# Instant Messaging

Team - G


Amruta Chavan agc9066

Anjali Pachpute avp9145

# What is Instant Messaging ?

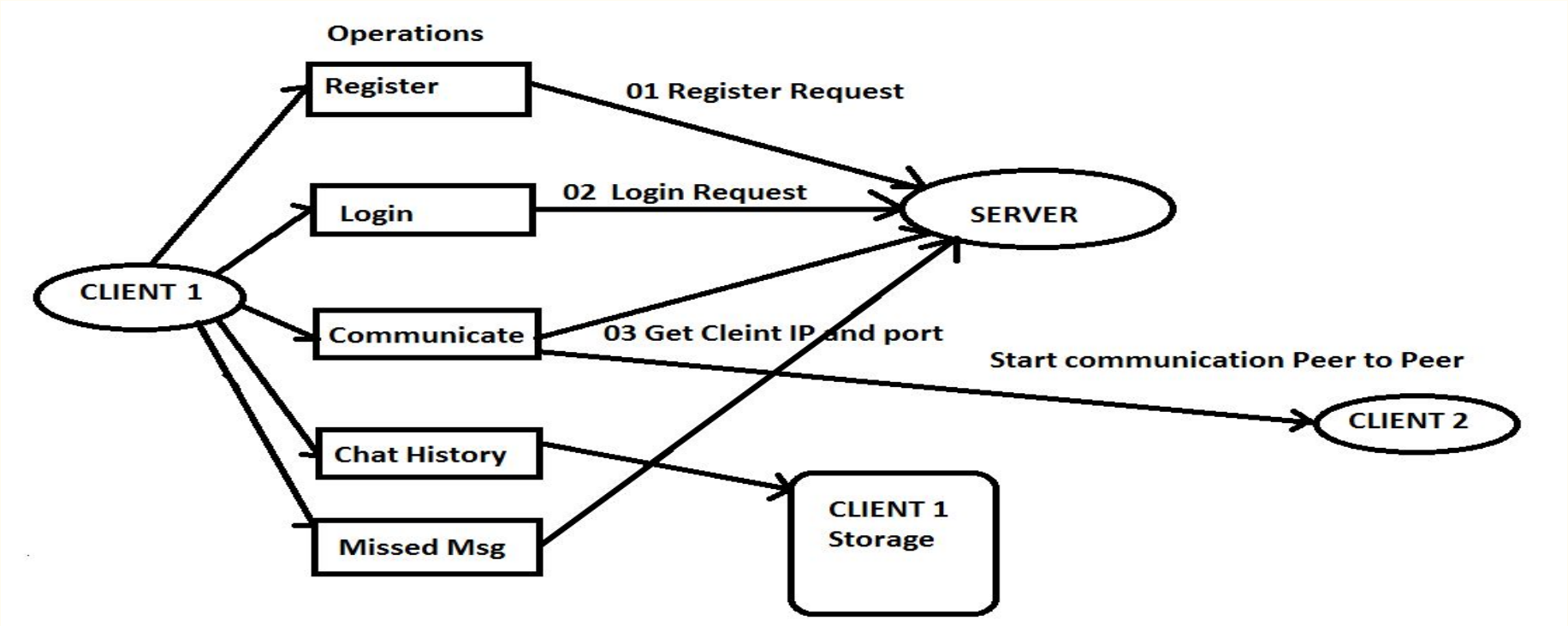
- It is an exchange of text messages between two or more people instantly
- One can easily find out whether the person he wants to communicate with is online or offline
- Communication is done instantly between online users
- Offline users receive messages once they come online
- It is a very fast and easy mode of communication

# Purpose/Usefulness of the Project

- To understand the concepts of networking
- To make communication fast and easy between two online users
- With Instant Messaging one can contact instantly with their friends

| **Features of Instant Messaging** | **Features implemented** |
| --- | --- |
| Reliability | Used **TCP** to implement Reliability |
| Instant messages between people who are online | Peer to Peer communication between multiple clients who are online |
| Scalability | **Multithreading** |
| Offline users receive messages when online | Buffering of messages for offline users |
| Chat History | Writing messages to a file to maintain chat history |

