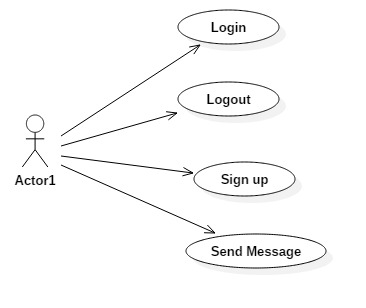
**SiliconValley Chatting Program**

Name: Kyubin Kyoung

1. **Introduction**

This is a project making a chatting program. Server is designed with java using Eclipse and client will be designed with Android studio. This chatting program will be sending data through socket and server will deal with the database. Database will be consisted of text file.

1. **Usecase Diagram**



1. **Usecase Description**

|  |  |
| --- | --- |
| **Name** | Login |
| **Actors** | User/Client |
| **Preconditions** | Start App/ unlogined |
| **Basic flow** | 1. Input ID into EditText layer 2. Input Password into EditText layer 3. Click ‘Sign in’ Button 4. Send the data to Server 5. If there is the same ID and Password, notify that the login is successed. 6. Go to the next scene (It will be the ‘friendList’ scene). |
| **Extensions** | 5a. The server cannot find the identical information.   * Notify to client that login failed and try login again |
| **Post-conditions** | Move to next scene and send the logined\_user data. The status of user will change to ONLINE from OFFLINE. |
| **Special Requirement** | User connected to Internet and use Android based mobile phone |

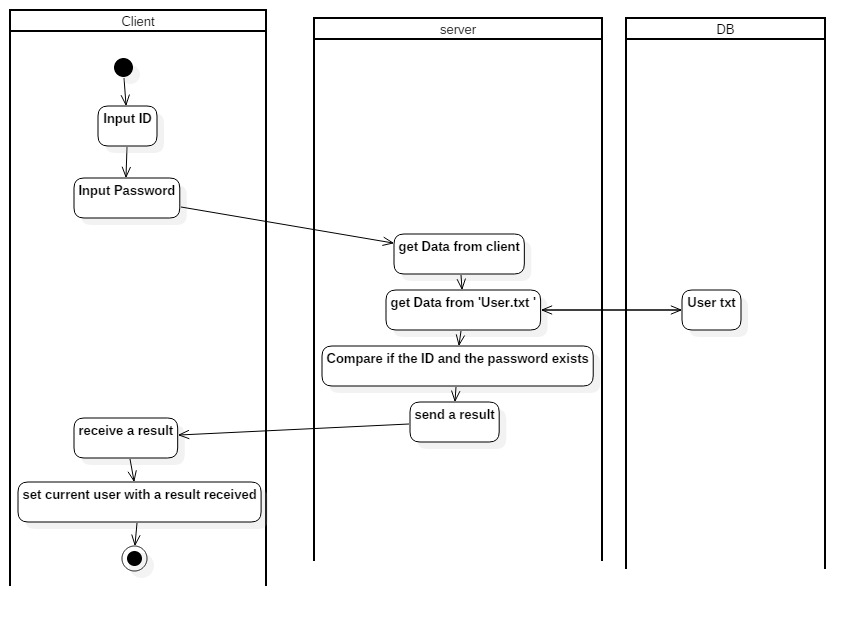
|  |  |
| --- | --- |
| **Name** | SignUp |
| **Actors** | User/Client |
| **Preconditions** | Start App/ unlogined |
| **Basic flow** | 1. Click SignUp button from the starting page. 2. Input user ID and click ‘verify ID’ to check if there is an existing ID in the DB. 3. Input user PW 4. Input user name. 5. Send the information to server. 6. Update user DB file. 7. Notify to client that sign up is done successfully. |
| **Extensions** | 2a. If there is an existing ID, clear the text from the editText layer so that user can write a data again.  5a. If PW and name is blank, toast a message in the scene. |
| **Post-conditions** | New user is created to DB and user can login with that ID and password. |
| **Special Requirement** | User connected to Internet and use Android based mobile phone |

