# Atm Interface

Project by

Saikiran Pagadala

# Introduction:

**Project Title**: Atm Interface

**Prepared by:** Saikiran Pagadala

Aim: To implement Atm interface using Jdbc connection.

Programming languages:

- Java
- MySQL

Software used:

- Eclipse
- MySQL Database

Synopsis:

**In java:**

There are 3 classes:

1. Mainmenu
2. BankingOperations
3. DatabaseConnection

And there are 8 main operations performed in Banking operation class in this application:

1. Deposit
2. Withdraw
3. Transfer
4. Check balance
5. Mini statement
6. Custom statement
7. Pin change
8. Exit

**Database:**

I have created two tables to manipulate operations and update transactions history:

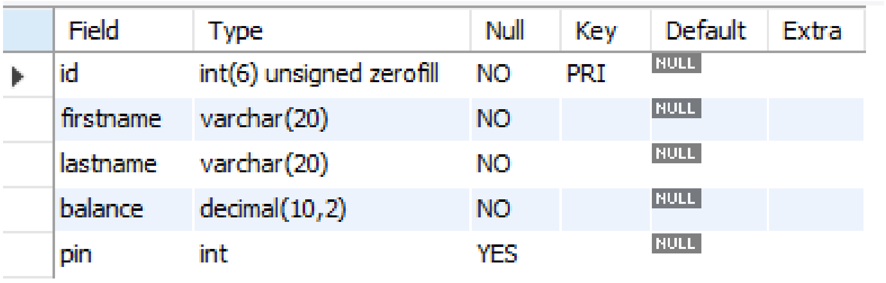
1. bankusers
2. ministatement
3.

**Project Creation**:

To perform those banking operations I need some data, for those I have created below tables:

**Bankusers**:

This table has 5columns, id accepts 6digits, firstname and lastname has regex to check no numeric and special characters, balance(>0), pin has condition (pin<9999 and pin>1000)as we can see below the structure of table.



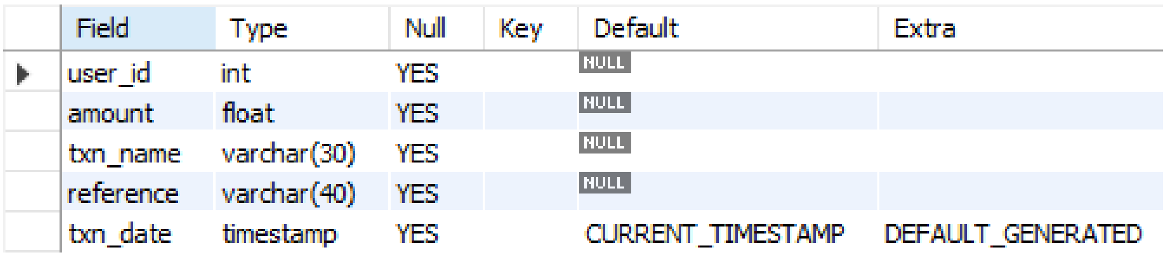
|  | Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|---|
| ▶ | id | int(6) unsigned zerofill | NO | PRI | NULL |  |
|  | firstname | varchar(20) | NO |  | NULL |  |
|  | lastname | varchar(20) | NO |  | NULL |  |
|  | balance | decimal(10,2) | NO |  | NULL |  |
|  | pin | int | YES |  | NULL |  |

Creation Query :

```
CREATE TABLE `bankusers` (

 `id` int(6) unsigned zerofill NOT NULL,

 `firstname` varchar(20) NOT NULL,

 `lastname` varchar(20) NOT NULL,

 `balance` decimal(10,2) NOT NULL,

 `pin` int DEFAULT NULL,

 PRIMARY KEY (`id`),

 CONSTRAINT `bankusers_chk_1` CHECK (((`pin` < 9999) and (`pin` > 1000))),

 CONSTRAINT `check_firstname_alpha` CHECK (regexp_like(`firstname`,_utf8mb4'^[A-Za-z]+$')),

 CONSTRAINT `check_lastname_alpha` CHECK (regexp_like(`lastname`,_utf8mb4'^[A-Za-z]+$')))
```

**Ministatement**:

This table has 5 columns are as user_id, amount, txn_name, reference, txn_date( default current timestamp).This table will stores the records through banking operations, I have implement like whenever the operations happens bankusers , I am storing those manipulations storing in this table. As we can see the table structure below :



|  | Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|---|
| ▶ | user_id | int | YES |  | NULL |  |
|  | amount | float | YES |  | NULL |  |
|  | txn_name | varchar(30) | YES |  | NULL |  |
|  | reference | varchar(40) | YES |  | NULL |  |
|  | txn_date | timestamp | YES |  | CURRENT_TIMESTAMP | DEFAULT_GENERATED |

