PROFFESIONAL TRAINING REPORT

at

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

by

ANISETTI SIDHARTHA

[Reg.no. 38110531]

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119 TAMILNADU

AUGUST 2020

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabamauniversity.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of **ANISETTI SIDHARTHA [Reg.no.38110531]** who carried out the project entitled -**Bank Management System Project** under my supervision from April 2019 to August 2020.

InternalGuide

Dr. R. Sathyabama Krishna M.E.,Ph.D.,

Head of the Department

Dr. S. Vigneswari, M.E.,Ph.D.,

Submitted for Viva voce Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

I **ANISETTI SIDHARTHA** [**Reg.no. 38110531**] hereby declare that the Professional Training Report on -**Bank Management System Project** done by me under the guidance of **Dr.R.Sathyabama Krishna M.E.,Ph.D.,** at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E**., **Ph.D,. Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide – **Dr.R.Sathyabama Krsihna M.E.,Ph.D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

ABSTRACT

The Bank Management System is an application for maintaining a person‘s account in bank. In this project I tried to show the working of a bank system and cover the basic functionality of a Bank Management System.

The Main Aim of this project -Bank Management system‖ is to provide the simplest management of bank account and transaction between users and bank system. . In short, this projects mainly focus on CRUD (Create, Read, Update, Delete).This project is written in Python Programming Language. The project file contains a python script and a database file.

This Project is a simple console based system which is very easy to understand and use. Talking about the system, it contains all the basic functions which include creating a new account, view account holders record, withdraws and deposit amount, balance inquiry, closing an account and edit account details. In this project, there is no such login system. This means he/she can use all those available features easily without any restriction. It is too easy to use he/she can check the total bank account records easily.

.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION to BANK MANAGEMENT SYSTEM

A bank is a financial institution which performs the deposit and lending function. A bank allows a person with excess money (Saver) to deposit his money in the bank and earns an interest rate. Similarly, the bank lends to a person who needs money (investor/borrower) at an interest rate. Thus, the banks act as an intermediary between the saver and the borrower.

The bank usually takes a deposit from the public at a much lower rate called deposit rate and lends the money to the borrower at a higher interest rate called lending rate.

The difference between the deposit and lending rate is called ‗net interest spread', and the interest spread constitutes the banks income.

Essential Features/functions of the Bank

Fig 1.1.1 essential features/functions of the bank

Online banking, also known as internet banking or web banking, is an electronic payment system that enables customers of a bank or other financial institution to conduct a range of financial transactions through the financial institution's website. The online banking system will typically connect to or be part of the core banking system operated by a bank and is in contrast to branch banking which was the traditional way customers accessed banking services.

Some banks operate as a "direct bank" (or -virtual bank‖), where they rely completely on internet banking.

Internet banking software provides personal and corporate banking services offering features such as viewing account balances, obtaining statements, checking recent transactions, transferring money between accounts, and making payments.

***Features of Banking*:**

The following are the basic characteristics to capture the essential features of Banking:

(i) Dealing in money: The banks accept deposits from the public and advance the same as loans to the needy people. The deposits may be of different types – current, fixed, savings, etc. accounts. The deposits are accepted on various terms and conditions.

(ii) Deposits must be withdrawable: The deposits (other than fixed deposits) made by the public can be withdrawable by cheques, draft or otherwise, i.e., the bank issue and pay cheques. The deposits are usually withdrawable on demand.

(iii) Dealing with credit: The banks are the institutions that can create credit i.e., creation of additional money for lending. Thus, -creation of credit‖ is the unique feature of banking.

(iv) Commercial in nature: Since all the banking functions are carried on with the aim of making profit, it is regarded as a commercial institution.

(v) Nature of agent: Besides the basic function of accepting deposits and lending money as loans, bank possesses the character of an agent because of its various agency services.

Features of Online Banking

- Check the account statement online.
- Open a fixed deposit account.
- Pay utility bills such as water bill and electricity bill.
- Make merchant payments.
- Transfer funds.
- Order for a cheque book.
- Buy general insurance.
- Recharge prepaid mobile/DTH.

