

Project Design Document

Project Design Document

1. Introduction
 - 1.1 Project Overview
 - 1.2 Objective
 - 1.3 Expected Customers and Market
 - 1.4 System Features
2. Background
3. Specification
 - 3.1 User Login and Registration
 - 3.2 User search and match result
 - 3.3 User upload questions or answer
 - 3.4 Ranking System
4. System Architecture
 - 4.1 Architecture Diagram
 - 4.2 System Components
 - 4.2.1 Frontend Development
 - 4.2.2 Functions
 - 4.2.3 Middleware and Communication
 - 4.3 Description of Major System Components by UML

1. Introduction

1.1 Project Overview

Our project, CUHKSZ-Stack-Overflow, mainly aims to provide a Q&A platform for CUHKSZ programmers who have technical questions about their Computer Science courses' projects or assignments in CUHK(SZ). It also offers a social channel where some of students, especially senior students, could share their programming and working experience, provide the guidance in CS learning, and give the specific suggestions to the other younger students who may encounter the same problem or situation. We hope to create an environment where students could bravely post their programming problems while other students or teachers would willingly to see and reply those questions. Hopefully this kind of communication will help students fully understand their homework, improve their programming skills and enable them to accumulate programming experience. Eventually, those communication and connection could save much time for students learning CS and also reduce the teaching load for TA at the same time, which definitely will enhance both learning and teaching qualities in Computer Science courses.

1.2 Objective

Our thoughts of the system design come from a famous Q&A application StackOverflow. It is a public platform building the definitive collection of coding Q&A for professional and enthusiast programmers. The system is divided into two parts, the frontend part where users could search, post and reply the questions, and the backend part where developers and servers deal with the data transfer and management. Our goal is to provide a satisfied and perfect CUHKSZ-Stack-Overflow Q&A system particularly for programmers in CUHKSZ.

1.3 Expected Customers and Market

The target users of our system are programmers (all students take Computer Science course in our school, TAs, professors) of CUHKSZ. Everyone can access the system, as long as they have an school email account for registration. Users could register an account in the system when they first access the website. Additionally, they are enabled to create a nickname and other background information like age, interested research career, CS courses already taken, etc. They also need to sign up a contract to make sure they have the obligations to maintain Community harmony.

1.4 System Features

Our system is mainly composed of 6 parts, including registration and login, searching relevant questions, posting new questions, replying the questions, displaying the highlight and hot questions or topic, ranking the excellent answers. Users could upload their questions online and also have the access to invite other people to answer their questions. They also could browse those hot questions that remain answered to see if they could help or join in the discussion. Moreover, the searching function is also valid for users. They are allowed to search the questions by themselves with various filter types that could narrow down their searching range. More details of these features will be included in later sections.

2. Background

Our group observe that many student-programmers from CUHKSZ may encounter many familiar programming questions in their project or assignment. Their solutions mainly include: searching online, emailing TAs or professors, uploading questions in WeChat group, or assigning an office hour. Searching online sometimes may not be an efficient way, because the blog or some guidance information may not directly answer the assignment question. Even worse, programmers need to spend much time on filtering the huge amount of information and it becomes hard for them to get answers when the homework question is not relevant to the results on website. Uploading questions on WeChat group could get the detailed guidance and answer. But the new WeChat group will be created every semester for other students who may encounter the same problems. The connection between students already taken this course with the students taking this course right now is broken in this way. Raising questions in office hour is not convenient for programmers to solve their question immediately since they need to make an appointment and wait until that day comes. Therefore, our group would like to take the first step to change the current situation.

Based on the inspiration of public Q&A platform Stack-Overflow, our group thinks it is necessary to design a similar Q&A platform for programmers in CUHKSZ. It could help them solve their technical questions in time which could save their much time in finishing homework. programmers could use this platform to browse their questions on website and they will get the relevant answer immediately since there are many students who have already taken this course and they may encounter the similar problems. Also, the platform is more friendly. Programmers could use their nickname rather than true name and it is totally private for others. Therefore it would not make them shame about their questions. The platform also supports many formats of questions, which means programmers could use code block to express their specific question rather than word. It is much more efficient and clearer to use code block to express their idea. So students could find the help on our website when they encounter some difficult bugs.

