



MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

Examination Tools applying Block Chain Technology

Group 4	
Group members	Nguyễn Minh Hưng – SE61794 Cao Trung Hiếu – SE61801 Trương Tân Sang – SE61926
Supervisor	Kiều Trọng Khánh
Ext. Supervisor	N/A
Capstone Project code	ExamTool

- Ho Chi Minh City, 11 September 2018 -

This page is intentionally left blank.

Table of Contents

Table of Contents.....	3
List Of Tables	10
List of Figures	14
Definitions, Acronyms, and Abbreviations	16
A. Introduction.....	17
1. Project Information	17
2. Introduction.....	17
3. Current Situation.....	17
4. Problem Definition	18
4.1. Proposed Solution	18
4.2. Feature Functions.....	18
5. Values and Challenges Value	18
6. Functional Requirement.....	18
7. Role and Responsibility.....	19
B. Software Project Management Plan.....	20
1. Problem Definition.....	20
1.1. Name of this Capstone Project.....	20
1.2. Problem Abstract	20
1.3. Project Overview	20
1.3.1 Current Situation.....	20
1.3.2 Boundaries of the system.....	20
1.3.3 Future Plans	21
1.3.4 Development Environment.....	21
1.3.4.1 Hardware requirements.....	21
1.3.4.2 Software requirements	21
2. Coding Convention	21
3. Project Organization	22
3.1. Software Process Model	22
3.2. Roles and Responsibilities	23
3.3. Tools and Techniques	24

4.	Project Management Plan	24
4.1.	Product Backlog.....	24
4.2.	Sprint Backlog	25
4.2.1	Sprint 1 (10.09.2018 – 23.09.2018): Project initiation	25
4.2.1.1	Goal.....	25
4.2.1.2	Development.....	26
4.2.2	Sprint 2 (24.09.2018 – 8.10.2018): Software Document.....	26
4.2.2.1	Goal.....	26
4.2.2.2	Development.....	26
4.2.3	Sprint 3 (9.10.2018 – 23.10.2018): Basic feature and core	27
4.2.3.1	Goal.....	27
4.2.3.2	Development.....	27
4.2.4	Sprint 4 (24.10.2018 – 8.11.2018): Software Document.....	27
4.2.4.1	Goal.....	27
4.2.4.2	Development.....	27
4.2.5	Sprint 5 (9.11.2018 – 23.11.2018): Software Document.....	28
4.2.5.1	Goal.....	28
4.2.5.2	Development.....	28
4.2.6	Sprint 6 (24.11.2018 – 13.12.2018): Complete Document.....	28
4.2.6.1	Goal.....	28
4.2.6.2	Development.....	28
4.3.	All Meeting minutes	29
C.	Software Requirement Specification	29
1.	User Requirement Specification	29
1.1.	Examination Requirement	29
1.2.	System Requirement	29
2.	System Requirement Specification	29
2.1.	External Interface Requirement	29
2.1.1	User Interface.....	29
2.1.2	Hardware Interface	29
2.1.3	Software Interface.....	29
2.1.4	Communication Protocol	29
2.2.	System Overview Diagram.....	30

2.3.	List of use case.....	34
2.3.1	<IT<IT> <i>Get list of killed processes.</i>	34
2.3.2	<IT> Add process to list of killed processes.....	35
2.3.3	<IT> Add process to list of exception processes	36
2.3.4	<IT> Kill Process.....	39
2.3.5	<Leader> Add chapter	40
2.3.6	<Leader> Delete chapter.....	42
2.3.7	<Leader> Edit chapter	44
2.3.8	<Teacher> Search Question.....	45
2.3.9	<Teacher> Add Question by Form	47
2.3.10	<Teacher> Add Question by Import File.....	50
2.3.11	<Teacher> Remove Question	51
2.3.12	<Teacher> Edit Questions	53
2.3.13	<Teacher> Approve Test Exam.....	55
3.	Software System Attribute.....	57
3.1.	Usability.....	57
3.2.	Reliability.....	58
3.3.	Availability	58
3.4.	Security	58
3.5.	Maintainability.....	58
3.6.	Portability.....	58
4.	Conceptual Diagram	59
D.	Software Design Description	61
1.	Design Overview	61
2.	System Architecture Design	61
2.1.	Web Server Architecture Design	62
2.2.	Client Architecture Design	62
3.	Component Diagram.....	64
4.	Detailed Description	65
4.1.	Class Diagram.....	65
4.2.	Class Diagram Explanation	68
4.2.1	Semester.....	68
4.2.2	Exam	68

4.2.3	ExamStudent.....	69
4.2.4	Student	69
4.2.5	BaseTest.....	70
4.2.6	BaseTestContent	71
4.2.7	Test.....	71
4.2.8	TestContent.....	72
4.2.9	Question	73
4.2.10	Option	73
4.2.11	Level	74
4.2.12	QuestionType.....	74
4.2.13	QuestionGroup.....	75
4.2.14	QuestionObjective	75
4.2.15	Chapter.....	76
4.2.16	Course	76
4.2.17	MainObjective	77
4.2.18	LearningOutcome	77
4.2.19	TeacherTest.....	78
4.2.20	TeacherTestContent	78
4.3.	Interaction Diagram	79
4.3.1	Activity Diagram	79
4.3.1.1	Teacher Import Question	79
4.3.1.2	Generate Test Exam.....	81
4.3.1.3	Teacher Approve Test Exam	81
4.3.1.4	Student Take a Test Exam	82
5.	Interface	83
5.1.	Manage Question Bank.....	84
5.2.	Import question	85
5.3.	Matching question.....	86
5.4.	Create Test Exam.....	87
5.5.	Test Exam List with “Pending” Status	91
5.6.	Take an Exam	92
5.7.	Create Exam.....	94
5.8.	Set Approver	95

6.	Database Design	97
6.1.	Entity Relationship Diagram	97
6.2.	Data Dictionary.....	98
6.3.	Class Diagram Explanation	98
6.3.1	Semester.....	98
6.3.2	Exam	99
6.3.3	Student	99
6.3.4	BaseTest.....	99
6.3.5	BaseTestContent	100
6.3.6	Test.....	100
6.3.7	TestContent.....	101
6.3.8	Question	101
6.3.9	Option	101
6.3.10	Level	101
6.3.11	QuestionType.....	102
6.3.12	QuestionGroup.....	102
6.3.13	QuestionObjective	102
6.3.14	Chapter.....	102
6.3.15	Course	103
6.3.16	MainObjective	103
6.3.17	LearningOutcome	103
6.3.18	Course Department	103
6.3.19	TeacherTest.....	103
6.3.20	TeacherTestContent	104
7.	Algorithms	105
7.1.	Calculating Percentage Similarity of 2 Question.....	105
7.1.1	Problem.....	105
7.1.2	Solution.....	105
7.1.3	Complexity.....	107
E.	System Implementation & Test	107
1.	Introduction.....	107
1.1.	Overview.....	107
1.2.	Test Approach.....	107

2.	Database Relationship Diagram.....	108
2.1.	Physical Diagram:.....	108
2.2.	Data Dictionary:.....	111
2.2.1	Semester.....	111
2.2.2	Exam	111
2.2.3	Student	111
2.2.4	BaseTest.....	111
2.2.5	BaseTestContent	112
2.2.6	Test.....	112
2.2.7	TestContent.....	113
2.2.8	Question	113
2.2.9	Option	114
2.2.10	Level	114
2.2.11	QuestionType.....	114
2.2.12	QuestionGroup.....	115
2.2.13	QuestionObjective	115
2.2.14	Chapter.....	115
2.2.15	Course	115
2.2.16	MainObjective	116
2.2.17	LearningOutcome	116
2.2.18	Course Department	116
2.2.19	TeacherTest.....	116
2.2.20	TeacherTestContent	117
2.2.21	AuthenticationApp.....	117
2.2.22	Import Result	118
2.2.23	DefaultProcess	118
3.	Performance Measures.....	118
3.1.	Parsing Question	118
3.2.	Validate Parsed Questions	118
3.3.	Matching Questions of a Course.....	118
3.4.	Random Question to Create Exam Base on Chapters/ Learning Outcome ...	118
3.5.	Get Exam Test from Server	119
3.6.	Submit Exam Tests of a Class	119

4.	Test Plan	119
4.1.	Features to be tested.....	119
4.2.	Features not to be tested.....	119
5.	System Testing Test Case	119
5.1.	Manage Exam System Test Case.....	119
5.1.1	Import Question	119
6.	System Implement	125
6.1.	Web Server Application.....	125
6.2.	Web Client Application	126
F.	Software User's Manual	127
1.	Installing Guide.....	127
2.	User Guide	130
2.1.	Admin Create Semester	130
2.2.	Teacher approve/reject exam	133
2.3.	Staff Create Exam	135
G.	Appendix.....	140

List Of Tables

Table 1: Definitions, Acronyms, and Abbreviations	16
Table 2: Role and Responsibilities	19
Table 3: Hardware Requirements.....	21
Table 4 Software Requirement.....	21
Table 5 Roles and Responsibilities Detail	23
Table 6 Tools and Techniques.....	24
Table 7 Product Backlog.....	25
Table 8 Sprint 1 Development	26
Table 9 SPRINT 2 DEVELOPMENT	27
Table 10 SPRINT 3 DEVELOPMENT	27
Table 11 SPRINT 4 DEVELOPMENT	28
Table 12 SPRINT 5 DEVELOPMENT	28
Table 13 SPRINT 6 DEVELOPMENT	28
Table 14 <IT> Get List Process Must Kill	35
Table 15 Add Must Be Killed Process to List	36
Table 16 Add process to list of exception processes	38
Table 17 Kill Process.....	40
Table 25 Add chapter.....	42
Table 26 Delete chapter.....	43
Table 27 <Leader> Edit chapter	45
Table 29 <Teacher> Search Question	47
Table 30 <TEACHER> ADD QUESTION BY FORM	49
Table 31 <Teacher> Add Question by Import File	51
Table 32 <Teacher> Remove Questions	53
Table 33 <Teacher> Edit Questions	55
Table 39 CONCEPTUAL DIAGRAM EXPLAINATION	60
Table 40 ARCHITECTURE OVERVIEW DESCRIPTION	62
Table 41 Component diagram dictionary	65
Table 42 Class diagram Description	68
Table 43 Class Semester Explanation	68
Table 44 Class Semester method explanation.....	68
Table 45 Class exam Explanation.....	69
Table 46 Class exam method explanation	69
Table 47 Class ExamStudent Explanation	69
Table 48 Class ExamStudent method explanation	69
Table 49 CLASS STUDENT EXPLANATION.....	70
Table 50 Class student method explanation	70
Table 51 CLASS BASETEST EXPLANATION	70
Table 52 Class basetest method explanation	71
Table 53 CLASS BASETESTCONTENT EXPLANATION	71
Table 54 Class basetestcontent method explanation	71
Table 55 CLASS TEST EXPLANATION	72
Table 56 Class test method explanation.....	72

Table 57 CLASS TESTCONTENT EXPLANATION	72
Table 58 Class testcontent method explanation	73
Table 59 CLASS QUESTION EXPLANATION.....	73
Table 60 Class question method explanation.....	73
Table 61 CLASS OPTION EXPLANATION	74
Table 62 Class option method explanation	74
Table 63 CLASS level EXPLANATION	74
Table 64 Class level method explanation	74
Table 65 CLASS QUESTIONTYPE EXPLANATION.....	75
Table 66 Class questiontype method explanation	75
Table 67 CLASS QUESTIONGROUP EXPLANATION.....	75
Table 68 Class questiongroup method explanation	75
Table 69 CLASS QUESTIONOBJECTIVE EXPLANATION	75
Table 70 Class questionobjective method explanation	76
Table 71 CLASS CHAPTER EXPLANATION	76
Table 72 Class chapter method explanation	76
Table 73 CLASS COURSE EXPLANATION.....	76
Table 74 Class course method explanation	77
Table 75 CLASS MAINOBJECTIVE EXPLANATION	77
Table 76 Class mainobjective method explanation.....	77
Table 77 CLASS LEARNINGOUTCOME EXPLANATION.....	77
Table 78 Class learningoutcome method explanation	78
Table 79 CLASS TeacherTest EXPLANATION	78
Table 80 Class teachertest method explanation	78
Table 81 CLASS TeacherTestContent EXPLANATION	79
Table 82 Class teachertestcontent method explanation.....	79
Table 83 manage question bank fields	84
Table 84 manage question bank buttons	84
Table 85 import question buttons	85
Table 86 matching question	86
Table 87create test exam fields.....	90
Table 88 create test exam buttons	90
Table 89 test exam list with “pending” status buttons	91
Table 90 take an exam fields	93
Table 91 Take an exam buttons.....	94
Table 92 create exam fields	95
Table 93 create exam buttons	95
Table 94 set approver fields	95
Table 95 set approver buttons.....	96
Table 96 ERD Dictionary.....	98
Table 97 Semester table Description	99
Table 98 Exam table Description	99
Table 99 Student table Description	99
Table 100 BaseTest table Description.....	100

Table 101 BaseTestContent table Description	100
Table 102 Test table Description	100
Table 103 TestContent table Description	101
Table 104 Question table Description	101
Table 105 Option table Description	101
Table 106 Level table Description.....	102
Table 107 QuestionType table Description	102
Table 108 QuestionGroup table Description	102
Table 109 QuestionObjective table Description.....	102
Table 110 Chapter table Description	102
Table 111 Course table Description.....	103
Table 112 MainObjective table Description	103
Table 113 LearningOutcome table Description	103
Table 114 CourseDepartment table Description.....	103
Table 115 TeacherTest table Description	104
Table 116 TeacherTestContent table Description	104
Table 117 Physical Diagram Description.....	110
Table 118 SEMESTER TABLE DESCRIPTION	111
Table 119 Exam table Description	111
Table 120 Student table Description	111
Table 121 BaseTest table Description.....	112
Table 122 BaseTestContent table Description	112
Table 123 Test table Description	113
Table 124 TestContent table Description	113
Table 125 Question table Description	114
Table 126 Option table Description.....	114
Table 127 Level table Description.....	114
Table 128 QuestionType table Description	115
Table 129 QuestionGroup table Description	115
Table 130 QuestionObjective table Description.....	115
Table 131 Chapter table Description	115
Table 132 Course table Description.....	115
Table 133 MainObjective table Description	116
Table 134 LearningOutcome table Description	116
Table 135 CourseDepartment table Description.....	116
Table 136 TeacherTest table Description	117
Table 137 TeacherTestContent table Description	117
Table 138 AuthenticationApp table Description	117
Table 139 ImportResult table Description.....	118
Table 140 DefaultProcess table Description.....	118
Table 141 Import Question Test Result 1	120
Table 142 Import question test result 2	121
Table 143 Import question test result 3	121
Table 144 Matching Question test result	121

Table 145 Generate Test Result.....	122
Table 146 Approve Exam Test result	122
Table 147 Run exam tool test result 1.....	123
Table 148 Run exam tool test result 2	123
Table 149 Run exam tool test result 3	123
Table 150 Take Exam test result 1	124
Table 151 Take Exam test result 2	124
Table 152 Take Exam test result 3	125

List of Figures

Figure 1 The Scrum Process	23
Figure 2: Use case overview Examination System.....	30
Figure 3: Teacher use case Detail	31
Figure 4: Leader use case detail.....	31
Figure 5: IT use case detail.....	32
Figure 6: Manager use case detail	32
Figure 7: Student use case detail	32
Figure 8: Training Department Staff use case detail	33
Figure 9 <IT> Get list of killed processes.	34
Figure 10 Add process to list of killed processes.....	35
Figure 11 Add process to list of exception processes.....	37
Figure 12 Kill Process	39
Figure 22 Add chapter	40
Figure 23 Delete chapter	42
Figure 24 <Leader> Edit chapter.....	44
Figure 25 <Teacher> Search Question	46
Figure 26 <Teacher> Add Question by Form	48
Figure 27 <Teacher> Add Question by Import File	50
Figure 28 <Teacher> Remove Question.....	52
Figure 29 <Teacher> Edit Questions	53
Figure 31 <Teacher> Approve Test Exam	55
Figure 34 Conceptual Diagram	59
Figure 35 System Architecture Design.....	61
Figure 36 Architecture Overview	62
Figure 37 Web Admin MVC Architecture	63
Figure 38 Component Diagram.....	64
Figure 39 Class diagram	66
Figure 40 Activity diagram Teacher Import Question	81
Figure 41 Activity diagram Training department staff generate Test Exam.....	81
Figure 42 Activity diagram Approve Test Exam	82
Figure 43 Activity diagram Student take a test exam.....	83
Figure 44 manage question bank	84
Figure 45 import question	85
Figure 46 matching question	86
Figure 47 create test exam	87
Figure 48 create test exam	87
Figure 49 create test exam	88
Figure 50 create test exam	88
Figure 51 create test exam	89
Figure 52 create test exam	89
Figure 53 test exam list with "pending" status.....	91
Figure 54 take an exam.....	92

Figure 55 take an exam.....	92
Figure 56 take an exam.....	93
Figure 57 take an exam.....	93
Figure 58 create exam	94
Figure 59 set approver.....	95
Figure 60 ER DIAGRAM	97
Figure 61 Levenshtein distance example result	107
Figure 62 Physical diagram	109
Figure 63 Onion Architecture	126
Figure 64 Event-driven architecture	126
Figure 65 Angular Component Example	127
Figure 66 Guide create semester 1.....	130
Figure 67 Guide create semester 2.....	130
Figure 68 Guide create semester 3.....	131
Figure 69 Guide create semester 4.....	131
Figure 70 Guide create semester 5.....	132
Figure 71 Guide teacher approve test exam	133
Figure 72 Guide teacher approve test exam 2	133
Figure 73 Guide teacher approve test exam 3	134
Figure 74 Guide teacher approve test exam 4	134
Figure 75 Guide teacher approve test exam 5	135
Figure 76 Guide teacher approve test exam 6	135
Figure 77 Guide Staff create exam 1	136
Figure 78 GUIDE STAFF CREATE EXAM 2	136
Figure 79 GUIDE STAFF CREATE EXAM 3	137
Figure 80 GUIDE STAFF CREATE EXAM 4	137
Figure 81 GUIDE STAFF CREATE EXAM 5	138
Figure 82 GUIDE STAFF CREATE EXAM 6	138
Figure 83 GUIDE STAFF CREATE EXAM 7	139

Definitions, Acronyms, and Abbreviations

Name	Definition
L.O	Learning Outcome
MVC	Model View Controller
API	Application Programming Interface
IDE	Integrated development environment
GUI	Graphic user interface
ERD	Entity relationship diagram
I/O	Input/Output

TABLE 1: DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

A. Introduction

1. Project Information

- Project name: **Examination Tools applying Block Chain Technology**
- Project Code: **ExamTool**
- Product Type: **Web Application, Desktop Application**
- Start Date: **September 11, 2018**
- End Date: **December 13, 2018**

2. Introduction

In this document, we will introduce a solution for FPT University Examination system. The current system have some problems like it does not guarantee the quality of the result and it may be used by students to exploit vulnerabilities for cheating. Based on our researches and analysis, we proposed a new solution for FPT University Examination system.

We will organize questions, exams by learning outcome and apply block-chain technology in there. L.O are statements that describe the knowledge or skills which the students should acquire by the end of a particular assignment, class, course, or program and help students understand why those knowledge and skills will be useful to them. A test exam created with L.O will ensure the coverage of a course content inside it. Beside the insurance of the test exam content, we also ensure the test exam will be protected by Block-chain technology. Block-chain is a technology which allows data to be transmitted safely and correctly by a very complex encryption. It will be used in the test exam approving phase to ensure that test exam will not be editable.

3. Current Situation

Two weeks before the exam start, staff will create test exam for each subject. The current question management system is manage questions by chapters. Depending on the teacher's requirements, the staff will randomly select questions based on the topic or chapter. Such random selection like that can not guarantee the test for being covered with the entire knowledge of the subject. After generated, lecture of the subject will be notified and review the test exam. During review process, lectures can remove or edit any questions in the test but the system could not update those modifications automatically in the question bank. Therefore, the reviewer has to update the question once over again in next time. If nothing is wrong, the test will be kept confidential until the exam. The reviewer has to sign in a document to confirm that the test can be used for the upcoming exam.

Upon exam arrival, students will enter the examination room, open the current examination software to take the exam. Before entering the room, students can turn on fraudulent applications and hide it. The examiners cannot know what students are doing outside the room so if the software is not in the blacklist it will not be destroyed

by the exam software and students can use actions such as using shortcut to trigger an event to capture a test page, or to black out a self-contained search text message, or, more importantly, to use TeamViewer to complete the test. Such behaviors will help students pass the exam easily and not evaluate the real ability of students. After the exam finish, the exam software can add those cheating applications to the blacklist, students will find another application and so on, cannot prevent students from cheating during the exam.

4. Problem Definition

4.1. Proposed Solution

- The course is divided by topic and chapter, not cover all the content of a course.
- The question updating is repeated and not synchronized. So that, reduce test exam quality by duplicate question and waste of server's storage.
- Cannot manage student take the exam process during examinations

4.2. Feature Functions

- **Random question base on L.O:** To generate test exam, staff will select L.O first, then all questions will be random base on them.
- **Update question in question bank:** After lecture edit questions in the test exam, that questions in question bank will be updated.
- **Anti-cheating:** We will prevent unnecessary actions and all processes in the list which was not necessary during the test, will be killed.

5. Values and Challenges Value

Values

- Provide better question bank management
- Generate exam that can cover all content of the course
- Kill processes which is not allowed while doing an exam.

Challenges

- Many import types is supported.
- Reduce cheating on exam.

6. Functional Requirement

Functional requirements of the system are listed as below:

Training Department Staff

- Manage exam.
- Support generate examination test from the question banks.

Teacher

- Import question from some different type.
- Edit question and warning when duplicate.
- Approve test exam quality.

Leader

- Manage course.
- Manage chapter.
- Manage learning outcome.

Student

- Support doing an exam.
- Get result of exam when submit test exam process has problem by connection, etc.

IT

- IT can manage student process

System Handler

- Reduced cheating
- Explicit generate test exam, exam, approve process by applying block chain.
- Using authenticate service from FPT University

7. Role and Responsibility

No	Full Name	Role	Position	Contact
1	Kiều Trọng Khánh	Project Owner	Supervisor	khanhkt@fe.edu.vn
2	Nguyễn Minh Hung	Scrum team members	Leader	hungnmse61794@fpt.edu.vn
3	Cao Trung Hiếu	Scrum master	Member	hieuctse61801@fpt.edu.vn
4	Trương Tân Sang	Scrum team members	Member	sangttse61926@fpt.edu.vn

TABLE 2: ROLE AND RESPONSIBILITIES

B. Software Project Management Plan

1. Problem Definition

1.1. Name of this Capstone Project

- **Official name:** Examination Tools applying Block Chain Technology.
- **Vietnamese name:** Xây dựng hệ thống quản lý đề thi và việc thi cử sử dụng công nghệ Block-Chain.
- **Abbreviation:** ExamTool.

1.2. Problem Abstract

This project is created to assist the school to fight against cheating in the examination as well as develop an exam administration to be more optimal. In the process of research and analysis, we decided to implement ExamTool by using extension and applying block-chain in test exam result. But this way have a lot of problem because extension can not intervene the system deeply to prevent cheating and block-chain technology is not suitable for test exam result. So that, we have to change the idea to using an application to kill processes of the system and applying block-chain in approved test exam process.

The Desktop application, which is built on Window, can deep embedding the system and kill unauthorized processes. Firstly, application will send a request to server to verify if this application is valid to access the website. After verified, application will be allowed to access resources for exam questions. Students will do the test during the given time.

Block-chain technology will be applied approved test exam process, which make sure them will not to be lost or changed. For each bad behavior once being discovered during test time, we are going to deduct time or stop the test immediately.

Regarding to the question management function, we find it difficult to input the question to the question bank as well as organize the examination. Therefore, a variety of template formats are going to be provided in the word file, which facilitates the lecturers to import the file. Parsing process will be resolved by the clients. In addition, we also look up the potential duplicate questions before importing them into the question bank, then notify to the lecturer for resolve.

The tests will be generated based on the L.O or chapters of a subject or manually selected by the staff. Thus, the exam will be guaranteed to contain all knowledge of a subject, then notify to the leaders for the approval. If any errors occur during the review process, the reviewer can directly edit the question right on the test. The question which has been corrected will be synchronized versus the question in the question bank.

1.3. Project Overview

1.3.1 Current Situation

Below are the problems encountered in this project:

- **New techniques:** The team is new to block-chain techniques.
- **Lack of knowledge** about the window system.
- **Lack of knowledge** about examination system.

1.3.2 Boundaries of the system

Our system supports:

- Import question file with many question types.
- Matching questions and show all questions can be duplicated.
- Synchronize question in question bank after edited in exam.
- Random questions in exam base on L.O or Chapter.
- Kill process real-time.
- ExamTool run best for Window 10.

Our system hasn't supported:

- ExamTool is not available for others O.S like Linux or Mac
- Exam Management System do not support Internet Explorer and Microsoft Edge.

1.3.3 Future Plans

- Flexible to custom template import question.
- More feature to manage student take exam process.
- Support all question type (reading, listen, writing...)

1.3.4 Development Environment

1.3.4.1 Hardware requirements

Hardware	Minimum Requirements	Recommended
Internet Connection	Cable	Cable
Operating System	Window Server 2008	Window Server 2016
Computer Processor	Intel® Xeon ® 3.0GHz	Intel® Xeon ® Processors
Computer Memory	4GB RAM	8GB RAM or more

TABLE 3: HARDWARE REQUIREMENTS

1.3.4.2 Software requirements

Software	Name / Version	Description
Environment		
Modeling tool	Star UML	
IDE	Visual Studio Enterprise 2017 Visual Studio Code 1.23.1 Webstorm 2018.2	Programming tools
DBMS	SQL Server 2017	Used to create & manage the database for system
Source control	Tortoise SVN 1.10.1	Used for source control

TABLE 4 SOFTWARE REQUIREMENT

2. Coding Convention

JavaScript:

Naming Conventions:

- Variable and function names written as **camelCase**.
- Global variables written in **UPPERCASE**.
- Constants written in **UPPERCASE**.

Function Conventions:

- Put the opening bracket at the end of the first line.
- Use one space before the opening bracket.
- Put the closing bracket on a new line, without leading spaces.
- Do not end a complex statement with a semicolon.

Others:

- Always put spaces around operators (= + - * /), and after commas.
- Always use 4 spaces for indentation of code blocks.
- Always end a simple statement with a semicolon.
- Always end an object definition with a semicolon.

C#:

Naming Conventions:

- Use **PascalCase** for public property, method and type name.
- For parameters and local variables, use **camelCase**.
- For private fields, use prefix **_camelCase** with an **_**.
- Named constants with **ALLCAPS**.
- Vertically align curly brackets.
- Prefix interfaces with the letter **I**. Interface names are noun (phrases) or adjectives.

Reference:

- [C# Style Guide and Coding Convention](#)
- [JavaScripts Style Guide and Coding Convention](#)

3. Project Organization

3.1. Software Process Model

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

- Our team only has 3 members, and tasks are assigned vertically, do all steps from design, coding, testing, and implementation. Scrum is the most suitable model for the 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 the deadline.
- The product owner can change the requirement or extend scope. The team will adapt to change better.

