2019-2020

By - Mehul Raj Singh

Class-12th science

Subject-

Computer Science

A PROJECT REPORT ON

"BANK MANAGEMENT SYSTEM"

IN PYTHON

CERTIFICATE

ACKNOWLEDGEMENT

It is with great pleasure that I find myself penning

Down these lines to express my sincere thanks to all

Those people who helped me a long way in

Complete this project.

The harmonious climate in our school provided

Proper atmosphere for creating this project. It was a

Privilege to have been guided by Ms. Darshana Soni.

I am so grateful to my parents and classmates who

helped me during the finalization of my project with

their constructive criticism and advice.

INDEX

INTRODUCTION

ABOUT PYTHON:

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level

debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

ABOUT MYSQL:

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB,

but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

ABOUT BANK MANAGEMENT SYSTEM:

Bank Management System is based on dot NET and is a major project fro students.It is used to Keep the records of clients,employee etc in Bank.The bank management system is an application for maintaining a person „C/S account in a bank. The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also to view reports of all accounts present. The following presentation provides the specification for the system.

SOURCE CODE

```
import pickle

import os

import pathlib

class Account :

    accNo = 0

    name = ''

    deposit=0

    type = ''


    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")
```

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

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 :"))

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

def withdrawAmount(self,amount):
```

```
        self.deposit -= amount


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


    def getAccountNo(self):
        return self.accNo
    def getAcccountHolderName(self):
        return self.name
    def getAccountType(self):
        return self.type
    def getDeposit(self):
        return self.deposit



def intro():
    print("\t\t\t\t**********************")
```

```
    print("\t\t\t\tBANK MANAGEMENT SYSTEM")
    print("\t\t\t\t**********************")

    input()



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


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")


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")


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')
```

```
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')
```

```
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')



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')



# start of the program

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 : ")
```

OUTPUT

C:\Windows\py.exe

```
                              **********************
                              BANK MANAGEMENT SYSTEM
                              **********************


        MAIN MENU
        1. NEW ACCOUNT
        2. DEPOSIT AMOUNT
        3. WITHDRAW AMOUNT
        4. BALANCE ENQUIRY
        5. ALL ACCOUNT HOLDER LIST
        6. CLOSE AN ACCOUNT
        7. MODIFY AN ACCOUNT
        8. EXIT
        Select Your Option (1-8)
1
Enter the account no : 1234
Enter the account holder name : MEHUL SINGH
Ente the type of account [C/S] : S
Enter The Initial amount(>=500 for Saving and >=1000 for current600



Account Created
```

C:\Windows\py.exe

```
                              **********************
                              BANK MANAGEMENT SYSTEM
                              **********************


        MAIN MENU
        1. NEW ACCOUNT
        2. DEPOSIT AMOUNT
        3. WITHDRAW AMOUNT
        4. BALANCE ENQUIRY
        5. ALL ACCOUNT HOLDER LIST
        6. CLOSE AN ACCOUNT
        7. MODIFY AN ACCOUNT
        8. EXIT
        Select Your Option (1-8)
2
        Enter The account No. : 1234
Enter the amount to deposit : 100
Your account is updted
```

C:\Windows\py.exe

```
                              ************************
                              BANK MANAGEMENT SYSTEM
                              ************************


        MAIN MENU
        1. NEW ACCOUNT
        2. DEPOSIT AMOUNT
        3. WITHDRAW AMOUNT
        4. BALANCE ENQUIRY
        5. ALL ACCOUNT HOLDER LIST
        6. CLOSE AN ACCOUNT
        7. MODIFY AN ACCOUNT
        8. EXIT
        Select Your Option (1-8)
3
        Enter The account No. : 1234
Enter the amount to withdraw : 200
```

C:\Windows\py.exe

```
                              ************************
                              BANK MANAGEMENT SYSTEM
                              ************************