Creation Query:

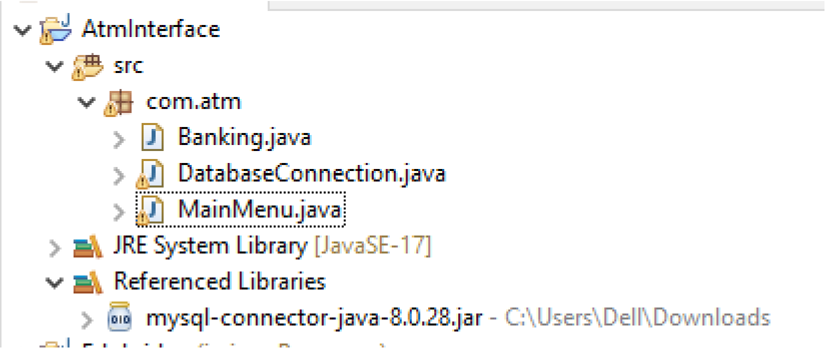
```
CREATE TABLE `ministatement` (

 `user_id` int DEFAULT NULL,

 `amount` float DEFAULT NULL,
 `txn_name` varchar(30) DEFAULT NULL,

 `reference` varchar(40) DEFAULT NULL,

 `txn_date` timestamp NULL DEFAULT CURRENT_TIMESTAMP)
```

Java Project creation:

Now, I have to create core java project with name “Atm Interface” and I need to add mySQL connector **jar file** to the project referenced library. This jar file  helps to connect with mySQL database and to have operations like table creation, inserting and update records, and to manipulations on tables.

In the project I have created a package called com.atm and created 3 classes in that, they are

MainMenu, DatabaseConnection and BankingOperations.java as we can see the project structure given below:



```
AtmInterface
  src
    com.atm
      Banking.java
      DatabaseConnection.java
      MainMenu.java
  JRE System Library [JavaSE-17]
  Referenced Libraries
    mysql-connector-java-8.0.28.jar - C:\Users\Dell\Downloads
```

The project codes are as we can see the according to classes:

**MainMenu.class**

```
package com.atm;


import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.InputMismatchException;

import java.util.Scanner;


public class MainMenu {


	public static void main(String[] args) throws ClassNotFoundException, SQLException{

		boolean flag = true;
```

```
        boolean id= true;

        ResultSet rs=null;


        while(id) {

            Scanner input = new Scanner(System.in);

    try {

            System.out.println("Enter userid");

            int userid = input.nextInt();

                Connection conn = DatabaseConnection.getConnection();

                Statement stmt = conn.createStatement();

                rs= stmt.executeQuery("select * from bankusers where id =" + userid);


    if(rs.next()) {//if user founds in the bank database, interface shows welcome user

                    System.out.println("Welcome to the banking: " +
rs.getString("firstname").toUpperCase() + "\n");

while (flag) {  // this loop runs until you call exit method,

//in which flag becomes false which don't satisfies loop condition

                        System.out.println("Choose option\n");

                        System.out.println("1.Deposit" );

                        System.out.println("2.Withdraw");

                        System.out.println("3.Transfer");

                        System.out.println("4.Mini Statment");

                        System.out.println("5.Custom statement");

                        System.out.println("6.Check Balance");

                        System.out.println("7.Change Pin");

                        System.out.println("8.Exit");

                        System.out.println("click the option");

 int choice=input.nextInt();

    switch (choice) {               //switch calls methods as per your choice

    case 1: Banking.deposit(userid);

    break;

    case 2: Banking.withdraw(userid);

    break;

                        case 3: Banking.transfer(userid);
case 3: Banking.transfer(userid);
    break;
case 4: Banking.miniStament(userid);
break;
case 5: Banking.customStatement(userid);
break;
case 6: Banking.checkBalance(userid);
break;
case 7: Banking.changePin(userid);
break;
case 8:         // Exit the program when the user chooses option 8
flag = false;id=false;
System.out.println("Thanks for using our services ");
break;
default: System.out.println("Invalid choice");
            }
        }
    }else {
    System.out.println("Invalid user id");
    }
    }catch (Exception e) {
                System.out.println("Userid must be Numeric");
            }
        }
    }
}
```

