

**CAPSTONE PROJECT REPORT**

**Report 2 – Project Management Plan**

– Ho Chi Minh, January 2022 –

**Table of Contents**

[II. Project Management Plan 4](#_Toc101828489)

[1. Overview 4](#_Toc101828490)

[1.1 WBS & Estimation 4](#_Toc101828491)

[1.2 Project Objectives 7](#_Toc101828492)

[1.3 Project Risks 7](#_Toc101828493)

[2. Management Approach 7](#_Toc101828494)

[2.1 Project Process 7](#_Toc101828495)

[2.2 Quality Management 8](#_Toc101828496)

[2.3 Training Plan 8](#_Toc101828497)

[3. Master Schedule 8](#_Toc101828498)

[4. Project Organization 10](#_Toc101828499)

[4.1 Team & Structures 10](#_Toc101828500)

[4.2 Roles & Responsibilities 10](#_Toc101828501)

[5. Project Communication 11](#_Toc101828502)

[5.1 Communication Plan 11](#_Toc101828503)

[5.2 External Interface 11](#_Toc101828504)

[a. FU Contacts 11](#_Toc101828505)

[b. Customer Contacts 11](#_Toc101828506)

[6. Configuration Management 11](#_Toc101828507)

[6.1 Tools & Infrastructures 11](#_Toc101828508)

[6.2 Document Management 12](#_Toc101828509)

[6.3 Source Code Management 12](#_Toc101828510)

**Table of Figures**

[Figure 1. Scrum Framework 8](#_Toc101828511)

[Figure 2. Team & Structures 10](#_Toc101828512)

**List of Tables**

[Table 1. WBS & Estimation 4](#_Toc101828513)

[Table 2. Project Objectives 7](#_Toc101828514)

[Table 3. Project Risks 7](#_Toc101828515)

[Table 4. Training Plan 8](#_Toc101828516)

[Table 5. Master Schedule 8](#_Toc101828517)

[Table 6. Roles & Responsibilities 10](#_Toc101828518)

[Table 7. Communication Plan 11](#_Toc101828519)

[Table 8. FU Contacts 11](#_Toc101828520)

[Table 9. Customer Contacts 11](#_Toc101828521)

[Table 10. Tools & Infrastructures 11](#_Toc101828522)

