

Table of Contents

- Brief Introduction about the system
- Full description of the system including the minimum requirements according to given description.
- Explain the classes with their attributes and methods.
- Explain exactly what the system can do (all tasks)
- User Manual
- Source Code
- Some Screenshots showing the errors or exceptions during testing the system.
- Screenshots showing the system is working without errors

Brief Introduction about the System.

- 1) The Banking Simulation System allows customers to
 - a) Create Bank Account (Savings, Checking)
 - b) Customers can view Current Balance, Deposit, Withdraw and Transfer between banks. c) Customer can view previous transaction
 - d) Pay Utility Bills (Hydro, Electricity, Gas and Maintenance Bills).
- Moreover, whenever the program is executed customer is presented with two options to create
- Savings and Checking Accounts.(Click 1 will create Savings Account || Click 2 will create Checking Account).
- Further, all the details entered by customer is stored in the file according to the selected account. (Example. Save.txt will store Savings Account details in the file.)
- Lastly, customer can view and update the details as and when required.

Full description of the system including the minimum requirements according to given description.

- 1) According to the given description all the requirements have been fulfilled however, we haven't explored any more accounts and have kept only two basic Accounts that is Savings and Checking.
- Firstly, when the code is executed user will be given a choice to create either a Savings or Checking Account.
 - Once, the user selects their choice a dialog box will appear on the screen with welcome message and some user details will be showed wherein user will enter the details and the will be saved in a text file.
 - After creating account, current balance will be shown which will be 0 as the new account is been created so, user will be given a list of options to select wherein user can deposit the money, withdraw and transfer the money between Saving and Checkings account or vice-versa.
 - Also, user can select an option to see the previous transaction done by them and even pay utility bills.
 - Further, utility bills like Hydro, Maintenance, Electricity and Gas Bills can be reimbursed from their account.
 - If user withdraws or transfers money \$10 fee is charged and lastly, user can view their details and even update as and when required.

Explain the classes with their attributes and methods.

- 1) JavaProject is the main class that is been created and under that all methods and attributes have been specified.

* Variables

`int Saving,Checking,saveacc,checkacc;` the variables used here are used in the tranfer method between the accounts.

`int Current_Balance,deposit_money,withdraw_money,transaction,fee = 5;` these variables are used in Deposit_Money, Withdraw_Money and Transaction methods.

`String Customer_Name, Customer_Mail, Country, Postal,phone,DOB;` the variables are used when user is asked to enter details during the creation of Saving and Checking account and the details entered are stored in this variables.

`Random rand = new Random();` the Random Class is used to generate random numbers for the account types.

`saveacc=rand.nextInt(90000000);` here saveacc is a variable used for Savings account to store the random numbers generated and later assigned to the Savings account during the transfer of money to different account.

`checkacc=rand.nextInt(90000000);` here checkacc is a variable used for Checking account to store the random numbers generated and later assigned to the Checking account during the transfer of money to different account.

`int Electricity_Bill, Hydro_Bill, Gas_Bill, Maintenance_Bill = 0;` this variables are used in Utility_Bills methods to reimburse the payment using this variables.

`int num = 0;` is used in Utility_Bills methods and in the switch case.

`int answer;` is used in Update_details methods in the if method to check the condition and if it's correct then user can update the details in the file.

`char option = '\0';` is used in Bank_Details() switch case.

* Methods

`public void Create_Acc()` is a method where user is given a choice to either create a new Savings or Checking.

`public int Deposit_Money (int deposit_money)` is a method used to deposit money

`public int Withdraw_Money (int withdraw_money)` is a method used to withdraw money

`public void Transaction()` is a method which will show the previous transaction done by the user.

`public void transfer(int withdraw_money)` is a method to transfer money within the accounts.

`public void Utility_Bills()` is a method to pay the utility bills

`public void Update_Details()` is a method which will show the data from the file and user can update details in the file.

`public int Bank_Details()` is a method which will call all the details mentioned above.

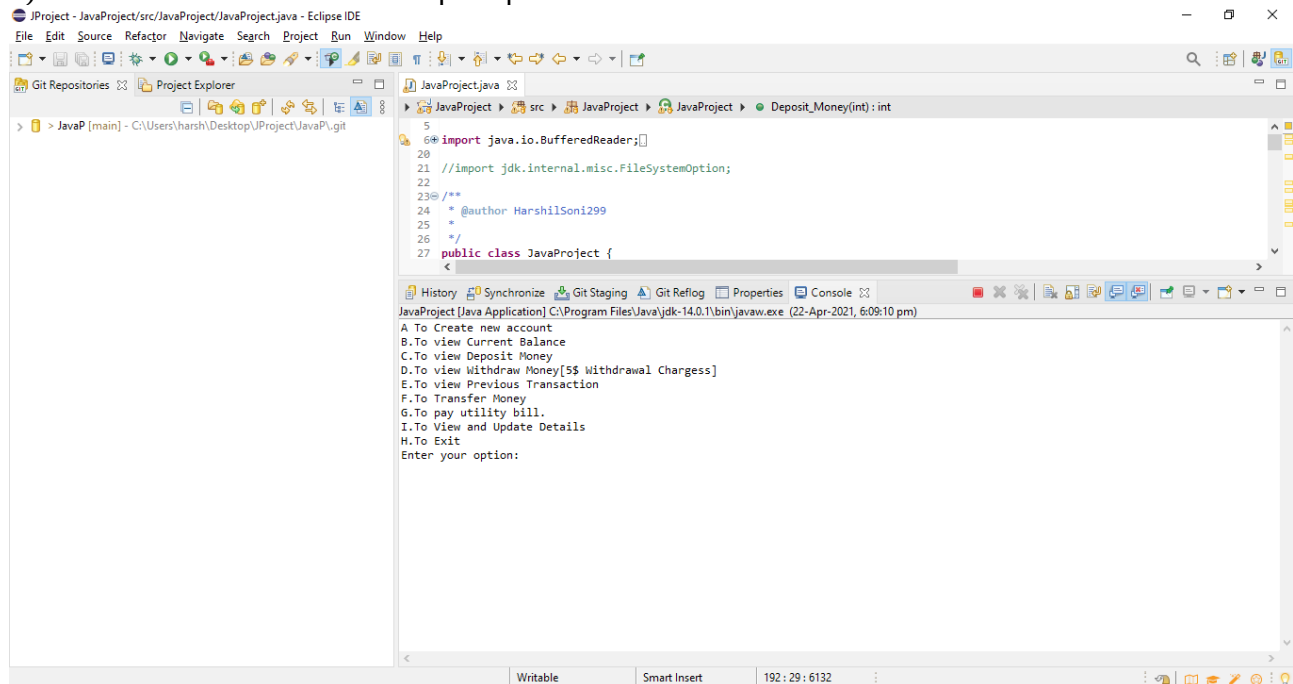
Explain exactly what the system can do (all tasks)

