A detailed report on

SECURE APPLICATION DEVELOPMENT

CPS 592-07

Assignment 2 - Secure and Robust Multi-Threaded Chat Server in Java

By

Dr.Phu phung

Submitted by

Venkateshwarlu Komuravelly

ID: 1015237060

Email: komuravellyv1@udayton.edu

Submitted Date: March 07, 2018

Link to my Bitbucket

https://bitbucket.org/komuravellyv1/venky2018sec-private/src/5cbcbe12c4ba80e5ac5f93611649f0e157aee653/assignments/assignment2/?at=master

2) Implementation and source code

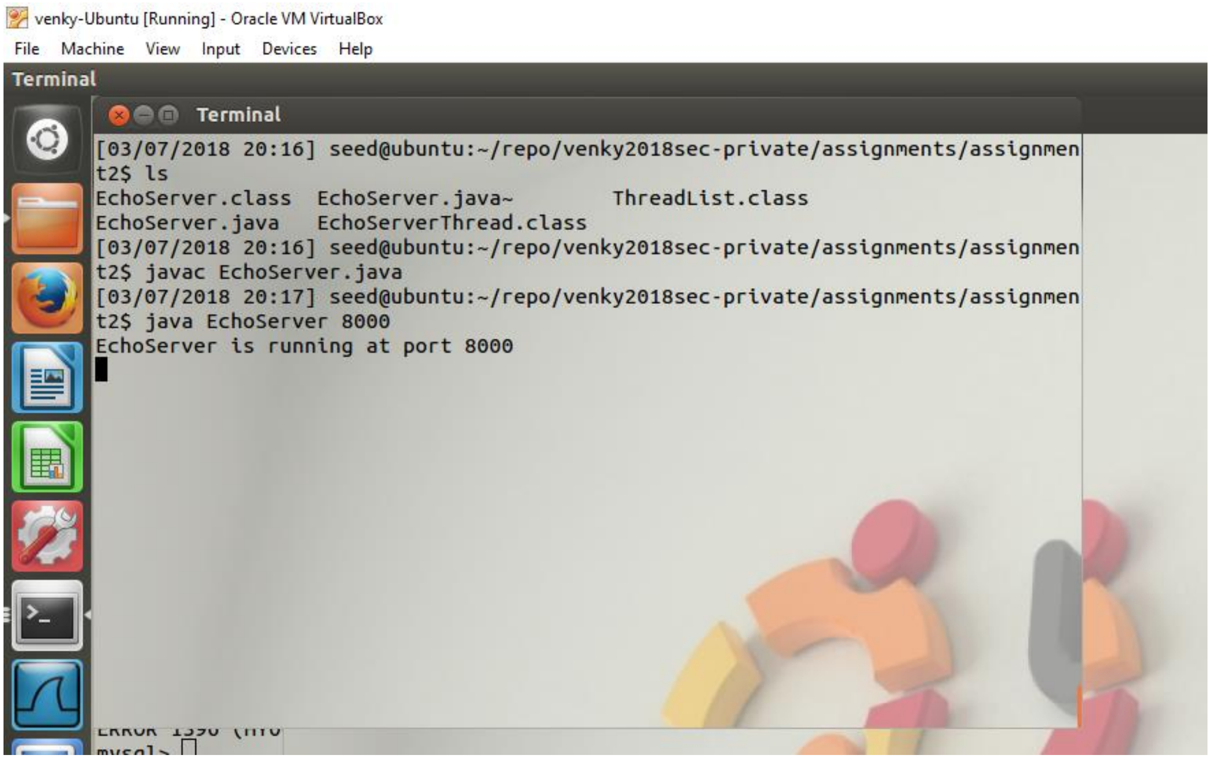
```
import java.net.*;
import java.io.*;
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.Set;
public class EchoServer {
static ThreadList threadlist = new ThreadList();
  public static void main(String[] args) throws IOException {
        if (args.length != 1) {
            System.err.println("Usage: java EchoServer <port number>");
            System.exit(1);
        }
//Check Input Validation Here.
        if(Integer.parseInt(args[0]) > 0) {
        int portNumber = Integer.parseInt(args[0]);

        try {
            ServerSocket serverSocket =   new
ServerSocket(Integer.parseInt(args[0]));
            System.out.println("EchoServer is running at port " +
Integer.parseInt(args[0]));

       while(true){

            Socket clientSocket = serverSocket.accept();
            System.out.println("A client is connected ");

       EchoServerThread newthread = new EchoServerThread(threadlist,clientSocket);
       threadlist.addThread(newthread);
       newthread.start();

    }
        }

       catch (IOException e) {
            System.out.println("Exception caught when trying to listen on port "
                + portNumber + " or listening for a connection");
            System.out.println(e.getMessage());
        }
    }
} }
class EchoServerThread extends Thread{
       private Socket clientSocket = null;
       private ThreadList threadlist = null;
```

```
        private PrintWriter out = null;
        private BufferedReader in = null;
        private String newusername;
        static ArrayList<String> thread_list = new ArrayList<String>();
        //***************        Constructors       ********************
        public EchoServerThread(Socket socket){
              clientSocket = socket;
        }
        public EchoServerThread(ThreadList threadlist, Socket socket){
        clientSocket = socket;
        this.threadlist = threadlist;
        }
        public EchoServerThread(String newusername) {
              this.newusername = newusername;
        }
        public void send(String message) {
        if (out!=null)
        out.println(message);
        }
        public synchronized void addListofUsers(String newusername) {
              thread_list.add(newusername);
        }
        public synchronized void getListofUsers() {
              for(int i=0;i< thread_list.size();i++) {
                    send(thread_list.get(i));
              }
              }
        public void run(){
        System.out.println("A new thread for client is running");

  //Hashtable to store usernames and passwords. It is Thread Safe.

        Hashtable<String, String> hashtable = new Hashtable<String, String>();
                    hashtable.put("Venkat","venk@03");
                    hashtable.put("Phu","Dayton1");
                    hashtable.put("Yesh", "Secure2");
                    hashtable.put("Jack", "Secure3");
                    hashtable.put("Matt", "Secure4");
        if(threadlist!=null)
        System.out.println("Inside thread:total clients: " +
  threadlist.getNumberofThreads());
        try{
          out = new PrintWriter(clientSocket.getOutputStream(), true);
       in = new BufferedReader(new
  InputStreamReader(clientSocket.getInputStream()));
       String inputLine;
        if(threadlist!=null){
        threadlist.sendToAll("the no of connected clients- "+
  threadlist.getNumberofThreads());
        }
             while ((inputLine = in.readLine()) != null) {
              String command = getCommand(inputLine);
              while(command.equals("<Join>")) {
                          String newusername = parseUsername(inputLine);
                          this.newusername = newusername;
                          String pass = parsePassword(inputLine);
                          Set<String> keys = hashtable.keySet();
                          while(keys.contains(newusername)) {
                                String value = hashtable.get(newusername);
```

