

**CAPSTONE PROJECT REPORT**

**Report 2 – Project Management Plan**

– Hanoi, August 2019 –

**Table of Contents**

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

[II. Project Management Plan 4](#_30j0zll)

[1. Overview 4](#_1fob9te)

[1.1 Scope & Estimation 4](#_3znysh7)

[1.2 Project Objectives 4](#_2et92p0)

[1.3 Project Risks 4](#_tyjcwt)

[2. Management Approach 5](#_3dy6vkm)

[2.1 Project Process 5](#_1t3h5sf)

[2.2 Quality Management 5](#_4d34og8)

[2.3 Training Plan 5](#_2s8eyo1)

[3. Project Deliverables 5](#_17dp8vu)

[4. Responsibility Assignments 5](#_3rdcrjn)

[5. Project Communications 6](#_26in1rg)

[6. Configuration Management 6](#_lnxbz9)

[6.1 Document Management 6](#_35nkun2)

[6.2 Source Code Management 6](#_1ksv4uv)

[6.3 Tools & Infrastructures 6](#_44sinio)

# 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

*[Create/Provide the list of software product following the table template as below. In this table, we categorize each software function into three levels of complexity (Simple, Medium, Complex) and estimate the total effort to complete each one in man-day]*

| **#** | **WBS Item** | **Complexity** | **Est. Effort**  **(man-days)** |
| --- | --- | --- | --- |
| ***1*** | ***Feature 1*** |  | ***21*** |
| 1.1 | Function 1.1 | Simple | 4 |
| 1.2 | Function 1.2 | Medium | 7 |
| 1.3 | Function 1.N | Complex | 10 |
| ***2*** | ***Feature 2*** |  |  |
| ***2.1*** | ***…*** |  |  |
| ***Total Estimated Effort (man-days)*** | | | ***21*** |

### 1.2 Project Objectives

*[Provide the overall project objective description and then the specific target metrics of your project in term of quality, time, and cost (allocated effort distribution for project activities: requirement, design, coding, testing, project management, etc). For example*

*Quality*

| **#** | **Testing Stage** | **Test Coverage** | **No. of Defects** | **% of Defect** | **Notes** |
| --- | --- | --- | --- | --- | --- |
| 1 | Reviewing | 100% | 0 | 0% |  |
| 2 | Unit Test | 95% | 2 | 20% |  |
| 3 | Integration Test | 90% | 3 | 30% |  |
| 4 | System Test | 85% | 4 | 40% |  |
| 5 | Acceptance Test | 90% | 1 | 10% |  |

*Milestone Timeliness (%):*

*Allocated Effort (man-days):*

]

### 1.3 Project Risks

*[List out the details on project risks in the table below]*

| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| --- | --- | --- | --- | --- |
| 1 | Lack of knowledge about Spring Boot and React | High | Medium | Organize internal training sessions |
| 2 | Missed deadline due to large workload | High | High | Divide work and manage time strictly |
| 3 | Schedule conflicts between members | Medium | Medium | Arrange flexible work schedules |

## 2. Management Approach

*[Describe the approach you would use the manage and implement your project]*

### 2.1 Project Process

- We will apply the Agile Scrum model to this project. Key steps include:

* Sprint Planning: Plan for each sprint.
* Daily Stand-up: Brief daily meeting to update progress.
* Sprint Review: Evaluate completed work after each sprint.
* Sprint Retrospective: Learn from experience and improve the process.

### 2.2 Quality Management

- We will apply the following methods to ensure project quality:

* Defect Prevention: Proactively prevent errors by reviewing code and writing complete test cases.
* Defect Detection: Use automated testing tools and perform manual testing.
* Defect Correction: Fix errors immediately after detection and check again to ensure no errors reappear.

### 2.3 Training Plan

*[You need to plan the training activities in case any of your team member lack of knowledge/skills to handle the project works]*

| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| --- | --- | --- | --- |
| Java Spring Boot | 5 |  | Mandatory |
| Git, Github | 5 |  | Mandatory |

## 3. Project Deliverables

*[Given the main project deliverables. Those can be internal and/or external deliverables. Students can prepare master schedule like the table format as below or in the more detailed structure as the sample in the attached sample file -* ***Report2\_Sample High Level Project Schedule.pdf****]*

| **#** | **Deliverable** | **Due Date** | **Notes** |
| --- | --- | --- | --- |
| 1 | … | dd/MM/yyyy | … |

## 4. Responsibility Assignments

*[Describe the main responsibilities in your project (to complete the outputs as defined in the above section), in the format as the sample below]*

