Group Members:

Ziwei Zhao (CCID: ziwei11)
Weixi Cheng (CCID: weixi)
Bingran Huang (CCID: bingran1)

1. Overview of System with A Small User Guide

For this project, we designed a system to allow the user to sign up and log in. After logging into the system, the user can post a question or search posts by inputting the keyword. The user can vote or answer the post after selecting it. As for the privileged user, in addition to the previously mentioned functions, they can also select a post to edit, add a tag or mark it as the accepted answer. Besides, the privileged user can give a badge to the poster by providing a badge name. On the homepage, users can log out at any time.

2. A Detailed Design of Software

(1) LogIn class:

The first screen of your system should provide options for both registered and unregistered users to login. Registered users should be able to login using a valid user id and password, respectively denoted with uid and pwd in table users. Unregistered users could be able to sign up by pressing the 'SIGN UP' button. After a successful login or signup, users should be able to perform the subsequent operations (chosen from a menu).

(2) SignUp class:

Unregistered users could be able to sign up by providing a unique uid and additionally a name, a city, and a password. After a successful login or signup, users should be able to perform the subsequent operations (chosen from a menu).

(3) SignIn class:

Main menu interface. There are three functions (Post Question, Search, Give Badge) for the privileged user and two functions (Post Question, Search) for others. By clicking the button, the user can jump to a specific interface and implement its function. There are both a logout button and a quit button on the interface.

(4) PostQ class:

The user could be able to post a question by providing title and body texts. The post should be properly recorded in the database tables. A unique pid should be assigned by your system, the post date should be set to the current date and the poster should be set to the user posting it.

(5) GiveB class (Privileged users):

The user can give a badge to the poster by providing a badge name. The information is recorded in the database with the badge date set to the current system date.

(6) Search class, ShowResult class:

The user could provide one or more keywords to search posts, and the system could retrieve all posts that contain at least one keyword either in title, body, or tag fields. If the user input more than one keyword, the keywords will be split by comma. For each matching post, in addition to the columns of the posts table, the number of votes, and the number of answers if the post is a question (or zero if the question has no answers) could be displayed. The result should be ordered based on the number of matching keywords with posts matching the largest number of keywords listed on top. If there are more than 5 matching posts, at most 5 matches will be shown at a time, letting the user select a post or see more matches. The user should be able to select a post and perform a post action.

(7) PERFORMPOSTACTION class:

If the user selected a question post, there will be an answer button and a vote button. If you click the answer button, the user can post an answer for the question by providing title and body texts. The answer should be properly recorded in the database tables. A unique pid should be assigned by your system, the post date should be set to the current date and the poster should be set to the user posting it. The answer should be also linked to the question. If you click the vote button, the user can vote on the post (if not voted already on the same post). The vote should be recorded in the database with a vno assigned by the system, the vote date set to the current date and the user id is set to the current user.

As for the privileged user, three more buttons (Mark, Add Tag, Edit) will appear on the interface.

(8) PERFORMPOSTACTION2 class:

If the user selected an answer post, there will only be a vote button. If you click the vote button, the user can vote on the post (if not voted already on the same post). The vote should be recorded in the database with a vno assigned by the system, the vote date set to the current date and the user id is set to the current user.

As for the privileged user, two more buttons (Add Tag, Edit) will appear on the interface.

(9) **Answer class, Confirm class, Compare class (Privileged users):**

The user could be able to mark the post (if it is an answer) as the accepted answer. If the question has already an accepted answer, the user will be prompted if s/he wants to change the accepted answer. The user can select to change the accepted answer or leave it unchanged.

(10) AddTag class (Privileged users):

If the selected post is a question, the privileged user could be able to see current tags of the post and can add tags to the post.

(11) EditP class (Privileged users):

If the selected post is a question, the privileged user could be able to edit the title and/or the body of the post.

(12) InputActionA class:

If the selected post is a question, the user can post an answer for the question by providing title and body texts. The answer should be properly recorded in the database tables. A unique pid should be assigned by the system, the post date should be set to the current date and the poster should be set to the user posting it. The answer should be also linked to the question.

3. Testing Strategy

We write 15 different classes, each class represents a different interface and function. So we can easily test our source code based on these classes.
Test cases include storing directly into the database before the system is running, and manually storing them when the system is running.
Here is our general strategy for testing, with the scenarios being tested.

For all users:

(1) Test registered user information (LogIn class, SignUp class, SignIn class)
(2) Test post a question (PostQ class)
(3) Test search a post (Search class, ShowResult class)
(4) Test answer a question (PERFORMPOSTACTION class, InputActionA class)
(5) Test vote a post (PERFORMPOSTACTION class, PERFORMPOSTACTION2 class)

For privileged users:

1. Test privileged users information (LogIn class, SignIn class)
2. Test give a badge (GiveB class)
3. Test mark an answer (Answer class, Confirm class, Compare class)
4. Test add tags for the post (AddTag class)
5. Test edit a post (EditP class)

4. GroupWork

Our team distributed tasks fairly evenly, we designed the system, interface and database together. It took about 5 days to complete the source code. In order to keep the project on track, we always discuss and share our own ideas. Also, no matter which team member encountered difficulties, others will help s/he to solve them together.
The following is the specific division of tasks.

Ziwei Zhao:
LogIn class, SignIn class, PostQ class, Search class, ShowResult class, PERFORMPOSTACTION class, PERFORMPOSTACTION2 class

Weixi Cheng:
LogIn class, SignUp class, SignIn class, ShowResult class, AddTag class, EditP class, Compare class

Bingran Huang:
GiveB class, Search class, ShowResult class, Answer class, Confirm class, Compare class, InputActionA class