

**CAPSTONE PROJECT REPORT**

**Report 2 – Project Management Plan**

– Hanoi, October 2023–

**Table of Contents**

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

[II. Project Management Plan 4](#_heading=h.30j0zll)

[1. Overview 4](#_heading=h.1t3h5sf)

[1.1 Scope & Estimation 4](#_heading=h.hox9ialoy0w5)

[1.2 Project Objectives 5](#_heading=h.fpbb1qexm4d5)

[1.3 Project Risks 6](#_heading=h.4d34og8)

[2. Management Approach 7](#_heading=h.c09gtstnq9dn)

[2.1 Project Process 7](#_heading=h.ws0bbnxvqnj0)

[2.2 Quality Management 8](#_heading=h.2s8eyo1)

[2.3 Training Plan 8](#_heading=h.17dp8vu)

[3. Project Deliverables 1](#_heading=h.df7aroqzmrjx)

[4. Responsibility Assignments 1](#_heading=h.3bkj9l5lbi57)

[4.1 Team & Structure 1](#_heading=h.pq7h8t99r9l3)

[4.2 Role & Responsibility 1](#_heading=h.zgke09fy66zv)

[5. Project Communications 1](#_heading=h.vayfvq8224ze)

[6. Configuration Management 1](#_heading=h.25irc81wo21a)

[6.1 Document Management 1](#_heading=h.111kx3o)

[6.2 Source Code Management 1](#_heading=h.3l18frh)

[6.3 Tools & Infrastructures 1](#_heading=h.44sinio)

# I. Record of Changes

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
| 17/09/2023 | A |  | Create Project Plan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Project Management Plan

## 1. Overview

### 1.1 Scope & Estimation

### 

| **#** | **WBS Item** | **Complexity** | **Est. Effort**  **(man-days)** |
| --- | --- | --- | --- |
| **1** | **Requirement gathering and analysis** |  | **48** |
| ***1.1*** | ***Initiating*** |  | ***17*** |
| 1.1.1 | Prepare project ideas | Simple | 5 |
| 1.1.2 | Actual research | Medium | 3 |
| 1.1.3 | Literature review | Medium | 5 |
| 1.1.4 | Define initial scope | Simple | 3 |
| 1.1.5 | Create report 1 (Introduction) | Simple | 1 |
| ***1.2*** | ***Planning*** |  | ***8*** |
| 1.2.1 | Choose project methodology | Simple | 1 |
| 1.2.2 | Create project schedule | Simple | 2 |
| 1.2.3 | Create risk management plan | Simple | 2 |
| 1.2.4 | Create report 2 (Project management plan) | Simple | 3 |
| ***1.3*** | ***Analysis*** |  | ***23*** |
| 1.3.1 | Create technical prototype | Medium | 5 |
| 1.3.2 | Interviewing user | Simple | 3 |
| 1.3.3 | Update scope & requirements | Medium | 5 |
| 1.3.4 | Create report 3 (SRS) | Medium | 5 |
| 1.3.5 | Create report 4 (SDD) | Medium | 5 |
| **2** | **Project execution** |  | **230** |
| 2.1 | Update SRS & SDS | Simple | 30 |
| 2.2 | Create test cases | Simple | 25 |
| 2.3 | Develop recommendation algorithm | Complex | 100 |
| 2.4 | Develop web application, console-based application | Complex | 30 |
| 2.5 | Execute unit tests, integration tests | Simple | 15 |
| 2.6 | Create Report 5 (Test Report) | Medium | 10 |
| 2.7 | Execute system test | Medium | 5 |
| 2.8 | Execute acceptance test | Medium | 5 |
| 2.9 | Create report 6 (User guides) | Medium | 3 |
| 2.10 | Create report 7 (Final report) | Medium | 3 |
| 2.11 | Prepare thesis presentation | Medium | 3 |
| 2.12 | Final thesis presentation | Simple | 1 |

| ***Total Estimated Effort (man-days)*** | | | ***246*** |
| --- | --- | --- | --- |

### 1.2 Project Objectives

***Objectives:***

* This project must be finished by 01/12/2023.
* All team members need to follow the task assigned by the PM.
* All team members give their best effort to complete the project.
* Project team members learn new knowledge, and new technology to apply to the project.