In today‘s world, the way of functioning and managing the system has been totally changed. There is a sudden and adrupt changes in the structure, maintenance and modification, handling, levelling inside every system. Without managing system through computer applications and programming, the development of infrastructures are unfinished, There are many errors and drawbacks without use of computer programming and applications.

As we know that, -necessity is the mother of invention‖, so in today‘s challenging world, every system is developed and launched by the use of computer software and programming.

Now talking about this project -Bank Management System‖ is written in Python. The project file contains a python script (main.py) and a database file. This is a simple console based system which is very easy to understand and use. Talking about the system, it contains all the basic functions which include creating a new account, view account holders record, withdraws and deposit amount, balance inquiry, closing an account and edit account details. In this mini project, there is no such login system. This means he/she can use all those available features easily without any restriction. It is too easy to use he/she can check the total bank account records easily.

In order to run the project, you must have installed Python, on your PC. This is a simple Console Based system, specially written for the beginners.

1.1 WHY DO WE NEED THIS MODEL?

This simple console based Bank Management system provides the simplest management of bank account and transaction

The application will be extremely beneficial for the customers intending to use and operate their bank account and will get various benefits in the field of management of accounts on a clean and user-friendly platform.

Bank Management System is a sample application, which is especially generated and designed for the bank in order to enter the applicant information about his or ban account and can perform other functions like deposit , withdraw. It is user name and ID protected as well.

Main points about why we have Banks:

The term 'bank' now refers to any establishment licensed to accept deposits, pay interest, clear cheques, make loans and act as an intermediary in financial transactions. They also provide a range of other financial services to their customers. Banks are places where individuals and organisations can store money which they don't wish to use at any given time. The money (and other precious items) can be stored more safely in banks than on their own premises.

Because of their access to this money, banks are therefore in a position to lend money to individuals and organisations that, for one reason or another, need to spend more money than they have access to at any given time.

The banks can thus use the money they care for to earn interest, which means that the money which individuals and organisations have deposited with them can be made to grow.

In this way, the banking system acts to channel society's surplus wealth to individuals and organisations that can use it productively.

Much of this wealth is used to make factories, machines, roads, airports, railways, shops and other economic assets, which help society as a whole to grow richer.

At the same time, it makes those who deposit their money with the banks become wealthier as well, and so encourages them to do so. Banks are therefore essential for economic growth.

Banks are also facilitators (they help make things happen). They play a vital role in helping governments and large corporations to raise funds by issuing bonds and shares in the capital markets. They offer advice and they underwrite the issues, which takes much of the risk out of them.

Banks also act as middle men in financial transactions, by providing short-term credit to enable large financial transactions to go through smoothly. This is a great help in keeping the wheels of commerce turning.

Banks also help companies manage their assets and make best use of money available for investment, often investing on their behalf. They also help companies trading internationally to bring their wealth into the country.

Central banks also play an important part in regulating the monetary supply to control inflation as well as ensuring financial stability throughout the financial system.

All these activities make banks essential. They use the huge amounts of money that they control to make even more money for their depositors and businesses. A stable, safe and secure banking system that functions properly and that organisations are confident to use is one of the key ingredients in a healthy and successful economy.

Main points why we have this model:

- It creates a user friendly environment, where a normal user can access through all the benefits of the system.
- It provides security from unauthorized access, only admin or authorized users are access granted to the system.
- It increases efficiency and save time.
- No any danger and obstacles from external entities.
- Easy access of saved data inside the system.
- Complex Banking operations and Transaction operations are efficiently handled by the application.
- It is cost effective.
- It has ease of use along with complete reference.

- It is less time consuming and highly secure; hence time wastage can be avoided.
- Up to date records of the customers are maintained by the database used in this system.