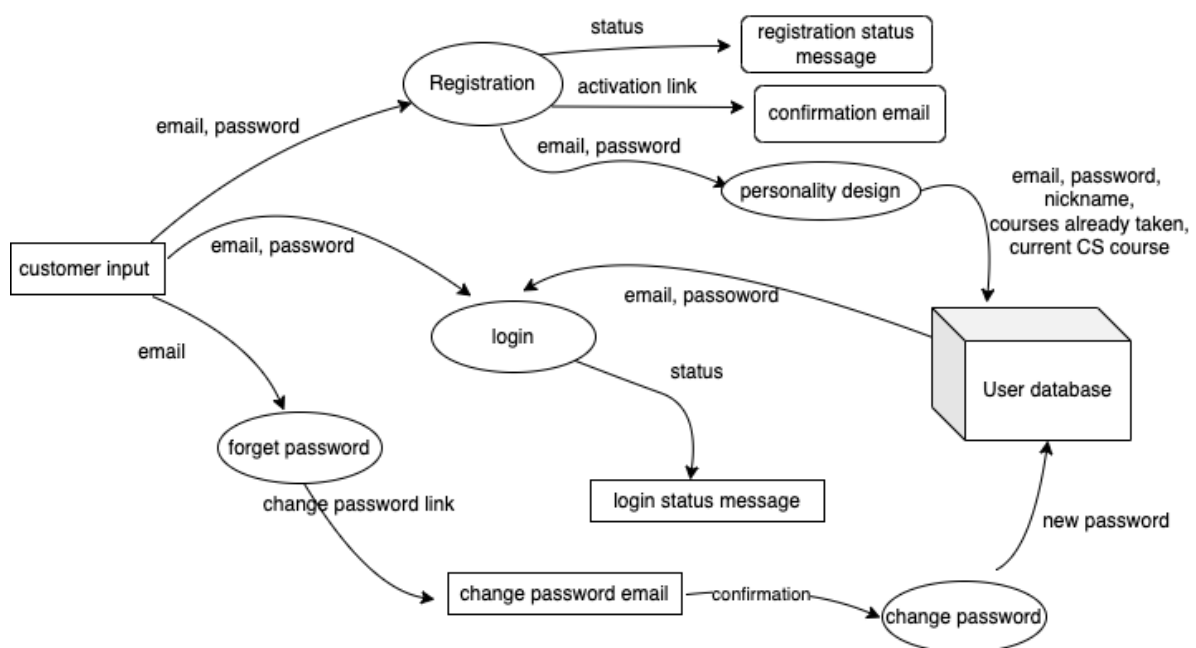
Our group also wants to build it as a social community for those programmers who want to raise some interesting topic to discuss with others. Programmers will have a platform to share their learning experience and their own suggestions to those younger students. And this friendly environment will also encourage more programmers to learn skills and be more creative and productive.

In the view of above, our groups borrow the ideas from the famous used app called Stack-Overflow in CS community, and we hope to create a CUHKSZ version for programmers in our school. This website can be used to upload the questions relevant to their course homework, support user use code block to express their problem and specific need. The website also will rank the high-quality answer and recommended every member in this community to read on their homepage, which raise interest for every community member to learn. By knowing the drawback of current Q&A mode, hopefully this system can boost the motivation for programmers to raise question and solve question immediately in CUHKSZ. Thus, empower every CUHKSZer programming skill.

3. Specification

We have divided our system into four main parts to show the data flow: User Login and Registration, User Search and match results, User upload questions, and Ranking System. This part of reports shows the data flow diagram and the description on the design specification of data flow and behavior of these four parts.