1) System can perform tasks like

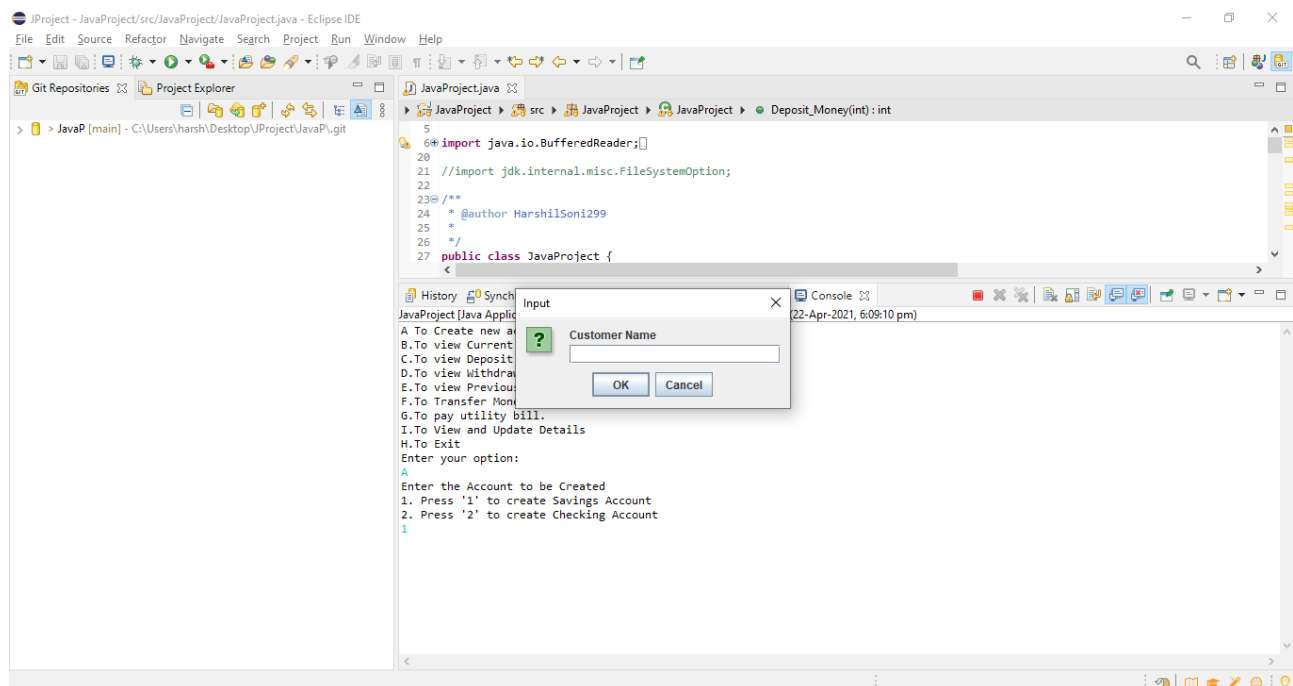
- Create Savings and Checking account.
- User can enter details about clients and account according to the account types like savings and checking.
- Users can perform deposit, withdraw money (\$10 fee is charged as well as with transfer within same bank accounts), transfer money and view previous transaction.
- Moreover, all details will be saved in a file.
- user can view the account details for checking as well as savings and update the details.
- Utility bills can be paid from the account which entails Gas, Hydro, Electricity and maintenance bills.

#User Manual

1) When code is executed user is prompted with the list of menu to choose from



As shown in the screenshot user can choose different options from the displayed menu.



When user selects option A account creation options are generated like on entering 1 Savings Account will be created and on entering 2 Checking Account will be created.

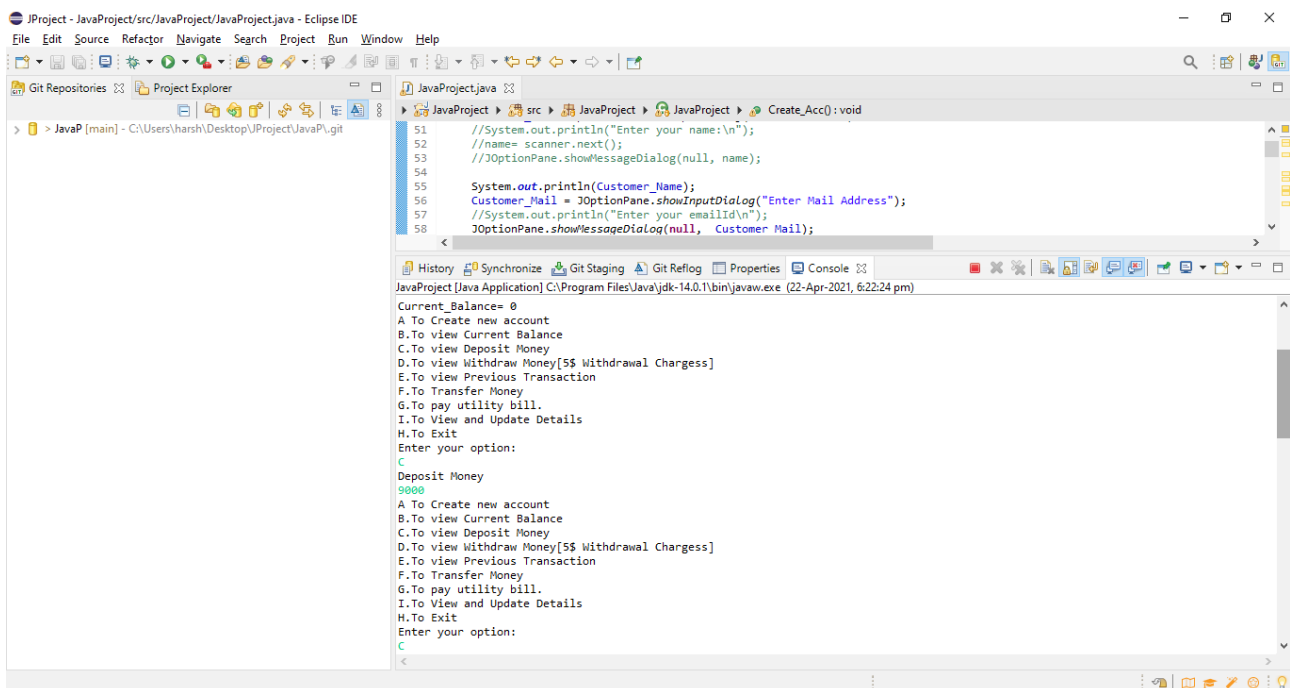
- Here, user selects 1 and is asked questions prior to open an account.
- As, user will enter all the details the information will be stored in a file.

Save.txt - Notepad

File Edit Format View Help

Customer name: HARSHIL SONI
Mail Address: harshil@gmail.com
Contact Info: 6477872878
Date of Birth: 09-03-1998
Country: Canada
Postal Code: m1g2r4

- Here user entered details are saved in a text file under savings account file.



The screenshot shows the Eclipse IDE with a Java project named 'JProject'. The 'Project Explorer' on the left shows the project structure. The 'JavaProject.java' file is open in the editor, showing the following code:

```
51 //System.out.println("Enter your name:\n");
52 //name= scanner.next();
53 //OptionPane.showMessageDialog(null, name);
54
55 System.out.println(Customer_Name);
56 Customer_Mail = JOptionPane.showInputDialog("Enter Mail Address");
57 //System.out.println("Enter your emailId\n");
58 JOptionPane.showMessageDialog(null, Customer_Mail);
```

The 'Console' view at the bottom shows the output of the program:

```
JavaProject [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (22-Apr-2021, 6:22:24 pm)
Current_Balance= 0
A To Create new account
B.To view Current Balance
C.To view Deposit Money
D.To view Withdraw Money[5$ Withdrawal Charegess]
E.To view Previous Transaction
F.To Transfer Money
G.To pay utility bill.
I.To View and Update Details
H.To Exit
Enter your option:
C
Deposit Money
9000
A To Create new account
B.To view Current Balance
C.To view Deposit Money
D.To view Withdraw Money[5$ Withdrawal Charegess]
E.To view Previous Transaction
F.To Transfer Money
G.To pay utility bill.
I.To View and Update Details
H.To Exit
Enter your option:
C
```

