# Multi-Client Chat Application Design Document

## Objective:

Design and implement a multi-client chat application using Java, JavaFX for GUI, and sockets for networking. The app will supports multiple clients connecting to a server, exchanging messages in real time.

## 1. Architecture Overview

* **Client Application (JavaFX)**: A JavaFX-based GUI for displaying messages and interacting with the chat server.
* **Server Application**: A Java-based server that manages client connections, maintains chat history, and handles broadcasting messages.

## 2. Functional Requirements

* Client:
  + Connect to the server.
  + Send messages to the server.
  + Receive messages from the server.
  + Display chat history.
* Server:
  + Accept multiple client connections.
  + Broadcast messages to connected clients.
  + Store chat history.
  + Handle user disconnects gracefully.

## 3. System Components

* Client Application:
  + GUI: JavaFX based interface to read/write messages.
  + Socket Communication: Handles message exchange with the server.
* Server Application:
  + Socket Server: Accepts and manages client connections.
  + Message Service: Broadcasts message to clients and stores chat history.
  + Client Handler: Manages individual client connections and communications.

## 4. Development and Testing

Initially, the project was designed without unit tests. This oversight led to a reliance on manual testing of the program.

## 5. Usage (README)

**Prerequisites**:

* Install Java Development Kit (JDK) version 8 or later.
* Set up the Eclipse IDE with JavaFX 21 support.

1. Server Application:
   1. Open the iMessageCore package.
   2. Run Server.java.
   3. Screenshot of server running:

A screenshot of a computer

Description automatically generated

1. Client Application:
   1. Open the iMessageGUI package.
   2. Run iMessage.java.
   3. Screenshot of client running:

A screenshot of a computer

Description automatically generated

1. Instructions:
   1. Use the text field to type messages.
   2. Click the “Send” button or press Enter to send the message to the server.
   3. Messages from other clients will appear in the chat window.
   4. The chat window will display messages exchanged during the session.
2. Testing the System:
   1. Start multiple instances of the client application.
   2. Verify that each client receives messages sent by others.