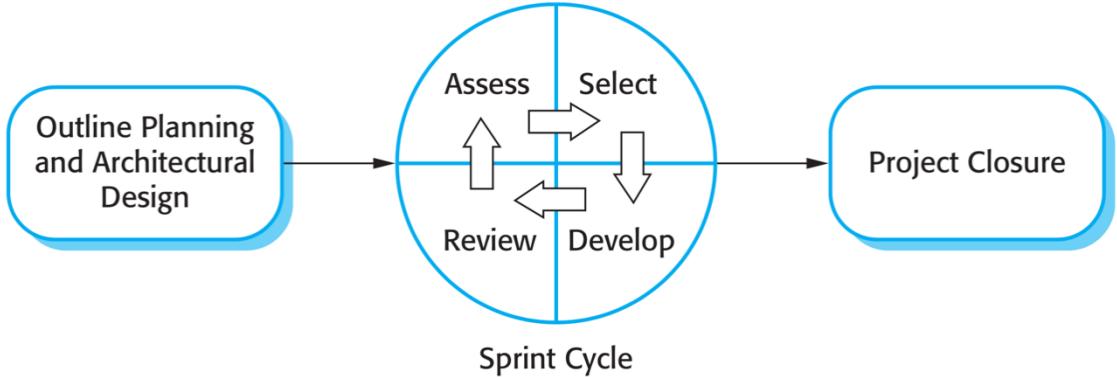


FIGURE 1 THE SCRUM PROCESS

Reference: Software Engineering 9th by Somerville, page 73

3.2. Roles and Responsibilities

No	Full name	Role in group	Responsibilities
1	Kiều Trọng Khánh	Project Owner	Specify scope and user requirement Give out technique and business analysis support Control the development process
2	Cao Trung Hiếu	Scrum master	Create Sprint Backlog and Product Backlog Make sure the Scrum teams understand and follow the process. Help the team master scrum artifacts such as Sprint Backlog, Product Backlog, ... Writing report Always be present to answer questions and give advice when product owner or scrum member needs.
3	Nguyễn Minh Hưng Trương Tân Sang	Scrum team members	Clarifying requirements Prepare documents Designing database GUI Design Coding Testing

TABLE 5 ROLES AND RESPONSIBILITIES DETAIL

3.3. Tools and Techniques

Tool/Technique	Name and version
Front-end	Angular 2+, Jquery, bootstrap
Back-end	.Net framework 4.7.2
IDE	Visual Studio Enterprise 2017, Visual Studio Code 1.23.1, Webstorm 2018.2
Database	SQL Server 2017
Modeling Tool	Star UML
Version Control	Tortoise SVN 1.10.1
Task Management	Trello

TABLE 6 TOOLS AND TECHNIQUES

4. Project Management Plan

4.1. Product Backlog

Spring	Story ID	Story	Task ID	Task
1	1	Introduction document	1.1	Project Information
			1.2	Introduction
			1.3	Current Situation
			1.4	Problem Definition
			1.5	Proposed Solution
			1.6	Role and Responsibility
			1.7	Functional Requirements
	2	Product Backlog	2.1	Create Product Backlog
	3	Project management plan	3.1	Problem Definition
			3.2	Project Organization
			3.3	Project management plan
			3.4	Coding Convention
2	4	Software Requirement Specification	4.1	User Requirement Specification
			4.2	Software System attribute
			4.3	Conceptual diagram
	5	Software Design Diagram	5.1	Design Overview
			5.2	System Architectural Design
			5.3	Component Diagram
			5.4	Interface
			5.5	Interface Design

			5.6	Algorithms
3	6	Anti duplicate question algorithms	6.1	System Problem definition
			6.2	Implement Algorithms
	7	Teacher import question exported from moodle system	7.1	Import question from Moodle
			7.2	Manage Question
			7.3	Manage Course
			7.4	Manage Exam
4	8	Leader Manage LO	8.1	Manage LO
			8.2	Manage Chapter
	9	Manage Test Exam	9.1	Staff Generate Test Exam
			9.2	Leader Approve Test Exam
			9.3	Teacher Review Student Test Exam
			9.4	Blockchain Implement
			9.5	Manage public test exam
	10	Manage Student take Test Exam Process	10.1	Student App Implement
			10.2	Real-time kill process
			10.3	Student take an exam
5	11	Get Student Screen	11.1	Get Student Screen
	12	FPT Authenticate merge	12.1	Use FPT authenticate
	13	Test System	13.1	Run test with FPT student
			13.2	Report bug
			13.3	Fix bug
			13.4	Test document
	14	Software User's Manual	14.1	Installing guide
			14.2	User manual
	15	Document paper	1.5	Paper document

TABLE 7 PRODUCT BACKLOG

4.2. Sprint Backlog

4.2.1 Sprint 1 (10.09.2018 – 23.09.2018): Project initiation

4.2.1.1 Goal

- 1.1 Project Information
- 1.2 Introduction
- 1.3 Current Situation
- 1.4 Problem Definition

- 1.5 Proposed Solution
- 1.6 Role and Responsibility
- 1.7 Functional Requirements
- 2.1 Create Product Backlog
- 3.1 Problem Definition
- 3.2 Project Organization
- 3.3 Project management plan
- 3.4 Coding Convention

4.2.1.2 Development

Task ID	Task	Responsible
1.1	Project Information	HungNM
1.2	Introduction	HungNM
1.3	Current Situation	HungNM
1.4	Problem Definition	HieuCT
1.5	Proposed Solution	HieuCT
1.6	Role and Responsibility	SangTT
1.7	Functional Requirements	SangTT
2.1	Create Product Backlog	HieuCT
3.1	Problem Definition	HungNM
3.2	Project Organization	HieuCT
3.3	Project management plan	HieuCT, SangTT
3.4	Coding Convention	SangTT

TABLE 8 SPRINT 1 DEVELOPMENT

4.2.2 Sprint 2 (24.09.2018 – 8.10.2018): Software Document

4.2.2.1 Goal

- 4 User Requirement Specification
- 4.1 Software Requirement Specification
- 4.2 Software System attribute
- 4.3 Conceptual diagram
- 5.1 Design Overview
- 5.2 System Architectural Design
- 5.3 Component Diagram
- 5.4 Detailed Description
- 5.5 Interface
- 5.6 Interface Design
- 5.7 Algorithms

4.2.2.2 Development

Task ID	Task	Responsible
4	User Requirement Specification	HungNM
4.1	Software Requirement Specification	HungNM
4.2	Software System attribute	HungNM
4.3	Conceptual diagram	HieuCT

5.1	Design Overview	HieuCT
5.2	System Architectural Design	SangTT
5.3	Component Diagram	SangTT
5.4	Detailed Description	HieuCT
5.5	Interface	HungNM
5.6	Interface Design	HieuCT, HungNM
5.7	Algorithms	HieuCT, SangTT

TABLE 9 SPRINT 2 DEVELOPMENT

4.2.3 Sprint 3 (9.10.2018 – 23.10.2018): Basic feature and core

4.2.3.1 Goal

- 6.1 System Problem definition
- 6.2 Implement Algorithms
- 7.1 Import question from Moodle
- 7.2 Manage Exam
- 7.3 Manage Course
- 7.4 Manage Question

4.2.3.2 Development

Task ID	Task	Responsible
6.1	System Problem definition	HieuCT
6.2	Implement Algorithms	HieuCT
7.1	Import question from Moodle	HungNM
7.2	Manage Exam	HungNM, SangTT
7.3	Manage Course	HungNM, SangTT
7.4	Manage Question	HungNM, SangTT

TABLE 10 SPRINT 3 DEVELOPMENT

4.2.4 Sprint 4 (24.10.2018 – 8.11.2018): Software Document

4.2.4.1 Goal

- 7.4 Teacher Import and Manage Question
- 8.1 Manage LO
- 8.2 Manage Chapter
- 9.1 Staff generate test exam
- 10.1 Student App Implement
- 10.2 Real-time kill Process Implement
- 9.2 Leader Approve Test Exam
- 10.3 Student Take Exam
- 9.3 Review Student Test Exam
- 9.4 Block-chain Implement

4.2.4.2 Development

Task ID	Task	Responsible
7.4	Teacher Import and Manage Question	HungNM

8.1	Manage LO	SangTT, HungNM
8.2	Manage Chapter	SangTT, HungNM
9.1	Staff generate test exam	SangTT, HungNM
10.1	Student App Implement	HieuCT
10.2	Real-time kill Process Implement	HieuCT
9.2	Leader Approve Test Exam	SangTT, HungNM
10.3	Student Take Exam	HieuCT
9.3	Review Student Test Exam	HieuCT
9.4	Block-chain Implement	SangTT

TABLE 11 SPRINT 4 DEVELOPMENT

4.2.5 Sprint 5 (9.11.2018 – 23.11.2018): Software Document

4.2.5.1 Goal

- 11.1 IT Get Student Screen
- 12.1 Login with FPT API
- 9.5 Manager public test Exam
- 13.1 Run test With FPT Student
- 13.2 Report bug
- 13.3 Fix bug
- 13.4 Test document

4.2.5.2 Development

Task ID	Task	Responsible
11.1	IT Get Student Screen	HieuCT
12.1	Login with FPT API	SangTT, HieuCT
9.5	Manager public test Exam	SangTT, HungNM
13.1	Run test With FPT Student	SangTT, HieuCT
13.2	Report bug	HieuCT
13.3	Fix bug	HieuCT
13.4	Test document	SangTT, HungNM

TABLE 12 SPRINT 5 DEVELOPMENT

4.2.6 Sprint 6 (24.11.2018 – 13.12.2018): Complete Document

4.2.6.1 Goal

- 14.1 Installation Guide
- 14.2 User Manual
- 15.1 Paper Document

4.2.6.2 Development

Task ID	Task	Responsible
14.1	Installation Guide	HieuCT
14.2	User Manual	SangTT, HungNM
15.1	Paper Document	SangTT, HungNM

TABLE 13 SPRINT 6 DEVELOPMENT

4.3. All Meeting minutes

All meeting minutes saved at: "<https://trello.com/capstonproject2>"

C. Software Requirement Specification

1. User Requirement Specification

1.1. Examination Requirement

The examination system can do the following functions:

- Allow students taking and submitting exams.
- Allow IT support summit student Exam when have technical problem.
- Block virtual machines.
- Block special key strokes.
- Ensure students exam doing process by saving it locally with advance hashing algorithm which password is needed to decrypt.
- Reconnect to FPT University Wi-Fi automatically when disconnected during exam.

1.2. System Requirement

The system does the following functions:

- Categorize questions by L.O.
- Secure and ensure exam creation process step by step by using block-chain technology.
- Kill process in the process must kill list when students start the ExamTool.
- Detect similar questions before import it to the question bank and notify to importer.
- Synchronize questions between question bank and exam bank.

2. System Requirement Specification

2.1. External Interface Requirement

2.1.1 User Interface

The user interface uses language is English for all web application and windows application.

2.1.2 Hardware Interface

N/A

2.1.3 Software Interface

Service 3rd party:

- Microsoft SignalR 2
- FPT Authenticate

2.1.4 Communication Protocol

Use HTTP protocol 1.1 for communication between:

- Web Application and Web Server.
- Windows Application and Web Server.

2.2. System Overview Diagram

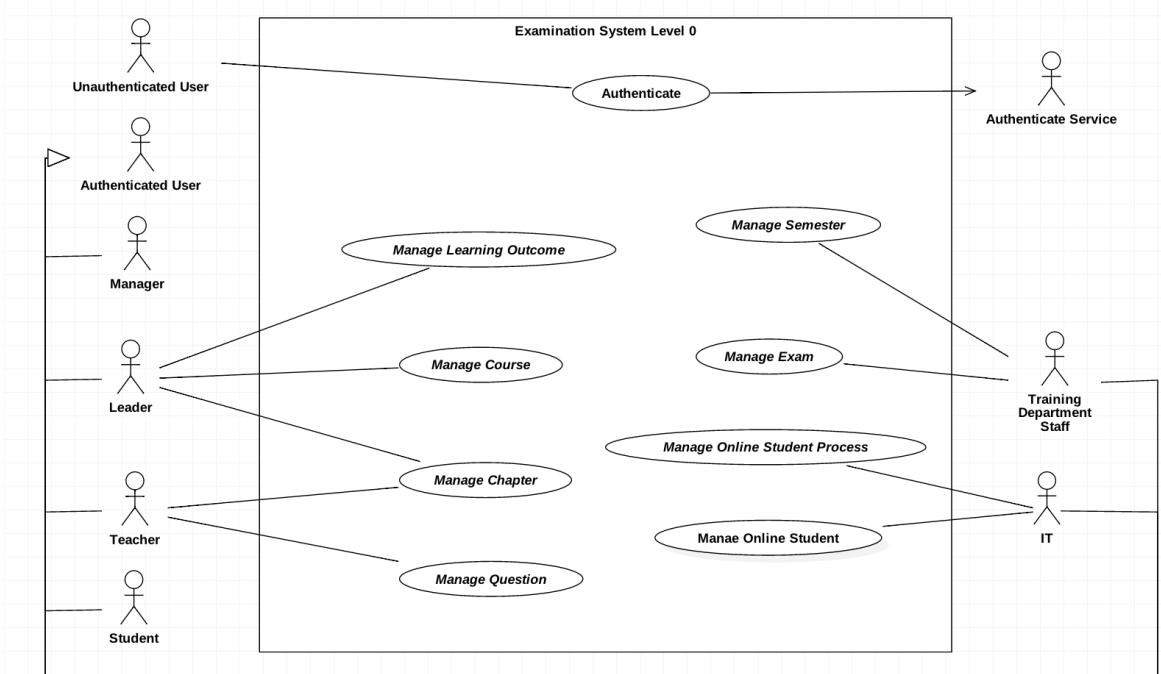


FIGURE 2: USE CASE OVERVIEW EXAMINATION SYSTEM

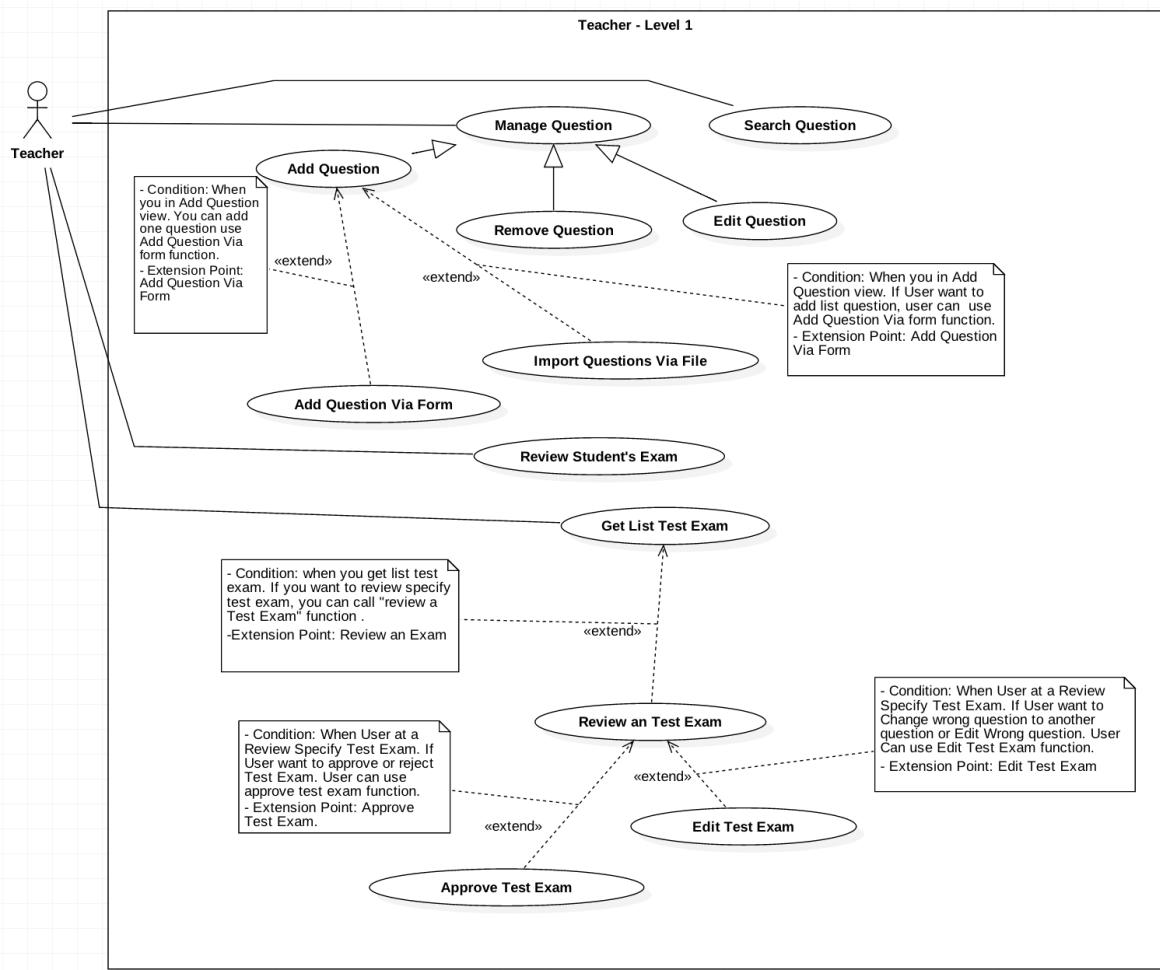


FIGURE 3: TEACHER USE CASE DETAIL

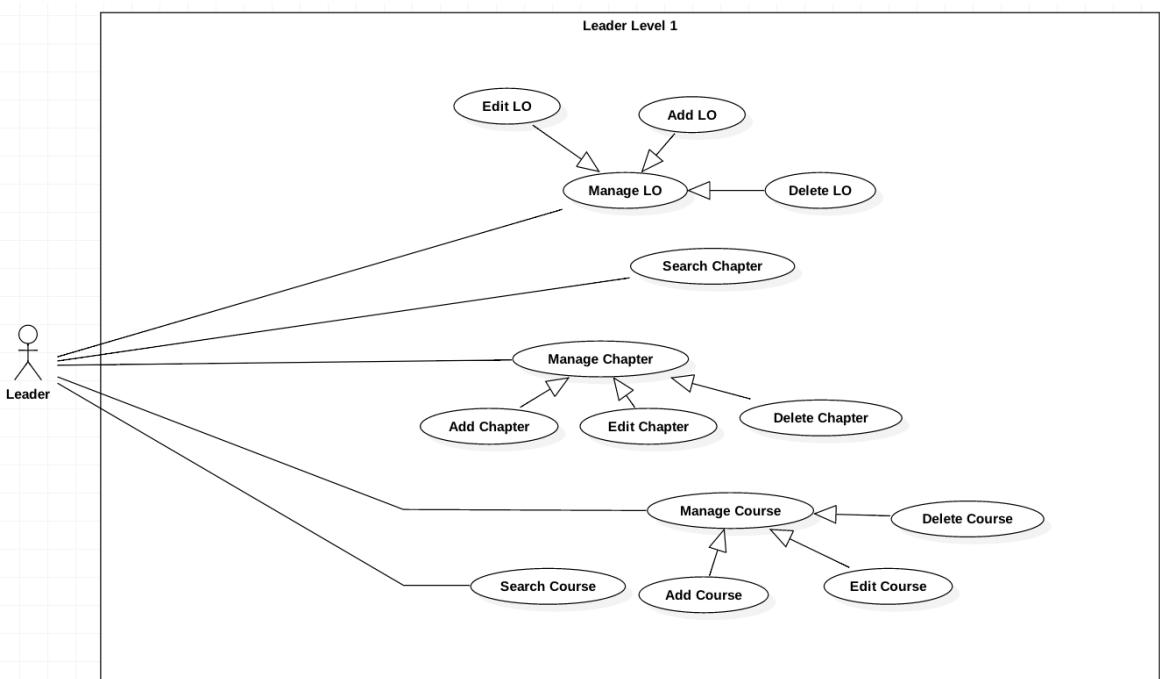


FIGURE 4: LEADER USE CASE DETAIL

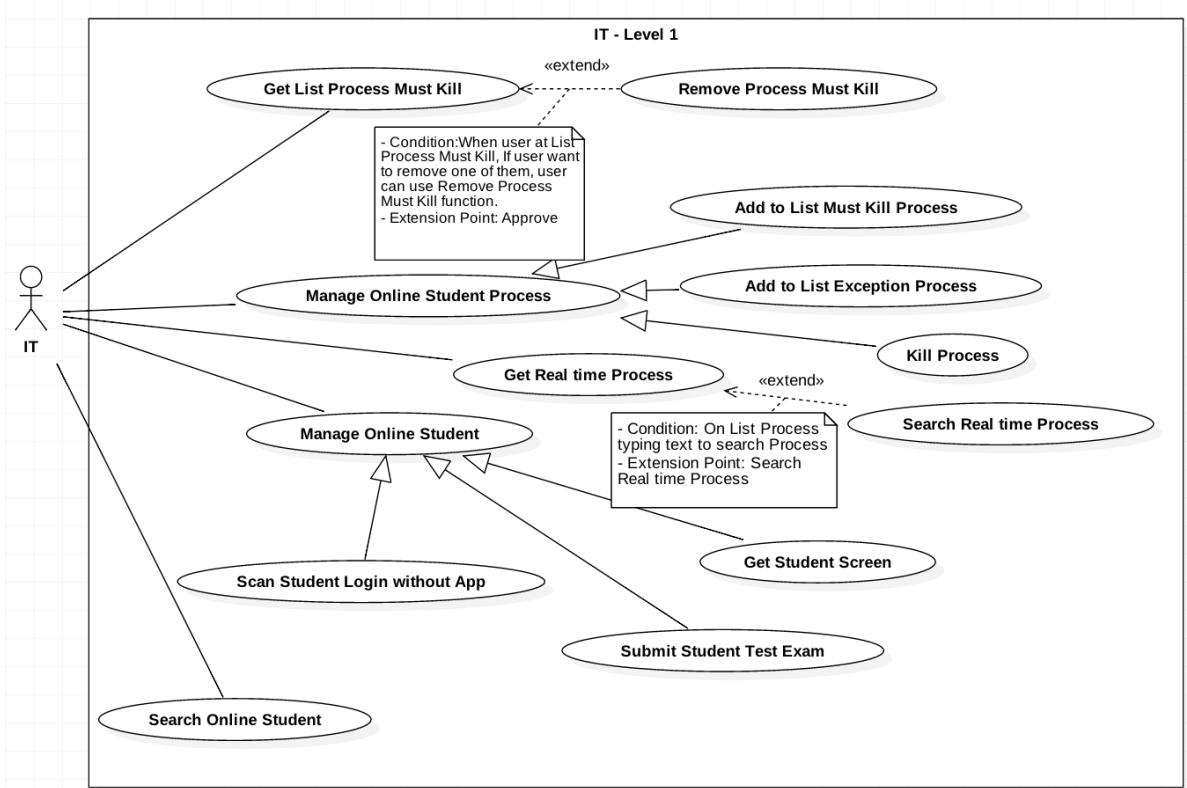


FIGURE 5: IT USE CASE DETAIL

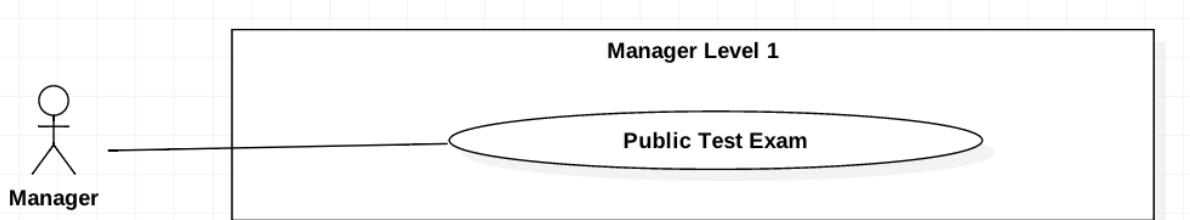


FIGURE 6: MANAGER USE CASE DETAIL

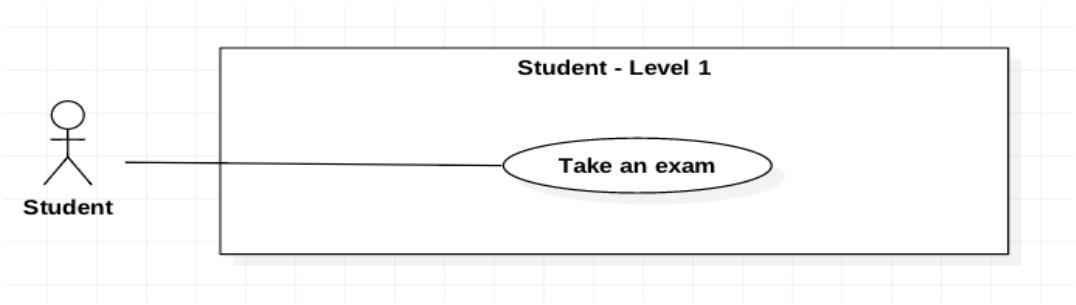


FIGURE 7: STUDENT USE CASE DETAIL

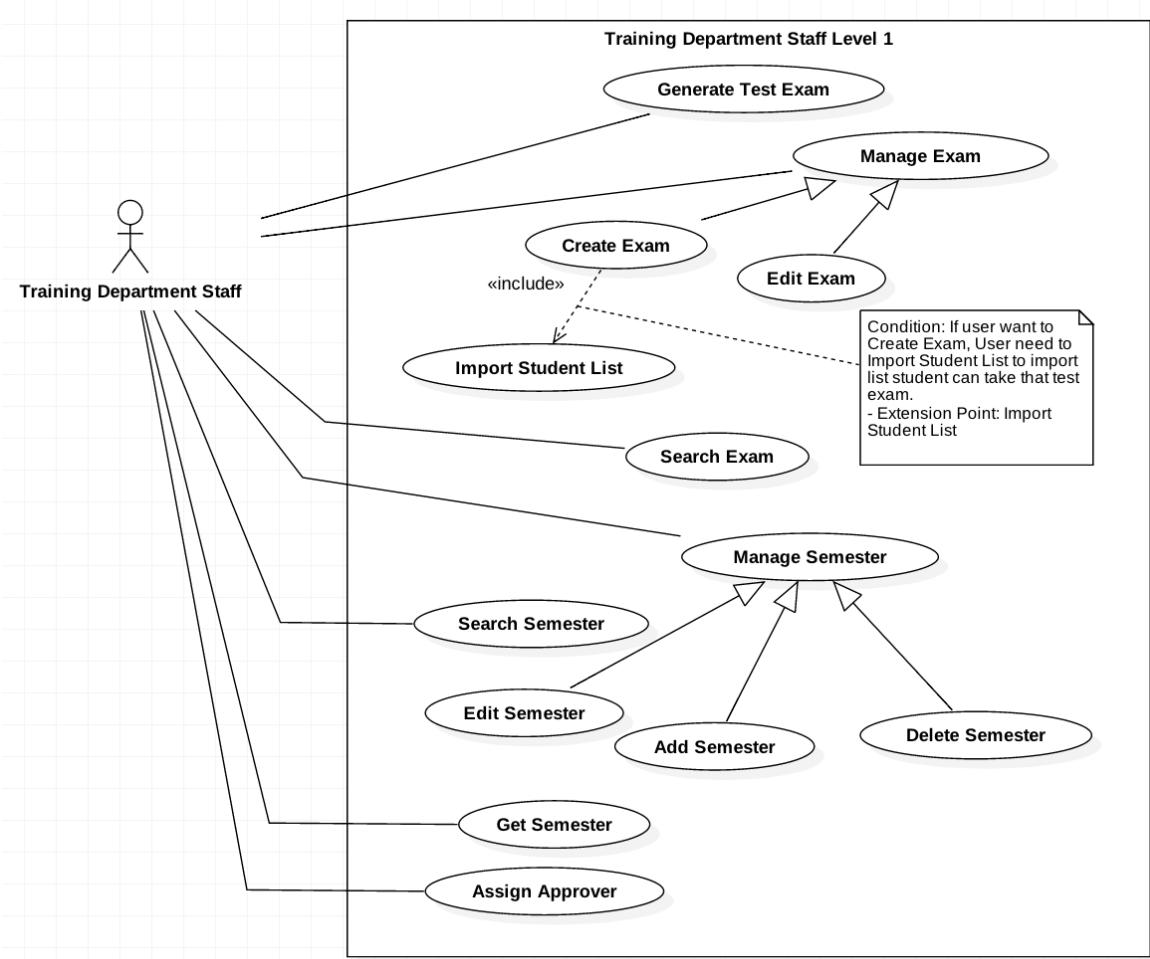


FIGURE 8: TRAINING DEPARTMENT STAFF USE CASE DETAIL

2.3. List of use case

2.3.1 <IT<IT> Get list of killed processes.

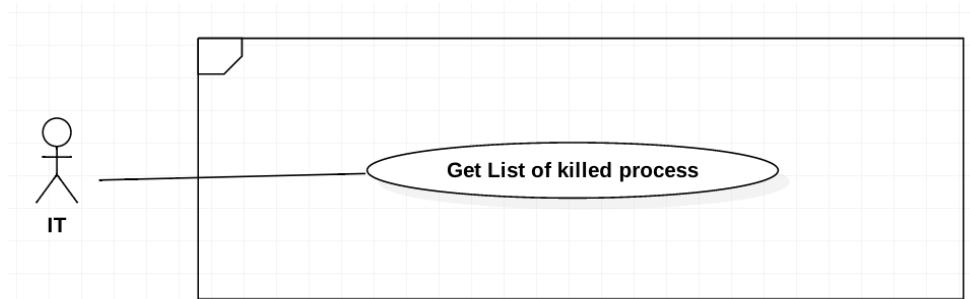


FIGURE 9 *GET LIST OF KILLED PROCESSES.*

USE CASE-UC_IT.01			
Use Case No.	UC_IT01	Use Case Version	2.0
Use Case Name	Get list of killed processes.		
Author	HieuCT		
Date	19/11/2018	Priority	Normal
<p>Actor:</p> <ul style="list-style-type: none"> - IT <p>Summary:</p> <ul style="list-style-type: none"> - Allows IT staff to get list of killed processes. - Help IT staff kill student's processes real-time. - Kill unallowed process when student starts ExanTool up. <p>Goal:</p> <ul style="list-style-type: none"> - IT staff can get list of killed processes to manage student's killed processes. <p>Triggers:</p> <ul style="list-style-type: none"> - User send get process command to server. <p>Preconditions:</p> <ul style="list-style-type: none"> - User log in as IT staff role. <p>Post Conditions:</p> <ul style="list-style-type: none"> - Success: Display list of killed processes. - Fail: Display error messages. <p>Main Success Scenario:</p>			
Step	Actor Action	System Response	
1	IT sends get process command	<p>The system will be displayed list process must kill with information:</p> <ul style="list-style-type: none"> - Process name - Application name 	

		- Process path
Alternative Scenario: N/A		
Exception: N/A		
Relationship: N/A		
Business Rules:		
<ul style="list-style-type: none"> - Staff can get list killed process with following information: <ul style="list-style-type: none"> o Process name o Application name o Process path 		

TABLE 14 <IT> GET LIST PROCESS MUST KILL

2.3.2 <IT> Add process to list of killed processes.

