

CHAT APPLICATION OVERVIEW

Chat Program using Socket Programming

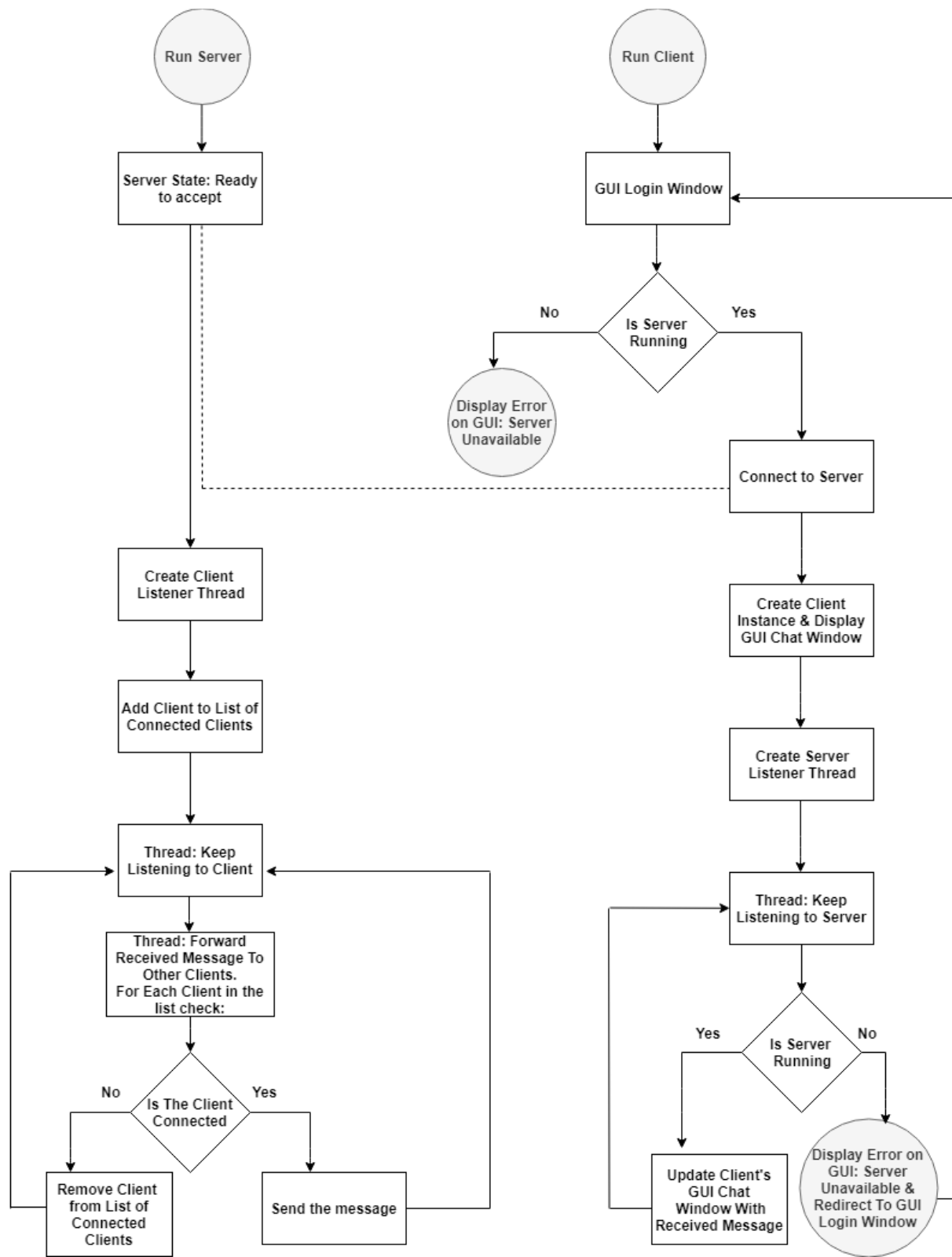
The project includes the following features

- A Server that can be run at a configurable host and port
- GUI for clients to login and chat
- Multiple clients can join the chat by logging in from different computers
- Exceptions in events such as abrupt disconnection from the server, logging out of a client during an ongoing chat are handled by conveying the appropriate information to the other active client(s)

Approach

- ✓ The chat application was designed using Java, Applets (JavaFx for GUI) and multi-threading at Client and Server
- ✓ The Server is created as a separate java class, which is to be run first
- ✓ The client instances are created and connected to the running server when the Chat Application is launched through the Java Applets UI
 - Each Client Instance is given a randomly generated internal unique identifier
- ✓ Whenever a new client logs in and connects to the server, a separate thread is created from
 - the newly connected client to listen information about other clients from the server
 - the server to listen to the incoming messages from the newly connected client and forward to other active client(s)
- ✓ The connection management between the client and the server is done through the creation of socket and its various phases (creation, accept, close)
- ✓ Any unanticipated connection loss from the server or disconnection of any client is handled through SocketExceptions in java and appropriate errors are displayed to the active clients on their GUI

