**MINISTRY OF EDUCATION AND TRAINING**

**FPT UNIVERSITY**

Capstone Project Document (Summary)

**Q & A Platform for Educators**

|  |  |
| --- | --- |
| **Group 11** | |
| **Group members** | Kha Hoàng Minh – Team Leader – SE60979  Trương Như Khang – Team Member – SE60992 |
| **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*

**

**CAPSTONE PROJECT REGISTER**

Class: Duration time: from 11/5/2015…. To /2014…..

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

x

(\*) Kinds of person make registers: Lecturer Students

x

1. Register information for supervisor (if have)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Full name** | **Phone** | **E-Mail** | **Title** |
| Supervisor 1 | Nguyễn Huy Hùng |  | hungnh@fpt.edu.vn | Mr. |
| Supervisor 2 |  |  |  |  |

2. Register information for students (if have)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Full name** | **Student code** | **Phone** | **E-mail** | **Role in Group** |
| Student 1 |  |  |  |  |  |
| Student 2 |  |  |  |  |  |
| Student 3 |  |  |  |  |  |
| Student 4 |  |  |  |  |  |

3. Register content of Capstone Project

(\*) 3.1. Capstone Project name:

English: Q & A Platform for Educators

Vietnamese: Hệ Thống Hỏi Đáp Cho Sinh Viên, Giảng Viên

Abbreviation: QAPE

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

1. Theory and practice (document):
   * Student should apply the software development process and the UML 2.0 in modeling the system
   * Software artifacts include User Requirement, Software Requirement Specification, Architecture Design, Detail Design, System Implementation and Testing Document, Installation Guide, sources code, and deployable software packages
   * Server side technique:
     + Database design, OOA, OOD, OOP, MVC, Java or .Net technology, …
   * Client side technique
     + HTML5, CSS, JavaScript, JQuery, Ajax
2. Program:

An online Q&A platform where students and lectures can exchange information and other educational materials. The system should support the following main features:

**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

1. Other products:

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

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

N/A

|  |  |
| --- | --- |
| **Supervisor (If have)**  *(Sign and full name)* | HCM city, date 14/4/2015 …..  **On behalf of Registers**  *(Sign and full name)* |

# Table of Contents

[Table of Contents 6](#_Toc427417635)

[List of Tables 8](#_Toc427417636)

[List of Figures 9](#_Toc427417637)

[Definitions, Acronyms, and Abbreviations 10](#_Toc427417638)

[A. Software Project Management Plan 11](#_Toc427417639)

[1. Problem Definition 11](#_Toc427417640)

[1.1 Name of this Capstone Project 11](#_Toc427417642)

[1.2 Problem Abstract 11](#_Toc427417643)

[1.3 Project Overview 11](#_Toc427417644)

[2. Project organization 13](#_Toc427417645)

[2.1 Software Process Model 13](#_Toc427417646)

[2.2 Tools and Techniques 13](#_Toc427417647)

[B. Software Requirement Specification 14](#_Toc427417648)

[1. User Requirement Specification 14](#_Toc427417649)

[1.1 Guest Requirement 14](#_Toc427417650)

[1.2 Student Requirement 14](#_Toc427417651)

[1.3 Teacher Requirement 14](#_Toc427417652)

[1.4 Admin Requirement 15](#_Toc427417653)

[2. System Overview Use Case 16](#_Toc427417654)

[3. Conceptual Diagram 17](#_Toc427417655)

[C. Software Design Description 18](#_Toc427417656)

[1. System architecture design 18](#_Toc427417657)

[1.1 Web architecture design 18](#_Toc427417658)

[2. Component Diagram 19](#_Toc427417659)

[3. Detailed Description 19](#_Toc427417660)

[3.1 Class Diagram 19](#_Toc427417661)

[3.2 Sequence Diagram 21](#_Toc427417662)

[4. Logical Diagram 29](#_Toc427417663)

[5. Algorithms 31](#_Toc427417664)

[5.1 Calculating question’s score 31](#_Toc427417665)

[5.2 Calculate similar score of two String 32](#_Toc427417666)

[5.3 Arrange strings by similar score 32](#_Toc427417667)

[D. Physical Diagram 34](#_Toc427417668)

[E. Demonstrations 36](#_Toc427417669)

# List of Tables

[Table 1: Definitions 10](#_Toc427417670)

[Table 2: Web architecture design 18](#_Toc427417671)

# List of Figures

[Figure 1: Scrum Development Model 13](#_Toc427417695)

[Figure 2: System Overview Use Case 16](#_Toc427417696)

[Figure 3: Conceptual diagram 17](#_Toc427417697)

[Figure 4: Component diagram 19](#_Toc427417698)

[Figure 5: Class diagram 20](#_Toc427417699)

[Figure 6: <Teacher> Create classroom sequence 21](#_Toc427417700)

[Figure 7: <Teacher> Invite student sequence 22](#_Toc427417701)

[Figure 8: <Teacher> Merge questions sequence 23](#_Toc427417702)

[Figure 9: <Teacher> Answer multi questions sequence 24](#_Toc427417703)

[Figure 10: <Teacher> Accept request sequence 25](#_Toc427417704)

[Figure 11: <Student> Request to join 26](#_Toc427417705)

[Figure 12: <Student> Add to folder 27](#_Toc427417706)

[Figure 13: <Teacher, Student> Create post 28](#_Toc427417707)

[Figure 14: <Teacher, Student> Answers 29](#_Toc427417708)

[Figure 15: Logical diagram 30](#_Toc427417709)

[Figure 16: Physical database diagram 35](#_Toc427417710)

[Figure 17: Demo Create classroom and invite student 36](#_Toc427417711)

[Figure 18: Demo Join classroom and request to join 36](#_Toc427417712)

[Figure 19: Demo Create question 37](#_Toc427417713)

[Figure 20: Demo answer, accept an answer and require answers 37](#_Toc427417714)

[Figure 21: Demo Invite teacher 38](#_Toc427417715)

[Figure 22: Demo Upload material and add material to folder 38](#_Toc427417716)

[Figure 23: Demo Merge question and answer multi questions 39](#_Toc427417717)

# Definitions, Acronyms, and Abbreviations

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

Table 1: Definitions

# 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. Teaching hour is pretty little so that teachers cannot answer all questions of students during class. Besides that, some students feel shy to ask about their problems that are not understood.

Therefore, 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 a Website application. Teachers can create their own classrooms for teaching on the website. These classrooms allow their students join and post question or article. Teachers answer the question in owned classroom or invite other teachers to join and answer that question. They can also upload materials and documents for sharing. Students ask question, answer existed questions if they know, share their knowledge by posting articles or download materials to PC or save to online folder.

### Project Overview

#### Current Situation

This project is developed in order to help teachers and students exchange information conveniently. This is the way that teachers share and answer as most students as possible. This is also the way that students get answer quickly and easily or get more ideas from different teachers and other students. This project also has restrictions. It comes from spamming, missing teacher’s rate, no private classroom. The system not yet supports reporting bad question, article or users.

#### The Proposed System

This project is going to build a website that help teachers and students communicate with each other easier and get the answer from who what they want.

#### Website

**Teachers:**

Create classrooms

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

Merge similar questions

Invite other teachers to answer questions

Invite students to join the discussion in a classroom

Follow a teacher

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

Real time notification

Real time in discussion

**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

**Admin:**

Manage users

Manage classroom

Manage post

#### Mobile application

N/A

## 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.

### 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

# 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
* 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)
* Merge questions, answer multi questions
* 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
* 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)

