Didactices IT Solutions LLP, Smart Work Business Centre Pvt. Ltd. Victoria Park Plot No. GN 37/2
Sector V 1st Floor, Saltlake Kolkata-91

Bank Account Management System

Done By : :

Soumyajit Pal, Techno India Batanagar (MAKAUT)
University Roll. : 33200115043
Reg. No : 153320110043 session : 2015-19
Email : soumyajit2pal@gmail.com

Table of contents

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty Prof. Arnab Chakraborty for his guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

Soumyajit Pal

Project Objective

The objective of this project is to maintain the accounts like saving account, withdrawing, deposit money. Bank provides account number to the customer for dealing transactions in the bank. At first a user registers himself/herself as customer. After that account can be opened. A customer can open more than one account.

Users of this project are

- Accountant
- Customer

Roles of Accountant are

- Login using his/her username and password
- Adding new account for customer
- Removing the account by using account number
- Viewing particular account details by giving account number
- Viewing all the account details
- Taking care of deposit and withdrawal operations

Roles of Customer are

- Login using his/her username and password
- Transfer the money from his account to other account
- Checking the transaction history

Requirement Specification

Processor	: Intel Pentium III or later
Main Memory (RAM)	: 256MB
Cache Memory	: 512KB
Monitor	: 14 inch Color Monitor
Keyboard	: 108 Keys
Mouse	: Optical Mouse
Hard Disk	: 160 GB

Software Requirement

- Python Interpreter
- Python package :
    - Pandas
    - Time
    - pandas

Database Design

Accountant Table

Field	Description
username	This is required to login as a accountant
password	

Customer Table

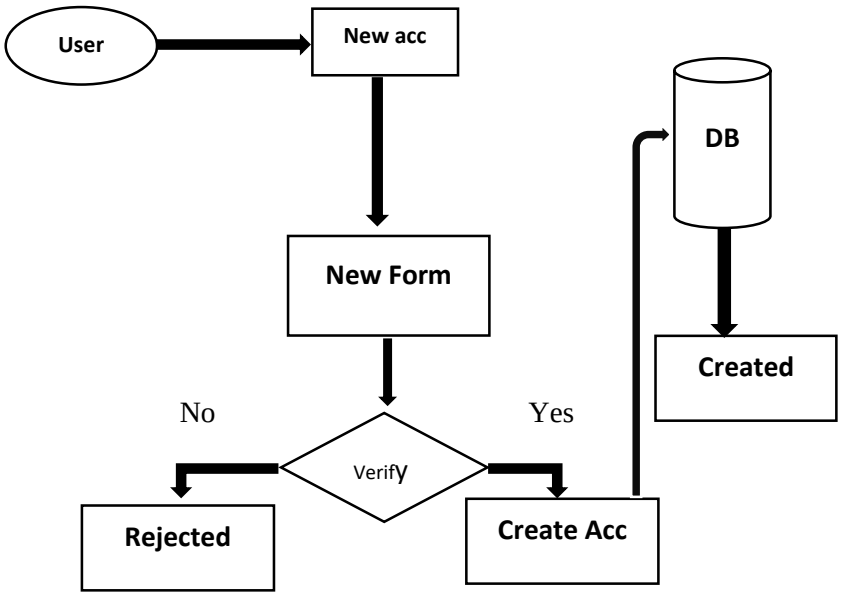
Field	Description
customerid	Create new id when account is created
acc_num	Create account number when account is created
name	Account holder's name
age	Account holder's age
sex	Account holder's sex
Address	Account holder's address
Phone	Account holder's phone number
Income	Account holder's income details
status	When account is active shows True otherwise False
date	Account creation date and time
amount	Amount of money in an account
userid	userid when account is created
passcode	passcode when account is created, it can be changed

Transaction Table

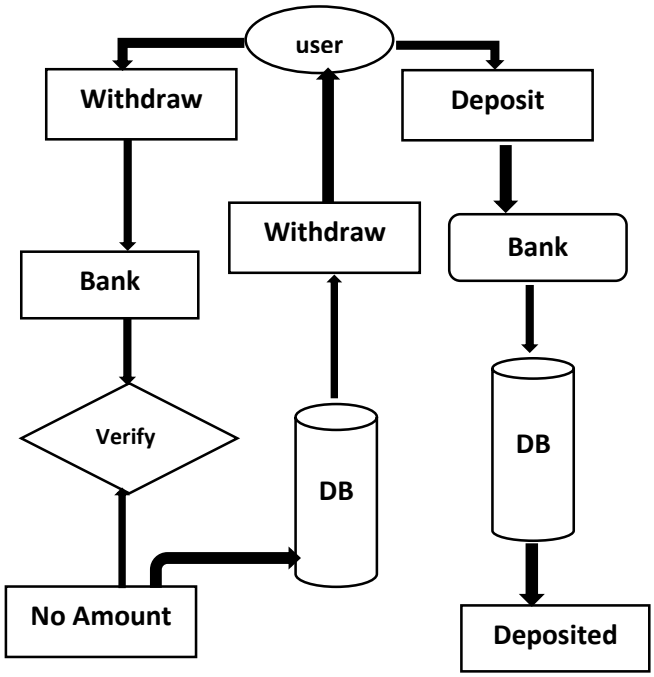
Field	Description
tid	Transaction id on every transaction
debit	Type of transaction(withdrawl)
credit	Type of transaction(deposit)
balance	After transaction amount of money in an account
date	Date of transaction
account	Account number of transaction
status	Type of transaction IMPS or self

