1 Limitation

The web application's primary limitation lies in its heavy reliance on user input for the estimation model to determine the SCRUM project completion date. Without user input, the algorithm cannot calculate the completion pace for sprint backlog items, thus hindering the application's ability to generate the final completion date and limiting its overall usability.

Additionally, the software design of the web application has not been fully optimized. The application implements the Service-Repository pattern in the backend, where business logic is intended to be separated from database logic. However, due to the complexity of the application, these logics are not entirely separated, resulting in less efficient code cleanliness.

8.2 Recommendation

The project currently relies heavily on user input for the estimation model, which can be improved by implementing an advanced AI algorithm trained on extensive SCRUM data. This AI-driven approach would provide more accurate estimations compared to the current version.

Additionally, addressing risks management in SCRUM projects is a challenge in the current industry. The web application could implement an algorithm that calculates estimation dates based on user task completion abilities, helping to mitigate project risks more effectively.

REFERENCES

Alexandrea, J. (2021). *MariaDB vs MySQL: What Are the Key Differences, Pros, and Cons*. [online] Hostinger Tutorials. Available at: https://www.hostinger.co.uk/tutorials/mariadb-vs-mysql [Accessed 26 Apr. 2024].

Amazon, n.d. *What is SDLC? - Software Development Lifecycle Explained - AWS*. [Online] Available at:<https://aws.amazon.com/what-is/sdlc/#:~:text=The%20software%20development%20lifecycle%20(SDLC.> [Accessed on 26 July 2023].

Ankitpal (2022). *Service Repository Pattern*. [online] Medium. Available at: https://medium.com/@ankitpal181/service-repository-pattern-802540254019 [Accessed 26 Apr. 2024].

Anushka, V. (2021). *Benefit of using MVC*. [Online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/benefit-of-using-mvc/> [Accessed 26 Apr 2024].

Bagiu, N., Avasilcăi, S. And Rusu, B., 2022. Traditional Vs. Agile: The Challenge Mindset Shift In Project Management. *Review of Management & Economic Engineering*, *21*(4).

Bergman, P.-E. (2019). *Repository Design Pattern*. [Online] Medium. Available at: <https://medium.com/@pererikbergman/repository-design-pattern- e28c0f3e4a30> [Accessed 26 Apr 2024].

Budiman, R., Raharjo, T. and Suhanto, A., 2022, July. SCRUM Project Management Challenges and Solutions: Systematic Literature Review. In *2022 IEEE 8th International Conference on Computing, Engineering and Design (ICCED)* (pp. 1-6). IEEE.

Duraisamy, G. and Atan, R., 2013. Requirement Traceability Matrix Through Documentation For SCRUM Methodology. *Journal of Theoretical & Applied Information Technology*, *52*(2), pp.154-159.

Džanić, A., Toroman, A. and Džanić, A., 2022. Agile Software Development: Model, Methods, Advantages And Disadvantages. *Acta Technica Corviniensis-Bulletin of Engineering*, *15*(4).

Gupta, 2023. *Time Is of the Essence: The Role of Time Tracking in Project Management.* Available at: <https://www.softwareadvice.com/resources/time-tracking-in-project-management/ > [Accessed on 24 July 2023].

Gurianov, D.A., Myshenkov, K.S. and Terekhov, V.I., 2023, March. Software Development Methodologies: Analysis and Classification. In *2023 5th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)* (Vol. 5, pp. 1-8). IEEE.

iluwatar (n.d.). *Component*. [Online] *Java Design Patterns.* Available at: <https://java-design-patterns.com/patterns/component/> [Accessed 26 Apr. 2024].

Indeed, 2022. *Project Tracking: Importance, Benefits, Tips and FAQs.* [Online] Available at: <https://in.indeed.com/career-advice/career-development/project-tracking> [Accessed on 24 July 2023].

jaydeepsathwara272, 2022. *Introduction to Laravel and MVC Framework.* [Online] Available at < https://www.geeksforgeeks.org/introduction-to-laravel-and-mvc-framework/#article-meta-div> [Accessed on 28 July 2023].

Kumar, D. M., n.d. *What is SCRUM? When to use SCRUM. Advantages and Disadvantages of SCRUM.* [Online] Available at: <https://www.linkedin.com/pulse/what-SCRUM-when-use-advantages-disadvantages-deepak-mohan-kumar/> [Accessed on 28 July 2023].

laravel.com. (n.d.). *Laravel - The PHP Framework For Web Artisans*. [Online] Available at: <https://laravel.com/docs/11.x/sanctum> [Accessed 26 Apr. 2024].

