

Team:

Wu, Qingjun

Sham Prasad PS

Padhy, Umakant

Title: WeChat like system

Summary: WeChat is an Instant Messenger like WhatsApp. It is a combination of Instant Message and Social Network. What we are trying to achieve here is a basic chat system where user can perform basic

Note: This system will include both UI and backend. The backend is basically being implemented for maintaining chat history and backup. We will be using a **SQL DB** (storage DB) to store the chat history. Also the DB would store static data like for maintaining user contact details, group members, etc.

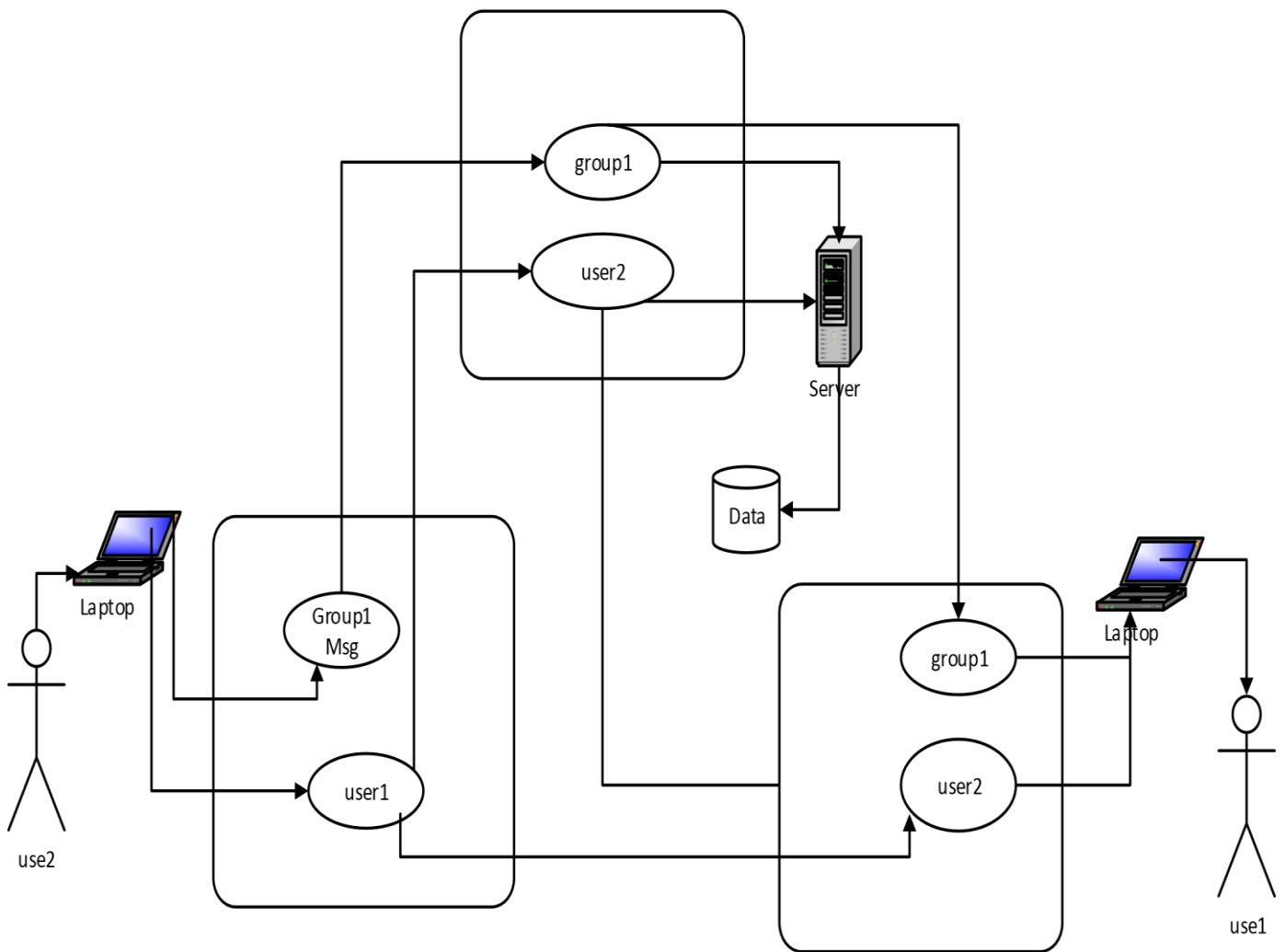
Functionality:

1. User Registration: register a new user;
2. User Login/Log out;
3. Add/Remove/block/unblock contacts (friends);
4. Send/Receive Message;
5. Create/Delete/Join/Quit group chatting;
6. Send/Receive message in group chat;

Key Technologies/Components of the system:

1. Message/post type: text, audio, video;
2. Backend Database will maintain contact list of user/group;
3. System will maintain IP/Port address of the user;
4. When user sends a message, a message will be sent to the server – this is the way the system keeps track of chat history;
5. Message is ordered: when a user is sending two messages to another, these two message should arrive in order;

Architecture Diagram:



Use Case Documentation:

Use Case ID	UC-01		
Use Case Name	Log In		
Description	User should be able to login		
Actor	User		
Pre-Condition	User should be registered		
Post-Condition	User should be able to view screen		
Frequency of Use	Multiple times by user		
Flow of Events		Actor Action	System Response
	1.	User clicks connect button	system loads chat window
	2.		
	3.		
	4.		
Variations	user provides incorrect password throw error message		
Notes and Issues			
Developer Notes			

Use Case ID	UC-02		
Use Case Name	Log out		
Description	User should be able to logout		
Actor	User		
Pre-Condition	User should be logged in User may be chatting User may be logged in		
Post-Condition	user will be shown login screen back		
Frequency of Use	Multiple times by user		
Flow of Events		Actor Action	System Response
	1.	User clicks logout button	system logs out and loads logon window
	2.		

