Ritchie Jao

Cole Murray

CS 166 Project Writeup

Our implantation for this project was rather simple. Since we were provided with the table required, we just needed to determine the queries to perform what we needed.

**Users**

**New User Registration (Ritchie):** For a new user to register an account, we take in the user’s name, password, and phone number and insert it into our table. We do not allow duplicates as the username is the primary key in our table.

**User Login/Logout (Provided) :** To login, we take input from the user. With the provided input, we perform a lookup into our user table to confirm that this is a valid login. After this is confirmed, the user is logged into our system. For logging out, we simply just take away the menu from the user.

**Delete One’s Own Account (Cole)**: When a user decided to logout, we perform a check to make sure they are not the init\_sender of any current chats. If they are, we throw an error message stating this. Otherwise, the account is removed from the database.

**Browse Contact/BlockList (Cole):** To browse a contact/blocklist, we perform a lookup into the user’s respective list. In the contact list, we also return the user’s status messages.

**Add/Delete Users From Contact/BlockList (Ritchie):** To add a contact, we perform a search by either the username or phone number. Depending on what is listed, we perform a lookup with the respective information. We perform a check to confirm the user was found, adding or displaying an error message accordingly.

**Browse Current Notifications (Cole):** To browse current notifications, we perform a lookup into the notifications table and print the results. Once we have printed the messages in the notifications, we removed them (marking them read).

**Browse List of Current Chats (Ritchie):** To browse the current chats, we perform a lookup into the chat\_list table. Inside this table we return any rows in which the member matches the login of the user browsing.

**Chats**

**Add/delete members (Ritchie):** A user is able to add or delete members to a chat if they are the initial sender. A user can also delete the chat, which causes all messages to be deleted as well.

**Chat Viewer (Ritchie):** The chat viewer recalls back the last ten messages. It also gives the option to view earlier messages, in increments of ten.

**Messages**

**Create New Message (Cole / Ritchie ):** When a user is in a chat, they are given the option to create a new message. This system will prompt them to enter their message text. It will also prompt for a destroy timestamp. After this, the system will ask if there is any attachments. If yes, it will take these inputs. From here it will add it into the correct tables. Each message has the author, creation timestamp and text.

Messages are checked to see if their destruction timestamp is up when displaying them in the chat window. If they are, they are deleted before they are displayed.

**Problems Encountered**

During this project, we didn’t encounter many errors. Difficulties we faced were making decisions on how to implement the features. Most of the errors we experienced were simply programming errors rather than logic errors.