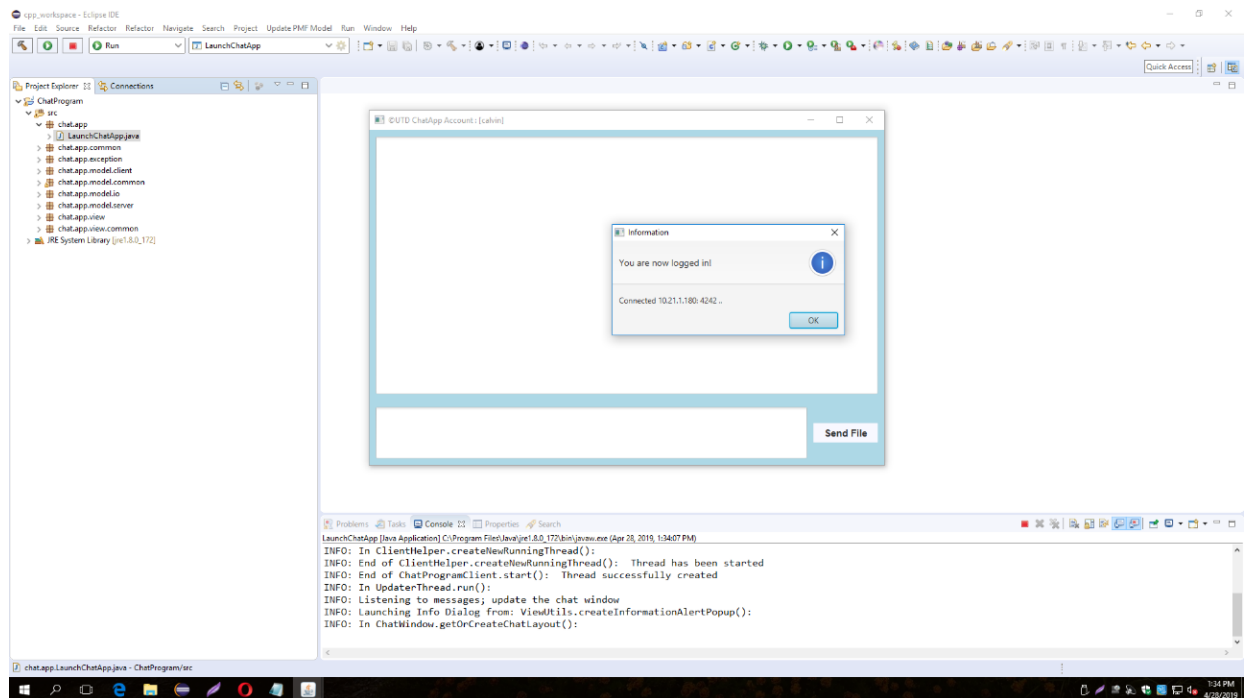
Flowchart



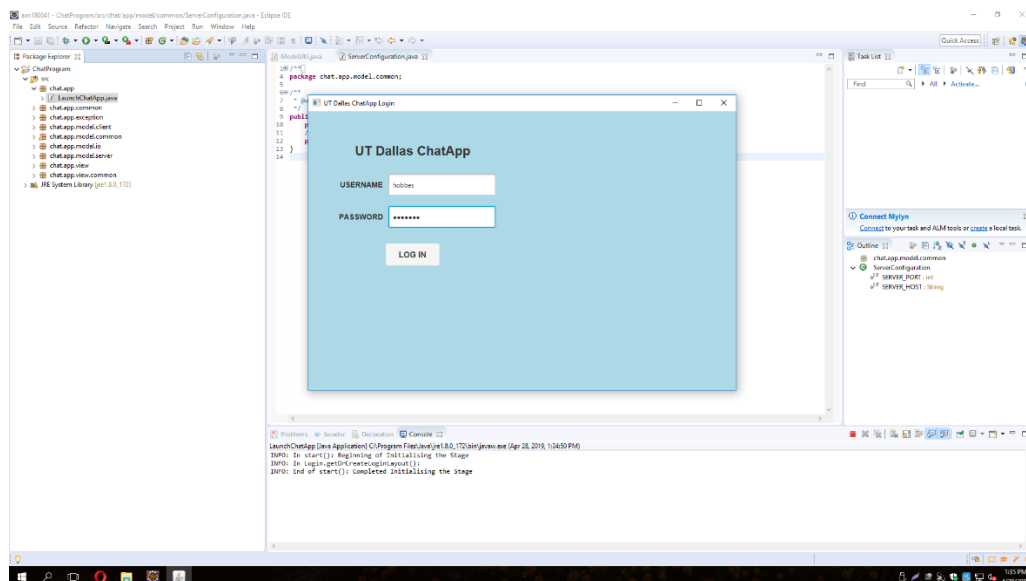
CODE EXECUTION SNAPSHOTS

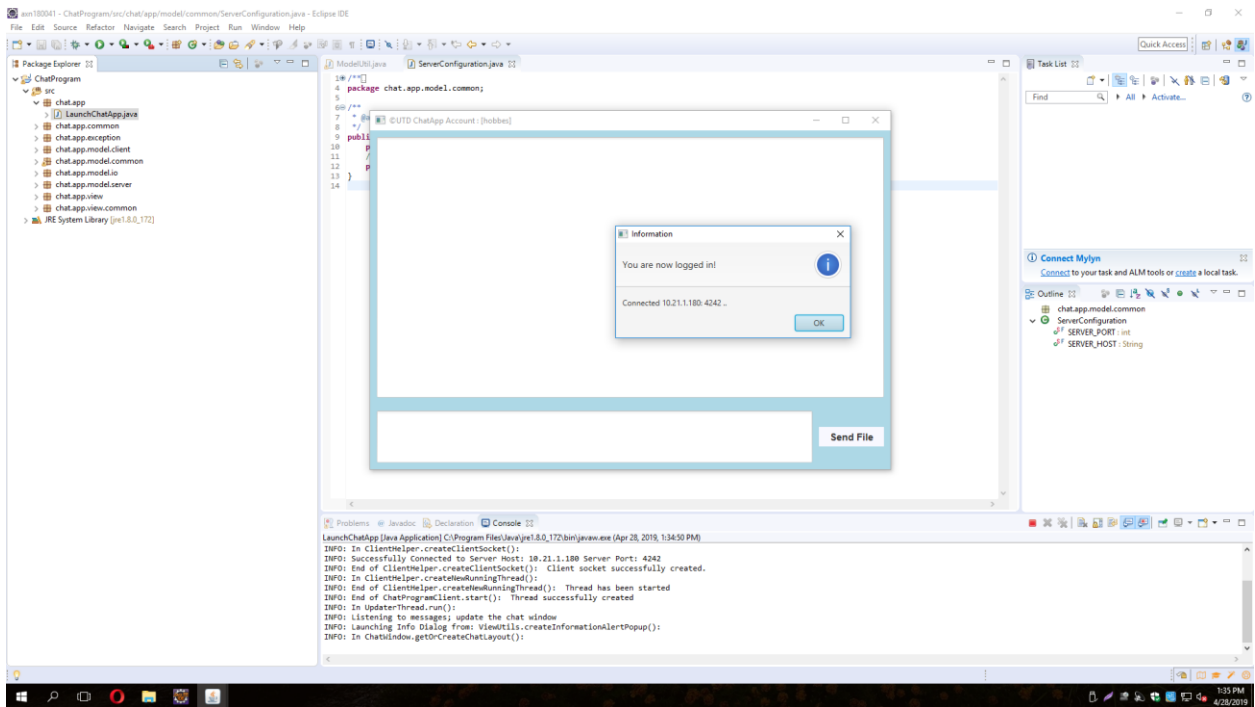
Server was made to run on one computer. The individual clients each were running on different computers and connecting to the server.