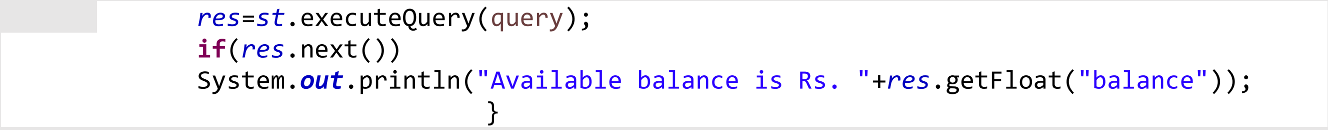
**BankingOperations.class**

```
package com.atm;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.Scanner;

public class Banking {

    private static Connection conn;
    private static Statement st;
    private static ResultSet res;
    static Scanner sc;

    public static void deposit(int userid) throws SQLException, ClassNotFoundException {
        conn=DatabaseConnection.getConnection();
        sc=new Scanner(System.in);
        st=conn.createStatement();
        int amount=0;
        String query="select * from bankusers where id="+userid;
        res=st.executeQuery(query);

        if(res.next()) {
            System.out.println("Enter you pin");
            int pin=sc.nextInt();
            if(res.getInt("pin")==pin) {
                System.out.println("Enter the deposit amount");
                amount=sc.nextInt();
                if(amount<=0) {
                    System.out.println("The amount should be greater than 0");
                    }
```

```
else {
// below statement updates the balance in the bankusers table
 st.executeUpdate("update bankusers set balance=balance+"+amount+" where id="+userid);
// below statement inserts the above updations into the ministatements to track the user transaction
st.executeUpdate("insert into ministatement (user_id,txn_name,reference,amount)
values("+userid+",'credit','by self',"+amount+")");


        res=st.executeQuery(query);
        if(res.next()) {
System.out.println("withdrawal amount of Rs "+amount+" Successfull \n\nclick Y to check Available balance,
otherwiese N ");
        char c=sc.next().toUpperCase().charAt(0);
        if(c=='Y') {
        System.out.println("Available balance is Rs. "+res.getFloat("balance"));}
                }}
            }
            else {
        System.out.println("Incorrect pin \n");
                }
            }
        }
        public static void withdraw(int userid) throws SQLException, ClassNotFoundException {
            conn=DatabaseConnection.getConnection();
            sc=new Scanner(System.in);
            st=conn.createStatement();
            int wdamount=0;
            String query="select * from bankusers where id="+userid;
            res=st.executeQuery(query);
            if(res.next()) {
                System.out.println("Enter you pin");
                int pin=sc.nextInt();
                if(res.getInt("pin")==pin) {
                    System.out.println("Enter the amount you want to withdraw :");
                    wdamount=sc.nextInt();
```

```
if(wdamount<=0) {
System.out.println("The Withdraw amount should be greate than 0 ");
}
else {
    if(wdamount>res.getFloat("balance")) {
    System.out.println("You have insufficient funds");
    }else {
    st.executeUpdate("update bankusers set balance=balance-"+wdamount+" where id="+userid);
    st.executeUpdate("insert into ministatement (user_id,txn_name,reference,amount)
    values("+userid+",'debit','by self',"+wdamount+")");
    res=st.executeQuery(query);
    if(res.next()) {
    System.out.println("withdrawal amount of Rs "+wdamount+" Successfull \n\nclick Y to check
    Available balance, otherwiese N ");
    char c=sc.next().toUpperCase().charAt(0);
    if(c=='Y') {
    System.out.println("Available balance is Rs. "+res.getFloat("balance"));
                }
            }
            }
        }
        }
    else {
        System.out.println("Incorrect pin\n");
    }
    }
}
public static void transfer(int userid) throws SQLException {
        conn=DatabaseConnection.getConnection();
        sc=new Scanner(System.in);
        st=conn.createStatement();
```

```
System.out.println("Enter the id to whom you want transfer");

int reciever =sc.nextInt();

String query="select * from bankusers where id ="+userid;

        res=st.executeQuery(query);

                if(res.next()) {

                System.out.println("Enter you pin");

                int pin=sc.nextInt();


                if(res.getInt("pin")==pin) {

                System.out.println("Enter the transfer amount :");

                int tamount=sc.nextInt();

                if(tamount<=0) {

                System.out.println("The amount should be greater than 0");

                }

                else if(tamount>res.getFloat("balance")) {

                System.out.println("You have insufficient funds");

        }

        else {

        st.executeUpdate("update bankusers set balance=balance- "+tamount+" where id="+userid);

        st.executeUpdate("update bankusers set balance=balance+ "+tamount+" where id="+reciever);

        st.executeUpdate("insert into ministatement (user_id,txn_name,reference,amount)
        values("+userid+",'debit','transfer to "+reciever+"',"+tamount+")");

        st.executeUpdate("insert into ministatement (user_id,txn_name,reference,amount)
        values("+reciever+",'credit','transfer from "+userid+"',"+tamount+")");

        res=st.executeQuery(query);

while(res.next()) {

System.out.println("Transfer amount of Rs"+tamount+" was suscessfull \n\nIf you want to check Available
balance, click Y otherwiese N ");

                                        }

                        char c=sc.next().toUpperCase().charAt(0);

                        if(c=='Y') {
```



```
                res=st.executeQuery(query);
                if(res.next())
                System.out.println("Available balance is Rs. "+res.getFloat("balance"));
                                        }
```

```
}           }else {
            System.out.println("Incorrect pin\n");
                        }           }
                        else {
                                    System.out.println("The recevier not found");
                        }}
public static void miniStament(int userid) throws SQLException {
                        conn=DatabaseConnection.getConnection();
                        sc=new Scanner(System.in);
                        st=conn.createStatement();

                        res=st.executeQuery("select * from bankusers where id="+userid);

                        if(res.next()) {
                                    System.out.println("Enter your pin :");
int pin=sc.nextInt();

if(pin==res.getInt("pin")) {

res=st.executeQuery("select * from ministatement where user_id= "+userid+" order by txn_date desc limit 5");

System.out.println("+------------+------------+-------------+--------------+-----------------------+");

System.out.println("|   userid    |    amount   |  txn_name   |  reference  |          Date          |");

System.out.println("+------------+------------+-------------+--------------+-----------------------+");

while(res.next()) {
System.out.println(String.format("|  %-10d |  %-10.2f |  %-10s  |  %-12s |  %-20s
|",res.getInt("user_id"),res.getFloat("amount"),res.getString("txn_name"),res.getString("reference"),res.getTim
estamp("txn_date")));    }

System.out.println("+------------+------------+-------------+--------------+-----------------------+");
                                                }
                                    else {
```

```
if(res.next()) {
                        System.out.println("Enter your pin :");
                        pin=sc.nextInt();
                        if(pin==res.getInt("pin")) {
                                res.close();
                                try {
                                        res=st.executeQuery("select balance from bankusers where
id="+userid);
                                        while(res.next()) {
                                                System.out.println("Your Current balance is:
"+res.getFloat("balance")+"\n");
                                        }
                                } catch (SQLException e) {
                                        // TODO Auto-generated catch block
                                        e.printStackTrace();
                                }
                        }
                        else {
                                System.out.println("Incorrect pin\n");


                        }
                }
        }
public static void changePin(int userid) throws SQLException {
                conn=DatabaseConnection.getConnection();
                sc=new Scanner(System.in);
                st=conn.createStatement();
                res=st.executeQuery("select * from bankusers where id="+userid);
                if(res.next()) {
                        System.out.println("Enter your pin");
                        int pin=sc.nextInt();
                        if(pin==res.getInt("pin")){
                                System.out.println("Enter your New pin you want to set :");
                                int newpin=sc.nextInt();
```

```
if(newpin==res.getInt("pin")) {
System.out.println("New pin should not be same as old pin");
            }
else if(newpin>9999&&newpin<1000){
System.out.println("The pin "+newpin+" is not valid");


}else {
int i=st.executeUpdate("update bankusers set pin="+newpin+" where id="+userid);
    if(i>0) {
            System.out.println("Pin changed succefully\n");
            }
            }
        }
else {
System.out.println("Incorrect pin");
        }
    }


}


public static void customStatement(int userid) throws SQLException {


        conn=DatabaseConnection.getConnection();
        sc=new Scanner(System.in);
        st=conn.createStatement();
        res=st.executeQuery("select * from bankusers where id="+userid);
        if(res.next()) {
            System.out.println("Enter your pin");
            int pin=sc.nextInt();
            if(pin==res.getInt("pin")){
                System.out.println("Enter the start Date(in YYYY-MM-DD): ");
                String start=sc.next();
```

