|  |  |
| --- | --- |
|  | **MINISTRY OF**  **EDUCATION AND TRAINING** |

**CAPSTONE PROJECT REPORT**

**Report 2 – Project Management Plan**

- Hanoi, December 2022 -

Table of Contents

[Acronyms and Definitions 1](#_Toc90846078)

[Record of changes 1](#_Toc90846079)

[Report II. Project Management Plan 2](#_Toc90846080)

[1. Overview 2](#_Toc90846081)

[1.1. WBS & Estimation 2](#_Toc90846082)

[1.2. Project Objectives 6](#_Toc90846083)

[1.2.1. Project Objectives 6](#_Toc90846084)

[1.2.2. Team Objectives 6](#_Toc90846085)

[1.3. Project Risks 6](#_Toc90846086)

[2. Management Approach 7](#_Toc90846087)

[2.1. Project Process 7](#_Toc90846088)

[2.2. Meeting Minutes 7](#_Toc90846089)

[2.3. Quality Management 8](#_Toc90846090)

[2.4. Training Plan 8](#_Toc90846091)

[3. Master Schedule 10](#_Toc90846092)

[4. Project Organization 11](#_Toc90846093)

[4.1. Team & Structures 11](#_Toc90846094)

[4.2. Roles & Responsibilities 11](#_Toc90846095)

[5. Project Communication 13](#_Toc90846096)

[5.1. Communication Plan 13](#_Toc90846097)

[5.2. External Interface 13](#_Toc90846098)

[6. Configuration Management 14](#_Toc90846099)

[6.1. Tools & Infrastructures 14](#_Toc90846100)

[6.2. Document Management 14](#_Toc90846101)

[6.2.1. Convention 14](#_Toc90846102)

[6.2.2. Management 14](#_Toc90846103)

[6.3. Source Code Management 15](#_Toc90846104)

[6.3.1. Convention 15](#_Toc90846105)

[6.3.2. Management 15](#_Toc90846106)

# Acronyms and Definitions

|  |  |  |
| --- | --- | --- |
| **#** | **Acronym** | **Definition** |
| 1 | WBS | Work Breakdown Structure |
| 2 | SRS | Software Requirement Specification |
| 3 | UX | User Experience |
| 4 | UI | User Interface |
| 5 | VPS | Virtual Private Server |
| 6 | API | Application Programming Interface |
| 7 | DBMS | Database Management System |
| 8 | UML | Unified Modeling Language |

Acronyms and Definitions Table

# Record of changes

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Status** | **In charge** | **Change description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