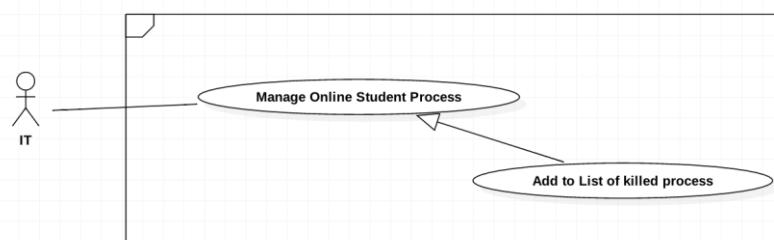


FIGURE 10 ADD PROCESS TO LIST OF KILLED PROCESSES

USE CASE-UC_IT.03			
Use Case No.	UC_IT03	Use Case Version	2.0
Use Case Name	Add process to list of killed processes		
Author	HieuCT		
Date	19/11/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - IT 		
Summary:	<ul style="list-style-type: none"> - When IT staff wants to forbid a process, IT staff can add process(es) to list of killed processes. 		
Goal:	<ul style="list-style-type: none"> - IT staff can add process(es) to list of killed processes. 		
Trigger:	<ul style="list-style-type: none"> - IT staff sends create command to add process(es) to killed processes list. 		
Preconditions:	<ul style="list-style-type: none"> - Log in as IT staff role. 		
Post Conditions:	<ul style="list-style-type: none"> - Success: Process is added to list of killed processes. 		

- Fail: Missing process's informations.

Main Success Scenario:

Step	Actor Action	System Response
1	IT sends get list killed process command.	The system will be displayed list process must kill with information: <ul style="list-style-type: none"> - Process name - Application name - Process path
2	Select the process row which IT want to Add	Parse process data to process information format
3	IT send add to process must kill command	Add new process to processes must killed. System check if process information is missing Process will be added to list of killed process. [Exception]

Exceptions:

Step	Cause	System Response
1	Process information is missing	Show message to IT client

Relationships:

- N/A

Business Rules:

- Get processes must kill from system (A).
- The system get current processes of student (B).
- Get user selected processes from B (C).
- Push C to A.

TABLE 15 ADD MUST BE KILLED PROCESS TO LIST

2.3.3 <IT> Add process to list of exception processes

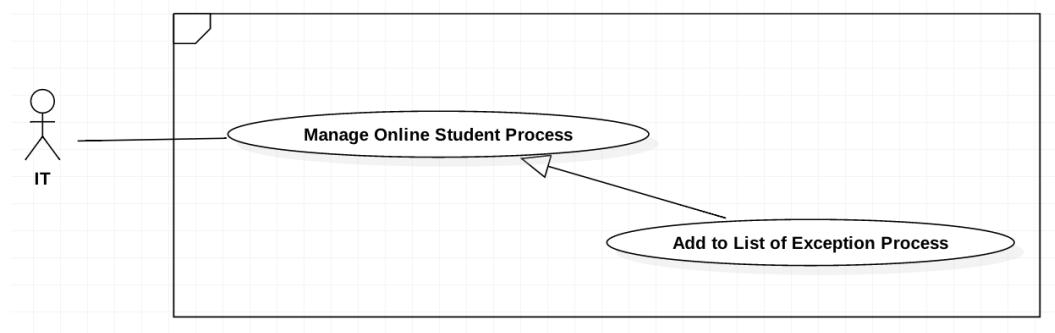


FIGURE 11 ADD PROCESS TO LIST OF EXCEPTION PROCESSES

USE CASE-UC_IT.04												
Use Case No.	UC_IT04	Use Case Version	2.0									
Use Case Name	Add process to list of exception processes											
Author	HieuCT											
Date	19/11/2018	Priority	Normal									
Actor:	<ul style="list-style-type: none"> - IT 											
Summary:	<ul style="list-style-type: none"> - When IT add process(es) to list of exception processes list. 											
Goal:	<ul style="list-style-type: none"> - IT can add process(es) to list of exception processes list. 											
Trigger:	<ul style="list-style-type: none"> - IT send create command to add process(es) to exception processes list. 											
Preconditions:	<ul style="list-style-type: none"> - Log in as IT staff role. 											
Post Conditions:	<ul style="list-style-type: none"> - Success: System add new process(es) in list exception processes - Fail: Process information is missing 											
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>IT sends get list killed process command.</td><td> <p>The system will be displayed list process must kill with information:</p> <ul style="list-style-type: none"> - Process Name - Application Name - Process Path </td></tr> <tr> <td>2</td><td>Select the process row which IT want to Add</td><td>Parse process data to process information format</td></tr> </tbody> </table>			Step	Actor Action	System Response	1	IT sends get list killed process command.	<p>The system will be displayed list process must kill with information:</p> <ul style="list-style-type: none"> - Process Name - Application Name - Process Path 	2	Select the process row which IT want to Add	Parse process data to process information format
Step	Actor Action	System Response										
1	IT sends get list killed process command.	<p>The system will be displayed list process must kill with information:</p> <ul style="list-style-type: none"> - Process Name - Application Name - Process Path 										
2	Select the process row which IT want to Add	Parse process data to process information format										

3	<i>IT send add to process exception command</i>	<i>Add new Process to Process must killed. System check if process information is missing [Exception]</i>
---	---	---

Exceptions:

Step	Cause	System Response
1	<i>Process information is existed</i>	<i>The system will display error messeges.</i>

Relationships:

- *N/A.*

Business Rules:

- *Get exception processes from system (A).*
- *The system get current processes of student (B).*
- *Get user selected processes from B (C).*
- *Exception C from A.*

TABLE 16 ADD PROCESS TO LIST OF EXCEPTION PROCESSES

2.3.4 <IT> Kill Process

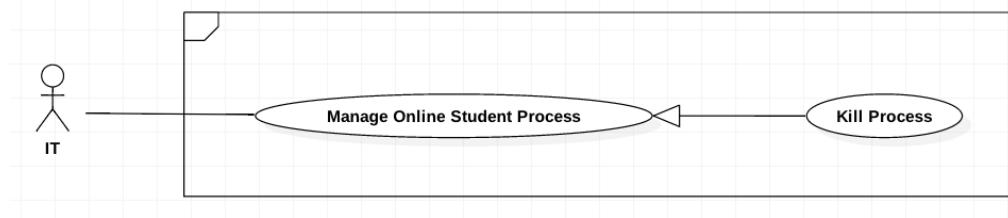


FIGURE 12 KILL PROCESS

USE CASE-UC_IT.05									
Use Case No.	UC_IT05	Use Case Version	2.0						
Use Case Name	Kill Process								
Author	HieuCT								
Date	19/11/2018	Priority	Normal						
Actor:	<ul style="list-style-type: none"> - IT 								
Summary:	<ul style="list-style-type: none"> - When IT kill student's process(es) real-time. 								
Goal:	<ul style="list-style-type: none"> - IT can kill student's process(es) real-time. 								
Triggers:	<ul style="list-style-type: none"> - IT send kill command to server. - Student need to start ExamTool up. 								
Preconditions:	<ul style="list-style-type: none"> - Need to scanned student's processes. - Need to defined cheating processes. 								
Post Conditions:	<ul style="list-style-type: none"> - Success: Process is killed on student's computer. - Fail: Display 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>IT goes to list process real time which show student computer process real time.</td><td> <p><i>The system will be displayed list process must kill with information:</i></p> <ul style="list-style-type: none"> - Process Name - Application Name - Process Path </td></tr> </tbody> </table>			Step	Actor Action	System Response	1	IT goes to list process real time which show student computer process real time.	<p><i>The system will be displayed list process must kill with information:</i></p> <ul style="list-style-type: none"> - Process Name - Application Name - Process Path
Step	Actor Action	System Response							
1	IT goes to list process real time which show student computer process real time.	<p><i>The system will be displayed list process must kill with information:</i></p> <ul style="list-style-type: none"> - Process Name - Application Name - Process Path 							

2	Select the process row which IT want to Kill	Parse process data to process information format
3	IT send Kill Process command	
4		Process selected will kill on all student computer if it existed. System check if process information is missing [Exception]

Relationships:

- N/A.

Business Rules:

- The system get current processes of student (A).
- Get user selected processes from A (B).
- Send request to kill B from A.

TABLE 17 KILL PROCESS

2.3.5 <Leader> Add chapter

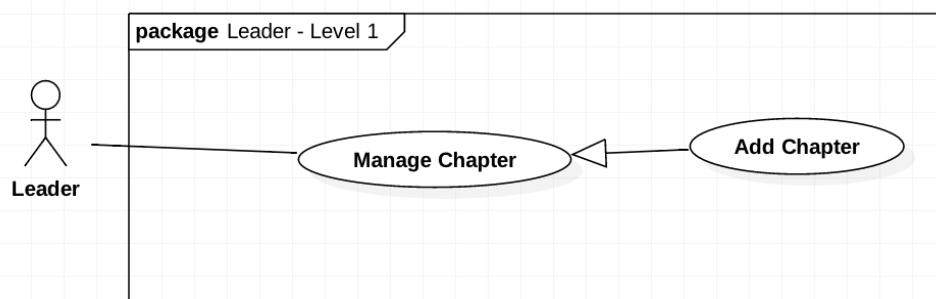


FIGURE 13 ADD CHAPTER

USE CASE-UC_ Leader.151			
Use Case No.	UC_Leader 15	Use Case Version	2.0
Use Case Name	Add Chapter		
Author	HieuCT		
Date	19/11/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Leader 		

Summary:

- Allow user to add chapters.
- Support for organize question by chapters.
- Support for generate question by chapters.

Goal:

- Add new chapter to specify course.

Triggers:

- Send an create command to server.

Preconditions:

N/A

Post Conditions:

- Success: Chapters will be added and display success message
- Fail: Display error messages.

Main Success Scenario:

Step	Actor Action	System Response
1	Leader goes to add chapter view	The system require to fill information: <ul style="list-style-type: none"> - Chapter - Level - Mark - Chapter Content - Options
2	Leader send a save command to save the chapters locally	
3		The system show response message and display new list chapter. [Exception]

Exception:

No	Cause	System Response
1	<i>The chapter info is missing</i>	<i>Show message to refill form to add new chapter.</i>

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

- Get the user input for a new chapter(A).
- Get all chapter available base on course code in system(B).
- If A not existed in B, then insert A to system.

TABLE 18 ADD CHAPTER

2.3.6 <Leader> Delete chapter

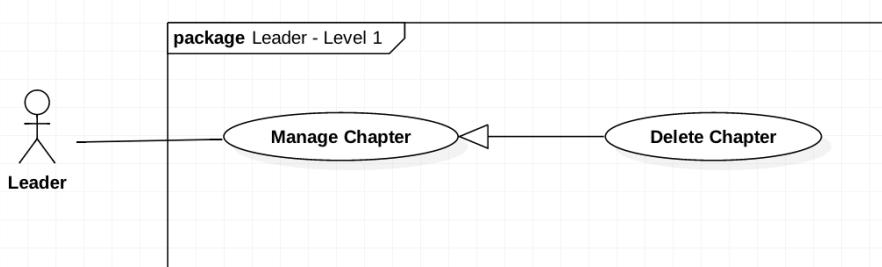


FIGURE 14 DELETE CHAPTER

USE CASE-UC_ Leader.16			
Use Case No.	UC_Loader16	Use Case Version	2.0
Use Case Name	Delete Chapter		
Author	HieuCT		
Date	19/11/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> ○ <i>Leader</i> 		
Summary:	<ul style="list-style-type: none"> ○ <i>This use case allows user remove unnecessary chapters.</i> ○ <i>Support for generate test exam more high quality.</i> 		
Goal:	<ul style="list-style-type: none"> ○ <i>Remove the chapter from list chapter of this course</i> 		
Triggers:	<ul style="list-style-type: none"> ○ <i>send delete chapter command.</i> 		
Preconditions:	<ul style="list-style-type: none"> ○ <i>Login the system with Leader role.</i> 		
Post Conditions:	<ul style="list-style-type: none"> ○ <i>Success: Chapters will be added.</i> 		

- Fail: Chapters will not be added and show error message

Main Success Scenario:

Step	Actor Action	System Response
	<i>Leader goes to list chapter</i>	<i>The system will display list chapter with information:</i> - Chapter - Level - Mark - Chapter Content - Options
	<i>Leader select chapter want to remove</i>	
3	<i>Leader send a delete chapter command.</i>	<i>The system will show a popup notify if that chapter has been deleted successfully</i>

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Get the user input for a chapter(A).
- Get all chapter available base on course code in system(B).
- If A existed in B, then remove A from system.

TABLE 19 DELETE CHAPTER

2.3.7 <Leader> Edit chapter

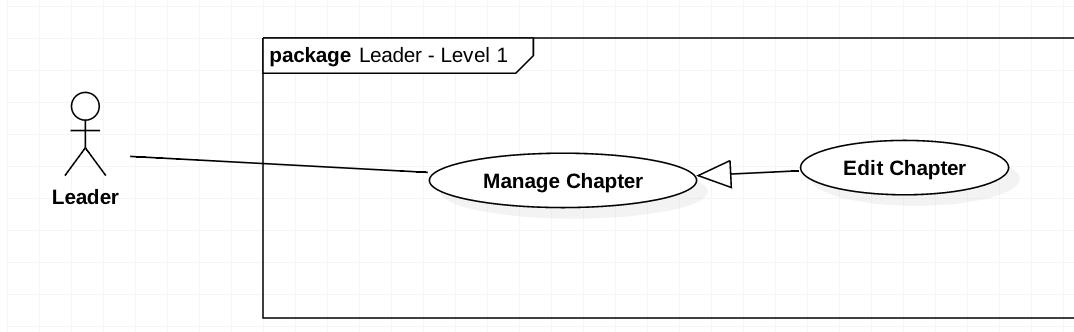


FIGURE 15 <LEADER> EDIT CHAPTER

USE CASE-UC_ Leader.18			
Use Case No.	UC_Leader18	Use Case Version	2.0
Use Case Name	Edit Chapter		
Author	HieuCT		
Date	19/11/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - <i>Leader</i> 		
Summary:	<ul style="list-style-type: none"> - <i>This use case allows user to edit chapter name and chapter order.</i> - <i>Support organize question and course more complete.</i> 		
Goal:	<ul style="list-style-type: none"> - <i>Edit chapter with new name.</i> 		
Triggers:	<ul style="list-style-type: none"> - <i>User send an edit chapter command.</i> - <i>User input new chapter Name.</i> 		
Preconditions:	<ul style="list-style-type: none"> - <i>Login the system with Leader role.</i> - <i>Change chapter to new information (chapter name, chapter order)</i> 		
Post Conditions:	<ul style="list-style-type: none"> - <i>Success: Chapters will be edited show success message</i> - <i>Fail: Chapters will not be edit and show message error</i> 		

Main Success Scenario:

Step	Actor Action	System Response
1	<i>Leader goes to List chapter view</i>	<i>The system will display list chapter with information:</i> - Chapter - Level - Mark - Chapter Content - Options
2	<i>Leader fill the form</i>	<i>The system requires Leader to input:</i> - Chapter - Level - Mark - Chapter Content - Options
3	<i>Leader send a save command to save the chapters locally</i>	[Exception]
4	<i>Leader send save command</i>	<i>The system will run save change command to save all changed</i>

Exception:

No	Cause	System Response
1	<i>The chapter info is missing</i>	<i>Show message to fill full form,</i>

Alternative Scenario: N/A

Relationships: Manage Chapter

Business Rules:

- Get the chapter base on chapter's id (A).
- Update A by user input (name, order, ...).
- Save A to the system.

TABLE 20 <LEADER> EDIT CHAPTER

2.3.8 <Teacher> Search Question

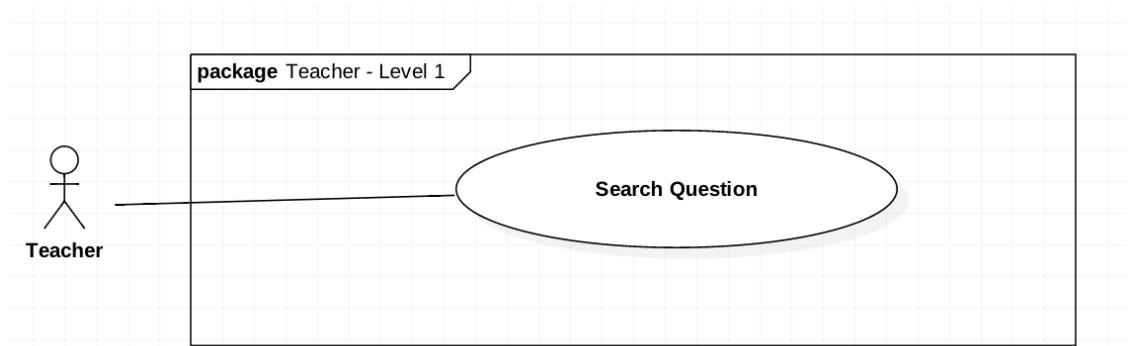


FIGURE 16 <TEACHER> SEARCH QUESTION

USE CASE-UC_Teacher.19			
Use Case No.	UC_Teacher 19	Use Case Version	2.0
Use Case Name	Search Question		
Author	HieuCT		
Date	19/11/2018	Priority	Normal
Actor:			
- Teacher			
Summary:			
<ul style="list-style-type: none"> - Allows search questions by question's code and content - Support user can find question to edit question or remove question from question bank. 			
Goal:			
<ul style="list-style-type: none"> - Finding the question with matching code and question content. 			
Triggers:			
<ul style="list-style-type: none"> - Teacher send search command 			
Preconditions:			
<ul style="list-style-type: none"> - Login the system with teacher role. - User input question information (question content, question code..) 			
Post Conditions:			
<ul style="list-style-type: none"> - Success: Questions will be found or not be found - Fail: Display error messages 			
Main Success Scenario:			
Step	Actor Action	System Response	

<i>1</i>	<i>Teacher will go to list question page</i>	<i>The system will display list question information at specify page:</i> <ul style="list-style-type: none"> - <i>Code</i> - <i>Course</i> - <i>Mark</i> - <i>Level</i>
<i>1</i>	<i>Teacher input name or code in the search field</i>	
<i>2</i>	<i>Teacher sends a command to get questions to follow name or code in the search field</i>	<i>The system run command and show questions have been searching on the question list:</i> <ul style="list-style-type: none"> - <i>Code</i> - <i>Course</i> - <i>Level</i> - <i>Mark</i>

Alternative Scenario:

Step	Actor Action	System Response
<i>1</i>	<i>Teacher sends a command to get questions to follow name or code in the search field</i>	<i>Show result not found on the screens</i>

Exceptions: N/A.

Relationships: N/A.

Business Rules:

- *Get question list base on course code (A).*
- *Filter all questions has content contain search value in A (B).*
- *System send back B to client.*

TABLE 21 <TEACHER> SEARCH QUESTION

2.3.9 <Teacher> Add Question by Form

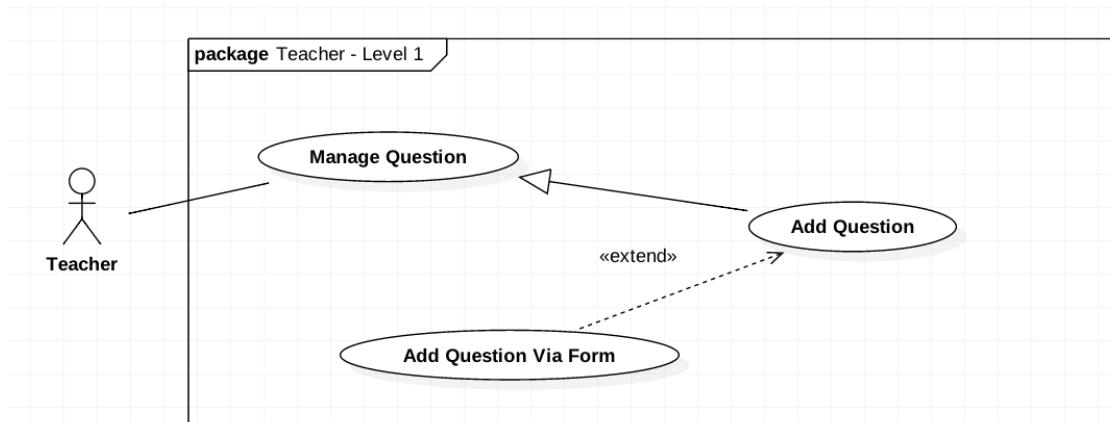


FIGURE 17 <TEACHER> ADD QUESTION BY FORM

USE CASE-UC_Teacher.20			
Use Case No.	UC_Teacher 20	Use Case Version	2.0
Use Case Name	Add Question by Form		
Author	HieuCT		
Date	19/11/2018	Priority	Normal
Actor:	<ul style="list-style-type: none"> - Teacher 		
Summary:	<ul style="list-style-type: none"> - <i>This use case allows teacher to add questions by form.</i> 		
Goal:	<ul style="list-style-type: none"> - <i>Teacher can add question by form.</i> 		
Triggers:	<ul style="list-style-type: none"> - <i>Teacher send an create command.</i> 		
Preconditions:	<ul style="list-style-type: none"> - <i>Login the system with teacher role.</i> 		
Post Conditions:	<ul style="list-style-type: none"> - <i>Success: Questions will be added.</i> - <i>Fail: Questions will not be added.</i> 		

Main Success Scenario:

Step	Actor Action	System Response
1	<i>Teacher fill the form</i>	<i>The system requires teacher to input:</i> - Chapter - Level - Mark - Question Content - Options
2	<i>Teacher send a save command to save the questions locally</i>	
3	<i>Teacher send a send command add question.</i>	<i>The system will run command and question and comeback to question list. [Exception]</i>

Exception:

No	Cause	System Response
1	<i>Teacher not fill full question info</i>	<i>Show message to fill full form.</i>
2	<i>Teacher not choose write answer for question.</i>	<i>Show message require answer</i>

Alternative Scenario:

Step	Actor Action	System Response
1	<i>Teacher add question with content matching >80% question already in database</i>	<i>Show warning may question already in database and confirm continuous to add or cancel action.</i>

Relationships: Add Question

Business Rules:

- Teacher fill question form (A).
- A has format:

Content	Options	Mark	Level	Chapters	Learning outcomes
---------	---------	------	-------	----------	-------------------

- Add A to question list.

TABLE 22 <TEACHER> ADD QUESTION BY FORM

2.3.10 <Teacher> Add Question by Import File

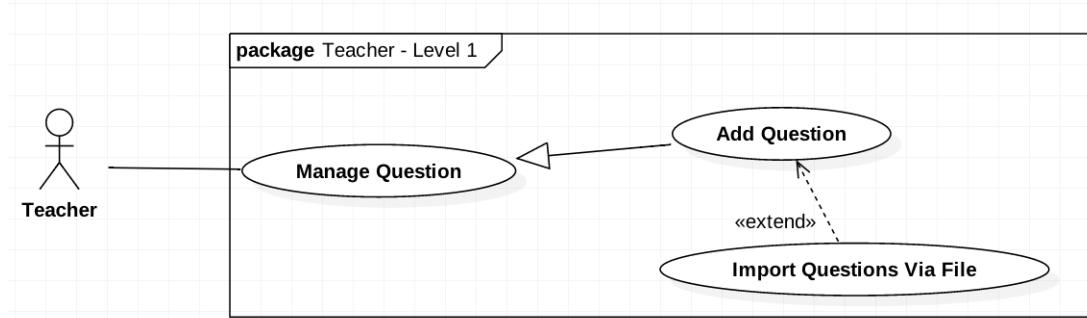


FIGURE 18 <TEACHER> ADD QUESTION BY IMPORT FILE

USE CASE-UC_Teacher.21			
Use Case No.	UC_Teacher 21	Use Case Version	2.0
Use Case Name	Add Question by Import File		
Author	HieuCT		
Date	19/11/2018	Priority	Normal
Actor:			
- Teacher			
Summary:			
- This use case allows teacher to add questions by uploading file.			
Goal:			
- Upload multiple question list exported by Moodle (old system)			
Triggers:			
- Teacher send an create command.			
Preconditions:			
- Login the system with teacher role.			
Post Conditions:			
- Success: Questions will be added. - Fail: Questions will not be added.			

Main Success Scenario:

Step	Actor Action	System Response
1	<i>Teacher upload the file</i>	<i>The system requires teacher to upload a file with GIFT format or file extensions: .doc,.xml</i>
		<i>System run command to validate and check duplicate question.</i>
2	<i>Teacher review the uploaded questions</i>	<i>Show question format status:</i> - Success - Warning - Error
3	<i>Teacher send a submit command to save the questions to the database</i>	<i>System run add questions to add question list</i>

Alternative Scenario: N/A

Exceptions: N/A

Relationships: Add Question

Business Rules:

- *Uploaded a file (A).*
- *Validate A by checking file type (.xml, .docx, .txt) and file format (GIFT format)(B)*
- *If B valid, parse B to question list.*
- *Question information has format:*

Content	Options	Mark	Level	Chapters	Learning outcomes
---------	---------	------	-------	----------	-------------------

TABLE 23 <TEACHER> ADD QUESTION BY IMPORT FILE

2.3.11 <Teacher> Remove Question

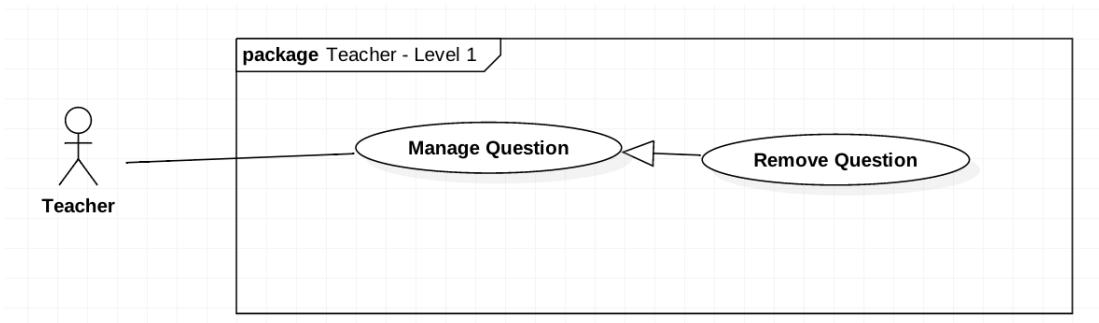


FIGURE 19 <TEACHER> REMOVE QUESTION

USE CASE-UC_Teacher.22			
Use Case No.	UC_Teacher 22	Use Case Version	2.0
Use Case Name	Remove Question		
Author	HieuCT		
Date	19/11/2018	Priority	Normal
Actor:			
- Teacher			
Summary:			
- Remove question from question bank.			
Goal:			
- Remove question from question bank.			
Triggers:			
- Teacher send delete command.			
Preconditions:			
- Login the system with teacher role.			
Post Conditions:			
- Success: Question will be removed. - Fail: Notify error when application cannot connect to server.			
Main Success Scenario:			

Step	Actor Action	System Response
1	<i>Teacher goes to list question list</i>	<i>The system will display list question information:</i> - Course - Mark - Level
2	<i>Teacher choose delete question</i>	<i>The system will show a warning popup to ensure delete action again.</i>
3	<i>Teacher confirm remove question in popup.</i>	
4	<i>Teacher send a delete question command.</i>	<i>The system will show a popup notify that question has been deleted successfully</i>

Alternative Scenario: N/A

Exceptions: N/A

Relationships: N/A

Business Rules:

- Get all questions in system (A).
- Filter question in A base on question's code (B).
- If B existed in A, remove B from system.

TABLE 24 <TEACHER> REMOVE QUESTIONS

2.3.12 <Teacher> Edit Questions

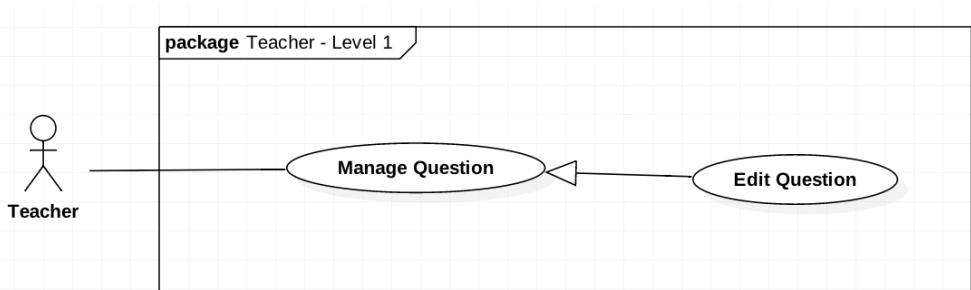


FIGURE 20 <TEACHER> EDIT QUESTIONS

USE CASE-UC_Teacher.23

Use Case No.	UC_Teacher 23	Use Case Version	2.0
Use Case Name	Edit Questions		
Author	HieuCT		
Date	19/11/2018	Priority	Normal

Actor:

- *Teacher*

Summary:

- *Update question information.*

Goal:

- *Update question information.*

Triggers:

- *Teacher send an update command.*

Preconditions:

- *Login the system with teacher role.*

Post Conditions:

- *Success: Questions will be edited.*
- *Fail: Notify error when user does not full fill information or application cannot connect to server.*

Main Success Scenario:

Step	Actor Action	System Response
1	<i>Teacher goes to list question view</i>	<i>The system will display list question information:</i> - Course - Mark <i>Level</i>
2	<i>Teacher choose to view detail question</i>	<i>The system will display detail information of this question:</i>

		<ul style="list-style-type: none"> - Question code - Question content - Option - Learning Outcome - Chapter - Mark - Percent - Level - Right answer
3	Teacher edit question information	The system require fill full info except learning outcome.
4	Teacher send a submit command to save the questions.	The system run save question command to save and comeback to list question.

