

**Capstone Project Report**

**Report 2 – Project Management Plan**

– Hanoi, Jan 18th, 2021 –

**Table of Contents**

[I. Project Management Plan 3](#_Toc62069322)

[1. Overview 3](#_Toc62069323)

[1.1 WBS & Estimation 3](#_Toc62069324)

[1.2 Project Objectives 5](#_Toc62069325)

[1.3 Project Risks 5](#_Toc62069326)

[2. Management Approach 5](#_Toc62069327)

[2.1 Project Process 6](#_Toc62069328)

[2.2 Quality Management 6](#_Toc62069329)

[2.3 Training Plan 7](#_Toc62069330)

[3. Master Schedule 8](#_Toc62069331)

[4. Project Organization 9](#_Toc62069332)

[4.1 Team & Structures 9](#_Toc62069333)

[4.2 Roles & Responsibilities 9](#_Toc62069334)

[5. Project Communication 9](#_Toc62069335)

[5.1 Communication Plan 9](#_Toc62069336)

[5.2 External Interface 10](#_Toc62069337)

[6. Configuration Management 10](#_Toc62069338)

[6.1 Tools & Infrastructures 10](#_Toc62069339)

[6.2 Document Management 10](#_Toc62069340)

[6.3 Source Code Management 11](#_Toc62069341)

# I. Project Management Plan

## 1. Overview

### 1.1 WBS & Estimation

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **WBS Item** | **Complexity** | **Est. Effort (Man-days)** |
| **I** | **SOFA Project** |  | **375** |
| **1** | **Pre-Initiating Phase** |  | **5** |
| 1.1 | Select Project Manager | Simple | 1 |
| 1.2 | Determine the role of team member | Simple | 1 |
| 1.3 | Choose topic | Simple | 2 |
| 1.4 | Register capstone project | Simple | 1 |
| **2** | **Initiating phase** |  | **9** |
| 2.1 | Determine stakeholder | Medium | 1 |
| 2.2 | List up requirement | Simple | 2.5 |
| 2.3 | Define scope of project | Simple | 2.5 |
| 2.4 | Create report 1 - project introduction document | Medium | 2.5 |
| 2.6 | Modify project introduction document | Simple | 0.5 |
| **3** | **Project Plan Phase** |  | **8** |
| 3.1 | Choose working model process | Simple | 2.5 |
| 3.2 | Select tools and techniques | Simple | 0.5 |
| 3.3 | Create project schedule | Simple | 1 |
| 3.4 | Create risk management plan | Medium | 1 |
| 3.5 | Organize project resources | Simple | 2 |
| 3.6 | Create report 2 - software project management plan | Medium | 1 |
| **4** | **Executing Phase** |  | **346** |
| **4.1** | **Design** |  | **59.5** |
| 4.1.1 | List up use case | Simple | 1 |
| 4.1.2 | Create use case description | Medium | 1.5 |
| 4.1.3 | Create report 3 -Software requirement specification | Medium | 5 |
| 4.1.4 | Create Architecture design | Complex | 10 |
| 4.1.4.1 | Data Architecture design | Medium | 2 |
| 4.1.4.2 | Data flow architectures | Medium | 2 |
| 4.1.4.3 | Call and Return architectures | Medium | 2 |
| 4.1.4.4 | Object Oriented architecture | Medium | 2 |
| 4.1.4.5 | Layered architecture | Medium | 2 |
| 4.1.5 | Database design | Medium | 10 |
| 4.1.6 | GUI layout design | Complex | 25 |
| 4.1.7 | Create report 4 - Software design | Simple | 5 |
| 4.1.8 | Redesign use case | Simple | 2 |
| **4.2** | **Implementation** |  | **237** |
| 4.2.1 | Back-end (RESTful API) |  | 92 |
| 4.2.1.1 | Configuration and setup environment | Simple | 2 |
| 4.2.1.2 | Create database | Simple | 15 |
| 4.2.1.3 | Coding common, Middleware, DBContext | Medium | 5 |
| 4.2.1.4 | Coding Model and Data Access Object | Medium | 10 |
| 4.2.1.5 | Coding controller handler request of manager page | Complex | 25 |
| 4.2.1.6 | Coding controller handler request of user app | Complex | 35 |
| 4.2.2 | Front-end |  | 140 |
| 4.2.2.1 | SOFA Manager (Web flatform) | Complex | 40 |
| 4.2.2.2 | SOFA (Android app flatform) | Complex | 100 |
| 4.2.3 | Deploy |  | 5 |
| 4.2.3.1 | Deploy host and domain config | Medium | 5 |
| 4.3 | **Test** |  | **49** |
| 4.3.1 | Test manager website |  | 24 |
| 4.3.1.1 | Create test cases for unit test | Simple | 2 |
| 4.3.1.2 | Unit test | Simple | 2 |
| 4.3.1.3 | Fix bug | Medium | 2 |
| 4.3.1.4 | Create test cases for integration test | Simple | 2 |
| 4.3.1.5 | Integration test | Simple | 2 |
| 4.3.1.6 | Fix bug | Medium | 2 |
| 4.3.1.7 | Create test cases for system test | Simple | 2 |
| 4.3.1.8 | System test | Simple | 2 |
| 4.3.1.9 | Fix bug | Medium | 2 |
| 4.3.1.10 | Create test case for acceptance test | Simple | 2 |
| 4.3.1.11 | Acceptance test | Simple | 2 |
| 4.3.1.12 | Fix bug | Medium | 2 |
| 4.3.2 | Test user application |  | 24 |
| 4.3.2.1 | Create test cases for unit test | Simple | 2 |
| 4.3.2.2 | Unit test | Simple | 2 |
| 4.3.2.3 | Fix bug | Medium | 2 |
| 4.3.2.4 | Create test cases for integration test | Simple | 2 |
| 4.3.2.5 | Integration test | Simple | 2 |
| 4.3.2.6 | Fix bug | Medium | 2 |
| 4.3.2.7 | Create test cases for system test | Simple | 2 |
| 4.3.2.8 | System test | Simple | 2 |
| 4.3.2.9 | Fix bug | Medium | 2 |
| 4.3.2.10 | Create test case for acceptance test | Simple | 2 |
| 4.3.2.11 | Acceptance test | Simple | 2 |
| 4.3.2.12 | Fix bug | Medium | 2 |
| 4.3.3 | Summary and evaluation | Simple | 1 |
| 4.1.9 | Create report 5 - Test case document | Simple | 2.5 |
| 4.1.10 | Create report 6 - Software user guide | Simple | 2.5 |
| **4.4** | **Finish executing phase** |  | **0.5** |
| **5** | **Monitoring and controlling** |  | **2** |
| 5.1 | Progress report | Simple | 2 |
| **6** | **Close** |  | 5 |
| 6.1 | Create final report | Medium | 5 |
| 6.2 | Project complete | Simple | 0 |