A – Added, M – Modified, D – deleted

# Report II. Project Management Plan

## 1. Overview

### 1.1. WBS & Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **WBS Item** | **Complexity** | **Est. Effort (Man-days)** |
| ***1*** | ***Pre-Initiating Phase*** |  | ***20*** |
| 1.1 | Select project manager | Simple | 2 |
| 1.2 | Determine the role of team members | Simple | 2 |
| 1.3 | Choose topic | Complex | 15 |
| 1.4 | Register Capstone Project | Simple | 1 |
| ***2*** | ***Initiating Phase*** |  | ***32*** |
| 2.1 | Kick-off meeting | Simple | 4 |
| 2.2 | Develop project introduction document | Complex | 28 |
| ***3*** | ***Planning Phase*** |  | ***50*** |
| **3.1** | **Project Planning** |  | ***20*** |
| 3.1.1 | Define scope | Medium | 3 |
| 3.1.2 | Choose working model process | Simple | 1 |
| 3.1.3 | Choose software process model | Simple | 1 |
| 3.1.4 | Select tools and techniques | Simple | 1 |
| 3.1.5 | Develop WBS | Medium | 4 |
| 3.1.6 | Develop project schedule | Medium | 3 |
| 3.1.7 | Develop risk management plan | Medium | 1 |
| 3.1.8 | Develop project communication plan | Simple | 1 |
| 3.1.9 | Organize project resources | Simple | 1 |
| 3.1.10 | Develop software project management plan | Medium | 4 |
| **3.2** | **Project Preparing** |  | ***5*** |
| 3.2.1 | Create software mockup | Complex | 3 |
| 3.2.2 | Technical Training/Self study | Simple | 2 |
| **3.3** | **Analyzing & Requirement** |  | ***25*** |
| 3.3.1 | User Requirements | Medium | 3 |
| 3.3.2 | Functional Requirements | Medium | 3 |
| 3.3.3 | Create Entity Relationship Diagram | Complex | 5 |
| 3.3.4 | Non-Functional Requirements | Simple | 2 |
| 3.3.5 | Create test cases for system test | Medium | 3 |
| 3.3.6 | Develop Software Requirement Specification | Complex | 9 |
| ***4*** | ***Executing*** |  |  |
| **4.1** | **Designing** |  | ***40*** |
| 4.1.1 | Basic architecture design | Medium | 3 |
| 4.1.2 | Create test cases for integration test | Medium | 5 |
| 4.1.3 | Detail design | Medium | 6 |
| 4.1.4 | Data & Database Design | Medium | 6 |
| 4.1.5 | Create test cases for unit test | Medium | 5 |
| 4.1.6 | Develop Software Design Document | Medium | 15 |
| **4.2** | **Implementation - Iteration 1: Develop user features** |  | ***72*** |
| 4.2.1 | *Requirements* |  | **3** |
| 4.2.1.1 | Update Software Requirement Specification | Medium | 3 |
| 4.2.2 | *Design* |  | ***4*** |
| 4.2.2.1 | Update Software Design Document | Medium | 4 |
| 4.2.3 | *Coding* |  | **56** |
| 4.2.3.1 | Feature 1: Authentication user |  | *5* |
| 4.2.3.1.1 | Code front-end | Medium | 3 |
| 4.2.3.1.2 | Code backend | Complex | 2 |
| 4.2.3.2 | Feature 2: Manage personal profile |  | *5* |
| 4.2.3.2.1 | Code front-end | Medium | 3 |
| 4.2.3.2.2 | Code backend | Medium | 2 |
| 4.2.3.2 | Feature 3: Receive diet |  | *24* |
| 4.2.3.2.1 | Code front-end | Medium | 6 |
| 4.2.3.2.2 | Code backend | Complex | 18 |
| 4.2.3.3 | Feature 4: Search food |  | *5* |
| 4.2.3.3.1 | Code front-end | Medium | 3 |
| 4.2.3.3.2 | Code backend | Medium | 2 |
| 4.2.3.4 | Feature 5: Change personal diet |  | *12* |
| 4.2.3.4.1 | Code front-end | Medium | 3 |
| 4.2.3.4.2 | Code backend | Complex | 9 |
| 4.2.3.5 | Feature 6: Manage food |  | *5* |
| 4.2.3.5.1 | Code front-end | Medium | 3 |
| 4.2.3.5.2 | Code backend | Medium | 2 |
| 4.2.4 | *Testing* |  | **15** |
| 4.2.4.1 | Unit test | Medium | 10 |
| 4.2.4.2 | Integration test | Medium | 5 |
| 4.2.5 | *Evaluation* | Simple | **5** |
| **4.3** | **Implementation - Iteration 2: Develop admin features** |  | ***35*** |
| 4.3.1 | *Requirements* |  | **4** |
| 4.3.1.1 | Update Software Requirement Specification | Medium | 4 |
| 4.3.2 | *Design* |  | **4** |
| 4.3.2.1 | Update Software Design Document | Medium | 4 |
| 4.3.3 | *Coding* |  | **6** |
| 4.3.3.1 | Feature 1: Manage system account |  | 6 |
| 4.3.3.1.1 | Code front-end | Medium | 3 |
| 4.3.3.1.2 | Code backend | Medium | 3 |
| 4.3.4 | *Testing* |  | **10** |
| 4.3.4.1 | Unit test | Medium | 5 |
| 4.3.4.2 | Integration test | Medium | 5 |
| 4.3.5 | *Evaluation* | Simple | **5** |
| 4.4 | **Full System testing** |  | **30** |
| 4.4.1 | Execute Full System Test & Fix Bugs | Medium | 25 |
| 4.4.2 | Update Software Test Report Document | Medium | 5 |
| 4.5 | **Transitioning** |  | **45** |
| 4.5.1 | *Software Optimization* |  | *30* |
| 4.5.1.1 | Optimize/Refactor Codes | Medium | 5 |
| 4.5.1.2 | Execute Full System Test Round 2 & Fix Bugs | Simple | 15 |
| 4.5.2 | *User Guides* |  | *5* |
| 4.5.2.1 | Develop Software User Guides | Medium | 5 |
| 4.5.2.2 | Develop Final Project Report | Medium | 10 |
| ***5*** | ***Monitoring and Controlling*** |  | ***5*** |
| 5.1 | Meet instructor | Medium | 2 |
| 5.2 | Progress Report | Complex | 3 |
| ***6*** | ***Closing*** |  | ***25*** |
| 6.1 | Prepare for final project presentation | Medium | 15 |
| 6.2 | Present final project | Medium | 5 |
| 6.3 | Project completed | Simple | 5 |
| **Total Estimated Effort (man-days)** | | | **292** |

