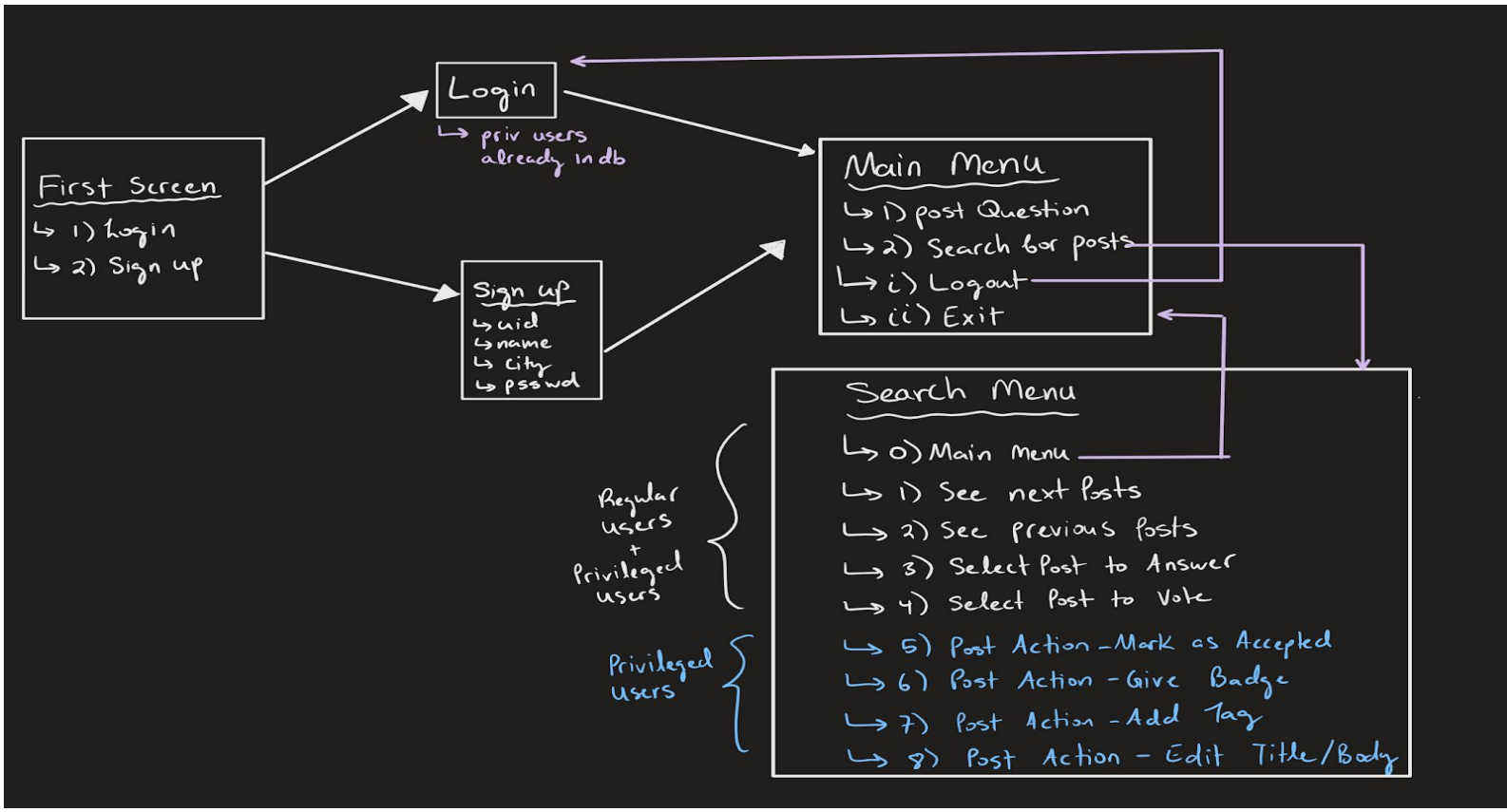
MINI PROJECT 1 DESIGN REPORT

USER GUIDE

To run the program in the command line type the following: python3 project1.db database-name(argument1)
The use starts off on the first screen and navigates through the application by entering a number corresponding to the menu option. Once signed up with or logged in with proper credentials the user must either post a question or search a keyword to get a result table. From there the user can perform any of options in the search menu by specifying a pid. When answering a post the pid selected must be that of a question. Similarly when “marking” an answer to be the accepted answer the user must choose an answer to a question post. The other options follow straight forward procedures and work as the title suggests.

GENERAL OVERVIEW



Overall the design is based around a question forum with various options for users to interact with posts such as answers, votes, scrolling through posts, etc. The data flow given illustrates the data flow throughout the program.

SOFTWARE DESIGN CHOICES

The connection to the database and cursor object are global variables since the data-base is queried from various functions so utilizing the global scope makes it possible to access these variables from anywhere. The overall design philosophy was to modularize all code blocks into functions so that each user option is self contained within its own function and can be called from a menu. Every user input option is verified for correctness and will loop until a valid input is received.

FIRST SCREEN AND MAIN MENU

The first screen allows users options to login or sign up by giving a unique uid, name, city, and password. Once logged in the user will be forwarded to the main menu which allows options for to post a question, search for a post, logout, or exit. When a question (any post) is created a unique pid is assigned by the system, in order to uniquely identify posts. Logging Out allows for users to switch accounts and login in with a different user. There is also an option to exit the program entirely. Lastly the Search Posts will take user input to search for a keyword within the title, body, or tag. All invalid input will result in a loop back until proper input is given.

SEARCH MENU

The results of the keyword search will be outputted to screen with the following columns pid, post date, post title, post body, poster, number of votes, and number of answers ordered by the count of keywords found. In particular the search utilizes the dictionary data structure to associate each row (key) of the posts table with a count (value). Options one and two allow the user to scroll through the list of posts while displaying at maximum five