Lei, H., Ganjeizadeh, F., Jayachandran, P.K. and Ozcan, P., 2017. A statistical analysis of the effects of SCRUM and Kanban on software development projects. *Robotics and Computer-Integrated Manufacturing*, *43*, pp.59-67.

Martin. M, 2023. *Prototype Model in Software Engineering.* [Online] Available at <https://www.guru99.com/software-engineering-prototyping-model.html> [Accessed on 28 July 2023].

OWASP (2023). *Cross Site Request Forgery (CSRF) | OWASP*. [online] owasp.org. Available at: https://owasp.org/www-community/attacks/csrf.

Ozkan, N., Bal, S., Erdogan, T.G. and Gök, M.Ş., 2022, September. SCRUM, Kanban or a Mix of Both? A Systematic Literature Review. In *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)* (pp. 883-893). IEEE.

Petrova, S. n.d. *Adopting Agile: The Latest Reports About The Popular Mindset.* [Online] Available at: <https://adevait.com/blog/remote-work/adopting-agile-the-latest-reports-about-the-popular-mindset> [Accessed on 28 July 2023].

Refactoring.guru. (2014). *Composite*. [online] Available at: https://refactoring.guru/design-patterns/composite.

Schwaber, K., and Sutherland, J., 2020. *The SCRUM Guide 2020*. [Online] Available at: <https://SCRUMguides.org/docs/SCRUMguide/v2020/2020-SCRUM-Guide-US.pdf> [Accessed on 28 July 2023].

Shahzad, B., Awan, W.N., Abro, A., Shoaib, M. and Alyahya, S., 2023. Framework for Effective Utilization of Distributed SCRUM in Software Projects. *Computer Systems Science & Engineering*, *44*(1).

TryQa, n.d. *What Is Prototype Model- Advantages, Disadvantages and When To Use It?* [Online] Available at <https://tryqa.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/> [Accessed on 28 July 2023].

Visual Paradigm, n.d. *What is Timeboxing in SCRUM?* [Online] Available at <https://www.visual-paradigm.com/SCRUM/what-is-time-boxing-in-SCRUM/> [Accessed on 28 July 2023].

web.dev. (n.d.). *Cross-Origin Resource Sharing (CORS) | Articles*. [Online] Available at: <https://web.dev/articles/cross-origin-resource-sharing> [Accessed 26 Apr. 2024].

APPENDICES

APPENDIX A: Usability Test Form

Tester #	1				
Testing Date	23/4/2024				
	Strongly Disagree 1	**Disagree 2**	**Neutral 3**	**Agree 4**	**Strongly Agree 5**
1. I think that I would like to use the web application to manage my SCRUM project					x
2. I found the web application to be unnecessarily complex.		x			
3. I thought the web application was easy to use.				x	
4. I think that I would need the support of a technical person to be able to use the web application.	x				
5. I found the role-based authorization of the web application are useful.					x

6. I thought there was too much inconsistency in the web application.	x				
7. I would imagine that most people would learn to use the web application very quickly.					x
8. I found the web application to be very confusing to use.	x				
9. I felt very confident using the web application.					x
10. I needed to learn a lot of things before I could get going with the web application.	x				

Tester #	2				
Testing Date	23/4/2024				
	Strongly Disagree 1	**Disagree 2**	**Neutral 3**	**Agree 4**	**Strongly Agree 5**
1. I think that I would like to use					x

the web application to manage my SCRUM project					
2. I found the web application to be unnecessarily complex.		x			
3. I thought the web application was easy to use.				x	
4. I think that I would need the support of a technical person to be able to use the web application.	x				
5. I found the role-based authorization of the web application are useful.					x
6. I thought there was too much inconsistency in the web application.	x				
7. I would imagine that most people would learn to use the					x

web application very quickly.					
8. I found the web application to be very confusing to use.	x				
9. I felt very confident using the web application.					x
10. I needed to learn a lot of things before I could get going with the web application.	x				

Tester #	3				
Testing Date	23/4/2024				
	Strongly Disagree 1	**Disagree 2**	**Neutral 3**	**Agree 4**	**Strongly Agree 5**
1. I think that I would like to use the web application to manage my SCRUM project					x
2. I found the web application to be unnecessarily complex.	x				

3. I thought the web application was easy to use.				x	
4. I think that I would need the support of a technical person to be able to use the web application.	x				
5. I found the role-based authorization of the web application are useful.					x
6. I thought there was too much inconsistency in the web application.	x				
7. I would imagine that most people would learn to use the web application very quickly.					x
8. I found the web application to be very confusing to use.	x				
9. I felt very confident using					x

the web application.					
10. I needed to learn a lot of things before I could get going with the web application.	x				

