Jeremy Mah
Ryan Goo

**Project 3**

We worked together on every part of this project. Our program starts with a main menu where it asks for user login, create user, and exit. On the create user part, we added an option to add status message. After you login successfully, it will bring you to our main menu.

**Main menu**

1.Add to contact list - Adds
2. Browse contact list - Shows all contacts along with their status
3. Read notification list - Shows a list of message id's of messages that are unread. The notifications will delete themselves after viewing this.
4. Browse block list - Shows list blocked users
5. Delete this account - Deletes the user's account and everything associated with it.
6. Add to blocklist - Adds recipient to block list but does not remove recipient from contact list.
7. Browse chats - Shows list of chat_id of the chats the user belongs to.
8. Open Chat - Prompts user for a chat_id. Brings user to chat menu if successful.

Opening a Chat will bring you to a new menu.

**Chat Menu**

1. Browse members of chat - Shows the Names of
2. Add member to chat - Adds member to chat. It it is more than two people, then it is group chat.
3. Delete member from chat - Deletes member from chat. He can still see the old messages of the chat but not any new mesages.
4. Delete entire chat- Deletes the entire chat. We had to use on delete cascade for this.
5. View Messages- Views messages and shows url and type. URL and type displays null if they do not exist.
6. Create New Message- Creates new message for chat. This is in this menu and not the main menu because you have to open a chat before typing in a message. (similar to how facebook messenger works)
7. Delete Message- Deletes the message from everything related to it.
8. Edit Message-Edits the body of a message.

**Running into problems**

We ran into numerous problems while implementing the project. Here is a list of what we had problems with and how we solved those issues.

Browse Contact\Block list - It was simple to browse a contact/block list, but to show user status messages was a little bit harder. We added a projection on the status along with the list_member of the lists to show status messages. This was a problem because status messages would display "null" if there is no status.

Starting new chat- We spent quite a while on starting a new chat because we weren't sure how to assign a new chat_id. We eventually found out we were supposed to use getCurrSeqVal, and then it took more time to understand how it worked. The example of creating lists for a new user helped a lot in understanding how to use getCurrSeqVal. We also had to look into load_data.sql to figure out we had to use "chat_chat_id_seq". It also took us quite some time to sort it in chronological order. We joined messages and chat to to sort by timestamp but it showed a lot of results. We fixed this by using max(msg_timestamp) and group by chat_id.

Delete Entire Chat - When we initially coded our deleting one's own account, it had errors regarding foreign keys in other tables, even though we had queries to delete those as well. We fixed this problem by modifying create_tables.sql by adding "ON DELETE CASCADE" onto the foreign keys.

Chat Viewer- Initially, we had no idea how to show only the last 10 messages but we searched online and figured out what to do. We had to learn how to use LIMIT to display a certain number of messages and OFFSET to show the next batch of values. We also used ORDER BY to sort the messages via timestamp.

Show Messages- We had a lot of trouble showing media attachment along with a message. We initially tried to do it all in one query, but that only showed only the messages with media attachments. We then tried to do a union of all the messages and the messages with attachments we had errors because the amount of columns were different. To fix this we had to use NULL AS URL. This gave us all messages and included the attachments, but for the messages without attachments, it showed null.

**Modifications**

create_indexes.sql - We created indexes for media_attachment and message for quicker viewing of messages. It is being indexed by msg_id and chat_id because if it is indexed by chat_id it will receive all the messages faster and indexing my msg_id, we can search for for media attachment's quicker.

create_tables.sql - We added 'on delete cascade' to chat,chat_list, and message.

Messenger.java-import java.util.scanner
included timestamp when showing chat list
For block list: We do not delete recipients from contact list when adding them to block list (similar to aol messenger).