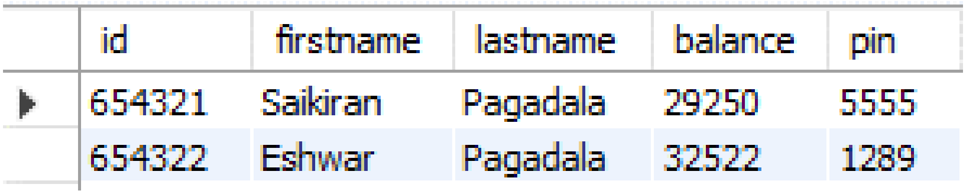
```
public static void customStatement(int userid) throws SQLException {


                    conn=DatabaseConnection.getConnection();

                    sc=new Scanner(System.in);

                    st=conn.createStatement();

                    res=st.executeQuery("select * from bankusers where id="+userid);

                    if(res.next()) {

                              System.out.println("Enter your pin");

                              int pin=sc.nextInt();

                              if(pin==res.getInt("pin")){

                                        System.out.println("Enter the start Date(in YYYY-MM-DD): ");

                                        String start=sc.next();

System.out.println("Enter end date(in YYYY-MM-DD ): ");

          String end=sc.next();

res=st.executeQuery("select * from ministatement where user_id= "+userid+" and txn_date between
'"+start+"' and '"+end+"'");


System.out.println("+-------------+-------------+--------------+--------------+----------------------+");

System.out.println("|   userid    |    amount   |  txn_name   |  reference  |        Date          |");

System.out.println("+-------------+-------------+--------------+--------------+----------------------+");

while(res.next()) {

System.out.println(String.format("|  %-10d |  %-10.2f |  %-10s  |  %-12s |  %-20s  |", res.getInt("user_id"),
res.getFloat("amount"),res.getString("txn_name"),res.getString("reference"),res.getTimestamp("txn_date")));

                                        }

System.out.println("+-------------+-------------+--------------+--------------+----------------------+");

                                        }

          System.out.println();

                              }else {

                                        System.out.println("Incorrect Pin");

                              }


          }


}
```