### 1.2 Project Objectives

Objectives:

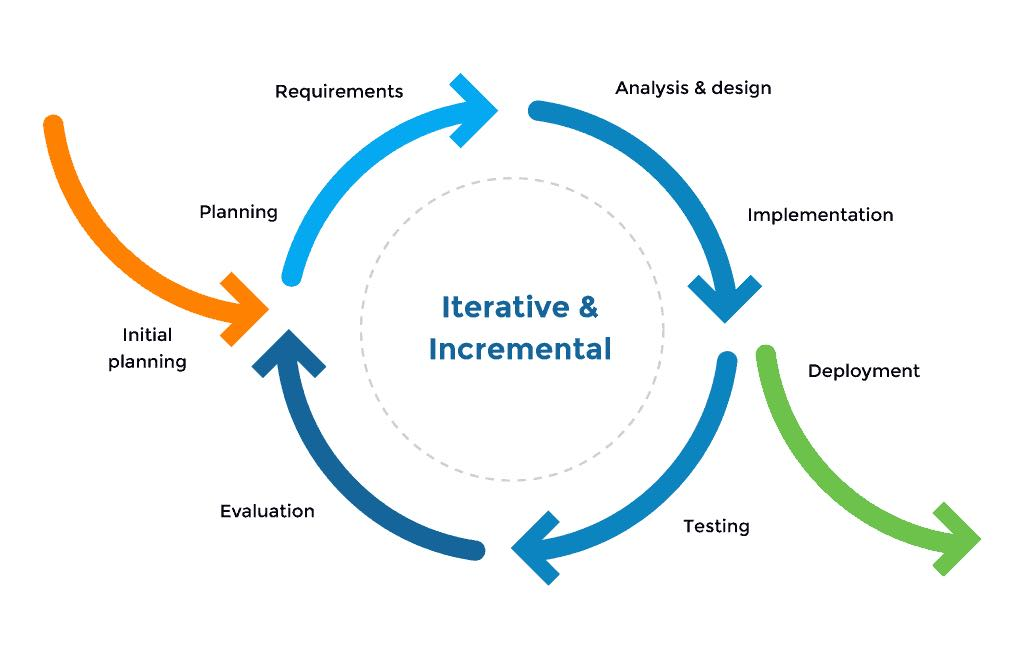
* This project must be finished by 30/04/2021
* All team members will follow the task assigned
* All team members learn new knowledge, new technology
* Coding phase begin before 05/02/2021
* Have real user used the app
* Have a beta test before finish implementation phase

