Capstone Project Final Report

FPTU CAPSTONE PROJECT MANAGEMENT

	FCPM Team
Group Members	Do Dinh Duc – SE04936 Bui Trung Kien – SE05182 Nguyen Ngoc Khanh - SE05045 Nguyen Duc Long – SE05906 Nguyen Minh Khang – SE05904
Supervisor	Nghiem Thi Lan Phuong
Capstone Project code	FCPM

Hanoi, Sep 1st, 2020

Table of Contents

Acknowledgements 5
DEFINITIONS AND ACRONYMS 6
CHAPTER 1: INTRODUCTION 7
1.1 Purpose 7
1.2 Project Information 7
1.3 The people 7
1.4 Background 8
1.5 The Existing System 9
1.6 Proposal of the system 10
1.6.1 System functions 10
1.6.2 Out-of-scope functions 11
1.6.3 Special approaches 11
CHAPTER 2: SOFTWARE PROJECT MANAGEMENT PLAN (SPMP) 12
2.1 Purpose 12
2.1.1 Software processing model 12
2.1.2 Roles and Responsibilities 13
2.2 PROJECT MANAGEMENT PLAN 15
2.2.1 Estimated Tasks 15
2.2.2 Meeting sessions 16
2.2.3 Coding Convention 17
2.2.4 Risk Management Plan 17
2.2.5 Communication Plan 18
CHAPTER 3: SOFTWARE REQUIREMENTS SPECIFICATION (SRS) 19
3.1 Purpose 19
3.2 Functional Requirements 19
3.2.1 Use Case Diagram 19
3.2.2 Business Rules 21

3.2.3 Use Cases 21
3.2.4 Use Cases Detail 25
3.3 Non-functional Requirement 54
3.3.1 Security 54
3.3.2 Supportability 55
3.3.3 Accuracy 55
3.3.4 Maintainability and Extensibility 55
3.3.5 Performance 55
3.3.6 Usability 55
3.4 Entity Relationship Diagram 56
CHAPTER 4: SOFTWARE DESGIN 57
4.1 Purpose 57
4.2 Architectural overview 57
4.2.1 System Architecture 57
4.2.2 System Architectural Explanation 58
4.3 Design of FPT Project Management System 61
4.3.1 Architectural Layers Design 61
4.3.2 Database Design 63
4.4 SEQUENCE DIAGRAM 74
4.4.1: Description of the Sequence Diagram 74
4.5. User Interface Design 154
4.5.1 Description of the User Interface 154
5. Software Test Documentation 235
5.1. Introduction 235
5.1.1 Purpose 235
5.1.2 Scope of testing 235
5.2. Test Plan 236
5.2.1 Testing tools and Environment 236
5.2.2 Resources and Responsibilities 237
5.2.3 Test Strategy 238
5.3. Test Approach 241
5.3.1 Unit test 241
5.3.2 Integration and System test 241

5.4. Test Report 242
5.4.1 Unit test 242
5.4.2 System test 244
6. Software User's Manual 247
6.1. Installation Guide 247
6.2. User's Guide 247
6.2.1 General function 247

Acknowledgements

We wish to express our deepest gratitude to our supervisor Mrs. Nghiem Thi Lan Phuong for her continuous sharing and motivating throughout the project. Under the instructions of Mrs. Phuong, the project team always felt comfortable and confident. Apart from all the technologies and methodologies, Mrs. Phuong shared with us the philosophy of keeping the right attitude towards problems. We believe that was the most important factor that led to the success of this project.

We would also like to thank the instructors at FPT University for all the classes. We hope you would find this project as a reflection of the knowledge and experiences you have given us during this period of four years.

Finally, we truly appreciate the random fellows on the internet who has discussed with us.

Without them, it would have likely been a real struggle to solve every tiny problem.

DEFINITIONS AND ACRONYMS

Terminologies

No	Definition	Explanation
1	**FCPM**	FPTU Capstone Project Management website.
2	**FCPM**	A stand-alone application that manages students' system in the 9th semester of the school.
3	**FCPM Store**	FPTU Capstone Project Management Website Store
4	**FCPM Console**	FPTU Capstone Project Management Website Console
5	**FCPM Developer**	People who develops FPTU Capstone Project Management for web applications.
6	**FCPM User**	People who need FPTU Capstone Project Management Website for their projects
7	**User**	The term includes FCPM Developer and FCPM User

Acronyms

Acronym	Definition
CPU	Central Processing Unit
RDBMS	Relational Database Management System
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment
UML	Unified Modelling Language
SRS	Software Requirement Specification
UC	Use Case
UI	User Interface
UX	User Experience
URL	Uniform Resource Locator
FCPM	FPT Capstone Project Management

CHAPTER 1: INTRODUCTION

1.1 Purpose

This chapter provides an overview of the project including background information, a literature review of the existing system and raising a proposal for ideas of improvement.

1.2 Project Information

- Project name: FPTU Capstone Project Management
- Project code: **FCPM**
- Project group name: **FCPM**
- Product type: **Web Application**
- Timeline: **Sep 1st 2020**

1.3 The people

NO	NAME	ROLES	E-MAIL	PHONE
1	*Nghiem Thi Lan Phuong*	Supervisor	phuongntl@fpt.edu.vn	0989376681
2	*Do Dinh Duc*	Leader	ducddse04936@fpt.edu.vn	0968480997
3	*Bui Trung Kien*	Member	kienbtse05182@fpt.edu.vn	0983268048
4	*Nguyen Ngoc Khanh*	Member	khanhnnse05045@fpt.edu.vn	0914092466
5	*Nguyen Duc Long*	Member	longndse05906@fpt.edu.vn	0978299648
6	*Nguyen Minh Khang*	Member	khangnmse05904@fpt.edu.vn	0971091547

1.4 Background

Each student studying at FPT University has a 4-year path to complete subjects according to the school's schedule. Each student must complete a graduation thesis as the final step to graduation. This made us think of a project of creating a software that provides students with functions such as registering, discussing, completing deadlines, exchanging with teachers ... With the abilities of reducing cost and time, it would be advantageous.

Figure 1-1: FPT University

1.5 The Existing System

As far as we know, most universities in Vietnam still manage projects manually.

Currently, FPT University has a number of websites for students to easily track their study progress and register their graduation project such as http://fap.fpt.edu.vn/ to help students keep track of everything. subject information, exam schedule, score report, ...

As far as we know, the page http://fap.fpt.edu.vn/ only has the project registration feature, not the project management feature for the training room. So we had the idea to create a project management system to help students, supervisors, training department and head of department easily and most convenient to use.

1.6 Proposal of the system

1.6.1 System functions

- Allow the student to:

  - ✓ Allow the student to login with FPT email accounts.
  - ✓ Allow the student to discuss and find teammates for capstone project in forum.
  - ✓ Allow the student to communicate via chat feature.
  - ✓ Allow the student to view and research project management models.
  - ✓ Allow the student to register projects.
  - ✓ Allow the student to view information and choose supervisors.
  - ✓ Allow the student to edit the capstone project's name.
  - ✓ Allow the student to report the working process of the group.
  - ✓ Allow the student to report following Jira's task template.
  - ✓ Allow the student to view news and notifications.

- Allow supervisor to:

  - ✓ Allow the supervisor to login with FPT email accounts.
  - ✓ Allow the supervisor to discuss and gather team's members for capstone project in forum.
  - ✓ Allow the supervisor to communicate via chat feature.
  - ✓ Allow the supervisor to view the list of registrants.
  - ✓ Allow the supervisor to confirm / reject the project.
  - ✓ Allow the supervisor to confirm / reject the request of changing project name.
  - ✓ Allow the supervisor to view the group's working process.
  - ✓ Allow the supervisor to view and comment on reports.

- Allow the Training Department to:

  - ✓ Allow the training department to login with FPT email accounts.
  - ✓ Allow the training department to view capstone project in forum.
  - ✓ Allow the training department to communicate via chat feature.
  - ✓ Allow the training department to view the list of registrants.
  - ✓ Allow the training department to manage capstone projects.

- ✓ Allows the training department to confirm / reject the request of changing project name.
- ✓ Allow the training department to manage users.

- ➢ Allow the Head of Department to:

  - ✓ Allow the head of department to login with FPT email account.
  - ✓ Allow the head of department to view capstone projects in forum.
  - ✓ Allow the head of department to communicate via chat feature.
  - ✓ Allow the head of department to confirm / reject the project.
  - ✓ Allow the head of department to view dashboard of all capstone projects.
  - ✓ Allow the head of department to view process of working in group.

1.6.2 Out-of-scope functions

Because of the time limitation, we did not implement these following functions in the FCPM initial version 1.0.0 release of the project but they are not permanently excluded. We are aware of their importance and they will be developed in future version 2.0.0:

- FCPM adds functions for affairs department.
- FCPM adds functions for examination council.
- FCPM completes functions of assessing the process of project.
- FCPM integrates API of project management systems.

Finally, we used some new approaches to solve the stated problems. Having awareness of the importance of new technologies, we aimed at applying them to our project.

1.6.3 Special approaches

- Using Spring boot for business development.
- Using Spring data JPA.
- Using IntelliJ IDEA 2020.2.3.
- Using MySQL Workbench 8.0 CE to manage data.
- Using HTML, CSS, Bootstrap to create frontend.
- Using MS Excel create templates for imported data.

CHAPTER 2: SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)

2.1 Purpose

- Helping readers understand the organization and activities of the project and making it easy for readers to visualize how the project is going.
- Setting out the principles that the team members must adhere to and strictly follow throughout the project.
- Each team member must be conscious of the team project, so all team members must use this document to clarify their responsibilities.

2.1.1 Software processing model

Figure: Iterative development model

The project used the Iterative Process Model

The Iterative Process 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 processing model, we broke down the developing system tasks into series of smaller tasks which were completed separately, allowing us to take advantage of what was learned during the development of earlier parts of the system. In addition, the iterative model is easier than other models when the issues are discovered. They are reported back to the team, and solutions will be found while the project is still in development.

2.1.2 Roles and Responsibilities

2.1.2.1 Organization Structure

Role	Responsibility
Project Manager	Planning, dividing works among team members, developing schedules, supporting team members, coordinating communication with team members, maintaining a comfortable and proficient working atmosphere, and keeping the team focused on working for the main goal.
Technical Leader	Choosing and deciding what technologies should be used, instruct and support team members in using technologies they may not be familiar with, overseeing the work being done by other developers
Test Leader	Test execution, including test set-up and test run, evaluating test runs, recovering errors, defect loggings, and recording test results.
Developer	Code the product, discuss, exchange and review code of other developers.
Designer	Design product's user interface
Tester	Test the product

Table 2-1: Project Structure

2.1.2.2 Project Team Members

Team Member	Role
DucDDSE04936	Project Manager, Technical Leader, Developer, Designer
KienBTSE05182	Test Leader, Developer, Designer, Tester
KhanhNNSE05045	Developer, Designer, Tester
LongNDSE05906	Developer, Designer, Tester
KhangNMSE05904	Developer, Designer, Tester

Table: Project Team Members

2.1.2.3 Tools and Techniques

Software Tool	Version	Description
Jira	Online	Management tools
GitHub	Online	Source code control
Source Tree	3.1.3	Source code control
Microsoft Words	2019	Document tools
Microsoft Excel	2019	Document tools
MySQL Workbench 8.0 CE	2019	My SQL
Facebook	Online	Communication tool
Skype	Online	Communication tool
Process Model		Iterative Model
Draw.io	Online	Diagram, UML tool, chart
Spring Boot	2.3.3	IDE for coding
Spring Data JPA	2.0.3	IDE for coding
Hibernate		IDE for coding
Visual Studio Code	1.34.0	IDE for coding
HTML		IDE for coding
CSS		IDE for coding
Bootstrap		IDE for coding
JS, JQuery		IDE for coding
IntelliJ IDEA	2020.2.3	IDE for coding

2.2 PROJECT MANAGEMENT PLAN

2.2.1 Estimated Tasks

ICON	KEY	SUMMARY	STATUS	TIME TRACKING (SPENT)	TIME SPENT	START DATE	END DATE
	FCPM-5	Start capstone project	DONE	0%	0d	29/Sep/20	25/Oct/20
	FCPM-6	In the process	DONE	0%	0d	26/Oct/20	23/Nov/20
	2	FCPM Sprint 1					
	FCPM-17	Comment	DONE	100%	2d	02/Oct/20	04/Oct/20
	3	FCPM Sprint 2					
	FCPM-20	Search and filter forum	DONE	100%	4d	05/Oct/20	08/Oct/20
	FCPM-21	Create common upload file	DONE	100%	1m	05/Oct/20	08/Oct/20
	FCPM-22	Create common send email	DONE	100%	3m	09/Oct/20	11/Oct/20
	FCPM-23	Create notification component	DONE	100%	1w 2d	05/Oct/20	11/Oct/20
	FCPM-25	Create booking lecturers feature	DONE	100%	1w 2d	05/Oct/20	11/Oct/20
	FCPM-24	Create capstone project detail page	DONE	100%	1w 2d	05/Oct/20	11/Oct/20
	FCPM-13	Register project to admin	DONE	100%	2d	29/Sep/20	04/Oct/20
	FCPM-12	Create login with google account	DONE	100%	4d	29/Sep/20	04/Oct/20
	FCPM-15	View detail information of user	DONE	100%	1w 1d	29/Sep/20	04/Oct/20
	FCPM-14	Show list lecturers and booking	DONE	100%	1w 1d	29/Sep/20	04/Oct/20
	FCPM-16	Post	DONE	100%	4d	29/Sep/20	01/Oct/20
	7	FCPM Sprint 3					
	FCPM-27	Append send email to register project	DONE	100%	1w 2d	12/Oct/20	18/Oct/20
	FCPM-28	Append send email to booking lecturers	DONE	100%	1w 2d	12/Oct/20	18/Oct/20
	FCPM-29	Append upload file to forum	DONE	100%	1w 1d	12/Oct/20	18/Oct/20
	FCPM-34	Append upload file to register project	DONE	100%	1w 2d	12/Oct/20	18/Oct/20
	FCPM-26	Create confirm join team feature	DONE	100%	1w 2d	12/Oct/20	18/Oct/20
	FCPM-18	Create group for chatting	DONE	100%	1w 3d	09/Oct/20	11/Oct/20
	8	FCPM Sprint 4					
	FCPM-30	Trainning department view infomation, status of list student	DONE	100%	1w 2d	19/Oct/20	25/Oct/20
	FCPM-32	Trainning department request information to admin	DONE	100%	1w 2d	19/Oct/20	25/Oct/20
	FCPM-31	Trainning department view and edit the list of registered projects	DONE	100%	1w 2d	19/Oct/20	25/Oct/20
	FCPM-35	Admin view list capstone project	DONE	100%	1w 2d	19/Oct/20	25/Oct/20
	FCPM-36	Admin accept or reject project	DONE	100%	1w 2d	19/Oct/20	25/Oct/20
	9	FCPM Sprint 5					
	FCPM-38	Daily report of member to leader	DONE	100%	2w 4d	26/Oct/20	08/Nov/20
	FCPM-39	Weekly report of leader to lecturer	DONE	100%	2w 4d	26/Oct/20	08/Nov/20
	10	FCPM Sprint 6					
	FCPM-47	Calculate data with the import file for statistics	DONE	100%	3w	09/Nov/20	22/Nov/20
	FCPM-40	Dashboard all project for admin	DONE	100%	2w 4d	09/Nov/20	22/Nov/20
	FCPM-41	Detail project for admin	DONE	100%	2w	09/Nov/20	22/Nov/20

2.2.2 Meeting sessions

All meeting sessions were be written following this template:

Meeting/Project Name:	FCPM		
Date of meeting:	25/8/2020	*Time: (Type)*	*1 hours (Face-to-face)*
Meeting Called by:	DucDD	***Location:***	*FPT University – P205*
Note Taker:	KhangNM	***Time Keeper:***	DucDD
1. Meeting Objectives 1. Assigning new tasks for members 2. Conducting report No.2			
2. Attendance			
Name	***Roles***	***E-mail***	***Phone***
Do Dinh Duc	Project Manager	Ducddse04936@fpt.edu.vn	0968480997
Bui Trung Kien	Developer ,Tester Leader, Doc	Kienbtse05182@fpt.edu.vn	0986268048
Nguyen Ngoc Khanh	Developer ,Tester, Doc	Khanhnnse05045@fpt.edu.vn	0914092466
Nguyen Minh Khang	Developer ,Tester, Doc	Khangnmse05904@fpt.edu.vn	0971091547
Nguyen Duc Long	Developer ,Tester, Doc	Longndse05906@fpt.edu.vn	09782996448
3. Done tasks 1. Building project structure. 2. Configuring git repository with security. 3. Conducting report No.1			
4. New tasks 1. Backend 2. Frontend			
5. Risks & Difficulties			

2.2.3 Coding Convention

Sonar Lint tool for checking convention of Java

2.2.4 Risk Management Plan

No	Description	Avoidance plan	Contingency plan	Status
R1	Data loss	Using GitHub for version control. - Teaching members how to use Git and resolving conflicts. - Always have important backups.	- Restoring backed up data from GitHub.	Closed
R2	Illness or absence of team members	- Providing schedules in advance. - For long periods of absence, members should notify the group in advance.	- Assigning the tasks of absent member to other members. - Working overtime if necessary.	Closed
R3	Misunderstanding of Requirements	- Discussing requirements carefully with the customer. - Always ask for clarification if requirements are unclear. - Commenting needs to meet	-Making sure idea's business logic is carefully analysed.	Closed
R4	Changes in requirements	- Every new update of requirement has to be reviewed by all team members and supervisor. - Team member has to analyse requirements carefully before raising up to team.	- If requirement has new update, all members have to join the meeting to be aware and to make decisions.	Closed
R5	Conflicts between team members	- Everything must be documented. - Every team member has to express clearly and carefully.	- Making sure any miscommunication has to be resolved.	Closed
R6	Failed to meet deadline	- Planning and developing schedule carefully. - Assigning tasks carefully. - Defining punishment for team members who neglect	- Finding the root cause of the problem. - Reassigning tasks. - Focusing on important functions first.	Closed

		work.		

Table 2-5: Risk Management

2.2.5 Communication Plan

Weekly meeting schedule: We used Iterative and Incremental Process Model, then we divided the system into two sub-systems, each sub-system was divided into series of small tasks. The amount of work by the whole team, after that the task will be assigned to team members by the Team Leader, and depending on difficulty levels, the Technical Leader set deadlines for each task. We had a meeting inform to all team about what each member finished last week, the status (fast, on time or slow), the issues met and how to solve them. When any member raised any issue, the whole team helped to find out a solution together. After that, the team defined detailed stories for next week tasks.

Daily meeting schedule: Each sub-system had one development team with different schedule. Starting each workday, each team had a stand-up meeting to inform to others: "What did I do yesterday?", "What will I do today?" and "Is there any difficulty?". By focusing on what each person accomplished the day before and will accomplish on that day, the team gained an excellent understanding of what had been done and what remained.