|  |  |
| --- | --- |
| **Name** | Send Message |
| **Actors** | User/Client |
| **Preconditions** | logined |
| **Basic flow** | 1. Select a person to send a message 2. Find a file which name is like “(SenderID) (ReceiverID) 3. Load a file and show it on the scene. 4. Write a text into editText layer. 5. Click send button 6. Show the message on the scene. (loading) 7. Send the text to the server 8. Server saves the message into the DB as a form of <SenderID, ReceiverID, message, time> 9. Notify to client that the update is done. 10. Show the time of messages in the scene. 11. If the receiver is online, the scene of the receiver will be changed and notify the user that the message is received using a toast message due to thread. |
| **Extensions** | 2a. if there is no file like that, new file will be created.  7a. if the server is not available for 30 seconds the message will be deleted.  11a. if the receiver is offline, only the DB will be changed.  11b. if the receiver is busy, toast message won’t be available. |
| **Post-conditions** | Message is saved to DB.  Receiver can notice that the message has arrived. |
| **Special Requirement** | User connected to Internet and use Android based mobile phone. |

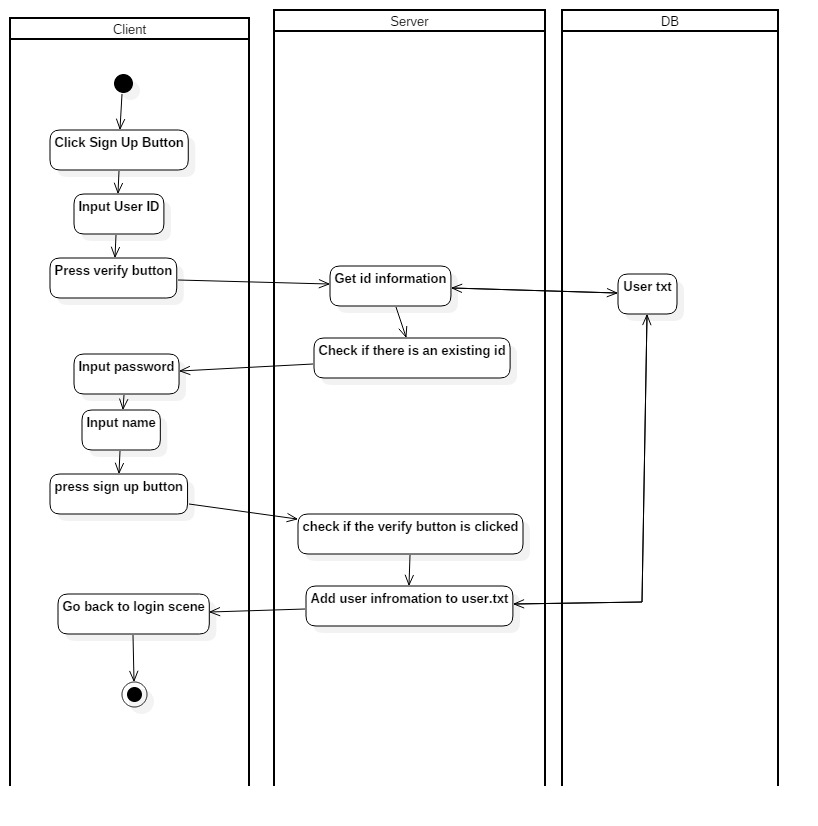
|  |  |
| --- | --- |
| **Name** | Logout |
| **Actors** | User/Client |
| **Preconditions** | Logined |
| **Basic flow** | 1. Click logout button 2. Change the status of a user from ONLINE/BUSY to OFFLINE 3. Socket will be closed 4. Go to the title scene |
| **Extensions** | - |
| **Post-conditions** | unlogin |
| **Special Requirement** | User connected to Internet and use Android based mobile phone. |

