

**Capstone Project Report**

**Report 2 – Project Management Plan**

– Hanoi, September 2021 –

**Table of Contents**

[I. Project Report 3](#_Toc47946885)

[1. Status Report 3](#_Toc47946886)

[2. Team Involvements 3](#_Toc47946887)

[3. Issues/Suggestions 3](#_Toc47946888)

[II. Project Management Plan 4](#_Toc47946889)

[1. Overview 4](#_Toc47946890)

[1.1 WBS & Estimation 4](#_Toc47946891)

[1.2 Project Objectives 4](#_Toc47946892)

[1.3 Project Risks 4](#_Toc47946893)

[2. Management Approach 5](#_Toc47946894)

[2.1 Project Process 5](#_Toc47946895)

[2.2 Quality Management 5](#_Toc47946896)

[2.3 Training Plan 5](#_Toc47946897)

[3. Master Schedule 5](#_Toc47946898)

[4. Project Organization 6](#_Toc47946899)

[4.1 Team & Structures 6](#_Toc47946900)

[4.2 Roles & Responsibilities 6](#_Toc47946901)

[5. Project Communication 6](#_Toc47946902)

[5.1 Communication Plan 6](#_Toc47946903)

[5.2 External Interface 7](#_Toc47946904)

[6. Configuration Management 7](#_Toc47946905)

[6.1 Tools & Infrastructures 7](#_Toc47946906)

[6.2 Document Management 8](#_Toc47946907)

[6.3 Source Code Management 8](#_Toc47946908)

# I. Project Report

## 1. Status Report

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Work Item** | **Status** | **Notes (Work Item in Details)** |
| 1 | Overview | Completed |  |
| 2 | Management Approach | Completed |  |
| 3 | Master Schedule | Completed |  |
| 4 | Project Organization | Completed |  |
| 5 | Project Communication | Completed |  |
| 6 | Configuration Management | Completed |  |

## 2. Team Involvements

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task** | **Member** | **Notes (Task Details, etc.)** |
| 1 | Overview | HaiND |  |
| 2 | Management Approach | HieuMT |  |
| 3 | Master Schedule | DanHN |  |
| 4 | Project Organization | HuongDN |  |
| 5 | Project Communication | MinhNC |  |
| 6 | Configuration Management | MinhNC |  |

## 3. Issues/Suggestions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Issue** | **Status** | **Notes (Solution, Suggestion, etc.)** |
| 1 |  | Pending |  |
| 2 |  | In Progress |  |
| 3 |  | Completed |  |

# II. Project Management Plan

## 1. Overview

### 1.1 WBS & Estimation(Estimation sua theo menday)

|  |  |  |
| --- | --- | --- |
| **WBS Code** | **WBS Item** | **Duration** |
| **1** | **Initializing** | **2 days** |
| 1.1 | Kick-off Meeting | 1 day |
| 1.2 | Create report 1 | 2 days |
| **2** | **Planning** | **3 days** |
| 2.1 | Create report 2 | 2 days |
| 2.2 | Create Scope plan | 2 hours |
| 2.3 | Create Resources plan | 2 hours |
| 2.4 | Create Time plan | 2 hours |
| 2.5 | Create pre-development plan | 2 hours |
| 2.6 | Create Project Scheduling management | 1 day |
| 2.7 | Create project plan | 1 hour |
| **3** | **Predevelopment Training** | **60 days** |
| 3.1 | C# | 14 days |
| 3.2 | .Net | 21 days |
| 3.3 | Javascript | 7 days |
| 3.4 | React Js | 14 days |
| 3.5 | Bootstrap libraries | 1 day |
| 4 | **Requirement Analysis** | **10 days** |
| 4.1 | Requirement gathering | 2 days |
| 4.2 | Business Analysis | 5 days |
| 4.3 | Gathering business information | 2 days |
| 4.4 | Learning business in similar systems | 1 day |
| 4.5 | Create draft use-cases diagrams | 1 day |
| 4.6 | Validate use-cases diagram with Mentor | 1 day |
| 4.7 | Update use-cases diagram | 2 days |
| 4.8 | Requirements Analysis Completed | 1 day |
| 4.9 | Deliver SRS | 1 day |
| **5** | **Design** | **14 days** |
| 5.1 | Design Database | 4 days |
| 5.2 | High-level design | 2 days |
| 5.3 | Detailed Design / Modules Design | 2 days |
| 5.4 | Interface Design | 3 days |
| 5.5 | Technical Specification | 1 day |
| 5.6 | Design Completed | 2 days |
| **6** | **Development** | **65 days** |
| 6.1 | Iteration 1 | 23 days |
| 6.1.1 | Create report 3 | 8 days |
| 6.2 | Iteration 2 | 21 days |
| 6.2.1 | Create report 4 | 8 days |
| 6.3 | Iteration 3 | 21 days |
| 6.3.1 | Create report 5 | 8 days |
| **7** | **Implementation** | **10 days** |
| 7.1 | Create report 6 | 3 days |
| 7.2 | Development | 2 days |
| 7.3 | Training | 3 days |
| 7.4 | Create final report | 3 days |
| 7.5 | UAT | 8 days |
| 7.6 | Support | 3 days |
| 7.7 | Development/Implementation Completed | 2 days |

