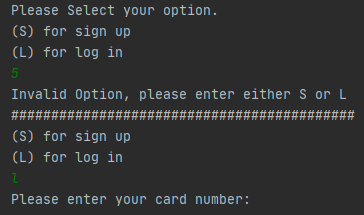
Our program uses three text files, “cardNumber” which stores card number, card pin , email and balance. And the “transactions” file which stores the transaction history of a user. And the “last transaction” file which stores the last transaction made by the user in the session.

We use 12 different functions.

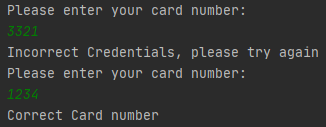
**#1: main()**

This function initializes the program, it makes chose whether you want to login or sign up



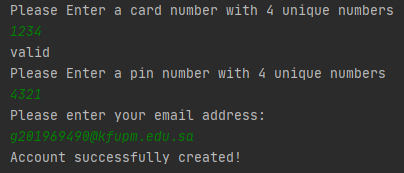
**#2: notSame(number):**

This function takes the input given to it and validates whether the entered string was an integer with 4 different numbers. And returns a Boolean statement



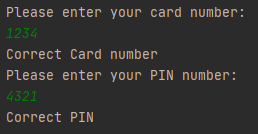
**#3 create():**

This function asks the user to enter a valid card number and pin, which have non- repeating characters using the notSame() function, it also asks for the email of the user and stores all that data in the “cardNumber” text file. It also makes the user start with 0 balance. Then moves the user to the menu



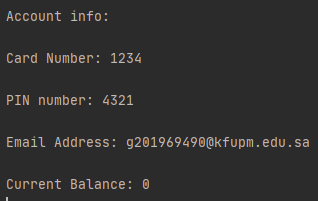
**#4 login():**

This function asks the user to enter his credentials and makes sure they are identical to the ones in the “cardNumber” text file. If they are correct it will take you to the menu



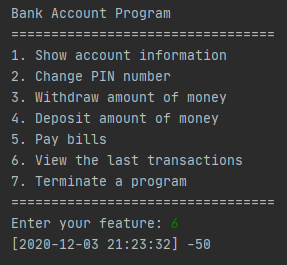
**#5 show(file):**

This function reads the lines on the “cardNumber” text files and displays them to the user.



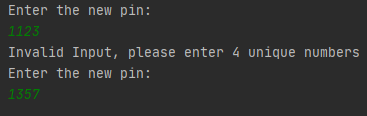
**#6 menu():**

This function asks the user to enter one of 7 functions if the input is invalid it will ask for the input again.



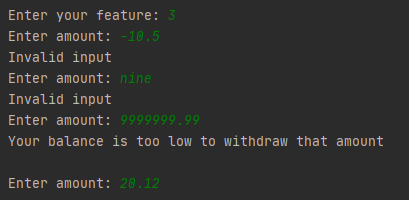
**#7 changePINFun(currentPIN, cardNumber, file):**

This function reads the previous values of the “cardNumber” file and asks the user to input a new pin which will replace the old one in the text file, after the input is taken it replaces al the other values to where they were originally but re-writes the pin.



**#8 withdrawFun(money, cardNumber, file):**

Gets all the lines from “cardNumber” file and assigns each line to its own variable and then takes the user input and checks if it’s valid then subtracts it to the current balance. After that, it will open the “lastTransaction” file and write the balance into the file. Finally, it will append it to the “transactions” file.



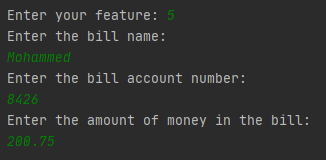
**#9 depositFun(nMoney, cardNumber, file):**

It does everything like withdrawFun() except it takes the user input and adds it to the current balance.



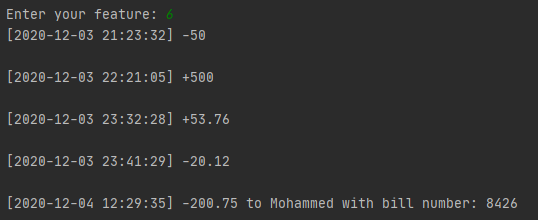
**#10 payBillFun(file, nMoney, cardNumber):**

Receives the bill name, number and amount. Then processes it and writes the transaction info to the “transactions” file.



**#11 viewTransactionsFun(cardNumber):**

Reads the data that is inside the “transactions” file and prints it out in the console.



**#12 terminateFun(file, nMoney, cardNumber):**

Checks whether if there were any transactions made during a session and then prints it out in the console, then it will terminate the program.

