

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/282218734>

ATM MACHINE PROGRAMMING WITH JAVA

Conference Paper · December 2014

DOI: 10.13140/RG.2.1.4945.4560

CITATIONS

0

READS

64,549

1 author:



Syed Hasan

Asia Pacific University of Technology and Innovation

20 PUBLICATIONS 2 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Online Assignment Submission & Feedback System [View project](#)

Table of Contents

ACKNOWLEDGEMENT	2
INTRODUCTION	3
EXPLANATIONS	4
<i>First Things First:</i>	4
<i>Execution Procedures:</i>	5
<i>Function (Balance):</i>	7
<i>Function (Deposit):</i>	7
<i>Function (Withdraw):</i>	8
<i>Function (Transfer):</i>	8
<i>Function (Change Pin):</i>	9
<i>NB:</i>	10
OBJECT ORIENTTED EXPLANATION	11
<i>Object Oriented Samples:</i>	11
<i>Sample 1:</i>	11
<i>Sample 2:</i>	12
<i>Sample 3:</i>	13
ASSUMPTION	14
ADDITIONAL FEATURES	15
<i>Feature 1:</i>	15
<i>Feature 2:</i>	16
<i>Sample Code For Validation:</i>	16
<i>Feature 3:</i>	17
<i>Sample Code For Password Character:</i>	17
<i>Feature 4:</i>	18
<i>Sample Code For Withdraw Validation:</i>	18
REFERENCE.....	19
Bibliography	19

ACKNOWLEDGEMENT

First of all, I would like to thank my lecturer **Mr. KAU GUAN KIAT** for helping me to acquire some basic knowledge of “**Java Programming Language**”. At the same time, he gave me the opportunity to learn something new related to our module like constructors, methods, arrays, JFrames etc.

Beside from my lecturer, I like to thank my other classmates for helping to understand the assignment related questions more clearly. They gave their best for completing this report on time. I thank them for their efforts.

INTRODUCTION

This assignment is based on developing an ATM (Automated Teller machine) using “**Java Programming Language**”. For that we used GUI (Graphical User Interface) in this development so that it will become more users friendly to interact.

Besides, we also added text files for user’s records that are directly linked with this program. It is so called a heart of this program where all the functions depend on it.

EXPLANATIONS

In this documentation we have given explanations of how to interact successfully with this ATM (Automated Teller Machine). We have explained here step by step so that it will surely help users to become more user friendly with it. Below are our explanations:

First Things First:

Before execute this program users need to do some works so that it will run properly into their system. First they need to make sure their system is having “JDK”. If they don’t have it then they can download from this below link:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>

Depending on their system (Windows 64bit/32bit) they need to download and install. Then they need to add the “JAVA” files to their system “PATH” so that the system can run the program from CMD (Command Prompt). The path will show something like this “**C:\Program Files (x86)\Java\jre1.8.0_25\bin;**”. Now just add the address besides the current path directory and save it.

The other way they can execute this program in to download the IDE (Integrated Development Environment) on their system. They can download ECLIPSE or NETBEANS depending on the windows (32bit/64bit). Below is the link:

NETBEANS:

<https://netbeans.org/downloads/>

ECLIPSE:

<http://www.eclipse.org/downloads/>

We developed this program using “**ECLIPSE**”.

Execution Procedures:

When user executes this program it will show the details (username, password and current balance) from the text file in the console. They will also see the startup GUI (Graphical User Interface) of this program.