Application Work flow

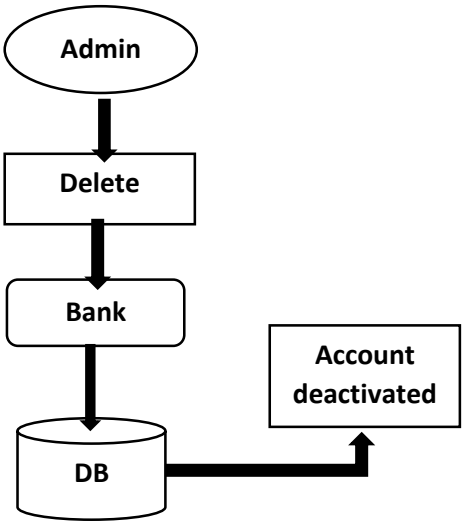
Create new account Data Flow Diagram



Data Flow Diagram for withdraw or deposit account



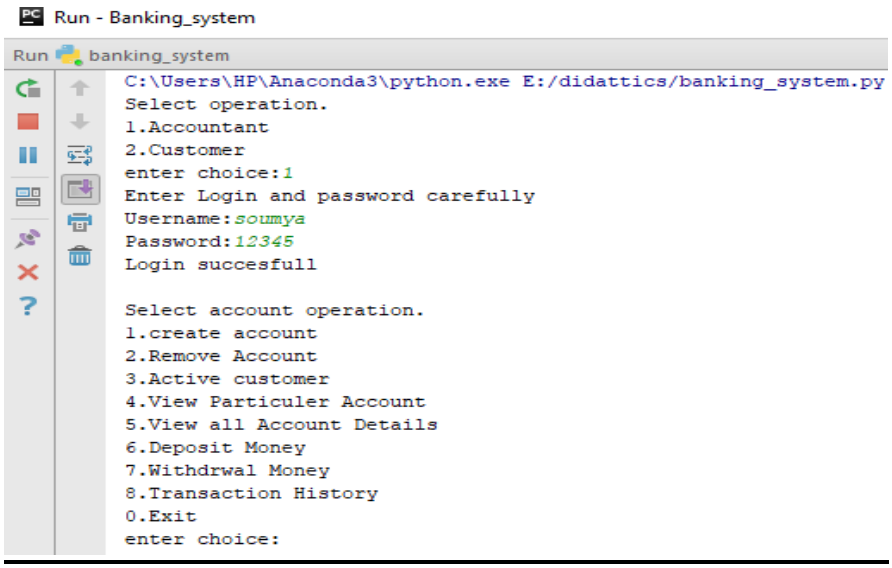
Data Flow Diagram for deleting an account



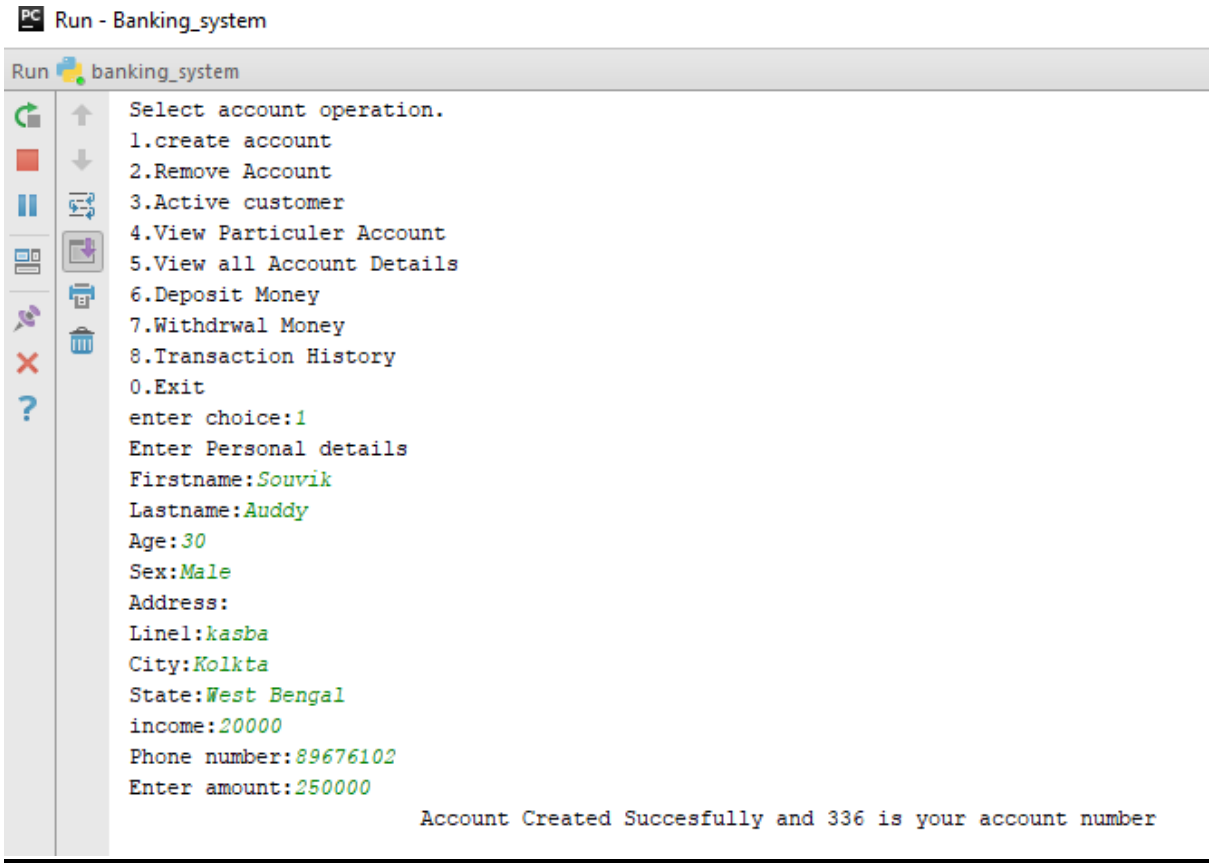
Screenshots

Accountant Menu

Login :