Alternative Scenario: N/A

Exceptions: N/A

Relationships: Manage Question

Business Rules:

- Get the question base on question's code (A)
- Update A by user input as format :

Content	Options	Mark	Learning outcomes	Chapters	Level

- Save A to system.

TABLE 25 <TEACHER> EDIT QUESTIONS

2.3.13 <Teacher> Approve Test Exam

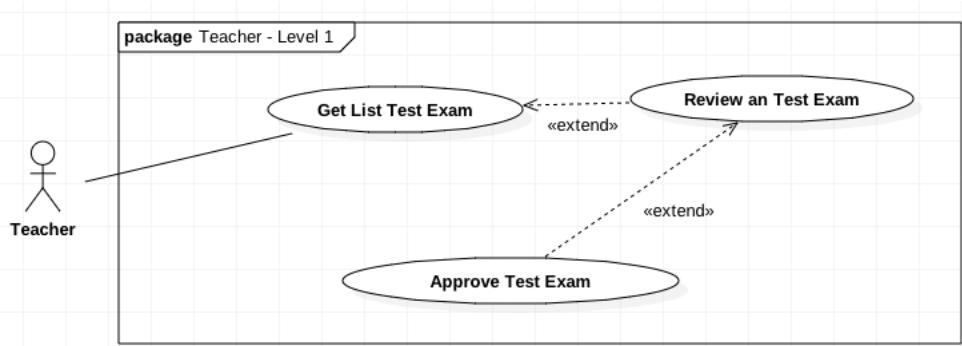


FIGURE 21 <TEACHER> APPROVE TEST EXAM

USE CASE-UC_Teacher.25			
Use Case No.	UC_Teacher 25	Use Case Version	2.0
Use Case Name	Approve Test Exam		
Author	HieuCT		
Date	19/11/2018	Priority	Normal

Actor:

- Teacher

Summary:

- User approve test exam.
- Test exam which status is “Approve” can be release to create exam.

Goal:

- Approve test exam.

Triggers:

- Teacher send an update command.

Preconditions:

- Login the system with teacher role.

Post Conditions:

- Success: Status of test exam will be changed to “Approved”.
- Fail: Notify error when application cannot connect to server.

Main Success Scenario:

Step	Actor Action	System Response
1	Teacher send get list test exam command	<p>The system will display list test exam information assigned to that teacher:</p> <ul style="list-style-type: none"> - Course - Number of questions - Duration
	Teacher send get detail test exam command.	The system will display list question and require teacher take that exam.

3	<i>Teacher submit test exam</i>	<i>The system will calculate mark and change Test Exam status to taken and display Test Exam result to interface with information:</i> - <i>Mark</i> - <i>Test Exam Result</i>
4	<i>Teacher send command approve test exam. [Alternative Scenario 1]</i>	<i>The system will change Test Exam to approved or rejected and come back display list exam assigned to approve.</i>

Alternative Scenario:

Step	Actor Action	System Response
1	<i>Teacher send command edit questions</i>	<i>The system will require teacher to retake exam</i>

Exceptions: N/A

Relationships: Extend “Review an test exam”

Business Rules:

- *Get test exam base on code (A).*
 - o *If A's status is “Pending”, teacher take an exam. After A finished, A's status change to “Taken”.*
 - o *If A's status is “Taken”:*
 - *If teacher update A, A's status change to “Edited”.*
 - *If teacher approve. A's status change to “Approved”.*
 - *If teacher reject, A's status change to “Rejected”.*
 - o *If A's status is “Edited”, teacher available to retake and A's status change to “Pending”.*

3. Software System Attribute

3.1. Usability

- Font size: 8px – 30px.
- Color: Light Red, Light Green, Blue, White, Light Gray
- Background: White

3.2. Reliability

- All appointment is recorded, not missing a single one. Web application never crash in any circumstance.

3.3. Availability

- System replies in maximum 5 seconds.

3.4. Security

- Web application have multiple role and each role has a specific permission to interact with.

3.5. Maintainability

- All server software, web application is divided into separated modules for easy maintain.

3.6. Portability

- Web application can be run on Chrome browser version 42 or later.

4. Conceptual Diagram

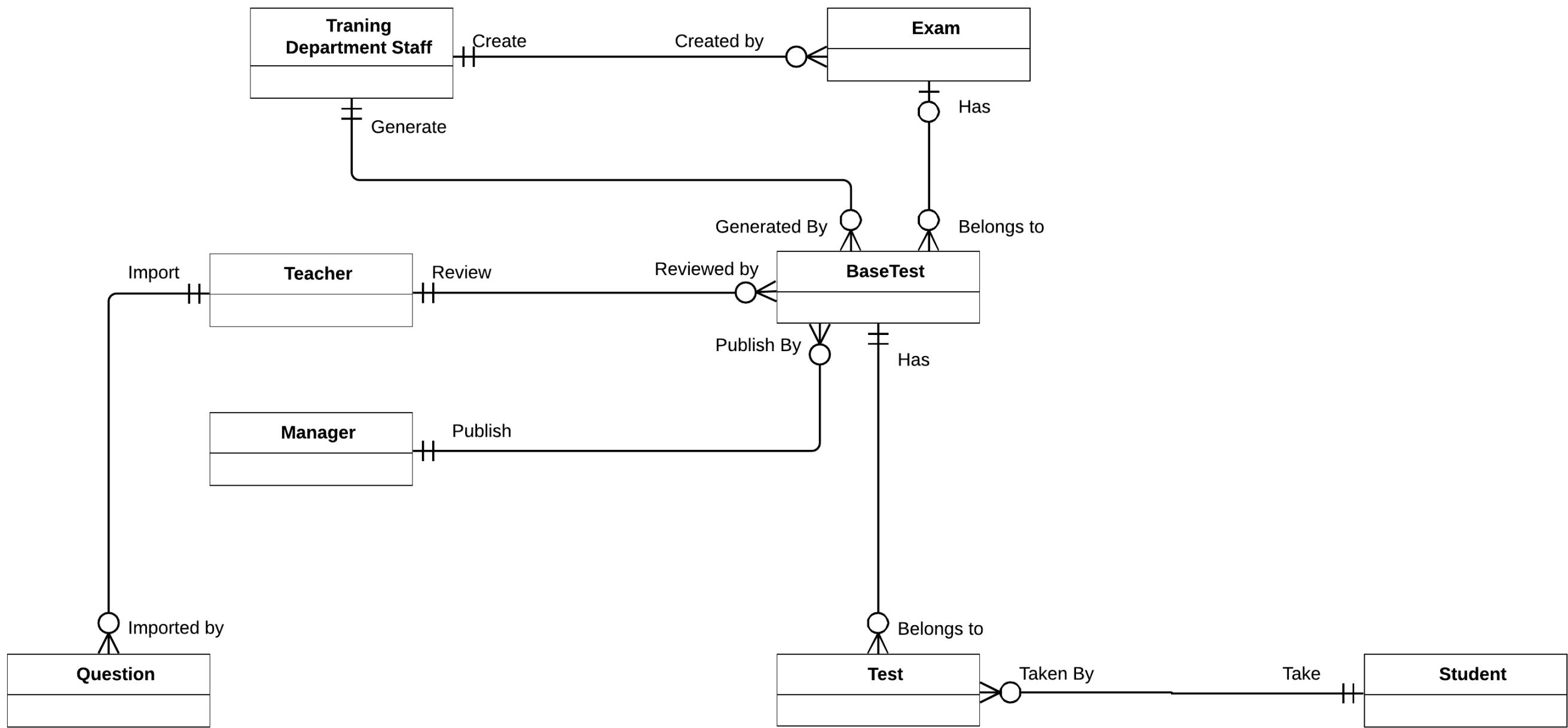


FIGURE 22 CONCEPTUAL DIAGRAM

Entity Name	Description
Training Department Staff	The Staff who generate exam, Test exam, semester.
Exam	Created by Training department Staff for every end of semester
BaseTest	Contain test exam information
Teacher	The people who import question and approve test exam
Manager	The people who publish test exam
Question	Contain questions information
Test	Contain student's test information
Student	Contain Student's information

TABLE 26 CONCEPTUAL DIAGRAM EXPLANATION

D. Software Design Description

1. Design Overview

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

The architectural design describes the overall architecture of the system and the architecture of each main component and subsystem.

The detailed design describes a static and dynamic structure for each component and functions. It includes class diagrams, class explanations and sequence diagrams for each use cases.

The database design describes the relationships between entities and details of each entity.

Document overview:

Section 2: gives an overall description of the system architecture design.

Section 3: gives component diagrams that describe the connection and integration of the system.

Section 4: gives the detail design description which includes a class diagram, class explanation, activity diagram and sequence diagram to details the application functions.

Section 5: describe screen design.

Section 6: describe fully attribute ERD.

Section 7: describe algorithms.

2. System Architecture Design

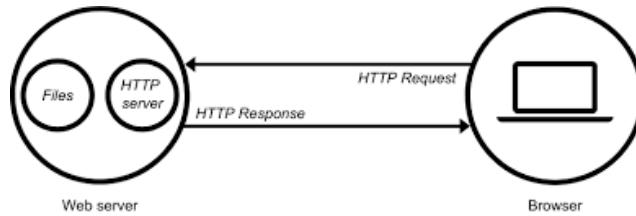


FIGURE 23 SYSTEM ARCHITECTURE DESIGN

2.1. Web Server Architecture Design

Overall architecture of Examination Tool System with functional layers and the collaboration between the system and the external systems is shown in figure 3 below.

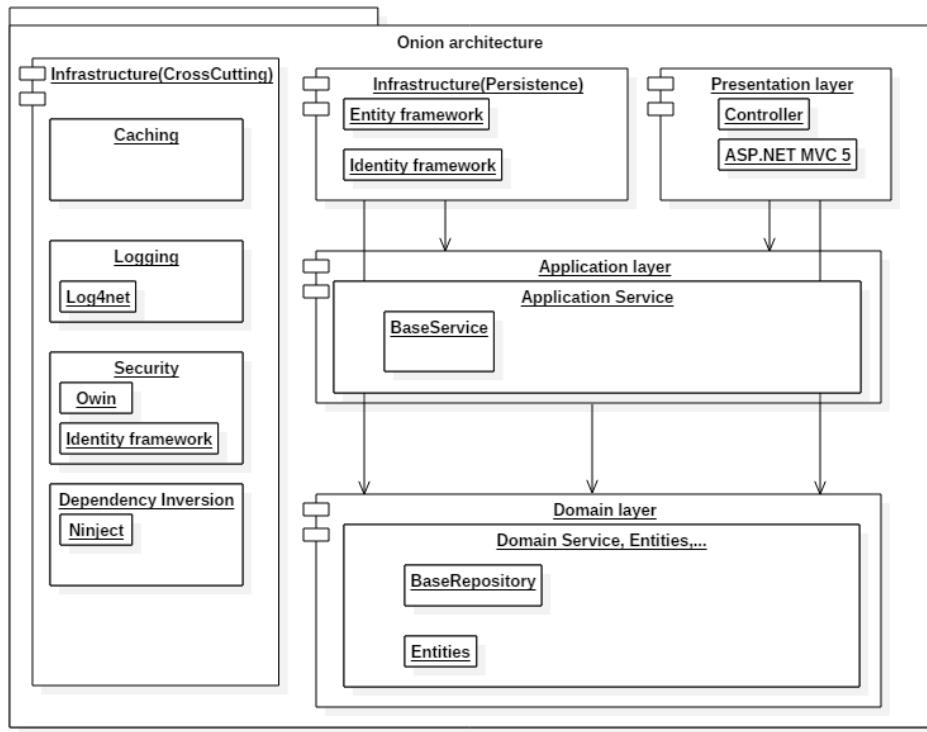


FIGURE 24 ARCHITECTURE OVERVIEW

No.	Layer Name	Description
1	Presentation layer	Contains UI, handle request
2	Application layer	Provides services for application
3	Domain layer	Provides full services, stores entities
4	Infrastructure layer	Communicate with other framework, work with database

TABLE 27 ARCHITECTURE OVERVIEW DESCRIPTION

2.2. Client Architecture Design

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

- The Model-View-Controller pattern highly supports the separation of concerns. This advantage not only increases the testability of the code but it also makes it easier to extend, allowing a fairly easy implementation of new features.

- If the Views respect the single responsibility principle then their role is just to update the Controller for every user event and just display data from the Model, without implementing any business logic. In this case, UI tests should be enough to cover the functionalities of the View.

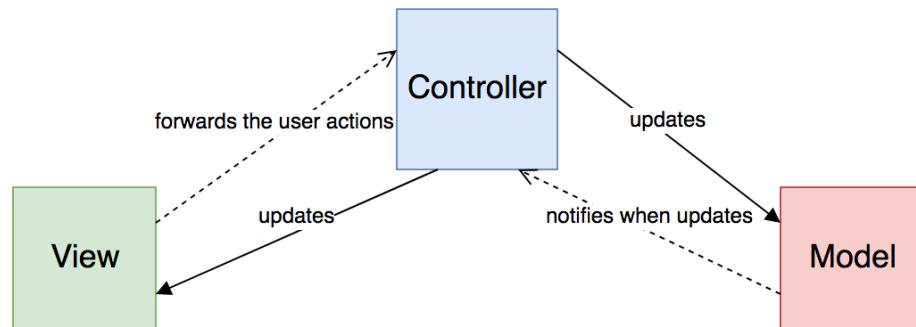


FIGURE 25 WEB ADMIN MVC ARCHITECTURE

(Reference: <https://medium.com/swlh/ios-design-patterns-a9bd07818129>)

- **Model:** which represents the underlying, logical structure of data in a software application and the high-level class associated with it. This object model does not contain any information about the user interface.
- **View:** which is a collection of classes representing the elements in the user interface (all of the things the user can see and respond to on the screen, such as buttons, display boxes, and so forth)
- **Controller:** which represents the classes connecting the model and the view, and is used to communicate between classes in the model and view.

3. Component Diagram

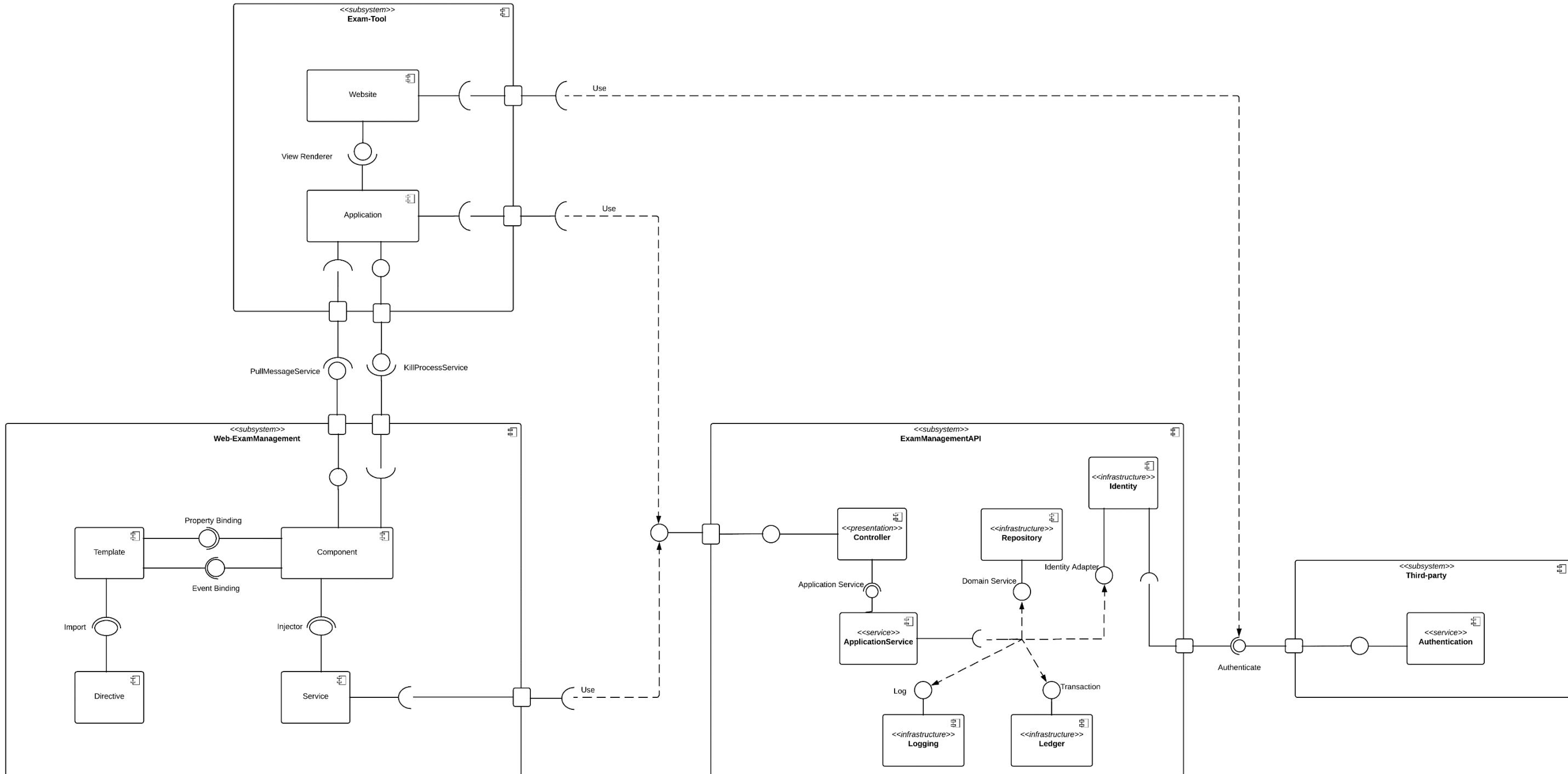


FIGURE 26 COMPONENT DIAGRAM

Entity Name	Description
<<subsystem>> EXAM-TOOL	<p>This subsystem provide tool help Student can take a test exam and through this tool IT can manage process on student computer. This subsystem include 2 main component:</p> <ul style="list-style-type: none"> - Website take a test exam - Application
Website	Provide interface for student take a test exam
Application	Provide tool for IT can collect process on student's computer and send to IT management website. It can screenshot student screen.
Web-Exam Management	This web build for manage and build by angular with angular structure. This website use API provide by ExamManagementAPI subsystem
Authenticate service	This service provide by FPT University. (Login, Get User info...)
ExamManagementAPI	Provide API for System.

TABLE 28 COMPONENT DIAGRAM DICTIONARY

4. Detailed Description

4.1. Class Diagram

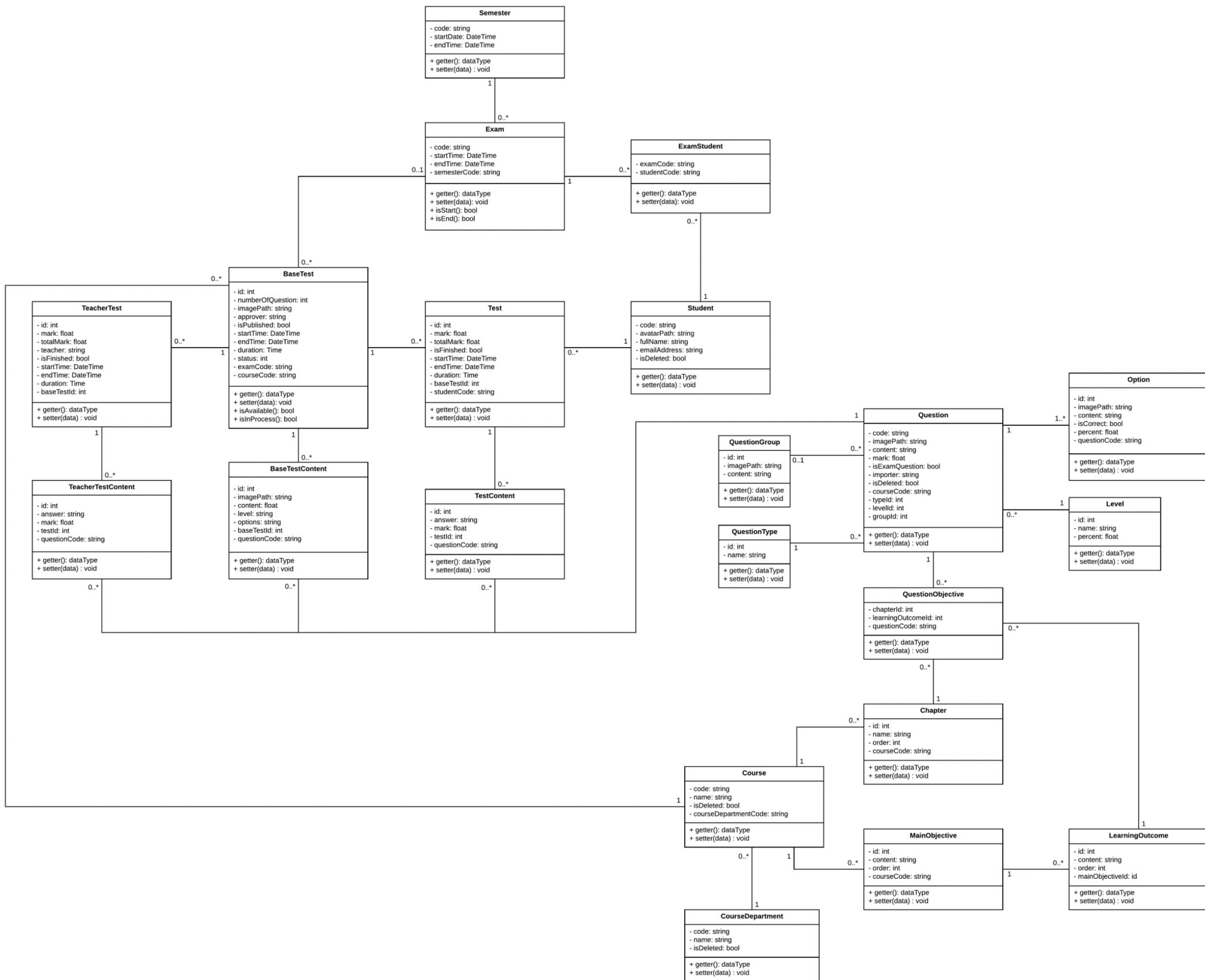


FIGURE 27 CLASS DIAGRAM

Entity Data dictionary: describe all content of all entities		
Class Name	Mapping column with conceptual diagram	Description
BaseTest	BaseTest	Contain the test exam information
Semester	N/A	Contain the Semester information
Exam	Exam	Contain the exam information
ExamStudent	N/A	Contain relationship between Student and Exam, this describe the accessible to the test exam of student
Student	Student	Contain the Student's information
Test	Test	Contain student's Test exam information
TeacherTest	N/A	Contain teacher's test exam information (because teacher need to take exam to approve test exam)
TeacherTestContent	N/A	Contain answer of each teacher's question
BaseTestContent	N/A	Contain test exam's question information
TestContent	N/A	Contain Student's answer information
Question	Question	Contain question information (question bank)
QuestionGroup	N/A	Contain some group question like reading question which contain some question in reading question
Option	N/A	Contain option can choice of each question

QuestionType	N/A	Type of question information (ex: single choice, multiple choice)
Level	N/A	Contain some level of question (hard, easy ...)
QuestionObjective	N/A	Question corresponding with LO
Chapter	N/A	Contain chapter information
Course	N/A	Contain Course information
MainObjective	N/A	Contain Purpose of course information
LearningOutcome	N/A	Decompose of big Main Objective
CourseDepartment	N/A	Contain Course Department information

TABLE 29 CLASS DIAGRAM DESCRIPTION

4.2. Class Diagram Explanation

4.2.1 Semester

Attribute

Attribute	Type	Visibility	Description
Code	string	private	Code of Semester (ex: Fall 2018)
StartDate	DateTime	private	Start Date of semester
EndDate	DateTime	private	End Date of semester

TABLE 30 CLASS SEMESTER EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 31 CLASS SEMESTER METHOD EXPLANATION

4.2.2 Exam

Attribute

Attribute	Type	Visibility	Description
Code	string	private	Code of Exam (ex: FinalX)

StartDate	DateTime	private	Start Date of Exam
EndDate	DateTime	private	End Date of Exam
SemesterCode	string	private	Semester Of that Exam
CourseCode	string	private	Course of that Exam

TABLE 32 CLASS EXAM EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 33 CLASS EXAM METHOD EXPLANATION

4.2.3 ExamStudent

Attribute

Attribute	Type	Visibility	Description
ExamCode	string	private	Code of Exam (ex FinalX)
StudentCode	string	private	Code of a Student (ex SE61801)

TABLE 34 CLASS EXAMSTUDENT EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 35 CLASS EXAMSTUDENT METHOD EXPLANATION

4.2.4 Student

Attribute

Attribute	Type	Visibility	Description
Code	string	private	Code of a Student (ex SE61801)
AvatarPath	string	private	Relative image path of Student Avatar
FullName	string	private	Student's Full Name

EmailAddress	string	private	Student's Email
isDeleted	boolean	private	Flag to know is this student deleted (true = is deleted)

TABLE 36 CLASS STUDENT EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 37 CLASS STUDENT METHOD EXPLANATION

4.2.5 BaseTest

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Base Test Id
NumberOfQuestion	int	private	Number question of this Test Exam
ImagePath	string	private	Relative folder storage image of base test.
ExamCode	string	private	Exam of that base test
CourseCode	string	private	Course of that base test
StartTime	DateTime	private	Start time for student take test exam
EndTime	DateTime	private	End time for student take test exam
Status	int	private	Status of that base test (approved, rejected, Edited...)
Approver	string	private	The people who approve quality of that base test
IsPublished	boolean	private	Is that base test published for student

TABLE 38 CLASS BASETEST EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 39 CLASS BASETEST METHOD EXPLANATION

4.2.6 BaseTestContent

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Question Id of base test
ImagePath	string	private	Relative folder storage image of base test content.
Content	string	private	Content of that question
Level	string	private	Level of question (describe how question hard)
BaseTestId	int	private	Base test of that question
QuestionCode	string	private	Code of question (ex PRX-L01)
Options	string	private	Json string list option can choice in that question

TABLE 40 CLASS BASETESTCONTENT EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 41 CLASS BASETESTCONTENT METHOD EXPLANATION

4.2.7 Test

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Id Student's Test Exam
Mark	double	private	Mark's Student Test Exam
TotalMark	double	private	Total mark of that test exam

isFinished	boolean	private	Is student done this test exam
StudentCode	string	private	Student who take this test exam
StartTime	DateTime	private	Time when student take test exam
EndTime	DateTime	private	End time when student take test exam
Duration	Time	private	Time of student take that test exam
BaseTestId	int	private	Base test of that test exam

TABLE 42 CLASS TEST EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 43 CLASS TEST METHOD EXPLANATION

4.2.8 TestContent

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Id Student's answer
TestId	int	private	Test id of this Student's answer
QuestionCode	double	private	Question code of this Student's answer
Answers	string	private	Json string list student answer of this question
Mark	double	private	Mark of this answer in this question

TABLE 44 CLASS TESTCONTENT EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value

Setter	Void	private	Set attribute Value
--------	------	---------	---------------------

TABLE 45 CLASS TESTCONTENT METHOD EXPLANATION

4.2.9 Question

Attribute

Attribute	Type	Visibility	Description
Code	string	private	Question Code
ImagePath	string	private	Relative path storage image of question
Content	string	private	Question content
LevelId	int	private	Level of question (hard, easy...)
GroupId	int	private	Mark of this answer in this question
IsExamQuestion	boolean	private	Is question already in test exam
CourseCode	string	private	Course of this question
Mark	double	Private	Mark of this question
IsDeleted	boolean	Private	Is this question deleted
TypeId	int	Private	Type of question (ex: single choice, multiple choice, matching...)
importer	string	Private	The teacher who import this question

TABLE 46 CLASS QUESTION EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 47 CLASS QUESTION METHOD EXPLANATION

4.2.10 Option

Attribute

Attribute	Type	Visibility	Description
-----------	------	------------	-------------

Id	int	private	Id of this option
ImagePath	string	private	Relative path storage image of option
Content	string	private	Option content
IsCorrect	boolean	private	Is this option correct
QuestionCode	string	private	Question Code
Percent	double	Private	Percent will Minus if choice wrong

TABLE 48 CLASS OPTION EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 49 CLASS OPTION METHOD EXPLANATION

4.2.11 Level

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Level Id
Name	string	private	Level name (ex: hard, easy)
Percent	Float	private	How important the question is

TABLE 50 CLASS LEVEL EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 51 CLASS LEVEL METHOD EXPLANATION

4.2.12 QuestionType

Attribute

Attribute	Type	Visibility	Description

Id	int	private	Type Id
Name	string	private	Type name (ex: single choice, multiple choice)

TABLE 52 CLASS QUESTIONTYPE EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 53 CLASS QUESTIONTYPE METHOD EXPLANATION

4.2.13 QuestionGroup

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Question Group Id
ImagePath	string	private	Relative Path of this question group
Content	String	private	Content of this question group

TABLE 54 CLASS QUESTIONGROUP EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 55 CLASS QUESTIONGROUP METHOD EXPLANATION

4.2.14 QuestionObjective

Attribute

Attribute	Type	Visibility	Description
ChapterId	int	private	Chapter Id
LearningOutcomeId	int	private	Learning Outcome Id
QuestionCode	string	private	Question

TABLE 56 CLASS QUESTIONOBJECTIVE EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

*TABLE 57 CLASS QUESTIONOBJECTIVE METHOD EXPLANATION***4.2.15 Chapter****Attribute**

Attribute	Type	Visibility	Description
Id	int	private	Chapter Id
Name	string	private	Name of chapter (ex: chapter 1)
CourseCode	string	private	Code of course
Order	int	private	Order of chapter (1, 2...)