CHAPTER 2

AIM AND SCOPE OF THIS PRESENT INVESTIGATION

2.1 AIM:

The project that we have undertaken aims to develop a banking system that is clean, user-friendly, simple and multi-functional. Development of this application includes a number of fields such that user feels comfortable and the system appears as dynamic to him. The project -Bank Management System‖ includes the following functionalities:

- All customer data are stored in a database file.
- Users should give their details to create new account.
- Customers can view their own account details and can use them as necessary
- Users can view their account balance.
- Users will be able to withdrawn deposit amount into their accounts.
- Users can modify their account details by giving their account details.
- Users can view all other account holder details.
- To provide flexibility for secure and safe transactions.
- For better performance.
- Reducing man power.
- For doing work more accurately.
- Faster performance.

For this model we are mainly concerned with developing a banking system where a user can submit his/her deposit amount to bank if he/she has an account or create a new account in this bank. User can also view the status. Can view account balance. One can easily maintain the above things if he/she has an account by login through his unique account number.

2.2 PROBLEM STATEMENT

The Bank Management System is an application for maintaining a person's account in a bank Organization need to effectively define and managerequirements to ensure they are meeting needs of the customer, So here we try to create simple banking environment for users which includes all the basic bank functionalities.

2.2 SCOPE

This application will be extremely beneficial for the Users intending to use and operate their bank account and will get various benefits in the field of management of accounts on a clean and user-friendly platform.Bank Management System, is a simple application, which is especially generated and designed for the bank in order to enter the application information about his or her bank account and can perform other functions like withdraw, deposit, balance enquiry, modifications, closing their account. It is user name and account number protected as well.

Following are the major objectives behind the proposed system:

- It creates a user friendly environment, where a normal user can access through all the benefits of the system.
- It can increase efficiency and saves the time.
- No any danger and obstacles from external entities.
- Easy access of saved data inside the system.
- Complete Banking operations and Transaction operations are efficiently handled by the application.
- It is cost effective.
- It has ease of use along with complete reference.
- It is highly secured and less time consuming; hence wastage can be avoided.
- Up to date records of the customers are maintained in the database files.

CHAPTER 3

EXPERIMENTAL OR MATERIALS AND METHODS, ALGORITHMS USED

3.1 MODULE DESCRIPTION

Several modules are used in developing this project code they are:

i. Intro
ii. WriteAccountsFile
iii. displayAll
iv. displaySp
v. depositAndWithdraw
vi. deleteAccount
vii. modifyAccount

Description of modules:

i. intro: This module is used to present the introduction screen, and take input

From the user to specific operation to be performed by the program

ii. WriteAccountsFile : This module is used to create new account of user by

Taking in account holder name, account number, type: savings/current,

Deposit amount. And save it in -accounts.data‖ file.

iii. displayAll : This module is used to display all the existing user accounts details

User names, account number, deposited amount. If there are no existing user details it prints -No records to display‖.

iv. displaySp : This module is used display the account details of the particular

Account holders by taking in account holders account number if you didn‘t enter an existing account number it prints an error message showing -no existing record

with this number‖. If at all there is no data in the database file it prints -no records to search‖.

v. depositAndWithdraw : This module is used to withdraw and deposit amount From/to the users account. To withdraw amount, accounts balance amount must be higher than the required amount. Or else it pints -You cannot withdraw larger amount‖. To deposit amount user has to give his valid account number or else it prints -No records to search‖. After withdraw/deposit it prints -Your account is updated‖.

E-R Diagram for understanding deposit and withdraw:

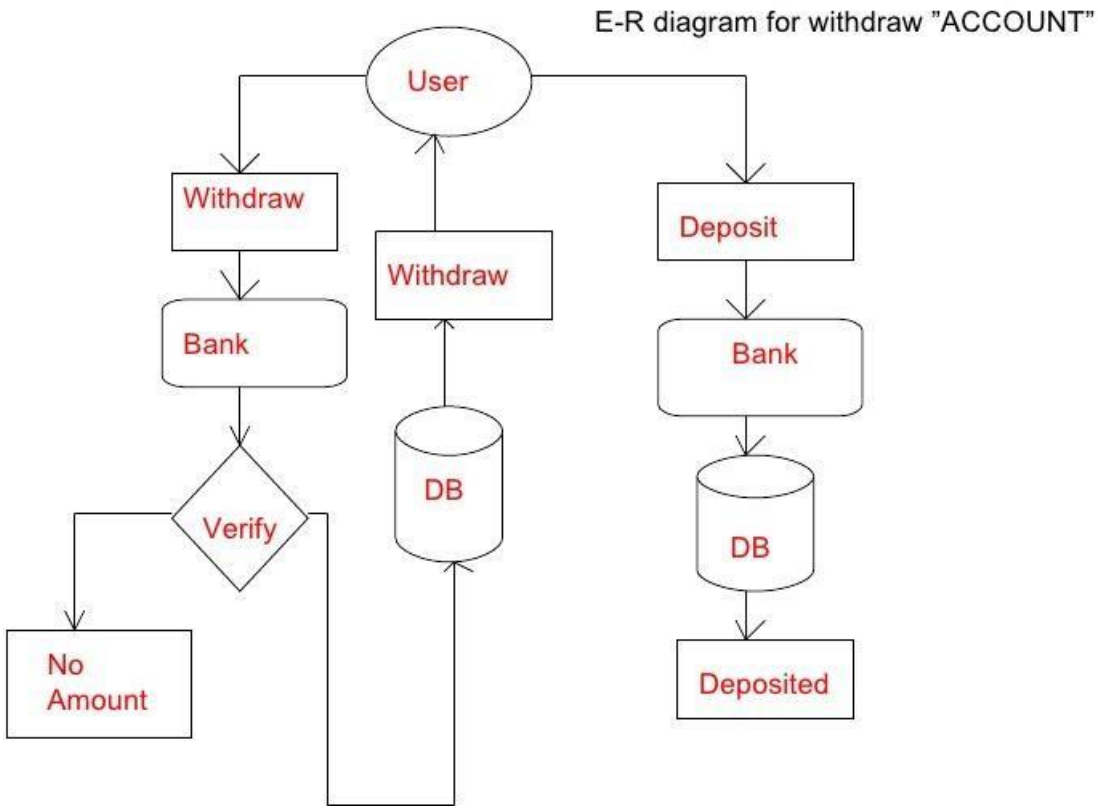


Fig 3.1.1

vi. deleteAccount : This module is used to close an existing account from the accounts.data data file by taking in the users account number.

vii. modifyAccount : This module is used change user account name, if there is any mistake or if you wish to. The changes will be permanent.

3.2 EXTERNAL LIBRARIES

There are 3 external libraries used in developing this project they are:

i. Pathlib
ii. Os
iii. pickle

These modules are preinstalled in anaconda distribution all you have to do is just import these modules to use them.

i. ***pathlib*** — Object-oriented filesystem paths

This module offers classes representing filesystem paths with semantics appropriate for different operating systems. Path classes are divided between pure paths, which provide purely computational operations without I/O, and concrete paths, which inherit from pure paths but also provide I/O operations.

