**TABLE OF CONTENTS**

INTRODUCTION ii

EXPLANATIONS iii

*First Things First* iii

*Execution Procedures* iv

*Function (withdraw)* v

*Function (deposit)* v

UML-DIAGRAM ......................................................................................................................... vi

OBJECT ORIENTED SAMPLE ..............…………………………………………………. vii-viii

LIMITATIONS ………………………….……………………………………………………... ix

**INTRODUCTION**

This assignment is based on developing an ATM (Automated Teller Machine) using “Java Programming Language.” This was a semester project for my Computer Science Programming II class (FALL 2017). For this project, we were told only to develop a console application to allow us to focus strictly on the logic of this program.

**EXPLANATIONS**

In this documentation, I have given explanations of how to interact successfully with this ATM (Automated Teller Machine) simulator below. This will help make the program more user friendly.

**First Things First:**

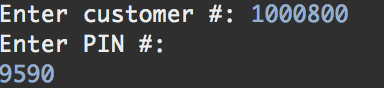
Before executing this program, users will need to download and install Java SE Development Kit to their operating system if they do not already have it. Otherwise this program will not run. Depending on your operating system, you may need to add the “JAVA” files to your system “PATH” so that the system can run the program from the CMD (Command Prompt or Shell).

You may also download a IDE (Integrated Development Environment) on your system such as Eclipse or NetBeans. I developed this program using Eclipse.

**Execution Procedures:**

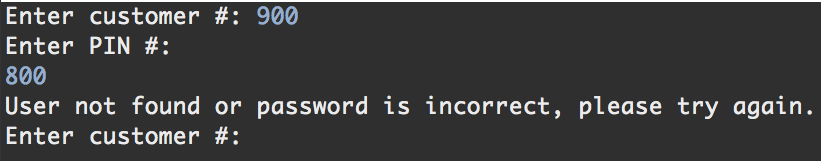
**USER LOGIN:**

When users execute this program, the user will be prompted to enter their customer # followed by their password.



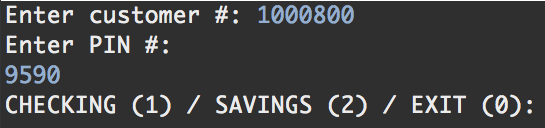
**ERROR:**

If the user enters an invalid customer # and/or password, they will be prompted to try again.

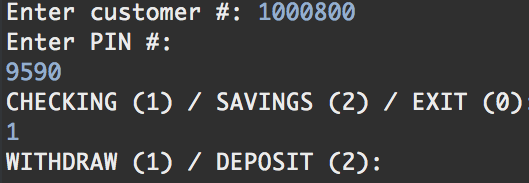


**MENU:**

If the user enters a valid customer # and password, the program will prompt the user to select which bank account the user wants to deposit or withdraw from.

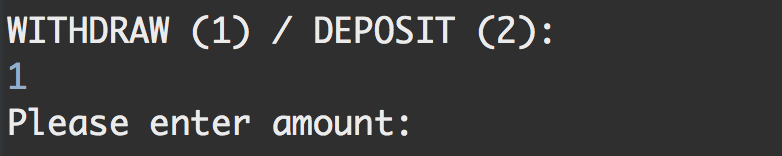


If the user chooses to select their checking or savings account, the program will prompt the user to select whether they want to deposit or withdraw from their bank account.



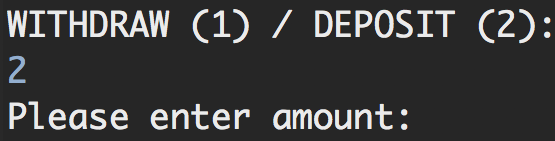
***Function (Withdraw):***

When the user chooses the “WITHDRAW” option, it will allow user to enter the amount they want to withdraw from selected account.

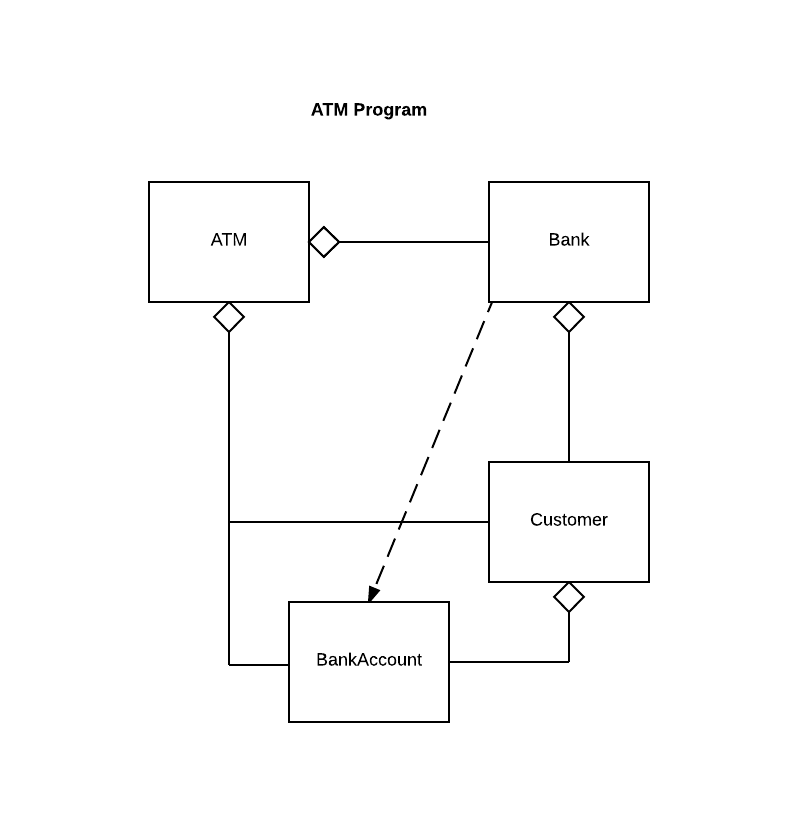


***Function (Deposit):***

When the user chooses the “DEPOSIT” option, it will allow user to enter the amount they want to deposit from selected account.



**UML-Diagram**

****

* **The ATM class uses the BankAccount, Bank, and Customer classes.**
* **The Bank class uses the Customer class.**
* **The Customer class uses the BankAccount Class.**

**Object Oriented Sample**

I am using an ArrayList named “**customers**”to store all Customer objects which reads from a text file named “**customers.txt**” once the program is launched. When the user enters their customer # and password, the input read from the file will verify if the customer’s account exists and if the password matches.



**LIMITATIONS**

This program does not have a GUI and is only ran on a console. Only registered customers can execute this program. All the details of the registered users have been inserted into the text file named “customers.txt”. For further implementation of this ATM program, developers will need to connect with a database. I did not add an option for deleting customers. If admin wants to delete users, he or she must open the text file named “customers.txt” and delete from there.