Unscheduled meeting: When someone had an important problem that he wanted to solve immediately, we had a meeting for discussion, usually via some online channels: Skype, Facebook or Phone.

Communication channel: Our main communication channel was Skype. On the other hand, we used face-to-face meetings, Facebook group to comment issues. However, we sometimes made phone calls or instant messages if someone had a problem.

CHAPTER 3: SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

3.1 Purpose

Software Requirements Specifications chapter provides general, as well as specific information of the system, it outlines functional and non-functional requirements of our website. Following these requirements, FCPM performs its intended functionality used in designing developing and testing. All members work based on the information that provided in this document.

3.2 Functional Requirements

3.2.1 Use Case Diagram

The figure below is displayed the use cases we had in this project.

Figure: Use case diagram

3.2.2 Business Rules

No	Description
B1	Username must not be empty.
B2	Username must be unique.
B3	User's full name must not be empty.
B4	The email must not be empty.
B5	Email address must be valid.
B6	Each email address can be used for only one account.
B7	Capstone Project must have sites and semesters
B8	Phone number must be valid.
B9	The administrator is able to: active/ deactivate account, change account role, change request state.
B10	Only users whose role is: Head of department, Supervisor view statistical dashboard of the screen.
B11	The administrator is able to: active/ deactivate account, change account role, change request state of student, supervisor.
B12	Exported files must be Microsoft Excel files with valid formats.
B13	Only leader can be created weekly report.
B14	Only student can create reports.
B15	Only training department add news.
B16	Only training department can manage users.
B17	Users must have sites and semesters

Table: Business rules

3.2.3 Use Cases

Actor	Description
Student	FPT students who have qualified for the project, are preparing to do a project can easily find their project group and register for the project in the fastest and most convenient way. Students will know the information of instructors to choose the right supervisor for the group. And in the process of making projects, students can also exchange and report on weekly work with supervisors.
Supervisor	The supervisor looks at the project teams of his choice and makes a decision to accept or reject the instruction. During the project, the supervisor can see the daily and weekly reports sent by the leader of team, and the supervisor will decide whether the project team is protected or not.
Training department	The training department will view the project registration list and view information and status of the project group. The training department will decide what the group will do and the supervisor will guide the group.

	Training department will synthesizes the results and decide the success or failure of the project team
Head of department	The Head of Department will see all of: the information, the project team list and the details of the team's project making process.

Table: Actor description

	Actor	Name	Description
UC-001	Student, Supervisors, Training department, Head of department	Login	Login to system with FCPM account.
UC-002	Student, Supervisors, Training department, Head of department	Logout	Logout to system with FCPM account
UC-003	Student	Register Capstone Project	Students register capstone project to system
UC-004	Student	Add member	Students add members to register project
UC-005	Student	Research model	Students research models for reference
UC-006	Student	Show list supervisors and booking	Students choose supervisor of their group
UC-007	Student	View GUIDE	View GUIDE to reference
UC-008	Student	Upload file	Students upload file to register projects or add posts
UC-009	Student	Send mail	Students send mail to register project and booking supervisor
UC-010	Student, Supervisor, Training department, Head of department	Chat	Users chat together
UC-011	Student	Create chat	Students create group for chatting and invite other student who do not have project group to talk, exchange.
UC-012	Student	Confirm join team	After talking through a group chat,

		feature	Students will agree to form a group to register project
UC-013	Student	Edit Profile	Students can edit profile
UC-014	Student, Supervisors, Training department, Head of department	Notification component	When there is any information, there will be a notice to the student, supervisors, training department, head of department
UC-015	Student, Supervisors, Training department, Head of department	Comment	Students, supervisors, training department, head of department can comment, exchange information with each other
UC-016	Student, Supervisors	Daily report to member to leader	Every day, members who complete tasks must report to leader
UC-017	Leader	Weekly report to leader to supervisor	Every week, the leader must report to supervisor during the past week, team has performed
UC-018	Training department	Manage User	Training department can manage all information of user
UC-019	Training department	View information, status of list student	Training department can view information of list student and view status of students who are eligible to do projects or students don not have groups to add to another group
UC-020	Training department	Add student and edit supervisor of Capstone Project	Training department can add student and edit supervisor of Capstone Project
UC-021	Head of department, Supervisor	View list capstone project	Allow students, supervisor see the list capstone project
UC-022	Head of department, Supervisor	View detail project	Head of department, supervisors can view detail project
UC-023	Head of department, Supervisor	Manage Capstone Project	Head of department and supervisors can manage capstone project
UC-024	Supervisors, Head of department	View task detail of Capstone Project	Supervisors and head of department can view task detail of Capstone Project
UC-025	Supervisors, Head of department	Dashboard statistical of project	Supervisors and head of department will dashboard statistical all information project of registered groups
UC-026	Student, Supervisor, Head	View information of user	Students, supervisors, head of department and training department

	of department, Training department		can view information of users
UC-027	Head of department	Approve capstone project in the project registration stage	Head of department can approve capstone project in the project registration stage
UC-028	Training department	Change pass / fail results	Training department can Change pass / fail results capstone project
UC-029	Training department	Create notification	Training department can create notification
UC-030	Training department	View history record	Training department can view history record
UC-031	Student	Change name capstone project	Student can change name capstone project
UC-032	Student, Supervisor, Head of department, Training department	View List Post	Student, Supervisor, Head of department, Training department view list post
UC-033	Student, Supervisor, Head of department, Training department	Get comment	Student, Supervisor, Head of department, Training department get comment
UC-034	Student, Supervisor, Head of department, Training department	Add post	Student, Supervisor, Head of department, Training department add post
UC-035	Student, Supervisor, Head of department, Training department	Edit post	Student, Supervisor, Head of department, Training department edit post
UC-036	Student, Supervisor, Head of department, Training department	Delete post	Student, Supervisor, Head of department, Training department delete post
UC-037	Student, Supervisor, Head of department, Training department	Search post	Student, Supervisor, Head of department, Training department search post
UC-038	Training	Add user	Training department add user

	department		
UC-039	Training department	Approve capstone project in the changing name stage	Training department approve capstone project in the changing name stage

Table: Use Case List

3.2.4 Use Cases Detail

UC-001. Login

Use Case Specification

UC ID and Name:	**UC- Log in**		
Create by:	DucDD	Date Created:	10/08/2020
Primary Actor:	Students, Supervisors, Training department, Head of department	Secondary Actor:	
Trigger:	N/A		
Description:	Allow students, supervisors, training department, head of department who have FCPM account to login.		
Preconditions:	Students supervisors, training department, head of department account has been set up and username and password has been generated.		
Post conditions:	Students, supervisors, training department, head of department are logged into FCPM successfully. System displays Login Page.		
Normal Flow:	**Login:** 1. Student visits the login page. 2. Student click " Sign in with Google " 3. Systems displays account verification 4. Student choose account FPT 5. System displays forum		
Alternative Flows:	Students violates some business rule in step 4 1. Students choose a non-FPT account 3. Go to step 1 in normal flow.		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			

Other Information:	N/A
Assumptions:	N/A

UC-002. Logout

Use Case Specification

UC ID and Name:	**UC- Logout**		
Create by:	KienBT	Date Created:	10/08/2020
Primary Actor:	Students, Supervisors, Training department, Head of department	Secondary Actor:	
Trigger:	N/A		
Description:	To prevent other users from stealing information		
Preconditions:	User has logged into FCPM system		
Post conditions:	Students, supervisors, training department, head of department are logged into FCPM successfully. System displays forum.		
Normal Flow:	**Logout**: 1.Students successfully login to the forum 2.Student select their username in the top right corner of the forum 3.Student click "Logout" 4.System display Login Page		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-003. Register Capstone Project

Use Case Specification

UC ID and Name:	**UC- Register Capstone Project**

Create by:	KhanhNN	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Student register capstone project to system		
Preconditions:	The student has formed his project team		
Post conditions:	1.The student has successfully formed his project team 2.Students login to their account successfully 3.The system displays forum		
Normal Flow:	**Register**: 1.Students login to their account successfully 2.Students access the forum 3.Students click on " Capstone Project " and choose " Create Capstone " 4.System display Register Project Page 5.Students register project and click “submit” to register		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-004. Add Member

Use Case Specification

UC ID and Name:	**UC- Add member**		
Create by:	LongND	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Student add member to register capstone project		
Preconditions:	The student has formed his project team		
Post conditions:	1.The student has successfully formed his project team 2.Students login to their account successfully 3.The system displays forum		
Normal Flow:	**Add member:** 1.Students login to their account successfully 2.Students access the forum 3.Students click on " Capstone Project " and choose " Create Capstone " 4.System display Register Project Page 5.Student add account member of team 6.Student click button “Add”		
Alternative Flows:	N/A		

Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-005. Research model

Use Case Specification

UC ID and Name:	**UC- Research model**		
Create by:	KhangNM	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Student research model for reference		
Preconditions:	The student has formed his project team		
Post conditions:	1.The student has successfully formed his project team 2.Students login to their account successfully 3.The system displays forum		
Normal Flow:	**Research model:** 1.Students login to their account successfully 2.Students access the forum 3.Students click on "Model" to research model		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-006. Show list supervisor and booking

Use Case Specification

UC ID and Name:	**UC- Show list supervisor and booking**

Create by:	LongND	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Student choose instructors of their group		
Preconditions:	Students have registered for the project		
Post conditions:	Students found the details of the supervisors		
Normal Flow:	**Booking supervisor:** 1. Students login to their account successfully 2. Students click on “Capstone Project” at the top of the page, Choose supervisors 3. System display a list supervisors’ page 4. Students click on name of supervisor to show information of supervisor to booking		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-007. View GUIDE

Use Case Specification

UC ID and Name:	**UC- View GUIDE**		
Create by:	DucDD	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Student view GUIDE to reference		
Preconditions:	Students have registered for the project		
Post conditions:	Students found the details of the supervisors		
Normal Flow:	**View GUIDE:** 1.Students login to their account successfully 2.Students access the forum 3.Students click on "GUIDE" to research GUIDE		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		

Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-008. Upload file

Use Case Specification

UC ID and Name:	**UC- Upload file**		
Create by:	KhangNM	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Student upload file to register project or add post		
Preconditions:	Leader gathered enough information about the group members to complete the word file Leader complete the word file to upload		
Post conditions:	1.Students login to their account successfully 2.Leader gathered enough information about the group members to complete the word file to upload 3.The system displays the Register Project page to upload file		
Normal Flow:	**Upload file:** 1.Students login to their account successfully 2.Students access the forum 3.Students click on " Capstone Project " and choose " Create Capstone " 4.System display Register Project Page to upload file 5. Students click on “ Upload” to upload file		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-009. Send mail

Use Case Specification

UC ID and Name:	**UC- Send mail**		
Create by:	KhanhNN	Date Created:	10/08/2020

Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Students send mail to register capstone project and booking supervisor		
Preconditions:	Group has enough to member		
Post conditions:	1.The group has all the information of its members 2.The group had the project's topic		
Normal Flow:	**Send mail:** 1.Student access to email page 2.Student create email 3.Student enter information of team 4.Student send mail to register capstone project and booking supervisor to training department		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-010. Chat

Use Case Specification

UC ID and Name:	**UC- Chat**		
Create by:	KienBT	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor, Training department, Head of department	Secondary Actor:	
Trigger:	N/A		
Description:	User chat together		
Preconditions:	User login successfully		
Post conditions:	User access successfully to forum		
Normal Flow:	**Chat:** 1. Users login to their account successfully 2. Users access to forum 3. Users click button “Messenger” to chat		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server		

	1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-011. Create chat

Use Case Specification

UC ID and Name:	**UC- Create chat**		
Create by:	DucDD	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Students Create chat and invite other student who do not have project group to talk, exchange		
Preconditions:	Students have not found their own project group		
Post conditions:	Project group does not have enough members		
Normal Flow:	**Create chat:** 1.Students login to their account successfully 2.Students access the forum page to post 3.Student click icon on screen 4.Show room to student chat		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-012. Confirm join team feature

Use Case Specification

UC ID and Name:	**UC- Confirm join team feature**		
Create by:	KhangNM	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		

Description:	After talking through a group chat, Students will agree to form a group to register project
Preconditions:	Students have not found their own project group
Post conditions:	Project group not enough members
Normal Flow:	**Confirm join team feature:** -Students login to their account successfully -Students access the forum page to post -Students Create chat - If the student is interested, press the join button, it will automatically add to the chat group - After the group chat members feel ok, the students in that group will form a group to register the project group.
Alternative Flows:	N/A
Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-013. Edit Profile

Use Case Specification

UC ID and Name:	**UC- Edit Profile**		
Create by:	KienBT	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Student can edit profile		
Preconditions:	Student login to their account successfully		
Post conditions:	Student access successfully to forum		
Normal Flow:	**Edit profile:** 1.Students login to their account successfully 2.Students access the forum 3.Student click to account 4.Student click button “Edit profile”		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		

Assumptions:	N/A

UC-014. Notification component

Use Case Specification

UC ID and Name:	**UC- Notification component**		
Create by:	LongND	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor, Training department, Head of department	Secondary Actor:	
Trigger:	N/A		
Description:	When there is any information, there will be a notice to the student, supervisors, training department, head of department		
Preconditions:	1. Users can access the system. 2. Users logins successfully		
Post conditions:	N/A		
Normal Flow:	**Notification component:** 1.Users login to their account successfully 2.Users access the forum page 3.Select the notifications you want to see 4.Displays details of the notifications you want to see		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-015. Comment

Use Case Specification

UC ID and Name:	**UC- Comment**		
Create by:	KhanhNN	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor	Secondary Actor:	

Trigger:	N/A
Description:	Student and supervisors can comment, exchange information with each other
Preconditions:	1. Student, supervisor can access the system. 2. Student, supervisor logins successfully
Post conditions:	N/A
Normal Flow:	**Comment:** 1.Students, supervisor login to their account successfully 2. Students, supervisor access the forum page 3. Student, supervisor can enter information you want comment, click send in post
Alternative Flows:	N/A
Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-016. Daily report to member to leader

Use Case Specification

UC ID and Name:	**UC- Daily report to member to leader**		
Create by:	DucDD	Date Created:	10/08/2020
Primary Actor:	Student	Secondary Actor:	
Trigger:	N/A		
Description:	Every day, members who complete tasks must report to leader		
Preconditions:	Members complete their task		
Post conditions:	N/A		
Normal Flow:	**Report**: 1.Students login to their account successfully 2.Students access the forum page 3. Select the Reports section at the top of the page, then select create report 4. Enter information report, then click button submit 5. Display reports of yourself and team members 6. Click name of title to view detail information of report 7. Display report detail		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		

Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-017. Weekly report to leader to supervisor

Use Case Specification

UC ID and Name:	**UC- Weekly report to leader to supervisor**		
Create by:	KienBT	Date Created:	10/08/2020
Primary Actor:	Leader	Secondary Actor:	
Trigger:	N/A		
Description:	Every week, the leader must report to supervisor during the past week, team has performed		
Preconditions:	Team complete their task		
Post conditions:	N/A		
Normal Flow:	**Report:** 1.Leader login to their account successfully 2.Leader access the forum page 3. Select the Reports section at the top of the page, then select create report 4. Enter information report, then click button submit 5. Display reports of yourself and team members 6. Click name of title to view detail information of report 7. Display report detail		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-018. Manage User

Use Case Specification

UC ID and Name:	**UC- Manage User**		
Create by:	LongND	Date Created:	10/08/2020
Primary Actor:	Training	Secondary Actor:	

	department		
Trigger:	N/A		
Description:	Training department can manage all information of users		
Preconditions:	Training department login to their account successfully		
Post conditions:	Training department access successfully the forum page		
Normal Flow:	**Manage User:** 1.Training department login to their account successfully 2.Training department access the forum page 3.Training department manage users 4. Show detail information of users		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-019. View information, status of list student

Use Case Specification

UC ID and Name:	**UC- View information, status of list student**		
Create by:	KhangNM	Date Created:	10/08/2020
Primary Actor:	Training department	Secondary Actor:	
Trigger:	N/A		
Description:	Training department can view information of list student and view status of students who are eligible to do projects or students don't have groups to add to another group		
Preconditions:	Training department has logged into FCPM website		
Post conditions:	"Registered project page" screen will be shown		
Normal Flow:	**View:** 1.Training department login to their account successfully 2.System display "Registered projects list" screen 3. The dropdown section containing the supervisors and student information will be displayed on the screen 4. Training department will select a group in the dropdown to see information, status of list student 5.Training department can switch status from registered _ capstone -> approve_ capstone _ supervisors ->		

	approve_ capstone_ head ->doing_ capstone
Alternative Flows:	N/A
Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-020. Add student and edit supervisor of Capstone Project

Use Case Specification

UC ID and Name:	**UC- Add student and edit supervisor of Capstone Project**		
Create by:	LongND	Date Created:	10/08/2020
Primary Actor:	Training department	Secondary Actor:	
Trigger:	N/A		
Description:	Training department can add student and edit supervisor of Capstone Project		
Preconditions:	Training department has logged into FCPM website		
Post conditions:	Training department access successfully to forum		
Normal Flow:	**View and edit:** 1.Training department login to their account successfully 2.System display forum 3. Training department click on Capstone Project 4. Training department choose Capstone Project List 5.Training department add student and edit supervisor of Capstone Project		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-021. View list capstone project

Use Case Specification

UC ID and Name:	**UC- View list capstone project**		
Create by:	DucDD	Date Created:	10/08/2020
Primary Actor:	Head of department, Supervisor	Secondary Actor:	
Trigger:	N/A		
Description:	Allow students, supervisor see the list capstone project		
Preconditions:	Students, supervisor have logged into FCPM website		
Post conditions:	"Capstone project list" screen will be shown		
Normal Flow:	**View:** 1.Head of department, supervisor access in FCPM website. 2.System display "List capstone project" screen 3. Head of department, supervisor click on Capstone Project 4. Head of department, supervisor choose List Capstone Project		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-022. View detail Project

Use Case Specification

UC ID and Name:	**UC- View detail Project**		
Create by:	KhangNM	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor	Secondary Actor:	
Trigger:	N/A		
Description:	Head of department, supervisor can view detail project		

Preconditions:	Head of department, supervisor have logged into FCPM website.
Post conditions:	Head of department, supervisor access successfully to forum
Normal Flow:	**View detail:** 1. Head of department, supervisor logins to their account successfully 2. Head of department, supervisors access the forum page 3. Head of department, supervisors click on Capstone Project 4. Head of department, supervisors choose Capstone Project List 5. Head of department, supervisors click on one capstone project to view detail project
Alternative Flows:	N/A
Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-023. Manage Capstone Project