posts on the screen at a time. Option three allows the user to give an answer by selecting a post that is a question existing in the data-base. Option four allows the user to vote on a post that has not been voted on by the user already. These options one through four can be performed by all users. The following options five to eight can only be performed by privileged users. Option five allows the user to select an answer and mark it as the accepted answer for the question it is answering. This can be changed to another answer if the user so chooses. Option six allows the user to give a badge to another poster(user) by specifying the badge name, however the poster can only receive one badge per day. Option seven allows the user to add a tag to a post. Option eight allows the user to edit the title or body or both at once for a single post specified by the user selecting a post id.

TESTING STRATEGY

To perform adequate testing a data file with inserted test cases was created and used to populate the database with predictable results so the program can be properly verified. Consistent tests were run every time a new feature was created, in order to keep a verified code base and keep errors from propagating throughout the subsequent code. The coverage of test cases include all possible branches from a particular action in the code. Examples include, but are not limited to the following, answering questions multiple times with the same user and different users, searching for body, tag, title keywords including phrases. Voting on all possible posts, and various other functions with different users. The most common bugs encountered were related to indices of the list returned from queries in search posts and properly dealing with values where columns of the table can be none. Additionally, some errors encountered were related to queries not being committed before a function was being returned resulting in no updates happening in the program.

GROUP WORK STRATEGY

Initial planning was performed together to discuss design and programming structure (variable names, formatting, function design, etc.). To keep the project on track and manage progression a github repository was maintained with uploads consisting of updating versions of the code each time a new feature was implemented.

Smriti - Worked on the login/sign up menu, post action mark, and post action edit **(15 hours)**

Areez - Worked on the main menu, post questions, post action tag (regular queries 1 and privileged query 3), debugging search **(15 hours)**

Gurick - Worked on Search Posts, post action answer, post action vote, post action badge **(18 hours)**

All members focused on testing and general debugging throughout the whole code