

**CAPSTONE PROJECT REPORT**

**Report 2 – Project Management Plan**

– Hanoi, Jan 2025 –

**Table of Contents**

[I. Record of Changes 3](#_heading=h.30j0zll)

[II. Project Management Plan 4](#_heading=h.1fob9te)

[1. Overview 4](#_heading=h.3znysh7)

[1.1 Scope & Estimation 4](#_heading=h.q40ozz1ijbwy)

[1.2 Project Objectives 11](#_heading=h.tyjcwt)

[1.3 Project Risks 12](#_heading=h.3dy6vkm)

[2. Management Approach 13](#_heading=h.1t3h5sf)

[2.1 Quality Management 13](#_heading=h.4d34og8)

[2.2 Project Training Plan 16](#_heading=h.2s8eyo1)

[3. Project Deliverables 16](#_heading=h.25sax7nontn1)

[4. Responsibility Assignments 17](#_heading=h.17dp8vu)

[5. Project Communications 19](#_heading=h.3rdcrjn)

[6. Configuration Management 19](#_heading=h.26in1rg)

[6.1 Document Management 19](#_heading=h.lnxbz9)

[6.2 Source Code Management 20](#_heading=h.35nkun2)

[6.3 Tools & Infrastructures 21](#_heading=h.t3y3pyihx6q6)

# I. Record of Changes

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 15/01/2025 | A | quydt | Add Project Management Plan/Management Approach |
| 15/01/2025 | A | tungbd | Add Project Communication, Configuration Management |
| 16/01/2025 | A | quydt | Add 3 Project Deliverables |
| 9/5/2025 | M | tungbd | Modify the report according to the council's comments. |
| 9/5/2025 | M | thinhnt | Modify the report according to the council's comments. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