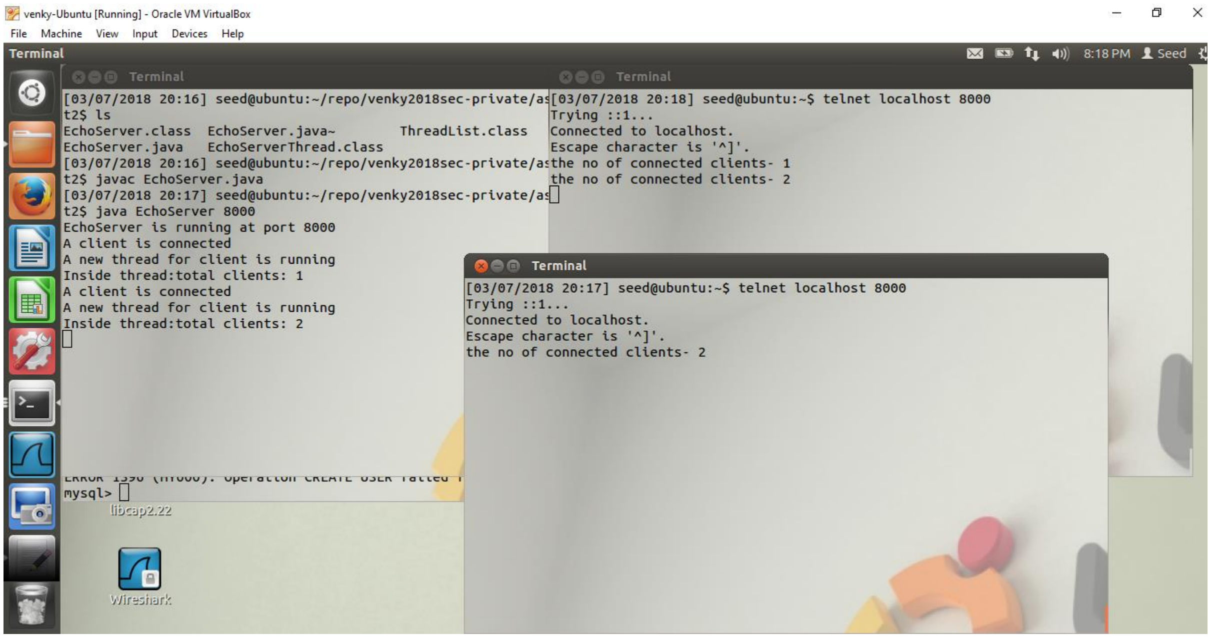
```
      if(value.equals(pass)) {
       send("Hi "+ newusername +" Welcome to chat room!");
       addListofUsers(newusername);
    threadlist.sendToAll("To All <new message>:"+ newusername + " Joined");

send("************************************");
send("Type <List> To get list of users");
send("Type <Exit> To exit Chat room");
send("Type <Chat>Message -To chat with Everyone");
send("Type <Priv>Private_user:Message -To chat privately");

send("************************************");
 break;
}else {
send("Invalid username:password"); break;
                                                    }
                                    }
            break;        }
    if(command.equals("<Chat>")) {
        String str = parseStringMessage(inputLine);
        threadlist.sendToAll("To All <Chat Message>"+ str);
      }
    else if(command.equals("<Priv>")) {
             parsePrivateMessage(inputLine);
      }
      else if(inputLine.equals("<Exit>")){
threadlist.sendToAll("To All: A client exists, the number of connected client:" +
(threadlist.getNumberofThreads()-1));
                          thread_List.remove(newusername);
                          threadlist.removeThread(this);
                          send("Updated Users List: ");
                          getListofUsers();
                          clientSocket.close();

                    }
      else if(inputLine.equals("<List>")){
      send("The List of users in the chat room:");
      getListofUsers();
    }
      }
      }
      catch (IOException ioe) {
            System.out.println("Exception caught is " + ioe.getMessage());
       }
            }

private String getCommand(String data) {
      if(data.isEmpty() || (data.length()<6))
             return "UNKNOWN";
             try {
                   String command = data.substring(0, 6).trim();
                   return command;
             }catch(Exception e) {
                   return "UNKNOWN";
             }
      }
private String parseStringMessage(String logindata) {
      String s = logindata.substring(6);
      return s;
    }
    private void parsePrivateMessage(String logindata) {
           String s = logindata.substring(6);
           String[] pmsg = s.split(":");
           String p_user = pmsg[0];
           String p_msg = pmsg[1];
           threadlist.sendPrivate(newusername, p_user, p_msg);
    }

              private String parseUsername(String logindata){
                String s = logindata.substring(6);
                String[] user = s.split(":"); // user array of type String to store
new username.
                 return user[0];
              }
              private String parsePassword(String logindata){
                String st = logindata.substring(6);
                String[] pass = st.split(":");
                 return pass[1];
              }
              public String getUserName() {
                return this.newusername;
              }
    }
    class ThreadList{
           //private ... threadlist = //store the list of threads in this variable
           private ArrayList<EchoServerThread> threadlist = new
    ArrayList<EchoServerThread>();

           public ThreadList()
           {
           }
           public  synchronized int getNumberofThreads(){
           //return the number of current threads
           return threadlist.size();
           }
           public synchronized void addThread(EchoServerThread newthread){
           //add the newthread object to the threadlist
           threadlist.add(newthread);
           }
           public synchronized void removeThread(EchoServerThread thread){
           //remove the given thread from the threadlist
           threadlist.remove(thread);
           }
           public synchronized void sendPrivate(String sender, String username, String
    message){
             for(EchoServerThread thread : threadlist){
                 if(thread.getUserName().equals(username)){
                     thread.send("<private> "+sender+ ":" +message);
                 }
             }
        }
           public synchronized void sendToAll(String message){
           Iterator<EchoServerThread> threadlistIterator = threadlist.iterator();
           while(threadlistIterator.hasNext()){
           EchoServerThread thread = threadlistIterator.next();
           thread.send(message);
           //ask each thread in the threadlist to send the given message to its client
        }}
}
```