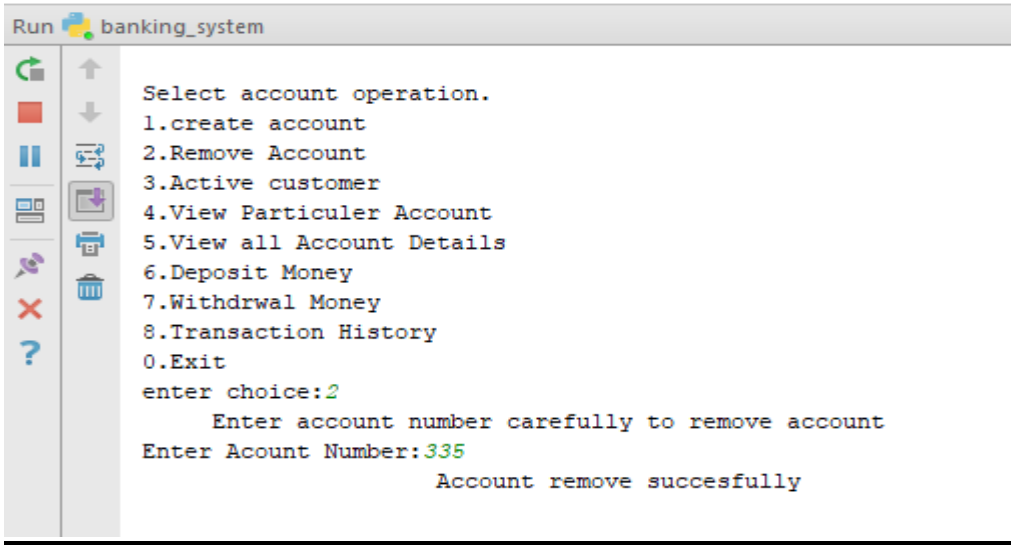


Create new account:



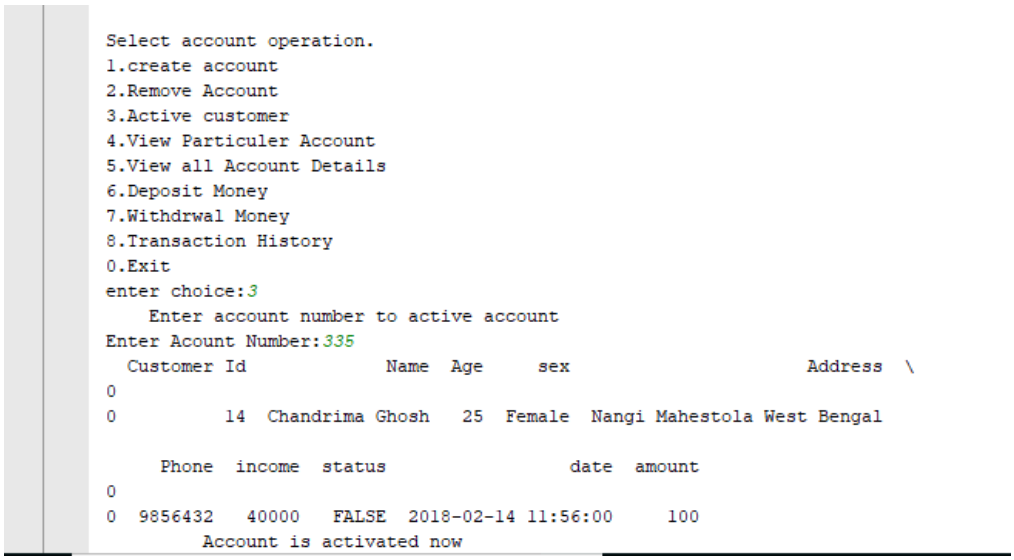
Remove Account

Run - Banking_system



```
Run  banking_system

Select account operation.
1.create account
2.Remove Account
3.Active customer
4.View Particuler Account
5.View all Account Details
6.Deposit Money
7.Withdrwal Money
8.Transaction History
0.Exit
enter choice:2
     Enter account number carefully to remove account
Enter Acount Number:335
                        Account remove succesfully
```

Active Customer :



```
Select account operation.
1.create account
2.Remove Account
3.Active customer
4.View Particuler Account
5.View all Account Details
6.Deposit Money
7.Withdrwal Money
8.Transaction History
0.Exit
enter choice:3
    Enter account number to active account
Enter Acount Number:335
  Customer Id             Name  Age     sex                       Address  \
0
0          14  Chandrima Ghosh   25  Female  Nangi Mahestola West Bengal

     Phone  income  status                date  amount
0
0  9856432   40000   FALSE  2018-02-14 11:56:00     100
         Account is activated now
```

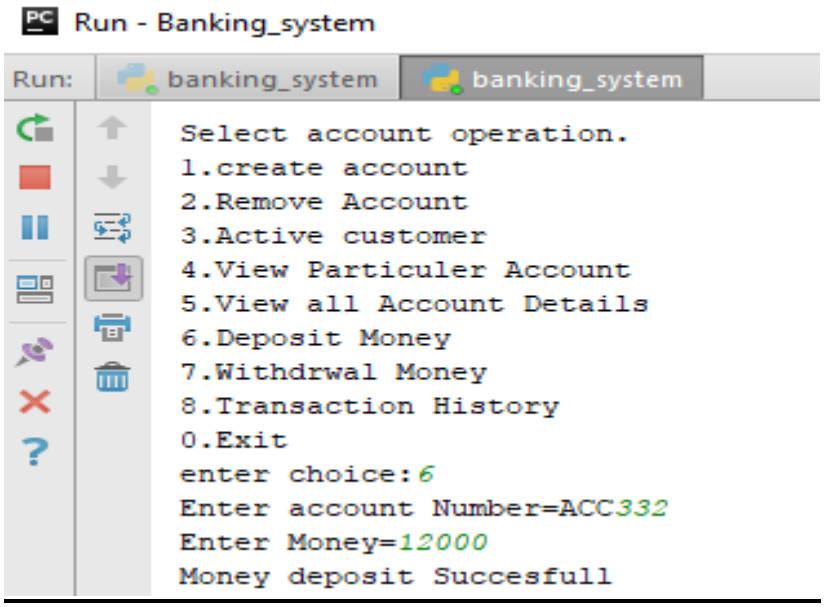
View particular Account:

```
Select account operation.
1.create account
2.Remove Account
3.Active customer
4.View Particuler Account
5.View all Account Details
6.Deposit Money
7.Withdrwal Money
8.Transaction History
0.Exit
enter choice:4
     Enter acoount number to view account details
Enter Acount Number:333
  Customer Id          Name  Age   sex                     Address  \
0
0          12  Souvik Auddy   21  Male  Entally Kolkata West Bengal

       Phone  income  status                date  amount
0
0  896761027   20000    TRUE  2018-02-13 12:42:25   24100
```