Table 1.1. WBS & Estimation

### 1.2. Project Objectives

#### 1.2.1. Project Objectives

* This product is a mobile application, suitable for anglers and fishing spots owners who want to connect to each other’s.
* This product has the main features:
  + Provide information about fishing spots.
  + Manage fishing spots information and staff.

#### 1.2.2. Team Objectives

* The team finishes the project before 17th December, 2022.
* Team members improve their skills in working in a group.
* Each member improves both technical skills and soft skills.
* Team successfully applied a software development process.

### 1.3. Project Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | Member illness | High | Medium | Other members should join forces to fulfill the work of the absent members. |
| 2 | Conflict among members | Medium | Medium | All members should discuss ways to resolve the conflict. |
| 3 | Lack of skill | High | Medium | Members should be given time to train in their field before coding. |
| 4 | Ambiguous business requirement | High | High | Study about other existing systems.  Study about the needs of end-users.  State the requirements clearly in documents.  Analyze each requirement carefully. |
| 5 | Facing hard problems | High | High | All members should discuss the problems and find out the best solution.  Ask the supervisor and people who have experience and knowledge about the problems. |
| 6 | Requirement changes | High | Medium | All members should discuss carefully  at the beginning of each iteration to  define scope and requirement. |

Table 1.3. Project Risks

## 2. Management Approach

### 2.1. Project Process

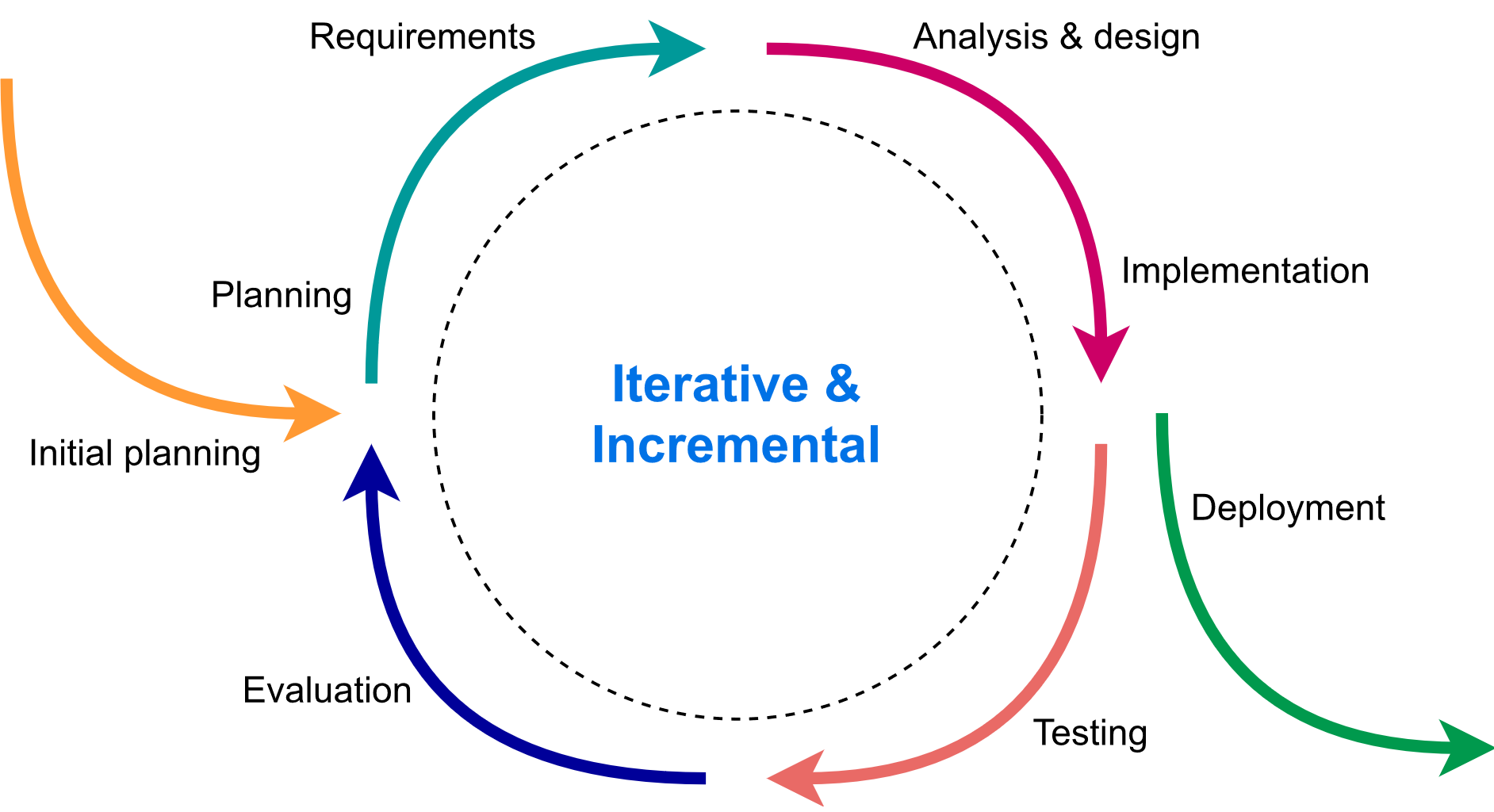


Figure 2.1. Iterative and Incremental software process model

We use the Iterative and Incremental software process model - a method in which the system is developed through repeated cycles and in smaller portions at a time. This model allows software developers to take advantage of what was learned during the development of earlier system parts. At each iteration, design modifications are made, and new functional capabilities are added.

This project will be divided into two iterations. In each iteration, we are going to finish and deliver a code package of essential features.

### 2.2. Meeting Minutes

All meeting minutes will follow this template:

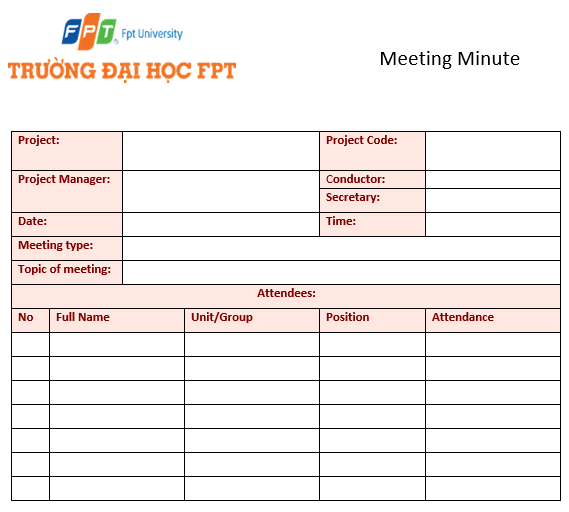


Figure 2.2. Meeting minute template

### 2.3. Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| Java Spring Boot | Nguyen Hoang Viet  Nguyen Viet Long | 27/9/2022, 1 week | Mandatory |
| ReactJS | Nguyen Duc Son  Le Manh Cuong | 27/9/2022, 1 week | Mandatory |
| GitLab | Nguyen Hoang Viet  Nguyen Viet Long  Nguyen Duc Son  Le Manh Cuong | 27/9/2022, 1 week | Mandatory |
| Figma | Nguyen Duc Son  Le Manh Cuong | 22/09/2022, 1 week | Mandatory |

