

FPT UNIVERSITY

Capstone Project Document School Feedback Management System

Group 11	
Group members	SE61897 - Nguyễn Việt Tú (Leader) SE61882 - Trần Hồ Minh Thuấn SE61745 - Hoàng Quốc Việt
Supervisor	Nguyễn Huy Hùng
Ext. Supervisor	N/A
Capstone Project code	SFMS

-Ho Chi Minh City, **8th January, 2018**-



CAPSTONE PROJECT REGISTER

Class: Duration time: from 8/1/2018 to 29/4/2018

(*) Profession: <Software Engineer> Specialty: <ES> <IS>

(*) Kinds of person make registers: Lecturer Students

1. Register information for supervisor (if have)

	Full name	Phone	E-Mail	Title
Supervisor 1	Nguyễn Huy Hùng		hungnh@fpt.edu.vn	Mr.

2. Register information for students (if have)

	Full name	Student code	Phone	E-mail	Role in Group
Student 1	Nguyễn Việt Tú	SE61897	0903123324	tu.nguyen1110@gmail.com	Leader
Student 2	Hoàng Quốc Việt	SE61745	01682301813	viethoangquoc2711@gmail.com	Member
Student 3	Trần Hồ Minh Thuấn	SE61882	01673340263	makuryuu@gmail.com	Member

3. Register content of Capstone Project

(*) 3.1. Capstone Project name:

English: School Feedback Management System

Vietnamese: Hệ Thống Quản Lý Feedback Trường Học

Abbreviation: SFMS

(*) 3.2. Main proposal content (including result and product)

a) Theory and practice (document):

- Student should apply the software development process and the UML 2.0 in modeling the system
- Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages
- Server side technique:
 - Database design, OOA, OOD, OOP, MVC, Java or .Net technology, ...
 - Apache Lucene, Elastic Search

- o Client side technique
 - HTML5, CSS, JavaScript, JQuery, Ajax
 - Mobile Platform (iOS, Android)

b) Program:

Building web application that allows school to create and manage feedbacks. Feedbacks may be used to evaluate staff, teachers, and other departments. Therefore, the system should support flexible templates that can be customized for each department, each course, each major, and even for each lecture. The following main features should be implemented:

- Support different roles
- Support default templates as well provide a way to customize template for each object
- Support create template by drag-n-drop feedback elements
- Generate reports from feedback results
- Analyze and give suggested improvement based on generated reports
- Set interval time for each feedback
- Remind users to conduct feedbacks

c) Other products:

- All of management functions of the system must be implemented to support the operating system.

4. Other comment (propose all relative thing if have)

N/A

HCM city, date 18/04/2018

Supervisor (If have)
(Sign and full name)

On behalf of Registers
(Sign and full name)

Nguyễn Huy Hùng

Table of Contents

A. Introduction.....	1
1. Project Information	1
2. Introduction.....	1
3. Current Situation	1
4. Problem Definition	1
5. Proposed Solution	2
5.1. Feature functions	2
5.2. Advantages and disadvantages	2
6. Functional Requirement	3
7. Role and Responsibilities.....	4
B. Software Project Management Plan.....	5
1. Problem Definition	5
1.1. Name of this Capstone Project	5
1.2. Problem Abstract.....	5
1.3. Project Overview	5
2. Project organization	9
2.1. Software Process Model.....	9
2.2. Roles and responsibilities	10
2.3. Tools and Techniques	10
3. Project Management Plan	10
3.1. Product Backlog	10
3.2. Sprint Backlog.....	10
3.3. Deliverables	11
3.4. All Meeting Minutes	11
4. Coding Convention.....	11
C. Software Requirement Specification	12
1. User Requirement Specification	12
1.1. Guest Requirement.....	12
1.2. Student Requirement	12
1.3. Lecturer Requirement.....	12
1.4. Staff Requirement.....	12

1.5. Head-of-Academic Requirement	13
1.6. System Requirement	14
2. System Requirement Specification	14
2.1. External Interface Requirement	14
2.2. System Overview Use Case	15
2.3. List of Use Case.....	16
3. Software System Attribute	54
3.1. Usability	54
3.2. Availability	54
3.3. Security	54
3.4. Maintainability.....	54
3.5. Portability	54
3.6. Performance	54
4. Conceptual Diagram	55
D. Software Design Description	57
1. Design Overview.....	57
2. System Architectural Design.....	57
3. Component Diagram	58
4. Detailed Description.....	60
4.1. Class Diagram.....	60
4.2. Interaction Diagram	65
5. Interface	70
5.1. Component Interface	70
5.2. User Interface Design	72
6. Database Design	107
6.1. Entity relationship diagram (ERD).....	107
6.2. Data Dictionary	107
7. Algorithms	108
7.1. Edit Feedback Content.....	108
7.2. Match conductors to target	108
7.3. Statistic and generate report.....	109
E. System Implementation & Test.....	113
1. Introduction	113
1.1. Overview	113

1.2. Test Approach	113
2. Database Relationship Diagram	113
2.1. Physical Diagram	113
2.2. Data Dictionary	115
3. Test Plan	117
3.1. Features to betested	117
3.2. Features not to be tested	118
4. System Testing TestCase	119
4.1. Communication Diagram	119
4.2. Test Case	120
F. Software User's Manual	128
1. Installation Guide	128
1.1. Setting up environment at server side	128
1.2. Deployment at serverside	128
1.3. Setting up the environment at client side	129
2. User Guide	129
<i>2.1. Web application for all users</i>	129
2.2. Web application for staffs	131
2.3. Web application for Head of Academic.....	139
2.4. Mobile application for all users.....	140

List of Tables

Table 1 Roles and Responsibilities	4
Table 2 Hardware Requirement for Server.....	8
Table 3 Hardware Requirement for PC.....	8
Table 4 Hardware Requirement for PC	8
Table 5 Roles and Responsibilities Details.....	10
Table 6 Deliverables	11
Table 7 Conceptual diagram data dictionary	56
Table 8 Component Dictionary	59
Table 9 Class Diagram Explanation.....	61
Table 10 Entity Data dictionary	108
Table 11 Describe content of all tables	115
Table 12 Attribute Data Dictionary.....	117
Table 13 Hardware Requirement.....	128
Table 14 Software Requirement.....	128
Table 15 Hardware Requirement for PC.....	129
Table 16 Hardware Requirement for Mobile.....	129
Table 17 <Guest> Login	129
Table 18 <All user> Feedback List	130
Table 19 <All user> Conduct feedback	130
Table 20 <All user> Notify user	131
Table 21 <Staff> Initialize new feedback.....	131
Table 22 <Staff> Create feedback content	132
Table 23 <Staff> Modifying Feedback Information	133
Table 24 <Staff, Lecturer> View list self's report	134
Table 25 <Head of Academic > View all report	135
Table 26 <Head of Academic, Staff, Lecturer> View report details	135
Table 27 <Head of Academic, Staff, Lecturer> View semester report.....	136
Table 28 <Head of Academic, Staff > View list users.....	137
Table 29 <Head of Academic, Staff> Create new user	137
Table 30 <Head of Academic, Staff> Create new user	138
Table 31 <Head of Academic > View all reports	139
Table 32 <Head of Academic > Manage criteria	140
Table 33 <Guest> Login	141
Table 34 <All users> View list feedbacks.....	141
Table 35 <All users> Conduct feedback on mobile	142

List of Figures

Figure 1 Scrum model.....	9
Figure 2 System Overview Use Case.....	15
Figure 3 Guest Overview Use Case.....	16
Figure 4 <Guest>Login.....	16
Figure 5 <Student> Overview Use Case.....	17
Figure 6 <Student> View List Feedback.....	18
Figure 7 <Student> Conduct Feedback.....	19
Figure 8 <Student> Logout.....	20
Figure 9 <Lecturer> Overview Use Case	21
Figure 10 <Lecturer> View List Report.....	21
Figure 11 <Lecturer> View Report Details.....	22
Figure 12 <Staff> Overview Use Case.....	24
Figure 13 <Staff> Create new account.....	25
Figure 14 <Staff> Edit account	26
Figure 15 <Staff> Create New Feedback Content	28
Figure 16 <Staff> Edit Feedback Content.....	30
Figure 17 <Staff> Choose Template	31
Figure 18 <Staff> Set Feedback Name and Description	32
Figure 19 <Staff> Add Suggested Improvement	33
Figure 20 <Staff> Edit Suggested Improvement.....	35
Figure 21 <Staff> Set Feedback Semester.....	36
Figure 22 <Staff> Set Feedback Interval Time	37
Figure 23 <Staff> Set Feedback Scope	38
Figure 24 <Staff> Set Feedback Target.....	39
Figure 25 <Staff> Set Feedback Conductor.....	40
Figure 26 <Staff> Publish Feedback.....	42
Figure 27 <Staff> Save Feedback as Template.....	43
Figure 28 <Staff> Update Feedback Template	44
Figure 29 <Staff> Unpublish Feedback.....	45
Figure 30 <Head of Academic> Overview Use Case	46
Figure 31 <Head of Academic> Create new criteria	46
Figure 32 <Head of Academic> Edit criteria	47
Figure 33 <Head of Academic> Deactivate criteria	48
Figure 34 <Head of Academic> Deactivate account.....	49
Figure 35 <System (Actor)> Overview Use Case.....	50
Figure 36 <System (Actor)> Analyze Feedback	50
Figure 37 <System (Actor)> Visualize Analyzed Data	52
Figure 38 <System (Actor)> Alert undone Feedback	53
Figure 39 Conceptual diagram	55
Figure 40 System Architecture Design	57
Figure 41 Component Diagram.....	58
Figure 42 Class Diagram.....	60
Figure 43 Sequence diagram - Create new Feedback	66
Figure 44 Sequence diagram - Conduct Feedback.....	67
Figure 45 Sequence diagram - View Feedback Report	67
Figure 46 Sequence diagram - Alert Undone Feedback	68
Figure 47 Sequence diagram - Notify Undone Feedback by Email.....	68
Figure 48 Sequence diagram - Create User	69
Figure 49 Sequence diagram - Create Criteria	69
Figure 50 <Web Application> Login	72
Figure 51 <Web Application> Side menu.....	73
Figure 52 <Web Application> Login	75
Figure 53 <Web Application> Alert Undone Feedback	76
Figure 54 <Web Application> Manage Criteria	76
Figure 55 <Web Application> Add Criteria	77
Figure 56 <Web Application> Select Template.....	78

Figure 57 <Web Application> Preview Template	79
Figure 58 <Web Application> Create New Blank Feedback	80
Figure 59 <Web Application> Create Feedback Content	81
Figure 60 <Web Application> Overview Creating Feedback	82
Figure 61 <Web Application> Table List Class Targets	83
Figure 62 <Web Application> Table List Course Targets	84
Figure 63 <Web Application> Table List Major Targets	84
Figure 64 <Web Application> Table List Department Targets	85
Figure 65 <Web Application> Preview Edit Feedback Content	86
Figure 66 <Web Application> Modify Suggestions	87
Figure 67 <Web Application> Modify Class Targets	88
Figure 68 <Web Application> Modify Course Targets	89
Figure 69 <Web Application> Modify Major Targets	90
Figure 70 <Web Application> Modify Department Targets	90
Figure 71 <Web Application> Modify Conductors	91
Figure 72 <Web Application> View List Feedbacks	92
Figure 73 <Web Application> Conduct Feedbacks	93
Figure 74 <Web Application> View List Feedbacks for Staff and Head Academic	94
Figure 75 <Web Application> Publish/Unpublish Feedback	95
Figure 76 <Web Application> View list Reports	96
Figure 77 <Web Application> View Report Details	97
Figure 78 <Web Application> View Semester Report Details	99
Figure 79 <Web Application> View List Users	100
Figure 80 <Web Application> Add New User	101
Figure 81 <Web Application> Edit User	102
Figure 82 <Mobiel Application> Login	104
Figure 83 <Mobiel Application> View List Feedbacks	105
Figure 84 <Mobiel Application> Conduct Feedback	106
Figure 85 Entity relationship diagram	107
Figure 86 Bar Chart report	110
Figure 87 Pie Chart report	110
Figure 88 Physical Diagram	114
Figure 89 Student Communication Diagram	119
Figure 90 Staff Communication Diagram	119
Figure 91 Head of Academic Communication Diagram	119
Figure 92 System Communication Diagram	119
Figure 93 <Guest> Login page	129
Figure 94 <All users> Feedback List	130
Figure 95 <All users> Conduct feedback	130
Figure 96 <All users> Notify user	131
Figure 97 <Staff> Initialize new feedback	131
Figure 98 <Staff> Create feedback content	132
Figure 99 <Staff> Modifying Feedback Information	133
Figure 100 <Staff, Teacher> View list self's report	134
Figure 101 <Head of Academic> View all report	134
Figure 102 <Head of Academic, Staff, Lecturer> View report details	135
Figure 103 <Head of Academic, Staff, Lecturer> View semester report details	136
Figure 104 <Head of Academic, Staff> View list users	136
Figure 105 <Head of Academic, Staff> Create new user	137
Figure 106 <Head of Academic, Staff> Unpublish Feedback	138
Figure 107 <Head of Academic> View all reports	139
Figure 108 <Head of Academic> Manage criteria	140
Figure 109 <Guest> Login	140
Figure 110 <All users> View list feedbacks	141
Figure 111 <All users> Conduct feedback on mobile	142

Definitions, Acronyms, and Abbreviations

Name	Definition
SFMS	School Feedback Management System

A. Introduction

1. Project Information

- Project name: School Feedback Management System
- Project Code: SFMS
- Product Type: **Web app, Mobile App**
- Start Date: **08/01/2018**
- End Date: **29/04/2018**

2. Introduction

In this document, we introduce a solution for School Feedback System. Current feedback systems have some problems like impossible to design and customize feedback form that suitable for each major, lecture, course, or inconvenient in analyzing results. Base on our reaserches and analysis, we proposed a solution for university in Vietnam and other developed countries.

We build a system, which helps the universities solve current problems. In the process of analysis, we believe our system is capable to resolve the problem by let users create, design, and customize feedback form through dragging and dropping items. Besides that, we also analyze feedback results and provide reports and suggested improvements, based on the results.

This document also describes our working process in 4 months includes our perspective in the system, component designs and detailed core workflows. We hope the system and our solution will help resolve the problems from universities in Vietnam and other developed countries.

3. Current Situation

Currently, in university, we use the same feedback form for every courses and major. So we can just gain feedback of some general information like: if lecturer is on time, if the students can understand the lecture, how skillful the lecturer is... When feedback period's over, the results will send to lecturer without analysing or providing suggested improvement.

4. Problem Definition

- **Customize Feedback form:**

- Use same questions set for all classes, courses, majors
- Can't question unique aspect of each course (For example: Can't question about accent of English lecturer)
- Lack of usefulness and practicality (Can only ask same general questions that used for all courses)
- Hardly use for personal improvement
- Can't modify targets or conductors of Feedback
- Can't choose type of feedback survey (Feedback for class, course, major, or for department)

- **Analyze results and Report**
 - Can't generate graphical charts of statistical results from feedback answers
 - Can't compare results in the past to current
 - Don't provide suggested improvement to lecturers
- **Conduct Feedback**
 - Edit answers

5. Proposed Solution

Our proposed solution is to build a system named School Feedback Manage System can drag-and-drop feedback items to create new feedback form to resolve those problems of current situations.

SFMS includes a web application to manage and create feedback, and mobile app to conduct feedback, with following functions:

5.1. Feature functions

- **Web Application:**
 - Create new Feedback form: choose existed template or customize by dragging and dropping items to the form
 - Save created feedback template
 - Choose scope of Feedback: Feedback for a lecture, a major, a course or a department
 - Set suggested improvement based on average points of each feedback
 - Set interval time for each Feedback
 - Remind users to conduct Feedback
 - Search and Filter feedback
 - Conduct Feedback
 - Edit Answers
 - Search and Filter reports
 - Compare points in the past to current through semester
 - View report in details
- **Mobile Application:**
 - View list Feedback
 - Conduct Feedback
 - Edit answer

5.2. Advantages and disadvantages

- **Advantages:**
 - Make realistic improvement from feedback
 - Help Head of Academic follows real performance of lecturers
 - Feedback can use in different subjects, courses, majors, class, departments
 - Represent results as graphical charts
 - Save created feedback form to
- **Disadvantages:**
 - Staffs have to spend time to design feedback form
 - Input suggested improvement manually

6. Functional Requirement

Base component

- Login
- Logout

• Student component

- View list of feedbacks
- Filter feedbacks
- Conduct Feedback
- Be alerted of undone feedback

• Lecturer component

- View list of feedbacks
- Filter feedbacks
- Conduct Feedback
- Be alerted of undone feedback
- View list of Report for that relates to their classes
- Filter Reports
- View Report comparisions through semesters
- View Report details

• Staff component

- View list of feedbacks
- Filter feedbacks
- Conduct Feedback
- Be alerted of undone feedback
- View list of Report for that relates to their departments
- Filter Reports
- View Report comparisions through semesters
- View Report details
- Create blank new feedback form:
 - + Create blank new feedabck
 - + Drag-n-drop feedback items to create content
 - + Edit feedback content
 - + Set suggested improvement
 - + Set name and description for feedback
 - + Set interval time for feedback
 - + Set semester for feedback
 - + Choose scope (lecture, major, course, department)
 - + Choose and modify feedback target
- Create new feedback from template
- Save Feedback:
 - + Save and publish feedback
 - + Save feedback as a new template
 - + Update current template
- Unpublish/Publish Feedback
- Manage Users:
 - + View list users
 - + Add users
 - + Update users

- **Head-of-Academic component**
 - View list of feedbacks
 - Filter feedbacks
 - Conduct Feedback
 - Be alerted of undone feedback
 - View list of all Reports
 - Filter Reports
 - View Report comparisions through semesters
 - View Report details
 - Create blank new feedback form:
 - + Create blank new feedabck
 - + Drag-n-drop feedback items to create content
 - + Edit feedback content
 - + Set suggested improvement
 - + Set name and description for feedback
 - + Set interval time for feedback
 - + Set semester for feedback
 - + Choose scope (lecture, major, course, department)
 - + Choose and modify feedback target
- Create new feedback from template
- Save Feedback:
 - + Save and publish feedback
 - + Save feedback as a new template
 - + Update current template
- Unpublish/Publish Feedback
- Manage Users:
 - + View list users
 - + Add users
 - + Update users
 - + Deactive users
- Manage Criterias:
 - + View list criterias
 - + Add criterias
 - + Update criterias
 - + Deactive criterias

7. Role and Responsibilities

No	Full Name	Role	Position	Contact
1	Nguyễn Huy Hùng	Project Manager	Supervisor	hungng@fpt.edu.vn
2	Nguyễn Việt Tú	Developer	Leader	tunvse61897@fpt.edu.vn
4	Hoàng Quốc Việt	Developer	Member	viethqse61745@fpt.edu.vn
5	Trần Hồ Minh Thuấn	Developer	Member	thuanhmse61882@fpt.edu.vn

Table 1 Roles and Responsibilities

B. Software Project Management Plan

1. Problem Definition

1.1. Name of this Capstone Project

- i. Official name: School Feedback Management System
- ii. Vietnamese name: Hệ Thống Quản Lý Feedback Trường Học
- iii. Abbreviation: SFMS

1.2. Problem Abstract

This project is our exertion about School Feedback System in Viet Nam. We called it is School Feedback Management System (SFMS). SFMS will provide users a user-friendly interface to create and conduct feedback.

Organizing a feedback survey in an effectively way is always a problem to improve quality. With the current Feedback system, users use the same feedback for all department, majors and course. Therefore, it's hard to evaluate the true quality while each department, major and course has its unique aspects.

SFMS will provide a system where users can create several Feedback forms easily by dragging and dropping Feedback items and save them as templates to reuse later.

To qualify the results, SFMS uses a new critical points-counting system for each question, provides suggested improvement when critical points is low.

Finally, SFMS analyzed results and presents as charts, help users to review and compare the results through semesters.

1.3. Project Overview

1.3.1. Current Situation

Below are the problems encountered in this project:

- Lack of the amount of necessary data: students, lecturers, staffs... data
- Limit in human resources and time: Team has only 4 members and time for all project is about 13 weeks for writing document, implementing the products and testing
- Lack of UI, UX (user experience) design skill: Our team members all study IS major and no one has studied UI, UX design.
- Lack of knowledge about back-office business: How point is counted in survey, how feedback results are treated...
- New technique: Some team members are new to the techniques used in the project. The team needs an amount of time to get familiar with those techniques.

1.3.2. The Proposed System

The system will have three sub-systems:

- An API application to serve API for mobile application and web application.
- A web application for staffs, lecturers, head of academic and students. Head of Academic can manage users, majors, courses, departments. Staffs and head of academic can create new feedback or choose existed template. Staff, Head of academic and lecturers can view reports when feedback period finishes. All users can conduct feedback.
- A mobile application for students conduct feedback

Task will be assigned vertically to team members, so that if one member quits, the team will not lack of resources.

1.3.2.1. Web Site

- **Base component**

- Login
- Logout

- **Student component**

- View list of feedbacks
- Filter feedbacks
- Conduct Feedback
- Be alerted of undone feedback

- **Lecturer component**

- View list of feedbacks
- Filter feedbacks
- Conduct Feedback
- Be alerted of undone feedback
- View list of Report for that relates to their classes
- Filter Reports
- View Report comparisions through semesters
- View Report details

- **Staff component**

- View list of feedbacks
- Filter feedbacks
- Conduct Feedback
- Be alerted of undone feedback
- View list of Report for that relates to their departments
- Filter Reports
- View Report comparisions through semesters
- View Report details
- Create blank new feedback form:
 - + Create blank new feedabck
 - + Drag-n-drop feedback items to create content
 - + Edit feedback content
 - + Set suggested improvement
 - + Set name and description for feedback
 - + Set interval time for feedback
 - + Set semester for feedback
 - + Choose scope (lecture, major, course, department)
 - + Choose and modify feedback target
- Create new feedback from template
- Save Feedback:
 - + Save and publish feedback
 - + Save feedback as a new template
 - + Update current template
- Unpublish/Publish Feedback

- Manage Users:
 - + View list users
 - + Add users
 - + Update users
- **Head-of-Academic component**
 - View list of feedbacks
 - Filter feedbacks
 - Conduct Feedback
 - Be alerted of undone feedback
 - View list of all Reports
 - Filter Reports
 - View Report comparisions through semesters
 - View Report details
 - Create blank new feedback form:
 - + Create blank new feedabck
 - + Drag-n-drop feedback items to create content
 - + Edit feedback content
 - + Set suggested improvement
 - + Set name and description for feedback
 - + Set interval time for feedback
 - + Set semester for feedback
 - + Choose scope (lecture, major, course, department)
 - + Choose and modify feedback target
 - Create new feedback from template
 - Save Feedback:
 - + Save and publish feedback
 - + Save feedback as a new template
 - + Update current template
 - Unpublish/Publish Feedback
 - Manage Users:
 - + View list users
 - + Add users
 - + Update users
 - + Deactive users
 - Manage Criterias:
 - + View list criterias
 - + Add criterias
 - + Update criterias
 - + Deactive criterias

1.3.2.2. Mobile Application

- View list of feedbacks
- Filter feedbacks
- Conduct Feedback

1.3.2.3. API Application

The server system takes responsibility to respond all the requests and also manages and processes data.

- Provide

- APIs for Mobile Application, Web Application

1.3.3. Boundaries of the System

The system can:

- Allow Head of academic and Staff to manage users.
- Allow Head of academic to manage criteria.
- Allow create new feedbacks
- Allow save feedbacks as templates to reuse
- Allow conduct feedback
- Notify users when there is undone feedback
- Count feedback points based on result
- Suggest improvement based on Feedback result
- View feedback reports

1.3.4. Future Plans

Current system is concentrated on core business flow. Therefore, some supporting features are restricted for the development team. These features may be expanded in the future:

- Analyze if users really spend time to do feedback
- Analyze “Other comment” paragraph to see if it’s positive or negative comment and count point

1.3.5. Development environment

1.3.5.1. Hardware requirements

For server		
Windows	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (4 Mbps)	Cable, Wi-Fi (8 Mbps)
Operating System	Window Server 2008	Window Server 2008
Computer Processor	Intel® Xeon ® 1.4GHz	Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz)
Computer Memory	1GB RAM	2GB or more

Table 2 Hardware Requirement for Server

For PC		
Windows	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (2 Mbps)	Cable, Wi-Fi (4 Mbps)
Operating System	Windows 7 or above	Window 7 or above
Computer Processor	Intel® Pentium 4 1.60Ghz	Intel® Pentium 4 2.00Ghz
Computer Memory	1GB RAM	2GB or more

Table 3 Hardware Requirement for PC

For mobile		
Windows	Minimum Requirements	Recommended
Internet Connection	Wi-Fi (2 Mbps)	Wi-Fi (4 Mbps)
Operating System	Android 4.4.2 or above	Android 6.0 or above
Memory	1GB RAM	2GB or more

Table 4 Hardware Requirement for PC

1.3.5.2. Software requirement

Software	Name / Version	Description
Operating system	Window 7 or above	Operating system and platform for development
Environment	Java EE 8, Spring, Gradle	Specification for developing web application
Modeling tool	StarUML 2.8.0	Use to draw model models and diagrams
IDE	NetBeans 8.2, IntelliJ IDEA 17.2, Android Studio 3.0	Programming tools
DBMS	MySQL 5.7	Used to create & manage the database for system
Source control	Github	Used for source control
Web browser	Chrome 42 or above	Testing browser
Application Server	Tomcat 8.0	Server to deploy app

2. Project organization

2.1. Software Process Model

This project is developed using Scrum model – part of an agile framework for Software development project. Our team choose Scrum model because of the following reasons:

- Our team only has 4 members, and tasks are assigned vertically, do all steps from design, coding, testing and implementation. Scrum is the most suitable model for small and medium project.
- In the project there are many new technologies that need to be learned. With the Scrum model, the team can learn and develop in parallel to meet deadline.
- There is no leader, no hierarchy in team, so team members work cheerfully, stimulating the initiative and creativity of each member.
- Product owner can change requirement or extend scope. The team will adapt to change better.

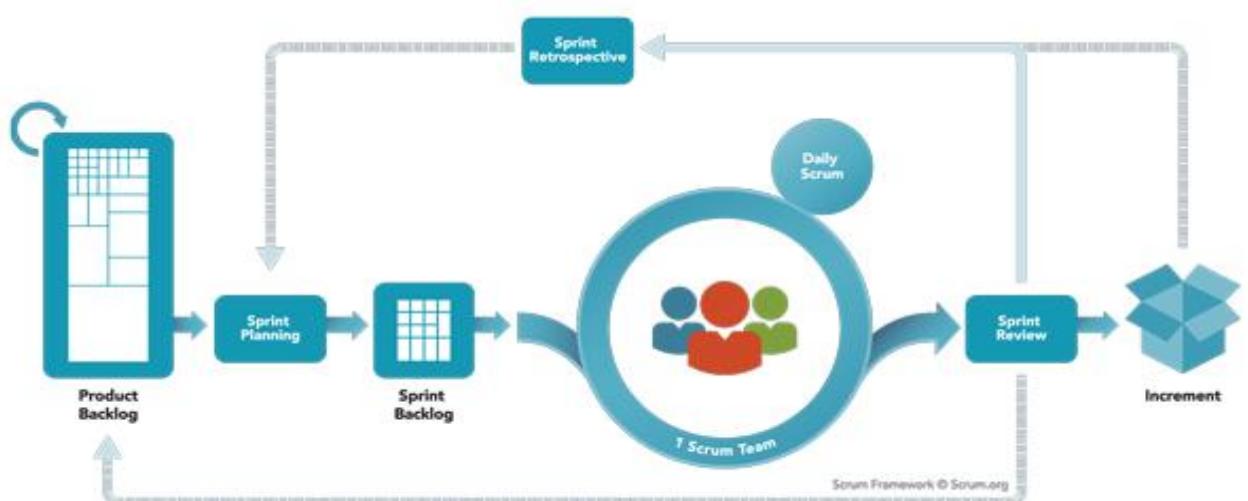


Figure 1 Scrum model

<https://www.scrum.org/resources/what-is-scrum>

2.2. Roles and responsibilities

No	Full name	Role in Group	Responsibilities
1	Nguyễn Huy Hùng	Project manager	<ul style="list-style-type: none"> Specify scope and user requirement Control the development process Give out technique and business analysis support
2	Nguyễn Việt Tú	Scrum master	<ul style="list-style-type: none"> Create Sprint Backlog and Product Backlog. Make sure the Scrum teams understand and follow the process. Always be present to answer questions and give advice when product owner or scrum member needs. Help the team master scrum artifacts such as: Sprint Backlog, Product Backlog, ... Writing report
3	Nguyễn Việt Tú Trần Hồ Minh Thuấn Hoàng Quốc Việt	Scrum team members	<ul style="list-style-type: none"> Designing database Clarifying requirements Prepare documents GUI Design Coding Testing

Table 5 Roles and Responsibilities Details

2.3. Tools and Techniques

Tool / Technique	Name / version
Frontend	HTML, CSS, JavaScript, Bootstrap, Thymeleaf
Backend	SpringBoot framework, Java
IDE	NetBeans 8.2, IntelliJ IDEA 17.2, Android Studio 3.0
Database	MySQL 5.7
Modelling tool	Star UML 2.8.0

3. Project Management Plan

3.1. Product Backlog

All product backlog could be found [here](#)

3.2. Sprint Backlog

All sprint backlog could be found [here](#)

3.3. Deliverables

No	Deliverable	Deliverable date	Deliverable location	Note
1	Introduction, Project Management Plan, Concept Diagram, Class Diagram, Entity Relationship Diagram, Use Case Overview, Mock UI			Sprint 1
2	Study Spring Boot Framework, Design User Interface for Web Application			Sprint 2
3	Design User Interface for Web Application (continue), Web Service Architecture, Code core flow RESTful API Web services			Sprint 3
4	Code core flow Web application and Mobile application; User Requirement Specification			Sprint 4 -> 8
5	Code low priority functions, Conceptual Diagram, Design Overview, System Architectural Design, Component Diagram.			Sprint 9-10
6	Testing, Entity Relationship Diagram, Database Diagram, Algorithms			Sprint 11-12

Table 6 Deliverables

- For each Sprint, deliverables are potentially shippable products, which can be a part of document or prototype implemented based on the project's core flow.
- Each Sprint has a fixed duration of one week.

3.4. All Meeting Minutes

All sprint meeting minutes could be found [here](#)

4. Coding Convention

- **Naming convention:**
 - Variable and method names are in mixed case, with first letter of each internal word capitalized except first word.
- **Method names should be verbs.**
 - Class names should be nouns, in mixed case with first letter of each internal word capitalized.
 - Constant names should be all uppercase with words separated by underscore.
- **Comment:**
 - Using /* */ for block comments.
 - Using // for line comments.

Using Java coding convention from:

<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

C. Software Requirement Specification

1. User Requirement Specification

1.1. Guest Requirement

Guest is a person who wants to create or conduct feedback. Guest can do the following functions:

- Login

1.2. Student Requirement

Student is a person who attends in a class. Student conducts most of feedbacks.

Student can do the following functions:

- Manage Feedbacks:
 - + View list feedbacks
 - + Filter feedbacks
- Conduct Feedback
- Be alerted of undone feedback

1.3. Lecturer Requirement

Lecturer is a person who teach a class. Lecturer is the target of most of feedbacks. Teacher can do the following functions:

- Manage Feedbacks:
 - + View list feedbacks
 - + Filter feedbacks
- Conduct Feedback
- Be alerted of undone feedback
- Manage Reports:
 - + View list reports that relates to their classes
 - + Compare reports by semesters
 - + View report details
 - + Filter reports

1.4. Staff Requirement

Staff is a person who work for the school, but doesn't have the highest role. Staff can do the following functions:

- Conduct Feedback
- Be alerted of undone feedback
- Create new feedback form:
 - + Create blank new feedabck
 - + Drag-n-drop feedback items to create content
 - + Edit feedback content
 - + Set suggested improvement
 - + Set name and description for feedback
 - + Set interval time for feedback

- + Set semester for feedback
- + Choose scope (lecture, major, course, department)
- + Choose and modify feedback target
- + Choose and modify feedback conductor
- Create new feedback from template
- Save Feedback:
 - + Save and publish feedback
 - + Save feedback as a new template
 - + Update current template
- Manage Feedbacks:
 - + View list feedbacks
 - + Filter feedbacks
 - + Publish/Unpublish Feedback
- Manage Reports:
 - + View list reports that relates to their departments
 - + Compare reports by semesters
 - + View report details
 - + Filter reports
- Manage Users:
 - + View list users
 - + Add users
 - + Update users

1.5. Head-of-Academic Requirement

Head of Academic is a person manage everything and have the highest role. Head of Academic can manage criterias of feedbacks, deactivate users and can view all reports. Head of Academic can do the following functions:

- Conduct Feedback
- Be alerted of undone feedback
- Create new feedback form:
 - + Create blank new feedabck
 - + Drag-n-drop feedback items to create content
 - + Edit feedback content
 - + Set suggested improvement
 - + Set name and description for feedback
 - + Set interval time for feedback
 - + Set semester for feedback
 - + Choose scope (lecture, major, course, department)
 - + Choose and modify feedback target
 - + Choose and modify feedback conductor
- Create new feedback from template
- Save Feedback:
 - + Save and publish feedback
 - + Save feedback as a new template
 - + Update current template

- Manage Feedbacks:
 - + View list feedbacks
 - + Filter feedbacks
 - + Publish/Unpublish Feedback
- Manage Reports:
 - + View all reports
 - + Compare reports by semesters
 - + View report details
 - + Filter reports
- Manage Users:
 - + View list users
 - + Add users
 - + Update users
 - + Deactivate users
- Manage Criterias:
 - + View list criterias
 - + Add criterias
 - + Update criterias
 - + Deactivate criterias

1.6. System Requirement

System performs functions that run underground and connects to external system (deposit money, notification system). System does following functions:

- Calculate statistic results for report
- Send notification when users have undone feedbacks

2. System Requirement Specification

2.1. External Interface Requirement

2.1.1. User Interface

- Mobile and Web application use Vietnamese interface, suitable for Vietnamese user.

2.1.2. Hardware Interface

- N/A

2.1.3. Software Interface

- Web application: work with Firefox, Chrome, Internet Explorer browsers.
- Mobile application: Android operating system.

2.1.4. Communication Protocol

- Use HTTP protocol 1.1 for communication between the web browser and the web server, mobile application and the server.

2.2. System Overview Use Case

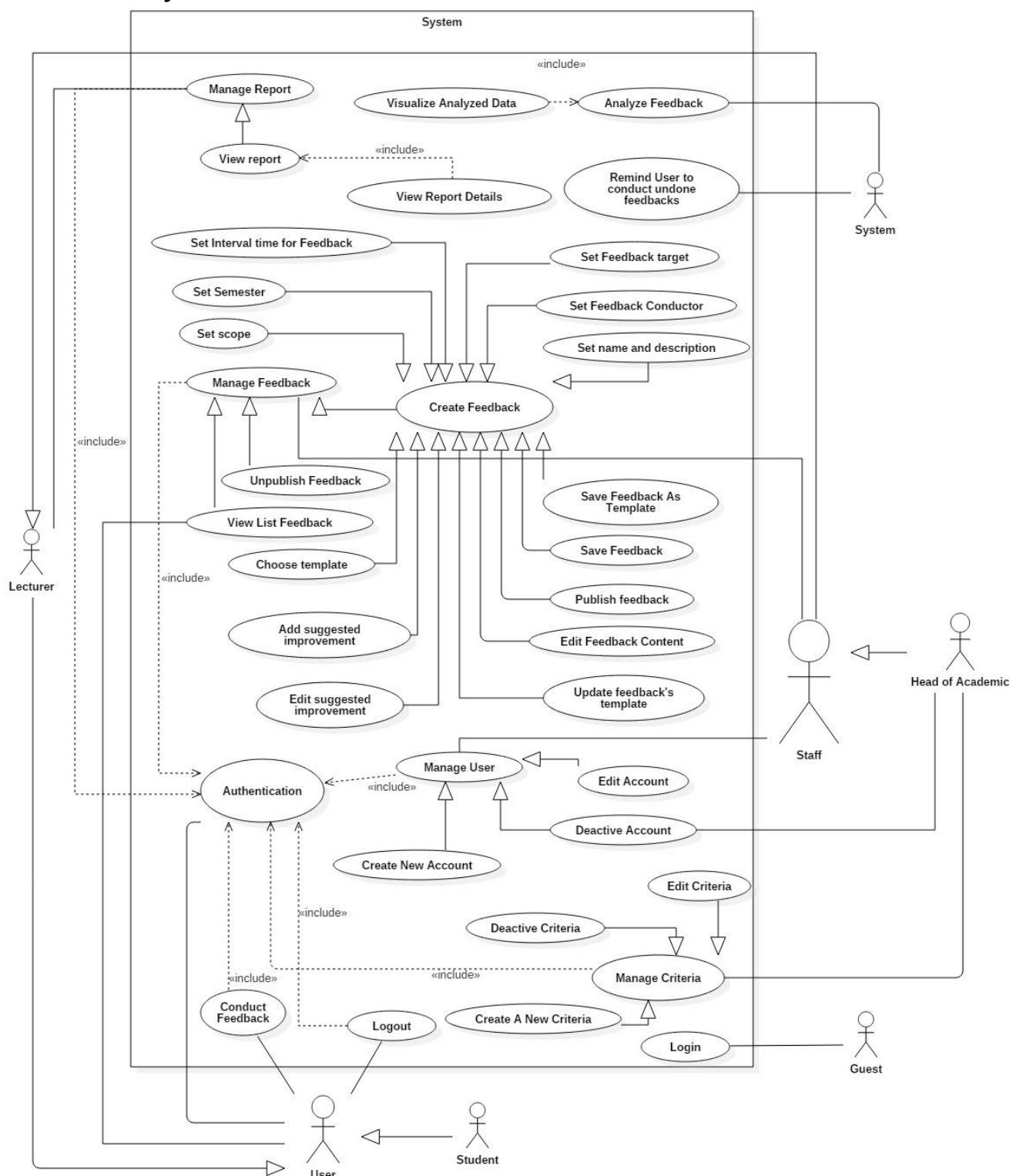


Figure 2 System Overview Use Case

2.3. List of Use Case

2.3.1. Guest Overview Use Case

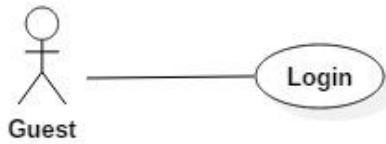


Figure 3 Guest Overview Use Case

2.3.1.1. Guest Login

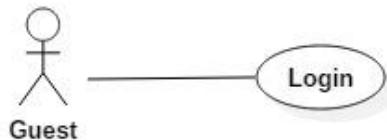


Figure 4 <Guest>Login

USE CASE - UC001															
Use Case No.	001	Use Case Version	1.0												
Use Case Name	Login														
Author	VietHQ														
Date	27/03/2018	Priority	High												
Actor:	<ul style="list-style-type: none"> - Guest 														
Summary:	<ul style="list-style-type: none"> - This use case allows Guest login to the system on website and mobile application. 														
Goal:	<ul style="list-style-type: none"> - Guest login successfully with the proper role. 														
Triggers:	<ul style="list-style-type: none"> - Guest sends the login command. 														
Preconditions:	<ul style="list-style-type: none"> - Guest has an account. 														
Post Conditions:	<ul style="list-style-type: none"> - Success: Guest accesses the system successfully. - Fail: System shows error message "Invalid username or password". 														
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Guest sends command to request login.</td><td> System requires identity information from Guest: <ul style="list-style-type: none"> - Username: free text input. - Password: free text input. </td></tr> <tr> <td>2</td><td>Guest inputs information</td><td></td></tr> <tr> <td>3</td><td>Guest sends command to login to system. [Alternative 1]</td><td>Guests will login system with their specific role.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Guest sends command to request login.	System requires identity information from Guest: <ul style="list-style-type: none"> - Username: free text input. - Password: free text input. 	2	Guest inputs information		3	Guest sends command to login to system. [Alternative 1]	Guests will login system with their specific role.
Step	Actor Action	System Response													
1	Guest sends command to request login.	System requires identity information from Guest: <ul style="list-style-type: none"> - Username: free text input. - Password: free text input. 													
2	Guest inputs information														
3	Guest sends command to login to system. [Alternative 1]	Guests will login system with their specific role.													