1. **Activity Diagram**

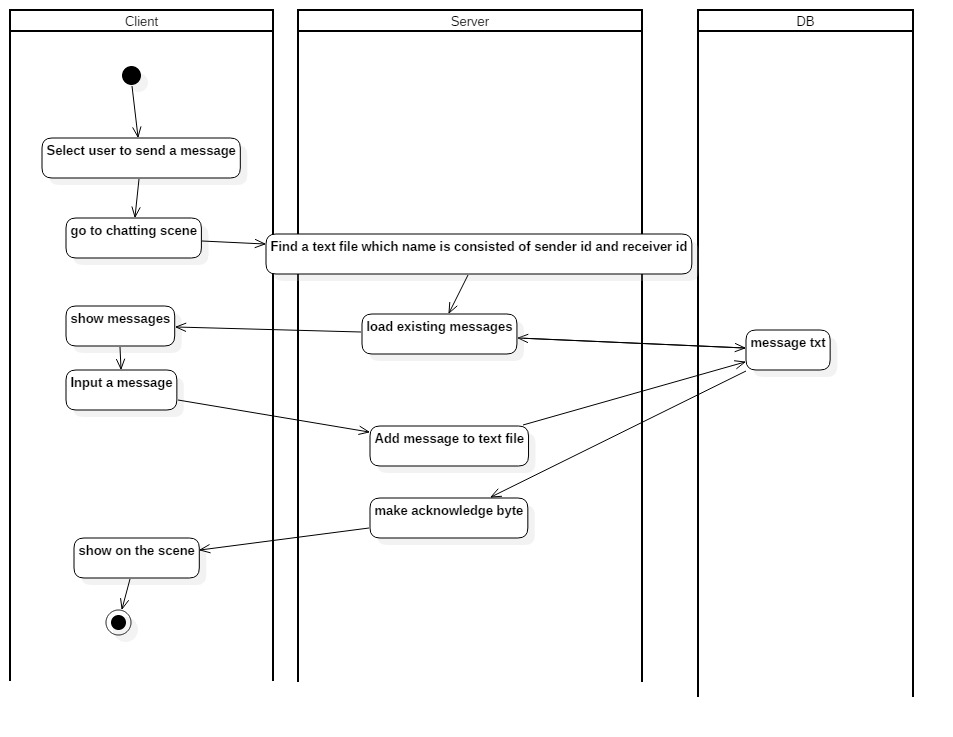
**a) Login**



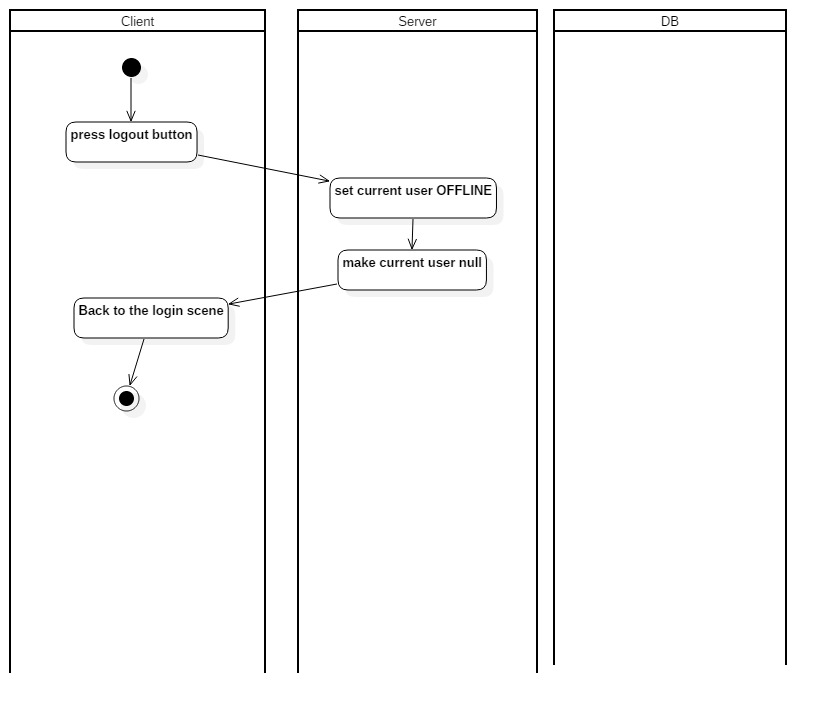
**b) Sign up**