***Target Metrics:***

* Timelines: 100%
* Allocated Effort: 246 mans-days
* Defect Distribution:

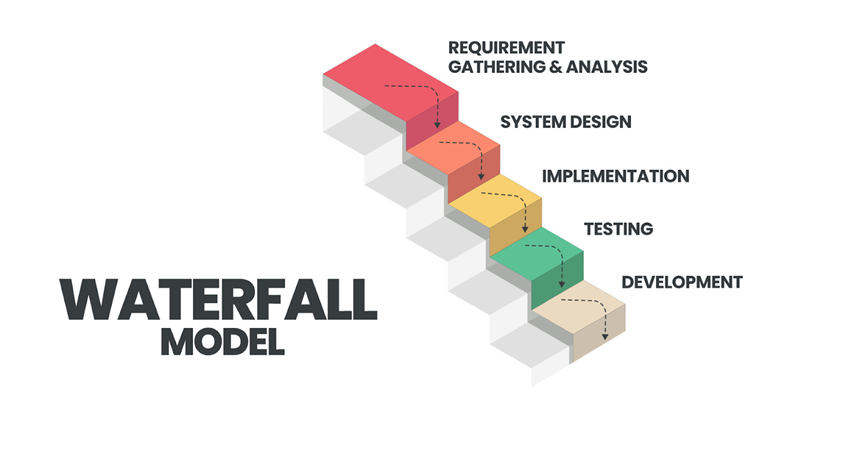
| **#** | **Testing Stage** | **Test Coverage** | **No. of Defects** | **% of Defect** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| 1 | Unit Test | 100% | 44 | 47.8% | Written and implemented by software developers to ensure that a part of an application (called a "unit") follows its design and functions as expected |
| 2 | Integration Test | 100% | 28 | 30.4% | Individual software modules are combined and tested as a group |
| 3 | System Test | 100% | 20 | 21.8% | Taken across the entire system, testing not only the design but also the behavior and even the raw expectations of the customer |
| 4 | Acceptance Test | 100% | 0 | 0% | System users perform tests in accordance with what will happen in real-life scenarios |
|  | **Total** |  | **92** | **100%** |  |

### 1.3 Project Risks

| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| --- | --- | --- | --- | --- |
| 1 | Insufficient resources (human) | High - Could lead to delays or failure | High - Unavoidable | -Ensure that team members work efficiently and maximize their skills. |
| 2 | Requirement changes | Medium - Could affect project scope | Medium - Likely | - Clearly define project objectives and requirements. Establish a change request process and communicate changes effectively. |
| 3 | Team conflicts | Medium - Could lead to productivity loss | Medium - Possible | - Promote effective communication and conflict resolution within the team. Establish conflict resolution procedures. |
| 4 | Unclear project objectives | Medium - May result in misalignment | Medium - Possible | - Clearly define and communicate project objectives. Ensure team members have a shared understanding of the goals. |
| 5 | Changes in legal and healthcare regulations | High | Medium | - Continuously monitor legal and healthcare regulation changes.  - Collaborate with legal experts to ensure compliance.  - Update the system according to new changes. |

## 2. Management Approach

### 2.1 Project Process



**The *HCS*** project uses the Waterfall Model.

The waterfall model is a linear software development method in which the development phases are executed sequentially and only move to the next stage after the previous phase is completely completed. This model is suitable for projects that require stability and high predictability.

Specifically, the steps to implement the project's Waterfall model are as follows:

* ***Design***: Based on requirements, design appropriate IT services. Including technology infrastructure design and service management process.
* ***Implementation***: Deploy technology infrastructure and build technologies to support IT services according to predefined requirements and designs.
* ***Testing***: Test system functions for efficient and reliable operation, ensuring that features are integrated without errors.
* ***Deployment***: Deploy the HCS system into the real environment. Including training and service manual

Particularly, the final phase, Maintenance, will not be implemented in the context of a graduation project.