*TABLE 58 CLASS CHAPTER EXPLANATION***Method**

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

*TABLE 59 CLASS CHAPTER METHOD EXPLANATION***4.2.16 Course****Attribute**

Attribute	Type	Visibility	Description
Code	string	private	Course Code
Name	string	private	Course Name
CourseDepartmentCode	string	private	Department of course
IsDeleted	boolean	private	Is this course deleted

*TABLE 60 CLASS COURSE EXPLANATION***Method**

Attribute	Return Type	Visibility	Description

Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 61 CLASS COURSE METHOD EXPLANATION

4.2.17 MainObjective

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Main Objective Id
Content	string	private	Main objective information
CourseCode	string	private	Course of this main objective
Order	int	private	Order of this main objective

TABLE 62 CLASS MAINOBJECTIVE EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 63 CLASS MAINOBJECTIVE METHOD EXPLANATION

4.2.18 LearningOutcome

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Learning Outcome Id
Content	string	private	Learning outcome information
CourseCode	string	private	Big main objective
Order	int	private	Order of this Learning outcome

TABLE 64 CLASS LEARNINGOUTCOME EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 65 CLASS LEARNINGOUTCOME METHOD EXPLANATION

4.2.19 TeacherTest

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Teacher Test Id
Mark	float	private	Mark of this test exam
TotalMark	Float	private	Total mark of this test exam
Teacher	string	private	Get from FPT Authenticate Service
IsFinished	boolean	private	Describe: is this test exam finished
StartTime	DateTime	private	Time teacher start to take test exam
EndTime	DateTime	private	Time teacher End take test exam
Duration	Time	private	Duration teacher take that test exam
BaseTestId	int	Private	Base test id

TABLE 66 CLASS TEACHERTEST EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 67 CLASS TEACHERTEST METHOD EXPLANATION

4.2.20 TeacherTestContent

Attribute

Attribute	Type	Visibility	Description
Id	int	private	Teacher Test Content Id
Answer	string	private	Answer of question
Mark	Float	private	Mark of this answer
TestId	int	private	Test Exam of this answer

QuestionCode	boolean	private	Question of this answer
--------------	---------	---------	-------------------------

TABLE 68 CLASS TEACHERTESTCONTENT EXPLANATION

Method

Attribute	Return Type	Visibility	Description
Getter	Attribute Type	private	Get attribute value
Setter	Void	private	Set attribute Value

TABLE 69 CLASS TEACHERTESTMETHOD EXPLANATION

4.3. Interaction Diagram

4.3.1 Activity Diagram

4.3.1.1 Teacher Import Question

Summary: This diagram shows how Teacher import question

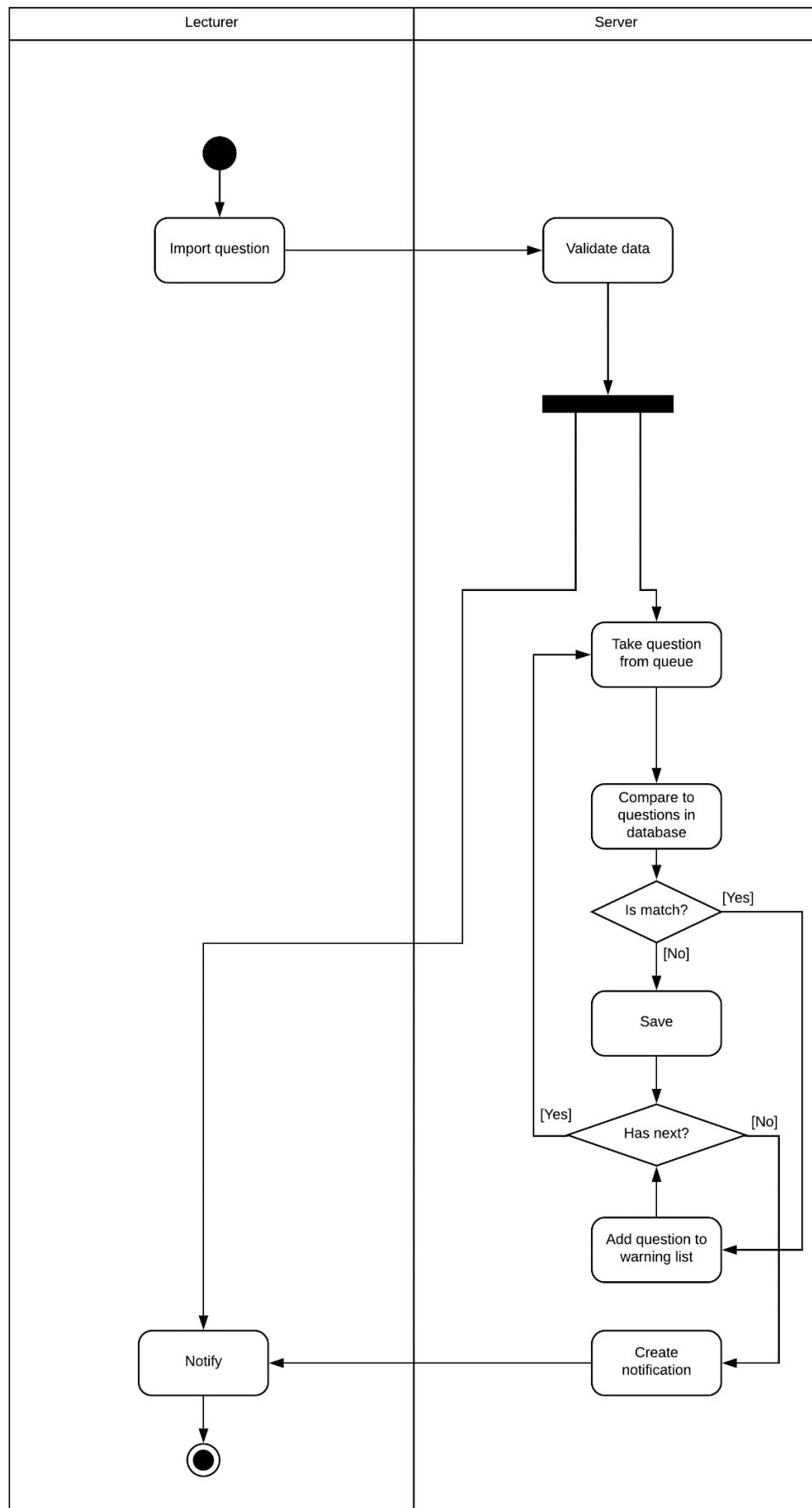


FIGURE 28 ACTIVITY DIAGRAM TEACHER IMPORT QUESTION

4.3.1.2 Generate Test Exam

Summary: This diagram shows how Staff Create Test Exam

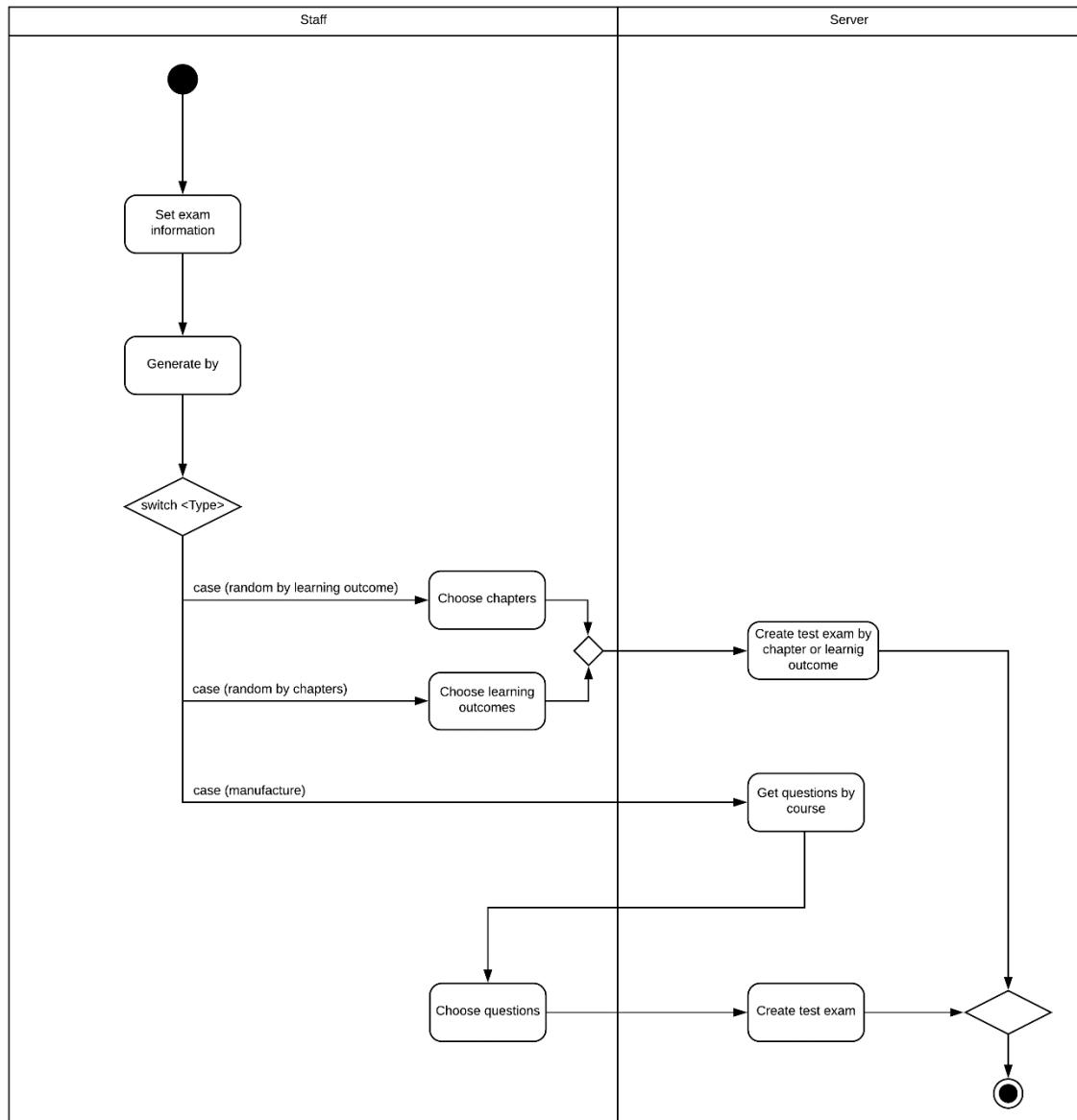


FIGURE 29 ACTIVITY DIAGRAM TRAINING DEPARTMENT STAFF GENERATE TEST EXAM

4.3.1.3 Teacher Approve Test Exam

Summary: This diagram shows how Teacher approve test exam

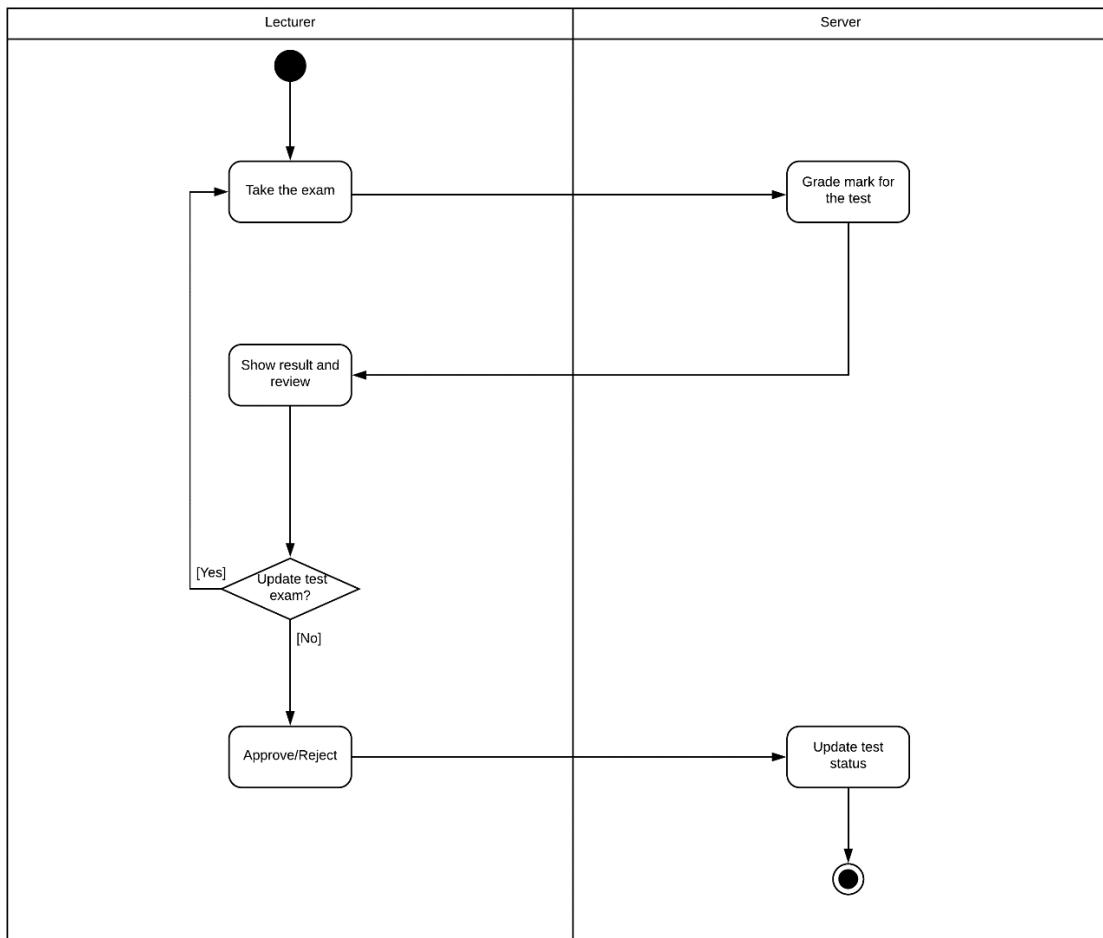


FIGURE 30 ACTIVITY DIAGRAM APPROVE TEST EXAM

4.3.1.4 Student Take a Test Exam

Summary: This diagram shows how Student Take a Test Exam

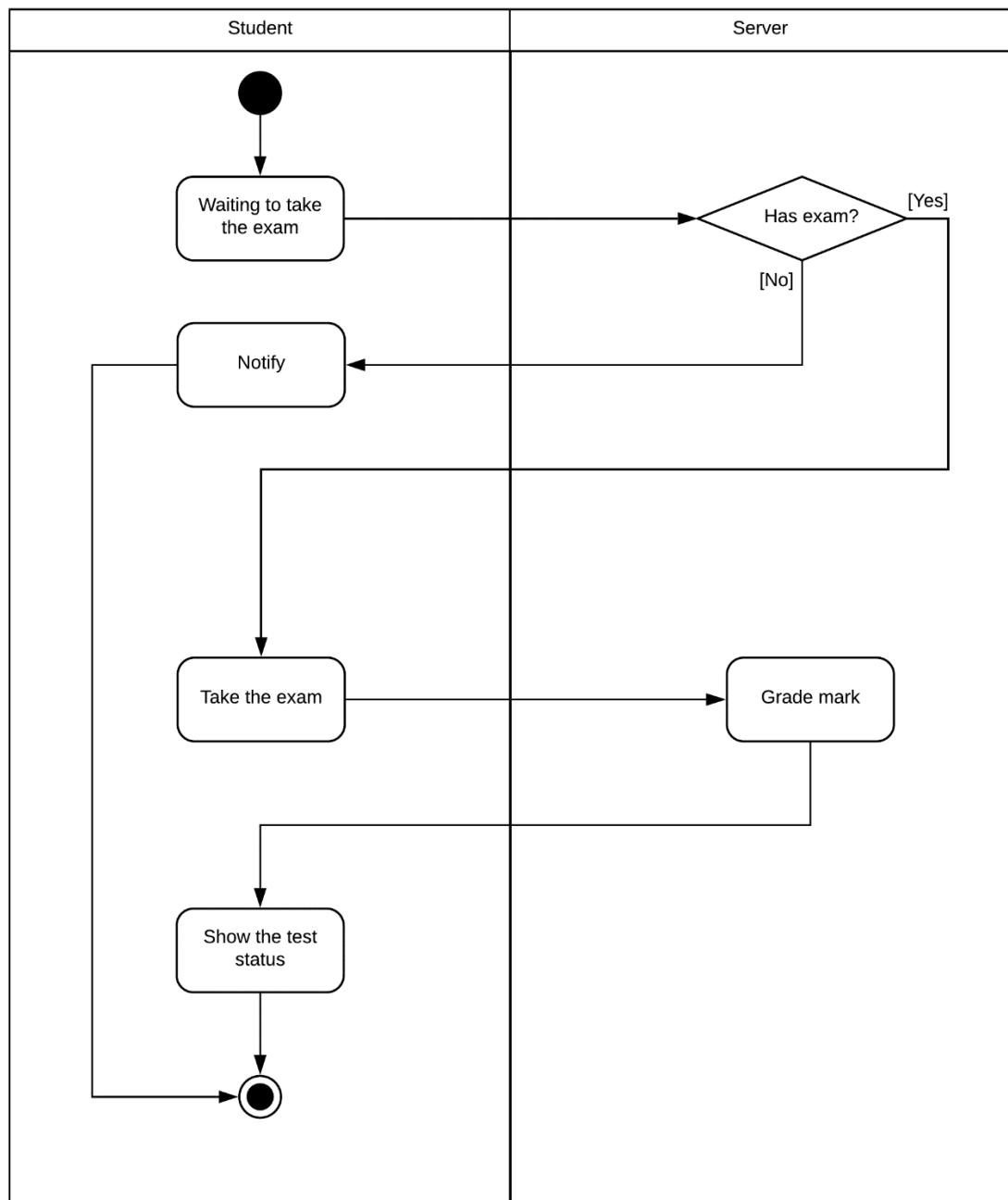


FIGURE 31 ACTIVITY DIAGRAM STUDENT TAKE A TEST EXAM

5. Interface

5.1. Manage Question Bank

The screenshot shows a web-based application titled "Exam Management". The main header includes "Exam Management", "Logout", and "Import Question". Below the header, there are tabs: "Manage Question Bank" (selected), "Manage Question", "Manage Question Bank", and "Manage Question". A search bar at the top left contains the placeholder "Search" and a dropdown menu for "Course Name or Question". On the right side, there is a red-bordered button labeled "Import Question". The main content area displays a table of questions with columns: Action, Mark, Grade, Course, and ID. Each row has a red-bordered "Action" column containing two icons: a red square with a white minus sign and a green square with a white checkmark. The table includes a header row and 10 data rows. At the bottom, there is a navigation bar with buttons for "First", "Previous", "Next", "Last", and "Refresh".

FIGURE 32 MANAGE QUESTION BANK

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	SearchText	Fill in question code or question content to filter question list	No	No	Text	String	0-255 characters

TABLE 70 MANAGE QUESTION BANK FIELDS

Buttons

No	Function	Description	Validation	Outcome
2	RedirectToImportQuestion	Go to import question page	N/A	Move to import question page
3	EditQuestion	Edit current question	N/A	Show a popup with question detail
4	DeleteQuestion	Delete current question	N/A	Delete and refresh the question list

TABLE 71 MANAGE QUESTION BANK BUTTONS

5.2. Import question

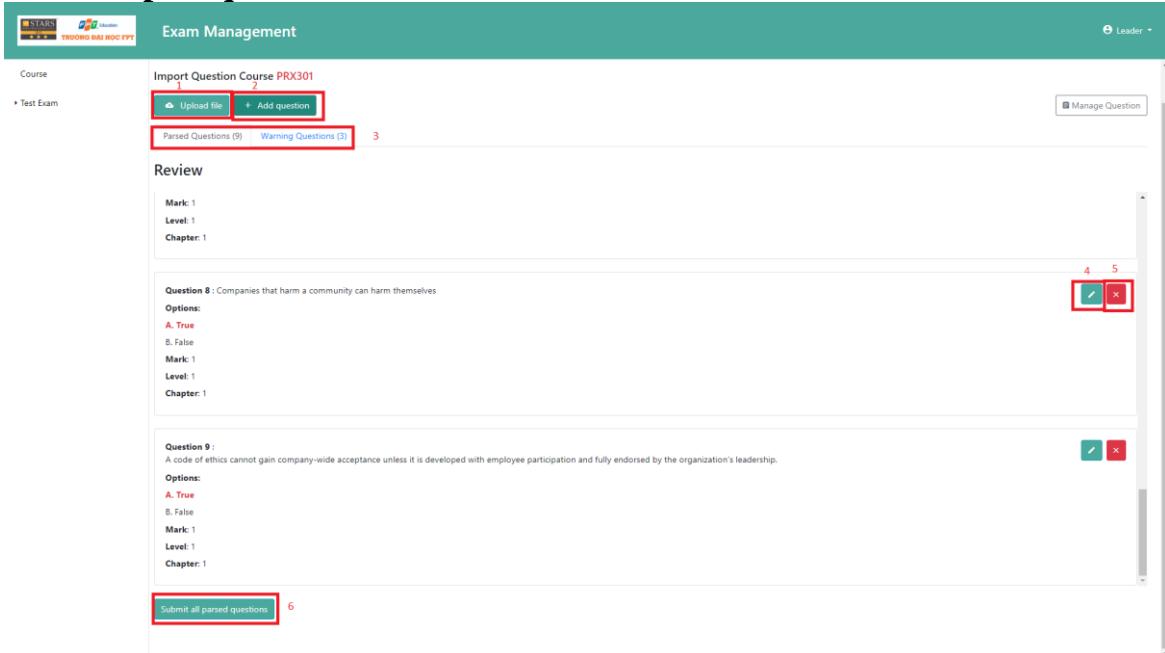


FIGURE 33 IMPORT QUESTION

Buttons

No	Function	Description	Validation	Outcome
1	UploadFile	Upload a file	File must be XML, docx or txt (Gift Format).	Parsed file to question list
2	AddQuestion	Add a question by using form	N/A	Show a popup
3	ChangeTab	Show all parsed questions successful or show all invalid questions	N/A	Show content of tab is chosen
4	EditQuestion	Edit current question	N/A	Show a popup with question detail
5	DeleteQuestion	Delete current question	N/A	Delete question in the question list
6	Submit	Send question list to server to matching	Question list cannot be emptied	Validate and start marching question.

TABLE 72 IMPORT QUESTION BUTTONS

5.3. Matching question

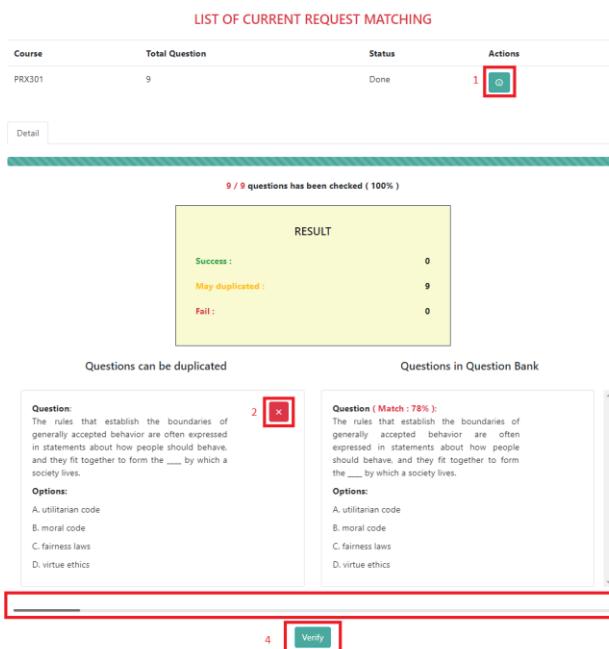


FIGURE 34 MATCHING QUESTION

Buttons

No	Function	Description	Validation	Outcome
1	ShowDetail	Show detail of current process	N/A	Show detail of process below the table
2	DeleteQuestion	Delete question from list	Current process must finish matching.	Delete question.
3	MoveNext/MovePrev	Drag to move next or move previous question in the list	Current process must finish matching	Show next or previous question.
4	Verify	Send question list to server	Current process must finish matching	Save the question list to question bank

TABLE 73 MATCHING QUESTION

5.4. Create Test Exam

The screenshot shows a software interface titled 'Exam Management'. On the left, a sidebar lists navigation options: Semester (Manage Semester, Manage Exam), Course (Select a Course), and Test Exam (Create Test Exam, Manage Test Exam, Review Test Exam). The main area displays a horizontal progress bar with five steps: 'FIND COURSE' (green dot), 'EXAM DETAIL' (white circle), 'SETTING LEARNING OUTCOMES OR CHAPTERS' (white circle), 'EXAM CONTENT' (white circle), and 'FINISH' (white circle). Below the progress bar, a red box highlights a dropdown menu labeled 'Course' with the sub-option '1 Select a Course'. To the right of the dropdown is a green 'Next' button with the number '2' above it.

FIGURE 35 CREATE TEST EXAM

This screenshot is identical to Figure 35, showing the 'Exam Management' interface. The sidebar and progress bar are the same. A red box highlights the 'Course' dropdown menu with '1 Select a Course' selected. The green 'Next' button to its right is also highlighted with a red box and has the number '2' above it.

FIGURE 36 CREATE TEST EXAM

Exam Management

FIND COURSE EXAM DETAIL SETTING LEARNING OUTCOMES OR CHAPTERS EXAM CONTENT FINISH

Course : PRX301

Number of Questions 3 Duration (minute) 4

1 1

Number of Tests 5

2

If you generate exam using Random feature, u may pass this field.

Previous 15 Next

FIGURE 37 CREATE TEST EXAM

Exam Management

FIND COURSE EXAM DETAIL SETTING LEARNING OUTCOMES OR CHAPTERS EXAM CONTENT FINISH

Learning Outcome Chapter

List of Chapter

Chapter 1
Chapter 2
Chapter 3
Chapter 4
Chapter 5
Chapter 6
Chapter 7

Random base on Chapter Custom Exam Content

Previous Next

FIGURE 38 CREATE TEST EXAM

Exam Management

- Semester
 - Manage Semester
 - Manage Exam
- Course
 - Test Exam**
 - Create Test Exam
 - Manage Test Exam
 - Review Test Exam

FIND COURSE EXAM DETAIL SETTING LEARNING OUTCOMES OR CHAPTERS EXAM CONTENT FINISH

Content of Exam 0/10

List of question: 11

Question level 1:

12

13 Load more...

14 Finish

2 Next

FIGURE 39 CREATE TEST EXAM

Exam Management

- Semester
 - Manage Semester
 - Manage Exam
- Course
 - Test Exam**
 - Create Test Exam
 - Manage Test Exam
 - Review Test Exam

FIND COURSE EXAM DETAIL SETTING LEARNING OUTCOMES OR CHAPTERS EXAM CONTENT FINISH

Course : PRX301

Number of Test : 2

Number of questions : 10

Duration : 00:10:00

14 Finish

FIGURE 40 CREATE TEST EXAM

Fields

N o	Field Name	Description	Rea d only	Mandator y	Contro l Type	Data Typ e	Lengt h
1	CourseCode	Fill a code course	No	Yes	Select	String	1
3	NumberOfQuestions	Fill number of questions in the test exam	No	Yes	EditText	Integer	1-100
4	Duration	Fill duration of a test exam	No	Yes	EditText	Integer	1-1439

5	NumberOfTest	Fill number of test exam will be generated	No	Yes	EditText	Integer	1-10
11	Level	Fill in level to filter question list of a chapter/learning outcome	No	No	Select	Integer	1

TABLE 74 CREATE TEST EXAM FIELDS

Buttons

No	Function	Description	Validation	Outcome
2	NextStep	Move to next step	N/A	Move to next step
6	CheckAll	Select all chapters or learning outcome	No	Select all chapters or learning outcomes
7	Check	Select a chapter or learning outcome	No	Select a chapter or learning outcome
8	RandomTestExam	Random test base on NumberOfTest and chapter list or learning outcome list.	N/A	Show a popup with question detail
9	SetList	Delete current question	Chapters list or Learning Outcome list cannot be emptied	Delete and refresh the question list
10	ChangeTab	Show all chapters or learning outcomes	N/A	Show content of tab is chosen
12	SetQuestion	Drag to left column to add this question to test exam content	Test Exam Content is lower than NumberOfQuestions	Add question to test exam content
13	LoadMore	Load next page and merge to question list of current chapter or learning outcome.	N/A	Show more questions.
14	Finish	Move back to step one	N/A	Move back to step one
15	PreviousStep	Move to previous step	N/A	Move to previous step

TABLE 75 CREATE TEST EXAM BUTTONS

5.5. Test Exam List with “Pending” Status

The screenshot shows a web-based application titled "Exam Management". In the top right corner, there is a user profile icon with the email "Khanhkt@fe.edu.vn". The main content area displays a table of test exams. The table has columns: Course, Start Date, End Date, Duration, Status, and Actions. A single row is visible, representing a test exam for course PRX301. The start date is 09/12/2018 - 07:57, and the end date is 10/12/2018 - 07:57. The duration is 00:10:00, and the status is "Pending". The "Actions" column contains a blue button with a white circle icon, which is highlighted with a red rectangular box.

