**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document

**Q & A Platform for Educators**

|  |  |
| --- | --- |
| **Group 11** | |
| **Group members** | Kha Hoàng Minh – Team Leader – SE60979  Trương Như Khang – Team Member – SE60992  Trần Đìng Thi – Team Member – SE61137 (Dropped out)  Tạ Đức Tùng – Team Member – 60416 (Dropped out) |
| **Supervisor** | Nguyễn Huy Hùng |
| **Ext. Supervisor** | N/A |
| **Capstone Project code** | QAPE |

-Ho Chi Minh City, May 11 2015-

*This page is intentionally left blank*

# Table of Contents

[Table of Contents 3](#_Toc425411055)

[List of Tables 4](#_Toc425411056)

[List of Figures 4](#_Toc425411057)

[Definitions, Acronyms, and Abbreviations 5](#_Toc425411058)

[A. Report No. 1 Introduction 6](#_Toc425411059)

[1. Project Information 6](#_Toc425411060)

[2. Introduction 6](#_Toc425411061)

[3. Current Situation 6](#_Toc425411062)

[4. Problem Definition 6](#_Toc425411063)

[5. Proposed Solution 7](#_Toc425411064)

[5.1 Feature functions 7](#_Toc425411065)

[5.2 Advantages and disadvantages 7](#_Toc425411066)

[6. Functional Requirements 7](#_Toc425411067)

[7. Role and Responsibility 8](#_Toc425411068)

[B. Report No.2 Software Project Management Plan 9](#_Toc425411069)

[1. Problem Definition 9](#_Toc425411070)

[1.1 Name of this Capstone Project 9](#_Toc425411072)

[1.2 Problem Abstract 9](#_Toc425411073)

[1.3 Project Overview 9](#_Toc425411074)

[2. Project organization 11](#_Toc425411075)

[2.1 Software Process Model 11](#_Toc425411076)

[2.2 Roles and responsibilities 12](#_Toc425411077)

[2.3 Tools and Techniques 13](#_Toc425411078)

[3. Project Management Plan 14](#_Toc425411079)

[3.1 Product Backlog 14](#_Toc425411080)

[3.2 Sprint Backlog 15](#_Toc425411081)

[3.3 All Meeting Minutes 15](#_Toc425411082)

[4. Coding Convention 15](#_Toc425411083)

[C. Report No. 3 Software Requirement Specification 16](#_Toc425411084)

[1. User Requirement Specification 16](#_Toc425411085)

[1.1 Guest Requirement 16](#_Toc425411086)

[1.2 Student Requirement 16](#_Toc425411087)

[1.3 Teacher Requirement 16](#_Toc425411088)

[1.4 Admin Requirement 17](#_Toc425411089)

[2. System Requirement Specification 17](#_Toc425411090)

[2.1 External Interface Requirement 17](#_Toc425411091)

[2.2 System Overview Use Case 17](#_Toc425411092)

[2.3 List of Use Case 18](#_Toc425411093)

[3. Software System Attribute 81](#_Toc425411094)

[3.1 Usability 81](#_Toc425411095)

[3.2 Reliability 81](#_Toc425411096)

[3.3 Availability 81](#_Toc425411097)

[3.4 Security 81](#_Toc425411098)

[3.5 Maintainability 81](#_Toc425411099)

[3.6 Portability 81](#_Toc425411100)

[3.7 Performance 81](#_Toc425411101)

[4. Conceptual Diagram 81](#_Toc425411102)

[D. Report No. 4 Software Design Description 82](#_Toc425411103)

[1. Design Overview 82](#_Toc425411104)

# List of Tables

[Table 1: Definitions 4](#_Toc419743945)

[Table 2: Roles and Responsibilities 7](#_Toc419743946)

[Table 3 Current Situation 8](#_Toc419743947)

[Table 4: Hardware Requirement for Server 10](#_Toc419743948)

[Table 5: Roles and Responsibilities Details 11](#_Toc419743949)

[Table 6: Product Backlog Detail 12](#_Toc419743950)

[Table 7: Sprint 1 13](#_Toc419743951)

# List of Figures

[Figure 1: Scrum Development Model\* 13](#_Toc425411163)

[Figure 2: System Overview Use Case 19](#_Toc425411164)

[Figure 3: <Guest> Overview Use Case 19](#_Toc425411165)

[Figure 4: <Guest> Login Use Case Diagram 20](#_Toc425411166)

[Figure 5: <Guest> Register Use Case Diagram 21](#_Toc425411167)

[Figure 6: <Teacher> Overview Use Case 23](#_Toc425411168)

[Figure 7: <Teacher> Create class Use Case Diagram 24](#_Toc425411169)

[Figure 7: <Teacher> Invite student Use Case Diagram 26](#_Toc425411170)

[Figure 9: <Teacher> Edit class Use Case Diagram 28](#_Toc425411171)

[Figure 10: <Teacher> Close class Use Case Diagram 29](#_Toc425411172)

[Figure 11: <Teacher> Reopen class Use Case Diagram 31](#_Toc425411173)

[Figure 12: <Teacher> Accept request of student Use Case Diagram 32](#_Toc425411174)

[Figure 13: <Teacher> Remove student Use Case Diagram 34](#_Toc425411175)

[Figure 14: <Teacher> Create post Use Case Diagram 35](#_Toc425411176)

[Figure 15: <Teacher> Edit post Use Case Diagram 37](#_Toc425411177)

[Figure 16: <Teacher> Close post Use Case Diagram 39](#_Toc425411178)

[Figure 17: <Teacher> Open post Use Case Diagram 40](#_Toc425411179)

[Figure 18: <Teacher> Delete post Use Case Diagram 42](#_Toc425411180)

[Figure 19: <Teacher> Upload material Use Case Diagram 43](#_Toc425411181)

[Figure 20: <Teacher> Delete material Use Case Diagram 45](#_Toc425411182)

[Figure 21: <Teacher> Invite teacher answering question Use Case Diagram 46](#_Toc425411183)

[Figure 22: <Teacher> Want answer Use Case Diagram 48](#_Toc425411184)

[Figure 7: <Teacher> Un-want answer Use Case Diagram 49](#_Toc425411185)

[Figure 24: <Teacher> Follow teacher Use Case Diagram 50](#_Toc425411186)

[Figure 24: <Teacher> Un-follow teacher Use Case Diagram 51](#_Toc425411187)

[Figure 26: <Teacher> Edit profile Use Case Diagram 53](#_Toc425411188)

[Figure 27: <Teacher> Merge question Use Case Diagram 55](#_Toc425411189)

[Figure 28: <Teacher> Search Use Case Diagram 57](#_Toc425411190)

[Figure 29: <Student> Overview Use Case 59](#_Toc425411191)

[Figure 30: <Student> Accept Invitation Use Case Diagram 59](#_Toc425411192)

[Figure 31: <Student> Request for joining class Use Case Diagram 61](#_Toc425411193)

[Figure 32: <Student> Leave class Use Case Diagram 62](#_Toc425411194)

[Figure 33: <Student> Create folder Use Case Diagram 64](#_Toc425411195)

[Figure 34: <Student> Delete folder Use Case Diagram 65](#_Toc425411196)

[Figure 35: <Student> Add material to folder Use Case Diagram 66](#_Toc425411197)

[Figure 36: <Student> Delete material Use Case Diagram 68](#_Toc425411198)

[Figure 37: <Student> Create post Use Case Diagram 68](#_Toc425411199)

[Figure 38: <Student> Edit post Use Case Diagram 69](#_Toc425411200)

[Figure 39: <Student> Delete post Use Case Diagram 70](#_Toc425411201)

[Figure 40: < Student> Want answer Use Case Diagram 70](#_Toc425411202)

[Figure 7: < Student> Un-want answer Use Case Diagram 71](#_Toc425411203)

[Figure 42: < Student> Follow teacher Use Case Diagram 71](#_Toc425411204)

[Figure 43: < Student> Un-follow teacher Use Case Diagram 72](#_Toc425411205)

[Figure 44: <Student> Edit profile Use Case Diagram 72](#_Toc425411206)

[Figure 45: <Student> Search Use Case Diagram 73](#_Toc425411207)

[Figure 46: <System> Overview Use Case 73](#_Toc425411208)

[Figure 47: <System> Push Notification Use Case Diagram 74](#_Toc425411209)

[Figure 48: <System> Feed News Use Case Diagram 75](#_Toc425411210)

[Figure 49: <Admin> Overview Use Case 77](#_Toc425411211)

[Figure 13: <Admin> Manage user Use Case Diagram 77](#_Toc425411212)

[Figure 51: <Admin> Search Use Case Diagram 78](#_Toc425411213)

[Figure 52: <Admin> Delete post Use Case Diagram 79](#_Toc425411214)

[Figure 53: <Admin> Close class Use Case Diagram 80](#_Toc425411215)

[Figure 54: <Admin> Reopen class Use Case Diagram 80](#_Toc425411216)

[Figure 55: <Admin> Delete material Use Case Diagram 81](#_Toc425411217)

[Figure 56: Conceptual diagram 83](#_Toc425411218)

# Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Name** | **Definition** |
| QAPE | Q & A Platform for Educators |

Table 1: Definitions

# Report No. 1 Introduction

## Project Information

* Project name: **Q&A Platform for Educators**
* Project Code: **QAPT11**
* Product Type: **Web Application**
* Start Date: **May 11th, 2015**
* End Date: **August 11th, 2015**

## Introduction

Nowadays, the development of internet and information technology led to the dramatic increase in trend of studying online. More and more information together with limited school time make having a website that supports studying is needed.

This project is a Q&A Web application where students and teachers can exchange information and other educational materials. Students can learn more things about the subject from posts of teachers or find the answers from other people.

## Current Situation

There is a lot of popular Q&A websites like Quora, Yahoo! Answers, Stack Overflow… People can post threads or answer the questions pretty easy using its account or connect to other accounts. These sites also have rating system for questions and responses in order to rate users as well.

Quora is rated as the best Q&A websites where you can find the answer for your question from real doctors, lawyers, chefs, engineers and more.

Yahoo! Answer is such as a longstanding Q&A website that everyone can join a thread and answer their question. The response will be awarded 10 points if the rate is highly recommended or 2 points if the rate is lower.

Stack Overflow is a Q&A websites for IT people with a lot of the professional and enthusiasm users. You can post a thread and wait responses from a lot of experienced programmers.

## Problem Definition

Most of Q&A problems are about the rewards of their good contributions. Somebody complained they should deserve better.

Quora and Yahoo! Answer are known as the Q&A famous website but their objects are about a lot of different issues in life. These websites doesn’t only focus on education so that students and teachers will be difficult for sharing their knowledge. Spamming questions are also problems of them. On the other hand, Stack Overflow only focuses on programing rather than diversifying subjects. All of them do not yet support the true studying environment.

## Proposed Solution

We suggest developing a Q&A website that focuses on studying. This website is such as a university in reality that includes many classes, teachers and students. It supports teachers share their additional lectures because of lacking of time. It also helps students ask self-confidently because some of them feel shy when asking in school.

### **Feature functions**

* Teachers have their classes with lectures and students. They can share additional lectures and studying materials, invites students join their classes. They also can invite other teachers together discuss about lectures or question.
* Students can ask about not yet understanding issues. They also join in classes to learn additional subject, follow their favorite teachers and manage studying material that uploaded by teachers.

### Advantages and disadvantages

The advantages and disadvantages of the proposed solution:

* Advantage:
* Teachers can upload their studying material (.doc, .pdf…) and students can manage it online.
* By supporting real time notification and discussion, students can imagine this website as the university with many classes, teachers and friends. They will feel free to ask anything.
* Users will have their right answers faster than they do on their own. That leads to learn better in real classes.
* System has a variety of categories that is better than Stack Overflow but it is not a mess such as Quora or Yahoo! Answers.
* Disadvantage:
* System do not support the users suggest teacher to join threads to answer the questions.
* Cost time waiting to approve to join a class.

## Functional Requirements

Function requirements of the system are listed as below:

**For Teachers:**

* Create classes.
* Create a post. Post types include: articles, questions, answer, and upload material.
* Invite other teachers to answer questions in the classes.
* Invite students to join the discussion in a class.
* Track students’ questions to see which topics a student is less known about.
* Support real time notification.
* Support real time in discussion.

**For Students:**

* Create a post: articles, questions, answer.
* Manage learning materials: allowing students to create folders to store different kinds of studying documents.
* Support real time notification.
* Support real time in discussion.
* Follow a teacher.
* Send a request to join in a teacher’s class.
* Support an easy way to add mathematic equation, code highlighting besides other normal text formatting for editor.
* Suggest related posts.

## Role and Responsibility

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full Name** | **Role** | **Position** | **Contact** |
| **1** | Nguyễn Huy Hùng | Project Manager | Instructor | HungNH@fpt.edu.vn |
| **2** | Kha Hoàng Minh | Developer | Leader | MinhKHSE60979@fpt.edu.vn |
| **3** | Trương Như Khang | Developer | Member | KhangTNSE60992@fpt.edu.vn |
| **4** | Trần Đình Thi | Developer | Member | ThiTDSE61137@fpt.edu.vn |
| **5** | Tạ Đức Tùng | Developer | Member | TùngTD60416@fpt.edu.vn |

Table 2: Roles and Responsibilities

# Report No.2 Software Project Management Plan

## Problem Definition



### Name of this Capstone Project

* Q&A Platform for Educators (QAPE)

### Problem Abstract

Nowadays, in almost schools and university, the level of exchange knowledge between students and teachers are not much. A result of learning and teaching has not reached the highest efficiency with the following reasons:

Teaching hours of pretty little, much knowledge so that teachers cannot answer all questions of students during class, students fell shy to ask about their problems are not understood.

So building online Q&A system is urgently needed employment in the current educational context. This system will help teachers and students interact with each other through an application Website. Teachers can create their own classes are teaching on the Website and allow their student can join and post all question in that. Teacher can answer question in owned classroom or invites other teachers to join and answer that question. They also can upload material and document for students that help them understand clearly the problem they do not understand.

.

### Project Overview

#### Current Situation

|  |  |  |
| --- | --- | --- |
| Website | Advantages | Disadvantages |
| Quora.com | -Can get answer by a prof.  -There are more Category  -User can edit question and edit answer from other user. | -Too much Category  -Everyone can answer question.  -Hard to find old answer.  -There is not real time discussion. Answer only. |
| Yahoo! Answer | -Can best answer by rating point  -Can ask everything: travel, book, films, social…. | -Everyone can answer question.  -No good interface design (too much word) |
| Stack Overflow | -Always pop Google search.  -So much data and load fast. | -No good design interface.  -There is not Category.  -There are much Basic question. |

Table 3 Current Situation

#### The Proposed System

From above disadvantages, this project is going to build a website that help teacher and student can communicate with each other easier and can get the answer from who what they want.

