# Chatting Application Specifications

## ***Client Side Application Specification:***

- Registration
    - User must be able to register for the application through a valid phone number (The users phone number will be the unique identifier of his/her account on Chat Application).
    - (Phone Number, Display Name, email, picture, password, confirmation password, gender, country, date of birth, bio.)
- Sign in
    - When running the application, user must be prompted to register their phone number. If user skips this step, application should close.
    - After he/she entered the valid phone number they'd be prompted to enter the password.
- Adding New Contact
    - The user will enter their contact phone number/numbers at same time.
    - Then the contact will receive an invitation by email if it exists to join as his/her friend (Bonus).
    - When the contact signs-in, it will receive the invitation to accept or reject this invitation.
    - If the invitation is still pending and the contact user adds also the user, this invitation will be accepted automatically (Bonus).
- Showing Contacts Status
    - Show status whether they are online or offline.
    - If the contact is online, it should be classified in any mode (Available, Busy, and Away) of clients on the contacts list according to its current status.

- 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.)*
  - Appear offline (Bonus)

    *(Hint: contact can see messages without seen mode to the other partner and when becoming online all seen message in all seen conversation to be displayed as seen.)*
- Contact Block (Bonus)
  - Allow the user to block certain clients.
  - In such a 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.
- User notification
  - When a contact becomes online/offline and update they status in contact list.
  - When a contact receives messages.
- Send Messages
  - Ability to conduct one to one chat
  - Ability to conduct group chat (more than two contacts).
  - All sent message must be with these attributes (message, font style, font color, text background, font size, bold, italic, Underline, timestamp, emoji (Bonus))
  - Message can be rich text as html (Bonus).
  - Offline Messages (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) (Bonus).

- Group Contact (Bonus)
  - Provide a feature to categorize the names in your contact list under certain categories (e.g. Friends, Family, Work …etc.). In such case, the contact list should display each category type with some names under it.
- Voice Chat (Bonus)
  - Allow voice chat! (Hint: See Sound API, and Java Media Framework).
- Video Chat (Bonus)
  - Allow clients to perform video chat using the internal laptop camera
- File Transfer
  - Ability to transfer files to another client or clients from the contact list (text, sound, short video, long movie ... etc.).
- Sign out
  - Ability to sign out from the application and the application should forget the password for the user but not his id (Note: to use it to sign in again for this user by password only).
- Exit
  - Ability to exit application without forgetting user so the user can open the application directly without Sign in in secured configuration file.
- Social Network Integration (Bonus)
  - Allow clients to view and post feeds to their social networks (Facebook-Twitter-LinkedIn).
- Application installer into system programs (Bonus)
  - Allow users to install your application into system and run as service in background.
- Chatbot service
  - Allow users to enable chatbot to automatic answer on behalf of me instead of not answering him.

    *(Hint: You can use any third party library to do that like (cleverbot, pandorabots, jabberwacky, AIML*(Bonus)*) )*

## ***Server Side Application:***

- Manage Service
  - Ability to start / stop the service on server.
- User Registration
  - Allow users to signup (i.e. registration of first-time users) with password to be changed for first user sign in in first time.
- Server availability
  - Accept connections and keep track with information about clients' status.
- Server Announcement
  - Ability to send an announcement message (Formatted text 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).
  - Announcement message to be rich text message as html (Bonus)
- Server Statistics
  - View the number of online and offline users.
  - Show statistics about the users' gender.
  - Show statistics about the users' country.
  - Show statistics about the users' entries number on system (Bonus).
- Data Availability
  - 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.
  - Maintain the user's chat sessions (Bonus).
- User Data Modification (Bonus)
  - Allow the administrator to view all clients' information from the database and display it in a tabulation form. Moreover, allow editing in the table and reflect the updates to the database.

## *Notes:*

- **Please design a USER-FRIENDLY GUI for your application. (Make the user happy to use your program ;-).**
- **Use JavaFX with CSS to represent the GUI of project.**
- **All communication between server and clients or between clients themselves should be encrypted.**
- **Use XML to save chat session data and any configuration as needed.**
- **Use Maven as project management Tool for your project with modules.**
- **Use Design principles and patterns in your project.**
- **Also provide your unit test for your services.**
- **Please package your work in an executable jar file (easy to use by users).**
- **Your connections will be RMI based.**

  ***Remember: All mandatory functionalities SHOULD BE COMPLETED (and tested) before attempting to work on any bonus features.***

## ***Project Delivery***

Delivery Date: 13th of February 2025

Delivery Package:

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

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

## ***Milestones:***

- Milestone: 18th of January 2025:
  - Database Schema.
  - GUI design V1.0.
- Milestone: 26th of January 2025:
  - Database Layer implementation.
  - Final UI Screens.