FIGURE 41 TEST EXAM LIST WITH "PENDING" STATUS

Buttons

No	Function	Description	Validation	Outcome
1	Detail	View detail of test exam	Start Date of test exam must greater than or equal current time. End Date of test exam must lower than or equal current time.	Go to “Detail” page

TABLE 76 TEST EXAM LIST WITH “PENDING” STATUS BUTTONS

5.6. Take an Exam

The screenshot shows a web-based exam management system. At the top, there are logos for STARS and TRƯỜNG ĐẠI HỌC FPT. The title "Exam Management" is centered. On the right, there is a "Lecturer" dropdown menu. Below the title, there's a "Course" section with a "Test Exam" link. A large central area displays a question and its code. The question is labeled "Question 1" and asks about ethical focus. It lists four options: "Virtue ethics", "Utilitarian", "Common good", and "Fairness". A red box highlights the "Submit" button at the bottom left of the question area. To the right, another question is partially visible, labeled "Question 2", which contains a block of JavaScript code.

FIGURE 42 TAKE AN EXAM

This screenshot shows a different part of the Exam Management interface. At the top, it has the same header with STARS and TRƯỜNG ĐẠI HỌC FPT logos, "Exam Management" title, and "Lecturer" dropdown. The "Course" and "Test Exam" sections are present. A question is displayed with a red box around the "Edit Question" and "Switch Question" buttons in the top right corner. The question asks when an XPathException is used and lists five options: A. The XPath instance is created with the newXPath method, B. All of the others, C. The evaluate method of XPath class is used, D. The XPathFactory is created with the newInstance method of XPathFactory, and E. None of the above. Below this, another question asks which statement is false and lists five options: A. Objects do not have the property of information hiding, B. None of the others, C. Characters are the building blocks of JavaScript programs, D. Browsers have a set of objects that encapsulate the HTML document, and E. Programs communicate with objects by using well-defined interfaces. At the bottom, there are "Approve" and "Reject" buttons, with a red box around the "Approve" button.

FIGURE 43 TAKE AN EXAM

Exam Management

Course: Test Exam

Question: Consider the following XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<nsl:Company1 xmlns:nsl="http://www.acompany.com/" xmlns="http://www.bcompany.com/">
<Company2>
<Company3 xmlns="http://www.ccompany.com/">
<Company4 ID="4"/>
</Company3>
</Company2>
</nsl:Company1>
```

Which of the following statements is true?

- A. The usage of namespace is not correct, default namespace can't be changed
- B. None of the others
- C. Namespace of Element Company2 belongs to http://www.acompany.com/
- D. Namespace of Element Company3 belongs to http://www.bcompany.com/
- E. Namespace of Element Company4 belongs to http://www.ccompany.com/

Question: Which of the following two statements must be used to validate an XML instance document with SAX?

- A. parser.setFeature("http://xml.org/sax/features/validation", true);
- B. XMLReader parser = new XMLReader();
- C. XMLParser parser = XMLParserFactory.createXMLParser();
- D. XMLReader parser = XMLReaderFactory.createXMLReader();
- E. parser.setProperty("http://xml.org/sax/features/validation", true);

7 **Retake Exam**

FIGURE 44 TAKE AN EXAM

Swap Question

Course: Test Exam

Code	Course	Level	Mark	Action
PRX301-L00-Q1	PRX301	1	1	Swap with current question
PRX301-L00-Q10	PRX301	1	1	
PRX301-L00-Q100	PRX301	1	1	
PRX301-L00-Q101	PRX301	1	1	
PRX301-L00-Q102	PRX301	1	1	
PRX301-L00-Q103	PRX301	1	1	
PRX301-L00-Q105	PRX301	1	1	
PRX301-L00-Q106	PRX301	1	1	
PRX301-L00-Q107	PRX301	1	1	

8 Course Name or Question **Search**

9 10

Previous 1 2 3 4 5 ... 18 Next

Approve Reject

FIGURE 45 TAKE AN EXAM

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
8	SearchText	Fill in question code or question content to filter question list	No	No	Text	String	0-255 characters

TABLE 77 TAKE AN EXAM FIELDS

Buttons

No	Function	Description	Validation	Outcome
----	----------	-------------	------------	---------

1	Submit	Send result of exam to server	N/A	Send result to server to grade test exam
2	ExpandMenu	Open menu which contain edit question and swap question	N/A	Show a small menu.
4	ShowEditModal	Show a popup which contain current question	N/A	Show a pop up
5	ShowSwapModal	Show a popup which contain all questions of current course	N/A	Show a pop up
6	ChangeStatus	Change status test exam to approve/reject	N/A	Status of test exam change to approve/reject
7	Retake	Take a test exam again	N/A	Change status test exam to pending and take a test exam again
9	ShowInformation	Show information of question	N/A	An information of question is expanded below the question row in the table.
10	SwapQuestion	Swap question in question bank with current question	N/A	Test exam content has been changed and status exam change to edited, button retake exam is shown and hide approve/reject button

TABLE 78 TAKE AN EXAM BUTTONS

5.7. Create Exam

The screenshot shows the 'CREATE' exam setup interface. On the left, a sidebar lists navigation options: Semester, Manage Semester; Course, Manage Exam; Test Exam, Create Test Exam, Manage Test Exam, Review Test Exam. The main area is titled 'CREATE' and contains the following fields:

- Semester: FALL_2018 (highlighted by a red box)
- Course: PRX01 (highlighted by a red box)
- Base Test: PRX01-277 (highlighted by a red box)
- Start Date: 09/12/2018 08:25 (highlighted by a red box)
- End Date: 09/12/2018 08:25 (highlighted by a red box)

To the right, there is a 'Student List' table with columns: No., Student Code, FullName. A red box highlights the 'Add' button (+) in the top right corner of the table header. At the bottom right of the interface, there is a 'Save' button (highlighted by a red box) and a 'Cancel' button.

FIGURE 46 CREATE EXAM

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	SemesterCode	Fill semester code	No	Yes	Text	String	0-50 characters
2	CourdeCode	Fill course code	No	Yes	Text	String	0-50 characters
3	BaseTest	Fill basetest	No	Yes	Text	String	0-20 characters

TABLE 79 CREATE EXAM FIELDS

Buttons

No	Function	Description	Validation	Outcome
4	StartDate	Select a date	Pick a date which equal or greater than current date	A date is set
5	EndDate	Select a date	Pick a date which equal or greater than start date	A date is set
6	ImportFile	Select a student file.	File must be excel format and have 2 column Fullname and Membercode	A student list will be add to form
7	Submit	Validate all field and create an exam.	Validate all fields	Send request to create an exam

TABLE 80 CREATE EXAM BUTTONS

5.8. Set Approver

FIGURE 47 SET APPROVER

Fields

No	Field Name	Description	Read only	Mandatory	Control Type	Data Type	Length
1	Approver	Fill approver	No	Yes	Text	String	0-50 characters

TABLE 81 SET APPROVER FIELDS

Buttons

No	Function	Description	Validation	Outcome
2	StartDate	Select a date	Pick a date which equal or greater than current date	A date is set
3	EndDate	Select a date	Pick a date which equal or greater than start date	A date is set
4	Submit	Validate all field and create an exam.	Validate all fields	Send request to create an exam

TABLE 82 SET APPROVER BUTTONS

6. Database Design

6.1. Entity Relationship Diagram

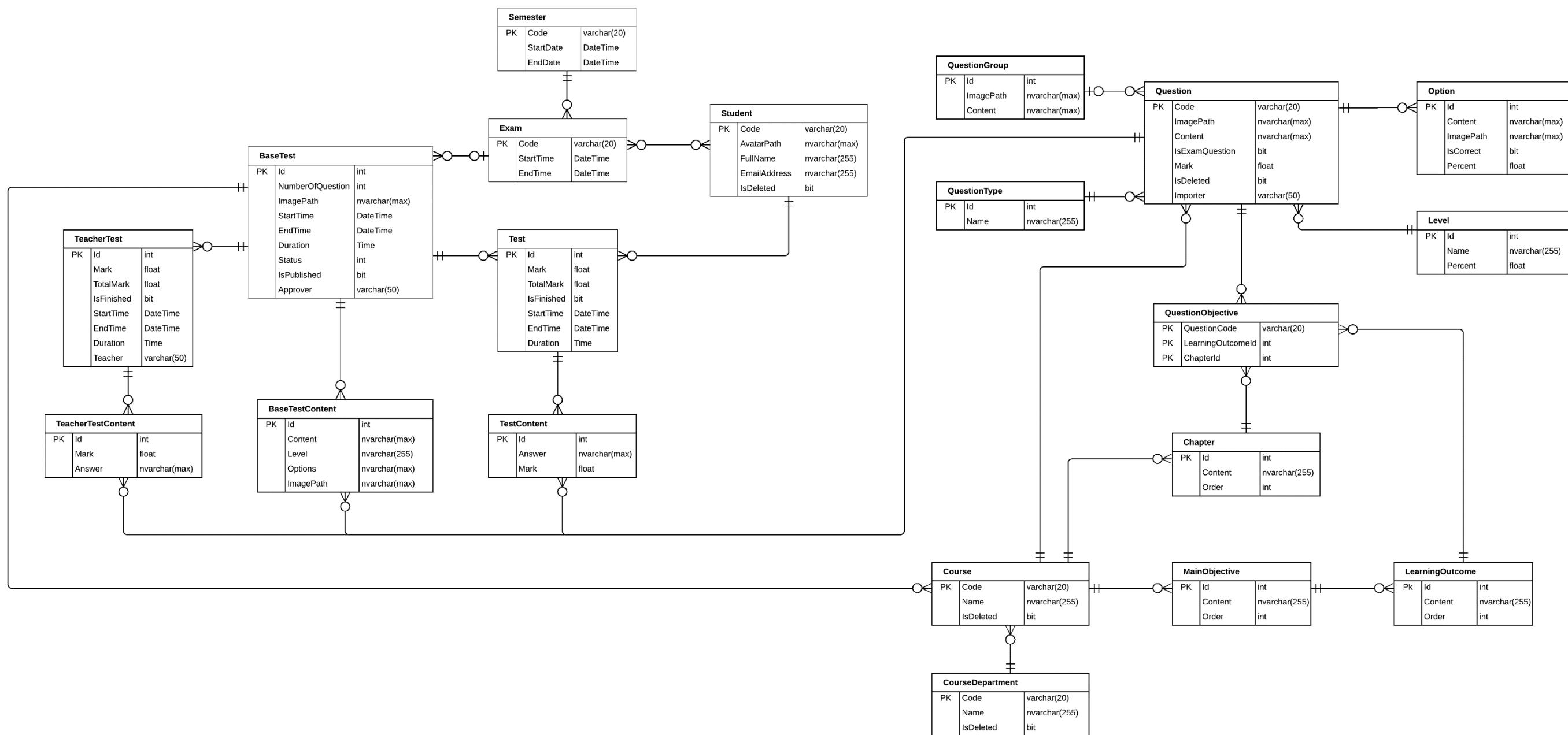


FIGURE 48 ER DIAGRAM

6.2. Data Dictionary

Entity Name	Description
BaseTest	Contain the test exam information
Semester	Contain the Semester information
Exam	Contain the exam information
ExamStudent	Contain between Student and Exam, it describe the accessible to the test exam of student
Student	Contain the Student information
Test	Contain student Test exam information
TeacherTest	Contain teacher test exam information (because teacher need to take exam to approve test exam)
TeacherTestContent	Contain answer of each teacher's question
BaseTestContent	Contain test exam's question information
TestContent	Contain Student's answer information
Question	Contain question information (question bank)
QuestionGroup	Contain some group question like reading question which contain some question in reading question
Option	Contain option can choice of each question
QuestionType	Type of question information (ex: single choice, multiple choice)
Level	Contain some level of question (hard, easy...)
QuestionObjective	Question corresponding with LO
Chapter	Contain chapter information
Course	Contain Course information
MainObjective	Contain Purpose of course information
LearningOutcome	Decompose of big Main Objective
CourseDepartment	Contain Course Department information

TABLE 83 ERD DICTIONARY

6.3. Class Diagram Explanation

6.3.1 Semester

Attribute	Type	Description
-----------	------	-------------

Code	varchar(20)	Code of Semester (ex: Fall 2018)
StartDate	Datetime	Start Date of semester
EndDate	Datetime	End Date of semester

TABLE 84 SEMESTER TABLE DESCRIPTION

6.3.2 Exam

Attribute	Type	Description
Code	varchar(20)	Code of Exam (ex: FinalX)
StartDate	Datetime	Start Date of Exam
EndDate	Datetime	End Date of Exam

TABLE 85 EXAM TABLE DESCRIPTION

6.3.3 Student

Attribute	Type	Description
Code	varchar(20)	Code of a Student (ex SE61801)
AvatarPath	nvarchar(Max)	Relative image path of Student Avatar
FullName	nvarchar(255)	Student's Full Name
EmailAddress	nvarchar(255)	Student's Email
IsDeleted	bit	Flag to know is this student deleted (true = is deleted)

TABLE 86 STUDENT TABLE DESCRIPTION

6.3.4 BaseTest

Attribute	Type	Description
Id	int	Base Test Id
NumberOfQuestion	int	Number question of this Test Exam
ImagePath	nvarchar(max)	Relative folder storage image of base test.
StartTime	DateTime	Start time for student take test exam
EndTime	DateTime	End time for student take test exam

Duration	Time	Duration of test exam
Status	int	Status of that base test (approved, rejected, edited...)
Approver	varchar(50)	The people who approve quality of that base test
IsPublished	bit	Is that base test published for student

TABLE 87 BASETEST TABLE DESCRIPTION

6.3.5 BaseTestContent

Attribute	Type	Description
Id	int	Question Id of base test
ImagePath	nvarchar(Max)	Relative folder storage image of base test content.
Content	nvarchar(max)	Content of that question
Level	nvarchar(255)	Level of question (describe how question hard)
Options	nvarchar(max)	Json string list option can choice in that question

TABLE 88 BASETESTCONTENT TABLE DESCRIPTION

6.3.6 Test

Attribute	Type	Description
Id	int	Id Student's Test Exam
Mark	float	Mark's Student Test Exam
TotalMark	float	Total mark of that test exam
IsFinished	bit	Is student done this test exam
StartTime	DateTime	Time when student take test exam
EndTime	DateTime	End time when student take test exam
Duration	Time	Time of student take that test

TABLE 89 TEST TABLE DESCRIPTION

6.3.7 TestContent

Attribute	Type	Description
Id	Int	Id Student's answer
Answer	nvarchar(max)	Json string list student answer of this question
Mark	Float	Mark of this answer in this question

TABLE 90 TESTCONTENT TABLE DESCRIPTION

6.3.8 Question

Attribute	Type	Description
Code	nvarchar(20)	Question Code
ImagePath	nvarchar(max)	Relative path storage image of question
Content	nvarchar(max)	Question content
IsExamQuestion	bit	Is question already in test exam
Mark	float	Mark of this question
IsDeleted	bit	Is this question deleted
Importer	varchar(50)	The people who import this question

TABLE 91 QUESTION TABLE DESCRIPTION

6.3.9 Option

Attribute	Type	Description
Id	int	Id of this option
ImagePath	nvarchar(max)	Relative path storage image of option
Content	nvarchar(max)	Option content
IsCorrect	bit	Is this option correct
Percent	float	Percent will Minus if choice wrong

TABLE 92 OPTION TABLE DESCRIPTION

6.3.10 Level

Attribute	Type	Description
Id	Int	Level Id
Name	nvarchar(255)	Level name (ex: hard, easy)
Percent	float	How important it is

TABLE 93 LEVEL TABLE DESCRIPTION

6.3.11 QuestionType

Attribute	Type	Description
Id	Int	Type Id
Name	nvarchar(255)	Type name (ex: single choice, multiple choice)

TABLE 94 QUESTIONTYPE TABLE DESCRIPTION

6.3.12 QuestionGroup

Attribute	Type	Description
Id	int	QuestionGroup Id
ImagePath	nvarchar(max)	Relative Path of this question group
Content	nvarchar(max)	Content of this question group

TABLE 95 QUESTIONGROUP TABLE DESCRIPTION

6.3.13 QuestionObjective

Attribute	Type	Description
ChapterId	int	Chapter Id
LearningOutcomeId	int	Learning Outcome Id
QuestionCode	varchar(20)	Question

TABLE 96 QUESTIONOBJECTIVE TABLE DESCRIPTION

6.3.14 Chapter

Attribute	Type	Description
Id	Int	Chapter Id
Name	nvarchar(255)	Name of chapter (ex: chapter 1)
Order	int	Order of chapter (1, 2...)

TABLE 97 CHAPTER TABLE DESCRIPTION

6.3.15 Course

Attribute	Type	Description
Code	Int	Course Code
Name	nvarchar(255)	Course Name
IsDeleted	bit	Is this course deleted

TABLE 98 COURSE TABLE DESCRIPTION

6.3.16 MainObjective

Attribute	Type	Description
Id	Int	MainObjective Id
Content	nvarchar(max)	Main objective information
Order	int	Order of this main objective

TABLE 99 MAINOBJECTIVE TABLE DESCRIPTION

6.3.17 LearningOutcome

Attribute	Type	Description
Id	Int	Learning Outcome Id
Content	nvarchar(max)	Learning outcome information
Order	int	Order of this Learning outcome

TABLE 100 LEARNINGOUTCOME TABLE DESCRIPTION

6.3.18 Course Department

Attribute	Type	Description
Code	Varchar(20)	Learning Outcome Id
Name	nvarchar(255)	Learning outcome information
IsDeleted	bit	Order of this Learning outcome

TABLE 101 COURSEDEPARTMENT TABLE DESCRIPTION

6.3.19 TeacherTest

Attribute	Type	Description
Id	int	Teacher Test Id

Mark	float	Mark of this test exam of teacher
TotalMark	float	Total mark of test test exam
IsFinished	bit	Is this teacher test finished
StartTime	DateTime	Time teacher start to take the test exam
EndTime	DateTime	Time teacher end the test exam
Duration	Time	Duration teacher done this test exam
Teacher	Varchar(255)	Teacher code get from FPT authentication service

TABLE 102 TEACHERTEST TABLE DESCRIPTION

6.3.20 TeacherTestContent

Attribute	Type	Description
Id	int	Teacher Test Id
Mark	float	Mark of this test exam of teacher
Answer	nvarchar(max)	Answer of teacher test exam

TABLE 103 TEACHERTESTCONTENT TABLE DESCRIPTION

7. Algorithms

7.1. Calculating Percentage Similarity of 2 Question

7.1.1 Problem

When a teacher imports a question list into the question bank, some questions may be duplicated. This can lead to a waste of server's storage.

7.1.2 Solution

In order to solve the mentioned problem, we decide to apply the Levenshtein distance algorithm.

Levenshtein distance algorithm is used to identify the differences between the ranges of 2 sequences. The range between these sequences is the minimum steps to make one sequence become the other one. This algorithm includes 3 changing functions:

- Remove a character
- Add a new character
- Replace with another character

Example: To change "kitten" to "sitting", we have to use at least 3 changing times as following:

1. kitten -> sitten (replace "k" with "s")
2. sitten -> sittin (replace "e" with "i")
3. sittin -> sitting (add "g")

Therefore, the range between "kitten" and sitting" is 3.

The nature of Levenshtein distance algorithm is based on Dynamic Programming. Mathematically, the Levenshtein distance between two strings a, b (of length $|a|$ and $|b|$ respectively) is given by $\text{lev}_{a,b}(|a|, |b|)$, where:

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i - 1, j) + 1 \\ \text{lev}_{a,b}(i, j - 1) + 1 \\ \text{lev}_{a,b}(i - 1, j - 1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

With

- $1_{(a_i \neq b_j)}$ equal to 0 when $a_i = b_j$ and equal to 1 otherwise
- $\text{lev}_{a,b}(i, j)$ is the distance between the first i characters of a and the first j characters of b .

Note that the first element in the minimum corresponds to deletion (from a to b), the second to insertion and the third to match or mismatch, depending on whether the respective symbols are the same.

Here is a straightforward pseudo code for a function called LevenshteinDistance that takes two strings:

- s of length m

- t of length n

And returns the Levenshtein distance between them:

```
function LevenshteinDistance(char s[1..m], char t[1..n]):  
    // for all i and j, d[i,j] will hold the Levenshtein distance between  
    // the first i characters of s and the first j characters of t  
    // note that d has (m+1)*(n+1) values  
    declare int d[0..m, 0..n]  
  
    set each element in d to zero  
  
    // source prefixes can be transformed into empty string by  
    // dropping all characters  
    for i from 1 to m:  
        d[i, 0] := i  
  
    // target prefixes can be reached from empty source prefix  
    // by inserting every character  
    for j from 1 to n:  
        d[0, j] := j  
  
    for j from 1 to n:  
        for i from 1 to m:  
            if s[i] = t[j]:  
                substitutionCost := 0  
            else:  
                substitutionCost := 1  
            d[i, j] := minimum(d[i-1, j] + 1,           // deletion  
                               d[i, j-1] + 1,           // insertion  
                               d[i-1, j-1] + substitutionCost) // substitution  
  