## System Overview Use Case

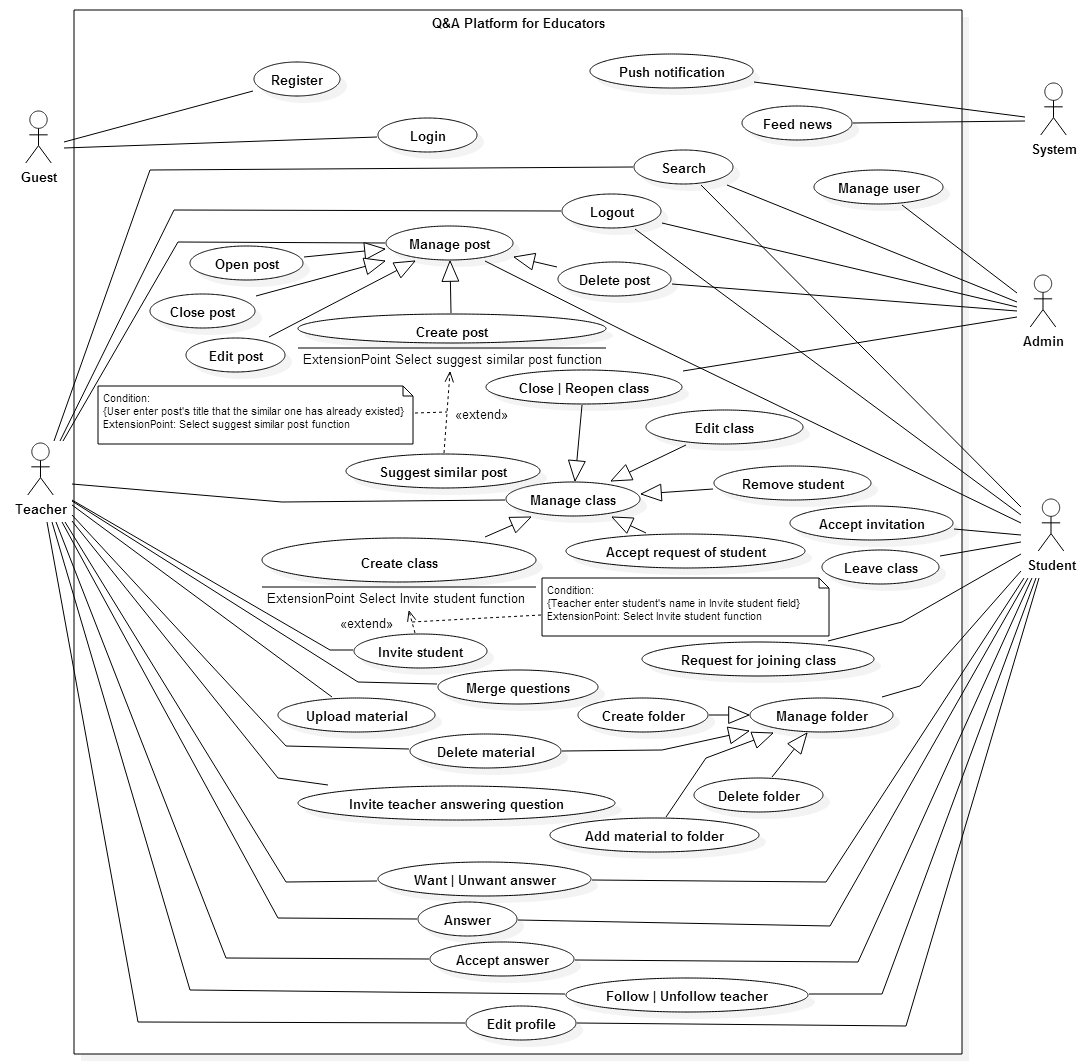
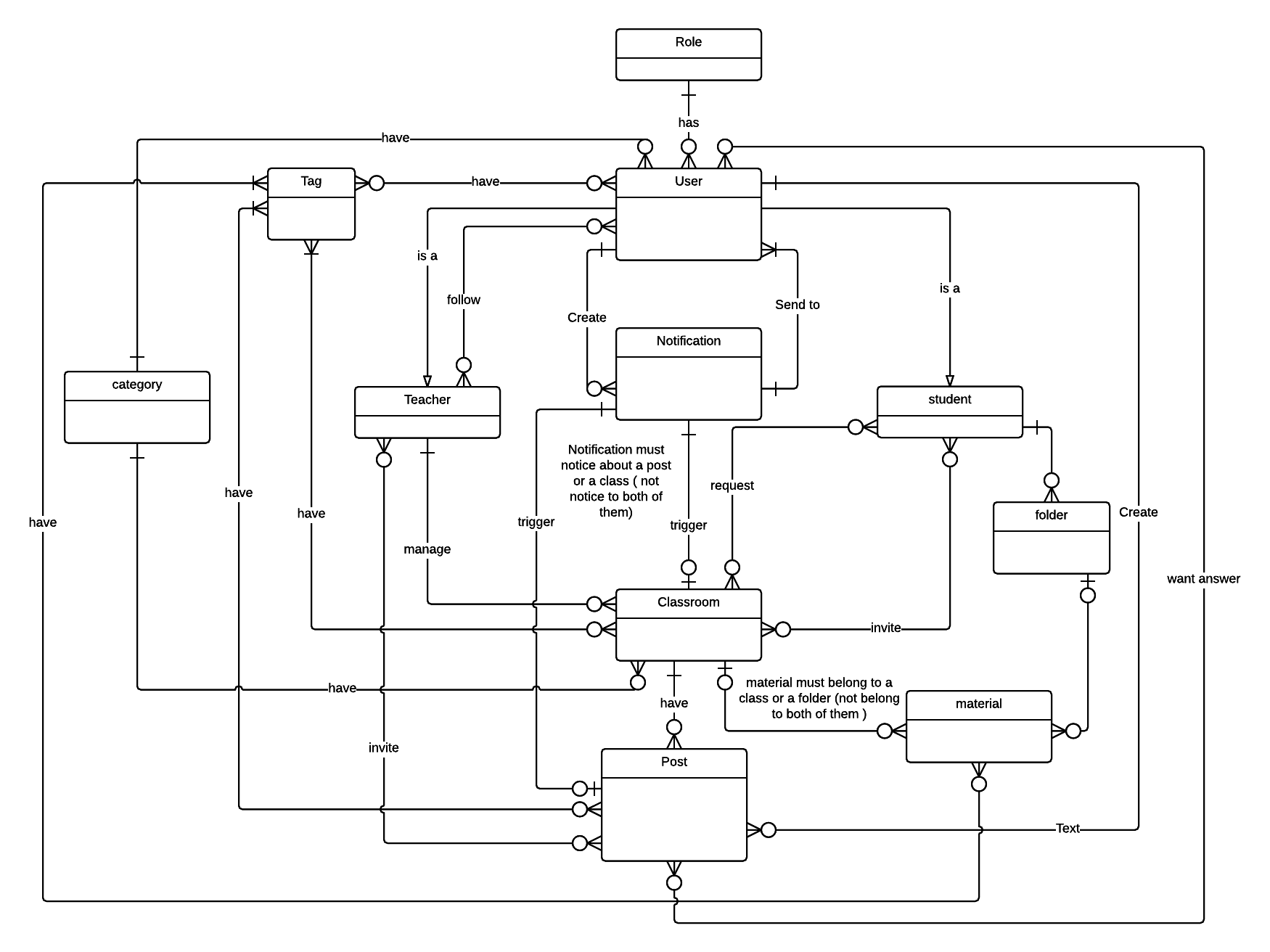


Figure 2: System Overview Use Case

## Conceptual Diagram

****

**Figure 3: Conceptual diagram**

# Software Design Description

## 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.

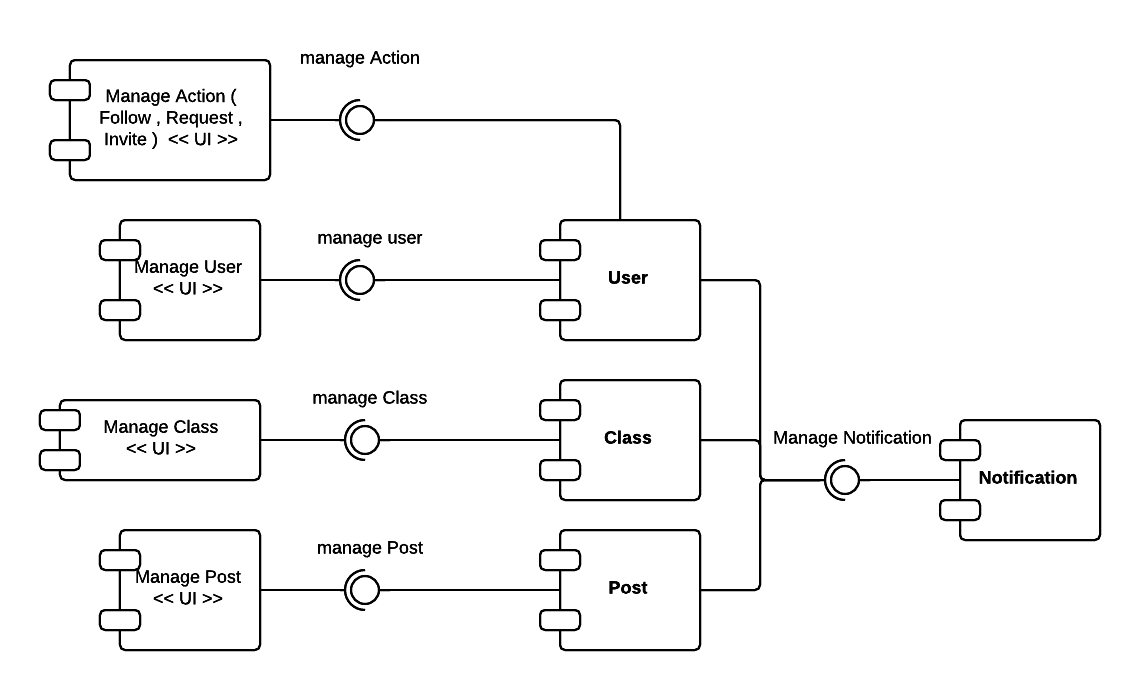
### Web architecture design

|  |  |
| --- | --- |
| 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 |  |

Table 2: Web architecture design

* 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.