*Table 1: Record of changes*

# II. Project Management Plan

## 1. Overview

### 1.1 Scope & Estimation

| **#** | **Work Package** | **Est. Effort**  **(pds)** | **Deadline** |
| --- | --- | --- | --- |
| **1.** | ***Requirement Gathering and Analysis*** |  |  |
| **1.1** | **Initiation** |  |  |
| 1.1.1 | Create Report 1: Project Introduction | 8 | Week 1 |
| **1.2** | **Planning** |  |  |
| 1.2.1 | Create Report 2: Project Management Plan v1.0 | 8 | Week 2 |
| 1.2.2 | Create Report 3: SRS v1.0 (Overall Requirement) | 8 | Week 2 |
| **1.3** | **Analysis** |  |  |
| 1.3.1 | Create Report 4: SDS v1.0 (High-Level Design Specifications) | 8 | Week 3 |
| 1.3.1.1 | Create System Architecture | 2 | Week 3 |
| 1.3.1.2 | Create Database Design | 8 | Week 3 |
| 1.3.1.3 | Create Package Diagram | 4 | Week 3 |
| 1.3.2 | Update SRS: Update Use Case List | 4 | Week 3 |
| 1.3.3 | Create Test Plan Documents | 5 | Week 4 |
| **2.** | **Implementation** |  |  |
| **2.1** | **Iteration 1: Common & Pharmacist Features** |  | Week 5-6 |
| 2.1.1 | Requirements Analysis |  |  |
| 2.1.1.1 | Collect User Requirements | 4 | Week 5 |
| 2.1.1.1 | Identify Main Features | 4 | Week 5 |
| 2.1.1.1 | Update SRS | 4 | Week 5 |
| **2.1.2** | **Design** |  |  |
| 2.1.2.1 | Update System Architecture/Package Diagram | 2 | Week 5 |
| 2.1.2.2 | Create Mock-up/Prototypes | 3 | Week 5 |
| 2.1.2.3 | Create Class Diagram | 4 | Week 5 |
| 2.1.2.4 | Create Sequence Diagram | 5 | Week 5 |
| **2.1.3** | **Coding** |  |  |
| 2.1.3.1 | Home | 3 | Week 6 |
| 2.1.3.2 | User Login | 3 | Week 6 |
| 2.1.3.3 | Reset Password | 6 | Week 6 |
| 2.1.3.4 | Change Password | 3 | Week 6 |
| 2.1.3.5 | View List Medical Supplies | 3 | Week 6 |
| 2.1.3.6 | Add Medical Supplies | 4 | Week 6 |
| 2.1.3.7 | View List Usage of Medical Supplies | 5 | Week 6 |
| 2.1.3.8 | View Medical Supplies Quantity | 3 | Week 6 |
| 2.1.3.9 | Medical Supplies Usage Report | 5 | Week 6 |
| 2.1.3.10 | Medical Supplies Import/Export Report | 3 | Week 6 |
| 2.1.3.11 | Medical Supplies Consumption Settlement Report | 3 | Week 6 |
| 2.1.3.12 | View Medicines List | 3 | Week 6 |
| 2.1.3.13 | View List of Medicines in Inventory | 3 | Week 6 |
| 2.1.3.14 | Add Medicine to Inventory | 5 | Week 6 |
| 2.1.3.15 | Dispense Medical Supplies | 5 | Week 6 |
| 2.1.3.16 | Medicine Usage Report | 3 | Week 6 |
| 2.1.3.17 | Medicine Import/Export Report | 3 | Week 6 |
| 2.1.3.18 | View Quantity of Medicines in Pharmacy Inventory | 3 | Week 6 |
| **2.1.4** | **Testing** |  |  |
| 2.1.4.1 | Create Test Cases | 8 | Week 6 |
| 2.1.4.2 | Execute Unit Tests | 8 | Week 6 |
| 2.1.4.3 | Execute Integration Tests | 8 | Week 6 |
| 2.1.4.4 | Create Test Report | 4 | Week 6 |
| **2.2** | **Common & Doctor Features** |  | Week 7- 8 |
| **2.2.1** | **Requirements Analysis** |  |  |
| 2.2.1.1 | Collect User Requirements | 4 | Week 7 |
| 2.2.1.1 | Identify Main Features | 4 | Week 7 |
| 2.2.1.1 | Update SRS | 4 | Week 7 |
| **2.2.2** | **Design** |  |  |
| 2.2.2.1 | Update Package Diagram | 2 | Week 7 |
| 2.2.2.1 | Create Mock-up/Prototypes | 3 | Week 7 |
