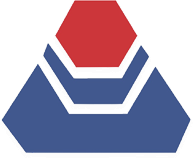
UNITED COLLEGE OF ENGINEERING & RESEARCH

NAINI, PRAYAGRAJ

*Affiliated to Dr. APJ Abdul Kalam Technical University , Lucknow , Uttar Pradesh*



***SESSION 2022-23***

**PROJECT REPORT -**

***“BANK BOOT SYSTEM in Python”***

**Submitted by :-*VIVEK VISHWAKARMA***

**University Roll Number :- *2000100100194***

**Submitted by :-ANUPAM SHUKLA University Roll Number :- *2000100100036***

**Guided by :- *Mr. Bhanu Pratap Rai***

**Submitted To**

***Department of Computer Science & Engineering , UCER***

***DATE :- 21/12/2023***

***PLACE :- Prayagraj***

TABLE OF CONTENTS

|  |  |
| --- | --- |
| **Topics** | **Page No.** |
| **DECLARATION** | **……………………………… 3** |
| **CERTIFICATE** | **……………………………… 4** |
| **ACKNOWLEDGEMENTS** | **……………………………… 5** |
| **CHAPTER 1 : Introduction** | **……………………………… 6** |
| **CHAPTER 2 : Design** | **……………………………… 8** |
| **CHAPTER 3 : Requirement Analysis** | **……………………………… 11** |
| **CHAPTER 4 : User Interface** | **……………………………… 12** |
| **CHAPTER 5 : Conclusion** | **……………………………… 23** |
| **REFERENCES** | **……………………………… 24** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE** |  | **PAGE NO.** |
| **FIG 2.1** | **………………………………** | **8** |
| **FIG 2.2** | **………………………………** | **9** |
| **FIG 2.3** | **………………………………** | **10** |
| **FIG 4.1** | **………………………………** | **20** |
| **FIG 4.2** | **………………………………** | **20** |
| **FIG 4.3** | **………………………………** | **21** |
| **FIG 4.4** | **………………………………** | **21** |
| **FIG 4.5** | **………………………………** | **22** |
| **FIG 4.6** | **………………………………** | **22** |

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: Vivek Vishwakarma

Anupam Shukla

CERTIFICATE

This is to certify that Project Report entitled “**BANK BOT SYSTEM**” which is submitted by **VIVEK VISHWAKARMA, ANUPAM SHUKLA** in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, is a record of the candidate own work carried out by him under my/our supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

**Date:**21/12/ 23

***ACKNOWLEDGEMENT***

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Third Year. We owe special debt of gratitude to Professor Mr. Vijay Divwedi Head of Department Computer Science & Engineering, United College Of Engineering And Research, Prayagraj for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor, Head, Department of Computer Science & Engineering, United College Of Engineering And Research, Prayagraj for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

SIGNATURE

Vivek Vishwakarma (2000100100194)

Anupam Shukla

(2000100100036)

CHAPTER 1 INTRODUCTION

* 1. Real Life Problem Analysis

To develop a software for solving financial work of a customer in banking environment in order to nurture the needs of an end banking user by providing simple way platform to perform banking task

* 1. Problem Identification

This project effort to provide facility to customers from various backgrounds and a chance to manage their account without visiting there respective bank branch

* 1. Problem Statement

Current Banking System has old mechanism that is for depositing of cash in bank accounts& management of accounts.

* 1. Existing Solution

Customers has to visit Bank’s Branch for account management & depositing cash and other account related work

* 1. Problem With Existing Solutions

Main Problem is long waiting queue and too much work load on Branch, one has to wait for 1 hour for depositing cash and managing his accounts services

* 1. Proposed Work

The system is designed to be user-friendly and easy to navigate, and it is hoped that it will be a valuable tool for Bank Customers & Staffs. It is simple console based BANK BOT SYSTEM provides the management of bank account and transaction to enable the user’s workspace to have additional functionalities which are not provided under a conventional banking software

,The main function of the Bank Bot System is to manage the details of account holder

.Depositing Cash Amount in Bank without visiting its Branch

CHAPTER 1 MODULE DESCRIPTION

* 1. DESCRIPTION

**Data File**

* + - This is used to store data file.data for the user by accepting input suchas account number, name and amount

Open account

* + - Opens a new account for the user by accepting input such as account number, name ,type,and minimum balance

Balance Enquiry

* + - Enables to search for the details of the given account number.
    - Displays only one account balance at a time