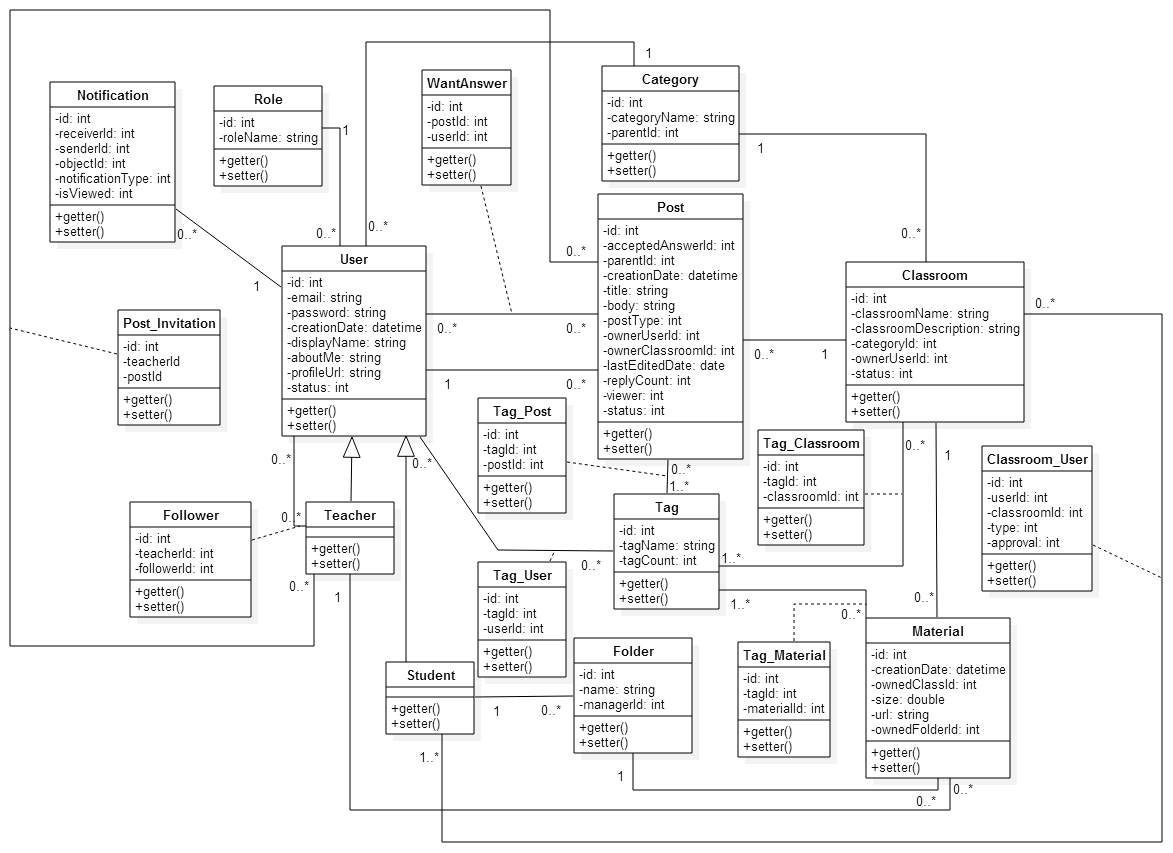
## Component Diagram



**Figure 4: Component diagram**

## Detailed Description

### Class Diagram

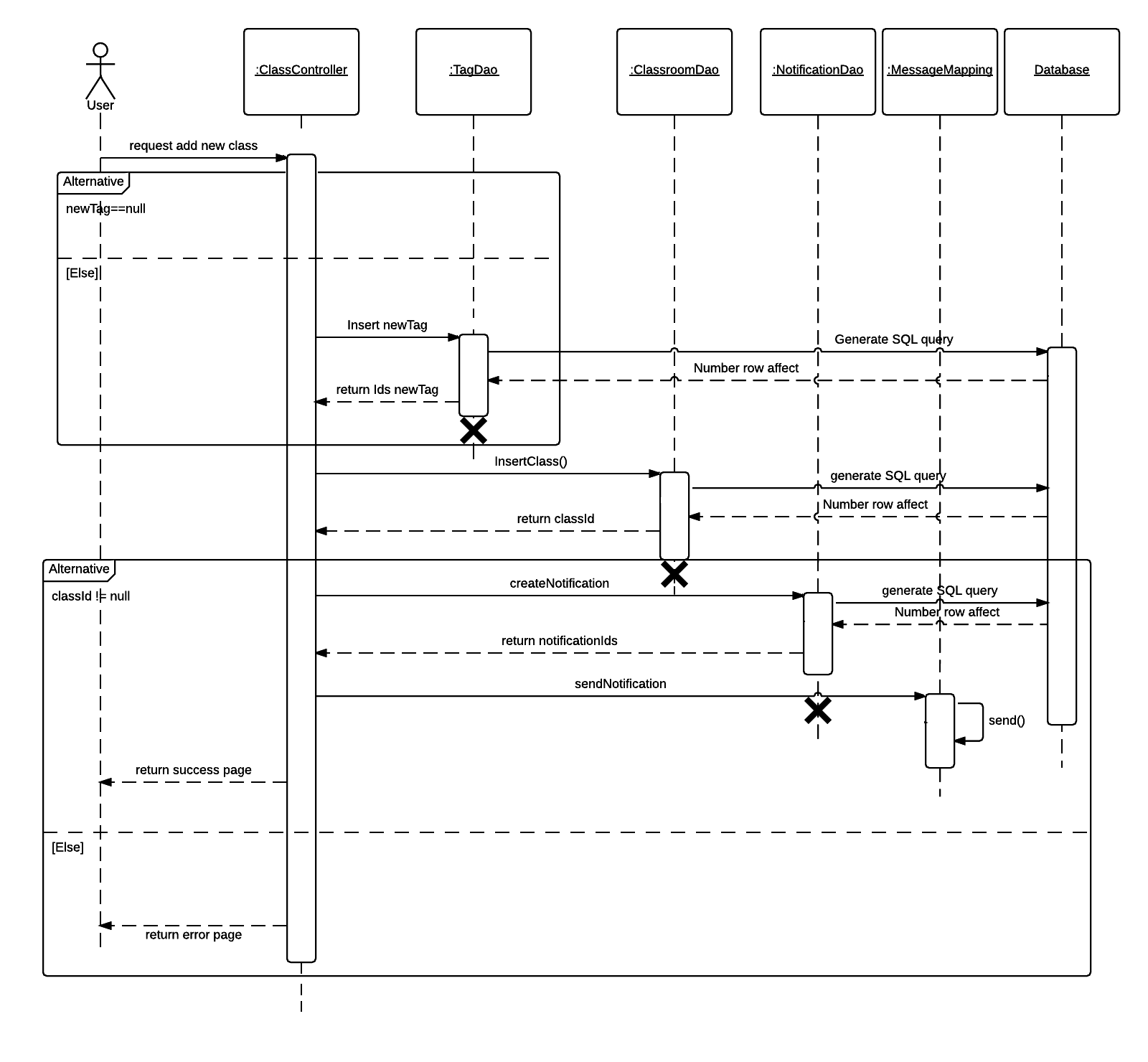
****

**Figure 5: Class diagram**

### Sequence Diagram

#### <Teacher> Create classroom

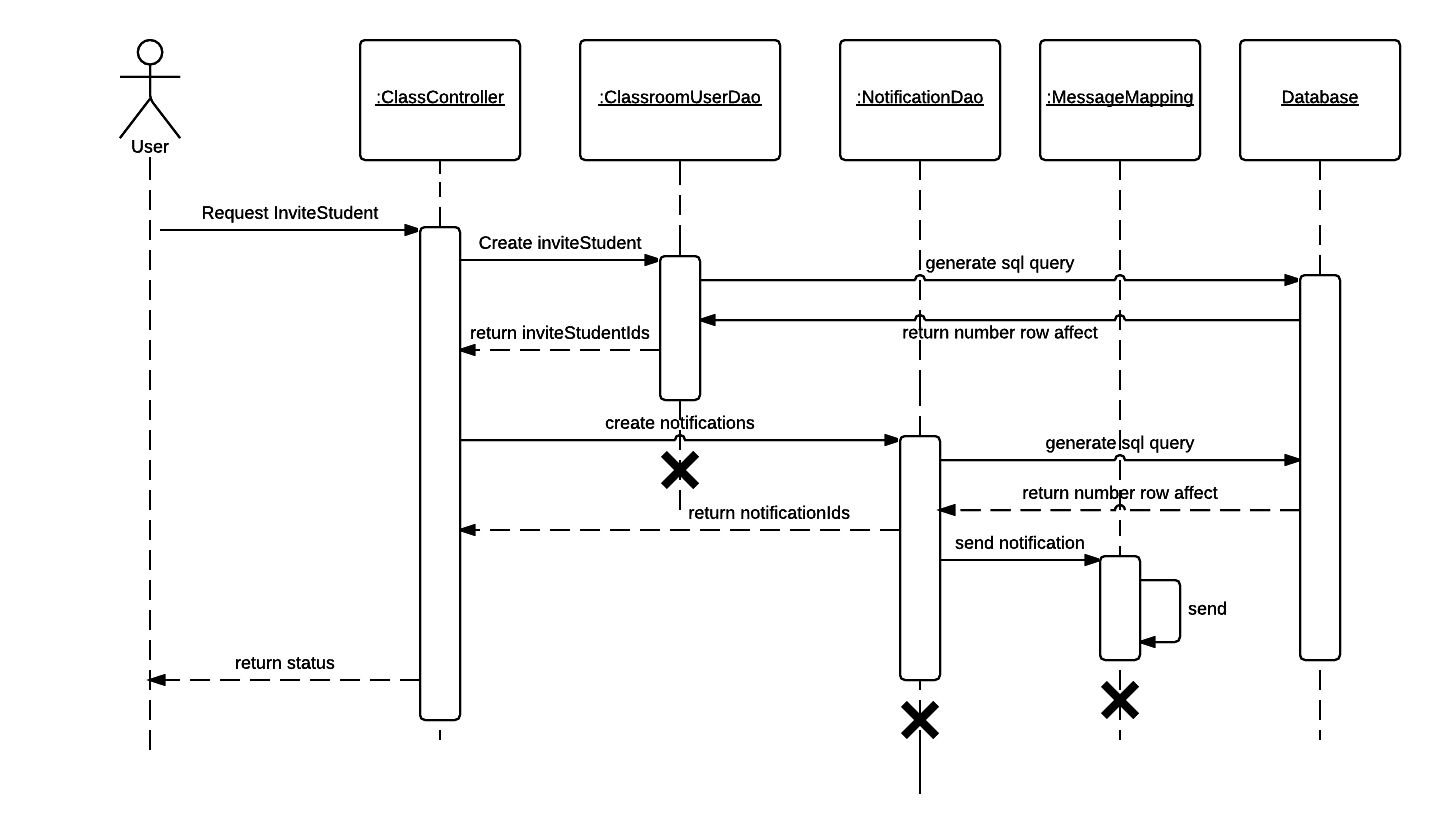
**Summary**: This diagram shows how teacher create a classroom



**Figure 