| 2.2.2.1 | Create Class Diagram | 4 | Week 7 |
| 2.2.2.1 | Create Sequence Diagram | 4 | Week 7 |
| **2.2.3** | **Coding** |  |  |
| 2.2.3.1 | User Profile | 3 | Week 8 |
| 2.2.3.2 | User Authorization | 7 | Week 8 |
| 2.2.3.3 | View Medical Records | 3 | Week 8 |
| 2.2.3.4 | View Medical Record Histories of Patient | 3 | Week 8 |
| 2.2.3.5 | View Prescription Details | 3 | Week 8 |
| 2.2.3.6 | Create Prescription | 5 | Week 8 |
| 2.2.3.7 | Edit Prescription | 5 | Week 8 |
| 2.2.4 | Testing |  |  |
| 2.2.4.1 | Create Test Cases | 8 | Week 8 |
| 2.2.4.2 | Execute Unit Tests | ***8*** | Week 8 |
| 2.2.4.3 | Execute Integration Tests | 8 | Week 8 |
| 2.2.4.4 | Create Test Report | 4 | Week 8 |
| **2.3** | **Iteration 3: Doctor, Pharmacist, Admin Feature** |  | Weeks 9-10 |
| **2.3.1** | **Requirements Analysis** |  |  |
| 2.3.1.1 | Collect User Requirements | 4 | week 9 |
| 2.3.1.2 | Identify Main Features | 4 | week 9 |
| 2.3.1.3 | Update SRS | 4 | week 9 |
| **2.3.2** | **Design** |  |  |
| 2.3.2.1 | Update Package Diagram/System Architecture | 2 | week 9 |
| 2.3.2.2 | Create Mock-up/Prototypes | 3 | week 9 |
| 2.3.2.3 | Create Class Diagram | 4 | week 9 |
| 2.3.2.4 | Create Sequence Diagram | 4 | week 9 |
| **2.3.3** | **Coding** |  |  |
| 2.3.3.1 | Add Medical Record | 5 | week 10 |
| 2.3.3.2 | Add Medical Record History | 4 | week 10 |
| 2.3.3.3 | Edit Medical Record History | 4 | week 10 |
| 2.3.3.4 | View Today’s Medical Record History | 3 | week 10 |
| 2.3.3.5 | View List of Prescriptions in Medical Record History | 3 | week 10 |
| 2.3.3.6 | View List of External Prescriptions | 3 | week 10 |
| 2.3.3.7 | View External Prescription Details | 3 | week 10 |
| 2.3.3.8 | Create External Prescription | 5 | week 10 |
| 2.3.3.9 | Edit External Prescription | 5 | week 10 |
| 2.3.3.10 | View List of Medical Supply Usage Forms | 3 | week 10 |
| 2.3.3.11 | View Medical Supply Usage Form Details | 3 | week 10 |
| 2.3.3.12 | Create Medical Supply Usage Form | 5 | week 10 |
| 2.3.3.13 | Edit Medical Supply Usage Form | 5 | week 10 |
| 2.3.3.14 | List of Medicine Dispensing Orders | 3 | week 10 |
| 2.3.3.15 | Dispense Medicine | 5 | week 10 |
| 2.3.3.16 | List of Medical Supply Dispensing Orders | 3 | week 10 |
| 2.3.3.17 | View user list | 3 | week 10 |
| 2.3.3.18 | Add User | 4 | week 10 |
| **2.3.4** | **Testing** |  |  |
| 2.3.4.1 | Create Test Cases | 8 | week 10 |
| 2.3.4.2 | Execute Unit Tests | 8 | week 10 |
| 2.3.4.3 | Execute Integration Tests | 8 | week 10 |
| 2.3.4.4 | Create System Test Cases | 4 | week 10 |
| 2.3.4.5 | Execute System Tests | 8 | week 10 |
| **3** | **Verification** |  |  |
| 3.1 | Create Report 6: User Guides (Installation Guides, User Manuals) | 7 | Week 11 |
| 3.2 | Complete and Submit UT, IT, ST, AT Reports | 6 | Week 11 |
| 3.3 | Create Acceptance Test Support | 5 | Week 11 |
| 3.4 | Final Software Package (Documents, Source Codes, etc.) | 8 | Week 12-13 |
| **4** | **Closing** |  |  |
| 4.1 | Create Project Final Report | 5 | Week 14 |
| 4.2 | Make Slides for Presentation | 4 | Week 14 |