- **Graphical User Interface Login**
 - **Client 1 Logged in**

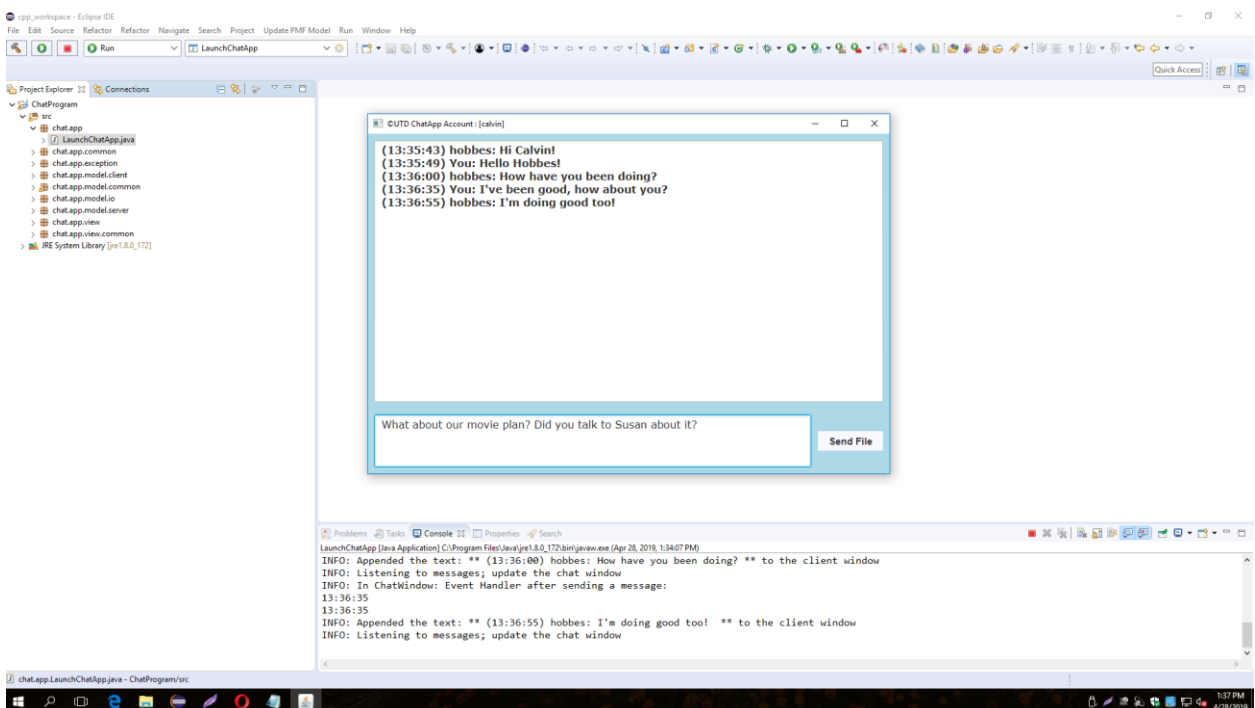


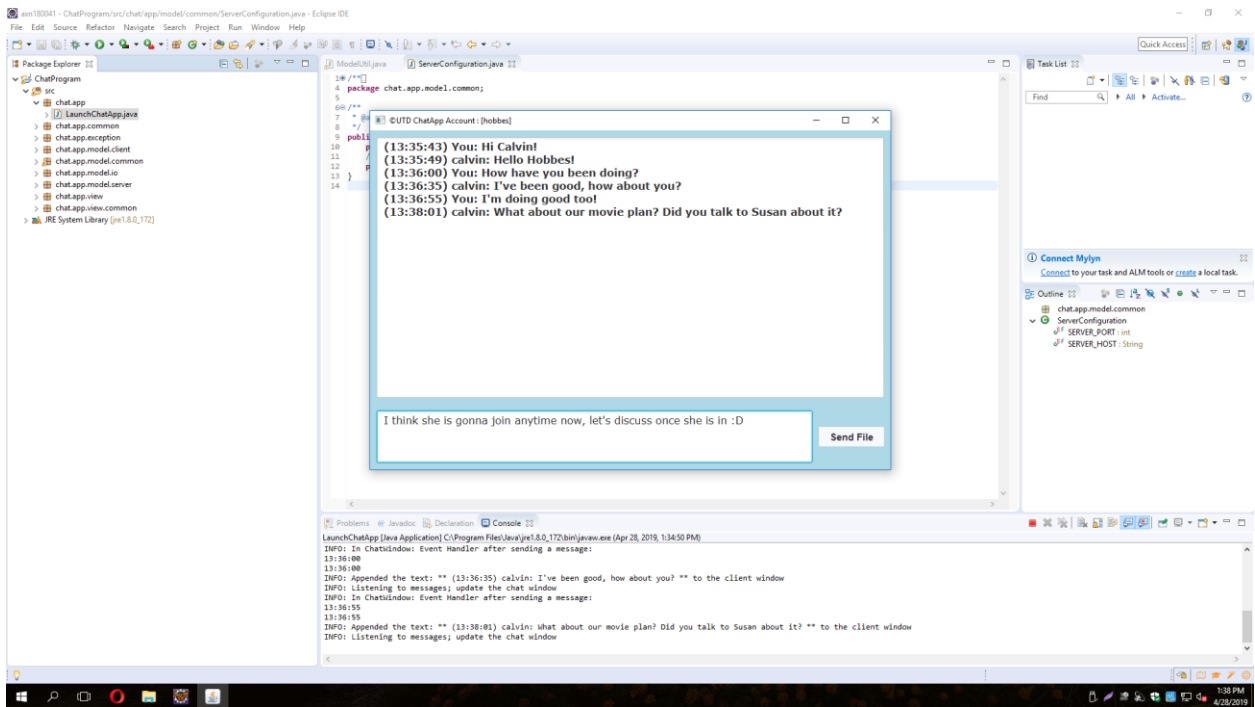
- **Client 2 Logged In**



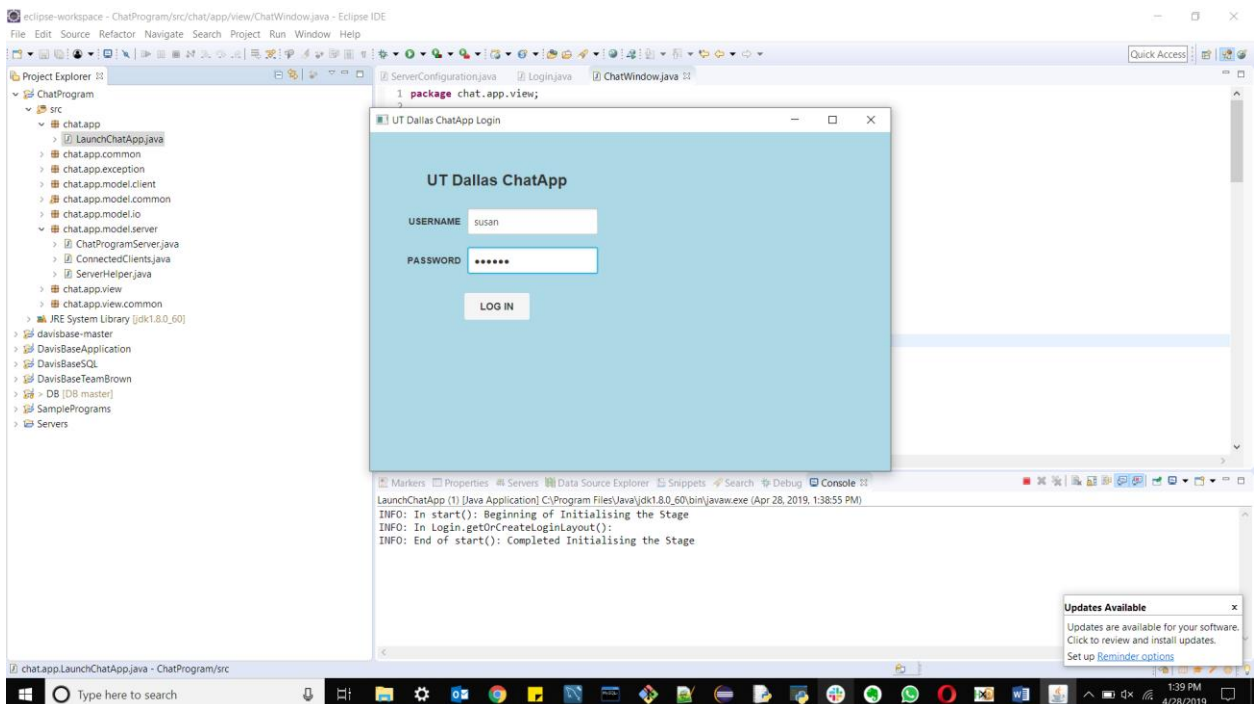


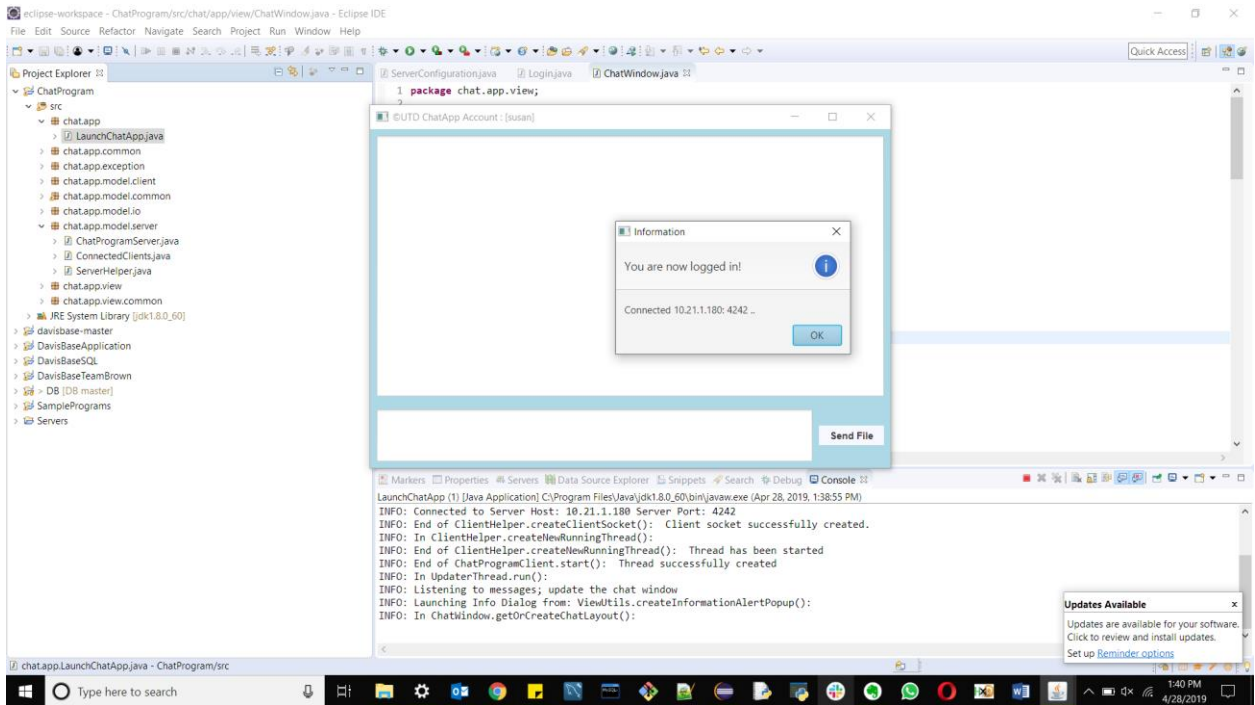
- Two Clients Communication
 - Client 2 and Client 1 communicating