Table 2.4. Training Plan

## 3. Master Schedule

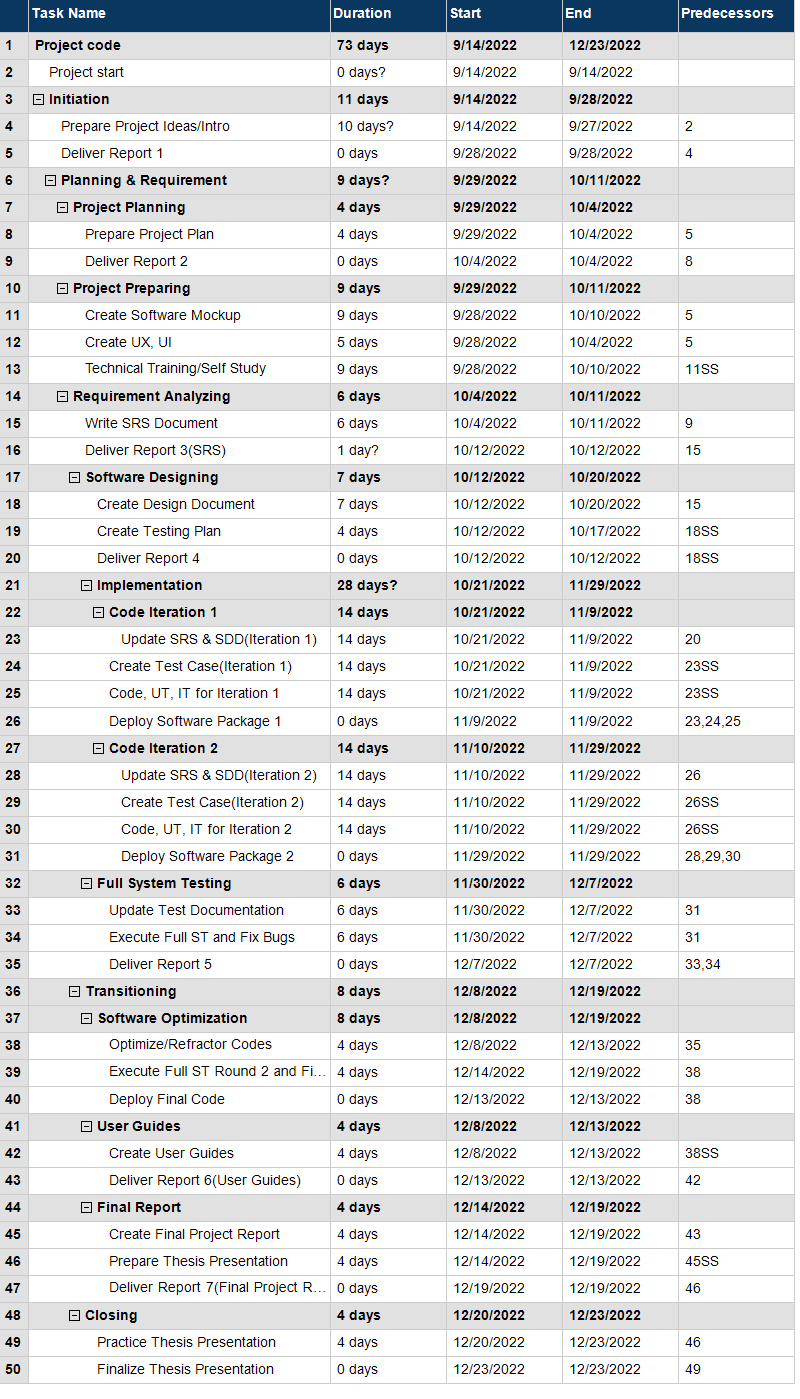


Figure 3.1 Master Schedule

