Fpt University

TRƯỜNG ĐẠI HỌC FPT

CAPSTONE PROJECT REPORT

Report 2 – Project Management Plan

– Hanoi, December 2024 –

Table of Contents

I. Record of Changes

Date	A* M, D	In charge	Change Description
02/09/2024	A	NinhNT	Initiation
03/09/2024	A	NinhNT	Add Scope & Estimation
03/09/2024	A	NinhNT	Add Project Communication
03/09/2024	A	NinhNT	Add Management approach
03/09/2024	A	ThuyDTT	Add manage approach: project process, quality management
04/09/2024	A	HauNX	Add Project Objectives
04/09/2024	M	ThuyDTT	Edit Scope & Estimation
04/09/2024	A	HauNX	Add Project Deliverables
05/09/2024	M	HauNX	Edit Project Objectives and Configuration Management
05/09/2024	M	ThuyDTT	Edit Quality Management
06/09/2024	M	HauNX	Edit Project Communications
09/09/2024	M	HauNX	Update project plan for iteration 1
07/10/2024	M	HauNX	Update project plan for iteration 2
08/11/2024	M	NinhNT	Update project plan for iteration 3

*A - Added M - Modified D - Deleted

II. Project Management Plan

1. Overview

1.1 Scope & Estimation

In this project, we define 1 man-day equals to 6 hours

#	WBS Item	Complexity	Est. Effort (man-days)
1	***Account Management***		***56***
1.1	View users list	Complex	10
1.2	Import user list	Complex	10
1.3	Forgot password	Complex	10
1.4	View Profile	Medium	8
1.5	Create account	Medium	8
1.6	Login	Complex	10
2	***Course Management***		***18***
2.1	View course list	Medium	8
2.2	Create course	Complex	10
3	***Student management***		***44***
3.1	View student registration list	Complex	10
3.2	Student registration details	Medium	8
3.3	View student details	Medium	8
3.4	View student list	Complex	10
3.5	View student result	Medium	8
4	***Student group management***		***43***
4.1	View student group list	Normal	5
4.2	Create student group	Complex	10
4.3	View student group details	Complex	10
4.4	Add students to a student group	Medium	8
4.5	Edit student group	Complex	10
5	***Supervisor management***		***18***
5.1	View supervisor list	Medium	8

5.2	Add supervisors to a course	Complex	10
6	***Team management***		***42***
6.1	View team list	Medium	8
6.2	Create team	Medium	8
6.3	View team details	Complex	10
6.4	Edit team	Medium	8
6.5	Add volunteers to a team	Medium	8
7	***Volunteer management***		***41***
7.1	View volunteer registration list	Complex	10
7.2	Volunteer registration details	Normal	5
7.3	Volunteer details	Medium	8
7.4	View volunteer list	Complex	10
7.5	Accept/Reject registration	Medium	8
8	***Night Shift management***		***68***
8.1	View night shift list	Complex	10
8.2	Register for available time	Complex	10
8.3	View staffs' freetime	Medium	8
8.4	View night shift details	Complex	10
8.5	View list of night shift rejection	Complex	10
8.6	View my night shift assignment	Complex	10
8.7	Night shift config	Complex	10
9	***Post management***		***37***
9.1	View post list (Guest's page)	Medium	8
9.2	Guest home page	Medium	8
9.3	View post list	Normal	5
9.4	View post details	Medium	8
9.5	View post details (Guest's page)	Medium	8
10	***Report management***		***20***
10.1	View report details	Complex	10
10.2	View report list	Complex	10

11	***Feedback management***		***18***
11.1	View feedback list	Normal	5
11.2	View feedback details	Normal	5
11.3	Send feedback	Medium	8
Total Estimated Effort (man-days)			**405**

Table 1: Scope and Estimation

1.2 Project Objectives

1.2.1 Allocated Effort

#	Members	Weekdays	Weekends
1	Nguyễn Tuấn Ninh	6 hours	8 hours
2	Nguyễn Xuân Hậu	6 hours	8 hours
3	Nguyễn Duy Phúc	6 hours	8 hours
4	Đỗ Thị Thanh Thủy	6 hours	8 hours
5	Đậu Đình Mạnh	6 hours	8 hours