### 2.2 Quality Management

To improve Project quality, we plan to use several approaches as follows:

* **Define Quality Requirements for Requirements Documents:** In this phase, identify quality requirements for the requirements documentation. Ensure that the requirements documentation is clear, complete, and unambiguous.
* **Check the Quality of Design**: Ensure that the system's design is quality-checked to meet the defined quality requirements.
* **Establish Design Rules:** Set up design rules and standards for the project to ensure correctness and quality of the design.
* **Quality Check of Source Code:** Conduct source code quality checks to ensure correctness, efficiency, and compliance with coding standard
* **Unit Testing**: Members perform unit tests for their own modules to ensure that basic errors do not occur
* **Integration Testing**: Conduct integrated testing of the combination of components and modules in the system to ensure that they can be compatible with each other without causing errors, and overcome problems of asynchrony between modules in the system.
* **System Testing**: Perform comprehensive testing on HCS to ensure operation, performance, and meet set quality requirements.

### 2.3 Training Plan

| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| --- | --- | --- | --- |
| Source Control Training on Git, Github | All team members | 18/09/2023 (1 day) | Mandatory |
| Coding Convention & Bug Logging Convention | All team members | 30/09/2023 (1 day) | Mandatory |
| React js(FrontEnd) | All team members | 01/10/2023- 20/10/2023(10 days) | Mandatory |
| C#/.Net 6 (BackEnd) | All team members | 21/10/2023 - 28/10/2023(7 days) | Mandatory |

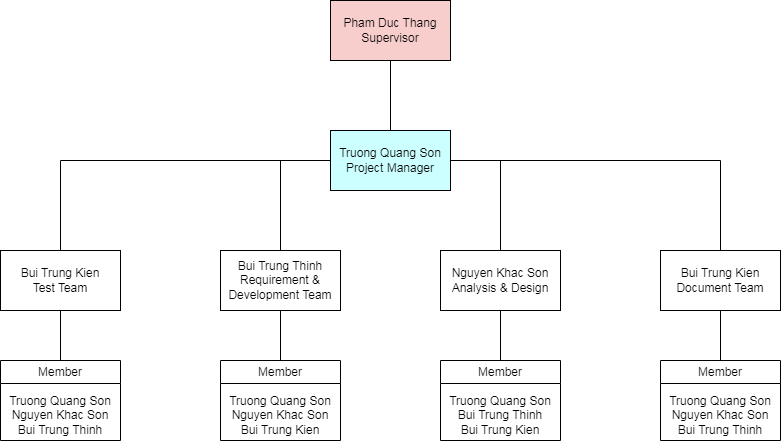
## 3. Project Deliverables

###### 

| **#** | **Deliverable** | **Start Date** | **Due Date** | **Duration** | **Deliverable scope** | **Note** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Project introduction Report 1 | 18/09/2023 | 20/09/2023 | 6 days | Overview project, Feature list | Week 2 |
| 2 | Project plan  Report 2 | 21/09/2023 | 25/09/2023 | 5 days | Time management, Risk management, Resource management plan | Week 3 |
| 3 | System Design Report 3  Report 4 | 26/09/2023 | 05/10/2023 | 15 days | Screen Design Architectural Design Database Design | Week 4 and week 5 |
| 4 | Code package | 06/10/2023 | 10/11/2023 | 36 days | Code & unit TCs, integration TCs | Week 6 to Week 10 |
| 5 | Testing documents Report 5 | 11/11/2023 | 20/11/2023 | 10 days | Test plan, System TCs, integration TCs, acceptance TCs | Week 11 and week 12 |
| 6 | User Guides Documents  Report 6 | 21/11/2023 | 25/11/2023 | 5 days | User manuals | Week 13 |
| 7 | Final report | 26/11/2023 | 30/11/2023 | 5 days | All previous reports | Week 14 |
| 8 | Final code package | 01/12/2023 | 01/12/2023 | 1 days | Final code & documents | Week 14 |