6: <Teacher> Create classroom sequence**

#### <Teacher> Invite student

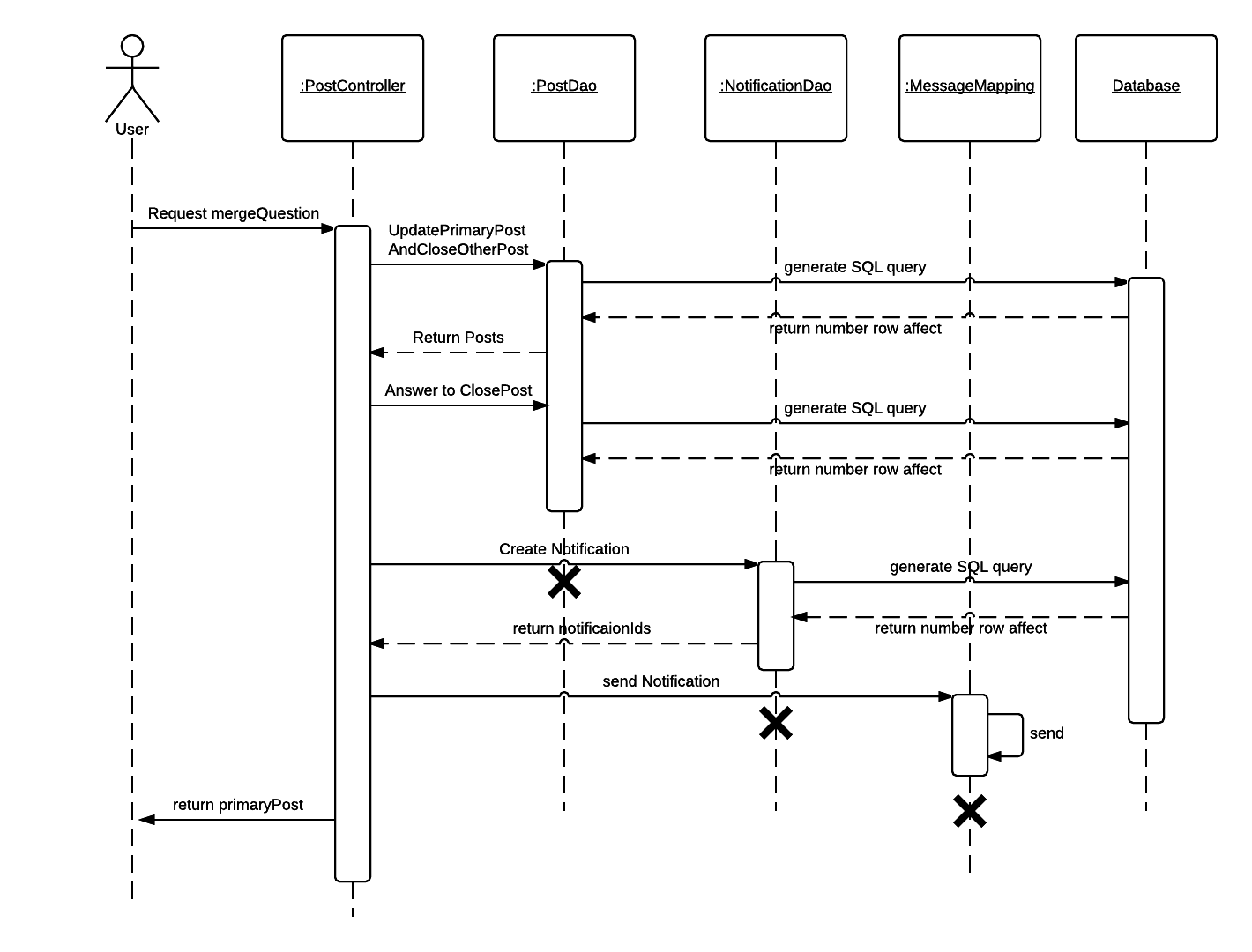
**Summary**: This diagram shows how teacher invites student to join classroom



**Figure 7: <Teacher> Invite student sequence**

#### <Teacher> Merge questions

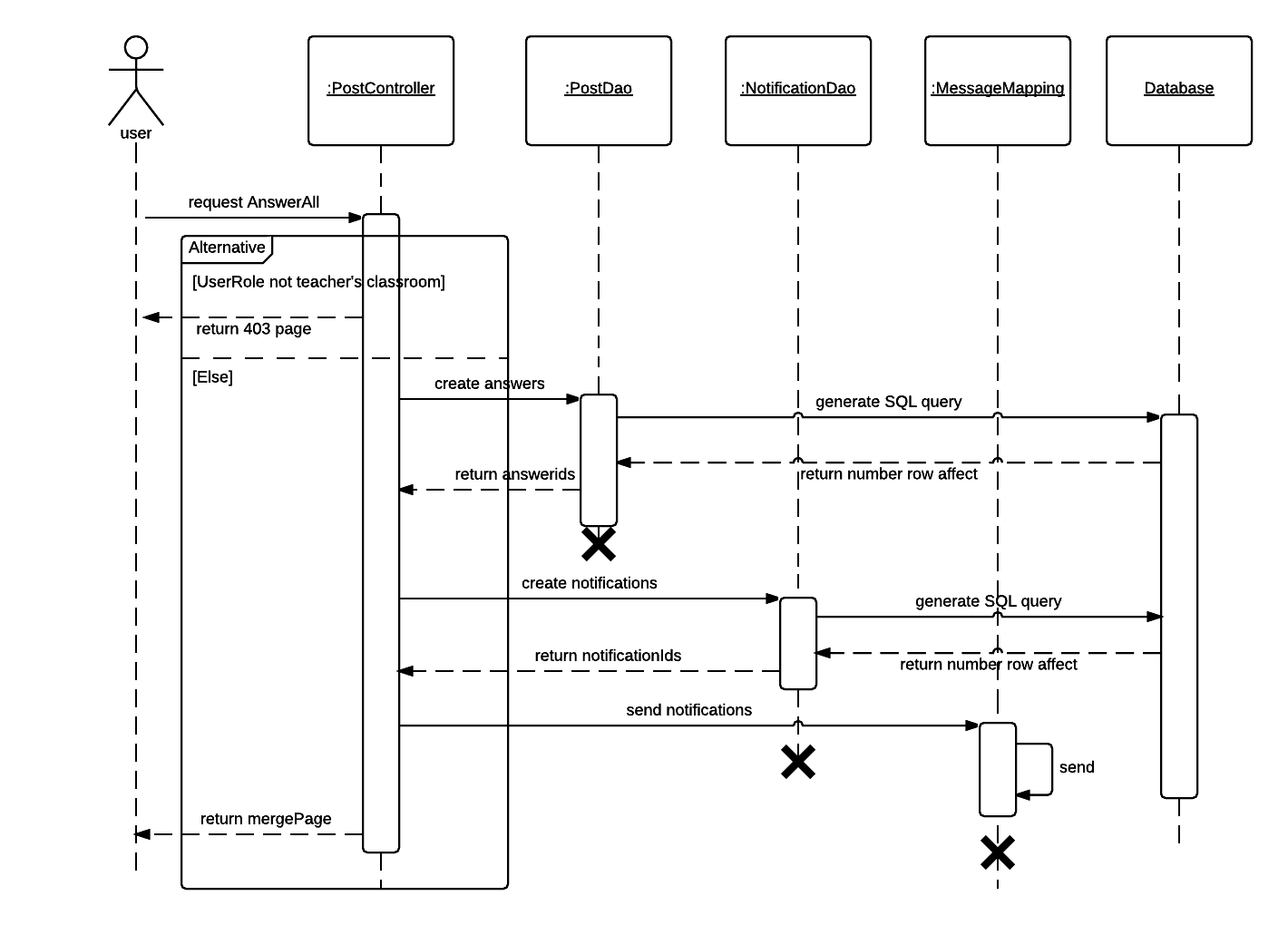
**Summary**: This diagram shows how teacher merges similar questions



