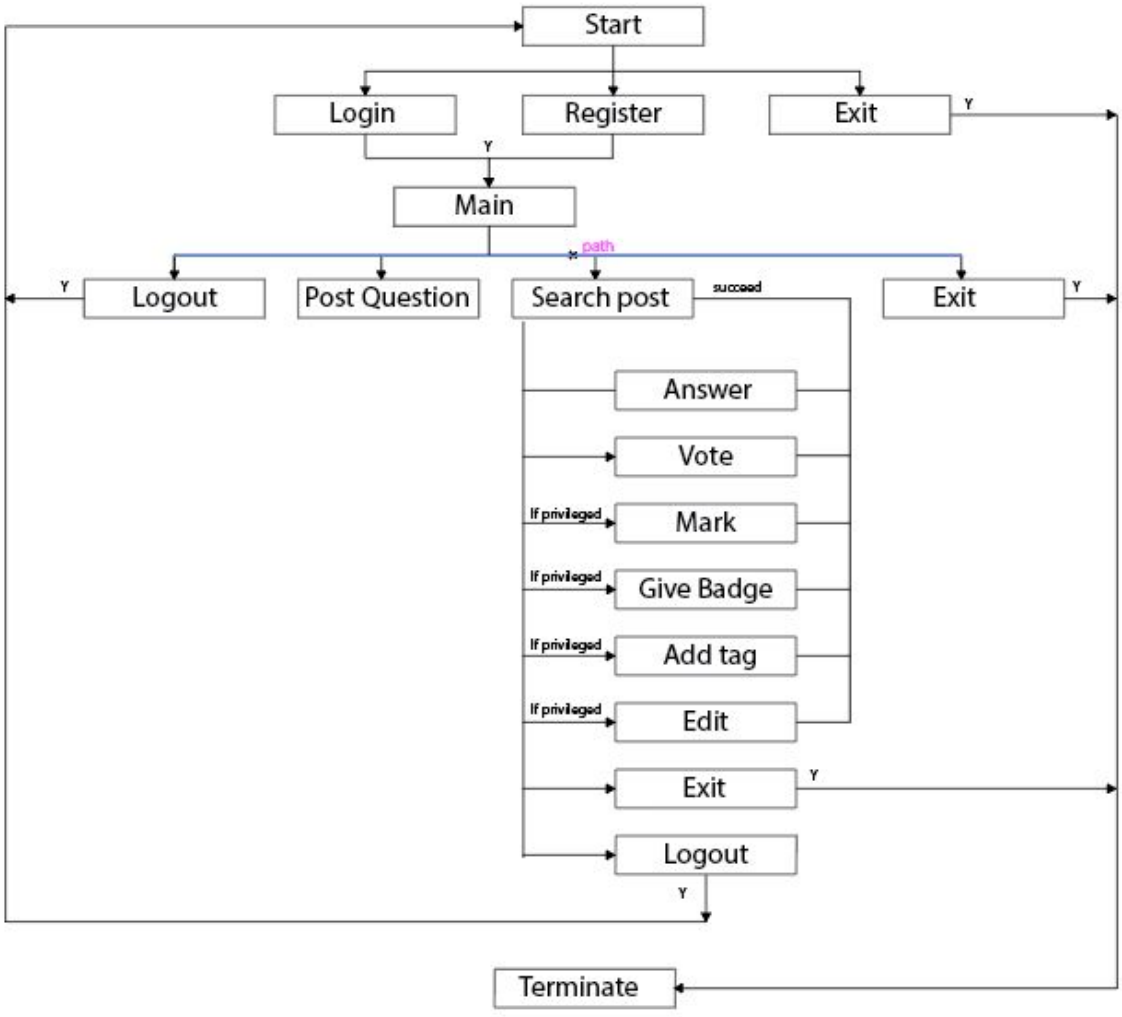
Report

1. General overview of the system

The program is a system that keeps the enterprise data in a database and to provide services to users. The uppermost screen of the system provides 3 options, including login, register and quit. After successfully login/register, users can choose to post the question, search for posts, logout or quit the program. If they choose the first option, the users need to provide the title and the body to the question. Option 2 is searching for posts: users enter keywords and the system will return all posts with matching keywords in descending order. After that, there are various post-action options such as answer/vote (both non-privileged and privileged users) and edit, add tags, mark an accepted answer, and give badges (only privileged users). Any of the actions done by users will be updated to the database. Besides, the system also provides 2 other options: log out (returning to the first screen), and exit (terminating the program).