*Table 2: Cost and Time Estimations*

### 1.2 Project Objectives

| **#** | **Metric** | **Unit** | **Planned** | **Actual** | **Notes / References** |
| --- | --- | --- | --- | --- | --- |
| 1 | Effort Usage | Person-day | 350 pds | TBD | (5 people x 14 weeks x 5 working days/week) |
| 2 | Review Defects | No of defects | 15 | TBD | Errors found in code review |
| 3 | Unit Test Defects | No of defects | 20 | TBD | Errors detected through Unit Test (xUnit) |
| 4 | Integration Test Defects | No of defects | 15 | TBD | Errors detected during API integration testing (Postman, Swagger) |
| 5 | System Test Defects | No of defects | 12 | TBD | Errors detected in System Test |
| 6 | Acceptance Test Defects | No of defects | 10 | TBD | Bugs discovered by customers or test users |
| 7 | Timeliness | % | >= 90% | TBD | Total on-time deliverables / Total project deliverables |
| 8 | Requirement Completeness | % | ≥ 85% | TBD | Failed if <=75% |
| 9 | Code Coverage (Unit Test) | % | ≥ 50% | TBD | Project test coverage goals |

*Table 3: Project Objectives*

### 1.3 Project Risks

| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| --- | --- | --- | --- | --- |
| 1 | Limited resources and  manpower | High | High | Promote teamwork and task sharing among team members. Prioritize critical tasks if necessary. |
| 2 | Challenges with new technology and unfamiliar environments. | Medium | Medium | Providing comprehensive training for all team members before commencing the project. |
| 3 | Team members may be absent or have limited time for communication with each other. | Medium | Medium | Schedule meetings every Monday to Saturday evening, and potential overtime on Sundays if necessary. Ensure that if anyone is absent, their tasks are completed and there are comprehensive notes. |
| 4 | Changes in requirements | High | Medium | Maintain continuous communication with the project lead to finalize requirements for each phase. Document changes and update project plans accordingly. |
| 5 | Team members may be involved in unpredictable or unavoidable accidents (traffic accident, family members’ passing), rendering them incapable of performing tasks | High | Medium | Team members who are still able are responsible for dividing the task(s) assigned to the incapacitated member(s) amongst themselves during their recovery period. After the member(s) have recovered, the team will temporarily increase time spent per day on the project to keep up with the timeline. Depending on the severity of the issue, the scope of the project might be reduced to accommodate for the lack of manpower. |

*Table 4: Project Risks*

## 2. Management Approach

### 2.1 Quality Management

**2.1.1 Quality Objectives**

Ensure that the software meets the requirements, including the completeness of the requirements, the quality of the source code, and the level of stability.

Check the quality through the evaluation process, testing the software at many levels.

Ensure the source code has strict version control, limiting errors arising from loss of synchronization.

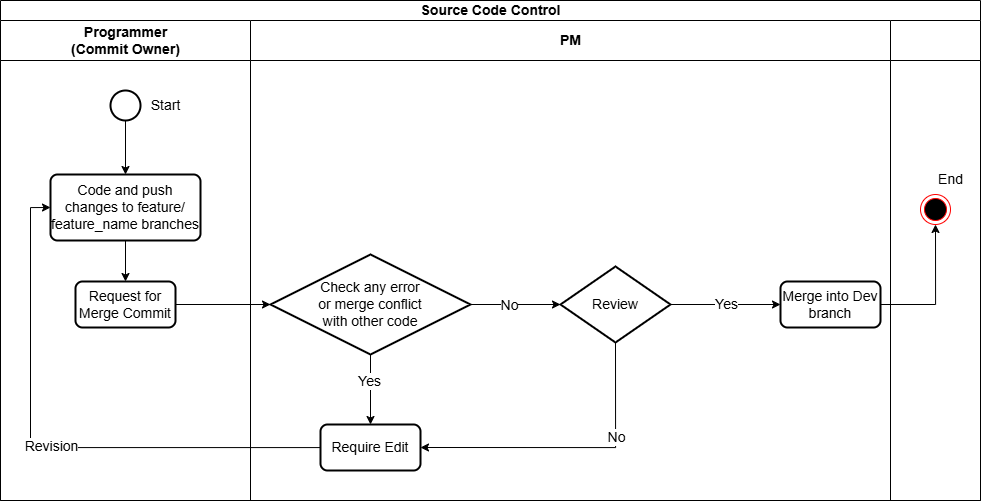
**2.1.2 Quality Assurance Process**

**2.1.2.1 Source Code Control**

Use GitHub for source control.

Require commits to follow clear naming conventions (feature/bugfix).

Mandatory code review before merging into the main branch.

****

*Figure 1 – Source Code Control Process*

The diagram above describes the source code control process between the programmer and the project manager. The process starts when the programmer writes code and pushes it to the feature branch. Then, they submit a merge commit. The PM or the system checks for errors and conflicts in the source code. If there are errors, the programmer needs to edit and resubmit the request. If there are no errors, the PM reviews the source code. If the code does not meet the requirements, the programmer needs to fix it. When the source code meets the standards, it is merged into the Dev branch, and the process ends. This ensures quality control before integrating it into the main system.