Table 2: Allocated Effort

Total allocated effort: 428 man-days

1.2.2 Defect Rate

We identify 1 man-month (MM) as 20 man-days. Since the team is developing a new system, we keep the defect rate (No. Def/MM) at around 4.2. Therefore the number of defects we want to identify after delivering the product to the customer is around 90. We will divide the defects into categories as below:

#	Testing Stage	Test Coverage	No. of Defects	% of Defect	Notes
1	Review	100%	50	55.55%	Perform to identify defects at an early stage
1	Unit Test	100%	0	0%	They are written and run by software developers to ensure that a section of an application (known as the "unit")

					meets its design and behaves as intended.
2	Integration Test	100%	25	27.78%	Individual software modules are combined and tested as a group.
3	System Test	100%	10	11.11%	Performed on the entire system, tests not only the design but also the behavior and even the believed expectations of the customer.
4	Acceptance Test	100%	5	5.56%	Users of the system perform tests in line with what would occur in real-life scenarios

Table 3: Defect distribution

Review Stage (55.55%): This high percentage indicates that many defects were found at an early stage, highlighting the importance of the review process in identifying issues before formal testing begins, thus helping to prevent costly fixes later in the development process.

Unit Test Stage (0%): The defect is 0% at this stage means that all the bugs at this level are fixed before coming to the next stage. If there are any defects remaining at this level, the project cannot run properly.

Integration Test Stage (27.78%): The number of defects discovered drops further in integration testing, which implies that the individual modules, once tested, tend to work well together. This stage identifies issues that may arise when integrating different modules, but since the number of defects found is relatively low, it indicates that the earlier tests were effective.

System Test Stage (11.11%): The number of defects identified in system testing is the lowest among the stages, suggesting that the system, when tested as a whole, performed better. This stage evaluates overall functionality and compliance with specifications, highlighting effective testing practices in the earlier stages.

Acceptance Test Stage (5.56%): A small number of defects were found during acceptance testing, indicating that the system met user requirements and expectations. This is a strong indicator of quality and thoroughness throughout the development and testing process.

1.2.3 Timeliness

We meet at least 90% of the assigned deadlines to ensure the project is completed on time.

1.2.4 Requirement Completeness

With the number of use cases identified, we aim to complete at 100%.

1.3 Project Risks

#	Risk Description	Impact	Possibility	Response Plans
1	Poor communication among members leads to missing critical information	High	High	Set up meetings and use project management tools to keep information updated. Encourage direct and clear communication to avoid misunderstandings.
2	Insufficient time spent on the capstone project of team members	High	High	Each member regularly reports any abnormal events to the whole team to quickly support. Make a clear commitment measured by working hours per day from the project commencement.
3	Customers change requirements and scopes are expanded beyond expectation	Medium	Medium	Establish a clear scope and regularly review it to ensure that it remains under control. Any proposed changes should be carefully reviewed before being implemented.
4	Lack of members due to a member being forced to	High	Low	Members agree to commit to spending enough time. Members have to learn additional skills necessary to serve the

	leave due to not being qualified to defend the project.			project and agree to fulfill their responsibilities, avoiding being forced to leave the group due to not completing the minimum tasks.