3) Test Cases and Demo

Started Server at port 8000. And connected two clients.


venky-Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Terminal

```
[03/07/2018 20:16] seed@ubuntu:~/repo/venky2018sec-private/assignments/assignmen
t2$ ls
EchoServer.class  EchoServer.java~        ThreadList.class
EchoServer.java   EchoServerThread.class
[03/07/2018 20:16] seed@ubuntu:~/repo/venky2018sec-private/assignments/assignmen
t2$ javac EchoServer.java
[03/07/2018 20:17] seed@ubuntu:~/repo/venky2018sec-private/assignments/assignmen
t2$ java EchoServer 8000
EchoServer is running at port 8000
```


venky-Ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
8:18 PM
Seed
Terminal
Terminal
Terminal
libcap2.22
Wireshark

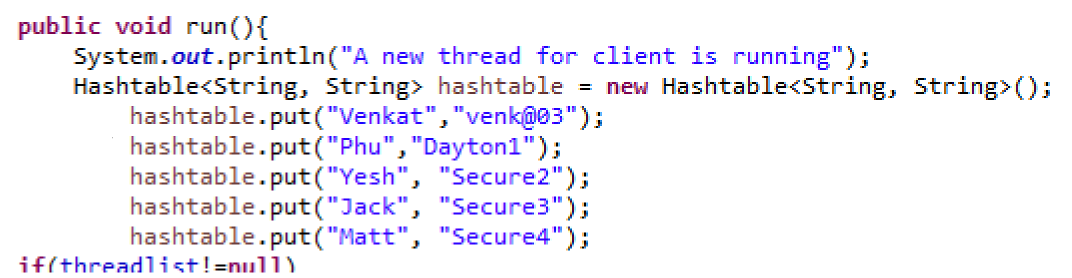
```
[03/07/2018 20:16] seed@ubuntu:~/repo/venky2018sec-private/as
t2$ ls
EchoServer.class  EchoServer.java~        ThreadList.class
EchoServer.java   EchoServerThread.class
[03/07/2018 20:16] seed@ubuntu:~/repo/venky2018sec-private/as
t2$ javac EchoServer.java
[03/07/2018 20:17] seed@ubuntu:~/repo/venky2018sec-private/as
t2$ java EchoServer 8000
EchoServer is running at port 8000
A client is connected
A new thread for client is running
Inside thread:total clients: 1
A client is connected
A new thread for client is running
Inside thread:total clients: 2
```

```
mysql>
```

```
[03/07/2018 20:18] seed@ubuntu:~$ telnet localhost 8000
Trying ::1...
Connected to localhost.
Escape character is '^]'.
the no of connected clients- 1
the no of connected clients- 2
```

```
[03/07/2018 20:17] seed@ubuntu:~$ telnet localhost 8000
Trying ::1...
Connected to localhost.
Escape character is '^]'.
the no of connected clients- 2
```

It Won't let to do anything until user enters chat room. Here In my code there are 5 users.

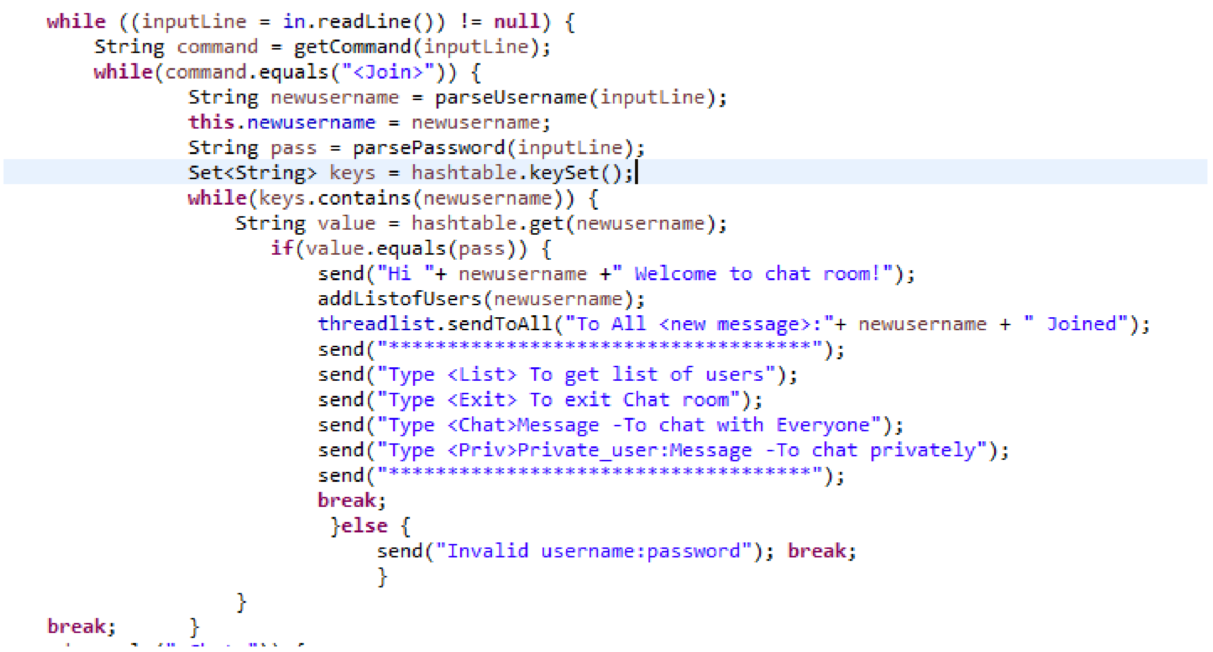


```
public void run(){
    System.out.println("A new thread for client is running");
    Hashtable<String, String> hashtable = new Hashtable<String, String>();
        hashtable.put("Venkat","venk@03");
        hashtable.put("Phu","Dayton1");
        hashtable.put("Yesh", "Secure2");
        hashtable.put("Jack", "Secure3");
        hashtable.put("Matt", "Secure4");
if(threadlist!=null)
```

