

**Capstone Project Report**

**Report 2 – Project Management Plan**

– Hanoi, September 2022 –

**Table of Contents**

[I. Record of Changes 3](#_Toc83330363)

[II. Project Management Plan 4](#_Toc83330364)

[1. Overview 4](#_Toc83330365)

[1.1 Scope & Estimation 4](#_Toc83330366)

[1.2 Project Objectives 4](#_Toc83330367)

[1.3 Project Risks 4](#_Toc83330368)

[2. Management Approach 5](#_Toc83330369)

[2.1 Project Process 5](#_Toc83330370)

[2.2 Quality Management 5](#_Toc83330371)

[2.3 Training Plan 5](#_Toc83330372)

[3. Project Deliverables 5](#_Toc83330373)

[4. Responsibility Assignments 5](#_Toc83330374)

[5. Project Communications 6](#_Toc83330375)

[6. Configuration Management 6](#_Toc83330376)

[6.1 Document Management 6](#_Toc83330377)

[6.2 Source Code Management 6](#_Toc83330378)

[6.3 Tools & Infrastructures 6](#_Toc83330379)

# I. Record of Changes

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Project Management Plan

## 1. Overview

### 1.1 Scope & Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **WBS Item** | **Complexity** | **Est. Effort**  **(man-days)** |
| **1** | **Pre-Initiation Phase** |  | ***5*** |
| 1.1 | Come up with ideas | Complex | 2.5 |
| 1.2 | Determine the role of team members | Simple | 1 |
| 1.3 | Choose topic | Medium | 1 |
| 1.4 | Register Capstone Project | Simple | 0.5 |
| **2** | **Initiation Phase** |  | ***10*** |
| 2.1 | Kick-offs meeting | Simple | 1 |
| 2.2 | Identify stakeholders | Medium | 4 |
| 2.3 | Meet Instructor | Simple | 1 |
| 2.4 | Project introduction | Medium | 4 |
| **3** | **Planning Phase** |  | ***23*** |
| 3.1 | Define project scope | Simple | 3 |
| 3.2 | Choose working model process | Simple | 2 |
| 3.3 | Select tools and techniques | Simple | 1 |
| 3.4 | Project schedule planning | Complex | 7 |
| 3.5 | Risk management plan | Medium | 5 |
| 3.6 | Project organization | Simple | 3 |
| 3.7 | Software project management planning | Medium | 3 |
| **4** | ***Execution Phase*** |  | ***225*** |
| 4.1 | **Requirements Analysis** |  | ***20*** |
| 4.1.1 | User Requirements | Medium | *8* |
| 4.1.2 | Functional Requirements | Medium | *8* |
| 4.1.3 | Non-Functional Requirements | Simple | *4* |
| 4.2 | **Design** |  | ***25*** |
| 4.2.1 | Basic architecture design | Medium | *5* |
| 4.2.2 | Detail design | Medium | *10* |
| 4.2.3 | Data & Database Diagram design | Complex | *10* |
| 4.3 | **Coding** |  | ***160*** |
| 4.3.1 | Back-end | Complex | *65* |
| 4.3.2 | Front-end | Complex | *65* |
| 4.3.3 | Fix bugs | Medium | *30* |
| 4.4 | **Testing** |  | ***40*** |
| 4.4.1 | Unit test | Medium | *14* |
| 4.4.2 | Integration test | Medium | *14* |
| 4.4.3 | System test | Medium | *8* |
| 4.4.4 | Acceptance test | Medium | *4* |
| **5** | **Monitoring and Controlling Phase** |  | ***23*** |
| 5.1 | Meet instructor | Simple | 3 |
| 5.2 | Control risk scope | Medium | 5 |
| 5.3 | Control cost | Medium | 5 |
| 5.4 | Process report | Medium | 10 |
| **6** | **Closing Phase** |  | ***25*** |
| 6.1 | Complete final report | Complex | 10 |
| 6.2 | Prepare for final project presentation | Medium | 9 |
| 6.3 | Present final report | Complex | 5 |
| 6.4 | Project closed | Simple | 1 |
| **Total Estimated Effort (man-days)** | | | ***311*** |