#### Website

**For Teachers:**

Create classes

Create a post. Post types include: articles, questions, answer, and upload material

Invite other teachers to answer questions in the classes

Invite students to join the discussion in a class

Track students’ questions to see which topics a student is less known about

Real time notification

Real time in discussion

**For Students:**

Create a post: articles, questions, answer

Manage learning materials: allowing students to create folders to store different kinds of studying documents.

Real time notification

Real time in discussion

Follow a teacher

Send a request to join in a teacher’s class.

Note: the editor must support an easy way to add mathematic equation, code highlighting besides other normal text formatting.

Suggest related posts

#### Boundaries of the System

This system is a Web application that helps student and teacher communicate with each other by Question and Answer.

There are three type of account: Student, Teacher, and Admin. They have different features.

#### Development Environment

##### Hardware requirements

**For server**

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

Table 4: Hardware Requirement for Server

##### Software requirements

* Handle Products and sprints backlog: MS Excel 2013.
* Implement Website and service: IntelliJ v13.1.6.
* Create and manage database for system: MySQL v5.6.24 and MySQL Workbench v6.6.3.
* Source control: Github.
* Create models and diagrams: Software Ideas Modeler.
* Communicate and meeting: Skype 7.0 and Google mail.
* System and platform development: Window 7.

## Project organization

### Software Process Model



Figure 1: Scrum Development Model\*

Because of the project characteristics, we decide to make progress in series of sprints which are time boxed iterations in one week. At the beginning of one sprint, through sprint planning meeting, all team members will discuss together to define sprint backlog which is suitable to be completed within a week. Daily meeting and online conference are used to manage all activities and issues troubleshoot. At the end of sprint, product owner and developer team will review completed product to figuring out the necessary changes for products.

\*Reference from http://www.hanoiscrum.net

### Roles and responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Full name** | **Role in Group** | **Responsibilities** |
| **1** | Nguyễn Huy Hùng | Scrum Master/Product Owner | * Specify user requirement * Control the development process * Give out technique and business analysis support |
| **2** | Kha Hoàng Minh | Team Leader, BA, DEV, Tester | * Managing process * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **3** | Trương Như Khang | Team Member, BA, DEV, Tester | * Designing database * Clarifying requirements * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **4** | Trần Đình Thi | DEV | * Prepare documents * GUI Design * Create test plan * Coding * Testing |
| **5** | Tạ Đức Tùng | DEV | * Prepare documents * GUI Design * Create test plan * Coding * Testing |

Table 5: Roles and Responsibilities Details

### Tools and Techniques

- Front-end technologies: HTML5, CSS3, JavaScript, jQuery, AJAX.

- Back-end technologies:

+Website: Java

+Framework: Spring.

- Web Server: Installed Tomcat 7.0.61.

- Database Management System: MySQL 5.6.

## Project Management Plan

### Product Backlog

|  |  |  |  |
| --- | --- | --- | --- |
| **Project: Q&A Platform for Educators** | | | |
|
| **ID** | **Function** | **Description** | **Added Sprint** |
| 1 | Register | Registering to become member of website |  |
| 2 | Login/Logout | Accessing the account to use their features |  |
| 3 | Get forgotten password | Resetting password if forgetting |  |
| 4 | Edit profile | Users edit their profile |  |
| 5 | Create class | Teachers create class for student's joining |  |
| 6 | Create article, question, answer | User post an article, ask a question or answer a question |  |
| 7 | Upload material | Teachers upload their studying documents |  |
| 8 | Invite teacher | Teachers invite other teachers to answer a question on |  |
| 9 | Invite student | Teachers invite students to join the discussion in class |  |
| 10 | Accept/Deny an invitation | Users accept/deny teachers' invitation |  |
| 11 | Accept/Deny a request | Teachers accept/deny a joining request from student |  |
| 12 | Track student | Teachers track students’ questions to see which topics a student is less known about |  |
| 13 | Manage post | Users manage their post (edit, remove post), Admin only can remove post. |  |
| 14 | Manage class | Teachers manage students in their class |  |
| 15 | Real time notify | System support real time notification when followed teachers have new posts or posts have new reply |  |
| 16 | Real time discuss | New answers or replies will show at real time without refreshing page |  |
| 17 | Manage learning materials | Students to create folders to store different kinds of studying documents |  |
| 18 | Follow teacher | Students follow their favorite teachers |  |
| 19 | Send a request | Students send a request to join in a teacher’s class. |  |
| 20 | Suggest related post | System suggest related posts (related articles, related questions) for users |  |
| 21 | Search | Users search posts or other users |  |
| 22 | Manage users | Admin can activate/deactivate users |  |

Table 6: Product Backlog Detail

### Sprint Backlog

#### Sprint 1

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Task** | **Implementer** | **Estimated hours** |
| 1 | Design homepage | MinhKH | 10 |
| 2 | Design register page | ThiTD | 6 |
| 3 | Design login form | ThiTD | 6 |
| 4 | Design get forget password page | ThiTD | 6 |
| 5 | Design student dashboard | TungTD | 8 |
| 6 | Design teacher dashboard | TungTD | 8 |
| 7 | Design admin dashboard | TungTD | 8 |
| 8 | Report 1 | ThiTD | 4 |
| 9 | Review report 1 | MinhKH | 3 |
| 10 | Setup environment turorial | KhangTN | 8 |
| 11 | Create overall use-case | KhangTN | 15 |
| 12 | Create guest use-cases | ThiTD | 10 |
| 12 | Create ERD | MinhKH | 12 |
| 13 | Create System Architectural Design | KhangTN | 10 |
| 14 | Report 2 | TungTD | 8 |
| 15 | Review report 2 | MinhKH | 3 |

Table 7: Sprint 1

### All Meeting Minutes

Place at Meeting Minute folder in Github.

## Coding Convention

Use Java coding convention to develop website.

Summary:

Naming Convention:

Use camel case for both variable and function name.

Use Pascal case for class name.

Indentation:

Avoid lines longer than 80 characters, since they are not handled well by many terminals and tools.

Declaration:

One declaration per line is recommended since it encourages commenting.

In absolutely no case should variables and functions be declared on the same line.

Do not put different types on the same line.

Code Examples:

Follow “Code Conventions for the Java TM Programming Language, by Sun Microsystems, rev April 20, 1999”.

<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

# Report No. 3 Software Requirement Specification

## User Requirement Specification

### Guest Requirement

Guest is a person who doesn’t access to the system. There are some functions guest can use:

* Register
* Login

### Student Requirement

Student is a person who accesses to the system with role “student”. There are functions student can use:

* Manage owned questions/articles/answers (create new, edit and delete)
* Request to join/Leave classrooms
* Manage materials (add material to folder, delete material in folder)
* Manage folders (create new, delete)
* Want answer the questions
* Accept answers
* Like/Unlike articles
* Follow/Un-follow teachers
* Edit profile
* Search
* Logout

### Teacher Requirement

Teacher is a person who accesses to the system with role “teacher”. There are functions teacher can use:

* Mange owned classrooms (create new, edit, close)
* Manage questions/articles/answers in owned classrooms (create new, edit, close)
* Manage owned questions/articles/answers (create new, edit, delete)
* Upload/Delete materials in owned classrooms
* Accept/Remove students in owned classrooms
* Invite students to join owned classrooms
* Invite teachers to answer questions in owned classrooms
* Merge questions in owned classrooms
* Answer multiple questions at once
* Want answer the questions
* Accept answers
* Like/Unlike articles
* Follow/Un-follow teachers
* Edit profile
* Search
* Logout

### Admin Requirement

Admin is a person who has the right of administration. There are functions admin can use:

* Manage articles/questions/answers (delete)
* Manage users (activate, deactivate)
* Manage category (create new, delete)

## System Requirement Specification

### External Interface Requirement

#### User Interface

* General requirement for graphics user interface is the GUI should be simple, clear, intuitive, and reminiscent.
* The interface design is an iterate process includes design, sketching, prototyping, user assessment.

#### Hardware Interface

* N/A

#### Software Interface

* Run with Chrome (Version 43.0.2357.132 m) and Cốc Cốc (Version 43.0.2357.126) browsers and must support JavaScript and html5.
* The screen of browsers must be at least 1024 x 768px to browse complete part of website.

#### Communication Protocol

* Using HTTP/HTTPS protocol

### System Overview Use Case

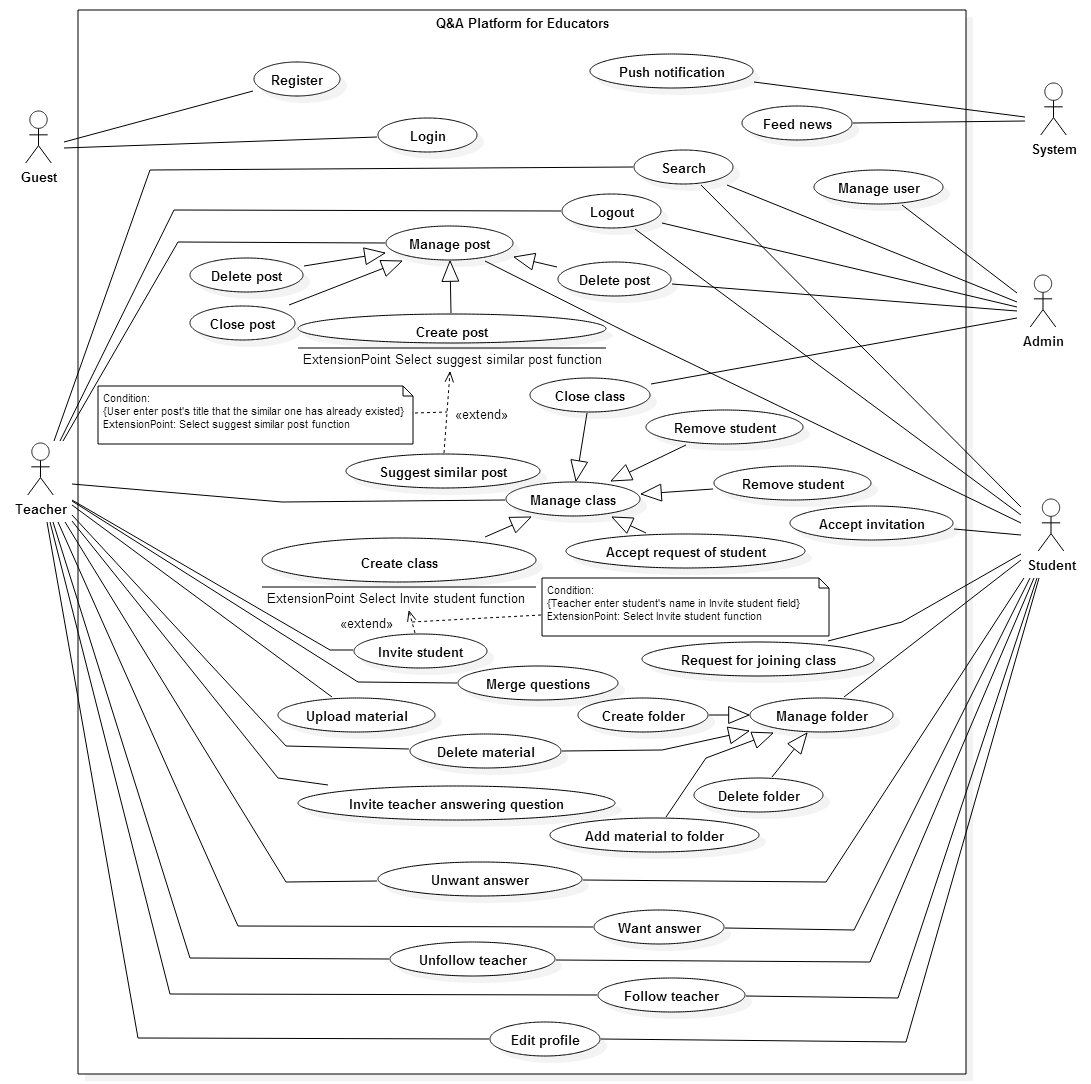
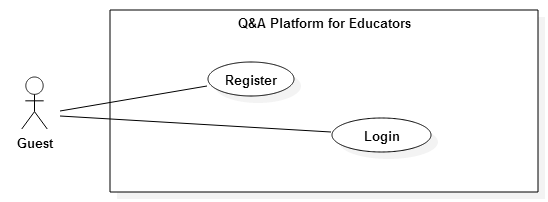


Figure 2: System Overview Use Case

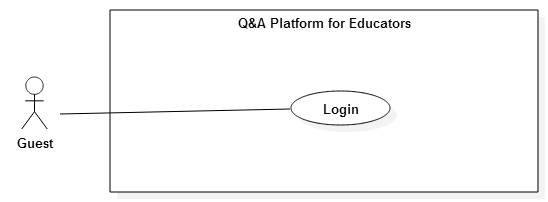
### List of Use Case

#### <Guest> Overview Use Case



**Figure 3: <Guest> Overview Use Case**

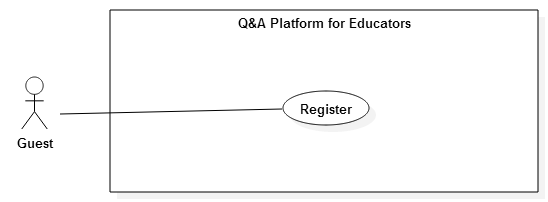
##### <Guest> Login Use Case Diagram

****

**Figure 4: <Guest> Login Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -1 SPECIFICATION** | | | |
| **Use-case No.** | QAPE001 | **Use-case Version** | 2.0 |
| **Use-case Name** | Login | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | High |
| **Actor:**   * Guest   **Summary:**   * This use case allows guest login to the system   **Goal:**   * Allows guest login to the system   **Triggers**   * Guest clicks Login button on Welcome screen.   **Preconditions:**   * N/A   **Post Conditions:**   * **On Success**: Guest logged in to the system. * **On Failure**: Show Login screen with error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | System displays Welcome screen:  - “Email”: textbox, min length 8, max length 255.  - “Password”: textbox, min length: 8, max length 50.  -“Login”: button. | | 2 | Guest inputs email and password |  | | 3 | Click “Login” button | System navigates to News feed screen.  [Exceptions 1, 2, 3] |   **Alternative Scenario:**   * -N/A   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Guest leaves “Email” or “Password” textboxes blank. | Show login screen with message: “Invalid email or password”. | | 2 | Guest logins with wrong username or password. | Show login screen with message: “Invalid email or password”. | | 3 | Guest logins with non-existed account. | Show login screen with message: “Invalid email or password”. |   **Relationships:**   * N/A   **Business Rules:**   * Guest have to login with Student account, Teacher account or Admin account to access the web application. | | | |