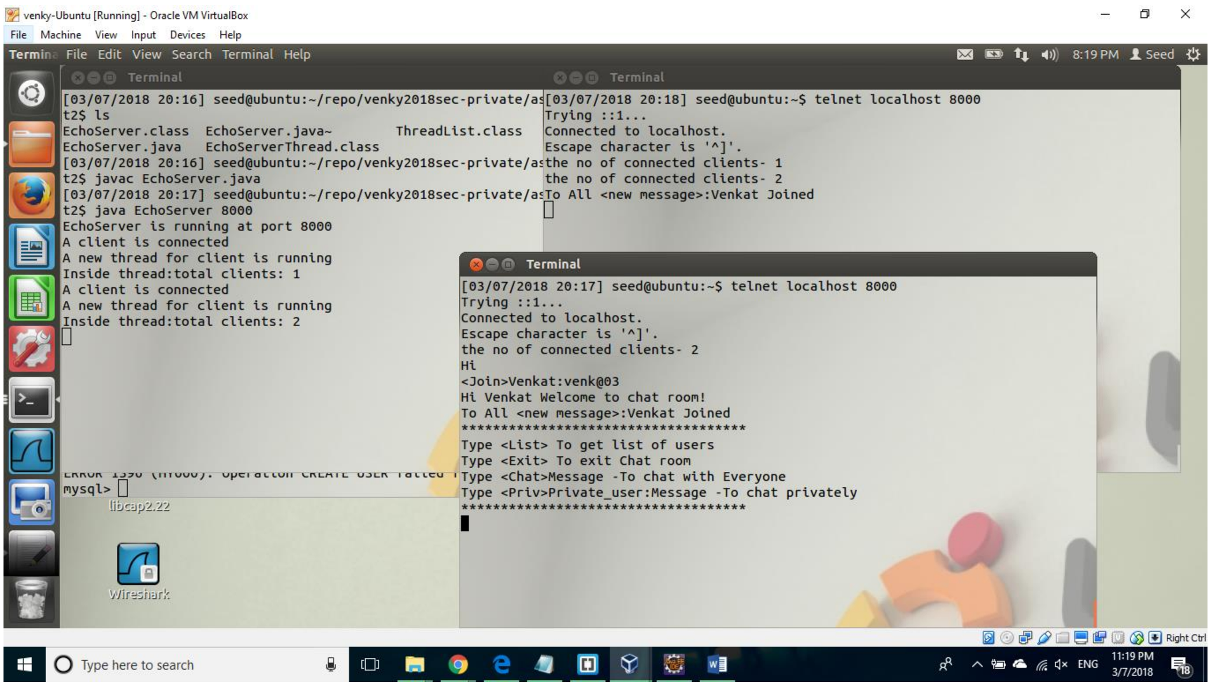
By Joining using <Join>username:password, Below things should happen.

1.Username and passwords should be matched with given.

2.Upon successful login it should show below commands.



```
  while ((inputLine = in.readLine()) != null) {
      String command = getCommand(inputLine);
      while(command.equals("<Join>")) {
              String newusername = parseUsername(inputLine);
              this.newusername = newusername;
              String pass = parsePassword(inputLine);
              Set<String> keys = hashtable.keySet();
              while(keys.contains(newusername)) {
                  String value = hashtable.get(newusername);
                     if(value.equals(pass)) {
                         send("Hi "+ newusername +" Welcome to chat room!");
                         addListofUsers(newusername);
                         threadlist.sendToAll("To All <new message>:"+ newusername + " Joined");
                         send("***********************************");
                         send("Type <List> To get list of users");
                         send("Type <Exit> To exit Chat room");
                         send("Type <Chat>Message -To chat with Everyone");
                         send("Type <Priv>Private_user:Message -To chat privately");
                         send("***********************************");
                         break;
                          }else {
                              send("Invalid username:password"); break;
                              }
                  }
  break;      }
```

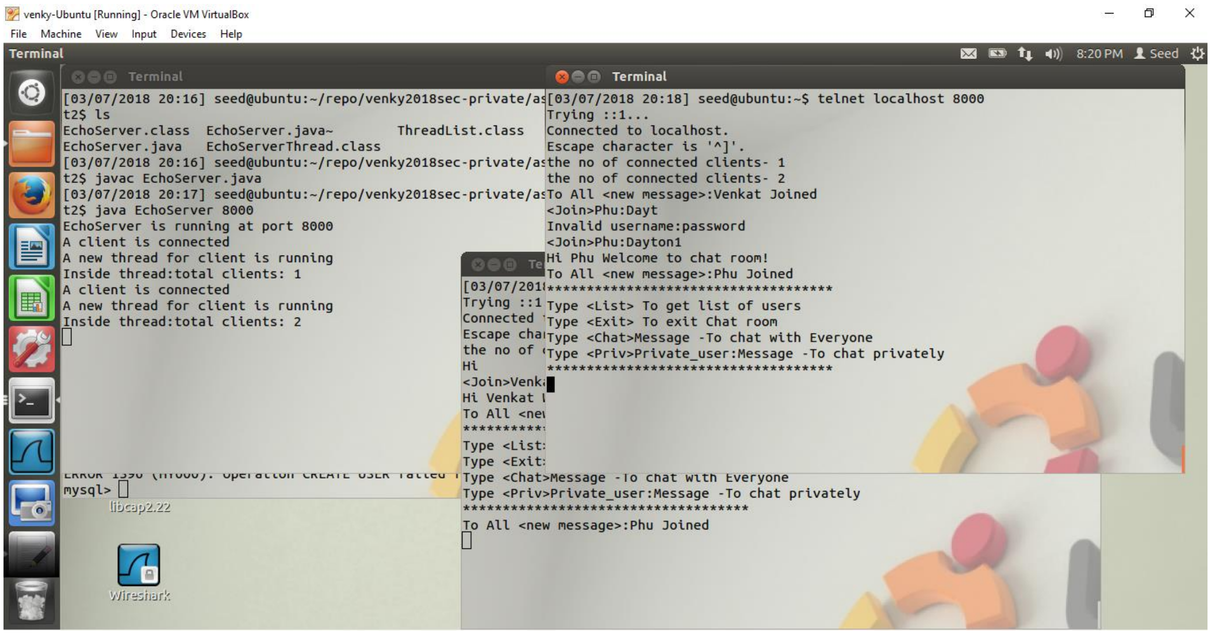


```
[03/07/2018 20:16] seed@ubuntu:~/repo/venky2018sec-private/as
t2$ ls
EchoServer.class  EchoServer.java~         ThreadList.class
EchoServer.java   EchoServerThread.class
[03/07/2018 20:16] seed@ubuntu:~/repo/venky2018sec-private/as
t2$ javac EchoServer.java
[03/07/2018 20:17] seed@ubuntu:~/repo/venky2018sec-private/as
t2$ java EchoServer 8000
EchoServer is running at port 8000
A client is connected
A new thread for client is running
Inside thread:total clients: 1
A client is connected
A new thread for client is running
Inside thread:total clients: 2
```

```
[03/07/2018 20:18] seed@ubuntu:~$ telnet localhost 8000
Trying ::1...
Connected to localhost.
Escape character is '^]'.
the no of connected clients- 1
the no of connected clients- 2
To All <new message>:Venkat Joined
```

```
[03/07/2018 20:17] seed@ubuntu:~$ telnet localhost 8000
Trying ::1...
Connected to localhost.
Escape character is '^]'.
the no of connected clients- 2
Hi
<Join>Venkat:venk@03
Hi Venkat Welcome to chat room!
To All <new message>:Venkat Joined
************************************
Type <List> To get list of users
Type <Exit> To exit Chat room
Type <Chat>Message -To chat with Everyone
Type <Priv>Private_user:Message -To chat privately
************************************
```