Alternative Scenario:

Step	Actor action	System Response
1	Guest input invalid username or password.	System shows error message "Tên đăng nhập hoặc mật khẩu không đúng".

Exceptions : N/A**Relationships:** N/A**Business Rules:**

- After login to system, guest will be redirected to specific view based on their role on the system: student, lecturer, staff or head of academic.

- If role is "Student", the system will display to Student view.
- If role is "Lecturer", the system will display to Lecturer view.
- If role is "Staff", the system will display to Staff view.
- If role is "Head of Academic", the system will display to Head of Academic view.

2.3.2. Student Overview Use Case

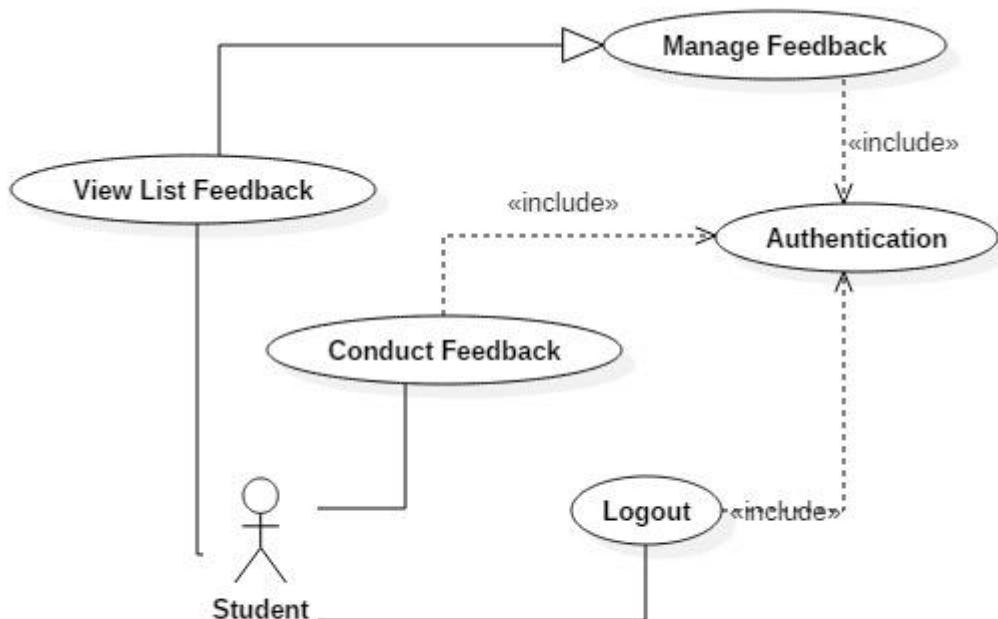


Figure 5 <Student> Overview Use Case

2.3.2.1. Student View List Feedback

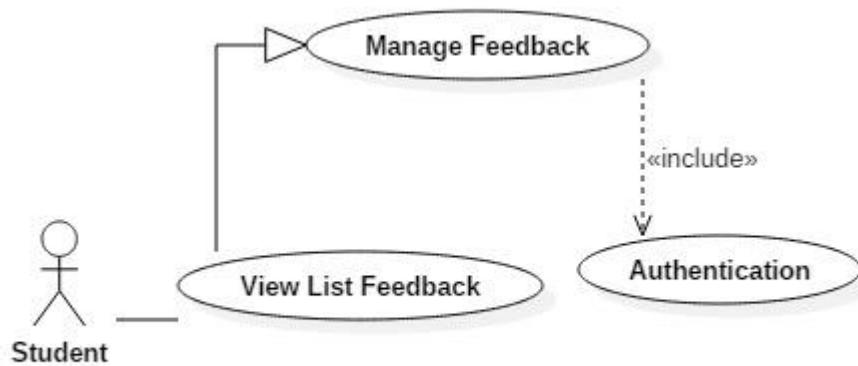


Figure 6 <Student> View List Feedback

USE CASE - UC002									
Use Case No.	002	Use Case Version	1.0						
Use Case Name	View List Feedback								
Author	VietHQ								
Date	27/03/2018	Priority	High						
Actor:	<ul style="list-style-type: none"> - Student - Staff - Lecturer - Head of Academic 								
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to get feedback list. 								
Goal:	<ul style="list-style-type: none"> - System gets feedback list and display it to Actor. 								
Triggers:	<ul style="list-style-type: none"> - Actor sends get feedback list command. 								
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system. 								
Post Conditions:	<ul style="list-style-type: none"> - Success: Feedback list is displayed. - Fail: Show error messages. 								
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor sends get feedback list command.</td><td>[Exception 1] System displays feedback list to Actor.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Actor sends get feedback list command.	[Exception 1] System displays feedback list to Actor.
Step	Actor Action	System Response							
1	Actor sends get feedback list command.	[Exception 1] System displays feedback list to Actor.							
Exceptions 1:	<table border="1"> <thead> <tr> <th>Step</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Feedback list is empty</td><td>System shows error message "Không có Phản hồi".</td></tr> </tbody> </table>			Step	Cause	System Response	1	Feedback list is empty	System shows error message "Không có Phản hồi".
Step	Cause	System Response							
1	Feedback list is empty	System shows error message "Không có Phản hồi".							
Relationships:	N/A								
Business Rules:	<ul style="list-style-type: none"> - Actor can view list of feedbacks that they are in conductors list. - Actor can conduct feedbacks that are still in available time from this list. 								

2.3.2.2. Student Conduct Feedback

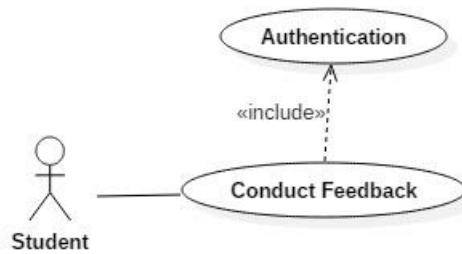


Figure 7 <Student> Conduct Feedback

USE CASE - UC003			
Use Case No.	003	Use Case Version	1.0
Use Case Name	Conduct Feedback		
Author	VietHQ		
Date	27/03/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Student - Staff - Lecturer - Head of Academic 		
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to get conduct feedback. 		
Goal:	<ul style="list-style-type: none"> - Actor submits answers successfully. 		
Triggers:	<ul style="list-style-type: none"> - Actor clicks button “Hoàn thành ngay” from list feedback. - Actor clicks corresponding feedback from alert button. 		
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system. - Actor has feedback needed to conduct. - Feedback is still in available interval time. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Actor submits answers successfully. - Fail: Show error messages. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Actor sends conduct feedback command.	System displays list of feedback questions to Actor.	
2	Actor chooses options		
3	Actor submits answers	System checks answers. [Exception 1] System directs to view list feedback screen and changes button “Hoàn Thành Ngay” to “Đã Hoàn Thành”	
Exceptions 1:			
Step	Cause	System Response	
1	Actor doesn't answer all required questions	System shows error message “Xin hoàn thành những câu hỏi bắt buộc.”	

Relationships: N/A

Business Rules:

- Actor answers question of the feedbacks.
- Actor has to answer all required questions.

2.3.2.3. Student Logout

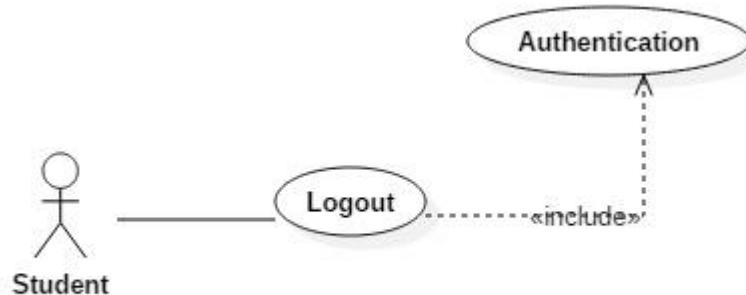


Figure 8 <Student> Logout

USE CASE - UC003

Use Case No.	004	Use Case Version	1.0
Use Case Name	Logout		
Author	VietHQ		
Date	27/03/2018	Priority	High

Actor:

- Student
- Staff
- Lecturer
- Head of Academic

Summary:

- This use case allows Actor to logout.

Goal:

- Actor logsouts successfully.

Triggers:

- Actor clicks button "Logout".

Preconditions:

- Actor has been authorized and accessed the system.

Post Conditions:

- **Success:** Actor logsouts succesfully.
- **Fail:** N/A

Main Success Scenario:

Step	Actor Action	System Response
1	Actor sends command to Logout.	System clears session state if any, takes user out of the system. System displays sign in view.

Exceptions 1: N/A

Relationships: N/A

Business Rules:

- After logout, role "Authenticated User" will become "Guest".

2.3.3. Lecturer Overview Use Case

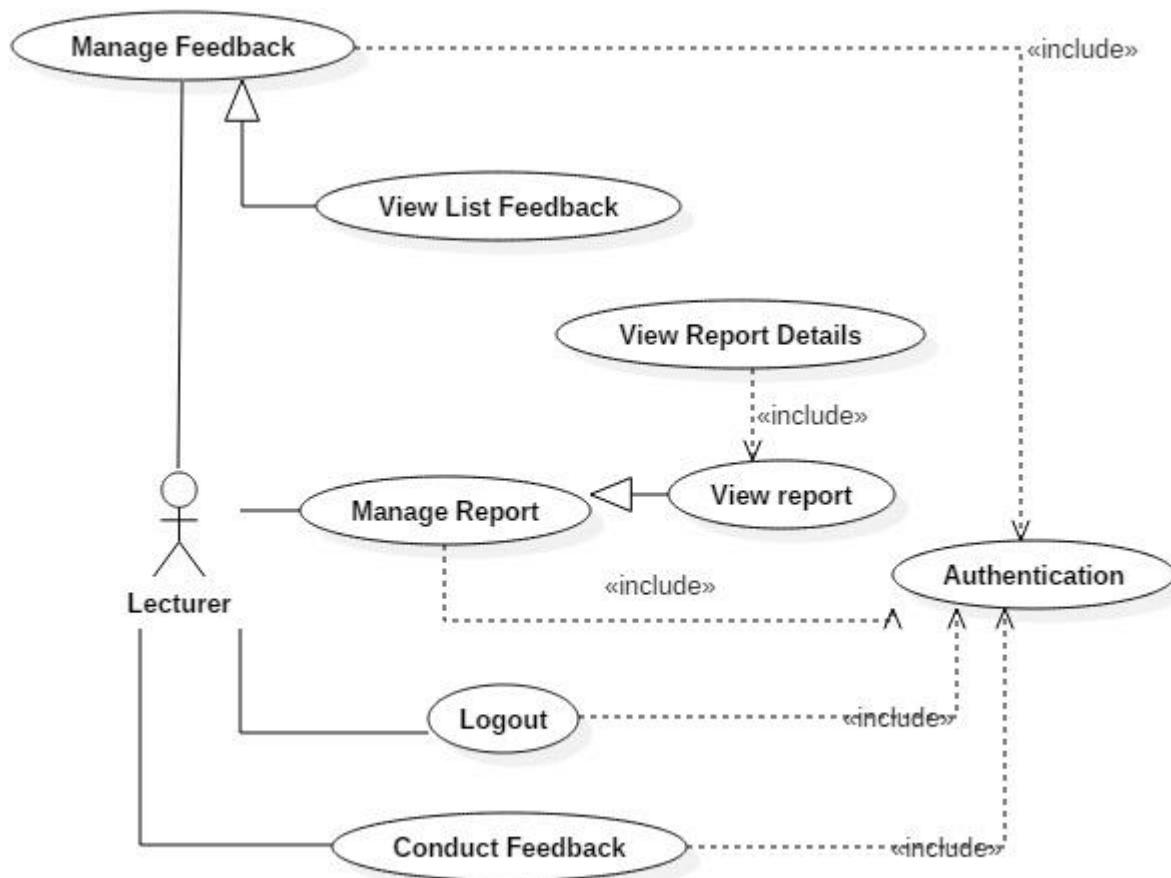


Figure 9 <Lecturer> Overview Use Case

2.3.3.1. Lecturer View List Report



Figure 10 <Lecturer> View List Report

USE CASE - UC005			
Use Case No.	005	Use Case Version	1.0
Use Case Name	View List Report		
Author	TuNV		
Date	27/03/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff - Lecturer - Head of Academic 		
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to get report list. 		

Goal:	<ul style="list-style-type: none"> - System gets report list and display it to Actor. 							
Triggers:	<ul style="list-style-type: none"> - Actor sends get report list command. 							
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system. 							
Post Conditions:	<ul style="list-style-type: none"> - Success: Report list is displayed. - Fail: Show error messages. 							
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor sends get report list command.</td><td>[Exception 1] System displays report list to Actor.</td></tr> </tbody> </table>		Step	Actor Action	System Response	1	Actor sends get report list command.	[Exception 1] System displays report list to Actor.
Step	Actor Action	System Response						
1	Actor sends get report list command.	[Exception 1] System displays report list to Actor.						
Exceptions 1:	<table border="1"> <thead> <tr> <th>Step</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Report list is empty</td><td>System shows error message "Không có Báo cáo".</td></tr> </tbody> </table>		Step	Cause	System Response	1	Report list is empty	System shows error message "Không có Báo cáo".
Step	Cause	System Response						
1	Report list is empty	System shows error message "Không có Báo cáo".						
Relationships: N/A								
Business Rules:	<ul style="list-style-type: none"> - Actor can view list of report that corresponds to their roles. - Lecturer can view reports of their classes. - Staff can view reports of their departments. - Head of Academic can view all reports. 							

2.3.3.2. *Lecturer View Report Details*

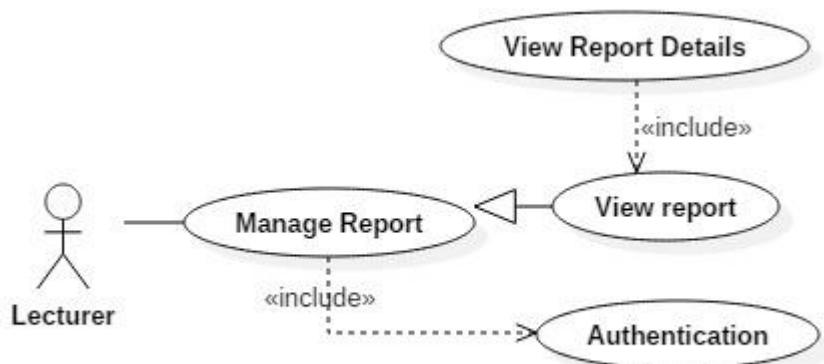


Figure 11 <Lecturer> View Report Details

USE CASE - UC006			
Use Case No.	006	Use Case Version	1.0
Use Case Name	View List Report Details		
Author	TuNV		
Date	27/03/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff - Lecturer - Head of Academic 		
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to view report details. 		
Goal:			

- System gets details of a report and displays it to Actor.

Triggers:

- Actor sends get report details command.

Preconditions:

- Actor has been authorized and accessed the system.

Post Conditions:

- **Success:** Details of a report is displayed.
- **Fail:** Show error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor sends get report details command.	System calculates answer points, statistics and displays report list to Actor.

Relationships: N/A

Business Rules:

- Actor can view details of a report.
- Details of a report included: Statistic number of answers of each questions, proportions of answers, average point of each criteria, average point of the report.
- Average point of each criteria is the average point of all questions belong to that criteria.
- Average point of the report is the average point of all criteria belong to that report.

2.3.4. Staff Overview Use Case

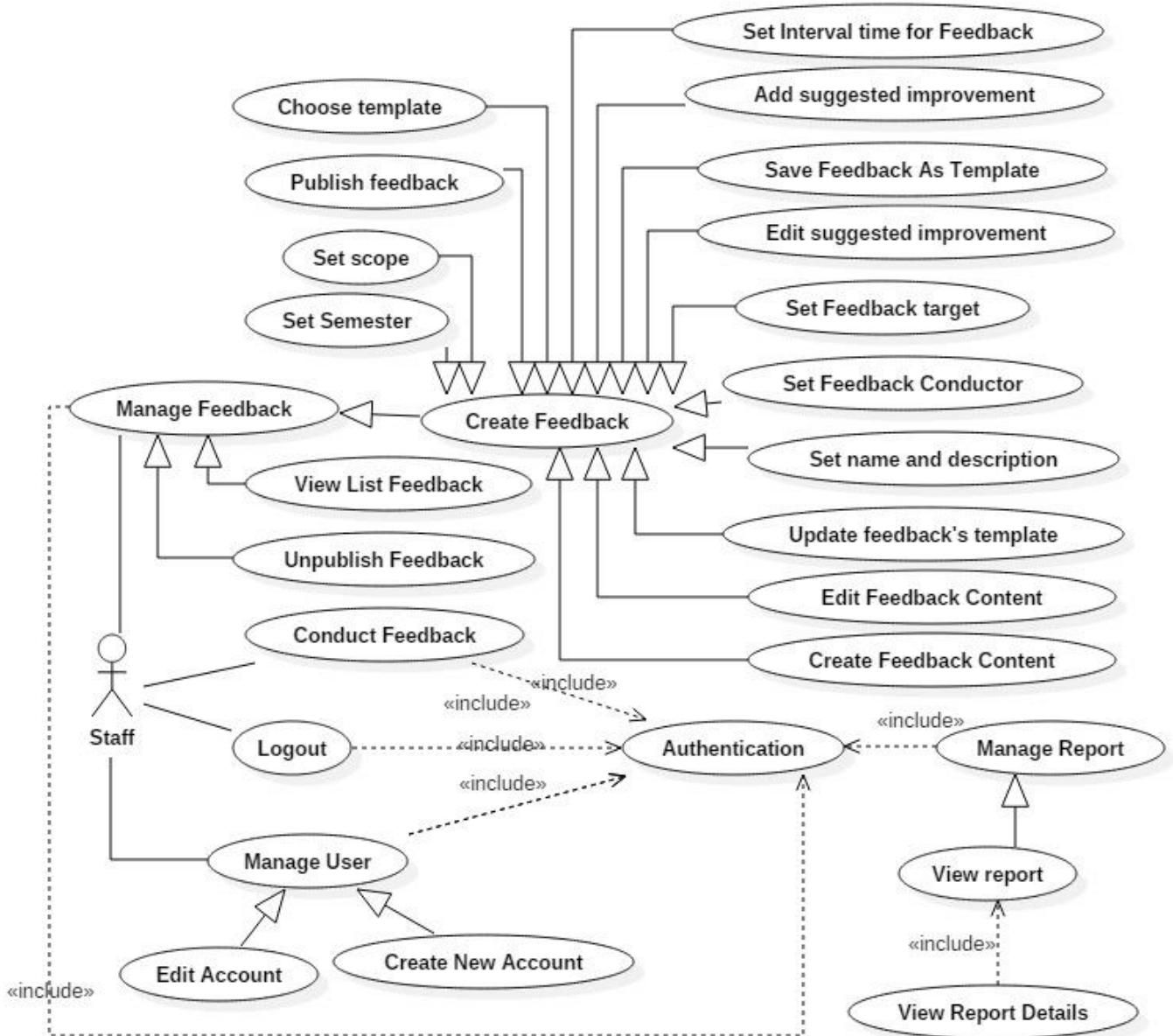


Figure 12 <Staff> Overview Use Case

2.3.4.1. Staff Create New Account

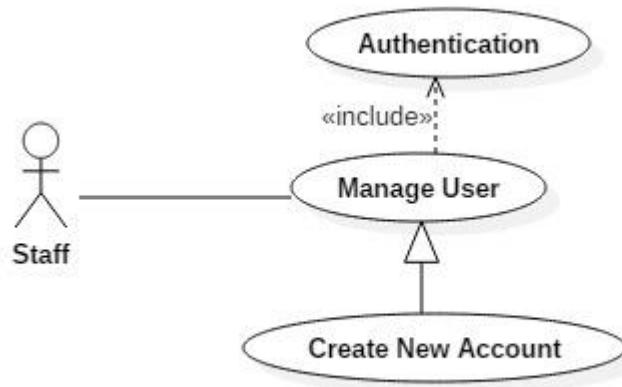


Figure 13 <Staff> Create new account

USE CASE - UC007			
Use Case No.	007	Use Case Version	1.0
Use Case Name	Create new account		
Author	VietHQ		
Date	27/03/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 		
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to create new account. 		
Goal:	<ul style="list-style-type: none"> - Account is created successfully and store in database of the system. 		
Triggers:	<ul style="list-style-type: none"> - Actor clicks on create new account button. 		
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: New account is created. - Fail: Account is not created. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Actor sends create new account command.	System requires information: <ul style="list-style-type: none"> - Username: free text input, required, length (9-20), unique - Password: free text input, required, length (9-20) - Full name: free text input, required, length (10-50) - Birthdate: date - Email: free text input, email format, unique - Sex: male or female - Role: Student, Lecturer, Staff, or Head of Academic - Major: Only for Lecturer and Student - Department: Only for Staff and Student 	

		<ul style="list-style-type: none"> - Status: Active or Deactive - Code
2	Actor inputs information and sends command to "Save". [Alternative 1]	<p>Account created.</p> <p>[Exception 1] [Exception 2] [Exception 3] [Exception 4]</p>

Alternative Scenario 1:

Step	Actor Action	System Response
	Actor sends command to reset.	System reset all field to blank.

Exceptions :

No	Cause	System Response
1	Actor inputs username already exist.	System show warning message "Tên đăng nhập này đã có người sử dụng".
2	Actor inputs email already exist.	System show warning message "Mail này đã được sử dụng".
3	Actor does not input required field.	<p>System notices that actor need to input all these field:</p> <ul style="list-style-type: none"> - "Full name" - "Username" - "Password" - "Code" - "Email"
4	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.

Relationships: N/A

Business Rules:

- After creating new account and active, user can login with new account.
- If choose role "Lecturer" or "Student", actor can choose field "Major", field "Department" will be disabled.
- If choose role "Staff" or "Head of Academic", actor can choose field "Department", field "Major" will be disabled.

2.3.4.2. Staff Edit Account

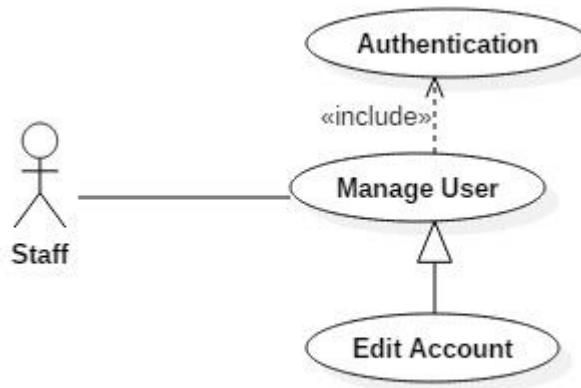


Figure 14 <Staff> Edit account

USE CASE - UC008			
Use Case No.	008	Use Case Version	1.0
Use Case Name	Edit account		
Author	VietHQ		
Date	27/03/2018	Priority	High

Actor:

- Staff
- Head of Academic

Summary:

- This use case allows Actor to edit account.

Goal:

- Profile of account is updated.

Triggers:

- Actor clicks on edit account button.

Preconditions:

- Actor has been authorized and accessed the system with proper role.

Post Conditions:

- **Success:** Information of account is updated and display.
- **Fail:** Information of account is not updated.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor sends edit account command.	<p>System requires information:</p> <ul style="list-style-type: none"> - Username: free text input, required, length (9-20), unique - Password: free text input, required, length (9-20) - Full name: free text input, required, length (10-50) - Birthdate: date - Email: free text input, email format, unique - Sex: male or female - Role: Student, Lecturer, Staff, or Head of Academic - Major: Only for Lecturer and Student - Department: Only for Staff and Student - Status: Active or Deactive - Code
2	Actor inputs information and sends command to "Save". [Alternative 1]	<p>Account updated.</p> <p>[Exception 1] [Exception 2] [Exception 3]</p>

Alternative Scenario 1:

Step	Actor Action	System Response
	Actor sends command to reset.	System reset all field to blank.

Exceptions :

No	Cause	System Response
1	Actor inputs email already exist.	System show warning message "Mail này đã được sử dụng".
2	Actor does not input required field.	<p>System notices that actor need to input all these field:</p> <ul style="list-style-type: none"> - "Full name"

		- "Password" - "Email"
3	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.
Relationships: N/A		
Business Rules:		
<ul style="list-style-type: none"> - After edited, new information will be save to profile of account. - Actor can't change Username, Code field. 		

2.3.4.3. Staff Create New Feedback Content

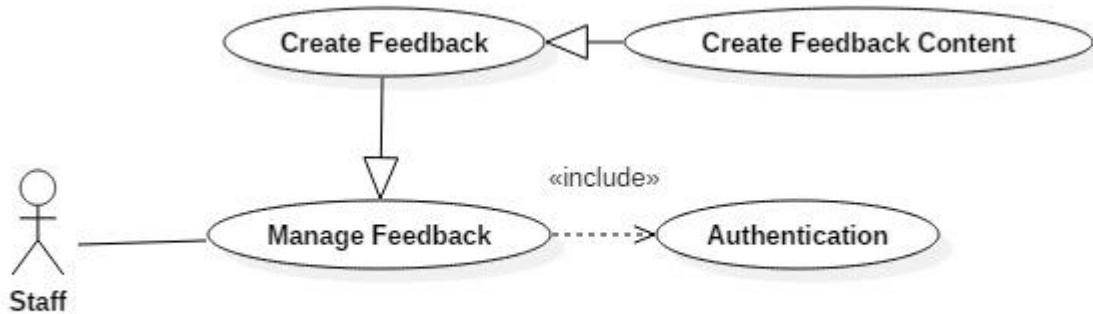


Figure 15 <Staff> Create New Feedback Content

USE CASE - UC009			
Use Case No.	009	Use Case Version	1.0
Use Case Name	Create Feedback Content		
Author	TuNV		
Date	27/03/2018	Priority	High
Actor:			
<ul style="list-style-type: none"> - Staff - Head of Academic 			
Summary:			
<ul style="list-style-type: none"> - This use case allows Actor to create new feedback content. 			
Goal:			
<ul style="list-style-type: none"> - Actor successfully creates new feedback content. 			
Triggers:			
<ul style="list-style-type: none"> - Actor clicks on create new feedback button. 			
Preconditions:			
<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 			
Post Conditions:			
<ul style="list-style-type: none"> - Success: New Feedback with created content is stored in database. - Fail: New Feedback content cannot be created. 			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Actor sends create blank new feedback command. [Alternative 1]	System displays create blank new feedback screen offering 4 types of feedback elements: radio, multiple choices, text or paragraph.	
2	Actor drags and drops suitable feedback elements to create content.	System requires information: <ul style="list-style-type: none"> - Radio: Question, Options, "Other" or not, "Required" or not, Point of each Option, Criteria 	

		<ul style="list-style-type: none"> - Multiple Choices: Question, Options, "Other" or not, "Required" or not, Point of each Option, Criteria - Text: Question, Criteria - Paragraph: Question, Criteria
2	Actor sends command to "Save". [Alternative 2] [Alternative 3] [Alternative 4]	Feedback created. [Exception 1]

Alternative Scenario 1:

Step	Actor Action	System Response
	Actor creates new feedback from an existed template.	System loads and displays questions and options from chosen template.

Alternative Scenario 2:

Step	Actor Action	System Response
	Actor sends command to delete a created question.	System deletes chosen question.

Alternative Scenario 3:

Step	Actor Action	System Response
	Actor sends command to delete a created option.	System deletes chosen option.

Alternative Scenario 4:

Step	Actor Action	System Response
	Actor drags question to reorder.	System reorders chosen questions.

Exceptions :

No	Cause	System Response
1	Actor does not input required field.	<p>System notices that actor need to input all these field:</p> <ul style="list-style-type: none"> - Question - Option - Point

Relationships: N/A

Business Rules:

- After clicking save, all questions and options will be stored in database.
- Point will be limit from 1 to 5 points.

2.3.4.4. Staff Edit Feedback Content

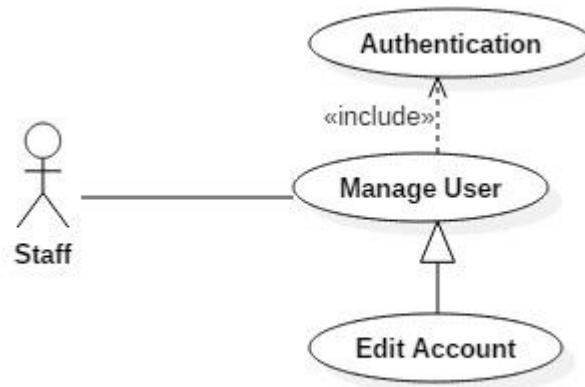


Figure 16 <Staff> Edit Feedback Content

USE CASE - UC010			
Use Case No.	010	Use Case Version	1.0
Use Case Name	Edit Feedback Content		
Author	TuNV		
Date	27/03/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 		
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to edit feedback content. 		
Goal:	<ul style="list-style-type: none"> - Actor successfully edits feedback content. 		
Triggers:	<ul style="list-style-type: none"> - Actor clicks on edit feedback button. 		
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: New contents of feedback are stored in database. - Fail: New content cannot be edited. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Actor sends edit feedback command.	System displays edit feedback screen offering 4 types of feedback elements: radio, multiple choices, text or paragraph.	
2	Actor drags and drops suitable feedback elements to create content.	System requires information: <ul style="list-style-type: none"> - Radio: Question, Options, "Other" or not, "Required" or not, Point of each Option, Criteria - Multiple Choices: Question, Options, "Other" or not, "Required" or not, Point of each Option, Criteria - Text: Question, Criteria - Paragraph: Question, Criteria 	
2	Actor sends command to "Save". [Alternative 1] [Alternative 2] [Alternative 3]	Feedback updated. [Exception 1]	

Alternative Scenario 1:

Step	Actor Action	System Response
	Actor sends command to delete a created question.	System deletes chosen question.

Alternative Scenario 2:

Step	Actor Action	System Response
	Actor sends command to delete a created option.	System deletes chosen option.

Alternative Scenario 3:

Step	Actor Action	System Response
	Actor drags question to reorder.	System reorders chosen questions.

Exceptions :

No	Cause	System Response
1	Actor does not input required field.	System notices that actor need to input all these field: - Question - Option - Point

Relationships: N/A

Business Rules:

- After clicking save, all questions and options will be updated in database.
- Point will be limit from 1 to 5 points.
- Edit Feedback screen contains all created questions and options before

2.3.4.5. Staff Choose Template

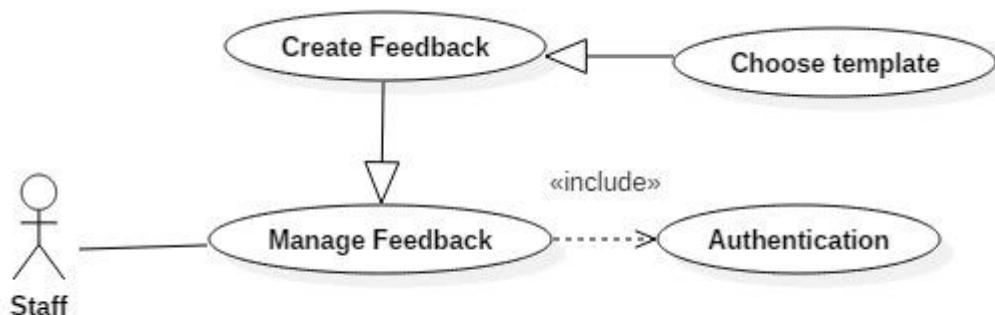


Figure 17 <Staff> Choose Template

USE CASE - UC011			
Use Case No.	011	Use Case Version	1.0
Use Case Name	Choose Template		
Author	TuNV		
Date	27/03/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 		
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to choose feedback template. 		

Goal:	<ul style="list-style-type: none"> - Actor successfully choose feedback template. 							
Triggers:	<ul style="list-style-type: none"> - Actor clicks on chosen template. 							
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 							
Post Conditions:	<ul style="list-style-type: none"> - Success: Chosen template has been successfully loaded. - Fail: Cannot choose template. 							
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor sends choose template command. [Alternative 1]</td><td>System chooses corresponding template and loads template content.</td></tr> </tbody> </table>		Step	Actor Action	System Response	1	Actor sends choose template command. [Alternative 1]	System chooses corresponding template and loads template content.
Step	Actor Action	System Response						
1	Actor sends choose template command. [Alternative 1]	System chooses corresponding template and loads template content.						
Alternative Scenario 1:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td></td><td>Actor clicks on "Preview button"</td><td>System loads template content and display to actor.</td></tr> </tbody> </table>		Step	Actor Action	System Response		Actor clicks on "Preview button"	System loads template content and display to actor.
Step	Actor Action	System Response						
	Actor clicks on "Preview button"	System loads template content and display to actor.						
Relationships: N/A								
Business Rules:	<ul style="list-style-type: none"> - System will display update template screen with all existed questions. 							

2.3.4.6. Staff Set Feedback Name and Description

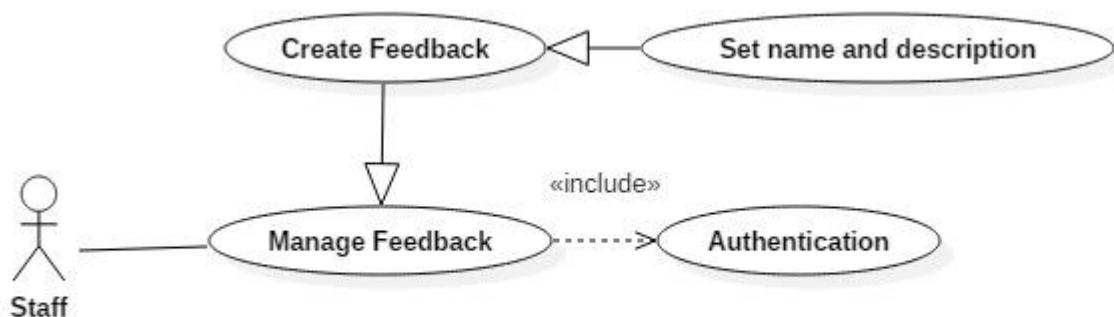


Figure 18 <Staff> Set Feedback Name and Description

USE CASE – UC012			
Use Case No.	012	Use Case Version	1.0
Use Case Name	Set Feedback Name and Description		
Author	TuNV		
Date	27/03/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 		
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to set Feedback name and description. 		
Goal:	<ul style="list-style-type: none"> - Actor successfully sets Feedback name and description. 		
Triggers:	<ul style="list-style-type: none"> - Actor sends set Feedback name and description command. 		

Preconditions:

- Actor has been authorized and accessed the system with proper role.

Post Conditions:

- **Success:** Feedback name and description are stored in database.
- **Fail:** Feedback name and description cannot be set.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor sends set Feedback name and description command. [Alternative 1]	System requires information: <ul style="list-style-type: none"> - Title: free text input, required, length (9-20) - Description: free text input, required, length (9-150)
2	Actor inputs information and sends command to "Save".	Feedback Name and Description are saved. [Exception 1] [Exception 2]

Alternative Scenario 1:

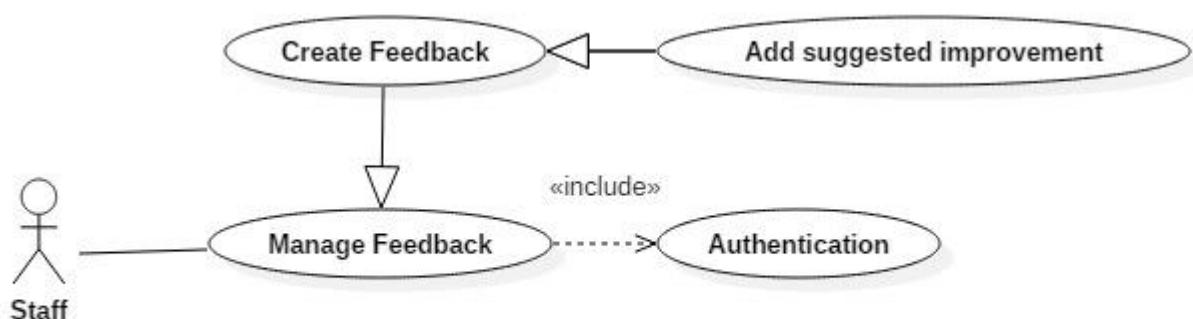
Step	Actor Action	System Response
	Actor uses existed name and description from template.	System loads existed name and description from template.

Exceptions :

No	Cause	System Response
1	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.
2	Actor does not input required field.	System notices that actor need to input all these field: <ul style="list-style-type: none"> - Title - Description

Relationships: N/A**Business Rules:**

N/A

2.3.4.7. Staff Add Suggested Improvement**Figure 19 <Staff> Add Suggested Improvement**

USE CASE - UC013					
Use Case No.	013	Use Case Version	1.0		
Use Case Name	Add suggested improvement				
Author	TuNV				
Date	27/03/2018	Priority	High		
Actor:					
<ul style="list-style-type: none"> - Staff - Head of Academic 					
Summary:					
<ul style="list-style-type: none"> - This use case allows Actor to add suggested improvement. 					
Goal:					
<ul style="list-style-type: none"> - Actor successfully adds suggested improvement. 					
Triggers:					
<ul style="list-style-type: none"> - Actor clicks add suggested improvement button. 					
Preconditions:					
<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 					
Post Conditions:					
<ul style="list-style-type: none"> - Success: Suggested improvements are successfully stored in database. - Fail: Suggested improvement cannot be saved. 					
Main Success Scenario:					
Step	Actor Action	System Response			
1	Actor sends add suggested improvement command. [Alternative 1]	System loads all created questions and requires improvement for each question: <ul style="list-style-type: none"> - Improvement: free text input, length (9-200) 			
2	Actor inputs information and sends command to "Save".	Feedback suggested improvements are saved. [Exception 1] [Exception 2]			
Alternative Scenario 1:					
Step	Actor Action	System Response			
	Actor chose an existed template and adds improvements.	System loads existed questions and improvements from chosen template.			
Exceptions :					
No	Cause	System Response			
1	Feedback has no created questions.	System shows error questions "Không có câu hỏi nào"			
2	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.			
Relationships: N/A					
Business Rules:					
<ul style="list-style-type: none"> - Improvements will be displayed in report if point of question is lower than average point. 					

2.3.4.8. Staff Edit Suggested Improvement

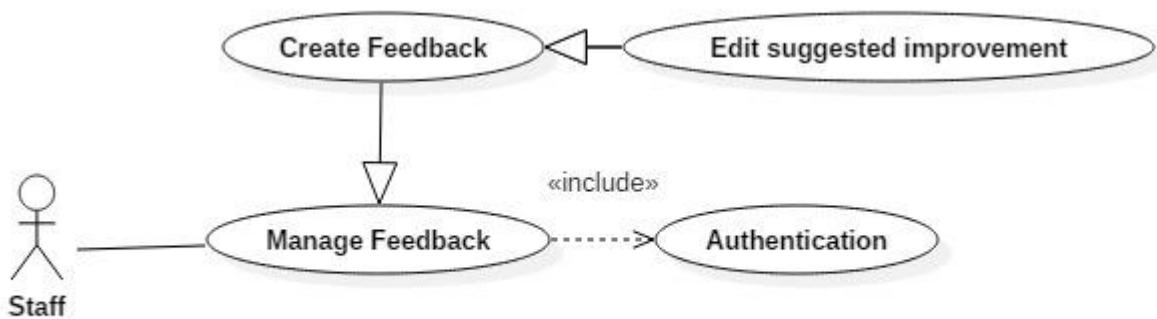


Figure 20 <Staff> Edit Suggested Improvement

USE CASE - UC014			
Use Case No.	014	Use Case Version	1.0
Use Case Name	Edit suggested improvement		
Author	TuNV		
Date	27/03/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 		
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to edit suggested improvement. 		
Goal:	<ul style="list-style-type: none"> - Actor successfully edits suggested improvement. 		
Triggers:	<ul style="list-style-type: none"> - Actor clicks edit suggested improvement button. 		
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Suggested improvements are successfully updated in database. - Fail: Suggested improvement cannot be update. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Actor sends edit suggested improvement command.	System loads all created questions and improvements - Improvement: free text input, length (9-200)	
2	Actor inputs information and sends command to "Save".	Feedback suggested improvements are updated. [Exception 1] [Exception 2]	

Exceptions :

No	Cause	System Response
1	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.
2	Feedback has no created questions.	System shows error questions "Không có câu hỏi nào"

Relationships: N/A

Business Rules:

- Improvements will be displayed in report if point of question is lower than average point.