	3.		
	4.		
Variations	User clicks logout button throw error message if any		
Notes and Issues			
Developer Notes			

Use Case ID	UC-03		
Use Case Name	Search Contact List		
Description	User should be able to lookup someone		
Actor	User		
Pre-Condition	User must be logged in User selects search User types name to search		
Post-Condition	Show list of contacts matching the name		
Frequency of Use	Multiple times by user		
Flow of Events		Actor Action	System Response
	1.	User logs in User searches in contact list User enters a name	System shows the list of contacts
	2.		
	3.		
	4.		
Variations	User is not logged in The contact is not present is in the contact list		
Notes and Issues			
Developer Notes			

Use Case ID	UC-04
Use Case Name	Add contact
Description	add a new user
Actor	User
Pre-Condition	User should be logged in User selects on Contacts button User selects on add new user

Post-Condition	Open a new window for adding a contact Save the contact		
Frequency of Use	Multiple times by user		
Flow of Events		Actor Action	System Response
	1.	User logs in	System should open the edit contacts page
	2.	User navigates to contacts window	
	3.	User selects add new user option	
	4.		
Variations		User may not be logged in User could not provide mandatory fields	System throws error
Notes and Issues			
Developer Notes			

Use Case ID	UC-05		
Use Case Name	Chatting		
Description	User should be able to chat with a person or a group		
Actor	User		
Pre-Condition	User selects buddy list or group User types text User hits send		
Post-Condition	Message should be delivered to correct client		
Frequency of Use	Multiple times by user		
Flow of Events		Actor Action	System Response
	1.	User should be logged in There should be some friends in contacts Select any contact	System should deliver message to client
	2.	Type message	

	3.		
	4.		
Variations	User not logged in		
Notes and Issues			
Developer Notes			

Use Case ID	UC-06		
Use Case Name	Create a group		
Description	User should be able to create a group of contacts		
Actor	User		
Pre-Condition	User selects create group		
	User chooses contacts to be added		
Post-Condition	Group with the contacts should be created		
Frequency of Use	Multiple times by user		
Flow of Events		Actor Action	System Response
	1	User is logged in User chooses create group User chooses some contacts to be added User selects create group	System should create a group
	2	Type message	
	3		
	4		
Variations	User not logged in User not able to find contacts		
Notes and Issues			
Developer Notes			

Use Case ID	UC-07
Use Case Name	Add user to group
Description	User should be able to add members to group that user created
Actor	User

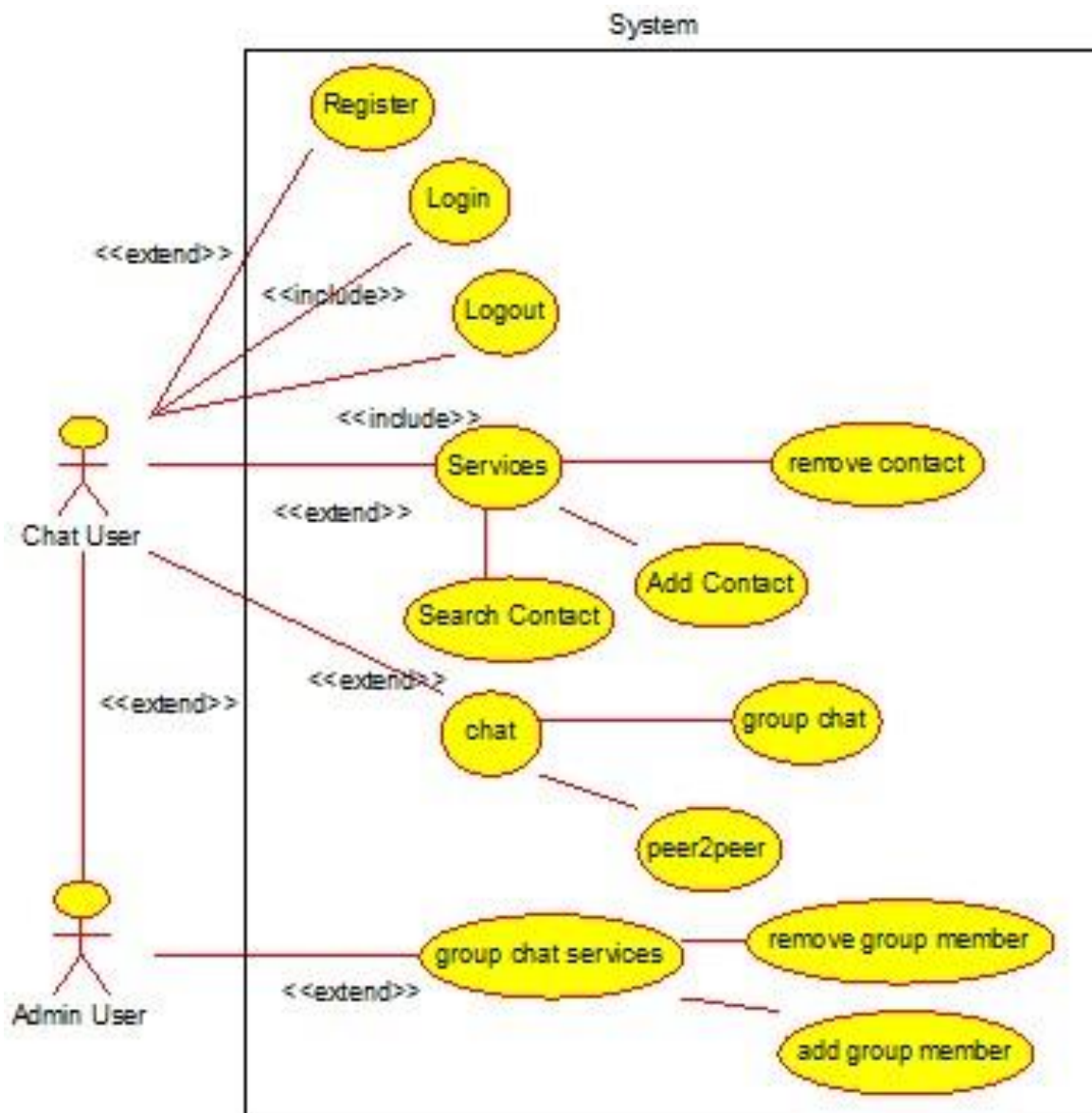
Pre-Condition	User selects buddy list or group User adds members of the group		
Post-Condition	Members should be added to the group		
Frequency of Use	Multiple times by user		
Flow of Events		Actor Action	System Response
	1	User should be logged in There should be some friends in contacts Select any contact	System should add those members to group
	2	add members to group	
	3		
	4		
Variations	User not logged in User is not the one who created the group		
Notes and Issues			
Developer Notes			