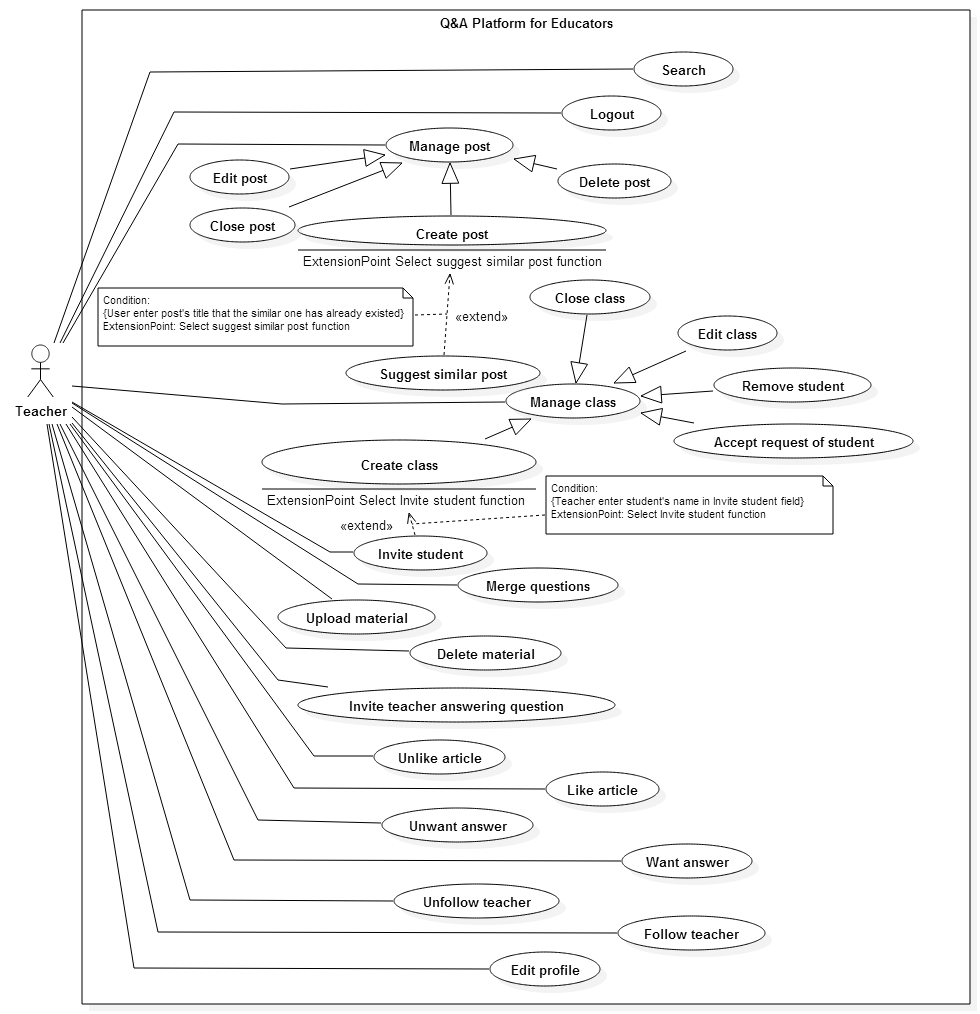
##### <Guest> Register Use Case Diagram



**Figure 5: <Guest> Register Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -2 SPECIFICATION** | | | |
| **Use-case No.** | QAPE002 | **Use-case Version** | 2.0 |
| **Use-case Name** | Register | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | High |
| **Actor:**   * Guest   **Summary:**   * This use case allows guest register an account   **Goal:**   * Allows guest login register an account   **Triggers**   * Guest clicks Register button on Welcome screen.   **Preconditions:**   * N/A   **Post Conditions:**   * **On Success**: New account is created and guest logged in to the system. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 |  | System displays Welcome screen:  - “Email\*”: textbox, min length 8, max length 255, regular expression: “^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$" required.  - “Password\*”: textbox, min length: 8, max length 50, required.  - “Confirm password\*”: textbox, min length: 8, max length 50, required.  - “Knowledge about\*”: drop down list, required.  - “You are\*”: select radio button, required.  -“Register”: button. | | 2 | Guest inputs all required fields |  | | 3 | Click “Register” button | New account is created and system navigates to News feed screen.  [Exceptions 1, 2, 3, 4, 5]  [Alternative 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Guest enters existed email address and clicks Register button | Show message: “This email has been used”. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Guest leaves “Email\*” blank or enters one not matched regular expression | Show error message: “Please enter a valid email address.” | | 2 | Guest leaves “Password\*” or “Confirm password\*” blank | Show error message: “Please provide a password. | | 3 | Guest enters “Password\*” or “Confirm password\*” is not matched min and max length | Show error message: “Your password must be between 8 and 50 characters long.” | | 4 | Guest enters “Password\*” and “Confirm password\*” are not matched | Show error message: “Please enter the same password as above.” | | 5 | Guest leaves “Knowledge about\*” as default | Show error message: “Please choose your knowledge.” |   **Relationships:**   * N/A   **Business Rules:**   * All required fields must be filled. * Email address must be unique. * Password and confirm password must be same. | | | |

#### <Teacher> Overview Use Case

****

**Figure 6: <Teacher> Overview Use Case**

##### <Teacher> Create class Use Case Diagram

****

**Figure 7: <Teacher> Create class Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -3 SPECIFICATION** | | | |
| **Use-case No.** | QAPE003 | **Use-case Version** | 2.0 |
| **Use-case Name** | Create class | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user create the classroom and invite student to join classroom   **Goal:**   * Create new classroom   **Triggers**   * User clicks “Create class” button in Dashboard screen.   **Preconditions:**   * User must login as Teacher role.   **Post Conditions:**   * **On Success**: New classroom is created. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Dashboard screen and clicks “Create class” button | System displays Create classroom screen:  - “Class name\*”: textbox, min length 20, max length 255, required.  - “Tag\*”: textbox, required.  - “Invite student”: textbox.  - “Category\*”: drop down list, required.  - “Class’s Description\*”: text-area, , min length 20, max length 350, required.  - “Create class”: button. | | 2 | User inputs all required fields |  | | 3 | Click “Create class” button | New classroom created and system navigates to classroom screen.  [Exceptions 1, 2, 3, 4, 5, 6] |   **Alternative Scenario:**   * N/A   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Class name\*” blank | Show error message: “Please provide the class's name.” | | 2 | User enters “Class name\*” is not match min and max length | Show error message: “The class name must be between 20 and 255 characters long.” | | 3 | User leaves “Tag\*” blank | Show error message: “Please provide at least one tag.” | | 4 | User leaves “Category\*” as default | Show error message: “Please select category.” | | 5 | User leaves “Class’s description\*” blank | Show error message: “Please provide the description.” | | 6 | User enters “Class’s description\*” is not match min and max length | Show error message: “The description must be between 20 and 255 characters long.” |   **Relationships:**   * Extend for Invite student (Select invite student function)   **Business Rules:**   * All required fields must be filled. * Number of Tag is not over 5. * Duplicated tags in Tag or students in Invite student do not allow. * Invite student can be blank. | | | |

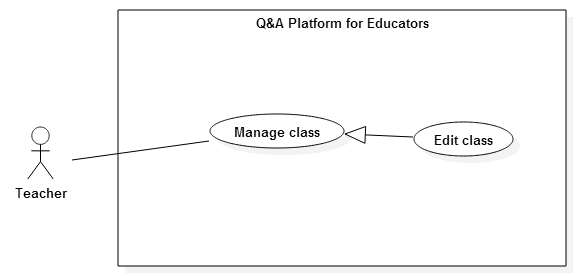
##### <Teacher> Invite student Use Case Diagram

****

**Figure 7: <Teacher> Invite student Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -4 SPECIFICATION** | | | |
| **Use-case No.** | QAPE004 | **Use-case Version** | 2.0 |
| **Use-case Name** | Invite student | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user invite students to join classroom   **Goal:**   * Invite students to join classroom   **Triggers**   * User clicks “Invite” button in Classroom screen. * User enters student’s name in Invite student when Create classroom.   **Preconditions:**   * User must login as Teacher role. * In case of invite student in Classroom screen, classroom must be already created, in “open” state and owned by user.   **Post Conditions:**   * **On Success**: Students are invited received invitation. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks “Invite” button | System displays popup “Invite student to join class”:  - “Student\*”: textbox, required  - “Invite”: button.  - “x” icon: button.  [Alternative 1] | | 2 | User inputs student’s name | System show drop down list with suggest students base on student’s name | | 3 | User chooses student from drop down list |  | | 4 | Click “Invite” button | Popup is closed and invited students receive invitations.  [Alternative 2]  [Exceptions 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks “x” icon | Popup is closed and no student is invited | | 2 | User types in text box instead of chooses from dropdown list | Popup is closed and no student is invited |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Student\*” blank | Show error message: “Please provide at least one student.” |   **Relationships:**   * Extend by Create class   **Business Rules:**   * User must choose student from suggest drop down list instead of types text. * Suggested students must be active accounts. | | | |

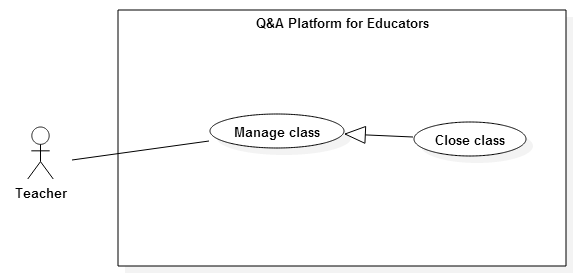
##### <Teacher> Edit class Use Case Diagram

****

**Figure 9: <Teacher> Edit class Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -5 SPECIFICATION** | | | |
| **Use-case No.** | QAPE005 | **Use-case Version** | 2.0 |
| **Use-case Name** | Edit class | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | Medium |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user update owned classroom’s information   **Goal:**   * Update classroom’s information   **Triggers**   * User clicks setting icon and clicks “Update class” in Classroom screen.   **Preconditions:**   * User must login as Teacher role. * Classroom must be in “open” state and owned by user.   **Post Conditions:**   * **On Success**: Classroom’s information is updated * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks setting icon | System displays drop down list | | 2 | User clicks “Update class”  [Alternative 1] | System displays Update classroom screen:  - “Tag\*”: textbox, required.  - “Category\*”: drop down list, required.  - “Class’s Description\*”: text-area, , min length 20, max length 350, required.  - “Update class”: button. | | 3 | User updates classroom’s information |  | | 4 | Click “Update class” button | Classroom’s information is updated and system navigates to classroom screen.  [Exceptions 1, 2, 3, 4] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks different place in page or clicks setting icon again | Drop down list is hidden |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Tag\*” blank | Show error message: “Please provide at least one tag.” | | 2 | User leaves “Category\*” as “Select a type” | Show error message: “Please select category.” | | 3 | User leaves “Class’s description\*” blank | Show error message: “Please provide the description.” | | 4 | User enters “Class’s description\*” is not match min and max length | Show error message: “The description must be between 20 and 255 characters long.” |   **Relationships:**   * N/A   **Business Rules:**   * All required fields must be filled. * Number of Tag is not over 5. * User cannot input duplicated tags. * User cannot edit classroom’s name. | | | |

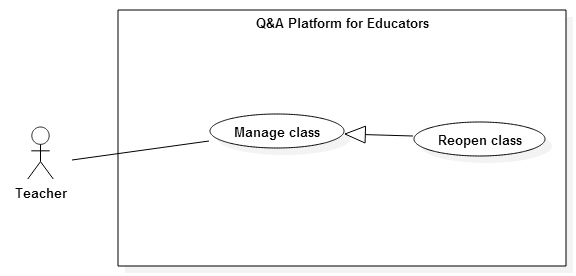
##### <Teacher> Close class Use Case Diagram

****

**Figure 10: <Teacher> Close class Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -6 SPECIFICATION** | | | |
| **Use-case No.** | QAPE006 | **Use-case Version** | 2.0 |
| **Use-case Name** | Close class | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | Medium |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user close owned classroom   **Goal:**   * Close the classroom to prevent more request, question, article or material   **Triggers**   * User clicks setting icon and clicks “Close class” in Classroom screen.   **Preconditions:**   * User must login as Teacher role. * Classroom must be in “open” state and owned by user.   **Post Conditions:**   * **On Success**: Classroom status is changed to close * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks setting icon | System displays drop down list | | 2 | User clicks “Close class”  [Alternative 1] | Classroom status is changed to close  System displays Classroom screen with no “Create post” button and show “Classroom is closed” in “About classroom” widget |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks different place in page or clicks setting icon again | Drop down list is hidden. Nothing change |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User cannot create new article/question in closed classroom * Student cannot request to join closed classroom * User cannot edit classroom information when it is closed * User can manage question, article, material, request, student in closed classroom except create new question or article, upload material and invite student. | | | |

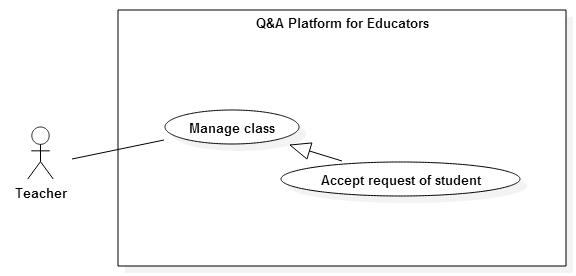
##### <Teacher> Reopen class Use Case Diagram

****

**Figure 11: <Teacher> Reopen class Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -7 SPECIFICATION** | | | |
| **Use-case No.** | QAPE007 | **Use-case Version** | 2.0 |
| **Use-case Name** | Reopen class | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | Medium |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user reopen closed classroom   **Goal:**   * Reopen closed classroom   **Triggers**   * User clicks setting icon and clicks “Reopen class” in Classroom screen.   **Preconditions:**   * User must login as Teacher role. * Classroom must be in “close” state and owned by teacher.   **Post Conditions:**   * **On Success**: Classroom status is changed to open * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks setting icon | System displays drop down list | | 2 | User clicks “Reopen class”  [Alternative 1] | System displays Classroom screen with full classroom management functions |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks different place in page or clicks setting icon again | Drop down list is hidden. Nothing change |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User has full management functions with owned opened classroom * Student can send request to join opened classroom | | | |

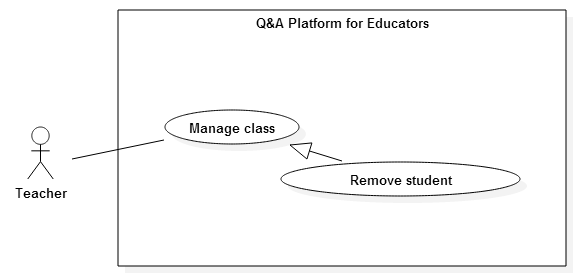
##### <Teacher> Accept request of student Use Case Diagram

