CLASH OF CLIENTS

- **PROPOSED APPLICATION**

The application is regarding a game that is to be played between 6 clients(player) and one server(host) using socket programming. The protocol followed in this TCP. TCP is chosen because we need to create 6 connections that are persistent till they get eliminated which may not be possible using UDP.

- CONNECTION ESTABLISHMENT **:**

Server:

TCP ServerSocket is first created at port "9876". This creates a Server with the local host ip address available at 9876 port number.

ss.accept() is called in a infinite loop, which runs until 6 clients are connected. A count variable is used to maintain the number of clients connected so far.

Client:

Socket object is created at the client which connects with Local Host and port number 9876.This will create a socket at localhost connecting to localhost with 9876 port number that is our server.

Six such clients are created.

- GAME RULES AND CONDITIONS :

The game begins after running 1 server and 6 clients. Two clients are teamed up. Hence there will be 3 teams.

In total there will be three rounds.

Round 1: Fastest Fingers First

Rules of Round 1:

- Question will be regarding sorting of numbers
- The fastest two answers will be taken into consideration and the remaining team is terminated
- Only the leader of the team is supposed to send answer

NOTE: In this round we dont have to enter anything from the command prompt. It is automatically done in the Client side code.We just have to wait until one of the pair is disconnected

Round 2: Online Quiz

Rules of Round 2:

- Three questions will be asked to each of the teams alternativel
- Each correct answer will be awarded 10 points
- No points will be awarded for wrong answer
- In case of a draw, musical chair game will be played (Random number is generated and that team is removed). The team that lost the game will be diconnected.

NOTE: In this round the user has to type input from the command prompt where it is mentioned as "Team Leader please type your input".The answer is case INSENSITIVE

Round 3: Kitne Prathishat

Rules of Round 3

- Single Question will be asked to both the members simultaneously
- The questions are related to percentages(%)
- Three such questions will be asked
- Calculation of marks:
  - Difference between actual % and answer given is calculated for each question and added
  - The Client with minimum score is declared as the WINNER!!

NOTE: In this round also user has to type input from both the command prompts.If the input is not integer then score will be added with 100points(as client with minimum points is WINNER)

- **SYSTEM REQUIREMENTS**

➢ HARDWARE REQUIREMENTS :

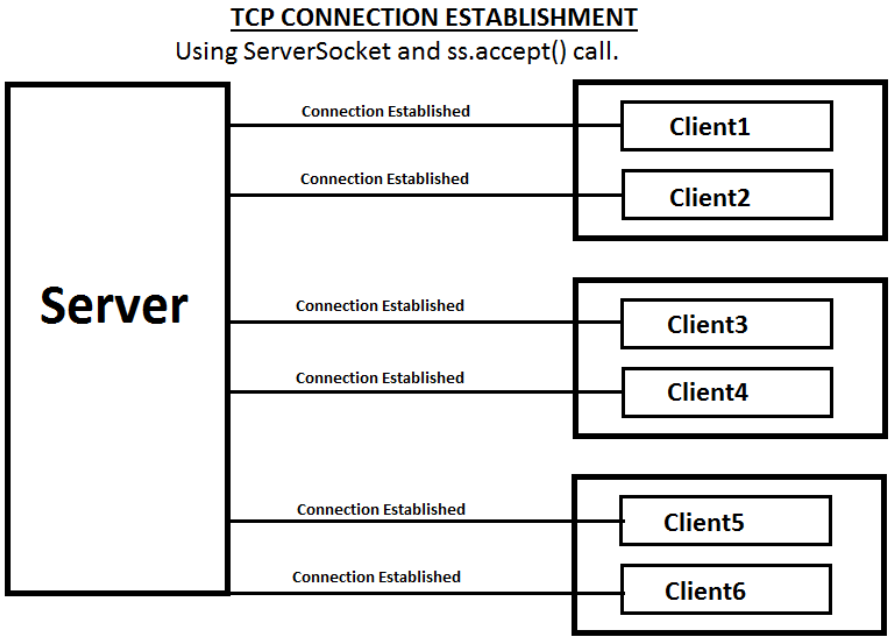
PROCESSOR : i3
RAM : 4 GB

➢ SOFTWARE REQUIREMENTS:

OPERATING SYSTEM : Windows(64 bit)
CODING LANGUAGE : Java

- **APPLICATION DESIGN**

- ARCHITECTURE:



An object of SocketServer is created at the Server end that creates a new socket which listens and binds to the specified port. The port number mentioned in our application is 9876. Once the socket is created it starts listening on this port. It makes a blocking call accept() which blocks the further process until one client gets connected.

The accept() calls made 6 times so that 6 client connections are ensured. Once the connections are made the game begins. In between if there is a client that gets disconnected then the game will be closed by broadcasting a message **"**One of the clients got disconnected hence ending the show!!".
In order to broadcast message,the message is individually sent to each of the clients by looping over the ObjectOutputStreamsof the corresponding sockets.

Marshalling and UnMarshalling: In the it is obsereved that Serializable is not imlemented anywhere as the Object used for transmiting is "String" throughout the application and String class internally implements Serializable class. Hence there is no need to imlement it.

Seperate lists are maintained inordered to store the ObjectOutputStreams and ObjectInputSTreams because once owerwritten by any other ObjectOutputSTream object there is no way we can trace back the same object. Maintaining a seperate list ensures that the output stream object is not destroyed until there is close in the connection. Throughout the code the Input Stream and Output Stream objects are not closed. They are closed only when there is a disconnection.

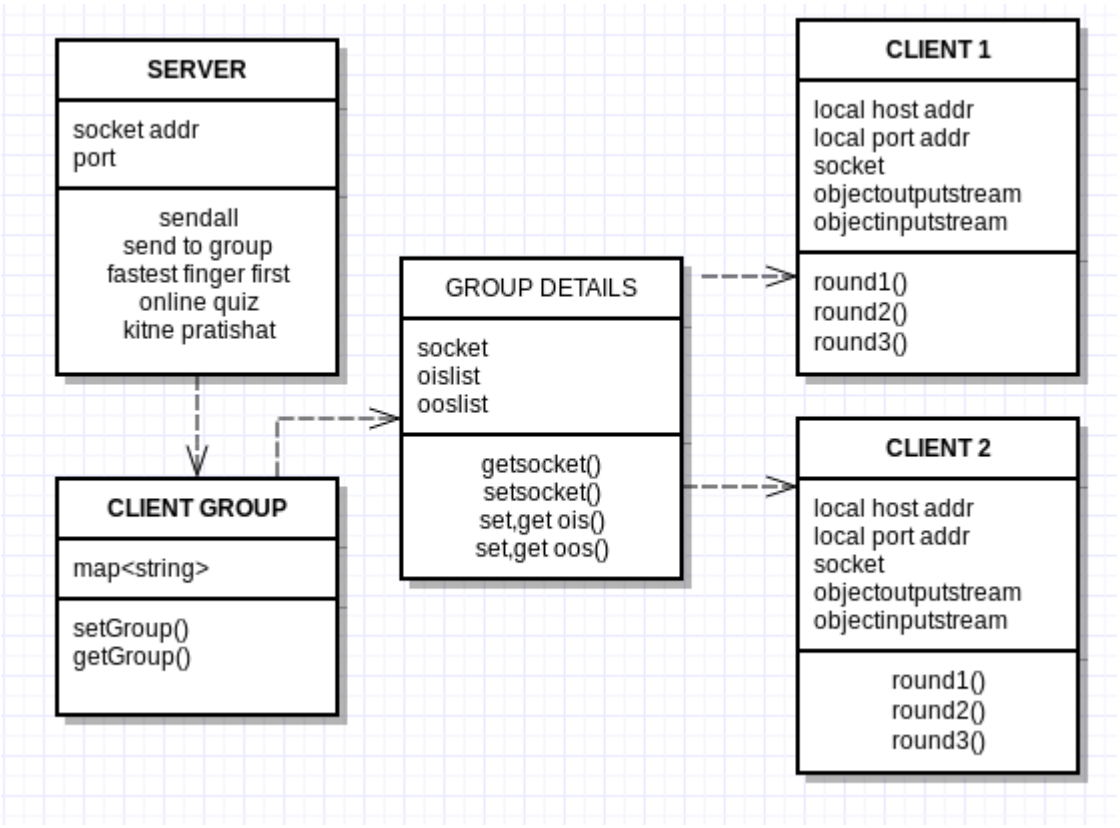
- **UML DIAGRAMS**:

Here we are including 4 UML diagrams to visualise,specify and document the construction and implementation our application “CLASH OF CLIENTS”.
The four uml diagrams here are:

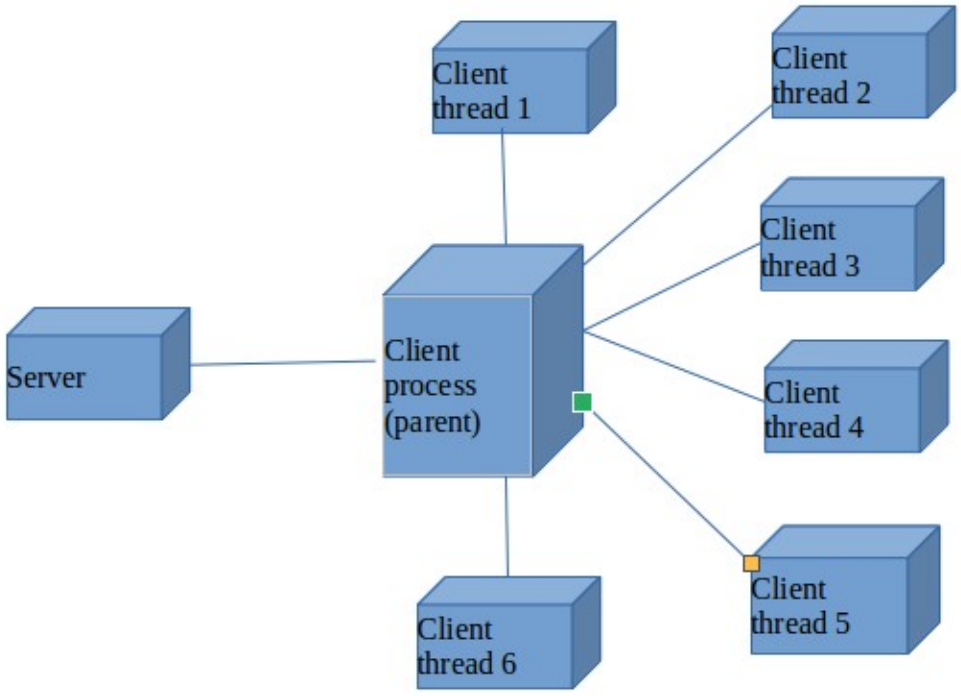
- Class diagrams
- Package diagrams
- Sequence diagram
- use case diagram

1. **Class Diagram**: It gives the static view of our application,by displaying classes, interfaces ,attributes and methods involved.

The below diagram is a simplified class diagram of our application, here in the diagram we are just representing only 2 clients instead of 6 clients.