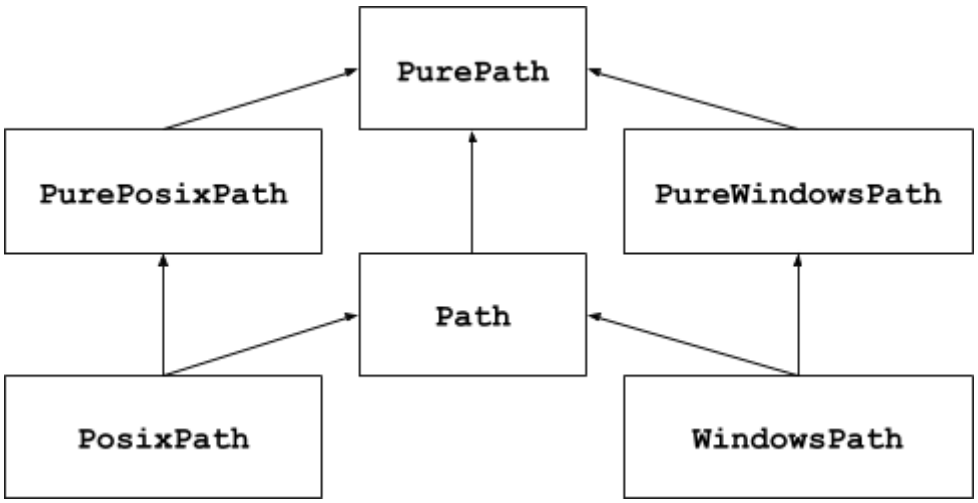


Fig 3.2.1.1

If you've never used this module before or just aren't sure which class is right for your task, path is most likely what you need. It instantiates a concrete path for the platform the code is running on.

Pure paths are useful in some special cases; for example:

1. If you want to manipulate Windows paths on a Unix machine (or vice versa). You cannot instantiate a windows path when running on Unix, but you can instantiate pure windows path.
2. You want to make sure that your code only manipulates paths without actually accessing the OS. In this case, instantiating one of the pure classes may be useful since those simply don‘t have any OS-accessing operations.

Basic use

Importing the main class:

```
>>>
>>> from pathlib import Path
```

Listing subdirectories:

```
>>>
>>> p = Path('.')
>>> [x for x in p.iterdir() if x.is_dir()]
[PosixPath('.hg'), PosixPath('docs'), PosixPath('dist'),
 PosixPath('_pycache_'), PosixPath('build')]
```

Listing Python source files in this directory tree:

```
>>>
>>> list(p.glob('**/*.py'))
[PosixPath('test_pathlib.py'), PosixPath('setup.py'),
 PosixPath('pathlib.py'), PosixPath('docs/conf.py'),
 PosixPath('build/lib/pathlib.py')]
```

Navigating inside a directory tree:

```
>>>
>>> p = Path('/etc')
>>> q = p / 'init.d' / 'reboot'
>>> q
PosixPath('/etc/init.d/reboot')
>>> q.resolve()
PosixPath('/etc/rc.d/init.d/halt')
```

Querying path properties:

```
>>>
```

```
>>> q.exists()
True
>>> q.is_dir()
False
```

Opening a file:

```
>>>
>>> with q.open() as f: f.readline()
...
'#!/bin/bash\n'
```

ii. *os* — Miscellaneous operating system interfaces

This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see open(), if you want to manipulate paths, see the os.path module, and if you want to read all the lines in all the files on the command line see the file input module. For creating temporary files and directories see the tempfile module, and for high-level file and directory handling see the shutil module.

Notes on the availability of these functions:

- The design of all built-in operating system dependent modules of Python is such that as long as the same functionality is available, it uses the same interface; for example, the function `os.stat(path)` returns stat information about *path* in the same format (which happens to have originated with the POSIX interface).
- Extensions peculiar to a particular operating system are also available through the os module, but using them is of course a threat to portability.
- All functions accepting path or file names accept both bytes and string objects, and result in an object of the same type, if a path or file name is returned.
- On VxWorks, os.fork, os.execv and os.spawn*p* are not supported.

exception `os.`**`error`**

An alias for the built-in oserror exception.

`os.name`

The name of the operating system dependent module imported. The following names have currently been registered: `'posix'`, `'nt'`, `'java'`.