****

**Figure 12: <Teacher> Accept request of student Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -8 SPECIFICATION** | | | |
| **Use-case No.** | QAPE008 | **Use-case Version** | 2.0 |
| **Use-case Name** | Accept request of student | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user accept joining requests of students   **Goal:**   * Accept students’ requests   **Triggers**   * User clicks Confirm button on request.   **Preconditions:**   * User must login as Teacher role. * Classroom must be owned by user and have at least one joining request from student.   **Post Conditions:**   * **On Success**: Request is removed and student is added to classroom * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks “Join Request” tab | System displays list of requests from students. | | 2 | User clicks “Confirm” button on request that user wants to accepts  [Alternative 1] | Message “You have confirmed the request of <student’s name>” is showed and this student is added to classroom. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks “Ignore” button | Message “You have ignored the request of <student’s name>” is showed and the request is removed |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User can confirm or ignore only with students’ requests to joined owned classrooms * User cannot confirm the request after ignoring * User can invite students after ignoring their requests | | | |

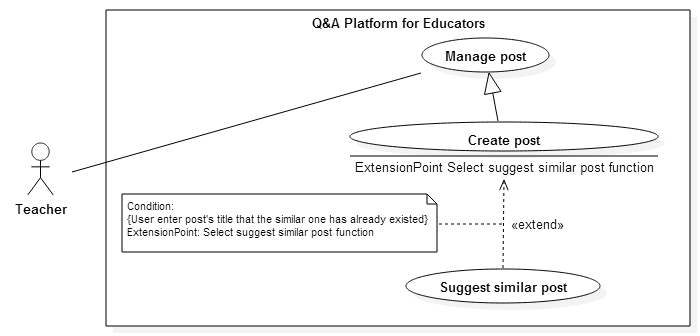
##### <Teacher> Remove student Use Case Diagram

****

**Figure 13: <Teacher> Remove student Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -9 SPECIFICATION** | | | |
| **Use-case No.** | QAPE009 | **Use-case Version** | 2.0 |
| **Use-case Name** | Remove student | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user remove students from classroom   **Goal:**   * Remove students   **Triggers**   * User clicks “Remove” button on student.   **Preconditions:**   * User must login as Teacher role. * Classroom must be owned by user and have at least one student.   **Post Conditions:**   * **On Success**: Student is removed from classroom * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks tab “Student” | System displays list of students in classroom. | | 2 | User clicks “Remove” button on student that user wants to remove | System displays “Remove student” popup:  - “OK”: button  - “Cancel”: button | | 3 | User clicks “OK” button  [Alternative 1] | Message “You have remove <student’s name> from classroom.” is showed and student is removed from classroom |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks “Cancel” button | Popup disappears and no student is removed. |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User can only remove students in owned classrooms. * User cannot add students after removing them. * User can invite students after removing them. | | | |

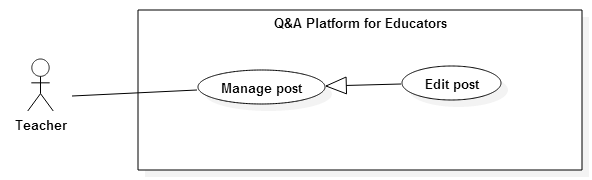
##### <Teacher> Create post Use Case Diagram

****

**Figure 14: <Teacher> Create post Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 10 SPECIFICATION** | | | |
| **Use-case No.** | QAPE010 | **Use-case Version** | 2.0 |
| **Use-case Name** | Create post | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 16/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user create new question or article in classroom.   **Goal:**   * Create new question or article   **Triggers**   * Teacher clicks “Create post” button in Classroom screen.   **Preconditions:**   * User must login as Teacher role. * User is owner of classroom * Classroom is “open” state   **Post Conditions:**   * **On Success**: New question or article is created. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks “Create post” button | System displays “Create post” screen:  - “Title\*”: textbox, min length 20, max length 255, required.  - “Tag\*”: textbox, required.  - “Type\*”: drop down list, required.  - “Detail\*”: text-area, min length 120, required.  - “Publish your post”: button. | | 2 | User inputs all required fields | After inputting “Title\*”, system suggest posts have similar titles  [Alternative 1] | | 3 | Click “Publish your post” button  [Alternative 2] | New article/question (base on Type) is created and system navigates to article/question screen.  [Exceptions 1, 2, 3, 4, 5, 6] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 |  | Nothing is suggested when there are no post have similar title | | 2 | User clicks post’s title in suggested list. | System opens new tab in browser and show this post |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Title\*” blank | Show error message: “Please provide the title.” | | 2 | User enters “Title\*” is not match min and max length | Show error message: “The class name must be between 20 and 255 characters long.” | | 3 | User leaves “Tag\*” blank | Show error message: “Please provide at least one tag.” | | 4 | User leaves “Type\*” as default | Show error message: “Please choose question or article.” | | 5 | User leaves “Details\*” blank | Show error message: “Please provide the details.” | | 6 | User enters “Details \*” is not match min length | Show error message: “The details must be at least 120 characters long.” |   **Relationships:**   * Extend for suggest similar post (Select suggest similar post function)   **Business Rules:**   * Post is only created if classroom is open. * User can only create post in owned classroom where post will be created. * All required fields must be filled. * Number of Tag is not over 5. * User cannot input duplicated tags. | | | |

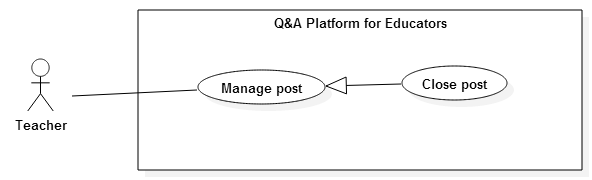
##### <Teacher> Edit post Use Case Diagram

****

**Figure 15: <Teacher> Edit post Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 11 SPECIFICATION** | | | |
| **Use-case No.** | QAPE011 | **Use-case Version** | 2.0 |
| **Use-case Name** | Edit post | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 16/07/2015 | **Priority** | Medium |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user edit owned question or article   **Goal:**   * Edit question or article   **Triggers**   * User clicks setting icon and clicks “Edit” button in Question or Article screen.   **Preconditions:**   * User must login as Teacher role. * User is owner of question or article. * Article or question in “Open” state.   **Post Conditions:**   * **On Success**: Question or Article is updated * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks “Create post” button | System displays Update post screen:  - “Title\*”: textbox, min length 20, max length 255, required.  - “Tag\*”: textbox, required.  - “Type\*”: drop down list, required.  - “Detail\*”: text-area, min length 120, required.  - “Save”: button. | | 2 | User changes fields |  | | 3 | Click “Save” button | Question/Article is updated and system navigates to question/article screen  [Exceptions 1, 2, 3, 4, 5, 6] |   **Alternative Scenario:**   * N/A   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Title\*” blank | Show error message: “Please provide the title.” | | 2 | User enters “Title\*” is not match min and max length | Show error message: “The class name must be between 20 and 255 characters long.” | | 3 | User leaves “Tag\*” blank | Show error message: “Please provide at least one tag.” | | 4 | User leaves “Type\*” as default | Show error message: “Please choose question or article.” | | 5 | User leaves “Details\*” blank | Show error message: “Please provide the details.” | | 6 | User enters “Details \*” is not match min length | Show error message: “The details must be at least 120 characters long.” |   **Relationships:**   * N/A   **Business Rules:**   * User can only edit owned post. * All required fields must be filled. * Number of Tag is not over 5. * User cannot input duplicated tags. | | | |

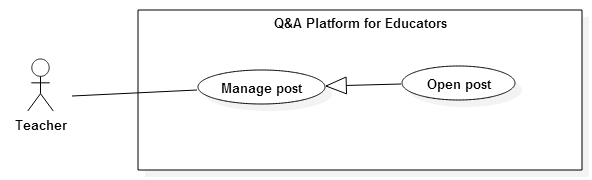
##### <Teacher> Close post Use Case Diagram

****

**Figure 16: <Teacher> Close post Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 12 SPECIFICATION** | | | |
| **Use-case No.** | QAPE012 | **Use-case Version** | 2.0 |
| **Use-case Name** | Close post | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 16/07/2015 | **Priority** | Medium |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user close question or article   **Goal:**   * Close question or article to prevent more answers   **Triggers**   * User clicks setting icon and clicks “Close” in Question or Article screen.   **Preconditions:**   * User must login as Teacher role. * Question or article is in user’s owned classroom. * Article or question in “Open” state.   **Post Conditions:**   * **On Success**: Question or Article is closed * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Question/Article screen and clicks setting icon. | System displays drop down list. | | 2 | User clicks “Close”.  [Alternative 1] | System displays question/article screen with no “Leave a reply” area.  [Alternative 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks different place in page or clicks setting icon again. | Drop down list is hidden. Nothing is changed. | | 2 |  | If post is question, “Ask teacher” widget will be hidden too. |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User cannot leave answer in closed post or edit closed post. * User can delete closed owned post. * User can close all posts in owned classroom | | | |

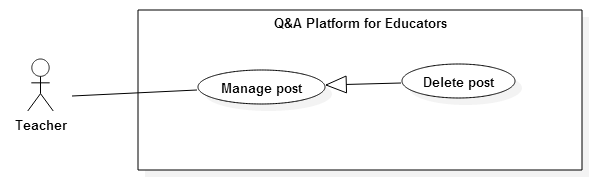
##### <Teacher> Open post Use Case Diagram

****

**Figure 17: <Teacher> Open post Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 13 SPECIFICATION** | | | |
| **Use-case No.** | QAPE013 | **Use-case Version** | 2.0 |
| **Use-case Name** | Open post | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 16/07/2015 | **Priority** | Medium |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user open question or article   **Goal:**   * Open question or article   **Triggers**   * User clicks setting icon and clicks “Open” in Question or Article screen.   **Preconditions:**   * User must login as Teacher role. * Question or article is in user’s owned classroom. * Article or question in “Close” state.   **Post Conditions:**   * **On Success**: Question or Article is opened * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Question/Article screen and clicks setting icon. | System displays drop down list. | | 2 | User clicks “Open”.  [Alternative 1] | System displays question/article screen with “Leave a reply” area.  [Alternative 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks different place in page or clicks setting icon again. | Drop down list is hidden. Nothing is changed. | | 2 |  | If post is question, “Ask teacher” will be display too. |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User can reply or if post is question, teacher can also invite other teachers to answer it. | | | |

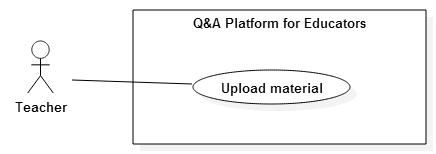
##### <Teacher> Delete post Use Case Diagram

****

**Figure 18: <Teacher> Delete post Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 14 SPECIFICATION** | | | |
| **Use-case No.** | QAPE014 | **Use-case Version** | 2.0 |
| **Use-case Name** | Delete post | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 16/07/2015 | **Priority** | Medium |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user delete owned question or article   **Goal:**   * Delete question or article   **Triggers**   * User clicks setting icon and clicks “Delete” button in Question or Article screen.   **Preconditions:**   * User must login as Teacher role. * User is owner of question or article.   **Post Conditions:**   * **On Success**: Question or Article is deleted * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Question/Article screen and clicks setting icon | System displays drop down list | | 2 | User clicks “Delete” | System displays “Delete post” popup:  - “OK”: button  - “Cancel”: button | | 3 | User clicks “OK” button  [Alternative 1] | Post is removed and system navigates to post’s owned classroom screen. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks “Cancel” button | Popup disappears and no student is removed. |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User cannot leave answer in closed post or edit closed post. * User cannot delete other users’ posts. | | | |

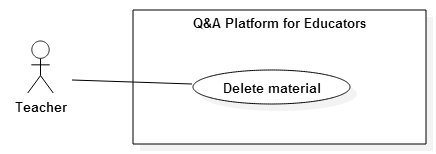
##### <Teacher> Upload material Use Case Diagram

****

**Figure 19: <Teacher> Upload material Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -15 SPECIFICATION** | | | |
| **Use-case No.** | QAPE015 | **Use-case Version** | 2.0 |
| **Use-case Name** | Upload material | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 16/07/2015 | **Priority** | Medium |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user upload material in owned classroom   **Goal:**   * Upload material   **Triggers**   * User clicks Upload button in Material tab.   **Preconditions:**   * User must login as Teacher role. * Classroom must be in “open” state and owned by user.   **Post Conditions:**   * **On Success**: New material is uploaded * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks tab “Material” | System displays list of materials in classroom. | | 2 | User clicks “Upload” button | System displays “Upload material” popup:  - “Tag\*”: textbox, required.  - “Material\*”: file input, required  - “x” icon: button  - “Choose File”: button  - “Upload”: button | | 3 | User inputs “Tag\*”  [Alternative 1] |  | | 4 | User clicks “Choose file” and select file from computer | System displays name of selected material. | | 5 | User clicks “Upload” button | Material is uploaded and systems show it in material list  [Exception 1, 2] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks “x” icon button | Popup disappears and no material is uploaded. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Tag\*” blank | Show error message: “Please provide at least one tag.” | | 2 | User doesn’t choose material to upload | Show error message: “Please choose material.” |   **Relationships:**   * N/A   **Business Rules:**   * Teach User er can only upload material in owned opened classrooms. * User can only upload one material at time. * Number of Tag is not over 5. * Duplicated tags in Tag do not allow. | | | |

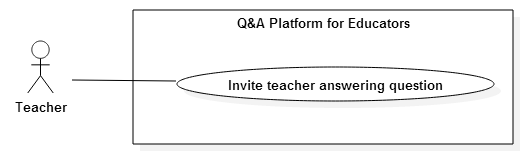
##### <Teacher> Delete material Use Case Diagram

****

**Figure 20: <Teacher> Delete material Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -16 SPECIFICATION** | | | |
| **Use-case No.** | QAPE016 | **Use-case Version** | 2.0 |
| **Use-case Name** | Delete material | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user delete material in owned classroom   **Goal:**   * Delete materials   **Triggers**   * User clicks “Delete” button in Material tab.   **Preconditions:**   * User must login as Teacher role. * Classroom must be owned by teacher and have at least one material.   **Post Conditions:**   * **On Success**: Material is deleted * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks tab “Material” | System displays list of materials. | | 2 | User clicks “Delete” button | System displays “Delete material” popup:  - “OK”: button  - “Cancel”: button | | 3 | User clicks “OK” button  [Alternative 1] | Material is deleted and disappears in material list |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks “Cancel” button | Popup disappears and no material is deleted. |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User can only delete materials in owned classrooms. * Deleted material is only deleted from classrooms; it’s still available in student’s folder if it has already added to. | | | |

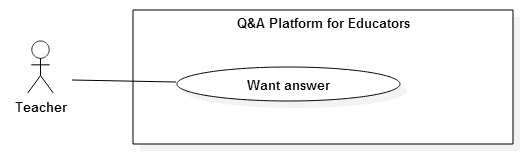
##### <Teacher> Invite teacher answering question Use Case Diagram