### 1.2 Project Objectives

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Testing Stage** | **Test Coverage** | **No. of Defects** | **% of Defect** | **Notes** |
| 1 | Unit Test | 100% | 50 | 95% | Reviewing and tests |
| 2 | Integration Test | 100% | 500 | 95% | Tester creates and tests |
| 3 | System Test | 100% | 200 | 95% | Tester creates and tests |
| 4 | Acceptance Test | 100% | 20 | 95% | User verifies system |

Project must be finished before 15/12/2022 with three main deliverables:

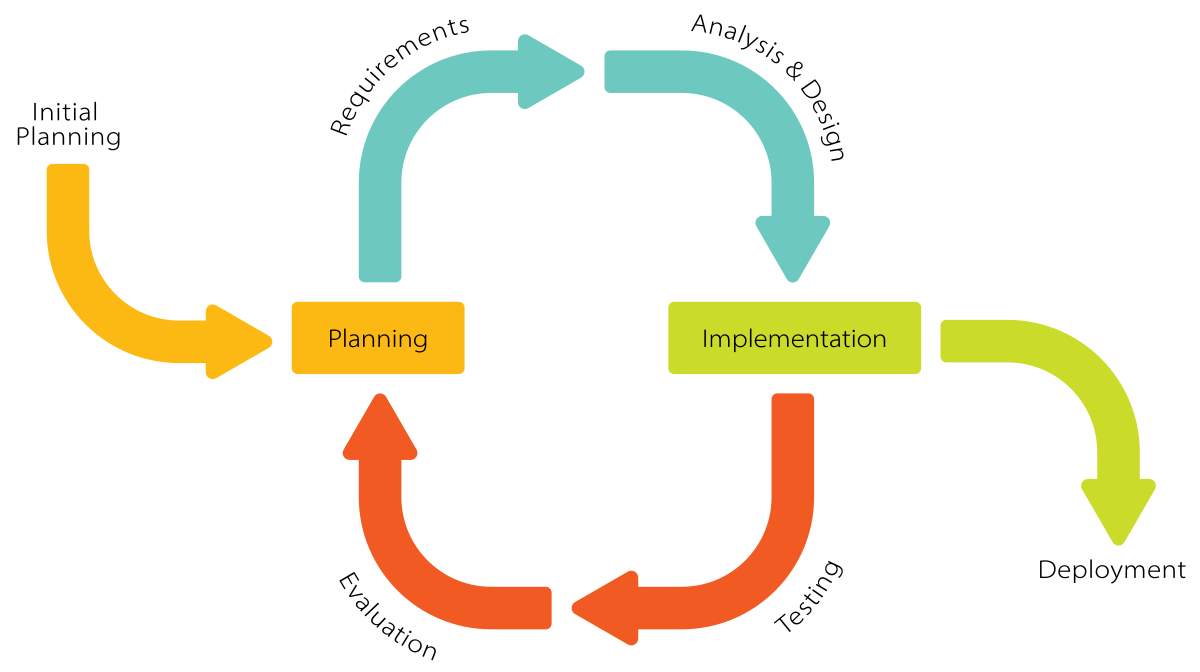
* Web App (Cover 100% requirement)
* Project documentation (FPT University capstone project)
* Project presentation slides

### 1.3 Project Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | Team members lack of knowledge, skill, experience about project | High | High | Members need to learn about technologies that require for his/her part of the project.  Who has skills will train for other members. |
| 2 | Absence of team members, especially by mental and physical illness | High | Low | Members must notify the team about the absence period, personal schedule, and the plan of how to keep up with the work process.  Ensure that the absence of a member won’t affect others and always have plans to deal with this problem. The PM reminds team members about task deadlines. |
| 3 | Conflict between team members | Medium | Medium | Define, divide the tasks clearly, estimate the time and agree before starting the task. All members meet to solve the problem.  If no agreement is reached, team will decide by voting. |
| 4 | Broken devices or internet during working on project | Medium | Low | Carefully prepare personal device.  If the network is disconnected suddenly then use 4G. |