As, the user enters C option he can deposit the money and 9000 has been deposited successfully.

```
//Project - JavaProject/src/JavaProject/JavaProject.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

JavaProject.java
51 //System.out.println("Enter your name:\n");
52 //name= scanner.next();
53 //OptionPane.showMessageDialog(null, name);
54
55 System.out.println(Customer_Name);
56 Customer_Mail = JOptionPane.showInputDialog("Enter Mail Address");
57 //System.out.println("Enter your emailid\n");
58 JOptionPane.showMessageDialog(null, Customer Mail);

History Synchronize Git Staging Git Reflog Properties Console
JavaProject [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (22-Apr-2021, 6:22:24 pm)

A To Create new account
B.To view Current Balance
C.To view Deposit Money
D.To view Withdraw Money[$$ Withdrawal Charegss]
E.To view Previous Transaction
F.To Transfer Money
G.To pay utility bill.
I.To View and Update Details
H.To Exit
Enter your option:
F
Select the account to transfer money
Savings
Funds Withdrawn from Checking Account 84773916 balance is 890
Funds deposited in the Saving Account 18803455 balance is 890
Funds successfully transferred.
A To Create new account
B.To view Current Balance
C.To view Deposit Money
D.To view Withdraw Money[$$ Withdrawal Charegss]
E.To view Previous Transaction
F.To Transfer Money
G.To pay utility bill.
I.To View and Update Details
H.To Exit
```

- User enters option F to transfer money to Savings account from checking account and the withdrawal amount is sent to Savings Accounts(Note: to transfer money use option D to withdraw money and then select the transfer money option).

```
//Project - JavaProject/src/JavaProject/JavaProject.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

JavaProject.java
51 //System.out.println("Enter your name:\n");
52 //name= scanner.next();
53 //OptionPane.showMessageDialog(null, name);
54
55 System.out.println(Customer_Name);
56 Customer_Mail = JOptionPane.showInputDialog("Enter Mail Address");
57 //System.out.println("Enter your emailid\n");
58 JOptionPane.showMessageDialog(null, Customer Mail);

History Synchronize Git Staging Git Reflog Properties Console
JavaProject [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (22-Apr-2021, 6:34:58 pm)

A To Create new account
B.To view Current Balance
C.To view Deposit Money
D.To view Withdraw Money[$$ Withdrawal Charegss]
E.To view Previous Transaction
F.To Transfer Money
G.To pay utility bill.
I.To View and Update Details
H.To Exit
Enter your option:
I
Enter File Name:Save.txt
Correct the File name is Save.txt
Customer name: HARSHIL SONI

Mail Address: harshil@gmail.com

Contact Info: 6477872878

Date of Birth: 09-03-1998

Country: Canada

Postal Code: m1g2r4
```

- Here, using I option user can view and update information and if file name is correct then only user can view the details from the file and if file name is incorrect the details will not be displayed.

The screenshot shows the Eclipse IDE with a Java project named 'JavaProject'. The code in 'JavaProject.java' includes a menu-driven application with options A through H. The console output shows the user has selected option 'I' (View and Update Details) and entered the file name 'save'. The application then displays an error message: 'save (The system cannot find the file specified)'.

```

51 //System.out.println("Enter your name:\n");
52 //name= scanner.next();
53 //OptionPane.showMessageDialog(null, name);
54
55 System.out.println(Customer_Name);
56 Customer_Mail = JOptionPane.showInputDialog("Enter Mail Address");
57 //System.out.println("Enter your emailId\n");
58 JOptionPane.showMessageDialog(null, Customer_Mail);

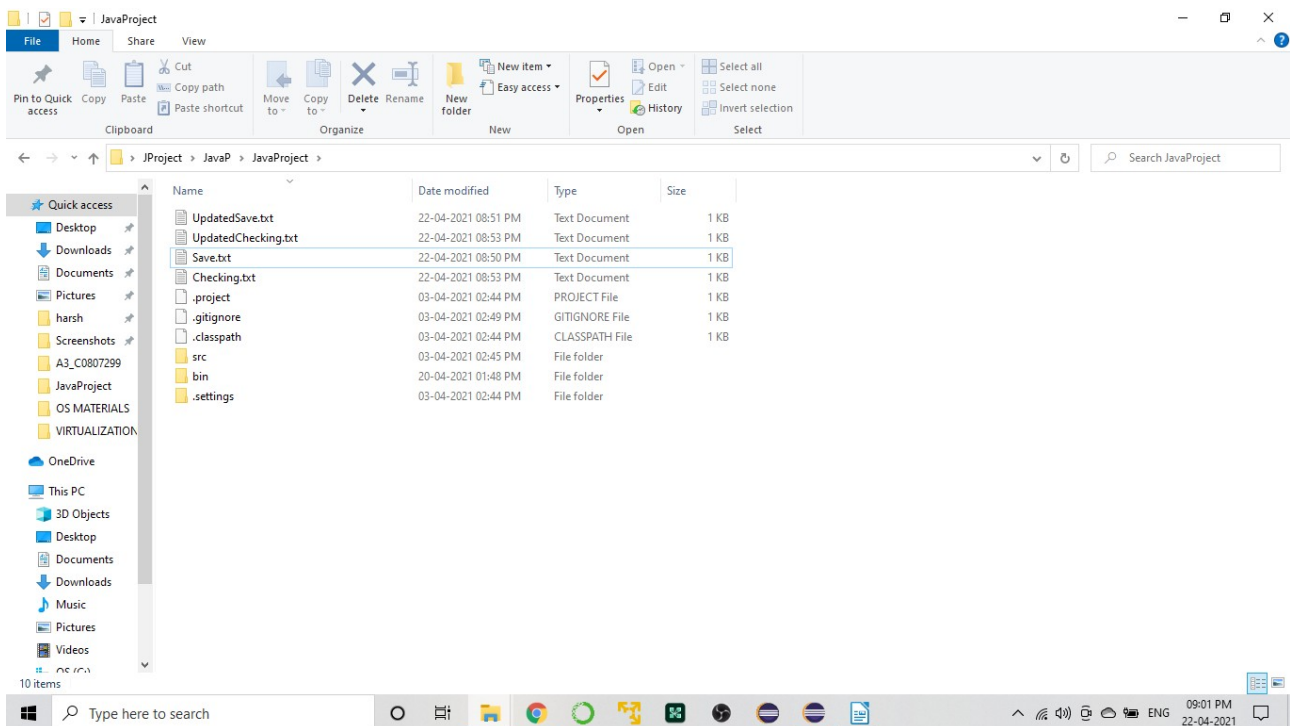
```

```

A To Create new account
B.To view Current Balance
C.To view Deposit Money
D.To view Withdraw Money[$$ Withdrawal Charegss]
E.To view Previous Transaction
F.To Transfer Money
G.To pay utility bill.
I.To View and Update Details
H.To Exit
Enter your option:
I
Enter File Name:save
save (The system cannot find the file specified)

```

- As, the validation failed to find the particular file name the user is not displayed with the requested file details.



- As seen in the screenshot the two files created are the updated ones are dynamic names taken during the file validation and saves the updated details taken from the user.