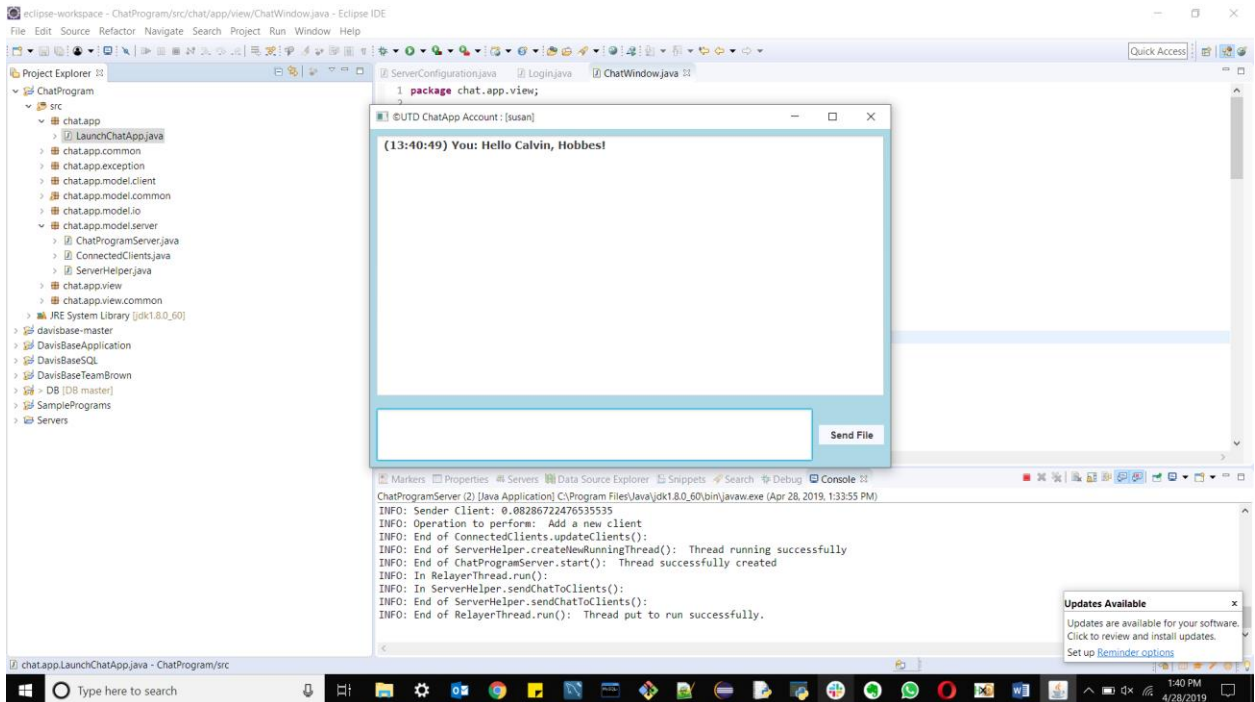


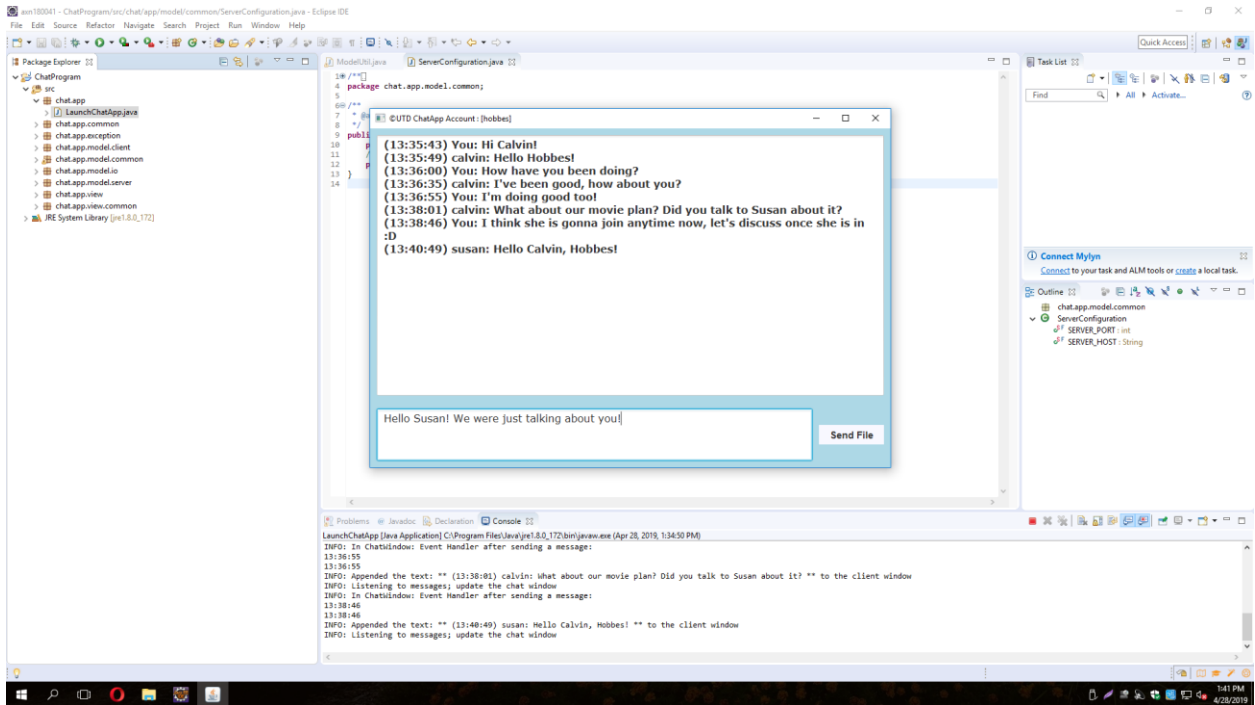
- **Multi-party communication**
 - **Client 3 Logs in**