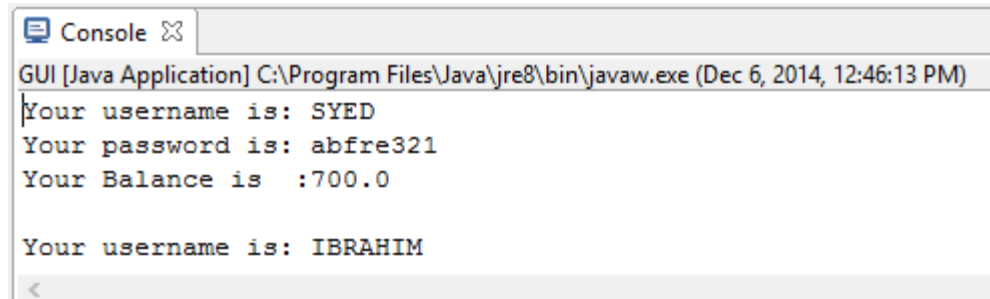


FIGURE 1: CONSOLE

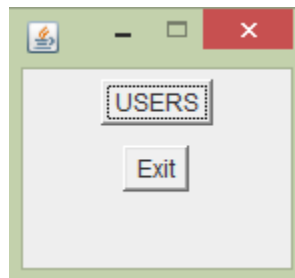


FIGURE 2: STARTUP GUI

User need to click the button “**USERS**” for further proceed. If they click the button “**Exit**”, it will terminate the program.

When user click the button “USERS” it will show the new GUI for entering their registered username and password. They need to enter the correct username and password.

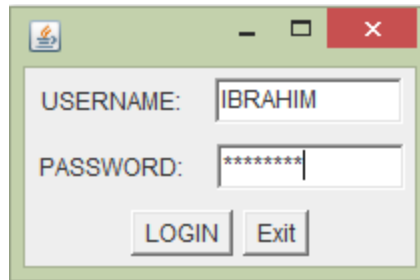


FIGURE 3: USERS LOGIN

If they enter invalid username or password they system will show the warning message.

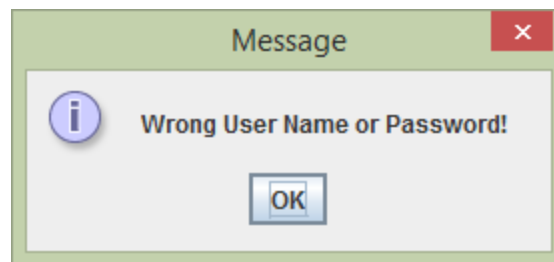


FIGURE 4: ERROR

If the user enters valid username and password then it will show the “MENU” on their screen.

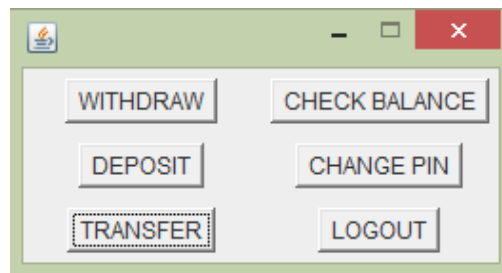


FIGURE 5: MENU

Function (Balance):

When users click the button “BALANCE”, it will show their current balance.

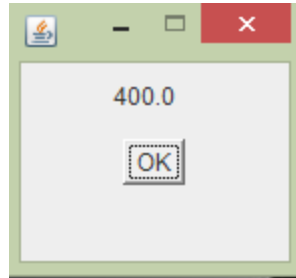


FIGURE 6: CURRENT BALANCE

Function (Deposit):

To deposit into the account click on the button “**DEPOSIT**”. It will show a deposit GUI on the screen. They need to enter the amount to deposit into their account.

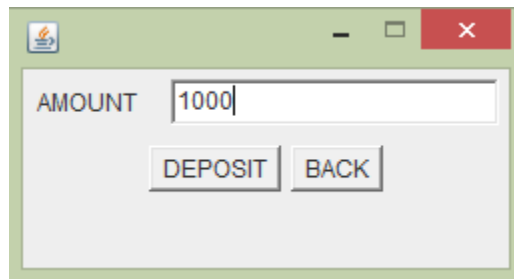


FIGURE 7: DEPOSIT AMOUNT

To check the new balance click on the button “**BALANCE**”. It will show the users new balance after deposit.

