A general overview of the system with a small user guide

To supply the filename for the database use the command -i filename.db
So to run on the command line the user can type python3 miniproj1.py -i filename.db

The system begins with a login screen that prompts the user to select and option from:

1 - Sign up
2 - Login
3 – Logout

The user must type the number 1 or 2 or 3 to select an action.
For every invalid user input throughout the program, the program displays an error message and prompts the user to enter a valid input.

If the user selects **option 1**:

The user is prompted to enter a uid which starts with u and is 4 characters longs.

The user is prompted to enter their name.

The user is prompted to enter a password twice.

Then the user is asked which city they live in.

Then a thank you message displays and the user is brought back to the login screen.

If the user selects **option 2:**

The user is prompted to enter their uid.

The user is prompted to enter their password.

If the password is correct an interface for the tasks menu appears and the user has to select which task they want to perform. The task menu contains 5 options for a normal user (4 tasks and an option to logout) while the task menu for a privileged user contains 9 options (8 tasks and an option to logout).

If the user selects **option 3** they logout.

Details of the design of the software

The project begins with all necessary import statements and setting up the connection to sqlite and the database file using command line argument.

The project was approached using functional programming.

Each function has a block comment which describes how the function does its task and the parameters used and the return value. There are inline comments too to explain specific lines of code.

There is a function for the menu, login and signup. The menu function calls login and signup functions when necessary.

There is a function called filter which is used for task selection and when a task is selected it calls the function for that task.

Each question has 1 function to perform the task except for Q2 which uses a helper function too.

There are 2 other functions namely user and maincheck which are helper functions to check for valid users in the database.

Testing Strategy

We used the data from assignment 2 and modified the tables to meet the spec and added privileged users. All testing was done on pycharm since it allows us to see all the data for each table.

After finishing our individual questions, we all tested our questions on our own.

In every meeting we held, we tested the whole code from scratch and fixed mistakes and debugged errors. We analyzed and tested each function and checked if it is making the necessary changes in the database. We also checked for invalid inputs.

During the meetings, the testing and debugging was done by Tanyaradzwa(gozhora) and he shared his screen and we all used our combined ideas and efforts to fix the bugs and mistakes we had.

We ensured SQL injection is avoided by sanitizing all input by parametrization.

Group Work Breakdown

The platform for discussion and sharing the code was a discord group which only had 3 of us group members and the platform to hold meetings was google meet using our ualberta emails.

The login interface was done by Akrash(akrash) which took approximately 2hours and modified by Robert(rjoseph1) which took approximately 30 minutes.
Reading the filename by a command line argument was done by Akrash(akrash) which took approximately 30 minutes.

Q1 was done by Robert(rjoseph1) which took approximately 2hours.

Q3 was done by Akrash(akrash) which took approximately 2hours.

Q4 was done by Tanyaradzwa(gozhora) which took approximately 2hours.

Then we held a meeting where implemented a search_easy() function which simply asks the user to enter a pid and returns that pid. We used the search_easy() function to test our codes. The meeting took approximately 3hours.

Since Q2 seemed bulky we decided to have one person do Q2 then each of the other two members to do 2 questions of the privileged users.

So Q2 was done by Tanyaradzwa(gozhora) which took approximately 6hours.

Q1 and Q2 of the privileged users was done by Robert(rjoseph1) which took approximately 4hours.

Q3 and Q4 of the privileged users was done by Akrash(akrash) which took approximately 4hours.

Then we held a meeting and tested all our questions from scratch and did a lot of debugging and error fixing. The meeting took approximately 6hours. We had a 20 minutes break in the meeting.

The report was made by Akrash (akrash) which took approximately 1hour.

SQL injection was handled by Tanyaradzwa(gozhora) which took approximately 30 minutes.

String matching was done by Tanyaradzwa(gozhora) while he did Q2.

Making the password nonvisible was done by Robert(rjoseph1) which took approximately 30 minutes.

We held another meeting to do all possible error checking and testing and debugging to finalize our project. The meeting took approximately 3hours.