2.3.4.9. Staff Set Feedback Semester

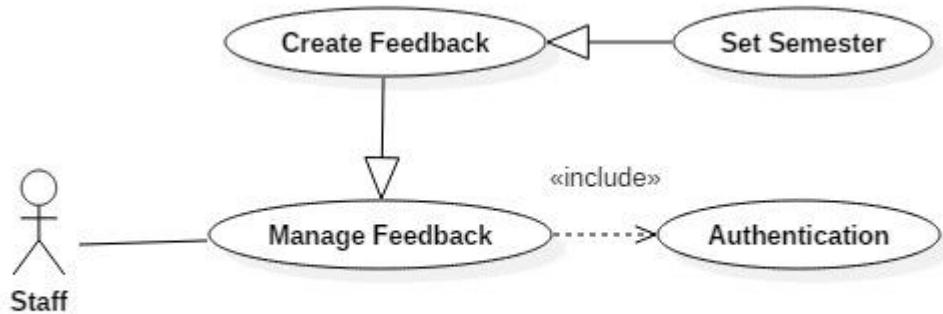


Figure 21 <Staff> Set Feedback Semester

USE CASE - UC015

Use Case No.	015	Use Case Version	1.0
Use Case Name	Set Feedback Semester		
Author	ThuanTHM		
Date	27/03/2018	Priority	High

Actor:

- Staff
- Head of Academic

Summary:

- This use case allows Actor to set Feedback semester.

Goal:

- Actor successfully sets Feedback semester.

Triggers:

- Actor clicks set Feedback semester button.

Preconditions:

- Actor has been authorized and accessed the system with proper role.

Post Conditions:

- **Success:** Feedback semester is successfully saved in database.
- **Fail:** Feedback semester cannot be saved.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor sends feedback semester command.	System requires information: - Semester: Dropdown list
2	Actor inputs information and sends command to "Save".	Feedback semester is saved. [Exception 1]

Exceptions :

No	Cause	System Response
1	Actor does not input required field.	System notices that actor need to input semester field.

Relationships: N/A

Business Rules:

- Feedback semester identifies the semester that feedback is available in.

2.3.4.10. Staff Set Feedback Interval Time

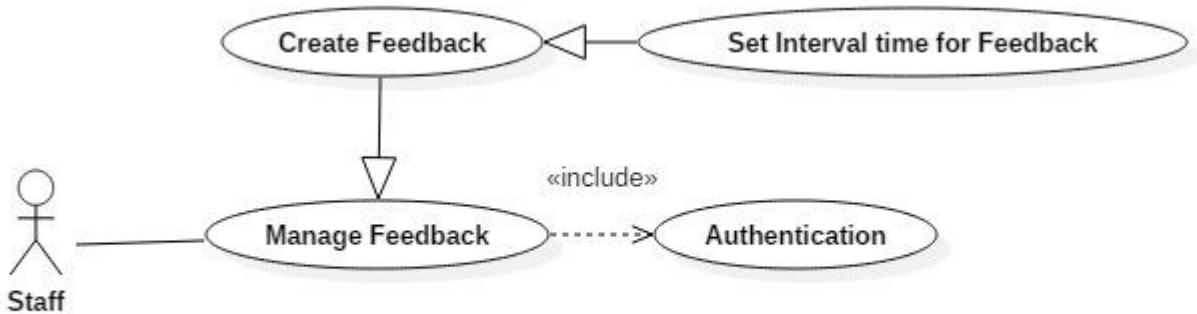


Figure 22 <Staff> Set Feedback Interval Time

USE CASE - UC016

Use Case No.	016	Use Case Version	1.0
Use Case Name	Set Feedback Interval Time		
Author	ThuanTHM		
Date	27/03/2018	Priority	High

Actor:

- Staff
- Head of Academic

Summary:

- This use case allows Actor to set Feedback Interval Time.

Goal:

- Actor successfully sets Feedback Interval Time.

Triggers:

- Actor clicks set Feedback Interval Time button.

Preconditions:

- Actor has been authorized and accessed the system with proper role.

Post Conditions:

- **Success:** Feedback Interval Time is successfully saved in database.
- **Fail:** Feedback Interval Time cannot be saved.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor sends Feedback Interval Time command.	System requires information: - From: Date picker - To: Date picker
2	Actor inputs information and sends command to "Save".	Feedback Interval Time is saved. [Exception 1] [Exception 2]

Exceptions :

No	Cause	System Response
1	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.

2	Actor does not input required field.	System notices that actor need to input semester field.
---	--------------------------------------	---

Relationships: N/A

Business Rules:

- Interval time is the period of time that Feedback is available to conduct.
- Interval time must be between start and end date of chosen semester.

2.3.4.11. Staff Set Feedback Scope

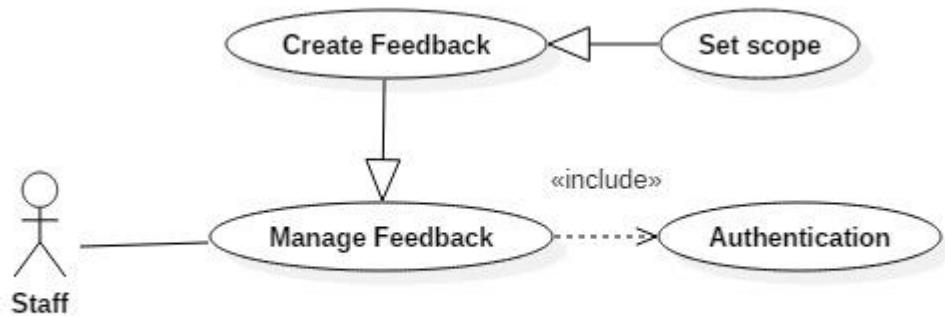


Figure 23 <Staff> Set Feedback Scope

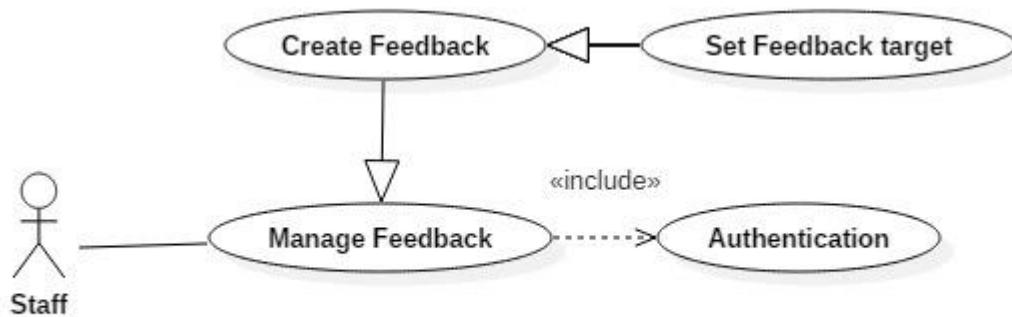
USE CASE - UC017												
Use Case No.	017	Use Case Version	1.0									
Use Case Name	Set Feedback Scope											
Author	ThuanTHM											
Date	27/03/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 											
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to set Feedback Scope. 											
Goal:	<ul style="list-style-type: none"> - Actor successfully sets Feedback Scope. 											
Triggers:	<ul style="list-style-type: none"> - Actor clicks set Feedback Scope button. 											
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: Feedback Scope is successfully saved in database. - Fail: Feedback Scope cannot be saved. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor sends Feedback Scope command.</td><td> System requires information: - Scope: Dropdown list </td></tr> <tr> <td>2</td><td>Actor inputs information and sends command to "Save".</td><td> Feedback Scope is saved. [Exception 1] [Exception 2] </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Actor sends Feedback Scope command.	System requires information: - Scope: Dropdown list	2	Actor inputs information and sends command to "Save".	Feedback Scope is saved. [Exception 1] [Exception 2]
Step	Actor Action	System Response										
1	Actor sends Feedback Scope command.	System requires information: - Scope: Dropdown list										
2	Actor inputs information and sends command to "Save".	Feedback Scope is saved. [Exception 1] [Exception 2]										

Exceptions :

No	Cause	System Response
2	Actor does not input required field.	System notices that actor need to input semester field.

Relationships: N/A**Business Rules:**

- Feedback Scope can be Major, Course, Department, Class.
- Each Feedback can only have one scope.

2.3.4.12. Staff Set Feedback Target**Figure 24 <Staff> Set Feedback Target****USE CASE - UC018**

Use Case No.	018	Use Case Version	1.0
Use Case Name	Set Feedback Target		
Author	ThuanTHM		
Date	27/03/2018	Priority	High

Actor:

- Staff
- Head of Academic

Summary:

- This use case allows Actor to set Feedback Target.

Goal:

- Actor successfully sets Feedback Target.

Triggers:

- Actor clicks set Feedback Target button.

Preconditions:

- Actor has been authorized and accessed the system with proper role.

Post Conditions:

- **Success:** Feedback Target is successfully saved in database.
- **Fail:** Feedback Target cannot be saved.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor sends Feedback Target command.	System displays list Feedback target that suitable to existed scope. [Exception 1]
2	Actor choose Target and sends command to "Save". [Alternative 1] [Alternative 2]	Feedback target is saved. [Exception 2]

	[Alternative 3]										
Alternative Scenario 1:											
<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td></td><td>Actor clicks on button "Deselect" target.</td><td>System deselects that target</td></tr> </tbody> </table>			Step	Actor Action	System Response		Actor clicks on button "Deselect" target.	System deselects that target			
Step	Actor Action	System Response									
	Actor clicks on button "Deselect" target.	System deselects that target									
Alternative Scenario 2:											
<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td></td><td>Actor clicks on "Delete" button.</td><td>System deletes that chosen target</td></tr> </tbody> </table>			Step	Actor Action	System Response		Actor clicks on "Delete" button.	System deletes that chosen target			
Step	Actor Action	System Response									
	Actor clicks on "Delete" button.	System deletes that chosen target									
Alternative Scenario 3:											
<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td></td><td>Actor chooses another scope</td><td>System removes all selected targets in previous scope</td></tr> </tbody> </table>			Step	Actor Action	System Response		Actor chooses another scope	System removes all selected targets in previous scope			
Step	Actor Action	System Response									
	Actor chooses another scope	System removes all selected targets in previous scope									
Exceptions :											
<table border="1"> <thead> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>There is no suitable target for chosen scope</td><td>System shows error message "Không có đối tượng nào"</td></tr> <tr> <td>2</td><td>Actor does not choose any target.</td><td>System notices that actor need to choose at least one target.</td></tr> </tbody> </table>			No	Cause	System Response	1	There is no suitable target for chosen scope	System shows error message "Không có đối tượng nào"	2	Actor does not choose any target.	System notices that actor need to choose at least one target.
No	Cause	System Response									
1	There is no suitable target for chosen scope	System shows error message "Không có đối tượng nào"									
2	Actor does not choose any target.	System notices that actor need to choose at least one target.									
Relationships: N/A											
Business Rules:											
<ul style="list-style-type: none"> - Actor can choose multiple targets for one feedback. - All targets that are chosen for a feedback must be in the same Scope. 											

2.3.4.13. Staff Set Feedback Conductor

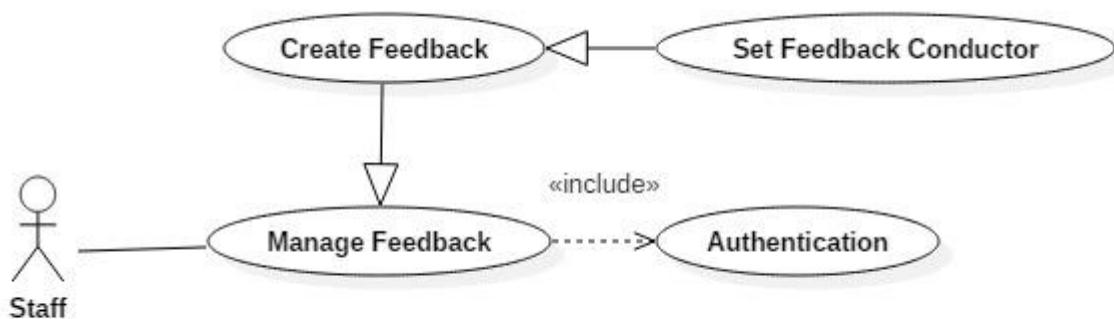


Figure 25 <Staff> Set Feedback Conductor

USE CASE - UC019			
Use Case No.	019	Use Case Version	1.0
Use Case Name	Set Feedback Conductor		
Author	ThuanTHM		
Date	27/03/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 		

Summary:

- This use case allows Actor to set Feedback Conductor.

Goal:

- Actor successfully sets Feedback Conductor.

Triggers:

- Actor clicks set Feedback Conductor button.

Preconditions:

- Actor has been authorized and accessed the system with proper role.

Post Conditions:

- **Success:** Feedback Conductor is successfully saved in database.
- **Fail:** Feedback Conductor cannot be saved.

Main Success Scenario:

Step	Actor Action	System Response
1	Actor sends Feedback Conductor command.	System displays list Feedback Conductor that suitable to existed target. [Exception 1]
2	Actor choose Conductor and sends command to "Save". [Alternative 1] [Alternative 2]	Feedback Conductor is saved. [Exception 2]

Alternative Scenario 1:

Step	Actor Action	System Response
	Actor clicks on button "Deselect" conductor.	System deselects that conductor

Alternative Scenario 2:

Step	Actor Action	System Response
	Actor clicks on "Delete" target button.	System deletes that all selected conductors for that target.

Exceptions :

No	Cause	System Response
1	There is no suitable conductors for chosen target	System shows error message "Không có người thực hiện nào"
2	Actor does not choose any conductor.	System notices that actor need to choose at least one conductor.

Relationships: N/A**Business Rules:**

- Actor can choose multiple conductors for one target.
- All corresponding conductors for that target must be auto selected.

2.3.4.14. Staff Publish Feedback

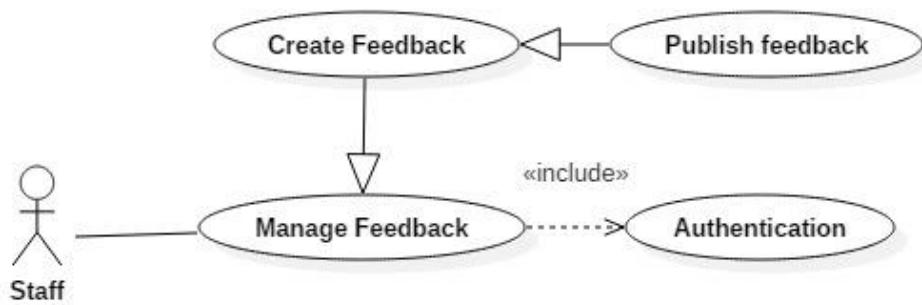


Figure 26 <Staff> Publish Feedback

USE CASE - UC020												
Use Case No.	020	Use Case Version	1.0									
Use Case Name	Publish Feedback											
Author	ThuanTHM											
Date	27/03/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 											
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to publish Feedback. 											
Goal:	<ul style="list-style-type: none"> - Actor successfully publishes Feedback. 											
Triggers:	<ul style="list-style-type: none"> - Actor clicks Publish Feedback button. 											
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: Feedback is successfully saved in database and published. - Fail: Feedback cannot be saved and published. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor sends Publish Feedback command.</td><td>System saves all Feedback Information and publishes Feedback. [Exception 1] [Exception 2]</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Actor sends Publish Feedback command.	System saves all Feedback Information and publishes Feedback. [Exception 1] [Exception 2]			
Step	Actor Action	System Response										
1	Actor sends Publish Feedback command.	System saves all Feedback Information and publishes Feedback. [Exception 1] [Exception 2]										
Exceptions :	<table border="1"> <thead> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor does not input required field.</td><td>System notices that actor need to input all these fields.</td></tr> <tr> <td>2</td><td>Actor inputs wrong some fields with requirement.</td><td>System notices that actor need to re-input all those fields.</td></tr> </tbody> </table>			No	Cause	System Response	1	Actor does not input required field.	System notices that actor need to input all these fields.	2	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.
No	Cause	System Response										
1	Actor does not input required field.	System notices that actor need to input all these fields.										
2	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.										
Relationships:	N/A											
Business Rules:	<ul style="list-style-type: none"> - When Feedback is published, it will be available for corresponding conductors to do. - Once feedback is published, it cannot be edited anymore. 											

2.3.4.15. Staff Save Feedback as Template

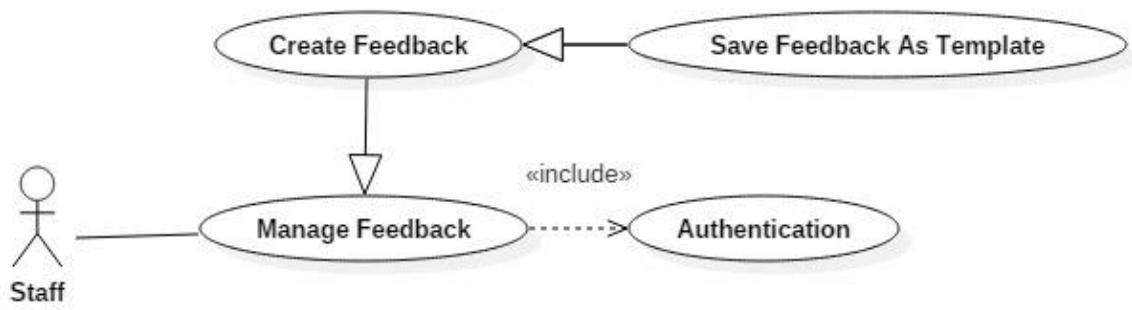


Figure 27 <Staff> Save Feedback as Template

USE CASE - UC021												
Use Case No.	021	Use Case Version	1.0									
Use Case Name	Save Feedback as Template											
Author	ThuanTHM											
Date	27/03/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 											
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to Save Feedback as Template. 											
Goal:	<ul style="list-style-type: none"> - Actor successfully Save Feedback as Template. 											
Triggers:	<ul style="list-style-type: none"> - Actor clicks Save Feedback as Template button. 											
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: Feedback is successfully saved as template in database. - Fail: Feedback cannot be saved as template. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor sends Save Feedback as Template command.</td><td> System saves all Feedback Information as new template Feedback. [Exception 1] [Exception 2] </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Actor sends Save Feedback as Template command.	System saves all Feedback Information as new template Feedback. [Exception 1] [Exception 2]			
Step	Actor Action	System Response										
1	Actor sends Save Feedback as Template command.	System saves all Feedback Information as new template Feedback. [Exception 1] [Exception 2]										
Exceptions:	<table border="1"> <thead> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor does not input required field.</td><td>System notices that actor need to input all these fields.</td></tr> <tr> <td>2</td><td>Actor inputs wrong some fields with requirement.</td><td>System notices that actor need to re-input all those fields.</td></tr> </tbody> </table>			No	Cause	System Response	1	Actor does not input required field.	System notices that actor need to input all these fields.	2	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.
No	Cause	System Response										
1	Actor does not input required field.	System notices that actor need to input all these fields.										
2	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.										
Relationships:	N/A											
Business Rules:	<ul style="list-style-type: none"> - When Feedback is saved as template, actor can choose it again in Select template screen 											

2.3.4.16. Staff Update Feedback Template

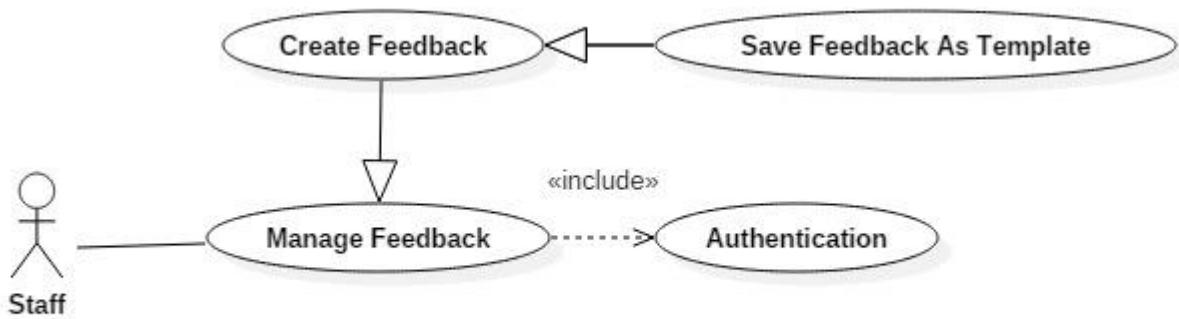


Figure 28 <Staff> Update Feedback Template

USE CASE - UC022												
Use Case No.	022	Use Case Version	1.0									
Use Case Name	Update Feedback Template											
Author	ThuanTHM											
Date	27/03/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 											
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to Update Feedback Template. 											
Goal:	<ul style="list-style-type: none"> - Actor successfully Update Feedback Template. 											
Triggers:	<ul style="list-style-type: none"> - Actor clicks Update Feedback Template button. 											
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: Template is successfully updated in database. - Fail: Template cannot be updated. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor sends Update Feedback Template command.</td><td>System updates all Template Information. [Exception 1] [Exception 2]</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Actor sends Update Feedback Template command.	System updates all Template Information. [Exception 1] [Exception 2]			
Step	Actor Action	System Response										
1	Actor sends Update Feedback Template command.	System updates all Template Information. [Exception 1] [Exception 2]										
Exceptions :	<table border="1"> <thead> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor does not input required field.</td><td>System notices that actor need to input all these fields.</td></tr> <tr> <td>2</td><td>Actor inputs wrong some fields with requirement.</td><td>System notices that actor need to re-input all those fields.</td></tr> </tbody> </table>			No	Cause	System Response	1	Actor does not input required field.	System notices that actor need to input all these fields.	2	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.
No	Cause	System Response										
1	Actor does not input required field.	System notices that actor need to input all these fields.										
2	Actor inputs wrong some fields with requirement.	System notices that actor need to re-input all those fields.										
Relationships: N/A												
Business Rules:	<ul style="list-style-type: none"> - Template is updated. Actor can chose template again in list template, customize and publish or save as another template. 											

2.3.4.17. Staff Unpublish Feedback

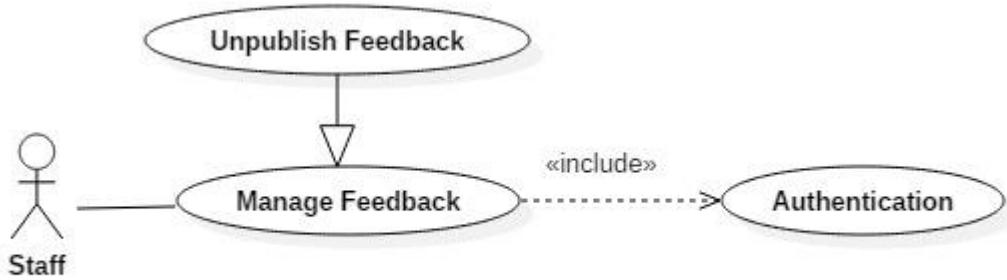


Figure 29 <Staff> Unpublish Feedback

USE CASE - UC023												
Use Case No.	023	Use Case Version	1.0									
Use Case Name	Unpublish Feedback											
Author	VietHQ											
Date	27/03/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Staff - Head of Academic 											
Summary:	<ul style="list-style-type: none"> - This use case allows Actor to Unpublish Feedback. 											
Goal:	<ul style="list-style-type: none"> - Actor successfully Unpublish Feedback. 											
Triggers:	<ul style="list-style-type: none"> - Actor clicks Unpublish Feedback button. 											
Preconditions:	<ul style="list-style-type: none"> - Actor has been authorized and accessed the system with proper role. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: Feedback is successfully unpublished. - Fail: Feedback cannot be unpublished. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor sends Unpublish Feedback command.</td><td>System confirms actor command.</td></tr> <tr> <td>2</td><td>Actor send confirm command. [Alternative 1]</td><td>System changes selected Feedback to unpublish state.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Actor sends Unpublish Feedback command.	System confirms actor command.	2	Actor send confirm command. [Alternative 1]	System changes selected Feedback to unpublish state.
Step	Actor Action	System Response										
1	Actor sends Unpublish Feedback command.	System confirms actor command.										
2	Actor send confirm command. [Alternative 1]	System changes selected Feedback to unpublish state.										
Alternative :	<table border="1"> <thead> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Actor send cancel action command.</td><td>System keeps Feedback publish state.</td></tr> </tbody> </table>			No	Cause	System Response	1	Actor send cancel action command.	System keeps Feedback publish state.			
No	Cause	System Response										
1	Actor send cancel action command.	System keeps Feedback publish state.										
Relationships: N/A												
Business Rules:	<ul style="list-style-type: none"> - Once Feedback is unpublished, all users cannot see or conduct that Feedback. - Report will not be generated for unpublished Feedback. - Actor can republish report after it is unpublished 											

2.3.5. Head of Academic Overview Use Case

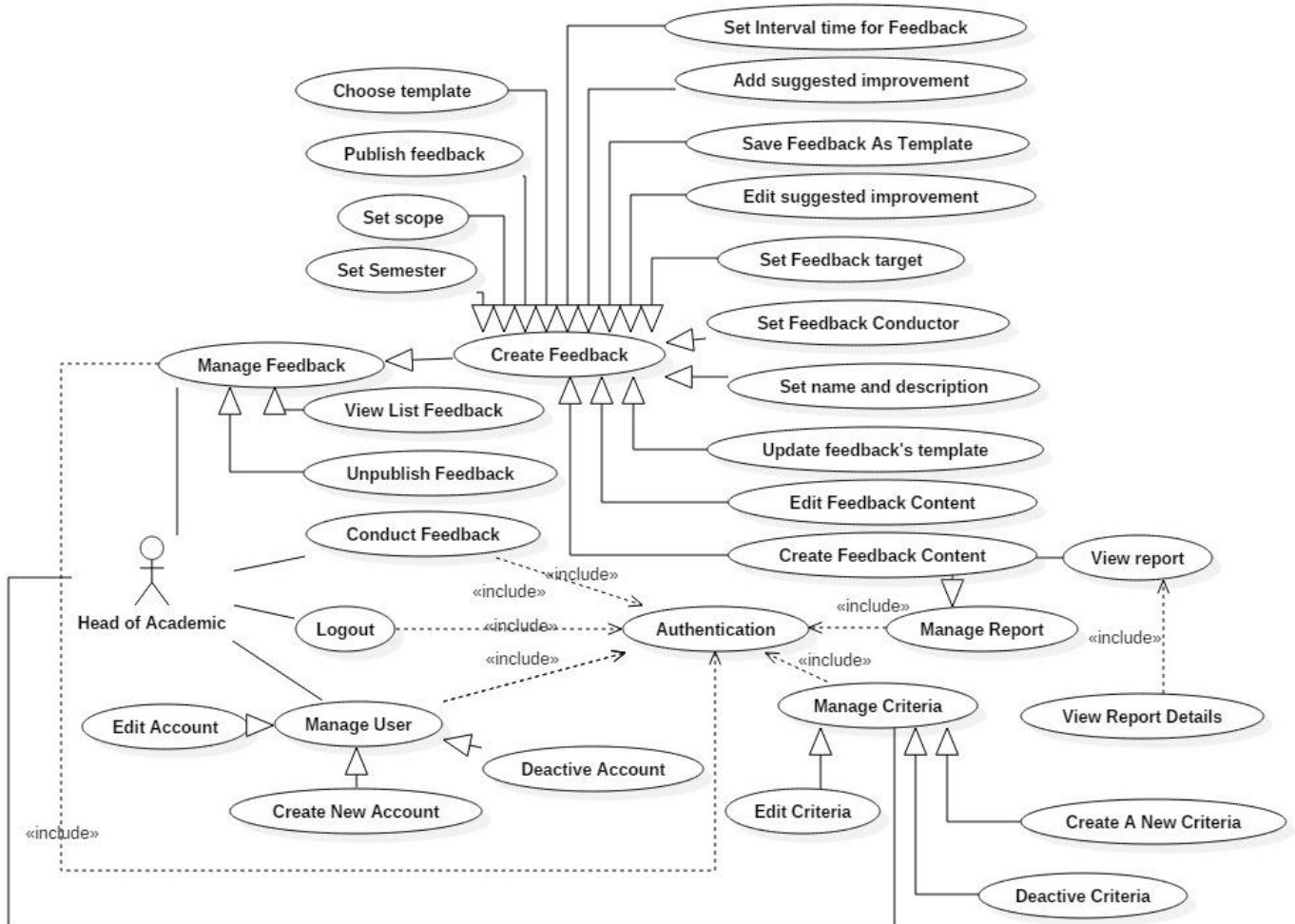


Figure 30 <Head of Academic> Overview Use Case

2.3.5.1. Head of Academic Create New Criteria

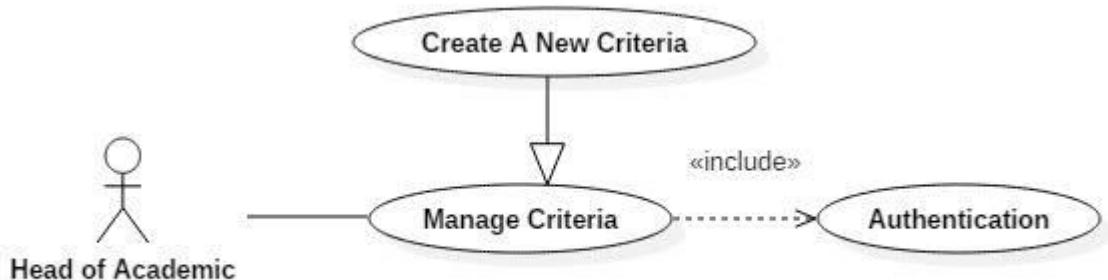


Figure 31 <Head of Academic> Create new criteria

USE CASE - UC024												
Use Case No.	024	Use Case Version	1.0									
Use Case Name	Create new criteria											
Author	ThuanTHM											
Date	03/04/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Head of Academic 											
Summary:	<ul style="list-style-type: none"> - This use case allows Head to create new criteria. 											
Goal:	<ul style="list-style-type: none"> - Criteria is created successfully and store in database of the system. 											
Triggers:	<ul style="list-style-type: none"> - Head clicks on create new criteria button. 											
Preconditions:	<ul style="list-style-type: none"> - Head has been authorized and accessed the system with proper role. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: New criteria is created. - Fail: Criteria is not created. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Head sends create new criteria command.</td><td>System requires information: <ul style="list-style-type: none"> - Criteria name - Status </td></tr> <tr> <td>2</td><td>Head inputs information and sends command to "Save".</td><td>Criteria created.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Head sends create new criteria command.	System requires information: <ul style="list-style-type: none"> - Criteria name - Status 	2	Head inputs information and sends command to "Save".	Criteria created.
Step	Actor Action	System Response										
1	Head sends create new criteria command.	System requires information: <ul style="list-style-type: none"> - Criteria name - Status 										
2	Head inputs information and sends command to "Save".	Criteria created.										
Alternative Scenario 1:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td></td><td>Head sends command to reset.</td><td>System reset all field to blank.</td></tr> </tbody> </table>			Step	Actor Action	System Response		Head sends command to reset.	System reset all field to blank.			
Step	Actor Action	System Response										
	Head sends command to reset.	System reset all field to blank.										
Exceptions :	<table border="1"> <thead> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Head does not input required field.</td><td>System notices that Head need to input all these field: <ul style="list-style-type: none"> - "Criteria name" </td></tr> </tbody> </table>			No	Cause	System Response	1	Head does not input required field.	System notices that Head need to input all these field: <ul style="list-style-type: none"> - "Criteria name" 			
No	Cause	System Response										
1	Head does not input required field.	System notices that Head need to input all these field: <ul style="list-style-type: none"> - "Criteria name" 										
Relationships: N/A												
Business Rules:	<ul style="list-style-type: none"> - After creating new criteria and active, actor can create question with new criteria. 											

2.3.5.2. Head of Academic Edit Criteria

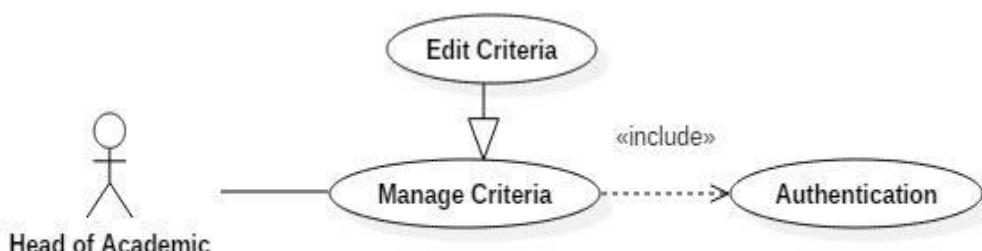


Figure 32 <Head of Academic> Edit criteria

USE CASE - UC025												
Use Case No.	025	Use Case Version	1.0									
Use Case Name	Edit Criteria											
Author	ThuanTHM											
Date	03/04/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Head of Academic 											
Summary:	<ul style="list-style-type: none"> - This use case allows Head to edit criteria. 											
Goal:	<ul style="list-style-type: none"> - Criteria is updated. 											
Triggers:	<ul style="list-style-type: none"> - Head clicks on edit criteria button. 											
Preconditions:	<ul style="list-style-type: none"> - Head has been authorized and accessed the system with proper role. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: Information of criteria is updated and display. - Fail: Information of criteria is not updated. 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Head sends edit criteria command.</td><td> System requires information: <ul style="list-style-type: none"> - Criteria name: free text input, required, length (9-20), unique - Status: Active or Deactive </td></tr> <tr> <td>2</td><td>Head inputs information and sends command to "Save".</td><td>Criteria updated.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Head sends edit criteria command.	System requires information: <ul style="list-style-type: none"> - Criteria name: free text input, required, length (9-20), unique - Status: Active or Deactive 	2	Head inputs information and sends command to "Save".	Criteria updated.
Step	Actor Action	System Response										
1	Head sends edit criteria command.	System requires information: <ul style="list-style-type: none"> - Criteria name: free text input, required, length (9-20), unique - Status: Active or Deactive 										
2	Head inputs information and sends command to "Save".	Criteria updated.										
Alternative Scenario 1:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td></td><td>Head sends command to reset.</td><td>System reset all field to blank.</td></tr> </tbody> </table>			Step	Actor Action	System Response		Head sends command to reset.	System reset all field to blank.			
Step	Actor Action	System Response										
	Head sends command to reset.	System reset all field to blank.										
Exceptions :	<table border="1"> <thead> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Head does not input required field.</td><td> System notices that Head need to input all these field: <ul style="list-style-type: none"> - "Criteria name" </td></tr> </tbody> </table>			No	Cause	System Response	1	Head does not input required field.	System notices that Head need to input all these field: <ul style="list-style-type: none"> - "Criteria name" 			
No	Cause	System Response										
1	Head does not input required field.	System notices that Head need to input all these field: <ul style="list-style-type: none"> - "Criteria name" 										
Relationships: N/A												
Business Rules:	<ul style="list-style-type: none"> - After edited, new information will be save to database. 											

2.3.5.3. Head of Academic Deactivate Edit

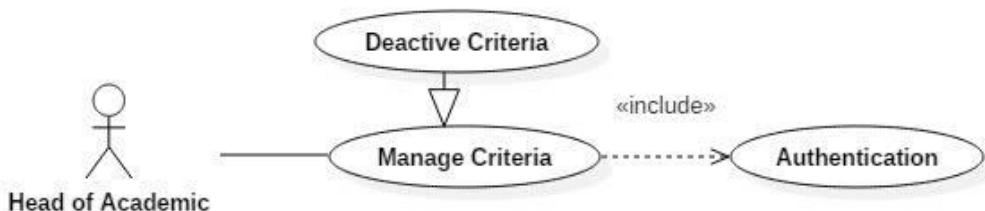


Figure 33 <Head of Academic> Deactivate criteria

USE CASE - UC026												
Use Case No.	026	Use Case Version	1.0									
Use Case Name	Deactivate Criteria											
Author	ThuanTHM											
Date	03/04/2018	Priority	High									
Actor:	<ul style="list-style-type: none"> - Head of Academic 											
Summary:	<ul style="list-style-type: none"> - This use case allows Head to deactivate criteria. 											
Goal:	<ul style="list-style-type: none"> - Criteria is deactivated. 											
Triggers:	<ul style="list-style-type: none"> - Head clicks on deactivate criteria button. 											
Preconditions:	<ul style="list-style-type: none"> - Head has been authorized and accessed the system with proper role. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: Criteria is deactivated - Fail: Criteria is not deactivated 											
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Head sends deactivate criteria command.</td><td>System requires information: - Criteria Id</td></tr> <tr> <td>2</td><td>Head hits confirm button</td><td>Criteria deactivated.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Head sends deactivate criteria command.	System requires information: - Criteria Id	2	Head hits confirm button	Criteria deactivated.
Step	Actor Action	System Response										
1	Head sends deactivate criteria command.	System requires information: - Criteria Id										
2	Head hits confirm button	Criteria deactivated.										
Exceptions :	<table border="1"> <thead> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Id does not exist in database</td><td>System notices that cannot find criteria in database -</td></tr> </tbody> </table>			No	Cause	System Response	1	Id does not exist in database	System notices that cannot find criteria in database -			
No	Cause	System Response										
1	Id does not exist in database	System notices that cannot find criteria in database -										
Relationships: N/A												
Business Rules:	<ul style="list-style-type: none"> - Deactivated criteria cannot be used in creating feedback content. 											

2.3.5.4. Head of Academic Deactivate Account

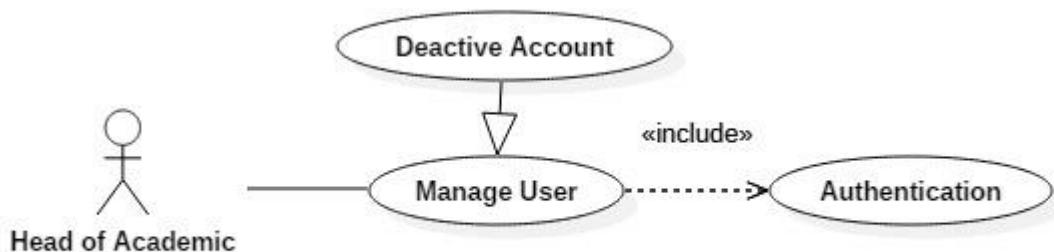


Figure 34 <Head of Academic> Deactivate account

USE CASE - UC027			
Use Case No.	027	Use Case Version	1.0
Use Case Name	Deactivate Account		
Author	VietHQ		
Date	03/04/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - Head of Academic 		

Summary:

- This use case allows Head to deactivate account.

Goal:

- Account is deactivated.

Triggers:

- Head clicks on deactivate account button.

Preconditions:

- Head has been authorized and accessed the system with proper role.

Post Conditions:

- **Success:** Account is deactivated
- **Fail:** Account is not deactivated

Main Success Scenario:

Step	Actor Action	System Response
1	Head sends deactivate account command.	System requires information: - Account Id
2	Head hits confirm button	Account deactivated.

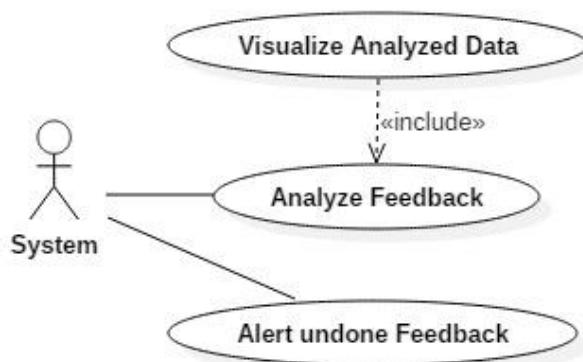
Exceptions :

No	Cause	System Response
1	Id does not exist in database	System notices that cannot find account in database -

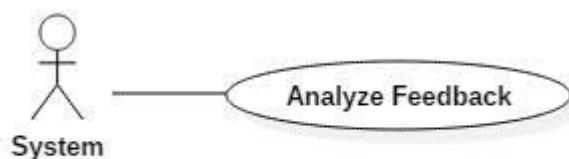
Relationships: N/A**Business Rules:**

- Deactivated account cannot login and access to the system.