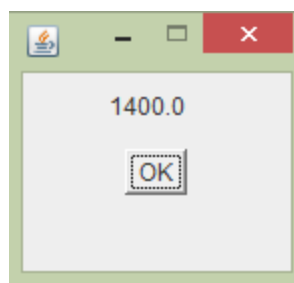


FIGURE 8: NEW BALANCE

Function (Withdraw):

To withdraw from their account, click on the button “WITHDRAW”. It will pop out new withdraw screen where user need to key in the total amount to withdraw from their account.

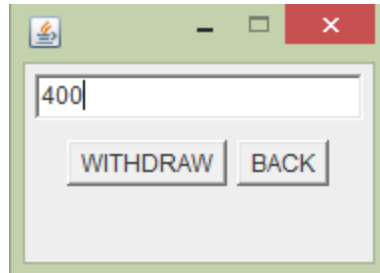


FIGURE 9: WITHDRAW

To check new balance after withdraw, click on the button “**BALANCE**”. It will show a new balance.

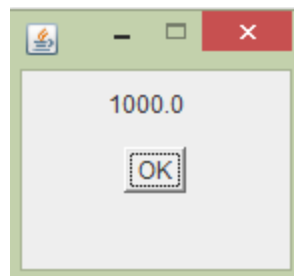


FIGURE 10: NEW BALANCE

Function (Transfer):

User can also transfer amount to other users account. For that, user needs to click on the button “TRANSFER”. It will show the new transfer GUI. Then key in the other user name and the total amount for transfer. As for example other users like “SYED” having RM 700 in his account. If SYED receive more from other user it will add to his account and the total account will become RM 1200.

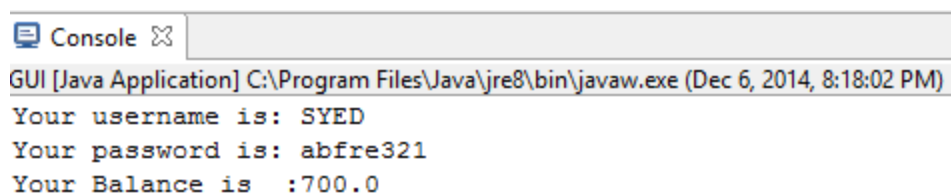


FIGURE 11: CONSOLE

Now user wants to transfer the amount to SYED. After entering the name and amount click on the button “TRANSFER”. It will transfer to SYED on his account.

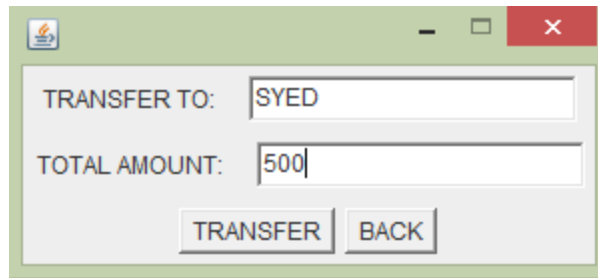


FIGURE 12: TRANSFER

Now check again the new balance. It will show that RM 500 has been deducted from user account and at the same time added to SYED’s account.

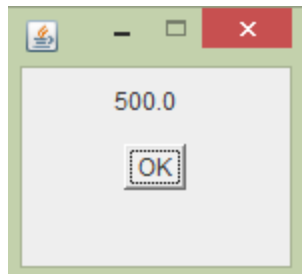


FIGURE 13: USER BALANCE

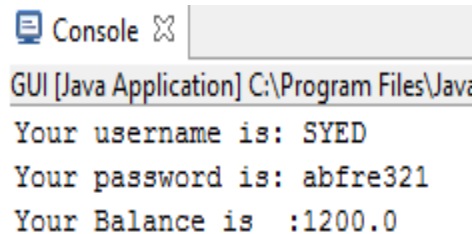


FIGURE 14: SYED's BALANCE

Function (Change Pin):

Users can also change their password (pin). For that they need to click the button “**CHANGE PIN**”. It will show the screen for changing the pin where users need to enter their current username and password for verification. Then click the button “**CHANGE**”.

