***Abstract***

*Java Socket programming is used for communication between the applications running on different JRE.Java Socket programming can be connection-oriented or connection-less.*

*This project describes a very basic one-way Client and Server setup where a Client connects, sends messages to server and the server shows them using socket connection. There’s a lot of low-level stuff that needs to happen for these things to work but the Java API networking package (java.net) takes care of all of that, making network programming very easy for programmers.*

*TABLE OF CONTENTS*

*Abstract*

*List of Figures*

***CHAPTER 1 : INTRODUCTION***

*\* INTRODUCTION*

**CHAPTER 2: PROJECT REQUIREMENTS**

\* SOFTWARE HARDWARE REQUIREMENT

**CHAPTER3:PROBLEM STATEMENT**

**CHAPTER4:SYSTEM STUDY AND ANALYSIS**

\*FEASIBILITY STUDY

\*BEHAVIOURAL STUDY

**CHAPTER 5: SYSTEM DESIGN**

\*BLOCK DIAGRAM

**CHAPTER6: IMPLEMENTATION**

\*MODULES,LIBRARIES,PACKAGES USED

**CHAPTER 7:TESTING**

\*ESTABLISHING THE CONNECTION

\*COMMUNICATION

\*CLOSING THE CONNECTION

**CHAPTER 8:EXECUTION AND IMPORTANT POINTS ABOUT THE PROJECT**

**CHAPTER 8:SCREENSHOTS**

**CHAPTER 9:CONCLUSION**

**CHAPTER 10:REFERENCE**

***CHAPTER 1 : INTRODUCTION***

This project describes a very basic one-way Client and Server setup where a Client connects, sends messages to server and the server shows them using socket connection. There’s a lot of low-level stuff that needs to happen for these things to work but the Java API networking package (java.net) takes care of all of that, making network programming very easy for programmers.

**CHAPTER 2: PROJECT REQUIREMENTS**

SOFTWARE HARDWARE REQUIREMENTS

We require the following softwares to be installed to run the java code .

1.jdk1.8.0\_144

2.jre1.8.0\_144

Current java version is J2SE 8

Where jdk is Java Development Kit

jre is Java Runtime Environment

**CHAPTER3:PROBLEM STATEMENT**

Client and Server setup where a Client connects, sends messages to server and the server shows them using socket connection.