3.1 User Login and Registration

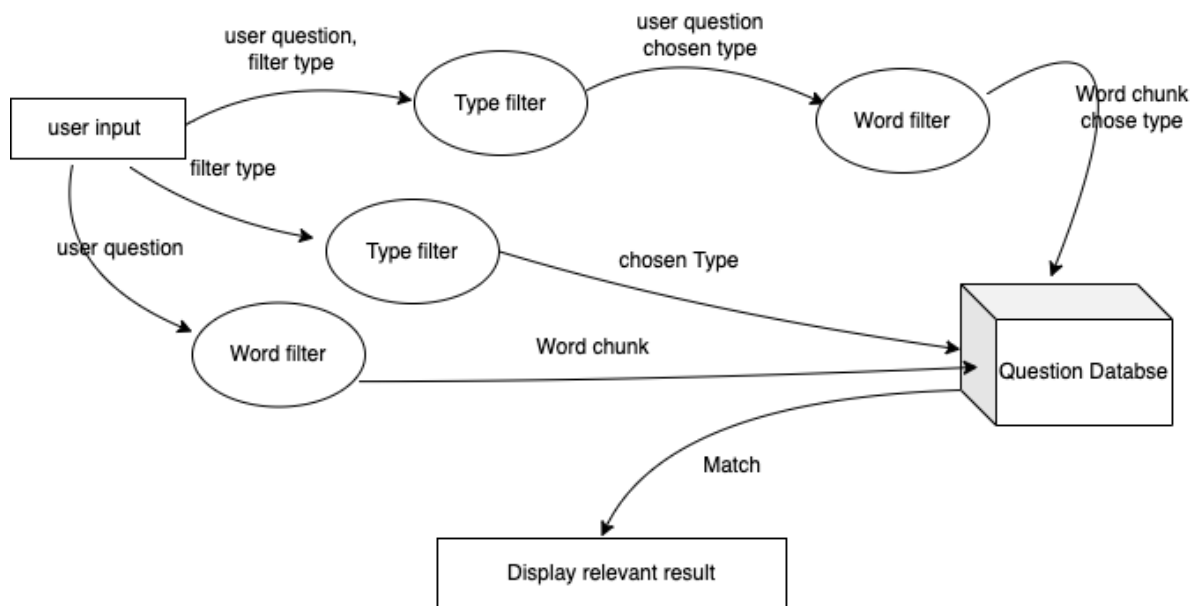


Users will have to go over a user registration process before they could use our system. The user registration function is mainly for the programmer to be personalized. To create an account, we require users to enter their email and password. A format validation check will be applied to ensure that the email format is like "xxx@link.cuhk.edu.cn". There would also be a data verification process that asks users to confirm their password by retyping in their desired password in another field. An email will be sent to the email account of the user afterwards and they are required to activate their account by clicking on the link provided in the confirmation email. Users who did not log in to the system could also view the contents of the pages. For those users who would like to post comments of menus, they will have to log in by typing their email as username and their own password. If users have any login problems, our system provides a

function for users to change their password. These users will be asked to type in their email for receiving an email, and in the email, they can click into a link to change password. By submitting the new password in the link, our database will update the user's data in the password field.

After creating the account, we will provide various types for users to personalize their information. They are able to have a nickname and some basic information like courses already taken and courses taking right now, language preference, etc. This basic information is public, and other users have the access to see these information. This function mainly help user to recognize many friends who have the same preference, and it also will be used to invite people to deal with some specific problem.