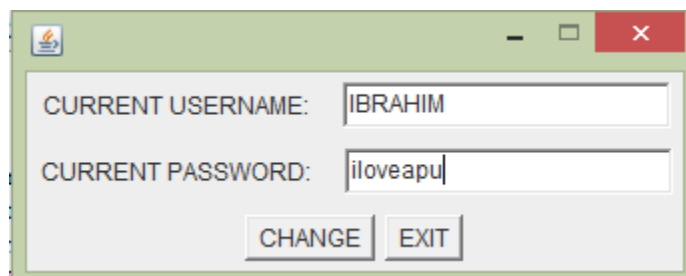


FIGURE 15: CHANGE PIN

It will appear a new “**JOPTIONPANE**” for entering the new password. Then click “**OK**”. It will change the current password to new password.

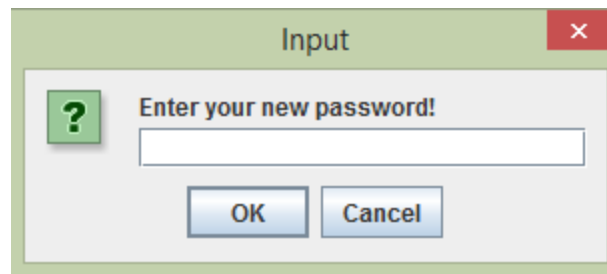


FIGURE 16: ENTER NEW PASSWORD

Now when user start the program they will see the new password has been updated into the file. It will show in their console.

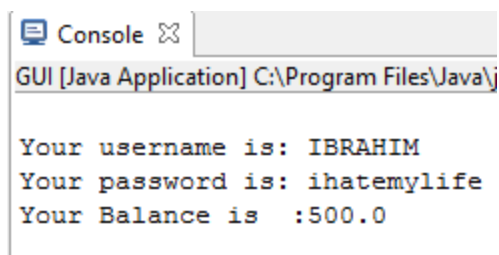


FIGURE 18: CURRENT PASSWORD

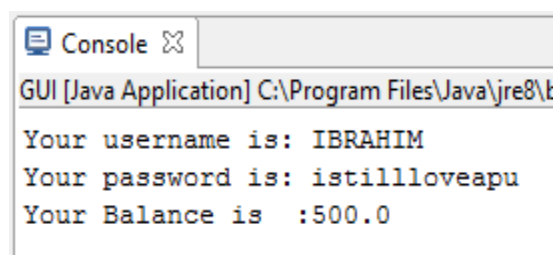


FIGURE 17: NEW PASSWORD

NB:

As we are using array where every details are there, user need to terminate the program to see the updated results in their file.

OBJECT ORIENTED EXPLANATION

In object-oriented programming, for example, an object is a self-contained entity that consists of both data and procedures to manipulate the data. In other way, object oriented is the software engineering concept where it is represented using the “**OBJECTS**”. Below are the objected oriented samples we used in this “**Java Programming Language**”:

Object Oriented Samples:

Sample 1:

We are using the “USER ARRAY” where every detail becomes store when the program launches. When users enter their username and password, the input read from the file for verification purposes.

```
public USER[] readFromFile() throws IOException {
    File q = new File("password.txt");
    Scanner y = new Scanner(q);
    USER[] z = new USER[10];
    String input1 = "";
    String input2 = "";
    double input3 = 0;
    int counter = 0;
    while(y.hasNext()){
        input1 = y.nextLine();
        input2 = y.nextLine();
        input3 = Double.parseDouble(y.nextLine());

        z[counter] = new USER(input1, input2, input3);
        counter++;
    }
    for(int i=0; i<10; i++){
        z[i].printDetails();
        System.out.println();
    }

    return z;
}
```

Sample 2:

For “LOGIN GUI” we are having the “Extends JFrame” where we declared the button, text field and labels for users’ interaction. This GUI we added with our “MAIN MENU”.

```
import java.awt.*;

public class LOGIN extends JFrame {
    public Button login, Exit;
    public TextField username, password;
    public Label username1, password1;
    private Panel a1, a2, a3;
    private UserGui x;

    public LOGIN() {
        login = new Button("LOGIN");
        Exit = new Button("Exit");
        username = new TextField(10);
        username1 = new Label("USERNAME:");
        password1 = new Label("PASSWORD:");
        password = new TextField(10);
        password.setEchoChar('*');
        a1 = new Panel();
        a2 = new Panel();
        a3 = new Panel();
        x = new UserGui();
        x.buildGUI();
    }
}
```

Sample 3:

This is the user class where we inserted all the details about users like username, password and their individual balance. When the console starts, it will read from the file and print it in the console.

```
1 import javax.swing.JOptionPane;
2
3
4
5 public class USER {
6     private String NAME;
7     private String PASSWORD;
8     private double balance;
9     private double transfer;
10    private double withdraw;
11    public USER(String a, String b , double c){
12        NAME = a;
13        PASSWORD = b;
14        balance = c;
15    }
16    public void printDetails(){
17
18        System.out.println("Your username is: "+NAME);
19        System.out.println("Your password is: "+PASSWORD);
20        System.out.println("Your Balance is :"+balance);
21    }
22    public String getName(){
23        return NAME;
24    }
25    public String getPassword(){
```

Console

```
<terminated> GUI [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (Dec 8, 2014, 12:17:
Your username is: JEREMY
Your password is: jer120
Your Balance is :50.0
```

ASSUMPTION

Although we tried to develop this program more user friendly, it has some limitations too. Some of the limitations we have given in details:

We did not add any registration function (only GUI) for users. Only the registered users can execute this program. All the details of the registered users have been inserted into the text file. For further implementation of this ATM program, developers need to connect with database. Here we are only using “**TEXT FILE**”.

We did not add any button for deleting the registered users from this program. If admin want to delete the users, he or she needs to open the text file “PASSWORD.TXT” and delete from there.

In future we can add more functions like scanning user’s finger print for verification. We can also add the function for third party payment (Users can pay their bills through atm machine).

ADDITIONAL FEATURES

In this we added some additional features for validation purposes. Below is on details:

Feature 1:

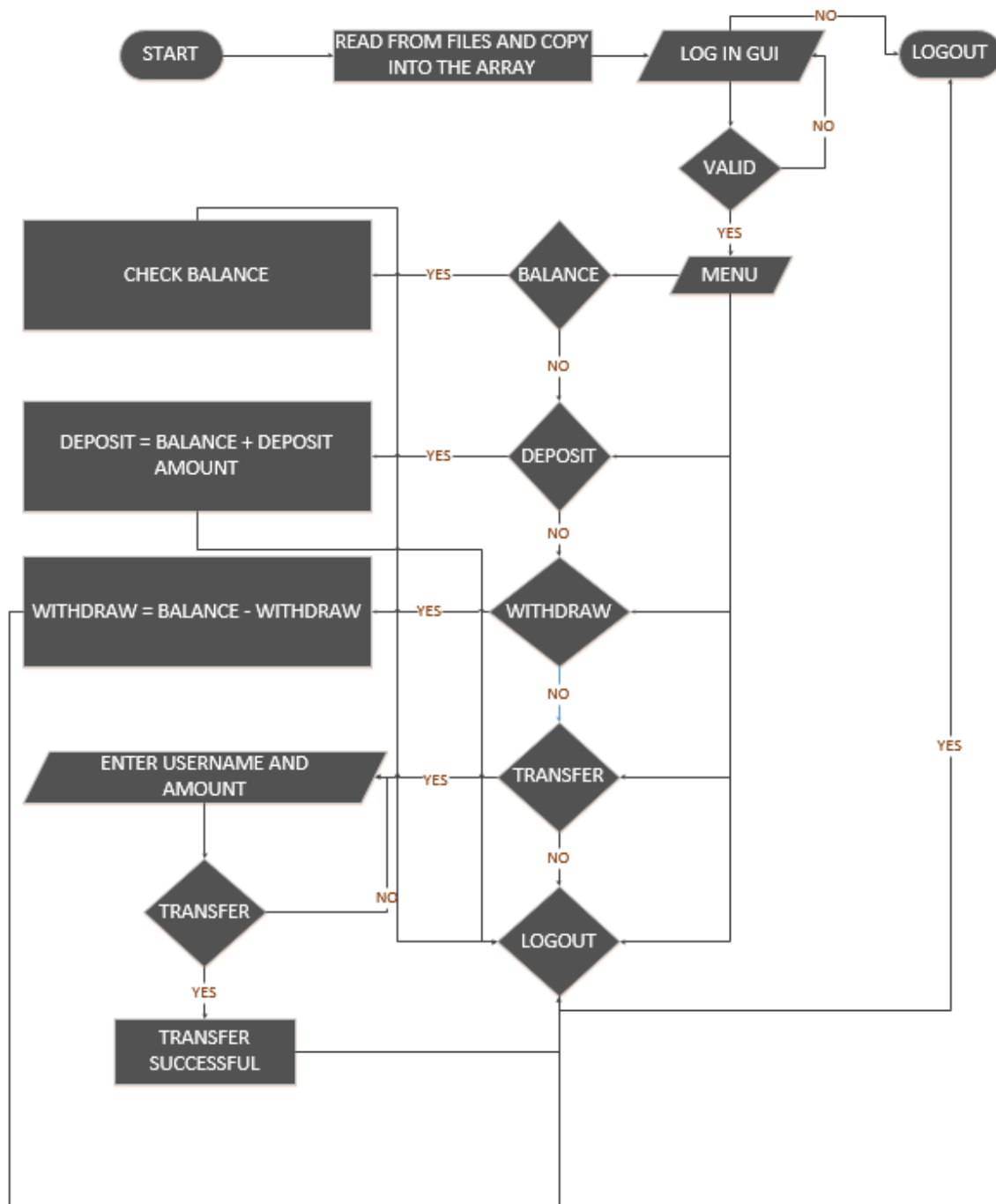


FIGURE 19: FLOW CHART

Feature 2:

We added the validation for log in. It will block unregistered users to log in into the system. If the users enter invalid username or password, it will show the error message.

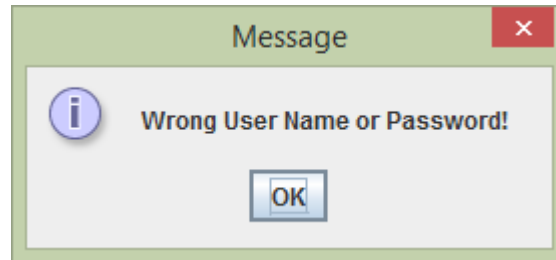


Figure 20: ERROR MESSAGE

Sample Code For Validation:

```
int i = 0;

String fromFile = "";
String fromFileA = "";
while(true){
    fromFile = f[i].getName();
    fromFileA = f[i].getPassword();
    if( name.equals(fromFile) && password.equals(fromFileA) ){
        userNo = i;
        a.setVisible(false);
        b.username.setText("");
        b.password.setText("");
        c.setVisible(true);
        break;
    }
    i++;
    if( i==10 ){
        a.setVisible(false);
        JOptionPane.showMessageDialog(null,"Wrong User Name or Password!");
        b.username.setText("");
        b.password.setText("");
        b.setVisible(true);
        break;
    }
}
```

Feature 3:

We also added special character for user password. It is because if any users type his or her user password so that no one can know except the user what he or she is typing.

