PARALLEL BANKING SYSTEM

Final Report

In fulfillment of the requirements for the

CS 4211 Parallel & Distributed Computing Project

At NIIT University

Under: Prof PRASHANT SRIVASTAVA

Submitted by

S.No	NAME	Enrol No.
1	Pisupati VNSSK Chaitanya	085
2	Tavva GNRSN Prudhvith	142
3	CVN Sai Koushik	024
4	Nandini Sinha	075
5	Kattekola Vaishnavi	270

1 INTRODUCTION

In this implementation of the distributed banking system we use java to create a modular executable distributed system and each of the mentioned modules interact using Remote Method Invocation(RMI). Here multiple clients can access the same server at same time without any collisions. This is reminiscent to virtual client-server system as the data structure will serve similar functionality as that of server storage and a java program is written to manage this data structure whereas the client will be emulated by multiple instances of the windows terminal. The client will first login (necessary for server to authorize the client) and after doing so client will have access to some basic operations like deposit, withdraw, check balance and view transaction history. All the client-side instructions will have to be given in the form of queries written in the command prompt, which the java program of the client side will parse and communicate the request to the server side. The pros for decentralizing the processing conquer the cons so, here we are implementing a distributed banking system that consists of a server and a network of several clients where server maintains and manages all users account data and the consumers could interact with clients and perform their operations offered by server.

2 MOTIVATION

On demand services are experiencing increasing need as more and more people are interested in the possibility of accessing services on the go. So, with this trend in mind we tried to find an application of the course CS-4211 Parallel and Distributed Computing and incorporate it into our project. Since this course deals with parallel execution and the communication between tasks we had the idea of implementing a distributed banking system where the different terminal windows will simulate the independent clients/users which will try to simultaneously access a server emulated by a database and controlled by a java program. Once they are granted the access they can perform some basic transaction operations.

3. PROPOSED WORK

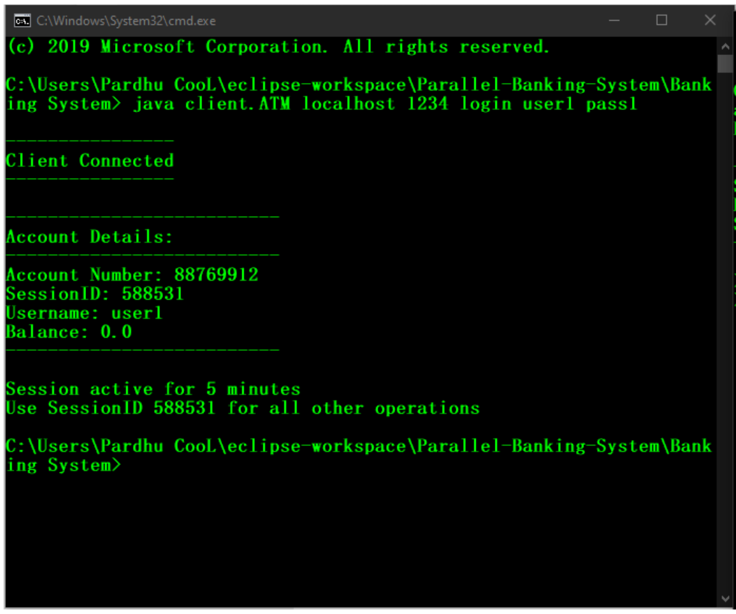
3.1. Aim of the proposed Work

The aim of this work to create a client-server based program which communicated via Java RMI. Here the client i.e. terminal initiates an operation by calling a remote method on the bank server to execute some basic transaction functionalities but first of all login is required to authenticate the user account for which simple array based data structure has been used.