### 1.2 Project Objectives

* *Timeliness: The project must be finished before December 8, 2021:*
* *Allocated Effort (man-days): 101*
* *Defect Distribution:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Quality Stage** | **No. of Defects** | **% of Defect** | **Notes** |
| 1 | Reviewing |  |  | Technical leader reviews code of members before merging |
| 2 | Unit Test |  |  | Developer creates and tests |
| 3 | Integration Test |  |  | The tester creates and tests |
| 4 | System Test |  |  | The tester creates and tests |
| 5 | User Acceptance Test |  |  | Customer verifies system |
| ***Total*** | |  | ***100%*** |  |

### 1.3 Project Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | Requirement changes | Medium | Medium | All members discuss carefully the project requirements before starting implementation. |
| 2 | Members have  argued,  conflicted with  others, leads to  stressful  working  environments | Medium | High | Define clear tasks for each member and agree on ideas before starting work. |
| 3 | Illness or  absence of team  members so  that they cannot  complete tasks  under deadline | Low | Medium | Members have to notify the team about illness or absence period and the plan of how to keep up with the work process. |
| 4 | Members lack  the knowledge  and skills to  complete a  particular task | High | Medium | Training all members before starting the project. |
| 5 | The library used  in the project is  no longer  supported | Low | Low | Choose a reputable library with active maintenance on GitHub. |
| 6 | Internet connection is down and team members cannot submit work or merge code. | Low | Medium | All developers have to set up the isolated development environment and have an offline copy of the documentation. |

## 2. Management Approach

### 2.1 Project Process

### 2.2 Quality Management

**2.2.1 Defect Prevention:**

● If any defect is found, the related person must be notified immediately at that time.

● Defects must be carefully evaluated such as "How bad is the defect and can it damage

the system?", "How long is the time to fix that defect?".

● The deadline for fixing the defect must be specified clearly.

● There is always a plan to prepare for what could happen at any time.

**2.2.2 Reviewing:**

● The curator must be honest and not biased towards any of the project members. If

there is an error, the person must immediately notify the person responsible for the

defect.

● Defects should be recorded on the Bug Tracking software with details such as priority.

● The person responsible for defects found must-have solutions to fix the defect as

quickly as possible.

**2.2.3 Unit Testing:**

● The curator must prepare the test cases carefully and accurately and must not ignore

any cases. Test cases should be appropriate for the functionality of the system.

● Defects should be recorded on the Bug Tracking software with details such as priority.

● The person responsible for defects found must-have solutions to fix the defect as

quickly as possible.

**2.2.4 Integration Testing:**

● The curator must prepare the test cases carefully and accurately and must not ignore

any cases. Test cases should be appropriate for the functionality of the system.

● Defects should be recorded on the Bug Tracking software with details such as priority.

● The person responsible for defects found must-have solutions to fix the defect as

quickly as possible.

● Internal modules within the system work smoothly.

**2.2.5 System Testing:**

● The curator must prepare test cases carefully and accurately. The test cases must

match well with the system and system and architecture design.

● Defects should be recorded on the Bug Tracking software with details such as priority.

● The person responsible for defects found must-have solutions to fix the defect as

quickly as possible.

● System testing test cases cover the entire system functionality and the communication

under development with external systems.

**2.2.6 Acceptance Testing:**

● The curator must prepare test cases carefully and accurately. The test cases must

match well with the system and system and architecture design.

● Defects should be recorded on the Bug Tracking software with details such as priority.

● The person responsible for defects found must-have solutions to fix the defect as

quickly as possible.

The test should cover non-functional issues such as load and performance defects.