Use Case Specification

UC ID and Name:	**UC- Manage Capstone Project**		
Create by:	KienBT	Date Created:	10/08/2020
Primary Actor:	Head of department, Supervisor	Secondary Actor:	
Trigger:	N/A		
Description:	Head of department, Supervisors can manage capstone project		
Preconditions:	Head of department, Supervisors login to their account successfully		
Post conditions:	Head of department, Supervisors access successfully the forum page		
Normal Flow:	**Manage User:** 1.Training department login to their account successfully 2.Training department access the forum page 3. Head of department, Supervisors manage capstone project		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		

Assumptions:	N/A

UC-024. View task detail of Capstone Project

Use Case Specification

UC ID and Name:	**UC- View detail Project**		
Create by:	KhanhNN	Date Created:	10/08/2020
Primary Actor:	Head of department, Supervisor	Secondary Actor:	
Trigger:	N/A		
Description:	Head of department, supervisors can view task detail of Capstone Project		
Preconditions:	Head of department, supervisors have logged into FCPM website.		
Post conditions:	Head of department, supervisors access successfully to forum		
Normal Flow:	**View detail:** 1. Head of department, supervisor logins to their account successfully 2. Head of department, supervisors access the forum page 3. Head of department, supervisors click on Capstone Project 4. Head of department, supervisors choose Capstone Project List 5. Head of department, supervisors click on one capstone project to view task detail of capstone project		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-025. Dashboard statistical of project

Use Case Specification

UC ID and Name:	***UC- Dashboard statistical of project***		
Create by:	KienBT	Date Created:	10/08/2020
Primary Actor:	Supervisor, Head of department	Secondary Actor:	
Trigger:	N/A		
Description:	Supervisors and head of department will dashboard statistical all		

	information project of registered groups
Preconditions:	1.Supervisors, head of department can access the system. 2.Supervisors, head of department login successfully
Post conditions:	N/A
Normal Flow:	**Dashboard statistical:** 1.Supervisors, head of department access in FCPM website. 2.System display "Home page" screen 3.Supervisors, head of department clicks on "Dashboard" 4.System displays dashboard statistical of project
Alternative Flows:	N/A
Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-026. View information of user

Use Case Specification

UC ID and Name:	**UC- View information of user**		
Create by:	KhangNM	Date Created:	10/08/2020
Primary Actor:	Student, Training department, Head of department, Supervisor	Secondary Actor:	
Trigger:	N/A		
Description:	Students, Training department, Head of department, Supervisors can view information of user		
Preconditions:	Students, Training department, Head of department, Supervisors have logged into FCPM website		
Post conditions:	Students, Training department, Head of department, Supervisors access successfully to forum		
Normal Flow:	**View:** 1. Students, Training department, Head of department, Supervisors login successfully 2. Students, Training department, Head of department, Supervisors access successfully to forum 3. Students, Training department, Head of department, Supervisors click to name of user 4. System display information of users		
Alternative Flows:	N/A		

Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-027. Approve capstone project in the project registration stage

Use Case Specification

UC ID and Name:	**UC- Approve capstone project in the project registration stage**		
Create by:	LongND	Date Created:	10/08/2020
Primary Actor:	Head of department	Secondary Actor:	
Trigger:	N/A		
Description:	Head of department can Approve capstone project in the project registration stage of students		
Preconditions:	Head of department has logged into FCPM website		
Post conditions:	Head of department has logged successfully into FCPM website		
Normal Flow:	**Approve capstone project in the project registration stage:** 1. Head of department login successfully to system 2. Head of department access successfully to forum 3. Head of department click on capstone project 4. Head of department choose list capstone project 5. System display capstone projects list 6. Head of department can click button “Accept” or “Reject” project		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-028. Change pass / fail results

Use Case Specification

UC ID and Name:	**UC- Change pass / fail results**		
Create by:	DucDD	Date Created:	10/08/2020
Primary Actor:	Training department	Secondary Actor:	
Trigger:	N/A		
Description:	Training department can Change pass / fail results capstone project		
Preconditions:	Training department has logged into FCPM website		
Post conditions:	Training department has logged successfully into FCPM website		
Normal Flow:	**Change pass / fail results**: 1. Training department login successfully to system 2. Training department access successfully to forum 3. Training department click on capstone project 4. Training department choose list capstone project 5. System display list of capstone project 6. Training department can click button "Update"		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-029. Create notification

Use Case Specification

UC ID and Name:	**UC- Create notification**		
Create by:	KienBT	Date Created:	10/08/2020
Primary Actor:	Training department	Secondary Actor:	
Trigger:	N/A		
Description:	Training department can create notification		
Preconditions:	Training department has logged into FCPM website		
Post conditions:	Training department has logged successfully into FCPM website		
Normal Flow:	**Create notification:** 1. Training department login successfully to system 2. Training department access successfully to forum 3. Training department click on Notification 4. Training department choose Create notification 5. System display list of Create notification page		
Alternative Flows:	N/A		

Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-030. View history record

Use Case Specification

UC ID and Name:	**UC- View history record**		
Create by:	KhanhNN	Date Created:	10/08/2020
Primary Actor:	Training department	Secondary Actor:	
Trigger:	N/A		
Description:	Training department can view history record		
Preconditions:	Training department has logged into FCPM website		
Post conditions:	Training department has logged successfully into FCPM website		
Normal Flow:	**View history record:** 1. Training department login successfully to system 2. Training department access successfully to forum 3. Training department click on History 4. System display list of History page		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-031. Change name capstone project

Use Case Specification

UC ID and Name:	**UC- Change name capstone project**		
Create by:	LongND	Date Created:	10/08/2020
Primary Actor:	Students	Secondary Actor:	
Trigger:	N/A		

Description:	Students can change name capstone project
Preconditions:	Students have logged into FCPM website
Post conditions:	Students have logged successfully into FCPM website
Normal Flow:	**Change name capstone project:** 1. Students login successfully to system 2. Students access successfully to forum 3. Students click on Capstone Project 4. Students choose My Capstone 5. Students click button "Change name"
Alternative Flows:	N/A
Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-032. View list post

Use Case Specification

UC ID and Name:	**UC- View list post**		
Create by:	DucDD	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor, Head of department, Training department	Secondary Actor:	
Trigger:	N/A		
Description:	Student, Supervisor, Head of department, Training department view list post		
Preconditions:	Student, Supervisor, Head of department, Training department have logged into FCPM website		
Post conditions:	Student, Supervisor, Head of department, Training department have logged successfully into FCPM website		
Normal Flow:	**View list post:** 1 Student, Supervisor, Head of department, Training department login successfully to system 2. Student, Supervisor, Head of department, Training department access successfully to forum 3. Student, Supervisor, Head of department, Training department click		

	on List Post
Alternative Flows:	N/A
Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-033. Get comment

Use Case Specification

UC ID and Name:	**UC- Get comment**		
Create by:	KienBT	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor, Head of department, Training department	Secondary Actor:	
Trigger:	N/A		
Description:	Student, Supervisor, Head of department, Training department get comment		
Preconditions:	Student, Supervisor, Head of department, Training department have logged into FCPM website		
Post conditions:	Student, Supervisor, Head of department, Training department have logged successfully into FCPM website		
Normal Flow:	**Get comment:** 1 Student, Supervisor, Head of department, Training department login successfully to system 2. Student, Supervisor, Head of department, Training department access successfully to forum 3. Student, Supervisor, Head of department, Training department comment		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		

Assumptions:	N/A

UC-034. Add post

Use Case Specification

UC ID and Name:	**UC- Add post**		
Create by:	KhanhNN	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor, Head of department, Training department	Secondary Actor:	
Trigger:	N/A		
Description:	Student, Supervisor, Head of department, Training department add post		
Preconditions:	Student, Supervisor, Head of department, Training department have logged into FCPM website		
Post conditions:	Student, Supervisor, Head of department, Training department have logged successfully into FCPM website		
Normal Flow:	**Add post:** 1 Student, Supervisor, Head of department, Training department login successfully to system 2. Student, Supervisor, Head of department, Training department access successfully to forum 3. Student, Supervisor, Head of department, Training department click to button "Add" to add post		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-035. Edit post

Use Case Specification

UC ID and Name:	**UC- Edit post**		
Create by:	KhangNM	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor, Head of department,	Secondary Actor:	

	Training department		
Trigger:	N/A		
Description:	Student, Supervisor, Head of department, Training department edit post		
Preconditions:	Student, Supervisor, Head of department, Training department have logged into FCPM website		
Post conditions:	Student, Supervisor, Head of department, Training department have logged successfully into FCPM website		
Normal Flow:	**Edit post:** 1 Student, Supervisor, Head of department, Training department login successfully to system 2. Student, Supervisor, Head of department, Training department access successfully to forum 3. Student, Supervisor, Head of department, Training department click on Edit post		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-036. Delete post

Use Case Specification

UC ID and Name:	**UC- Delete post**		
Create by:	LongND	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor, Head of department,	Secondary Actor:	

	Training department		
Trigger:	N/A		
Description:	Student, Supervisor, Head of department, Training department delete post		
Preconditions:	Student, Supervisor, Head of department, Training department have logged into FCPM website		
Post conditions:	Student, Supervisor, Head of department, Training department have logged successfully into FCPM website		
Normal Flow:	**Delete post:** 1 Student, Supervisor, Head of department, Training department login successfully to system 2. Student, Supervisor, Head of department, Training department access successfully to forum 3. Student, Supervisor, Head of department, Training department click to button “Delete” to delete post		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-037. Search post

Use Case Specification

UC ID and Name:	**UC- Search post**		
Create by:	DucDD	Date Created:	10/08/2020
Primary Actor:	Student, Supervisor, Head of department,	Secondary Actor:	

	Training department		
Trigger:	N/A		
Description:	Student, Supervisor, Head of department, Training department search post		
Preconditions:	Student, Supervisor, Head of department, Training department have logged into FCPM website		
Post conditions:	Student, Supervisor, Head of department, Training department have logged successfully into FCPM website		
Normal Flow:	**Search post:** 1 Student, Supervisor, Head of department, Training department login successfully to system 2. Student, Supervisor, Head of department, Training department access successfully to forum 3. Student, Supervisor, Head of department, Training department click to "Search" to search post		
Alternative Flows:	N/A		
Exceptions:	Cannot communicate with server 1. FCPM displays error message.		
Priority:	High		
Frequency of Use:	High		
Business Rules:			
Other Information:	N/A		
Assumptions:	N/A		

UC-038 Add user

Use Case Specification

UC ID and Name:	**UC- Add user**		
Create by:	KienBT	Date Created:	10/08/2020
Primary Actor:	Training department	Secondary Actor:	
Trigger:	N/A		

Description:	Training department add user
Preconditions:	Training department has logged into FCPM website
Post conditions:	Training department has logged successfully into FCPM website
Normal Flow:	**Add user:** 1. Training department login successfully to system 2. Training department access successfully to forum 3. Training department click on Capstone Project 4. Training department choose Manage user 5. Training department button "Add" to add user
Alternative Flows:	N/A
Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

UC-039 Approve capstone project in the changing name stage

Use Case Specification

UC ID and Name:	**UC- Approve capstone project in the changing name stage**		
Create by:	KhanhNN	Date Created:	10/08/2020
Primary Actor:	Training department	Secondary Actor:	
Trigger:	N/A		
Description:	Training department add user		

Preconditions:	Training department has logged into FCPM website
Post conditions:	Training department has logged successfully into FCPM website
Normal Flow:	**Approve capstone project in the changing name stage** : 1. Training department login successfully to system 2. Training department access successfully to forum 3. Training department click on Capstone Project 4. Training department choose Manage user 5. Training department button “change” to changing name
Alternative Flows:	N/A
Exceptions:	Cannot communicate with server 1. FCPM displays error message.
Priority:	High
Frequency of Use:	High
Business Rules:	
Other Information:	N/A
Assumptions:	N/A

3.3 Non-functional Requirement

3.3.1 Security

- The system must use authentication with google
- The authentication server returns information of user logged in with email.
- The security matrix is as the following table:

Function	Student	Supervisor	Training department	Head of department
Login	X	X	X	X
Logout	X	X	X	X
Register Capstone Project	X			

Add member	X			
Research model	X			
Show list supervisors and booking	X			
View GUIDE	X			
Upload file	X			
Send mail	X			
Chat	X	X	X	X
Create chat	X			
Confirm join team feature	X			
Edit Profile	X			
Notification component	X	X	X	X
Comment	X	X	X	X
Daily report to member to leader	X	X		
Weekly report to leader to supervisor	X	X		
Manage User			X	
View information, status of list student			X	
Add student and edit supervisor of Capstone Project			X	
View list capstone project		X		X
View detail project		X		X
Manage Capstone Project		X		X
View task detail of Capstone Project		X		X
Dashboard statistical of project		X		X
View information of user	X	X	X	X
Approve capstone project in the project registration stage				X

3.3.2 Supportability

- The application should be easy to use for the user
- The application must have all necessary features to support the user
- The application can be easily extended

3.3.3 Accuracy

- The overall accuracy is calculated by diving total number of correct answers by the number of questions asked

3.3.4 Maintainability and Extensibility

- Source code must follow the coding convention in order to help developers to understand, improve readability of source code and make the website more maintainable or modify the code.

3.3.5 Performance

- Web frontend uses bootstrap to make the website design process simpler and faster, it is also a free toolkit for open-source code to help website templates become more completed and eye-catching.
- This software requires a powerful server machine with high band internet access.
- The app must work on all platforms.
- The system must be usable when many people use it at the same time or at any time.
- Web uses spring boot to help us programmers to simplify the process of programming an application with Spring, focusing only on developing business for the application.

3.3.6 Usability

- The interface is beautiful, good-looking for users, simple and easy to use, it does not break at various zoom levels
- Links and buttons are easily clickable
- Must help users to use the application in the most satisfied way
- The screen layout is clearly streamlined
- Screen should be attractive and attractive to the user
- Main functions are organized into tabs for easy and convenient user access
- Having response buttons and clickable menus to help guide end users to their answer when needed
- Buttons containing important commands must be unique

3.4 Entity Relationship Diagram

CHAPTER 4: SOFTWARE DESGIN

4.1 Purpose

- This chapter is to give the developer team an overview of what the system's architecture is, and how they should be implemented. This chapter consists of:

✓ Architecture overview
✓ Component diagram
✓ Detailed design
✓ Detailed description of components
✓ Database design

4.2 Architectural overview

4.2.1 System Architecture

4.2.1.1 Diagram

Figure: System overview

4.2.2 System Architectural Explanation

4.2.2.1 MVC Architectural

4.2.2.1.1 View application – HTML5 & CSS3 & JavaScript & JQuery & Bootstrap.

- Using HTML5 & CSS3 & JavaScript & JQuery & Bootstrap provides advantage of making the website design process simple, fast, eye-catching, and allows users to freely create ideas, create more and more diverse products. This design also helps to save maximum time designing website interface. Not only that, it also brings many advantages for businesses to develop, web developers can easily use and use in the most convenient way.

4.2.2.2 Controller

✓ In Spring Boot, @Controller annotation indicates that the annotated class is a controller. It is a specialization of @Component and is auto detected through class path scanning. It is typically used in combination with annotated handler methods based on the @Request Mapping annotation.

✓ In the following application, we demonstrate the usage of @Controller. The application returns current data and time to the client.

4.2.2.3 Model

4.2.2.3.1 SQL Database – MySQL Workbench 8.0 CE

Figure: MySQL Workbench 8.0 CE

- MySQL Workbench is a visually modelled and designed database access tool for use with
  a relational MySQL server database. MySQL Workbench helps to create new physical data models and supports modification of existing MySQL databases with reverse / forward techniques, optional management functions.

4.2.2.4 Service providers

4.2.2.4.1 Session state

- The session state service manages session state for every user. They store the state in the memory, in external process or in MySQL database. The membership service stores usernames, passwords and other membership data in MySQL

4.2.2.4.2 Mail service

Figure: Gmail

4.3 Design of FPT Project Management System

4.3.1 Architectural Layers Design

4.3.1.1 FCPM – Layers Design

Figure: FCPM- Layers Design

4.3.1.2 Package design

No.	Directory	Description	Convention
1	/CONFIG	Where the basic configuration of the website is contained	N/A
2	/Controller	The place we call from the service to handle the URL request	N/A
3	/DTO	Where DTO to Entity and Entity to DTO Conversion	N/A
4	/Entity	Where we create entity classes. Each entity will describe a table in DB	N/A
5	/Repository	The place we write the database query to call from the database	N/A
6	/Service	The place we call from the repository to process logic and then push back to the controller	N/A
7	/UTILS	Where we use the web	N/A
8	/Common	Where supporting code for the backend	N/A
9	/Static	Where contains html, image files to design the homepage interface	N/A

4.3.2 Database Design

4.3.2.1 Database diagram

Figure: FCPM database document diagram

4.3.2.2 Explanation of database decision

The image above explains the basic structure of the client-server communication model. A client machine communicates with the server machine in a given network. Each client can send a request from the Graphical user interface (GUI) on the screen, and the server will return the desired result as long as both understand each other. The same goes for the main operation in a MySQL environment:

- ✓ MySQL creates tables to store data, defining the relationships between those tables.
- ✓ The client will send the SQL request with a special command on MySQL.
- ✓ The application on the server will respond to the information and return the results on the client machine.

So, we chose MySQL because:

- ✓ Flexibility and convenience: You can modify the source code to meet your needs without paying any extra costs. The installation process is also very simple and usually takes less than 30 minutes.
- ✓ High Performance: Many server clusters use MySQL. Whether you store big data from e-commerce sites or do heavy IT businesses, MySQL can handle it with smooth, high speeds.
- ✓ Industry standard: The data and technology industry has been using MySQL for many years, so it is a basic skill as a programmer. MySQL users can also deploy projects quickly and hire data experts for a fee if they need to.
- ✓ Safety: Data security is always the most important issue when choosing RDBMS software. With decentralized access and account management, MySQL has a very high standard of security. Host-based authentication and credential encryption are available.

4.3.2.3 Data Dictionary

4.3.2.3.1 Class Diagram, Table description & Table design

- **Class Diagram**

Figure: Class diagram

- **Table description**

No	Table Name	Description
1	users	Save information users.
2	semesters	Save semesters information of students
3	roles	5 roles: Student members, Supervisors, Training department, Head of department, Student leaders.
4	post	Save posts of users.
5	comments	Save comments of users.
6	status	Show and save the status of users and capstone projects.
7	files	Save file names and paths.
8	reports	Save daily and weekly reports.
9	user-roles	To save which users corresponds to which roles.
10	notifications	Save news and notifications.
11	report-details	To save which users corresponds to which reports.
12	evaluations	Send evaluation information from instructors to students.
13	users-posts	To save which users corresponds to which posts.
14	capstone-projects	Save information of capstone project.
15	history-records	Save history with transactions such as adding students, delete students ...
16	evaluation-details	Send evaluation detail information from instructors to students.
17	profession	Save profession information of students.
18	capstone-projects-details	Save detail information of capstone projects.
19	chat	Save chats.
20	notification-details	Save detail notification the training department to students, or instructors to students.
21	statistics	Dashboard statistical all projects.
22	task-details	Save task details information of capstone project.
23	sites	Save sites.
24	chat-details	Save chat details.