2.3.6. System (Actor) Overview Use Case

**Figure 35 <System (Actor)> Overview Use Case**

2.3.6.1. System (Actor) Analyze Feedback

**Figure 36 <System (Actor)> Analyze Feedback**

USE CASE - UC028																		
Use Case No.	028	Use Case Version	1.0															
Use Case Name	Analyze Feedback																	
Author	TuNV, VietHQ																	
Date	03/04/2018	Priority	High															
Actor:	<ul style="list-style-type: none"> - System 																	
Summary:	<ul style="list-style-type: none"> - This use case allows system to analyze from feedbacks' answers. 																	
Goal:	<ul style="list-style-type: none"> - Feedbacks' answer will be sum up and calculated by overall semester or criteria or classes. 																	
Triggers:	<ul style="list-style-type: none"> - Users with see all report or see self's report privileges click the "Xem báo cáo" link. 																	
Preconditions:	<ul style="list-style-type: none"> - Users triggers the event has been authorized and accessed the system with proper role. 																	
Post Conditions:	<ul style="list-style-type: none"> - Success: Return an overall average point and component (class, criteria) average point. - Fail: Return NaN for any kind of point. 																	
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Users click on "Xem báo cáo" menu item</td><td>Redirect users to view list reports pages.</td></tr> <tr> <td>2</td><td>Users click on "Xem báo cáo" on any record show on table.</td><td>Redirect users to view reports throughout semesters.</td></tr> <tr> <td>3</td><td>Actor composes feedbacks' answers then calculate overall average point throughout semesters.</td><td>Return the result to visualize these data.</td></tr> <tr> <td>4.</td><td>Actor selects a semester to view detail point.</td><td>Return the result to visualize these data.</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Users click on "Xem báo cáo" menu item	Redirect users to view list reports pages.	2	Users click on "Xem báo cáo" on any record show on table.	Redirect users to view reports throughout semesters.	3	Actor composes feedbacks' answers then calculate overall average point throughout semesters.	Return the result to visualize these data.	4.	Actor selects a semester to view detail point.	Return the result to visualize these data.
Step	Actor Action	System Response																
1	Users click on "Xem báo cáo" menu item	Redirect users to view list reports pages.																
2	Users click on "Xem báo cáo" on any record show on table.	Redirect users to view reports throughout semesters.																
3	Actor composes feedbacks' answers then calculate overall average point throughout semesters.	Return the result to visualize these data.																
4.	Actor selects a semester to view detail point.	Return the result to visualize these data.																
Alternative Scenario 1:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>3a</td><td>Users filter from which semester to which semester.</td><td>System re-draws to satisfy user desire.</td></tr> <tr> <td>4a</td><td>Actor selects a semester for a subject by a lecturer.</td><td>System provides users with which class should be viewed.</td></tr> </tbody> </table>			Step	Actor Action	System Response	3a	Users filter from which semester to which semester.	System re-draws to satisfy user desire.	4a	Actor selects a semester for a subject by a lecturer.	System provides users with which class should be viewed.						
Step	Actor Action	System Response																
3a	Users filter from which semester to which semester.	System re-draws to satisfy user desire.																
4a	Actor selects a semester for a subject by a lecturer.	System provides users with which class should be viewed.																
Exceptions :	<table border="1"> <thead> <tr> <th>No</th><th>Cause</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Divided by 0</td><td>System removes this calculation from the result.</td></tr> <tr> <td>2</td><td>No feedback for this target has been done.</td><td>System show warning message "Đối tượng này chưa có feedback nào".</td></tr> </tbody> </table>			No	Cause	System Response	1	Divided by 0	System removes this calculation from the result.	2	No feedback for this target has been done.	System show warning message "Đối tượng này chưa có feedback nào".						
No	Cause	System Response																
1	Divided by 0	System removes this calculation from the result.																
2	No feedback for this target has been done.	System show warning message "Đối tượng này chưa có feedback nào".																
Relationships: N/A																		
Business Rules:	<ul style="list-style-type: none"> - Only user with "Head of Academic" and "Admin" can see report for all departments and classes. - User with "Lecturer" role can see only report for classes who he/she is in charge of. - User with "Staff" role can see only report for his/her department. - User with "Student" role cannot access these data. 																	

2.3.6.2. System (Actor) Visualize Data

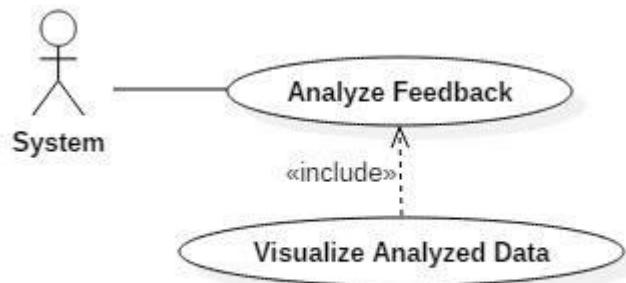


Figure 37 <System (Actor)> Visualize Analyzed Data

USE CASE - UC029																		
Use Case No.	029	Use Case Version	1.0															
Use Case Name	Visualize Analyzed Data																	
Author	TuNV, VietHQ																	
Date	03/04/2018	Priority	High															
Actor:	<ul style="list-style-type: none"> - System 																	
Summary:	<ul style="list-style-type: none"> - This use case allows system to visualize analyzed data. 																	
Goal:	<ul style="list-style-type: none"> - Calculated overall point will be shown as bar charts or pie charts, enable users to see the progress or which criteria need to improved 																	
Triggers:	<ul style="list-style-type: none"> - Users with see all report or see self's report privileges click the "Xem báo cáo" link and data has been calculated. 																	
Preconditions:	<ul style="list-style-type: none"> - User triggers the event has been authorized and accessed the system with proper role. - Overall average points and component points had been calculated. 																	
Post Conditions:	<ul style="list-style-type: none"> - Success: Visualize the calculated data as pie chart or bar chart. - Fail: Cannot show the data as charts. 																	
Main Success Scenario:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1</td><td>Users click on "Xem báo cáo" menu item</td><td>Redirect users to view list reports pages.</td></tr> <tr> <td>2</td><td>Users click on "Xem báo cáo" on any record show on table.</td><td>Redirect users to view reports throughout semesters.</td></tr> <tr> <td>3</td><td>Actor get the calculated data and display as bar chart throughout semesters</td><td>Display bar charts to users' screens.</td></tr> <tr> <td>4.</td><td>Actor selects a semester to view detail point.</td><td>Display pie charts to users' screens</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	Users click on "Xem báo cáo" menu item	Redirect users to view list reports pages.	2	Users click on "Xem báo cáo" on any record show on table.	Redirect users to view reports throughout semesters.	3	Actor get the calculated data and display as bar chart throughout semesters	Display bar charts to users' screens.	4.	Actor selects a semester to view detail point.	Display pie charts to users' screens
Step	Actor Action	System Response																
1	Users click on "Xem báo cáo" menu item	Redirect users to view list reports pages.																
2	Users click on "Xem báo cáo" on any record show on table.	Redirect users to view reports throughout semesters.																
3	Actor get the calculated data and display as bar chart throughout semesters	Display bar charts to users' screens.																
4.	Actor selects a semester to view detail point.	Display pie charts to users' screens																
Alternative Scenario 1:	<table border="1"> <thead> <tr> <th>Step</th><th>Actor Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>3a</td><td>Users filter from which semester to which semester.</td><td>System re-draws charts to satisfy user desire.</td></tr> </tbody> </table>			Step	Actor Action	System Response	3a	Users filter from which semester to which semester.	System re-draws charts to satisfy user desire.									
Step	Actor Action	System Response																
3a	Users filter from which semester to which semester.	System re-draws charts to satisfy user desire.																

4a	Actor selects a semester for a subject by a lecturer.	System provides users with which class should be viewed.
----	---	--

Exceptions :

No	Cause	System Response
1	Nan point	Corresponding chart will be blank.
2	No feedback for this target has been done.	System shows warning message “Đối tượng này chưa có feedback nào”.

Relationships: N/A

Business Rules:

- Only user with “Head of Academic” and “Admin” can see report for all departments, classes.
- User with “Lecturer” role can see only report for classes who he/she is in charge of.
- User with “Staff” role can see only report for his/her department.
- User with “Student” role cannot access these data.

2.3.6.3. System (Actor) Alert undone Feedback



Figure 38 <System (Actor)> Alert undone Feedback

USE CASE - UC030			
Use Case No.	030	Use Case Version	1.0
Use Case Name	Alert undone Feedback		
Author	VietHQ		
Date	04/04/2018	Priority	High
Actor:	<ul style="list-style-type: none"> - System 		
Summary:	<ul style="list-style-type: none"> - This use case allows system to alert undone Feedback to users. 		
Goal:	<ul style="list-style-type: none"> - Alert undone Feedback to corresponding users. 		
Triggers:	<ul style="list-style-type: none"> - Feedback is published. 		
Preconditions:	<ul style="list-style-type: none"> - Feedback is still in available time and has at least one conductor. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Alert undone Feedback to users. - Fail: Feedback is not alerted. 		
Main Success Scenario:			
Step	Actor Action	System Response	
1	Users click on Publish Feedback button.	System checks list conductors and send email, alert them to do feedback. [Exception 1]	
Exceptions :			
No	Cause	System Response	
1	There is no conductors.	System shows error message “Xin chọn ít nhất một người thực hiện”	

Relationships: N/A

Business Rules:

- Alert icon will be keep displaying until Feedback is conducted.
- Once user conducts Feedback, user can conduct again as long as Feedback is still available.

3. Software System Attribute

3.1. Usability

- Head of Academic and staff should need less than 0.5 day of training to use the system.
- Mobile and web application use Vietnamese including all dialogs and messages.
- Student and Lecturer can understand and perform functions immediately without the training process.

3.2. Availability

- System is divided into modules, if a function is down, it will not influence others.
- The system can be adapted for a large number of requests.

3.3. Security

- All input data should be validated before saving to database.
- Roles permission should be specified clearly and user should be authenticated and authorized when accessing to the system.

3.4. Maintainability

- System is divided into modules.
- When a module of a function is down, it is easy to take it down to fix without impact other functions.

3.5. Portability

- Web application can run on Chrome, Firefox, IE browser.
- User can use the mobile application on devices running Android 6 or later.

3.6. Performance

- System converts and returns results in 1 seconds or less depend on information of inputted driver.

4. Conceptual Diagram

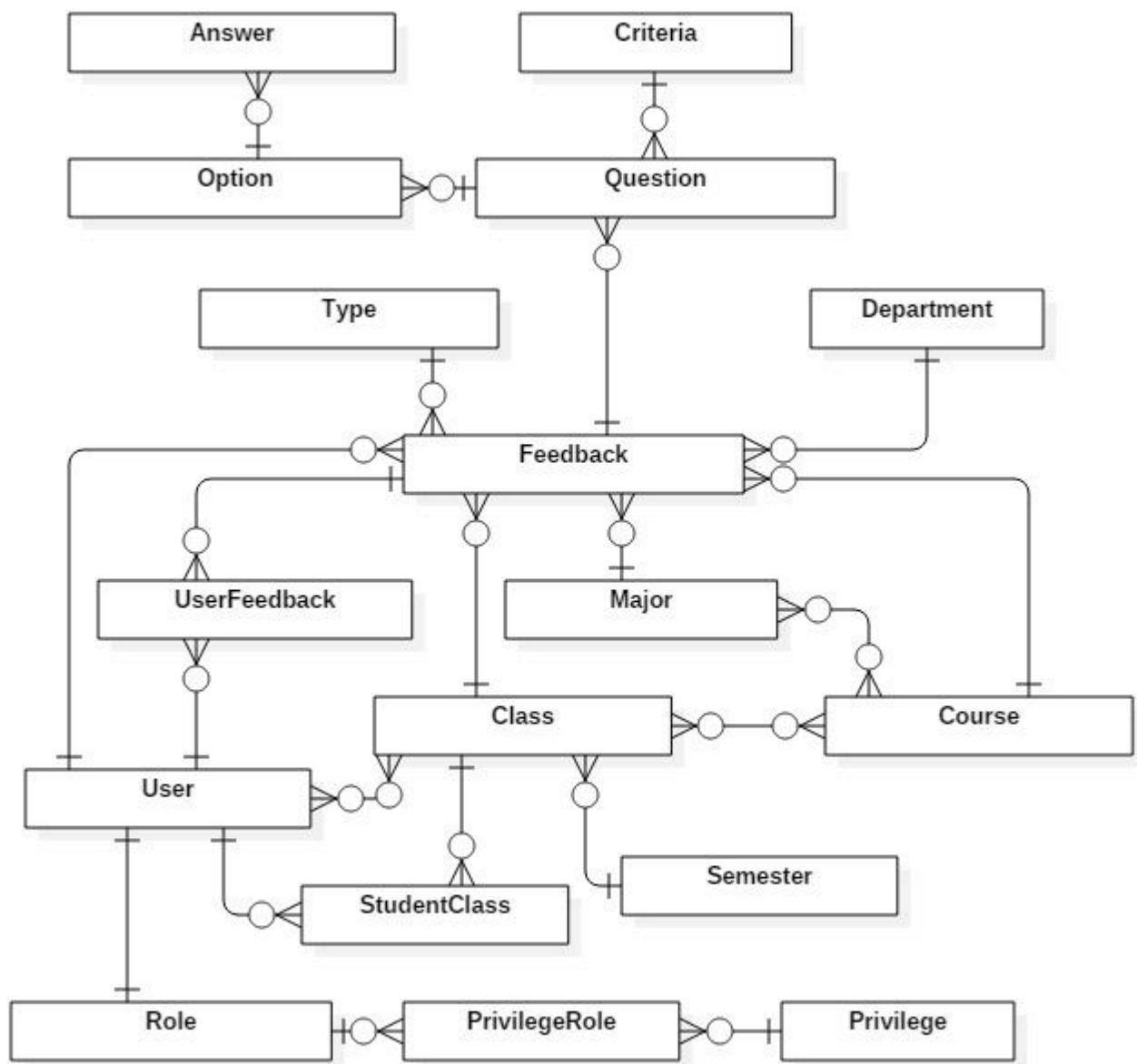


Figure 39 Conceptual diagram

Data Dictionary

Entity Data dictionary: describe content of all entities	
Entity Name	Description
Feedback	Contain Feedback information
Question	Contain Question information
Option	Contain Option information
Answer	Contain Answer information
Criteria	Contain Criteria information
Semester	Contain Semester information
Type	Contain Feedback type information
User	Contain User information
UserFeedback	Contain information about relationship between a user with a feedback
Role	Contain Role information
Privilege	Contain Privilege information
PrivilegeRole	Contain Privilege information for each role
StudentClass	Contain information about class of each student
Class	Contain Class information
Course	Contain Course information
Major	Contain Major information
Department	Contain Department information

Table 7 Conceptual diagram data dictionary

D. Software Design Description

1. Design Overview

This document describes the technical and user interface design of SFMS System. It includes the architectural design, the detailed design of common functions and business functions and the design of database model. Document overview:

- Section 2: gives an overall description of the system architecture design.
- Section 3: gives component diagrams that describes the connection and integration of the system.
- Section 4: gives the detail design description which includes class diagram, class explanation and sequence diagram to details the application functions.
- Section 5: describes screen design.
- Section 6: describes a fully attributed ERD.
- Section 7: describes algorithms.

2. System Architectural Design

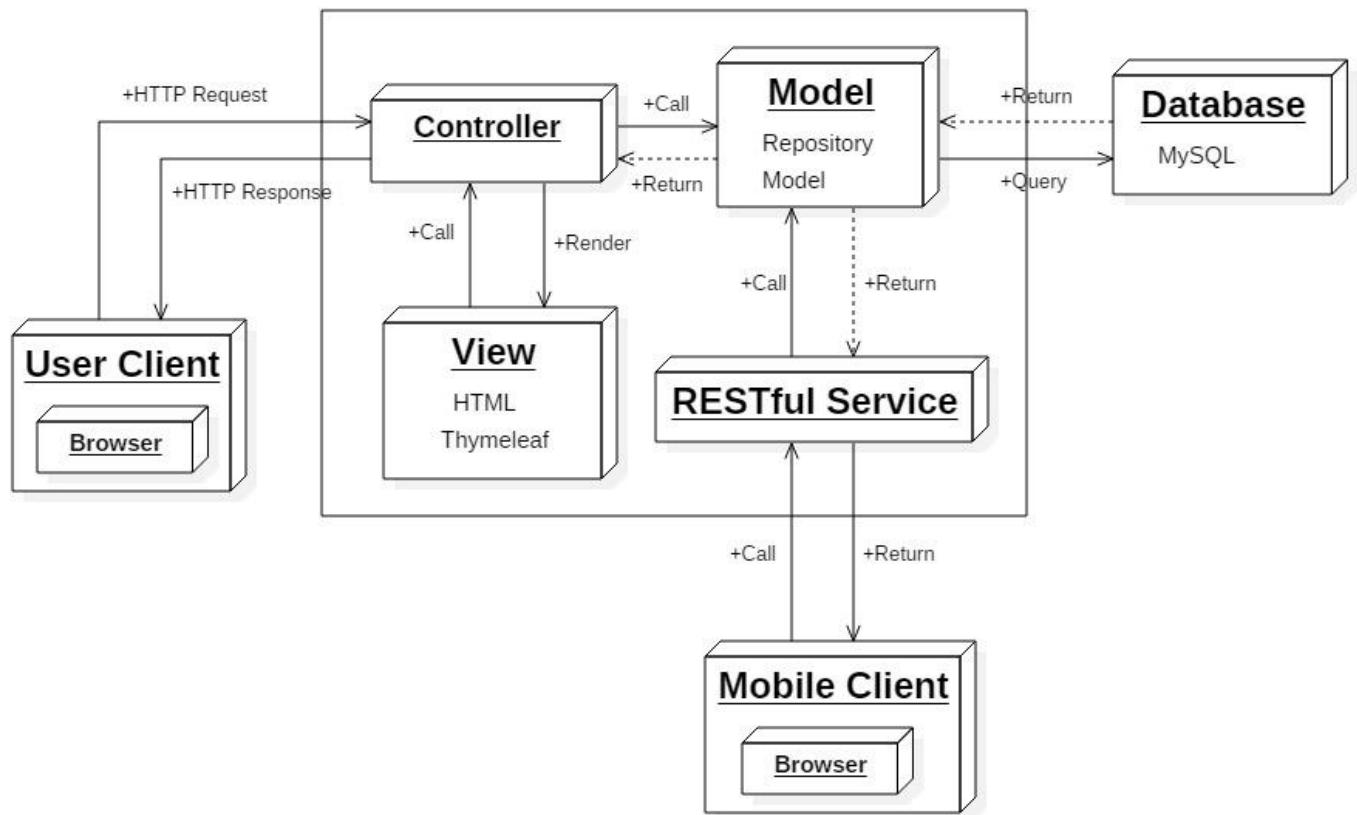


Figure 40 System Architecture Design

In Web Application, the system is developed under MVC architecture style. We choose this architecture for Web Application because of following advantages:

- The application will be divided into three clear parts.
- Testing each component in the architecture become easier.
- Ease of maintenance, we can change any component without effect to other component.

- Faster development process: MVC supports rapid and parallel development. With MVC, one programmer can work on the view while other can work on the controller to create business logic of the web application. The application developed using MVC can be three times faster than application developed using other development patterns.

This project follows MVC architecture with following components:

- Controller is the parts of the application that acts like event handler to handles user interaction. Typically, controller read data from a request and calls appropriate Business's method then selects view to return to user.
- View is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
- The model is responsible for managing the data of the application. It responds to the request from the view and it also responds to instructions from the controller to update itself.

3. Component Diagram

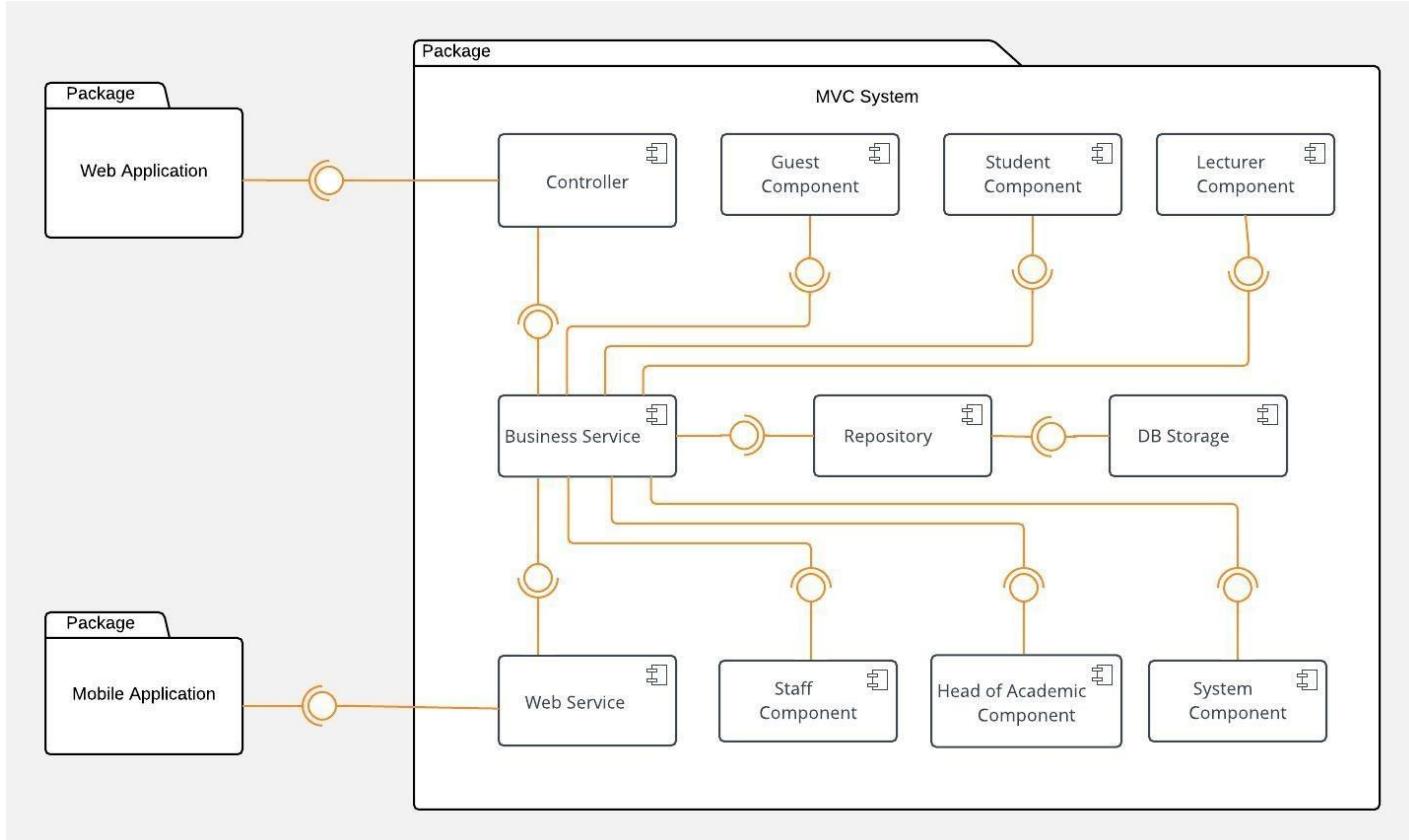


Figure 41 Component Diagram

Component Dictionary: Describes components	
Web Application	Web application package contains operation of SFMS on web
Mobile Application	Mobile application package contains operation of SFMS on mobile device
Business Service	Handle Business logic
Controller	Handle Request and Response. Accepts input and converts it to commands for the model and view.

Web Service	Provide API for mobile applications to interact with the system
Guest Component	Component to handle customer activities in the system
Student Component	Component to handle student activities in the system
Lecturer Component	Component to handle lecturer activities in the system
Staff Component	Component to handle staff activities in the system
Head of Academic Component	Component to handle head of academic activities in the system
System Component	Component to handle system
Repository	Store data
DB Storage	Database on hard disk. Contain method for tranferring data between database and data object

Table 8 Component Dictionary

4. Detailed Description

4.1. Class Diagram

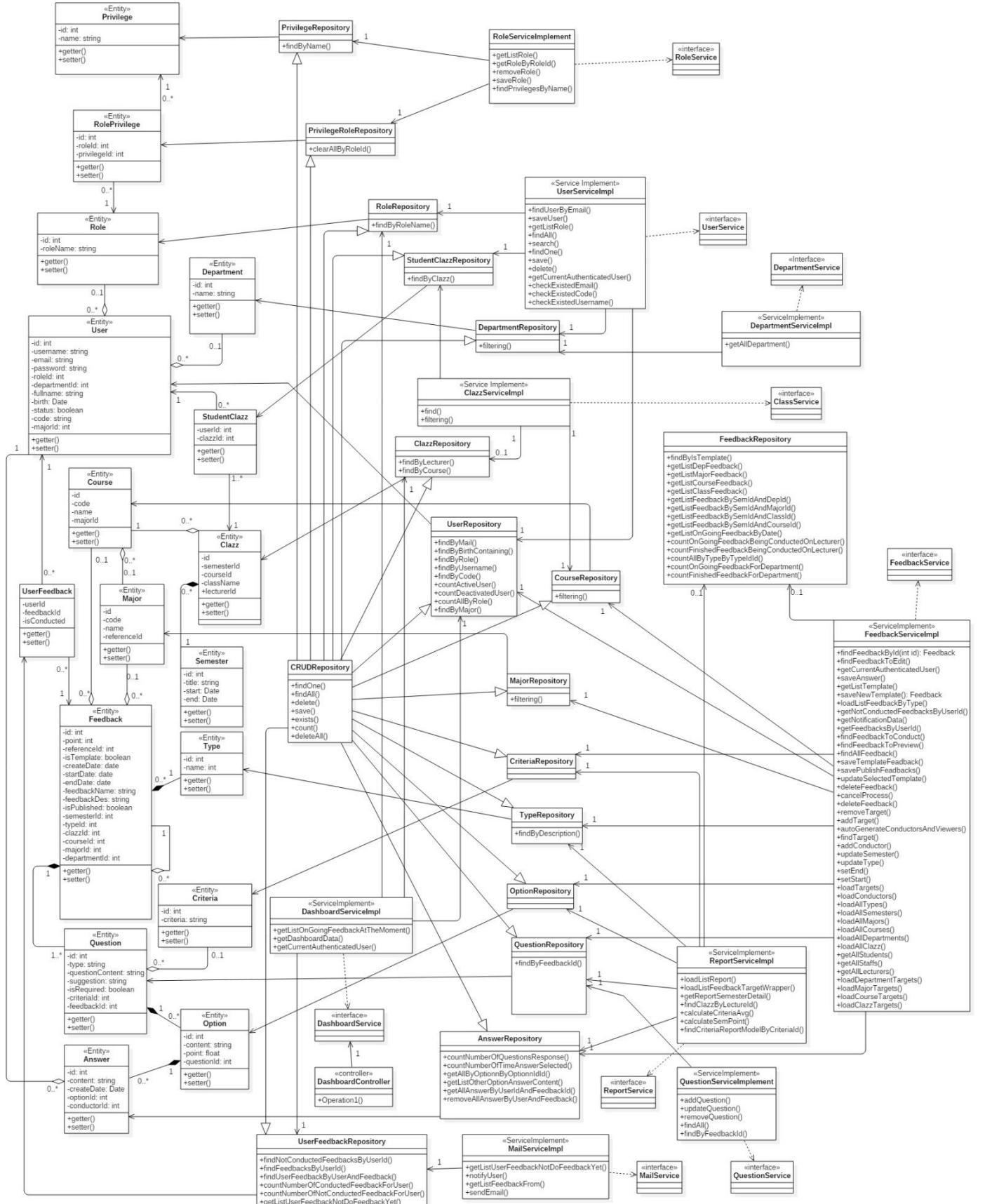


Figure 42 Class Diagram

Class dictionary: describe Class	
Class Name	Description
User	Contain the user's account & profile information
Privilege	Contain the privilege information
Role	Contain the role information
RolePrivilege	Contain the information about the relationship between role & privilege
Department	Contain the department information
Major	Contain the major information
Course	Contain the course information
Clazz	Contain the class information
StudentClazz	Contain the information the relationship between user whom role is Student and Clazz which he's enrolled
Type	Contain the feedback's type information
Semester	Contain the semester information
Feedback	Contain the feedback information
UserFeedback	Contain the information the relationship between user (whom role is Student) and Clazz which he's enrolled
Criteria	Contain the criteria information
Question	Contain the question information
Option	Contain the option information
Answer	Contain the answer information

Table 9 Class Diagram Explanation

4.1.1. User

Attribute	Type	Visibility	Description
id	int	private	unique identifier of a user's account
username	string	private	unique username of a user's account
email	string	private	unique email of a user's account
password	string	private	password of a user's account
roleId	int	private	identifier of the role of account
departmentId	int	private	Identifier of the department of account
fullname	string	private	fullname of user
code	string	private	code of user's account
birth	date	private	date of birth of user
status	boolean	private	status of user's account
majorId	int	private	identifier of the department of user

4.1.2. Priviledge

Attribute	Type	Visibility	Description
id	int	private	Unique identifier of a privilege
name	string	private	name of privilege

4.1.3. Role

Attribute	Type	Visibility	Description
id	int	private	Unique identifier of a role
roleName	string	private	name of role

4.1.4. RolePriviledge

Attribute	Type	Visibility	Description
id	int	private	Unique identifier of a role
roleId	int	private	Identifier of RolePrivilege's role
privilegeId	int	private	Identifier of RolePrivilege's privilege

4.1.5. Department

Attribute	Type	Visibility	Description
id	int	private	Unique identifier of a department
name	int	private	name of a department

4.1.6. Major

Attribute	Type	Visibility	Description
id	int	private	Unique identifier of a major
code	string	private	code of a major
name	string	private	name of a major
referenceId	int	private	Identifier of major's reference

4.1.7. Course

Attribute	Type	Visibility	Description
id	int	private	Unique identifier of a

			course
code	string	private	code of a course
name	string	private	name of a course
majorId	int	private	Identifier of course's major

4.1.8. Clazz

Attribute	Type	Visibility	Description
id	int	private	Unique identifier of a course
courseld	int	private	Identifier of class's course
className	string	private	name of a course
lecturerId	int	private	Identifier of class's lecturer
semesterId	int	private	Identifier of class's semester

4.1.9. StudentClazz

Attribute	Type	Visibility	Description
userid	int	private	identifier of student
clazzId	int	private	Identifier of class

4.1.10. Semester

Attribute	Type	Visibility	Description
id	int	private	Unique identifier of a semester
title	string	private	title of a semester
start	date	private	a day when semester start
end	date	private	a day when semester will end

4.1.11. Type

Attribute	Type	Visibility	Description
id	int	private	Unique identifier of a type
name	string	private	name of type

4.1.12. Feedback

Attribute	Type	Visibility	Description

<code>id</code>	<code>int</code>	<code>private</code>	unique identifier of a feedback
<code>point</code>	<code>string</code>	<code>private</code>	point of e feedback
<code>referenceId</code>	<code>int</code>	<code>private</code>	identifier of feedback's reference
<code>isTemplate</code>	<code>boolean</code>	<code>private</code>	the feedback is template or not
<code>isPublished</code>	<code>boolean</code>	<code>private</code>	the feedback is published or not
<code>createDate</code>	<code>date</code>	<code>private</code>	create date of feedback
<code>startDate</code>	<code>date</code>	<code>private</code>	when the feedback start to receive answers
<code>endDate</code>	<code>date</code>	<code>private</code>	when the feedback end to receive answers
<code>feedbackName</code>	<code>string</code>	<code>private</code>	name of feedback
<code>feedbackDes</code>	<code>string</code>	<code>private</code>	description of feedback
<code>semesterId</code>	<code>int</code>	<code>private</code>	identifier of feedback's semester
<code>typeId</code>	<code>int</code>	<code>private</code>	identifier of feedback's type
<code>majorId</code>	<code>int</code>	<code>private</code>	identifier of feedback's major
<code>courseId</code>	<code>int</code>	<code>private</code>	identifier of feedback's course
<code>clazzId</code>	<code>int</code>	<code>private</code>	identifier of feedback's class
<code>departmentId</code>	<code>int</code>	<code>private</code>	identifier of feedback's department

4.1.13. *UserFeedback*

Attribute	Type	Visibility	Description
<code>userId</code>	<code>int</code>	<code>private</code>	Identifier of conductor
<code>feedbackId</code>	<code>string</code>	<code>private</code>	Identifier of feedback
<code>isConducted</code>	<code>boolean</code>	<code>private</code>	the feedback has been conducted by user or not

4.1.14. *Criteria*

Attribute	Type	Visibility	Description
<code>id</code>	<code>int</code>	<code>private</code>	Unique identifier of a criteria
<code>criteria</code>	<code>string</code>	<code>private</code>	name of criteria
<code>status</code>	<code>boolean</code>	<code>private</code>	the criteria is activated to be use or not

4.1.15. *Question*

Attribute	Type	Visibility	Description
<code>id</code>	<code>int</code>	<code>private</code>	unique identifier of a question
<code>type</code>	<code>string</code>	<code>private</code>	type of question, depend on the way we answer it
<code>questionContent</code>	<code>boolean</code>	<code>private</code>	content of question
<code>suggestion</code>	<code>string</code>	<code>private</code>	suggestion for improving question's result
<code>isRequired</code>	<code>boolean</code>	<code>private</code>	question must be answered or not
<code>criteriaId</code>	<code>int</code>	<code>private</code>	identify of question's criteria
<code>feedbackId</code>	<code>int</code>	<code>private</code>	identify of question's feedback

4.1.16. *Option*

Attribute	Type	Visibility	Description
id	int	private	unique identifier of an option
isRequired	boolean	private	question must be answered or not
point	int	private	point of option
questionId	int	private	identify of option's question

4.1.17. *Answer*

Attribute	Type	Visibility	Description
id	int	private	unique identifier of an answer
answerContent	int	private	content of answer as text
createDate	date	private	when the answer was created
optionId	int	private	identify of answer's option
conductorId	int	private	identify of answer's conductor