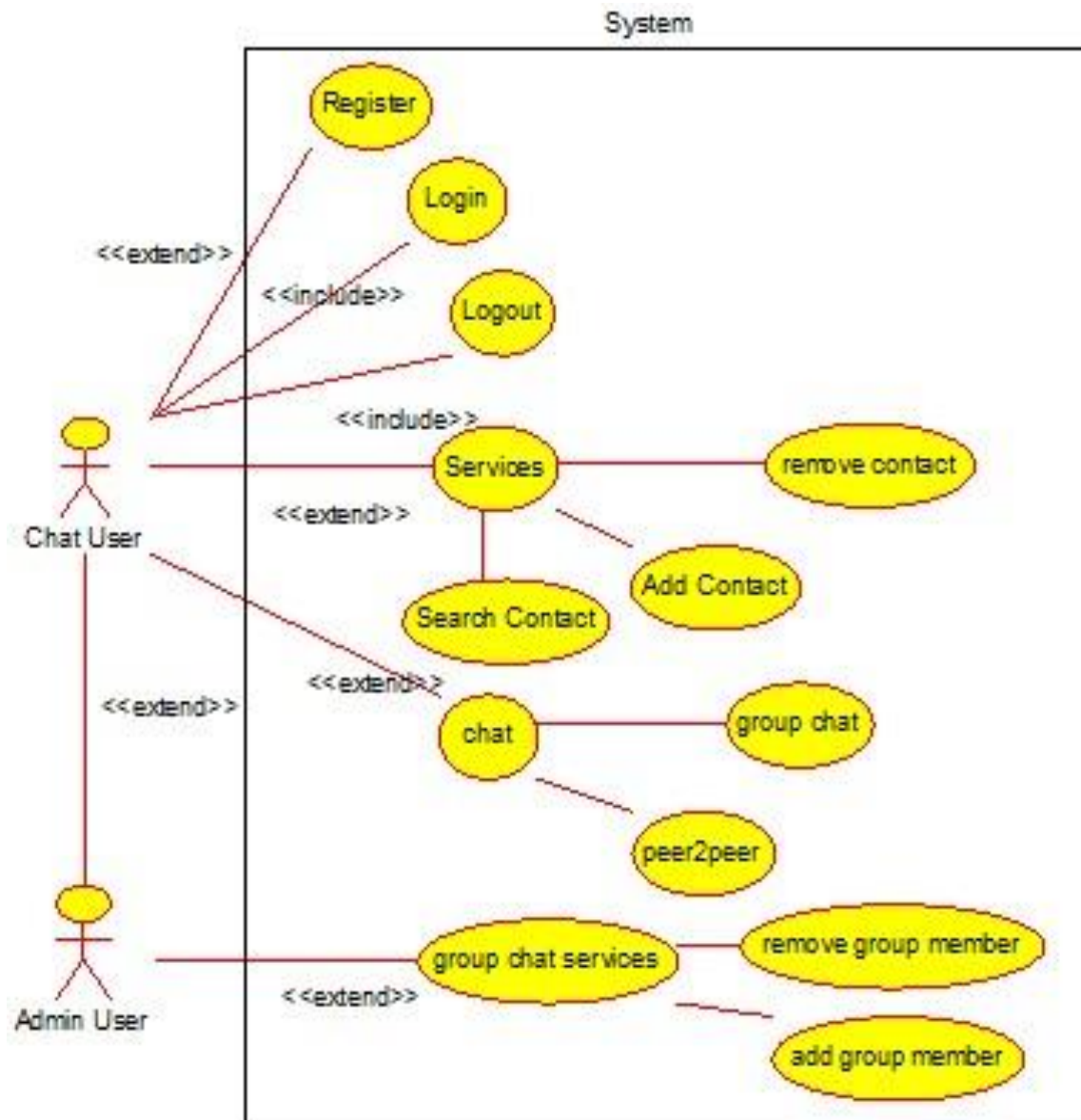
Use Case ID	UC-08		
Use Case Name	remove user		
Description	User should be able to remove members to group that user created		
Actor	User		
Pre-Condition	User is logged in User selects buddy list or group User removes members of the group		
Post-Condition	Members should be removed from the group		
Frequency of Use	Multiple times by user		
Flow of Events		Actor Action	System Response

	1	User should be logged in User selects a group User removes members from it	System should remove those members from group
	2		
	3		
	4		

Variations	User not logged in User is not the one who created the group
Notes and Issues	
Developer Notes	