Deposit

* + - Provides options to deposit amount from the given account number

Withdraw

* + - Provides options to withdraw amount from the given account number

Close Account

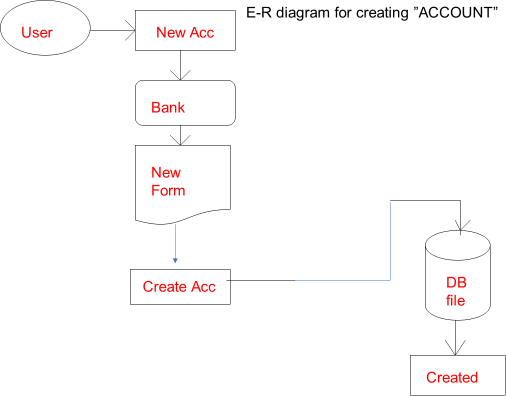
* + - Close the selected account from the bank

Display Account list

* + - Displays the list of all account details comprising of account number,name and balance amount
  1. Creating Account

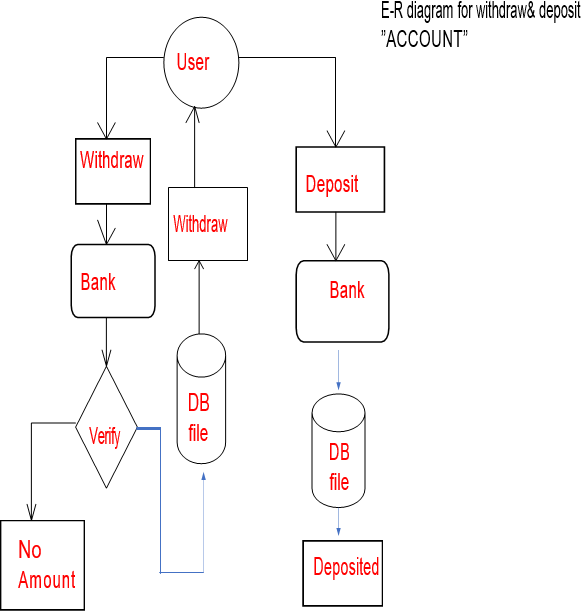
v **CHAPTER 2 DESIGN**

E-R DIAGRAM



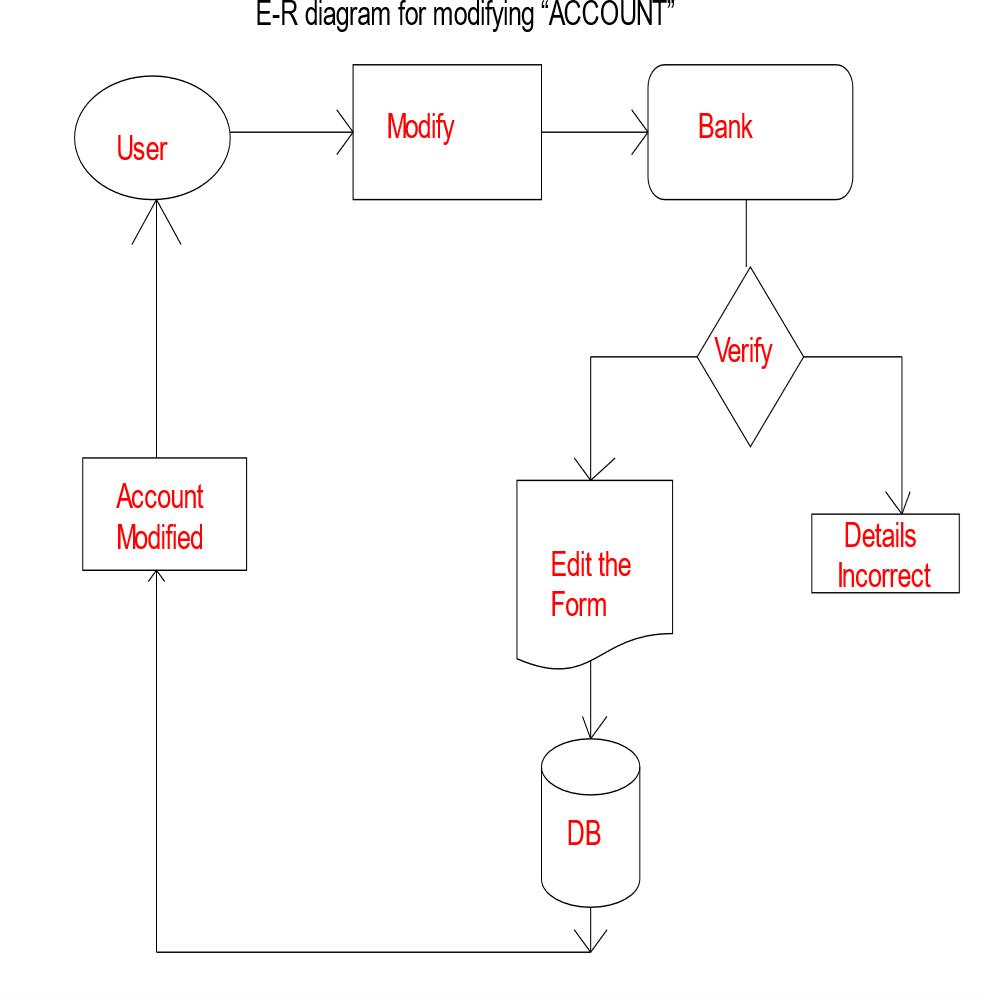
**Fig 2.1: ER Diagram for Creating Account**