## 2. Management Approach

### 2.1 Project Process



After researching the software development model carefully, the project will use the Iterative and Incremental Software Process Model. In an Iterative & Incremental model, initially, a partial implementation of a total system is constructed so that it will be in a deliverable state. The Iterative & Incremental model is mostly used when the scope of the project is big, the major requirements are defined clearly, some more details will be added later in software development. By using this software process model, we break down the developing system task into a series of smaller tasks which will be completed separately, allowing us to take advantage of what was learned during the development of earlier parts of the system. The reasons for the project to choose this model are:

* You can develop some main functions that prioritize requirements first.
* Requirements changes can be easily accommodated.
* Testing and debugging during smaller iterations is easy.
* Easier to manage risk easier because risk segments are identified and dealt with during its  iteration.
* Customers can provide feedback to each product increment, thus avoiding surprises at the end of development.
* Customers get important functionality early

### 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 |
| Java Spring Boot | Backend Members | 15/09/2022 to 29/09/2022 | Mandatory |
| React JS | Frontend Members | 15/09/2022 to 29/09/2022 | Mandatory |
| Gitlab | All Team members | 15/09/2022 to 21/09/2022 | Mandatory |

## 3. Project Deliverables

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Deliverable** | **Due Date** | **Notes** |
| 1 | Project Introduction | 15/09/2022 | Prepare project ideals/ Intro |
| 2 | Project Management Plan | 17/09/2022 | Prepare Project Plan, Create Technical Prototype |
| 3 | Software Requirement Specification (SRS) | 25/09/2022 | Write SRS Document and Create Test Plan |
| 4 | Software Design Specification (SDS) | 30/09/2022 | Architecture Design, Detailed Design, Database |
| 5 | Code and Implement Iteration 1 | 13/10/2022 | Code & Unit test, Integration test and System test |
| 6 | Code and Implement Iteration 2 | 27/10/2022 | Code & Unit test, Integration test and System test |
| 7 | Code and Implement Iteration 3 | 10/11/2022 | Code & Unit test, Integration test and System test |
| 8 | Test Documentation | 24/11/2022 | Write test documents and synthesize test cases |
| 9 | Software User Guildes | 30/11/2022 | Write user manuals document |
| 10 | Final code and report | 08/12/2022 | Final Code, System test reports |

## 4. Responsibility Assignments

*D~Do; R~Review; S~Support; I~Informed; <blank>- Omitted*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Responsibility** | **ChuongNNHE14028** | **DungNTHE140858** | **TrinhPKHE140643** | **LongTNHE140955** | **HaHMHE140295** |
| Project Planning | S | S | R | R | D |
| Project Tracking | I | I | I | S | D |
| Project Introduction Document | S | D | R | R | D |
| Prepare SRS Document | S | I | I | S | D |
| Prepare SDD Document | S | I | I | S | D |
| Code Backend | I | I | D | D | I |
| Code Fronted App | D | D | I | S | I |
| Test Document and Report | S | S | S | D | R |

## 5. Project Communications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| Weekly Meeting with Supervisor | Team members and supervisor | Report project progress.  Seek advice.  Submit documentation | Thursday | Offline |
| Weekly Meeting | Team members | Reports Review, tracking progress | Sunday, Tuesday | Google Meet |
| Daily Meeting | Team members | Task Review | Daily | Google Meet |
| Additional Meeting | Team members | Solving Issues | Unspecified | Google Meet |

## 6. Configuration Management

### 6.1 Document Management

Our project uses Confluence and Google Drive to manage project documents.