### 1.3 Project Risks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Risk** | **Description** | **Impact** | **Possibility** | **Response Plans** |
| 1 | Inaccurate Estimations | Schedule is inaccurate | Medium | Medium | 1) Create schedule carefully 2) Add backup time (or backup schedule) 3) Focus on close targets |
| 2 | Poor Quality Code | The code may be difficult to read, meaning it is difficult for other developers to review or make changes | High | Low | 1) Code reviews 2) Clear coding standards and guides 3) Testing of all code |
| 3 | Inadequate Human Resources | Don't have enough member to complete project be on time | Medium | Low | 1) Register overtime for team member |
| 4 | Technical Difficulties | Have issues during coding | Medium | Medium | 1) Choose design pattern appropriate 2) Create architecture design more detail and exactly 3) Have a professional in team |
| 5 | Security | The system is unsafe for user, hacker can attack and take user information… | High | High | 1) Using token for improving backend security 2) Add more algorithm for hash private information 3) Take preventive measures hacking |
| 6 | Serious error in synchronization between web and mobile platform | display on browser and mobile is more difference. Maybe, the component just can use on web but not mobile | Medium | High | 1) Choose another component to fix 2) Make components difference but similar feature |
| 7 | lack of requirement | Developer create code was not enough requirement | High | Medium | 1) Code reviews 2) Clear coding standards and guides 3) Testing of all code |

## 2. Management Approach

### 2.1 Project Process



*Figure 2.1 – Iterative and incremental Model*

SOFA project uses the Iterative and incremental model.

The iterative model feels like mini waterfalls chained together. Instead of designing and gathering the requirements for the whole applications, we slice the project into functional portions that progress through the waterfall steps. Also, this model has these useful advantages:

* You can develop prioritized requirements first.
* Initial product delivery is faster.
* Customers gets important functionality early.
* Lowers initial delivery cost.
* Each release is a product increment, so that the customer will have a working product at hand all the time.
* Customer can provide feedback to each product increment, thus avoiding surprises at the end of development.
* Requirements changes can be easily accommodated.

### Quality Management

### Defect Prevention:

### Once a defect is found, the related person in charge should be notified immediately.

### The defect must be assessed carefully such as “how bad is the defect?”, “how long does it take to fix the defect?”.

### The deadline for fixing defects must be clearly stated.

### Reviewing:

### The person in charge must be honest and show no favor over any member. If something goes wrong, that person must notify the person who takes responsibility for that defect.

### The defect must be logged on Bug Tracking software with detail such as priority.

### The person who takes responsibility for the found defects must take actions accordingly.

### Unit Testing:

### The person in charge must prepare test cases carefully and accurately. The test cases must match well with the system.

### The defect must be logged on Bug Tracking software with detail such as priority.

### The person who takes responsibility for the found defects must take action accordingly.

### Integration Testing

### The person in charge must prepare test cases carefully and accurately. The test cases must match well with the system and architecture design.

### The defect must be logged on Bug Tracking software with detail such as priority.

### The person who takes responsibility for the found defects must take action accordingly.

### Internal modules within the system work smoothly.

### System Testing

### The person in charge must prepare test cases carefully and accurately. The test cases must match well with the system and system design.