* 1. Depositing & Withdrwal



**Fig 2.2: ER Diagram for Depositing & Withdrwal**

* 1. MODIFYING ACCOUNT



**Fig 2.3: ER Diagram for Modifying Account**

CHAPTER 3 REQUIREMENT ANALYSIS

* 1. Software Requirement
     1. OPERATING SYSTEM- WINDOW 8 - 11 ( AVAILABLE )
     2. Python IDLE
     3. Python Interpreter- Pycharm
  2. Hardware Requirement
     1. RAM - 4GB TO 16GB (available)
     2. HDD STORAGE CAPACITY - 256 GB (minimum)
     3. PROCESSOR - Quad Core (Minimum)

CHAPTER 4 ALGORITHM & USER INTERFACE

* + 1. Importing the libraries

import pickle import os import pathlib

* + 1. Creating a class account

class Account :

accNo=0 name = '' deposit=0 type = ''

4.1.4. 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)

4.1.3.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")

* + 1. 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 :"))

* + 1. Function to show the deposited or withdrawn amounts

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

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

= amount

* + 1. Function to show sessional report with all details

def report(self):

print(self.accNo, " ",self.name ," ",self.type," ", self.deposit)

* + 1. 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

* + 1. **Function to create an account** def writeAccount(): account = Account() account.createAccount() writeAccountsFile(account)
    2. 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")

* + 1. 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")

* + 1. 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')

* + 1. 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')

* + 1. 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 16item.name = input("Enter the account holder

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')

* + 1. 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')

* + 1. Asking the user for the task that he needs to do

ch='' num=0 intro()

while ch != 8:

#system("cls"); print("

⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛")

print("🏻\n\tç†:M† 𝑀𝘗𝐼𝖭 𝑀𝐸𝖭𝑡 M† :ç† \t🏻") print("🏻\n\t÷Ç1. NEW ACCOUNT⬛➵ \t🏻")

print("🏻\n\t÷Ç2. DEPOSIT AMOUNTˆ˘-Iw-.±˘

print("🏻\n\t÷Ç3. WITHDRAW AMOUNT ,×‘C‡ˇ8

* ")

\t🏻")

print("🏻\n\t÷Ç4. BALANCE ENQUIRY Ç \t🏻") print("🏻\n\t÷Ç5. ALL ACCOUNT HOLDER LIST🗄 🏻") print("🏻\n\t÷Ç6. CLOSE AN ACCOUNT + \t🏻")

print("🏻\n\tÇ÷7. MODIFY AN ACCOUNT•¸)‘|

print("🏻\n\t÷Ç8. EXIT \t🏻")

\t🏻")

print( "🏻" "\n\tSelect Your Option (1-8) \t🏻 ") print("

⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛⬛")

ch = input('\t\n⬛➜') #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': displayA

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 BOT SYSTEM ") break

else :

print("Invalid choice Selected ")

ch = input("PRESS M for MAIN MENU:⬛➜ ")