Table 4: Project risk

2. Management Approach

2.1 Project Process

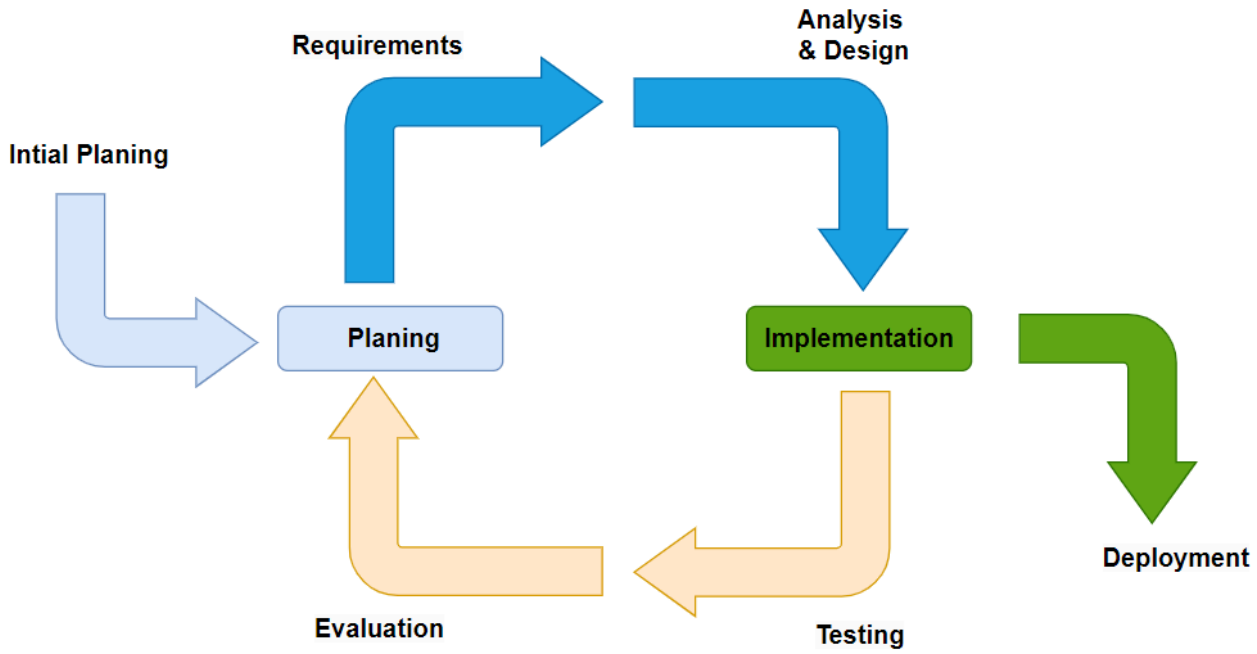

Figure 1: Iterative and Incremental Model Process

After carefully reviewing the models for software development, we decided to apply the **Iterative and Incremental model** for this project. This software development model is an approach to software development through building the system step by step and iterating many times. The software parts will be developed and improved gradually through each development cycle, continuously integrating user input and gradually adding functions. Each part of the project will have complete functions and can operate independently and each version will be supplemented with new and expanded functions. Some reasons for choosing this development model:

- Flexibility: Because the model is iterative, it is easy to change the requirements through iterations.

- Risk management: Reduce risks by detecting and handling problems at an early stage.
- Continuous feedback: By developing software in small steps, customers can test and give feedback continuously.
- Quality assurance: Because it is tested at each stage, the quality of the product is very reliable.
- Measurable progress: Set clear goals to track the progress of the project.

However, Iterative and Incremental model also has some disadvantages that we need to know when using it:

- Difficulty in managing the organization: Continuous iterations and step-by-step delivery can be difficult to manage and require close coordination.
- Need for a long-term vision: Continuous loops develop step by step, but to ensure smoothness between iterations, an overall view of the product is needed.

In this project, we plan to divide it into 4 specific iterations as follows:

- **Project Initiating (1 week - from 02/09/2024 to 08/09/2024)**
  - Kick-off Meeting
  - Determine Project Scope and Features
  - Prepare Project Introduction (Report 1)
  - Create preliminary Project Plan (Report 2 - Version 0.9)

- **Iteration 1 (4 weeks - from 09/09/2024 to 06/10/2024)**
  - Identify functions for iteration 1, update Project Plan (Report 2 - Version 1.0)
  - Create high-level requirements and analyze detailed requirements for iteration 1 and start Software Requirement Specification (Report 3 - Version 1.0)
  - Create high-level design and analyze detailed design for iteration 1 features and start Software Design Document (Report 4- Version 1.0)
  - Coding the planned functions
  - Conduct testing and bug fixing (Report 5.1, 5.2, 5.3 - Version 1.0)
  - Evaluate iteration effectiveness