**Figure 8: <Teacher> Merge questions sequence**

#### <Teacher> Answer multi questions

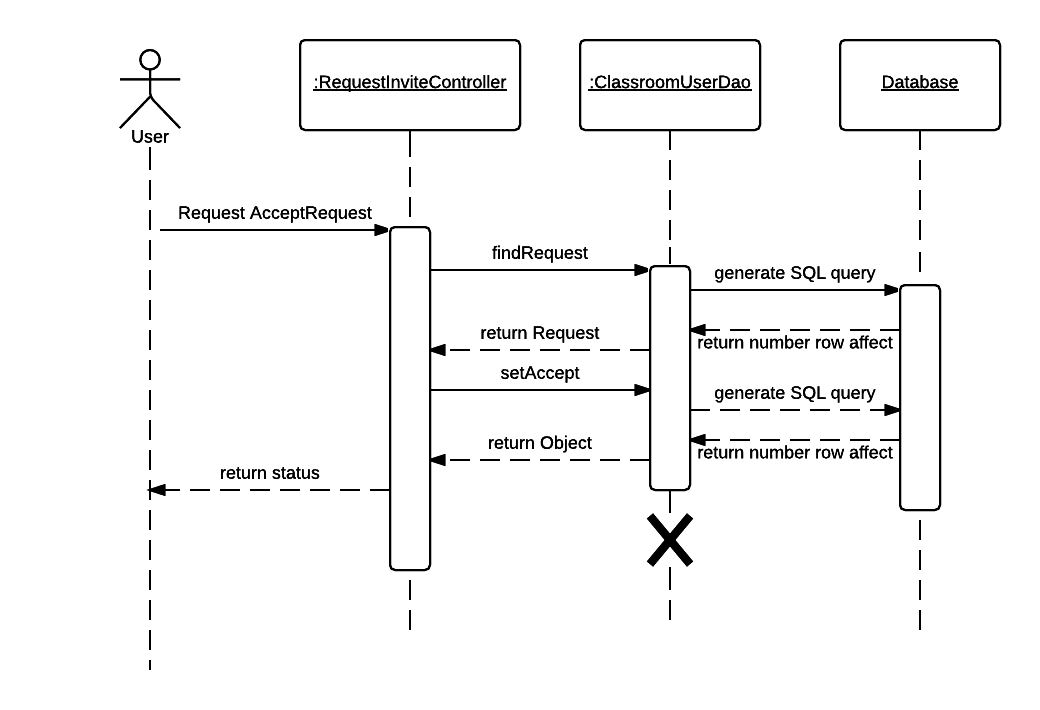
**Summary**: This diagram shows how teacher answers multi similar questions when merging



**Figure 9: <Teacher> Answer multi questions sequence**

#### <Teacher> Accept request

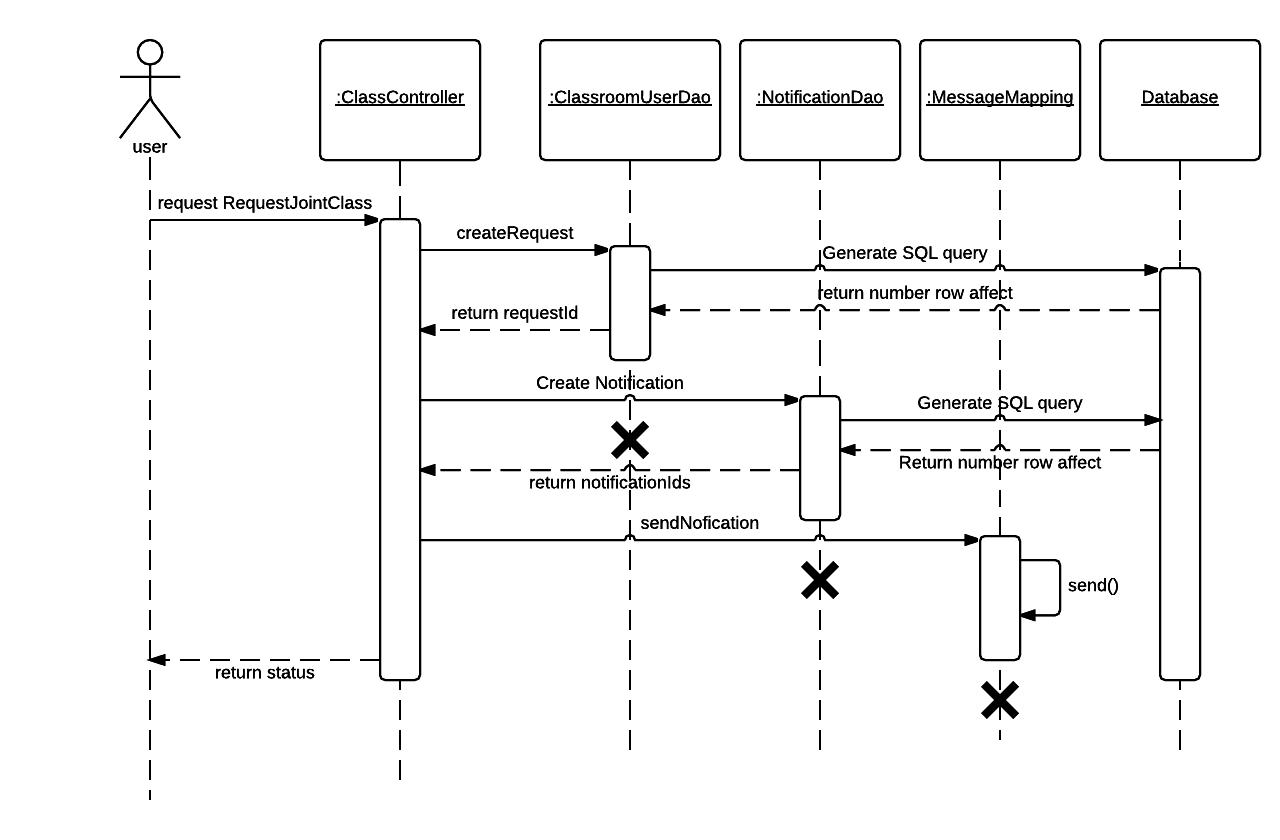
**Summary**: This diagram shows how teacher accepts student’s request