        MAIN MENU
        1. NEW ACCOUNT
        2. DEPOSIT AMOUNT
        3. WITHDRAW AMOUNT
        4. BALANCE ENQUIRY
        5. ALL ACCOUNT HOLDER LIST
        6. CLOSE AN ACCOUNT
        7. MODIFY AN ACCOUNT
        8. EXIT
        Select Your Option (1-8)
4
        Enter The account No. : 1234
Your account Balance is =  500
```

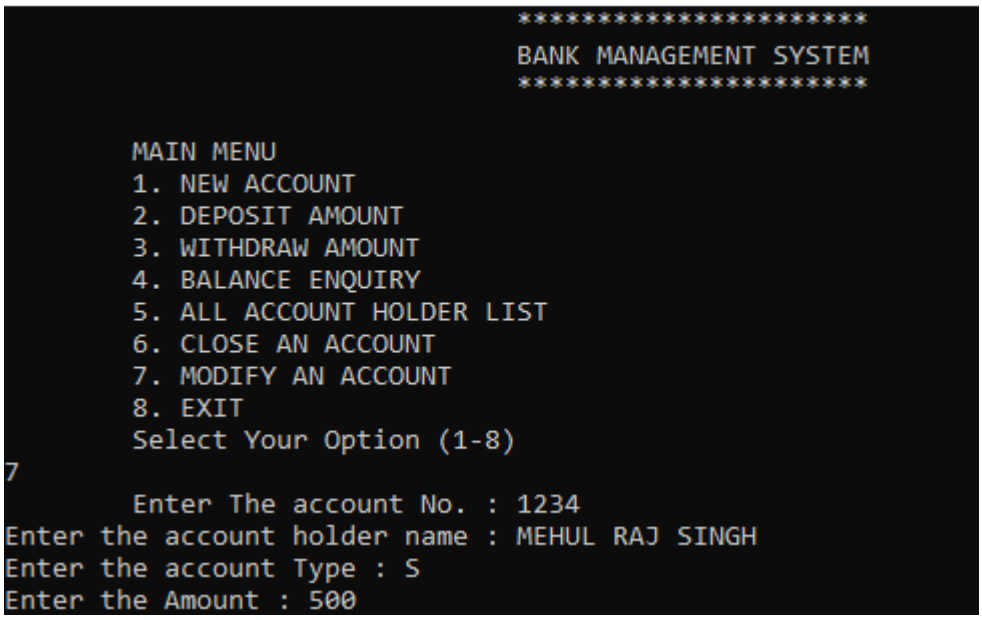
C:\Windows\py.exe

```
                              ************************
                              BANK MANAGEMENT SYSTEM
                              ************************


        MAIN MENU
        1. NEW ACCOUNT
        2. DEPOSIT AMOUNT
        3. WITHDRAW AMOUNT
        4. BALANCE ENQUIRY
        5. ALL ACCOUNT HOLDER LIST
        6. CLOSE AN ACCOUNT
        7. MODIFY AN ACCOUNT
        8. EXIT
        Select Your Option (1-8)
5
1234    MEHUL SINGH    S    500
```

C:\Windows\py.exe

```
                              ************************
                              BANK MANAGEMENT SYSTEM
                              ************************


        MAIN MENU
        1. NEW ACCOUNT
        2. DEPOSIT AMOUNT
        3. WITHDRAW AMOUNT
        4. BALANCE ENQUIRY
        5. ALL ACCOUNT HOLDER LIST
        6. CLOSE AN ACCOUNT
        7. MODIFY AN ACCOUNT
        8. EXIT
        Select Your Option (1-8)
6
        Enter The account No. : 1234
Enter your choice :
        MAIN MENU
        1. NEW ACCOUNT
        2. DEPOSIT AMOUNT
        3. WITHDRAW AMOUNT
        4. BALANCE ENQUIRY
        5. ALL ACCOUNT HOLDER LIST
        6. CLOSE AN ACCOUNT
        7. MODIFY AN ACCOUNT
        8. EXIT
        Select Your Option (1-8)
4
        Enter The account No. : 1234
No existing record with this number
```

C:\Windows\py.exe



BIBLIOGRAPHY

BOOKS:

- SUMITA ARORA-COMPUTER SCIENCE WITH PYTHON
- ARIHANT- ALL IN ONE COMPUTER SCIENCE CBSE

INTERNET:

- WEBSITE: WWW.PYTHON.ORG
- WEBSITE: WWW.WIKIPEDIA.ORG