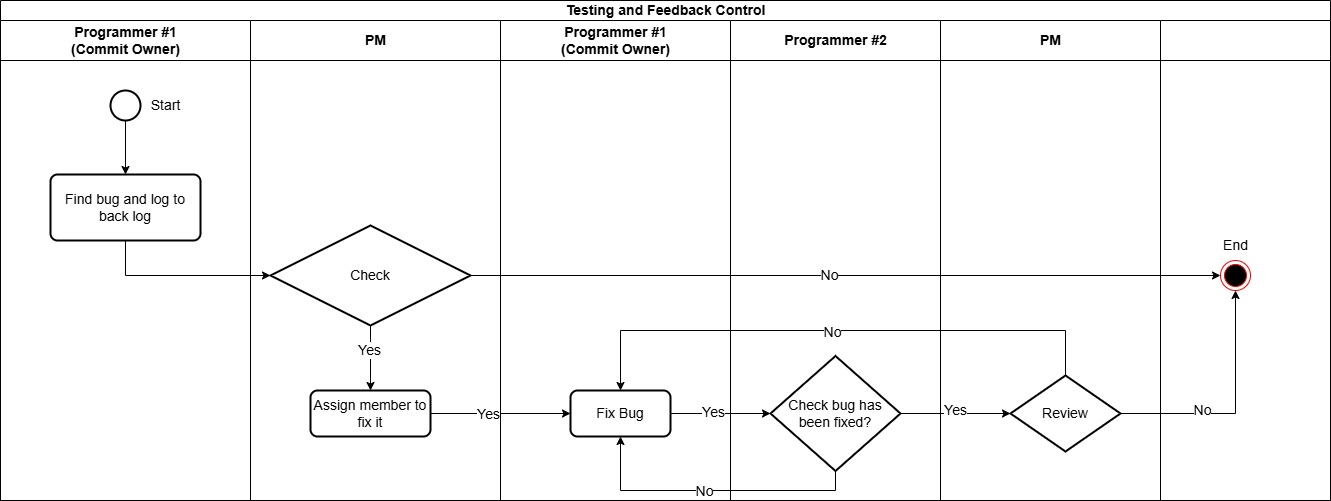
**2.1.2.2 Testing and Feedback Control**

Unit Test: Performed with xUnit / MSTest to ensure each component is working properly.

Integration Test: API testing using Postman, Swagger.

System Test: Run on a simulated environment to test the entire system.

Acceptance Test: Performed with the customer to ensure the system meets the requirements.

****

*Figure 2 – Testing and Feedback Control Process*

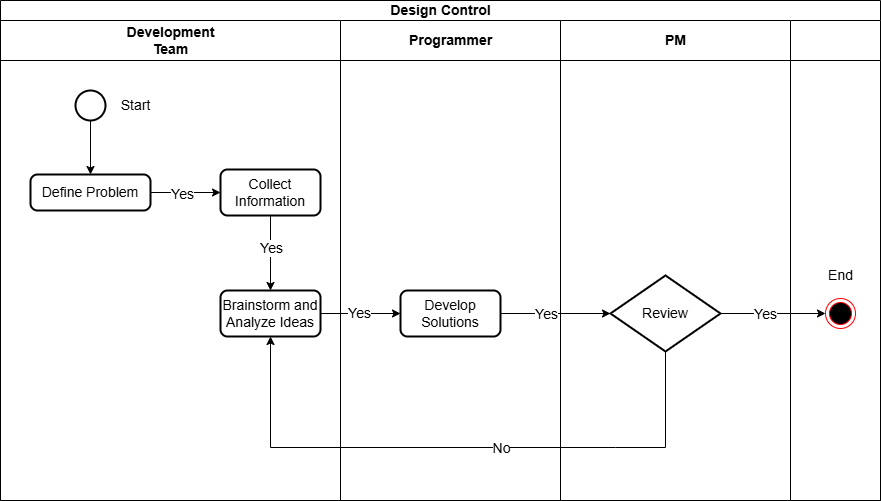
The diagram above depicts the Testing and Feedback Control process, which involves the steps of bug handling between the programmer and the project manager. The process begins when Programmer #1 finds a bug and logs it in the backlog. The PM checks the bug, if it needs fixing, they assign a programmer to fix it. Programmer #1 fixes the bug, then Programmer #2 checks if the bug is fixed. If not, the bug is sent back for further fixing. If the bug is fixed, the PM reviews it. If it passes, the process ends; otherwise, the bug goes back to the fix step. This process ensures that all bugs are logged, fixed, and confirmed before closing.

**2.1.2.3 Design Control Process**

Design review: Check the design before implementation.