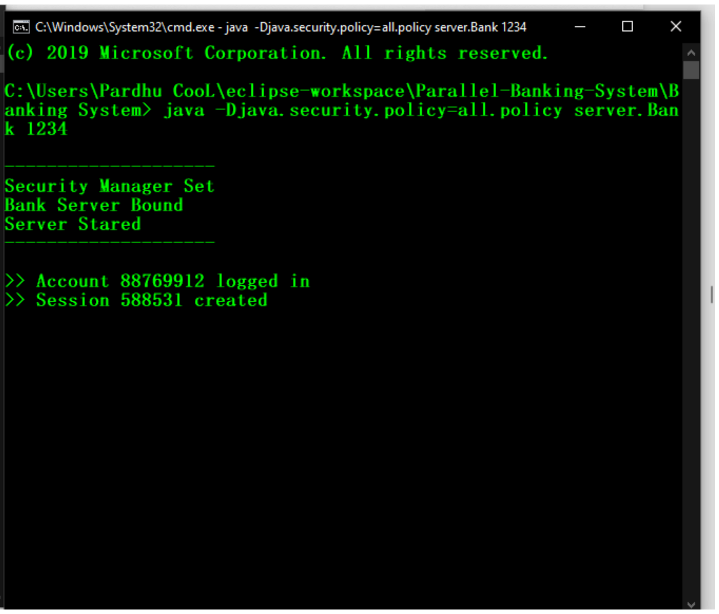
3.2. Work Done

- First, we have to register for RMI port
- Let it be any random port 1234
- Then login has to dine with right credentials
- If successful:
- Login for user gets valid for 5 minutes
- Now ask for any option user wants among the provided functionalities
- Let it be inquiry
- If inquiry account number is typed
- The current balance of account number is amount
- Now let the asked functionality be deposit
- deposit account number: deposit amount
- Successfully deposited deposit amount to account number