# Architecture



# Message Format

| | |
|---|---|
| msgId | This field is used to store the message Id.<br>Each message has a message ID which will help the server or client to identify the type of message received. |
| userName | userName field is used to store the client's userName |
| passWord | passWord field is used to store the client's password |
| IP | IP field is used to store the client's IP address |
| port | port is used to store the client's port |
| ErrMsg | This field is basically a string field which is used to store the messages that are to be exchanged between the client and the server. |

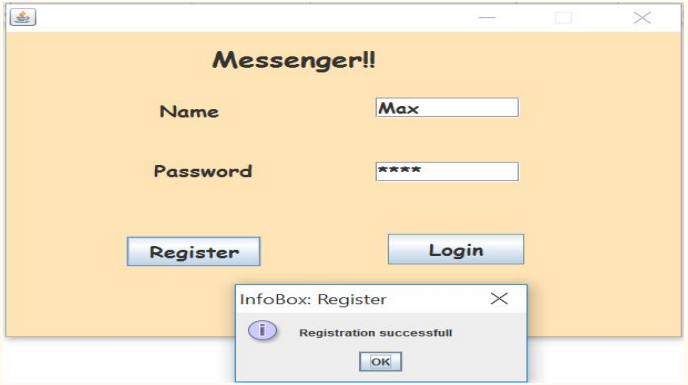
# Implementation Details

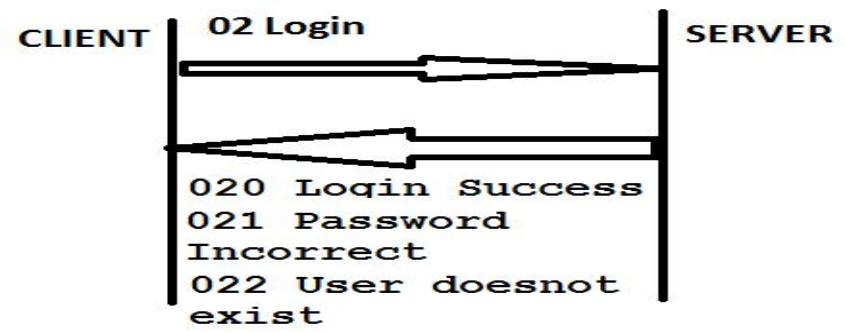
The most important feature of Instant Messaging is that it provides reliable transfer of messages.Messages are guaranteed to be delivered.

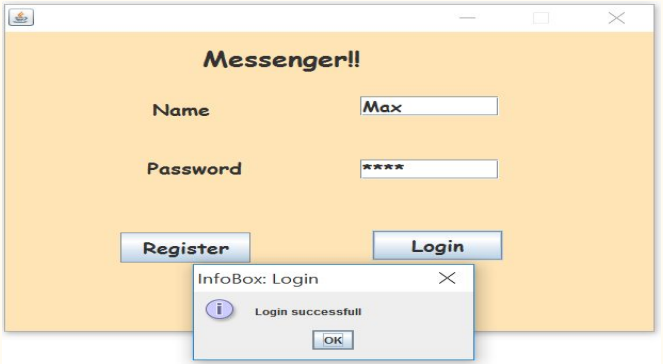
In order to provide this feature **TCP Protocol** was used.An Hybrid approach was used to implement an Instant Messenger.

1) **Client/Server Approach**
   A single multithreaded server accepts connections from multiple clients to login and register.