Agree on coding convention: Follow common coding standards within the team.

Periodically review the architecture: Adjust the design if necessary.



*Figure 3 – Design Control Process*

The diagram above shows the Design Control Process with three main roles: Development Team, Programmer, and PM (Project Manager). The process starts with the development team defining the problem and collecting information to understand the requirements. The team then brainstorms and analyzes ideas, from which the programmer develops solutions. The solutions are sent to the project manager for review. If the solution is approved, the process ends. If not, the solution is returned for improvement, and the process is repeated. This process ensures that design solutions are developed systematically, meet requirements, and are thoroughly tested before implementation.

### 

### 2.2 Project Training Plan

| **Training Area** | **Participants** | **When, Duration** |
| --- | --- | --- |
| HTML, CSS, JS | Everyone | 10/1/2025 14h-17h |
| Selenium | Everyone | 11/1/2025  8h-11h30p |
| Coding-Convention | Everyone | 11/1/2025  20h-22h |
| Contestant guide for using the platform | End-users | 14/4/2025 |
| Q&A and Support Sessions | End-users |  |

## 

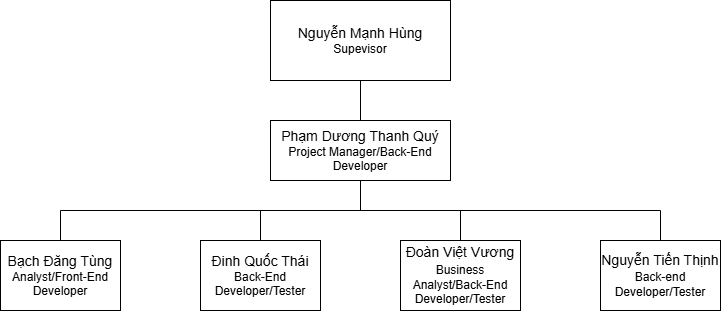
*Table 5: Project Training Plan*

## 3. Project Deliverables

| **#** | **Deliverable** | **Due Date** | **Notes** |
| --- | --- | --- | --- |
| 1 | Report 1 –Project Introduction | 8/01/2025 | Project Overview, Background, Vision... |
| 2 | Report 2 – Project Management Plan | 15/01/2025 | Scope & Estimations, Management Approach, Deliverables... |
| 3 | Report 3 – Software Requirement Specification | 03/04/2025 | Overall Requirements, Use Case Specifications, Functional Requirements... |
| 4 | Report 4 – Software Design Specification | 03/04/2025 | System Design, Database Design, Detailed Design... |
| 5 | Iteration 1(Code,Test and Implement) | 28/02/2025 | Coding and Testing |
| 6 | Iteration 2(Code,Test and Implement) | 30/03/2025 | Coding and Testing |
| 7 | Iteration 3(Code,Test and Implement) | 20/04/2025 | Coding and Testing |
| 8 | Report 5 – Test Document | 24/04/2025 | Testing Scope, Test Strategy, Test Cases... |
| 9 | Report 6 – Software User Guide | 24/04/2025 | Release Package, User Guide |
| 10 | Report 7 – Final Report | 25/04/2025 | All the above |

*Table 6: Project Deliverables*

## 4. Responsibility Assignments

**

*Figure 4 – RACI Chart*

**RACI Chart**: R~Responsible, A~Accountable, C~Consulted, I~Informed

| **Work Package** | **Quypdthe163964** | **Tungbdhe172875** | **Vuongdvhe160717** | **Thaidqhe170966** | **Thinhnthe171628** |
| --- | --- | --- | --- | --- | --- |
| Requirements analysis | A | I | R | R | C |
| Report 1: Project Introduction | A | R | C | I | I |
| Report 2: Project Management Plan | A | R | C | I | I |
| Report 3: SRS | A | R | R | R | C |
| Report 4: SDS | A | R | R | R | C |
| Report 5: Test Documentation | A | C | I | R | C |
| Report 6: Software User Guides | A | R | R | C | I |
| Report 7: Final Project Report | A | R | R | C | I |
| Back-end Building | A | R | R | R | R |
| Front-end Building | A | R | I | I | I |
| Testing | A | I | I | R | C |