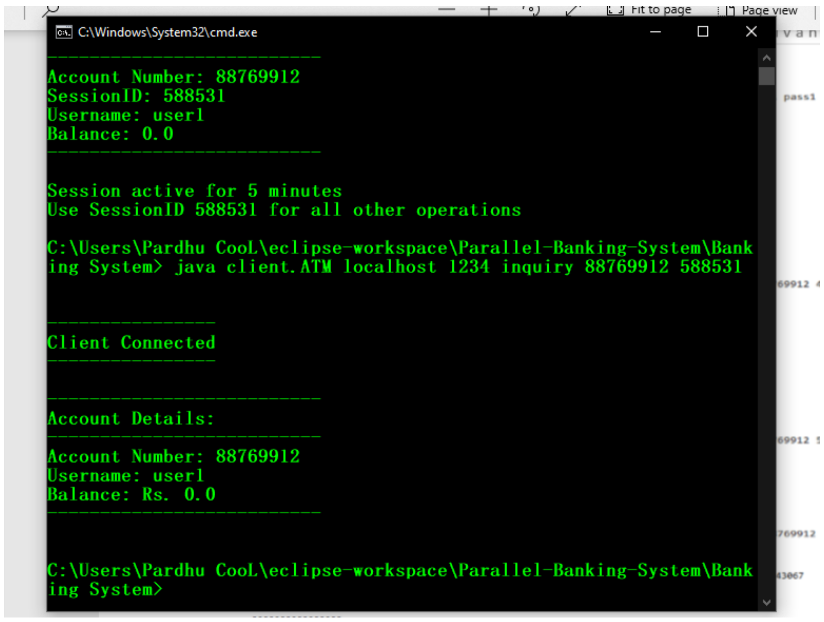
SCREENSHOTS OF THE WORK DONE



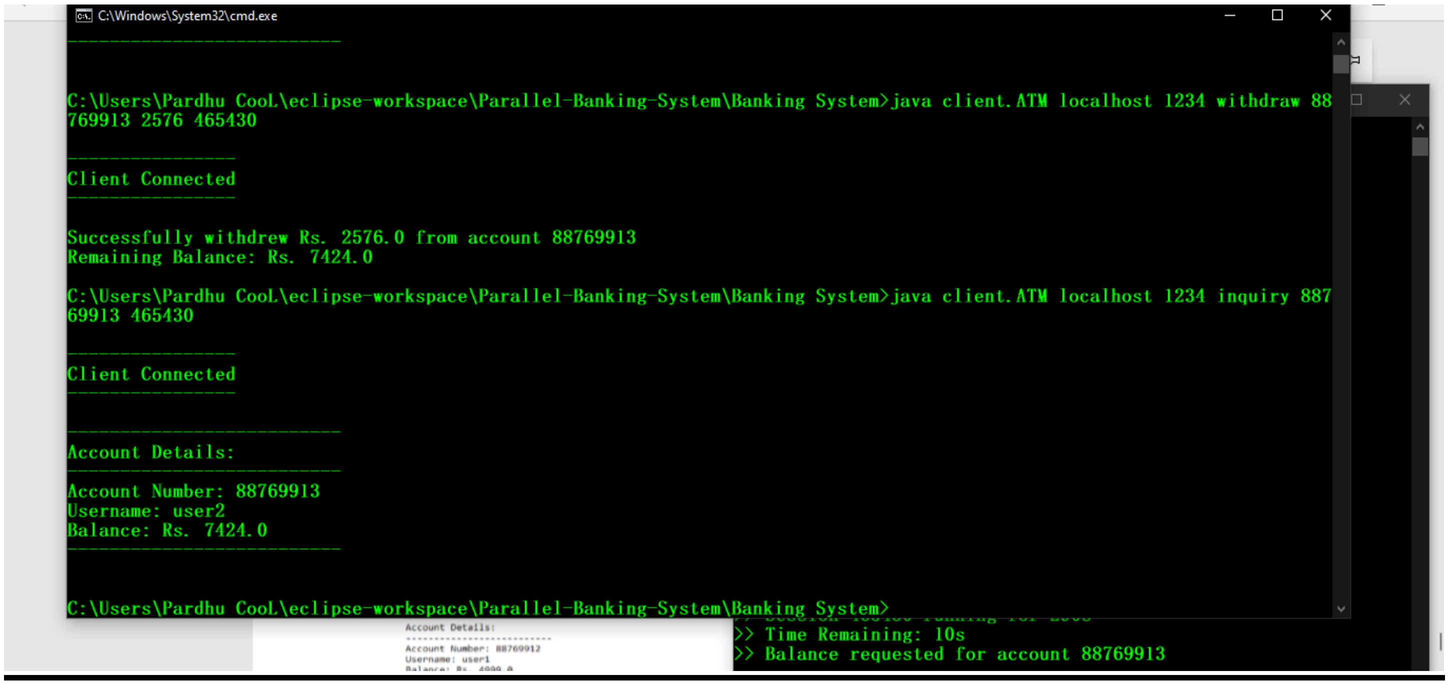
```
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Bank
ing System> java client.ATM localhost 1234 login user1 pass1

----------------
Client Connected
----------------

--------------------------
Account Details:
--------------------------
Account Number: 88769912
SessionID: 588531
Username: user1
Balance: 0.0
--------------------------

Session active for 5 minutes
Use SessionID 588531 for all other operations

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Bank
ing System>
```



```
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\B
anking System> java -Djava.security.policy=all.policy server.Ban
k 1234

--------------------
Security Manager Set
Bank Server Bound
Server Stared
--------------------

>> Account 88769912 logged in
>> Session 588531 created
```

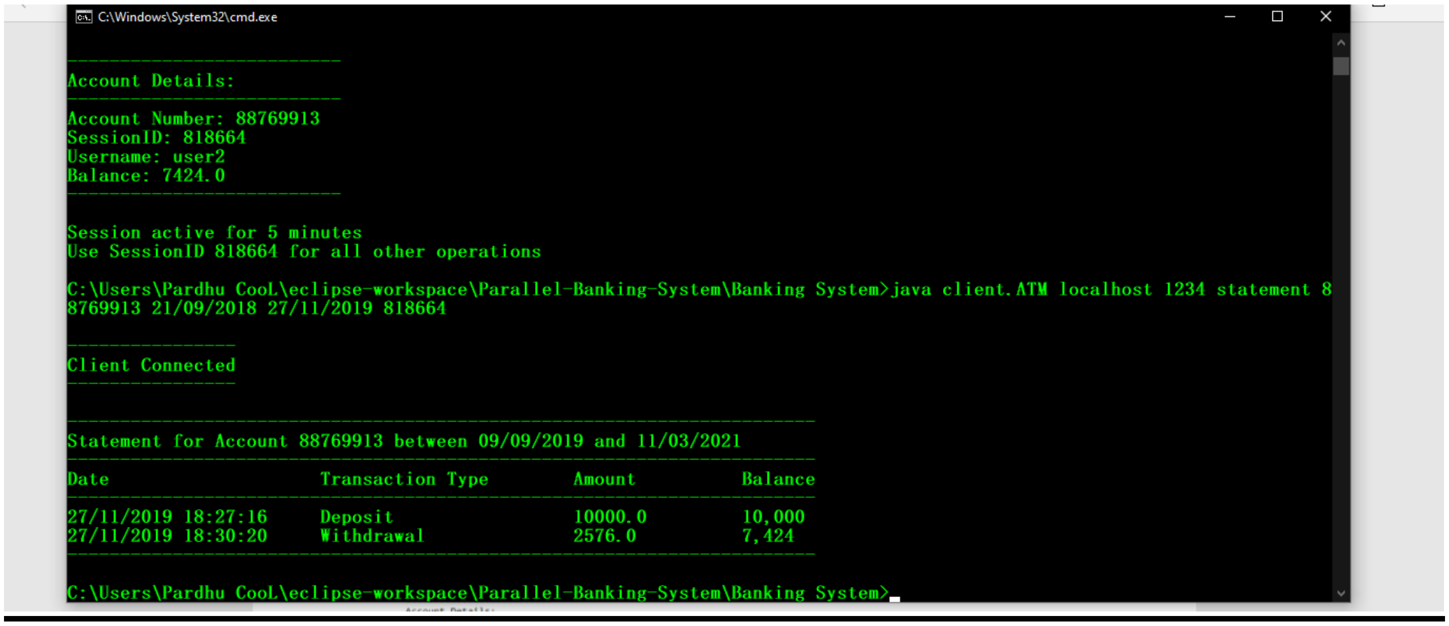


```
--------------------------
Account Number: 88769912
SessionID: 588531
Username: user1
Balance: 0.0
--------------------------

Session active for 5 minutes
Use SessionID 588531 for all other operations

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Bank
ing System> java client.ATM localhost 1234 inquiry 88769912 588531

----------------
Client Connected
----------------

--------------------------
Account Details:
--------------------------
Account Number: 88769912
Username: user1
Balance: Rs. 0.0
--------------------------

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Bank
ing System>
```