- **Iteration 2 (3 weeks - from 07/10/2024 to 27/10/2024)**
  - Identify functions for iteration 2, update Project Plan (Report 2 - Version 1.1)

- Revise requirements (if requirements change) and analyze detailed requirements for iteration 2 features and update Software Requirement Specification (Report 3 - Version 1.1)
- Analyze detailed design for iteration 2 features and update Software Design Document (Report 4 - Version 1.1)
- Proceed with coding the planned functions
- Conduct testing and bug fixing (Update report 5.1, 5.2, 5.3 - Version 1.1)
- Evaluate iteration effectiveness

- **Iteration 3 (3 weeks - from 28/10/2024 to 17/11/2024)**
  - Identify functions for iteration 3, update Project Plan (Report 2 - Version 1.2)
  - Revise requirements (if requirements change) and analyze detailed requirements for iteration 3 features and update Software Requirement Specification (Report 3 - Version 1.2)
  - Analyze detailed design for iteration 3 features and update Software Design Document (Report 4 - Version 1.2)
  - Proceed with coding the planned functions
  - Conduct testing and bug fixing (Update report 5.1, 5.2, 5.3 - Version 1.2)
  - Evaluate iteration effectiveness

- **Iteration 4 (3 weeks - from 18/11/2024 to 08/12/2024)**
  - Identify functions for iteration 4, update project plan (Report 2 - Version 1.3)
  - Revise requirements (if requirements change) and analyze detailed requirements for iteration 4 features and update Software Requirement Specification (Report 3 - Version 1.3)
  - Analyze detailed design for iteration 4 features and update Software Design Document (Report 4 - Version 1.3)
  - Proceed with coding the planned functions
  - Conduct testing and bug fixing (Update report 5.1, 5.2, 5.3 - Version 1.3)
  - Proceed with project deployment
  - Prepare Report 6 - User Guide
  - Evaluate iteration effectiveness

- **Project Closing (1 week - from 09/12/2024 to 15/12/2024)**
  - Report 7 - Final Report

- Submit The Capstone Project
- Prepare Thesis Defense Slides and Practical Presentation
- Evaluate project effectiveness

2.2 Quality Management

2.2.1 Defect Prevention

- To minimize potential errors before developing the product
- Design the system architecture carefully
- Testers must collect a list of common errors and plan to make changes.
- There must be a plan to train and guide members on common problems and errors during the development process.
- Managers need to determine the appropriate plan and should make some changes if necessary.

2.2.2 Reviewing

- After understanding the customer needs, review the requirements from members, developers, BAs or customers to ensure all perspectives are in sync and can be thoroughly reviewed before proceeding to the next steps.
- Review documents thoroughly to ensure accuracy and completeness.
- Developers perform code reviews to detect logical or syntactic errors.
- Compare the application with existing systems to ensure the work is being done in the right direction.

2.2.3 Unit Testing

- Ensure that each small component of the software works as expected by writing small, separate unit tests with logical names for maintenance.
- Use automated testing frameworks to easily automate testing.
- Ensure that the tests run quickly and are reusable.
- Ensure that all edge cases are fully tested.

2.2.4 Integration Testing

- Make sure that unit testing is done before moving on to Integration Testing.
- Make sure the modules are working properly after integration.
- Prepare integration test plans and reports
- Fully prepare test cases and test data.

2.2.5 System Testing

- Ensure integration testing phases are completed.
- Ensure functional and non-functional requirements are met.
- To ensure smooth and reliable operation of the software, system testing must be planned and testing should mainly focus on performance, security, recovery, external interface, user-system interaction, etc.

2.2.6 Acceptance Testing

- Ensure system testing phases are completed.
- Ensure that all business, functional, and non-functional requirements are met.
- Ensure smooth and intuitive interaction between the user and the system, with workflows reflecting actual use cases.
- Verify that the software meets all acceptance criteria as defined by stakeholders for final approval and sign-off for deployment.