3.2 User search and match result

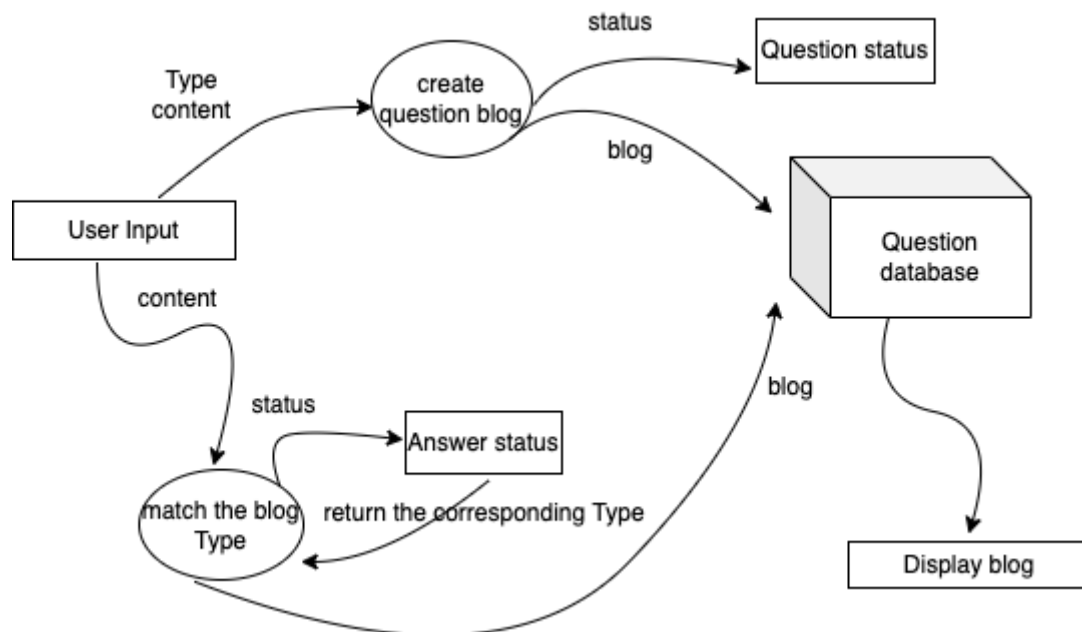


When users try to search some relevant information with its problem. The System will ask them to input the question content and the question type. The question type is defined by the system admin. The question type mainly include CSC3100, CSC3050, etc. The filter is composed of two parts. The first filter is course number like CSC3100, users could only pick one of those course numbers. The second filter is about the homework number, such as project 1. By this filter, the user can search the content in detail like CSC3100 project 1.

The users also could upload the question content. The system will cut the word into different chunks. And the database system will search some of those chunks by some algorithm and present the most relevant blogs to users. Users can search information by either the question content or question type.

The database will use the input word chunk or course type to match the corresponding blog. During the match process, we will develop a matching algorithm to match the source information and target blogs. The first blog that is found will be present on the top, which will be regarded as the most relevant blog. The system will show all the relevant blogs. And user can filter those blogs by upload time, by most approved, etc.

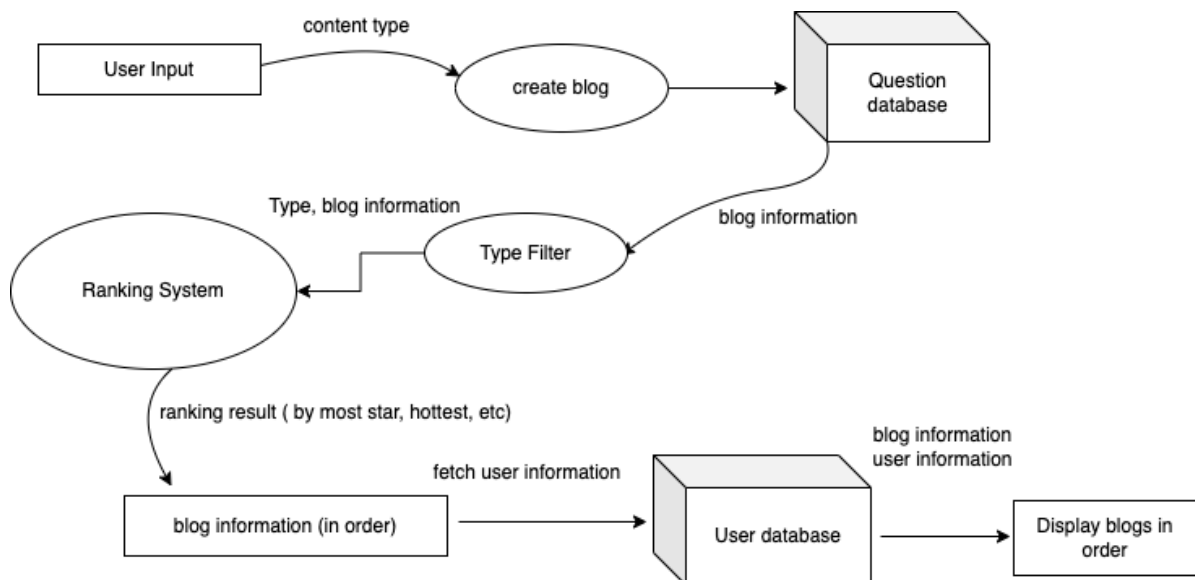
3.3 User upload questions or answer



When users try to raise a question in the system, they have to determine which type the question is. The types include the course numbers like CSC3150. After users choose the type, they are required to type their question content. The content support many formats such as markdown, word, text, code block, etc. Users feel free to type any question they want to ask within the type. The users will create the blog after they fill in the content, and the blog will be sent to the question database system with users' information.