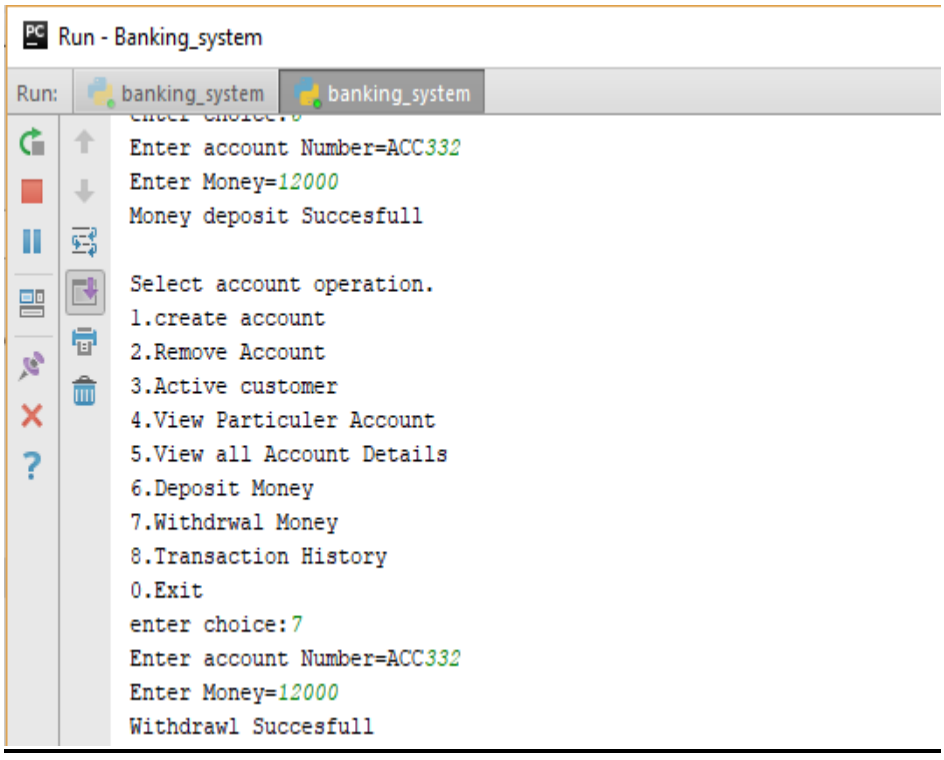
View all account details:

```
Run - Banking_system
Run banking_system
4.View Particuler Account
5.View all Account Details
6.Deposit Money
7.Withdrwal Money
8.Transaction History
0.Exit
enter choice:5
     You choose to view all acount details

  Customer Id Account Number             Name Age     sex  \
0
0           8            332    Soumyajit Pal  21    Male
0          12            333     Souvik Auddy  21    Male
0          14            334  Chandrima Ghosh  25  Female
0          15            335     Souvik Auddy  30    Male

                       Address      Phone Income status                 date  \
0
0                        Kasba  943215525  20000   TRUE  2018-02-12 15:35:54
0  Entally Kolkata West Bengal  896761027  20000   TRUE  2018-02-13 12:42:25
0  Nangi Mahestola West Bengal   98564321  40000   TRUE  2018-02-14 11:56:00
0     kasba Kolkta West Bengal   89676102  20000   TRUE  2018-02-14 12:27:48

   amount
0
0   38000
0   18977
0   28900
0  250000
```

Deposit Money:



Withdrawl Money:

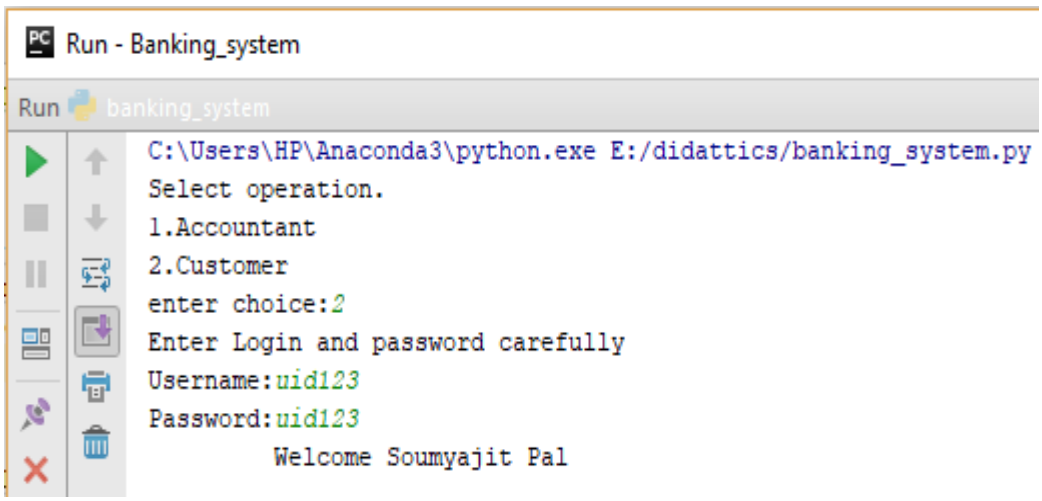


Transaction History:

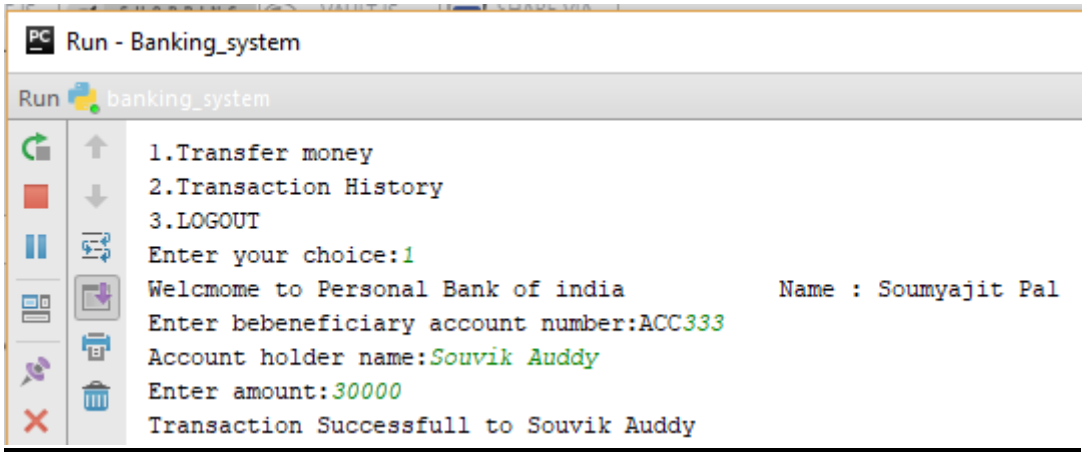
```
Select account operation.
1.create account
2.Remove Account
3.Active customer
4.View Particuler Account
5.View all Account Details
6.Deposit Money
7.Withdrwal Money
8.Transaction History
0.Exit
enter choice:8
Enter account number:ACC335
  Transaction id  Debit  Credit  Balance                 Date  Type
0
0              58      0   21000    20900  2018-02-14 12:41:41  SELF
0              55      0    1000     1100  2018-02-14 12:31:41  SELF
0              54  12000       0      100  2018-02-14 12:03:48  IMPS
0              51    100       0    12000  2018-02-14 11:58:37  SELF
0              50      0     100    12100  2018-02-14 11:58:14  SELF
0              52      0     100    12100  2018-02-14 12:02:08  SELF
```

Customer Menu

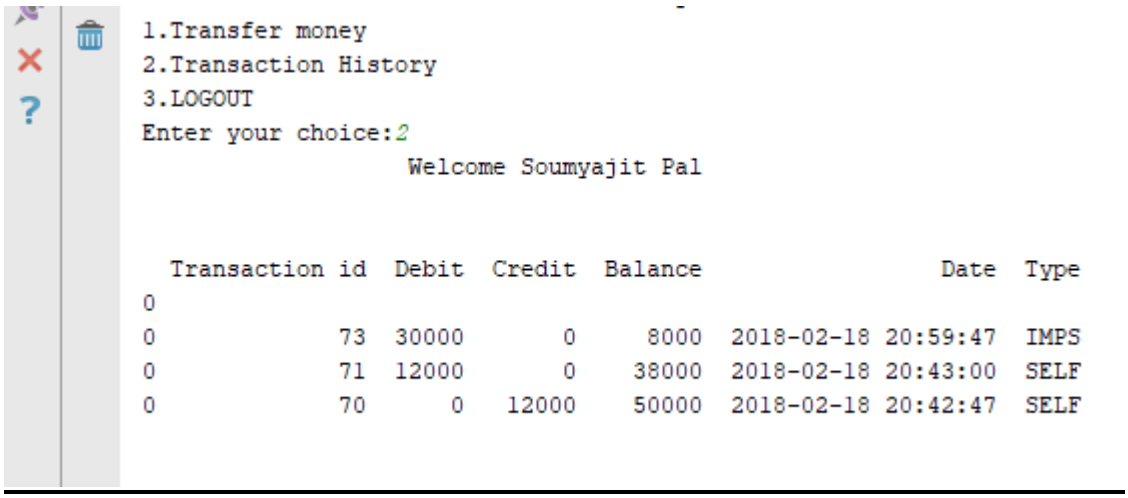
User Login:



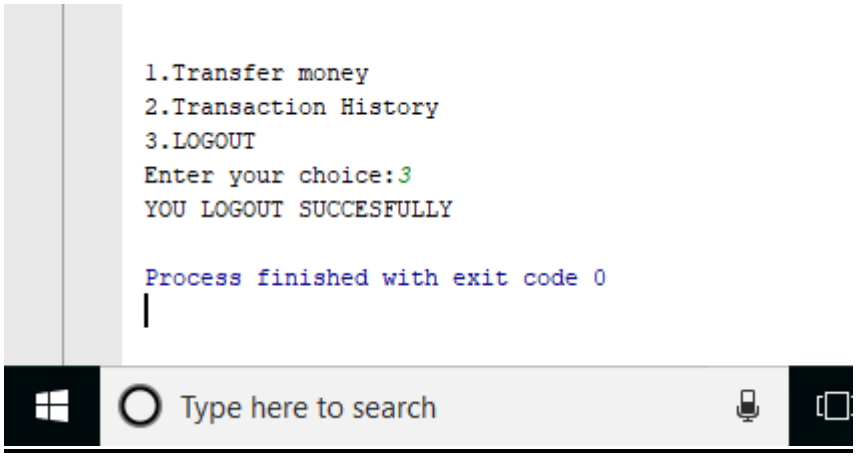
Transfer Money to another account:



Transaction History:



Logout :