Tester #	4				
Testing Date	23/4/2024				
	Strongly Disagree 1	**Disagree 2**	**Neutral 3**	**Agree 4**	**Strongly Agree 5**
1. I think that I would like to use the web application to manage my SCRUM project					x
2. I found the web application to be unnecessarily complex.		x			
3. I thought the web application was easy to use.					x
4. I think that I would need the support of a technical person to be able to use	x				

the web application.					
5. I found the role-based authorization of the web application are useful.					x
6. I thought there was too much inconsistency in the web application.	x				
7. I would imagine that most people would learn to use the web application very quickly.				x	
8. I found the web application to be very confusing to use.	x				
9. I felt very confident using the web application.					x
10. I needed to learn a lot of things before I could get going with the web application.	x				

Tester #	5				
Testing Date	23/4/2024				
	Strongly Disagree 1	**Disagree 2**	**Neutral 3**	**Agree 4**	**Strongly Agree 5**
1. I think that I would like to use the web application to manage my SCRUM project					x
2. I found the web application to be unnecessarily complex.	x				
3. I thought the web application was easy to use.					x
4. I think that I would need the support of a technical person to be able to use the web application.	x				
5. I found the role-based authorization of the web application are useful.					x

6. I thought there was too much inconsistency in the web application.	x				
7. I would imagine that most people would learn to use the web application very quickly.					x
8. I found the web application to be very confusing to use.	x				
9. I felt very confident using the web application.					x
10. I needed to learn a lot of things before I could get going with the web application.	x				

APPENDIX B: User Acceptance Test Form

Tester #	1		
Testing Date	23/4/2024		
UAT ID	**Modules**	**Test Scenario**	**Result**
UAT01	Register	1. Register an account with all input filled.	Pass

UAT02	Login and Logout	1. Login with valid credentials. 2. Logout from the web application.	Pass
UAT03	Project	1. Create a new project with no team member invited. 2. Create a new project with team member invited (tanengian@gmail.com). 3. Choose a project to manage. 4. Remove a team member from the project. 5. Add new team member into the project. 6. Edit the team member role (Set it to SCRUM Master, Product Owner and/or Team Member)	Pass
UAT04	Backlog	1. Create new product backlog. 2. Create new sprint backlog.	Pass
UAT05	RTM	1. View the RTM. 2. Search the RTM with valid product backlog keyword. 3. Search the RTM with valid sprint backlog keyword.	Pass

		4. Search the RTM with no keyword matches.	
UAT06	Sprint	1. Initiate a new sprint. 2. End a sprint.	Pass
UAT07	Estimation	1. Estimate the sprint backlog item from the estimate modal. 2. Change the estimation unit. 3. Initiate a planning poker session. 4. Join the planning poker session from the notification page. 5. Estimate a sprint backlog item with planning poker (all user chooses the different estimation card). 6. Estimate a sprint backlog item with planning poker (all user chooses the same estimation card). 7. Initiate a revote in planning poker. 8. Change the planning poker sequence. 9. Add a note in the planning poker session.	Pass

Tester #	2

Testing Date	23/4/2024		
UAT ID	**Modules**	**Test Scenario**	**Result**
UAT01	Register	1. Register an account with all input filled.	Pass
UAT02	Login and Logout	1. Login with valid credentials. 2. Logout from the web application.	Pass
UAT03	Project	1. Create a new project with no team member invited. 2. Create a new project with team member invited (tanengian@gmail.com). 3. Choose a project to manage. 4. Remove a team member from the project. 5. Add new team member into the project. 6. Edit the team member role (Set it to SCRUM Master, Product Owner and/or Team Member)	Pass
UAT04	Backlog	1. Create new product backlog. 2. Create new sprint backlog.	Pass
UAT05	RTM	1. View the RTM.	Pass

		2. Search the RTM with valid product backlog keyword. 3. Search the RTM with valid sprint backlog keyword. 4. Search the RTM with no keyword matches.	
UAT06	Sprint	1. Initiate a new sprint. 2. End a sprint.	Pass
UAT07	Estimation	1. Estimate the sprint backlog item from the estimate modal. 2. Change the estimation unit. 3. Initiate a planning poker session. 4. Join the planning poker session from the notification page. 5. Estimate a sprint backlog item with planning poker (all user chooses the different estimation card). 6. Estimate a sprint backlog item with planning poker (all user chooses the same estimation card). 7. Initiate a revote in planning poker.	Pass