Venkat,Phu Two users joined chat room.

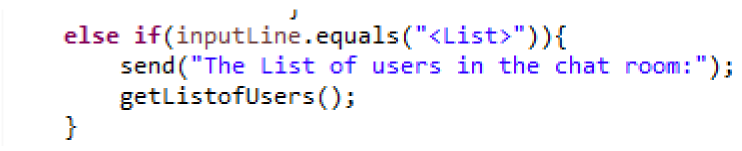


```
[03/07/2018 20:16] seed@ubuntu:~/repo/venky2018sec-private/as
t2$ ls
EchoServer.class  EchoServer.java~         ThreadList.class
EchoServer.java   EchoServerThread.class
[03/07/2018 20:16] seed@ubuntu:~/repo/venky2018sec-private/as
t2$ javac EchoServer.java
[03/07/2018 20:17] seed@ubuntu:~/repo/venky2018sec-private/as
t2$ java EchoServer 8000
EchoServer is running at port 8000
A client is connected
A new thread for client is running
Inside thread:total clients: 1
A client is connected
A new thread for client is running
Inside thread:total clients: 2
```

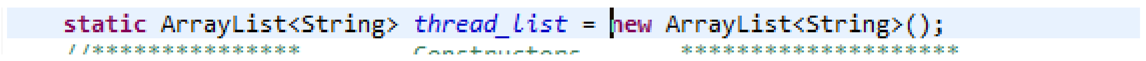
```
[03/07/2018 20:18] seed@ubuntu:~$ telnet localhost 8000
Trying ::1...
Connected to localhost.
Escape character is '^]'.
the no of connected clients- 1
the no of connected clients- 2
To All <new message>:Venkat Joined
<Join>Phu:Dayt
Invalid username:password
<Join>Phu:Dayton1
Hi Phu Welcome to chat room!
To All <new message>:Phu Joined
************************************
Type <List> To get list of users
Type <Exit> To exit Chat room
Type <Chat>Message -To chat with Everyone
Type <Priv>Private_user:Message -To chat privately
************************************
```

```
Type <Chat>Message -To chat with Everyone
Type <Priv>Private_user:Message -To chat privately
************************************
To All <new message>:Phu Joined
```

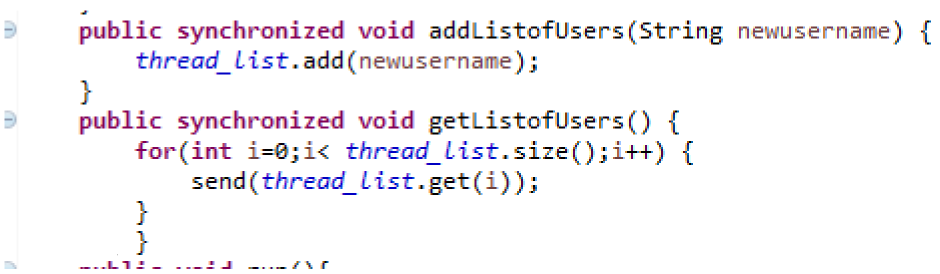
<List> should show list of users in chat room as below.



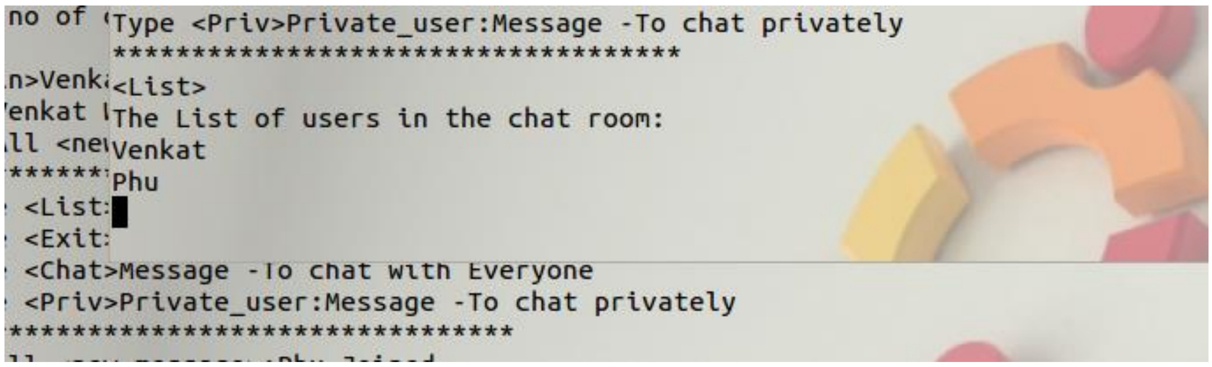
```
else if(inputLine.equals("<List>")){
    send("The List of users in the chat room:");
    getListofUsers();
}
```



```
static ArrayList<String> thread_list = new ArrayList<String>();
```

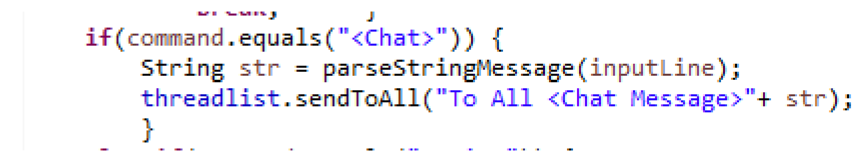


```
    public synchronized void addListofUsers(String newusername) {
        thread_list.add(newusername);
    }
    public synchronized void getListofUsers() {
        for(int i=0;i< thread_list.size();i++) {
            send(thread_list.get(i));
        }
    }
```

