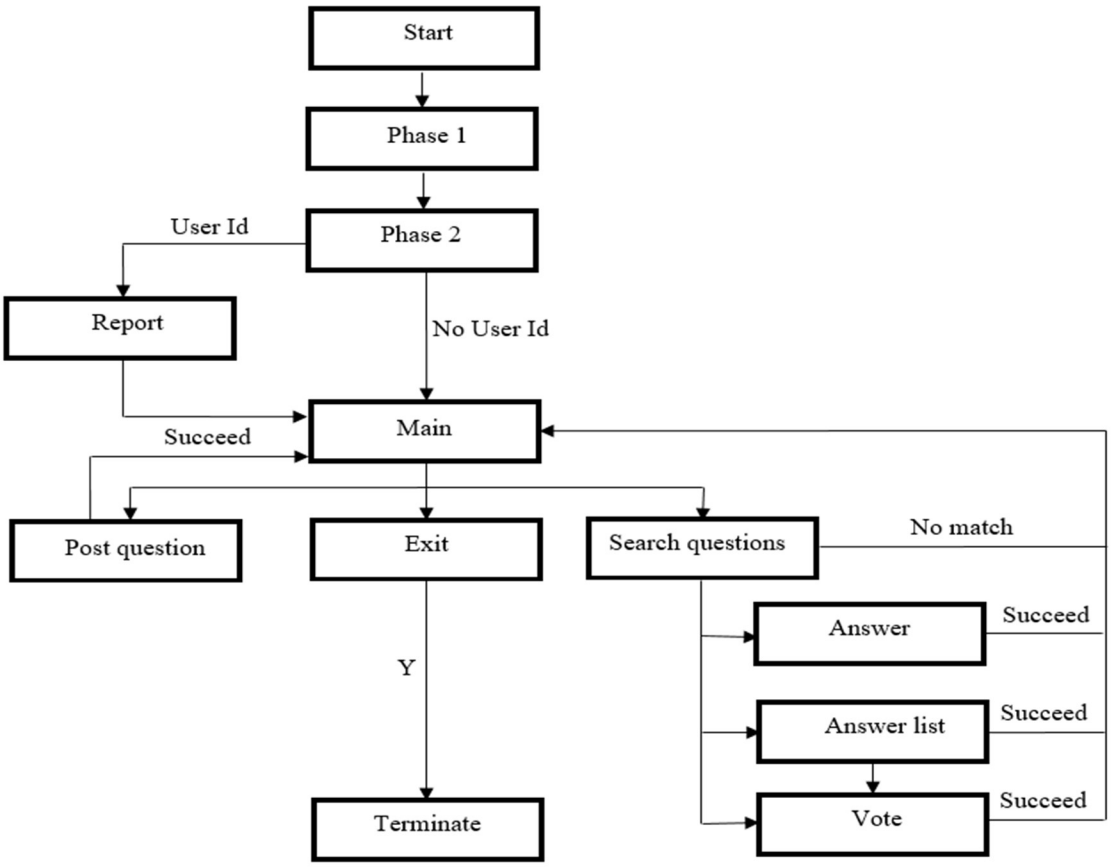
Report

1. General overview of the system

The program is a system storing data in MongoDB and providing basic functions for searches and updates. This program is divided in 2 phases: the first phase takes port number as user's input to connect with the server, then creates the 291 database (If it does not exist). Next, the program creates 3 collections (The system should drop them if they existed beforehand). In the next phase, users have the choice to provide user Id (A report showing number of user's questions, average score for questions, number of answer owned, score for the answer, and the number of votes for that user). After that, users can perform various tasks: post a question, search for questions, answer the questions, list answers of the question, and vote for questions/answers. After these actions, user can return to the main menu for further operations. Also, users can end the program if they want to. Any actions that change the data will be saved and updated to the database.



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

Overall, the software contains 8 primary functions in total. For the detailed function description, we also comment in the code for each function.

Function	Description
Phase 1	- Reads the JSON files, constructs the collection for each. - Asks for port number input, connects to the server according to the port number. - Creates 291 database. If any database/collections exist beforehand, drops them. - Extracts all terms (length > =3) in body and title field of Posts, adds an array to Posts collection, build indexes on those terms.
Report	- Asks user to provide user Id. - Shows: + Numbers of questions owned, average score for those questions; + Numbers of answers owned, average score for those answers; + Numbers of votes for the user.
Post question	- Controls the post option in the program. - Users need to provide a valid title, body text, and tag to the post. - Post Id, date and other fields are assigned by system.
Search question	- Controls the search for post and questions options in the system. - Users can provide 1 or more keywords in either title, body, or tag field. - Displays all posts, ranging from the post with most matching keywords to the least.
Answer	- Control the answer option in the program, called within the search question function. - Asks user to provide body text to the post. - Post Id, date and other data are assigned by system, updates database.
Answer list	- Controls the list, vote option in the program, called within the search question function. - Enables user to see all answers of a selected question (accepted answer comes first). - Display all fields of the answer from Posts. - Controls Vote function.
Vote	- Control the question/answer action: vote in the system, called within the list function. - Enable users choose a valid post to vote. - Check if the user has been voted (apply with user providing Id only). - If not, score field of posts increases; vote type, post Id, vote Id, dates are assigned by the system, data is saved in the database.
Main	- Controls the flow of the whole system. - Controls the files input. - Creates the uppermost screen of the system, asking users to provide user Id. - Users can choose to post the question, search for post, or quit the program.

3. Testing strategy:

a. **Static testing:** test each function separately to see if it runs smoothly in the system
b. **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.
c. **Database**: For any changes we make in the program, use MongoDB commands to confirm.
d. **User inputs:** Check the inputs the user types in every time with the existing data and lower all input strings to prevent case insensitivity.

- **Edge cases:**

- Phase 1: Users provide the wrong type of port number.
- Phase 1: Drop the database/collections if they are already existed.
- Post question: If users provide an empty/wrong type of post.
- Search for question: If users provide no keywords, or keywords provided are wrong type/not existed.
- Search for question: If the user provides same keyword more than once.
- Answer a question: Search for keywords but no result returns.
- List answer: If there are no answer for the question.
- Vote: If user chooses a post that has already voted.
- Vote: Check if the constraint applied for registered user only or not.

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	Writes phase 2 in python (answer question, list answer, vote).	Approx. 3 hours/week
	Assigns the tasks and monitor the progress of each member.	Approx. 1-2 hours/week
	Tests, debugs the code.	Approx. 3 hours/week
Hoang Nguyen	Writes phase 2 in python (post question, search question).	Approx. 2-3 hours/week
	Monitors the forum for questions, communicates with TA/professor for clarifications.	Approx. 2 hours/week
	Tests and debugs the code.	Approx. 3 hours/week
Duong Hoang Son	Develops the database for testing the system.	Approx. 3 hours/week
	Writes phase 1 in python.	Approx. 3 hours/week
	Prepares all the documentation.	Approx. 1-2 hour/week (2nd week only)