# Register and Login

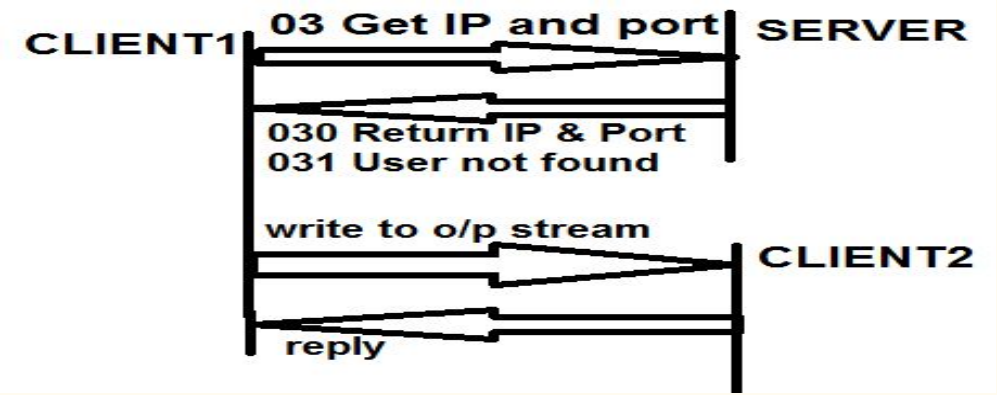
# Implementation Details (contd..)

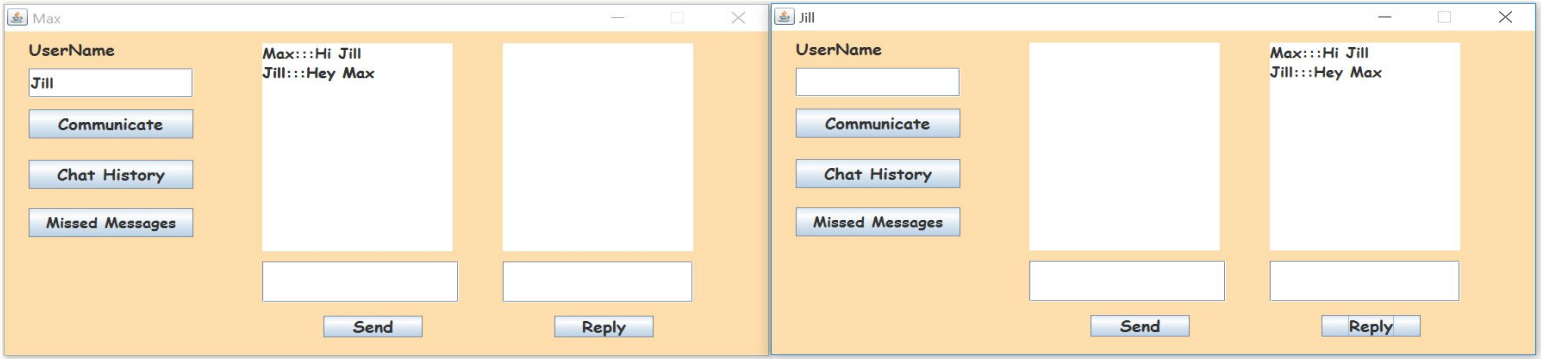
**2) Peer-to-Peer Approach**

Once a client registers or logs in, he can send a request to server to communicate with a friend by providing the friends username.

If the username is registered at the server, server returns the IP address of the user. After receiving the IP address, client starts a peer-to-peer communication with his friend.