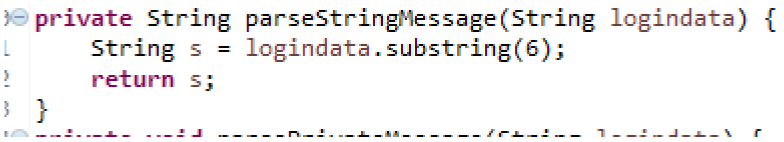


```
no of  Type <Priv>Private_user:Message -To chat privately
       ************************************
n>Venk <List>
enkat  The List of users in the chat room:
ll <ne Venkat
****** Phu
 <List:
 <Exit:
 <Chat>Message -To chat with Everyone
 <Priv>Private_user:Message -To chat privately
*********************************
```

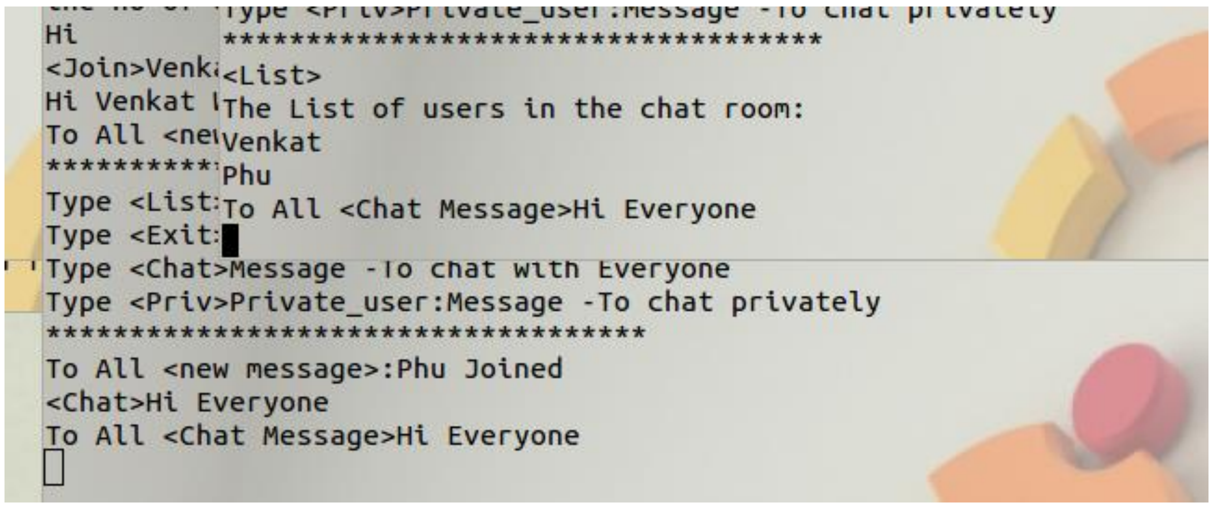
<Chat>message To chat with everyone as shown below.



```
    if(command.equals("<Chat>")) {
        String str = parseStringMessage(inputLine);
        threadlist.sendToAll("To All <Chat Message>"+ str);
    }
```

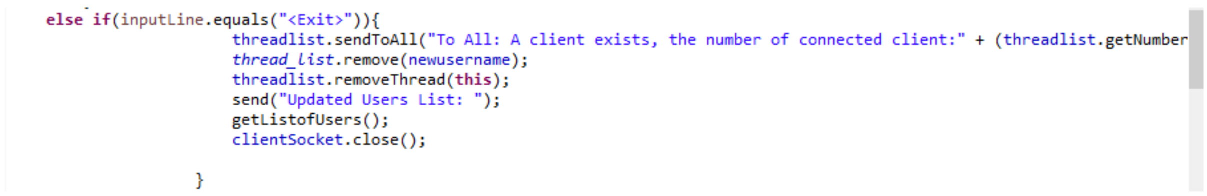


```
private String parseStringMessage(String logindata) {
    String s = logindata.substring(6);
    return s;
}
```

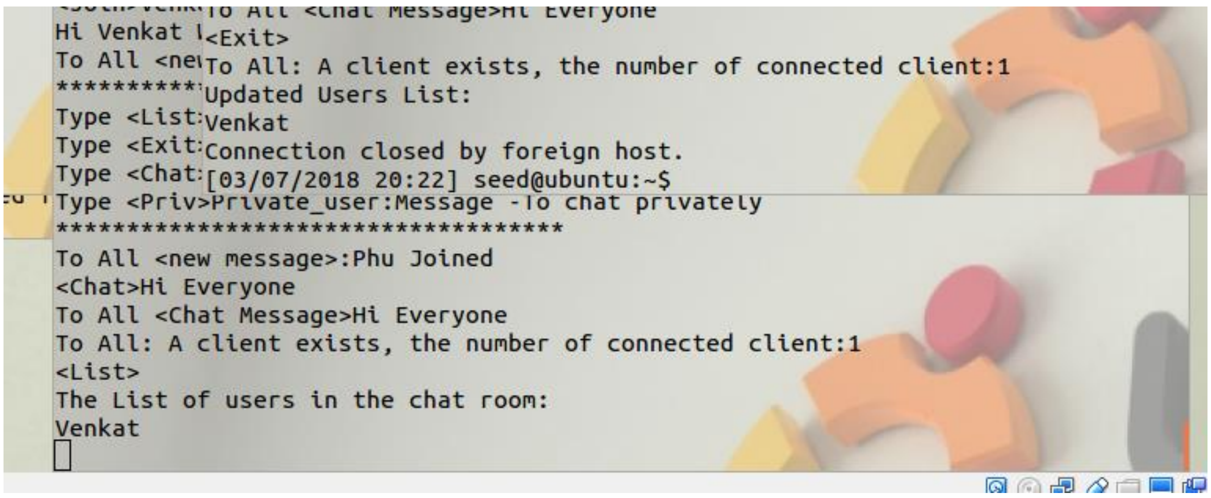


```
Hi          ************************************
<Join>Venk  <List>
Hi Venkat   The List of users in the chat room:
To All <ne  Venkat
**********  Phu
Type <List: To All <Chat Message>Hi Everyone
Type <Exit:
Type <Chat>Message -To chat with Everyone
Type <Priv>Private_user:Message -To chat privately
************************************
To All <new message>:Phu Joined
<Chat>Hi Everyone
To All <Chat Message>Hi Everyone
```

<Exit> to exit the chat room.