2.3 Project Training Plan

Training Area	Participants	When, Duration	Waiver Criteria
ASP.NET	NinhNT, ManhDD, PhucND, ThuyDTT	14/9- 20/9	Mandatory
React	NinhNT, ManhDD, HauNX	14/9- 20/9	Mandatory
Git, Github	All Member	14/9- 15/9	Mandatory
Testing	All Member	20/9- 25/9	Mandatory

Table 5: Training plan

3. Project Deliverables

#	Deliverable	Due Date	Notes
Project Initiating (from 02/09/2024 to 08/09/2024)			
1	Project Introduction	08/09/2024	Prepare an overview of the project: background, objective, and scope

2	Project Plan v0.9	08/09/2024	Prepare Project plan at a high level
Iteration 1 (from 09/09/2024 to 06/10/2024)			
3	Project Plan v1.0	09/09/2024	Identify functions for iteration 1, update Project Plan
4	Software Requirement Specification v1.0	16/09/2024	High-level requirements and analyze detailed requirements for iteration 1
5	Software Design Document v1.0	22/09/2024	High-level design and analyze detailed design for iteration 1 features and start Software Design Document
6	Software Test Document (STD)	01/10/2024	Prepare testing results: Unit Test, Integration Test, System Test, Acceptance Test
6	Product Release 1	06/10/2024	Product Release after Iteration 1
Iteration 2 (from 07/10/2024 to 27/10/2024)			
7	Project Plan v1.1	07/10/2024	Identify functions for iteration 2, update Project Plan
8	Software Requirement Specification v1.1	12/10/2024	Update Software Requirement Specification with analysis of detailed requirements for iteration 2
9	Software Design Document v1.1	17/10/2024	Update the Software Design Document with the analysis of the detailed design for iteration 2 features
10	Software Test Document (STD)	23/10/2024	Prepare testing results: Unit Test, Integration Test, System Test, Acceptance Test
11	Product Release 2	27/10/2024	Product Release after Iteration 2
Iteration 3 (from 28/10/2024 to 17/11/2024)			
12	Project Plan v1.2	28/10/2024	Identify functions for iteration 3, update Project Plan
13	Software Requirement Specification v1.2	02/11/2024	Update Software Requirement Specification with analysis of detailed requirements for iteration 3

14	Software Design Document v1.2	07/11/2024	Update the Software Design Document with analysis of the detailed design for iteration 3 features
15	Software Test Document (STD)	13/11/2024	Prepare testing results: Unit Test, Integration Test, System Test, Acceptance Test
16	Product Release 3	17/11/2024	Product Release after Iteration 3
Iteration 4 (from 18/11/2024 to 08/12/2024)			
17	Project Plan v1.3	19/11/2024	Identify functions for iteration 4, update Project Plan
18	Software Requirement Specification v1.3	24/11/2024	Update Software Requirement Specification with analysis of detailed requirements for iteration 4
19	Software Design Document v1.3	29/11/2024	Update the Software Design Document with analysis of the detailed design for iteration 4 features
20	Software Test Document (STD)	30/11/2024	Prepare testing results: Unit Test, Integration Test, System Test, Acceptance Test
21	Product Release 4	04/12/2024	Product Release after Iteration 4
22	User Guide Document	04/12/2024	Provide straightforward instructions for using the developed software, ensuring the language is simple and direct.
Project Closing (from 09/12/2024 to 15/12/2024)			
23	Final Project Report	09/12/2024	Includes all the previous project documents in combined, source code, and thesis presentation slide

