

Bank Account Repository

Develop a menu driven complete program that can perform the illustrated features in Figure 1. This program works for an individual customer only; the program will save the bank account information into a text file.



Figure 1: account bank menu

Description:

The program starts by asking the user to select either login (L) or sign up (S) feature. Once the user selects a feature, it asks him for (**Card Number**, then **PIN**) and after filling the required information for that feature, it shows the above menu for other features. Otherwise, it keeps asking for the required information. Each feature has to be designed using a separate function and it might be called by other functions including the main function. The description of the login feature, sign up feature, and the above features are given below:

1. **Sign up Feature – `def create()`:** this feature allows new user to create an account and it saves his information in **cardNumber.txt**. The required information to be asked in this feature as follows:
 - (a) **Card Number:** user will enter a card number and should be four digits, each digit is unique and it should not be repeated in the 4-digits card number. This feature will keep running until the user enters the required number with same length and type.
 - (b) **PIN:** user will enter a PIN number and should be four digits, each digit is unique and it should not be repeated in the 4-digits PIN number. This feature will keep running until the user enters the required number with same length and type.
 - (c) **Email:** user needs to enter his KFUPM email (g20XXXXXXX@kfupm.edu.sa) in the registration stage. If he did something else, it will ask again for email with same specifications.
 - (d) **Extra services:** user needs to add at least one service for the new account.

After creating your account, you are required to save the information in **cardNumber.txt** file in same path of your **jupyter** file and each item will be saved in a line. Then, the program will sleep for few seconds and it will redirect you to the login feature.

2. **Login Feature – `def login()`:** it allows the user to login to his saved account after creating the account using the first function (`create()`). The required information to be asked in this feature as follows:
 - (a) It asks the user to enter his card number and it should be available in same path (**cardNumber.txt**). The program will check the first line in the **cardNumber.txt** for the card number. This feature will keep running until the user enters his saved number.
 - (b) Then, the program will check the second line in the **cardNumber.txt** for the PIN number. This feature will keep running until the user enters his saved number.
 - (c) If the user enters the correct info, the menu in Figure 1 will appear directly.
3. **Show Feature – `def show(file)`:** it allows the user to show his saved account details based on the reading of his .txt file. After doing this step, the program will sleep for few seconds and it will redirect you to the menu again.

4. **ChangePIN Feature** – `def changePINFun(currentPIN, cardNumber, file)`: it allows the user to change his PIN number in the .txt file. Here, you are required to replace the saved PIN by new PIN and it should have the same structure. After doing this step, the program will sleep for few seconds and it will redirect you to the menu again.
5. **Withdraw Feature** – `def withdrawFun(money, cardNumber, file)`: it allows the user to withdraw money from his account and then updates the account details in the .txt file accordingly along with the date and the time of this operation. Remind that, if you don't have enough money in your account, it will give a notification message and it will ask for new amount until you entered an amount that you can withdraw it. After doing this step, the program will sleep for few seconds and it will redirect you to the menu again.
6. **Deposit Feature** – `def depositFun(file, nMoney, cardNumber)`: it allows the user to deposit money to his account and then updates the account details in the .txt file accordingly along with the date and the time of this operation. After doing this step, the program will sleep for few seconds and it will redirect you to the menu again.
7. **Bills Feature** – `def payBillFun(file, nMoney, cardNumber)`: it allows the user to pay a bill from his account, deducts this bill from the account and updates the account details in the .txt file accordingly along with the date and the time of this operation. Following some details of the bill that might use in the function:
 - (a) Enter the name of the bill.
 - (b) Enter the account number of this bill.
 - (c) Enter the value of this bill.

Remind that, if you don't have enough money in your account, it will give a notification message and it will ask you to deposit amount to deduct the bill. After doing this step, the program will sleep for few seconds and it will redirect you to the menu again.
8. **View Transactions Feature** – `def viewTransactionsFun(cardNumber)`: it allows the user to show the history of his account based on the saved .txt file. If there are transactions in the account, it will be presented along with its (date and time). Otherwise, it will show "no transactions". After doing this step, the program will sleep for few seconds and it will redirect you to the menu again.
9. **Terminate Feature** – `def terminateFun(file, nMoney, cardNumber)`: it allows the user to terminate the program and at same time it will show the last transactions on the account during the session.