Correspondence to tools in the os module:

Below is a table mapping various os functions to their corresponding purepath /path equivalent.

os and os.path	pathlib
os.path.abspath()	Path.resolve()
os.chmod()	Path.chmod()
os.mkdir()	Path.mkdir()
os.rename()	Path.rename()
os.replace()	Path.replace()
os.rmdir()	Path.rmdir()
os.remove(), os.unlink()	Path.unlink()
os.getcwd()	Path.cwd()
os.path.exists()	Path.exists()
os.path.expanduser()	Path.expanduser() and Path.home()
os.listdir()	Path.iterdir()

os and os.path	pathlib
os.path.isdir()	Path.is_dir()
os.path.isfile()	Path.is_file()
os.path.islink()	Path.is_symlink()
os.link()	Path.link_to()
os.symlink()	Path.symlink_to()
os.stat()	Path.stat(), Path.owner(), Path.group()
os.path.isabs()	PurePath.is_absolute()
os.path.join()	PurePath.joinpath()
os.path.basename()	PurePath.name
os.path.dirname()	PurePath.parent
os.path.samefile()	Path.samefile()
os.path.splitext()	PurePath.suffix

Table 3.2.1

iii. pickle :

In Python, you can use pickle to serialize (deserialize) an object structure into (from) a byte stream. Here are best practices for secure Python pickling.

Pickle in Python is primarily used in serializing and deserializing a Python object structure. In other words, it‘s the process of converting a Python object into a byte stream to store it in a file/database, maintain program state across sessions, or

transport data over the network. The pickled byte stream can be used to re-create the original object hierarchy by unpickling the stream. This whole process is similar to object serialization in Java or .Net.

When a byte stream is unpickled, the pickle module creates an instance of the original object first and then populates the instance with the correct data. To achieve this, the byte stream contains only the data specific to the original object instance. But having just the data alone may not be sufficient. To successfully unpickle the object, the pickled byte stream contains instructions to the unpickler to reconstruct the original object structure along with instruction operands, which help in populating the object structure.

According to the pickle module documentation, the following types can be pickled:

- None, true, and false
- Integers, long integers, floating point numbers, complex numbers
- Normal and Unicode strings
- Tuples, lists, sets, and dictionaries containing only picklable objects
- Functions defined at the top level of a module
- Built-in functions defined at the top level of a module
- Classes that are defined at the top level of a module

Pickle allows different objects to declare how they should be pickled using the reduce_method. Whenever an object is pickled, the__reduce_method defined by it gets called. This method returns either a string, which may represent the name of a Python global, or a tuple describing how to reconstruct this object when unpickling.

Generally the tuple consists of two arguments:

- A callable (which in most cases would be the name of the class to call)
- Arguments to be passed to the above callable

The pickle library will pickle each component of the tuple separately, and will call the callable on the provided arguments to construct the new object during the process of unpickling.

3.3 SOFTWARE AND HARDWARE SPECIFICATIONS:

It does not need any additional hardware of software to operate the program, but the following requirements should be strongly maintained.

Hardware Requirements:

- Pentium ii or above hardware.
- 512MB of RAM or higher.
- 800MHz processor or above.
- 20mb of hard disk space.

Software Requirements:

- Operating system WINDOWS 98 or higher.
- Anaconda distribution environment/ Jupyter notebook or any other editor to run python program.

3.4 INPUT SPECIFICATIONS:

After executing the code the user has to give certain inputs to get the required outputs they are:

i For creating new account the user has to give in their name, account number they intend to, type of their account savings/current, deposit amount, the user must deposit more than 500 for savings account and more than 1000 for current account.

ii To display balance deposit of a user, the user has to give in his/her account number.

iii To withdraw amount from his/her account the user has to give in his/her account number which is already created or else it prints an error message, amount to withdraw, which must be lower than his/her account balance or else it prints an error message.

iv To deposit amount from his/her account the user has to give in his/her account number which is already created or else it prints an error message.

v To close an account the user has to enter his/her account number which is already created or else it prints an error message.

vi To modify user account details, user has to enter his/her account number and give in his new account details.

CHAPTER 4

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

This Project is very easy to understand and use. The major objective for this project is to create a simple interface between user and bank management system so that user can understand how e-banking works. As of the current output we are successful in achieving our objective and creating an efficient bank management system. Talking about the system, it contains all the basic functions which include creating a new account, view account holders record, withdraws and deposit amount, balance inquiry, closing an account and edit account details. In this project, there is no such login system. This means he/she can use all those available features easily without any restriction. It is too easy to use he/she can check the total bank account records easily.