DatabaseConnection.class :

```
package com.atm;


import java.sql.Connection;

import java.sql.Driver;

import java.sql.DriverManager;


public class DatabaseConnection {

        private static String driver="com.mysql.cj.jdbc.Driver";

        private static String url="jdbc:mysql://localhost:3306/my_database";

        private static String un="root";

        private static String pass="root";

private  static Connection conn;

        public static Connection getConnection() {

                try {

                        Class.forName(driver);

 conn=DriverManager.getConnection(url,un,pass);


                } catch (Exception e) {

                        e.printStackTrace();

                }

                return conn;

                }

        }
```

**Bankusers:**

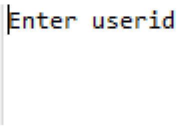
I have two records in table as bankusers, Now I'm going to run the application using these records below to show how the application works:



| id | firstname | lastname | balance | pin |
|---|---|---|---|---|
| 654321 | Saikiran | Pagadala | 29250 | 5555 |
| 654322 | Eshwar | Pagadala | 32522 | 1289 |

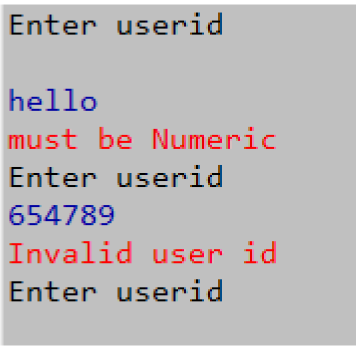
Output:

After running the application user will be asked to enter the user id as below:



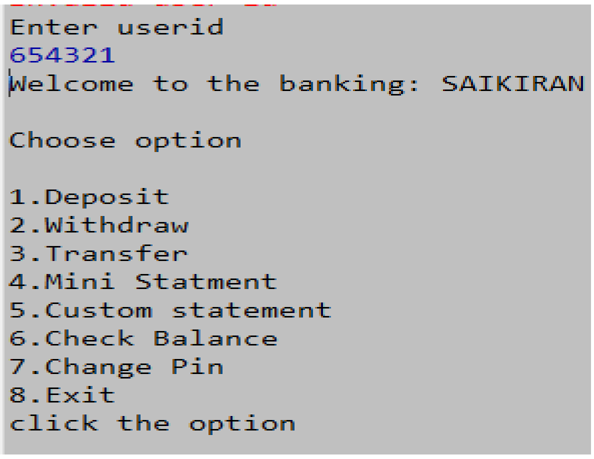
```
Enter userid
```

If user enters the enter the id, which is checks in the database if not found it show the "Inavalid userid"  and if input not a numeric value it as shown below:



```
Enter userid

hello
must be Numeric
Enter userid
654789
Invalid user id
Enter userid
```

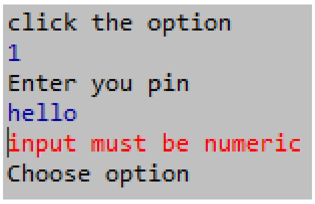
After entering existing user id, the user will get banking operations by welcome greeting like in the figure:



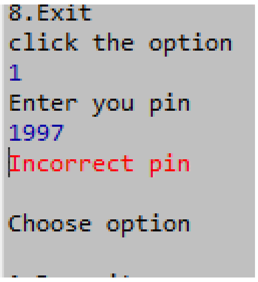
```
Enter userid
654321
Welcome to the banking: SAIKIRAN

Choose option

1.Deposit
2.Withdraw
3.Transfer
4.Mini Statment
5.Custom statement
6.Check Balance
7.Change Pin
8.Exit
click the option
```

Deposit Operation:

When we click 1, it asks to enter the pin. If the pin is entered incorrectly or string values instead then it redirects to main menu asks again to choose the option as below