### The defect must be logged on Bug Tracking software with detail such as priority.

### The person who takes responsibility for the found defects must take action accordingly.

### System testing test cases cover the entire system functionality and the communication under development with external systems.

### Acceptance Testing

### The person in charge must prepare test cases carefully and accurately. The test cases must match well with the system and business requirements.

### The person who takes responsibility for the found defects must take action accordingly.

### The defect must be logged on Bug Tracking software with detail such as priority.

### If there are customers, customers should take part in Acceptance testing.

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

### 2.3 Training Plan

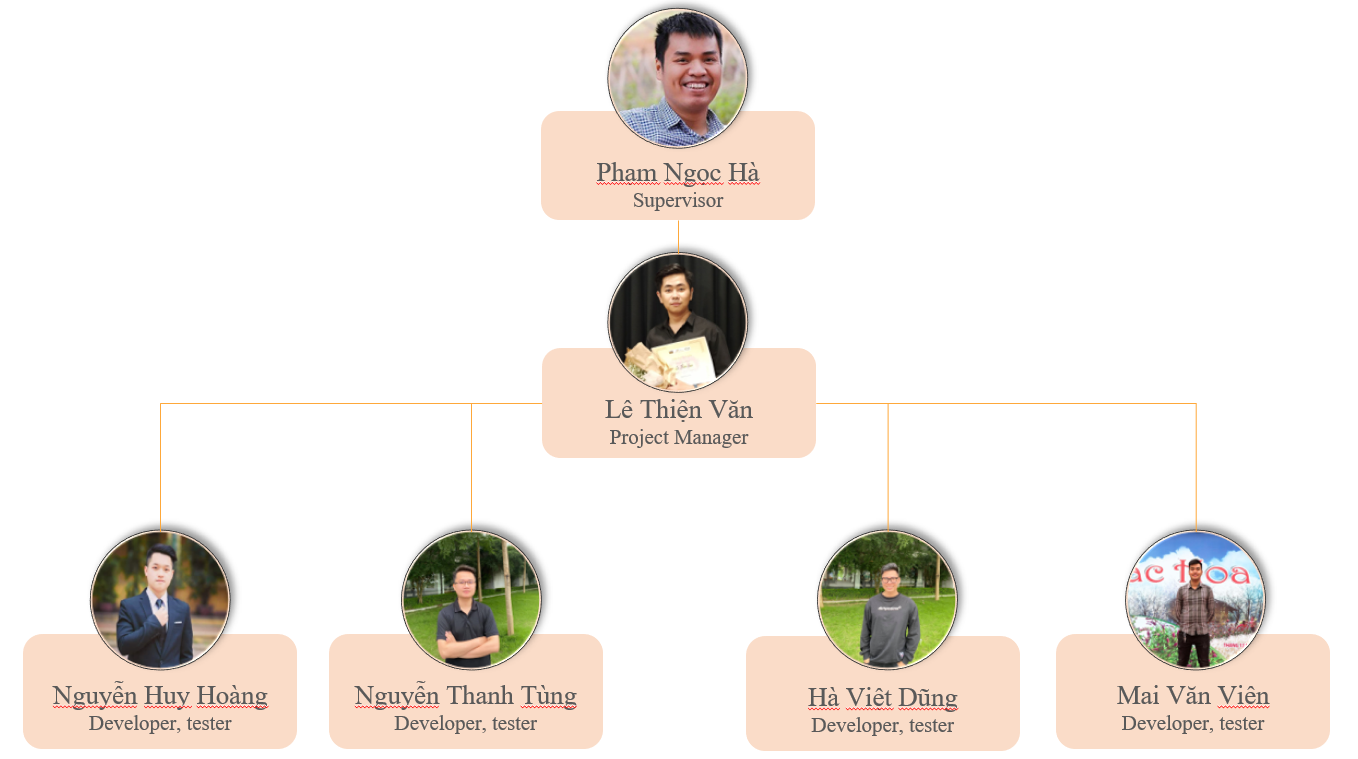
|  |  |  |  |
| --- | --- | --- | --- |
| Training Area | Participants | When, Duration | Waiver Criteria |
| Asp.net core API | All team members | 16/01/2021,  (1 day) | Mandatory |
| Python | All team members | 17/01/2021  (1 day) |  |
| Angular 8 | All team members | 18/01/2021  - 22/01/2021  (1 week) |  |
| React native | All team members | 01/02/2021  - 05/02/2021  (1 week) |  |
| Coding convention & bug logging convention | All team members | 06/02/2021  (1 day) |  |