- **Table design**

4.3.2.3.1.1: Users

Field	Type	Allow Nulls	Description
ID	VARCHAR (50)		
BIRTH_DATE	DATE	NULL	
CREATED_DATE	DATETIME		
DESCRIPTION	VARCHAR (256)	NULL	
EMAIL	VARCHAR (64)		
ENCRYTED_PASSWORD	VARCHAR (256)		
FIRST_NAME	VARCHAR (30)		
GENDER	INT		
IMAGE	LONGTEXT	NULL	
LAST_NAME	VARCHAR (30)		
PHONE	VARCHAR (15)		
USER_NAME	VARCHAR (50)		
EVALUATION_DETAIL _ID	INT	NULL	
LOCATIONS_ID	INT	NULL	
REPORT_DETAIL_ID	INT	NULL	
REPORT_RECEIVE_ID	INT	NULL	
REPORT_ID	INT	NULL	
STATUS_ID	INT	NULL	
NOTIFICATION_ID	INT	NULL	

4.3.2.3.1.2: Semesters

Field	Type	Allow Nulls	Description
ID	INT		
END_DATE	DATE	NULL	
END_REGISTER_CAPSTON E	DATE	NULL	
NAME	VARCHAR		
START_DATE	DATE	NULL	
START_REGISTER_CAPSTO NE	DATE	NULL	

4.3.2.3.1.3: Roles

Field	Type	Allow Nulls	Description
ID	INT		
NAME	VARCHAR (20)		

4.3.2.3.1.4: Post

Field	Type	Allow Nulls	Description
ID	INT		
CREATED_DATE	DATETIME	NULL	
DESCRIPTION	LONGTEXT	NULL	
TITLE	VARCHAR	NULL	
AUTHOR_ID	(256)	NULL	
POST_AUTHOR	VARCHAR (50)	NULL	
	VARCHAR (50)		

4.3.2.3.1.5: Comments

Field	Type	Allow Nulls	Description
ID	INT		
CONTENT	VARCHAR (256)		
CREATED_DATE	DATETIME		
POST_ID	INT	NULL	
REPORT_DETAIL_ID	INT	NULL	
SENDER_ID	VARCHAR (50)	NULL	

4.3.2.3.1.6: Status

Field	Type	Allow Nulls	Description
DESCRIPTION	VARCHAR	NULL	
	(256)		
NAME	VARCHAR (50)		

4.3.2.3.1.7: Files

Field	Type	Allow Nulls	Description

DESCRIPTION	INT	
FILE_NAME	VARCHAR (256)	NULL
PATH	VARCHAR (256)	NULL
CAPSTONE_PROJECT_ID	VARCHAR (256)	NULL
EVALUATION_ID	INT	NULL
POST_ID	INT	NULL
REPORT_DETAIL_ID	INT	NULL

4.3.2.3.1.8: Reports

Field	Type	Allow Nulls	Description
ID	INT		
NAME	VARCHAR (50)		
TYPE	VARCHAR (50)		
SENDER_ID	VARCHAR (50)		

4.3.2.3.1.9: User-roles

Field	Type	Allow Nulls	Description
USER_ID	VARCHAR (50)		
ROLE_ID	INT		

4.3.2.3.1.10: Notifications

Field	Type	Allow Nulls	Description
ID	INT		
CONTENT	LONGTEXT		
TITLE	VARCHAR (256)		
TYPE	VARCHAR (50)		
CREATED_DATE	DATETIME	NULL	
SENDER_ID	VARCHAR (50)	NULL	

4.3.2.3.1.11: Report-details

Field	Type	Allow Nulls	Description
LAST_MODIFIED_DATE	DATETIME		

REPORT_ID	INT	NULL	

4.3.2.3.1.12: Evaluations

Field	Type	Allow Nulls	Description
ID	INT		
NAME	VARCHAR (50)		
TYPE	VARCHAR (50)		
CAPSTONE_PROJECT _ID	INT	NULL	

4.3.2.3.1.13: Users-post

Field	Type	Allow Nulls	Description
USERS_ID	VARCHAR(50)		
POSTS_ID	INT		

4.3.2.3.1.14: Capstone-projects

Field	Type	Allow Nulls	Description
ID	INT		
DESCRIPTION_ACTI ON	LONGTEXT	NULL	
DESCRIPTION	VARCHAR (256)		
DOCUMENT	VARCHAR (256)		
NAME	VARCHAR (100)	NULL	
NAME_ABBREVIATI ON	VARCHAR (20)	NULL	
NAME_LANG_OTHE R	VARCHAR (20)	NULL	
NAME_VI	VARCHAR (256)		
PROGRAM	VARCHAR (10)		
SPECIALTY	INT	NULL	
PROFESSION_ID	INT	NULL	
STATUS_ID			

4.3.2.3.1.15: History-records

Field	Type	Allow Nulls	Description
ID	INT		
CONTENT	VARCHAR (256)		
CREATED_DATE	DATETIME	NULL	
LAST_MODIFIED_DA	DATETIME	NULL	
TE	INT	NULL	
CAPSTONE_PROJECT	INT	NULL	
_ID	INT	NULL	
EVALUATION_ID	INT	NULL	
POST_ID	VARCHAR (50)	NULL	
REPORT_ID	INT	NULL	
USER_ID			
NOTIFICATION_ID			

4.3.2.3.1.16: Evaluation-details

Field	Type	Allow Nulls	Description
ID	INT		
CONTENT	VARCHAR (256)		
GRADE	FLOAT		
TYPE	VARCHAR (50)		
EVALUATION_ID	INT		
ASSESSOR_ID	VARCHAR (50)		

4.3.2.3.1.17: Profession

Field	Type	Allow Nulls	Description
ID	INT		
MAX_MEMBER	INT	NULL	
MIN_MEMBER	INT	NULL	
NAME	VARCHAR (100)		
SUBJECT_CODE	VARCHAR (50)		

4.3.2.3.1.18: Capstone-projects-details

Field	Type	Allow Nulls	Description
ID	INT		

DESCRIPTION_ACTIO	VARCHAR (50)	NULL	
N	VARCHAR (256)	NULL	
CAPSTONE_PROJECT	INT	NULL	
_ID	INT	NULL	
STATUS_ID			
USER_ID			

4.3.2.3.1.19: Chat

Field	Type	Allow Nulls	Description
ROOM	VARCHAR (50)		
USER_ID	VARCHAR (50)	NULL	

4.3.2.3.1.20: Notification-details

Field	Type	Allow Nulls	Description
NOTIFICATION_ID	INT	NULL	
USER_ID	VARCHAR (50)	NULL	

4.3.2.3.1.21: Statistics

Field	Type	Allow Nulls	Description
ID	INT		
END_DATE	DATE	NULL	
START_DATE	DATE	NULL	
TIME_TRACKING_CURRE	DOUBLE	NULL	
NT	DOUBLE	NULL	
TIME_TRACKING_DONE	DOUBLE	NULL	
TIME_TRACKING_PROGR	DOUBLE	NULL	
ESS	INT	NULL	
TIME_TRACKING_TODO	INT	NULL	
WEEK			
CAPSTONE_PROJECT_ID			

4.3.2.3.1.22: Task-details

Field	Type	Allow Nulls	Description
ID	INT		
ASSIGNEE	VARCHAR (256)	NULL	

END_DATE	DATE	NULL	
START_DATE	DATE	NULL	
STATUS	VARCHAR (100)	NULL	
SUMMARY	LONGTEXT	NULL	
TIME_SPENT	VARCHAR (100)	NULL	
TIME_TRACKING	DOUBLE	NULL	
WEEK	INT	NULL	
CAPSTONE_PROJECT _ID	INT	NULL	

4.3.2.3.1.23: Sites

Field	Type	Allow Nulls	Description
ID	INT		
READ_STATUS	VARCHAR (50)		
CHAT_ID	INT	NULL	
USER_ID	VARCHAR(50)	NULL	

4.3.2.3.1.24: Chat-details

Field	Type	Allow Nulls	Description
NOTIFICATION_ID	INT	NULL	
USER_ID	VARCHAR(50)	NULL	

4.4 SEQUENCE DIAGRAM

4.4.1: Description of the Sequence Diagram

4.4.1.1: Login

Class Diagram:

Users
getId() String
setId(String) void
getUsername() String
setUsername(String) void
getFirstName() String
setFirstName(String) void
getLastName() String
setLastName(String) void
getBirthDate() Date
setBirthDate(Date) void
getGender() Integer
setGender(Integer) void
getPhone() String
setPhone(String) void
getImage() String
setImage(String) void
getEmail() String
setEmail(String) void
getCreatedDate() Date
setCreatedDate(Date) void
getAddress() String
setAddress(String) void
getDescription() String
setDescription(String) void
getStatus() Status
setStatus(Status) void
getRoleUser() List<UserRoles>
setRoleUser(List<UserRoles>) void
getHistoryRecords() List<HistoryRecords>
setHistoryRecords(List<HistoryRecords>) void
getCapstoneProjectDetails() List<CapstoneProjectDetails>
setCapstoneProjectDetails(List<CapstoneProjectDetails>) void
getNotificationDetails() List<NotificationDetails>
setNotificationDetails(List<NotificationDetails>) void
getReportReceives() List<Reports>
setReportReceives(List<Reports>) void
getComments() List<Comments>
setComments(List<Comments>) void
getEvaluationDetail() EvaluationDetails
setEvaluationDetail(EvaluationDetails) void
getPosts() List<Posts>
setPosts(List<Posts>) void
getChats() List<Chat>
setChats(List<Chat>) void
getReport() List<Reports>
setReport(List<Reports>) void
getSemester() Semesters
setSemester(Semesters) void
getSite() Sites
setSite(Sites) void
getChatDetail() List<ChatDetails>
setChatDetail(List<ChatDetails>) void
equals(Object) boolean
hashCode() int
UserRepository
getAllUserStudent(String, String) List<UserManagementDTO>
findByUsername(String) List<Users>
findByEmail(String) Optional<Users>
getUserByRoleId(Pageable, Integer) Page<Users>
findReportByUserId(Pageable, String) Page<Reports>
getLectureNoteBooked2ByRoleId(Pageable, Integer, String, String) Page<Users>
getLectureNoteBookedByRoleId(Pageable, Integer, String) Page<Users>
getUserByUserName(String) Users
getUserByUserRoleAndProjectId(Integer, Integer) List<Users>
getUserStudentByStatusId(Integer) List<UserManagementDTO>
countStudent(Integer, String, String) Integer
getAllUserStudentHasNoTeam(String, String) List<UserManagementDTO>
countStudentHasNoTeam(String, String) Integer
countAllStudent(String, String) Integer
countStudentEligibleCapstone(String, String) Integer
updateProfileByUserId(String, String, String, String, Date, String) void
UserRoleKey
getRole() Roles
setRole(Roles) void
getUser() Users
setUser(Users) void
getSerialversionuid() long
UserRoles
getUserRoleKey() UserRoleKey
setUserRoleKey(UserRoleKey) void
getSerialversionuid() long
UserService
findById(String) Users
getAllUser() List<Users>
getAllUserStudent(String, String) List<UserManagementDTO>
deleteUser(String) boolean
save(Users) boolean
saveAll(List<Users>) boolean
findAll() List<Users>
getUserByRoleId(Integer) List<Users>
findByUsername(String) List<Users>
findByEmail(String) Users
findPaginated(Pageable) Page<Users>
findPaginatedNotLecture2Booked(Pageable, String, String) Page<Users>
findPaginatedNotLectureBooked(Pageable, String) Page<Users>
getUserByUserName(String) Users
getUserByUserRoleAndProjectId(Integer, Integer) List<Users>
getUserStudentByStatusId(Integer) List<UserManagementDTO>
countStudent(Integer, String, String) Integer
getAllUserStudentHasNoTeam(String, String) List<UserManagementDTO>
countStudentHasNoTeam(String, String) Integer
countAllStudent(String, String) Integer
countStudentEligibleCapstone(String, String) Integer
updateProfileByUserId(String, String, String, String, Date, String) boolean
UserRoleService
getRoleNamesByUserId(String) List<String>
saveRoleUser(UserRoles) boolean
removeAllRoleOfUserByUserId(String) boolean
isExists(UserRoleKey) UserRoles
getRoleNamesByEmail(String) List<String>
updateRoleStudentReject(String) void
updateRoleLeader(String) void
Roles
getId() Integer
setId(Integer) void
getName() String
setName(String) void
getUserRole() List<UserRoles>
setUserRole(List<UserRoles>) void
getSerialversionuid() long
UserRolesRepository
getRoleNamesByUserId(String) List<String>
removeByUserId(String) void
getUserByRoleId(Integer) List<Users>
getRoleNamesByEmail(String) List<String>
updateRoleStudentReject(String) void
updateRoleLeader(String) void
GoogleUtils
getToken(String) String
getUserInfo(String) String
buildUser(String, Users) UserDetails
RoleRepository
findRolesByName(String) Roles
RoleService
findByName(String) Roles
findAll() List<Roles>
findById(Integer) Roles
LoginController
login(HttpServletRequest, Model) String
loginGoogle(HttpServletRequest, HttpServletResponse) String
userInfo(Model, Principal) String
admin() String
accessDenied() String
Powered by yFiles

Sequence Diagram:

4.4.1.2: Logout

Sequence Diagram:

4.4.1.3: Register Capstone Project

Class Diagram:

Sequence Diagram:

4.4.1.4: Add member

Class Diagram:

Sequence Diagram:

4.4.1.5: Research model

4.4.1.6: Show list supervisors and choose

Class Diagram:

Sequence Diagram:

4.4.1.7: view GUIDE

4.4.1.8: Upload file

Class Diagram:

Sequence Diagram:

4.4.1.9: Send mail

Sequence Diagram:

4.4.1.10: Chat

Class Diagram:

Sequence Diagram:

4.4.1.11: Create chat

Class Diagram:

Sequence Diagram:

4.4.1.12: Confirm join team feature

Class Diagram:

CapstoneProjectDetails
getSupType() String
setSupType(String) void
getDesAction() String
setDesAction(String) void
getId() Integer
setId(Integer) void
getCapstoneProject() CapstoneProjects
setCapstoneProject(CapstoneProjects) void
getUser() Users
setUser(Users) void
getStatus() Status
setStatus(Status) void
CapstoneProjectDetailService
addCapstonprojectDetail(CapstoneProjectDetails) boolean
getDetailByCapstoneProjectId(Integer) List<CapstoneProjectDetails>
getUserByCapstionelD(Integer) List<CapstoneProjectDetails>
getProjectIdByUserId(String) List<Integer>
getIdProjectByUserIDCheckApprove(String) List<Integer>
getStatuByCapstoneProjectDetailIdAndUserId(Integer, Integer) Integer
getUserByCapstoneProjectDetailId(Integer) List<Users>
countLecturersIdAndCapstoneProjectIdOP1(Integer) Integer
countLecturersIdAndCapstoneProjectIdOP2(Integer) Integer
countLecturersByProjectId(Integer) Integer
updateStatusUserProject(String, Integer) Integer
findUserByStatusRegisted(String) Users
deleteCapstoneProjectDetailsByUserId(String, Integer) Integer
deleteRejectCapstoneProjectDetailsByUserId(String, Integer) Integer
findCapstoneProjectByUserId(String) CapstoneProjects
getUserStudentMemberByProjectId(Integer) List<Users>
getOneProjectIdByUserId(String) Integer
getUserById(Integer) List<Users>
getStatusById(Integer) List<Status>
getByProjectId(Integer) List<Object[]>
findById(Integer) CapstoneProjectDetails
save(CapstoneProjectDetails) boolean
userLecturersIdAndCapstoneProjectIdOP1(Integer) Users
userLecturersIdAndCapstoneProjectIdOP2(Integer) Users
getUserWaitingApproveByCapstoneProjectDetailId(Integer) List<Users>
getUserByCapstoneProject(Integer) List<Users>
listUserRoleByProjectId(Integer) List<UserRoles>
getUserEditByCapstoneProject(Integer) List<Object[]>
checkUserCapstoneDetail(String, Integer) CapstoneProjectDetails
findUserByCapstoneProjectDetailId(Integer) List<Users>
CapstoneProjectDetailRepository
getDetailByCapstoneProjectId(Integer) List<CapstoneProjectDetails>
getIdProjectByUserID(String) List<Integer>
getIdProjectByUserIDCheckApprove(String) List<Integer>
getStatuByCapstoneProjectDetailIdAndUserId(Integer, Integer) Integer
getUserByCapstoneProjectDetailId(Integer) List<Users>
getUserWaitingApproveByCapstoneProjectDetailId(Integer) List<Users>
getUserByCapstoneProject(Integer) List<Users>
countLecturersByCapstoneProjectId(Integer) Integer
countLecturersIdAndCapstoneProjectIdOP1(Integer) Integer
countLecturersIdAndCapstoneProjectIdOP2(Integer) Integer
getUserIdByCapstoneProjectDetailId(Integer) List<CapstoneProjectDetails>
updateStatusUserProject(String, Integer) Integer
findUserByStatusRegisted(String) Users
deleteCapstoneProjectDetailsByUserId(String, Integer) Integer
deleteRejectCapstoneProjectDetailsByUserId(String, Integer) Integer
findCapstoneProjectByUserId(String) CapstoneProjects
getUserStudentMemberByProjectId(Integer) List<Users>
getOneProjectIdByUserId(String) Integer
getByProjectId(Integer) List<Object[]>
getUserByCapstoneProjectDetailId1(Integer) CapstoneProjectDetails
getUserById(Integer) List<Users>
getStatusById(Integer) List<Status>
userLecturersIdAndCapstoneProjectIdOP1(Integer) Users
userLecturersIdAndCapstoneProjectIdOP2(Integer) Users
listUserRoleByProjectId(Integer) List<UserRoles>
getUserEditByCapstoneProject(Integer) List<Object[]>
checkUserCapstoneDetail(String, Integer) CapstoneProjectDetails
deleteCapstoneProjectDetailsByUserNotIn(Collection<Users>) void
findUserByCapstoneProjectDetailId(Integer) List<Users>
1
1
DetailProjectController
detailProject(Integer, Model, Principal) String
approve(Integer, Model, Principal) String
reject(Integer, Model, Principal) String
editReport(Integer, Model, Principal) String
editProject(Integer, Model, Principal) String
getMember(Principal) String
getMember(String) String
registerProject(CapstoneProjectDTO, BindingResult, Model, Principal, HttpServletRequest, RedirectAttributes) String
chagingNameProject(Integer, Model, Principal) String
saveChangingNameProject(CapstoneProjects, RedirectAttributes, Model, Principal) String
Powered by yFiles