## 4. Responsibility Assignments

### 4.1 Team & Structure



### 4.2 Role & Responsibility

| Role | Name | Responsibilities |
| --- | --- | --- |
| ***Supervisor*** | Pham Duc Thang | * Give instruction to the project team * Verify deliverables * Supervise project team’s status |
| ***Project Manager*** | Trương Quang Sơn | * Tracking member’s work * Report working status to the instructor * Review, discuss each stage completed on time * Assign task to each member * Have overall responsibility of the project |
|  |  | **Documentation Report Team** |
| ***Leader*** | Bùi Trung Kiên | * Assign task to members * Responsible for SRS (Software Requirement Specification) * Review all of the documents * Support in other documents |
| ***Members*** | Trương Quang Sơn  Bùi Trung Thịnh  Nguyễn Khắc Sơn | * Support, complete in other documents |
|  |  | **Analysis & Design Team** |
| ***Leader*** | Nguyễn Khắc Sơn | * Analysis Requirements * Create SRS and define scopes * Define high level architecture based on SRS * Use case specification * System Architecture Design * Screen Design * Sequence Diagram * Workflow |
| ***Members*** | Trương Quang Sơn  Bùi Trung Thịnh  Bùi Trung Kiên | * Complete Screen Design * Screen Layout * Class Diagram * Flow Chart * Context Diagram * ERD Diagram * Database Design * Define business process flow and object state |
|  |  | **Requirement & Development Team** |
| ***Leader*** | Bùi Trung Thịnh | * Decide technique and tools to be used * Ensure assignment or directly assigns specific tasks to Code Team members * Ensure that the team has sufficient performance levels and is also responsible for conflict prevention and resolution. * Provide feedback on quality plan, help determine metrics and criteria for this project * Control source code and merge code * Provide feedback on deliverables and quality reviews * Keeping track of development work |
| ***Members*** | Trương Quang Sơn  Nguyễn Khắc Sơn  Bùi Trung Kiên | * Coding back-end * Coding front-end * Provide the best possible application reception by the user * Help decide what technique and tools to be used * Deploy web application to host server in each release version |
|  |  | **Testing Team** |
| ***Leader*** | Bùi Trung Kiên | * Create test plan and define test strategy * Assign tasks to members * Create test case and testing documents * Controlling testing activities * Implement test case and report test result * Collect and analyse customer information |
| ***Member*** | Trương Quang Sơn  Nguyễn Khắc Sơn  Bùi Trung Thịnh | * Create test case in Unit test * Create test case in Integration test * Create test case in System test * Create test case in Acceptance test * Implement test case, fix bugs and log defect |

## 5. Project Communications

| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| --- | --- | --- | --- | --- |
| Kickoff meeting | Project team + Supervisor | Introduce the project, confirm the project objectives, goals, and deliverables | A one-off event, at the beginning of the project | Online meetings in Google Meet |
| Meeting with Supervisor | Project team + Supervisor | Review project status and discuss potential issues. | Weekly  (Every Wednesday) | Meet face-to-face in Alpha Building in FPT University |
| Weekly meeting schedule | Project team | Discuss what each team member did, and what they’ll do | Weekly  (Every Thursday, Friday) | Meetings in Google Meet |
| Daily meeting schedule | Each small team (Design, FrontEnd, BackEnd, Test,...) | Each team will have an online meeting to inform the others: “What did I do yesterday?”, “What will I do today?” | Everyday | Online meetings in Google Meet |

## 6. Configuration Management

### 6.1 Document Management

* Document tools: Microsoft Office 365, Google Docs/Sheets/Slides
* File management: Google Drive

### 6.2 Source Code Management

We use Git as a tool for Code management and Version control. The source code is hosted on GitHub.

#### 6.2.1 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.
* All developers must update the newest code before committing code to Github.

#### 6.2.2 Management

* Source code managed by Github.
* Technique leader is the person who will review and merge the code of team members.
* Only team members can access the repository on Github.

### 6.3 Tools & Infrastructures

| **Category** | **Tools / Infrastructure** |
| --- | --- |
| **Technology** | React js (FrontEnd), C#/.Net 7 (BackEnd) |
| **Database** | SQL SERVER |
| **IDEs/Editors** | Visual Studio, VS Code, SSMS |
| **Diagramming** | Visio, DrawIO, Visual Paradigm |
| **Documentation** | Ms Office, Google Docs/Sheets/Slides |
| **Version Control** | GitHub (Source Codes), Google Drive (Documents) |