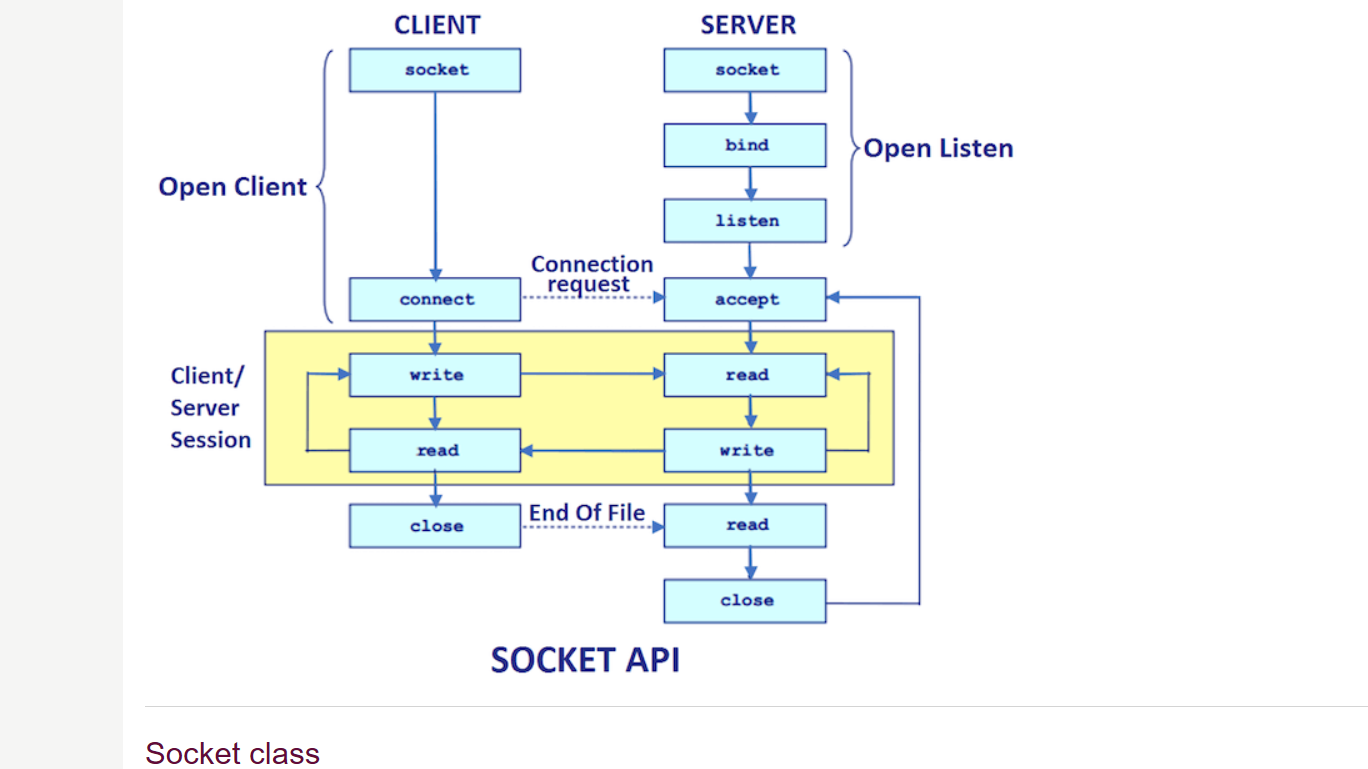
**CHAPTER4:SYSTEM STUDY AND ANALYSIS**

THIS PROJECT IS FEASIBLE BECAUSE WE ARE USING JAVA WHICH IS PLATFORM INDEPENDENT,ROBUST AND IT USES VERY SIMPLE PROGRAMMING TECHNIQUES

OUR PROJECT IS TO ESTABLISH A SOCKET CONNECTION BETWEEN CLIENT AND SERVER .ITS VERY EASY TO ANALYSE THE WAY IN WHICH THE CONNECTION IS ESTABLISHED .

**CHAPTER 5: SYSTEM DESIGN**

\*BLOCK DIAGRAM

******

**CHAPTER6: IMPLEMENTATION**

***MODULES,LIBRARIES,PACKAGES USED***

*Packages:*

Java API networking package (java.net)

java.io.\*

*Classes used:*

Server

Client

Socket

***CHAPTER 7:TESTING***

ESTABLISHING THE CONNECTION

To connect to other machine we need a socket connection. A socket connection means the two machines have information about each other’s network location (IP Address) and TCP port.The java.net.Socket class represents a Socket. To open a socket:

Socket socket = new Socket(“127.0.0.1”, 5000)

First argument – IP address of Server. ( 127.0.0.1 is the IP address of localhost, where code will run on single stand-alone machine).

Second argument – TCP Port. (Just a number representing which application to run on a server. For example, HTTP runs on port 80. Port number can be from 0 to 65535)

***Communication***

To communicate over a socket connection, streams are used to both input and output the data.

***Closing the connection***

The socket connection is closed explicitly once the message to server is sent.

In the program, Client keeps reading input from user and sends to the server until “Over” is typed.

***CHAPTER 8:EXECUTION AND IMPORTANT POINTS ABOUT THE PROJECT***

***EXECUTION:***

Open two windows one for Server and another for Client

1. First run the Server application as ,

$ java Server

Server started  
Waiting for a client …

2. Then run the Client application on another terminal as,

$ java Client

It will show – Connected and the server accepts the client and shows,

Client accepted

3. Then you can start typing messages in the Client window. Here is a sample input to the Client

Hello

I made my first socket connection

Over

Which the Server simultaneously receives and shows,

Hello

I made my first socket connection

Over

Closing connection

Sending “Over” closes the connection between the Client and the Server .