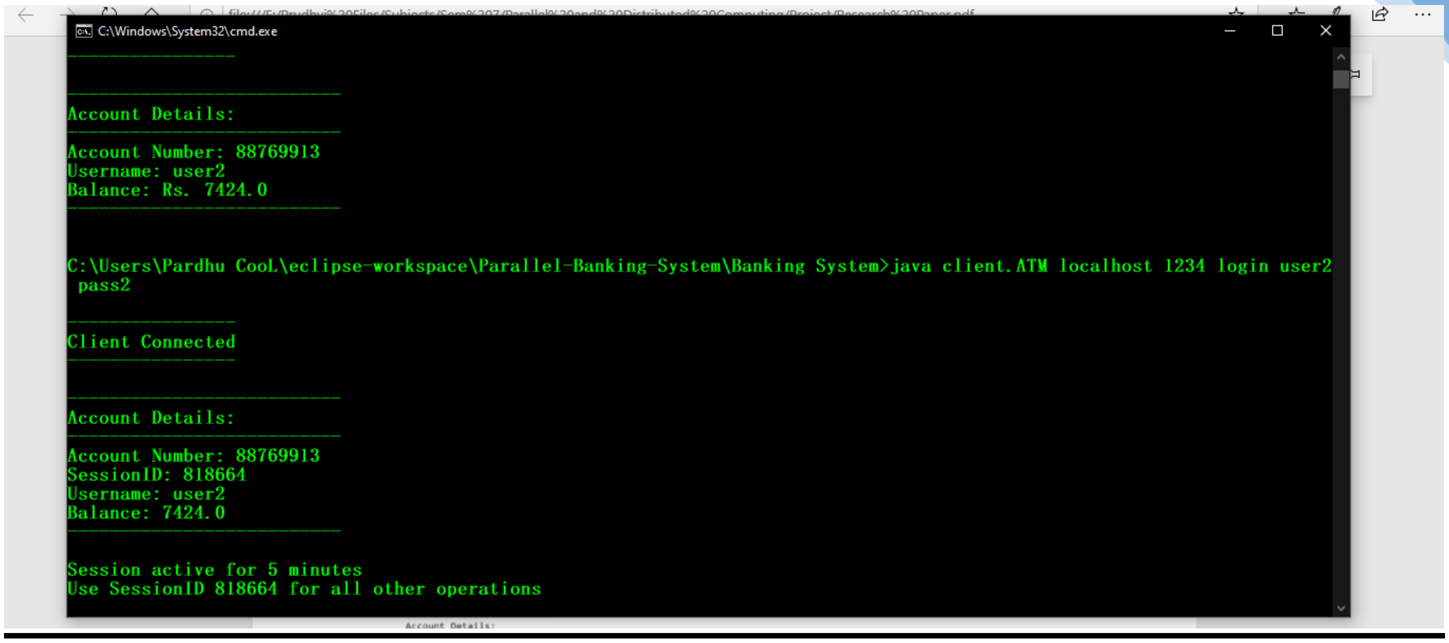


```
C:\Windows\System32\cmd.exe
-----------------------------

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Banking System>java client.ATM localhost 1234 withdraw 88
769913 2576 465430

----------------
Client Connected
----------------

Successfully withdrew Rs. 2576.0 from account 88769913
Remaining Balance: Rs. 7424.0

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Banking System>java client.ATM localhost 1234 inquiry 887
69913 465430

----------------
Client Connected
----------------

--------------------------
Account Details:
--------------------------
Account Number: 88769913
Username: user2
Balance: Rs. 7424.0
--------------------------

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Banking System>
```