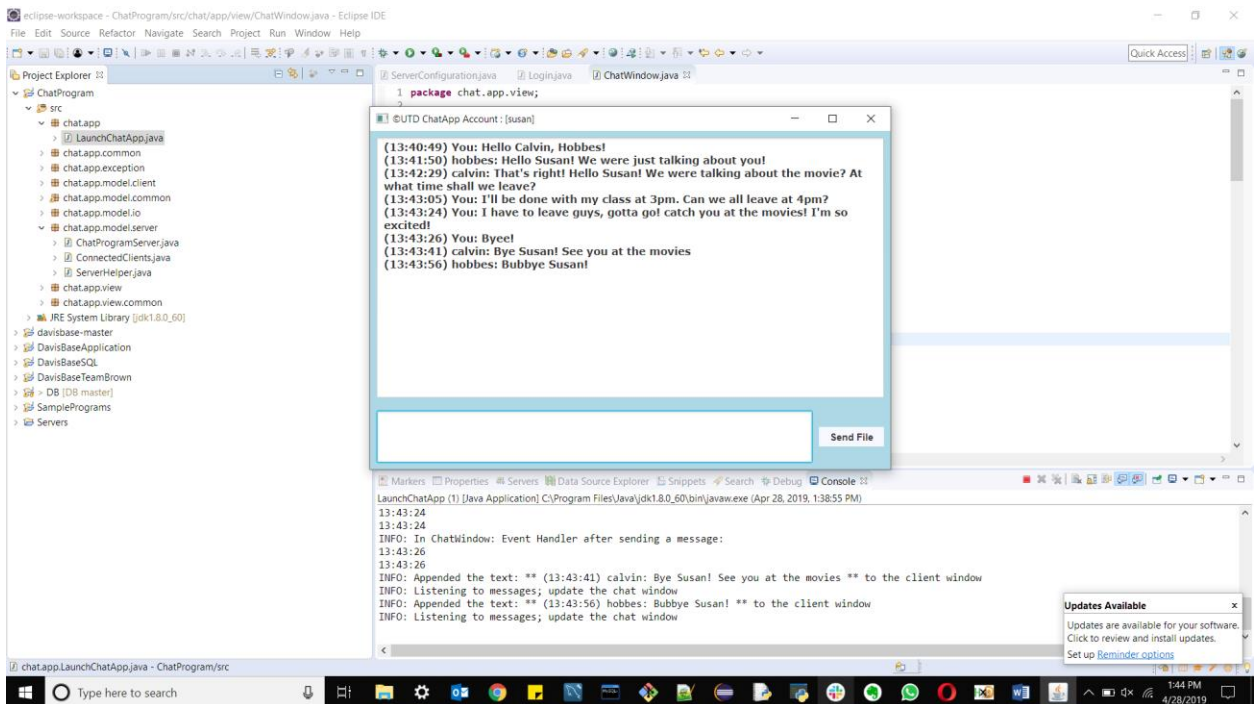


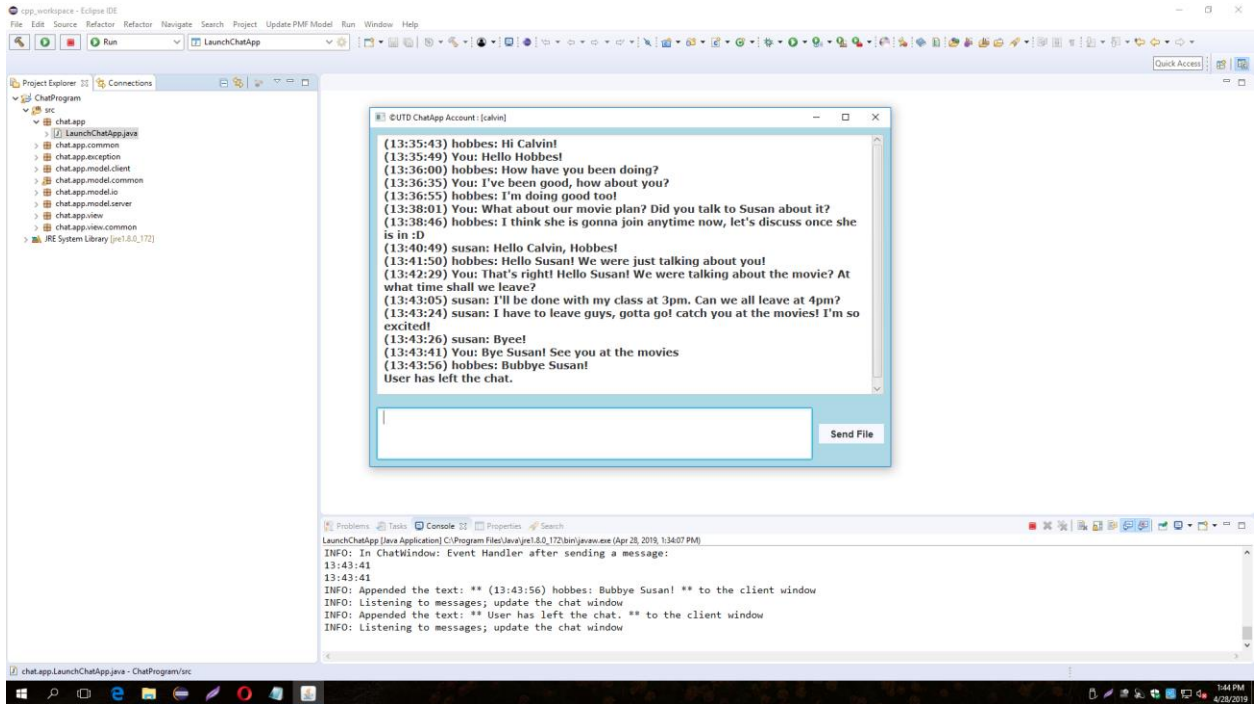
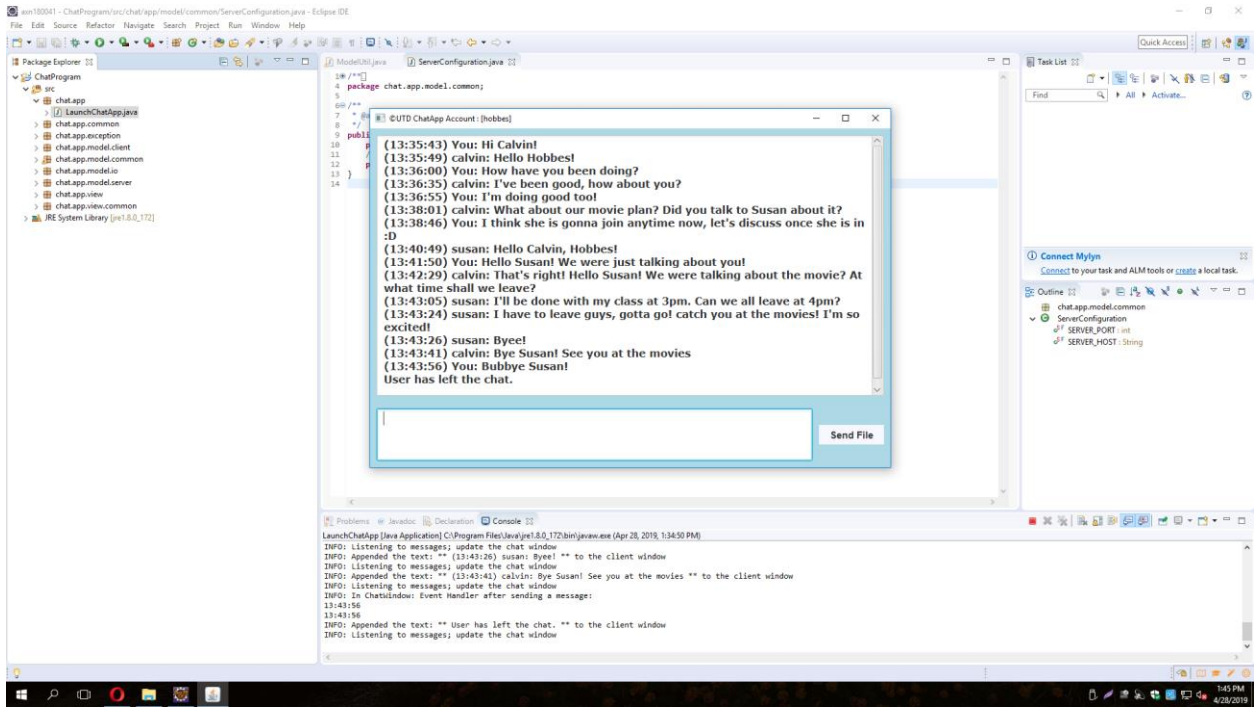
○ Client 3, Client 2 and Client 1 communicate with each other