Sample Code For Password Character:

```
public class LOGIN extends JFrame {
    public Button login, Exit;
    public TextField username, password;
    public Label username1, password1;
    private Panel a1, a2, a3;
    private UserGui x;

    public LOGIN() {
        login = new Button("LOGIN");
        Exit = new Button("Exit");
        username = new TextField(10);
        username1 = new Label("USERNAME:");
        password1 = new Label("PASSWORD:");
        password = new TextField(10);
        password.setEchoChar('*');
        a1 = new Panel();
        a2 = new Panel();
        a3 = new Panel();
        x = new UserGui();
        x.buildGUI();
    }
}
```

Feature 4:

We added the extra function for withdraw balance. If user input the total withdraw amount more than their current balance the system will show error message.

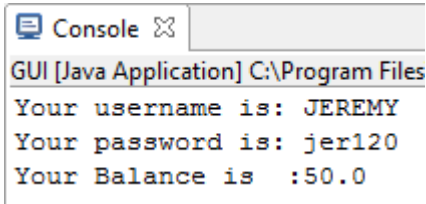


FIGURE 21: CONSOLE

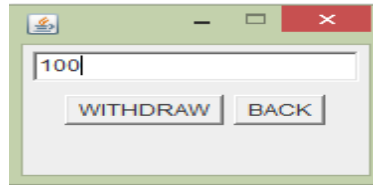


Figure 22: WITHDRAW

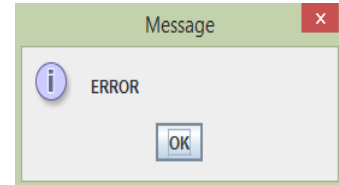


FIGURE 23: ERROR MESSAGE

Sample Code For Withdraw Validation:

```
public void withdrawbalance(double z){  
  
    if(z>balance){  
        JOptionPane.showMessageDialog(null,"ERROR");  
    }  
    else{  
        balance = balance - z;  
        JOptionPane.showMessageDialog(null,"SUCCESSFULL");  
    }  
}
```

REFERENCE

Bibliography

BRYANT, J., 2012. *JAVA 7 FOR ABSOLUTE BEGINNER*. NEW YORK: APRESS.

DREAMCODE.NET, 2014. *DREAMCODE.NET*. [Online]

Available at: <http://www.dreamincode.net/forums/topic/206344-basic-gui-in-java-using-jframes/>
[Accessed 5 December 2014].

EXAMPLES.JAVACODEGEEKS, 2014. *EXAMPLES.JAVACODEGEEKS*. [Online]

Available at: <http://examples.javacodegeeks.com/desktop-java/swing/jbutton/set-action-command-for-jbutton/>
[Accessed 5 December 2014].

HOME&LEARN, 2014. *HOME&LEARN*. [Online]

Available at: http://www.homeandlearn.co.uk/java/read_a_textfile_in_java.html
[Accessed 5 December 2014].

JAVA2S, 2014. *JAVA2S.COM*. [Online]

Available at: <http://www.java2s.com/Questions And Answers/Java-File/text-file/password.htm>
[Accessed 5 December 2014].

NETBEANS.ORG, 2014. *NETBEANS.ORG*. [Online]

Available at: <https://netbeans.org/kb/articles/gui-functionality.html>
[Accessed 7 December 2014].

ORACLE.COM, 2014. *docs.oracle.com*. [Online]

Available at: <http://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>
[Accessed 5 December 2014].

PROGRAMCREEK.COM, 2008. *PROGRAMCREEK.COM*. [Online]

Available at: <http://www.programcreek.com/2011/03/java-write-to-a-file-code-example/>
[Accessed 5 December 2014].

REALAPPLETS.COM, 2014. *REALAPPLETS.COM*. [Online]

Available at: <http://www.realapplets.com/tutorial/actionexample.html>
[Accessed 5 December 2014].

STACKOVERFLOW, 2011. *stackoverflow*. [Online]

Available at: <http://stackoverflow.com/questions/4716503/best-way-to-read-a-text-file>
[Accessed 6 December 2014].

TUTORIALPOINT, 2014. *TUTORIALPOINT*. [Online]

Available at: http://www.tutorialspoint.com/javaexamples/java_simple_gui.htm

[Accessed 4 December 2014].