When users want to answer somebody's question. Their answer will just be put under the original question blog. And the information will be stored into the database system, the course type of the answer will be the same as the question blog. The blog information will be displayed on the website for every user to see.

3.4 Ranking System

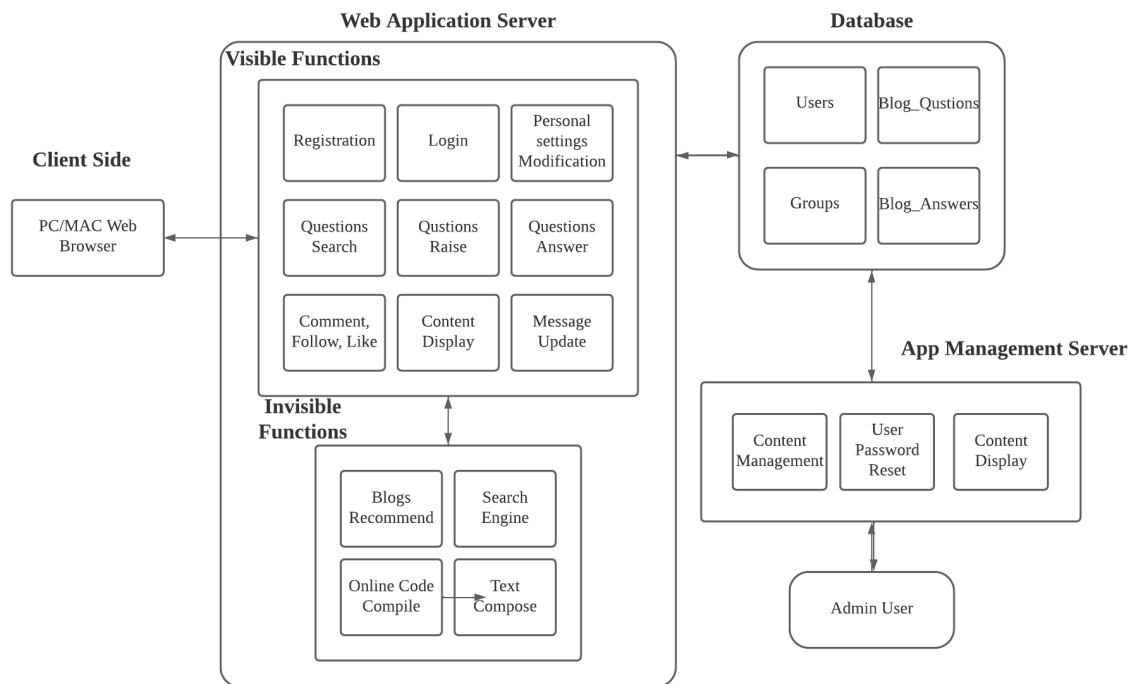


The ranking system is used when there are many blogs in the database system. All the blog information will be classified into many classes by their types. In each type, the ranking system will take some particular algorithm to rank all the blog information. The system will rank blogs by created time, most approved, most necessary questions needed to be answered, etc. The system will display the result to the user and recommend them to read or answer questions based on their background information.

4. System Architecture

This part mainly provides an overall ideas related to the design of our system architecture. In the first subsection of this section, an architecture diagram will be used to illustrate the structure of design in client sides, application server, and database server. In the second subsection, basic system components as well as the functions we plan to deploy in these components will be introduced. In the third subsection, we will describe the relationship of our major system components using UML diagrams.

4.1 Architecture Diagram



The figure above provides an understanding of the overall design of our system. Totally, our system can be divided into 5 parts, where 2 parts are regarding users, 2 parts for admin, and 1 part regards database.