4.2. *Interaction Diagram*

4.2.1. *Create new Feedback*

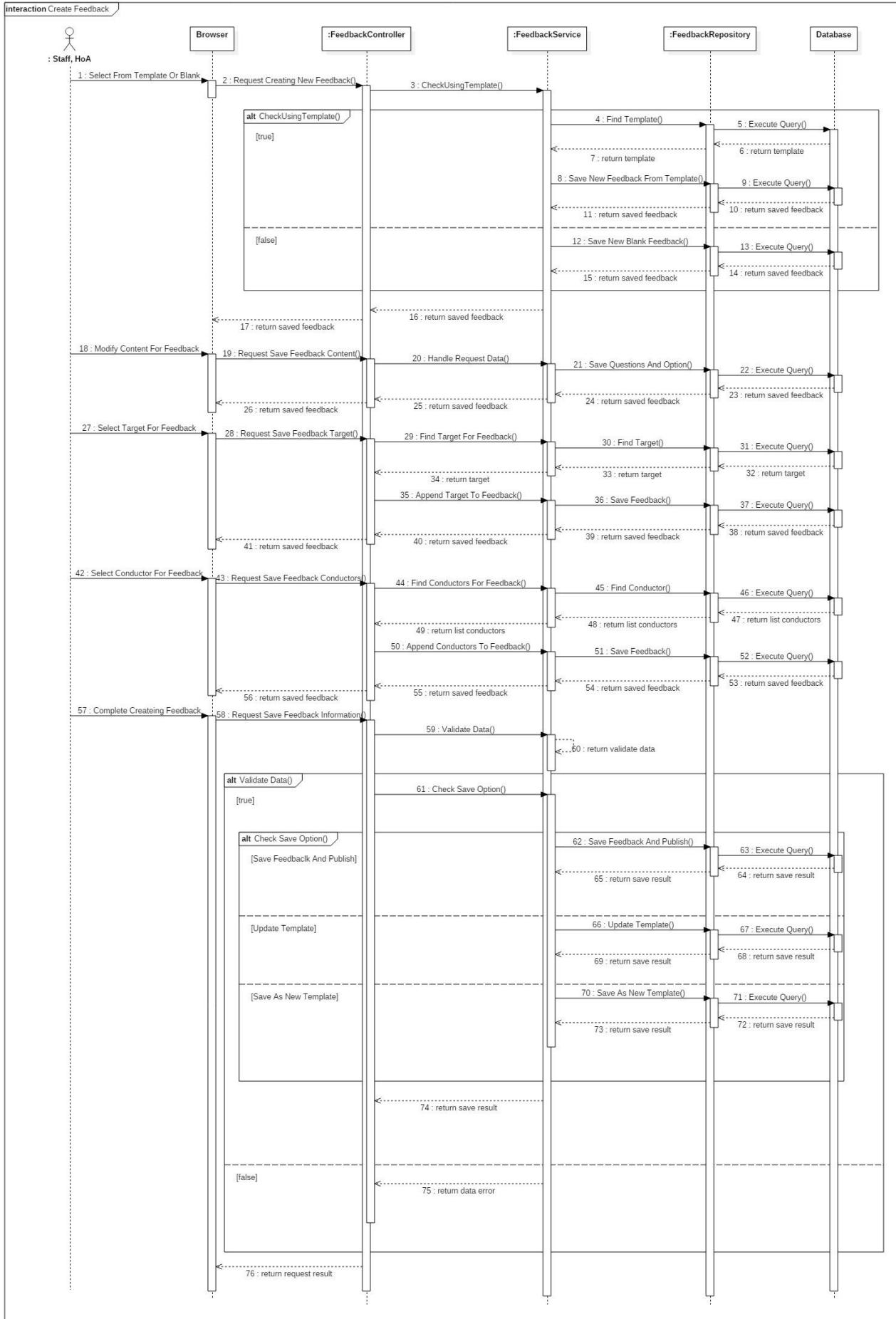


Figure 43 Sequence diagram - Create new Feedback

4.2.2. Conduct Feedback

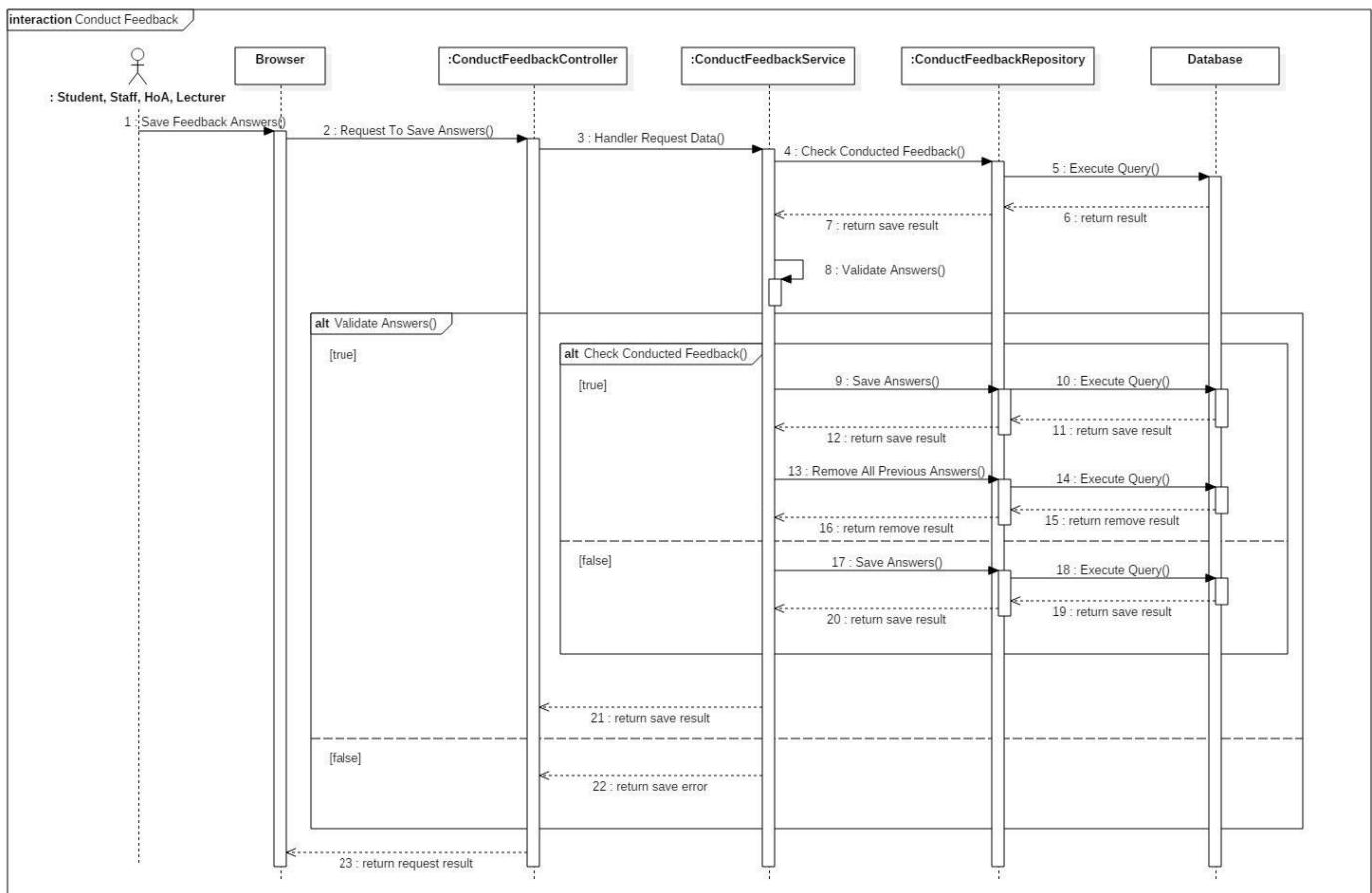


Figure 44 Sequence diagram - Conduct Feedback

4.2.3. View Feedback Report

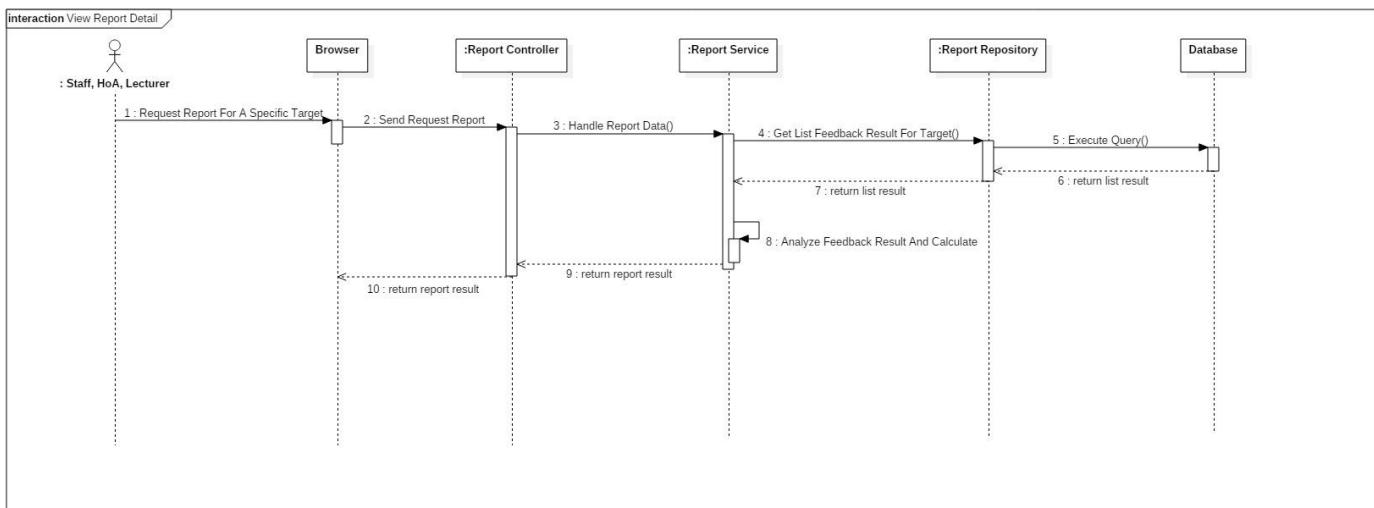


Figure 45 Sequence diagram - View Feedback Report

4.2.4. Alert Undone Feedback

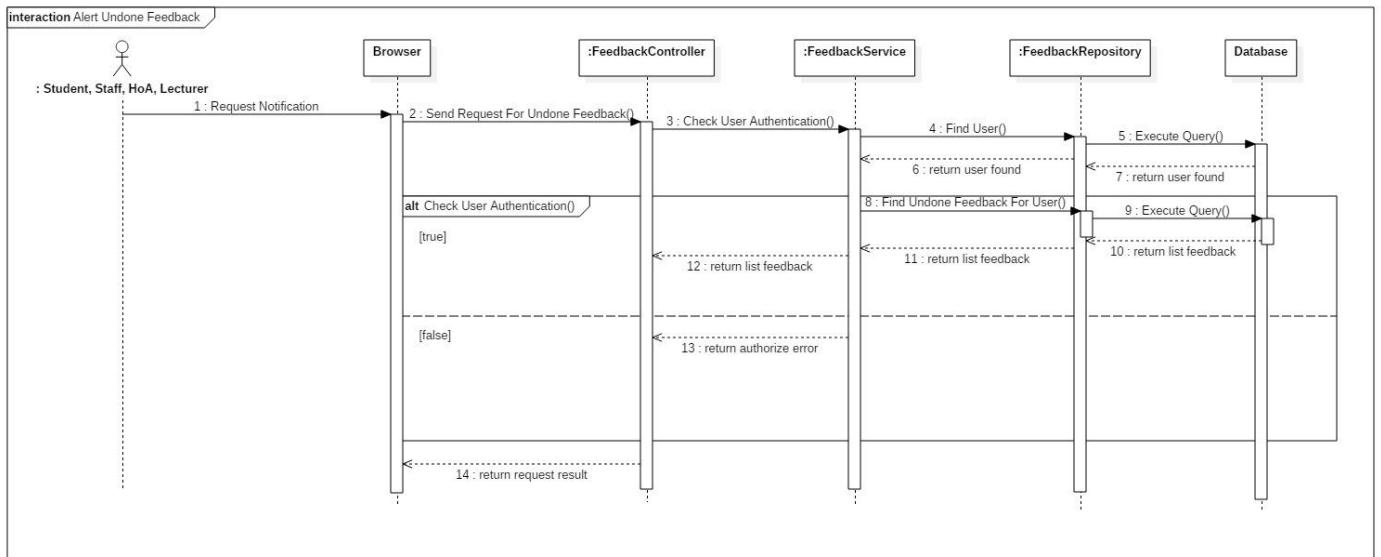


Figure 46 Sequence diagram - Alert Undone Feedback

4.2.5. Notify Undone Feedback by Email

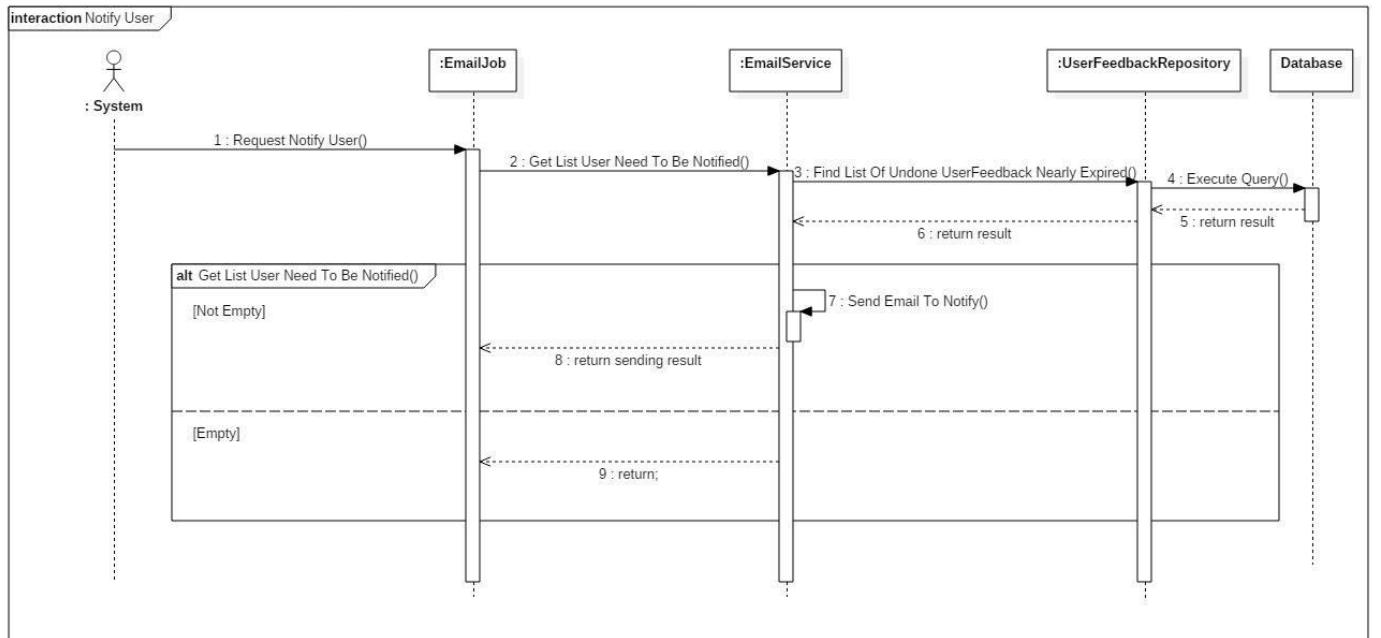


Figure 47 Sequence diagram - Notify Undone Feedback by Email

4.2.6. Create User

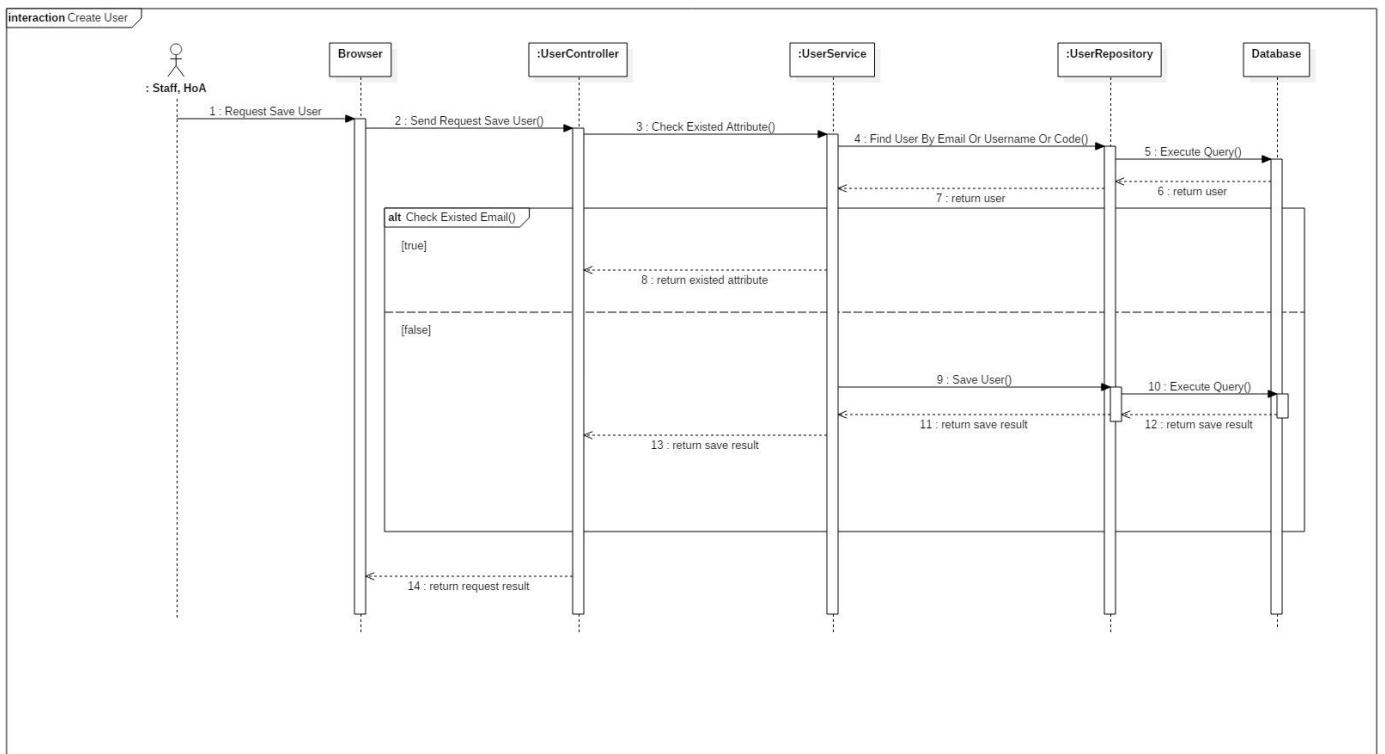


Figure 48 Sequence diagram – Create User

4.2.7. Create criteria

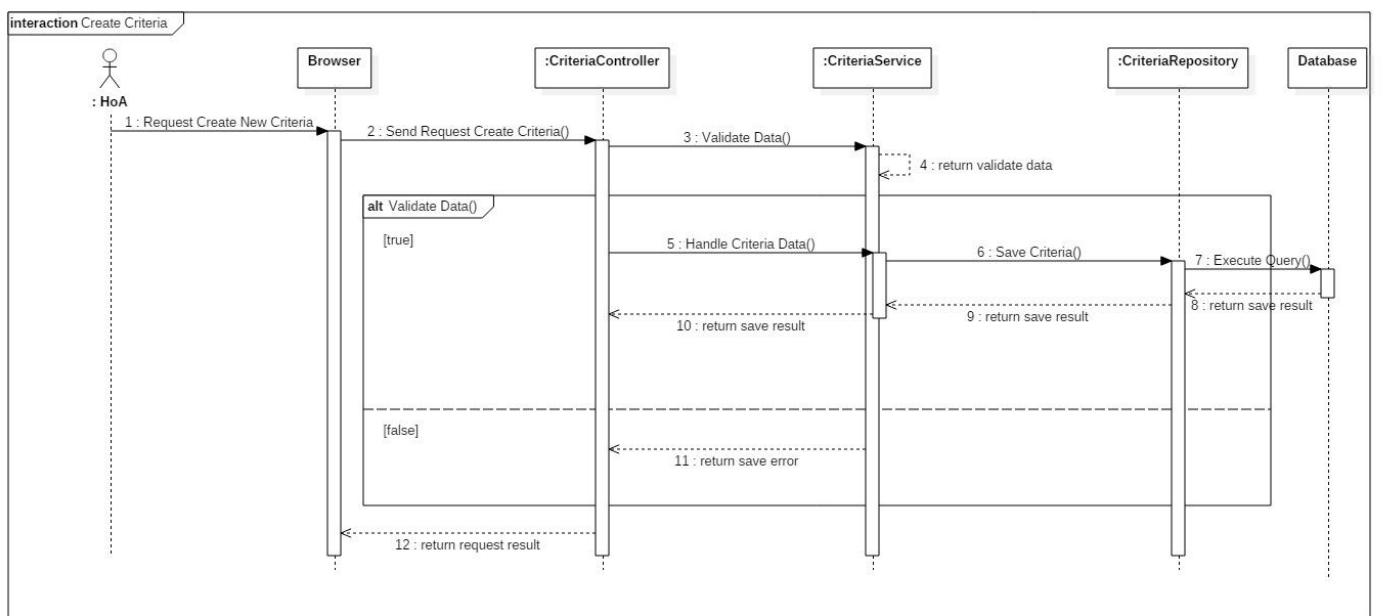


Figure 49 Sequence diagram – Create Criteria

5. Interface

5.1. Component Interface

5.1.1. Module of User Api

Signature	Description	Output	Exception
mobileLogin(String username, String password)	Login into Mobile Application	boolean	N/A
findAll	Find all users	JSON of List Users	N/A
getUser(int id)	Get user information	JSON of User	N/A
saveUser(SaveUserModel userModel)	Save user	ResponseEntity	N/A
updateUser(SaveUserModel userModel)	Update user	ResponseEntity	N/A
deleteUser(int id)	Delete user	N/A	N/A
checkExistedEmail(String email)	Check if email is existed when create new user	ResponseEntity	N/A
checkExistedCode(String code)	Check if code is existed when create new user	ResponseEntity	N/A
checkExistedUsername(String username)	Check if username is existed when create new user	ResponseEntity	N/A

5.1.2. Module of Feedback Api

Signature	Description	Output	Exception
getFeedback(int id)	Get content of Feedback	ResponseEntity	N/A
listClazzesTargets(HttpSession session)	Get list target for Feedback type Clazz	ResponseEntity	N/A
listDepartmentTargets(HttpSession session)	Get list target for Feedback type Department	ResponseEntity	N/A
listMajorTargets(HttpSession session)	Get list target for Feedback type Major	ResponseEntity	N/A
listCourseTargets(HttpSession session)	Get list target for Feedback type Course	ResponseEntity	N/A
createFeedbackFromTemplate (int id, HttpSession session)	Create Feedback from a Template	ResponseEntity	N/A
editTitle(Feedback feedback, HttpSession session)	Edit Title when creating Feedback	ResponseEntity	N/A
editDescription(Feedback feedback, HttpSession session)	Edit Description when creating Feedback	ResponseEntity	N/A
editStart(Feedback feedback, HttpSession session)	Edit Start date of Feedback when creating Feedback	ResponseEntity	N/A
editEnd(Feedback feedback, HttpSession session)	Edit End date of Feedback when creating Feedback	ResponseEntity	N/A
editType(Feedback feedback, HttpSession session)	Edit Type of Feedback when creating Feedback	ResponseEntity	N/A
editSemester(Feedback feedback, HttpSession session)	Edit Semester of Feedback when creating Feedback	ResponseEntity	N/A

addTarget(Target t, HttpSession session)	Add target of creating Feedback	ResponseEntity	N/A
removeTarget(Target t, HttpSession session)	Remove target when creating Feedback	ResponseEntity	N/A
addConductor(int targetId, User conductor, HttpSession session)	Add conductor for a target when creating Feedback	ResponseEntity	N/A
removeConductor(int id, User conductor, HttpSession session)	Remove conductor when creating Feedback	ResponseEntity	N/A
addQuestion(AddQuestionModel model)	Add created Question	Response	N/A
addMultipleQuestions(AddQuestionModel[] model)	Add multiple created Questions at the same time	Response	N/A
updateQuestion(UpdateQuestionModel model)	Update created Question	Response	N/A
removeQuestion(RemoveQuestionModel model)	Remove created Question	Response	N/A
saveFeedback(int opt, HttpSession session)	Save Feedback and complete creating Feedback process	ResponseEntity	N/A
getListTemplate(boolean isTemplate)	Get list templates	JSON of List Templates	N/A
getTemplateContent(int id)	Get content of template	JSON of Feedback	N/A
saveQuestion(FeedbackCreateModel model)	Save questions	Response	N/A
deactiveTemplate(String templateIDS)	Deactive template	Response	N/A
modifyQuestion(FeedbackUpdateModel model)	Modify questions	Response	N/A
modifySuggestion(FeedbackModifySuggestionModel model)	Modify suggestions	Response	N/A
getListNotConductedFeedbackByAuthorizedUser()	Get list of undone Feedback	ResponseEntity	N/A
conductFeedback(int id)	Load content of Feedback by Id to conduct	JSON of Feedback	N/A
conductFeedback(int id) saveConductedFeedback(ConductAnswerWrapper conductAnswerWrapper)	Save answers after conducting Feedback	ResponseEntity	N/A

5.1.3. Module of Report Api

Signature	Description	Output	Exception
getListReport(String type)	Get list of Reports	JSON of List Reports	N/A
getListSemester()	Get list semester for Bar chart Report	ResponseEntity	N/A
getSemesterStatistic(int courseId, List<Integer> semesterIds)	Get statistic results for Bar chart Reports	ResponseEntity	N/A

5.1.4. Module of Criteria Api

Signature	Description	Output	Exception
getAll()	Get list of Criterias	ResponseEntity	N/A
create(Criteria c)	Create new Criteria	ResponseEntity	N/A
update(Criteria c)	Update new Criteria	ResponseEntity	N/A
active(Criteria c)	Active new Criteria	ResponseEntity	N/A
deactive(Criteria c)	Deactive new Criteria	ResponseEntity	N/A

5.2. User Interface Design

5.2.1. Login

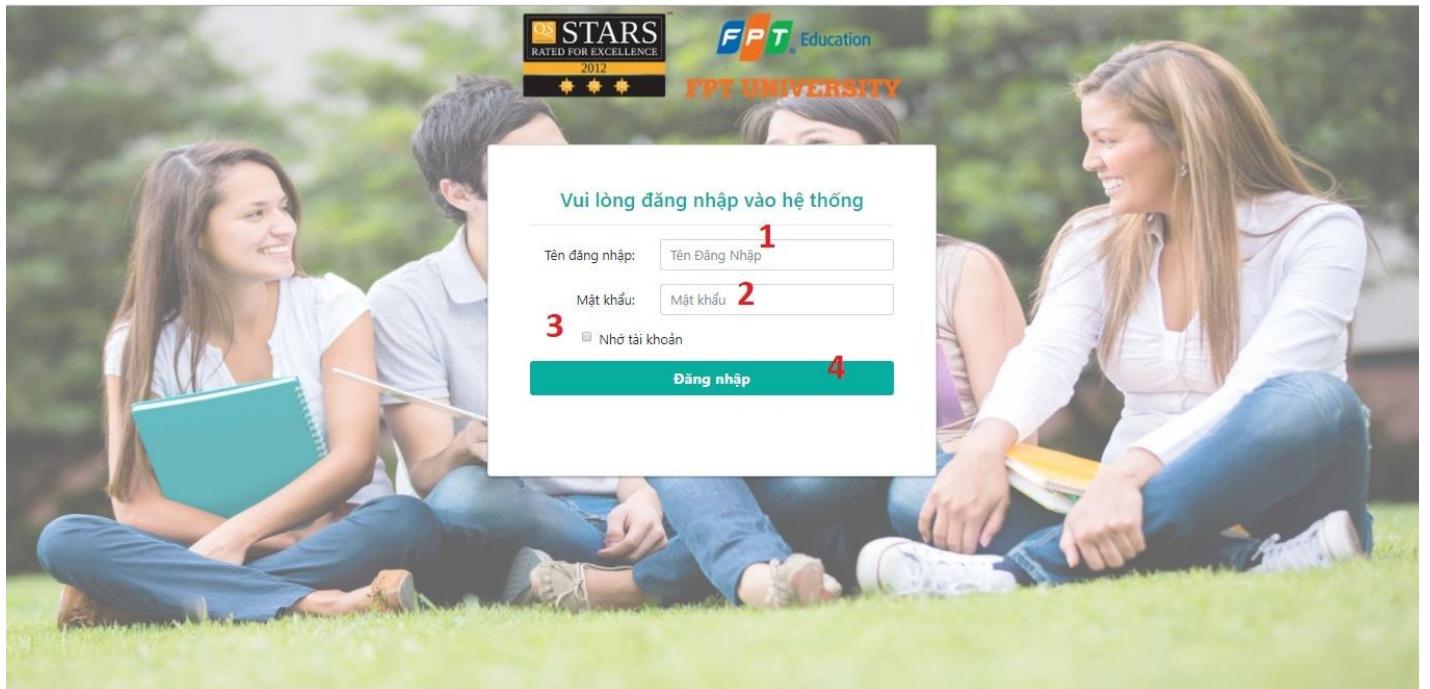


Figure 50 <Web Application> Login

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	Username	Fill username	No	Yes	Textbox	String
2	Password	Fill password	No	Yes	Password	String
3	rememberAccount	Remember account	No	No	RadioButton	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
4	Login	Log-in into the system	N/A	Transfer to Dashboard

5.2.2. Side Menu

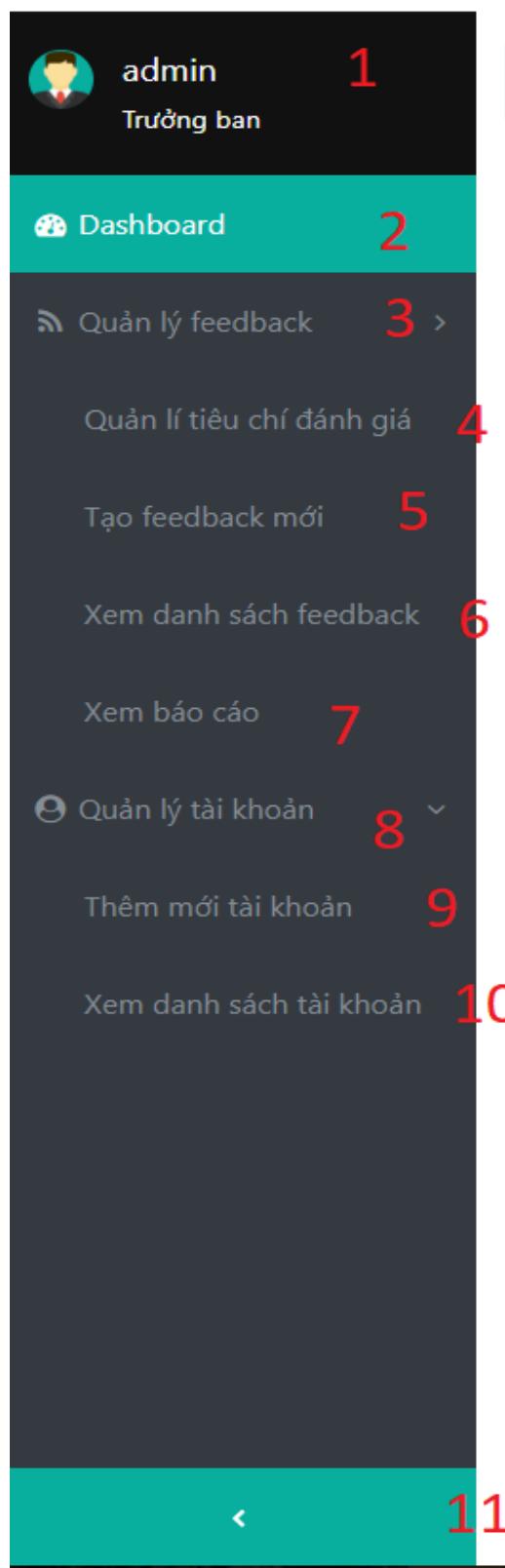


Figure 51 <Web Application> Side menu

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	accountInformation	Display user's full name and role.	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	Dashboard	Redirect to dashboard.	User must had been authorized.	Redirect to dashboard.
3	manageFeedback	Menu contains submenu with corresponding function.	N/A	Collapse sub-menus inside.
4	manageCriteria	Sub-menu contains hyperlink to manage criteria	User must have one of these roles: Head of Academic	Redirect to manage criteria.
5	createNewCriteria	Sub-menu contains hyperlink to create new feedback.	User must have one of these roles: -Head of Academic. -Staff.	Redirect to create new feedback page.
6	seeListFeedback	Sub-menu contains hyperlink to list conduct feedback page.	User must have been authorized.	Redirect to see list feedback page.
7	seeReports	Sub-menu contains hyperlink to see reports.	User must have one of these roles: -Head of Academic. -Staff. -Lecturer.	Redirect to see reports page.
8	mngUser	Menu contains sub-menu with corresponding function.	User must have one of these roles: -Head of Academic. -Staff.	Collapse sub-menus inside it.
9	createUser	Sub-menu contains hyperlink to create new user page.	User must have one of these roles: -Head of Academic. -Staff.	Redirect to create new user page.
10	seeUser	Sub-menu contains hyperlink to see users page.	User must have one of these roles: -Head of Academic. -Staff.	Redirect to see users.
11	toggleSideMenu	Button to toggle show/hide state of side menu.	N/A	Show/hide side menu.

5.2.3. Dashboard



Figure 52 <Web Application> Login

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	statisticChart	Pie Chart statistics number of users, feedbacks	Yes	No	N/A	Chart
2	pageFilter	Change page of table Feedback list	No	No	Dropdown List	String
3	search	Search Feedback	No	No	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
4	conductFeedb ack	Conduct the Feedback	N/A	Transfer to conduct Feedback Page

5.2.4. Alert Undone Feedback



Figure 53 <Web Application> Alert Undone Feedback

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
2	fullname	Fullname of user	Yes	Yes	Text	String
4	feedbackDes	Description of undone Feedback	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	iconAlert	Icon alerts User that they have Feedback to do	N/A	Dropdown the list of Undone Feedback
3	feedbackName	Name of undone Feedback	N/A	Transfer to conduct Feedback page

5.2.5. Manage Criteria

Tiêu chí	Trạng thái	Tùy chọn
ahihiid	7+8	7+8
Bảo đảm giáo trình	4	+
Chuyên cần	5	+
Cơ sở hạ tầng kỹ thuật	+	+
Cơ sở vật chất	+	+
Giáo trình	+	+
Giải đáp thắc mắc	+	+

Figure 54 <Web Application> Manage Criteria

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	pageFilter	Change page of table Feedback list	No	No	Dropdown List	String
2	search	Search Criteria or Type	No	No	Text	String
3	critName	Criteria Name	Yes	Yes	Text	String
5	actToggle	Active a criteria	No	Yes	Toggle	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
7	addCrit	Add new Criteria	N/A	Open popup window to create new Criteria
8	editCrit	Edit new Criteria	N/A	Open popup window to edit Criteria

5.2.6. Add Criteria

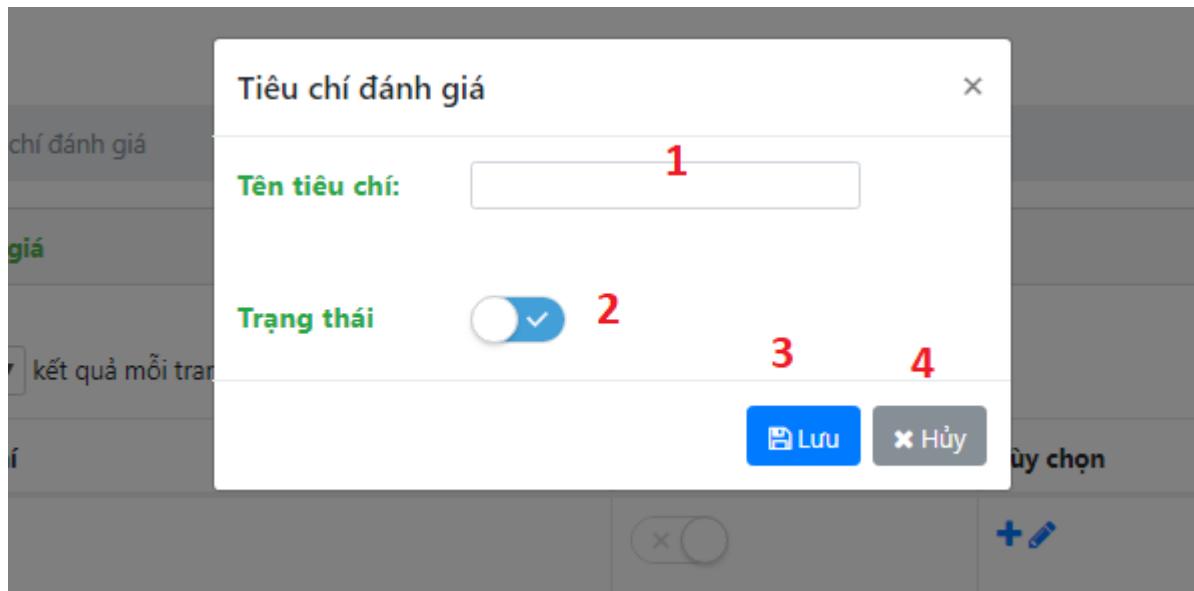


Figure 55 <Web Application> Add Criteria

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	critName	Criteria Name	No	Yes	Text	String
2	actToggle	Active a criteria	No	Yes	Toggle	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	save	Save new Criteria	N/A	Add new Criteria
4	cancel	Cancel	N/A	Cancel Creating

5.2.7. Select template

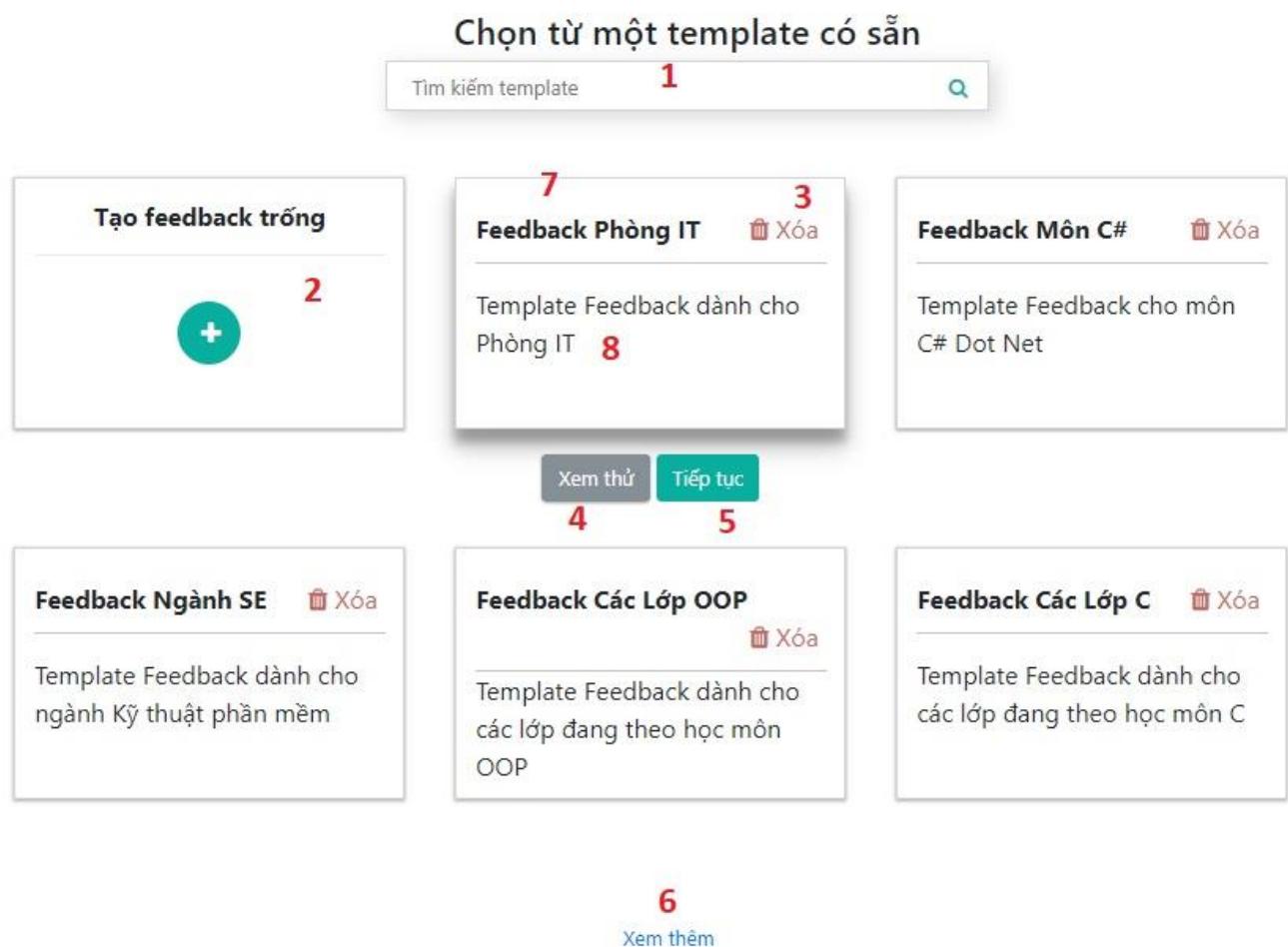


Figure 56 <Web Application> Select Template

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
7	tempName	Template Name	No	Yes	Text	String
8	tempDes	Template Description	No	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	search	Search Template	N/A	Template match with search keyword
2	createFb	Create blank new Feedback	N/A	Open popup window to create blank new Feedback
3	removeTemp	Remove Template	N/A	Remove a Template
4	previewTemp	Preview content of a Template	N/A	Open popup window to show content of a Template
5	nextTemp	Choose Template	N/A	Transfer to create content Feedback page
3	loadMore	Load more Template	N/A	Load more Templates

5.2.8. Preview Template

Feedback Các Lớp C#

Câu hỏi 1 * **1**
Tốc độ giảng dạy của giáo viên thế nào? **2**

- Nhanh **3**
- Vừa
- Chậm
- Khác

Câu hỏi 2

Tốc độ ra sao?

- Có

4 Chọn template này **5** Đóng

Figure 57 <Web Application> Preview Template

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	questionNum	Question Number	Yes	Yes	Text	String
2	questionContent	Content of Question	Yes	Yes	Text	String
3	optionContent	Content of Option	No	Yes	Radio	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	selectButton	Choose this Template	N/A	Transfer to create content Feedback page
2	cancelButton	Cancel	N/A	Cancel

5.2.9. Create New Blank Feedback

The screenshot shows a modal dialog box with a title bar 'Tạo phản hồi mới' (Create New Blank Feedback). Inside, there are two text input fields: 'Tên feedback' (Name of Feedback) and 'Mô tả' (Description of Feedback). Below the fields are two buttons: a green 'Tiếp tục' (Continue) button and a grey 'Đóng' (Close) button. The entire dialog is set against a dark gray background.

