



# Chatting Application Specifications

## Client Side Application Specification:

- Sign in or Sign up as a first time user.
- Adding other friends to the contacts list.  
*(Hint: The email address may be used as a unique identifier for each registered client.)*
- Showing Online/Offline status of clients on the contacts list.
- User notification when another user becomes online.
- Ability to set status to different modes: Available, Busy, and Away.  
*(Hint: Use a different icon for each status. No special features are required for each status. This is just an icon that indicates the availability of the user.)*
- Ability to conduct one to one chat or chat group (more than two contacts).
- Ability to save a chat session.  
*(Hint: You may use XML and XML APIs)*
- Ability to transfer files to another client from the contact list (text, sound, movie ... etc).
- Sign out
- Exit



### **Server Side Application:**

The server application should have a simple GUI that starts the service and allows some administration facilities (i.e. allow the administrator to view certain reports and statistics about the users). The details are as follows:

- Ability to start / stop the server.
- Allow users to signup (i.e. registration of first time users) and store their information (name, email, username, password, gender, country, ... etc)
- Accept connections and store information about clients' status.
- Ability to send an announcement message to all online users (For Example to advertise for a product or to announce the release of a newer version of the chat program).
- View the number of online and offline users.
- Show some statistics about the users. Maintain the user's contact list so that the user can sign in from any machine he wants, without having to worry about re-creating his contact list.
- **Please design a USER FRIENDLY GUI for your application. (Make the user happy to use your program ;-)** (Use Swing Components).
- It is recommended that you use a 3<sup>rd</sup> party LookAndFeel.
- Please package your work in an executable jar file (easy to use by users).
- Your connections may be RMI based.

***Remember: All the above mentioned basic functionality SHOULD BE COMPLETED (and tested) before attempting to work on any bonus features from the list below.***



## **“Optional” Bonus Features**

**Each group may choose up to 3 bonus features from the list below:**

- Allow the client to sort his contact list names. Provide a feature to categorize the names in your contact list under certain categories (eg. Friends, Family, Work ...etc). In such case, the contact list should display each category type with some names under it.
- Allow the user to block certain clients. Moreover, allow the user to log in and yet be invisible to all the clients on his contact list. In such case, the others are not notified of that user's presence. However, the user is allowed to chat with anyone from his contact list if he wants. In such case, the user only appears online to the clients he has engaged in a chat with.
- Provide a facility for offline text messages, which means that a user can send a message to another client even if the other is offline. When that client logs in at any time, he should then receive all the messages that were sent to him while he was offline.  
***If you choose this bonus point, then make sure that the administrator can send the announcement to ALL users, whether online or offline.***
- Allow the administrator to view all clients' information from the database, and display in a JTable. Moreover, allow editing in the table and reflect the updates to the database.
- Allow voice chat! (Hint: See Sound API, and Java Media Framework)
- Allow clients to perform video chat using the internal laptop camera
- Allow clients to view and post feeds to their social networks (Facebook-Twitter-LinkedIn)
- Allow clients for remote desktop assistance from one of the contacts (Use Robot Class)
- Allow clients to send emails to each other through your chatting program. Then make sure that when the client registers and signs up for the first time, his username and password should be automatically sent to his email address. (Hint: See Java Mail API.)



### **Project Delivery**

Delivery Date: 3<sup>rd</sup> of February, 2016

Delivery Package:

- Executable JAR files
- Client's Netbeans project
- Server's Netbeans project
- Database schema and/or database backup
- Any 3<sup>rd</sup> party APIs (if used)
- Read me file that describes how to run and use the project and the team members' names.

Please, pack all of the pre-mentioned items on a CD and deliver it on the Date.

### **Milestones:**

1<sup>st</sup> Milestone: 28<sup>th</sup> of January, 2016:

- Database Schema and implementation.
- GUI V1.0
- Draft UML, plan or any description for how do you think about implementing your project.
- Show the tasks assignment to team members.