Sequence Diagram:

4.4.1.13: Edit profile

Class Diagram:

Sequence Diagram:

4.4.1.14: Notification component

Class Diagram:

Sequence Diagram:

4.4.1.15: Comment

Class Diagram:

Sequence Diagram:

4.4.1.16: Daily report to member to leader

Class Diagram:

Sequence Diagram:

4.4.1.17: Weekly report to leader to supervisor

Class Diagram:

Sequence Diagram:

4.4.1.18: Manage user

Class Diagram:

Sequence Diagram:

4.4.1.19: View information, status of list student

Class Diagram:

Status
getId() Integer
setId(Integer) void
getName() String
setName(String) void
getUsers() List<Users>
setUsers(List<Users>) void
getCapstoneProjectDetails() List<CapstoneProjectDetails>
setCapstoneProjectDetails(List<CapstoneProjectDetails>) void
Users
getId() String
setId(String) void
getUsername() String
setUsername(String) void
getFirstName() String
setFirstName(String) void
getLastName() String
setLastName(String) void
getBirthDate() Date
setBirthDate(Date) void
getGender() Integer
setGender(Integer) void
getPhone() String
setPhone(String) void
getImage() String
setImage(String) void
getEmail() String
setEmail(String) void
getCreatedDate() Date
setCreatedDate(Date) void
getAddress() String
setAddress(String) void
getDescription() String
setDescription(String) void
getStatus() Status
setStatus(Status) void
getRoleUser() List<UserRoles>
setRoleUser(List<UserRoles>) void
getHistoryRecords() List<HistoryRecords>
setHistoryRecords(List<HistoryRecords>) void
getCapstoneProjectDetails() List<CapstoneProjectDetails>
setCapstoneProjectDetails(List<CapstoneProjectDetails>) void
getNotificationDetails() List<NotificationDetails>
setNotificationDetails(List<NotificationDetails>) void
getReportReceives() List<Reports>
setReportReceives(List<Reports>) void
getComments() List<Comments>
setComments(List<Comments>) void
getEvaluationDetail() EvaluationDetails
setEvaluationDetail(EvaluationDetails) void
getPosts() List<Posts>
setPosts(List<Posts>) void
getChats() List<Chat>
setChats(List<Chat>) void
getReport() List<Reports>
setReport(List<Reports>) void
getSemester() Semesters
setSemester(Semesters) void
getSite() Sites
setSite(Sites) void
getChatDetail() List<ChatDetails>
setChatDetail(List<ChatDetails>) void
equals(Object) boolean
hashCode() int
UserRepository
getAllUserStudent(String, String) List<UserManagementDTO>
findByUsername(String) List<Users>
findByEmail(String) Optional<Users>
getUserByRoleId(Pageable, Integer) Page<Users>
findReportByUserId(Pageable, String) Page<Reports>
getLectureNoteBooked2ByRoleId(Pageable, Integer, String, String) Page<Users>
getLectureNoteBookedByRoleId(Pageable, Integer, String) Page<Users>
getUserByUserName(String) Users
getUserByUserRoleAndProjectId(Integer, Integer) List<Users>
getUserStudentByStatusId(Integer) List<UserManagementDTO>
countStudent(Integer, String, String) Integer
getAllUserStudentHasNoTeam(String, String) List<UserManagementDTO>
countStudentHasNoTeam(String, String) Integer
countAllStudent(String, String) Integer
countStudentEligibleCapstone(String, String) Integer
updateProfileByUserId(String, String, String, String, Date, String) void
StatusService
getStatusById(int) Status
findByName(String) Status
getAll() List<Status>
findStatusByUserId(String) Status
findStatusByCapstoneProject(Integer) Status
UserService
findById(String) Users
getAllUser() List<Users>
getAllUserStudent(String, String) List<UserManagementDTO>
deleteUser(String) boolean
save(Users) boolean
saveAll(List<Users>) boolean
findAll() List<Users>
getUserByRoleId(Integer) List<Users>
findByUsername(String) List<Users>
findByEmail(String) Users
findPaginated(Pageable) Page<Users>
findPaginatedNotLecture2Booked(Pageable, String, String) Page<Users>
findPaginatedNotLectureBooked(Pageable, String) Page<Users>
getUserByUserName(String) Users
getUserByUserRoleAndProjectId(Integer, Integer) List<Users>
getUserStudentByStatusId(Integer) List<UserManagementDTO>
countStudent(Integer, String, String) Integer
getAllUserStudentHasNoTeam(String, String) List<UserManagementDTO>
countStudentHasNoTeam(String, String) Integer
countAllStudent(String, String) Integer
countStudentEligibleCapstone(String, String) Integer
updateProfileByUserId(String, String, String, String, Date, String) boolean
StatusRepository
findById(Integer) Optional<Status>
findByName(String) Status
findByUsers(Users) Status
findStatusByUserId(String) Status
findStatusByCapstoneProject(Integer) Status
CapstoneProjectController
forum(Model, Principal) String
getProjects(Model, Principal, String, String, Integer, Integer, String) String
listStudentProject(Model, String, Principal) String
updateStatusDetail(String, String, Principal) String
updateStatus(String, String, Principal) String
reject(String, String, Principal) String
rejectDetail(String, String, Principal) String
updateStatusList(String, String, Principal) String
rejectList(String, String, Principal) String
insertDetail(String, String, Principal) String
editSupervisors(String, String, Principal) String
getMemberProject(String, String, Principal) String
getSupervisorsProject(String, String, Principal) String
getRegisterProject(Model, Principal) String
getMember(String) String
registerProject(CapstoneProjectDTO, BindingResult, Model, Principal, HttpServletRequest) String
getForm(Principal) String
1
1
1
1
1
1
1
1
Powered by yFiles

Sequence Diagram:

4.4.1.20: Add student and edit supervisor of Capstone Project

Class Diagram:

Sequence Diagram:

4.4.1.21: View list capstone project

Class Diagram:

Sequence Diagram:

4.4.1.22: View detail project

Class Diagram:

Sequence Diagram:

4.4.1.23: Manage Capstone Project

Class Diagram:

Supervisor,
Head of department
List Capstone Project Page
Controller
Service
Repository
Database
Go to the page and view list capstone project
SendRequest
GetNotification
SendRequest
querySQL
If capstone project doesn't exist, error message
Else
Success ()
Display list capstone project
Manage capstone project

4.4.1.24: View task detail of Capstone Project

Class Diagram:

Sequence Diagram:

4.4.1.25: Dashboard statistical of project

Class Diagram:

Sequence diagram:

4.4.1.26: View information of user

Class Diagram:

Sequence Diagram:

4.4.1.27: Approve capstone project in the project registration stage

Class Diagram:

Powered by yFiles

Sequence Diagram:

4.4.1.28: Change pass / fail results

Class Diagram:

Status
getId() Integer
setId(Integer) void
getName() String
setName(String) void
getUsers() List<Users>
setUsers(List<Users>) void
getCapstoneProjectDetails() List<CapstoneProjectDetails>
setCapstoneProjectDetails(List<CapstoneProjectDetails>) void
CapstoneProjects
getStatistics() List<Statistics>
setStatistics(List<Statistics>) void
getDesAction() String
setDesAction(String) void
getId() Integer
setId(Integer) void
getName() String
setName(String) void
getNameOther() String
setNameOther(String) void
getNameVi() String
setNameVi(String) void
getNameAbbreviation() String
setNameAbbreviation(String) void
getProfession() Profession
setProfession(Profession) void
getSpecialty() String
setSpecialty(String) void
getDocument() String
setDocument(String) void
getProgram() String
setProgram(String) void
getDescription() String
setDescription(String) void
getStatus() Status
setStatus(Status) void
getHistoryRecords() List<HistoryRecords>
setHistoryRecords(List<HistoryRecords>) void
getFiles() List<Files>
setFiles(List<Files>) void
getCapstoneProjectDetails() List<CapstoneProjectDetails>
setCapstoneProjectDetails(List<CapstoneProjectDetails>) void
getEvaluations() List<Evaluations>
setEvaluations(List<Evaluations>) void
getSemester() Semesters
setSemester(Semesters) void
getSite() Sites
setSite(Sites) void
getNameChanging() String
setNameChanging(String) void
getNameViChanging() String
setNameViChanging(String) void
CapstoneProjectService
getCapstoneProjectNameByUserId(String) List<String>
saveRegisterProject(CapstoneProjects) boolean
getCapstonProjectById(Integer) CapstoneProjects
getAllProject() List<CapstoneProjects>
registerProject(CapstoneProjectDTO, Principal, String) String
getAllByUserId(String, Integer, Integer, Integer, Integer, String) List<Object[]>
findById(Integer) CapstoneProjects
getCountStudent(Integer) Integer
getCapstoneProjectByUserId(String) CapstoneProjects
updateStatusCapstoneProjectSendTD(Integer) boolean
deleteUserNotSubmitCapstone(Integer) boolean
updateProject(CapstoneProjectDTO, Principal, String, Integer) String
updateStatusCapstoneProjectChangingName(Integer) boolean
capstoneProjectChangingName(String, String, Integer) boolean
getCapstoneProjecRegistingtByUserId(String) CapstoneProjects
getCapstoneProjectRegistedByUserId(String) CapstoneProjects
findUserByCapstoneProjectId(Integer) List<Users>
findCapstoneProjectRegistedBySupervisorId(String) List<CapstoneProjects>
updateSupervisorsSubmitCapstone(Integer) boolean
CapstoneProjectRepository
findByName(String) List<CapstoneProjects>
getCapstoneProjectNameByUserId(String) List<String>
findById(Integer) Optional<CapstoneProjects>
getAll(String, Integer, Integer, Integer, Integer, String) List<Object[]>
getCountStudent(Integer) Integer
getCapstoneProjectByUserId(String) CapstoneProjects
updateStatusCapstoneProjectSendTD(Integer) void
updateStatusCapstoneProjectChangingName(Integer) void
deleteUserNotSubmitCapstone(Integer) void
capstoneProjectChangingName(String, String, Integer) void
getCapstoneProjecRegistingtByUserId(String) CapstoneProjects
getCapstoneProjectRegistedByUserId(String) CapstoneProjects
findUserByCapstoneProjectId(Integer) List<Users>
findCapstoneProjectRegistedBySupervisorId(String) List<CapstoneProjects>
updateSupervisorsSubmitCapstone(Integer) void
CapstoneProjectController
forum(Model, Principal) String
getProjects(Model, Principal, String, String, Integer, Integer, String) String
listStudentProject(Model, String, Principal) String
updateStatusDetail(String, String, Principal) String
updateStatus(String, String, Principal) String
reject(String, String, Principal) String
rejectDetail(String, String, Principal) String
updateStatusList(String, String, Principal) String
rejectList(String, String, Principal) String
insertDetail(String, String, Principal) String
editSupervisors(String, String, Principal) String
getMemberProject(String, String, Principal) String
getSupervisorsProject(String, String, Principal) String
getRegisterProject(Model, Principal) String
getMember(String) String
registerProject(CapstoneProjectDTO, BindingResult, Model, Principal, HttpServletRequest) String
getForm(Principal) String
1
1
1
1
Powered by yFiles

Sequence Diagram:

4.4.1.29: Create notification

Class Diagram:

Sequence Diagram:

4.4.1.30: View history record

Class Diagram:

HistoryRecords
getNotification() Notifications
setNotification(Notifications) void
getId() Integer
setId(Integer) void
getUser() Users
setUser(Users) void
getCreatedDate() Date
setCreatedDate(Date) void
getLastModifiedDate() Date
setLastModifiedDate(Date) void
getContent() String
setContent(String) void
getCapstoneProject() CapstoneProjects
setCapstoneProject(CapstoneProjects) void
getReport() Reports
setReport(Reports) void
getEvaluation() Evaluations
setEvaluation(Evaluations) void
getPost() Posts
setPost(Posts) void
HistoryRecordService
findById(Integer) HistoryRecords
deleteHistoryRecord(Integer) boolean
save(HistoryRecords) boolean
findAll() List<HistoryRecords>
getHistoryRecordsByUserId(String) List<HistoryRecords>
findHistoryByProjectId(Integer) HistoryRecords
findHistoryByUserId(String) HistoryRecords
getByReportId(Integer) HistoryRecords
findHistoryByUserIdCapstoneId(String, Integer) HistoryRecords
getDataRoleStudent() List<HistoryRecords>
getDataRoleHead() List<HistoryRecords>
getDataRoleTrainingDepartment() List<HistoryRecords>
getDataRoleLecture() List<HistoryRecords>
getCapstoneRegister() List<HistoryRecords>
getCapstoneApprove() List<HistoryRecords>
getCapstoneReject() List<HistoryRecords>
getCapstoneBookingSupervisors() List<HistoryRecords>
getAddSupervisorsForCapstone() List<HistoryRecords>
HistoryRecordsRepository
getHistoryRecordsByUserId(String) List<HistoryRecords>
findHistoryByProjectId(Integer) HistoryRecords
findHistoryByUserId(String) HistoryRecords
getByReportId(Integer) HistoryRecords
findHistoryByUserIdCapstoneId(String, Integer) HistoryRecords
getDataRoleStudent() List<HistoryRecords>
getDataRoleTrainingDepartment() List<HistoryRecords>
getDataRoleHead() List<HistoryRecords>
getDataRoleLecture() List<HistoryRecords>
getCapstoneRegister() List<HistoryRecords>
getCapstoneApprove() List<HistoryRecords>
getCapstoneReject() List<HistoryRecords>
getCapstoneBookingSupervisors() List<HistoryRecords>
getAddSupervisorsForCapstone() List<HistoryRecords>
1
1
HistoryRecordsController
loadHistory(Model, Integer, Principal) String
loadDataTableToTheDropdown(Model, Integer, Integer, Principal) String
Powered by yFiles

Sequence Diagram:

4.4.1.31: Change name capstone project

Class Diagram:

CapstoneProjects
getStatistics() List<Statistics>
setStatistics(List<Statistics>) void
getDesAction() String
setDesAction(String) void
getId() Integer
setId(Integer) void
getName() String
setName(String) void
getNameOther() String
setNameOther(String) void
getNameVi() String
setNameVi(String) void
getNameAbbreviation() String
setNameAbbreviation(String) void
getProfession() Profession
setProfession(Profession) void
getSpecialty() String
setSpecialty(String) void
getDocument() String
setDocument(String) void
getProgram() String
setProgram(String) void
getDescription() String
setDescription(String) void
getStatus() Status
setStatus(Status) void
getHistoryRecords() List<HistoryRecords>
setHistoryRecords(List<HistoryRecords>) void
getFiles() List<Files>
setFiles(List<Files>) void
getCapstoneProjectDetails() List<CapstoneProjectDetails>
setCapstoneProjectDetails(List<CapstoneProjectDetails>) void
getEvaluations() List<Evaluations>
setEvaluations(List<Evaluations>) void
getSemester() Semesters
setSemester(Semesters) void
getSite() Sites
setSite(Sites) void
getNameChanging() String
setNameChanging(String) void
getNameViChanging() String
setNameViChanging(String) void
CapstoneProjectService
getCapstoneProjectNameByUserId(String) List<String>
saveRegisterProject(CapstoneProjects) boolean
getCapstonProjectById(Integer) CapstoneProjects
getAllProject() List<CapstoneProjects>
registerProject(CapstoneProjectDTO, Principal, String) String
getAllByUserId(String, Integer, Integer, Integer, Integer, String) List<Object[]>
findById(Integer) CapstoneProjects
getCountStudent(Integer) Integer
getCapstoneProjectByUserId(String) CapstoneProjects
updateStatusCapstoneProjectSendTD(Integer) boolean
deleteUserNotSubmitCapstone(Integer) boolean
updateProject(CapstoneProjectDTO, Principal, String, Integer) String
updateStatusCapstoneProjectChangingName(Integer) boolean
capstoneProjectChangingName(String, String, Integer) boolean
getCapstoneProjecRegistingtByUserId(String) CapstoneProjects
getCapstoneProjectRegistedByUserId(String) CapstoneProjects
findUserByCapstoneProjectId(Integer) List<Users>
findCapstoneProjectRegistedBySupervisorId(String) List<CapstoneProjects>
updateSupervisorsSubmitCapstone(Integer) boolean
CapstoneProjectRepository
findByName(String) List<CapstoneProjects>
getCapstoneProjectNameByUserId(String) List<String>
findById(Integer) Optional<CapstoneProjects>
getAll(String, Integer, Integer, Integer, Integer, String) List<Object[]>
getCountStudent(Integer) Integer
getCapstoneProjectByUserId(String) CapstoneProjects
updateStatusCapstoneProjectSendTD(Integer) void
updateStatusCapstoneProjectChangingName(Integer) void
deleteUserNotSubmitCapstone(Integer) void
capstoneProjectChangingName(String, String, Integer) void
getCapstoneProjecRegistingtByUserId(String) CapstoneProjects
getCapstoneProjectRegistedByUserId(String) CapstoneProjects
findUserByCapstoneProjectId(Integer) List<Users>
findCapstoneProjectRegistedBySupervisorId(String) List<CapstoneProjects>
updateSupervisorsSubmitCapstone(Integer) void
«create»
1
1
DetailProjectController
detailProject(Integer, Model, Principal) String
approve(Integer, Model, Principal) String
reject(Integer, Model, Principal) String
editReport(Integer, Model, Principal) String
editProject(Integer, Model, Principal) String
getMember(Principal) String
getMember(String) String
registerProject(CapstoneProjectDTO, BindingResult, Model, Principal, HttpServletRequest, RedirectAttributes) String
chagingNameProject(Integer, Model, Principal) String
saveChangingNameProject(CapstoneProjects, RedirectAttributes, Model, Principal) String
Powered by yFiles

Sequence Diagram:

4.4.1.32: View list post

Class Diagram:

Sequence Diagram:

4.4.1.33: Get comment

Class Diagram:

Sequence Diagram:

4.4.1.34: Add post

Class Diagram:

Sequence Diagram:

4.4.1.35: Edit post

Class Diagram:

Sequence Diagram:

4.4.1.36: Delete post

Class Diagram:

Sequence Diagram:

4.4.1.37: Search post

Class Diagram:

Sequence Diagram:

4.4.1.38: Add user

Class Diagram:

Sequence Diagram:

4.4.1.39: Approve capstone project in the changing name stage

4.5. User Interface Design

4.5.1 Description of the User Interface

4.5.1.1. Login

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Login	Click to Sign in With Google	No	Navigate to account for authentication
2	Login	Choose FPT email	No	Navigate to forum

4.5.1.2. Logout

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Logout	Click to Do Thao	No	Show logout
2	Logout	Clock to Logout	No	Navigate to Login Page

4.5.1.3. Register Capstone Project

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Register	Click to Create Capstone	No	Navigate to Register Project Page
2	Choose leader or member	Choose leader or member	No	Show leader or Member
3	Add member	Click to Add	No	Show name of member
4	Add file	Click to choose file	No	Show file
5	Submit Register	Click to submit	No	Show submit success or fail
6	Reset register	Click to reset	No	Re-register

Fields

No	Field Name	Descripti	Read Only	Mandatory	Control Type	Data Type	Length
7	Name Project	Name of project	No	Yes	Text	String	120
8	Name project other language	Name project other language	No	Yes	Text	String	120
9	Name project Vietnamese	Name project Vietnam	No	Yes	Text	String	120

10	Name project Abbreviation	Name project Abbreviation	No	Yes	Text	String	120
11	Profession project	Profession Project	No	Yes	Text	String	120
12	Program project	Program Project	No	Yes	Text	String	120
13	Specialty project	Specialty Project	No	Yes	Text	String	120
14	Document project	Document Project	No	Yes	Text	String	120
12	Member	Member add	No	Yes	Text	String	120
13	Description project	Description Project	No	Yes	Text	String	120

4.5.1.4 Add member

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Add member	Click to "Add"	No	Show name of member

4.5.1.5 Research model

- **Screen Images:**

Spiral Model

Spiral model is one of the most important Software Development Life Cycle models, which provides support for Risk Handling. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process. The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using spiral model. The Radius of the spiral at any point represents the expenses(cost) of the project so far, and the angular dimension represents the progress made so far in the current phase. Below diagram shows the different phases of the Spiral Model.

