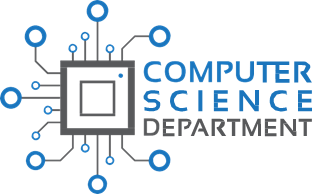
# AMERICAN UNIVERSITY OF SCIENCE AND TECHNOLOGY FACULTY OF ARTS AND SCIENCES



CSI 250 – Computer Programming II

Section A

Spring 2017-2018

BANK

By

Mohamad Cheikho 12130598

Shadi Naoura 12130598

Ibrahim jardani 12130598

Submitted to

Saeed Raheel, Ph.D.

30/5/2018

# **Project Evaluation**

# 

# **Objective**

The objective of this program is to make a user-friendly interface that will help bank employees manage the bank accounts in the most efficient and easiest way possible. The program should keep the employee up to date with all the accounts and the transactions that happen between accounts.

Such programs are found in every bank, this way data will be stored correctly and will never be lost. It is essential that the program has a simple graphical user interface that allows the employee manipulate the accounts easily and without any trouble.

# **Program Functionality**

The proposed program has one user that can access it and that’s the employee at the bank that does all the saving and transactions manually.

In order to save the data correctly, the program should be built in an object oriented approach. So the program is built based on classes that follow:

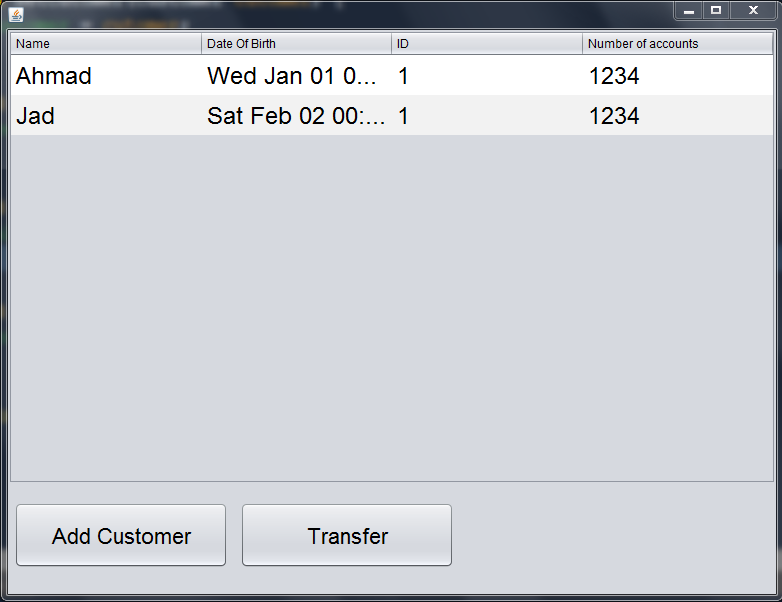
|  |  |
| --- | --- |
| **Class Name** | **Class Description** |
| Customer.java | This class contains the name, date of birth, address, phone number, id, and list of accounts that the customer possess. |
| Account.java | Account is the class responsible for all the types of the accounts and it contains a customer, date of its creation, balance, and ID. Moreover, it contains an abstract function that defines what all its subclasses are. |
| IntrestCheckingAccount.java | It is an account that has interest. |
| CheckingAccount.java | It is an account that is of type checking, |
| SavingsAccount.java | It is an account that allows user to have interest in the money saved. |
| CreditAccount.java | It is an account that is of type credit. |
| Transactions.java | This class contains static functions that allow the program to make transfers between an account and the other, to read the accounts from file, and to print changes into the file. |

The Account.java is the parent class of all the other accounts. It contains an abstract function called getAccountInfo() that is responsible to get the type of the account that is being selected. This abstract function is then implemented in the other subclasses that extend Account. The function basically returns a String that shows the type of the account for future selections.

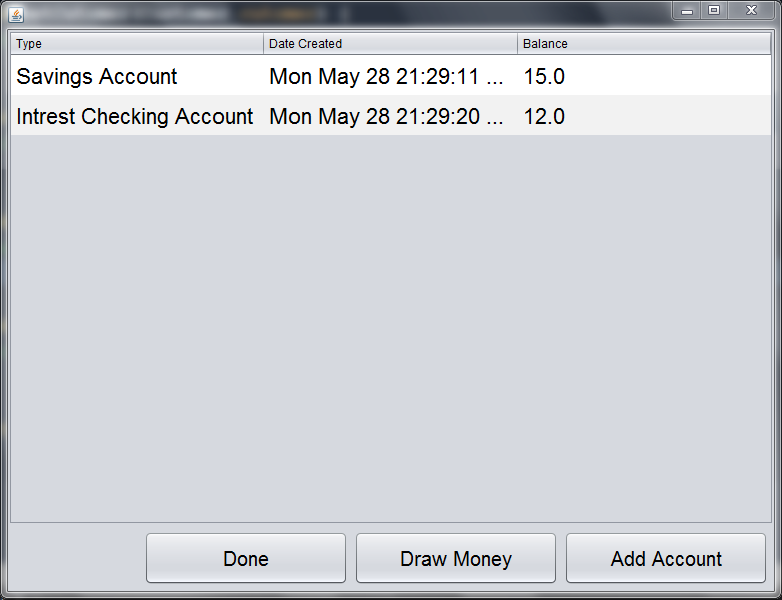
The program uses different frames to manipulate the inputs and outputs. The different frame classes are:

|  |  |
| --- | --- |
| **Class Name** | **Class Description** |
| Customers.java | This class extends JFrame and is the class that launches on startup and has all the customers in a table |
| Accounts.java | This shows all the accounts that are relate to a certain customer. |
| AddAccounts.java | GUI to allow the user to add an account to the system |
| AddCustomer.java | Similar to AddAccounts, allows the user to add a customer to the system. |
| CustomerDetails.java | Shows all the details that are related to the selected customer. |
| Transfer.java | Window that allows the transfer from one account to the other. |

Some different output of the program:



This page shows the different customers that the user currently can manipulate.



This output shows the different accounts for the selected customer that the employee chooses to manipulate.