Code

```
import pymysql
import time
import pandas as pd
import numpy as np

######################### Choice Menu##############

def customer_menu():
    print("Select account operation.")
    print("1.create account")
    print("2.Remove Account")
    print("3.Active customer")
    print("4.View Particuler Account")
    print("5.View all Account Details")
    print("6.Deposit Money")
    print("7.Withdrwal Money")
    print("8.Transaction History")
    print("0.Exit")

##############################   Choice Pickup ###################################

    choice = int(input("enter choice:"))
    if (choice == 1):
        print("Enter Personal details")
        db = pymysql.connect("localhost", "root", "", "banking")
        cursor = db.cursor()
        try:
            cursor.execute("SELECT MAX(`acc_num`) FROM customer_details")
            results = cursor.fetchall()
            for row in results:
                Accountnum = row[0]
        except:
            print ("        Error: unable to fetch data")
        db.close()
        create_account(Accountnum)
    elif (choice == 2):
        print("     Enter account number carefully to remove account")
        remove()
    elif (choice==3):
        print ("    Enter account number to active account")
        activeclient()
    elif (choice == 4):
        print("     Enter acoount number to view account details")
        view_account()
    elif (choice == 5):
        print("     You choose to view all acount details\n")
        view_allaccount()
    elif (choice == 6):
        deposit()
    elif (choice == 7):
        withdrawl()
    elif(choice==8):
        printhistory()
    elif(choice==0):
        print("    \n\n YOU LOGOUT SUCCESFULLY\n")
    else:
        print("enter right choice")

########################   create Account ####################################

def create_account(Accountnum):
    fname = input("Firstname:")
    lname = input("Lastname:")
    name = "%s %s" % (fname, lname)
    age=int(input("Age:"))
    sex=input("Sex:")
    print("Address:")
    line1=input("Line1:")
```

```
    city=input("City:")
    state=input("State:")
    address="%s %s %s"%(line1,city,state)
    income = int(input("income:"))
    phone=int(input("Phone number:"))
    status='TRUE'
    amount=int(input("Enter amount:"))
    acc_num=Accountnum+1
    userid='uid'+str(acc_num)
    passcode=userid
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    sql = "INSERT INTO `customer_details` (`customertid`, `acc_num`, `name`, `age`,
`sex`, `address`, `phone`, `income`, `status`, `date`, `amount`, `userid`, `passcode`)
VALUES (NULL, '%d', '%s', '%d', '%s', '%s', '%d', '%d', '%s', CURRENT_TIME(), '%d',
'%s', '%s')" %(acc_num, name, age, sex, address,
phone,income,status,amount,userid,passcode);
    try:
        cursor.execute(sql)
        print("                        Account Created Succesfully and {} is your
account number\n".format(acc_num))
        db.commit()
    except:
        db.rollback()
    print ("\n\n\n")
    customer_menu()
    db.close()

###################  Remove account ##########################

def remove():
    acc_num = int(input("Enter Acount Number:"))
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    sql = "UPDATE `customer_details` SET status='FALSE' WHERE `acc_num`='%d'"
%(acc_num)
    try:
        cursor.execute(sql)
        print("                      Account remove succesfully\n")
        db.commit()
    except:
        db.rollback()
    print ("\n\n")
    customer_menu()
    db.close()

#################### update to active account ############################

def activeclient():
    acc_num = int(input("Enter Acount Number:"))
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor=db.cursor()
    try:
        cursor.execute("SELECT * FROM `customer_details` WHERE `acc_num`='%d'" %
(acc_num))
        results=cursor.fetchall()
        columns = ['Customer Id', ' Name', ' Age', ' sex', ' Address', ' Phone', '
income', ' status', ' date',' amount']
        use = pd.DataFrame([['', '', '', '', '', '', '', '', '', '']], columns=columns)
        for row in results:
            status = row[8]
        if(status=='FALSE'):
            cursor.execute("UPDATE `customer_details` SET status='TRUE' WHERE
`acc_num`='%d'" % (acc_num))
            print("         Account is activated now\n")
        else:
            print("         Account is already active\n")
    except:
        print("invalid operation !!!!\n")
    print ("\n")
    customer_menu()
```

```
######################## View Particuler Account ##########################

def view_account():
    acc_num = int(input("Enter Acount Number:"))
    db = pymysql.connect("localhost","root","","banking" )
    cursor = db.cursor()
    try:
        cursor.execute("SELECT * FROM `customer_details` WHERE `acc_num`='%d'"
%(acc_num))
        results = cursor.fetchall()
        columns = ['Customer Id', ' Name', ' Age', ' sex', ' Address', ' Phone','
income',' status',' date',' amount']
        use = pd.DataFrame([['', '', '', '', '', '', '', '', '', '']], columns=columns)
        for row in results:
            customer_id=row[0]
            accountn=row[1]
            name = row[2]
            age = row[3]
            sex = row[4]
            address = row[5]
            phone = row[6]
            income=row[7]
            status=row[8]
            date=row[9]
            amount=row[10]
        if(acc_num==accountn):
            use =
use.append(pd.DataFrame([[customer_id,name,age,sex,address,phone,income,status,date,amo
unt]], columns=columns))
            print(use)
        else:
            print("Wrong Account Number\n")
    except:
        print ("Error: unable to fetch data\n")
    db.close()
    print ("\n\n")
    customer_menu()

###################### view All Account Details ##############################

def view_allaccount():
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    try:
        cursor.execute("SELECT * FROM `customer_details` WHERE `amount`>0")
        results = cursor.fetchall()
        columns = ['Customer Id','Account
Number','Name','Age','sex','Address','Phone','Income','status','date','amount']
        use = pd.DataFrame([['','','','','','','','','','','']], columns=columns)
        for row in results:
            customer_id = row[0]
            account_num=row[1]
            name = row[2]
            age = row[3]
            sex = row[4]
            address = row[5]