#Source Code

```
/**
 *
 */
package JavaProject;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.text.NumberFormat;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Random;

import javax.swing.JOptionPane;

//import jdk.internal.misc.FileSystemOption;

/**
 * @author HarshilSoni299
 *
 */
public class JavaProject {

    /**
     * @param args
     */
    // Variables used for transferring money between accounts
    int Saving,Checking,saveacc,checkacc;
    // Variables used here are used in the methods defined below
    int Current_Balance,deposit_money,withdraw_money,transaction,fee = 5;
    //Create Account Variables
    String Customer_Name, Customer_Mail, Country, Postal,phone,DOB;
    //Random class is created and will generate Random numbers for the Account types Savings
and Checking
    Random rand = new Random();

    //Create Account Method gives option to user to either create Savings or Checking Account
    public void Create_Acc()
    {
        System.out.println("Enter the Account to be Created");
        System.out.println("1. Press '1' to create Savings Account\n"+"2. Press '2' to create
Checking Account");
        Scanner scanner = new Scanner(System.in);
        int input;
        input=scanner.nextInt();
```



```

if(input==1)
{
    //JOptionPane shows the welcome message.
    JOptionPane.showMessageDialog(null, "Welcome to Savings Account");
    //Enter Customer Name
    Customer_Name=JOptionPane.showInputDialog("Customer Name");
    //Dialog box shows entered name
    JOptionPane.showMessageDialog(null,"Customer Name is "+Customer_Name);

    //Enter Mail Address
    Customer_Mail = JOptionPane.showInputDialog("Enter Mail Address");
    //Dialog box shows entered Mail Address
    JOptionPane.showMessageDialog(null,"Mail Id is "+Customer_Mail);

    //Enter Phone Number
    phone=JOptionPane.showInputDialog("Enter your phone number");
    //Dialog box shows entered Phone Number
    JOptionPane.showMessageDialog(null,"Phone number is"+phone);

    //Enter Date of Birth
    DOB=JOptionPane.showInputDialog("Date of Birth");
    //Dialog box shows entered date of birth
    JOptionPane.showMessageDialog(null,"Date of Birth is "+DOB);

    //Enter Country
    Country=JOptionPane.showInputDialog("Enter Country");
    //Dialog box shows entered Country name
    JOptionPane.showMessageDialog(null,"Country is "+Country);

    //Enter Postal Code
    Postal=JOptionPane.showInputDialog("Enter Postal Code");
    //Dialog box shows entered Postal Code
    JOptionPane.showMessageDialog(null,"Postal Code is "+Postal);

    //JOptionPane shows the success message.
    JOptionPane.showMessageDialog(null, "Successfully Saved the Saving Account
Details in File.");

    try{
        File file = new File("Save.txt");
        PrintWriter writer = new PrintWriter(file);
        writer.write("Customer name: "+ Customer_Name +"\n" + "Mail Address: "
+ Customer_Mail+"\n" + "Contact Info: " + phone+"\n"+"Date of Birth: " + DOB +"\n"+"Country:
"+Country+"\n"+"Postal Code: "+Postal);
        writer.close();
    }
    catch(IOException e)
    {
        e.printStackTrace();
    }

    System.out.println("Saving account created successfully");
}

```

```

else
{
    if(input==2)
    //OptionPane shows the welcome message.
    JOptionPane.showMessageDialog(null, "Welcome to Checking Account");
    //Enter Customer Name
    Customer_Name=JOptionPane.showInputDialog("Customer Name");
    //Dialog box shows entered name
    JOptionPane.showMessageDialog(null,"Customer Name is "+Customer_Name);

    //Enter Mail Address
    Customer_Mail = JOptionPane.showInputDialog("Enter Mail Address");
    //Dialog box shows entered Mail Address
    JOptionPane.showMessageDialog(null,"Mail Id is "+Customer_Mail);

    //Enter Phone Number
    phone=JOptionPane.showInputDialog("Enter your phone number");
    //Dialog box shows entered Phone Number
    JOptionPane.showMessageDialog(null,"Phone number is"+phone);

    //Enter Date of Birth
    DOB=JOptionPane.showInputDialog("Date of Birth");
    //Dialog box shows entered date of birth
    JOptionPane.showMessageDialog(null,"Date of Birth is "+DOB);

    //Enter Country
    Country=JOptionPane.showInputDialog("Enter Country");
    //Dialog box shows entered Country name
    JOptionPane.showMessageDialog(null,"Country is "+Country);

    //Enter Postal Code
    Postal=JOptionPane.showInputDialog("Enter Postal Code");
    //Dialog box shows entered Postal Code
    JOptionPane.showMessageDialog(null,"Postal Code is "+Postal);

    //OptionPane shows the success message.
    JOptionPane.showMessageDialog(null, "Successfully Saved Checking Account
Details in File.");

    try{
        File file = new File("Checking.txt");
        PrintWriter writer = new PrintWriter(file);
        writer.write("Customer name: " + Customer_Name + "\n" + "Mail Address: " +
Customer_Mail+"\n"+ "Contact Info: " + phone+"\n"+"Date of Birth: " + DOB + "\n"+"Country:
"+Country+"\n"+"Postal Code: "+Postal);
        writer.close();
    }
    catch(IOException e)
    {
        e.printStackTrace();
    }
}

```

```
}
```

```
/*Methods are created below
```

1. Deposit_Money()
2. Withdraw_Money()
3. Transaction()
4. transfer()
5. Utilty_Bills()
6. Update_Details()*/

```
//Deposit_Money method to deposit money
```

```
public int Deposit_Money (int deposit_money)
```

```
{
    if(deposit_money !=0 )
    {
        Current_Balance = Current_Balance+ deposit_money;
        transaction = deposit_money;
    }
    return transaction;
}
```

// Validates the transaction, then withdraws the specified amount from the account and \$10 fee is charged.

```
public int Withdraw_Money (int withdraw_money)
```

```
{
    if(withdraw_money !=0 )
    {
        Current_Balance = Current_Balance-withdraw_money - fee ;
        transaction = - withdraw_money;
    }
    return transaction;
}
```

```
//Returns Previous Transaction
```

```
public void Transaction()
```

```
{
    if(transaction > 0)
    {
        System.out.println("Deposited: " + transaction);
    }
    else if(transaction < 0)
    {
        System.out.println("Withdraw: " +Math.abs(transaction));
    }
    else
    {
        System.out.println("No Transaction");
    }
}
```

```
//Transfers money from one bank to another and vice-versa
```

```

public void transfer(int withdraw_money)
{
    saveacc=rand.nextInt(900000000);
    checkacc=rand.nextInt(900000000);
    System.out.println("Select the account to transfer money");
    Scanner scanner = new Scanner(System.in);
    String c =scanner.next();
    if("saving"==c)
    {
        if(deposit_money >= withdraw_money)
        {
            Saving = deposit_money-withdraw_money;
            Checking = Checking + withdraw_money;
            System.out.println("FundS Withdrawed from Saving Acccount "+saveacc+" balance
is " +Saving);
            System.out.println("FundS deposited in the Checking Account "+checkacc+" balance
is "+Checking);
            System.out.println("Funds successfully transfered.");
        }
        else
        {
            System.out.println("Insufficient funds");
        }
    }
    else
    {
        if(deposit_money >= withdraw_money)
        {
            Checking = deposit_money-withdraw_money;
            Saving = Checking + withdraw_money;
            System.out.println("FundS Withdrawed from Checking Acccount "+saveacc+"
balance is " +Checking);
            System.out.println("FundS deposited in the Saving Account "+checkacc+"
balance is "+Saving);
            System.out.println("Funds successfully transfered.");
        }
        else
        {
            System.out.println("Insufficient funds");
        }
    }
}

```

//Pays utility bills from the account

```
public void Utility_Bills()
```

```

{
    int num = 0;
    //Variables for bills
    int Electricity_Bill, Hydro_Bill, Gas_Bill, Maintenance_Bill = 0;
    //Welcome message
    System.out.println("Welcome to the bill payment window!!");
    Scanner sc = new Scanner(System.in);

```

```

//Menu to select the type of the bill for payment
System.out.println("Select the bill type:");
System.out.println("1. Electric bill.");
System.out.println("2. Hydro bill.");
System.out.println("3. Gas bill.");
System.out.println("4. Maintenance bill.");
num = sc.nextInt();
switch(num)
{
case 1:
    System.out.println("Enter the electric bill amount:");
    Electricity_Bill = sc.nextInt();
    if(Current_Balance>=Electricity_Bill)
    {
        Current_Balance = Current_Balance-Electricity_Bill;
        System.out.println("Transaction successful.");
    }
    else
    {
        System.out.println("Insufficient balance.");
    }

case 2:
    System.out.println("Enter the Hydro bill amount:");
    Hydro_Bill = sc.nextInt();
    if(Current_Balance>=Hydro_Bill)
    {
        Current_Balance = Current_Balance - Hydro_Bill;
        System.out.println("Transaction successful.");
    }
    else
    {
        System.out.println("Insufficient balance.");
    }

case 3:
    System.out.println("Enter the gas bill amount:");
    Gas_Bill = sc.nextInt();
    if(Current_Balance>=Gas_Bill)
    {
        Current_Balance = Current_Balance - Gas_Bill;
        System.out.println("Transaction successful.");
    }
    else
    {
        System.out.println("Insufficient balance.");
    }

case 4:
    System.out.println("Enter the maintenance bill amount:");
    Maintenance_Bill = sc.nextInt();
    if(Current_Balance>=Maintenance_Bill)
    {
        Current_Balance = Current_Balance-Maintenance_Bill;