*Table 7: Responsibility Assignment*

## 

## 5. Project Communications

In our project, we communicate in several ways, for several purposes. The table below shows the communication item, participating members, its purposes, time, meeting frequency, and methods.

| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| --- | --- | --- | --- | --- |
| Quick message  Group chat | All team members | Casual chat  Schedule meetings  Report progress | Daily | Messenger, Zalo |
| Daily Meeting | All team members | Check progress and assign a new task to team members | 8:30 PM every day except Tuesday and Sunday | Online via Google Meet |
| Tasks Assignment | All team members | The project manager allocates tasks to each team member | Every day | Google Sheet |
| Weekly Meeting with Supervisor | All team members + Supervisor | Review project status and discuss potential issues | 8 PM every Tuesday | Online via Google Meet |
| Emergency Meeting | All team members | Discussing and resolving urgent issues | When a problem that needs to be dealt with immediately | Online via Google Meet |

*Table 8: Project Communications*

## 6. Configuration Management

### 6.1 Document Management

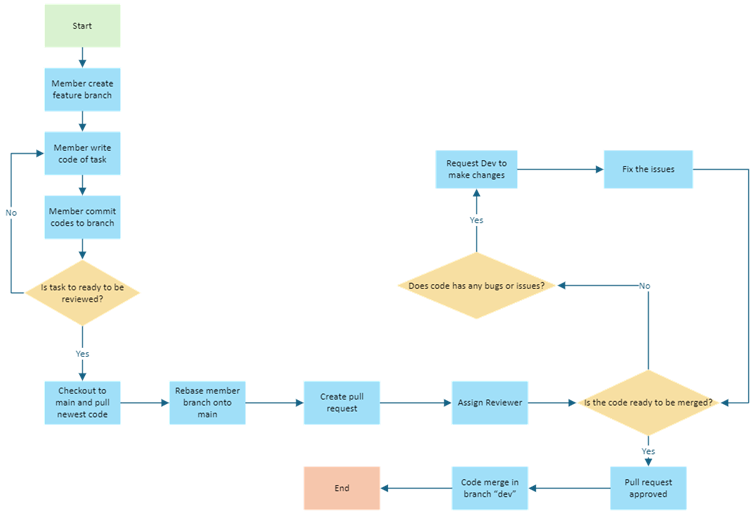
All documents will be managed in OneDrive with the following structure:

* Folder:
  + **Diagram:** Diagrams needed for documentation
  + **Management**: Spreadsheets for more detailed management
  + **Meeting Minutes:** Summary of meetings with mentor
  + **Record:** Recordings of meetings with the mentor
  + **Report:** Reports for submissions for review

### 

### 6.2 Source Code Management

Team members use GitHub for source code and version control, following this workflow:



*Figure 5 – Source Code Management*

#### **6.2.1. Convention**

* All coding branches must be reviewed by team developers before merging into the main branch.
* Team members are only allowed to commit code to the local repository.
* Code must be fully tested and can run well on the local side before sending pull requests to the master repository for finished implementations.

#### 6.2.2. Management

* Source code managed by GitHub.
* The technique leader is the person who must review carefully before merging the code of team members.
* Only team members and mentors can access the repository on GitHub.

### 

### 

### 

### 

### 

### 

### 6.3 Tools & Infrastructures

| **Category** | **Tools / Infrastructure** |
| --- | --- |
| **Technology** | NET C # (Backend), HTML, CSS, JS(Frontend) |
| **Database** | Microsoft SQL Server Management Studio |
| **IDEs/Editors** | Visual Studio Code, Visual Studio |
| **Diagramming** | Draw.io, Visual Paradigm, Plant UML Web Server, SequenceDiagram.org |
| **Documentation** | Microsoft Office, Google Docs, Google Sheets |
| **Version Control** | GitHub (Source Codes), OneDrive (Documents) |
| **Deployment server** | Azuze, Vietnix, Docker |
| **Project management** | GitHub (Defects, Issues), Google Drive |

*Table 9: Tools and Infrastructures*