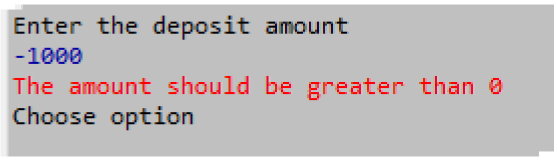


```
click the option
1
Enter you pin
hello
input must be numeric
Choose option
```

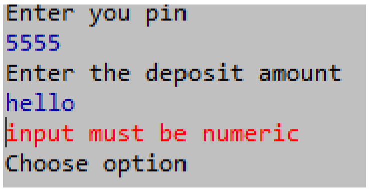


```
8.Exit
click the option
1
Enter you pin
1997
Incorrect pin

Choose option
```

After entering valid pin it user will asked to enter the amount,if entered amount  is negative or characters then the it will be shown like below

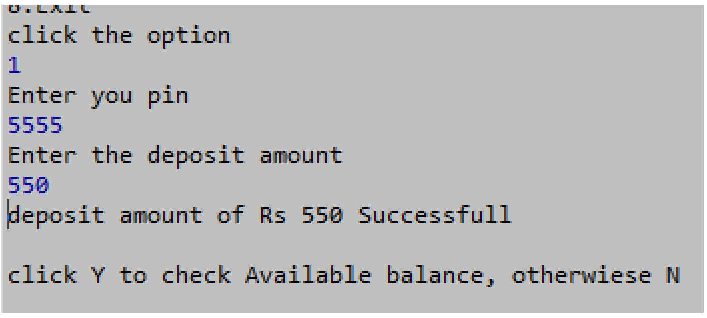


```
Enter the deposit amount
-1000
The amount should be greater than 0
Choose option
```



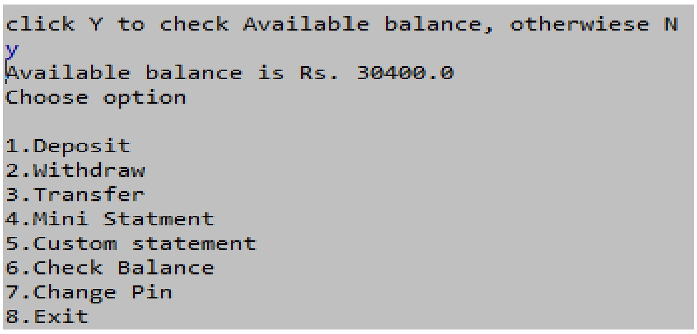
```
Enter you pin
5555
Enter the deposit amount
hello
input must be numeric
Choose option
```

When entering valid amount the process will happens and adds the amount to the balance of the user in the bank table.



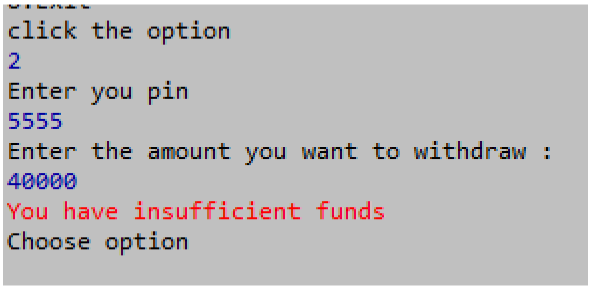
```
8.Exit
click the option
1
Enter you pin
5555
Enter the deposit amount
550
deposit amount of Rs 550 Successfull

click Y to check Available balance, otherwiese N
```

If you click the “Y” then it show the available balance and returns to mainmenu, otherwise  it directly returns to the main menu  and there user can  use the any operations  until   user calls the exit method.



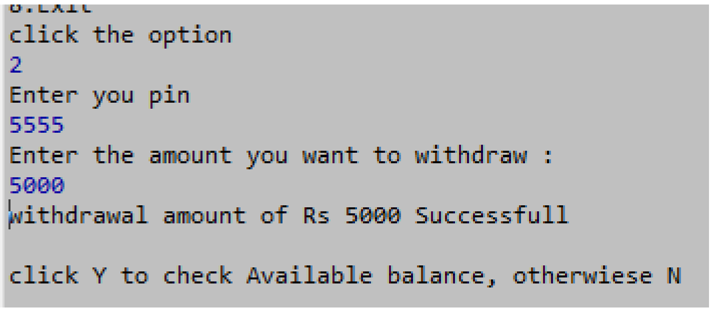
```
click Y to check Available balance, otherwiese N
y
Available balance is Rs. 30400.0
Choose option

1.Deposit
2.Withdraw
3.Transfer
4.Mini Statment
5.Custom statement
6.Check Balance
7.Change Pin
8.Exit
```

Now lets see the withdraw operation, we have already executed the negative testcases and mismatch types. But here we have to  see withdrawal of exceeding balance of the user, how exception handled as we can see below.



```
click the option
2
Enter you pin
5555
Enter the amount you want to withdraw :
40000
You have insufficient funds
Choose option
```