# II. Project Management Plan

## 1. Overview

### 1.1 WBS & Estimation

Table 1. WBS & Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **WBS Item** | **Complexity** | **Est. Effort** |
| **1** | **Analysis** |  | ***53*** |
| 1.1 | Finalize project ideas | Medium | 3 |
| 1.2 | Deliver Report 1 – Project Introduction | Simple | 10 |
| 1.2.1 | Research business opportunity and existing systems | Simple | 5 |
| 1.2.2 | Determine the major features and limitations | Simple | 5 |
| 1.3 | Deliver Report 2 – Project Management Plan | Simple | 10 |
| 1.3.1 | Breakdown work structures & choose the appropriate management approach method | Simple | 6 |
| 1.3.2 | Determine the project organization structure & project communication | Simple | 4 |
| 1.4 | Deliver Report 3 – System Requirement Specification | Medium | 30 |
| 1.4.1 | Gathering requirements & determine the business rules | Simple | 5 |
| 1.4.2 | Design use case diagram & use case specification | Medium | 22 |
| 1.4.3 | Design entity relationship diagram | Simple | 3 |
| **2** | **Design** |  | ***40*** |
| 2.1 | Deliver Report 4 – Software Design Document | Medium | 30 |
| 2.1.1 | Determine the appropriate system architecture | Medium | 2 |
| 2.1.2 | Design package diagram | Medium | 3 |
| 2.1.3 | Design class diagram | Medium | 2 |
| 2.1.4 | Design sequence diagram | Medium | 10 |
| 2.1.5 | Design state diagram | Medium | 5 |
| 2.1.6 | Design activity diagram | Medium | 5 |
| 2.1.7 | Design database | Medium | 3 |
| 2.2 | Design user interface prototype | Medium | 10 |
| **3** | **Training** |  | ***34*** |
| 3.1 | Learn machine learning and deep learning | Complex | 14 |
| 3.2 | Learn Git | Simple | 1 |
| 3.3 | Learn React and Spring Boot | Medium | 7 |
| 3.4 | Learn Flutter | Simple | 5 |
| 3.5 | Learn Python and Flask | Medium | 7 |
| **4** | **Implementation** |  | ***192*** |
| 4.1 | AI module |  | 50 |
| 4.1.1 | Research model and algorithm for face verify | Simple | 1 |
| 4.1.2 | Attendance face verify code implementation | Complex | 21 |
| 4.1.3 | Get data from real environment | Simple | 6 |
| 4.1.4 | Test face verify accuracy with multiple use cases (different angle, low resolution camera…) | Complex | 12 |
| 4.1.5 | Optimize the face verify accuracy | Complex | 10 |
| 4.2 | Backend module |  | 60 |
| 4.2.1 | Login and authentication | Simple | 2 |
| 4.2.2 | Manage account | Medium | 4 |
| 4.2.3 | Manage tool | Simpe | 2 |
| 4.2.4 | Manage room | Medium | 5 |
| 4.2.5 | Manage subject | Simple | 2 |
| 4.2.6 | Manage semester | Medium | 4 |
| 4.2.7 | Manage shift | Complex | 9 |
| 4.2.8 | Manage examinee | Medium | 5 |
| 4.2.9 | Manage exam room | Complex | 6 |
| 4.2.10 | Manage report | Simple | 2 |
| 4.2.11 | Manage attendance | Medium | 5 |
| 4.2.12 | Manage notification | Complex | 7 |
| 4.2.13 | Manage schedule | Medium | 5 |
| 4.2.14 | Manage profile | Simple | 2 |
| 4.3 | Web module |  | 59 |
| 4.3.1 | Initialize and setup routing | Medium | 5 |
| 4.3.2 | Login and logout | Simple | 1 |
| 4.3.3 | Manage profile page | Simple | 2 |
| 4.3.4 | Manage account page | Medium | 5 |
| 4.3.5 | Manage tool page | Simple | 2 |
| 4.3.6 | Manage room page | Medium | 5 |
| 4.3.7 | Manage semester page | Medium | 5 |
| 4.3.8 | Manage subject page | Simple | 2 |
| 4.3.9 | Manage shift page | Medium | 5 |
| 4.3.10 | Manage exam room page | Complex | 8 |
| 4.3.11 | Manage examinee page | Medium | 5 |
| 4.3.12 | Manage report page | Simple | 2 |
| 4.3.13 | Manage notification | Complex | 5 |
| 4.3.14 | Dashboard page | Complex | 7 |
| 4.4 | Mobile module |  | 23 |
| 4.4.1 | Initialize and setup routing | Simple | 1 |
| 4.4.2 | Login and logout | Simple | 1 |
| 4.4.3 | Manage profile screen | Simple | 3 |
| 4.4.4 | Manage attendance screen | Medium | 4 |
| 4.4.5 | Manage report screen | Simple | 2 |
| 4.4.6 | Manage staff registration screen | Simple | 2 |
| 4.4.7 | Manage schedule screen | Medium | 4 |
| 4.4.8 | Manage notification | Medium | 6 |
| **5** | **Testing** |  | ***35*** |
| 5.1 | Deliver Report 5 – Test documentation |  | 35 |
| 5.1.1 | Create test plan | Simple | 6 |
| 5.1.2 | Unit test | Medium | 12 |
| 5.1.3 | Integration test | Simple | 6 |
| 5.1.4 | System test | Simple | 5 |
| 5.1.5 | Acceptance test | Simple | 5 |
| 5.1.6 | Finalize test report | Simple | 1 |
| **6** | **Finalize** |  | ***26*** |
| 6.1 | Deliver Report 6 – Software User Guide | Simple | 7 |
| 6.2 | Deliver Report 7 – Final Project Report | Simple | 7 |
| 6.3 | Create slide and prepare script for final thesis defense | Simple | 12 |
| ***Total Estimated Effort (man-days)*** | | | ***380*** |

### 1.2 Project Objectives

* Timeliness: 80%
* Allocated Effort: 4 (members) \* 14(weeks) \* 6(days/week) = 336 man-days

Table 2. Project Objectives

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Quality Stage** | **No. of Defects** | **% Of Defect** | **Notes** |
| 1 | Reviewing | 18 | 18 |  |
| 2 | Unit Test | 54 | 56 |  |
| 3 | System test | 25 | 26 |  |
| ***Total*** | | ***97*** | ***100%*** |  |

### 1.3 Project Risks

Table 3. Project Risks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Risk Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | Member drops out | High | Low | Reduce the scope and adjust the project plan. |
| 2 | Change business rule | Medium | High | Host a meeting with members and supervisor. |
| 3 | Equipment of members is broken | Medium | Medium | Members use other equipment until the broken one is fixed or buy a new one. |
| 4 | Cloud service is down during demonstration | High | Medium | Continue the demonstration using local hosts |
| 5 | Member is unavailable due to COVID-19 situation | Medium | High | Shift the available workload to healthy members to work overtime. |

## 2. Management Approach

### 2.1 Project Process

VaLua use the Scrum Framework due to the following reasons:

* Easy to learn and use.
* Allow the team to identify risks early and make it easier to response to them.
* Through feedback and regular testing, increase the quality of the development process and the quality of the final product.