Detail More

Agile Development Models

In earlier days Iterative Waterfall model was very popular to complete a project. But nowadays developers face various problems while using it to develop a software. The main difficulties included handling change requests from customers during project development and the high cost and time required to incorporate these changes. To overcome these drawbacks of Waterfall model, in the mid-1990s the Agile Software Development model was proposed. The Agile model was primarily designed to help a project to adapt to change requests quickly. So, the main aim of the Agile model is to facilitate quick project completion. To accomplish this task agility is required. Agility is achieved by fitting the process to the project, removing activities that may not be essential for a specific project. Also, anything that is wastage of time and effort is avoided. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer..

Detail More

build holds all the features required by the customer..

Detail More

SDLC V-Model

The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it..

Detail More

RAD Model Models

Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product. In the RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process. RAD projects follow iterative and incremental model and have small teams comprising of developers, domain experts, customer representatives and other IT resources working progressively on their component or prototype. The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable..

Detail More

Iterative Model

In the Iterative model, iterative process starts with a simple implementation of a small set of the

RAD Model Models

Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product. In the RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process. RAD projects follow iterative and incremental model and have small teams comprising of developers, domain experts, customer representatives and other IT resources working progressively on their component or prototype. The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable..

Detail More

Iterative Model

In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

Detail More

Software Engineering | Classical Waterfall Model

*Classical waterfall model is the basic **software development life cycle** model. It is very simple but idealistic. Earlier this model was very popular but nowadays it is not used. But it is very important because all the other software development life cycle models are based on the classical waterfall model.*

Classical waterfall model divides the life cycle into a set of phases. This model considers that one phase can be started after completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus the development process can be considered as a sequential flow in the waterfall. Here the phases do not overlap with each other. The different sequential phases of the classical waterfall model are shown in the below figure:

Let us now learn about each of these phases in brief details:

1. ***Feasibility Study**: The main goal of this phase is to determine whether it would be financially and technically feasible to develop the software.*
   The feasibility study involves understanding the problem and then determine the various possible strategies to solve the problem. These different identified solutions are analyzed based on their benefits and drawbacks, The best solution is chosen and all the other phases are carried out as per this solution strategy.
2. ***Requirements analysis and specification**: The aim of the requirement analysis and specification*

2. ***Requirements analysis and specification**: The aim of the requirement analysis and specification*
phase is to understand the exact requirements of the customer and document them properly. This phase consists of two different activities.

- ***Requirement gathering and analysis:** Firstly all the requirements regarding the software are gathered from the customer and then the gathered requirements are analyzed. The goal of the analysis part is to remove incompleteness (an incomplete requirement is one in which some parts of the actual requirements have been omitted) and inconsistencies (inconsistent requirement is one in which some part of the requirement contradicts with some other part).*
- ***Requirement specification:** These analyzed requirements are documented in a software requirement specification (SRS) document. SRS document serves as a contract between development team and customers. Any future dispute between the customers and the developers can be settled by examining the SRS document.*

3. ***Design**: The aim of the design phase is to transform the requirements specified in the SRS document into a structure that is suitable for implementation in some programming language.*
4. ***Coding and Unit testing**: In coding phase software design is translated into source code using any suitable programming language. Thus each designed module is coded. The aim of the unit testing phase is to check whether each module is working properly or not.*
5. ***Integration and System testing**: Integration of different modules are undertaken soon after they have been coded and unit tested. Integration of various modules is carried out incrementally over a number of steps. During each integration step, previously planned modules are added to the partially integrated system and the resultant system is tested. Finally, after all the modules have been successfully integrated and tested, the full working system is obtained and system testing is carried out on this.*

5. ***Integration and System testing***: *Integration of different modules are undertaken soon after they have been coded and unit tested. Integration of various modules is carried out incrementally over a number of steps. During each integration step, previously planned modules are added to the partially integrated system and the resultant system is tested. Finally, after all the modules have been successfully integrated and tested, the full working system is obtained and system testing is carried out on this.*
   System testing consists three different kinds of testing activities as described below :

   - ***Alpha testing:*** *Alpha testing is the system testing performed by the development team.*
   - ***Beta testing:*** *Beta testing is the system testing performed by a friendly set of customers.*
   - ***Acceptance testing:*** *After the software has been delivered, the customer performed the acceptance testing to determine whether to accept the delivered software or to reject it.*
6. ***Maintainence:*** *Maintenance is the most important phase of a software life cycle. The effort spent on maintenance is the 60% of the total effort spent to develop a full software. There are basically three types of maintenance :*
   - ***Corrective Maintenance:*** *This type of maintenance is carried out to correct errors that were not discovered during the product development phase.*
   - ***Perfective Maintenance:*** *This type of maintenance is carried out to enhance the functionalities of the system based on the customer's request.*
   - ***Adaptive Maintenance:*** *Adaptive maintenance is usually required for porting the software to work in a new environment such as work on a new computer platform or with a new operating system.*

Advantages of Classical Waterfall Model

Classical waterfall model is an idealistic model for software development. It is very simple, so it can

- *Phases in this model are processed one at a time.*
- *Each stage in the model is clearly defined.*
- *This model has very clear and well undestood milestones.*
- *Process, actions and results are very well documented.*
- *Reinforces good habits: define-before- design, design-before-code.*
- *This model works well for smaller projects and projects where requirements are well understood.*

Drawbacks of Classical Waterfall Model

Classical waterfall model suffers from various shortcomings, basically we can't use it in real projects, but we use other software development lifecycle models which are based on the classical waterfall model. Below are some major drawbacks of this model:

- ***No feedback path:*** *In classical waterfall model evolution of a software from one phase to another phase is like a waterfall. It assumes that no error is ever committed by developers during any phases. Therefore, it does not incorporate any mechanism for error correction.*
- ***Difficult to accommodate change requests:*** *This model assumes that all the customer requirements can be completely and correctly defined at the beginning of the project, but actually customers' requirements keep on changing with time. It is difficult to accommodate any change requests after the requirements specification phase is complete.*
- ***No overlapping of phases:*** *This model recommends that new phase can start only after the completion of the previous phase. But in real projects, this can't be maintained. To increase the efficiency and reduce the cost, phases may overlap.*

Mọi góp ý, thắc mắc xin liên hệ: Phòng dịch vụ sinh viên: Email: dichvusinhvien@fe.edu.vn. Điện thoại: (024)7308.13.13
© Powered by FPT University | CMS | library | books24x7

Web Selling Online

You Can Use Model:

+ Waterfall model
+ SDLC V-Model

And Use Management Software :

+ Jira
+ Trello
+ MyXteam

Show Detail Model

Building Management System

You Can Use Model:

+ Agile Development Models
+ Spiral Model

And Use Management Software :

+ Jira
+ Trello
+ MyXteam

Show Detail Model

Attendance System By AI

You Can Use Model:

+ Classical Waterfall Model
+ Iterative Model

And Use Management Software :

+ Jira
+ Trello
+ MyXteam

Show Detail Model

Navigation and Navigation System

You Can Use Model:

+ Spiral Model

And Use Management Software :

+ Jira
+ Trello
+ MyXteam

Show Detail Model

Study Encryption Algorithms

You Can Use Model:

+ Iterative Model

And Use Management Software :

+ Jira
+ Trello
+ MyXteam

Show Detail Model

Build a Service Receiving and Management

You Can Use Model:

+ Agile Development Models
+ Spiral Model

And Use Management Software :

+ Jira
+ Trello
+ MyXteam

Show Detail Model

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Research model	Click to “About”	No	Show About model page
2	Research model	Click to “Tutorial model”	No	Show Tutorial model page
3	View detail about model	Click to button “Detail more”	No	Show detail about model
4	View detail tutorial model	Click to button “Show detail model”	No	Show detail tutorial model

4.5.1.6. Show list supervisors and booking

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Booking	Click to button “Supervisor 1”	No	Display notification confirm
2	Booking	Click to button “Supervisor 2”	No	Display notification confirm
3	Confirm booking	Click to button “Ok”	No	Display notification booking successful
4	Confirm booking	Click to button “Huy”	No	Keep the list supervisors page

4.5.1.7. View GUIDE

- **Screen Images**

Bước 1 : Access vào mình hình register capstone

Click vào "Create Capstone" như hình để tạo capstone

Bước 1 : Access vào Jira chứa Capstone của bạn

4.5.1.8. Upload file

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Upload file	Click to choose File	No	Show file
2	Post	Click to post	No	Navigate to forum
3	Reset post	Click to reset	No	Re-post

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
7	Title	Title post	No	Yes	Text	String	120
8	Description	Description post	No	Yes	Text	String	120

4.5.1.9. Send mail

- **Screen Images**

4.5.1.10 Chat

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	View group chat to chat	Click to icon	No	Navigate to group chat
2	Chat	Click to button “Send”	No	Display content chat

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
3	Content	Content chat	No	Yes	Text	String	120

4.5.1.11. Create chat

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Create Group for chatting	Click to send	No	Send message to group

4.5.1.12. Confirm join team feature

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Confirm join Team feature	Click to button "Confirm"	No	Display confirm Dialog
2	Reject	Click button "Reject"	No	Display confirm Dialog

4.5.1.13 Edit profile

- **Screen Images:**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Edit profile	Click to button "Edit profile"	No	Navigate to Profile page
2	Upload image	Click button "Upload"	No	Display folder that you choose image
3	Update profile	Click button "Update"	No	Display notification Update successful

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
4	Description	describe hobbies, forte	No	Yes	Text	String	120

5	Phone	Phone	No	Yes	Text	String	120
6	Date of birth	Date of birth	No	Yes	Text	Date	
7	Address	Address	No	Yes	Text	String	120

4.5.1.14. Notification component

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	View Notification	Click to name Of notification	No	Navigate to Notification detail page
2	View detail notification	Click to “view”	No	Navigate to detail information notification page
3	View list Notification	Click to “more”	No	Navigate to list Notification page

4.5.1.15. Comment

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Comment	Click to send	No	Send comment to Post

4.5.1.16. Daily report to member to leader

- **Screen Images**

List Report

Report by	Title	Created date
Khanh	sdsda	2020-11-08 03:27:47.0
Khanh	asdsadasd	2020-11-08 03:11:47.0
Khanh	1231212	2020-11-08 02:39:17.0
Khanh	12312312	2020-11-08 02:36:25.0
Khanh	eqweqw	2020-11-08 02:29:07.0
Duc	sadasdasd	2020-11-08 02:28:08.0
Khanh	w	2020-11-08 02:15:00.0
Khanh	sdsda	2020-11-08 03:27:47.0
Khanh	asdsadasd	2020-11-08 03:11:47.0
Khanh	1231212	2020-11-08 02:39:17.0

Report Detail

Sdsda

Item Name	Plan	Actual	Progress	Issue	Cover plan
Today Task					
Next Task					

Write a comment..

Send

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Add file	Click to choose File	No	Show file
2	Report	Click to submit	No	Navigate to List-report page

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Title	Title report	No	Yes	Text	String	N/A
2	Content	Content report	No	Yes	Text	String	N/A

4.5.1.17. Weekly report to leader to supervisor

- **Screen Images**

List Report

Report by	Title	Created date
Khanh	sdsda	2020-11-08 03:27:47.0
Khanh	asdsadasd	2020-11-08 03:11:47.0
Khanh	1231212	2020-11-08 02:39:17.0
Khanh	12312312	2020-11-08 02:36:25.0
Khanh	eqweqw	2020-11-08 02:29:07.0
Duc	sadasdasd	2020-11-08 02:28:08.0
Khanh	w	2020-11-08 02:15:00.0
Khanh	sdsda	2020-11-08 03:27:47.0
Khanh	asdsadasd	2020-11-08 03:11:47.0
Khanh	1231212	2020-11-08 02:39:17.0

Report Detail

Sdsda

Item Name	Plan	Actual	Progress	Issue	Cover plan
Today Task					
Next Task					

Write a comment..

Send

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Add file	Click to choose File	No	Show file
2	Report	Click to submit	No	Navigate to List-report page

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Title	Title report	No	Yes	Text	String	N/A
2	Content	Content report	No	Yes	Text	String	N/A

4.5.1.18 Manage user

- **Screen Images**

SE05594	khangnmse0004	Ngo Manh Khang	Male	2020-12-13 05:20:31.0	khangnmse0004@gmail.com		eligible_capstone	Fall2020	FU-Hoa Lac	
SE05515	kiendtse05515	Dong Trung Kien	Male	2020-12-13 05:11:41.0	kiendtse055151@fpt.edu.vn		eligible_capstone	Fall2020	FU-Hoa Lac	
SE05126	datnqse05126	Nguyen Quoc Dat	Male	2020-12-13 05:10:41.0	datnqse051261@fpt.edu.vn		eligible_capstone	Fall2020	FU-Hoa Lac	
SE05116	manhpxse05016	Pham Xuan Manh	Male	2020-12-13 05:09:44.0	manhpxse050161@fpt.edu.vn		eligible_capstone	Fall2020	FU-Hoa Lac	

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Search capstone project	Click “Search” button	No	Show list capstone project

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Lengt
1	ID	ID	No	Yes	Text	Integer	N/A
2	Username	Username	No	Yes	Text	String	N/A
3	Full Name	Full Name	No	Yes	Text	String	N/A
4	Gender	Gender	No	Yes	Text	String	N/A
5	Created date	Created date	No	Yes	Text	Date	N/A

6	Email	Email	No	Yes	Text	String	N/A
7	Capstone Project	Capstone Project	No	Yes	Text	String	N/A
8	Status	Status	No	Yes	Text	String	N/A
9	Semester	Semester	No	Yes	Text	Integer	N/A
10	Site	Site	No	Yes	Text	String	N/A

4.5.1.19. View information, status of list student

- **Screen Images**

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	User Name	User Name	No	Yes	Text	String	N/A
2	Capstone Project	Capstone Project	No	Yes	Text	String	N/A
3	Address	Address	No	Yes	Text	String	N/A
4	Status	Status	No	Yes	Text	String	N/A
5	Birth date	Birth date	No	Yes	Text	Date	N/A
6	Description	Descriptio n	No	Yes	Text	String	N/A
7	Email	Email	No	Yes	Text	String	N/A
8	Gender	Gender	No	Yes	Text	String	N/A
9	Phone	Phone	No	Yes	Text	INT	N/A

4.5.1.20. View and edit the list of registered project

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Choose all profession	Click to all profession	No	Show professions
2	Choose all profession	Click to all status	No	Show statuses
3	Search name project	Click to search	No	Show name project
4	Approve Project	Click to approve	No	Approve project
5	Reject project	Click to reject	No	Reject project
6	See next page	Click to next	No	Show next page
7	Edit project	Click to “v”	No	Edit project
8	Add Student	Click to add	No	Add student
9	Cancel Student	Click to cancel	No	Cancel student

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Name	Name project	No	Yes	Text	String	N/A
2	Description	Descriptio n project	No	Yes	Text	String	N/A
3	Name-vi	Name Vietnam project	No	Yes	Text	String	N/A
4	Name-Abbreviation	Name Abbreviati on	No	Yes	Text	String	N/A
5	Status	Status	No	Yes	Text	String	N/A

4.5.1.21. View list capstone project

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Choose all profession	Click to all profession	No	Show professions
2	Choose all profession	Click to all status	No	Show statuses
3	Search name project	Click to search	No	Show name project
4	Approve Project	Click to approve	No	Approve project
5	Reject project	Click to reject	No	Reject project
6	See next page	Click to next	No	Show next page

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Name	Name project	No	Yes	Text	String	N/A
2	Description	Descriptio n project	No	Yes	Text	String	N/A
3	Name-vi	Name Vietnam project	No	Yes	Text	String	N/A
4	Name-Abbreviation	Name Abbreviati on	No	Yes	Text	String	N/A
5	Status	Status	No	Yes	Text	String	N/A

4.5.1.22. View detail project

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Add student	Click to add student	No	Show information of add student
2	Cancel	Click to cancel	No	Cancel
3	Delete project	Click to “x”	No	Delete project

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Role	Role	No	Yes	Text	String	N/A
2	First Name	First Name	No	Yes	Text	String	N/A

3	Last Name	Last Name	No	Yes	Text	String	N/A
4	Phone	Phone	No	Yes	Text	INT	N/A
5	Email	Email	No	Yes	Text	String	N/A
6	Status	Status	No	Yes	Text	String	N/A

4.5.1.23 Manage Capstone Project

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	View detail project	Click to icon	No	Display detail project
2	Accept project	Click to icon V or button “Approve Selected”	No	Display notification “Do you want approve project”
3	Reject project	Click to icon X or button “Reject Selected”	No	Display notification “Do you want reject project”
4	Search project	Click to button “Search”	No	Display project
5	Choose Profession	Click to dropdown “Profession”	No	Display list profession

6	Choose Status	Click to dropdown "Status"	No	Display list status

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Name project	Name project	No	Yes	Text	String	N/A
2	Subject Code	Subject Code	No	Yes	Text	String	N/A
3	Name	Name	No	Yes	Text	String	N/A
4	Name-Vi	Name Vietnamese	No	Yes	Text	String	N/A
5	Name changing	Name changing	No	Yes	Text	String	N/A
6	Name changing vi	Name changing Vietnamese	No	Yes	Text	String	N/A
7	Name abbreviation	Name abbreviation	No	Yes	Text	String	N/A
8	Description	Description	No	Yes	Text	String	N/A
9	Status	Status	No	Yes	Text	String	N/A

