CSIT5100 Assignment 2

Objectives:

Multi-User Calendar Manager (MCM) system is an appointment-scheduling tool which supports multiple users and groups. It works based on server-client mode. It aims to allow users to manage their appointments and groups and provide a friendly GUI interface. It is targeted to both system administrator who are responsible for the database, and system users who utilize the system to manage their appointments and groups info.

Main Functionalities:

1. Manage valid users (register/unregister users)
2. Manage appointments (insert/delete appointments)
3. Manage groups (create/remove/join/withdraw groups)
4. Valid users logon/logoff the system
5. View the appointments info
6. View the groups info

Design Issue:

- Communication

MCM system works based on server-client mode, which consists of a console-based server and a GUI-based client. We need to build communication between server and clients. This feature could better support multiple users and groups.

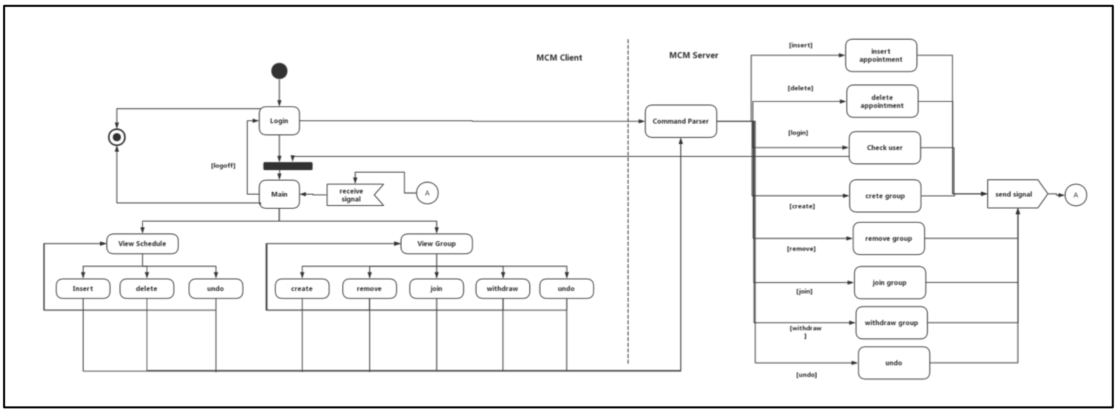
- GUI interface

MCM client is graphically interfaced. The GUI interface consists of various GUI components such as text field, button, table, check box, menu and etc. GUI interface could be more adapter to the wide range of user.

- Server response

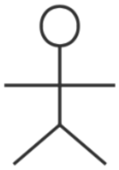
MCM server is designed to manage the data including user info, appointment info and group info. The server supports the core functionality. It is console-based which could be easier for communication between server and clients.

Activity Diagram



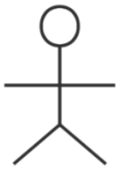
(really sorry that it is too long, it may require zoom-in operation, but it is clearly designed)

Major Actors of the system



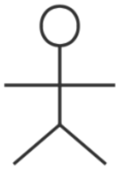
Name: System Administrator

System administrator is a person who manage the system. The person may work in an organization or company. He is responsible for the permission of valid users and the consistency of the system data. He needs to register/unregister valid users and manage the groups as well as appointments.



Name: System User

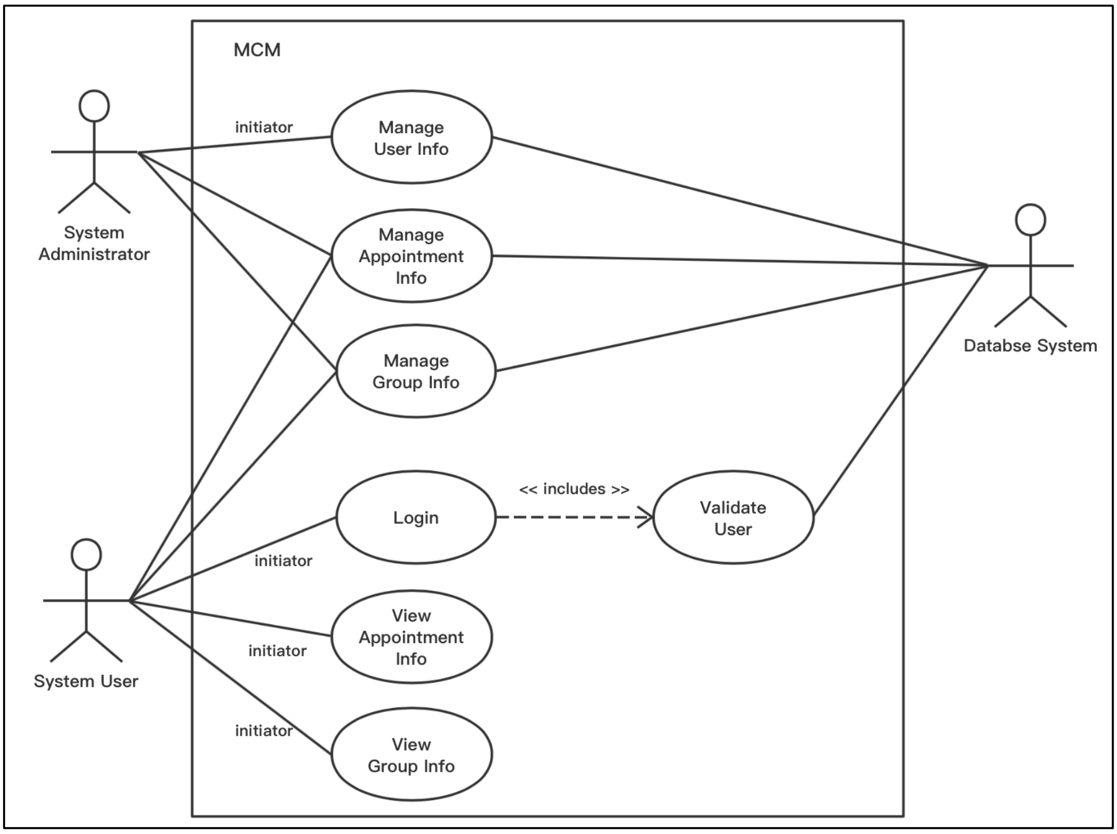
System user is a person who utilize the system. The person may work within an organization or work individually. The responsibilities is to manage his appointments or even group info. He needs the operation of managing appointments (i.e. insert/delete) and managing groups (i.e. create/remove/join/withdraw)



Name: Database system

The database system includes data records, i.e. appointment records, group records, partcipation records, user records. The system is in a file format, which is generated by MCM. It is responsible to manage appointment info, group info, participation info and user info. It needs to save system output and also needs to be loaded during intialization.