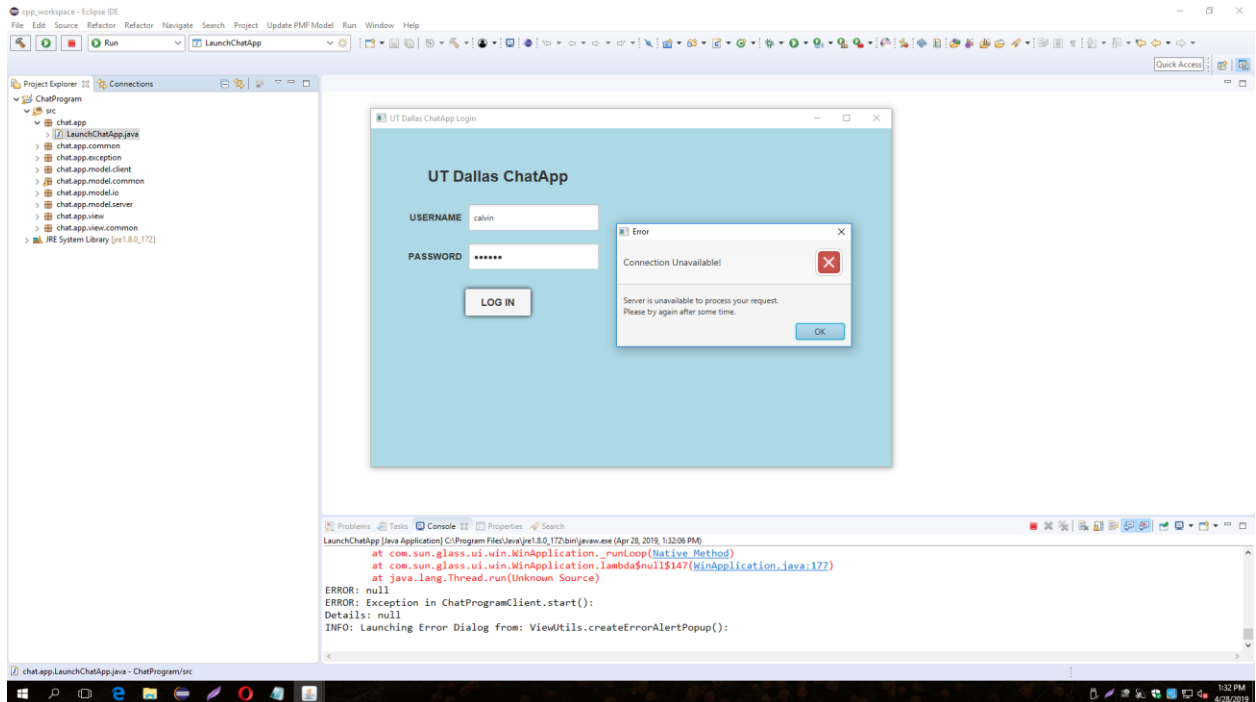


○ Client 3 leaves the chat group

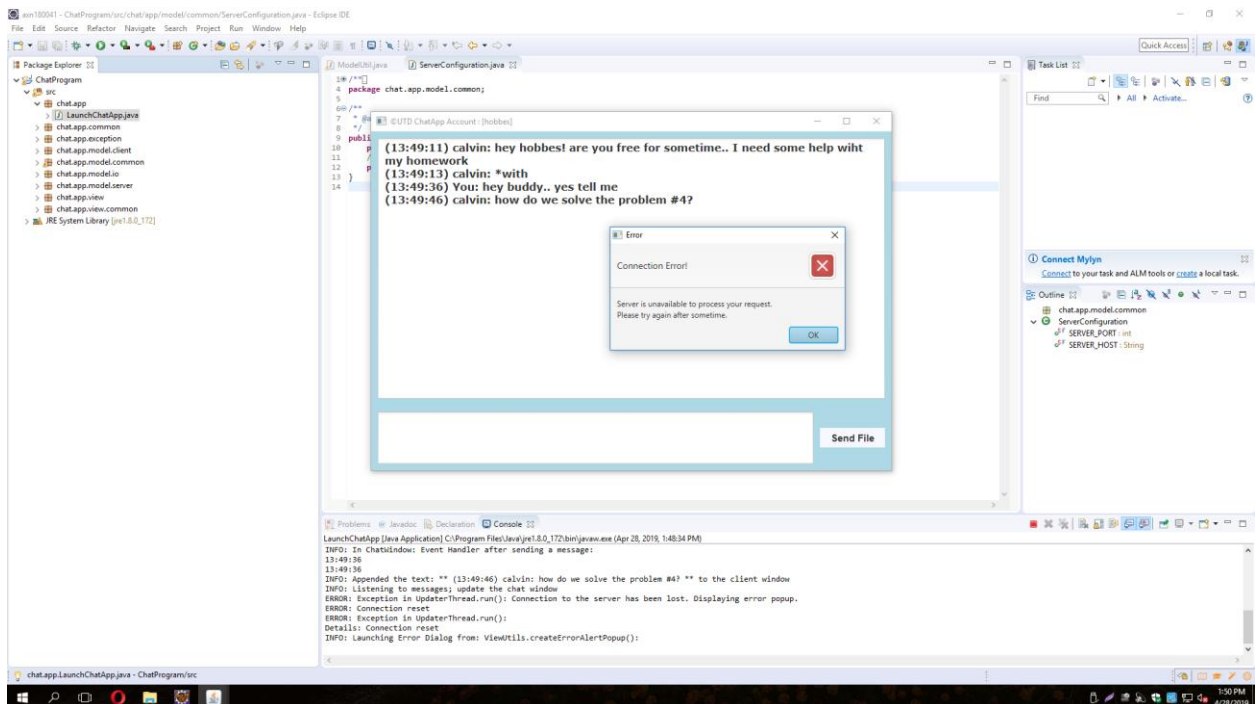


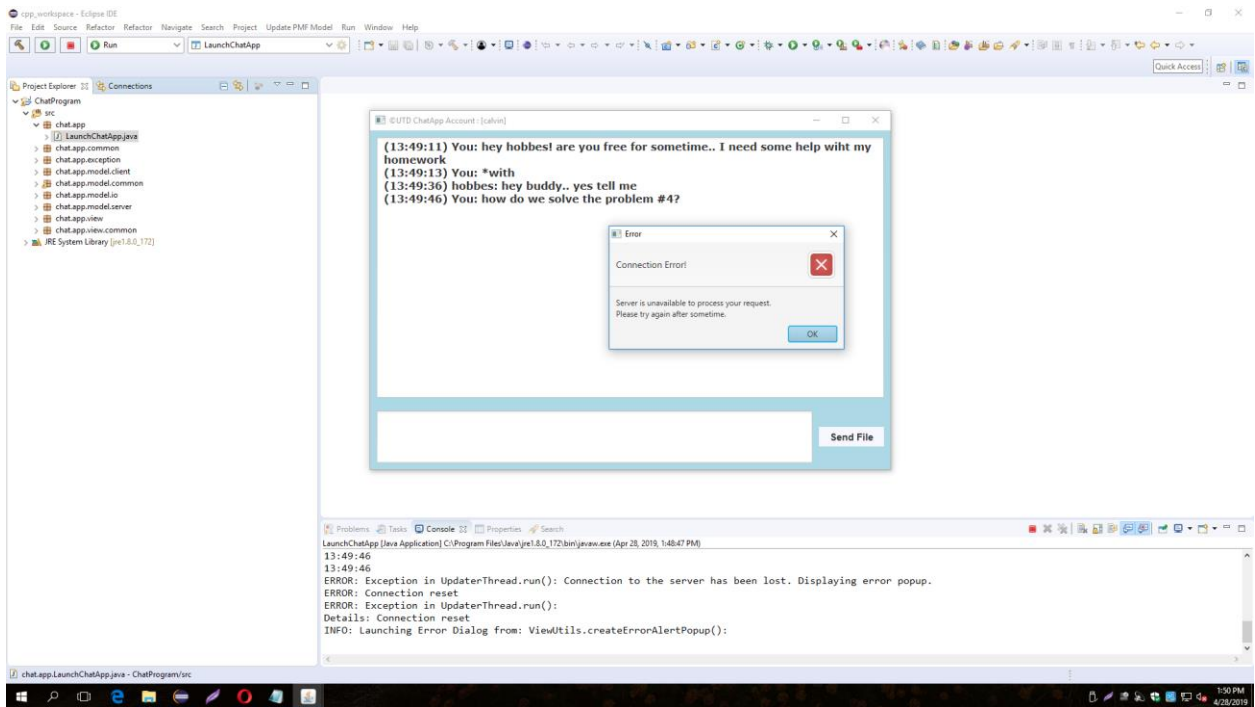


- **Handling Server Unavailability**
 - **Client trying to Log In when the Server is not available**