    return d[m, n]
```

Two examples of the resulting matrix (hovering over a tagged number reveals the operation performed to get that number):

		k	i	t	t	e	n	
	0	1	2	3	4	5	6	
s	1	1	2	3	4	5	6	
i	2	2	1	2	3	4	5	
t	3	3	2	1	2	3	4	
t	4	4	3	2	1	2	3	
i	5	5	4	3	2	2	3	
n	6	6	5	4	3	3	2	
g	7	7	6	5	4	4	3	

		S	a	t	u	r	d	a	y	
	0	1	2	3	4	5	6	7	8	
S	1	0	1	2	3	4	5	6	7	
u	2	1	1	2	2	3	4	5	6	
n	3	2	2	2	3	3	4	5	6	
d	4	3	3	3	3	4	3	4	5	
a	5	4	3	4	4	4	4	3	4	
y	6	5	4	4	5	5	5	4	3	

FIGURE 49 LEVENSHTEIN DISTANCE EXAMPLE RESULT

7.1.3 Complexity

In totally, the complexity of this algorithm is $O(n*m)$ with n is length of first question and m is length of second question need to compare

E. System Implementation & Test

1. Introduction

1.1. Overview

This chapter describes the testing and implementation Exam Tool. It includes test plans, test cases, test result and risks estimations and some modification to the previous design phase and system testing to minimize the programming and system error.

1.2. Test Approach

- Goal: Test all features in the whole Exam Tool system based on the core flow.
- Method: Interactive system testing.

Testing base on functionality of the software, customer requirements, and implementations. Testers will input data and watch result on the screen. Moreover, we deployment the tool in school's system, students have to do quiz test in software.

2. Database Relationship Diagram

2.1. Physical Diagram:

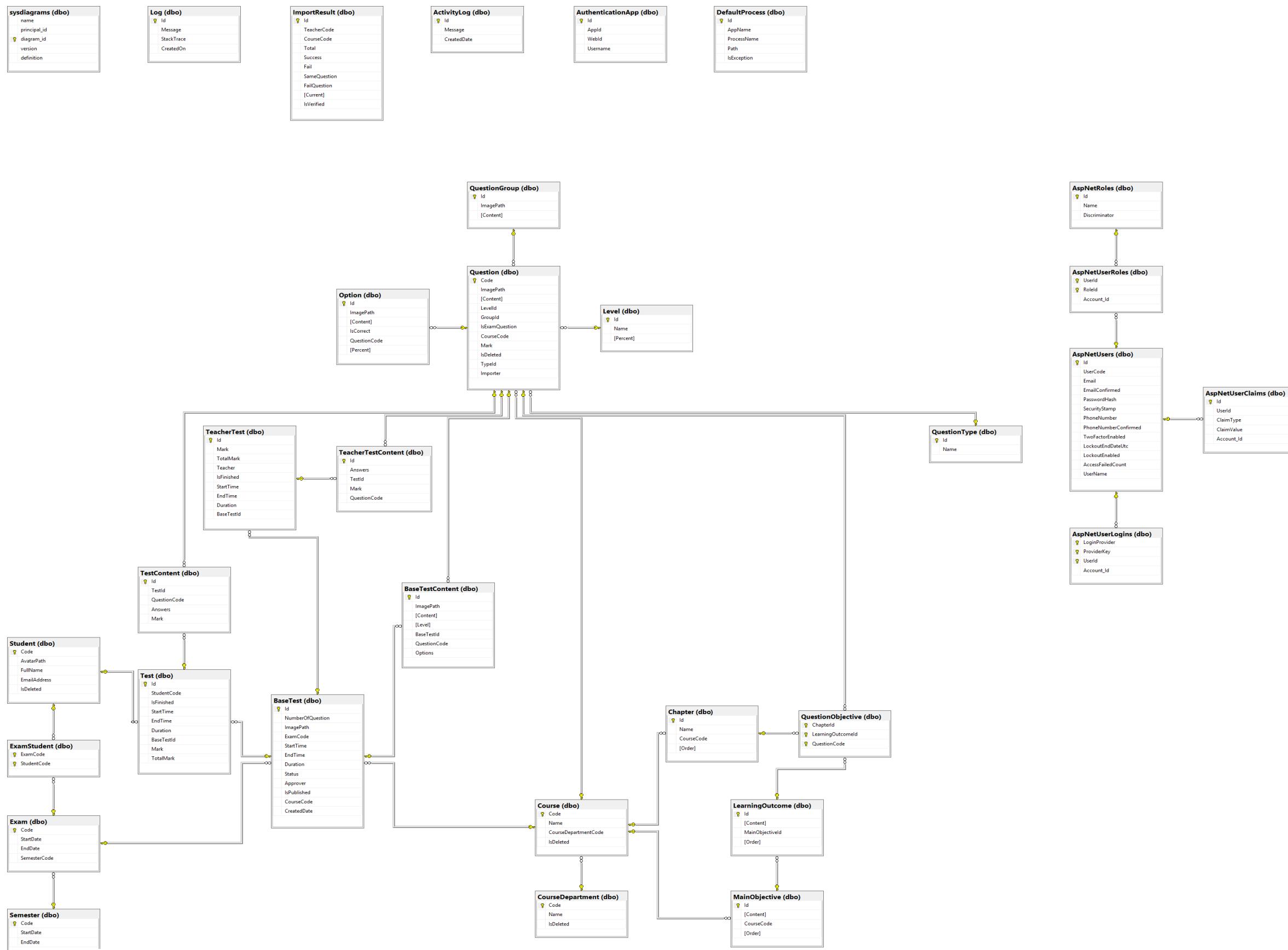


FIGURE 50 PHYSICAL DIAGRAM

Table Name	Description
BaseTest	Contain the test exam information
Semester	Contain the Semester information
Exam	Contain the exam information
ExamStudent	Contain between Student and Exam, this describe the accessible to the test exam of student
Student	Contain the Student information
Test	Contain student Test exam information
TeacherTest	Contain teacher test exam information (because teacher need to take exam to approve test exam)
TeacherTestContent	Contain answer of each teacher's question
BaseTestContent	Contain test exam's question information
TestContent	Contain Student's answer information
Question	Contain question information (question bank)
QuestionGroup	Contain some group question like reading question which contain some question in reading question
Option	Contain option can choice of each question
QuestionType	Type of question information (ex: single choice, multiple choice)
Level	Contain some level of question (hard, easy ...)
QuestionObjective	Question corresponding with LO
Chapter	Contain chapter information
Course	Contain Course information
MainObjective	Contain Purpose of course information
LearningOutcome	Decompose of big Main Objective
CourseDepartment	Contain Course Department information
ImportResult	Result of matching process
AuthenticationApp	Store info to Identify Student when using app
DefaultProcess	Storage Process must kill

TABLE 104 PHYSICAL DIAGRAM DESCRIPTION

2.2. Data Dictionary:

2.2.1 Semester

Attribute	Type	Description
Code	varchar(20)	Code of Semester (ex: Fall 2018)
StartDate	Datetime	Start Date of semester
EndDate	Datetime	End Date of semester

TABLE 105 SEMESTER TABLE DESCRIPTION

2.2.2 Exam

Attribute	Type	Description
Code	varchar(20)	Code of Exam (ex: FinalX)
StartDate	Datetime	Start Date of Exam
EndDate	Datetime	End Date of Exam
SemesterCode	varchar(20)	Code of Semester (ex: Fall 2018)

TABLE 106 EXAM TABLE DESCRIPTION

2.2.3 Student

Attribute	Type	Description
Code	varchar(20)	Code of a Student (ex SE61801)
AvatarPath	nvarchar(Max)	Relative image path of Student Avatar
FullName	nvarchar(255)	Student's Full Name
EmailAddress	nvarchar(255)	Student's Email
isDeleted	bit	Flag to know is this student deleted (true = is deleted)

TABLE 107 STUDENT TABLE DESCRIPTION

2.2.4 BaseTest

Attribute	Type	Description
Id	int	Base Test Id
NumberOfQuestion	int	Number question of this Test Exam

ImagePath	nvarchar(max)	Relative folder storage image of base test.
StartTime	DateTime	Start time for student take test exam
EndTime	DateTime	End time for student take test exam
Duration	Time	Duration of test exam
Status	int	Status of that base test (approved, rejected, Edited...)
Approver	varchar(50)	The people who approve quality of that base test
IsPublished	bit	Is that base test published for student
ExamCode	varchar(20)	Code of Semester (ex: Fall 2018)
CourseCode	Int	Course Code

TABLE 108 BASETEST TABLE DESCRIPTION

2.2.5 BaseTestContent

Attribute	Type	Description
Id	int	Question Id of base test
ImagePath	nvarchar(Max)	Relative folder storage image of base test content.
Content	nvarchar(max)	Content of that question
Level	nvarchar(255)	Level of question (describe how question hard)
Options	nvarchar(max)	Json string list option can choice in that question
baseTestId	int	Base Test Id
questionCode	varchar(20)	Question of this test exam

TABLE 109 BASETESTCONTENT TABLE DESCRIPTION

2.2.6 Test

Attribute	Type	Description

Id	int	Id Student's Test Exam
Mark	float	Mark's Student Test Exam
TotalMark	float	Total mark of that test exam
isFinished	bit	Is student done this test exam
StartTime	DateTime	Time when student take test exam
EndTime	DateTime	End time when student take test exam
Duration	Time	Time of student take that test
baseTestId	int	Base Test Id
StudentCode	varchar(20)	Code of a Student (ex SE61801)

TABLE 110 TEST TABLE DESCRIPTION

2.2.7 TestContent

Attribute	Type	Description
Id	Int	Id Student's answer
Answer	nvarchar(max)	Json string list student answer of this question
Mark	Float	Mark of this answer in this question
QuestionCode	nvarchar(20)	Question Code
TestId	int	Id Student's Test Exam

TABLE 111 TESTCONTENT TABLE DESCRIPTION

2.2.8 Question

Attribute	Type	Description
Code	nvarchar(20)	Question Code
ImagePath	nvarchar(max)	Relative path storage image of question
Content	nvarchar(max)	Question content
IsExamQuestion	bit	Is question already in test exam

Mark	float	Mark of this question
isDeleted	bit	Is this question deleted
importer	varchar(50)	The people who import this question
LevelId	Int	Level of this question
TypeId	Int	Type of this question (single choice, multiple choice)
GroupId	int	Group question in reading question

TABLE 112 QUESTION TABLE DESCRIPTION

2.2.9 Option

Attribute	Type	Description
Id	int	Id of this option
ImagePath	nvarchar(max)	Relative path storage image of option
Content	nvarchar(max)	Option content
IsCorrect	bit	Is this option correct
Percent	float	Percent will Minus if choice wrong
QuestionCode	nvarchar(20)	Question of this option

TABLE 113 OPTION TABLE DESCRIPTION

2.2.10 Level

Attribute	Type	Description
Id	Int	Level Id
Name	nvarchar(255)	Level name (ex: hard, easy)
Percent	float	How important it is

TABLE 114 LEVEL TABLE DESCRIPTION

2.2.11 QuestionType

Attribute	Type	Description
Id	Int	Type Id

Name	nvarchar(255)	Type name (ex: single choice, multiple choice)
------	---------------	--

TABLE 115 QUESTIONTYPE TABLE DESCRIPTION

2.2.12 QuestionGroup

Attribute	Type	Description
Id	int	QuestionGroup Id
ImagePath	nvarchar(max)	Relative Path of this question group
Content	nvarchar(max)	Content of this question group

TABLE 116 QUESTIONGROUP TABLE DESCRIPTION

2.2.13 QuestionObjective

Attribute	Type	Description
ChapterId	int	Chapter Id
LearningOutcomeId	int	Learning Outcome Id
QuestionCode	varchar(20)	Question

TABLE 117 QUESTIONOBJECTIVE TABLE DESCRIPTION

2.2.14 Chapter

Attribute	Type	Description
Id	Int	Chapter Id
Name	nvarchar(255)	Name of chapter (ex: chapter 1)
Order	int	Order of chapter (1, 2...)
CourseCode	Int	Course Code

TABLE 118 CHAPTER TABLE DESCRIPTION

2.2.15 Course

Attribute	Type	Description
Code	Int	Course Code
Name	nvarchar(255)	Course Name
IsDeleted	bit	Is this course deleted
CourseDepartmentCode	nvarchar(20)	Department of this course

TABLE 119 COURSE TABLE DESCRIPTION

2.2.16 MainObjective

Attribute	Type	Description
Id	Int	MainObjective Id
Content	nvarchar(max)	Main objective information
Order	int	Order of this main objective

TABLE 120 MAINOBJECTIVE TABLE DESCRIPTION

2.2.17 LearningOutcome

Attribute	Type	Description
Id	Int	Learning Outcome Id
Content	nvarchar(max)	Learning outcome information
Order	int	Order of this Learning outcome
MainObjectiveId	Int	MainObjective Id

TABLE 121 LEARNINGOUTCOME TABLE DESCRIPTION

2.2.18 Course Department

Attribute	Type	Description
Code	varchar(20)	Learning Outcome Id
Name	nvarchar(255)	Learning outcome information
IsDeleted	bit	Order of this Learning outcome

TABLE 122 COURSEDEPARTMENT TABLE DESCRIPTION

2.2.19 TeacherTest

Attribute	Type	Description
Id	int	Teacher Test Id
Mark	float	Mark of this test exam of teacher
TotalMark	float	Total mark of test test exam
isFinished	bit	Is this teacher test finished
StartTime	DateTime	Time teacher start to take the test exam
EndTime	DateTime	Time teacher end the test exam

Duration	Time	Duration teacher done this test exam
Teacher	varchar(255)	Teacher code get from FPT authentication service
baseTestId	int	Base test of this test exam

TABLE 123 TEACHERTEST TABLE DESCRIPTION

2.2.20 TeacherTestContent

Attribute	Type	Description
Id	int	Teacher Test Id
Mark	float	Mark of this test exam of teacher
Answer	nvarchar(max)	Answer of teacher test exam
questionCode	nvarchar(20)	Question of this answer

TABLE 124 TEACHERTESTCONTENT TABLE DESCRIPTION

2.2.21 AuthenticationApp

Attribute	Description
Id	Authentication app Id
AppId	Identify the app by Id, this Id generate by signalR Id
WebId	Id of web when load by the app identify by AppId. Generated by signalR
Username	Student Username

TABLE 125 AUTHENTICATIONAPP TABLE DESCRIPTION

2.2.22 Import Result

Attribute	Description
Id	Import Result Id
TeacherCode	Teacher who import that section
CourseCode	Course import
Total	Number of question
Success	Number of imported success
Fail	Number of imported fail
SameQuestion	Number of duplicate question
FailQuestion	Number of parse fail question

TABLE 126 IMPORTRESULT TABLE DESCRIPTION

2.2.23 DefaultProcess

Attribute	Description
Id	Identify Id
AppName	Name of the software run on Student computer
ProcessName	Process name
Path	Absolute location of software

TABLE 127 DEFAULTPROCESS TABLE DESCRIPTION

3. Performance Measures

3.1. Parsing Question

When a lecture import a file, client need at least 3 to 5 seconds to parse 100 questions.

3.2. Validate Parsed Questions

When question contain image, client need maximum 10 seconds to check that image is valid or not.

3.3. Matching Questions of a Course

3.4. Random Question to Create Exam Base on Chapters/ Learning Outcome

When random option is selected, systems need maximum 3 seconds to generate 1 exam of a course.

3.5. Get Exam Test from Server

When the waiting time has expired, Exam Tool should finish get exam request at maximum 15 seconds.

3.6. Submit Exam Tests of a Class

When the duration of a test has expired, Exam Tool should finish submit exam at maximum 10 seconds.

4. Test Plan

The purpose of this document is to describe the overall test plan for testing the Exam Tool System. It can verify and ensure that the Exam Tool meets its design specification and other requirements from the user.

4.1. Features to be tested

- **Exam Management**

- Import Question
- Matching Question
- Generate Test
- Approve/Reject Test

- **Exam Tool**

- Do Exam Test
- Submit Exam
- Prevent other applications
- Remember Student's answer
- Prevent multiple logged

4.2. Features not to be tested

- **Exam Management**

- Login
- Logout
- Manage Semester
- Manage Exam Code
- Manage Course
- Manage Chapter
- Manage Learning Outcome
- Manage Course
- Share Screen
- Force Submit
- Export Mark to Excel file
- Review Test

5. System Testing Test Case

5.1. Manage Exam System Test Case

5.1.1 Import Question

ID	Test Case	Test case	Expected output	Inter-test Case	From date	Succes s/Fail	Note
----	-----------	-----------	-----------------	-----------------	-----------	---------------	------

	Description	procedure		Dependence	to date		
IQ_1	Import Question of XML course from Gift Format	1. Questions parsed from file. 2. Submit Questions.	- All questions was parsed successfully - Client receive success notify.	Class SE1167, SE1168	From 18/10/2018 To 18/10/2018	194/42	-There are 26 questions had been duplicated. - There are 12 questions cannot be added into databases - There are 4 questions had wrong format with XML content .

TABLE 128 IMPORT QUESTION TEST RESULT I

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Success/Fail	Note
IQ_2	Import Question of Java Web course from Xml file	1. Questions parsed from file. 2. Submit	- All questions was parsed successfully - Client receive success notify.	Class SE1276, SE1675	From 24/10/2018 To 24/10/2018	124/0	N/A

		Questio ns.					
--	--	----------------	--	--	--	--	--

TABLE 129 IMPORT QUESTION TEST RESULT 2

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Success/Fail	Note
IQ_3	Import Question of XML course from XML file	1. Questions parsed from file. 2. Submit Questions.	- All questions was parsed successfully - Client receive success notify.	Class SE1166, SE1167, SE1168	From 29/10/2018 To 29/10/2018	175/7	- 7 questions had wrong format with XML content .

TABLE 130 IMPORT QUESTION TEST RESULT 3

5.1.2 Matching Question

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Success/Fail	Note
MQ_1	Compare questions import from file with all questions in question bank	1. Compare content. 2. Verify Questions.	- All questions was detect successful	N/A	From 25/11/2018 To 26/11/2018	45/0	N/A

TABLE 131 MATCHING QUESTION TEST RESULT

5.1.3 Generate Exam

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Success/Fail	Note
----	-----------------------	---------------------	-----------------	----------------------------	-------------------	--------------	------

GE _1	Generate Exam base on Chapters	1. Setting configuration of Exam 2. Generate Exam	- An Exam was created successfully	Class SE1166, SE1167, SE1168, SE1276, SE1275	From 24/10 /2018 To 30/10 /2018	7/0	N/A
------------------	--------------------------------	--	------------------------------------	--	---------------------------------	-----	-----

TABLE 132 GENERATE TEST RESULT

5.1.4 Approve Exam

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Succes s/Fail	Note
AP _1	Lecturers take an exam, edit/ swap questions and approve/reject the test	1. Take exam 2. Edit question in the test exam content will auto synchronize into question bank 3. Approve /Reject	- Lecturer can take an exam - Questions in question bank will be updated when teacher edit questions in text exam content.	N/A	From 23/11 /2018 To 27/11 /2018	112/5	- Lecturer cannot submit their test 2 times. - Lecturer cannot update question in question bank 3 times

TABLE 133 APPROVE EXAM TEST RESULT

5.2 Exam Tool Test Case

5.2.1 Run Exam tool

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Succes s/Fail	Note
-----------	------------------------------	----------------------------	------------------------	-----------------------------------	--------------------------	----------------------	-------------

RE _1	Students run the app and login	1. Run exam tool 2. Login	- Every students's laptop can run the app without error. - Login successful	Class SE1167, SE1168	From 18/10 /2018 To 18/10 /2018	26/4	- 4 Laptop s cannot run the app.
--------------	--------------------------------	------------------------------	--	----------------------	---------------------------------	------	----------------------------------

TABLE 134 RUN EXAM TOOL TEST RESULT 1

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Succes s/Fail	Note
RE _2	Students run the app and login	1. Run exam tool 2. Login	- Every students's laptop can run the app without error. - Login successful	Class SE1176, SE1175	From 24/10/ 2018 To 24/10/ 2018	41/4	- 4 Laptop s cannot run the app.

TABLE 135 RUN EXAM TOOL TEST RESULT 2

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Succes s/Fail	Note
RE _3	Students run the app and login	1. Run exam tool 2. Login	- Every students's laptop can run the app without error. - Login successful	Class SE1167, SE1168, SE1166	From 29/10/ 2018 To 29/10/ 2018	46/5	- 5 Laptop s cannot run the app.

TABLE 136 RUN EXAM TOOL TEST RESULT 3

5.2.2 Take Exam

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Succes s/Fail	Note