In the front-end part, users can access our application via browser. Users interact with our application with the fancy user interface we provide and is able to use the visible functions below:

- Registry via email verification
- Login\logout
- Set personal information includes modifying password and uploading profile photo
- View the contents recommended by our system
- Search questions
- Edit and raise questions
- Answer questions raised by others
- Compile and run codes online
- Follow and like others' questions
- Comment on others' answers
- View the updated message that relates to users

To empower the visible functions provided to users, we have some invisible functions that are invisible to users while embedded in our system. These functions benefit the QoE by delivering contents in which users might be interested, increasing the speed of searching, providing a pretty text composition, and enabling users to compile and run codes on the website so that questions

and answers about coding could be illustrated in a vivid way. The invisible functions we provide to users are listed below:

- Blog recommendation: in charge of delivering the recently hot blogs to users.
- Search Engine: in charge of analyzing the questions that users type in the search box and return the correlated results in an efficient way.
- Online code compiling, running: allow user to compile and run their codes written in blogs online.
- Text composer: compose what users type and display it in a pretty style.

For the part of database, our database server maintains mainly four tables to store the information about users, groups, blog questions, blog answers.

To allow the handy management of our application, we provide another user interface for the admin user. The admin user is allowed to access and manipulate on all the data in the database via the functions we provide in app management server.

4.2 System Components

4.2.1 Frontend Development

Our produce would be presented as a website composed of HTML, CSS, JavaScript. For the development of the frontend, we plan to apply the framework Vue. Additionally, we plan to make our user interface to be user-friendly and fancy by designing a pretty UI.

4.2.2 Functions

In previous parts, 4 major functions are mentioned:

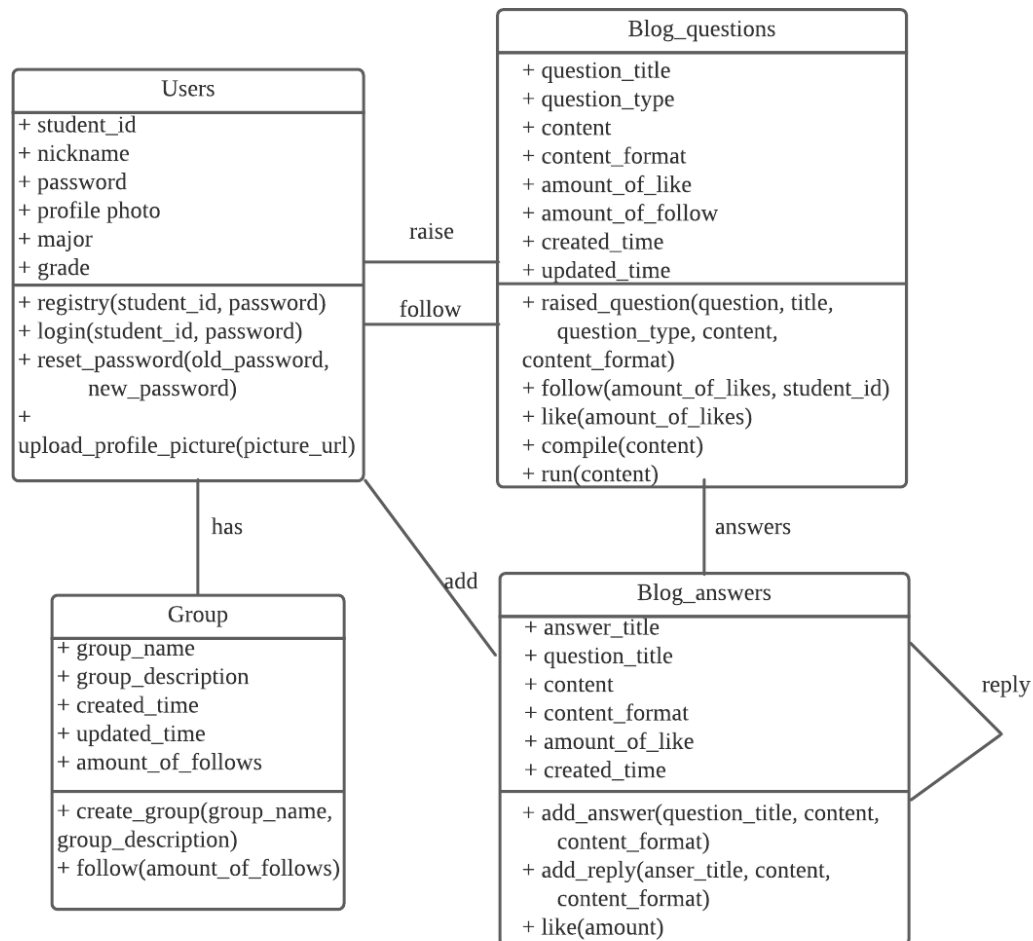
- User Login and Registration
 - This function requires the frontend to verify whether the user's input of email is whether the email account of CUHKSZ, the user's input of username and password is whether a correct format and then send these correct information to the backend. The backend is required to send the verification code to users via email. After the user type in the verification code, it should check whether it is matched and store the user's information into the database. Then the backend would return a signal to the frontend representing the user successfully register an account.
- User Search and Match Result
 - Users should be able to search for the questions and get the relevant results via this function. This function is empowered by the search engine we plan to implement in our system. Our system should chunk the questions users type in, also get the question types selected by users. Then it should use the algorithm we would design to find the relevant blogs in the database and present it to users. The search engines to be robust enough to deal with various questions from users, as well as efficient enough to return the search results within a time users can accept.
- User upload questions or answer
 - For this part, our system should allow users to type in the subject of questions, types of questions, and content of questions. What's more, the system should provide different formats of content like markdown, word, text, code block, etc. Our text composer should be robust enough to compose the various raw content user type in pretty. Our online compiling and running function should enable the users to execute the code written in code block.
- Ranking system