# Peer-to-Peer Communication

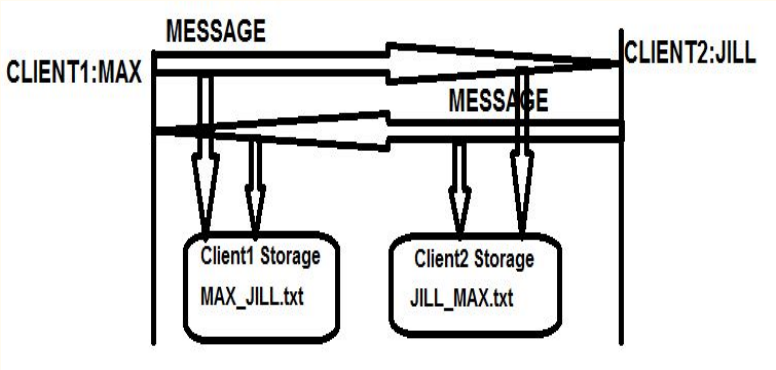


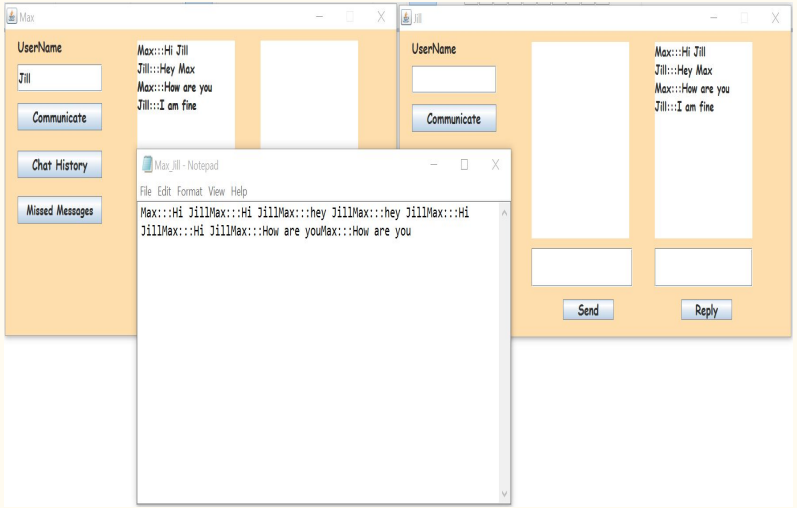



# Implementation Details(contd..)

A client has an option of looking at missed messages(Messages received when he was offline) and check the chat history(Log of chat) with a particular friend.