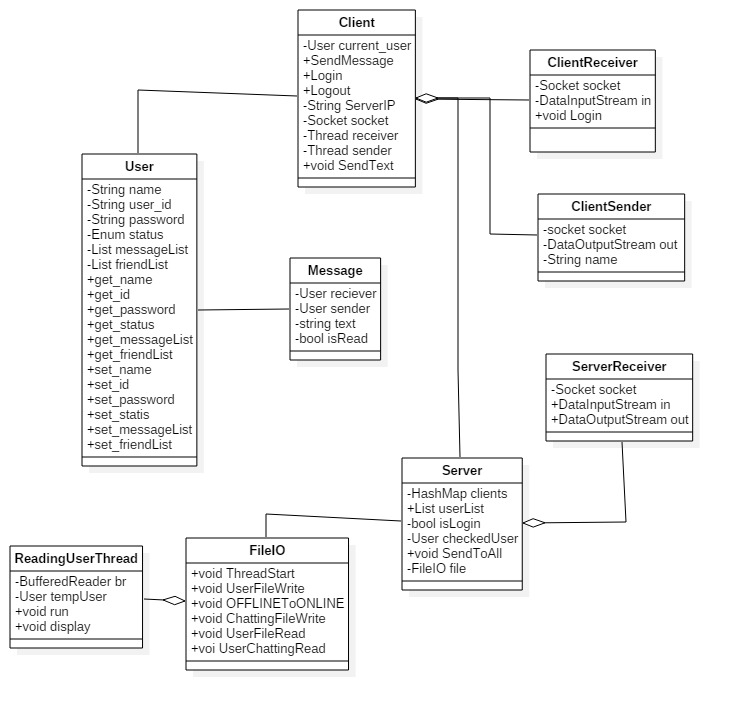
**c) Send Message**



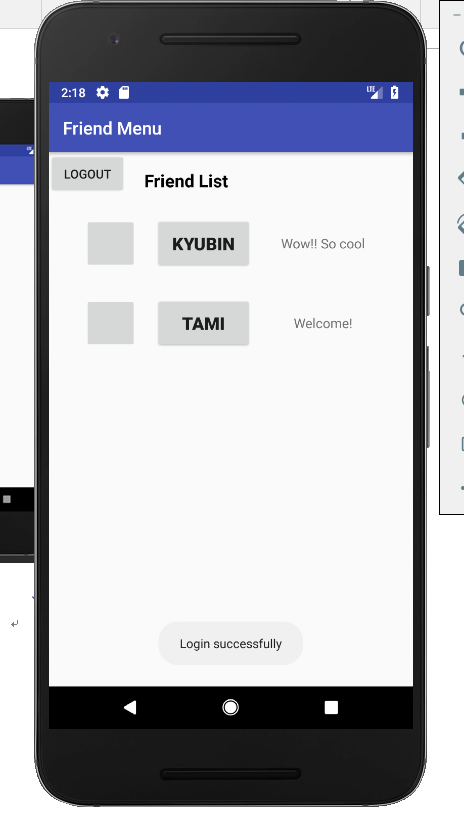
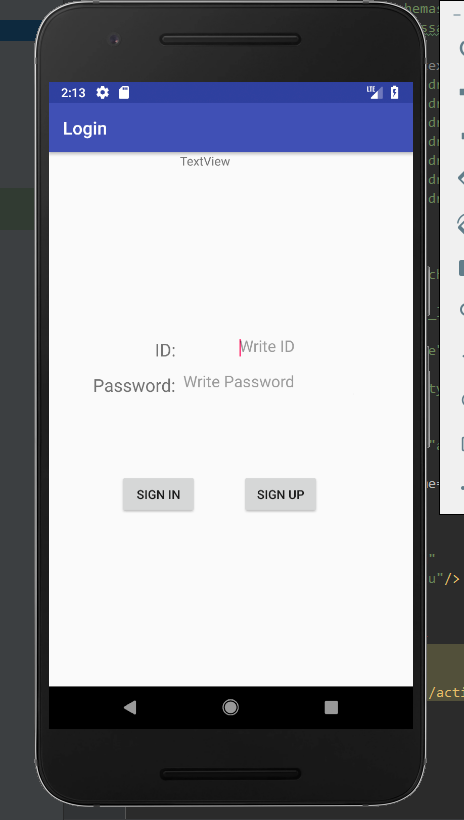
**d) Logout**