- This function should rank the hot blogs according to the amount of follows, likes, and views, of the blocks. It should keep updating the ranking of blogs in a reasonable period of time(every 30 minutes) according to the data stored in the database and recommend them to users.

4.2.3 Middleware and Communication

For the middleware, what we plan to be applied in our product is some middleware assisting text composition, code block's online compiling and running. These middleware would be together installed on the cloud server, and where our application on cloud server could directly use them. For the communication between frontend and backend, we plan to use POST and GET request to enable the backend to get the data from the frontend. Then, the backend would deliver the data back with data formatted in JSON. For manipulating data in database, we plan to directly use the API implemented in Django for some simple manipulations, and write some SQL statements for the complex manipulations.

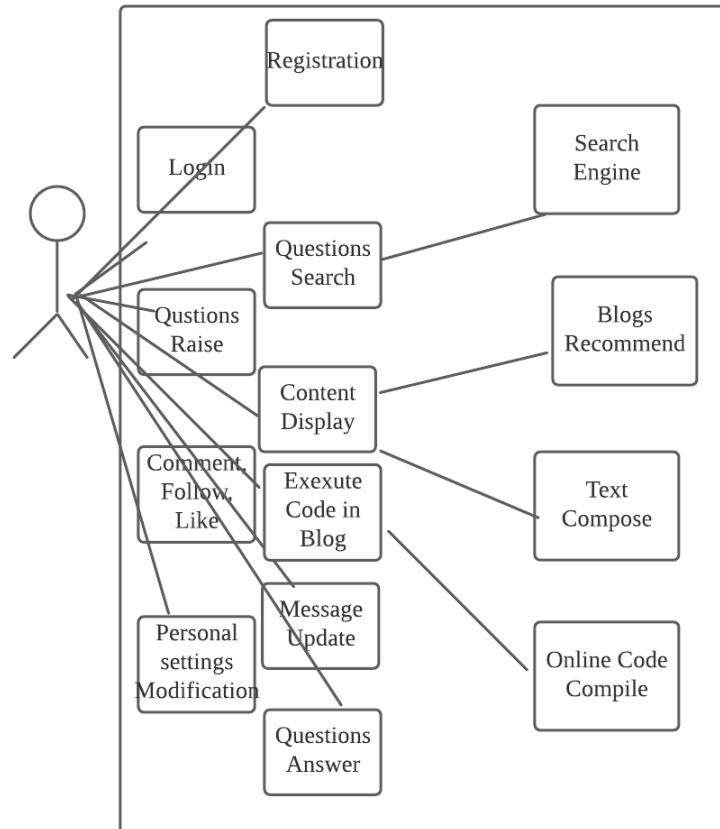
4.3 Description of Major System Components by UML