Table 6: Project Deliverables

4. Responsibility Assignments

4.1 Project Organization

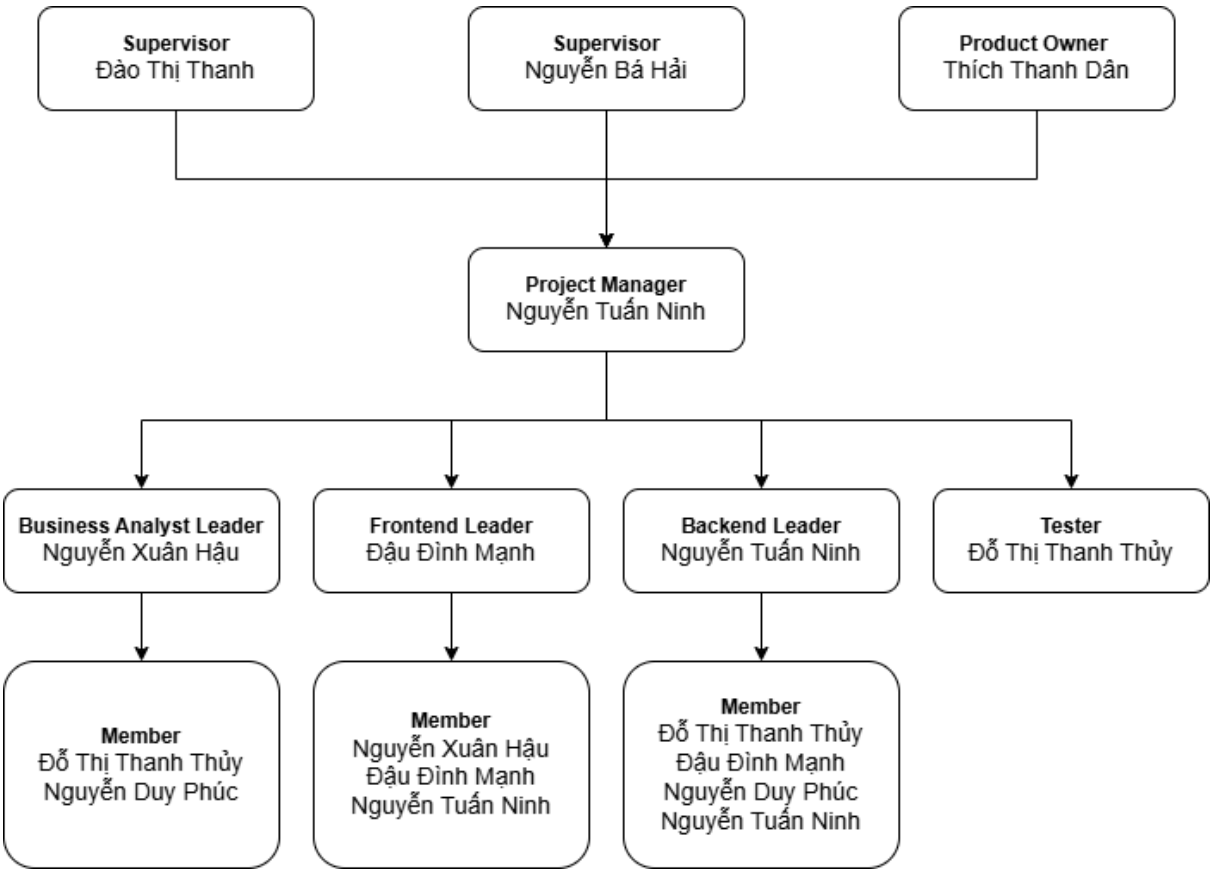

Figure 2: Project Organization

4.2 Role and responsibilities

Role	Responsibilities
Supervisor	Oversees the overall project to ensure alignment with strategic goals, provides guidance and mentorship to the project team, ensures that the project objectives are met, and approves major project deliverables and milestones.
Product Owner	The Product Owner defines and prioritizes the product backlog, ensuring alignment with business goals and user needs.
Project Manager	Manages the project from inception to completion, ensuring it stays on schedule, creates and maintains project plans, allocates resources and assigns tasks, and monitors progress and performance.