**Figure 10: <Teacher> Accept request sequence**

#### <Student> Request to join

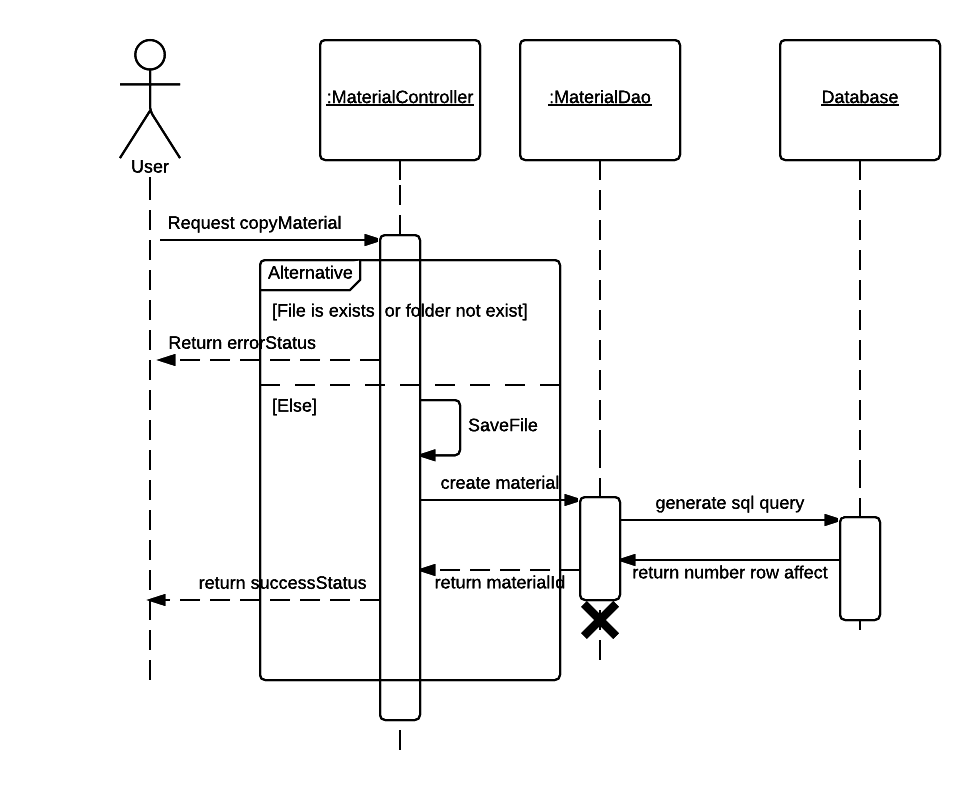
**Summary**: This diagram shows how student requests to join classroom



**Figure 11: <Student> Request to join**

#### <Student> Add to folder

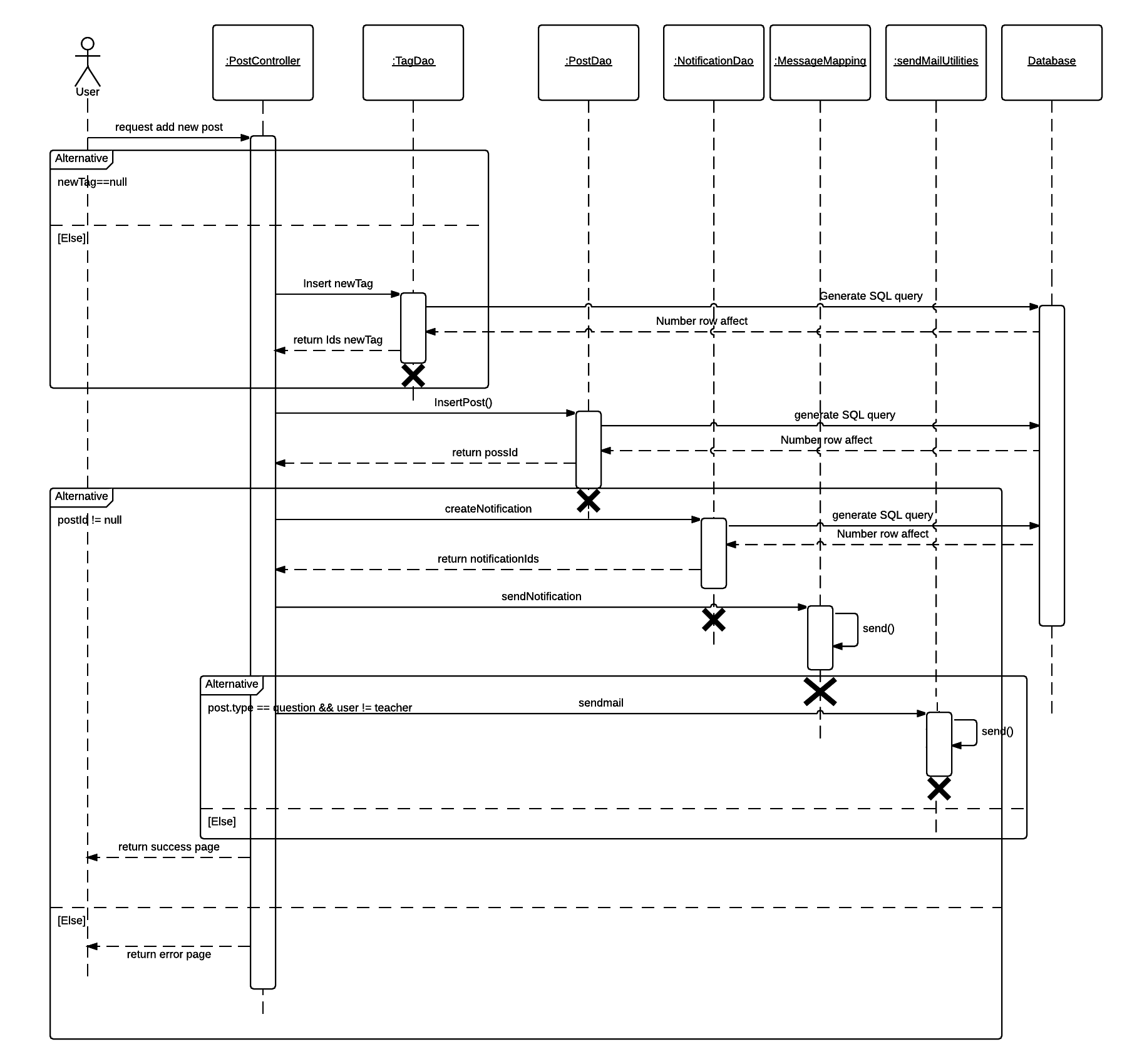
**Summary**: This diagram shows how student adds material to folder



**Figure 12: <Student> Add to folder**

#### <Teacher, Student> Create post

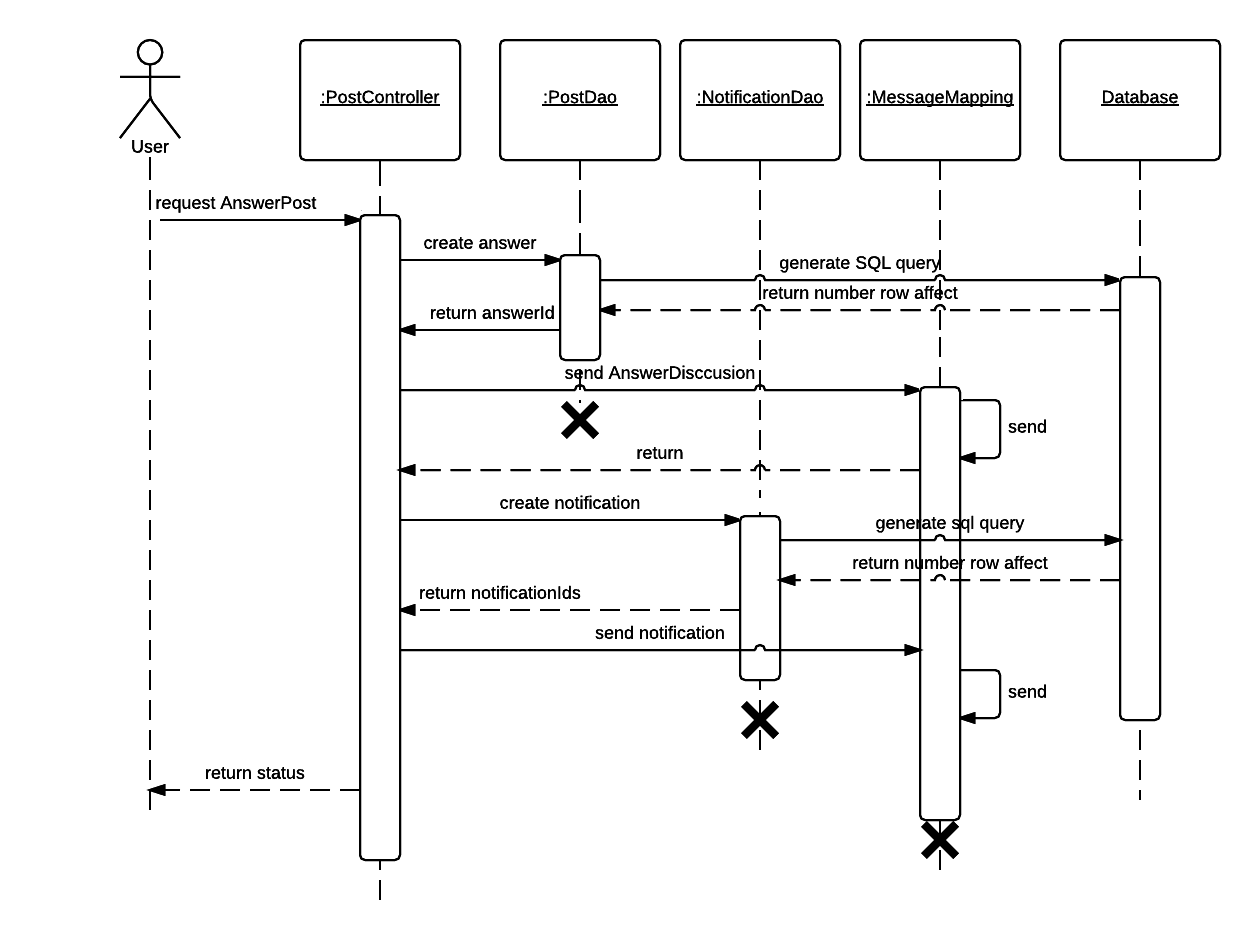
**Summary**: This diagram shows how teacher or student creates new post