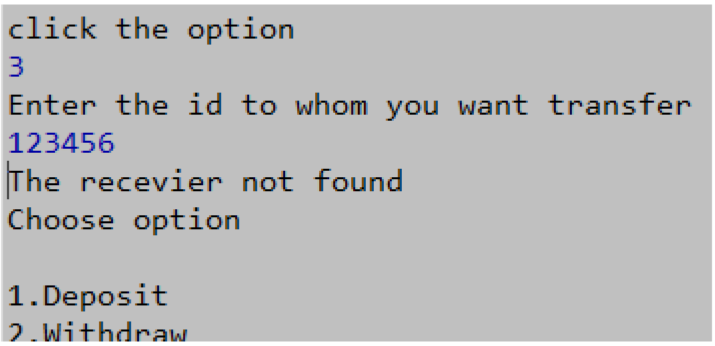
Here if user enters the amount in the range of his account balance user gets withdrawal succesfull and asks for view balance, as we already seen how it works.



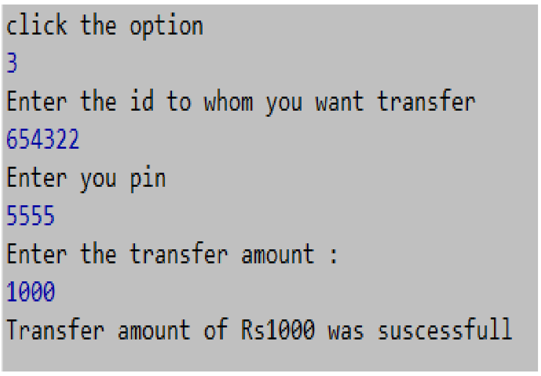
```
click the option
2
Enter you pin
5555
Enter the amount you want to withdraw :
5000
withdrawal amount of Rs 5000 Successfull

click Y to check Available balance, otherwiese N
```

**Transfer operation:**

Now we'll see if user enters a unregistered and registered ids then error shows as as below.



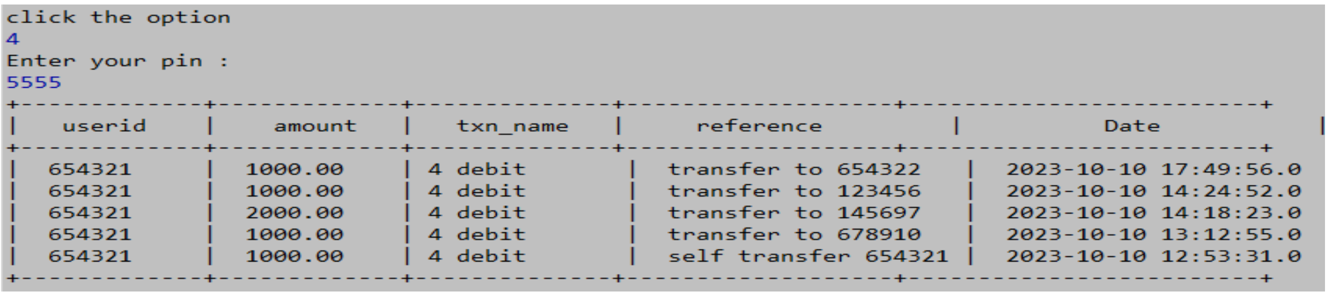
```
click the option
3
Enter the id to whom you want transfer
123456
The recevier not found
Choose option

1.Deposit
2.Withdraw
```



```
click the option
3
Enter the id to whom you want transfer
654322
Enter you pin
5555
Enter the transfer amount :
1000
Transfer amount of Rs1000 was sucessfull
```