----	-----------------------	---------------------	-----------------	----------------------------	-------------------	---------------	------

RE _1	Students can take exam then submit their test exam.	1. Take Exam 2. Submit	- Every students can take an exam - Every students can submit their test exam	Class SE1167, SE1168	From 18/10 /2018 To 18/10 /2018	28/2	- 2 students cannot submit exam.
--------------	---	---------------------------	--	----------------------	---------------------------------	------	----------------------------------

TABLE 137 TAKE EXAM TEST RESULT 1

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Succes s/Fail	Note
RE _2	Students can take exam then submit their test exam.	1. Take Exam 2. Submit	- Every students can take an exam - Every students can submit their test exam	Class SE1176, SE1175	From 24/10/ 2018 To 24/10/ 2018	42/3	- 3 students cannot submit exam.

TABLE 138 TAKE EXAM TEST RESULT 2

ID	Test Case Description	Test case procedure	Expected output	Inter-test Case Dependence	From date to date	Success/Fail	Note
RE_3	Students can take exam then submit their test exam.	1. Take Exam 2. Submit	- Every students can take an exam - Every students can submit their test exam	Class SE1167, SE1168, SE1166	From 29/10/2018 To 29/10/2018	47/4	- 3 students cannot submit exam. - 1 student lost exam test because of window update.

TABLE 139 TAKE EXAM TEST RESULT 3

6. System Implement

6.1. Web Server Application

Overall architecture of Examination Tool System with functional layers and the collaboration between the system and the external systems is shown in figure 3 below.

Onion Architecture

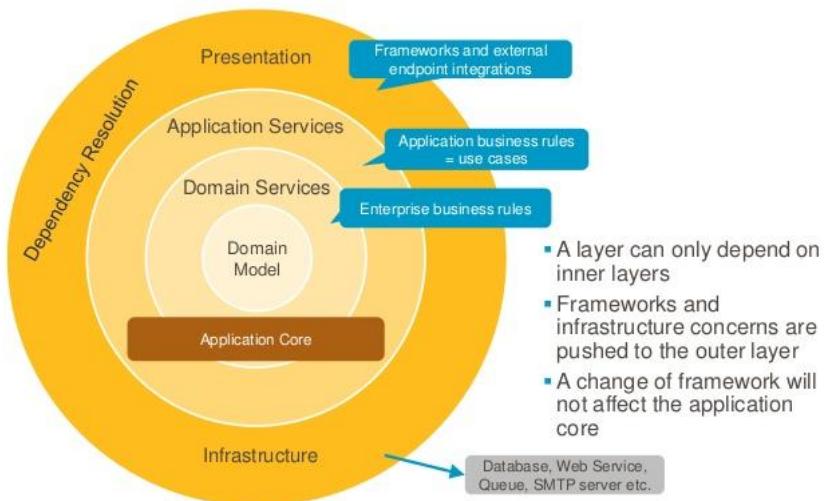


FIGURE 51 ONION ARCHITECTURE

6.2. Web Client Application

- Angular separate page to many components, each component is reusable, which make the application easy to debugging, maintenance.
- Component: The component controls the display, control View, so you can imagine the Component as a controller in the MVC model.
- Service classes are capable of performing some commonly used functions. Some common services are: logging service, data service, message bus, ...
- Using dependency injection: Allow the creation of class objects with all the additional modules / modules / services.
- The event-driven architecture caters to both the client-side and the server- side that are written in JavaScript and thus the synchronization process is fast and orderly. The event loop through web socket protocol which works on TCP handles the multi-user function and prevents the overhead of HTTP for web development.

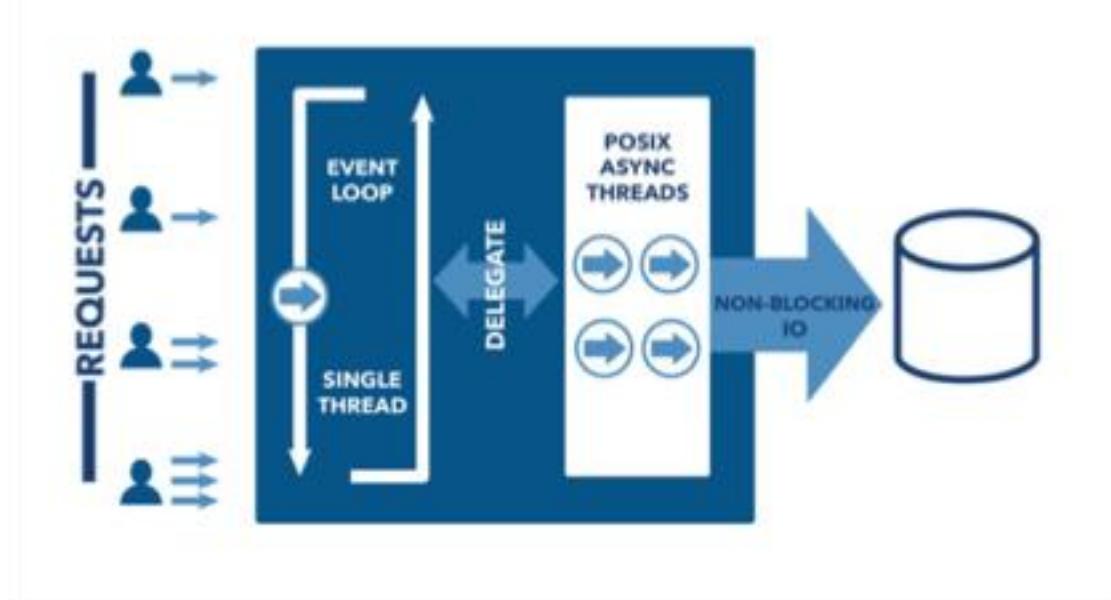


FIGURE 52 EVENT-DRIVEN ARCHITECTURE

In Web client application, we are using Angular under MVC architecture. We choose this Angular because of the following advantage:

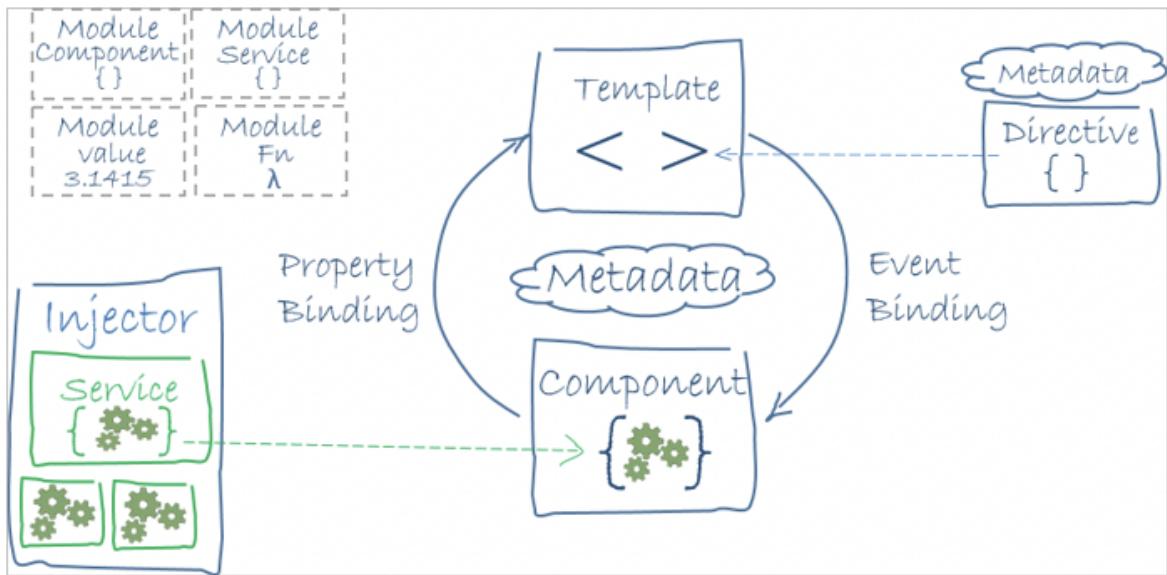


FIGURE 53 ANGULAR COMPONENT EXAMPLE

Reference: "<https://angular.io/guide/architecture>"

F. Software User's Manual

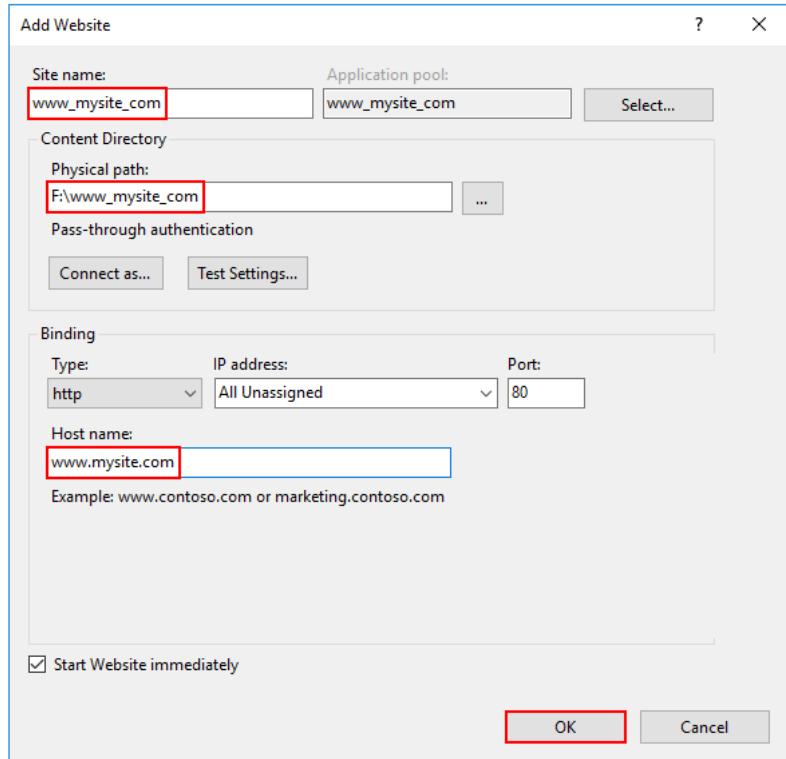
1. Installing Guide

1. On the hosting system, create a folder to contain the app's published folders and files. An app's deployment layout is described in the [Directory Structure](#) topic.
2. Within the new folder, create a *logs* folder to hold ASP.NET Core Module stdout logs when stdout logging is enabled. If the app is deployed with a *logs* folder in the payload, skip this step. For instructions on how to enable MSBuild to create the *logs* folder automatically when the project is built locally, see the [Directory structure](#) topic.

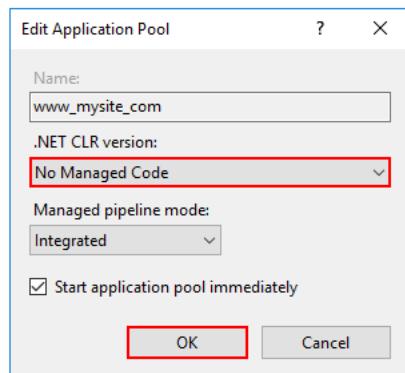
Important

Only use the stdout log to troubleshoot app startup failures. Never use stdout logging for routine app logging. There's no limit on log file size or the number of log files created. The app pool must have write access to the location where the logs are written. All of the folders on the path to the log location must exist. For more information on the stdout log, see [Log creation and redirection](#). For information on logging in an ASP.NET Core app, see the [Logging](#) topic.

3. In **IIS Manager**, open the server's node in the **Connections** panel. Right-click the **Sites** folder. Select **Add Website** from the contextual menu.
4. Provide a **Site name** and set the **Physical path** to the app's deployment folder. Provide the **Binding configuration** and create the website by selecting **OK**:



5. Under the server's node, select **Application Pools**.
6. Right-click the site's app pool and select **Basic Settings** from the contextual menu.
7. In the **Edit Application Pool** window, set the **.NET CLR version** to **No Managed Code**:



ASP.NET Core runs in a separate process and manages the runtime. ASP.NET Core doesn't rely on loading the desktop CLR. Setting the **.NET CLR version** to **No Managed Code** is optional.

8. Confirm the process model identity has the proper permissions.

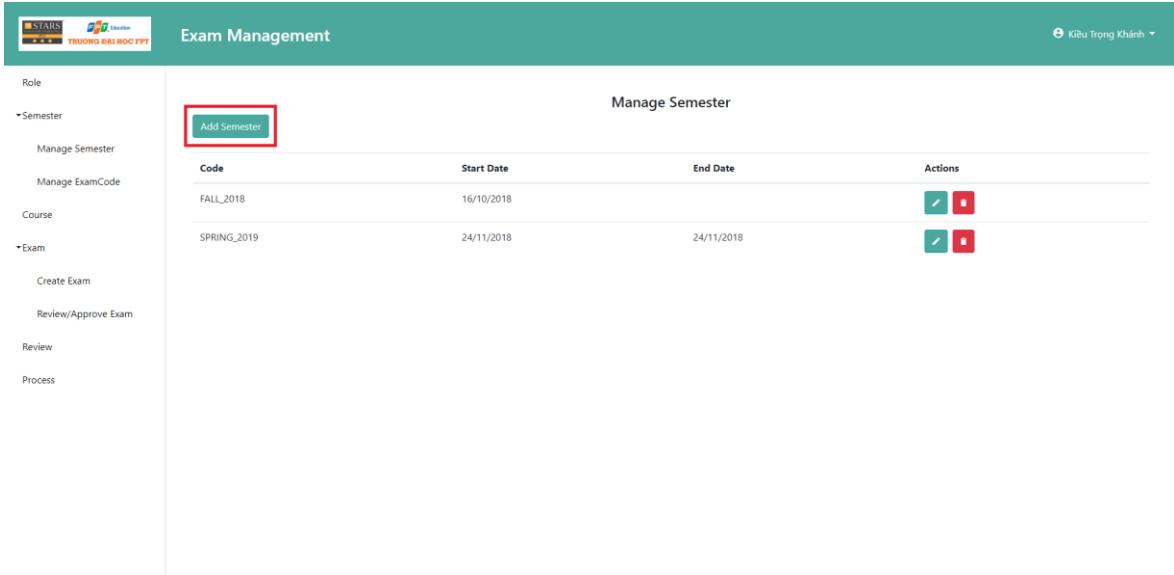
If the default identity of the app pool (**Process Model > Identity**) is changed from **ApplicationPoolIdentity** to another identity, verify that the new identity has the required permissions to access the app's folder, database, and other required

resources. For example, the app pool requires read and write access to folders where the app reads and writes files.

Reference: "<https://docs.microsoft.com/en-us/aspnet/web-forms/overview/deployment/visual-studio-web-deployment/deploying-to-iis>"

2. User Guide

2.1. Admin Create Semester



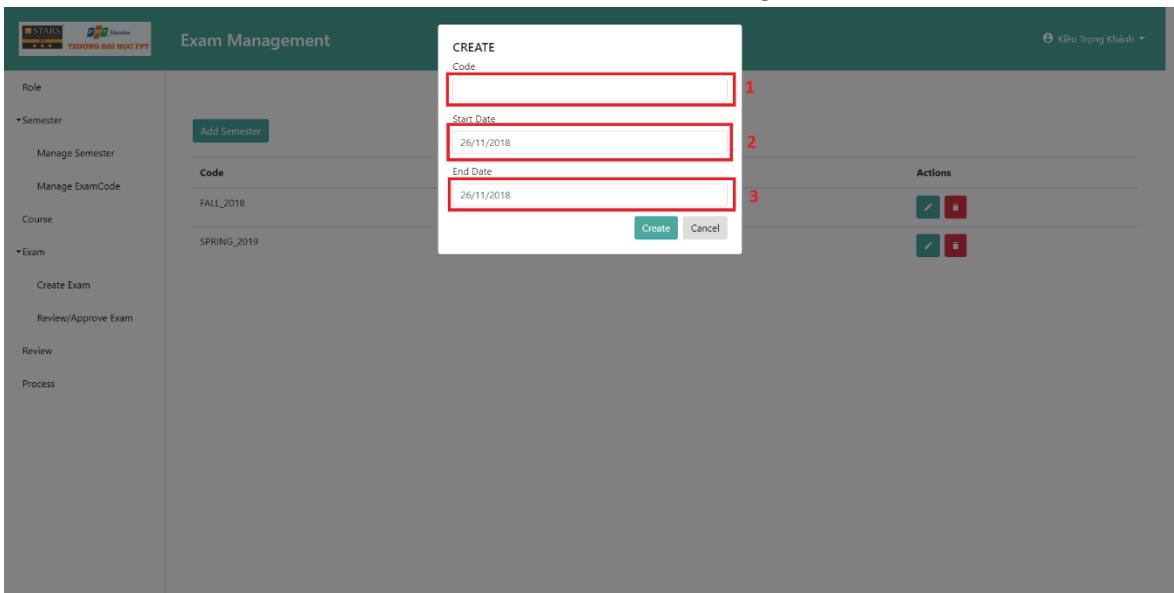
The screenshot shows the 'Exam Management' interface with a sidebar on the left containing navigation links for roles like 'Manage Semester', 'Manage ExamCode', 'Course', and 'Exam'. The main area is titled 'Manage Semester' and displays two rows of data:

Code	Start Date	End Date	Actions
FALL_2018	16/10/2018		
SPRING_2019	24/11/2018		

A red box highlights the 'Add Semester' button at the top of the table.

FIGURE 54 GUIDE CREATE SEMESTER 1

Admin Click the “Add Semester” button to start adding new semester.



The screenshot shows the 'Exam Management' interface with a sidebar on the left containing navigation links for roles like 'Manage Semester', 'Manage ExamCode', 'Course', and 'Exam'. The main area is titled 'CREATE' and contains three input fields:

- Code (labeled 1)
- Start Date (labeled 2)
- End Date (labeled 3)

The 'Create' and 'Cancel' buttons are located at the bottom right of the form.

FIGURE 55 GUIDE CREATE SEMESTER 2

Admin input the Semester Code (1) and the date for start and end of the semester (2) (3)

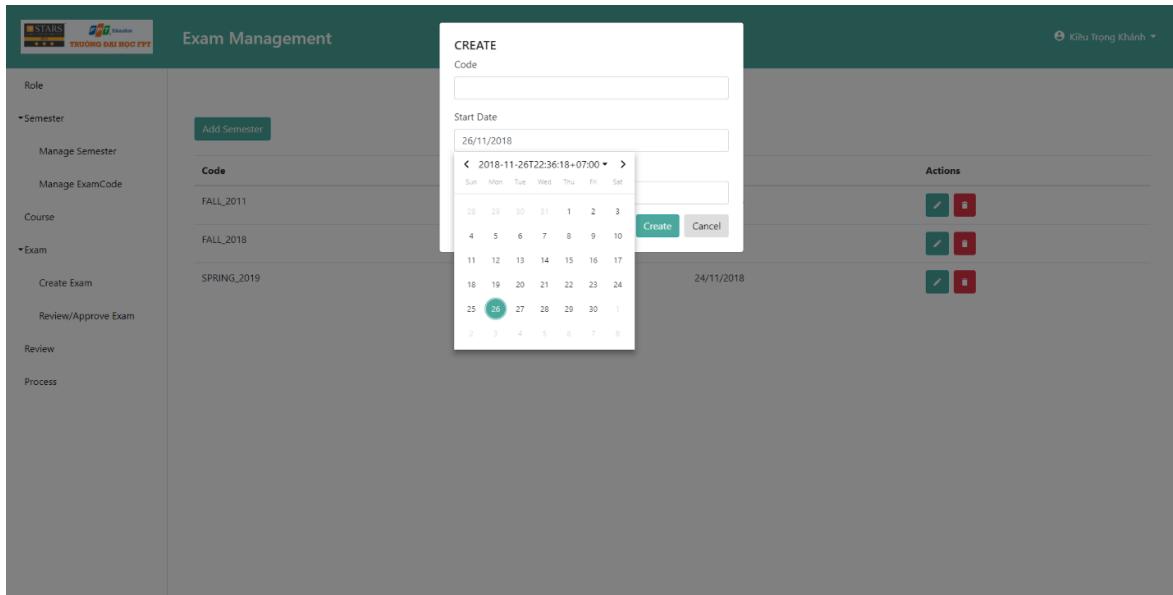


FIGURE 56 GUIDE CREATE SEMESTER 3

A Date picker will show up, admin can choose a date from it.

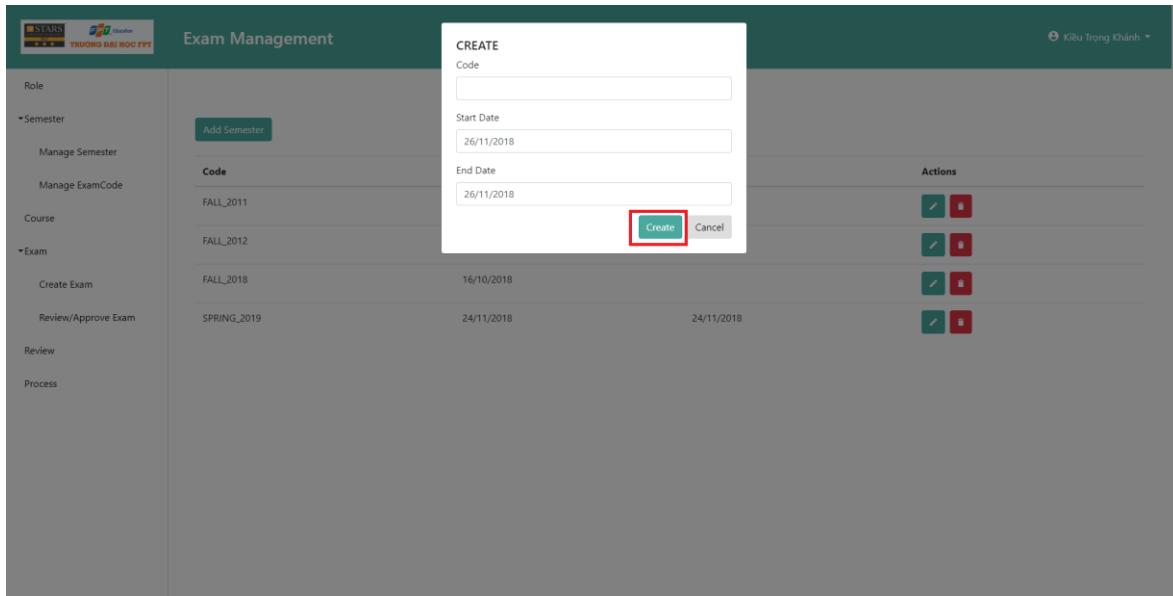


FIGURE 57 GUIDE CREATE SEMESTER 4

After complete all input fields, admin can click the “Create” button to complete and create a new semester.

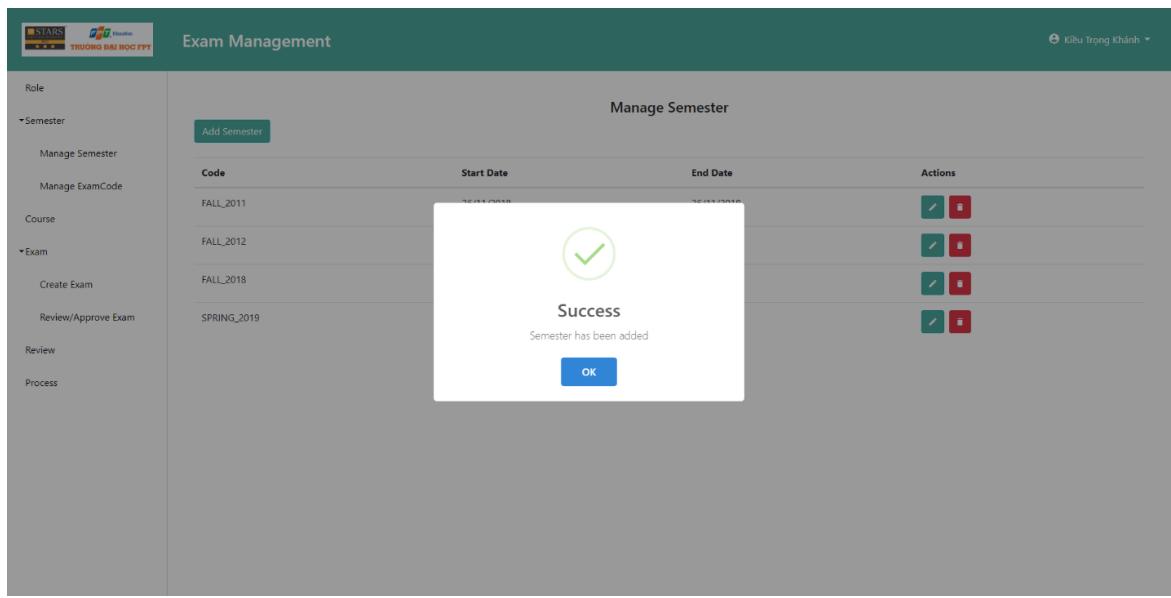


FIGURE 58 GUIDE CREATE SEMESTER 5

A Popup will indicate that the new semester has been successfully created or not.

2.2. Teacher approve/reject exam

The screenshot shows a web-based Exam Management system. At the top, there are logos for STARS and TRƯỜNG ĐẠI HỌC FPT. The main header is "Exam Management". On the right, a user profile for "Kiều Trọng Khánh" is visible. The left sidebar has a navigation menu with items like Role, Semester, Course, Exam, and Process. Under Exam, there are sub-options for Create Exam, Review/Approve Exam, and Release the Test. The main content area displays a table of exams with columns: Course, Start Date, End Date, Duration, Status, and Actions. The table lists several entries, each with a status (e.g., Taken, Pending) and an action button (a green square with a white circle). A red box highlights the "Actions" column.

Course	Start Date	End Date	Duration	Status	Actions
PRX301	26/11/2018 - 11:21	28/11/2018 - 11:21	00:10:00	Taken	
PRX301	26/11/2018 - 11:26	26/11/2018 - 11:29	00:10:00	Out of Date	
PRX301	26/11/2018 - 11:26	26/11/2018 - 11:29	00:10:00	Out of Date	
PRX301	26/11/2018 - 11:28	26/11/2018 - 11:35	00:10:00	Out of Date	
PRX301	26/11/2018 - 11:28	26/11/2018 - 11:35	00:10:00	Out of Date	
PRX301	26/11/2018 - 11:32	26/11/2018 - 11:32	00:10:00	Out of Date	
PRX301	26/11/2018 - 11:32	26/11/2018 - 11:32	00:10:00	Out of Date	
PRX301	27/11/2018 - 11:33	28/11/2018 - 11:33	00:10:00	Not In Time	
PRX301	27/11/2018 - 11:33	28/11/2018 - 11:33	00:10:00	Not In Time	
PRX301	26/11/2018 - 05:04	26/11/2018 - 08:04	00:10:00	Out of Date	
PRX301	26/11/2018 - 05:04	26/11/2018 - 08:04	00:10:00	Out of Date	
PRJ321	26/11/2018 - 11:01	26/11/2018 - 11:01	00:01:00	Out of Date	
PRJ321	26/11/2018 - 11:01	26/11/2018 - 11:01	00:01:00	Out of Date	
PRJ321	26/11/2018 - 11:03	27/11/2018 - 11:03	00:01:00	Taken	
PRJ321	26/11/2018 - 11:03	27/11/2018 - 11:03	00:01:00	Pending	

FIGURE 59 GUIDE TEACHER APPROVE TEST EXAM

Teacher go to the “Review/Approve Exam” Page and can take action to approve or reject exams. Exams which status is “Pending” or “Taken” can be taken action.

This screenshot shows a detailed view of an exam question. The left sidebar is identical to Figure 59. The main area displays a question: "Question 1 : Companies that harm a community can harm themselves". Below it are two radio buttons: "True" and "False", with "True" being selected. A red box highlights the "Submit" button, which is labeled with the number 2. Another red box highlights the radio button for "True", which is labeled with the number 1.

FIGURE 60 GUIDE TEACHER APPROVE TEST EXAM 2

Teacher can either do the test (1) or submit it (2).

The screenshot shows the 'Exam Management' section of a web application. On the left, a sidebar lists navigation options: Role, Semester (Manage Semester, Manage ExamCode), Course, Exam (Create Exam, Review/Approve Exam), Review, and Process. The main area displays a summary of a test for 'Course : PRJ321'. The summary includes: Start Date: 01/01/0001 - 06:42, End Date: 01/01/0001 - 06:42, Number of questions: 1, Duration: 00:01:00, Status: Taken, and Mark: 0. Below this, a question is listed: 'Question : Companies that harm a community can harm themselves' with options A. True and B. False. At the bottom right of the summary box are 'Approve' and 'Reject' buttons, with the 'Approve' button highlighted by a red box labeled '1'.

FIGURE 61 GUIDE TEACHER APPROVE TEST EXAM 3

After submitting the test, a summary about the test will be shown (1). Teacher then can click the “Approve” or “Reject” button.

This screenshot shows a confirmation dialog box over the Exam Management interface. The dialog asks 'Are you sure?' and states 'This Test will not be allowed Approved/Reject anymore!!!'. It features a 'Confirm' button (highlighted with a red box labeled '2') and a 'Cancel' button. In the background, the test summary and question list from Figure 61 are visible.

FIGURE 62 GUIDE TEACHER APPROVE TEST EXAM 4

Click confirm to Approve or Reject the Test Exam.

The screenshot shows a table of exam entries. A modal window in the center displays a green checkmark icon and the word "Success". Below it, a message says "The Test has been Approved!". A blue "OK" button is at the bottom of the modal. The table columns include Course, Start Date, End Date, Duration, Status, and Actions.

Role	Course	Start Date	End Date	Duration	Status	Actions
Semester	PRX301	26/11/2018 - 11:21	28/11/2018 - 11:21	00:10:00	Taken	
Course	PRX301	26/11/2018 - 11:26		00:10:00	Out of Date	
Exam	PRX301	26/11/2018 - 11:26		00:10:00	Out of Date	
Create Exam	PRX301	26/11/2018 - 11:28		00:10:00	Out of Date	
Review/Approve Exam	PRX301	26/11/2018 - 11:28		00:10:00	Out of Date	
Review	PRX301	26/11/2018 - 11:32		00:10:00	Out of Date	
Process	PRX301	27/11/2018 - 11:33		00:10:00	Not In Time	
	PRX301	27/11/2018 - 11:33	28/11/2018 - 11:33	00:10:00	Not In Time	
	PRX301	26/11/2018 - 05:04	26/11/2018 - 08:04	00:10:00	Out of Date	
	PRX301	26/11/2018 - 05:04	26/11/2018 - 08:04	00:10:00	Out of Date	
	PRJ321	26/11/2018 - 11:01	26/11/2018 - 11:01	00:01:00	Out of Date	
	PRJ321	26/11/2018 - 11:01	26/11/2018 - 11:01	00:01:00	Out of Date	
	PRJ321	26/11/2018 - 11:01	27/11/2018 - 11:01	00:01:00	Taken	

FIGURE 63 GUIDE TEACHER APPROVE TEST EXAM 5

Popup will be shown to indicate that the test exam is approved.

The screenshot shows a table of exam entries. A modal window in the center displays a green checkmark icon and the word "Success". Below it, a message says "The Test has been Rejected!". A blue "OK" button is at the bottom of the modal. The table columns include Course, Start Date, End Date, Duration, Status, and Actions.

Role	Course	Start Date	End Date	Duration	Status	Actions
Semester	PRX301	26/11/2018 - 11:26	26/11/2018 - 11:29	00:10:00	Out of Date	
Course	PRX301	26/11/2018 - 11:28	26/11/2018 - 11:35	00:10:00	Out of Date	
Exam	PRX301	26/11/2018 - 11:28	26/11/2018 - 11:35	00:10:00	Out of Date	
Create Exam	PRX301	26/11/2018 - 11:32	26/11/2018 - 11:32	00:10:00	Out of Date	
Review/Approve Exam	PRX301	26/11/2018 - 11:32		00:10:00	Out of Date	
Review	PRX301	27/11/2018 - 11:33		00:10:00	Not In Time	
Process	PRX301	27/11/2018 - 11:33		00:10:00	Not In Time	
	PRX301	26/11/2018 - 05:04		00:10:00	Out of Date	
	PRX301	26/11/2018 - 05:04		00:10:00	Out of Date	
	PRJ321	26/11/2018 - 11:01		00:01:00	Out of Date	
	PRJ321	26/11/2018 - 11:01		00:01:00	Out of Date	
	PRJ321	26/11/2018 - 11:03	27/11/2018 - 11:03	00:01:00	Taken	
	PRJ321	26/11/2018 - 11:03	27/11/2018 - 11:03	00:01:00	Taken	
	PRX301	26/11/2018 - 11:20	26/11/2018 - 11:20	00:01:00	Out of Date	
	PRX301	26/11/2018 - 11:20	26/11/2018 - 11:20	00:01:00	Out of Date	
	PRJ321	26/11/2018 - 11:21	27/11/2018 - 11:20	00:01:00	Pending	

FIGURE 64 GUIDE TEACHER APPROVE TEST EXAM 6

Popup will be shown to indicate that the test exam is rejected.

2.3. Staff Create Exam

Teacher go to the Create Exam page, a create wizard will appear, first is to choose the course.

Exam Management

Role

Semester

Course

Exam

Create Exam

Review/Approve Exam

Review

Process

FIND COURSE EXAM DETAIL SETTING LEARNING OUTCOMES OR CHAPTERS EXAM CONTENT FINISH

Course : Java Web

Start Date for Teacher take Exam **1**
26/11/2018 - 23:36

End Date for Teacher take Exam **2**
26/11/2018 - 23:36

Number of Questions **3**
1

Duration (minute) **4**
1

Number of Tests **5**
2

If you generate Exam using Random feature, u may pass this field.

Previous Next

FIGURE 65 GUIDE STAFF CREATE EXAM 1

After choosing which course will have the test created, teacher must fill in 4 required fields (1)(2)(3)(4), (5) is left for optional if learning outcome is chosen in the next stage.

Exam Management

Role

Semester

Course

Exam

Create Exam

Review/Approve Exam

Review

Process

FIND COURSE EXAM DETAIL SETTING LEARNING OUTCOMES OR CHAPTERS EXAM CONTENT FINISH

Course : Java Web

Start Date for Teacher take Exam
26/11/2018 - 23:36

End Date for Teacher take Exam
26/11/2018 - 23:36

Duration (minute)
1

Number of Tests
2

If you generate Exam using Random feature, u may pass this field.

Cancel Set

FIGURE 66 GUIDE STAFF CREATE EXAM 2

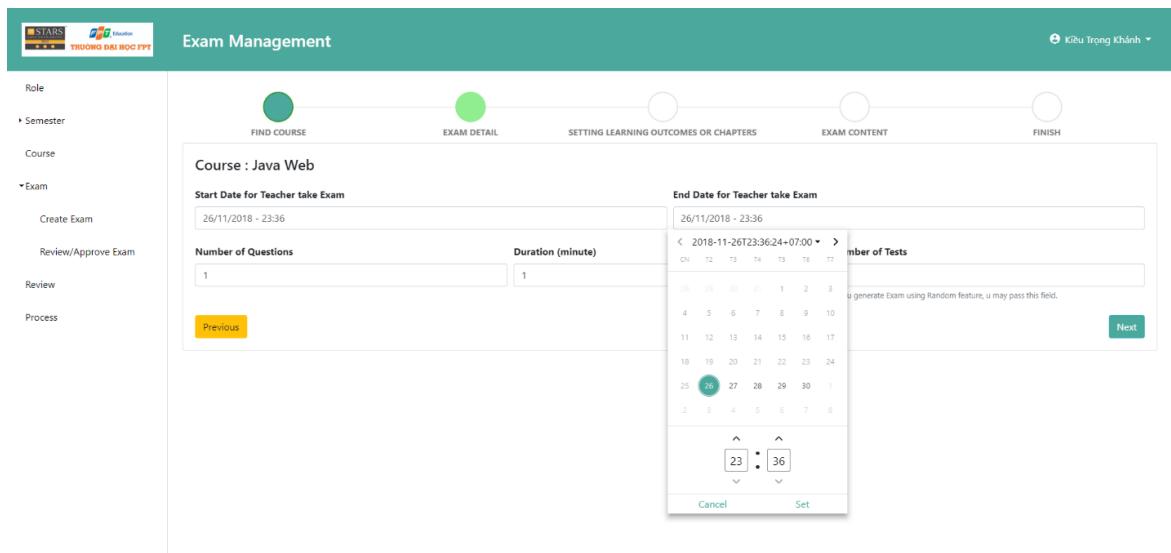


FIGURE 67 GUIDE STAFF CREATE EXAM 3

A date time picker will be show up to choose the start and end date time for teacher to take the exam.

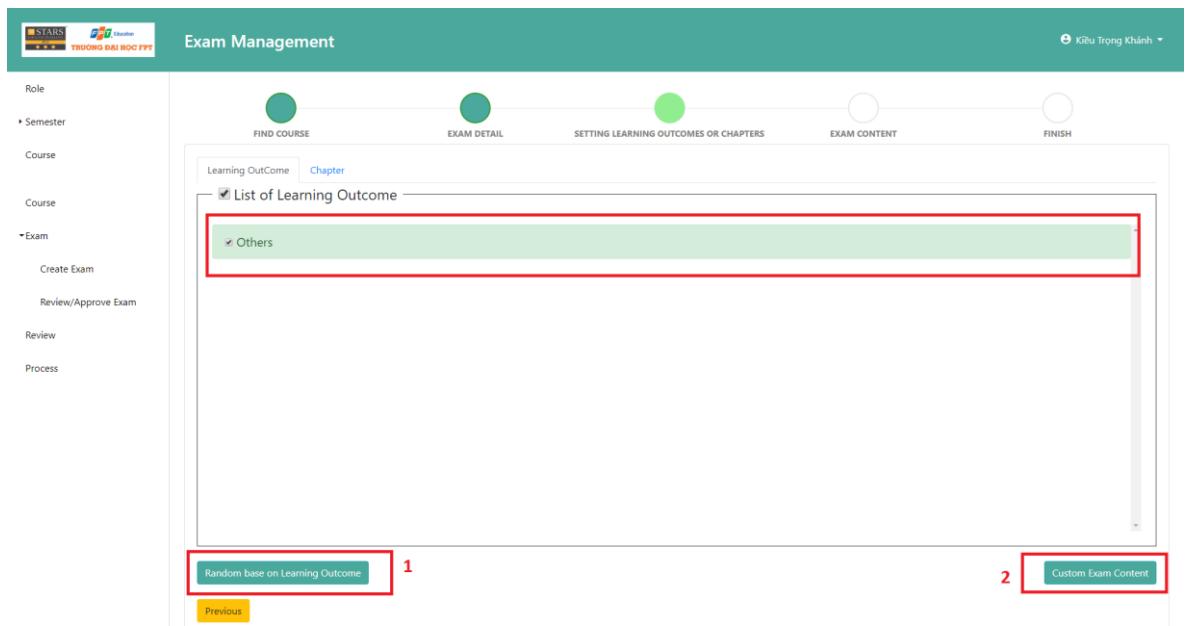


FIGURE 68 GUIDE STAFF CREATE EXAM 4

Teacher can choose which Learning Outcome they want in the test exam. They can randomly choose questions base on Learning Outcome by clicking (1) or manually custom the test exam questions by clicking (2).

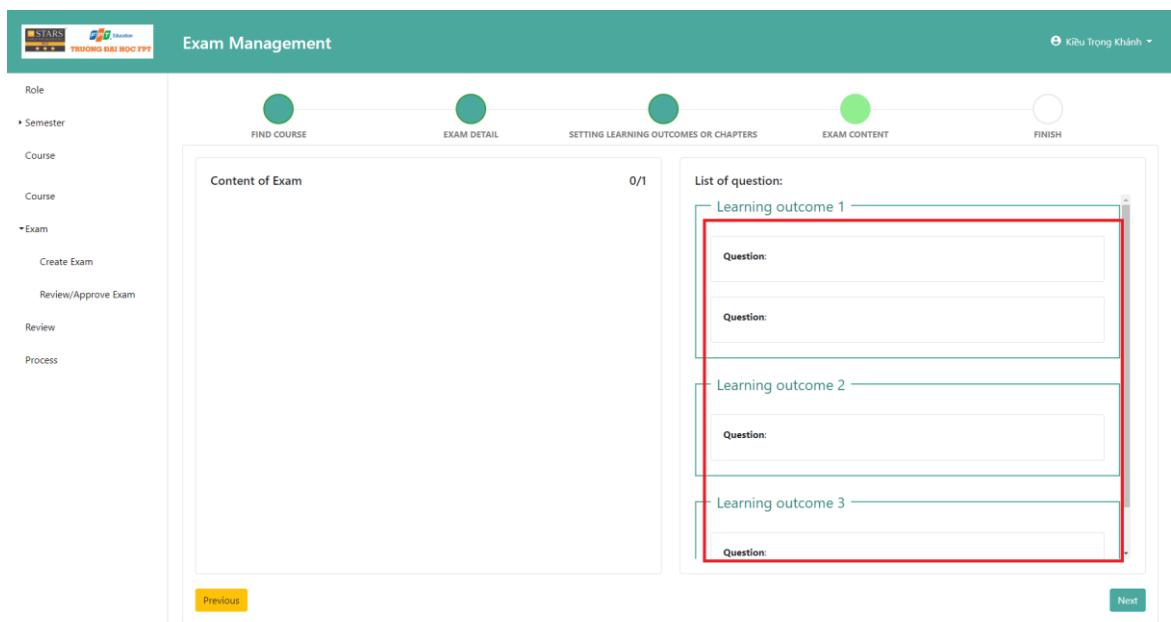


FIGURE 69 GUIDE STAFF CREATE EXAM 5

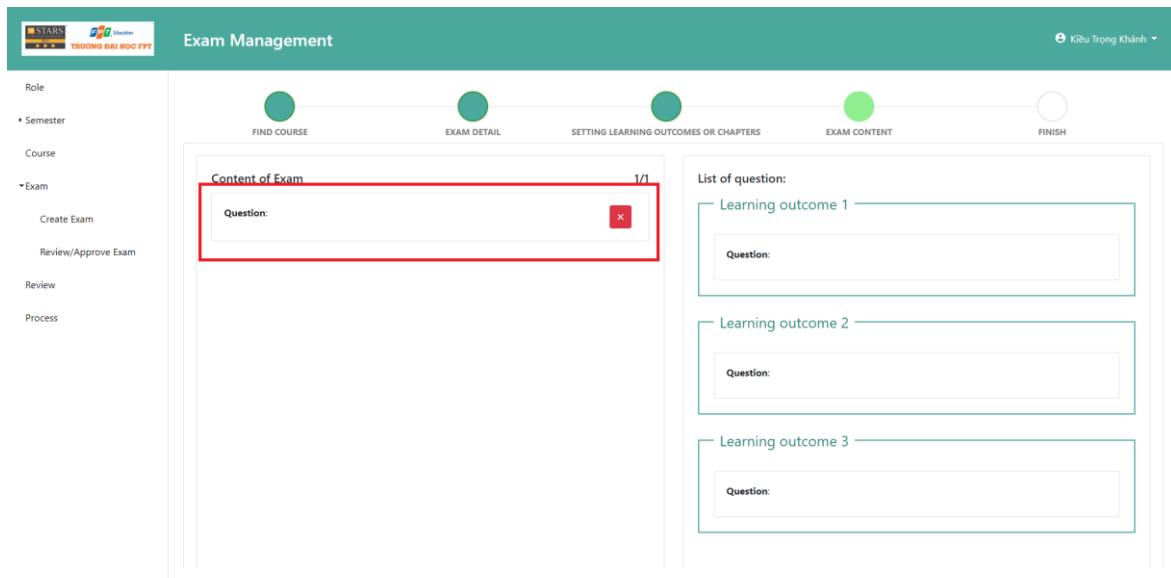


FIGURE 70 GUIDE STAFF CREATE EXAM 6

Questions can be drag and drop from the List to the Content of Exam.

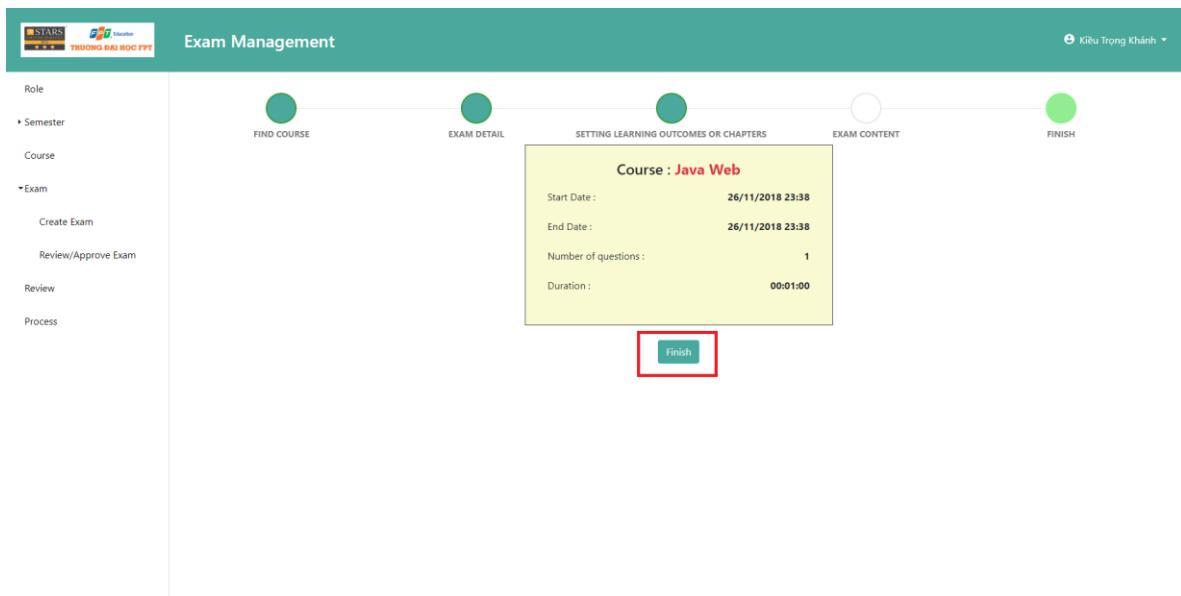


FIGURE 71 GUIDE STAFF CREATE EXAM 7

A Summary will be shown and teacher can click finish to finish the create exam wizard.

G. Appendix

1. Software Engineering 9th by Somerville, page 73
2. C# Code Style Rules: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>
3. Google JavaScript Style Guide: <https://google.github.io/styleguide/jsguide.html>
4. Angular - Style Guide: <https://angular.io/guide/styleguide#single-responsibility>
5. Angular - Architecture overview: <https://angular.io/guide/architecture>