**Figure 13: <Teacher, Student> Create post**

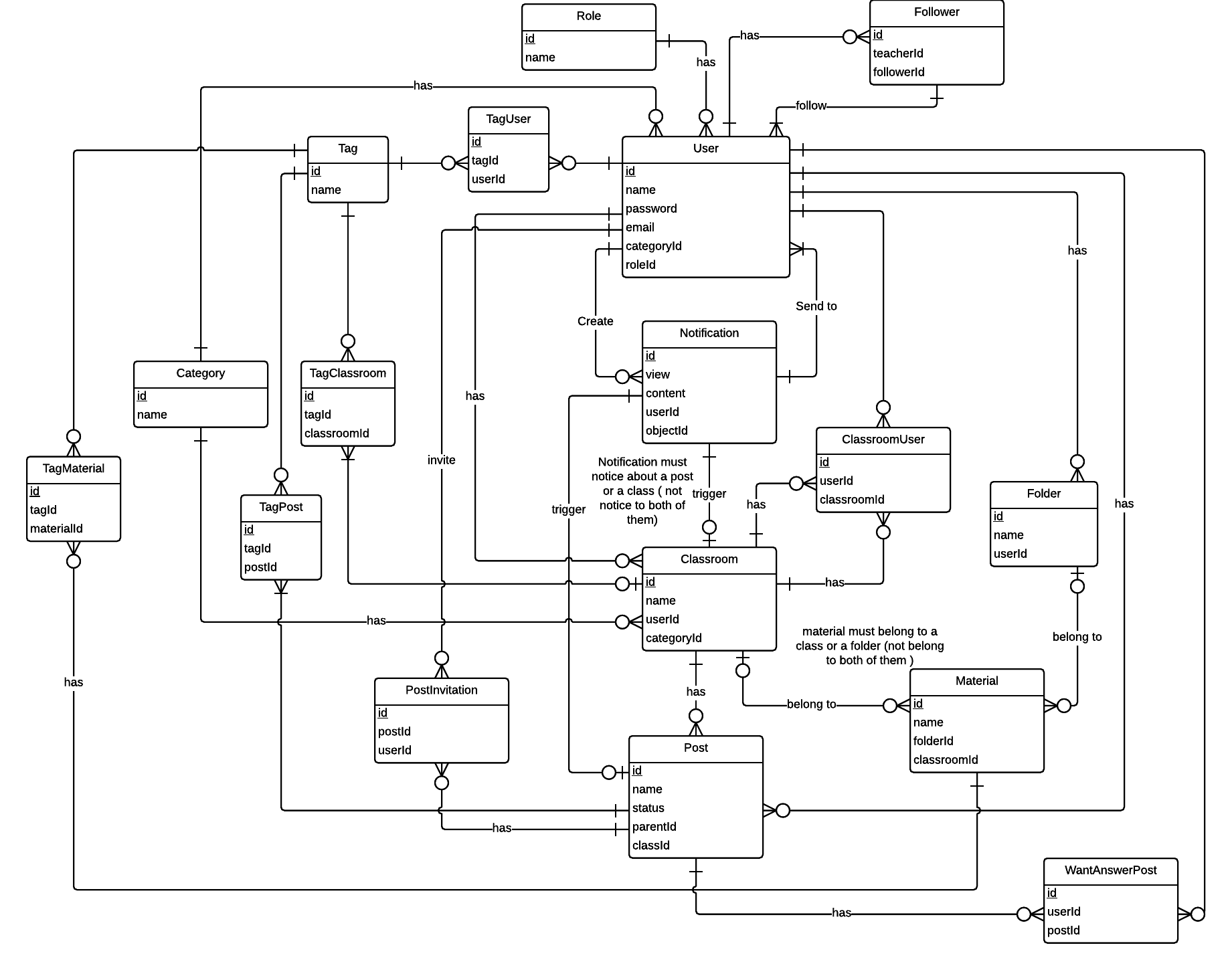
#### <Teacher, Student> Answer

**Summary**: This diagram shows how teacher or student answers a question



**Figure 14: <Teacher, Student> Answers**

## Logical Diagram



**Figure 15: Logical diagram**

## Algorithms

### Calculating question’s score

#### Definition

Calculating question’s score is the way to evaluate question’s importance. Base on question scores, system knows the order to show questions to user.

#### Definition Problem

It’s important to choose what question will be shown first and what question will be shown after to user. An algorithm that evaluates the user’s interest with questions (question’s scores) is needed.

#### Solution

Question’s score (S) is calculates base on the following attributes:

* *wa*: Number of users that want the answer for the question
* *v*: Number of users that view the question
* *a*: Number of answers for the question
* *a’*: Decrease value: a’=0 if the question hasn’t yet have accepted answer, a’=1 if the question has already have accepted answer
* *h*: distance between current time and last interaction (last edited time of the question or last edited time of answer of the question) time of question by hour

Because of the difference of attributes’ importance, each attribute have different weight: *wa* is 3, *v* is 2 and others is 1.

Define the formula to calculate question’s score:

Example: Given the question that has 30 users that want the answer, 100 views, 5 answers and lasted answer is posted at 5 hours ago. This question already has accepted answer.

Because the question has accepted answer, a’ =1. So question’s score:

#### Complexity

The complexity of algorithms is O(1)

### Calculate similar score of two String

#### Define Problem

* Given two strings. Calculate their matching percent

#### Requirement

* two strings

#### Solution

* + The solution follows “Jaccard index”
  + If a string contains many words, break it into a list of words.
  + For each word, we find out how many adjacent character pairs are contained in it.
  + Create a function pairs(s) which returns a list of adjacent character pairs of string s.
  + Then, we use below formula to calculate matching percent.

#### Example

Calculate the matching percent of 2 strings: *KHANG, KHUONG.*

Break string into list of adjacent character pairs:

KHANG 🡪 = {KH, HA, AN, NG}

KHUONG 🡪 = {KH, HU, UO, ON, NG}

Find number interaction of 2 Strings

Find number union of 2 Strings

Calculate matching percentage

### Arrange strings by similar score

#### Define Problem

* Given list of strings. Arrange them by similar score.

#### Requirement

* List of strings.

#### Solution

* Calculate total similar score of each element in list with other elements.
* Order list by element’s total score.

#### Example

* + - Given List of strings :
    - List A = {What is abstraction in OOP,

How to convert String to Int,

What is inheritance in OOP,

How to parse String to Int}

A1 = What is abstraction in OOP

A2 = How to convert String to Int

A3 = What is inheritance in OOP

A4 = How to parse String to Int

* + - Expected Result :

List B = {How to parse String to Int ,

How to convert String to Int ,

What is abstraction in OOP ,

What is inheritance in OOP }

* + - Calculate total similar score of First Element with other elements :
      * Total(A1) =
    - We do the same thing with other elements. We will get these results :
      * Total(A2) = 0.65
      * Total(A3)= 0.43
      * Total(A4) = 0.689
    - Order list decreasingly by total similar score we get:

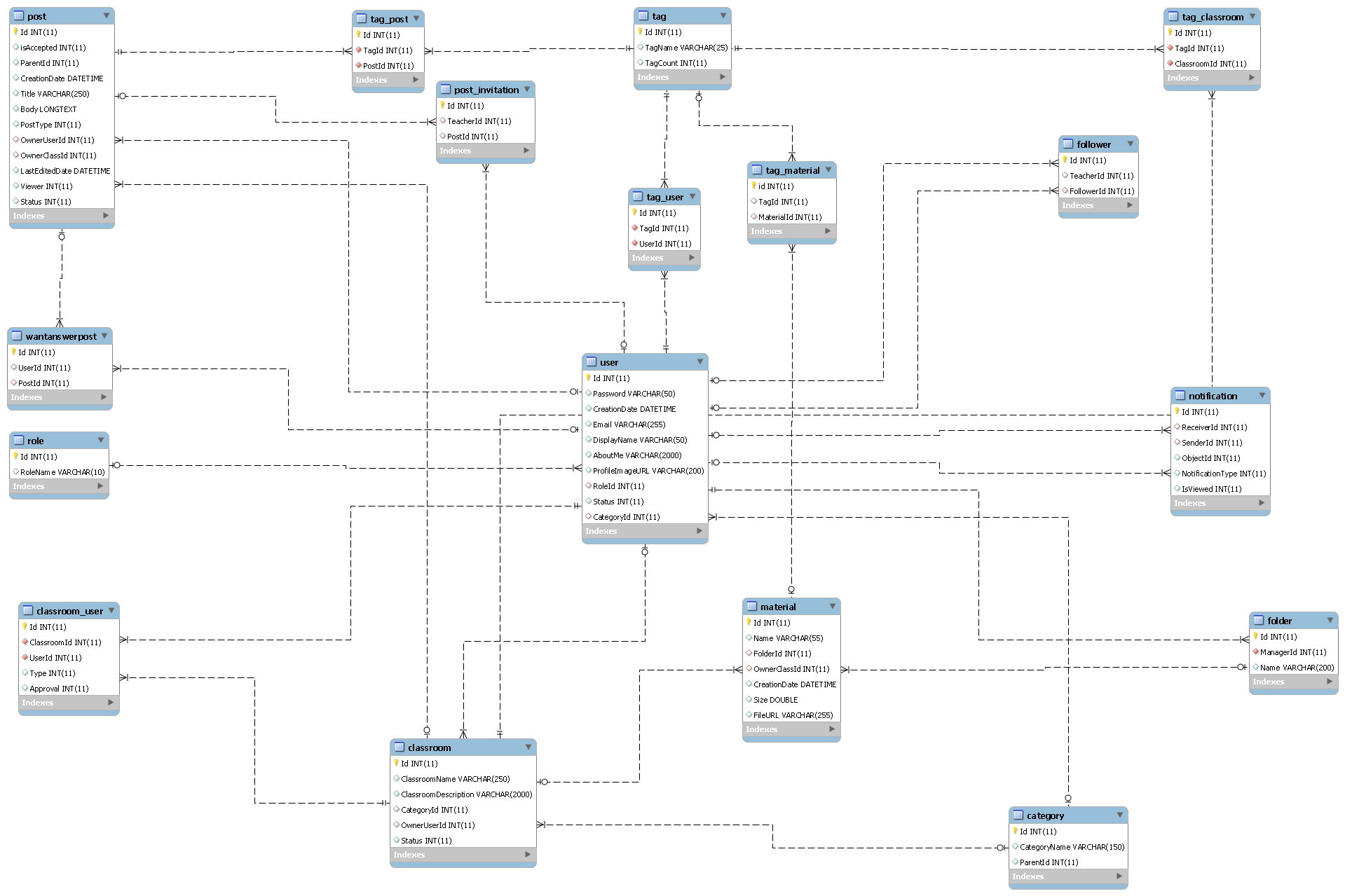
List B = {How to parse String to Int ,

How to convert String to Int ,

What is abstraction in OOP ,

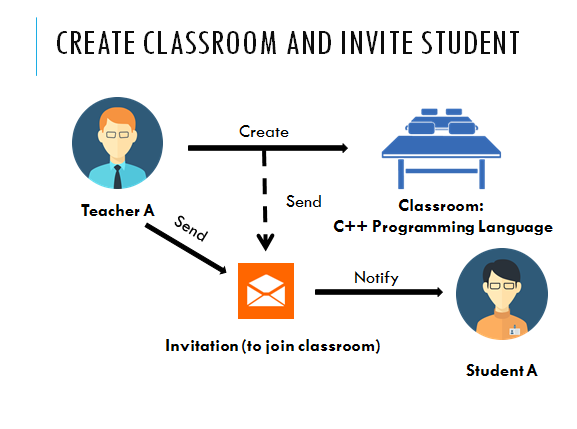
What is inheritance in OOP }

# Physical Diagram

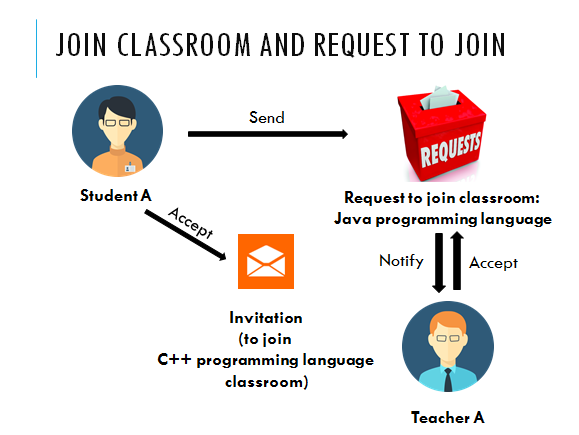


**Figure 16: Physical database diagram**

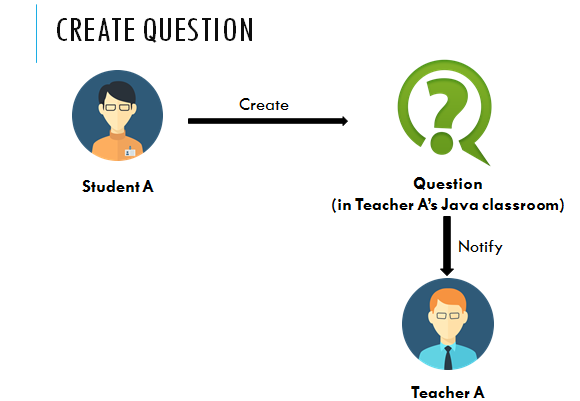
# Demonstrations



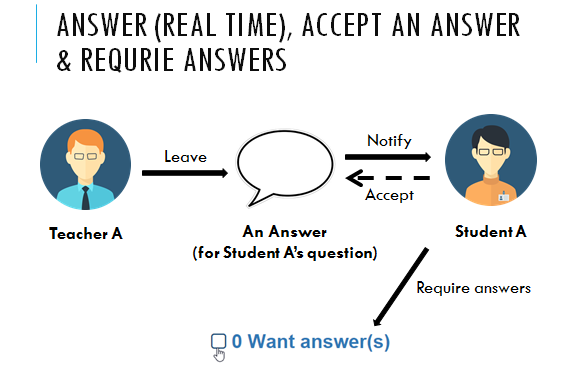
**Figure 17: Demo Create classroom and invite student**



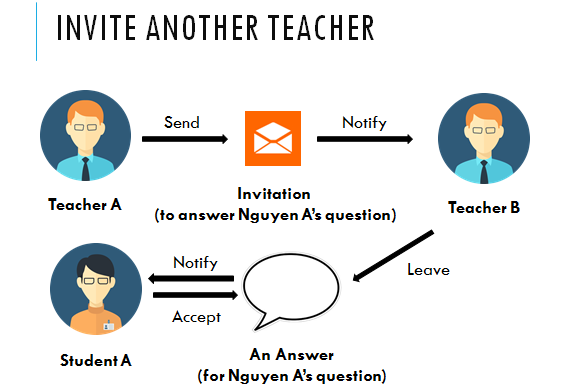
**Figure 18: Demo Join classroom and request to join**



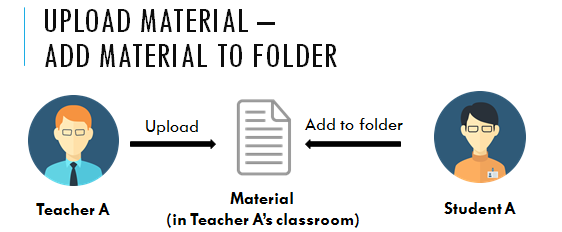
**Figure 19: Demo Create question**



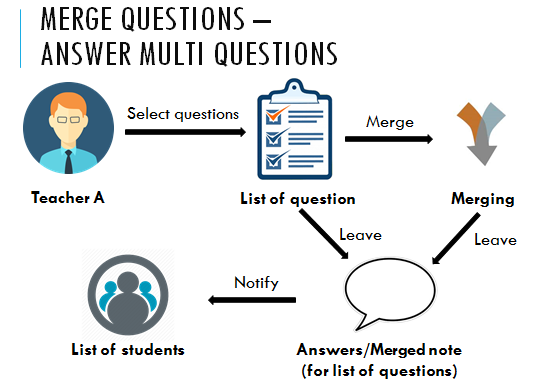
**Figure 20: Demo answer, accept an answer and require answers**



**Figure 21: Demo Invite teacher**



**Figure 22: Demo Upload material and add material to folder**



**Figure 23: Demo Merge question and answer multi questions**