***IMPORTANT POINTS:***

Server application makes a ServerSocket on a specific port which is 5000. This starts our Server listening for client requests coming in for port 5000.

Then Server makes a new Socket to communicate with the client.

socket = server.accept()

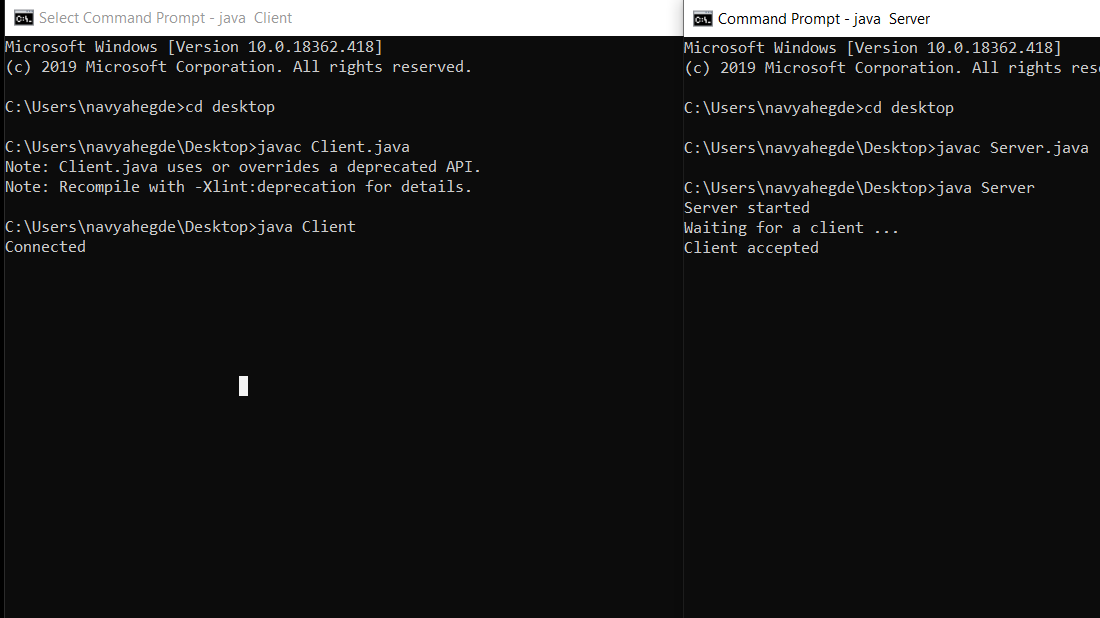
The accept() method blocks(just sits there) until a client connects to the server.

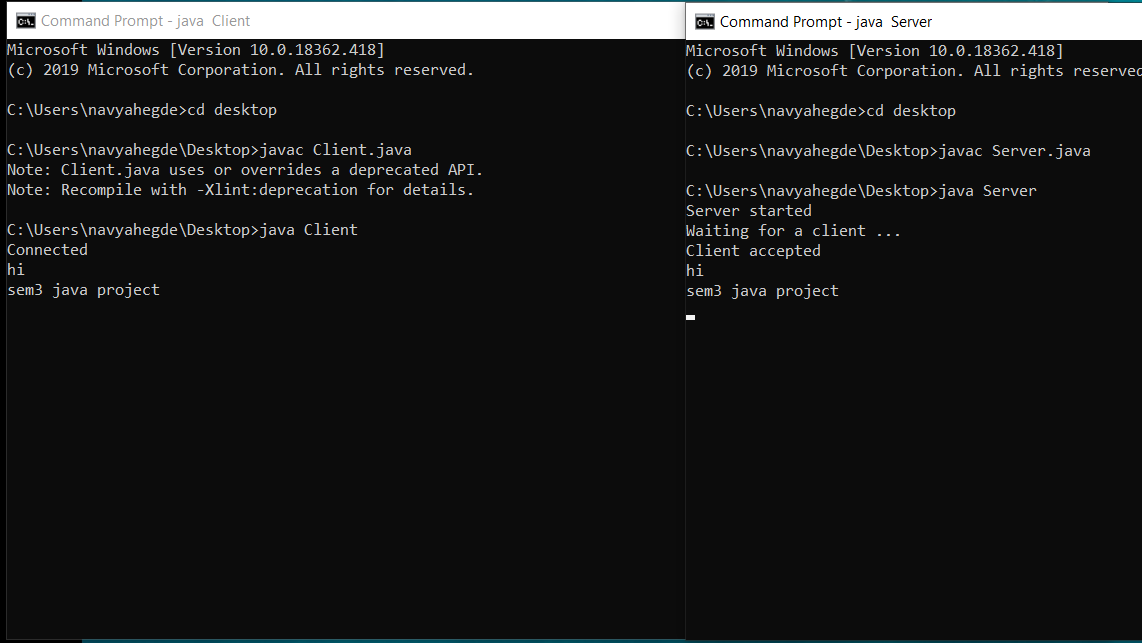
Then we take input from the socket using getInputStream() method. Our Server keeps receiving messages until the Client sends “Over”.