Figure 58 <Web Application> Create New Blank Feedback

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
1	fbName	Name of Feedback	No	Yes	Text	String
2	fbDes	Description of Feedback	No	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	selectButton	Create new blank Feedback	N/A	Transfer to create content Feedback page
2	cancelButton	Cancel	N/A	Cancel

5.2.10. Create Feedback Content

Xóa 1

Tốc độ giảng dạy của giáo viên thế nào? * 2

Nhanh 3

Vừa

Khác 4

Câu hỏi bắt buộc: 5

Nội dung: Tốc độ giảng dạy của giáo viên thế nào? 6

Loại đánh giá: Chuyên cần 7

Thêm câu trả lời Khác: 8

Nhanh 9 Trọng số: 1.0 10 × 11

Vừa Trọng số: 2.0 × 12 + Thêm 12

13 Nhóm radio
14 Nhóm checkbox
15 Trường nhập văn bản(đơn dòng)
16 Trường nhập văn bản(đa dòng)

Figure 59 <Web Application> Create Feedback Content

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type
2	quesContent	Content of Question	Yes	Yes	Text	String
3	optContent	Content of Option	Yes	Yes	Radio	String
4	otherText	Other Answer	Yes	No	Text	String
5	requireCheck	Check if this question is required	No	No	Checkbox	String
6	quesText	Edit content of question	No	Yes	Text	String
7	criteriaList	Choose question Criteria	No	Yes	Dropdown List	String
8	otherCheck	Check if this question has "Other Answer"	No	No	Checkbox	String
9	optText	Edit content of option	No	Yes	Text	String
10	point	Edit option point	No	Yes	Number	Integer

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	delQuesBtn	Delete this question	N/A	Question disappeared

11	delOptBtn	Delete this option	N/A	Option disappeared
12	moreOptBtn	Add more option	N/A	New option appeared
13	addRadioBtn	Drag-n-Drop more Radio question	N/A	New Radio question appreared
14	addChkBtn	Drag-n-Drop more Checkbox question	N/A	New Checkbox question appreared
15	addTextBtn	Drag-n-Drop more Text question	N/A	New Text question appreared
16	addTxtAreaBt n	Drag-n-Drop more Text Area question	N/A	New Text Area question appreared

5.2.11. Overview Creating Feedback

Vui lòng kiểm tra & hoàn tất những thông tin sau

1 Tiêu đề: Feedback cho các lớp Front-end web development - PRO201

2 Mô tả: Phục vụ cho việc đánh giá bộ môn học kì Spring 2018

3 Học kỳ: Spring 2018

4 Bắt đầu nhận phản hồi: 04/01/2018

5 Kết thúc nhận phản hồi: 04/30/2018

6 Đối tượng được đánh giá: Lớp

7 Xem nội dung feedback

8 Xem đề xuất cải tiến

9 10 kết quả mỗi trang

11 Lưu & tiến hành feedback

12 Hủy

Môn học	Mã môn	Học kỳ	Lớp	Giảng viên	Tùy chọn
Front-end web development	PRO201	Spring 2017	SE1061	Nguyễn Huy Hoàng	Danh sách người làm feedback + Xoá
Front-end web development	PRO201	Fall 2017	SE1061	Nguyễn Huy Hiệu	Danh sách người làm feedback + Xoá
Môn học	Mã môn	Học kỳ	Lớp	Giảng viên	Tùy chọn

Hiển thị từ 1 tới 2 trong số 2 kết quả

Trước 1 Sau

Lưu feedback thành:

10 Lưu & tiến hành feedback
 Mẫu mới
 Cập nhật mẫu đã chọn

11 [Lưu](#) [Hủy](#) 12

Figure 60 <Web Application> Overview Creating Feedback

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	feedbackName	Feedback's name	No	Yes	Text	String
2	feedbackDes	Feedback's description	No	Yes	Text	String
3	semester	Feedback's semester	No	Yes	Dropdown List	String
4	startdate	Feedback's start date	No	Yes	Date picker	Date
5	enddate	Feedback's end date	No	Yes	Date picker	Date
6	type	Type of feedback	No	Yes	Dropdown List	String
9	tableTarget	Table shows list targets of Feedback	No	Yes	Table	N/A
10	optionSave	Choose type of saving options	No	Yes	Radio	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
7	prevFeedbackContent	Preview content of selected feedback	N/A	Open a modal pop-up display content of feedback
8	viewSuggestion	Edit selected feedback's suggestions	N/A	Redirect to edit suggestion page
11	saveButton	Save Feedback	N/A	Redirect to list Feedback page
12	cancel	cancel	N/A	cancel

5.2.11.1. Table List Class Targets

Môn học	Mã môn	Học kỳ	Lớp	Giảng viên	Tùy chọn
Front-end web development	PRO201	Spring 2017	SE1061	Nguyễn Huy Hoàng	Edit Delete
C# & Dot Net	PRN292	Summer 2017	SE1062	Nguyễn Huy Hoàng	Edit Delete
Object-Oriented Programming using Java	PRO192	Summer 2017	SE1062	Nguyễn Huy Hiệu	Edit Delete
Front-end web development	PRO201	Fall 2017	SE1061	Nguyễn Huy Hiệu	Edit Delete
Môn học	Mã môn	Học kỳ	Lớp	Giảng viên	Tùy chọn

Hiển thị từ 1 tới 4 trong số 4 kết quả

Trước 1 Sau

Figure 61 <Web Application> Table List Class Targets

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	courseName	Name of class's course	Yes	Yes	Text	String
2	courseCode	Code of class's course	Yes	Yes	Text	String
3	semester	Class's semester	Yes	Yes	Text	String
4	className	Class's name	Yes	Yes	Text	String
5	lecturer	Class's Lecturer	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	editConductor	Modify conductors of this target	N/A	Go to customize list conductors page for selected target
7	editTarget	Modify this feedback target	N/A	Go to customize feedback's targets page
8	removeTarget	Remove selected target	N/A	Remove selected target

5.2.11.2. Table List Course Targets

Đối tượng được đánh giá		Môn học				
10	kết quả mỗi trang				Tìm:	
Môn học		Mã môn	Chuyên ngành	Tùy chọn		
1Front-end web development	2PRO201	3Kỹ thuật phần mềm	4 Danh sách người làm feedback	5 6		
Object-Oriented Programming using Java	PRO192	Kỹ thuật phần mềm	Danh sách người làm feedback			
Programming Fundamentals Using C	PRF192	Kỹ thuật phần mềm	Danh sách người làm feedback			
Môn học	Mã môn	Chuyên ngành	Tùy chọn			

Hiển thị từ 1 tới 3 trong số 3 kết quả

Trước 1 Sau

Figure 62 <Web Application> Table List Course Targets

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	courseName	Name of class's course	Yes	Yes	Text	String
2	courseCode	Code of class's course	Yes	Yes	Text	String
3	major	Name of course's major	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
4	editConductor	Modify conductors of this target	N/A	Go to customize list conductors page for selected target
5	editTarget	Modify this feedback target	N/A	Go to customize feedback's targets page
6	removeTarget	Remove selected target	N/A	Remove selected target

5.2.11.3. Table List Major Targets

Đối tượng được đánh giá		Chuyên ngành				
10	kết quả mỗi trang				Tìm:	
Tên	Mã ngành	Tùy chọn				
1An toàn thông tin	2IA	3 Danh sách người làm feedback	4 5			
Kỹ thuật phần mềm	SE	Danh sách người làm feedback				
Quản trị kinh doanh	SB	Danh sách người làm feedback				
Tên	Mã ngành	Tùy chọn				

Hiển thị từ 1 tới 3 trong số 3 kết quả

Trước 1 Sau

Figure 63 <Web Application> Table List Major Targets

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	majorName	Major's name	Yes	Yes	Text	String
2	majorCode	Major's code	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	editConductor	Modify conductors of this target	N/A	Go to customize list conductors page for selected target
4	editTarget	Modify this feedback target	N/A	Go to customize feedback's targets page
5	removeTarget	Remove selected target	N/A	Remove selected target

5.2.11.4. Table List Department Targets

The screenshot displays a table list of department targets. The columns are labeled 'Phòng ban' and 'Tùy chọn'. The 'Tùy chọn' column contains links to edit or delete targets, and a link to view the list of people who have given feedback. The table includes rows for Công tác Sinh viên, IT, Tuyển Sinh, and Đào tạo. A search bar and pagination controls are also visible.

Figure 64 <Web Application> Table List Department Targets

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	depName	Department's name	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	editConductor	Modify conductors of this target	N/A	Go to customize list conductors page for selected target
3	editTarget	Modify this feedback target	N/A	Go to customize feedback's targets page
4	removeTarget	Remove selected target	N/A	Remove selected target

5.2.11.5. Preview Edit Feedback Content

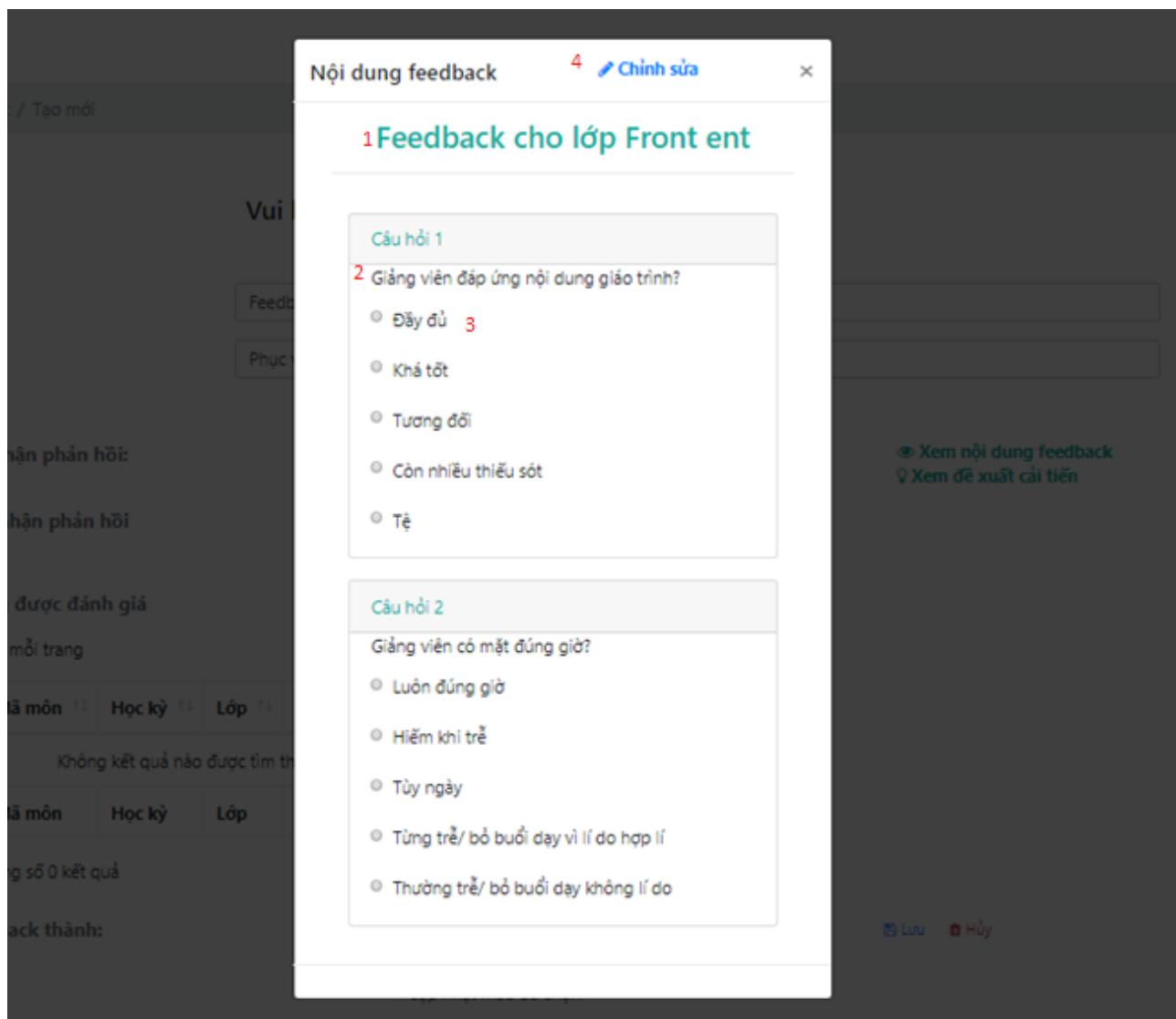


Figure 65 <Web Application> Preview Edit Feedback Content

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	title	Feedback's name	Yes	Yes	Text	String
2	questionContent	Content of question	Yes	Yes	Text	String
3	optionContent	Content of option	Yes	Yes	Radio	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
4	edit	Go to edit feedback's content page	N/A	edit feedback's content page

5.2.11.6. Modify Suggestions

Đưa ra các đề xuất cải tiến dựa theo câu hỏi

Câu hỏi 1
1

Tốc độ giảng dạy của giáo viên thế nào? 2

3

Câu hỏi 2
2

Giảng viên có đảm bảo đúng giáo trình?

Giảng viên nên đảm bảo nội dung giáo trình cho sinh viên.

Câu hỏi 3
3

Giảng viên có đến lớp đúng giờ?

4 Quay lại Tiếp tục 5 5

50% Gửi 6

Figure 66 <Web Application> Modify Suggestions

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	quesNum	Number of Question	Yes	Yes	Text	String
2	quesContent	Content of Question	Yes	Yes	Text	String
3	sugContent	Content of Suggestion	No	No	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
4	prevBtn	Previous Page	N/A	Transfers to previous page of modifying suggestion page
5	nextBtn	Next Page	N/A	Transfers to next page of modifying suggestion page
4	sendBtn	Submit	N/A	Transfers to overview page

5.2.11.7. Modify Targets

5.2.11.7.1. Modify Class Target

1 [Hoàn tất](#)

Chọn lớp được feedback

2 Chuyên ngành
3 Môn học
4 Học kỳ
5 Lớp
6 Giảng viên
7 Tất cả ▾

10 kết quả mỗi trang
Tìm:

Lớp	Môn học	Mã môn	Học kỳ	Giảng viên
<input checked="" type="checkbox"/> 11SE1061	12Front-end web development	13PRO201	14Spring 2017	15Nguyễn Huy Hoàng
<input type="checkbox"/> SE1061	Programming Fundamentals Using C	PRF192	Spring 2017	Nguyễn Huy Hiệu
<input checked="" type="checkbox"/> SE1062	C# & Dot Net	PRN292	Summer 2017	Nguyễn Huy Hoàng
<input type="checkbox"/> SE1062	Object-Oriented Programming using Java	PRO192	Summer 2017	Nguyễn Huy Hiệu
<input type="checkbox"/> SE1063	Programming Fundamentals Using C	PRF192	Fall 2017	Nguyễn Huy Hoàng
<input type="checkbox"/> SE1061	Front-end web development	PRO201	Fall 2017	Nguyễn Huy Hiệu
<input type="checkbox"/> SE1062	Alice	PRO001	Spring 2018	Nguyễn Huy Hoàng

Figure 67 <Web Application> Modify Class Targets

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
2	searchMajor	Input keyword for searching classes by major's name	No	No	Datalist	String
3	searchCourse	Input keyword for searching classes by course's name	No	No	Datalist	String
4	searchSemester	Input keyword for searching classes by semester's name	No	No	Datalist	String
5	searchClass	Input keyword for searching classes by class's name	No	No	Datalist	String
6	searchLecturer	Input keyword for searching classes by lecturer's name	No	No	Datalist	String
7	filterAllSelected	Filter results of targets displayed in table by whether they were selected/ get all result	No	No	Dropdown List	String
8	searchAllFields	Input keyword for searching classes by all fields in table	No	No	Text	String
11	className	Name of class	Yes	Yes	Text	String
12	courseName	Name of class's course	Yes	Yes	Text	String
13	courseCode	Code of class's course	Yes	Yes	Text	String
14	semester	Name of class's semester	Yes	Yes	Text	String
15	lecturer	Name of class's lecturer	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	completeBtn	Finish modify feedback's targets	N/A	Transfer to Feedback's overview page
9	removeTarget	Remove selected class as a target	N/A	Removed target & updated results in table
10	addTarget	Add selected class as target	N/A	Added target & updated results in table

5.2.11.7.2. Modify Course Target

Chọn môn học được feedback

Môn học	Mã môn	Chuyên ngành
C# & Dot Net	PRN292	SE - Kỹ thuật phần mềm
Object-Oriented Programming using Java	PRO192	SE - Kỹ thuật phần mềm
Programming Fundamentals Using C	PRF192	SE - Kỹ thuật phần mềm
Front-end web development	PRO201	SE - Kỹ thuật phần mềm
Alice	PRO001	SE - Kỹ thuật phần mềm
Applied Graphic Design	GDP101	SE - Kỹ thuật phần mềm
Introduction to Computing	CSI101	SE - Kỹ thuật phần mềm

Figure 68 <Web Application> Modify Course Targets

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	searchMajor	Input keyword for searching classes by major's name	No	No	Datalist	String
2	courseName	Name of course	Yes	Yes	Text	String
3	courseCode	Code of course	Yes	Yes	Text	String
4	major	Name of major whom course belong to	Yes	Yes	Text	String

5.2.11.7.3. Modify Major Target

Chọn chuyên ngành được feedback



A screenshot of a web application interface. At the top left, there is a dropdown menu set to '10' and a search bar containing 'kết quả mỗi trang'. On the right, there is a search input field labeled 'Tim:' with a magnifying glass icon. Below this is a table with two columns: 'Chuyên ngành' (Majors) and 'Mã ngành' (Codes). The table contains five rows, each with a checkbox in the first column. The data is as follows:

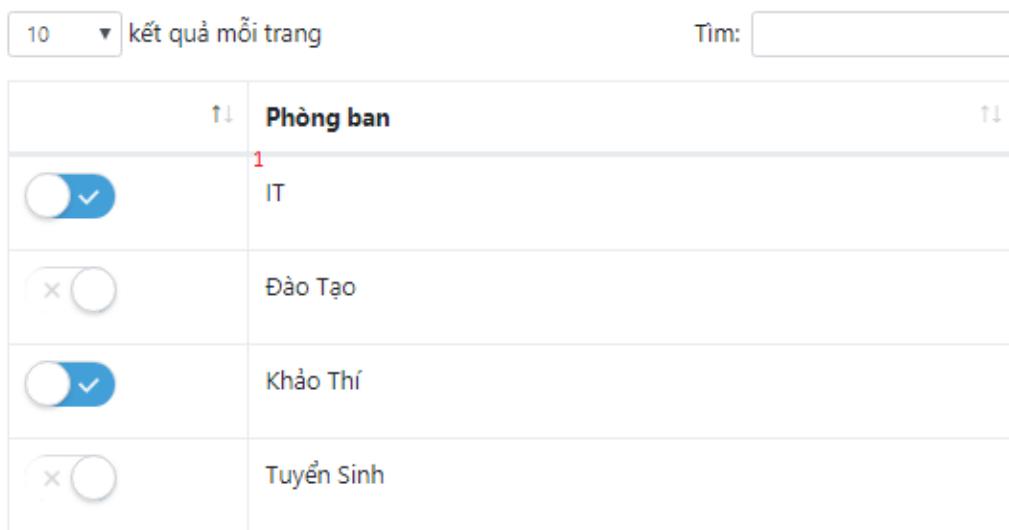
	Chuyên ngành	Mã ngành
<input checked="" type="checkbox"/>	Kỹ thuật phần mềm	SE
<input checked="" type="checkbox"/>	An toàn thông tin	IA
<input checked="" type="checkbox"/>	Quản trị kinh doanh	SB
<input checked="" type="checkbox"/>	Tài chính - Ngân hàng	
<input checked="" type="checkbox"/>	Thiết kế đồ họa	

Figure 69 <Web Application> Modify Major Targets

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	majorName	Major's name	Yes	Yes	Text	String
2	majorCode	Major's code	Yes	Yes	Text	String

5.2.11.7.4. Modify Department Target

Chọn phòng ban được feedback



A screenshot of a web application interface. At the top left, there is a dropdown menu set to '10' and a search bar containing 'kết quả mỗi trang'. On the right, there is a search input field labeled 'Tim:' with a magnifying glass icon. Below this is a table with one column: 'Phòng ban' (Departments). The table contains four rows, each with a checkbox in the first column. The data is as follows:

Phòng ban
<input checked="" type="checkbox"/> IT
<input checked="" type="checkbox"/> Đào Tạo
<input checked="" type="checkbox"/> Khảo Thí
<input checked="" type="checkbox"/> Tuyển Sinh

Figure 70 <Web Application> Modify Department Targets

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	depName	Department's name	Yes	Yes	Text	String

5.2.11.8. Modify Conductors

Chọn những người được làm feedback

Giảng viên Nhân viên Sinh viên

Chuyên ngành 2 Môn học 3 Học kì 4 Giảng viên 5 Lớp 6

10 kết quả mỗi trang Tim: 14

15 Chọn tất cả	Họ & tên	MSSV	Chuyên ngành	Môn đã đăng kí	Lớp đã đăng kí
<input checked="" type="checkbox"/> 7	Thái Tiến Hoàng 9	SE61882 10	SE - Kỹ thuật phần mềm 11	PRO201, PRF192, PRO001, PRJ311, CEA201, 12	SE1061, SE1063, SE1062, SE1265, SE1266, 13
<input type="checkbox"/>	Nguyễn Việt Tú	SE61882	SE - Kỹ thuật phần mềm	PRO201, PRN292, PRO192, GDP101, CSI101,	SE1061, SE1062, SE1163, SE1164,
<input checked="" type="checkbox"/> 8	Hoàng Quốc Việt	SE61745	SE - Kỹ thuật phần mềm		
Chọn tất cả	Họ & tên	MSSV	Chuyên ngành	Môn đã đăng kí	Lớp đã đăng kí

Hiển thị từ 1 tới 3 trong số 3 kết quả Trước 1 Sau

Figure 71 <Web Application> Modify Conductors

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
2	searchMajor	Input keyword for searching classes by major's name	No	No	Datalist	String
3	searchCourse	Input keyword for searching classes by course's name	No	No	Datalist	String
4	searchSemester	Input keyword for searching classes by semester's name	No	No	Datalist	String
5	searchLecturer	Input keyword for searching classes by lecturer's name	No	No	Datalist	String
6	searchClass	Input keyword for searching classes by class's name	No	No	Datalist	String
14	searchAllFields	Input keyword for searching classes by all fields in table	No	No	Text	String
9	fullname	User's fullname	Yes	Yes	Text	String
10	code	User's code	Yes	Yes	Text	String
11	major	User's major	Yes	Yes	Text	String
12	enrolledClasses	Names of enrolled classes	Yes	Yes	Text	String
13	learnedCourse	Name of learned courses	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	completeBtn	Finish modify feedback's	N/A	Transfer to Feedback's

		targets		overview page
8	removeTarget	Remove selected class as a target	N/A	Removed target & updated results in table
9	addTarget	Add selected class as target	N/A	Added target & updated results in table
15	selectAll	Select all targets	N/A	Added all target & updated results in table

5.2.12. View List Feedbacks

Danh sách Feedbacks										
Trạng thái	Tất cả	Loại Feedback	Xem toàn bộ	Sắp xếp	A-Z	3				
1		2								
Tên	4	Kì học	5	Ngày bắt đầu	6	Ngày kết thúc	7	Loại Feedback	8	Trạng thái
Feedback Các Lớp C		Spring 2018		25-03-2018		25-04-2018		Lớp		Làm lại feedback <input checked="" type="checkbox"/> 11
Feedback Các Lớp C#		Spring 2018		25-03-2018		25-04-2018		Lớp		Làm feedback <input checked="" type="checkbox"/> 10
Feedback Phòng IT		Spring 2018		13-03-2018		13-04-2018		Phòng ban		Feedback đã quá hạn hoặc chưa bắt đầu 9

Figure 72 <Web Application> View List Feedbacks

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	filterStatus	Filter Feedback by its Status (Conducted or Not)	No	Yes	Dropdown List	String
2	filterType	Filter Feedback by its Type (Class, Major, Course, Department)	No	Yes	Dropdown List	String
3	sort	Sort Feedback by name	No	Yes	Dropdown List	String
4	name	Feedback Name	Yes	Yes	Text	String
5	semester	Feedback Semester	Yes	Yes	Text	String
6	startDate	Feedback Start Date	Yes	Yes	Text	String
7	endDate	Feedback End Date	Yes	Yes	Text	String
8	type	Feedback Type	Yes	Yes	Text	String
9	fbStatus	Status states that Feedback is expired	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
10	conductBtn	Conduct Feedback	N/A	Transfer to Conduct Feedback Page
11	editBtn	Edit Feedback Answers	N/A	Transfer to Edit Answers Page

5.2.13. Conduct Feedbacks

Feedback Các Lớp Java 1

Câu hỏi 1 * 2

Giảng viên có đến lớp đúng giờ? 3

Có 4
 Không
 Lúc sớm/lúc không
 Khác

Nhập câu trả lời của bạn vào đây 5

Câu hỏi 2 *

Thắc mắc của anh/chị thường được giải đáp lúc nào?

Ngay tại chỗ
 Cuối buổi học 6
 Buổi học sau
 Không được giải đáp

Feedback Các Lớp Java

Câu hỏi 3

Anh/chị đánh giá kỹ năng chuyên môn của giảng viên thế nào?

7

Câu hỏi 4

Anh/chị có ý kiến gì khác?

10

8 Quay lại Tiếp tục 9

11 Quay lại Tiếp tục 12

50% Gửi kết quả 100%

Figure 73 <Web Application> Conduct Feedbacks

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	feedbackName	Name of Feedback	Yes	Yes	Text	String
2	quesNum	Number of Question	Yes	Yes	Text	String
3	quesContent	Content of Question	Yes	Yes	Text	String
4	optRadio	Content of Radio type Option	No	Yes	Radio	String
5	otherContent	Content when user chooses "Other" option	No	No	Text	String
6	optChkBox	Content of Checkbox type Option	No	Yes	Check Box	String
7	optText	Content of Text type Option	No	Yes	Text	String
10	optTextArea	Content of Text Area type	No	Yes	Text Area	String

		Option				
11	progressBar	Progress bar displays completed percentage	Yes	Yes	N/A	N/A

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
8	prevBtn	Previous Page of Conducting Feedback page	N/A	Transfer to previous page of Conduct Feedback Page
9	nextBtn	Next Page of Conducting Feedback page	N/A	Transfer to next page of Conduct Feedback Page
12	sendBtn	Submit all answers	N/A	Transfer to view list Feedbacks page

5.2.14. View List Feedbacks for Staff and Head Academic

Danh sách Feedbacks

Tên feedback	Kì báo cáo	Từ-đến	Loại	Đối tượng được báo cáo	Tình trạng
Feedback cho phòng IT 5	Spring 2017 6	4/10/2018 - 7 4/10/2018	Phòng ban 8	IT 9	Đã xuất bản 10 Ngưng 11
Tên feedback	Kì báo cáo	Từ-đến	Loại	Đối tượng được báo cáo	Tình trạng

Hiển thị 5 kết quả mỗi trang

Trước 1 Sau

Figure 74 <Web Application> View List Feedbacks for Staff and Head Academic

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	filterStatus	Filter Feedback by its Type	No	Yes	Dropdown List	String
3	filterPage	Number of Feedbacks is shown in one page	No	Yes	Dropdown List	Integer
4	search	Search Feedback	No	Yes	Text	String
5	name	Feedback Name	Yes	Yes	Text	String
6	semester	Feedback Semester	Yes	Yes	Text	String
7	avaiDate	Feedback Available Date	Yes	Yes	Text	String
8	type	Feedback Type	Yes	Yes	Text	String
9	target	Feedback Target	Yes	Yes	Text	String
10	fbStatus	Status states that Feedback is published or not	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	addFBBtn	Add more Feedback	N/A	Transfer to Select Template page
11	deactiveBtn	Unpublish a Feedback	N/A	Open a modal to confirm unpublish Feedback

5.2.15. Publish/Unpublish Feedback

The screenshot illustrates the Publish/Unpublish Feedback feature in a web application. It shows a list of feedback items and two confirmation dialogs.

Main View: A table displays feedback items with the following columns: Tên feedback, Kì báo cáo, Từ-đến, Loại, Đối tượng được báo cáo, Tình trạng, and an empty column. One item is shown: Feedback cho phòng IT, Spring 2017, 4/10/2018 - 4/10/2018, Phòng ban, IT, Đã xuất bản, and Ngung.

Modal 1 (Top): Xác nhận lại việc ngưng xuất bản

- Sau khi ngưng xuất bản, người dùng sẽ không thể làm feedback này. Bạn có chắc chắn muốn ngưng xuất bản feedback này không? **1**
- 2** Ngưng xuất bản
- 3** Close

Modal 2 (Bottom): Xác nhận lại việc xuất bản

- Sau khi xuất bản, nếu feedback này vẫn còn hạn, người dùng có thể tiếp tục làm feedback này. Bạn có chắc chắn muốn xuất bản feedback này không? **4**
- 5** Xuất bản
- Đóng

Figure 75 <Web Application> Publish/Unpublish Feedback

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	unpubWarn	Warning when unpublish Feedback	Yes	No	Text	String
4	pubWarn	Warning when publish Feedback	Yes	No	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
2	unpubBtn	Unpublish a Feedback	N/A	Unpublish a Feedback and reload screen
3	cancel	Cancel	N/A	Cancel
5	pubBtn	Publish a Feedback	N/A	Publish a Feedback and reload screen

5.2.16. View list Reports

The screenshot shows a web application titled "Báo cáo Feedbacks". On the left is a dark sidebar with five navigation icons (1, 2, 3, 4, 5). The main area has a teal header "Báo cáo Feedbacks". Below it are two dropdown menus: "Loại Feedback" set to "Lớp" and "Sắp xếp" set to "A-Z". The main content is a table with columns "Giảng viên" and "Môn học". The table lists seven rows of data, each with a "Xem báo cáo" button in a red box labeled "3". The data is as follows:

Giảng viên	Môn học
Nguyễn Huy Hiệu	Object-Oriented Paradigm using Java - PRO192
Nguyễn Huy Hiệu	Programming Fundamentals Using C
Nguyễn Huy Hiệu	Applied Graphic Design
Nguyễn Huy Hiệu	Desktop Java Applications
Nguyễn Huy Hiệu	Data Structure and Algorithm
Nguyễn Huy Hoàng	C# & Dot Net
Nguyễn Huy Hoàng	Front-end web development

Figure 76 <Web Application> View list Reports

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	filterType	This box contains 4 types of report: -Departments. -Courses. -Majors. -Courses taught by teachers.	No	Yes	Dropdown List	String
2	filterSort	This box allows user to sort from ascending order or descending order.	No	Yes	Dropdown List	String
4	attrName	Name of the attribute.	Yes	Yes	Text	String
5	tblTarget	Data of corresponding type selected.	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
3	seeReportDetails	Redirect to view reports for	User must	Redirect to view report detail

		this record throughout semesters.	have one of these roles: -Head of Academic. -Staff. -Lecturer.	page.
--	--	-----------------------------------	---	-------

5.2.17. View Report Details

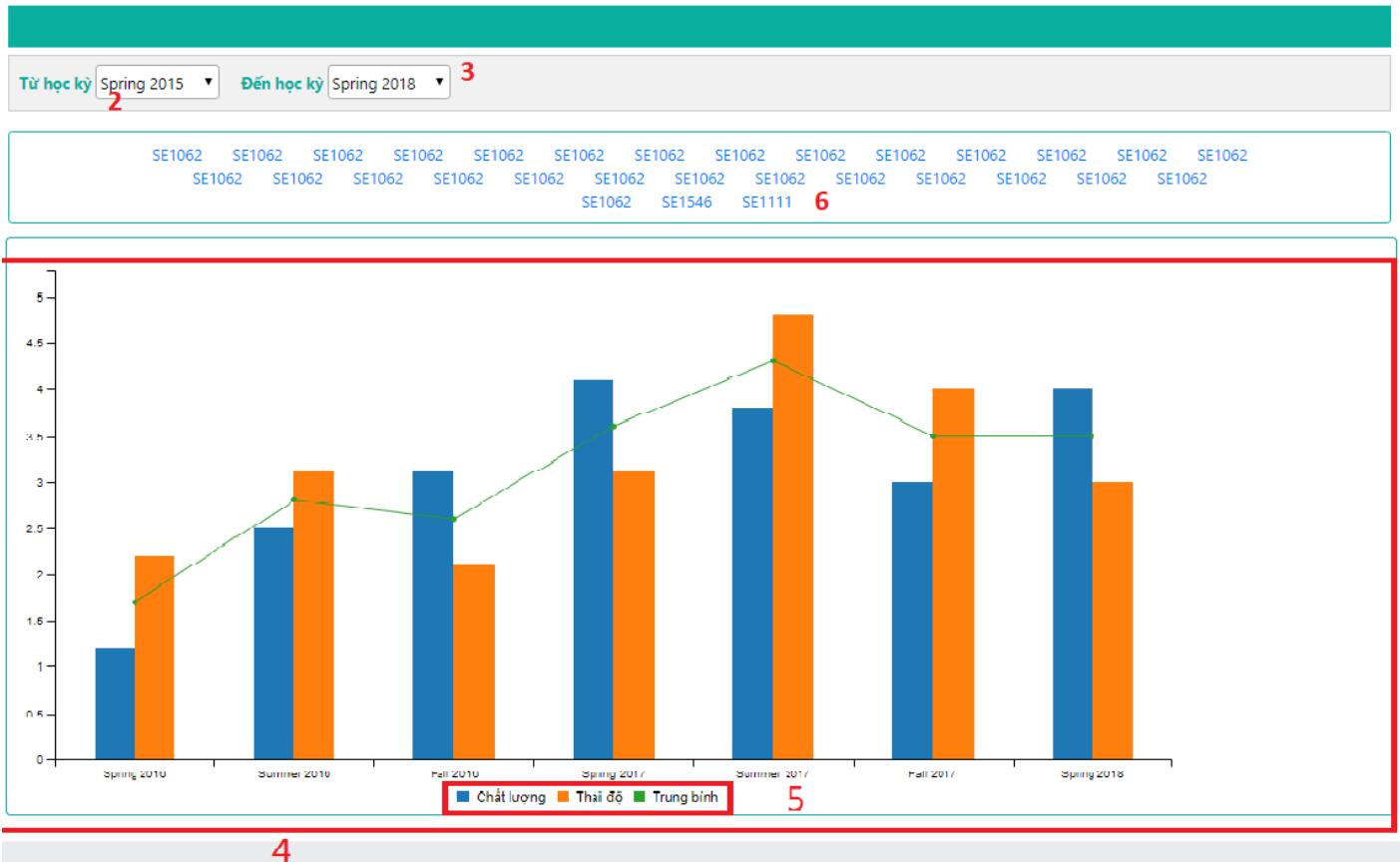


Figure 77 <Web Application> View Report Details

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
2	filterFromSemester	This box allow user to select which semester is starting point.	No	Yes	Dropdown List	String
4	filterToSemester	This box allow user to select which semester is end point.	Yes	Yes	Text	String
4	chartData	Visualized data calculated from feedbacks' result.	Yes	Yes	Bar chart	N/A
5	criteria	Name of criteria calculated. Click on this toggle will hide or unhide corresponding bar on chart	Yes	Yes	Toggle	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
6	targetName	List of target that corresponding with the chosen report bar		Transfer to View Report semester details page

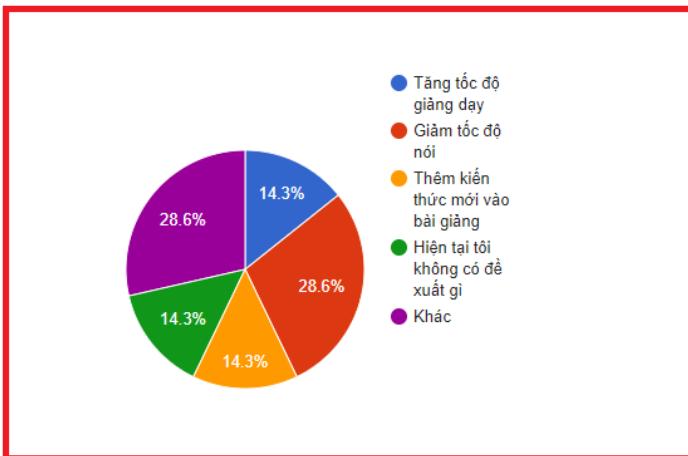
5.2.18. View Semester Report Details

The screenshot shows a user interface for viewing semester report details. The top navigation bar includes links to Dashboard, Quản lý feedback, and Thông tin chi tiết feedback. The main title is 'Kết quả feedback Lớp SE1061 - Spring 2017'. A summary box displays the average score as 'Điểm trung bình: 3.5/5'. Below this, a red box highlights the recommendation 'Chuyên cần-3.1/5'. The interface is annotated with numbers 1 through 7 to identify specific UI elements:

- 1: Top navigation link
- 2: Main title bar
- 3: Average score box
- 4: Recommendation box
- 5: Suggestion checkbox
- 6: Suggestion input field
- 7: Suggestion count

A pie chart at the bottom shows student responses to a question, with the following distribution:

Response Category	Percentage
Tăng tốc độ giảng dạy	14.3%
Giảm tốc độ nói	28.6%
Thêm kiến thức mới vào bài giảng	14.3%
Hiện tại tôi không có đề xuất gì	14.3%
Khác	28.6%



Ngày nhập	Nội dung
18-03-2018 19:41	Không có ý kiến gì :D
30-03-2018 00:23	Ahihi khong có gì luôn :3

Xem các ý kiến Khác



9

10



Figure 78 <Web Application> View Semester Report Details

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	targetName	Name of which this report is for	Yes	Yes	Text	String
2	semesterName	Name of the semester	Yes	Yes	Text	String
3	overallAvgPoint	Semester's overall average point.	Yes	Yes	Text	float
4	criteriaAvgPoint	Name of criteria and its average point.	Yes	Yes	Text	String
5	suggestion	Suggestion for this question.	Yes	Yes	Text	String
6	questionContent	Content for this question.	Yes	Yes	Text	String
7	responseCount	Number of user response for this question.	Yes	Yes	Text	int
8	pieChart	Pie chart representing options for this question and its number.	Yes	Yes	N/A	N/A
9	otherOptionTable	Table representing which users inputted after they select "Khác" option.	Yes	No	N/A	String
10	toggleOtherOptionTable	Show/hide users' inputted answers	Yes	No	Toggle	N/A

5.2.19. View List Users

Quản lý người dùng hệ thống

1 + Thêm người dùng

2 Tất cả phân quyền 3 Tất cả phòng 4 Tất cả chuyên ngành 5 Tất cả

5 kết quả mỗi trang 6

7 Tìm:

Table Data:

Họ & Tên	Phân quyền	Phòng ban	Chuyên ngành	Email	Trạng thái	
admin 11	Trưởng ban 12	Đào Tạo 13	N/A 14	viethoangquoc2711@gmail.com 15	Đang hoạt động 16	Chỉnh sửa
Hoàng Quốc Việt	Giảng viên	Đào Tạo	Kỹ thuật phần mềm	viethqse61745@fpt.edu.vn	Đang hoạt động	Chỉnh sửa
Hoàng Quốc Việt	Học sinh	N/A	Kỹ thuật phần mềm	viethoangquoc2711@gmail.com	Đang hoạt động	Chỉnh sửa
Nguyễn Huy Hiệu	Giảng viên	Đào Tạo	Kỹ thuật phần mềm	viethqse61745@fpt.edu.vn	Đang hoạt động	Chỉnh sửa
Nguyễn Huy Hoàng	Giảng viên	Đào Tạo	Kỹ thuật phần mềm	viethqse61745@fpt.edu.vn	Đang hoạt động	Chỉnh sửa
Họ & Tên	Phân quyền	Phòng ban	Chuyên ngành	Email	Trạng thái	

Hiển thị từ 1 tới 5 trong số 10 kết quả

Trước 1 2 Sau 9

Figure 79 <Web Application> View List Users

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
2	filterByRole	Filter user by role	No	Yes	Dropdown List	String
3	filterByDepartment	Filter user by department	No	Yes	Dropdown List	String
4	filterByMajor	Filter users by major	No	Yes	Dropdown List	String
5	filterByStatus	Filter users by status	No	Yes	Dropdown List	String
7	txtSearch	Search user	No	Yes	Text	String
8	tblUser	Table representing list of users	Yes	Yes	Table	String
11	fullname	Fullname of user	Yes	Yes	Text	String
12	role	Role of user	Yes	Yes	Text	String
13	department	Department of user	Yes	Yes	Text	String
14	major	Major of user	Yes	Yes	Text	String
15	email	Email of user	Yes	Yes	Text	String
16	status	Status of user	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
1	btnAddUser	Add New User Button	N/A	Redirect User to
9	btnPagination	Change page of table	N/A	Change which page is the table on
10	btnEditUser	Edit a user	N/A	Transfer to edit user page

5.2.20.Add new User

Vui lòng kiểm tra & hoàn tất những thông tin sau

1	Tên đăng nhập:	
2	Mật khẩu mới:	
3	Nhập lại mật khẩu:	
4	Họ & tên:	
5	Ngày sinh:	dd----yyyy
6	Email:	
7	Giới tính:	<input type="radio"/> Nam <input checked="" type="radio"/> Nữ
8	Phân quyền:	Giảng viên
9	Chuyên ngành:	N/A
10	Phòng ban:	Đào tạo
11	Đang hoạt động:	<input type="checkbox"/>
12	Mã code:	

 Lưu và tiếp tục
 Lưu và quay về danh sách
 Hủy

13 14 15

Figure 80 <Web Application> Add New User

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	txtUsername	User's username	No	Yes	Text	String
2	txtPassword	User's password	No	Yes	Password	String
3	txtConfirmPassworm	User retypes password	No	Yes	Password	String
4	txtFullname	User's full name	No	Yes	Text	String
5	txtBirth	User's birth	No	Yes	date	Date
6	txtEmail	User's email	No	Yes	email	String
7	rbtGender	User's gender	No	Yes	boolean	String
8	selectRole	User's role	No	Yes	Dropdown List	String
9	selectMajor	User's major	No	Yes	Dropdown List	String
10	selectDept	User's department	No	Yes	Dropdown List	String
11	chkStatus	User's status	No	Yes	Checkbox	String
12	txtCode	User's code	No	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
13	saveAndContinue	Save User and continue to add new user	All required field must be filled. Username, email, code must not be existed in database. Username must contain only character and number.	Save then reload page.

			Password must be at least 6 characters. Email must match format. Confirm password must match password.	
14	saveAndBackToList	Save User and then go back to see list users	All required field must be filled. Username, email, code must not be existed in database. Username must contain only character and number. Password must be at least 6 characters. Email must match format. Confirm password must match password.	Save then redirect user to see users page.
15	cancelInput	Clear all inputted data	N/A	All control return to its default.

5.2.21. Edit User

Vui lòng kiểm tra & hoàn tất những thông tin sau

1	Tên đăng nhập:	admin
2	Mật khẩu mới:	
3	Nhập lại mật khẩu:	
4	Họ & tên:	admin
5	Ngày sinh:	06-Jan-1996
6	Email:	viethoangquoc2711@gmail.com
7	Giới tính:	<input checked="" type="radio"/> Nam <input type="radio"/> Nữ
8	Phân quyền:	Trưởng ban
9	Chuyên ngành:	N/A
10	Phòng ban:	Đào Tạo
11	Đang hoạt động:	<input checked="" type="checkbox"/>
12	Mã code:	admin
<input type="button" value="Lưu và tiếp tục"/> <input type="button" value="Lưu và quay về danh sách"/> <input type="button" value="Hủy"/>		
13	14	15

Figure 81 <Web Application> Edit User

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	txtUsername	User's username	Yes	Yes	Text	String
2	txtPassword	User's password	No	Yes	Password	String
3	txtConfirmPasswrm	User retypes password	No	Yes	Password	String
4	txtFullname	User's full name	No	Yes	Text	String
5	txtBirth	User's birth	No	Yes	date	Date

6	txtEmail	User's email	No	Yes	email	String
7	rbtGender	User's gender	No	Yes	boolean	String
8	selectRole	User's role	No	Yes	Dropdown List	String
9	selectMajor	User's major	No	Yes	Dropdown List	String
10	selectDept	User's department	No	Yes	Dropdown List	String
11	chkStatus	User's status	No	Yes	Checkbox	String
12	txtCode	User's code	Yes	Yes	Text	String

Buttons/Hyperlinks

No	Function	Description	Validation	Outcome
13	saveAndContinue	Save User and continue to update user	All required field must be filled. Username, email, code must not be existed in database. Username must contain only character and number. Password must be at least 6 characters. Email must match format. Confirm password must match password.	Save then reload page.
14	saveAndBackToList	Save User and then go back to see list users	All required field must be filled. Username, email, code must not be existed in database. Username must contain only character and number. Password must be at least 6 characters. Email must match format. Confirm password must match password.	Save then redirect user to see users page.
15	cancelInput	Clear all inputted data	N/A	All control return to its default.