            phone = row[6]
            income = row[7]
            status = row[8]
            date = row[9]
            amount = row[10]
            use = use.append(pd.DataFrame([[customer_id,account_num, name, age, sex,
address, phone, income, status, date, amount]],columns=columns))
        print (use)
    except:
        print ("Error: unable to fetch data\n")
    db.close()
    print ("\n\n")
    customer_menu()
```

```
########### update withdrawl money in trans_his ############################3

def update_withdrawl(account_number,withdrawlmoney,status):
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    try:
        cursor.execute("SELECT * FROM `customer_details` WHERE `acc_num`='%d'" %
(account_number))
        results = cursor.fetchall()
        for row in results:
            amount=row[10]
        cursor.execute("INSERT INTO `transaction_histoty`(`tid`, `credit`, `debit`,
`balance`, `date`, `account`,`status`) VALUES
(NULL,0,'%d','%d',CURRENT_TIME,'%d','%s')"
%(withdrawlmoney,amount,account_number,status))
        print("Withdrawl Succesfull\n")
        db.commit()
    except:
        print ("Error: Invalid account details!\n")
    customer_menu()
    db.close()

#########   update deposit trans in tran_his ###################

def update_deposit(account_number,depositmoney,status):
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    try:
        cursor.execute("SELECT * FROM `customer_details` WHERE `acc_num`='%d'" %
(account_number))
        results = cursor.fetchall()
        for row in results:
            amount = row[10]
        cursor.execute("INSERT INTO `transaction_histoty`(`tid`, `credit`, `debit`,
`balance`, `date`, `account`,`status`) VALUES
(NULL,'%d',0,'%d',CURRENT_TIME,'%d','%s')" % (depositmoney, amount, account_number,
status))
        print("Money deposit Succesfull\n")
        db.commit()
    except:
        print ("Error: Invalid!\n")
    customer_menu()
    db.close()
####################  Deposit ######################

def deposit():
    account_number = int(input("Enter account Number=ACC"))
    depositmoney = int(input("Enter Money="))
    status="SELF"
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    sql="SELECT * FROM `customer_details` WHERE `acc_num`='%d'" % (account_number)
    sql1 = "UPDATE `customer_details` SET `amount`=amount+'%d' WHERE `acc_num` = '%d'"
% (depositmoney, account_number)
    try:
        cursor.execute(sql)
        results=cursor.fetchall()
        for row in results:
            stat_us = row[8]
            if (stat_us == "TRUE"):
                cursor.execute(sql1)
                update_deposit(account_number,depositmoney,status)
            else:
                print("Account is not activated,Active the account first\n\n")
                customer_menu()
        db.commit()
    except:
        db.rollback()
    db.close()
######################### Withdrawl money ############################
```

```
def withdrawl():
    account_number = int(input("Enter account Number=ACC"))
    withdrawlmoney = int(input("Enter Money="))
    status="SELF"
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    sql = "UPDATE `customer_details` SET `amount`=amount-'%d' WHERE `acc_num` = '%d'" %
(withdrawlmoney, account_number)
    try:
        cursor.execute("SELECT * FROM `customer_details` WHERE `acc_num`='%d'" %
(account_number))
        results = cursor.fetchall()
        for row in results:
            amount = row[10]
            stat_us=row[8]
            if(stat_us=="TRUE"):
                if (amount >withdrawlmoney):
                    cursor.execute(sql)
                    update_withdrawl(account_number,withdrawlmoney,status)
                    db.commit()
                else:
                    print("You dont have enough money to transfer.    current balance
is {}\n".format(amount))
                    customer_menu()
                db.commit()
            else:
                print("Account is not activated,Active the account first\n\n")
                customer_menu()
    except:
        db.rollback()
    db.close()
##############  Print History ######################

def printhistory():
    acc_num=int(input("Enter account number:ACC"))
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    sql="SELECT * FROM `transaction_histoty` WHERE `account`='%d'" % (acc_num)
    try:
        cursor.execute(sql)
        results = cursor.fetchall()
        columns = ['Transaction id', ' Debit', ' Credit', ' Balance', ' Date', ' Type']
        use = pd.DataFrame([['', '', '', '', '','']], columns=columns)
        for row in results:
            tid=row[0]
            debit=row[1]
            credit=row[2]
            balance=row[3]
            date=row[4]
            type=row[6]
            use=use.append(pd.DataFrame([[tid, debit, credit, balance,date,
type]],columns=columns))
        print(use)
        db.commit()
    except:
        print ("Error: Invalid account number!!\n")
    db.rollback()
    db.close()
    print ("\n\n")
    customer_menu()
############## Login operation for accountant ############################

def login_acc():
    count = 0
    while (count < 3):
        username = input("Username:")
        password = input("Password:")
        db = pymysql.connect("localhost", "root", "", "banking")
        cursor = db.cursor()
        try:
            cursor.execute("SELECT * FROM `login_accountant` WHERE `username`='%s' AND
```

```
`password`='%s'" % (username, password))
            results = cursor.fetchall()
            for row in results:
                Username = row[0]
                Password = row[1]
            if (Username == username):
                if (Password == password):
                    print("Login succesfull\n")
                    count=4
                    customer_menu()
                else:
                    print ("Error: Invalid username/Password!")
            db.commit()
        except:
            print("Error: Invalid username/Password!")
            count = count + 1
    db.rollback()
    db.close()
    if count == 3:
        print("More than 3 attempts are not allowed")

        ###################    update in trans_table     ########################

def trans_update(receiver,sender,final_value,final_value2,payble,name):
    status='IMPS'
    db=pymysql.connect("localhost","root","","banking")
    cursor=db.cursor()
    sql1="INSERT INTO `transaction_histoty`(`tid`, `credit`, `debit`, `balance`,