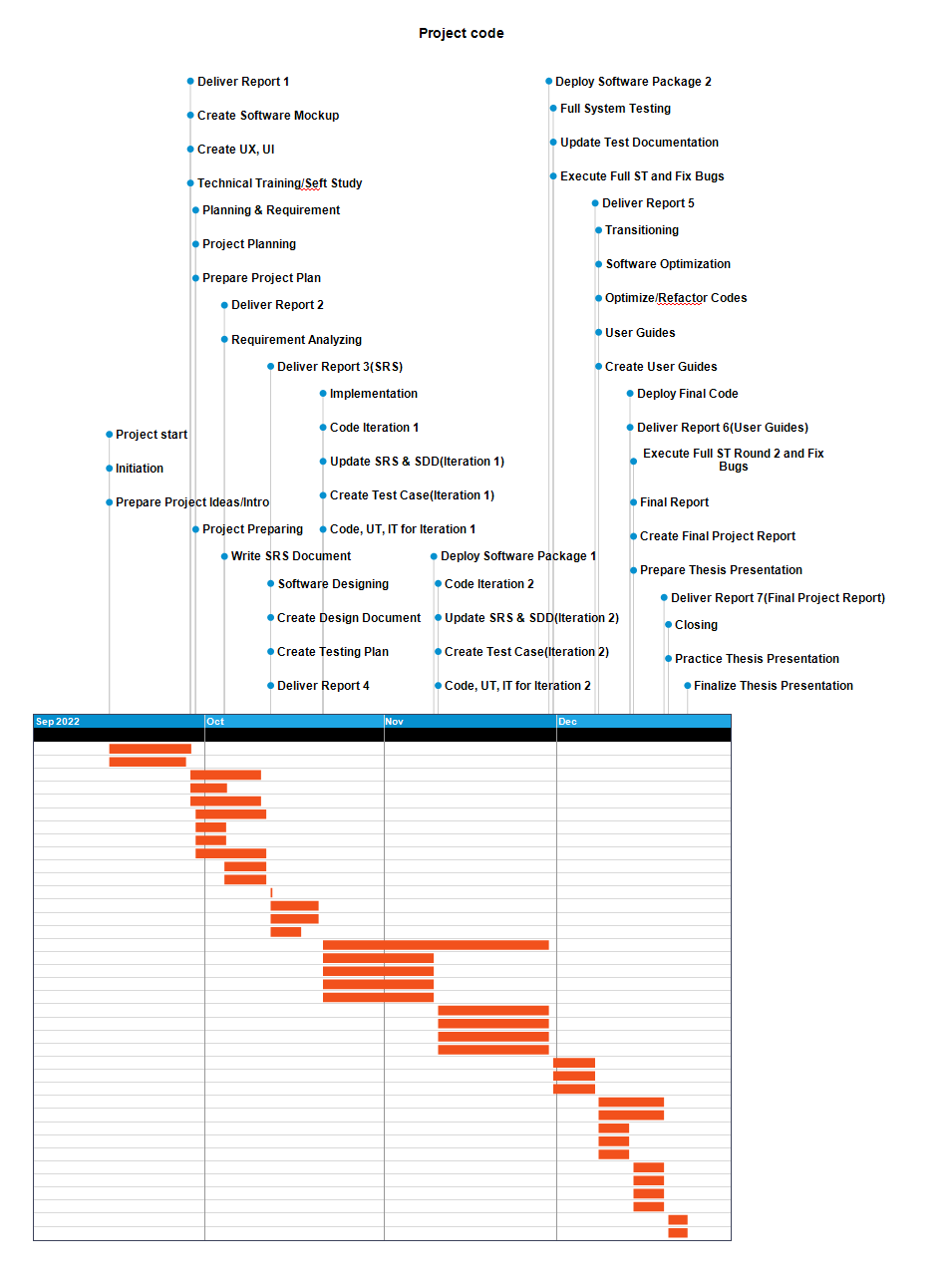


Figure 3.2 Master Schedule(Gantt Chart Timeline)