```

```

        System.out.println("Transaction successful.");
    }
    else
    {
        System.out.println("Insufficient balance.");
    }
    default:
        System.out.println("Enter valid option.");
}

}

//Update Details Method to view details from the file and update details
public void Update_Details() throws IOException
{
    //Enter the file name
    Scanner Read = new Scanner(System.in);
    System.out.printf("Enter File Name: "+"\\t"+"Note: Enter 'Save.txt' or
'Checking.txt'+"\\n");
    String fileName = Read.next();
    int answer;
    String Customer,Mail,Phone,country,postcode,line= " ";
    int choice;

    try
    {
        File inFile = new File(fileName);
        Scanner in = new Scanner(new File(fileName));
        System.out.println("Correct the File name is "+inFile.getName());

        if(inFile!=null)
        {
            BufferedReader br = new BufferedReader(new FileReader(inFile));
            String st;
            while ((st = br.readLine()) != null)
                System.out.println(st+"\\n");
        }

    }
    //if filename is incorrect the user cannot view the details from the file
    catch (FileNotFoundException ex)
    {
        System.out.println(ex.getMessage());
    }

    //Asks if the user want to update the details
    System.out.println("Do you want to update the details(Click 1 to continue)");
    answer=Read.nextInt();

    if(answer!=0)
    {
        Customer=JOptionPane.showInputDialog("Customer Name");
    }
}

```

```

Mail=JOptionPane.showInputDialog("Update Mail Address");

Phone=JOptionPane.showInputDialog("Update Phone Number");

DOB=JOptionPane.showInputDialog("Update Date of Birth");

country=JOptionPane.showInputDialog("Update Country");

postcode=JOptionPane.showInputDialog("Update Postal Code");

try
{
    /*It will take filename dynamically either savings or checking
    when typed during validation filename.*/
    File f = new File("Updated"+fileName);
    FileWriter file = new FileWriter(f);
        file.write("Updated Information"+"\\n"+"Customer name: "+ Customer +"\\n"
+ "Mail Address: " + Mail+"\\n"+ "Contact Info: "+ Phone+"\\n"+"Date of Birth: "+ DOB +"\\
n"+"Country: "+country+"\\n"+"Postal Code: "+postcode);
        file.close();
    }
    catch(IOException e)
    {
        e.printStackTrace();
    }
    System.out.println("Saving account created successfully");
}

else
{
    System.out.println("Successfully updated details in Savings Account");
}
}

```

/*Bank_Details Method contains switch case which will call all the defined methods*/

public int Bank_Details() throws IOException

```

{
    char option = '\0';
    Scanner scanner = new Scanner(System.in);

    do
    {
        System.out.println("A To Create new account");
        System.out.println("B.To view Current Balance");
        System.out.println("C.To view Deposit Money");
        System.out.println("D.To view Withdraw Money[5$ Withdrawal Chargess]");
        System.out.println("E.To view Previous Transaction");
        System.out.println("F.To Transfer Money");
    }
}

```

```
System.out.println("G.To pay utility bill.");
System.out.println("I.To View and Update Details");
System.out.println("H.To Exit");
System.out.println("Enter your option:");
option = scanner.next().charAt(0);
```

```
switch(option)
{
```

```
    case 'A':
```

```
        Create_Acc(); //It will call the account to either create savings
or checking account
```

```
    case 'B':
```

```
        System.out.println("Current_Balance= " +Current_Balance); //
to view current balance of the accounts
        break;
```

```
    case 'C':
```

```
        System.out.println("Deposit Money"); //It will call the deposit
money method

        deposit_money = scanner.nextInt();
        Deposit_Money(deposit_money);
        break;
```

```
    case 'D':
```

```
        System.out.println("Withdraw Money"); //It will call the
withdraw method to withdraw money

        withdraw_money = scanner.nextInt();
        Withdraw_Money(withdraw_money);
        System.out.println("\n");
        break;
```

```
    case 'E':
```

```
        Transaction(); //To view the previous transaction
        break;
```

```
    case 'F':
```

```
        transfer(withdraw_money); //Transfer method will transfer
money within accounts

        break;
```

```
    case 'G':
```

```
        Utility_Bills(); //To pay different bills
        break;
```

```
    case 'I':
```

```
        Update_Details(); // To view and update the account details
from the files

        break;
```

```
    default:
```

```
        System.out.println("Thanking you for Banking with us");
```



```

        break;
    }
}
while(option!= 'H');
{
    System.out.println("Have a good day!");
}
return 0;
}

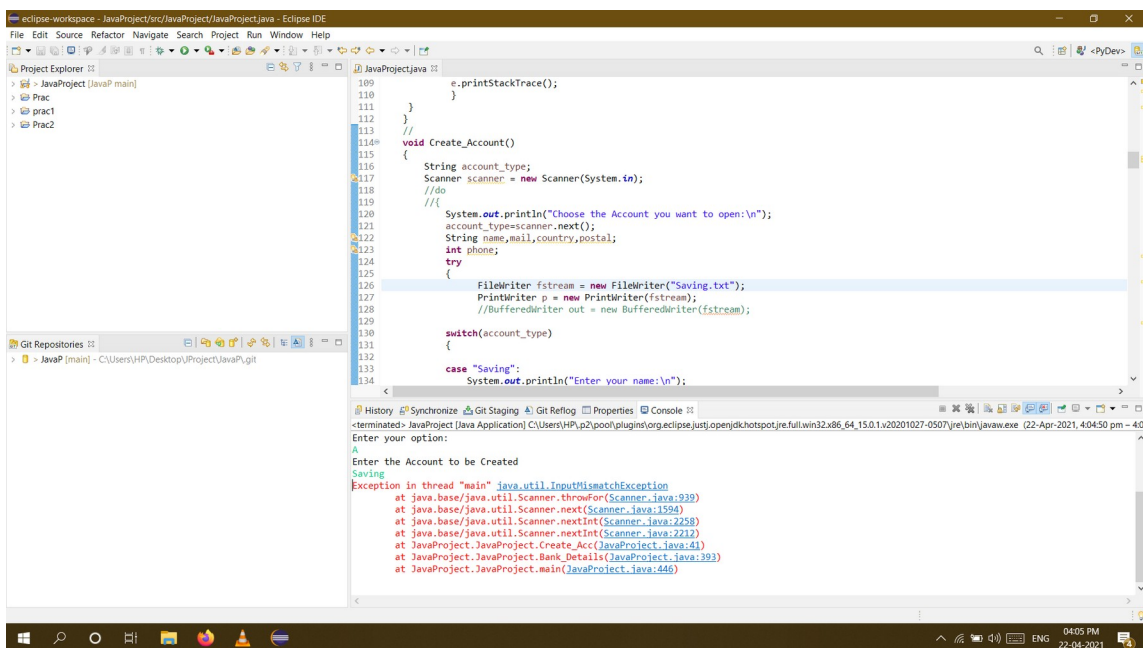
public static void main(String[] args) throws IOException {

    // Object obj is created.
    JavaProject obj = new JavaProject();
    obj.Bank_Details(); // Bank_Details method is called which contains the switch case.
}

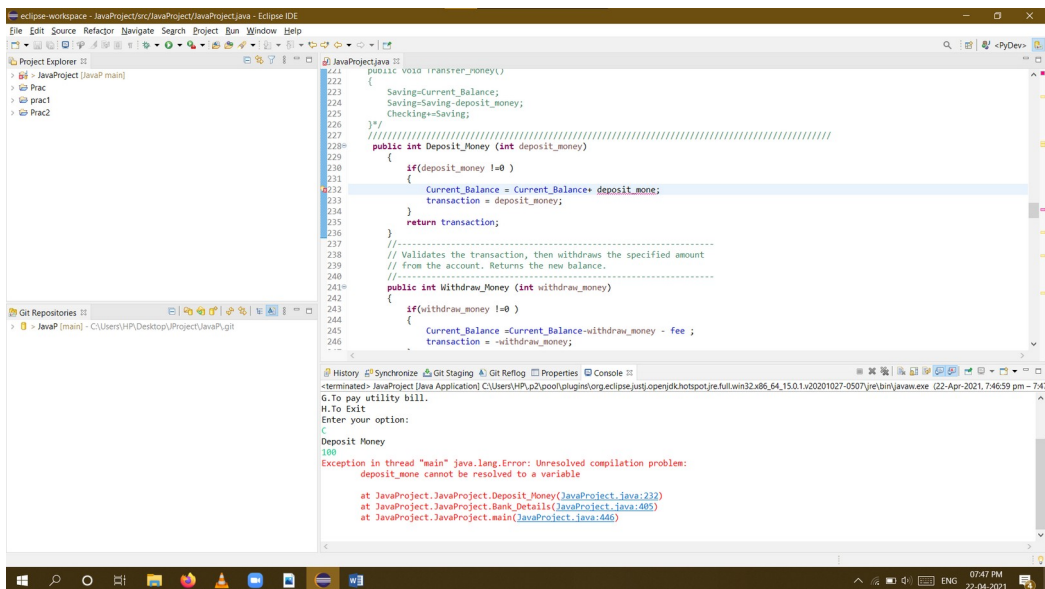
}