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

| **Responsibility** | **QuangNVHE16** | **CongVTHE160103** | **HieuPBHE163832** | **QuangNHHE16** | **AnhLQHE16** |
| --- | --- | --- | --- | --- | --- |
| Project Planning & Tracking | R/I | D | D | S | S |
| Prepare Project Introduction Document | S | S | D | R | I |
| Prepare SRS Document (Overview Part) | R | S | S | D | D |
| Prepare SRS Document (User Requirements) | R | S | S | D | D |
| Prepare SDS Document (Secig |  |  |  |  |  |
|  |  |  |  |  |  |

## 5. Project Communications

| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| --- | --- | --- | --- | --- |
| Kick-off Meeting | Project Team, Instructors | Introduce project objectives, scope, and team roles | Start of the project | In-person meeting, Zoom |
| Weekly Status Report | Project Team, Instructors | Update on project progress, identify issues | Every Monday | Email, Google Docs |
| Bi-weekly Team Meeting | Project Team | Review progress, discuss challenges, plan next steps | Every two weeks on Monday | In-person meeting, Zoom |
| Monthly Progress Report | Project Team, Instructors | Comprehensive update on project status | End of each month | Email, PDF report |
| Issue Log Update | Project Manager, Project Team | Track and manage project issues | As needed | Google Sheets, Trello |
| Final Presentation | Project Team, Instructors | Present final results and findings | End of the project | In-person presentation, PowerPoint |
| Feedback Session | Instructors, Project Team | Gather feedback on project outcomes | After final presentation | In-person/Online meeting, SurveyMonkey |

## 6. Configuration Management

### 6.1 Document Management

- To manage project documents and their changes/versions, I use Google Docs and Google Sheets. This approach ensures that all project documentation, including the Software Requirements Specification (SRS), Software Design Specification (SDS), and planning documents, are centralized and accessible to the entire team.

**6.1.1. Document Creation and Sharing:**

- Create initial versions of all necessary documents (e.g., SRS, SDS, planning) using Google Docs and Google Sheets.

- Share these documents with relevant team members, assigning appropriate permissions (e.g., view, comment, edit).

**6.1.2. Version Control:**

- Utilize Google Docs' and Sheets' built-in version history to track changes. This allows for easy comparison of document versions and rollback to previous versions if necessary.

- Name and timestamp significant versions for quick reference (e.g., "SRS\_v1.0\_2024-05-17").

**6.1.3. Change Management:**

- Implement a change request process where proposed modifications are reviewed and approved by relevant stakeholders before being incorporated into the documents.

- Use comments and suggestions in Google Docs to discuss and review changes collaboratively.

**6.1.4. Documentation Updates:**

- Regularly update documents to reflect approved changes. Ensure that all updates are logged in the version history with detailed descriptions of the changes made.

- Notify the team of significant updates and provide summaries of the changes to ensure everyone is aware of the latest information.

**6.1.5. Backup and Security:**

- Periodically download and store backup copies of critical documents to a secure location.

- Ensure that access to documents is controlled and that sensitive information is protected according to the project's security policies.

- By following these practices, project documents are effectively managed, ensuring that all changes are tracked, reviewed, and communicated efficiently.

### 6.2 Source Code Management

- Our project's source code will be managed using GitLab, providing a robust platform for collaboration and version control. Here's an overview of our approach:

**6.2.1. Repository Setup:** Create a Git repository on GitLab to serve as the central location for storing our source code.

**6.2.2. Version Control:** Use Git commands to track changes to the codebase, ensuring a detailed history of modifications.

**6.2.3. Branching Strategy:** Utilize branches to isolate work on new features or fixes, facilitating parallel development efforts.

**6.2.4. Collaboration:** Invite team members to collaborate on the project by granting them access to the GitLab repository. This enables seamless contribution and code sharing.

**6.2.5. Code Reviews:** Leverage GitLab's built-in code review capabilities to maintain code quality and facilitate feedback among team members.

**6.2.6. CI/CD Integration:** Implement CI/CD pipelines in GitLab to automate testing and deployment processes, streamlining the development lifecycle.

**6.2.7. Issue Tracking:** Utilize GitLab's issue tracking system to manage and prioritize tasks, ensuring efficient resolution of bugs and implementation of features.

**6.2.8. Documentation:** Maintain project documentation within the GitLab repository, covering essential information such as installation instructions, configuration details, and coding standards.

- By adhering to these practices, we can effectively manage our project's source code and ensure its integrity throughout the development lifecycle.

### 6.3 Tools & Infrastructures

| **Category** | **Tools / Infrastructure** |
| --- | --- |
| **Technology** | HTML/CSS ,ReactJS (FrontEnd) , Java/SpringBoot (BackEnd) |
| **Database** | MySQL |
| **IDEs/Editors** | Visual Studio Code, IntelliJ IDEA |
| **Diagramming** | DrawIO, Visual paradigm |
| **Documentation** | Google Docs/Sheets/Slides |
| **Version Control** | GitLab (Source Codes), Google Drive (Documents) |
| **Deployment server** |  |
| **Project management** | GitLab (Tasks, Defects) |