## 

## 4. Project Organization

### 4.1. Team & Structures

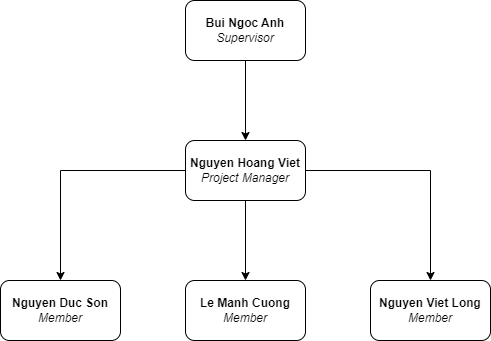


Figure 4.1. Team & Structures Diagram

### 4.2. Roles & Responsibilities

|  |  |  |
| --- | --- | --- |
| **Role** | **Member** | **Responsibility** |
| Project Manager | Nguyen Hoang Viet | Responsible for planning, creating schedules, coordinating communication.  Keeping track of the progress of the project.  Take responsibility with all requirement and schedule of school.  Keep project in direct goals. |
| Analysis Leader | Nguyen Hoang Viet | Analyze project requirement  Define project scope, create SRS document  Review SRS created by analysis members  Study and research about fishing activity |
| Analysis Member | Nguyen Viet Long  Nguyen Duc Son  Le Manh Cuong | Define requirements: functional and non-functional requirements  Keep track and contribute to SRS document  Define detail flow of features  Review other members and comment  Study and research about fishing activity |
| Designer | Nguyen Hoang Viet  Nguyen Duc Son | Define screens, flow between screens  Design UI, UX for application |
| Technical Leader | Nguyen Viet Long | Responsible for choosing technologies for the whole system and overseeing the work done by other programmers |
| Frontend Developer | Nguyen Duc Son  Le Manh Cuong | Initial and develop frontend  Study technical requirement to apply into  project  Review code for each other  Fix bugs |
| Backend Developer | Nguyen Hoang Viet  Nguyen Viet Long | Design database  Study technical requirement to apply into  project  Initial and develop backend server  Review code for each other  Fix bugs  Deploy code to server |
| Test Leader | Nguyen Hoang Viet | Define test plan  Assign tasks for other member  Study and research testing tools  Create test template  Take responsibility about coverage of test |
| Test Member | Le Manh Cuong | Create test cases  Implement tests follow test plan  Create test reports and inform development teams about bugs |

Table 4.2. Roles & Responsibilities

## 5. Project Communication

### 5.1. Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| Weekly meeting | Team members, supervisor | Review member’s work achievements.  Report the project's progress and status of each member’s work. | Every Tuesday | Google Meet |
| Daily meeting | Team members | Report the project's progress and status of each member’s work each day. | Every weekday | Google Meet |
| Unscheduled meeting | Team members | Resolve unpredicted problems | When any problems arise | Google Meet, Messenger |
| Supervisor daily communication | Team member, supervisor | Report project status, Q&A | Every weekday | Messenger |
| Team daily communication | Team members | Discuss project, sharing files, sharing source code | Every weekday | Discord, Messenger |

Table 5.1. Communication Plan

### 5.2. External Interface

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Contact Person**  **(name, position)** | **Contact address**  **(email, telephone)** | **Responsibility** |
| Supervisor | Bui Ngoc Anh | [anhbn5@fe.edu.vn](mailto:anhbn5@fe.edu.vn) | * Provide document template * Give instruction to project team * Review deliverables * Supervise project status |

Table 5.2. External Interface

## 6. Configuration Management

### 6.1. Tools & Infrastructures

|  |  |
| --- | --- |
| **Programming languages** | **Tools & Technical** |
| Framework | Java SpringBoot, React |
| Library | ReactJS |
| API | REST API |
| DBMS | MySQL Workbench |
| IDEs/Editors | Visual Studio Code, IntelliJ |
| Testing tools & Framework | IntelliJ, Postman |
| Design | Figma, Proto.io |
| UML tools | Draw.io |
| Version Control | GitLab |
| Deployment Server |  |
| Project Management Tools | Jira, MindView |

Table 6.1. Tools & Infrastructures

### 6.2. Document Management

#### 6.2.1. Convention

* Font family: Times New Roman
* Heading 1: 18pt Bold
* Heading 2: 16pt Bold
* Heading 3: 14pt Bold
* Heading 4: 12pt Bold
* Heading 5: 12pt Bold Italic
* Heading 6: 12pt Bold Italic Underline
* Image & Table Title: 10pt Italic
* Normal: 12pt

#### 6.2.2. Management

* All documents will be stored on Google Drive and Microsoft OneDrive.
* All reports will be delivered to the supervisor through Gmail.

### 6.3. Source Code Management

#### 6.3.1. Convention

* All coding branches must be reviewed by team developers before merging to the default branch.
* All developers must implement unit tests and review code before creating a merge request.

#### 6.3.2. Management

* All source code will be stored on GitLab.
* Only team members can access the repository on GitLab.