```

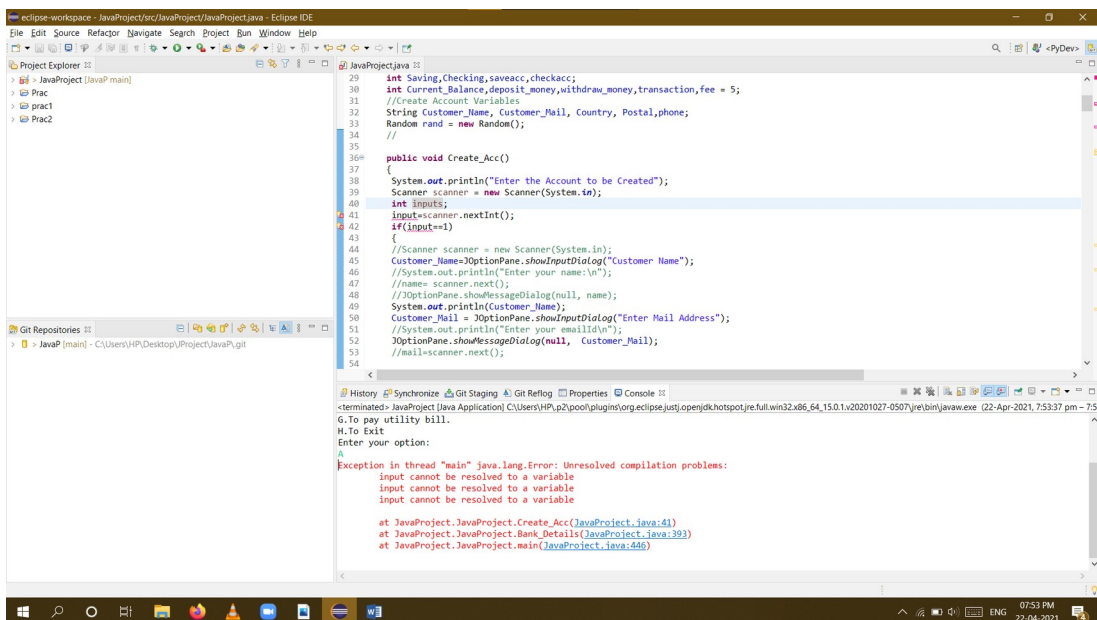
#Some Screenshots showing the errors or exceptions during testing the system.



- This error was caused due to the file mismatch where user enters to create an account and needs to enter the filename but generates mismatch error. However to resolve this issue I used file validation and else condition so code doesn't end abruptly but user is given a chance to enter something else with a prompt message.