****

**Figure 21: <Teacher> Invite teacher answering question Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -17 SPECIFICATION** | | | |
| **Use-case No.** | QAPE017 | **Use-case Version** | 2.0 |
| **Use-case Name** | Invite teacher answering question | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 15/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user invite another teachers to join into discussion in question   **Goal:**   * Invite teacher to answering question   **Triggers**   * User clicks “Ask teacher” button in Question screen.   **Preconditions:**   * User must login as Teacher role. * Question must be in “Open” state and in teacher’s owned classroom.   **Post Conditions:**   * **On Success**: Teachers are invited received invitation and can discuss in question. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Question screen and clicks “Ask teacher” button | System displays popup “Ask teacher student to join class”:  - “Teacher\*”: textbox, required  - “Invite”: button.  - “x” icon: button.  [Alternative 1] | | 2 | User inputs student’s name | System show drop down list with suggest teachers base on teacher’s name | | 3 | User chooses another teacher from drop down list |  | | 4 | Click “Invite” button | Popup is closed, invited teachers receive invitations and they can answer this question.  [Alternative 2]  [Exceptions 1] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks “x” icon | Popup is closed and no teacher is invited | | 2 | User types in text box instead of chooses from dropdown list | Popup is closed and no teacher is invited |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Teacher\*” blank | Show error message: “Please provide at least one teacher.” |   **Relationships:**   * N/A   **Business Rules:**   * User must choose teacher from suggest drop down list instead of types text. * Suggested teachers must be active accounts. * User can invite other teachers only in question and this question must be opened. | | | |

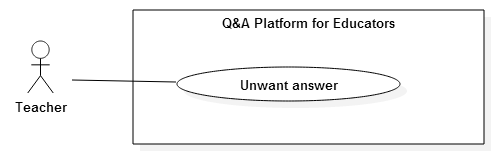
##### <Teacher> Want answer Use Case Diagram

****

**Figure 22: <Teacher> Want answer Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -18 SPECIFICATION** | | | |
| **Use-case No.** | QAPE018 | **Use-case Version** | 2.0 |
| **Use-case Name** | Want answer | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user want answer to increase the interest of question   **Goal:**   * Want answer to increase the interest of question   **Triggers**   * User checks in un-checked “Want answer” checkbox   **Preconditions:**   * User must login as Teacher role.   **Post Conditions:**   * **On Success**: Number of “Want answer” increases 1 * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Question screen and checks in “Want answer(s)” check box. | The number of “Want answer” increases 1 and checkbox is checked |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * Closed question allows “Want answer” function * User cannot use “Want answer” twice with same question (one user only can increase 1 of “Want answer” number) * User can use “Want answer” function with any question in any classroom | | | |

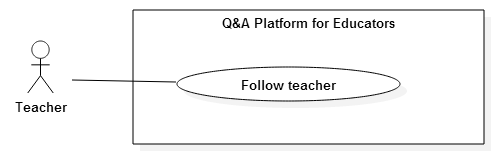
##### <Teacher> Un-want answer Use Case Diagram

****

**Figure 23: <Teacher> Un-want answer Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -19 SPECIFICATION** | | | |
| **Use-case No.** | QAPE019 | **Use-case Version** | 2.0 |
| **Use-case Name** | Un-want answer | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows teacher un-want answer to decrease the interest of question   **Goal:**   * Un-want answer to increase the interest of question   **Triggers**   * User unchecks in checked “Want answer” checkbox   **Preconditions:**   * User must login as Teacher role.   **Post Conditions:**   * **On Success**: Number of “Want answer” decreases 1 * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Question screen and unchecks in “Want answer(s)” checkbox. | The number of “Want answer” decreases 1 and checkbox is un-checked |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * Closed question allows “Un-want answer” function * User can only uncheck the checked checkbox. | | | |

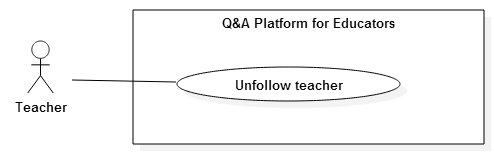
##### <Teacher> Follow teacher Use Case Diagram

****

**Figure 24: <Teacher> Follow teacher Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -20 SPECIFICATION** | | | |
| **Use-case No.** | QAPE020 | **Use-case Version** | 2.0 |
| **Use-case Name** | Follow teacher | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user follow other teachers to get more interest question for news feed   **Goal:**   * Follow teacher   **Triggers**   * User click “Follow” button in other teacher’s profile screen   **Preconditions:**   * User must login as Teacher role.   **Post Conditions:**   * **On Success**: Teacher follows another one. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to another one’s Profile screen and clicks “Follow” button | Teacher follows another one and button “Follow” change to “Unfollow” |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User cannot follow himself. * Only teachers can be followed. * Deactivated teacher accounts cannot be followed. | | | |

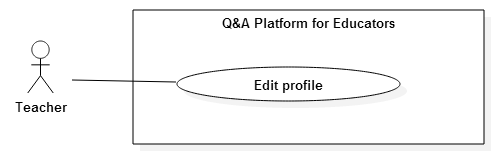
##### <Teacher> Un-follow teacher Use Case Diagram

****

**Figure 25: <Teacher> Un-follow teacher Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -21 SPECIFICATION** | | | |
| **Use-case No.** | QAPE021 | **Use-case Version** | 2.0 |
| **Use-case Name** | Un-follow teacher | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | Medium |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user un-follow other teachers to prevent get more news from them   **Goal:**   * Un-follow teacher   **Triggers**   * User click “Unfollow” button in other teacher’s profile screen   **Preconditions:**   * User must login as Teacher role.   **Post Conditions:**   * **On Success**: Teacher un-follows another one. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to another one’s Profile screen and clicks “Unfollow” button  [Alternative 1] | Teacher un-follows another one and button “Unfollow” change to “Follow” |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User goes to dashboard screen, clicks on tab “Followed teacher”. | System show list of followed teachers | |  | User clicks “Unfollow” button on teacher that user want to un-follow. | Message “You have unfollow <teacher’s name>” is showed and teacher un-follows another one. |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * Only followed teachers can be un-followed. * Deactivated teacher accounts can be un-followed. | | | |

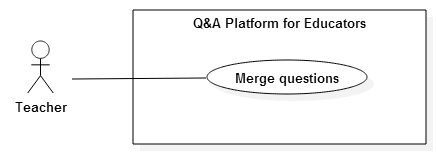
##### <Teacher> Edit profile Use Case Diagram

****

**Figure 26: <Teacher> Edit profile Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -22 SPECIFICATION** | | | |
| **Use-case No.** | QAPE022 | **Use-case Version** | 2.0 |
| **Use-case Name** | Edit profile | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user edit profile to update information   **Goal:**   * Edit profile   **Triggers**   * Teacher clicks “Edit profile” button on Profile screen.   **Preconditions:**   * User must login as Teacher role.   **Post Conditions:**   * **On Success**: Profile is updated. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Profile screen and clicks “Edit profile” button | System displays “Edit profile” screen:  - “Profile Picture”: file input.  - “Display Name\*”: textbox, min length: 8, max length: 30, required  - “Password\*”: textbox, min length: 8, max length 50, required.  - “Confirm password\*”: textbox, min length: 8, max length 50, required.  - “Knowledge about\*”: drop down list, required.  - “Your tags”: textbox  - “About Yourself”: text-area  -“Update”: button. | | 2 | User changes fields |  | | 3 | Click “Update” button | Profile is updated and system refreshes “Edit profile” screen with updated information  [Exception 1, 2, 3, 4, 5, 6] |   **Alternative Scenario:**   * N/A   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Display Name\*” blank | Show error message: “Please provide the display name.” | | 2 | User enters “Display Name\*” not match min and max length | Show error message: “The display name must be between 8 and 30 characters long.” | | 3 | User leaves “Password\*” or “Confirm password\*” blank | Show error message: “Please provide a password.” | | 4 | User enters “Password\*” or “Confirm password\*” is not matched min and max length | Show error message: “Your password must be between 8 and 50 characters long.” | | 5 | User enters “Password\*” and “Confirm password\*” are not matched | Show error message: “Please enter the same password as left.” | | 6 | User leaves “Knowledge about\*” as default | Show error message: “Please choose your knowledge.” |   **Relationships:**   * N/A   **Business Rules:**   * All required fields must be filled. * Password and confirm password must be same. * User can only edit his profile. * After profile is updated, news feed content changed earliest after 30 minutes | | | |

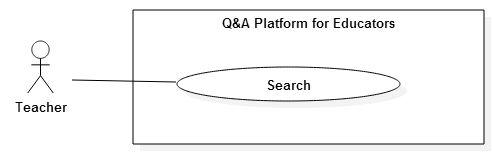
##### <Teacher> Merge question Use Case Diagram

****

**Figure 27: <Teacher> Merge question Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -23 SPECIFICATION** | | | |
| **Use-case No.** | QAPE023 | **Use-case Version** | 2.0 |
| **Use-case Name** | Merge question | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user merge multiple questions into one.   **Goal:**   * Merge multiple questions into one in order to prevent similar question in classroom   **Triggers**   * User clicks “Merge question” button on tab Question.   **Preconditions:**   * User must login as Teacher role. * User is owner of classroom. * Classroom has at least 2 questions.   **Post Conditions:**   * **On Success**: Main merged question is updated with new information. Other merge questions are closed * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks tab Question | System displays list of questions in classroom | | 2 | User clicks “Merge question”  [Alternative 1] | System navigates to “Merge questions” screen:  - Range bar: slide bar to control the percentage of similar questions.  - List of similar questions with checkbox: required  - “Merge checked questions”: button  - “Answer checked questions”: text area, min length 30  - “Submit your answer”: button | | 3 | User select questions and clicks “Merge checked questions”  [Alternative 2] | System navigates to “New merged question” screen:  - Reused question: radio button, required.  - “Title\*”: textbox, min length 20, max length 255, required.  - “Tag\*”: textbox, required.  - “Detail\*”: text-area, min length 120, required.  - “Save”: button. | | 4 | User inputs all required field |  | | 5 | Clicks “Save” button | Reused question is saved with new information.  Other merged questions are closed and have commented “This question was merged and answered by <teacher’s name>; you can read it in this Question”.  System navigates to “Question” screen with new information of reused question.  [Exception 1, 2, 3, 4, 5, 6, 7, 9] |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks “<Number> similar question(s)” on question that user want to merge | [Main success 3] | | 2 | User inputs “Answer checked question” text-area and click “Save” button  [Exception 9, 10] | All checked questions have answered with same answer that user has inputted. |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Title\*” blank | Show error message: “Please provide the title.” | | 2 | User enters “Title\*” is not match min and max length | Show error message: “The class name must be between 20 and 255 characters long.” | | 3 | User leaves “Tag\*” blank | Show error message: “Please provide at least one tag.” | | 4 | User leaves “Details\*” blank | Show error message: “Please provide the details.” | | 5 | User enters “Details \*” is not match min length | Show error message: “The details must be at least 120 characters long.” | | 6 | User doesn’t select any question | Show error message: “Please select at least two questions.” | | 7 | User selects only one question | Show error message: “Please select at least two questions.” | | 8 | User leaves “Answer checked question” text-area blank | Show error message: “Please provide the answer.” | | 9 | User enters Answer checked question” text-area not match min length | Show error message: “The answer must be at least 30 characters long.” | | 10 | User doesn’t select any question | Show error message: “Please select at least one question.” |   **Relationships:**   * N/A   **Business Rules:**   * All required fields must be filled. * User can only merge question in owned classroom. * Resolved questions and closed questions cannot be merged. * Students must receive notification if their questions are merged. | | | |

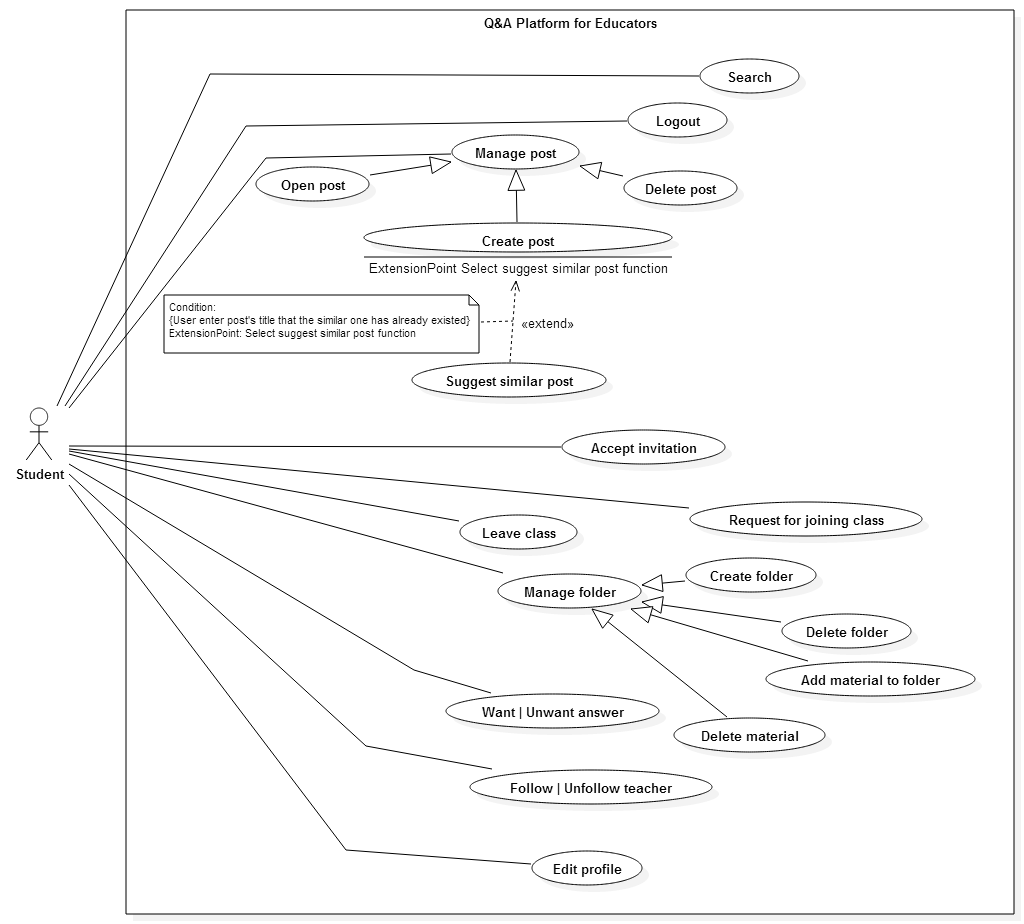
##### <Teacher> Search Use Case Diagram

****

**Figure 28: <Teacher> Search Use Case Diagram**

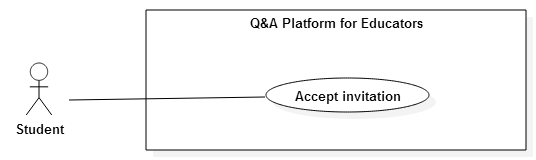
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -24 SPECIFICATION** | | | |
| **Use-case No.** | QAPE024 | **Use-case Version** | 2.0 |
| **Use-case Name** | Search | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Teacher   **Summary:**   * This use case allows user search question, article, material, classroom, teacher and student.   **Goal:**   * Search question, material, classroom, teacher, student in system   **Triggers**   * User clicks “Search” button on News feed screen.   **Preconditions:**   * User must login as Teacher role.   **Post Conditions:**   * **On Success**: Results of searching is showed. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to News feed screen  [Alternative 1] | System displays “News feed” screen with search area:  - Search filter: dropdown list  - Search textbox  - “Search”: button | | 2 | User chooses filter and inputs search key |  | | 3 | Click “Search” button  [Alternative 2] | System navigates to Search result screen and displays search results |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User goes to Classroom screen | System display “Classroom” screen with search textbox | |  | User enters search key and presses “Enter” key | System display search results in this classroom | | 2 | Press “Enter” key | System navigates to Search result screen and displays search results |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * Search results also are closed classrooms, questions, article; deactivated teacher, student; | | | |