4.5.1.24 View task detail of Capstone Project

- **Screen Images**

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Lengt
1	Summary	Summary	No	Yes	Text	String	N/A
2	Start Date	Start Date	No	Yes	Text	Date	N/A
3	End Date	End Date	No	Yes	Text	Date	N/A
4	Time Spent	Time Spent	No	Yes	Text	String	N/A
5	Status	Status	No	Yes	Text	String	N/A
6	Assignee	Assignee	No	Yes	Text	String	N/A
7	Week	Week	No	Yes	Text	Integer	N/A

4.5.1.25. Dashboard statistical of project

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Search user of project	Click to button “Search”	No	Display dashboard
2	Choose semester	Click to dropdown “Semesters”	No	Display list semester
3	Choose site	Click to dropdown “Sites”	No	Display list sites

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Name project	Name project	No	Yes	Text	String	N/A
2	Search user of project	Search user of project	No	Yes	Text	String	N/A

4.5.1.26. View information of user

- **Screen Images**

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	User Name	User Name	No	Yes	Text	String	N/A
2	Capstone Project	Capstone Project	No	Yes	Text	String	N/A
3	Address	Address	No	Yes	Text	String	N/A

4	Status	Status	No	Yes	Text	String	N/A
5	Birth date	Birth date	No	Yes	Text	Date	N/A
6	Description	Descriptio n	No	Yes	Text	String	N/A
7	Email	Email	No	Yes	Text	String	N/A
8	Gender	Gender	No	Yes	Text	String	N/A
9	Phone	Phone	No	Yes	Text	INT	N/A

4.5.1.27. Approve capstone project in the project registration stage

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Choose all profession	Click to all profession	No	Show professions
2	Choose all profession	Click to all status	No	Show statuses
3	Search name project	Click to search	No	Show name project
4	Approve Project	Click to approve	No	Approve project
5	Reject project	Click to reject	No	Reject project
6	See next page	Click to next	No	Show next page

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Name	Name project	No	Yes	Text	String	N/A
2	Description	Descriptio n project	No	Yes	Text	String	N/A
3	Name-vi	Name Vietnam project	No	Yes	Text	String	N/A
4	Name-Abbreviation	Name Abbreviati on	No	Yes	Text	String	N/A
5	Status	Status	No	Yes	Text	String	N/A

4.5.1.28 Change pass / fail results

- **Screen Images**

Forum
Capstone Project
Supervisors
Manage User
Notifications
History
Guide
Do Duc
Capstone Project
All Profession
Eligible Defence By Supervisor
Name Project
Search
Click here
Approve Selected
Reject Selected
SUBJECT CODE
NAME
NAME_VI
NAME CHANGING
NAME CHANGING VI
NAME_ABBREVIATION
DESCRIPTION
STATUS
SWP_19
asd
asd
asd
Eligible Defence By Supervisor
Copyright © 2020 FPT

Capstone Project
All Profession
Eligible Defence By Supervisor
Name Project
Search
Approve Selected
Reject Selected
Click here set to pass result
Click here set to fail result
SUBJECT CODE
NAME
NAME_VI
NAME CHANGING
NAME CHANGING VI
NAME_ABBREVIATION
DESCRIPTION
STATUS
SWP_6
Parking Management
Quản lí gửi xe
PMCPM
FPT Capstone Project Managment
Eligible Defence By Supervisor
SWP_7
App Map Directions
Ứng dụng chỉ đường
AMD
Ứnứng dụng chỉ đường trên điện thoại di động
Eligible Defence By Supervisor
SWP_8
FPT capstone
Fpt capstone
Fpt capstone
Eligible Defence By Supervisor
JS_10
Management System for Small/Family Business
Hệ thống quản lý cho doanh nghiệp nhỏ / gia đình
MSB
FPT Capstone Project Management
Eligible Defence By Supervisor
JS_12
Lmemo
Lmemo
LMM
FPT Capstone Project Management
Eligible Defence By Supervisor
Copyright © 2020 FPT

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Choose all profession	Click to all profession	No	Show professions
2	Choose all profession	Click to all status	No	Show statuses
3	Search name project	Click to search	No	Show name project
4	Approve Project	Click to approve	No	Approve project
5	Reject project	Click to reject	No	Reject project
6	See next page	Click to next	No	Show next page

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Name	Name project	No	Yes	Text	String	N/A
2	Description	Descriptio n project	No	Yes	Text	String	N/A
3	Name-vi	Name Vietnam project	No	Yes	Text	String	N/A
4	Name-Abbreviation	Name Abbreviati on	No	Yes	Text	String	N/A
5	Status	Status	No	Yes	Text	String	N/A

4.5.1.29 Create notification

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Choose all profession	Click to all profession	No	Show professions
2	Choose all profession	Click to all status	No	Show statuses
3	Search name project	Click to search	No	Show name project
4	Approve Project	Click to approve	No	Approve project
5	Reject project	Click to reject	No	Reject project
6	See next page	Click to next	No	Show next page

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Name	Name project	No	Yes	Text	String	N/A
2	Description	Descriptio n project	No	Yes	Text	String	N/A
3	Name-vi	Name Vietnam project	No	Yes	Text	String	N/A

4	Name-Abbreviation	Name Abbreviation	No	Yes	Text	String	N/A
5	Status	Status	No	Yes	Text	String	N/A
6	Title	Title	No	Yes	Text	String	N/A
7	Content	Content	No	Yes	Text	String	N/A

4.5.1.30 View History Record

- **Screen Images**

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Content	Content	No	Yes	Text	String	N/A
2	Create date	Create date	No	Yes	Text	Date	N/A
3	Last modified date	Last modified date	No	Yes	Text	Date	N/A
4	Capstone project	Capstone project	No	Yes	Text	String	N/A

5	Notification	Notification	No	Yes	Text	String	N/A
6	Post	Post	No	Yes	Text	String	N/A
7	Report	Report	No	Yes	Text	String	N/A

4.5.1.31 Change name capstone project

- **Screen Images**

Forum
Capstone Project
Supervisors
Reports
Model
Nguyen Khanh
Project Information
Change Name
Edit
Click here
Information
Name Project
Google Map
Name Project Other language
Google Map
Name Project VietNamese
Google Map
Name Project Abbreviation
Google Map
Profession
SE
Program
Google Map
Specialty
Google Map
Document
Google Map
Description
Status
Pending By Supervisor
Member
User
Role
Status
khanhnnse050452
Leader
Confirmed
Global News
FU-HL: Thông báo điểm thi lần 1 môn VOV141 và điểm thi lần 2 HK Fall 2020 (No 1)
2020-12-06 20:15:58.0
FU-HL: Thông báo DSSV thi lần 2 môn VOV114 HK Fall 2020.
2020-12-06 20:15:39.0
FU-HL: Thông báo về việc gia hạn đăng ký mua BHYT đợt 4 năm 2020
2020-12-06 20:15:14.0
THÔNG BÁO NỘP HỌC PHÍ KỲ SPRING 2021
2020-12-06 20:14:58.0
FU-HL: Thông báo lịch thi cuối kỳ lần 1 và lần 2 môn JPD112 HK Fall 2020.
2020-12-06 20:12:08.0
FU-HL: VV: Mở thêm lớp LAB231.E1. BL5
2020-12-06 20:11:48.0
More...
Notification
SE050457 has rejected your team
2020-12-22 23:36:53.0
SE050455 has joined your team
2020-12-22 23:36:13.0
SE050454 has joined your team
2020-12-22 23:34:45.0
SE050453 has joined your team
2020-12-22 23:30:13.0
More...

Warning:
You can only edit information without submitting this form for Training Department.
Name Project
Google Map
Name Project VietNamese
Google Map
Điền thông tin cần sửa
Name Project Other language
Google Map
Name Project Abbreviation
Google Map
Program Project
Google Map
Specialty Project
Google Map
Document Project
Google Map
Description Project
Update
Click here để update
FU-HL: Thông báo DSSV thi lần 2 môn VOV114 HK Fall 2020.
2020-12-06 20:15:39.0
FU-HL: Thông báo về việc gia hạn đăng ký mua BHYT đợt 4 năm 2020
2020-12-06 20:15:14.0
THÔNG BÁO NỘP HỌC PHÍ KỲ SPRING 2021
2020-12-06 20:14:58.0
FU-HL: Thông báo lịch thi cuối kỳ lần 1 và lần 2 môn JPD112 HK Fall 2020.
2020-12-06 20:12:08.0
FU-HL: VV: Mở thêm lớp LAB231.E1. BL5
2020-12-06 20:11:48.0
More...
Notification
SE050457 has rejected your team
2020-12-22 23:36:53.0
SE050455 has joined your team
2020-12-22 23:36:13.0
SE050454 has joined your team
2020-12-22 23:34:45.0
SE050453 has joined your team
2020-12-22 23:30:13.0
More...

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Change name	Click to Change name	No	Change game
2	Edit	Click to Edit	No	Navigate to Edit capstone project page
3	Update	Click to Update	No	Display successful notification

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
7	Name Project	Name of project	No	Yes	Text	String	120
8	Name project other language	Name project other language	No	Yes	Text	String	120
9	Name project Vietnamese	Name project Vietnam	No	Yes	Text	String	120
10	Name project Abbreviation	Name project Abbreviation	No	Yes	Text	String	120
11	Profession	Profession	No	Yes	Text	String	120

	project	Project					
12	Program project	Program Project	No	Yes	Text	String	120
13	Specialty project	Specialty Project	No	Yes	Text	String	120
14	Document project	Document Project	No	Yes	Text	String	120
12	Member	Member add	No	Yes	Text	String	120
13	Description project	Descriptio n Project	No	Yes	Text	String	120

4.5.1.32 View list post

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Search for title	Click to Search	No	Search for title
2	Add post	Click to Add post	No	Display screen add post
3	Send comment	Click to send	No	Comment

4.5.1.33 Get comment

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Search for title	Click to Search	No	Search for title
2	Add post	Click to Add post	No	Display screen add post
3	Send comment	Click to send	No	Comment

4.5.1.34 Add post

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Add post	Click to add post	No	Show file
2	Post	Click to reset	No	Navigate to forum
3	Reset post		No	Re-post

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
7	Title	Title post	No	Yes	Text	String	120
8	Description	Description post	No	Yes	Text	String	120

4.5.1.35 Edit post

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Add post	Click to add post	No	Show file
2	Post	Click to reset	No	Navigate to forum
3	Reset post		No	Re-post

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
7	Title	Title post	No	Yes	Text	String	120
8	Description	Description post	No	Yes	Text	String	120

4.5.1.36 Delete post

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Add post	Click to add post	No	Show file
2	Delete	Click to Delete	No	Display successful notification
3	Cancel	Click to Cancel	No	Back to the page

4.5.1.37 Search code

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Search for title	Click to Search	No	Search for title
2	Add post	Click to Add post	No	Display screen add post
3	Send comment	Click to send	No	Comment

4.5.1.38 Add user

- **Screen Images**

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Choose all profession	Click to all profession	No	Show professions
2	Choose all profession	Click to all status	No	Show statuses
3	Search name project	Click to search	No	Show name project
4	Approve Project	Click to approve	No	Approve project
5	Reject project	Click to reject	No	Reject project
6	See next page	Click to next	No	Show next page
7	Submit	Click to submit	No	Display successful notification

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Name	Name project	No	Yes	Text	String	N/A
2	Description	Descriptio n project	No	Yes	Text	String	N/A
3	Name-vi	Name Vietnam project	No	Yes	Text	String	N/A
4	Name-Abbreviation	Name Abbreviati on	No	Yes	Text	String	N/A
5	Status	Status	No	Yes	Text	String	N/A
6	Title	Title	No	Yes	Text	String	N/A
7	Content	Content	No	Yes	Text	String	N/A

4.5.1.39: Approve capstone project in the changing name stage

- **Screen Images**

Forum
Capstone Project
Supervisors
Manage User
Notifications
History
Guide
Do Duc
Capstone Project
Click here
All Profession
All Status
Name Project
Search
Approve Selected
Reject Selected
All Status
Registering Capstone
Registed Capstone
Approved By Supervisor
Approved By Training Department
Approved By Head
Doing
Not Eligible Defence
Eligible Defence
Rejected
Changing Name
Accepted Renaming By Supervisor
SUBJECT CODE
NAME CHANGING
NAME CHANGING VI
NAME_ABBREVIATION
DESCRIPTION
STATUS
SWP_19
asd
Doing
SWP_18
asd
asd
Doing
SWP_1
FPT Casptone Project Managment
Quản lý đồ án
FCPM
FPT Casptone Project Managment
Registed Capstone
SWP_2
FPT Casptone Project Managment 2
Quản lý đồ án 2
FCPM
FPT Casptone Project Managment 2
Registering Capstone
SWP_3
The web sells clothes online
trang bán quần áo trên mạng
sco
bán hàng online
Doing
SWP_4
App Map Directions
Ứng dụng chỉ đường
AMD
bản đồ chỉ đường
Accepted Renaming By Supervisor
SWP_5
salary calculator software
phần mềm tính tiền lương
SCC
Phầm mềm tính tiền lương cho công ty nhà nước
Doing
SWP_6
Take attendance with facial recognition
Điểm danh bằng nhận diện khuôn mặt
TAWFR
Điểm danh thông minh cho các công ty công nghệ
Doing

Forum
Capstone Project
Supervisors
Manage User
Notifications
History
Guide
Do Duc
Capstone Project
All Profession
Accepted Renaming By Supervisor
Name Project
Search
Approve Selected
Reject Selected
SUBJECT CODE
NAME
NAME_VI
NAME CHANGING
NAME CHANGING VI
NAME_ABBREVIATION
DESCRIPTION
STATUS
SWP_4
App Map Directions
Ứng dụng chỉ đường
AMD
bản đồ chỉ đường
Accepted Renaming By Supervisor
Copyright © 2020 FPT
Click here

Button/ Hyperlinks

No	Function	Description	Validation	Outcome
1	Choose all profession	Click to all profession	No	Show professions
2	Choose all profession	Click to all status	No	Show statuses
3	Search name project	Click to search	No	Show name project
4	Approve Project	Click to approve	No	Approve project
5	Reject project	Click to reject	No	Reject project
6	See next page	Click to next	No	Show next page

Fields

No	Field Name	Description	Read Only	Mandatory	Control Type	Data Type	Length
1	Name	Name project	No	Yes	Text	String	N/A
2	Description	Descriptio n project	No	Yes	Text	String	N/A
3	Name-vi	Name Vietnam project	No	Yes	Text	String	N/A
4	Name-Abbreviation	Name Abbreviati on	No	Yes	Text	String	N/A
5	Status	Status	No	Yes	Text	String	N/A

5. Software Test Documentation

5.1. Introduction

5.1.1 Purpose

- The primary purpose of this chapter is to detect and prevent defects which may be created by developers while developing the software and this may lead to software failures. Through testing, system defects can be discovered and fixed, then help to build better software quality. Using the contents of this chapter, developers and testers can work together to ensure that the system is thoroughly tested and performs functions as intended.

  - ✓ Scope of Testing.
  - ✓ Test Plan
  - ✓ Test Case
  - ✓ Test Report

5.1.2 Scope of testing

5.1.2.1 States of testing

The testing process is divided into 3 phases: Unit testing, Integration testing and System testing.

States of testing	Description
Unit test	Unit Tests are performed to guarantee that individual units of source code work well. In this phase, a unit can refer to a function, individual program or even a procedure, and a White-box Testing method is usually used to get the job done
Integration test	Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. Putting integration tests of all modules into the application is performed by CPM tester to check if the combined units were working as expected.
System test	System testing is performed to monitor and evaluate the behaviour of a completely integrated software product or software system, based on predefined specification and functional requirements. The goal at this level is to evaluate whether the system has

complied with all of the outlined requirements and to see that it meets Quality Standards.

Table 5.1 Scope of testing

5.1.2.2 Types of testing

The following types of testing are performed:

- Function testing:

  - ✓ Unit test
  - ✓ Integration test
  - ✓ System test

5.1.2.3 Range of testing

All features described in the System Requirement Specification are tested.

5.2. Test Plan

5.2.1 Testing tools and Environment

5.2.1.1 Testing tools

Tool	Purpose
Junit	Used to view app, bug login, etc..
Excel	Used to manage test cases
Google Chrome	Used to view the web page, bug logging page
IntelliJ	Used to view app, bug login, etc..
Jira	Used to manage bugs

Table 5.2 Testing tools

5.2.1.1 Testing environment

Type of testing	Software	Hardware
Unit test	Junit IntelliJ IDEA 2020.2.3	Dell Window 10 Intel core i7 RAM 8G 1T Disk Space
Integration test	Junit Google Chrome Microsoft Office 2016	Dell Window 10 Intel core i7 RAM 8G 1T Disk Space
System test	Google Chrome	Dell Window 10 Intel core i7 RAM 8G 1T Disk Space

Table 5.3 Testing environment

5.2.2 Resources and Responsibilities

ID	Resource	Responsibilities
1	Test Leader	• Review test cases and test reports. • Review overall quality of the project. • Define test strategy. • Define test plan, test approach.
2	Tester	Performing the actual system testing • Understand the requirements of the project. • Implement integration test following requirement. • Create test plan • Create test cases • Perform tests • Write test reports

		• Log bugs
3	Developer	Create and perform Unit Tests. • Fix bugs.

Table 5.4 Resources and Responsibilities

5.2.3 Test Strategy

5.2.3.1 Test model

Figure 5.1 Test model

More information: https://en.wikipedia.org/wiki/V-Model_(software_development)

5.2.3.2 Test types

ID	Type of test	Description
1	GUI (Graphical user interface) testing	• GUI test is used to test the user interface of the software. It will be performed fully on all screens. The target is to cover the verification of mocks defined before. It also includes fields: ✓ Test whether the interface is displayed correctly or not. ✓ Test if you can perform the intended functionality of the application using the GUI. ✓ Test if the error messages are displayed correctly. ✓ Test if the font used in the system is readable. ✓ Test the fonts and warning messages. ✓ Test if the images are clear or not. ✓ Test if that the images are properly aligned
2	Regression testing	• Regression testing is necessary, even when making very small changes to the code. • Regression test ensures that the bugs are fixed. • Regression testing ensures bugs fixing does not affect other parts of the system.
3	Functional testing	• Functional test is used to test for each logical path of use case. • Tests if functions of the system work as intended on the CPM system. • Functional testing is a type of black-box testing and as such, internal structure is rarely considered.