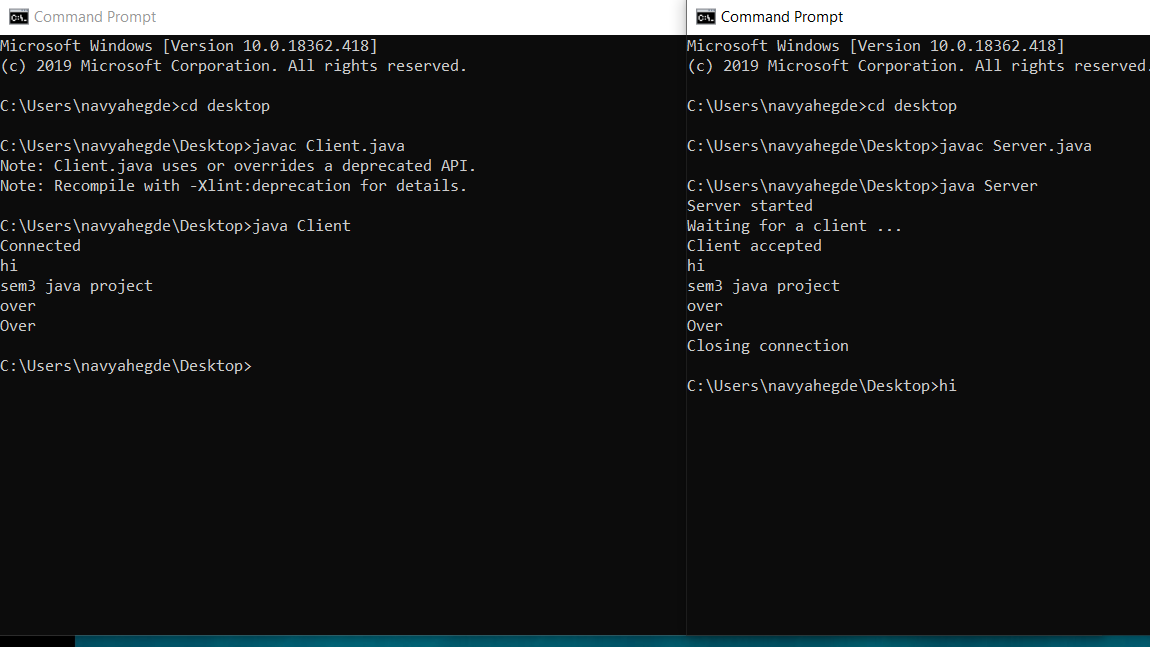
After we’re done we close the connection by closing the socket and the input stream.

To run the Client and Server application on your machine, compile both of them. Then first run the server application and then run the Client application.

***CHAPTER 9:SCREENSHOTS***







**CHAPTER 9:CONCLUSION**

**Thus we can conclude that we can connect a client to a server by using sockets in Java**

**CHAPTER 10:REFERENCE**

**We have made use of various internet sources,books and textbooks.**