Use Case Diagram



Use Case Description

1. Login

Purpose: allow system users to enter MCM system

Actors: System user

Flow of Events:

1) System user invokes the use case by beginning MCM client.

2) The system would pop up the GUI interface with text fields requiring user-name.

3) User type in his user-name and MCM server address and click enter button.

4) Use case terminates

2. Validate User

Purpose: validate the logging-in user

Actors: Database System

Flow of Events:

1) Database System invokes the use case after User’ s Login use case

2) MCM has read the user’ s user-name, and it load the user info from Database

3) MCM compare the user-name with the user info from Database

4) If match, then the user is valid, else it is not valid.

5) MCM system gives back the validation info to GUI

6) Use case terminates.

3. View Appointment Info

Purpose: allow users to see the details of appointments

Actors: System Users

Pre-conditions: Users are valid and they enter the MCM system successfully.

Flow of Events:

1) User invokes the use case by clicking the appointment-view check box

2) The MCM system would retrieve the appointment info

3) The system then shows the appointment info in the GUI given to users, i.e. starting time, end time, description, etc.

4) Use case terminates

4. View Group Info

Purpose: allow users to see the details of groups

Actors: System Users

Pre-conditions: Users are valid and they enter the MCM system successfully.

Flow of Events:

1) User invokes the use case by clicking the group-view check box

2) The MCM system would retrieve the group info

3) The system then shows the group info in the GUI given to users, i.e. group name, creator, etc.

4) Use case terminates

5. Manage User Info

Purpose: allow administrators to register or unregister valid users

Actors: System Administrator, Database

Flow of Events: (*register users*)

1) Administrator invokes the use case by beginning the MCM system
2) Administrator gives the commands to MCM server, in order to register the user
3) The MCM system would parse the commands to get the user info
4) If user info is not duplicated, MCM would insert it into Database User schema so that the user is registered; else MCM would not insert it into Database.
5) Use case terminates

Alternative Path: (*unregister users*)

1) Administrator invokes the use case by beginning the MCM system
2) Administrator gives the commands to MCM server, in order to unregister the user
3) The MCM system would parse the commands to get the user info
4) If user info is existed, MCM would delete it from Database User schema so that the user is unregistered; else MCM would not delete it.
5) Use case terminates

6. Manage Appointment Info

Purpose: allow administrators or users to manage the appointment info

Actors: System User/Administrator and Database

Pre-conditions:

1) Users are valid and they enter the MCM system successfully.
2) For inserting appointments, the appointment should be valid, i.e. start time is earlier than end time
3) For deleting appointments, the appointment list should include at least one appointment. Users can only delete the appointments belong to them, while administrators can delete any appointments based on commands

Flow of Events: (*insert appointments*)

1) User invokes the use case by entering the MCM system

2) User can type in the details of a new appointment and click the insert button

3) MCM system read the new appointment info

4) If the appointment is valid and not duplicated, MCM would insert it into Database Appointment schema; else, MCM would reject.

5) Use case terminates

Alternative Path: (*delete appointments*)

1) User invokes the use case by entering the MCM system

2) User can select one appointment from GUI and click the delete button

3) MCM system get the selected appointment info

4) The appointment is existed and it belongs to the corresponding user, so MCM would delete it from Database Appointment schema.

5) Use case terminates

7. Manage Group Info

Purpose: allow administrators or users to manage the group info

Actors: System User/Administrator and Database

Pre-conditions:

1) Users are valid and they enter the MCM system successfully.

2) For removing groups, the group list should include at least one group. Users can only remove the groups whose creator are themselves, while administrators can remove any.

3) For joining groups, the group list should include at least one group. Users can only join the groups whose creator are not themselves.

4) For withdrawing groups, the group list should include at least one group. Users can only withdraw the groups that they have already joined.

Flow of Events: (*create groups*)

1) User invokes the use case by entering the MCM system

2) User can type in the details of a new group and click the create button

3) MCM system read the new group info

4) If the group is not duplicated, MCM would insert it into Database Group schema; else, MCM would reject.

5) Use case terminates

Alternative Path:

There are three alternative paths following:

Alternative Path: (*remove groups*)

1) User invokes the use case by entering the MCM system
2) User can select one group from GUI and click the remove button
3) MCM system get the selected group info
4) If the group belongs to the corresponding user, MCM would delete it from Database Group schema; else, MCM would reject.
5) Use case terminates

Alternative Path: (*join groups*)

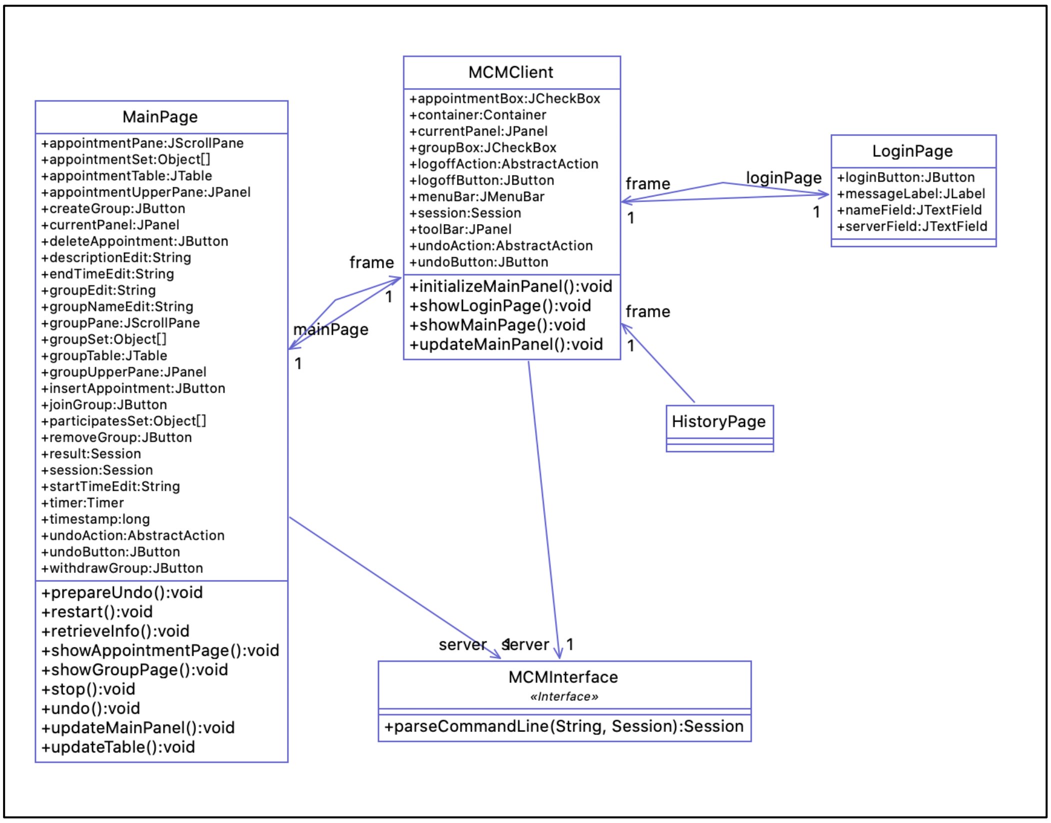
1) User invokes the use case by entering the MCM system
2) User can select one group from GUI and click the join button
3) MCM system get the selected group info
4) If the group does not belong to the corresponding user, MCM would insert the participation relationship into Database Participation schema; else, MCM would reject.
5) Use case terminates