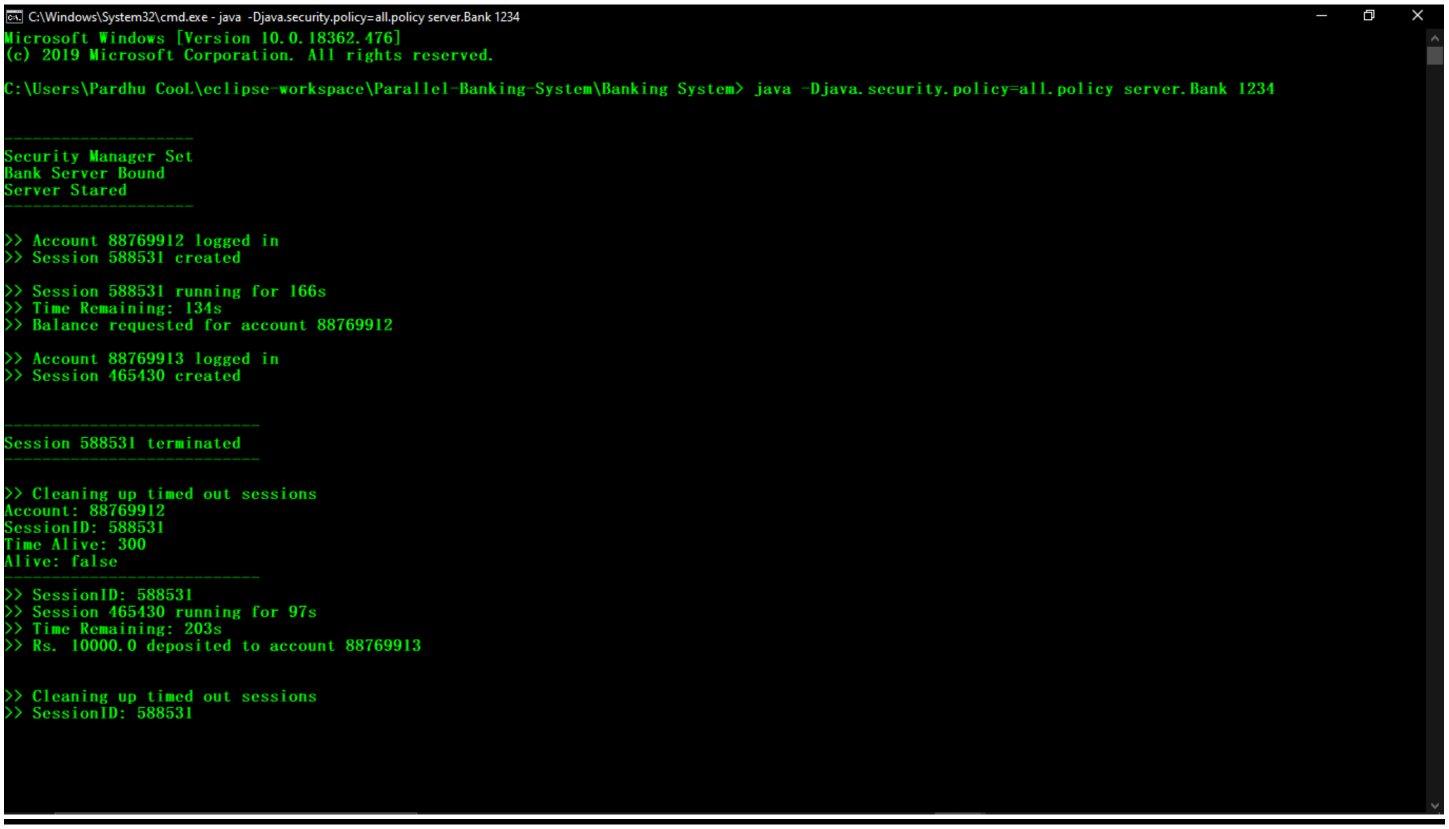
```
>> Time Remaining: 10s
>> Balance requested for account 88769913
```



```
C:\Windows\System32\cmd.exe
--------------------------
Account Details:
--------------------------
Account Number: 88769913
SessionID: 818664
Username: user2
Balance: 7424.0
--------------------------

Session active for 5 minutes
Use SessionID 818664 for all other operations

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Banking System>java client.ATM localhost 1234 statement 8
8769913 21/09/2018 27/11/2019 818664

----------------
Client Connected
----------------

----------------------------------------------------------------------
Statement for Account 88769913 between 09/09/2019 and 11/03/2021
----------------------------------------------------------------------
Date                    Transaction Type        Amount          Balance
----------------------------------------------------------------------
27/11/2019 18:27:16     Deposit                 10000.0         10,000
27/11/2019 18:30:20     Withdrawal              2576.0          7,424
----------------------------------------------------------------------

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Banking System>_
```



```
C:\Windows\System32\cmd.exe
----------------
------------------------------
Account Details:
------------------------------
Account Number: 88769913
Username: user2
Balance: Rs. 7424.0
------------------------------


C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Banking System>java client.ATM localhost 1234 login user2
 pass2

----------------
Client Connected
----------------

------------------------------
Account Details:
------------------------------
Account Number: 88769913
SessionID: 818664
Username: user2
Balance: 7424.0
------------------------------

Session active for 5 minutes
Use SessionID 818664 for all other operations
```



```
C:\Windows\System32\cmd.exe - java  -Djava.security.policy=all.policy server.Bank 1234
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Pardhu CooL\eclipse-workspace\Parallel-Banking-System\Banking System> java -Djava.security.policy=all.policy server.Bank 1234


--------------------
Security Manager Set
Bank Server Bound
Server Stared
--------------------

>> Account 88769912 logged in
>> Session 588531 created

>> Session 588531 running for 166s
>> Time Remaining: 134s
>> Balance requested for account 88769912

>> Account 88769913 logged in
>> Session 465430 created


---------------------------
Session 588531 terminated
---------------------------

>> Cleaning up timed out sessions
Account: 88769912
SessionID: 588531
Time Alive: 300
Alive: false
---------------------------
>> SessionID: 588531
>> Session 465430 running for 97s
>> Time Remaining: 203s
>> Rs. 10000.0 deposited to account 88769913


>> Cleaning up timed out sessions
>> SessionID: 588531
```