**4.2 MENU PAGE**

USER INTERFACE

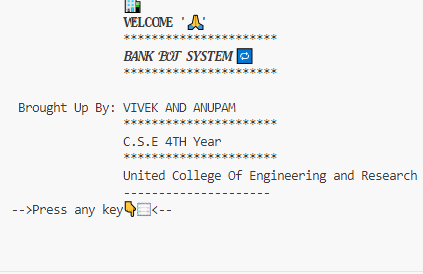


Fig 4.1: User interface for Menu page

Input- Choice we want 1-8

Output-Next phase of choice input

* + 1. **CREATE NEW ACCOUNT**

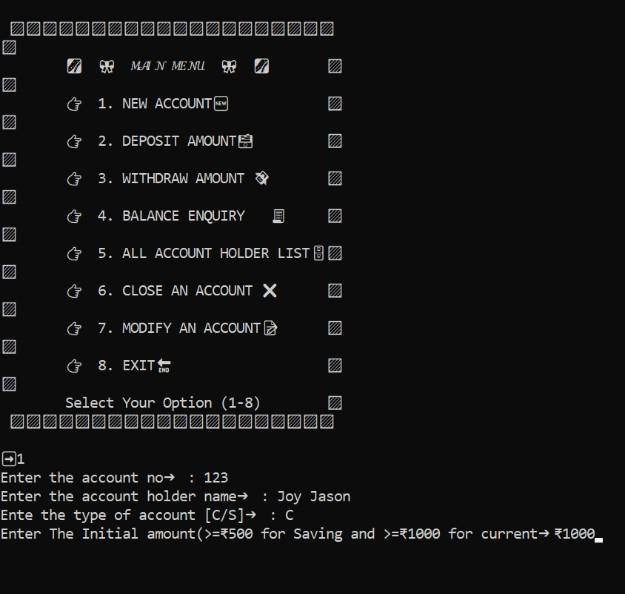


Fig 4.2: User interface for New Account

Input- Acoount No , Name , Account Type Output-

Created Account

* + 1. DESPOSIT

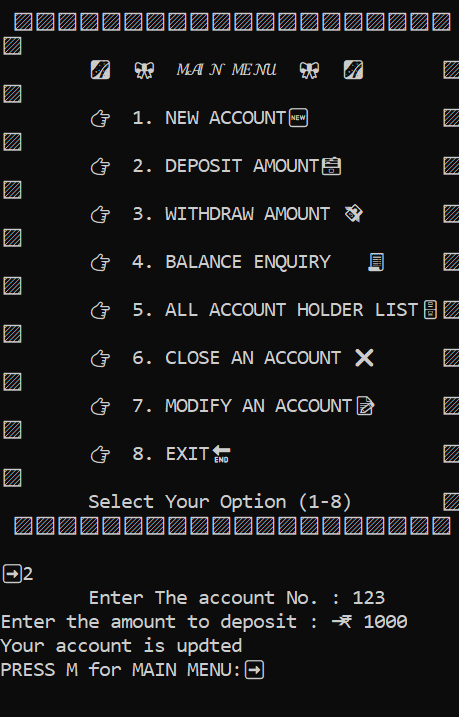


Fig 4.3: User interface for Deposit

Input-Account No. , Amount Output- Updated

* + 1. WITHDRAW

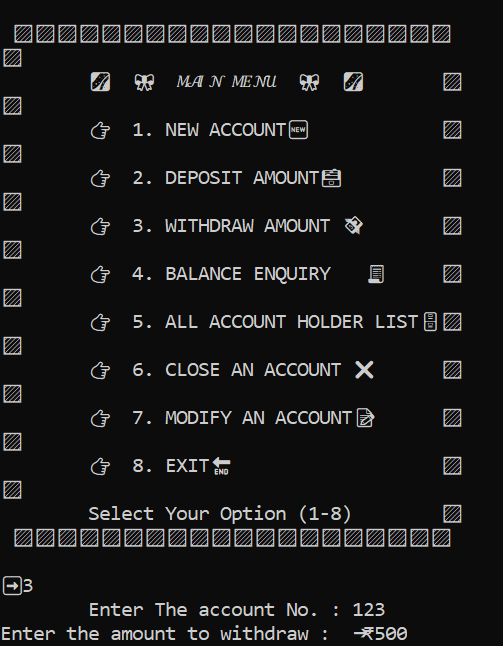
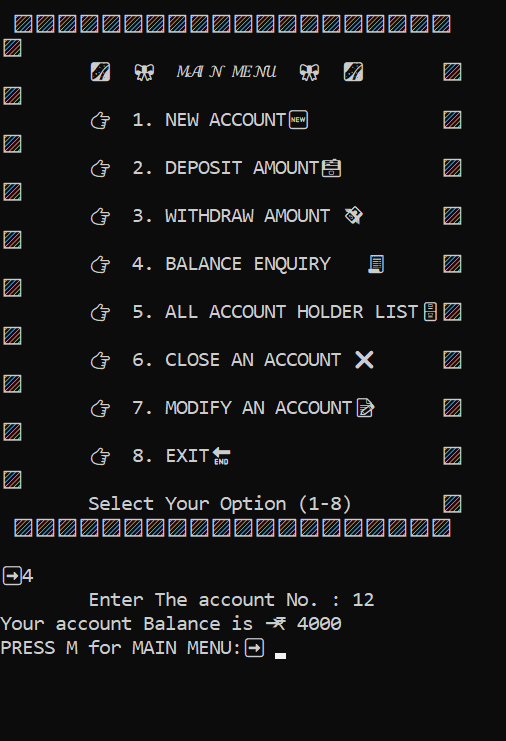