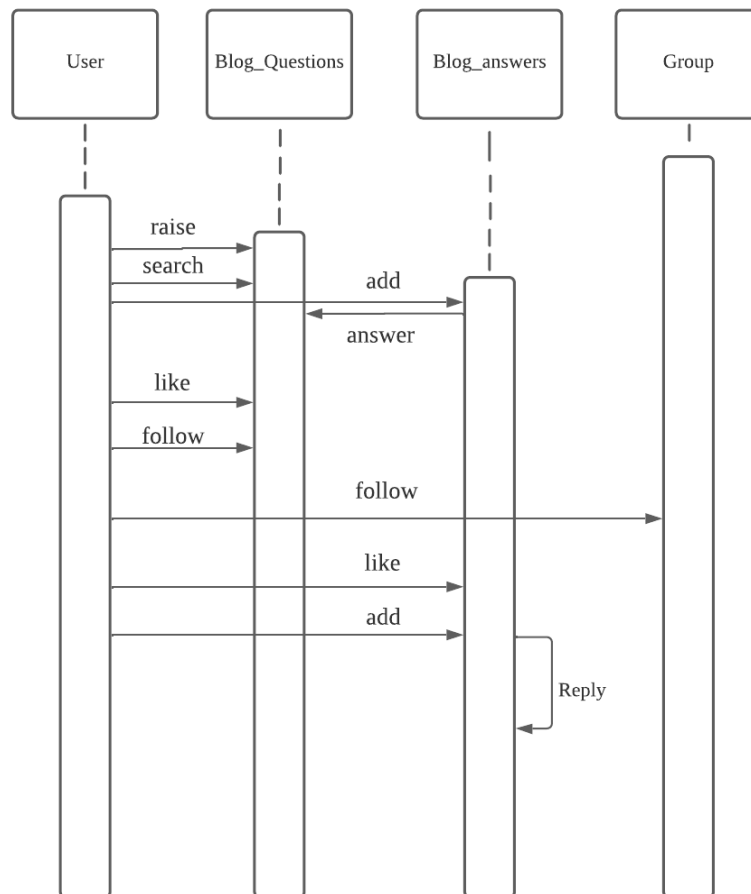
The above diagram shows the classes we plan to include in the development process and database. For each class, it has several functions which help to manipulate the information in each class and implement other functions we provided to the client. Class `Users` stores the use student id as the main key and contains other basic information of users. Class `Blog_questions` represents each blog of questions, which uses `question_title` as the main key and contains other feature describing the contents of this question. Class `Blog_answers` represents the answer answering to the question, which uses `answer_title` as its main key. As it shows, it contains

question_title which is the foreign key of class `Blog_questions`. For the class `Group`, it stores basic information of a certain group, which use group_name as main key.

Regarding the relationships between each class, `users` and `Group` has a multi-to-multi relationship, means one group contains several users and in users can follow several groups. For `users` and `Blog_question` as well as `users` and `Blog_answers`, one user can raise several questions and give several answers. Another relationship between `users` and `Blog_questions` is follow, which means one user could follow one question he/she is interested in. For the relationship between `users` and `Blog_questions` is that one user can add several answers. In addition, one special point is that `Blog_answers` can have relationship with itself, which means the one answer could be the reply to another answer but not limited to questions.



The relationship between the user and functions we plan to implement is shown in the above figure. The functions directly connected to the user are the ones user can directly use by clicking the icon or typing something. The functions not connected to users are the invisible functions which empower the functions that is visible to users.



The above figure is the sequence diagram which describes how does the process of QA works. First, the user raise a blog question and the question would be stored in the database. It allows other users to search then find out the question by the search engine. Then the user who want to answer this question can add an answer, and the answer would be automatically displayed under the corresponding blog of question. The the user can like and follow the questions, where the information would be added into the table of Blog_Questions in the database. The user could also follow a group and like a answer. Also, the user can add an answer to reply another answer, and this would also be stored in database and be queried from the database when displaying.