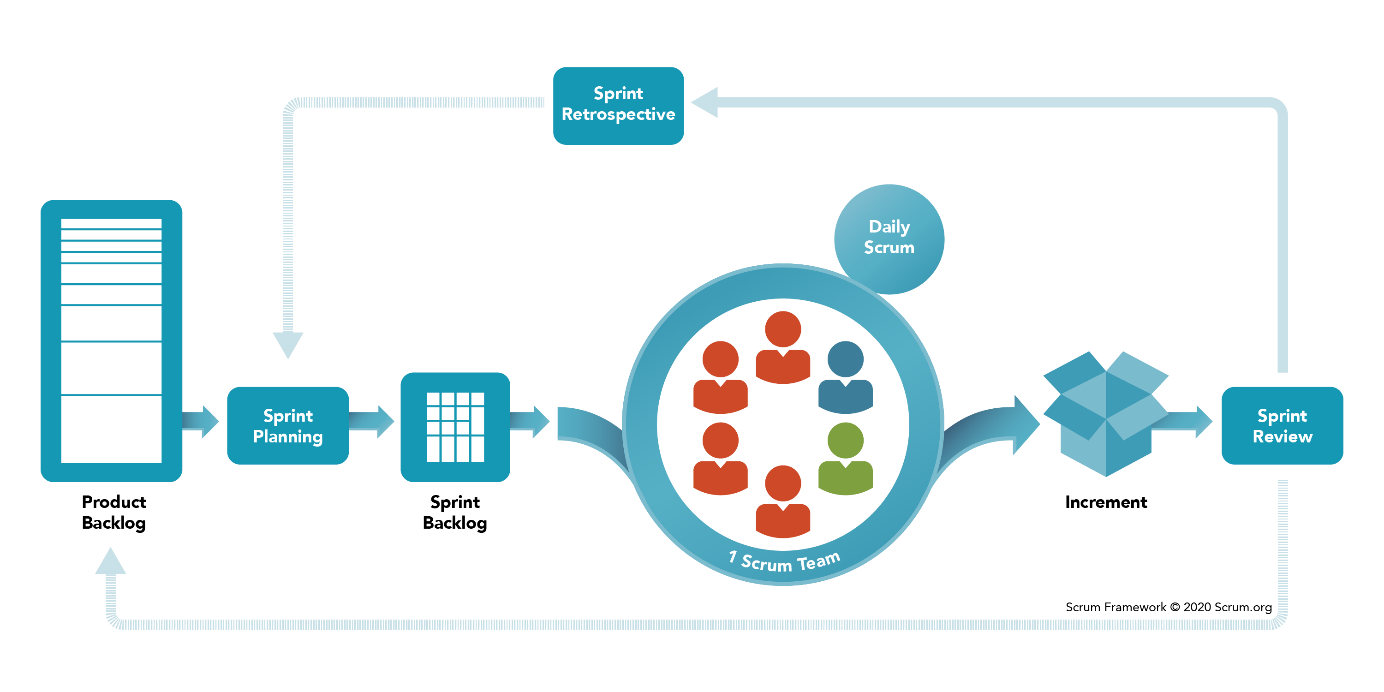


Figure 1. Scrum Framework

### 2.2 Quality Management

* Always cross-review code before merge.
* Using unit testing, integration testing, system testing and integration testing methodology to avoid defects and ensure the final product quality.