Fig 4.4: User interface for Withdraw

Input- Account No. , Amount

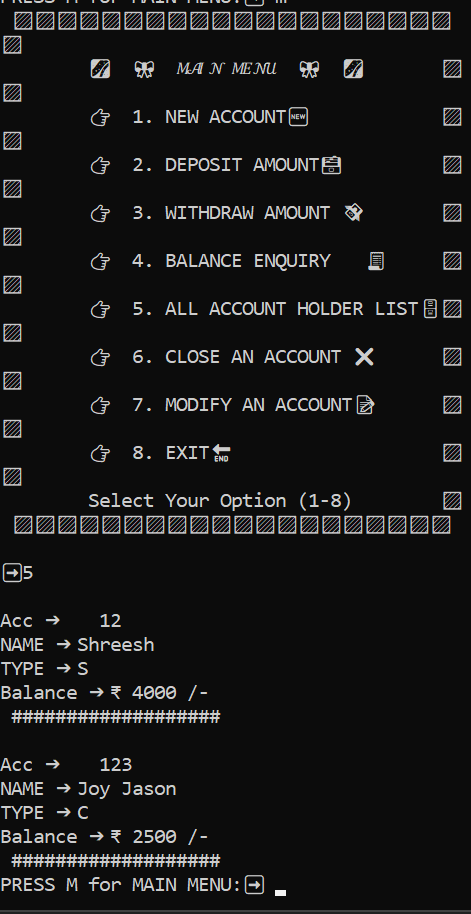
* + 1. BALANCE



Input- Account No. Output-Details Fetched

* + 1. ACCOUNT LIST

Fig 4.5: User interface for Balance



**Fig 4.6: User interface for Account list**

Input- 5 option choice

CONCLUSION

* + This project is developed to nurture the needs of a user in a banking sector by embedding all the tasks of transactions taking place in a bank.
  + Future version of this software will still be much enhanced than the current version1.0
  + Thus the Bank BOT System it is developed and executed successfully.

ADVANTAGES

* + Simplicity of Use
  + Reduction in Cost of Operation
  + Multi-banking Features
  + 24/7 Availability
  + Secure Transactions

REFERENCES

BOOKS

* + - *[Think Python](https://realpython.com/best-python-books/#think-python)*: The most basic of this list, *Think Python* provides a comprehensive Python reference.
    - *[Fluent Python](https://realpython.com/best-python-books/#fluent-python)*: While Python’s simplicity lets you quickly start coding, this book teaches you how to write idiomatic Python code, while going into several deep topics of the language.
    - *[Effective Python: 59 Ways to Write Better Python](https://realpython.com/best-python-books/#effective-python-59-ways-to-write-better-python)*: This relatively short book is a collection of 59 articles that, similarly to *Fluent Python*, focus on teaching you how to write truly Pythonic code.
    - *[Python Cookbook](https://realpython.com/best-python-books/#python-cookbook)*: As a cookbook, this will be a good reference on how to use Python to complete tasks you have done in another language.

REPORTS

* RESERVE BANK OF INDIA. (2007) Report of the Committee on Mechanism in Banking
* Industry
* RESERVE BANK OF INDIA. (1999) Report of the Committee on Computerization in
* Banking (The Rangarajan Committee Mumbai)
* RESERVE BANK OF INDIA. (2014) Report of the Committee on Reforms in
* Banking Sectors (The Narsimham Committee Mumbai)

WEBSITES

* Geeks for Geeks.com
* W3 Schools.com
* YouTube.com
* Template view.com
* Learncode.com