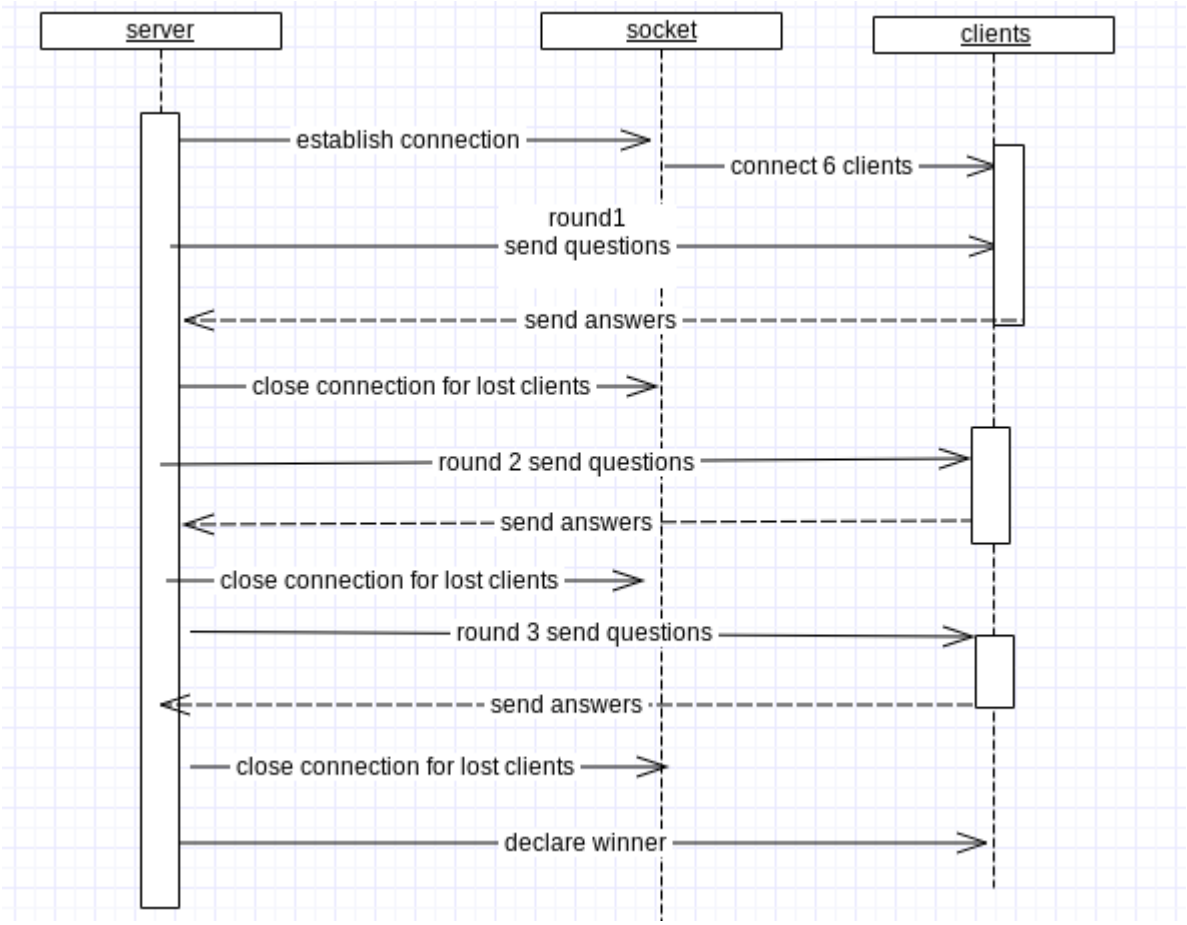


2. **Package Diagram**:



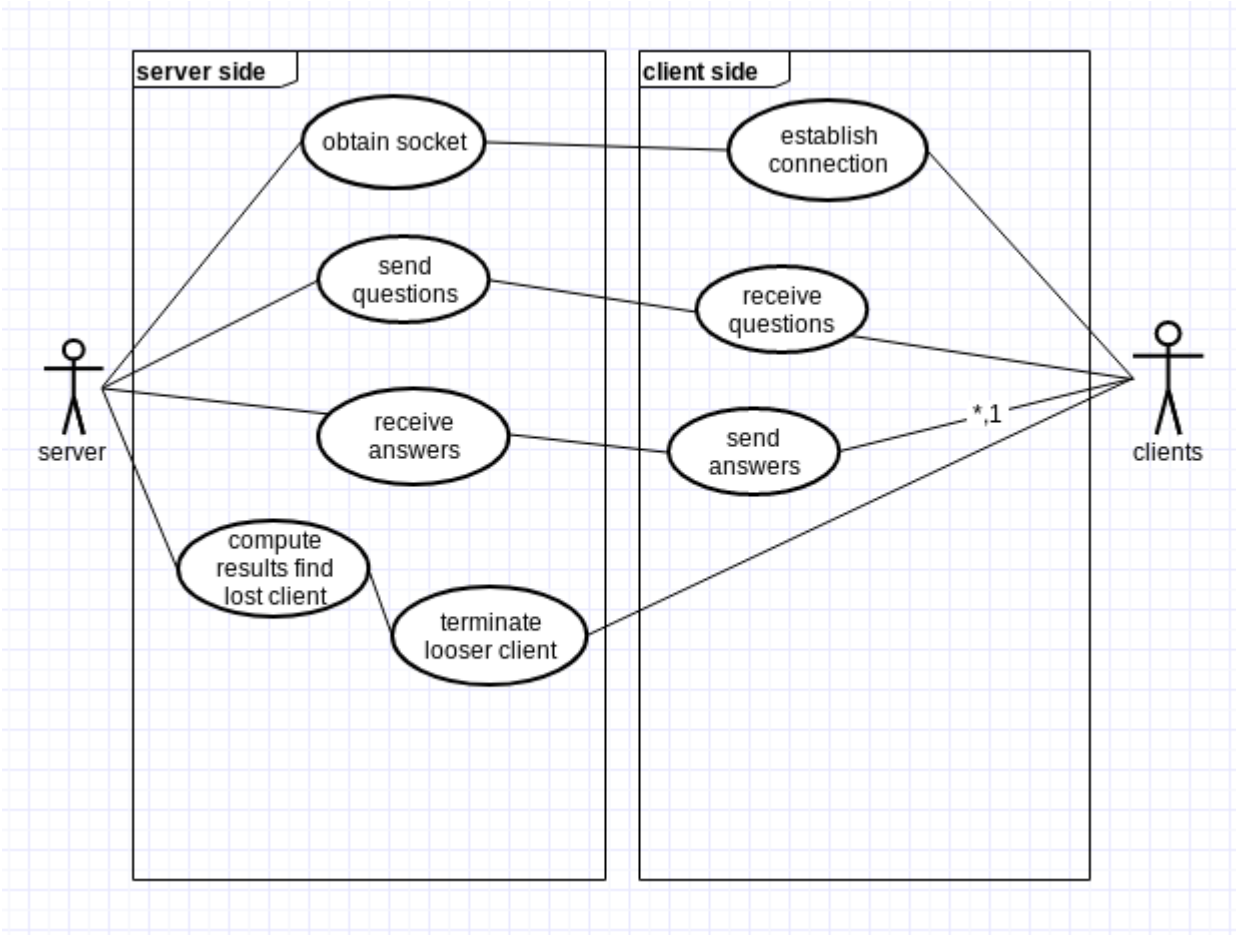
3. **Sequence Diagram**:

This UML diagram shows the sequence of operations carried out in the present CLASH OF CLIENTS application



4.Use-case Diagram:

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (**use** cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system



Exceptions Handled:

- If the clients are started without starting server:

  1. Message will be displayed saying "The Server is not yet started. Please Start the Server first and then the Clients"

- If Server stops ubruptly

  1. Message will be displayed on all Clients saying "There was some issue with the you or server. Hence closing the socket"

  2 . All connected Clients will be terminated

- If while making connections, if one of the clients gets disconnected:

  1. Message will be displayed on all Clients saying "One of the clients got disconnected hence ending the show!!"

  2 . All connected Clients will be terminated

- If in between the game one of the clients gets disconnected:

  1. Message will be displayed on all Clients saying "There was some issue with the you or server. Hence closing the socket"

  2. All connected Clients will be terminated

Future Enhancements:

- After winning, to establish **peer to peer** connection between the winner and the looser clients and make the Looser Client congratulate the winner client
- Catch block of IO Exceptions while closing the Server Socket is to be handled. Need to know when this situation arises.