#### <Student> Overview Use Case

****

**Figure 29: <Student> Overview Use Case**

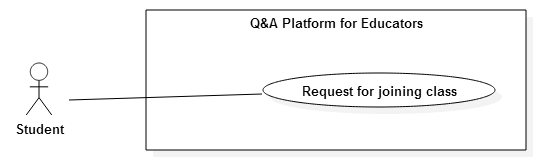
##### <Student> Accept Invitation Use Case Diagram

****

**Figure 30: <Student> Accept Invitation Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -25 SPECIFICATION** | | | |
| **Use-case No.** | QAPE025 | **Use-case Version** | 2.0 |
| **Use-case Name** | Accept Invitation | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Student   **Summary:**   * This use case allows user accept invitation to join classroom from teacher   **Goal:**   * Accept invitation to join classroom   **Triggers**   * User clicks “Confirm” button on invitation.   **Preconditions:**   * User must login as Student role. * Student has at least one invitation.   **Post Conditions:**   * **On Success**: Student joined in classroom. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Dashboard screen and clicks “Invitation” tab  [Alternative 1] | System displays list of invitations from teachers | | 2 | User clicks “Confirm” button on invitation that student wants to accept  [Alternative 2] | Message “You have joined <classroom’s name>” is showed and student is added to this classroom. |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User goes to Classroom screen that is invited and clicks “Accept invitation” button | “Accept invitation” button is hidden  “Leave” button is showed  “Create post” button is showed | | 2 | User clicks “Ignore” button | Message “You have ignored the invitation to join <classroom’s name>” is showed and the invitation is removed |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User cannot confirm the invitation after ignoring * User can be invited again to join the classroom that user has ignored * After confirming invitation, news feed content changed earliest after 30 minutes * User can only create post in classroom after joining classroom (by accepting teacher’s invitation or after teacher accepts student’s request) | | | |

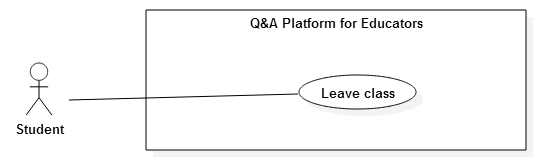
##### <Student> Request for joining class Use Case Diagram

****

**Figure 31: <Student> Request for joining class Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -26 SPECIFICATION** | | | |
| **Use-case No.** | QAPE026 | **Use-case Version** | 2.0 |
| **Use-case Name** | Request for joining class | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Student   **Summary:**   * This use case allows user request teacher for joining favorite classroom   **Goal:**   * Request for joining classroom   **Triggers**   * User clicks “Join” button on Classroom screen.   **Preconditions:**   * User must login as Student role.   **Post Conditions:**   * **On Success**: Request is sent to teacher. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks “Join” button | The request for joining classroom is sent to teacher  “Join” button is hidden  “Request sent” button is show |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User can only request for joining classroom if user has not yet invited by teacher. * User can only create post in classroom after joining classroom (by accepting teacher’s invitation or after teacher accepts student’s request) | | | |

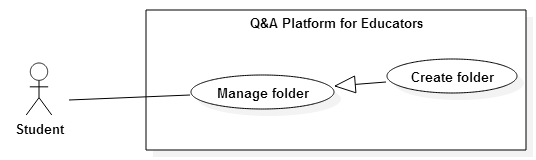
##### <Student> Leave class Use Case Diagram

****

**Figure 32: <Student> Leave class Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -27 SPECIFICATION** | | | |
| **Use-case No.** | QAPE027 | **Use-case Version** | 2.0 |
| **Use-case Name** | Leave class | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Student   **Summary:**   * This use case allows user leave joined classroom   **Goal:**   * Leave joined classroom   **Triggers**   * User clicks “Leave” button in Classroom screen.   **Preconditions:**   * User must login as Student role. * User has joined at least one classroom   **Post Conditions:**   * **On Success**: Request is sent to teacher. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks “Leave” button  [Alternative 1] | System displays “Leave classroom” popup:  - “OK”: button  - “Cancel”: button | | 2 | User clicks “OK” button  [Alternative 2] | “Create post” button is hidden  “Leave” button is hidden  “Join” button is showed |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User goes to Dashboard screen and clicks “Joined classroom” tab | System displays list of joined classrooms | |  | User clicks “Leave” button on classroom that user wants to leave | System displays “Leave classroom” popup:  - “OK”: button  - “Cancel”: button | |  | User clicks “OK” button  [Alternative 2] | “Create post” button is hidden  “Leave” button is hidden  “Join” button is showed | | 2 | User clicks “Cancel” button | Popup disappears and student has not left classroom. |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User can only leave joined classroom (if user is invited to join or user sends request for joining, user cannot leave classroom) * After leaving, user can request for joining or classroom’s teacher can invite user to join again. * After leaving, news feed content changed earliest after 30 minutes | | | |

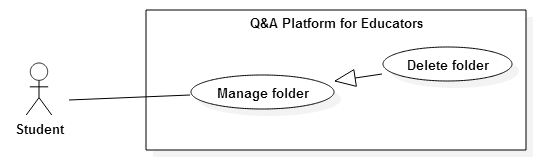
##### <Student> Create folder Use Case Diagram

****

**Figure 33: <Student> Create folder Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -28 SPECIFICATION** | | | |
| **Use-case No.** | QAPE028 | **Use-case Version** | 2.0 |
| **Use-case Name** | Create folder | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Student   **Summary:**   * This use case allows user create folder to manage material   **Goal:**   * Create new folder   **Triggers**   * User clicks “Create folder” button in Classroom screen.   **Preconditions:**   * User must login as Student role.   **Post Conditions:**   * **On Success**: New folder is created. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Material screen and clicks “Create folder” button | System displays “Create folder” popup:  - “Folder name\*”: textbox, min length: 8, max length: 30, required  - “Create new folder”: button  - “x” icon: button | | 2 | User fill “Folder name\*” |  | | 3 | User click “Create new folder”  [Alternative 1] | Folder is created, popup is hidden and system displays new folder in “Library” tab |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click “x” icon button | Popup is hidden and no folder is created |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User leaves “Folder name\*” blank | Show error message: “Please provide the folder name.” | | 2 | User enters “Folder Name\*” not match min and max length | Show error message: “The folder name must be between 8 and 30 characters long.” |   **Relationships:**   * N/A   **Business Rules:**   * User can create maximum 10 folders * Only students can create new folder * Newest folder is showed at first | | | |

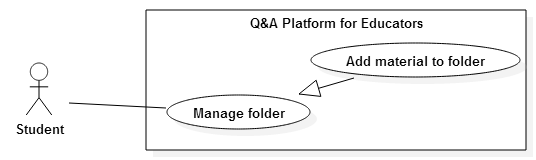
##### <Student> Delete folder Use Case Diagram

****

**Figure 34: <Student> Delete folder Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -29 SPECIFICATION** | | | |
| **Use-case No.** | QAPE029 | **Use-case Version** | 2.0 |
| **Use-case Name** | Delete folder | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Student   **Summary:**   * This use case allows user delete folder   **Goal:**   * Delete owned folder   **Triggers**   * Student clicks “x” icon in folder icon.   **Preconditions:**   * User must login as Student role. * User must have at least one folder   **Post Conditions:**   * **On Success**: Folder is deleted. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Material screen and clicks “x” icon on folder icon | System displays “Delete folder” popup:  - “OK”: button  - “Cancel”: button | | 2 | User clicks “OK” button  [Alternative 1] | Folder is deleted and removed from screen |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User clicks “Cancel” button | Popup disappears and no student is removed. |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * Delete folder include deleting all materials in it | | | |

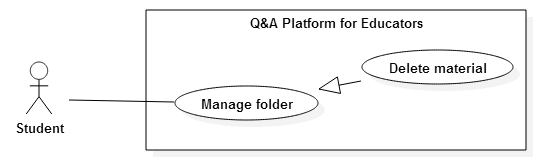
##### <Student> Add material to folder Use Case Diagram

****

**Figure 35: <Student> Add material to folder Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -30 SPECIFICATION** | | | |
| **Use-case No.** | QAPE030 | **Use-case Version** | 2.0 |
| **Use-case Name** | Add material to folder | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **Actor:**   * Student   **Summary:**   * This use case allows user add favorite material to folder   **Goal:**   * Add material to folder   **Triggers**   * User clicks “Folder” button on material that student want to add.   **Preconditions:**   * User must login as Student role. * User must have at least one folder.   **Post Conditions:**   * **On Success**: Material is added to folder. * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Classroom screen and clicks “Material” tab | System displays list of material of classroom | | 2 | User clicks “Folder” button on material that student want to added | System displays popup “Add to folder”:  - List of created folder  - “x” icon button | | 3 | User select folder  [Alternative 1] | Material is added to folder, and system displays message “Added” replace for “Folder” button |   **Alternative Scenario:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | User click “x” icon button | Popup is hidden and material is added to folder |   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * Student can create maximum 10 folders * Newest folder is showed at first | | | |

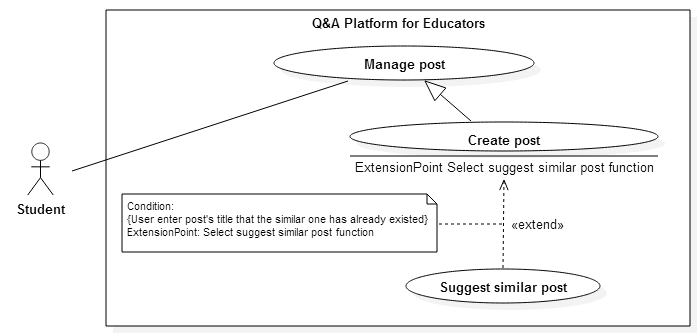
##### <Student> Delete material Use Case Diagram

****

**Figure 36: <Student> Delete material Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -31 SPECIFICATION** | | | |
| **Use-case No.** | QAPE031 | **Use-case Version** | 2.0 |
| **Use-case Name** | Delete material | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **REFERENCE TO USE CASE – 16 (QAPE016) WITH DIFFERENT POINTS:**  **Actor:**   * Student   **Triggers**   * User clicks “Delete” button in Folder screen.   **Preconditions:**   * User must login as Student role.   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | User goes to Folder screen. | System displays list of material | |  | **Other steps (begin from step 2) reference to QAPE016 USE CASE** | |   **Business Rules:**   * Deleted material is only deleted from folder; it’s still available in owned classroom. | | | |

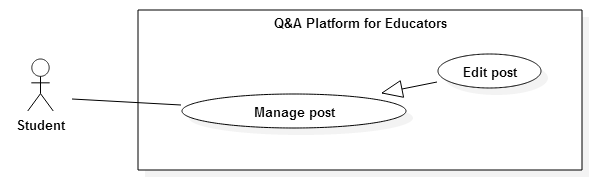
##### <Student> Create post Use Case Diagram

****

**Figure 37: <Student> Create post Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 32 SPECIFICATION** | | | |
| **Use-case No.** | QAPE032 | **Use-case Version** | 2.0 |
| **Use-case Name** | Create post | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **REFERENCE TO USE CASE – 10 (QAPE010) WITH DIFFERENT POINTS:**  **Actor:**   * Student   **Preconditions:**   * User must login as Student role. * User has joined classroom where post will be created * Classroom is “open” state | | | |

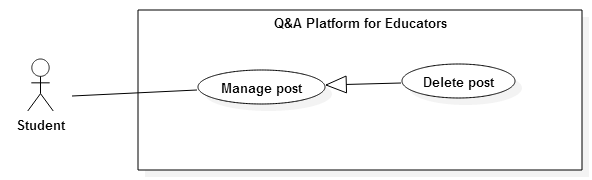
##### <Student> Edit post Use Case Diagram

****

**Figure 38: <Student> Edit post Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 33 SPECIFICATION** | | | |
| **Use-case No.** | QAPE033 | **Use-case Version** | 2.0 |
| **Use-case Name** | Edit post | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | Medium |
| **REFERENCE TO USE CASE – 11 (QAPE011) WITH DIFFERENT POINTS:**  **Actor:**   * Student | | | |

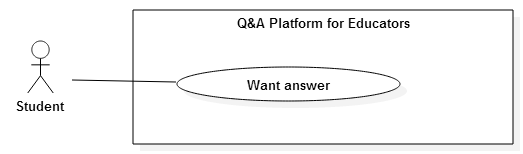
##### <Student> Delete post Use Case Diagram

****

**Figure 39: <Student> Delete post Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 34 SPECIFICATION** | | | |
| **Use-case No.** | QAPE034 | **Use-case Version** | 2.0 |
| **Use-case Name** | Delete post | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | Medium |
| **REFERENCE TO USE CASE – 14 (QAPE014) WITH DIFFERENT POINTS:**  **Actor:**   * Student | | | |

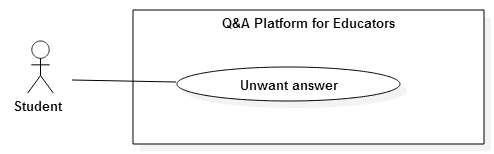
##### <Student> Want answer Use Case Diagram

****

**Figure 40: < Student> Want answer Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -35 SPECIFICATION** | | | |
| **Use-case No.** | QAPE035 | **Use-case Version** | 2.0 |
| **Use-case Name** | Want answer | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **REFERENCE TO USE CASE – 18 (QAPE018) WITH DIFFERENT POINTS:**  **Actor:**   * Student | | | |

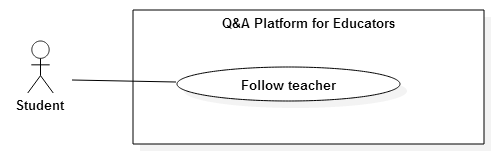
##### < Student> Un-want answer Use Case Diagram

****

**Figure 7: < Student> Un-want answer Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -36 SPECIFICATION** | | | |
| **Use-case No.** | QAPE036 | **Use-case Version** | 2.0 |
| **Use-case Name** | Un-want answer | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **REFERENCE TO USE CASE – 19 (QAPE019) WITH DIFFERENT POINTS:**  **Actor:**   * Student | | | |

