REPORT - CA2 - OOP

By C20372181 Adam O’Shea and C20424992 Ignas Prakapas

In our project there are four main classes, the customer class, the savings account class, checking account class and the account class.

PROJECT CLASSES

The customer class:

We have six variables that we initialise for this class Account ID, Account Type, Account User, Account Balance, Username, Account Details

Then we have the start menu, this is the menu where the user decides if they would like to make a new account or register into an already existing account, based on what the user decides, these lead to taking input from the user. For registering the new details are written into the “customers.txt” file, and for login the input is compared to the details in the same file.

The account class:

We have six variables that we initialise for this class Account ID, Account Type, Account User, Account Balance, Target, Amount, Target Account Balance. Also we have used inheritance from the customer class above.

This is our main menu after the start menu. We have multiple options here, you can check your account, make a new account, deposit money to an account, withdraw money to an account, close an account, transfer from one account to another, view an account’s transactions. First you would select create a new account and you will be redirected to choose between a savings or a checking account. You can check your account details by choosing to check your account, this shows all the account holders’ details of that account. We have deposit and withdraw; they deposit and withdraw from a selected account respectively. You can close an account; this effectively deletes the details of it in the “accounts.txt” file. You can transfer money from one account to another, if you know the unique IDs of the other account you are sending it to, whether that be your own account or a foreign account. Lastly, you can view the transactions of the accounts the username of the current active user is associated with.

The savings account class:

We only initialise username as a variable here. This class inherits the account class. This class basically writes all the input, and edits the input based on the users’ actions into the “accounts.txt” file. We have also designed a visual representation of transactions.

The checking accounts class:

We only initialise username here as well. This also inherits the account class. This is near enough the same to the savings account but is made for an account that handles more transfers than the savings account would.

USER MANUAL:

On start-up this is the first thing the user will see is the start-up menu. They are prompted to register or login.

If they hit register, then they must fill out a registration form.

If they hit login then they need to enter in their username and password that they have used before on the system.

If they press (x) to exit, then the program will terminate.

Once they are logged in, they are then greeted by the main menu, they have multiples choices to choose from, but most options are restricted until they have an account on their user account. As a new user, you do this by going into creating a new account and creating either a savings or checking account. **It is very important to remember your account ID.**

If they hit check account, then they are shown the details of their account.

If they hit deposit money, then they will need to enter in an account ID, and then they can specify the amount of money they would like to deposit.

If they hit withdraw money, they must enter in an account ID that is related to their user account. If the balance of the account is zero or below it, you cannot withdraw money from that account.

If they choose to close an account, then a confirmation will pop up if this is really what they want, if the user continues then the account is wiped.

If they choose to transfer money from one account to another, then they need the account ID of the account they want to send from and the account ID they are sending it to. If the sender has zero or is below it then they cannot send money.

If they select view transactions, then they enter their account ID to see all their recent transactions, you cannot enter in an account ID that is not related to your account.

If they select (x) to exit, then they will be brought back to the start menu.

TEAM MEMBERS CONTRIBUTIONS:

Ignas:

I created the main menu. I did the report. I helped make the deposit and withdraw functions. I made the output in the terminal more readable. I did the savings account and half of the methods in the account class including deposit, viewTransactions, etc. I also commented the code.

Adam:

I created the customer class and its respective methods. I coded the other half of the account class including withdraw, transfer, etc. I also created the checkingAccount class and its respective methods.

CREATIVITY:

For the additional program function outside of the brief, we created a reset password method in the account class. This allows a user to change their login password to something else. This is accomplished by prompting for the existing password, asking for a new password, and confirming the new password. The customer’s file is then read in to a list and the list is searched to find the old password. The old password is changed to the new password and the file is overwritten with the change.

CHALLENGES:

The biggest challenge we faced was when it came to editing our files. We found the solution for it to be reading the files to a list variable, and then edit them within the program by searching the list and changing elements. Then we would overwrite the file with the new changes.

We also used another method of writing the changes to a temporary file and then renaming the temp file to the actual files name to overwrite it.