Creating a simple Bank Management System using Python

Here I am trying build app that would try to manage all the bank details of users. This would inform about creating an account, and managing all the details of the account and his savings and deposits.

So here we need a class so that we can store and access all the details that are required for an user who uses this.

1. **Importing the libraries**

```
import pickle
import os
import pathlib
```

2. Creating a class account

```
class Account :
   accNo = 0
   name = ''
   deposit=0
   type = ''
```

3.Function to create an account

```
def createAccount(self):
      self.accNo= int(input("Enter the account no : "))
      self.name = input("Enter the account holder name : ")
      self.type = input("Ente the type of account [C/S] : ")
      self.deposit = int(input("Enter The Initial amount(>=500 for Saving and >=1000 )for current"))
      print("\n\n\nAccount Created")
```

4. Function to show the account

```
def showAccount(self):
      print("Account Number : ",self.accNo)
      print("Account Holder Name : ", self.name)
      print("Type of Account",self.type)
      print("Balance : ",self.deposit)
```

5. Function to change the account holder name

```
def modifyAccount(self):
      print("Account Number : ",self.accNo)
      self.name = input("Modify Account Holder Name :")
      self.type = input("Modify type of Account :")
      self.deposit = int(input("Modify Balance :"))
```

6. Function to show the deposited or withdrawn amounts

```
def depositAmount(self,amount):
      self.deposit += amount

   def withdrawAmount(self,amount):
      self.deposit -= amount
```

7. Function to show sessional report with all details.

```
def report(self):
      print(self.accNo, " ",self.name ," ",self.type," ", self.deposit)
```

8. Functions to get account number, account holder name and other details

```
def getAccountNo(self):
      return self.accNo

   def getAcccountHolderName(self):
      return self.name

   def getAccountType(self):
      return self.type

   def getDeposit(self):
      return self.deposit
```

9. Function to create an account

```
def writeAccount():
   account = Account()
   account.createAccount()
   writeAccountsFile(account)
```

10. Function to display all the details if the user data exists

```
def displayAll():
   file = pathlib.Path("accounts.data")
   if file.exists ():
      infile = open('accounts.data','rb')
      mylist = pickle.load(infile)
      for item in mylist :
         print(item.accNo," ", item.name, " ",item.type, " ",item.deposit )
      infile.close()
   else :
      print("No records to display")
```

11.Function to display the account_balance

```
def displaySp(num):
   file = pathlib.Path("accounts.data")
   if file.exists ():
      infile = open('accounts.data','rb')
      mylist = pickle.load(infile)
      infile.close()
      found = False
      for item in mylist :
         if item.accNo == num :
            print("Your account Balance is = ",item.deposit)
            found = True
   else :
      print("No records to Search")
   if not found :
      print("No existing record with this number")
```

12. Function that carries out the deposits and withdrawals

```
def depositAndWithdraw(num1,num2):
```

```
    file = pathlib.Path("accounts.data")
    if file.exists ():
        infile = open('accounts.data','rb')
        mylist = pickle.load(infile)
        infile.close()
        os.remove('accounts.data')
        for item in mylist :
            if item.accNo == num1 :
                if num2 == 1 :
                    amount = int(input("Enter the amount to deposit : "))
                    item.deposit += amount
                    print("Your account is updted")
                elif num2 == 2 :
                    amount = int(input("Enter the amount to withdraw : "))
                    if amount <= item.deposit :
                        item.deposit -=amount
                    else :
                        print("You cannot withdraw larger amount")

else :
        print("No records to Search")
    outfile = open('newaccounts.data','wb')
    pickle.dump(mylist, outfile)
    outfile.close()
    os.rename('newaccounts.data', 'accounts.data')
```

13. Function to delete the account

```
def deleteAccount(num):
    file = pathlib.Path("accounts.data")
    if file.exists ():
        infile = open('accounts.data','rb')
        oldlist = pickle.load(infile)
        infile.close()
        newlist = []
        for item in oldlist :
            if item.accNo != num :
                newlist.append(item)
        os.remove('accounts.data')
        outfile = open('newaccounts.data','wb')
        pickle.dump(newlist, outfile)
        outfile.close()
        os.rename('newaccounts.data', 'accounts.data')
```

14.Function to modify the accounnt details

```
def modifyAccount(num):
    file = pathlib.Path("accounts.data")
    if file.exists ():
        infile = open('accounts.data','rb')
        oldlist = pickle.load(infile)
        infile.close()
        os.remove('accounts.data')
        for item in oldlist :
            if item.accNo == num :
                item.name = input("Enter the account holder name : ")
                item.type = input("Enter the account Type : ")
                item.deposit = int(input("Enter the Amount : "))

outfile = open('newaccounts.data','wb')
        pickle.dump(oldlist, outfile)
        outfile.close()
        os.rename('newaccounts.data', 'accounts.data')
```