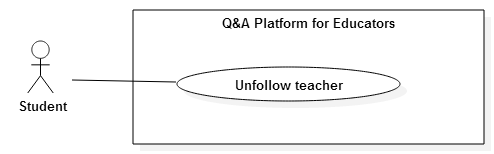
##### < Student> Follow teacher Use Case Diagram

****

**Figure 42: < Student> Follow teacher Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -37 SPECIFICATION** | | | |
| **Use-case No.** | QAPE037 | **Use-case Version** | 2.0 |
| **Use-case Name** | Follow teacher | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **REFERENCE TO USE CASE – 20 (QAPE020) WITH DIFFERENT POINTS:**  **Actor:**   * Student | | | |

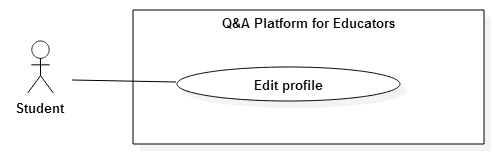
##### < Student> Un-follow teacher Use Case Diagram

****

**Figure 43: < Student> Un-follow teacher Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -38 SPECIFICATION** | | | |
| **Use-case No.** | QAPE038 | **Use-case Version** | 2.0 |
| **Use-case Name** | Un-follow teacher | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | Medium |
| **REFERENCE TO USE CASE – 21 (QAPE021) WITH DIFFERENT POINTS:**  **Actor:**   * Student | | | |

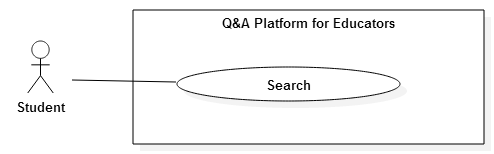
##### < Student> Edit profile Use Case Diagram

****

**Figure 44: <Student> Edit profile Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -39 SPECIFICATION** | | | |
| **Use-case No.** | QAPE039 | **Use-case Version** | 2.0 |
| **Use-case Name** | Edit profile | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **REFERENCE TO USE CASE – 22 (QAPE022) WITH DIFFERENT POINTS:**  **Actor:**   * Student | | | |

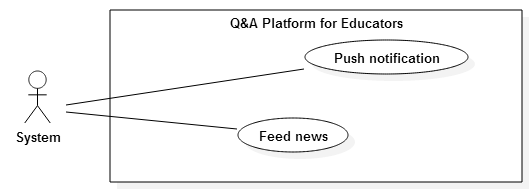
##### <Student> Search Use Case Diagram

****

**Figure 45: <Student> Search Use Case Diagram**

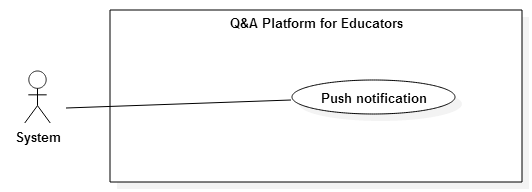
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -40 SPECIFICATION** | | | |
| **Use-case No.** | QAPE040 | **Use-case Version** | 2.0 |
| **Use-case Name** | Search | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **REFERENCE TO USE CASE – 24 (QAPE024) WITH DIFFERENT POINTS:**  **Actor:**   * Student | | | |

#### <System> Overview Use Case

****

**Figure 46: <System> Overview Use Case**

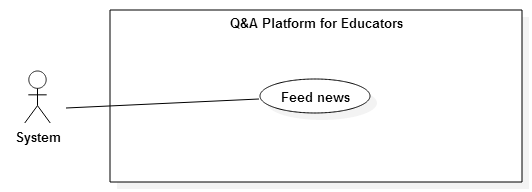
##### <System> Push Notification Use Case Diagram

****

**Figure 47: <System> Push Notification Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -41 SPECIFICATION** | | | |
| **Use-case No.** | QAPE041 | **Use-case Version** | 2.0 |
| **Use-case Name** | Push notification | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 18/07/2015 | **Priority** | High |
| **Actor:**   * System   **Summary:**   * This use case allows system send notification to related users   **Goal:**   * Notify followers when teacher creates classroom * Notify followers and student in classroom when teacher creates post in this classroom * Notify students in classroom and classroom’s owned teacher when student creates post in this classroom. * Notify question owner and all users discuss in question when having new reply * Notify classroom’s owned teacher when having new request * Notify student and teacher when having new invitation   **Triggers**   * One of above events (in **Goal**) occurs in real time   **Preconditions:**   * Teacher creates classroom * Teacher creates post in classroom * Student creates post in classroom * Teacher or student answers the question * Student requests to join classroom * Teacher invites student to join classroom * Teacher invites others to answering question   **Post Conditions:**   * **On Success**: Notify related user when event occurs * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | System find all users that relate to occurred event | System sends notification to related users. |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * After event occurs, notification will send to related user at the same time (real time notification) * Notification includes hyperlink that linked with occurred event. * User can only see top 30 recent notifications. | | | |

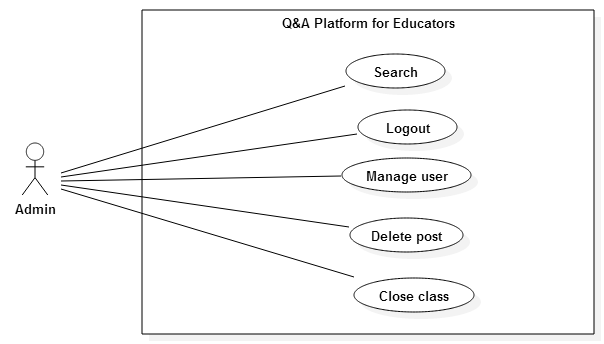
##### <System> Feed News Use Case Diagram

****

**Figure 48: <System> Feed News Use Case Diagram**

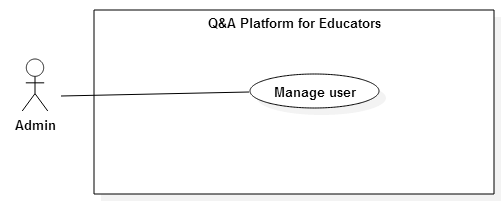
|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -42 SPECIFICATION** | | | |
| **Use-case No.** | QAPE042 | **Use-case Version** | 2.0 |
| **Use-case Name** | Feed news | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 18/07/2015 | **Priority** | High |
| **Actor:**   * System   **Summary:**   * This use case allows system calculates question’s, article’s, material’s to order and show on user’s newsfeed for each user.   **Goal:**   * Calculate question score base on view number, want answer number, created date, updated date * Check question is in owned/joined classroom, is owned followed teacher, is in user’s knowledge * Check article’s, material’s tags is in user tags * Check article, material is in user’s knowledge * Sort questions, articles, materials decreasingly by score and add them to supported database   **Triggers**   * At 30 minutes routine.   **Preconditions:**   * User must be log in as Student or Teacher role   **Post Conditions:**   * **On Success**: Sorted questions, articles, materials for each user are added to supported database * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Server checks time and compares with previous feed news function’s executed times. If distance between two times is 30 minutes, calculate score and sort | System calculates score of questions, articles, materials, and sorted each of three by score decreasingly. |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * In 30 minutes between two execution times, user’s newsfeed has nothing changed. * If user has not owned or joined any classroom, scores of questions that are in classrooms will be passed * If user has not follow any teacher, scores of questions that are owner by followed teachers will be passed * If user has not yet enter his tags in profile, scores of articles and material that their tags match user’s tags will be passed. | | | |

#### <Admin> Overview Use Case

****

**Figure 49: <Admin> Overview Use Case**

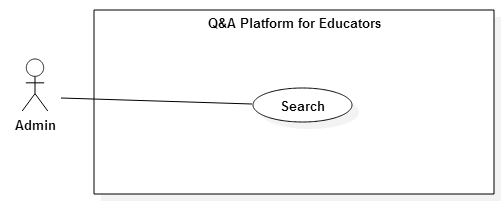
##### <Admin> Manage user Use Case Diagram

****

**Figure 13: <Admin> Manage user Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -43 SPECIFICATION** | | | |
| **Use-case No.** | QAPE043 | **Use-case Version** | 2.0 |
| **Use-case Name** | Manage user | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 18/07/2015 | **Priority** | High |
| **Actor:**   * Admin   **Summary:**   * This use case allows admin change users’ status in system   **Goal:**   * Activate/Deactivate user account   **Triggers**   * Admin clicks “Lock” icon / “Unlock” icon on user.   **Preconditions:**   * User must login as Admin role.   **Post Conditions:**   * **On Success**: User is activated or deactivated * **On Failure**: Show error message   **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Admin goes to Manage user screen. | System displays list of users in system.  If user account in row is deactivated: show “Unlock” icon  If user account in row is activated: show “Lock” icon | | 2 | Admin clicks “Lock”/”Unlock” icon on user that admin wants to change status | User’s status is updated  “Lock” icon changes to “Unlock” icon in case of deactivating account  “Unlock” icon change to “Lock” icon in case of activating account |   **Alternative Scenario:**   * N/A   **Exceptions:**   * N/A   **Relationships:**   * N/A   **Business Rules:**   * User account has two status Activated/Deactivated * After account is deactivated, user cannot login to system with this account * Admin cannot activate/deactivate himself | | | |

##### <Admin> Search Use Case Diagram

****

**Figure 51: <Admin> Search Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE -44 SPECIFICATION** | | | |
| **Use-case No.** | QAPE044 | **Use-case Version** | 2.0 |
| **Use-case Name** | Search | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | High |
| **REFERENCE TO USE CASE – 24 (QAPE024) WITH DIFFERENT POINTS:**  **Actor:**   * Admin | | | |

##### <Admin> Delete post Use Case Diagram

****

**Figure 52: <Admin> Delete post Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 45 SPECIFICATION** | | | |
| **Use-case No.** | QAPE045 | **Use-case Version** | 2.0 |
| **Use-case Name** | Delete post | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | Medium |
| **REFERENCE TO USE CASE – 14 (QAPE014) WITH DIFFERENT POINTS:**  **Actor:**   * Admin   **Preconditions:**   * User must login as Admin role.   **Business Rules:**   * User with admin role can delete any posts. | | | |

##### <Admin> Close class Use Case Diagram

****

**Figure 53: <Admin> Close class Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 46 SPECIFICATION** | | | |
| **Use-case No.** | QAPE046 | **Use-case Version** | 2.0 |
| **Use-case Name** | Close class | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | Medium |
| **REFERENCE TO USE CASE – 6 (QAPE06) WITH DIFFERENT POINTS:**  **Actor:**   * Admin   **Preconditions:**   * User must login as Admin role. * Classroom must be in “open” state.   **Business Rules:**   * User can close any classrooms * User cannot create new article/question in closed classroom * Student cannot request to join closed classroom * User can manage question, article, material, request, and student in closed classroom. | | | |

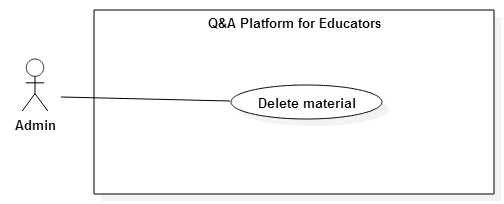
##### <Admin> Reopen class Use Case Diagram

****

**Figure 54: <Admin> Reopen class Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 47 SPECIFICATION** | | | |
| **Use-case No.** | QAPE047 | **Use-case Version** | 2.0 |
| **Use-case Name** | Reopen class | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | Medium |
| **REFERENCE TO USE CASE – 7 (QAPE07) WITH DIFFERENT POINTS:**  **Actor:**   * Admin   **Preconditions:**   * User must login as Admin role. * Classroom must be in “open” state.   **Business Rules:**   * User can reopen any classrooms. | | | |

##### <Admin> Delete material Use Case Diagram

****

**Figure 55: <Admin> Delete material Use Case Diagram**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE 48 SPECIFICATION** | | | |
| **Use-case No.** | QAPE048 | **Use-case Version** | 2.0 |
| **Use-case Name** | Delete material | | |
| **Author** | Kha Hoang Minh | | |
| **Date** | 17/07/2015 | **Priority** | Medium |
| **REFERENCE TO USE CASE – 16 (QAPE016) WITH DIFFERENT POINTS:**  **Actor:**   * Admin   **Preconditions:**   * User must login as Admin role.   **Business Rules:**   * User can delete materials in any classrooms. * Deleted material is only deleted from classrooms; it’s still available in student’s folder if it has already added to. | | | |

## Software System Attribute

### Usability

#### Graphic User Interface

* + All the texts, labels and should be written in English.

#### Usability

* + Admin only need one day of training to use this system.

#### Installation

* + The system must be easy to deploy. Customer can deploy successfully and learn to configure, maintain the system within one day of training.
  + The attached manual guide must be clear. User can read and do themselves without developer’s help.

### Reliability

* + .

### Availability

* + System is available 24/7.
  + User need an internet connection to access the system.

### Security

* + .

### Maintainability

* + .

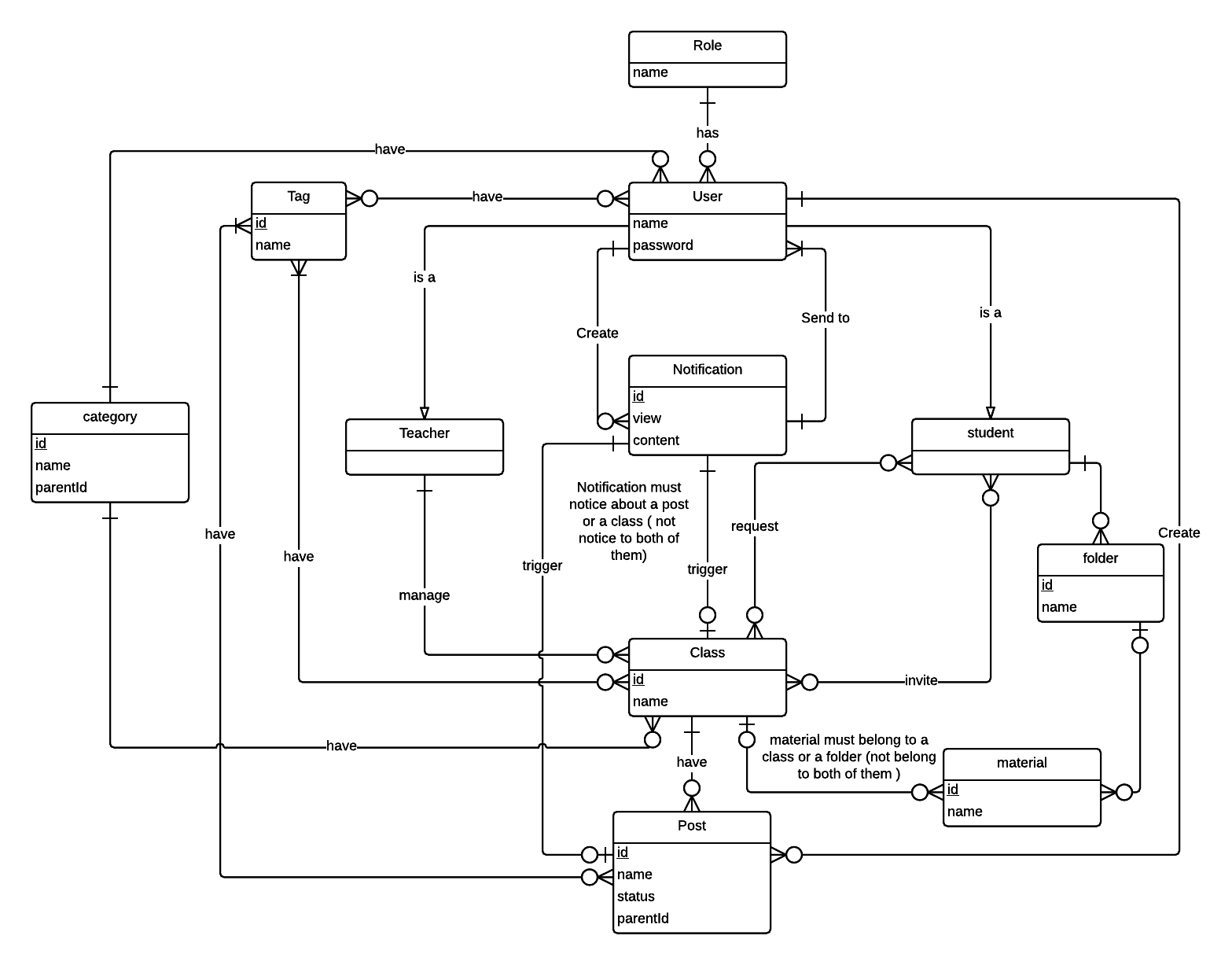
### Portability

* + .