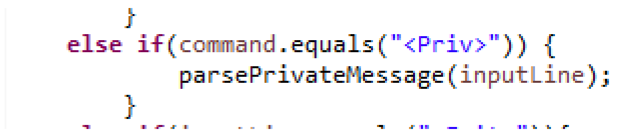


```
else if(inputLine.equals("<Exit>")){
                        threadlist.sendToAll("To All: A client exists, the number of connected client:" + (threadlist.getNumber
                        thread_list.remove(newusername);
                        threadlist.removeThread(this);
                        send("Updated Users List: ");
                        getListofUsers();
                        clientSocket.close();

                }
```

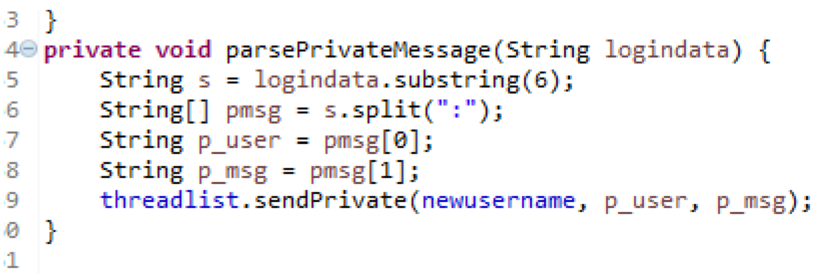


```
<Jon...>        To All <Chat Message>Hi Everyone
Hi Venkat l<Exit>
To All <ne To All: A client exists, the number of connected client:1
**********Updated Users List:
Type <List:Venkat
Type <Exit:Connection closed by foreign host.
Type <Chat:[03/07/2018 20:22] seed@ubuntu:~$
Type <Priv>Private_user:Message -To chat privately
************************************
To All <new message>:Phu Joined
<Chat>Hi Everyone
To All <Chat Message>Hi Everyone
To All: A client exists, the number of connected client:1
<List>
The List of users in the chat room:
Venkat
```

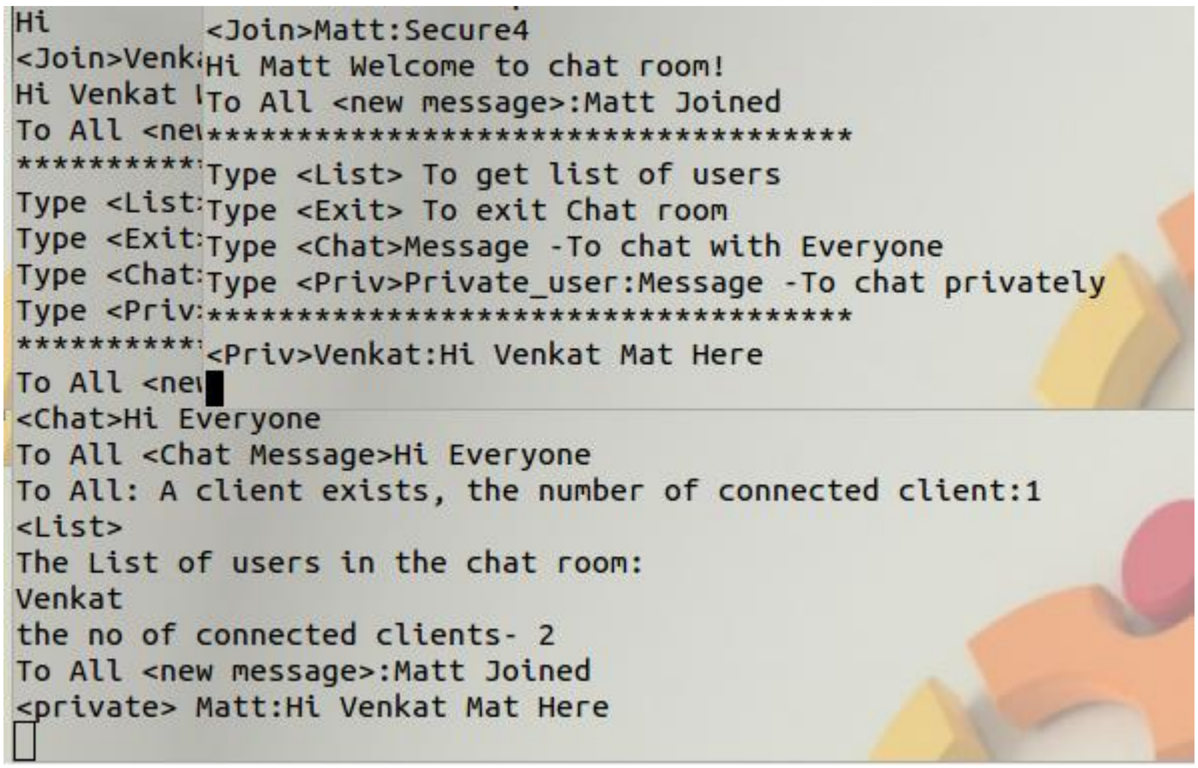
<Priv>Username:Message to send message privately.