15.Function that writes the account details into file

```
def writeAccountsFile(account) :

file = pathlib.Path("accounts.data")
   if file.exists ():
      infile = open('accounts.data','rb')
      oldlist = pickle.load(infile)
      oldlist.append(account)
      infile.close()
      os.remove('accounts.data')
   else :
      oldlist = [account]
   outfile = open('newaccounts.data','wb')
   pickle.dump(oldlist, outfile)
   outfile.close()
   os.rename('newaccounts.data', 'accounts.data')
```

16. Asking the user for the task that he needs to do

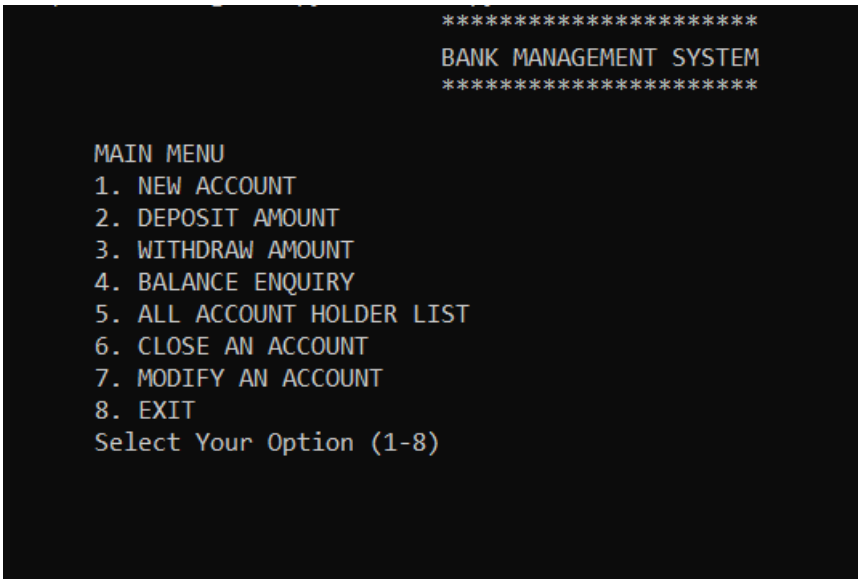
```
ch=''
num=0
intro()

while ch != 8:
   #system("cls");
   print("\tMAIN MENU")
   print("\t1. NEW ACCOUNT")
   print("\t2. DEPOSIT AMOUNT")
   print("\t3. WITHDRAW AMOUNT")
   print("\t4. BALANCE ENQUIRY")
   print("\t5. ALL ACCOUNT HOLDER LIST")
   print("\t6. CLOSE AN ACCOUNT")
   print("\t7. MODIFY AN ACCOUNT")
   print("\t8. EXIT")
   print("\tSelect Your Option (1-8) ")
   ch = input()
   #system("cls");

if ch == '1':
      writeAccount()
   elif ch =='2':
      num = int(input("\tEnter The account No. : "))
      depositAndWithdraw(num, 1)
   elif ch == '3':
      num = int(input("\tEnter The account No. : "))
      depositAndWithdraw(num, 2)
   elif ch == '4':
      num = int(input("\tEnter The account No. : "))
      displaySp(num)
   elif ch == '5':
      displayAll();
   elif ch == '6':
      num =int(input("\tEnter The account No. : "))
      deleteAccount(num)
   elif ch == '7':
      num = int(input("\tEnter The account No. : "))
      modifyAccount(num)
   elif ch == '8':
      print("\tThanks for using bank managemnt system")
      break
   else :
      print("Invalid choice")

ch = input("Enter your choice : ")
```

Now we have observed the coding structure so now we will see how the output appears :



Here we need to check the option which is necessary and select it inorder to proceed.

Thus we can save the details of the account user and his transactions using this simple project.

References:

link to github:

gullayeshwantkumarruler/Bank-Management-System
*Contribute to gullayeshwantkumarruler/Bank-Management-System development by creating an account on GitHub.*github.com

link to linkedin:

Yeshwant Kumar - Gandhi Institute of Engineering and Technology (GIET), Gunupur - Orissa, India |...
*I am a workaholic, always trying to tend my Hard-Work towards Smart-Work.I always apply my logical sense rather than...*www.linkedin.com

By Yeshwant Kumar on May 9, 2021.

Canonical link

Exported from Medium on May 9, 2021.