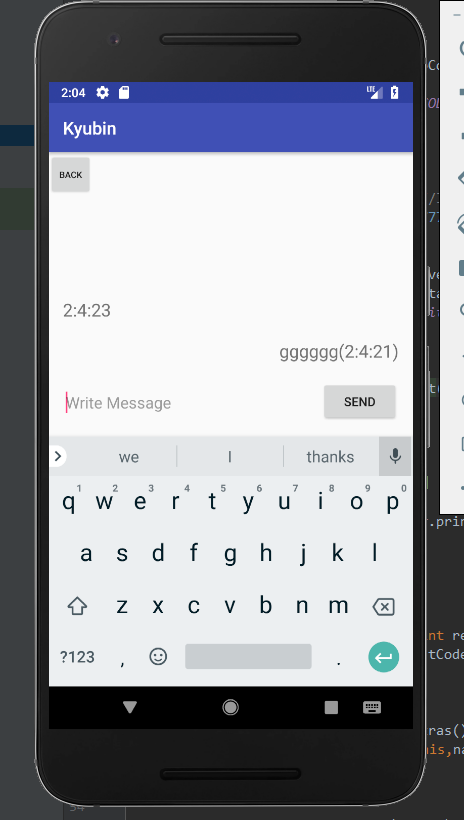
1. **Class Diagram**

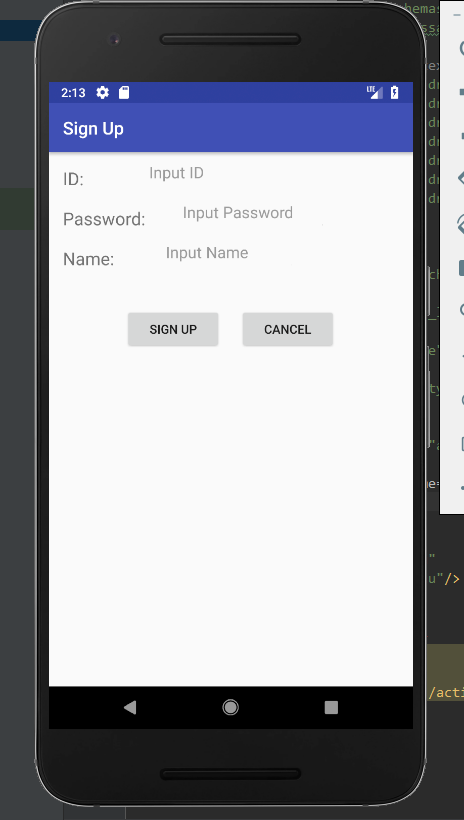


1. **UI Flow**



Start menu Friend List menu





Sign up menu Chatting Menu

**7. WBS**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **4/11** | **4/12** | **4/13** | **4/14** | **4/15** | **4/16** | **4/17** | **4/18** | **4/19** | **4/20** |
| **Login** | **Signup** | **Chatting** | **Chatting** | **Chatting** | **Chatting** | **UI** | **UI** | **Additional task** | **test** |