5.2.22.Login Mobile

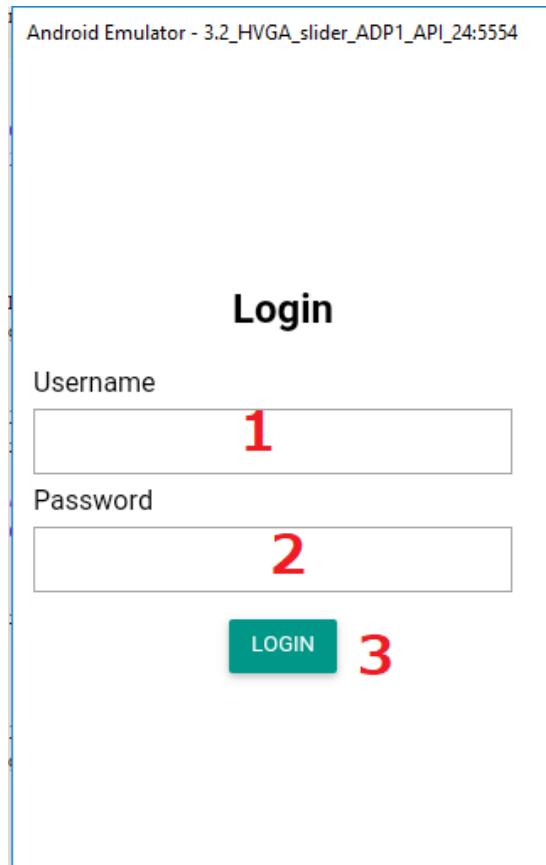


Figure 82 <Mobile Application> Login

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	username	Username	No	Yes	TextView	String
2	password	Password	No	Yes	TextView	String

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
3	loginBtn	Login	N/A	Transfer to View list Feedback page

5.2.23. View List Feedbacks Mobile

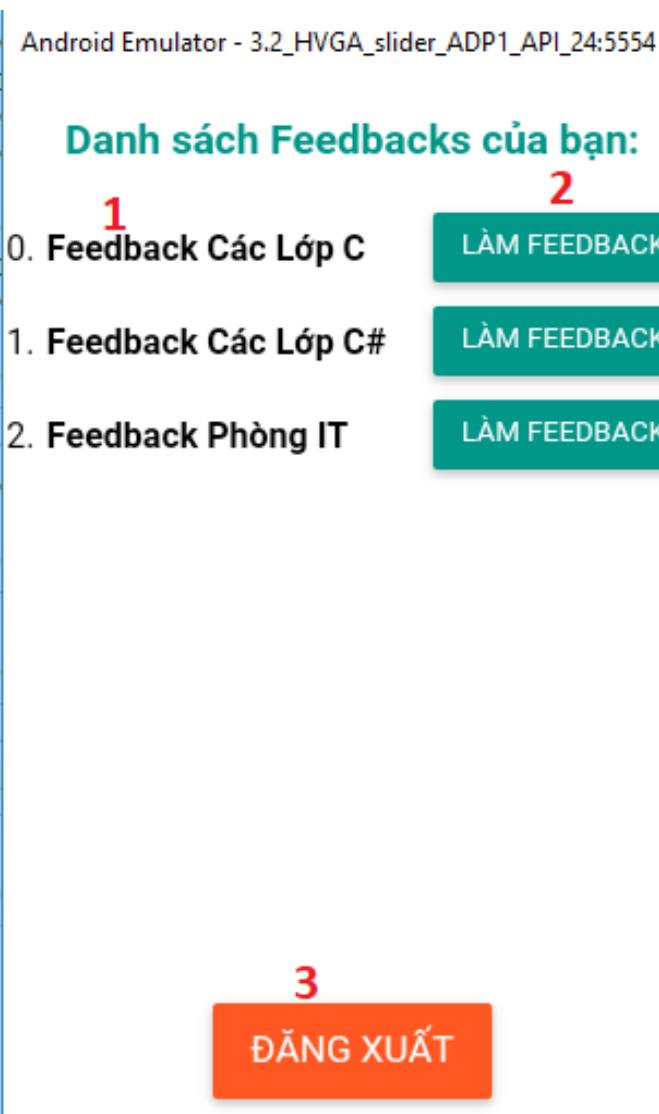


Figure 83 <Mobile Application> View List Feedbacks

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	fbName	Feedback Name	Yes	Yes	TextView	String

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
2	conductBtn	Conduct Feedback	N/A	Transfer to Conduct Feedback screen
3	logoutBtn	Log out	N/A	Transfer to Login Page

5.2.24. Conduct Feedback Mobile

Android Emulator - 3.2_HVGA_slider_ADP1_API_24:5554

Feedback

Câu hỏi 1 1

Tốc độ giảng dạy của giáo viên thế nào?

- Nhanh 2
- Vừa
- Chậm
- Khác

Nhập câu trả lời của bạn vào đây 4

5 6 GỬI KẾT QUẢ

Android Emulator - 3.2_HVGA_slider_ADP1_API_24:5554

Câu hỏi 2

Tốc độ ra sao?

Có 7

Câu hỏi 3

Nội dung của câu hỏi:

8

25% 7 GỬI KẾT QUẢ

Figure 84 <Mobile Application> Conduct Feedback

No	Field Names	Description	Read Only	Mandatory	Control Type	Data Type
1	quesNum	Number of Question	Yes	Yes	TextView	String
2	quesContent	Content of Question	Yes	Yes	TextView	String
3	optRadioContent	Content of Radio Type Option	No	Yes	Radio	String
4	otherText	Content of Option "Other"	No	No	Text	String
5	progressBar	Display completed percentage	Yes	Yes	N/a	N/a
7	optChkBoxContent	Content of Check Box Type Option	No	Yes	Check Box	String
8	optTextContent	Content of Text Type Option	No	Yes	Text	String

Buttons/Hyperlinks:

No	Function	Description	Validation	Outcome
6	submitBtn	Submit Answers	N/A	Submit answers and transfer to view list page

6. Database Design

6.1. Entity relationship diagram (ERD)

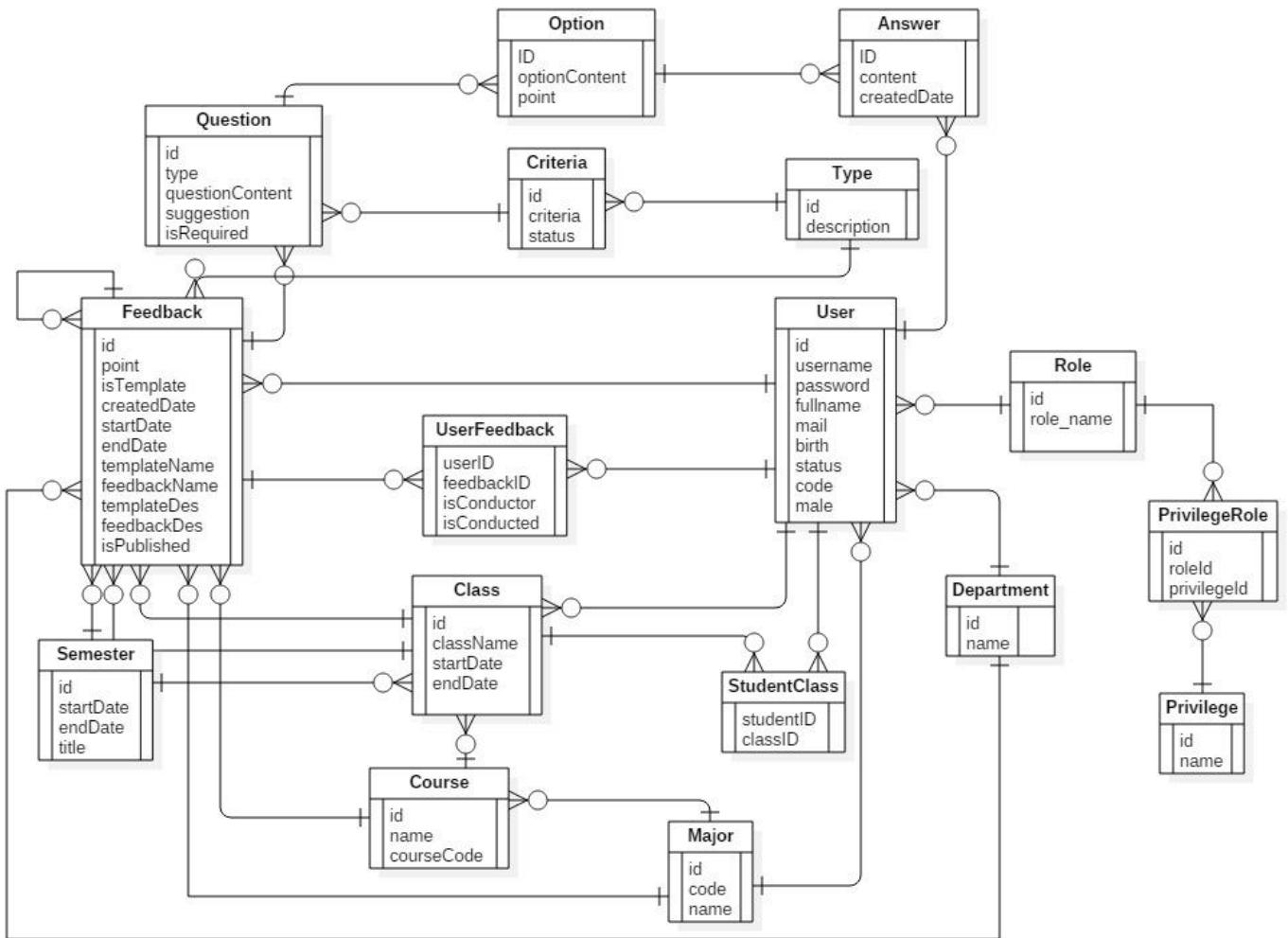


Figure 85 Entity relationship diagram

6.2. Data Dictionary

Entity Data dictionary: describe content of all entities	
Entity Name	Description
Feedback	Contain Feedback information
Question	Contain Question information
Option	Contain Option information
Answer	Contain Answer information
Criteria	Contain Criteria information
Semester	Contain Semester information

Type	Contain Feedback type information
User	Contain User information
UserFeedback	Contain information about relationship between a user with a feedback
Role	Contain Role information
Privilege	Contain Privilege information
PrivilegeRole	Contain Privilege information for each role
StudentClass	Contain information about class of each student
Class	Contain Class information
Course	Contain Course information
Major	Contain Major information
Department	Contain Department information

Table 10 Entity Data dictionary

7. Algorithms

7.1. Edit Feedback Content

7.1.1. Definition

Edit Feedback Content is the way to create new, update, delete existed questions and options that created before.

7.1.2. Define Problem

When choose “Edit Feedback content”, users is redirected to edit content screen which displays all created questions and options. On this screen, users can change type of question, edit question content, create new or delete question. Besides, users can also change point of a option, edit option content, create new or delete option of a question. All changes can only be saved once users click “Send” button. Systems has to recognize and distinguish those changes to take suitable action.

7.1.3. Solution

To solve this problem, we should follow these steps:

- Get all questions and options existing on screen. Attach Question ID for questions that created before. Questions created new don't have Question ID.
- Loop through list questions, if:
 - + Question doesn't have question ID, add new Question.
 - + Question has Question ID, update Question. In each Question, loop through list options, if:
 - Option doesn't have question ID, add new Option.
 - Option has Option ID, update Option.
 - Option has Option ID, but its ID currently isn't in the list of existed Option ID anymore, delete that Option.
 - + Question has Question ID, but its ID currently isn't in the list of existed Question ID anymore, delete that Question.

7.1.4. Complexity

- In total, the complexity of this algorithm is $O(N^2)$

7.2. Match conductors to target

7.2.1. Definition

Matching Conductors to Target is the way to auto finding suitable conductors for that Feedback and add to the list.

7.2.2. Define Problem

A Feedback can be one of four type: Feedback for Major, Course, Class or Department. Each type has a specific type of conductors (can be student in the class, staff in the department) and has different way to get the correct list.

7.2.3. Solution

To solve this problem, we should follow these steps:

- Create database with structure like this:
 - + Table Users: Including all types of users (Head of Academic, Staff, Lecturer, Student). Having field DepartmentID and MajorID. Those two fields are nullable.
 - + Table Class: Including all classes. Having CourseID field which presents which course the class belongs to.
 - + Table Student_Class: This table is used to connect table Users and table Class. It including all user has Student role and their class. Having field UserID and ClassID.
- Get selected type of Feedback. If:
 - + Type is Major: Conductors for Majors type is all users belong to that Major (including Head of Academic, Lecturer and Student). Select from Database, table Users, list users has corresponding Major.
 - + Type is Course: Conductors for Courses type is all students belong to that Course.
 - Select from Database, table Class, all classes has corresponding Course
 - Loop through Class list, select unique from database, table Student_Class, all students belong to each class
 - + Type is Class: Conductors for Class type is all students belong to that Class. Select from database, table Student_Class, all students has corresponding Class.
 - + Type is Department: Conductors for Department type is all users who not belong to that Department (including Head of Academic, Lecturer, Student, Staff from other Department). Select from Database, table Users, list users has Department field different from selected one.

7.2.4. Complexity

- In total, the complexity of this algorithm is $O(N)$ to $O(N^2)$

7.3. Statistic and generate report

7.3.1. Define Problem

Head of Academic, Staff and Lecturers can see reports of Feedbacks. Because we have 4 types of Feedback so as a result, we also have 4 types of Report: Report for Major, Course, Class and Department. Staff can only see report of their Department. Lecturer can only see report of the classes he teaches. Head of Academic can see all reports.

7.3.2. Requirement

- We have 2 types of Report Visualization. First, users will be shown a list of reports, after choosing one, users will be redirected to Bar Chart type report as Figure 81. Bar Chart compares points of the target through semester. When user clicks on a bar, user will be redirected to Pie Chart type report as Figure 82, which shows detailed information of selected semester.

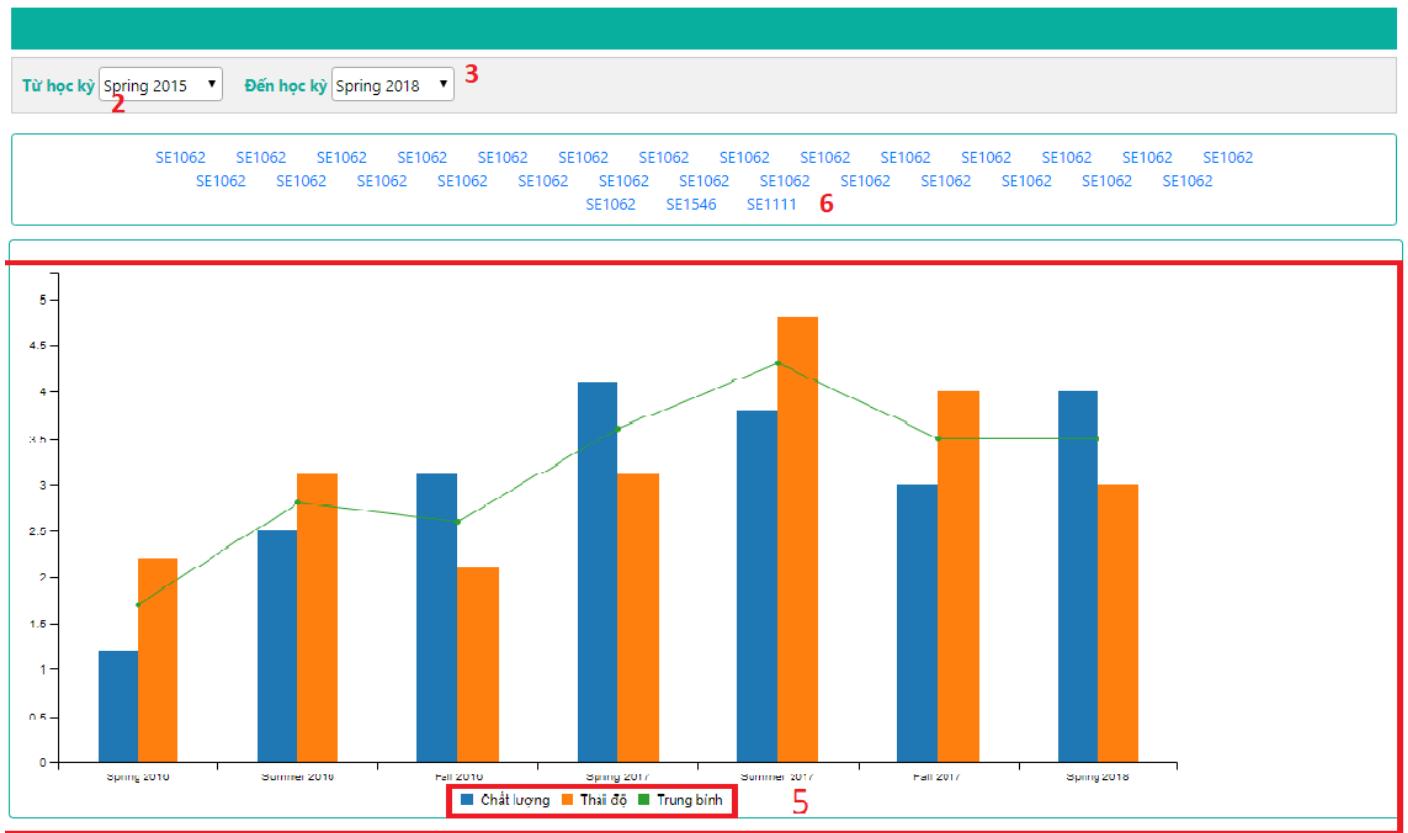


Figure 86 Bar Chart report

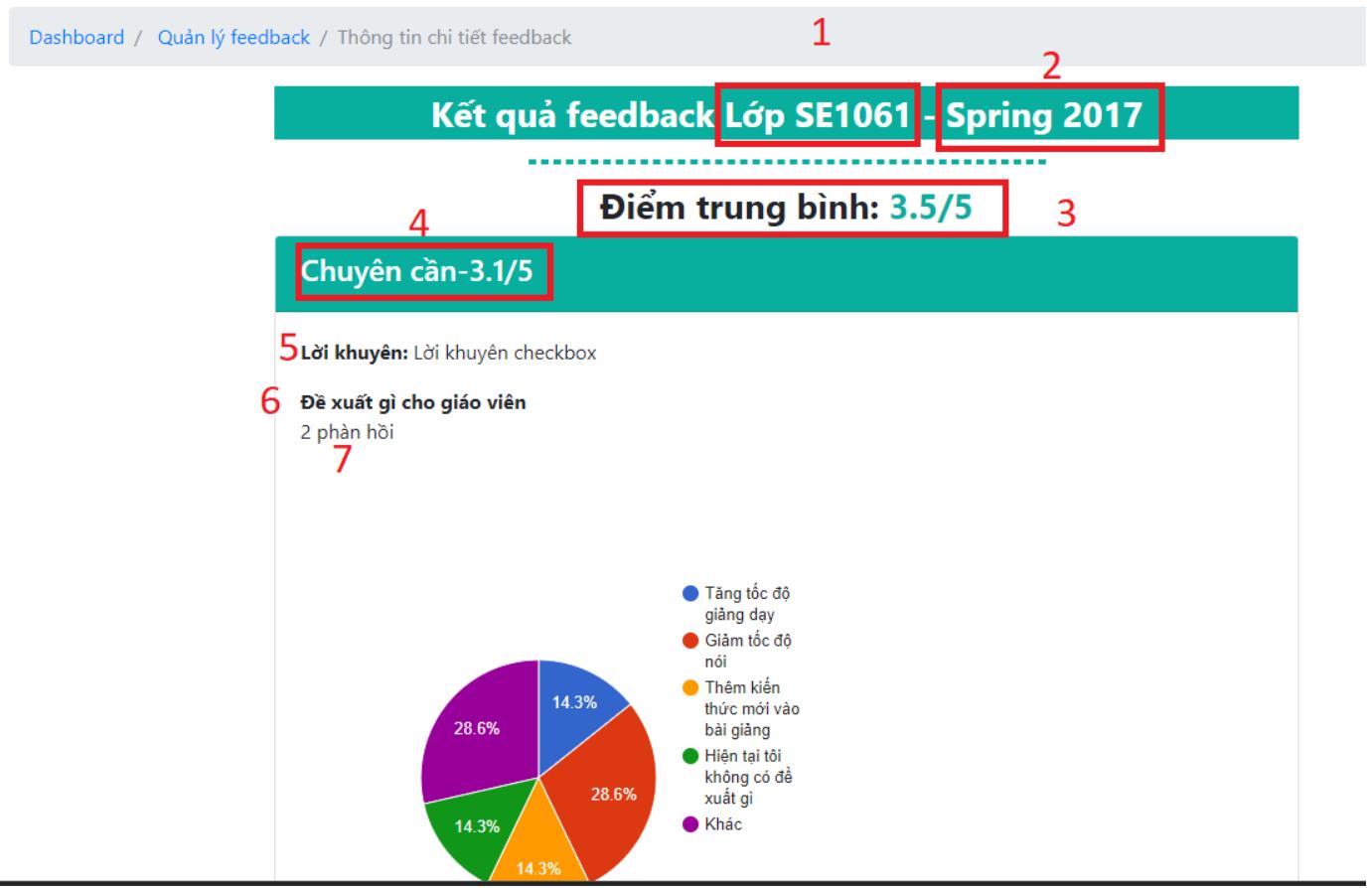


Figure 87 Pie Chart report

- Bar Chart Report:
 - + Major, Course, Department: Each bar shows the average points of a criteria in a semester. Line shows the total average point of a semester.
 - + Class: Each bar shows the average points of a criteria of all classes (That lecturer teaches) of one Course in a semester. Line shows the total average point of all classes (that lecturers teaches) of one Course in a semester.

For example: Lecture Nguyen Huy Hung, has 2 Java Web classes SE1061 and SE1063, in semester Spring 2018. His bar char report shows the average point from 2 classes SE1061 and SE1063.
- Pie Chart Report: Show total average point, average point of each Criteria, Suggestion (if point < 3.4), number of responses, percentage of answers in each question.

7.3.3. Solution

- Each question has many Options. Each Options will have a specific point. When user chooses one answer, its point will be counted.
- Average Criteria Point will be counted as:

$$\text{Average Criteria} = \frac{\sum_{i=1}^n \left(\frac{\sum_{j=1}^m (a_j * p_j)}{\sum_{j=1}^m a_j} \right)}{n}$$

where: n is number of Questions belong to that Criteria

m is number of Options belong to Question i

a is number of Answers of Option j

p is Point of Option j

- Total Average Point will be counted as:

$$\text{Total Average} = \frac{\sum_{i=1}^n (p_i)}{n}$$

where: n is number of Criterias

p is Average Point of Criteria i

7.3.4. Example

- Lecturer Nguyen Huy Hung , has 2 Java Web classes SE1061 and SE1063, in semester Spring 2018. His Feedback form for 2 those classes has the same questions set like this:

- + Question 1 – Criteria 1, Radio type question, has 2 Options:

Option 1 gives him 3 points
Option 2 gives him 5 points

- + Question 2 – Criteria 1, Checkbox type question, has 3 Options:

Option 1 gives him 5 points
Option 2 gives him 4 points
Option 3 gives him 2 points

+ Question 3 - Criteria 2, Radio type question, has 3 Options:

- Option 1 gives him 4 points
- Option 2 gives him 3 points
- Option 3 gives him 2 points

- Now, there are 10 students in his 2 classes, they give answers like this. The number represent number of people chooses that option:

	Option 1	Option 2	Option 3
Question 1	6	4	
Question 2	6	8	3
Question 3	3	4	3

- Average point of Criteria 1 is:

$$\frac{\left(\frac{6*3+4*5}{10} + \frac{6*5+8*4+3*2}{17} \right)}{2} = 3.9$$

- Average point of Criteria 2 is:

$$\frac{3*4+4*3+3*2}{10} = 3$$

- Total Average point is:

$$\frac{3.9+3}{2} = 3.5$$

E. System Implementation & Test

1. Introduction

1.1. Overview

This section describes the approach and methodologies used by group to plan, organize and manage the testing of SFMS system. It provides in the detail all necessary information about the implementation and testing procedure of the system included test plans, test cases, test result, test environments, pass/fail criteria and risks estimations as well as a checklist to cover all possible cases.

1.2. Test Approach

- Goal: Test all features in the whole SFMS system based on the core flow.
- Method: black-box testing
- Technique: check list
- The testing for this project will consists of Integration System test level. Testing the program which was integrated and as a complete system to ensure that the software requirements have been met. Integration testing would be performed by all member of team and approved by team leader
- System testing is focused on assessing the system's reliability. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.

2. Database Relationship Diagram

2.1. Physical Diagram

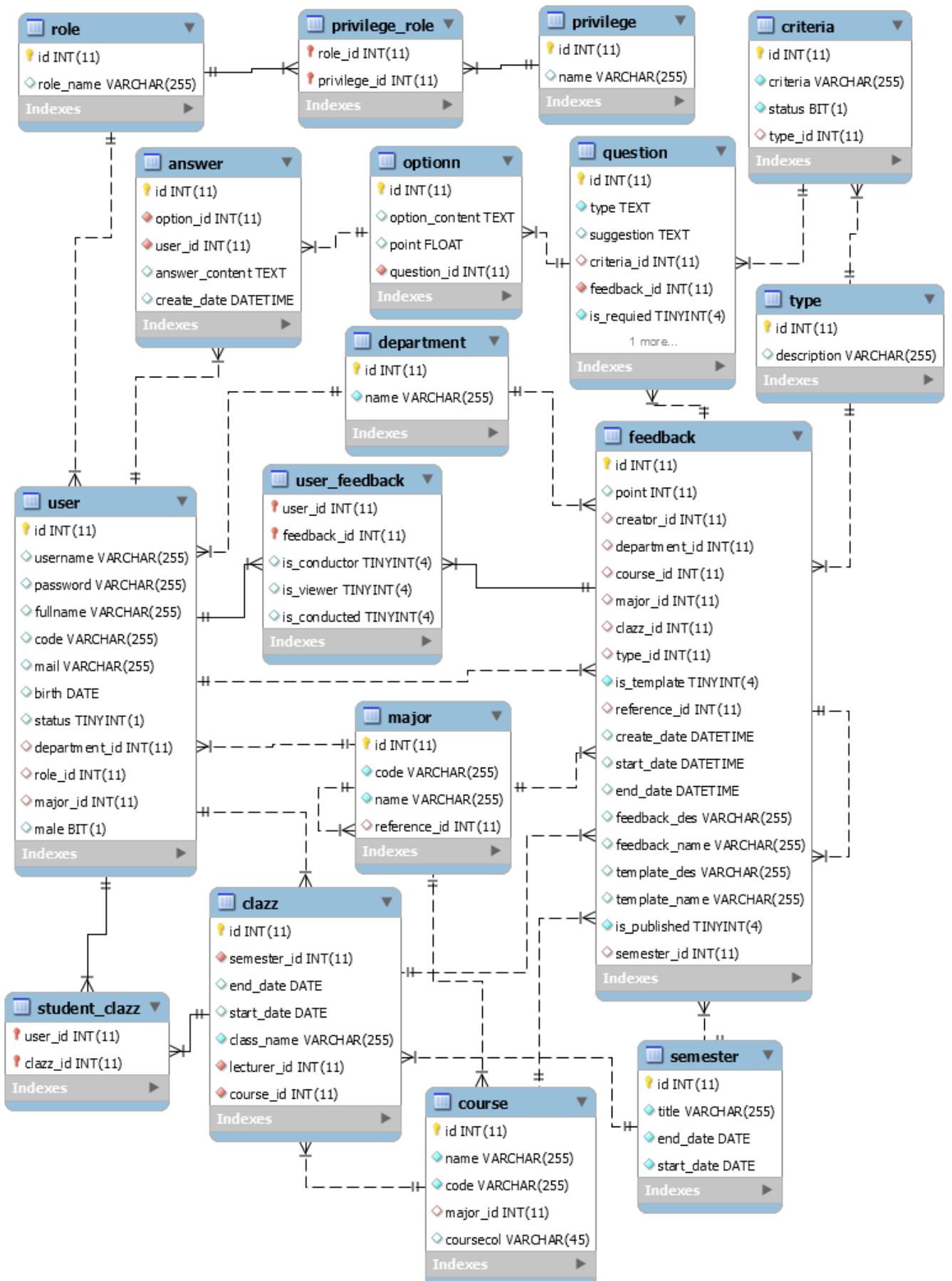


Figure 88 Physical Diagram

2.2. Data Dictionary

Entity Data dictionary: describe content of all tables	
Table Name	Description
Feedback	Contain Feedback information
Question	Contain Question information
Optionn	Contain Option information
Answer	Contain Answer information
Criteria	Contain Criteria information
Semester	Contain Semester information
Type	Contain Feedback type information
User	Contain User information
User_Feedback	Contain information about relationship between a user with a feedback
Role	Contain Role information
Privilege	Contain Privilege information
Privilege_Role	Contain Privilege information for each role
Student_Clazz	Contain information about class of each student
Clazz	Contain Class information
Course	Contain Course information
Major	Contain Major information
Department	Contain Department information

Table 11 Describe content of all tables

Entity name	Attributes	Description	Domain	Null
Feedback	id	Unique identifier of the Feedback	INT(11)	N
	point	Average point of Feedback	INT(11)	Y
	creator_id	Id of the user creates Feedback	INT(11)	N
	department_id	Id of the Department that Feedback belongs to	INT(11)	Y
	course_id	Id of the Course that Feedback belongs to	INT(11)	Y
	major_id	Id of the Major that Feedback belongs to	INT(11)	Y
	clazz_id	Id of the Class that Feedback belongs to	INT(11)	Y
	type_id	Id of the Type that Feedback belongs to	INT(11)	N
	is_template	Feedback is used as Template or not	BIT(1)	N
	reference_id	Id of the Feedback that Feedback uses as Template	INT(11)	Y
	create_date	Created Date of Feedback	DATETIME	N
	start_date	Start Date of Feedback	DATETIME	N
	end_date	End Date of Feedback	DATETIME	N
	feedback_des	Description of Feedback	VARCHAR(255)	N
	feedback_name	Name of Feedback	VARCHAR(255)	N
	template_des	Description of Template	VARCHAR(255)	Y
	template_name	Name of Template	VARCHAR(255)	Y
	is_published	Feedback is published or not	BIT(1)	N

	semester_id	Id of the Semester that Feedback belongs to	INT(11)	N
Question	id	Unique identifier of the Question	INT(11)	N
	type	Type of Question	TEXT	N
	suggestion	Suggestion for this Question	TEXT	Y
	criteria_id	Id of the Criteria that Question belongs to	INT(11)	N
	feedback_id	Id of the Feedback that Question belongs to	INT(11)	N
	question_content	Content of Question	VARCHAR(255)	N
	is_required	Question is required or not	BIT(1)	N
Optionn	id	Unique identifier of the Option	INT(11)	N
	point	Point of Option	FLOAT	N
	question_id	Id of the Question that Option belongs to	INT(11)	N
	option_content	Content of Option	TEXT	N
Answer	id	Unique identifier of the Answer	INT(11)	N
	option_id	Id of the Option that Answer belongs to	INT(11)	N
	user_id	Id of the User replies that Answer	INT(11)	N
	answer_content	Content of Answer	TEXT	Y
	create_date	Created Date of Answer	DATETIME	N
Criteria	id	Unique identifier of the Criteria	INT(11)	N
	type_id	Id of the Type that Criteria belongs to	INT(11)	Y
	status	Status of Criteria	BIT(1)	N
	criteria	Name of Criteria	VARCHAR(255)	N
Semester	id	Unique identifier of the Semester	INT(11)	N
	start_date	Start Date of Semester	DATETIME	N
	end_date	End Date of Semester	DATETIME	N
	title	Name of Semester	VARCHAR(255)	N
User	id	Unique identifier of the User	INT(11)	N
	department_id	Id of the Department that User belongs to	INT(11)	Y
	role_id	Id of the Role that User belongs to	INT(11)	N
	major_id	Id of the Major that User belongs to	INT(11)	Y
	male	Gender of User	BIT(1)	N
	status	Status of User	BIT(1)	N
	username	Username of User	VARCHAR(255)	N
	password	Password of User	VARCHAR(255)	N
	fullname	Name of User	VARCHAR(255)	N
	code	Code of User	VARCHAR(255)	N
	mail	Email of User	VARCHAR(255)	N
User_Feedback	birth	Birthdate of User	DATE	N
	user_id	Unique identifier of the User	INT(11)	N
	feedback_id	Id of the Feedback that User has relation to	INT(11)	N
	is_conductor	User is conductor of this Feedback or not	BIT(1)	N
	is_viewer	User is Report viewer of this Feedback or not	BIT(1)	N

	is_conducted	This Feedback is conducted or not	BIT(1)	N
Type	id	Unique identifier of the Type	INT(11)	N
	description	Description of Type	VARCHAR(255)	N
Role	id	Unique identifier of the Role	INT(11)	N
	role_name	Name of Role	VARCHAR(255)	N
Privilege	id	Unique identifier of the Privilege	INT(11)	N
	name	Name of Privilege	VARCHAR(255)	N
Privilege_Role	role_id	Unique identifier of the Role	INT(11)	N
	privilege_id	Id of the Privilege that the Role can do	INT(11)	N
Student_Clazz	user_id	Unique identifier of the User	INT(11)	N
	clazz_id	Id of the Class that User belongs to	INT(11)	N
Clazz	id	Unique identifier of the Class	INT(11)	N
	semester_id	Id of the Semester that Class belongs to	INT(11)	N
	end_date	Start Date of Clazz	DATE	N
	start_date	End Date of Clazz	DATE	N
	class_name	Name of Class	VARCHAR(255)	N
	lecturer_id	Id of the User that teaches this Class	INT(11)	N
	course_id	Id of the Course that Class belongs to	INT(11)	N
Course	id	Unique identifier of the Course	INT(11)	N
	name	Name of Course	VARCHAR(255)	N
	code	Code of Course	VARCHAR(255)	N
	major_id	Id of the Major that Course belongs to	INT(11)	N
Major	id	Unique identifier of the Major	INT(11)	N
	name	Name of Major	VARCHAR(255)	N
	code	Code of Major	VARCHAR(255)	N
Department	id	Unique identifier of the Department	INT(11)	N
	name	Name of Department	VARCHAR(255)	N

Table 12 Attribute Data Dictionary

3. Test Plan

The overall purpose of testing is to ensure SFMS meets its entire technical, functional and business requirement. The purpose of this document is to describe the overall test plan and strategy for testing the SFMS. The approach described in this document provides the framework for all testing related to this application. Individual test cases will be written for each version of the application that is released. This document will also be updated as required for each release.