**- Confluence:**

**Logo

Description automatically generated**

Confluence is a web-based corporate [wiki](https://en.wikipedia.org/wiki/Wiki) ([collaboration software](https://en.wikipedia.org/wiki/Collaboration_software)) developed by Australian software company [Atlassian](https://en.wikipedia.org/wiki/Atlassian). Atlassian wrote Confluence in the [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) programming language and first published it in 2004. Confluence Standalone comes with a built-in Tomcat web server and [hsql](https://en.wikipedia.org/wiki/Hsql) database, and also supports other databases.

### Break down team silos

An open, connected structure allows information to flow freely among everyone at the organization.

### Turn conversations into action

Built for lasting knowledge so you never lose great ideas or context in a transient notification or chat.

### Organize everything in one place

From quarterly planning docs to new hire blogs, everything lives on Confluence.

### Build a culture of open teamwork

With social features, employees at every level have a voice to contribute, share, and receive feedback.

Graphical user interface, text, application, email

Description automatically generated

*Figure 6.1.1: Confluence Overview*

This is an example about APIs management in Confluence

A screenshot of a computer

Description automatically generated

*Figure 6.1.2: APIs management in Confluence*

**- Google Drive:**

**Logo

Description automatically generated with low confidence**

Google Drive is a [file storage](https://en.wikipedia.org/wiki/File_hosting_service) and [synchronization service](https://en.wikipedia.org/wiki/File_synchronization) developed by [Google](https://en.wikipedia.org/wiki/Google). Launched on April 24, 2012, Google Drive allows users to store files in the [cloud](https://en.wikipedia.org/wiki/Cloud_storage) (on Google's servers), synchronize files across devices, and [share files](https://en.wikipedia.org/wiki/File_sharing). In addition to a [web interface](https://en.wikipedia.org/wiki/Web_application), Google Drive offers apps with offline capabilities for [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and [macOS](https://en.wikipedia.org/wiki/MacOS) computers, and [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) and [iOS](https://en.wikipedia.org/wiki/IOS) smartphones and tablets. Google Drive encompasses [Google Docs](https://en.wikipedia.org/wiki/Google_Docs), [Google Sheets](https://en.wikipedia.org/wiki/Google_Sheets), and [Google Slides](https://en.wikipedia.org/wiki/Google_Slides), which are a part of the [Google Docs Editors](https://en.wikipedia.org/wiki/Google_Docs_Editors) [office suite](https://en.wikipedia.org/wiki/Office_suite) that permits collaborative editing of documents, spreadsheets, presentations, drawings, forms, and more. Files created and edited through the Google Docs suite are saved in Google Drive.

*Graphical user interface, application

Description automatically generated*

*Figure 6.1.3: Google Drive Overview*

### 6.2 Source Code Management

Our project uses GitLab to manage source code.

**- Gitlab**

**Logo

Description automatically generated**

GitLab helps you automate the builds, integration, and verification of your code. With SAST, DAST, code quality analysis, plus pipelines that enable concurrent testing and parallel execution, your teams get quick insights about every commit so they can deliver higher quality code faster.

*A screenshot of a computer

Description automatically generated*

*Figure 6.2.1: Project management in GitLab*

This is a code management in GitLab.

*Graphical user interface, text, application, email

Description automatically generated*

*Figure 6.2.2: Code management in GitLab*

### 6.3 Tools & Infrastructures

|  |  |
| --- | --- |
| **Category** | **Tools / Infrastructure** |
| **Technology** | ReactJS (FrontEnd), Java/SpringBoot (BackEnd), Kafka |
| **Database** | MySQL |
| **IDEs/Editors** | Visual Studio Code, IntelliJ, WebStorm |
| **Diagramming** | Figma, DrawIO |
| **Documentation** | Ms Office, Google Docs/Sheets/Slides |
| **Version Control** | GitLab (Source Codes), Confluence (Documents) |
| **Deployment server** | Amazon Web Services |
| **Third part** | Google Map Platform |
| **Project management** | Jira (Schedule, Tasks, Defects) |
| **Communication** | Messenger, Google Meet |