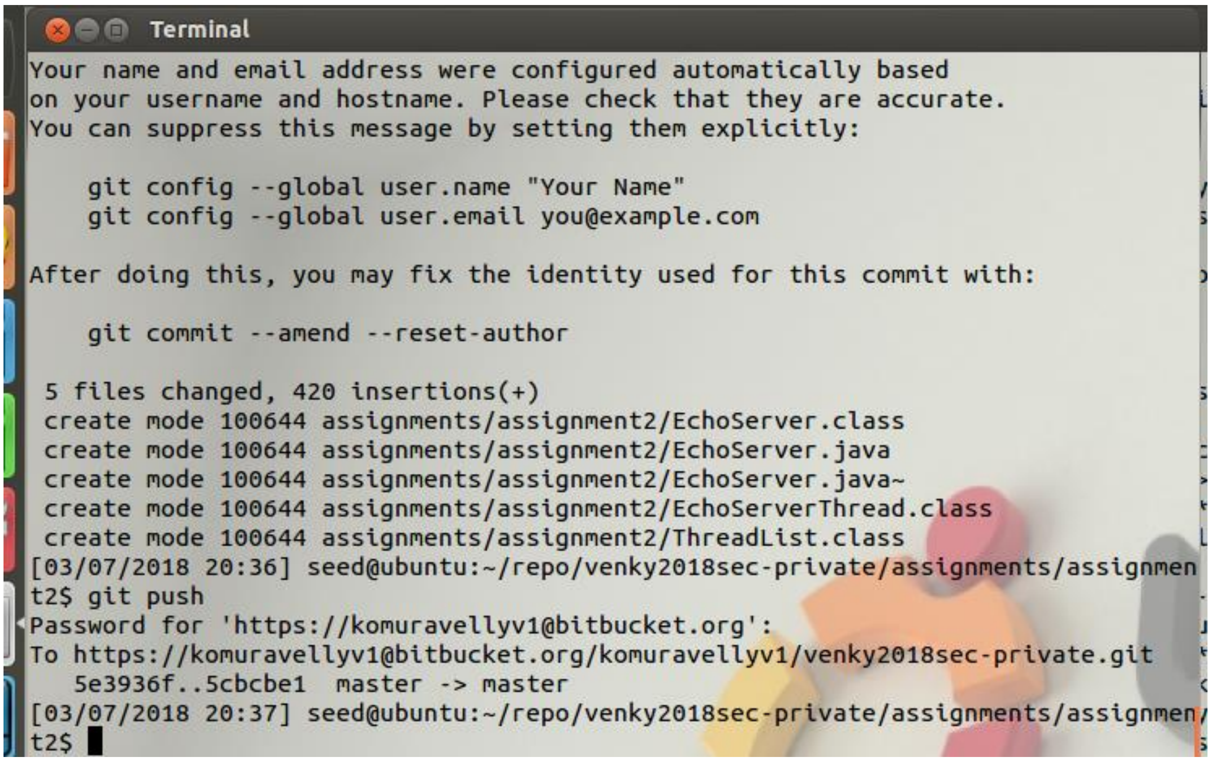
```
        }
    else if(command.equals("<Priv>")) {
            parsePrivateMessage(inputLine);
        }
```



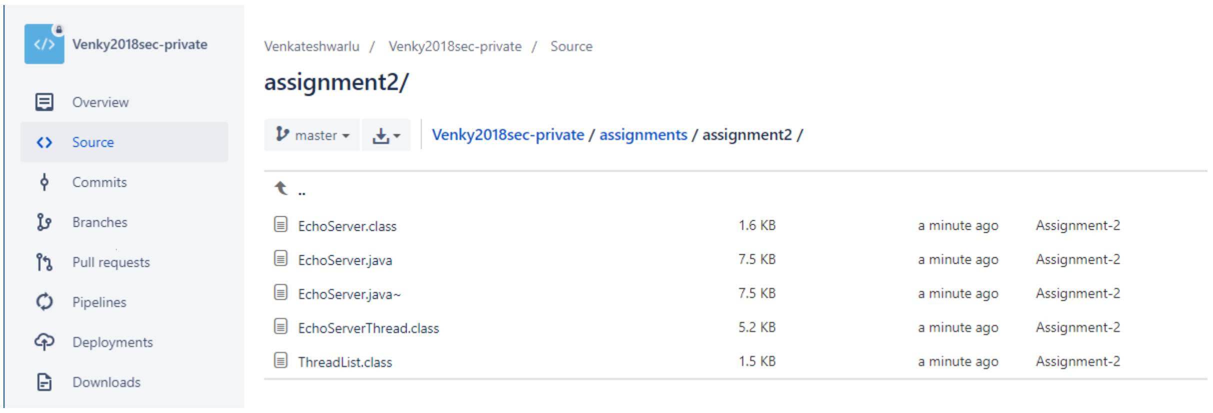
```
}
private void parsePrivateMessage(String logindata) {
    String s = logindata.substring(6);
    String[] pmsg = s.split(":");
    String p_user = pmsg[0];
    String p_msg = pmsg[1];
    threadlist.sendPrivate(newusername, p_user, p_msg);
}
```



```
Hi          <Join>Matt:Secure4
<Join>Venk  Hi Matt Welcome to chat room!
Hi Venkat   To All <new message>:Matt Joined
To All <ne  ************************************
**********  Type <List> To get list of users
Type <List: Type <Exit> To exit Chat room
Type <Exit: Type <Chat>Message -To chat with Everyone
Type <Chat: Type <Priv>Private_user:Message -To chat privately
Type <Priv: ************************************
**********  <Priv>Venkat:Hi Venkat Mat Here
To All <ne
<Chat>Hi Everyone
To All <Chat Message>Hi Everyone
To All: A client exists, the number of connected client:1
<List>
The List of users in the chat room:
Venkat
the no of connected clients- 2
To All <new message>:Matt Joined
<private> Matt:Hi Venkat Mat Here
```



```
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 5 files changed, 420 insertions(+)
 create mode 100644 assignments/assignment2/EchoServer.class
 create mode 100644 assignments/assignment2/EchoServer.java
 create mode 100644 assignments/assignment2/EchoServer.java~
 create mode 100644 assignments/assignment2/EchoServerThread.class
 create mode 100644 assignments/assignment2/ThreadList.class
[03/07/2018 20:36] seed@ubuntu:~/repo/venky2018sec-private/assignments/assignmen
t2$ git push
Password for 'https://komuravellyv1@bitbucket.org':
To https://komuravellyv1@bitbucket.org/komuravellyv1/venky2018sec-private.git
   5e3936f..5cbcbe1  master -> master
[03/07/2018 20:37] seed@ubuntu:~/repo/venky2018sec-private/assignments/assignmen
t2$
```


Venky2018sec-private

Overview
Source
Commits
Branches
Pull requests
Pipelines
Deployments
Downloads

Venkateshwarlu / Venky2018sec-private / Source

assignment2/

master Venky2018sec-private / assignments / assignment2 /

..			
EchoServer.class	1.6 KB	a minute ago	Assignment-2
EchoServer.java	7.5 KB	a minute ago	Assignment-2
EchoServer.java~	7.5 KB	a minute ago	Assignment-2
EchoServerThread.class	5.2 KB	a minute ago	Assignment-2
ThreadList.class	1.5 KB	a minute ago	Assignment-2

4.Security Analysis:

1. Port number is taking after checking.

2.Hast table is thread safe.

3.Users can not view or do anything if they don't login