3.1. Features to be tested

- Student: Conduct Feedback
- Staff:
 - + Preview Template
 - + Select Feedback Timeframe
 - + Modify Suggested Improvement
 - + Modify Feedback Targets
 - + Modify Feedback Conductors
 - + View list Feedbacks

- Head of Academic: View list Reports
- System: Alert undone Feedback

3.2. Features not to be tested

- Guest: Login
- Student: Filter Feedback
- Lecturer:
 - + Compares Report by Semester
 - + View Report Details
 - + Filter Report
- Staff:
 - + Create blank new feedback
 - + Drag-n-drop feedback items to create content
 - + Edit feedback content
 - + Set name and description for feedback
 - + Set semester for feedback
 - + Choose scope (lecture, major, course, department)
 - + Create new feedback from template
 - + Save Feedback
 - + Manage Users
- Head of Academic: Manage Criterias
- System: Statistic and generate reports

4. System Testing Test Case

4.1. Communication Diagram

Test cases are created from below communication diagram

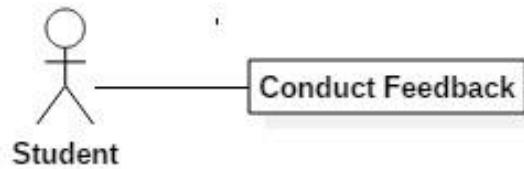


Figure 89 Student Communication Diagram

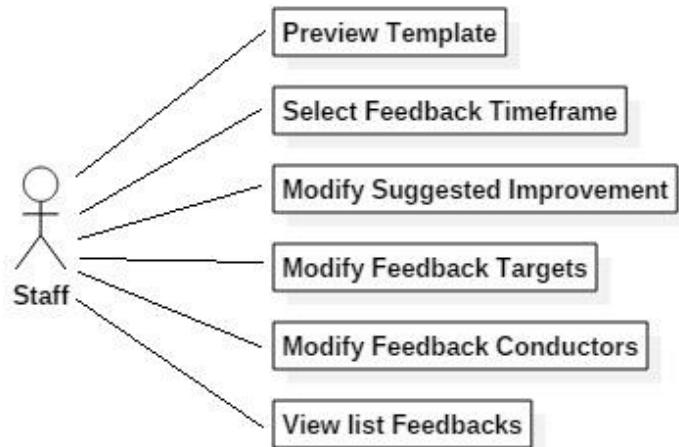


Figure 90 Staff Communication Diagram

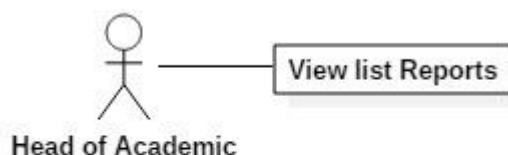


Figure 91 Head of Academic Communication Diagram

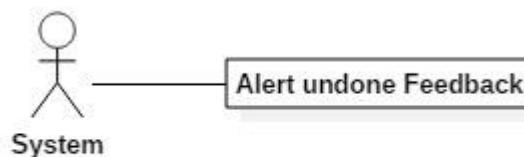


Figure 92 System Communication Diagram

4.2. Test Case

4.2.1. <Student> Conduct feedback

ID	Test Case Description	Precondition	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
Conduct feedback 1	Test student conduct feedback	There is at least 1 available Feedback in list	1.User login 2.User open notification panel 3.User select a feedback to conduct 4.User submit conduct feedback form after filled all required feedback	1. System alert user's answer saved 2. System show list feedbacks view	N/A	Pass	15/4/2018
Conduct feedback 2	Test student conduct feedback without input all required fields	There is at least 1 available Feedback in list	1.User login 2.User open notification panel 3.User select a feedback to conduct 4.User submit conduct feedback form without filled all required feedback	1.System display alert to prompt answer for each required question	N/A	Pass	1504/2018

4.2.2. <Staff> Preview template content

ID	Test Case Description	Precondition	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
Preview template content 1	Test staff previews template content	- There is at least 1 Template in list - Template has at least 1 question	1.User login as Staff 2.User clicks on "Quản lý feedback" menu. 3.User clicks on "Tạo feedback mới" option 4.User hovers on an icon of a template on screen 5.User click on button "Xem Thử"	1.System pop-up a modal on screen that show sample content of questions of template feedback	N/A	Pass	15/04/2018

Preview template content 2	Test staff previews template content with template has no questions	<ul style="list-style-type: none"> - There is at least 1 available Feedback in list - Template has no question 	<ol style="list-style-type: none"> 1.User login as Staff 2.User click on “Quản lý feedback” menu. 3.User click on “Tạo feedback mới” option 4.User hover on an icon of a template on screen 5.User click on button “xem thử” 	1.System pop-up a modal which has content is “Không có câu hỏi nào”	N/A	Pass	15/04/2018
-----------------------------------	---	--	---	---	-----	------	------------

4.2.3. <Staff> Select Feedback Timeframe

ID	Test Case Description	Precondition	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
Preview template content 1	Test staff select Start date of Feedback timeframe	N/A	<ol style="list-style-type: none"> 1.User login as Staff 2.User click on “Quản lý feedback” menu. 3.User creates new Feedback 4. User select dropdown for label “Học kỳ” then select an item of semester in dropdown list 5. User select input “Bắt đầu nhận phản hồi” field then select a date in allowed range on datetime picker shown up as feedback’s startdate 	<ol style="list-style-type: none"> 1.System change value of select “Bắt đầu nhận phản hồi” field to selected date 2.System close datetimepicker 3.System change min-value in allowed range for select “Kết thúc nhận phản hồi” field’s datetimepicker into selected startdate 	N/A	Pass	15/04/2018
Preview template content 2	Test staff select End date of Feedback timeframe	N/A	<ol style="list-style-type: none"> 1.User login as Staff 2.User click on “Quản lý feedback” menu. 3.User creates new Feedback 4. User select dropdown for label “Học kỳ” then select an item of semester in dropdown list 5. User select input field for 	<ol style="list-style-type: none"> 1.System change value of select “Kết thúc nhận phản hồi” field to selected date 2.System close datetimepicker 	N/A	Pass	15/04/2018

			label “Kết thúc nhận phản hồi” then select a date in allowed range on datetime picker shown up as feedback’s enddate				
Preview template content 3	Test staff Feedback timeframe that not match with select Semester	N/A	1.User login as Staff 2.User click on “Quản lý feedback” menu. 3.User creates new Feedback 4. User select dropdown for label “Học kỳ” then select an item of semester in dropdown list 5. User select Start date or End date out of allowed range on datetime picker shown up	1.System closes datetimepicker and changes nothing in shown up page	N/A	Pass	15/04/2018

4.2.4. <Staff> Modify Suggested Improvement

ID	Test Case Description	Precondition	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
Suggestion 1	Test staff Modify Suggested Improvement	N/A	1.User login as Staff 2.User clicks on “Quản lý feedback” menu. 3.User clicks on “Tạo feedback mới” option 4.User clicks on icon of “Tạo Feedback trống” 5.User click on link “Chỉnh sửa lời khuyên”	1.System shows screen of Modifying suggested improvement 2. Screen shows list of created Questions and textfield to input suggestion for each one	N/A	Pass	15/04/2018

Suggestion 2	Test staff Modify Suggested Improvement with creating from Template	<ul style="list-style-type: none"> - There is at least 1 template in list - There is at least 1 suggestion in the template 	<ol style="list-style-type: none"> 1.User login as Staff 2.User clicks on “Quản lý feedback” menu. 3.User clicks on “Tạo feedback mới” option 4.User clicks on button “Tiếp tục” of a template. 5.User click on link “Chỉnh sửa lời khuyên” 	<ol style="list-style-type: none"> 1.System shows screen of Modifying suggested improvement 2. Screen shows list of existed Questions, created suggestion that exists in template before, and textfield to re-input suggestion for each one 	N/A	Pass	15/04/2018
---------------------	---	--	--	---	-----	------	------------

4.2.5. <Staff> Modify Feedback Targets

ID	Test Case Description	Precondition	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
Target 1	Test staff modifies Feedback Targets	- There is at least 1 target in list	<ol style="list-style-type: none"> 1.User login as Staff 2.User clicks on “Quản lý feedback” menu. 3.User clicks on “Tạo feedback mới” option 4.User create new Feedback 5.User clicks on link “Thêm đối tượng cho Feedback” 	<ol style="list-style-type: none"> 1.System shows screen of Modifying Feedback Targets 2. Screen shows list of Target that matches with selected Type 	N/A	Pass	15/04/2018
Target 2	Test staff modifies Feedback Targets with having no Target in list	N/A	<ol style="list-style-type: none"> 1.User login as Staff 2.User clicks on “Quản lý feedback” menu. 3.User clicks on “Tạo feedback mới” option 4.User create new Feedback 5.User clicks on link “Thêm đối tượng cho Feedback” 	<ol style="list-style-type: none"> 1.System shows screen of Modifying Feedback Targets 2. Screen shows message “Không có đối tượng nào” 	N/A	Pass	15/04/2018

4.2.6. <Staff> Modify Feedback Conductors

ID	Test Case Description	Precondition	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
Target 1	Test staff modifies Feedback Conductors	- There is at least 1 Conductors in list	1.User login as Staff 2.User clicks on “Quản lý feedback” menu. 3.User clicks on “Tạo feedback mới” option 4.User create new Feedback 5.User adds target to list 6.User clicks on icon add more Conductors for a Target	1.System shows screen of Modifying Feedback Conductors 2.Screen shows list of Conductors that matches with selected Target.	N/A	Pass	15/04/2018
Target 2	Test staff modifies Feedback Conductors with having no Conductors in list	N/A	1.User login as Staff 2.User clicks on “Quản lý feedback” menu. 3.User clicks on “Tạo feedback mới” option 4.User create new Feedback 5.User adds target to list 6.User clicks on icon add more Conductors for a Target	1.System shows screen of Modifying Feedback Conductors 2.Screen shows message “Không có người thực hiện nào”	N/A	Pass	15/04/2018

4.2.7. <System> Alert undone Feedback

ID	Test Case Description	Precondition	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
Alert1	Test system alert undone feedback on panel	There is at least 1 new Feedback which user has to conduct	1.User login 2.User open notification panel	1.System shows undone feedback.	N/A	Pass	15/04/2018
Alert2	Test system alert undone feedback on Feedback List	There is at least 1 new Feedback which user has to conduct	1.User login 2.User clicks on "Quản lý Feedback" 3. User clicks on "Danh sách Feedback"	1.System shows Feedback along with button "Làm Feedback"	N/A	Pass	15/04/2018
Alert3	Test system alert undone feedback on panel when Feedback is expired	There is at least 1 new Feedback which user has to conduct	1.User login 2.User open notification panel	1.System will not show expired feedback	N/A	Pass	15/04/2018
Alert4	Test system alert undone feedback on Feedback list when Feedback is expired	There is at least 1 new Feedback which user has to conduct	1.User login 2.User clicks on "Quản lý Feedback" 3. User clicks on "Danh sách Feedback"	1.System shows Feedback along with text "Feedback đã hết hạn"	N/A	Pass	15/04/2018

4.2.8. <User> View list Feedbacks

ID	Test Case Description	Precondition	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
ViewListFeedback1	Test user view list feedback to conduct	N/A	1. User login 2. User click on “Quản lý feedback” menu. 3. User click on “Xem danh sách feedback sub-menu”	1. System shows list of feedback for logged in user. 2. System shows a hyperlink “Làm feedback” to answer	N/A	Pass	15/04/2018
ViewListFeedback2	Test user view list feedback with no result.	N/A	1. User login 2. User click on “Quản lý feedback” menu. 3. User click on “Xem danh sách feedback sub-menu”	1. System show a table with only 1 record whose content is “không có Feedback nào”	N/A	Pass	15/04/2018
ViewListFeedback3	Test user view list feedback with feedback at least conducted once and Feedback is still available	N/A	1. User login 2. User click on “Quản lý feedback” menu. 3. User click on “Xem danh sách feedback sub-menu”	1. System show list of feedback in a table. 2. System shows a hyperlink “Làm lại feedback” to re-answer	N/A	Pass	15/04/2018
ViewListFeedback4	Test user view list feedback when feedback is expired	N/A	1. User login 2. User click on “Quản lý feedback” menu. 3. User click on “Xem danh sách feedback sub-menu”	1. System show list of feedback in a table. 2. System shows a tesxt say: “Feedback đã quá hạn hoặc chưa bắt đầu”.	N/A	Pass	15/04/2018

ViewListFeedback5	Test user view list feedback with feedback at least conducted once but feedback is expired	N/A	1.User login 2.User click on “Quản lý feedback” menu. 3.User click on “Xem danh sách feedback sub-menu”	1. System show list of feedback in a table. 2. System shows a text say: “Đã hoàn thành”.	N/A	Pass	15/04/2018
--------------------------	--	-----	---	---	-----	------	------------

4.2.9. <Head of Academic> View list Reports

ID	Test Case Description	Precondition	Test case procedure	Expected output	Inter-test case dependence	Result	Test Date
ViewListReport1	Test Head of Academic when views list report.	N/A	1.User login as Head of Academic roles. 2.User click on “Xem báo cáo” menu	1.System show these to user screen: -Tables corresponding to 4 types of feedback. -Filter to show which type of feedback should appear on the screen. -Option to see to which reports.	N/A	Pass	15/04/2018

F. Software User's Manual

1. Installation Guide

1.1. Setting up environment at server side

The following software must be installed into the server machine:

1.1.1. Hardware requirements

Windows	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (4 Mbps)	Cable, Wi-Fi (8 Mbps)
Operating System	Window Server 2008	Window Server 2008
Computer Processor	Intel® Xeon ® 1.4GHz	Intel® Xeon ® Quad Core (12M Cache, 2.50 GHz)
Computer Memory	1GB RAM	2GB or more

Table 13 Hardware Requirement

1.1.2. Software requirements

Software	Name / Version	Description
Operating system	Window 7 or above	Operating system and platform for development
Environment	Java EE 8, Spring, Gradle	Specification for developing web application
IDE	IntelliJ IDEA 17.2, Android Studio 3.0	Programming tools
DBMS	MySQL 5.7	Used to create & manage the database for system
Web browser	Chrome 42 or above	Testing browser

Table 14 Software Requirement

1.2. Deployment at serverside

1.2.1. Prepare deployment package

- Install JDK 8.0
- Setup Tomcat
- Install IntelliJ IDEA 17.2, Spring and Gradle
- Install Android Studio 3.0
- Install MySQL 5.7 and restore database "Capstone"

1.2.2. Configure Server before deploy

- Open file application.properties in folder SFMS\src\main\resources
 - + Edit your port at property "server.port"
 - + Edit your username to MySQL "spring.datasource.username"
 - + Edit your password to MySQL "spring.datasource.password"
 - + Edit your username to Email "spring.mail.username"
 - + Edit your password to Email "spring.mail.password"

1.2.3. Deploy web application on server

- Run app
- Open web browser, input url to app

1.3. Setting up the environment at client side

1.3.1. Setting up for computer

Windows	Minimum Requirements	Recommended
Internet Connection	Cable, Wi-Fi (2 Mbps)	Cable, Wi-Fi (4 Mbps)
Operating System	Windows 7 or above	Window 7 or above
Computer Processor	Intel® Pentium 4 1.60Ghz	Intel® Pentium 4 2.00Ghz
Computer Memory	1GB RAM	2GB or more

Table 15 Hardware Requirement for PC

Windows	Minimum Requirements	Recommended
Internet Connection	Wi-Fi (2 Mbps)	Wi-Fi (4 Mbps)
Operating System	Android 4.4.2 or above	Android 6.0 or above
Memory	1GB RAM	2GB or more

Table 16 Hardware Requirement for Mobile

2. User Guide

2.1. Web application for all users

2.1.1. Login

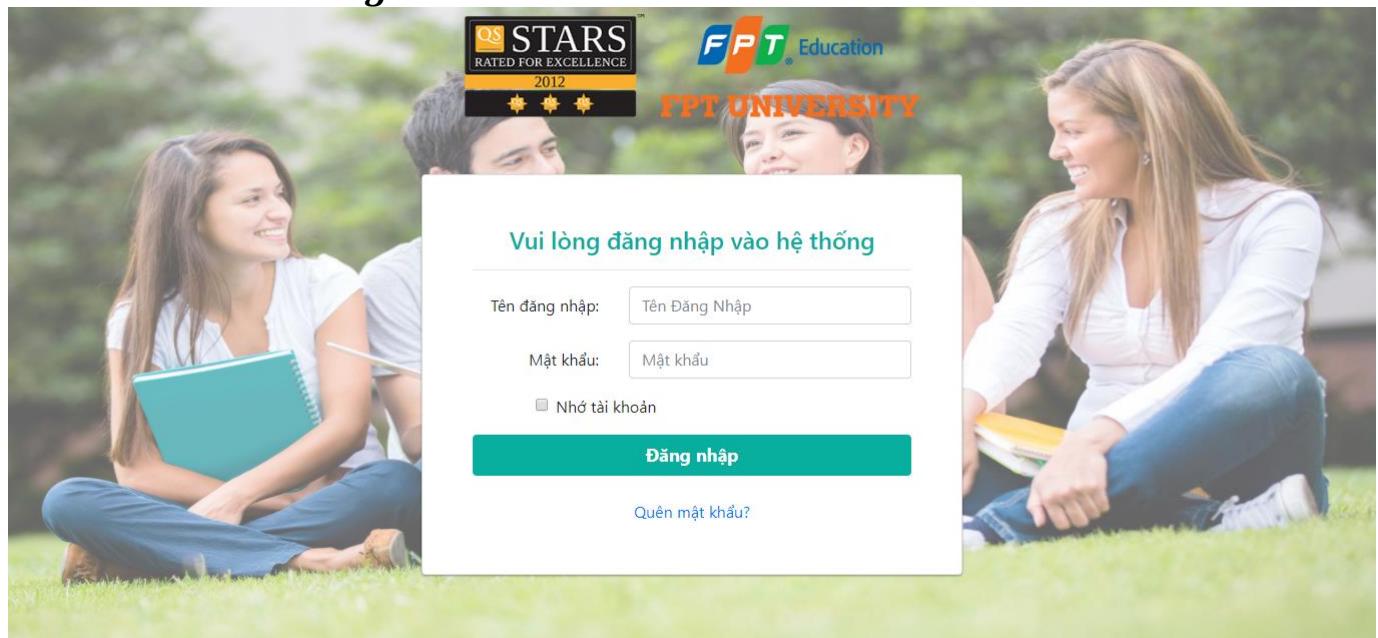


Figure 93 <Guest> Login page

Step	Description
1	Enter username, password.
2	Tap “Đăng nhập” button.

Table 17 <Guest> Login

2.1.2. Feedbacks need to be conducted list

The screenshot shows the SFMS application's interface. On the left is a dark sidebar with a user profile picture of 'Hoàng Quốc Việt' (Student), a 'Dashboard' link, and a 'Quản lý feedback' link. The main content area has a teal header 'Danh sách Feedbacks'. Below it are three filter dropdowns: 'Trạng thái' (All), 'Loại Feedback' (Show all), and 'Sắp xếp' (Sort A-Z). A table lists one feedback entry: 'Feedback cho phòng IT' (Category: Spring 2017, Start Date: 10-04-2018, End Date: 22-04-2018, Status: Phòng ban, Action: 'Làm feedback' with a pen icon). The bottom right of the main area says 'Copyright © Your Website 2017'.

Figure 94 <All users> Feedback List

Step	Description
1	Tap on “Quản lý feedback”
2	Tap on “Xem danh sách feedback”

Table 18 <All user> Feedback List

2.1.3. Conduct Feedback

The screenshot shows a modal dialog box titled 'Xác nhận gửi kết quả' (Confirm sending results). It contains the question 'Bạn có chắc muốn gửi kết quả này?' (Are you sure you want to send this result?). There are two buttons: 'Gửi' (Send) in green and 'Đóng' (Close) in grey. Below the dialog, there are navigation buttons 'Quay lại' (Back) and 'Tiếp tục' (Continue). At the bottom, there is a progress bar at 100% and a large green 'Gửi kết quả' (Send result) button.

Figure 95 <All users> Conduct feedback

Step	Description
1	Answer all required question.
2	Tap on “Quay lại” to return to previous page and “Tiếp tục” to continue to next page.
3	Tap on “Gửi kết quả” to send your answers.
4	Tap on “Gửi” again to cō

Table 19 <All user> Conduct feedback

2.1.4. Notify user

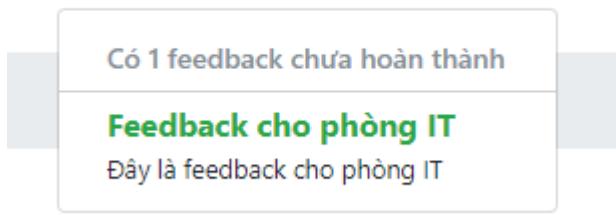
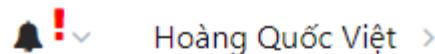


Figure 96 <All users> Notify user

Step	Description
1	Click on bell notification

Table 20 <All user> Notify user

2.2. Web application for staffs

2.2.1. Initialize new feedback

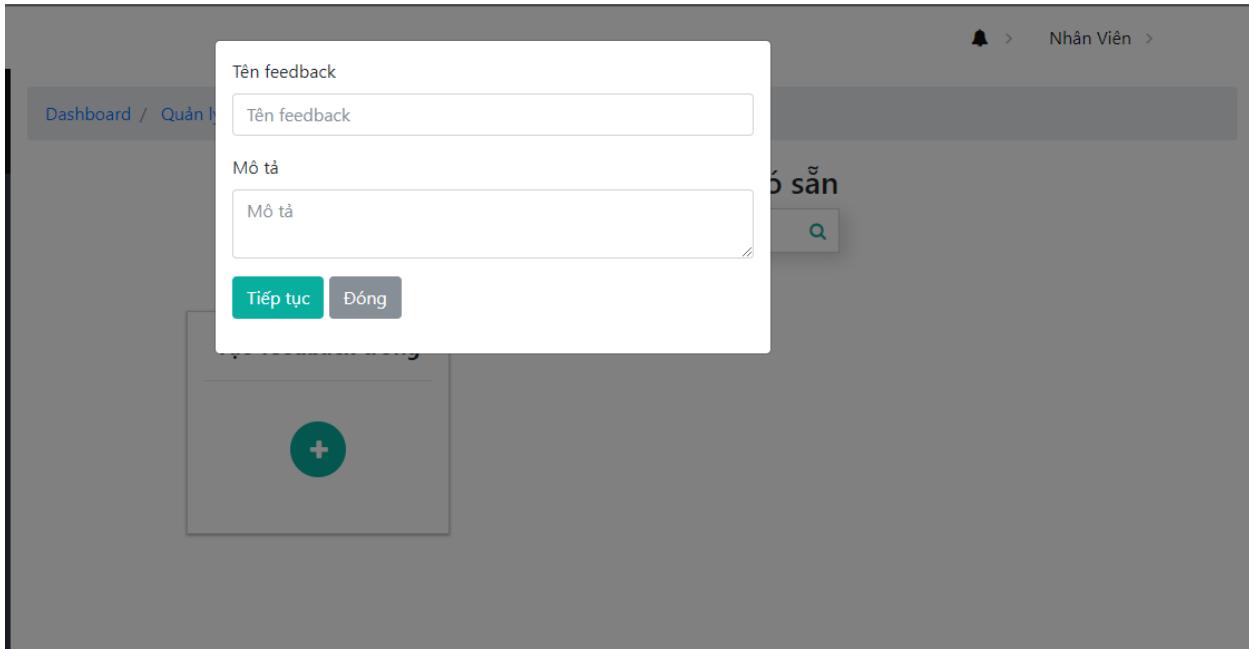


Figure 97 <Staff> Initialize new feedback

Step	Description
1	Click on any template or click on “Tạo feedback trống”.
2	Fill in name and description for this feedback.
3	Tap on “Tiếp tục”

Table 21 <Staff> Initialize new feedback

2.2.2. Create feedback content

SFMS

admin
Trưởng ban

Dashboard
Quản lý feedback
Xem báo cáo
Quản lý tiêu chí đánh giá
Quản lý tài khoản

Tạo nội dung cho feedback

Xóa

Nội dung của câu hỏi

Câu hỏi bắt buộc:

Nội dung: Nội dung của câu hỏi?

Loại đánh giá: Chuyên cần

Nhóm radio
Nhóm checkbox
Trường nhập văn bản(đơn dòng)
Trường nhập văn bản(đa dòng)

Lưu nội dung

Figure 98 <Staff> Create feedback content

Step	Description
1	Drag a question type from right menu to the greyscaled area.
2	Fill in question content.
3	Click on “Lưu nội dung” to save content for this feedback.

Table 22 <Staff> Create feedback content

2.2.3. Modifying feedback information

Vui lòng kiểm tra & hoàn tất những thông tin sau

Tiêu đề	Feedback cho phòng IT
Mô tả	Đây là feedback cho phòng IT
Học kỳ	Spring 2017
Bắt đầu nhận phản hồi:	dd----yy
Kết thúc nhận phản hồi	dd----yy
Đối tượng được đánh giá	Lớp
10 kết quả mỗi trang	
Tìm: _____	
Môn học ↑↓ Mã môn ↑↓ Học kỳ ↑↓ Lớp ↑↓ Giảng viên ↑↓ Tùy chọn ↑↓	

Xem nội dung feedback
Xem đề xuất cải tiến

Bắt đầu nhận phản hồi: 10-Apr-20

Kết thúc nhận phản hồi: 22-Apr-20

Đối tượng được đánh giá: Phòng ban

10 kết quả mỗi trang

Tìm: _____

Phòng ban	Tùy chọn
Không kết quả nào được tìm thấy. Thêm đối tượng	
Tên	Tùy chọn

Hiển thị 0 tới 0 trong số 0 kết quả

Trước Sau

Lưu feedback thành:

- Lưu & tiến hành feedback
- Mẫu mới
- Cập nhật mẫu đã chọn

[Lưu](#) [Hủy](#)

Figure 99 <Staff> Modifying Feedback Information

Step	Description
1	Select from date and end date for this feedback.
2	Select which this feedback is to.
3	Select who should conduct this feedback.
3	-Select “Lưu và tiến hành feedback” to save this feedback. -Select “Mẫu mới” to save this as new template. -Select “Cập nhật mẫu đã chọn” to update this template.
4	Click on “Lưu” to complete modifying feedback/template content.

Table 23 <Staff> Modifying Feedback Information

2.2.4. View list self's report for teacher and staff

Các báo cáo của tôi

Phòng ban của bạn:

Tuyển Sinh

[Xem báo cáo chi tiết ➔](#)

Các báo cáo của tôi

Hiển thị kết quả mỗi trang

Tìm:

#	Mã môn	Tên môn	Lớp	
1	PRO201	Front-end web development	SE1061	Xem báo cáo
2	PRN292	C# & Dot Net	SE1062	Xem báo cáo
3	PRF192	Programming Fundamentals Using C	SE1063	Xem báo cáo
4	PRO001	Alice	SE1062	Xem báo cáo
5	CSI101	Introduction to Computing	SE1164	Xem báo cáo
#	Mã môn	Tên môn	Lớp	

Hiển thị từ 1 tới 5 trong số 7 kết quả

Trước 1 2 Sau

Figure 100 <Staff, Teacher> View list self's report

Step	Description
1	Select a subject, department to see your report.

Table 24 <Staff, Lecturer> View list self's report

2.2.5. View all report for Head of Academic

Loại Feedback	Phòng ban	Sắp xếp
Tên phòng		
Công tác Sinh viên		Xem báo cáo
Đào Tạo		Xem báo cáo
IT		Xem báo cáo
Khảo Thí		Xem báo cáo
PDP		Xem báo cáo
Thư Viện		Xem báo cáo
Tuyển Sinh		Xem báo cáo
Y tế		Xem báo cáo

Figure 101 <Head of Academic> View all report

Step	Description
1	Select a feedback type to filter table.
2	Select sort order.
3	Click on “Xem báo cáo” to see report detail.

Table 25 <Head of Academic > View all report

2.2.6. View report details

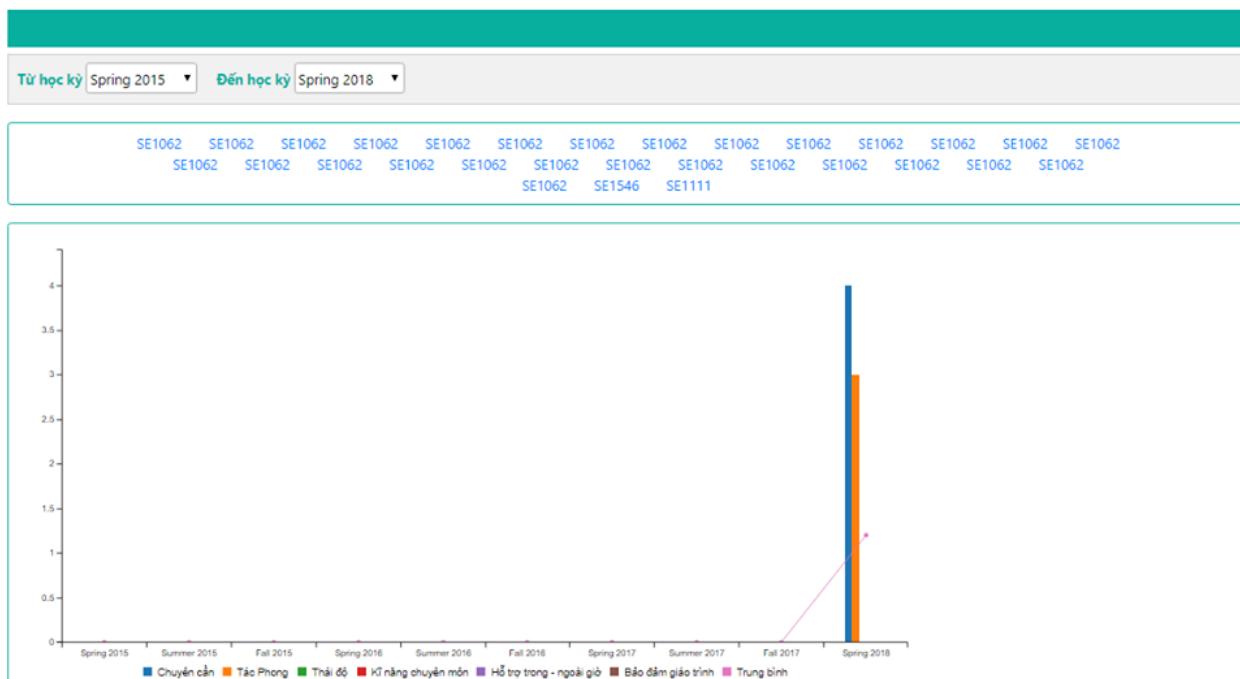


Figure 102 <Head of Academic, Staff, Lecturer> View report details

Step	Description
1	Select from semester and to semester to filter which period of time you want to see.
2	Click on bar chart to show which classes are in this semester.
3	Click on class's name to see detail for this semester.

Table 26 <Head of Academic, Staff, Lecturer> View report details

2.2.7. View semester report details

Điểm trung bình: 1.8/5

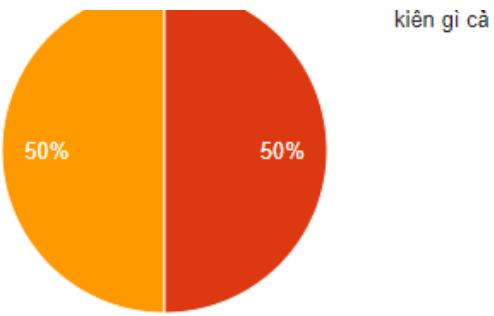
Chuyên cần-3.5/5

Điểm trung bình của câu hỏi: 3.0

Lời khuyên:

Phòng IT có đáp ứng được các nhu cầu của bạn không?
1 phản hồi

● Vừa đủ



kiên gì cà

Ngày nhập	Nội dung
-----------	----------

Xem các ý kiến Khác

Figure 103 <Head of Academic, Staff, Lecturer> View semester report details

Step	Description
1	Click on “Xem thêm các ý kiến khác” or “Xem nội dung câu trả lời” to see users' inputted answer.

Table 27 <Head of Academic, Staff, Lecturer> View semester report

2.2.8. View list users

+ Thêm người dùng						
Tất cả phân quyền		Tất cả phòng		Tất cả chuyên ngành		Tất cả
Họ & Tên		Phân quyền	Phòng ban	Chuyên ngành	Email	Trạng thái
admin	Trưởng ban	Đào Tạo	N/A	Kỹ thuật phần mềm	viethoangquoc2711@gmail.com	Đang hoạt động
Hoàng Quốc Việt	Giảng viên	Đào Tạo	N/A	Kỹ thuật phần mềm	viethqse61745@fpt.edu.vn	Đang hoạt động
Hoàng Quốc Việt	Sinh viên	N/A	N/A	Kỹ thuật phần mềm	viethoangquoc2711@gmail.com	Đang hoạt động
Nguyễn Huy Hiệu	Giảng viên	Đào Tạo	N/A	Kỹ thuật phần mềm	viethqse61745@fpt.edu.vn	Đang hoạt động

Figure 104 <Head of Academic, Staff> View list users

Step	Description
1	Select from a filter to filter table.
2	Click on “Thêm người dùng” to add new user.
3	Click on “Chỉnh sửa” to see user’s detail.

Table 28 <Head of Academic, Staff> View list users

2.2.9. Create new user

The screenshot shows a form titled "Vui lòng kiểm tra & hoàn tất những thông tin sau" (Please check and complete the following information). The form fields include:

- Tên đăng nhập: [Text input]
- Mật khẩu mới: [Text input]
- Nhập lại mật khẩu: [Text input]
- Họ & tên: [Text input]
- Ngày sinh: [Text input] (dd----yyyy)
- Email: [Text input] (Ex: example@example.com)
- Giới tính: [Radio buttons] (Nam or Nữ selected)
- Phân quyền: [Select dropdown] (Giảng viên)
- Chuyên ngành: [Select dropdown] (N/A)
- Phòng ban: [Select dropdown] (Đào Tạo)
- Đang hoạt động: [Check box] (unchecked)
- Mã code: [Text input] (Mã của người dùng này)

At the bottom are three buttons: "Lưu và tiếp tục" (Save and continue), "Lưu và quay về danh sách" (Save and return to list), and "Hủy" (Cancel).

Figure 105 <Head of Academic, Staff> Create new user

Step	Description
1	Fill in all field.
2	Click “Lưu và tiếp tục” to save and add another user. Click “Lưu và quay về danh sách” to save and return to users list. Click “Hủy” to reload page.

Table 29 <Head of Academic, Staff> Create new user

2.2.10. Unpublish Feedback

Danh sách Feedbacks

Lọc loại feedback ▾

Thêm feedback +

Hiển thị 5 kết quả mỗi trang

Tìm:

Tên feedback ↑↓	Kì báo cáo ↑↓	Từ-đến ↑↓	Loại ↑↓	Đối tượng được báo cáo ↑↓	Tình trạng ↑↓	
Feedback cho phòng IT	Spring 2017	4/10/2018 - 4/10/2018	Phòng ban	IT	Đã xuất bản	Ngưng
Tên feedback	Kì báo cáo	Từ-đến	Loại	Đối tượng được báo cáo	Tình trạng	

Hiển thị từ 1 tới 1 trong số 1 kết quả

Trước 1 Sau

Xác nhận lại việc ngưng xuất bản

Dashboard / Quản lý

Danh sách Fe

Lọc loại feedback ▾

Ngưng xuất bản

Close

Thêm feedback +

Hiển thị 5 kết quả mỗi trang

Tìm:

Tên feedback ↑↓	Kì báo cáo ↑↓	Từ-đến ↑↓	Loại ↑↓	Đối tượng được báo cáo ↑↓	Tình trạng ↑↓	
Feedback cho phòng IT	Spring 2017	4/10/2018 - 4/10/2018	Phòng ban	IT	Đã xuất bản	Ngưng
Tên feedback	Kì báo cáo	Từ-đến	Loại	Đối tượng được báo cáo	Tình trạng	

Hiển thị từ 1 tới 1 trong số 1 kết quả

Trước 1 Sau

Figure 106 <Head of Academic, Staff> Unpublish Feedback

Step	Description
1	Click “Ngưng” button
2	Click “Ngưng xuất bản” to unpublish Feedback. Click “Close” to reload page.

Table 30 <Head of Academic, Staff> Create new user

2.3. Web application for Head of Academic

2.3.1. View all report for Head of Academic

Loại Feedback	Phòng ban	Sắp xếp	A-Z
Tên phòng			
Công tác Sinh viên			Xem báo cáo
Đào Tạo			Xem báo cáo
IT			Xem báo cáo
Khảo Thí			Xem báo cáo
PDP			Xem báo cáo
Thư Viện			Xem báo cáo
Tuyển Sinh			Xem báo cáo
Y tế			Xem báo cáo

Figure 107 <Head of Academic > View all reports

Step	Description
1	Select a feedback type to filter table.
2	Select sort order.
3	Click on "Xem báo cáo" to see report detail.

Table 31 <Head of Academic > View all reports

2.3.2. Manage criteria

Dashboard / Quản lý tiêu chí đánh giá			
Quản lý tiêu chí đánh giá			
10	kết quả mỗi trang		Tìm:
Tiêu chí	Trạng thái	Tùy chọn	
ahihidd	<input checked="" type="checkbox"/>	+ edit	
Bảo đảm giáo trình	<input checked="" type="checkbox"/>	+ edit	
Chuyên cần	<input checked="" type="checkbox"/>	+ edit	
Cơ sở hạ tầng kỹ thuật	<input checked="" type="checkbox"/>	+ edit	
Cơ sở vật chất	<input checked="" type="checkbox"/>	+ edit	
Giáo trình	<input checked="" type="checkbox"/>	+ edit	
Giải đáp thắc mắc	<input checked="" type="checkbox"/>	+ edit	

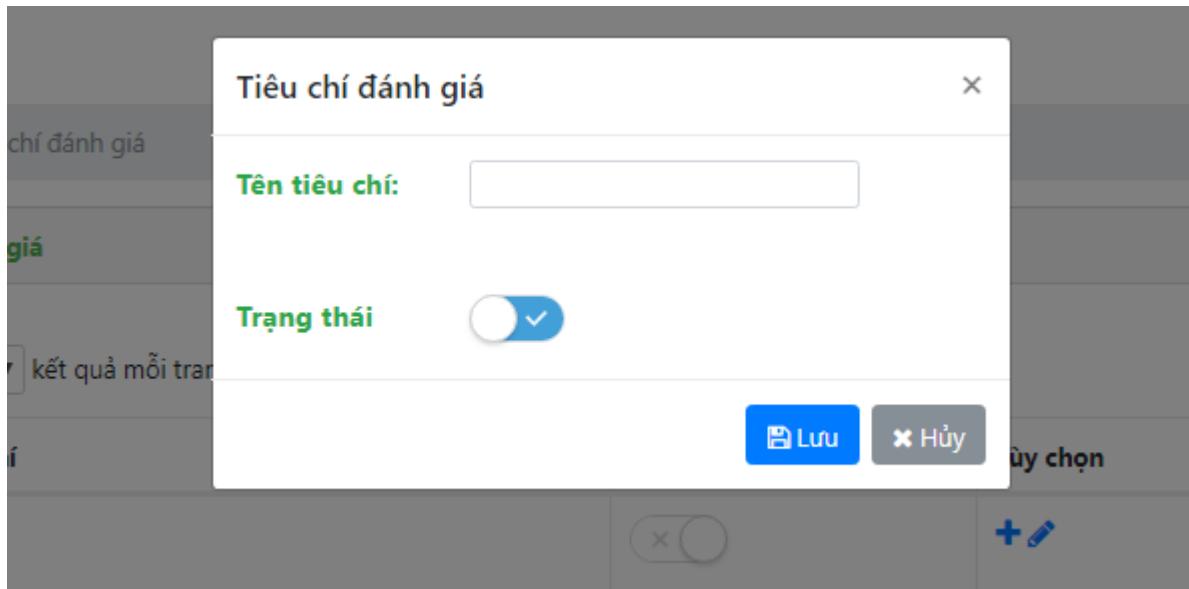


Figure 108 <Head of Academic > Manage criteria

Step	Description
1	Click on “+” button to add new criteria.
2	Fill in name, target and status for criteria.
3	Click on “Lưu” to save criteria.

Table 32 <Head of Academic > Manage criteria

2.4. Mobile application for all users

2.4.1. Login

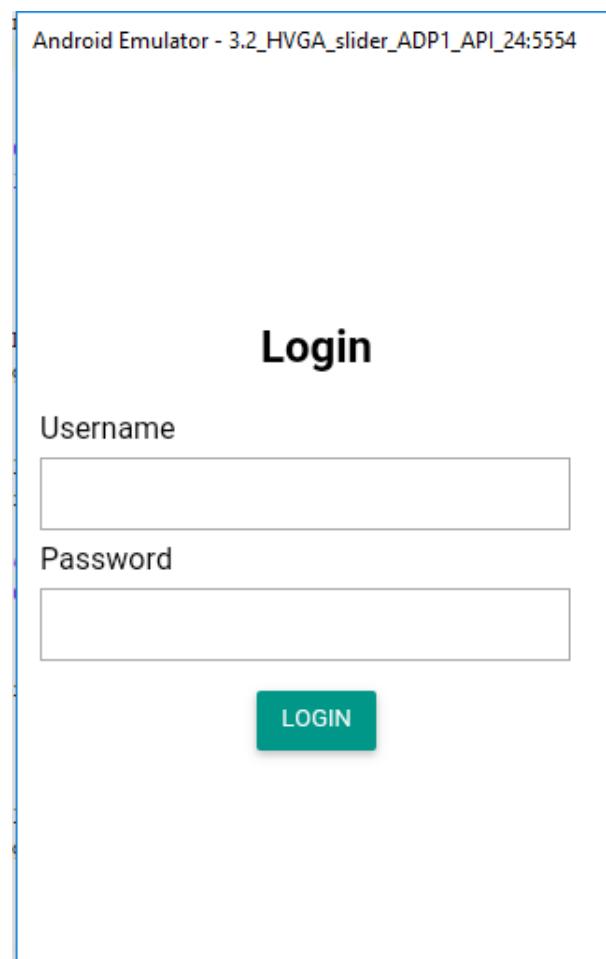


Figure 109 <Guest> Login

Step	Description
1	Fill in username and password.
2	Click on “Đăng nhập” to login.

Table 33 <Guest> Login

2.4.2. View list Feedbacks

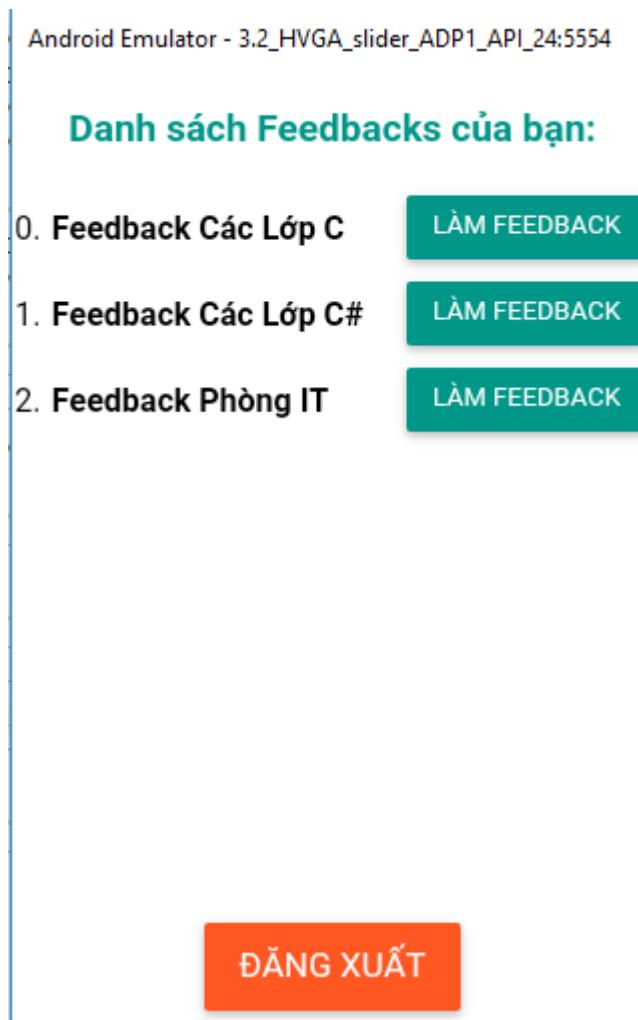


Figure 110 <All users> View list feedbacks

Step	Description
1	Click on “Làm Feedback” to conduct this feedback.

Table 34 <All users> View list feedbacks

2.4.3. Conduct Feedback on mobile

Android Emulator - 3.2_HVGA_slider_ADPI_API_24:5554

Feedback

Câu hỏi 1

Tốc độ giảng dạy của giáo viên thế nào?

- Nhanh
- Vừa
- Chậm
- Khác

Nhập câu trả lời của bạn vào đây

GỬI KẾT QUẢ

Android Emulator - 3.2_HVGA_slider_ADPI_API_24:5554

Câu hỏi 2

Tốc độ ra sao?

Có

Câu hỏi 3

Nội dung của câu hỏi:

25%

GỬI KẾT QUẢ

Figure 111 <All users> Conduct feedback on mobile

Step	Description
1	Answer all required question.
2	Click on “Gửi kết quả” to submit your answers.

Table 35 <All users> Conduct feedback on mobile