Alternative Path: (*withdraw groups*)

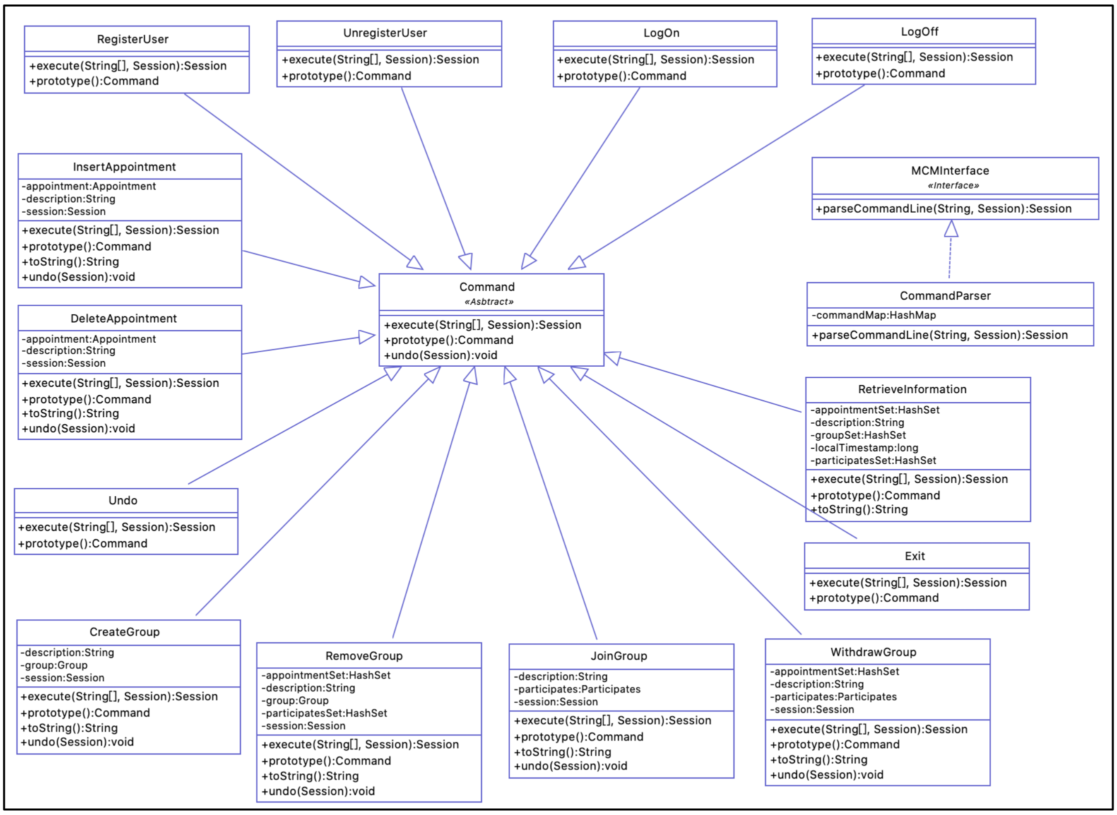
1) User invokes the use case by entering the MCM system
2) User can select one group from GUI and click the withdraw button
3) MCM system get the selected group info
4) If the group does not belong to the corresponding user, and the user has joined the group. MCM would delete the participation relationship from Database Participation schema; else, MCM would reject.
5) Use case terminates

Class Diagram

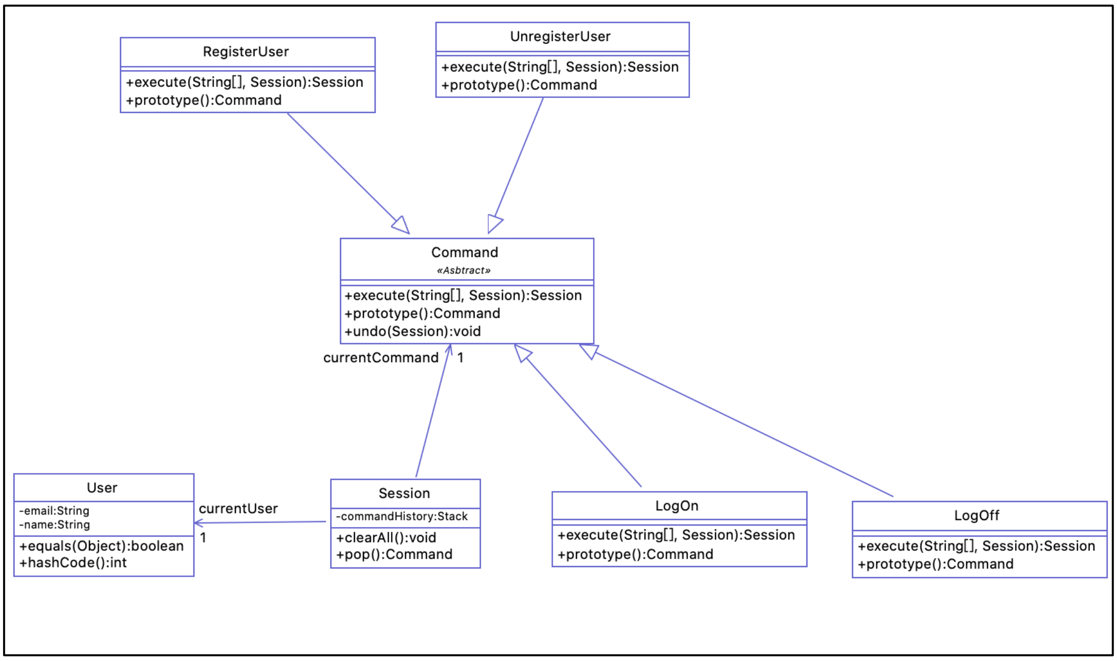
MCM Client CUI design:



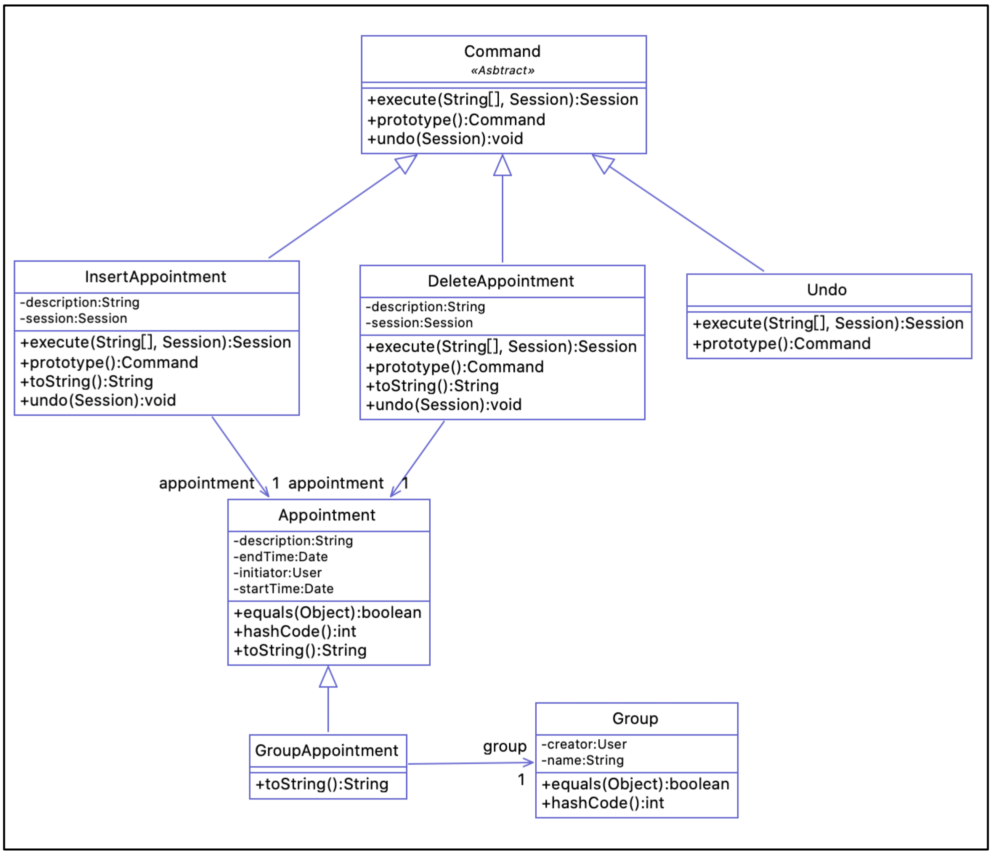
MCM Server design:



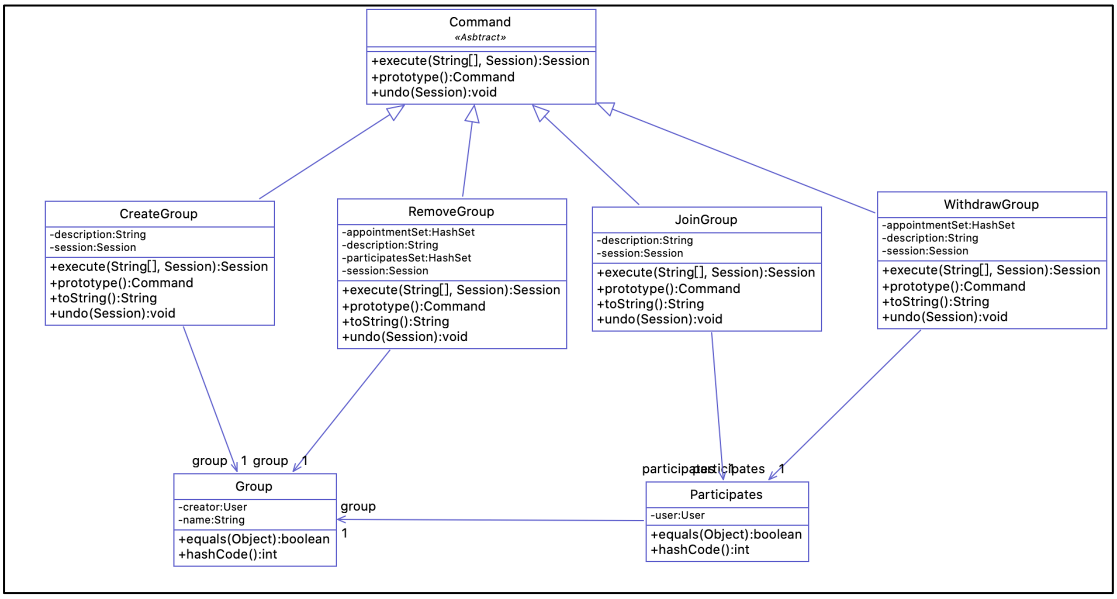
Manage User info:



Manage Appointment info:



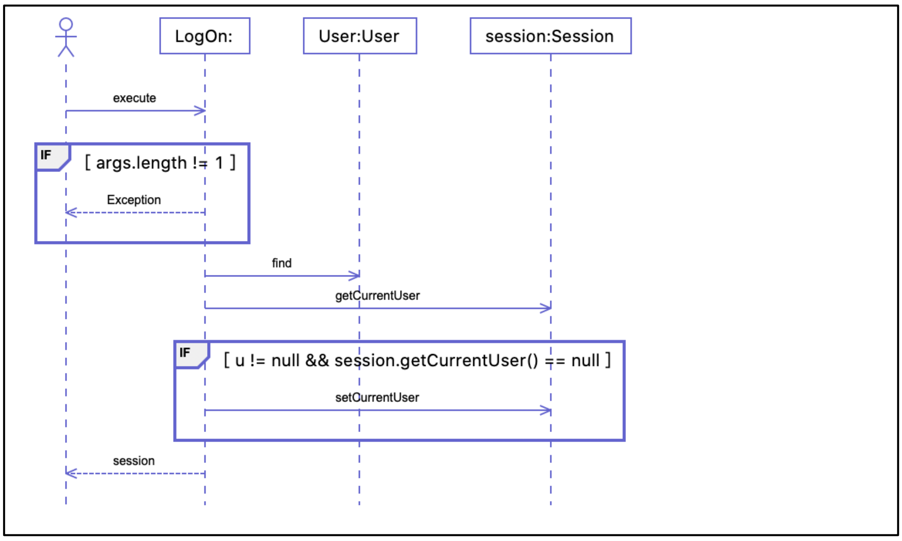
Manage Group info:



Sequence Diagram

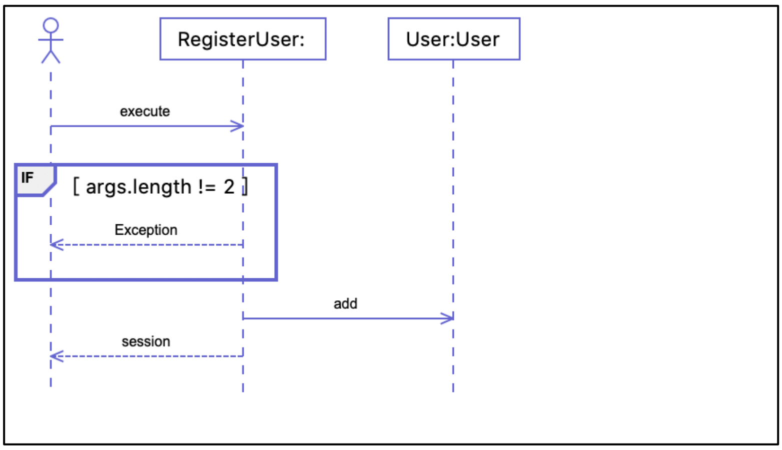
[Though some of them may need zoom-in operation, they are all well designed.]

User login:

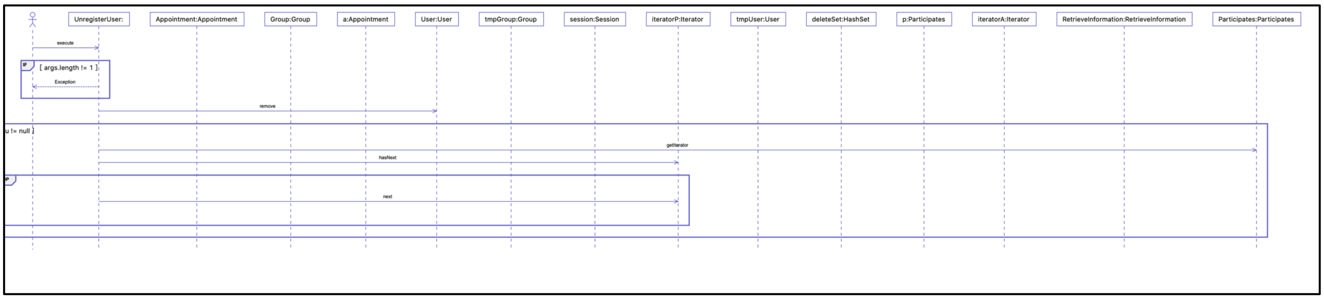


Manage User Info:

Register user:

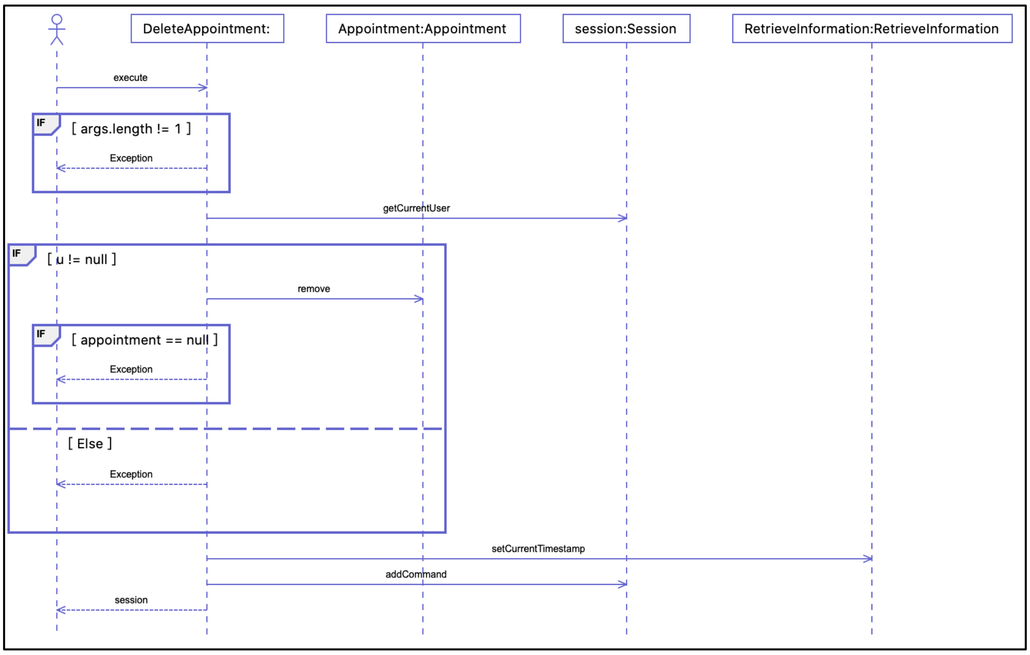


Unregister user:

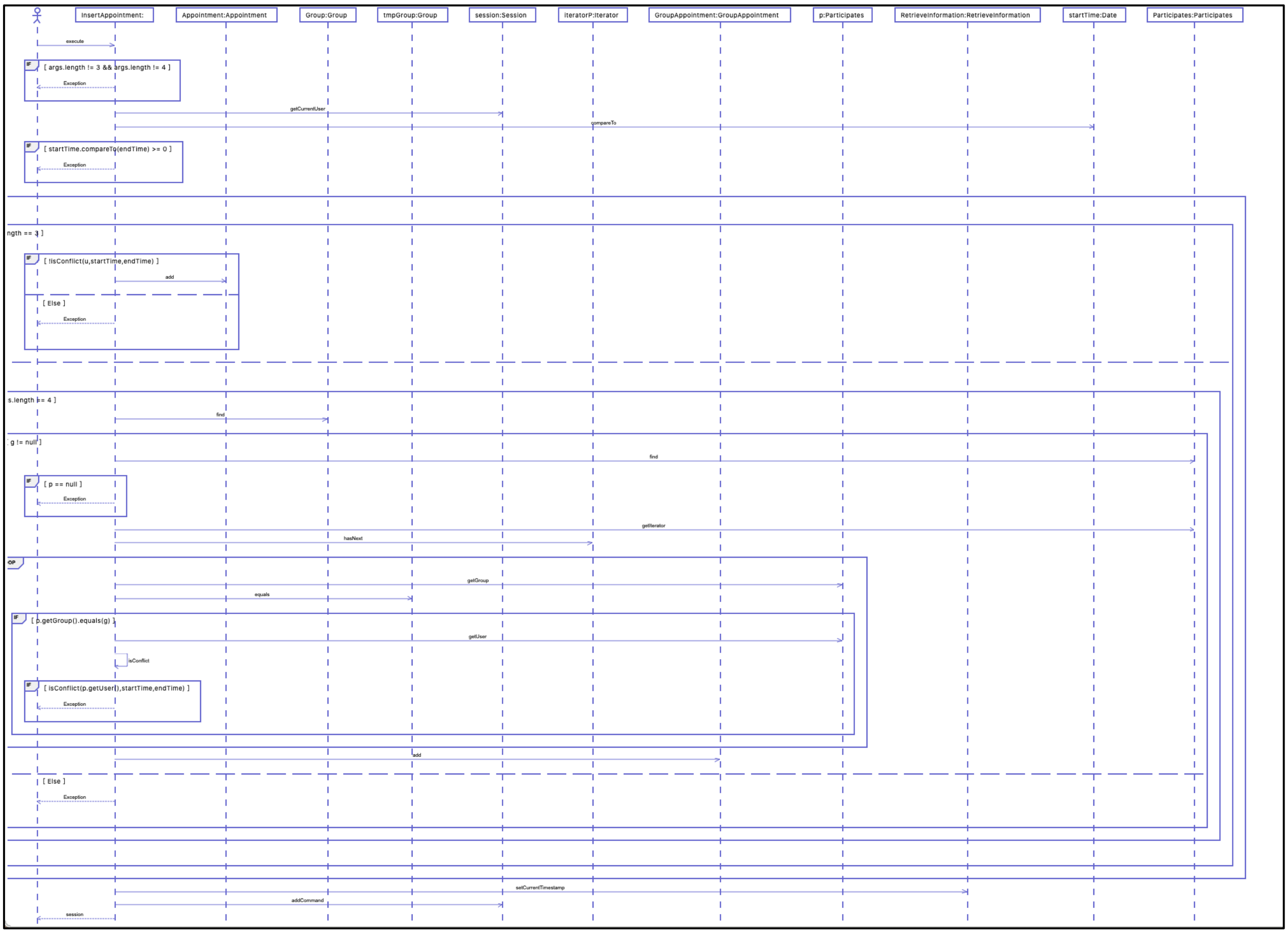


Manage appointment info:

Delete appointment:

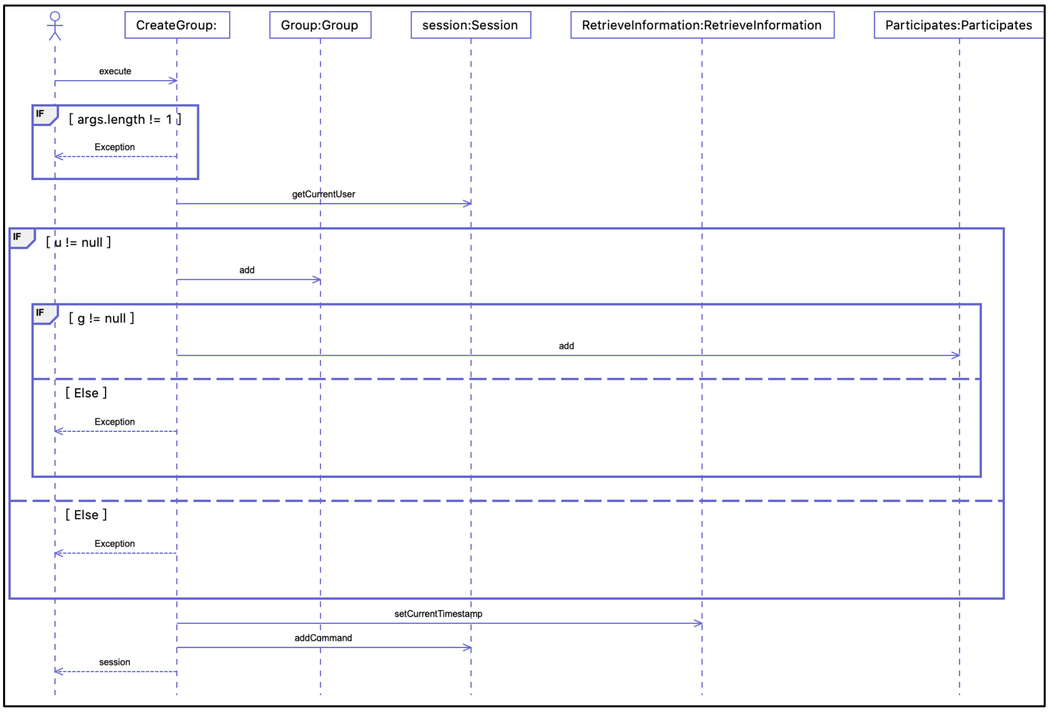


Insert appointment:

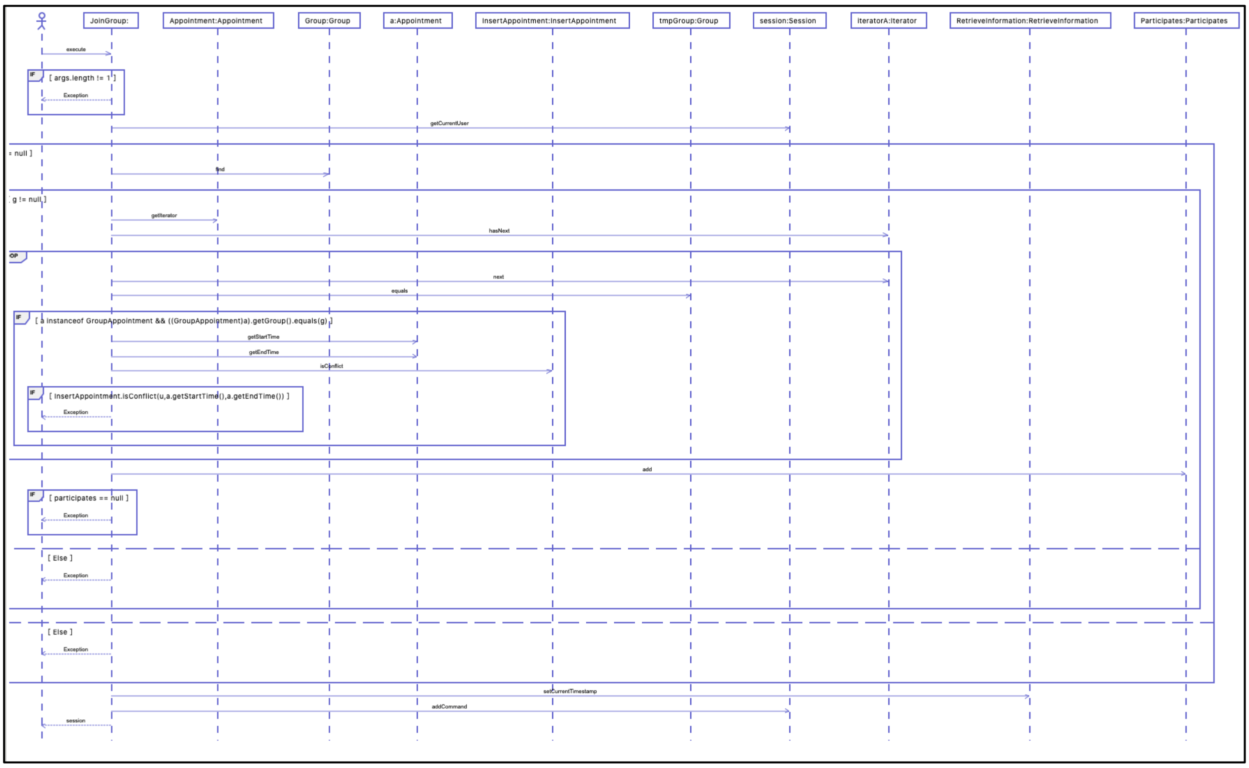


Mange group info:

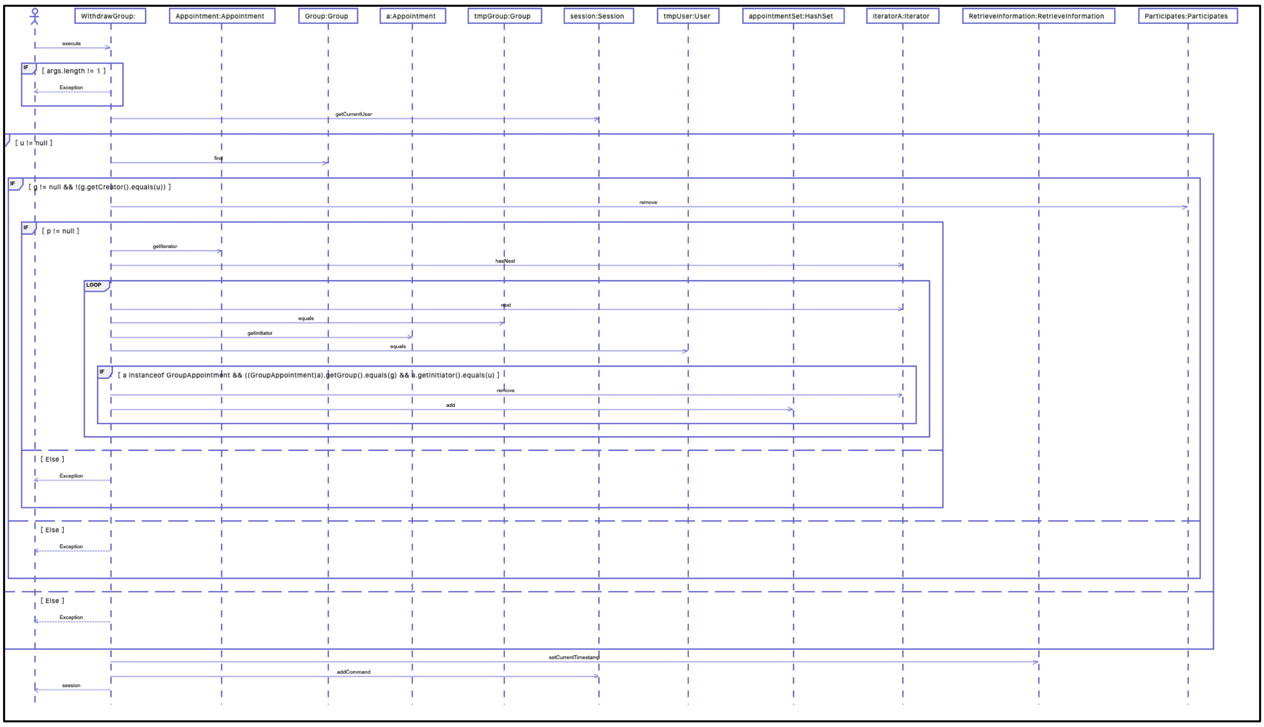
Create group:



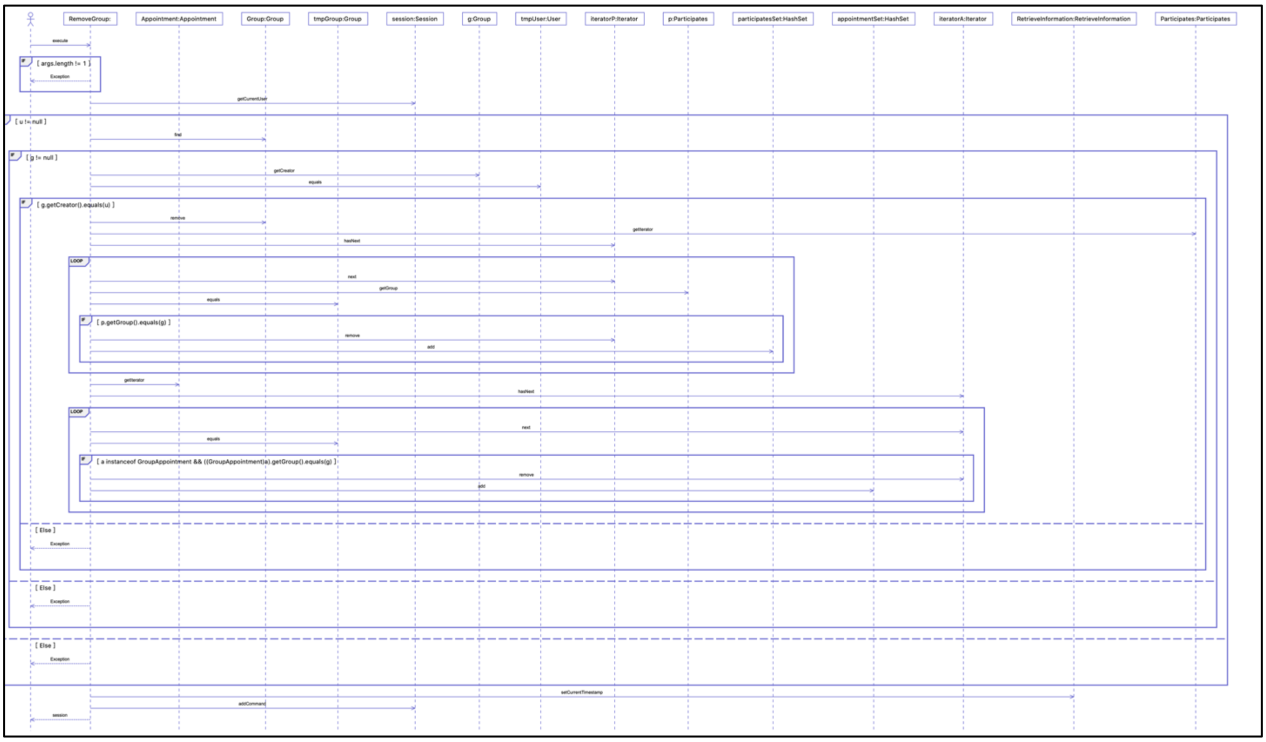
Join group:



Withdraw group:

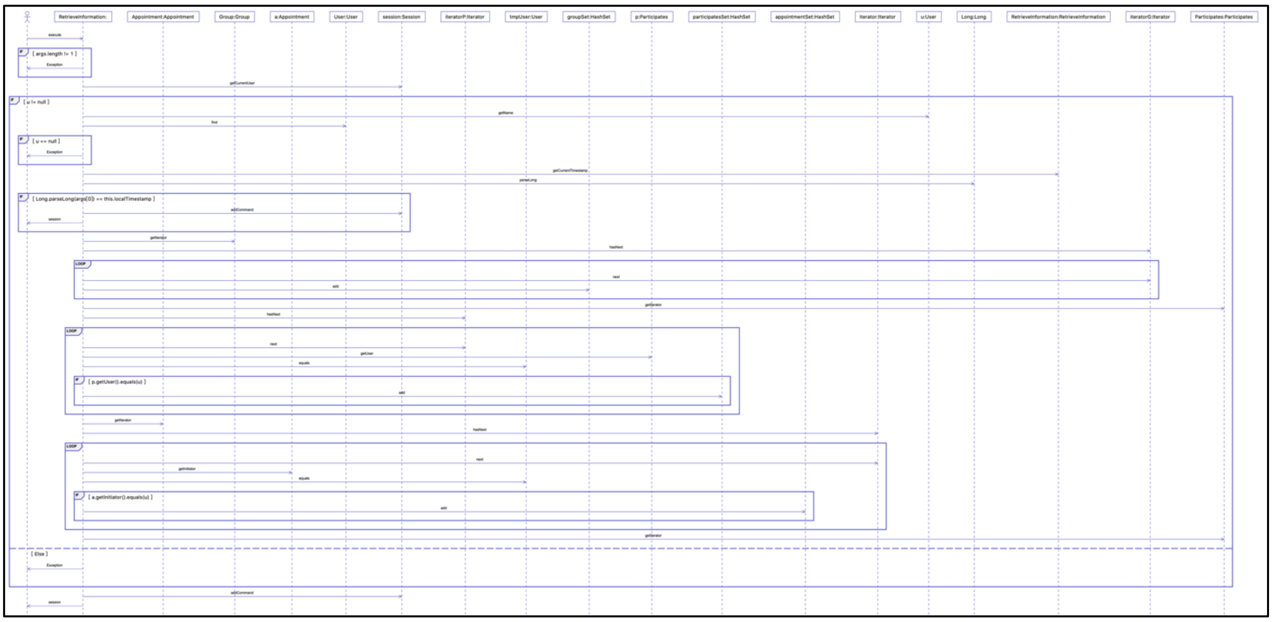


Remove group:



View appointment/group info:

Retrieve info:



Limitation and Future Improvement:

1. This MCM system supports multiple users, but it only needs user names to log in. In order words, other people instead of sam can also login the MCM system as “sam”, they can see and even change the schedule and group info of sam. This is very **unsecure**.

   Suggestion: we should verify the valid users by both their user name and corresponding **password**, and use **hash method** to hash the password. MCM system database only stores the hash string of passwords, so that even administrators have no idea what password is. But the valid user can type in his password in the MCM Client, which would hash the password string and send it to MCM server to compare. In this way, MCM system would be more secure than before.

2. **Users who create the group cannot see or manage the group member**. In reality, creators may need to validate his group member and even kick-out some members because of some reasons. But in MCM system, users can join any groups on their wish, as well as leave any groups freely.

   Suggestion: The group creator should have the right to view and manage the group members, which means, he can invite other users into his group or kick some members out of the group. Also, those who want to join the group should first hand their proposals to the creator and they would join the group only after the permission of creator.