## 3. Master Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Deliverable** | **Due Date** | **Deliverable Scope** | |
| 1 | Project Plan | 18/01/2021 | Project Schedule, Risk Management Plan, Resources Management Plan |
| 2 | SRS | 26/01/2021 | SRS |
| 3 | Design | 05/02/2021 | Architecture Design, Detailed Design, GUI Design, Database |
| 4 | Iteration 1 | 04/04/2021 | Code & Unit test, System test cases |
| 5 | Iteration 2 | 18/04/2021 | Code & Unit test, System test cases |
| 7 | UAT Package | 24/04/2021 | Codes, System test reports |
| 8 | Final Package | 30/04/2021 | Final Codes & documents, User manual |

## 4. Project Organization

### 4.1 Team & Structures



### 4.2 Roles & Responsibilities

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Project Manager | Le Thien Van |
| Analysis Leader | Nguyen Huy Hoang |
| Analysis Member | Nguyen Thanh Tung, Mai Van Vien |
| Design Leader | Ha Viet Dung |
| Design Member | Le Thien Van |
| Technical Leader | Le Thien Van |
| Developer | Mai Van Vien, Ha Viet Dung, Nguyen Thanh Tung, Nguyen Huy Hoang |
| Test Leader | Mai Van Vien |
| Test Member | Nguyen Huy Hoang, Ha Viet Dung |

## 5. Project Communication

### 5.1 Communication Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Communication Item** | **Who/ Target** | **Purpose** | **When, Frequency** | **Type, Tool, Method(s)** |
| Weekly meeting | Supervisor and team members | Review members’ work achievements during the week and report the project’s progress and status. | Every tuesday | Face to face |
| Daily meeting | Team members | Report the progress that members achieved each day. | 21h,  every day | Google meet |
| Unscheduled Meeting | Team members | Occurred when there's a critical problem that need to be resolved immediately |  | Face to face, google meet |

### 5.2 External Interface

#### a. FU Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| Function | Contact Person  (name, position) | Contact address  (email, telephone) | Responsibility |
| Supervisor | Phạm Ngọc Hà | [HaPN10@fe.edu.vn](mailto:HaPN10@fe.edu.vn)  0988623000 | - 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 |
| Supervisor | Phạm Ngọc Hà | [HaPN10@fe.edu.vn](mailto:HaPN10@fe.edu.vn)  0988623000 | - Provide document template - Give instruction to project team - Review deliverables - Supervise project status |

## 6. Configuration Management

### 6.1 Tools & Infrastructures

|  |  |
| --- | --- |
| **Programming languages** | C#, Python, Html, Css, JavaScript, TypeScript |
| **Framework** | ASP.Net core API, Angular 8, React Native, Django |
| **API** | Clarifai, ZaloPay |
| **DBMS** | SQL Server |
| **IDEs/Editors** | Visual Studio Code, Visual Studio 2019 |
| **UML tools** | Visio, Lucid chart, draw.io |
| **Version Control** | Git hub |
| **Deployment environment** | Window server 2012 |
| **App environment** | Android |
| **Deployment server** | IIS Server (Server machine) |
| **Project management tool** | Trello |

### 6.2 Document Management

We use Git hub and Google drive as our primary tool for sharing, editing and version control of our documents, along with Status Reports with-in each document. It allows us to see what is changed in the documents and who is responsible for that change, reverse and simultaneously compare between versions of the document.

### 6.3 Source Code Management

For version control of our source code, we use Git hub. It tracks the changes team members make to files, so we have a record of what has been done, and we can revert to specific versions should we ever need to. Git hub also makes collaboration easier, allowing changes by multiple people to all be merged into one source.

Git hub also features branches that provide an isolated environment for every change to our codebase. When a team member wants to start working on something - no matter how big or small - he creates a new branch. This ensures that the master branch always contains production - quality code.