FILE CLASS:

Here we stop 5 classes and we will list them below:
Class for menu : The relation it will have between this class is that it will direct the user to the different functions of the other classes.

Class for account opening
From the main menu if the user wants to create a new account this class with its functions will need to enter in actions, he will need his name, password, and account number.

Class for transfer
From the main menu, if the user wants to transfer money to another user, he will use the account number and user name of the person to whom he wants to transfer the money that has been saved by a class creation function, but before doing so, he will need to identify himself with the data that has been saved during creation.

Class for deposit
In the main menu, if the user wishes to make a deposit he will need his username and account number which is saved by a Class creation function.

Class for consultation
In the main menu, if the user wishes to make a consultation The user can use his username and password which has been saved by a function of the Class creation.

Class Menu:

```
def __init__(self    ):
  Def question():
```

In this method, we are going to ask the user which action he wants to perform (creation, transfer, deposit, consult)
If the user wants to create the account choose one: the program use Class Account (enter name, password, account number)

If the user wants to make the transfer choose two: the program use Class transfer (enter name and password)

If the user wants to make a deposit choose three: the program use Class deposit (enter name ,and enter account number)

If the user wants to consult his money four: we use Class deposit (enter name and enter password)

Test case number	Test case description	Test data	Expected result	Actual results	Pass/Fail
1	**verifies if the customer's choice of creating an account is given.**	**one**	**the choice must be successful**	**successful choice**	**pass**
2	**check if the customer wants to make a transfer**	**Two**	**the choice must be successful**	**successful choice**	**pass**
3	**check if the customer wants to make a deposit**	**Three**	**the choice must be successful**	**successful choice**	**pass**
4	**check if the customer wants to make a transfer**	**Four**	**the choice must be successful**	**successful choice**	**pass**
5	**Check if the print work**	**print**	**the choice must be successful**	**successful choice**	**pass**

Class Creation :

```
    def __init__(self,name,password,account_number)
    def save( name,password,account_number):
      name=input('please enter name:)
      password=input(please enter the password;)
      account_number=int(input("please enter number:")
 print("welcome",name)
print(account created)
```

Test case number	Test case description	Test data	Expected result	Actual results	Pass/Fail
1	test if the username saves	Name (in the file)	hopes to save	successful backup	pass
2	test if the password is saved	password (in the file)	hopes to save	successful backup	pass
3	test if the account number is saved	Account number (in the file)	hopes to save	successful backup	Pass
4	if the amount of the account creation is saved	amount (in the file)	hopes to save	successful backup	Pass
5	file can save a lot of data and we can use data	The file	hopes to save	successful backup	pass

Class Transfer:

def __init__(self,name,password ,account_number)
def ID (name,password)
If your name user and password it is your identification
print("welcome",name)

Enter the name of the person you want to make the transfer

def trans(name ,account_number)
print(how much do you want to send)
Account1-amount that user send
Amount that user send + account2 (receptionist)

Test case number	Test case description	Test data	Expected result	Actual results	Pass/Fail
1	**check if the user name gives**	**User name**	**hope to get the user data**	**Successful to get**	**pass**
2	**check if the password gives**	**password**	**hope to get the password**	**Successful to ge**	**pass**
3	**check if the account gives**	**Account number**	**hope to get the bank account**	**Successful to ge**	**pass**
4	**checks if the second user name gives**	**username2**	**hope to get the ,username 2**	**Successful to ge**	**pass**
5	**Checks if the transfers gives**	**amount**	**Hope to make the transfer**	**Successful for the transfer**	**pass**

class deposit :
def init (self,name,account number)

Def identification(name,account number)
Enter name
If name==name_user
Enter account number :
If account number == account_user
Enter amount :

Def calculated(account,number)
Account number account +=amount

Test case number	Test case description	Test data	Expected result	Actual results	Pass/Fail
1	**check if the user name gives**	**User name**	**hope to get the user data**	**Successful to get**	**pass**
2	**check if the password gives**	**password**	**hope to get the password**	**Successful to ge**	**pass**
3	**check if the account gives**	**Account number**	**hope to get the bank account**	**Successful to ge**	**pass**
4	**Give the news balance**	**Account number**	**Hope to give the new balance**	**Successful to ge**	**pass**
5	**Checks if the deposit works**	**amount**	**Hope to make the transfer**	**Successful for the transfer**	**pass**

Class consult:

Def init(self,name,password,account number)

Def know(name,password)

Enter name

Enter password ‘

If name == name user

password=input(‘enter password:’)

Print(welcome,name

Test case number	Test case description	Test data	Expected result	Actual results	Pass/Fail
1	**check if the user name gives**	**User name**	**hope to get the user data**	**Successful to get**	**pass**
2	**check if the password gives**	**password**	**hope to get the password**	**Successful to ge**	**pass**
3	**check if the account gives after enter name and password**	**Account number**	**hope to get the bank account**	**Successful to ge**	**pass**
4	**Give the news balance if we make the deposit or transfer**	**Account number**	**Hope to give the new balance or rising anchors**	**Successful to ge**	**pass**
5	**If all the condition work**	**Condition**	**Hope that the condition work**	**Successful to ge**	**pass**