## 2.3 Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Training Area | Participants | When, Duration | Waiver Criteria |
| C# | Development team | 06/09/2021-  19/09/2021 | Mandatory |
| Entity Framework | Development team | 06/09/2021-  26/9/2021 | Mandatory |
| .Net Core | Development team | 06/09/2021-  26/9/2021 | Mandatory |
| Javascript - ReactJS – Redux | Development team | 06/09/2021-  26/09/2021 | Mandatory |
| Git, Github | Development team | 13/09/2021 | Mandatory |

## 3. Master Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Deliverable** | **Due Date** | **Deliverable Scope** |
| 1 | |  | | --- | | Project Idea | | 06/09/2021 |  |
| 2 | |  | | --- | | Report 1 – Project  Introduction | | 11/09/2021 | Product Background, Existing Systems, Business Opportunity, Product Vision, Project Scope & Limitations |
| 3 | |  | | --- | | Report 2 – Project  Management Plan | | 12/09/2021 | WBS, Project Process, Plan and Schedule, Project Organization, Project Communication, and Configuration Management |
| 4 | |  | | --- | | Report 3 – SRS | | 20/09/2021 | Business Rules, Use Case Diagram & Use Case Description, Functional Requirements, and Non-Functional Requirements |
| 5 | |  | | --- | | Report 4 –  Software Design | | 15/10/2021 | System Architecture, Back-end Architecture, Front-end and Architecture Database design |
| 6 | |  | | --- | | Code &  Implement  Iteration 1 | | 05/10/2021 | Detailed Design Code & Unit test Integration & System test cases |
| 7 | |  | | --- | | Code &  Implement  Iteration 2 | | 25/10/2021 | Detailed Design Code & Unit test Integration & System test cases |
| 8 | |  | | --- | | Report 5 – Test  Document | | 20/11/2021 | Test Model, Test Plan, Test Cases, and Test Reports |
| 9 | |  | | --- | | Report 6 –  Software Guides | | 28/11/2021 | Deliverable Package Installation Guides User Manual |
| 10 | |  | | --- | | Code &  Implement  Iteration 3 | | 20/11/2021 | Detailed Design Code & Unit test Integration & System test cases |
| 11 | |  | | --- | | Final Code and  Report | | 19/12/2021 | Final Codes & documents, User manual |

## 

## 4. Project Organization

### 4.1 Team & Structures



### 4.2 Roles & Responsibilities

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Project Manager | • Planning, developing schedules, coordinating communication, responsible for keeping the team’s focus on the main goal |
| BA Leader | • Analyze requirement.  • Define scope and create an SRS document.  • Design entity relationship diagram. |
| BA Member | • Define business process flow and object state.  • Capture and specifically describe the use case. |
| Technical Leader | • Define high-level architecture base on SRS.  • Implement configuration and web server.  • Research and implement UI and UX design for front-end. |
| Developer | • Involve in coding the product and reviewing the code of other developers. |
| Test Leader | • Create a template testing document.  • Define test strategy, create a test plan, and defect log template |
| Project Manager | • Planning, developing schedules, coordinating communication, responsible for keeping the team’s focus on the main goal |
| BA Leader | • Analyze requirement.  • Define scope and create an SRS document.  • Design entity relationship diagram. |
| Test Member | • Create test cases.  • Implement test case and log defect |

## 5. Project Communication

### 5.1 Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| |  | | --- | | GitHub | | Professor Kiên,  Hải,  Minh,  Hiếu,  Hưởng,  Dân | |  | | --- | | Resource exchange | | |  | | --- | | Usually | | |  | | --- | | Type | |
| Facebook | Hải,  Minh,  Hiếu,  Hưởng,  Dân | |  | | --- | | Communication | | |  | | --- | | Usually | | |  | | --- | | Method | |
| Discord | Hải,  Minh,  Hiếu,  Hưởng,  Dân | Communication | Usually | Method |
| |  | | --- | | Google Meet | | Professor Kiên,  Hải,  Minh,  Hiếu,  Hưởng,  Dân | Communication | Usually | Method |
| Trello | Hải,  Minh,  Hiếu,  Hưởng,  Dân | |  | | --- | | Define team principle | | |  | | --- | | Usually | | |  | | --- | | Method | |

### 5.2 External Interface

#### a. FU Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Contact Person  (name, position) | Contact address  (email, telephone) | Responsibility |
| Supervisor | Luong Trung Kien | kienlt6@fe.edu.vn | - Provide document template - Give instruction to project team - Review deliverables - Supervise project status |

#### b. Customer Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Contact Person  (name, position) | Contact address  (email, telephone) | Responsibility |
|  |  |  |  |

## 6. Configuration Management

### 6.1 Tools & Infrastructures

|  |  |
| --- | --- |
| **Programming languages** | C#, React, JavaScript |
| **Framework** | Entity framework |
| **API** | Web API |
| **DBMS** | SQL Server |
| **IDEs/Editors** | Visual Studio Code, Visual Studio |
| **UML tools** | Draw.io |
| **Version Control** | Git, Github |
| **Deployment server** |  |
| **Project management tool** | Trello |

### 6.2 Document Management

● Document tools: Microsoft Office 365, 2016.

● File management: Google Drive

### 6.3 Source Code Management

Source code is managed by Git on github.com