Every module we used in developing this project has been working successfully and the external modules we imported for creating database to store the data also working efficiently. There is no preprocessed data in this system it only takes in inputs from the users and store it the database and use it to perform all other functionalities

There are many advantages of using this program as it contains various features like:

- It is actually user friendly software, as it is just easy to use by just following the instructions which appeared on the screen.
- This program needs user account number to access user information, so that only authorised users are allowed to accessed through internal main system.
- Once a record has been saved, duplicate record can‘t be made. All the records have different account number so that there will not be any misplace of the records entered.

CHAPTER 5

SUMMARY AND CONCLUSIONS

This project -Bank Management System‖ is developed as to create simple interface between users and banking environment. And we are successful in reaching our objective by using simple modules and other external libraries in python. So by using simple basic functions and libraries in python we have created a simple banking environment such as creating account, depositing and withdraw of money, closing an account, modifying an account.

As we know that, no any program can be 100% reliable and efficient. So there are also some drawbacks from my system which are as follows:

- It cannot perform all the required functions as bank required it‘s simply a record of account of customer.
- System is not sharply a graphical user interface.
- It‘s not a multiuser and multitasking program. It can‘t perform various task at a single time

REFERENCES

1 https://en.wikipedia.org/wiki/Bank

2 https://docs.python.org/3/library/pathlib.html

3 https://docs.python.org/3/library/os.html

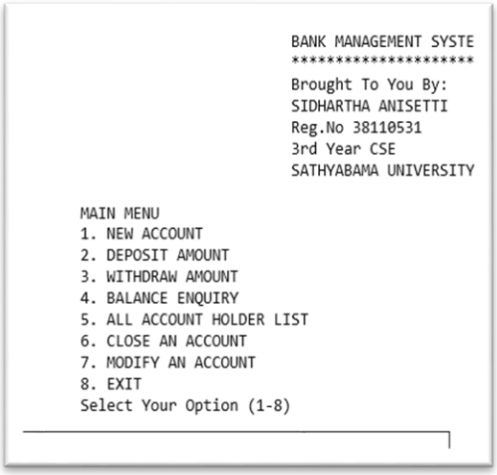
4 https://docs.python.org/3/library/pickle.html

5 https://www.udemy.com/course/python-the-complete-python-developer- course

APPENDIX

A. SCREENSHOTS

i. Introduction screen



ii. Creating account

iii. Depositing amount

```
nter any key : 2
       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)

       Enter The account No. : 2468
nter the amount to deposit : 5000
our account is updted

nter any key :
```

iv. Withdraw amount

```
       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)

       Enter The account No. : 13579
nter the amount to withdraw : 100

nter any key :
```

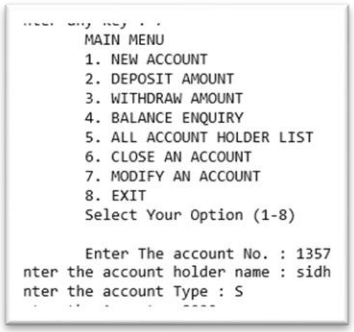
v. Balance enquiry

```
       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)

       Enter The account No. : 2468
our account Balance is =  8000

nter any key :
```

vi. All account holder list

vii. Closing an account

viii. Modifying an account



ix. All account holders list after deleting an account and after modifying an account

```
        Enter The account No. : 13579
Enter the account holder name : sidhu
Enter the account Type : S
Enter the Amount : 8999
Enter any key : 5
        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
```

x. Exit

```
Enter any key : 8
        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)
8
        Thanks for using bank managemnt system
```

B. 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**********")
```

```
    print("\t\t\t\tBrought To You By:")
    print("\t\t\t\t SIDHARTHA ANISETTI ")
    print("\t\t\t\t 3rd YEAR CSE ")
    print("\t\t\t\t red.no: 38110531 ")
    print("\t\t\t\t SATHYABAMA UNIVERSITY ")
    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 any key : ")
```