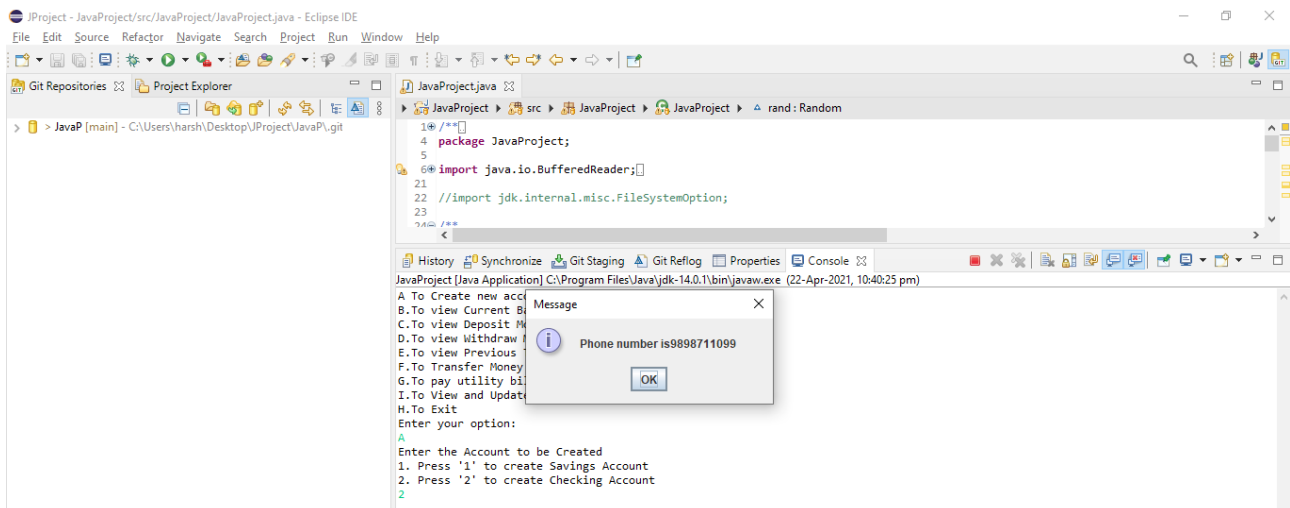
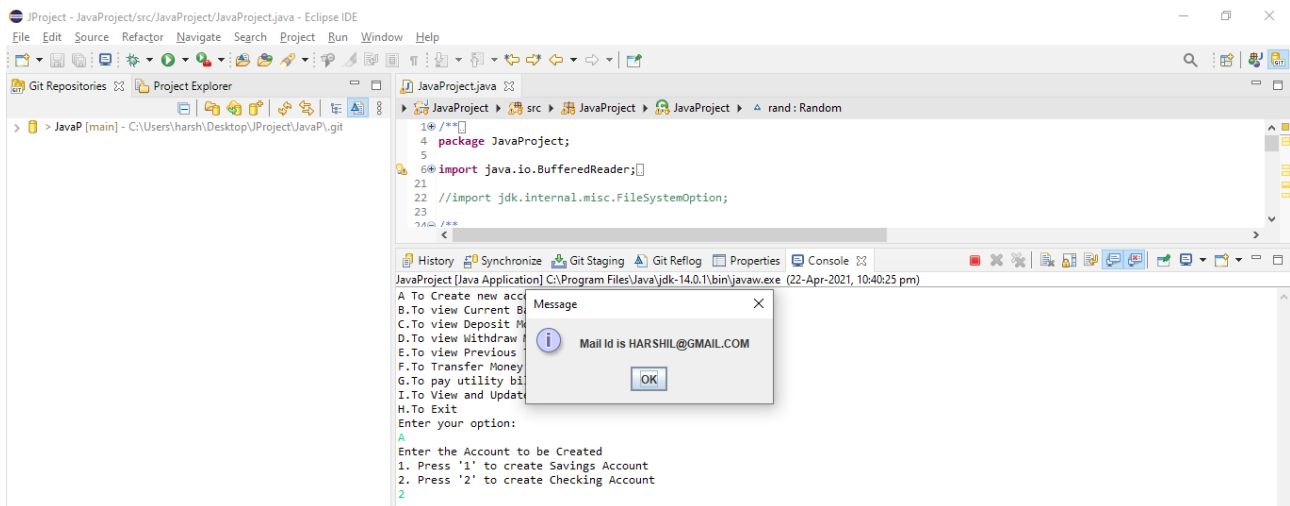
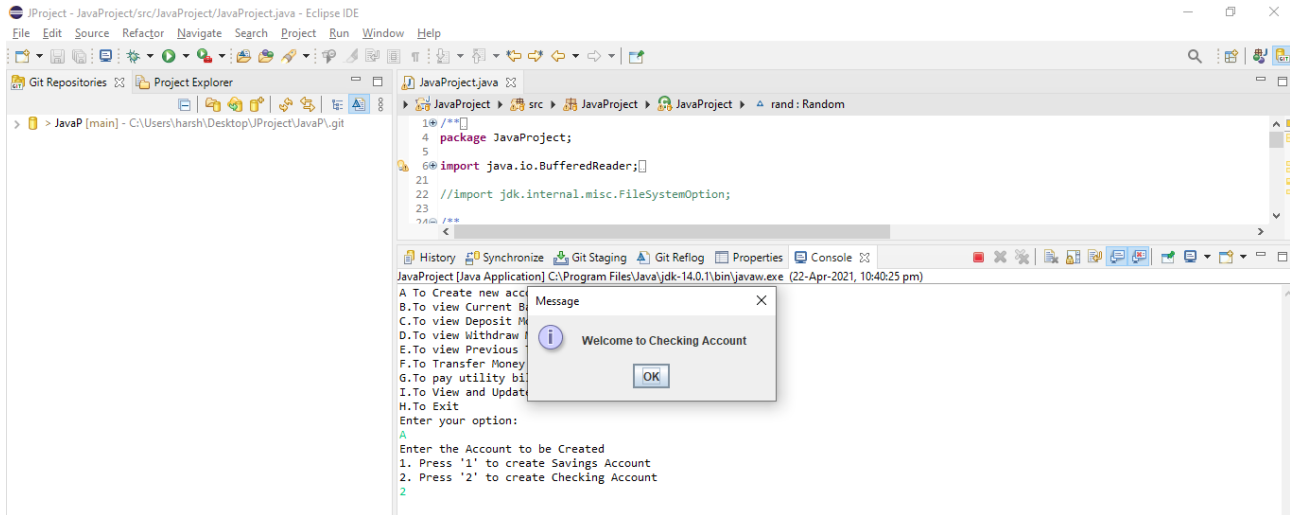
- This error was caused due to the invalid variable name.

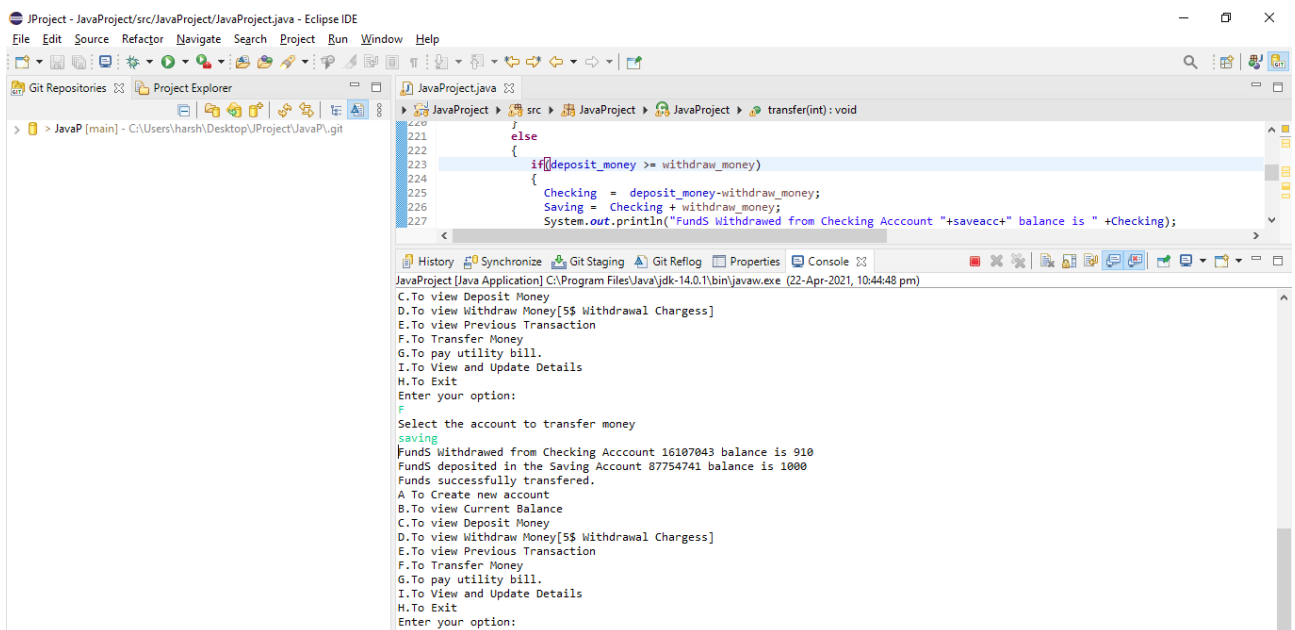
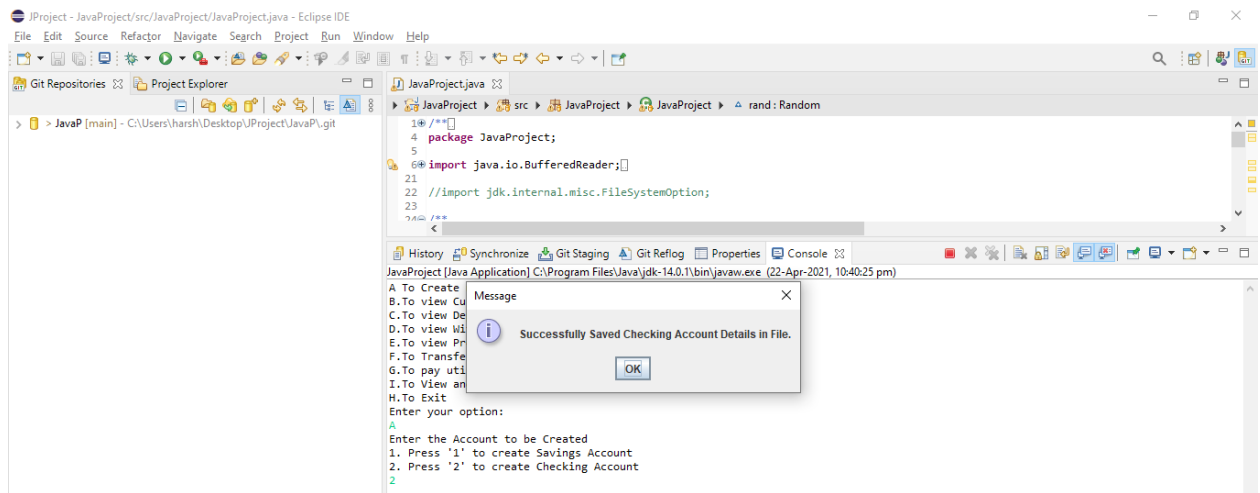
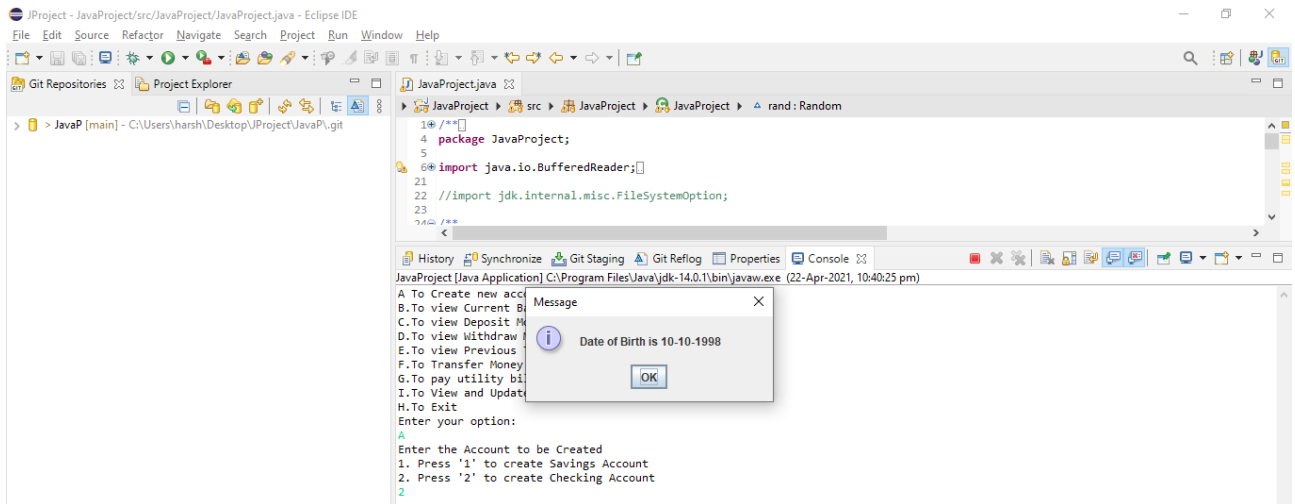