`date`, `account`,`status`) VALUES (NULL,'%d',0,'%d',CURRENT_TIME,'%d','%s')"
%(payble,final_value,receiver,status)
    sql2="INSERT INTO `transaction_histoty`(`tid`, `debit`, `credit`, `balance`,
`date`, `account`,`status`) VALUES (NULL,'%d',0,'%d',CURRENT_TIME,'%d','%s')"
%(payble,final_value2,sender,status)
    try:
        cursor.execute(sql1)
        cursor.execute(sql2)
        print("Transaction Successfull to {}".format(name))
        db.commit()
    except:
        print("wrong")
    db.rollback()
    db.close

############### money transfer  and update balance in customer details   ##########

def transfer_money(benf_name,benf_accNumber,amount,accountnumber):
    name=benf_name
    receiver=benf_accNumber
    sender=accountnumber
    payble=amount
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    sql1 = "UPDATE `customer_details` SET `amount`=amount-'%d' WHERE `acc_num` = '%d'"
% (payble, sender)
    sql2 = "UPDATE `customer_details` SET `amount`=amount+'%d' WHERE `acc_num` = '%d'"
% (payble, receiver)
    try:
        cursor.execute(sql1)
        cursor.execute(sql2)
        cursor.execute("SELECT * FROM `customer_details` WHERE `acc_num`='%d'" %
(receiver))
        results = cursor.fetchall()
        for row in results:
            final_value=row[10]
        cursor.execute("SELECT * FROM `customer_details` WHERE `acc_num`='%d'" %
(sender))
        results = cursor.fetchall()
        for row in results:
            final_value2 = row[10]
        trans_update(receiver,sender,final_value,final_value2,payble,name)
        db.commit()
```

```
    except:
        print("chech point")
        db.rollback()
    db.close()

    ###########  transfer account number  #  input name and number & amount  ####

def transfer(acc_number,name):
    accountnumber=acc_number
    print("Welcmome to Personal Bank of india           Name : {}".format(name))
    count = 0
    while(count<3):
        benf_accNumber=int(input("Enter bebeneficiary account number:ACC"))
        benf_name=input("Account holder name:")
        db = pymysql.connect("localhost", "root", "", "banking")
        cursor = db.cursor()
        try:
            cursor.execute("SELECT * FROM `customer_details` WHERE `name`='%s' AND
`acc_num`='%d'" % (benf_name,benf_accNumber))
            results = cursor.fetchall()
            for row in results:
                bname = row[2]
                bacc = row[1]
            if (bname == benf_name):
                if (bacc== benf_accNumber):
                    count=4
                    amount=int(input("Enter amount:"))
                    transfer_money(benf_name,benf_accNumber,amount,accountnumber)
            else:
                print("invalid name and account number")
                db.commit()
        except:
            count= count + 1
    entry_choice(name, acc_number)
    db.close()

   ##############   Login for customer  ###################

def trans_his(acc_number,name):
    print("                    Welcome {}    ".format(name))
    db = pymysql.connect("localhost", "root", "", "banking")
    cursor = db.cursor()
    try:
        cursor.execute("SELECT * FROM `transaction_histoty` WHERE `account`='%d'" %
(acc_number))
        results = cursor.fetchall()
        columns = ['Transaction id', ' Debit', ' Credit', ' Balance', ' Date', ' Type']
        use = pd.DataFrame([['', '', '', '', '', '']], columns=columns)
        for row in results:
            tid = row[0]
            debit = row[1]
            credit = row[2]
            balance = row[3]
            date = row[4]
            type = row[6]
            use = use.append(pd.DataFrame([[tid, debit, credit, balance, date, type]],
columns=columns))
        print ("\n")
        print(use)
        db.commit()
    except:
        print ("Error: Invalid Account number!")
    print ("\n\n")
    entry_choice(name, acc_number)
    db.rollback()
    db.close()
```

```
  ########################   entry choice  ####################

def entry_choice(name, acc_number):
    print("1.Transfer money")
    print ("2.Transaction History")
    print("3.LOGOUT")
    choice = int(input("Enter your choice:"))
    if (choice == 1):
        transfer(acc_number, name)
    elif (choice == 2):
        trans_his(acc_number,name)
    elif (choice == 3):
        print("YOU LOGOUT SUCCESFULLY")
    else:
        print("wrong choice")

###############    Login customer   ######################

def login_customer():
    count = 0
    while (count < 3):
        username = input("Username:")
        password = input("Password:")
        db = pymysql.connect("localhost", "root", "", "banking")
        cursor = db.cursor()
        try:
            cursor.execute("SELECT * FROM `customer_details` WHERE `userid`='%s' AND
`passcode`='%s'" %(username, password))
            results = cursor.fetchall()
            for row in results:
                acc_number=row[1]
                name=row[2]
            print("         Welcome {} \n".format(name))
            count=4
            entry_choice(name, acc_number)
            db.commit()
        except:
            print ("Error: Invalid username/Password!")
            count = count + 1
    db.close()
    if count == 3:
        print("More than 3 attempts are not allowed")

############## operation menu ################################################

print("Select operation.")
print("1.Accountant")
print("2.Customer")
choice=int(input("enter choice:"))
if(choice==1):
    print("Enter Login and password carefully")
    login_acc()
elif(choice==2):
    print("Enter Login and password carefully")
    login_customer()
else:
    print("enter right choice")

############################### End of the code ####################################
```

References

1. Learning **MYSQL**

Website: www.w3school.com

2. **PHP** and **MYSQL** video tutorials

Website: www.youtube.com, www.mysirg.com

3. Python Tutorial

Website: www.sololearn.com, www.tutorialpoints.com

4. Statistics and Machine Learning in Python Release 0.1 by **Edouard Duchesnay, Tommy Lofstedt**
5. Python notes provided by **Didactic It Solution Pvt. Ltds**

Certificate

This is to certify that Mr.Soumyajit Pal of Techno India Batanagar (MAKAUT), registration number: 153320110043 has successfully completed a project on "Bank management system" using Python under the guidance of Prof. Arnab Chakraborty

--- ----------

-

Director

Didactics IT solution LLP