		8. Change the planning poker sequence. 9. Add a note in the planning poker session.	

Tester #	3		
Testing Date	23/4/2024		
UAT ID	**Modules**	**Test Scenario**	**Result**
UAT01	Register	1. Register an account with all input filled.	Pass
UAT02	Login and Logout	1. Login with valid credentials. 2. Logout from the web application.	Pass
UAT03	Project	1. Create a new project with no team member invited. 2. Create a new project with team member invited (tanengian@gmail.com). 3. Choose a project to manage. 4. Remove a team member from the project. 5. Add new team member into the project. 6. Edit the team member role (Set it to SCRUM Master, Product Owner and/or Team Member)	Pass

UAT04	Backlog	1. Create new product backlog. 2. Create new sprint backlog.	Pass
UAT05	RTM	1. View the RTM. 2. Search the RTM with valid product backlog keyword. 3. Search the RTM with valid sprint backlog keyword. 4. Search the RTM with no keyword matches.	Pass
UAT06	Sprint	1. Initiate a new sprint. 2. End a sprint.	Pass
UAT07	Estimation	1. Estimate the sprint backlog item from the estimate modal. 2. Change the estimation unit. 3. Initiate a planning poker session. 4. Join the planning poker session from the notification page. 5. Estimate a sprint backlog item with planning poker (all user chooses the different estimation card). 6. Estimate a sprint backlog item with planning poker	Pass

		(all user chooses the same estimation card). 7. Initiate a revote in planning poker. 8. Change the planning poker sequence. 9. Add a note in the planning poker session.	

Tester #	4		
Testing Date	23/4/2024		
UAT ID	**Modules**	**Test Scenario**	**Result**
UAT01	Register	1. Register an account with all input filled.	Pass
UAT02	Login and Logout	1. Login with valid credentials. 2. Logout from the web application.	Pass
UAT03	Project	1. Create a new project with no team member invited. 2. Create a new project with team member invited (tanengian@gmail.com). 3. Choose a project to manage. 4. Remove a team member from the project. 5. Add new team member into the project. 6. Edit the team member role (Set it to SCRUM	Pass

		Master, Product Owner and/or Team Member)	
UAT04	Backlog	1. Create new product backlog. 2. Create new sprint backlog.	Pass
UAT05	RTM	1. View the RTM. 2. Search the RTM with valid product backlog keyword. 3. Search the RTM with valid sprint backlog keyword. 4. Search the RTM with no keyword matches.	Pass
UAT06	Sprint	1. Initiate a new sprint. 2. End a sprint.	Pass
UAT07	Estimation	1. Estimate the sprint backlog item from the estimate modal. 2. Change the estimation unit. 3. Initiate a planning poker session. 1. Join the planning poker session from the notification page. 2. Estimate a sprint backlog item with planning poker	Pass

		(all user chooses the different estimation card). 3. Estimate a sprint backlog item with planning poker (all user chooses the same estimation card). 4. Initiate a revote in planning poker. 5. Change the planning poker sequence. 6. Add a note in the planning poker session.	

Tester #	5		
Testing Date	23/4/2024		
UAT ID	**Modules**	**Test Scenario**	**Result**
UAT01	Register	1. Register an account with all input filled.	Pass
UAT02	Login and Logout	1. Login with valid credentials. 2. Logout from the web application.	Pass
UAT03	Project	1. Create a new project with no team member invited. 2. Create a new project with team member invited (tanengian@gmail.com). 3. Choose a project to manage. 4. Remove a team member from the project.	Pass

		5. Add new team member into the project. 6. Edit the team member role (Set it to SCRUM Master, Product Owner and/or Team Member)	
UAT04	Backlog	1. Create new product backlog. 2. Create new sprint backlog.	Pass
UAT05	RTM	1. View the RTM. 2. Search the RTM with valid product backlog keyword. 3. Search the RTM with valid sprint backlog keyword. 4. Search the RTM with no keyword matches.	Pass
UAT06	Sprint	1. Initiate a new sprint. 2. End a sprint.	Pass
UAT07	Estimation	1. Estimate the sprint backlog item from the estimate modal. 2. Change the estimation unit. 3. Initiate a planning poker session.	Pass

		4. Join the planning poker session from the notification page. 5. Estimate a sprint backlog item with planning poker (all user chooses the different estimation card). 6. Estimate a sprint backlog item with planning poker (all user chooses the same estimation card). 7. Initiate a revote in planning poker. 8. Change the planning poker sequence. 9. Add a note in the planning poker session.	