4 APPLICATION OF THE PROJECT

4.1. Problem Statement
To implement a distributed banking system where the different terminal windows will simulate the independent clients/users which will try to simultaneously access a server emulated by a database and controlled by a java program. Once they are granted the access they can perform basic transaction operations like deposit, withdrawal, inquiry and bank statement receipt.

4.2. Constituents
This distributed banking system consists of a server and a client. The server all users basic account information that can be invoked by customer via terminal. The functionalities are as follows:
- deposit: in this operation the user account's balance would get increased by the amount entered.
- withdraw: using this operation the user account's balance would get deducted by the amount entered.
- inquiry: using this operation the user would get to know his/her balance of respective account.
- getStatement: this operation returns a statement having transactions over a period of time.
- login: this operation is to authenticate the user account accessibility. Here we have written a client-server based program which communicated via Java RMI. Here the client i.e. terminal initiates an operation by calling a remote method on the bank server to execute one of the above-mentioned functionalities but first of all login is required to authenticate the user account for which simple array based data structure has been used. If the login succeeds a session ID is returned which is then valid for 5 minutes to use. The session ID acts as an authentication token that must be passed for each of the other remote methods.

5 CONCLUSION

In this day and age of developing advancements, endeavours are moving towards the Internet for trade and business. Individuals are surging towards the web based business applications for their everyday needs, which thus are making the Internet extremely well known. Internet Banking has given both an open door and a test to conventional managing an account. In the quickly developing world, managing an account is a need, which takes a great deal of time from our bustling schedule. How Distributed applications like web based Banking, can be created without much of a stretch utilizing Java and its conveyed model of distributed architecture, is shown in this paper. This project as intended has helped us to learn a lot how to write distributed programs in Java. We created a program with the aim of implementing a client-server based program which communicated via Java RMI. Here the client i.e. terminal initiates an operation by calling a remote method on the bank server to execute some basic transaction functionalities but first of all login is required to authenticate the user account for which simple array based data structure has been used.

6 FUTURE DEVELOPMENT POSSIBILITIES

This project can be enhanced by incorporating various other functionalities or features like enhanced security with encrypted implementation of login functionality. Also, GUI implementation instead of the current command line implementation will be a more practical and aesthetically pleasing method to go about it.