**Ministatement:**

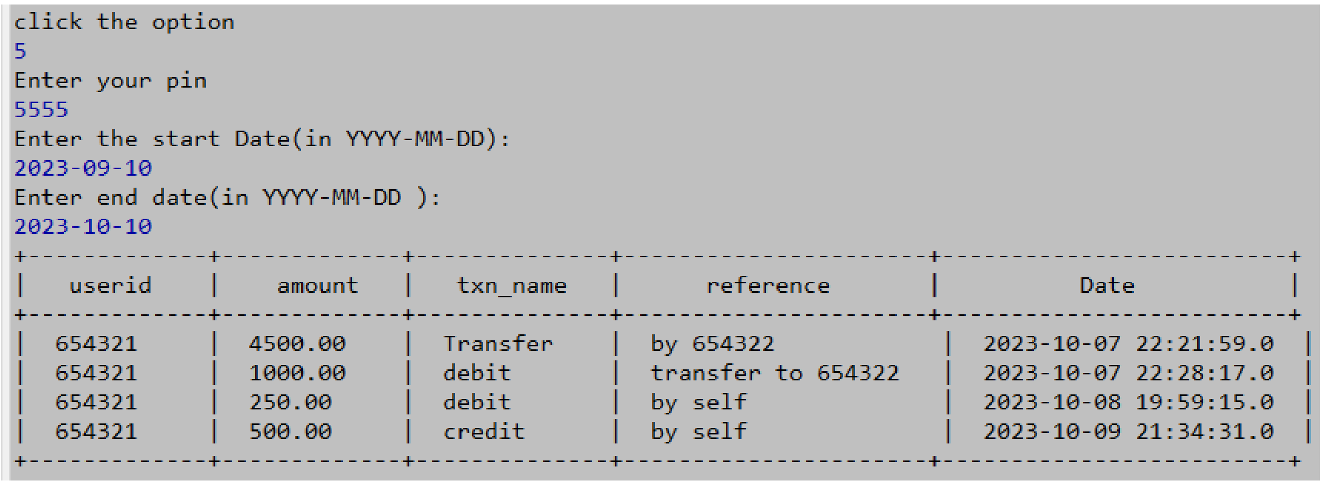
When user selects the ministatement operation it asks user to enter password. if entered pin is validates then only it shows the last transactions.



```
click the option
4
Enter your pin :
5555
+-------------+-------------+-------------+-------------------+-------------------------+
|   userid    |    amount   |   txn_name  |     reference     |           Date          |
+-------------+-------------+-------------+-------------------+-------------------------+
|  654321     |  1000.00    | 4 debit     |  transfer to 654322   |  2023-10-10 17:49:56.0
|  654321     |  1000.00    | 4 debit     |  transfer to 123456   |  2023-10-10 14:24:52.0
|  654321     |  2000.00    | 4 debit     |  transfer to 145697   |  2023-10-10 14:18:23.0
|  654321     |  1000.00    | 4 debit     |  transfer to 678910   |  2023-10-10 13:12:55.0
|  654321     |  1000.00    | 4 debit     |  self transfer 654321 |  2023-10-10 12:53:31.0
+-------------+-------------+-------------+-------------------+-------------------------+
```

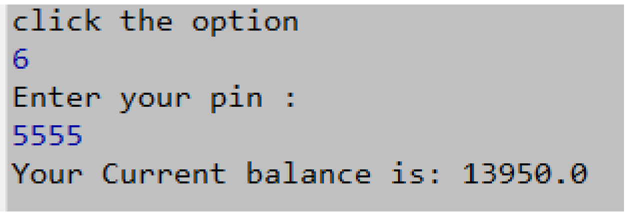
**CustomStatement:**

When user choose to custom statement, he has to enter Date range like below.



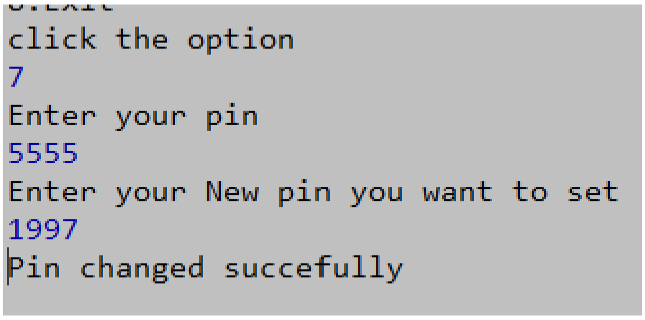
```
click the option
5
Enter your pin
5555
Enter the start Date(in YYYY-MM-DD):
2023-09-10
Enter end date(in YYYY-MM-DD ):
2023-10-10
+-------------+------------+-------------+--------------------+------------------------+
|   userid    |   amount   |  txn_name   |      reference     |          Date          |
+-------------+------------+-------------+--------------------+------------------------+
|  654321     |  4500.00   |  Transfer   |  by 654322          |  2023-10-07 22:21:59.0  |
|  654321     |  1000.00   |  debit      |  transfer to 654322 |  2023-10-07 22:28:17.0  |
|  654321     |  250.00    |  debit      |  by self            |  2023-10-08 19:59:15.0  |
|  654321     |  500.00    |  credit     |  by self            |  2023-10-09 21:34:31.0  |
+-------------+------------+-------------+--------------------+------------------------+
```

**Check Balance:**



```
click the option
6
Enter your pin :
5555
Your Current balance is: 13950.0
```

**Pin Change:**



```
click the option
7
Enter your pin
5555
Enter your New pin you want to set
1997
Pin changed succefully
```

**Exit:**

When we choose **Exit operation** it actual has no method, but I made while loop to false to stop further iteration in switch case

**Conclusion:**

The implementation of Atm interface application is successfully executed using jdbc connection.