Table 5.4 Resources and Responsibilities

5.2.3.3 State of testing

Test Type	States of testing		
	Unit Test	Integration Test	System Test
Unit test	✓		
GUI testing		✓	✓
Functional testing		✓	✓

Table 5.5 States of testing

5.2.3.4 Test schedule

Table below are the Test Schedule for CPM Project. That consist of two round:

Test Schedule	Round 1		Round 2	
	Start Date	End Date	Start Date	End Date
Unit Test	10/10	10/11		
Integration test	27/10	15/11	16/11	22/11
System test				

5.2.3.5 Deliverable

Table below is the Deliverables for FCPM Project:

Deliverables	Responsibilities	Completion date
Test plan	Test Leader	09-10-2020
Test cases	Testers	20-10-2020
Test case review	Testers + Test Leader	02-11-2020
Test data(Data Set)	Testers	19-11-2020
Defect report	All members	02-12-2020
Final test report	Test Leader	10-12-2020

5.3. Test Approach

5.3.1 Unit test

- Unit Test is a method of testing software by breaking the source code into separate parts and testing each part. Unit testing is done by developers. FCPM developer team embraces this feature to gain the following advantages:

- ✓ Ensuring code quality
- ✓ Early error detection
- ✓ Reducing the Number of Bugs
- ✓ Automation tests can be run as frequently as required.
- ✓ Saving development time.
- ✓ Reducing product costs
- ✓ Improving code design, especially in test-driven developments
- ✓ Being able to review each other's code, the teamwork is more cohesive and Agile.
- ✓ Providing clear documentation
- ✓ It is easier to maintain than Front-end tests which are difficult to maintain with the short release cycles and frequent changes and with a complex system.
- ✓ Reducing the cost of a resource to corresponding Front-end testing.

5.3.2 Integration and System test

- Integration and system tests are done by testers to ensure that combined units work correctly and that the system as a whole function as intended. Each test case is tested using Google Chrome, Android Devices. FCPM define that a test case is:

- ✓ Verifying the Layouts
- ✓ Testing user access privileges.
- ✓ Testing all the network features, like a request is passed over a HTTPS protocol or not.
- ✓ Verifying menu items and their functionality.
- ✓ Adherence to coding standards and practices.
- ✓ Testing whether after an action is performed, cleanup of data is performed. Database operations like insertion, deletion and updates should be done with an intimidation to the user.

- GUI testing is also done during this process to ensure that elements load and function correctly, text is readable, and the app interface looks good in various devices sizes.

- Test case details are described in the file: Report5_System Test Case.xlsx

5.4. Test Report

5.4.1 Unit test

```java
@RunWith(SpringRunner.class)
@SpringBootTest
public class UserServiceTest {
    @MockBean
    private UserRepository userRepository;
    @MockBean
    private UserRolesRepository userRoleRepository;
    @Autowired
    private UserService userService;

    @Test
    public void testfindById() {
        String id="1";
        Users user= new Users();
        user.setId("1");
        Optional<Users> optionausers = Optional.of(user);
        when(userRepository.findById(id)).thenReturn(optionausers);
        assertEquals(user, userService.findById(id));
    }
    @Test
    public void testgetAllUser() {

    List<Users> list = new ArrayList<~>();


    when(userRepository.findAll()).thenReturn(list);

    assertEquals(list, userService.getAllUser());
```

Figure: Unit test case sample UserServiceTest

The contents of the Unit Test Report are shown in the table below:

Test Case	Pass	Fail
Unit Test- TestfindByID	16	**0**
Unit Test- TestgetAllUser	26	0
Unit Test- TestdeleteeUserSuccess	18	0
Unit Test- TestGetAllDepartmentByID	19	0
Unit Test- TestSaveUserSuccess	74	0
Unit Test- TestFindAll	12	0
Unit Test- TestGetUserByRoleID	15	0

Unit Test- TestFindbyUsername	27	0
Unit Test- TestFindbyEmail	66	0
Total all test case	**273**	**0**

For each of the tested unit, controller are tested.

Figure: Unit test case completed

5.4.2 System test

No	Module code	Pass	Fail	Untested	N/A	Number of test cases
1	Login	7	0	0	0	7
2	Forum	38	0	0	0	38
3	Register	32	0	0	0	32
4	Supervisors	18	0	0	0	18
5	List Report	14	0	0	0	14
6	Notification	13	0	0	0	13
7	Model	47	0	0	0	47
8	My Report	14	0	0	0	14
9	Create Report	35	0	0	0	35
10	Dashboard	103	0	0	0	103
11	User Information	11	0	0	0	11
12	Manage Capstone Project	132	0	0	0	132
13	My Capstone	17	0	0	0	17
14	Manage User	57	0	0	0	57
15	Header	19	0	0	0	19
	Sub total	557	0	0	0	557

Test coverage	**100.00**	%
Test successful coverage	**100.00**	%

Test case list:

ID	Test Case Description	Test Case Procedure	Expected Output	Inter-test case Dependence	Result	Test date	Note
	Create report Member						
001	Cấu trúc màn hình, bố trí object màn hình, định nghĩa style	Access vào màn hình "Create Report "	Confirm :Layout màn hình đúng với mô tả ở mockup	Đang ở màn hình Create Report	Pass	11/19/2020	
002	Header-Footer	Access vào màn hình "Create Report "	Confirm :Phần header- footer hiển thị đầy đủ	Đang ở màn hình Create Report	Pass	11/19/2020	
003	Độ phân giải thiêu chuẩn trên window và tất cả cá thiết bị di động	Access vào màn hình "Create Report "	Confirm :Màn hình hiển thị không bị vỡ layout, không bị mất chữ	Đang ở màn hình Create Report	Pass	11/19/2020	
004	Resize màn hình	Access vào màn hình "Create Report "	Confirm :Màn hình hiển thị không bị vỡ layout, không bị mất chữ	Đang ở màn hình Create Report	Pass	11/19/2020	
005	Create report	Access vào màn hình Create report " Input title Input content Click "Submit"	Confirm :Tạo report thành công	Đang ở màn hình Create Report	Pass	11/25/2020	
006	Save report	Access vào màn hình Create report " Click vào biểu tượng save	Confirm : Report được lưu lại	Đang ở màn hình Create Report	Pass	11/25/2020	
007	Format report	Access vào màn hình Create report " Click vào biểu tượng Form Nhập các trường cần thiết để tạo form Click vào Ok để tạo form	Confirm : Report được format	Đang ở màn hình Create Report	Pass	11/25/2020	
008	Preview report	Access vào màn hình Create report " Click vào biểu tượng Preview	Confirm : Report được mở ra tab preview report	Đang ở màn hình Create Report	Pass	11/25/2020	
009	Print report	Access vào màn hình Create report " Click vào biểu tượng Print	Confirm :Mở ra màn print để Print Report	Đang ở màn hình Create Report	Pass	11/25/2020	
010	Template report	Access vào màn hình Create report " Click vào biểu tượng Template	Confirm :Mở ra màn template để chọn tem plate	Đang ở màn hình Create Report	Pass	11/25/2020	
011	Search report	Access vào màn hình Create report " Click vào biểu tượng Search Nhập ký tự cần tìm vào texbox Click vào find để tìm kiếm	Confirm :Các ký tự tìm kiếm được tô đậm màu trên màn report	Đang ở màn hình Create Report	Pass	11/26/2020	
012	Replace report	Access vào màn hình Create report " Click vào biểu tượng Search- Nhập ký tự cần thay thế vào textbox "Find what" Nhập ký tự để thay thế vào textbox "Replace with" Click vào replace để thay thế từng ký tự	Confirm :Ký tự được thay thế trong report	Đang ở màn hình Create Report	Pass	11/26/2020	
013	Replace report	Access vào màn hình Create report " Click vào biểu tượng Search- Nhập ký tự cần thay thế vào textbox "Find what" Nhập ký tự để thay thế vào textbox "Replace with" Click vào replace All để thay thế tất cả ký tự	Confirm :Ký tự được thay thế trong report	Đang ở màn hình Create Report	Pass	11/26/2020	
014	Form report	Access vào màn hình Create report " Click vào biểu tượng Form Nhập các trường cần thiết để tạo form Click vào Ok để tạo form	Confirm :Form được tạo trên màn report	Đang ở màn hình Create Report	Pass	11/26/2020	
015	Textbox report	Access vào màn hình Create report " Click vào biểu tượng Textbox -Nhập các trường cần thiết để tạo textbox Chọn kiểu textbox là Select or Required Click vào Ok để tạo textbox	Confirm :Textbox được tạo trên màn report	Đang ở màn hình Create Report	Pass	11/26/2020	
016	Radio button report	Access vào màn hình Create report " Click vào biểu tượng Radio Button Nhập các trường cần thiết để tạo textbox Chọn kiểu textbox là Select or Required Click vào Ok để tạo radio button	Confirm :Radio button được tạo trên màn report	Đang ở màn hình Create Report	Pass	11/26/2020	

Figure Test Case: Create report

ID	Test Case Description	Test Case Procedure	Expected Output	Inter-test case Dependence	Result	Test date	Note
	Manage_Capstone Head						
001	Cấu trúc màn hình, bố trí object màn hình, định nghĩa style	Access vào màn hình "Manage_Capstone Head"	Confirm :Layout màn hình đúng với mô tả ở mockup	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
002	Header-Footer	Access vào màn hình "Manage_Capstone Head"	Confirm :Phần header- footer hiển thị đầy đủ	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
003	Độ phân giải thiêu chuẩn trên window và tất cả cá thiết bị di động	Access vào màn hình "Manage_Capstone Head"	Confirm :Màn hình hiển thị không bị vỡ layout, không bị mất chữ	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
004	Resize màn hình	Access vào màn hình "Manage_Capstone Head"	Confirm :Màn hình hiển thị không bị vỡ layout, không bị mất chữ	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
005	Phân trang	Access vào màn hình "Manage Capstone" Click vào trang thứ nhất	Confirm :Hiện thị các capstone của trang thứ nhất	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
006	Phân trang	Access vào màn hình "Manage Capstone" Click vào trang thứ hai	Confirm :Hiện thị các capstone của trang thứ hai	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
007	Phân trang	Access vào màn hình "Manage Capstone" Click vào trang thứ ba	Confirm :Hiện thị các capstone của trang thứ ba	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
008	Phân trang	Access vào màn hình "Manage Capstone" Click vào Previous của phần phân trang	Confirm :Hiện thị các capstone của trang trước	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
009	Phân trang	Access vào màn hình "Manage Capstone" Click vào Next của phần phân trang	Confirm :Hiện thị các capstone của trang sau	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
010	Search	Access vào màn hình "Manage Capstone" Chọn All Profession Chọn All Status Không input name	Confirm : Tất cả các project hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
011	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn All Status Không input name	Confirm :Tất cả các project ngành SE hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
012	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status not_eligible_capstone Không input name	Confirm :Các project có status not_eligible_capstone hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
013	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status eligible_capstone Không input name	Confirm :Các project có status eligible_capstone hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
014	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status inActive_user Không input name	Confirm :Các project có status inActive_user hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
015	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status registering_capstone Không input name	Confirm :Các project có status registering_capstone hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
016	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status registed_capstone Không input name	Confirm :Các project có status registed_capstone hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
017	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status Không input name	Confirm :các project có status not_eligible_capstone hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
018	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status approve_capstone_lecturer Không input name	Confirm :các project có status approve_capstone_lecturer hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
019	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status approve_capstone_trainning Không input name	Confirm :các project có status approve_capstone_trainning hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
020	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status approve_capstone_head Không input name	Confirm :các project có status approve_capstone_head hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	
021	Search	Access vào màn hình "Manage Capstone" Chọn mã ngành SE Chọn status doing_capstone Không input name	Confirm :các project có status doing_capstone hiện ra	Đang ở màn hình Manage_Capstone Head	Pass	11/19/2020	

Figure Test Case: Manage Capstone Project

For more details about all test suites, please look at folder **"FCPM_TestCase"**

6. Software User's Manual

6.1. Installation Guide

6.2. User's Guide

6.2.1 General function

6.2.1.1 Login

Step 1: At the home page, click on "Sign in with Google" to navigate to Authenticate email FPT page

Step 2: After entering the Authenticate email FPT page, Select email FPT to login

Step 3: After login email FPT, to navigate to the forum

6.2.1.2 Logout

Step 1: Click on the username in the upper right corner of the screen, Choose logout

6.2.1.3 Register Capstone Project

Step 1: Click the Capstone Project in the top of the page, choose Create Capstone Project

Step 2: Enter information to register project

Step 3: Click button submit to register project

6.2.1.4 Add member

Step 1: Click the Capstone Project in the top of the page, choose Create Capstone Project

Step 2: Enter account of member, click button “Add”

6.2.1.5 Research model

Step 1: Access to forum, click button Model, choose About

Step 2: System display About Model

Step 3: Click button “Detail More” to view detail

Step 4: Return forum, click button Model, choose Tutorial Model

Step 5: System display Tutorial Model page to view

6.2.1.6 Show list supervisors and booking

Step 1: Click the Capstone Project in the top of the page, choose Create Capstone Project

Step 2: Enter information to register project

Step 3: Click button submit to register project

Step 4: After register successfully, system display list supervisor to booking, click button Supervisor 1

Step 5: Display notification “Are you sure you want to choose lecturer”, click “OK”

Step 6: After book supervisor 1 successfully, System display remain supervisors and don't supervisor 1 to book supervisor 2, click button "Supervisor 2"

6.2.1.7 View GUIDE

Step 1: Access forum, click GUIDE

Step 2: System display GUIDE Page

1.Dành cho Member

Bước 1 : Access vào mình hình Create Report

Click vào "Create Report" như hình để tạo Report

Bước 1 : Access vào Jira chứa Capstone của bạn

6.2.1.8 Upload file

Step 1: Click button "ADD POST"

Step 2: After click button “ADD POST”, show form to create post

Step 3: Enter the required information, click button “Post” to upload file to forum

6.2.1.9 Send mail

Step 1: Go to the email page, choose compose to send mail to register capstone project and booking supervisor

6.2.1.10 Chat

Step 1: Access forum, Click icon in the right screen

Step 2: System display group chat to chat together

6.2.1.11 Create chat

Step 1: Click icon on screen

Step 2: Show room to student chat

6.2.1.12 Confirm join team feature

Step 1: Access forum, Click Capstone Project, Choose My Capstone

Step 2: Click “Confirm” button

6.2.1.13 Edit profile

Step 1: Access forum, click account, choose “Edit Profile”

Step 2 : System display detail information of user, enter information, click “Update” to edit profile

6.2.1.14 Notification component

Step 1: Select the notifications you want to see

Step 2: Displays details of the notifications you want to see

6.2.1.15 Comment

Step 1: Enter information you want comment, click send

6.2.1.16 Daily report to member to leader

Step 1: Select the Reports section at the top of the page, then select create report

Step 2: Enter information report, then click button submit

Step 3: Display reports of yourself and team members

List Report

Report by	Title	Created date
Khanh	sdsda	2020-11-08 03:27:47.0
Khanh	asdsadasd	2020-11-08 03:11:47.0
Khanh	1231212	2020-11-08 02:39:17.0
Khanh	12312312	2020-11-08 02:36:25.0
Khanh	eqweqw	2020-11-08 02:29:07.0
Duc	sadasdasd	2020-11-08 02:28:08.0
Khanh	w	2020-11-08 02:15:00.0
Khanh	sdsda	2020-11-08 03:27:47.0
Khanh	asdsadasd	2020-11-08 03:11:47.0
Khanh	1231212	2020-11-08 02:39:17.0

Step 4: Click name of title to view detail information of report

Step 5: Display report detail

6.2.1.17 Weekly report to leader to supervisor

Step 1: Select the Reports section at the top of the page, then select create report

Step 2: Enter information report, then click button submit

Step 3: Display reports of yourself and team members

List Report

Report by	Title	Created date
Khanh	sdsda	2020-11-08 03:27:47.0
Khanh	asdsadasd	2020-11-08 03:11:47.0
Khanh	1231212	2020-11-08 02:39:17.0
Khanh	12312312	2020-11-08 02:36:25.0
Khanh	eqweqw	2020-11-08 02:29:07.0
Duc	sadasdasd	2020-11-08 02:28:08.0
Khanh	w	2020-11-08 02:15:00.0
Khanh	sdsda	2020-11-08 03:27:47.0
Khanh	asdsadasd	2020-11-08 03:11:47.0
Khanh	1231212	2020-11-08 02:39:17.0

Step 4: Click name of title to view detail information of report

Step 5: Display report detail

Report Detail

Test Rp 2

Item Name	Plan	Actual	Progress	Issue	Cover plan
Today Task					
Next Task					

Write a comment..

Send

6.2.1.18 Manage user

Step 1: Access forum, Click Manage User, choose Manage Student

Step 2: System display Student Management Page

6.2.1.19 View information, status of list student

Step 1: View information, status of list student

6.2.1.20 View and edit the list of registered project

Step 1: View and edit the list of registered project

6.2.1.21 View list capstone project

Step 1: View list capstone project

6.2.1.22 View detail project

Step 1: Access forum, click Capstone Project, choose Manage Capstone

Step 2: System display Capstone Project Page

Step 3: Click to icon as screen

Step 4: System display detail project

6.2.1.23 Manage Capstone Project

Step 1: Access forum, click Capstone Project, choose Manage Capstone

Step 2: System display Capstone Project Page

6.2.1.24 View task detail of Capstone Project

Step 1: Access Dashboard, click Detail in list project

Step 2: System display Task Detail Page

6.2.1.25. Dashboard statistical of project

Step 1: Access forum, Click to Dashboard

Step 2: System display Dashboard page

6.2.1.26 View information of user

Step 1: Access forum, click to account of user

Step 2: Display information of user

6.2.1.27 Approve capstone project in the project registration stage

Step 1: Click button “Approve” or “Reject” project

6.2.1.28 Change pass / fail results

Step 1: Access forum, Click Capstone Project, Choose Manage Capstone

Step 2: Choose status

Step 3: Click “Search” button

Step 4: Click icon as screen to pass result or fail result

6.2.1.29 Create notification

Step 1: Access forum, click Notifications, choose Create Notifications

Step 2: Enter information, click “Post” button

6.2.1.30 View History Record

Step 1: Access forum, click History, choose History List, system display History Records

6.2.1.31 Change name capstone project

Step 1: Access forum, click Capstone Project, choose My Capstone

Step 2: Click “Edit” button

Step 3: Enter information to correct, click “Update” button

6.2.1.32 View list post

Step 1: Access forum display list post

6.2.1.33 Get comment

Step1: Access forum, enter the article you want to comment on post

Step 2: Click “Send” button

6.2.1.34 Add post

Step 1: Access forum, click “Add Post” button

Step 2: Fill full the information, click “Post” button

6.2.1.35 Edit post

Step 1: Access forum, click on the icon as screen

Step 2: Edit post information, click “Post” button to edit

6.2.1.36 Delete post

Step 1: Access forum, click on the icon as screen

Step 2: Click “Delete” button

6.2.1.37 Search code

Step 1: Access forum, enter title, click “Search”

6.2.1.38 Add user

Step 1: Access forum, click Manage User, choose Other User

Step 2: Fill full the information

Step 3: Click “Submit” button

6.1.1.39: Approve capstone project in the changing name stage

Step 1: Access forum, Click Capstone Project, Choose Manage Capstone

Step 2: Choose status

Step 3: Click “Search” button

Step 4: Click icon as screen to approve