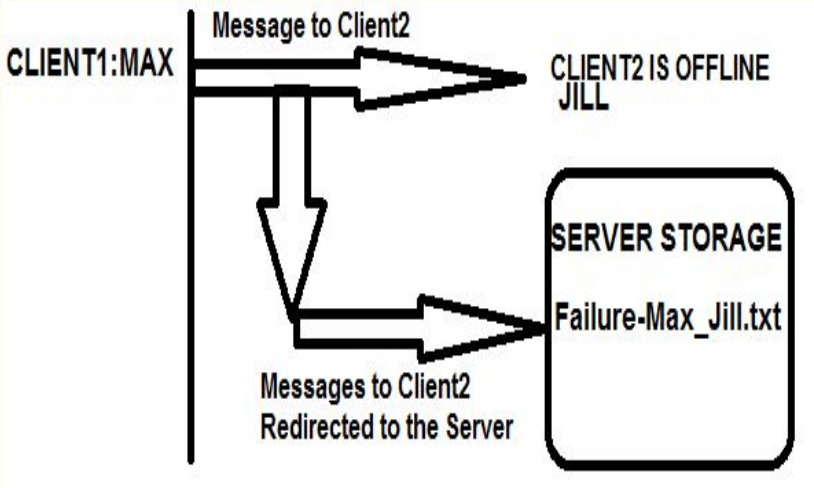
## Chat History

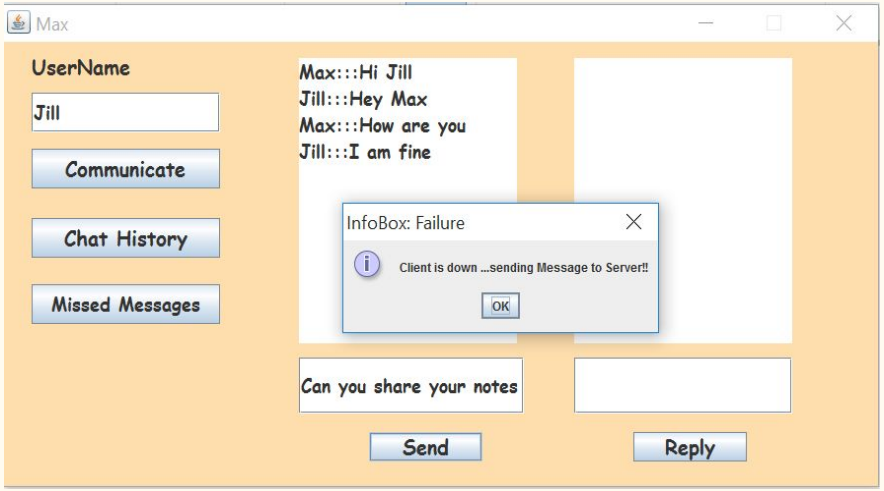


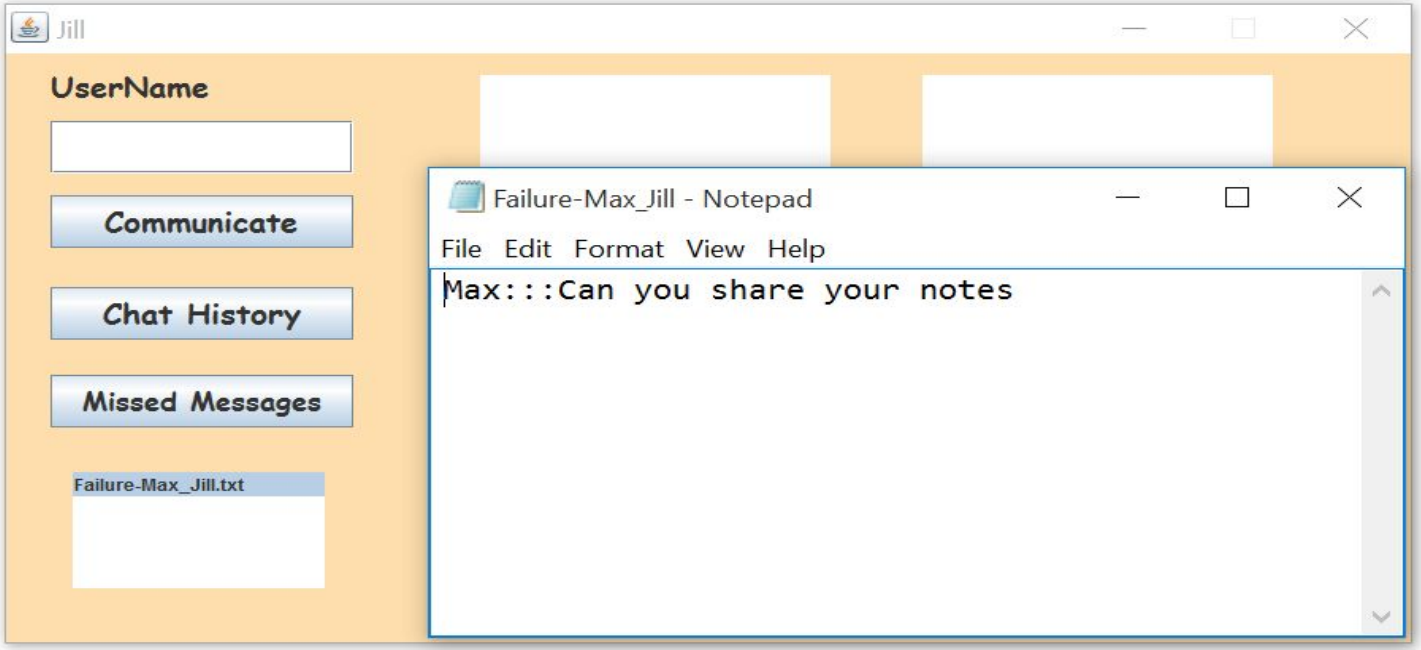
# Implementation Details(contd..)

## Missed Messages:



# Missed Messages(contd..)



# Difficulties Encountered so far

- Deciding the approach and functionalities to implement.
- Had issues dealing with multithreading while using command line interface

# How were the difficulties resolved

- To handle multithreading GUI was implemented

# DEMO

Thank you