### Performance

* + .

## Conceptual Diagram

****

**Figure 56: Conceptual diagram**

**Data Dictionary**

|  |  |
| --- | --- |
| **Entity Data dictionary: describe content of all entities** | |
| Entity Name | Description |
|  |  |

Table 8: Entity data dictionary

# Report No. 4 Software Design Description

## Design Overview

* This document describes the technical and user interface design of QAPE System. It includes the architectural design, the detailed design of common functions and business functions and the design of 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 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 class diagram, class explanation, and sequence diagram to details the application functions.
* Section 5: describe a fully attributed ERD.

## System architecture design

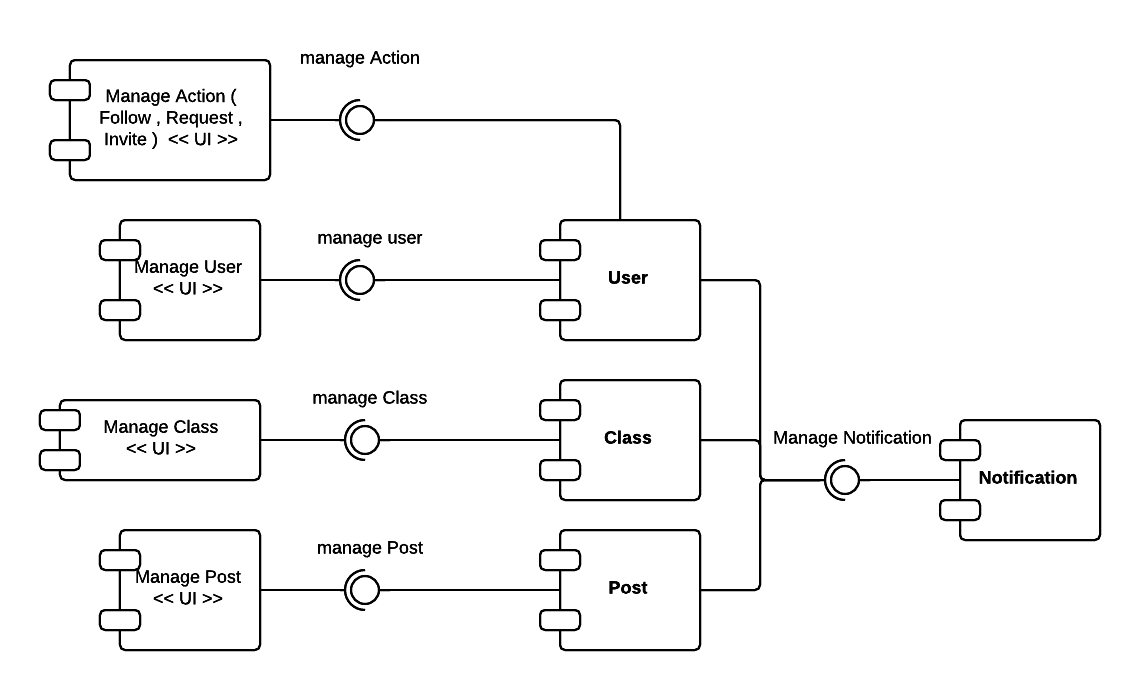
* Our system, QAPE is developed based on MVC pattern.
* QAPE is a big website and a kind of social website. MVC architecture separate a product (website, win form, phone app …) to three parts that help us easier to build – develop – test – control project.
* MVC architecture can run with any environment or language, so in the future if we want to another instance for our website on Mobile Phone or Desktop Computer, We can do it easily.

## Web application architecture description

|  |  |
| --- | --- |
| Model – View – Controller Architecture | |
| User (Web browser) | Business logic  Send Request  Process Request  Call  Call  View  Select  Result  Forward/Redirect |
| Controller | Forward/Redirect |
| Model |  |
| View |  |

* Model is the part of the application that handles the logic for the application data. Model often receives data from database, process business, and then sends result to Controller.
* View is the part of the application that handles the display of the data. View often receives data from Controller and shows the data to user.
* Controller is the part of the application that handles user interaction. Controller often receives request from user, sends request to suitable Model, receives result from Model, and sends result to suitable View.

## Component Diagram



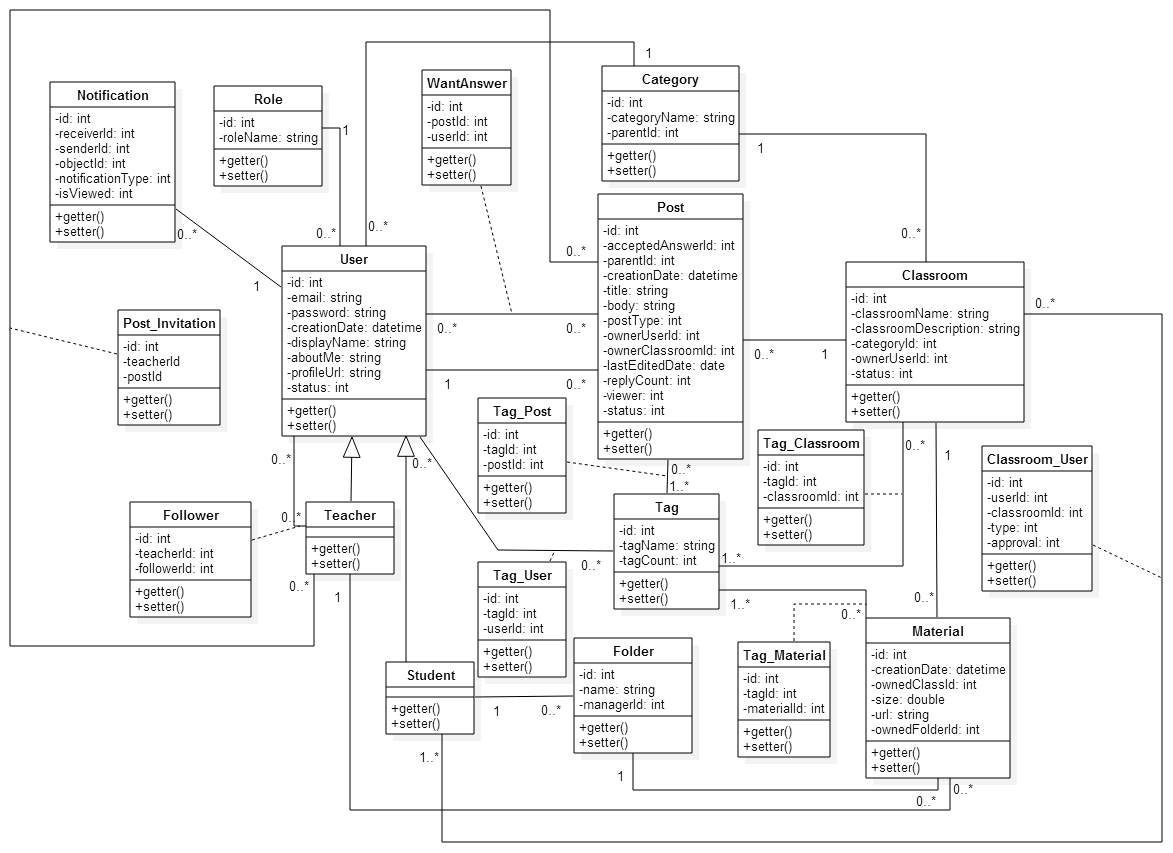
**Figure 57: Component diagram**

|  |  |
| --- | --- |
| **Component dictionary: describe component** | |
| Component Name | Description |
|  |  |

Table 9: Component dictionary

## Detailed Description

### Class Diagram

****

**Figure 58: Class diagram**

|  |  |
| --- | --- |
| **Class dictionary: describe class** | |
| Class Name | Description |
| Category | Describe attribute and operation of category |
| Classroom | Describe attribute and operation of classroom |
| Classroom\_User | Describe attribute and operation of relationship between classroom and student |
| Folder | Describe attribute and operation of folder |
| Follower | Describe attribute and operation of relationship between teacher and other users |
| Material | Describe attribute and operation of material |
| Notification | Describe attribute and operation of notification |
| Post | Describe attribute and operation of post |
| Post\_Invitation | Describe attribute and operation of relationship between post and teacher |
| Role | Describe attribute and operation of role |
| Student | Describe attribute and operation of student |
| Tag | Describe attribute and operation of tag |
| Tag\_Classroom | Describe attribute and operation of relationship between tag and classroom |
| Tag\_Material | Describe attribute and operation of relationship between tag and material |
| Tag\_Post | Describe attribute and operation of relationship between tag and post |
| Tag\_User | Describe attribute and operation of relationship between tag and user |
| Teacher | Describe attribute and operation of teacher |
| User | Describe attribute and operation of user |
| WantAnswerPost | Describe attribute and operation of relationship between post and user |

Table 10: Class dictionary

### Class Diagram Explanation

#### Category

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a category |
| categoryName | String | private | Name of category |
| parentId | Integer | private | Parent identifier of a category |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Classroom

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a classroom |
| classroomName | String | private | Name of classroom |
| classroomDescription | String | private | Description of classroom |
| categoryId | Integer | private | Identifier of a category that classroom belongs to |
| ownerUserId | Integer | private | Identifier of a user(teacher) that classroom belongs to |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Classroom\_User

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a relationship between classroom and user (student) |
| userId | Integer | private | Identifier of a user (student) |
| classroomId | Integer | private | Identifier of a classroom |
| type | Integer | private | Relationship type between student and classroom: 1: student request to join classroom  2: student is invited to join classroom |
| approval | Integer | private | Check if student has joined classroom or not:  0: pending  1: accept  2: ignore |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Folder

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a folder |
| name | String | private | Name of folder |
| managerId | String | private | Identifier of a user (student) owned the folder |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Follower

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a relationship between teacher and other users |
| teacherId | Integer | private | Identifier of a teacher that is followed |
| userId | Integer | private | Identifier of a user that is following |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Material

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a material |
| name | String | private | Name of material |
| creationDate | Date | private | Uploaded date of material |
| ownerClassId | Integer | private | Identifier of a classroom owned the material |
| folderId | Integer | private | Identifier of a folder the material belongs to |
| size | double | private | Size of material |
| fileURL | String | private | Download link of material |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Notification

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a notification |
| receiverId | Integer | private | Identifier of a user that received notification |
| senderId | Integer | private | Identifier of a user that makes notification occur |
| ownerClassId | Integer | private | Identifier of a classroom owned the material |
| objectId | Integer | private | Identifier of an object (classroom or post) reference to notification |
| notificationType | Integer | private | Notification type:  1: teacher creates new classroom  2: teacher creates new post  3: student creates post  4: teacher/student replies post  5: student requests to join classroom  6: teacher invites student to join classroom  7: teacher invites others to answer question  8: student accepts to join classroom |
| isViewed | Integer | private | Check if user has viewed notification or not:  0: not yet viewed  1: is viewed |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Post

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a post |
| isAccepted |  |  |  |
| parentId | Integer | private | Parent identifier of post (reply type) |
| creationDate | Date | private | Created date of post |
| title | String | private | Title of post (question/article) |
| body | String | private | Body of post |
| postType | Integer | private | Post type:  1: question  2: article  3: reply |
| ownerUserId | Integer | private | Identifier of user owned the post |
| ownerClassId | Integer | private | Identifier of classroom owned the post |
| lastEditedDate | Date | private | Last edited date of post |
| viewer | Integer | private | Number of viewers of post |
| status | Integer | private | Check if post is closed or opened |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Post\_Invitation

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a relationship between question and teacher |
| teacherId | Integer | private | Identifier of a teacher invited to answer question |
| postId | Integer | private | Identifier of a post teacher is invited to |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Role

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a relationship between question and teacher |
| roleName | String | private | Name of role |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Student

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Tag

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a tag |
| tagName | String | private | Name of tag |
| tagCount | Integer | private | Number of used times of tag |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Tag\_Classroom

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a relationship between tag and classroom |
| tagId | Integer | private | Identifier of tag |
| classroomId | Integer | private | Identifier of classroom |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Tag\_Material

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a relationship between tag and material |
| tagId | Integer | private | Identifier of tag |
| materialId | Integer | private | Identifier of material |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Tag\_Post

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a relationship between tag and post |
| tagId | Integer | private | Identifier of tag |
| postId | Integer | private | Identifier of post |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Tag\_User

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a relationship between tag and user |
| tagId | Integer | private | Identifier of tag |
| userId | Integer | private | Identifier of user |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### Teacher

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### User

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a user |
| email | String | private | Email of user |
| password | String | private | Password of user |
| creationDate | date | private | Created date of user |
| displayName | String | private | Display name of user |
| aboutMe | String | private | Introduction of user |
| profileImageURL | String | private | URL of image of user |
| status | Integer | private | Check if user is activated or deactivated |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |

#### WantAnswerPost

Attribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| id | Integer | private | Unique identifier of a relationship between post and user |
| postId | Integer | private | Identifier of post |
| userId | Integer | private | Identifier of user |

Method

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return type** | **Visibility** | **Description** |
| getter | attribute type | public | Get attribute value |
| setter | void | public | Set value of attribute |