2. Design of the software to deliver the major functions of the application:

Overall, the software contains 16 functions in total with 11 primary functions. For the detail function description, we also comment in the source code for each piece of code

Function	Description
Login handle	- Control login option for the user, is called within the main function. - Asks and verifies if users enter correct user id and password (invisible) - This function returns the user id.
Register handle	- Control the register option for the user, is called within the main function. - Users can create a new user id, name, city, and password (invisible). - Verify and save the data in the database. - This function returns the user id.
Post question	- Control the post option in the program. - Users need to provide a valid title and body text to the post. - Post id and date are assigned by system
Search for post	- Control the search for post options in the system. - Users can provide 1 or more keywords in either title, body, or tag field. - Display at most 5 posts at a time, ranging from the post with most matching keyword to the least one. - Users can choose to see more or not if more than 5 posts are matched.
Answer	- Control the answer option is the program, which is called within the search for post function. - Asks user to provides title and body to the post, post id and date are assigned by system - Save the data in the database
Vote	- Control the post- action: vote of the system, is called within the search for post function. - Users can choose a valid post id to vote. - Check if the user has been voted. If not, the number of votes increases, dates are assigned by the system, data is saved in the database.
Mark	- Control the post-action: mark accepted answer in the system, is called within the search for post function. - Asks for pid to mark - The answer chosen by the user will be assigned to the question if it's valid. - If the question already has the answer, ask users if they want to change or not. - The data will be saved in the database.
Give badge	- Control the post-action: give badge of the system, is called within the search for post function. - Prompt users to enter a valid name of the badge, and a valid user id

	- Check for the constraints of the ubadge table and save the data in the database
Add tag	- Control the post-action: add tags in the system, is called within the search for post function. - If the tag has not already existed, the data will be saved in the database.
Edit	- Control the post-action: edit in the system, is called within the search for post function. - Ask users which pid they want to edit. After verifying the pid, ask users to change title and body text of the post. Other information leaves unchanged - Data is saved in the database.
Main	- Control the flow of the whole system. - Control the data input, insert tables. - Create the uppermost screen of the system, asking users to either login, register, or quit the program. - After successfully login/register, users can choose to post the question, search for post, log out, or exit the program.

3. **<u>Testing strategy:</u>**

- **Static testing:** test each function separately to see if it runs smoothly in the system
- **Dynamic testing:** check how the function interacts with other features.
    - Performing a post action and search afterwards for the same keyword to see whether the displaying posts has a different information
- **Database**: For any changes we make in the program, open the sqlite to confirm those change
- **User inputs:** Check the inputs the user types in everytime with the domain constraint, and lower all input strings to prevent case insensitivity
- **Injection attack:**
    - Login using the following codes: “u1 or 1=1” and “u1; Drop Table users;”
- **Edge cases:**
    - Sign in: Users sign in with wrong password and wrong user_id
    - Register: Users sign up with empty password and empty user_id
    - Register: Users sign up with user id which is longer than 4 characters
    - Post question: If the system is full of posts (domain for pid is 4)
    - Search for post: If the user provides no keywords
    - Search for post: If the user provides all keywords not existing
    - Search for post: If the user provides same keyword more than once
    - Answer a question: Among displaying posts, there are no questions (all answers)
    - Answer a question: If the system is full of posts (domain for pid is 4)
    - Votes: Among displaying posts, current user has already voted all of them
    - Tags: Enter the already existed tags but capitalize all the letters.
    - Give badge: Give the same user with different badge on the same day

- Give badge: Among users have displaying posts after search, all of them has already received a badge on the day current users login
- Mark: Among displaying posts, there are no answers (all questions)

4. **Group work break-down:**

Method of coordination:

- Google Drive: For each day, create a separated folder and upload the document.
- In-person meeting: once a week to solve the difficult problems (approximately 3 hours/meeting) and assess the performance of each member
- Slack: online meeting and discussion
- Team leader: keep track the progress made by team member

Name	Tasks	Time
Chu Duc Thang	Write the program in python (search_post, answer, vote, mark, give_badge, add_tag)	Approx. 4-5 hours/week
	Test and debug the code	Approx 3 hours/week
	Assign the tasks and monitor the progress of each member	Approx 1-2 hours/week
Nguyen Minh Hoang	Write all the queries to process the data from the database	Approx. 2 hours/week (1st week only)
	Write the program in python (login_handle, register_handle, privileged_check, edit)	Approx. 2-3 hours
	Test and debug the code.	Approx. 4 hours/week
	Monitor the forum for questions and communicate with TA/professor for clarifications	Approx. 2 hours/week
Duong Hoang Son	Develops the database for testing the system	Approx. 3 hours/week
	Test and debug the code	Approx. 5 hours/week
	Prepare all the documentation	Approx. 1-2 hour/week (2nd week only)