### 2.3 Training Plan

Table 4. Training Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Training Area** | **Participants** | **When, Duration** | **Waiver Criteria** |
| Java Spring Boot | All members | Week 1, 1 weeks | Mandatory |
| React | All members | Week 1, 1 weeks | Mandatory |
| Python Flask | All members | Week 6, 1 weeks | Mandatory |
| Face Verify | All members | Week 6, 1 weeks | Mandatory |
| Flutter | All members | Week 3, 5 days | Mandatory |
| Git, GitHub | All members | Week 3, 1 day | Mandatory |

## 3. Master Schedule

Table 5. Master Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Deliverable** | **Effort** | **Due Date** | **Deliverable Scope** |
| 1 | Project Plan | 2 | 07/01/2022 | A high-level overview of the project |
| 2 | Report 1 – Project introduction document | 3 | 11/01/2022 | Product background, business opportunity, existing products, main features and limitations |
| 3 | Report 2 – Project Management Document | 5 | 16/01/2022 | Project objectives, WBS, master schedule, main features, project organization and communication |
| 4 | Report 3 - SRS Document | 36 | 28/02/2022 | Business rules, messages, use case diagram, use case specifications |
| 5 | Report 4 - Software Design Document | 22 | 12/03/2022 | Architecture Design, Detailed Design, Database |
| 6 | Report 5 - Software Testing Document | 22 | 15/04/2022 | System test cases |
| 7 | Report 6 - Release Package and User Guides | 12 | 22/04/2022 | User manual guides |
| 8 | Report 7 - Appendix | 2 | 23/04/2022 | Glossary and references used during the projects |
| 9 | Code package 1 | 38 | 26/02/2022 | Features for manager (API, front end web app) and shift manager (API, front-end web app) |
| 10 | Code package 2 | 44 | 16/03/2022 | Features for checking attendances (API, front end mobile apps, verify face) |
| 11 | Code package 3 | 21 | 08/04/2022 | Rework features for shift manager (API, front end web app) |
| 12 | Code package 4 | 80 | 20/04/2022 | Add automation functionalities and optimization for cloud deployment (API, Front end web app, mobile apps and AI services) |
| 13 | Final Package | 1 | 23/02/2022 | Final software and documents |

## 4. Project Organization

### 4.1 Team & Structures

Diagram

Description automatically generated

Figure 2. Team & Structures

### 4.2 Roles & Responsibilities

Table 6. Roles & Responsibilities

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Scrum Master | - Planning and assigning tasks to team members.  - Tracking progress of the project.  - Support team members in both technical and business analysis. |
| Analysis Team | - Analyze user requirements.  - Define functional and non-functional requirements.  - Determine the appropriate architecture for the project.  - Determine the core flow of each feature. |
| A.I Team | - Research and implement the face verification A.I. |
| Backend Team | - Design database  - Implement the API for both web and mobile app |
| Frontend Team | - Design and implement the user interface for the web and mobile app |

## 5. Project Communication

### 5.1 Communication Plan

Table 7. Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| Meeting Supervisor | Supervisors and team members | Review project documents and work progress, support technical problems. | Twice per week | Online Google Meet.  Offline FU Campus. |
| Daily meeting | Team members | Review task progress, discuss problems and help each other. | Every workday | Online Google Meet, Slack |

### 5.2 External Interface

#### a. FU Contacts

Table 8. FU Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Contact Person**  **(name, position)** | **Contact address**  **(email, telephone)** | **Responsibility** |
| Supervisor | - Đoàn Nguyễn Thành Hòa | - hoadnt@fe.edu.vn | - Provide document template  - Supervise project status  - Give instruction to project team  - Review deliverables |

#### b. Customer Contacts

Table 9. Customer Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **Contact Person**  **(name, position)** | **Contact address**  **(email, telephone)** | **Responsibility** |
| End-user | - Đoàn Nguyễn Thành Hòa | - hoadnt@fe.edu.vn | - Provide user stories  - Review deliverables  - Supervise project |

## 6. Configuration Management

### 6.1 Tools & Infrastructures

Table 10. Tools & Infrastructures

|  |  |
| --- | --- |
| **Programming languages** | Java, Python, JavaScript, Dart |
| **Framework** | Java Spring Boot, React, Flask |
| **API** | Rest API |
| **DBMS** | PostgreSQL |
| **IDEs/Editors** | Visual Studio Code, IntelliJ, PyCharm |
| **UML tools** | Draw.io |
| **Version Control** | GitHub |
| **Deployment server** | Microsoft Azure |
| **Project management tool** | Jira |

### 6.2 Document Management

We use Google Drive along with GitHub to manage our documents. Google Drive allows us to easily edit the document directly, while GitHub allows us to manage the version of our documents and their changes.

### 6.3 Source Code Management

We use GitHub as our main source code management tool. Developers need to perform the task appropriate to the issues assigned on Jira. Developers need to create branches whose names correspond to the issues. All functionalities must be tested locally by the developers before creating a pull request.