Business Analyst Leader	Leads the business analysis efforts, ensuring that all business requirements are properly understood and met, analyses and documents business processes and requirements, and works with stakeholders to gather feedback
Business Analyst	Supports the Business Analyst Leader in gathering and analyzing requirements, conducting interviews and workshops with stakeholders, and translating business requirements into technical specifications
Front end Leader	Leads the front-end development team, oversees the design and development of the front-end interface reviews code, and provides feedback to front-end members
Front end member	Develops and implements the front-end interface as per the project requirements, and writes front-end code.
Back end Leader	Leads the back-end development team, ensuring the proper functionality of the server, database, and application logic, Oversee back-end architecture and implementation.
Front end member	Implements the back-end services and ensures integration with the front end, writes back-end code
Tester leader	Leads the testing team to ensure the quality of the product before release, develops test plans and strategies, oversees the execution of test cases and reports defects, and ensures proper documentation of testing results.
Tester	Executes tests, and identifies defects in the system. Testers also document, and report bugs to the development team, and re-test fixes to ensure issues are resolved.

Table 7: Row and responsibilities

4.3 Responsibility assignments

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

Responsibility	NinhNT	HauNX	PhucND	ThuyDTT	ManhDD
Document and Report					
Project Planning & Tracking	D	S, R	I	I	I
Prepare Project Introduction Document	I	D, R	D	S	S
Prepare SRS Document	D	D, R	D	D	D
Prepare SDS Document	D, R	D	D	S	D
Prepare Test Documents	D	D	D	D, R	D
Prepare User Guides	S	D, R	D	S	S
Prepare Final Report	D	D, R	D	D	D
Prepare Presentation Slide	R	D	S	D	S
Design					
System Architecture Design	D, R	S	I	I	I
Class Diagram	R	I	S	S	D
Sequence Diagram	D, R	I	D	D	D
Use Case Specification	D, R	S	D	D	D
Screen Design	R	S	D	D	I
Database Design	D, R	S	I	s	I
Implementation					
Control source code	D, R	I	I	I	I

Code front-end	D	D	I	D	D, R
Code back-end	D, R	I	D	D	D
Deploy web	D, R	I	I	I	I
Test					
Create test plan	R	I	I	D	I
Create test report	D	I	D	R	D

Table 8: Responsibility assignments

5. Project Communications

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Daily meeting	All Member	- Check progress - Discuss problem	21:00 every day	Google Meet
Weekly meeting with supervisors	All Member, supervisor	- Report progress	3 times a week	Google Meet, Microsoft team
Team Messaging	All Member, supervisor	- Notice about unscheduled meeting - Ask supervisors about the issues to get advice	When necessary	Online - Messenger, Zalo
Unscheduled Meeting	All Member, supervisor	- Discuss and solve problems	When there are problems that needs to be resolved immediately	Google Meet/Offline
Meeting with customers	All Member, customer	- Discuss about requirements	1 time a month	Google Meet/Offline

		- Update the project status to customer		

Table 9: Communication Plan

6. Configuration Management

6.1 Document Management

The project uses Google Docs and Google Sheets as our primary tools for sharing, editing, and version control of our documents. It allows us to observe what has changed in the documents and who is responsible for that modification.

The project has a folder on Google Drive named Documents. It is shared between members and supervisors.

The folder includes following files:

- Report1_Project Introduction
- Report2_Project Management Plan
- Report3_Software Requirement Specification
- Report4_Software Design Document
- Report5.0_Test Documentation
- Report5.1_Unit Test
- Report5.2_Integration Test
- Report6_Software User Guides
- Report7_Final Project Report

6.2 Source Code Management

The project uses Gitlab for managing source code because It makes it easy for developers to share code files and collaborate with fellow developers on open-source projects. GitHub also serves as a social networking site where developers can openly network, collaborate, and pitch their work.

The project has 2 Gitlab repositories: SCCMS_API and SCCMS_WebApp

6.3 Tools & Infrastructures

Category	Tools / Infrastructure

Technology	React 18(FrontEnd), ASP.NET 8 (BackEnd), SignalR (notification)
Database	SQL Server
IDEs/Editors	Visual Studio Code, Visual Studio
Diagramming	DrawIO, Lucid Chart, Astah Professional, Figma, Visual Paradigm
Documentation	Ms Office, Google Docs/Sheets/Slides
Version Control	GitLab (Source Codes), Google Drive (Documents)
Deployment server	Azure Web Service
Project management	Google Sheet (Schedule, Tasks, Defects), GitLab (Issue Management)

Table 10: Tools and Infrastructures