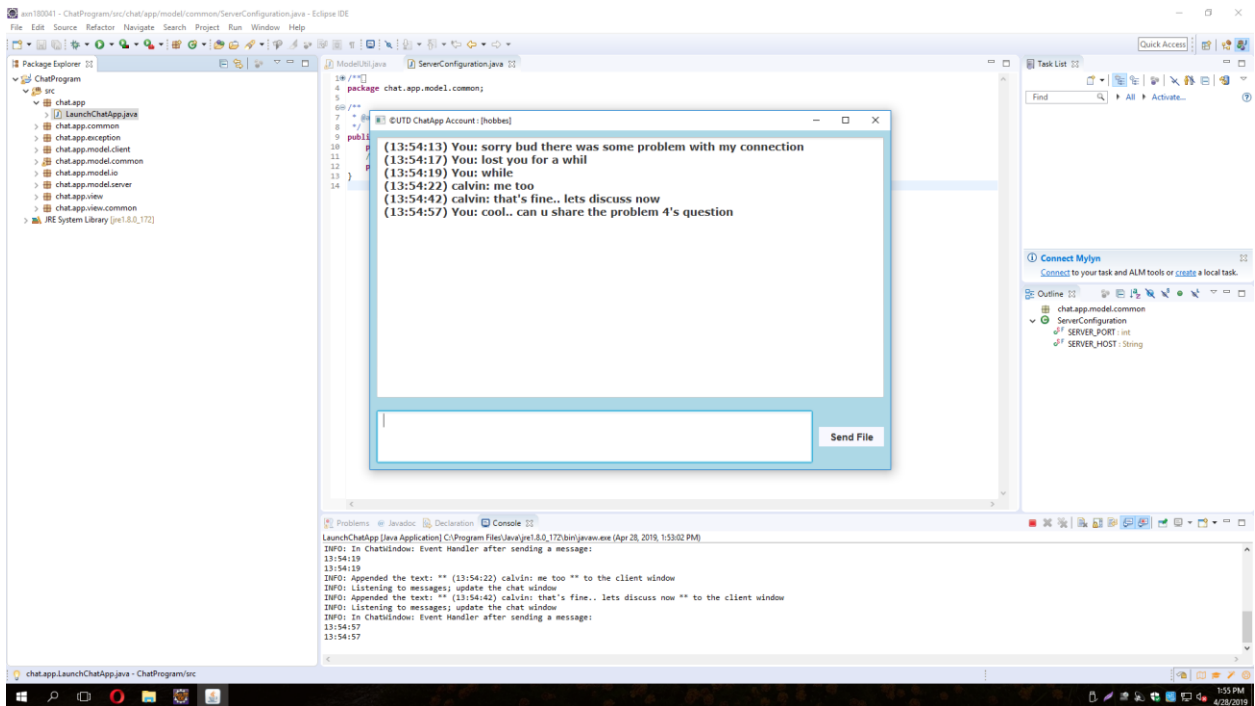


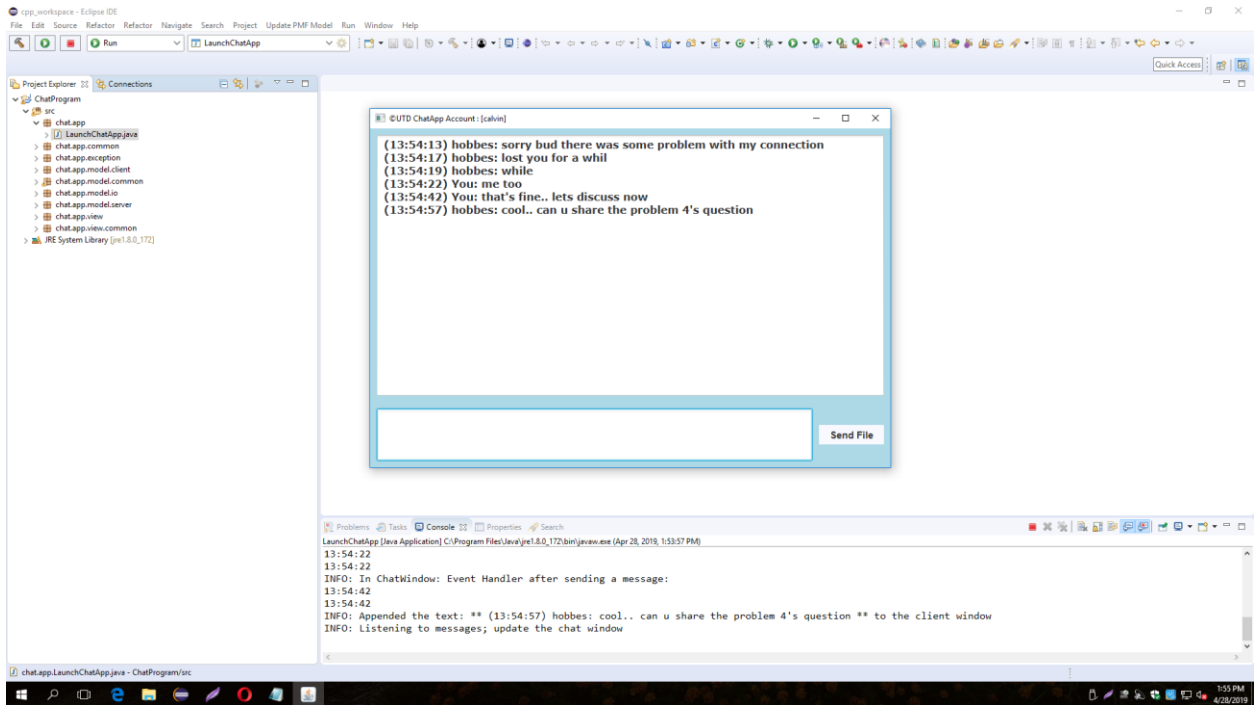
- **Server goes unavailable while Client 1 and Client 2 communicating**