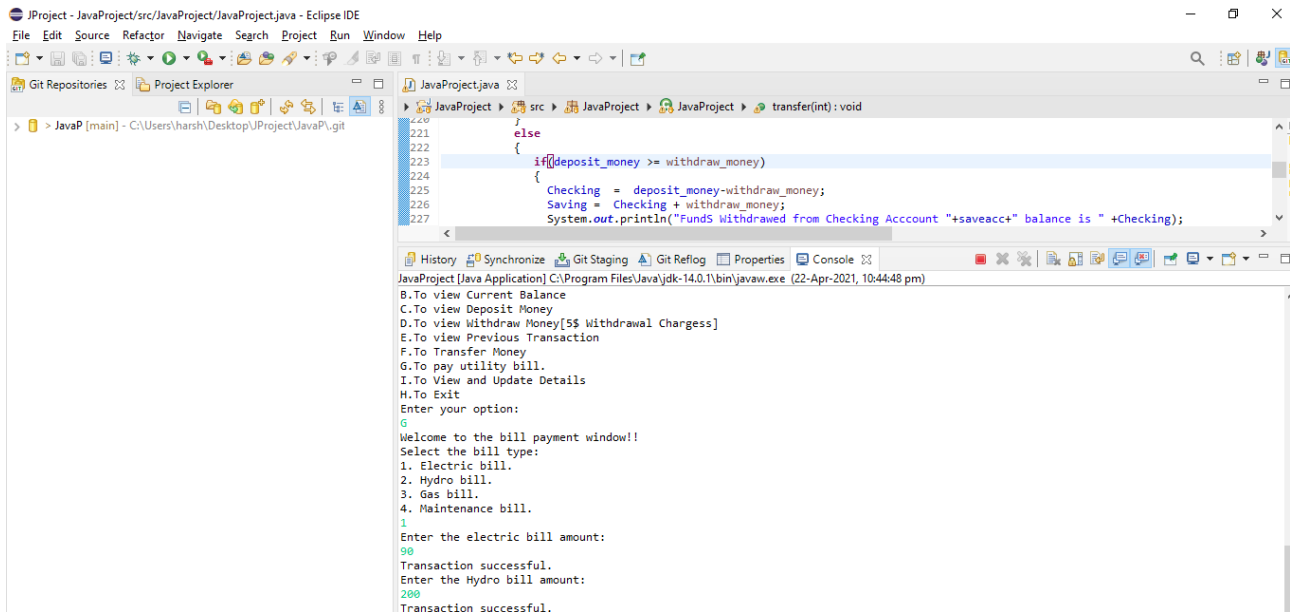
- During the debugging the user needs to enter an option with a character but integer data type was used and later changed the type to String.

- Lastly, there was an internal error which was not displayed on console but I was not able to write data into the file even after using buffer methods and filewriter. So, lastly we came to a conclusion to use File and instead of overriding in the same file I replicated it in the different filename and used dynamic file names for savings and checking accounts.

#Screenshots showing the system is working without errors







The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The Project Explorer on the left shows a project named 'JavaP' with a main folder. The central editor displays a Java file named 'JavaProject.java' with the following code:

```
221  
222  
223     else  
224     {  
225         if(deposit_money >= withdraw_money)  
226         {  
227             Checking = deposit_money-withdraw_money;  
228             Saving = Checking + withdraw_money;  
229             System.out.println("Funds Withdrawed from Checking Account "+saveacc+" balance is " +Checking);  
230         }  
231     }
```

The bottom console window shows the output of the Java application:

```
JavaProject [Java Application] C:\Program Files\Java\jdk-14.0.1\bin\javaw.exe (22-Apr-2021, 10:44:48 pm)  
B.To view Current Balance  
C.To view Deposit Money  
D.To view Withdraw Money[5$ Withdrawal Chagess]  
E.To view Previous Transaction  
F.To Transfer Money  
G.To pay utility bill.  
I.To View and Update Details  
H.To Exit  
Enter your option:  
6  
Welcome to the bill payment window!!  
Select the bill type:  
1. Electric bill.  
2. Hydro bill.  
3. Gas bill.  
4. Maintenance bill.  
1  
Enter the electric bill amount:  
90  
Transaction successful.  
Enter the Hydro bill amount:  
200  
Transaction successful.
```

#Name(s) of the file(s) the system is creating, and copy of their details

1. Save.txt file is created and used for storing Savings account information.
2. Checking.txt file is created and used for storing Checking account information.
3. UpdatedSave.txt is used to store the updated data in this file(Note:- Save.txt and UpdatedSave.txt file has different data and not replicated.)
4. UpdatedChecking.txt is used to store the updated data in this file(Note:- Checking.txt and UpdatedChecking.txt file has different data and not replicated.)

#List of References.

1. <https://forum.knime.com/t/how-to-create-dynamic-filename/5410/4>
2. <https://www.baeldung.com/java-write-to-file>
3. <https://www.javatpoint.com/java-filewriter-class>
4. <https://stackabuse.com/java-save-write-string-into-a-file/>
5. <https://www.dreamincode.net/forums/topic/165046-anyone-know-a-good-way-to-transfer-funds-between-2-accounts/>