

# Project Report

## I. Introduction

This project is to program a basic messaging app using Java. The app allows users to communicate with other users in a chat with the use of messages. Although the app front end is Java, the back end of the app uses SQL in the form of PostgreSQL. Due to the console-based nature of the application, we have included some additional functionality. In addition, there are indexes that are used to speed up the query efficiency for all queries that are run during the application.

## II. Interface and Function Description

### 1. Menu System:

The menu system contains entrance menu, main menu and five sub-menu.

- (1) Entrance menu consists of options: Create user, Login and Exit.
- (2) Main menus consists of options: Contact list, Block list, Chat list, Delete account and Log out.
- (3) Sub menu of Contact list consists of options: Browse contact list, Add to Contact, Delete contact and Return to main menu.
- (4) Sub menu of Block list consists of options: Browse block list, Add to Block, Delete block and Return to main menu.
- (5) Sub menu of Chat list consists of options: Browse chat list, Add a new chat, Delete chat and Return to main menu.
- (6) Under Chat list, the submenu consists of options: View Message, Add chat member, Remove chat member and Return to chat list.
- (7) Under View Message, the submenu consists of options: Add Message, Edit Message, Delete Message and View Earlier Message.

### 2. Functions:

- (1) CreateUser: Users are asked to enter Login ID and password through console in order to create a new account.
- (2) Login: Users are asked to enter Login ID and password through console in order to access the functions of message app.
- (3) DeleteAccount: After Log in, users can choose to delete account through main menu. Once accomplished, all the information associate with the Login ID will be deleted, and user will be enforced to log out.
- (4) AddToContact: Under sub menu of contact list, user is able to add contact member by entering login ID through console. If the Login ID is in block list, user can choose to move it to contact list or cancel the action. If the Login ID is already in contact list or not existing, the action will be vacated.
- (5) DeleteFromContact: Under sub menu of contact list, user is able to delete contact member by entering login ID through console. If the Login ID is not in contact list or not existing, the action will be vacated.
- (6) ListContacts: Under sub menu of contact list, user is able to list all the contact member in console.
- (7) AddToBlock: Under sub menu of block list, user is able to add block member by entering login ID through console. If the Login ID is in contact list, user can choose to move it to block

list or cancel the action. If the Login ID is already in block list or not existing, the action will be vacated.

- (8) DeleteFromBlock: Under sub menu of block list, user is able to delete block member by entering login ID through console. If the Login ID is not in block list or not existing, the action will be vacated.
- (9) ListBlocks: Under sub menu of block list, user is able to list all the block member in console.
- (10) CreateChat: In chat list submenu, user is able to create a new chat by entering members and message content through console.
- (11) DeleteChat: In chat list submenu, user is able to delete a new chat by entering chat ID if the user is the initial sender. If the user not the initial sender, the action will be vacated.
- (12) ListChat: Under sub menu of chat list, user is able to list all the block member in console. And under list chat mode. Users can choose to view message and manipulate group members through sub menu.
- (13) AddToChat: Under chat list mode, user is able to add member to chat in console if the user is the initial sender. If the user not the initial sender, the action will be vacated.
- (14) RemoveFromChat: Under chat list mode, user is able to remove member to chat in console if the user is the initial sender. If the user not the initial sender, the action will be vacated.
- (15) ChatViewer: Under chat list mode, user is able to View message. The default message number to be shown is 10. While the submenu to manipulate the listing message will be shown.
- (16) DisplayMessages: function will be called once enter into view chat message mode. And if the total message is larger than 10, it can be called to display earlier message.
- (17) NewMessage: Under view chat message mode, user can add a new message in the chat by entering message content through console.
- (18) EditMessage: Under view chat message mode, user can edit message by entering message id and new message content if the user is the message sender. If the user is not the sender, the action will be vacated.
- (19) DeleteMessage: Under view chat message mode, user can delete message by entering message id if the user is the message sender. If the user is not the sender, the action will be vacated.

### 3. Index

- (1) USER\_LIST table: list\_id
- (2) USR table: login, password, contact\_list, block\_list\_list
- (3) USER\_LIST\_CONTAINS table: list\_member
- (4) CHAT table: chat\_id, init\_sender
- (5) CHAT\_LIST: chat\_id, member
- (6) MESSAGE table: msg\_id, msg\_timestamp, sender\_login, chat\_id

## III. Findings and Workload

Except the main functionalities to implement, there are some details that we need to take into considerations like console interface construction, checking valid users, time stamp format and quote manipulation in string. To accelerate the speed of queries, it is also necessary to create index based on function implementations.

The whole project is split evenly: Wei Wang: 50%, Rebecca Hom: 50%