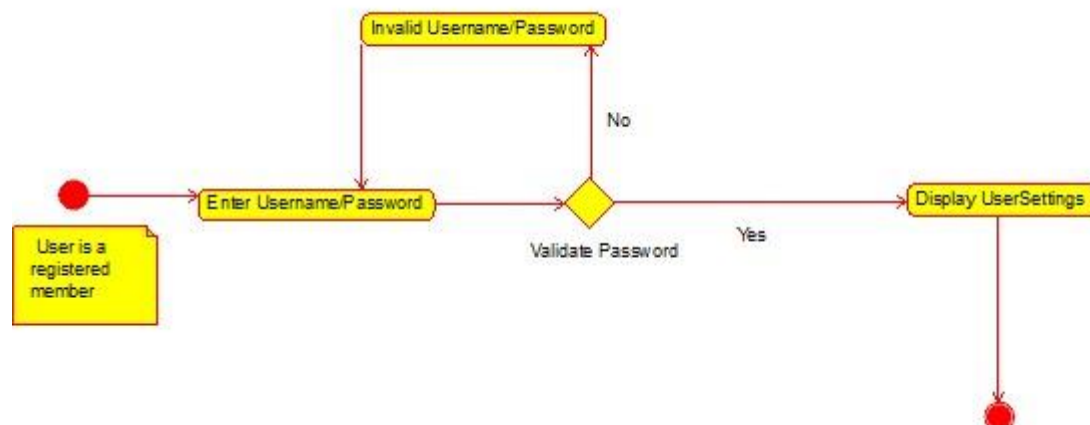
Use Case Diagram



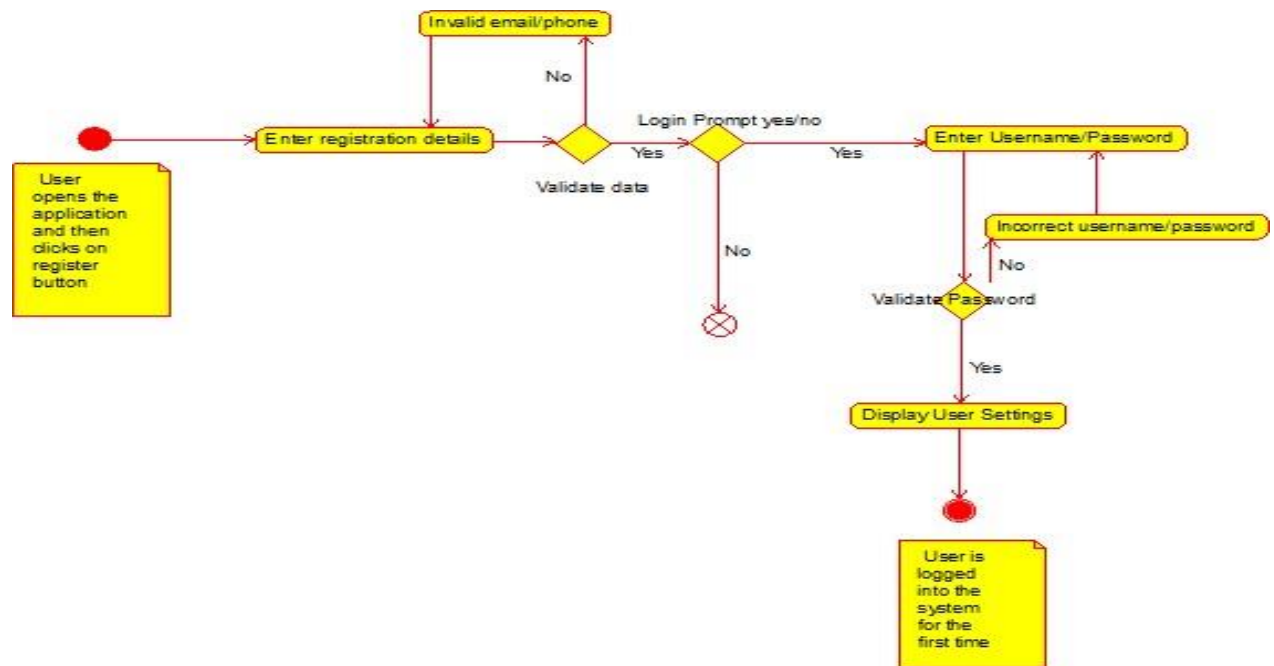


Activity Diagrams:

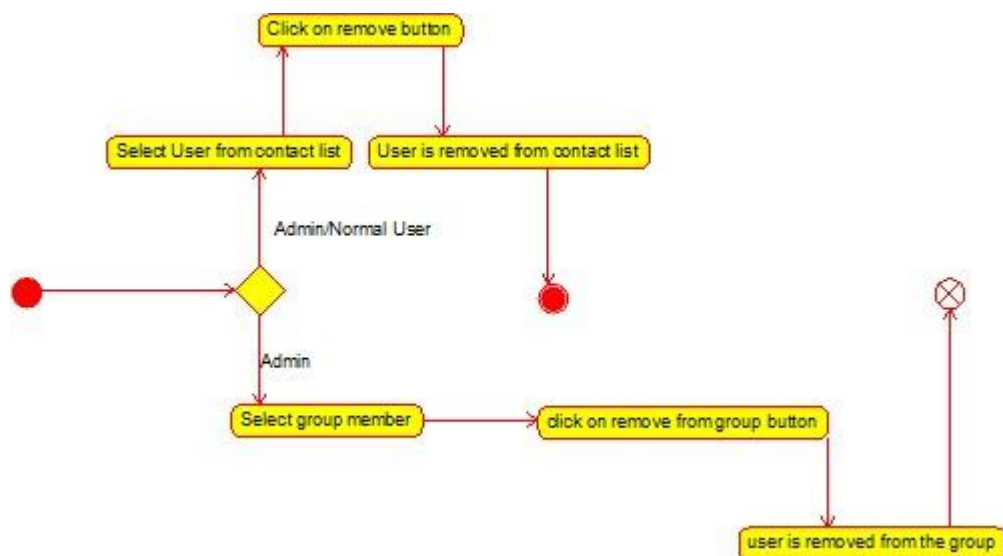
Login Activity:



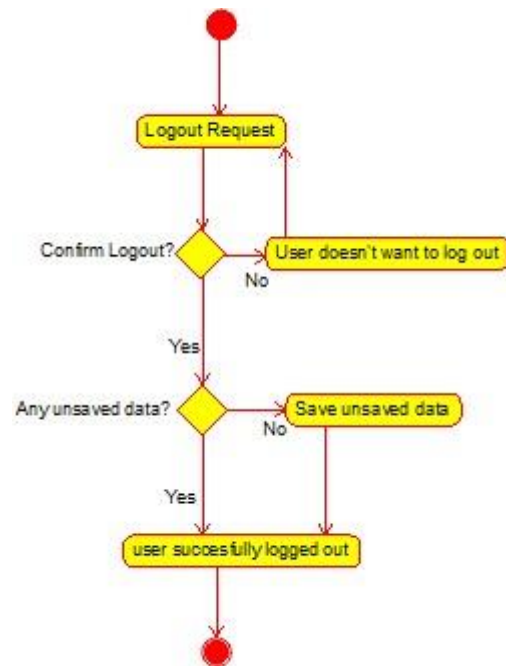
Register Activity:



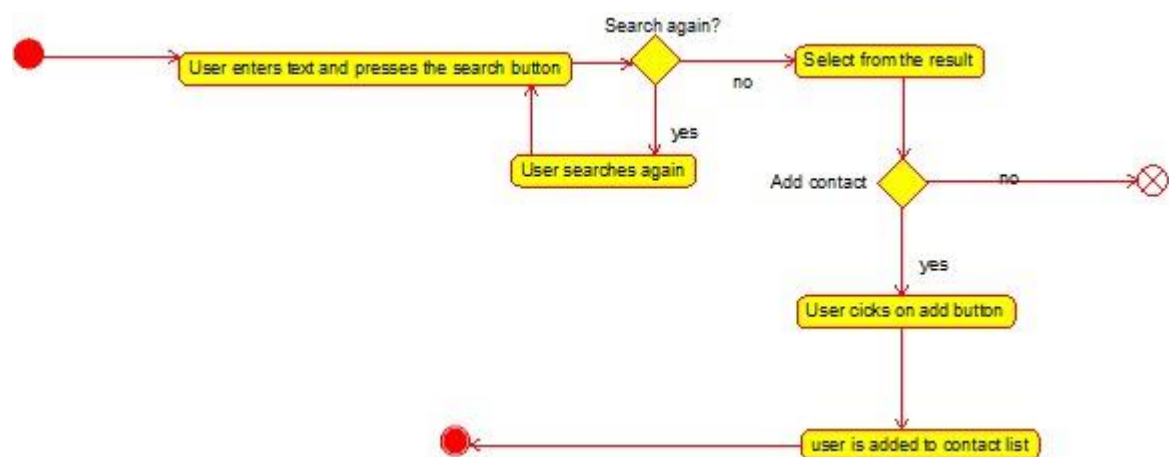
Remove User/group user from Contact List:



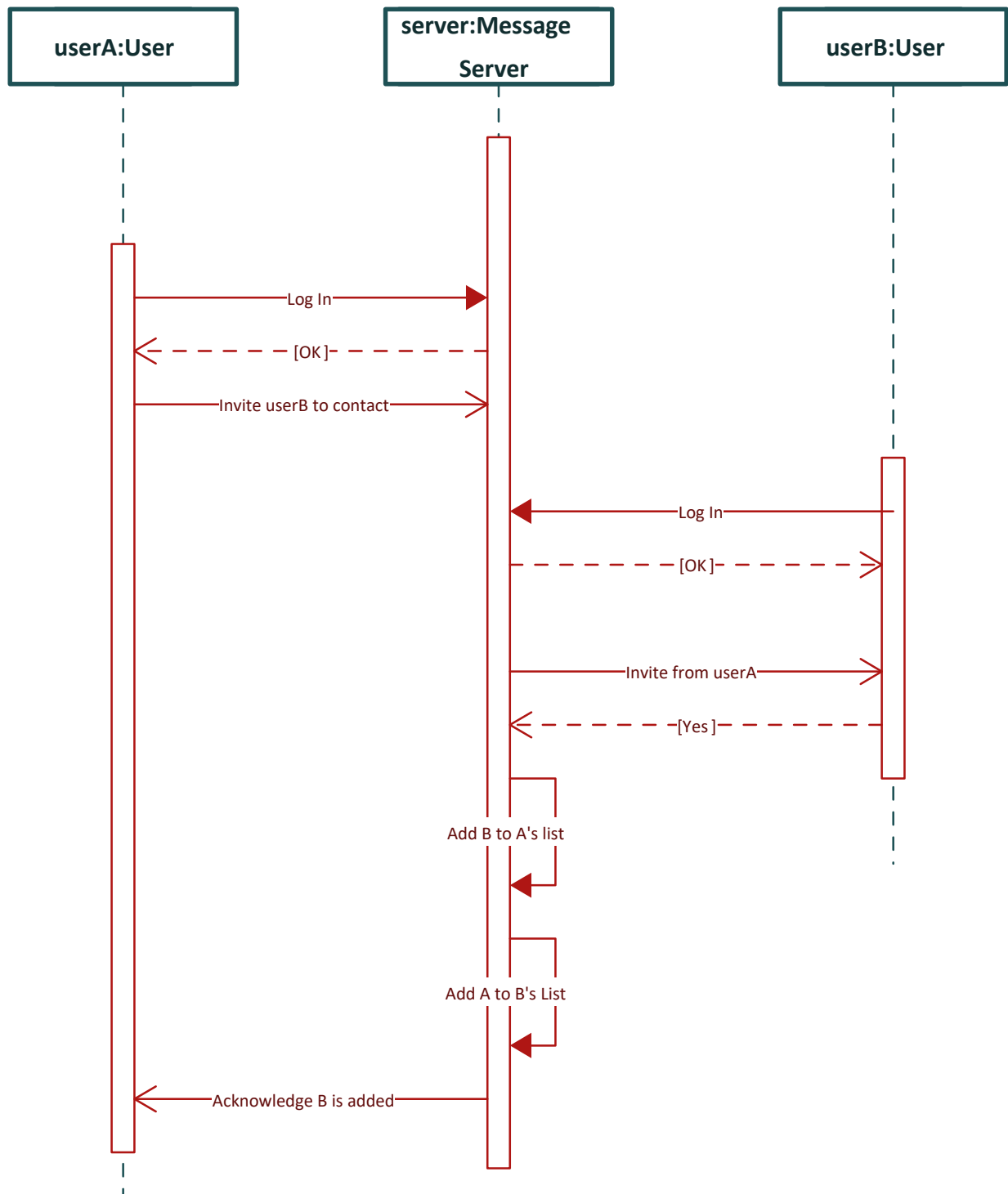
Logout Activity:



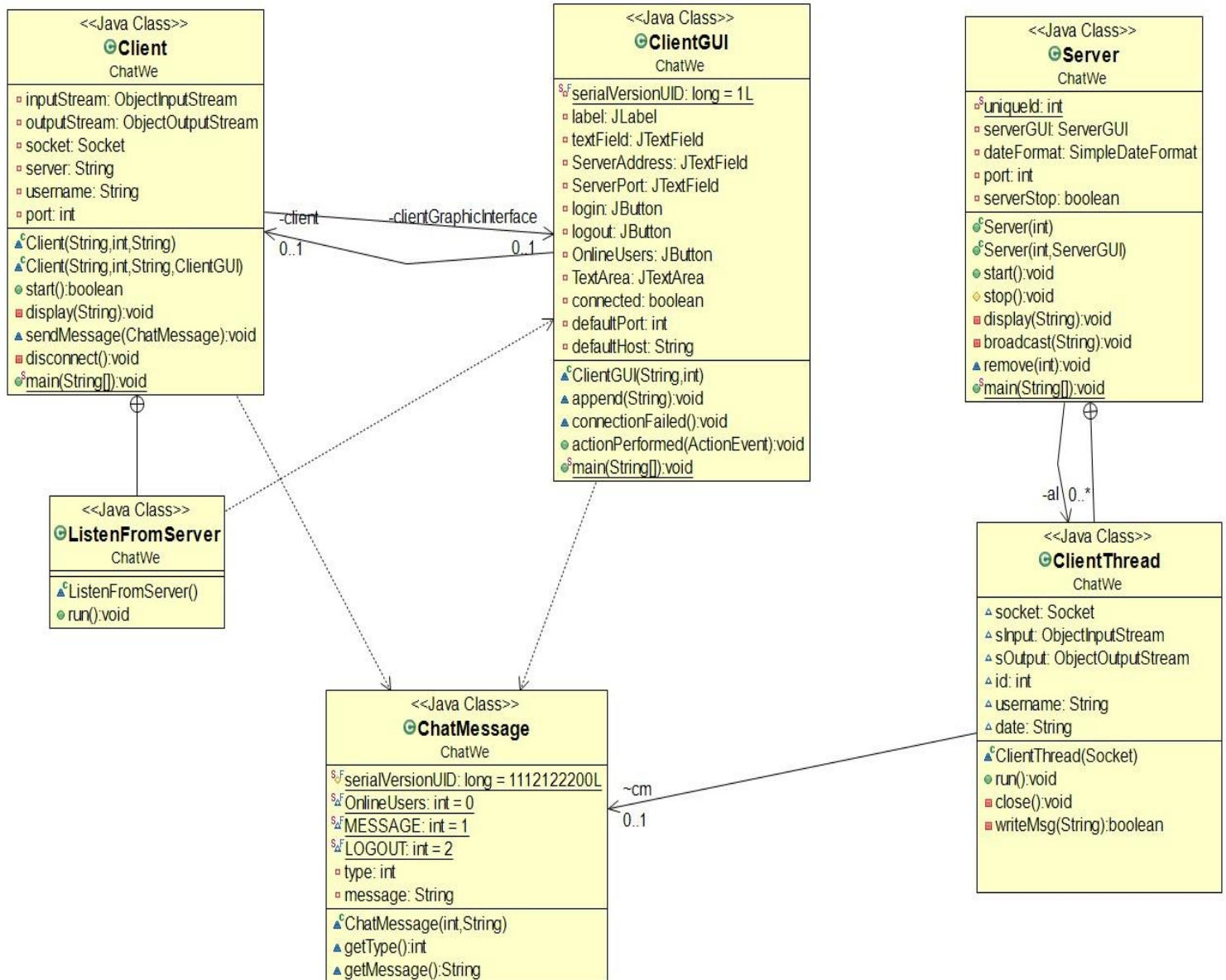
Search Activity:



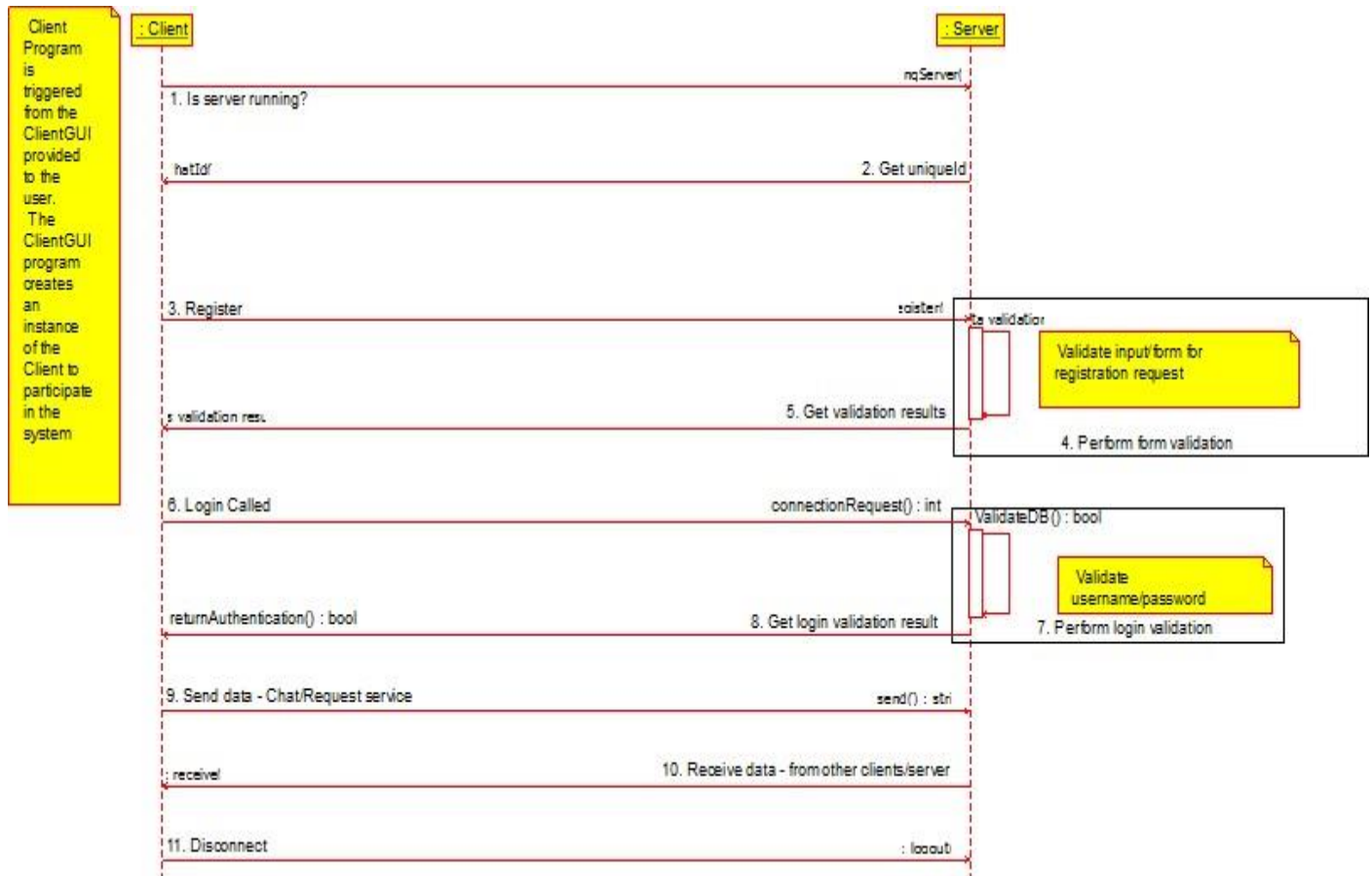
Sequence Diagram:



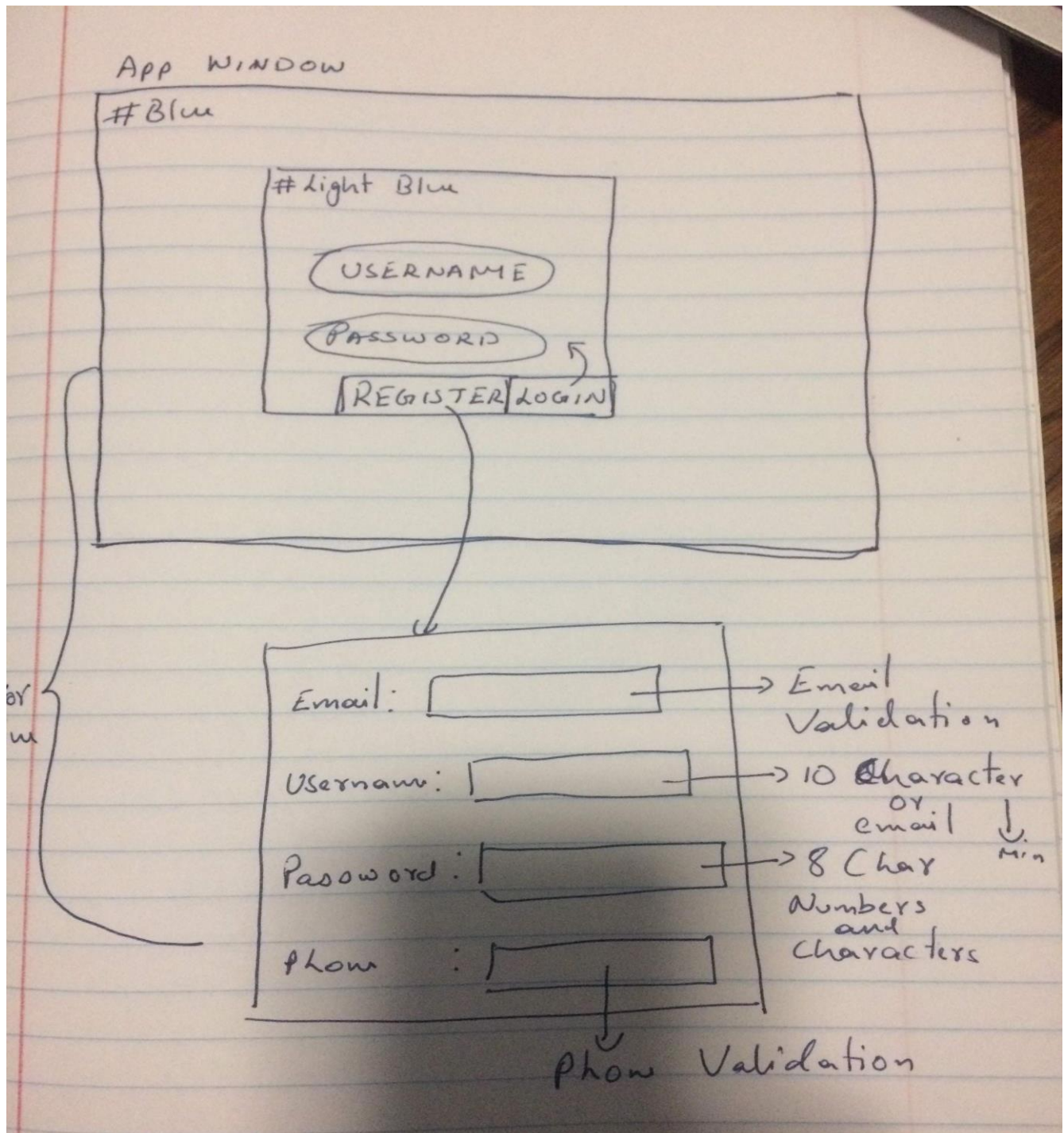
Class diagram:

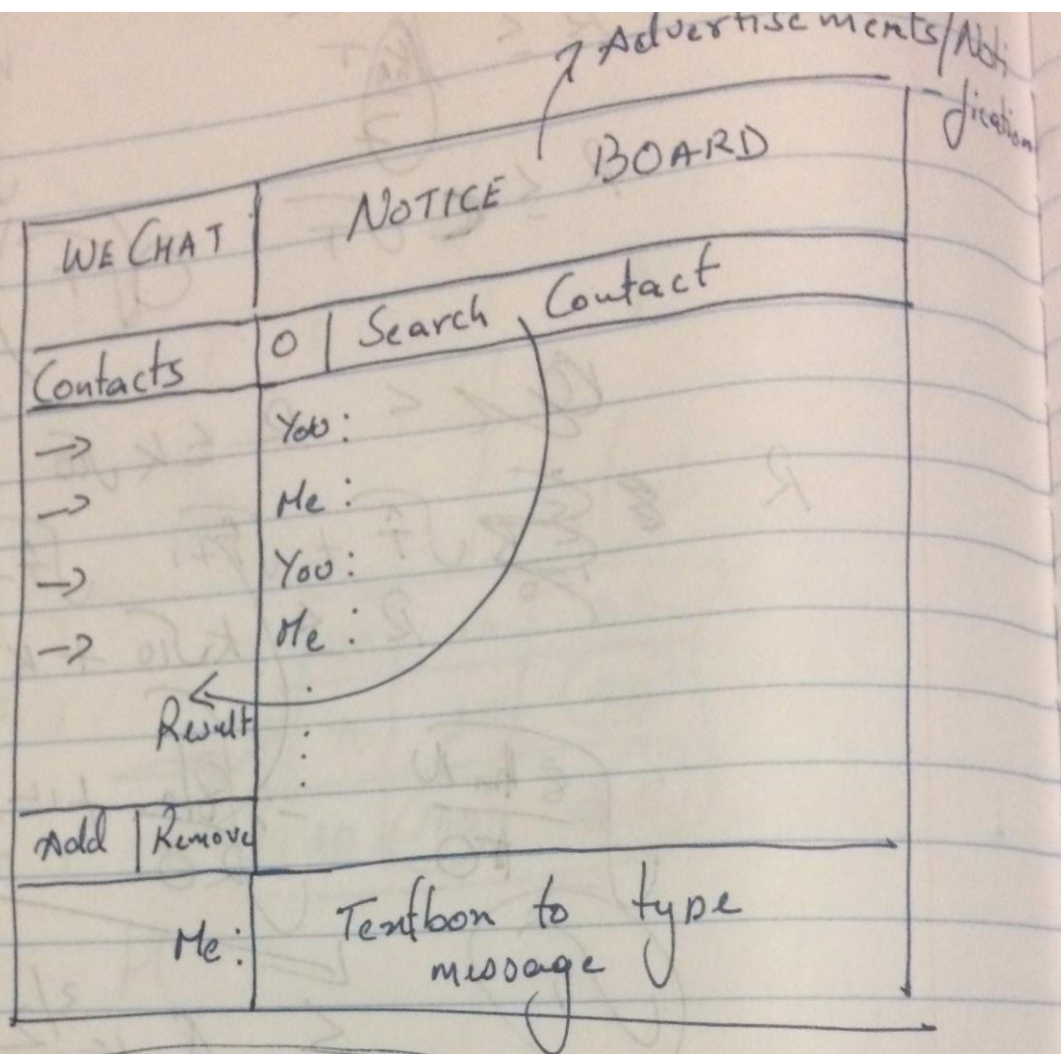


Client Server Interaction Sequence:



Basic UI Mock-up :





←
for
Tracking
and
Maintenance

SERVER GUI

Time Stamp	Data Logged
...	User1: ...
	User2: ...