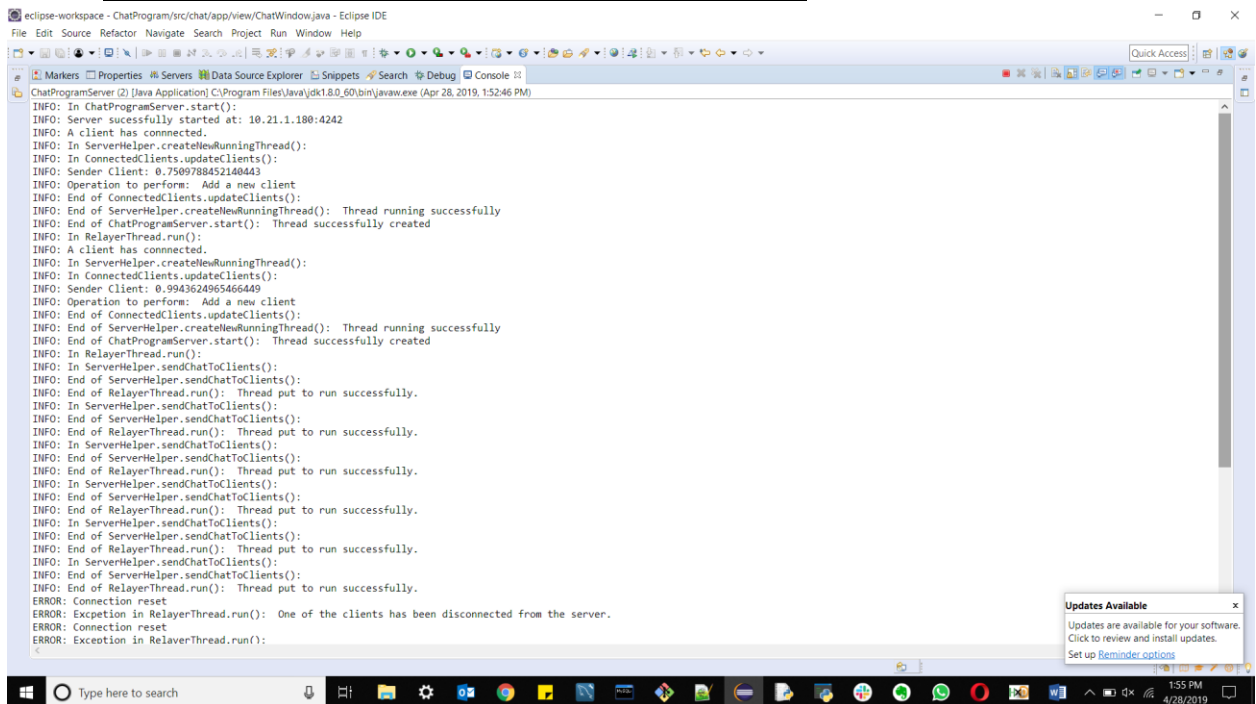


- **Client 1 and Client 2 resume communication once Server is available**

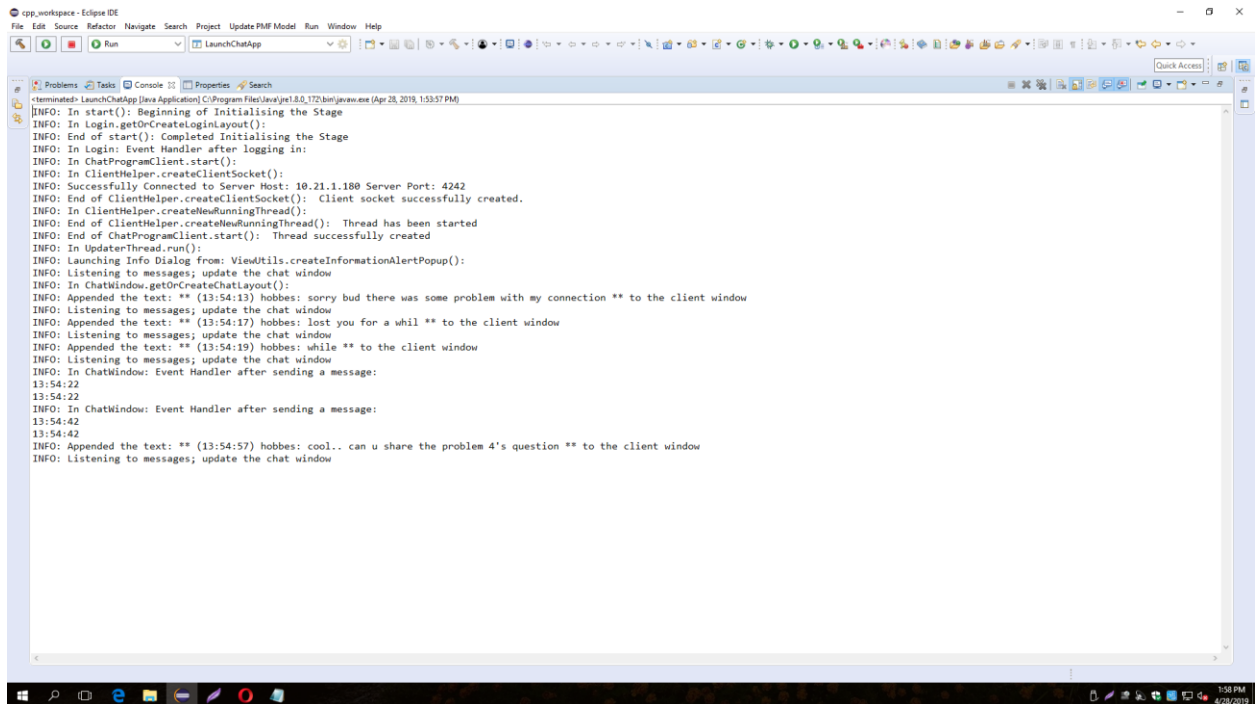




○ Server Logs for the Intermittent Server Unavailability



- **Sample Logs**
 - **Client Logs**



The screenshot shows the Eclipse IDE interface with the 'Console' view active. The console displays a series of log messages from a Java application named 'LaunchChatApp'. The logs indicate the application's startup sequence, including initialization, login handling, socket creation, and thread management. It also shows a simulated chat interaction where a user named 'hobbes' sends messages and the application responds with status updates.

```
terminated: LaunchChatApp [Java Application] C:\Program Files\Java\jre1.8.0_172\bin\java.exe (Apr 28, 2019, 1:53:57 PM)
INFO: In start(): Beginning of Initialising the Stage
INFO: In Login.getOrCreateLoginLayout():
INFO: End of start(): Completed Initialising the Stage
INFO: In Login: Event Handler after logging in:
INFO: In ChatProgramClient.start():
INFO: In ClientHelper.createClientSocket():
INFO: Successfully Connected to Server Host: 10.21.1.180 Server Port: 4242
INFO: End of ClientHelper.createClientSocket(): Client socket successfully created.
INFO: In ClientHelper.createNewRunningThread():
INFO: End of ClientHelper.createNewRunningThread(): Thread has been started
INFO: End of ChatProgramClient.start(): Thread successfully created
INFO: In UpdaterThread.run():
INFO: Launching Info Dialog from: ViewUtils.createInformationAlertPopup():
INFO: Listening to messages; update the chat window
INFO: In ChatWindow.getOrCreateChatLayout():
INFO: Appended the text: ** (13:54:13) hobbes: sorry bud there was some problem with my connection ** to the client window
INFO: Listening to messages; update the chat window
INFO: Listening to messages; update the chat window
INFO: Appended the text: ** (13:54:17) hobbes: lost you for a while ** to the client window
INFO: Listening to messages; update the chat window
INFO: Listening to messages; update the chat window
INFO: In ChatWindow: Event Handler after sending a message:
13:54:22
13:54:22
INFO: In ChatWindow: Event Handler after sending a message:
13:54:42
13:54:42
INFO: Appended the text: ** (13:54:57) hobbes: cool.. can u share the problem 4's question ** to the client window
INFO: Listening to messages; update the chat window
```

Sample Server Logs

[illegible]

INFO: End of ServerHelper.sendChatToClients():
INFO: End of RelayerThread.run(): Thread put to run successfully.
INFO: In ServerHelper.sendChatToClients():
INFO: End of ServerHelper.sendChatToClients():
INFO: End of RelayerThread.run(): Thread put to run successfully.
ERROR: Connection reset
ERROR: Excpetion in RelayerThread.run(): One of the clients has been disconnected from the server.
ERROR: Connection reset
ERROR: Exception in RelayerThread.run():
Details: Connection reset
INFO: In ServerHelper.sendUserLeftChat():
INFO: End of ServerHelper.sendUserLeftChat(): Message successfully updated to other clients.
INFO: In ConnectedClients.updateClients():
INFO: Sender Client: 0.08286722476535535
INFO: Operation to perform: Remove a disconnected client
INFO: End of ConnectedClients.updateClients():
INFO: In ServerHelper.sendChatToClients():
INFO: End of ServerHelper.sendChatToClients():
INFO: End of RelayerThread.run(): Thread put to run successfully.
INFO: In ServerHelper.sendChatToClients():
INFO: End of ServerHelper.sendChatToClients():
INFO: End of RelayerThread.run(): Thread put to run successfully.
ERROR: Connection reset
ERROR: Excpetion in RelayerThread.run(): One of the clients has been disconnected from the server.
ERROR: Connection reset
ERROR: Exception in RelayerThread.run():
Details: Connection reset
INFO: In ServerHelper.sendUserLeftChat():
INFO: End of ServerHelper.sendUserLeftChat(): Message successfully updated to other clients.
INFO: In ConnectedClients.updateClients():
INFO: Sender Client: 0.554851859588879
INFO: Operation to perform: Remove a disconnected client
INFO: End of ConnectedClients.updateClients():
ERROR: Connection reset
ERROR: Excpetion in RelayerThread.run(): One of the clients has been disconnected from the server.
ERROR: Connection reset
ERROR: Exception in RelayerThread.run():
Details: Connection reset
INFO: In ServerHelper.sendUserLeftChat():
INFO: End of ServerHelper.sendUserLeftChat(): Message successfully updated to other clients.
INFO: In ConnectedClients.updateClients():
INFO: Sender Client: 0.9772266450860949
INFO: Operation to perform: Remove a disconnected client
INFO: End of ConnectedClients.updateClients()