Solution Overview

These are the classes that I identified:

App: This class is my main class, this is the whole project and of course it is connected to the other three classes I made because this is the compilation of all the classes.

- The way I did it is very simple but it could be made simpler in a smart way that a modern coder would do. I would say I did it how a beginner in java would do it.
- First off, I made a restriction using a while loop so that if the user will input an option that is not given, it will loop to the menu.
- I made a big while loop that stated if the option will be equal to 1 or 2 or 3 or 4 or 5, this way it will still loop through it. I want it like this because at first I could not solve the problem of looping through any options. But with this loop I was able to solve that problem and wherever options the user input is at, it will still go back to if option is equal to 1 and from there going down to the last option and going back again to the first option.
- I put the restriction if the user will input the wrong option at the bottom part of my big while loop so that it can still detect it inside this loop.

Customer: This class has the personal information of the user input.

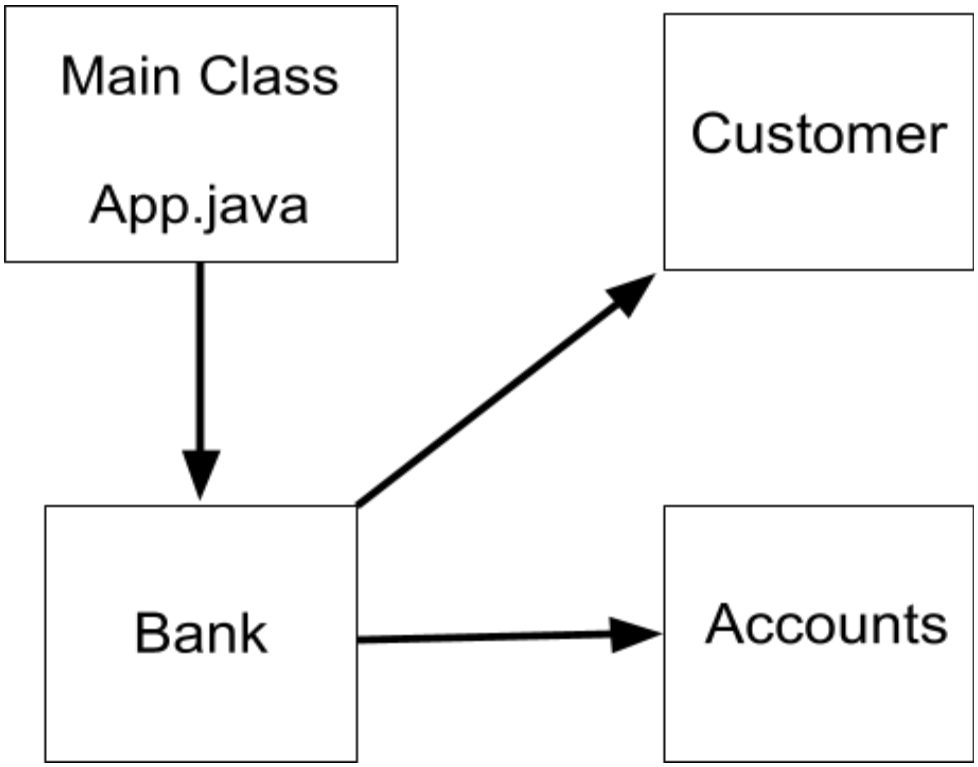
- This class has setters and getters for the first name, last name, and social security number of the input user. All of it is just the personal information for the user input which is connected to all of the classes because the options start at the personal information of the user input.

Account: This class has the code for withdrawing, closing, and depositing the said amount from any of the accounts the user will input.

- At first I made my constructors which I named Account method. This method will check if the account is already closed and generate the account number.
- The deposit method, if the amount is greater than zero and if the account is not closed then the account balance will add the amount of the user input.
- The withdrawal method, if the account balance is equal to zero then print failed withdrawal. Else, if the account balance minus the amount of the user will input is greater than or equal to zero then minus the amount of withdrawal that the user will input to the account balance. I put the if account balance is equal to zero first because I want the withdrawal to fail when the account balance gets to zero so that if ever the account was closed and the deposited amount becomes zero then the withdrawal will fail.
- This class is connected to the class Customer's personal information and to the accounts that the class Bank will list.
- The next following methods are the just setter and getters for each of those variables.

Bank: This class holds the list of the accounts that the user will create.

- First I made an arraylist from the Account class. In this way, the accounts that will be made by the user will be in arrays of lists.
- The bank method, putting the variable accounts to an array list.
- The addAccount method, this method will add any accounts that the user will input to the the Account class and generate a string that says the account number has been created. On the other hand, the account number that will be generated and will come from the Account class.
- The listAccounts method, just listing all the accounts that the user created.
- The depositAccount method, an account will be checked by putting the account number that was generated after the user created an account. If the account exists then the amount that the user will input will go to the method deposit in Account class and be processed with its own condition. If the account was not found then the string will come up saying the account was not created.
- The withdrawAccount method, this method does the same thing as what the depositAccount method does only it will go to the method withdraw in Account class.
- The last method in this class is closeAccount. This method will close an account and then check the account number that the user will input. If the account exists then the account will be closed with a string saying the account is closed and that the current balance or amount that the user will input will be posted and deposits are no longer possible.


Main Class
App.java
Customer
Bank
Accounts