KENDRIYA VIDYALAYA BALLYGUNGE

COMPUTER SCIENCE PROJECT

Academic Year: 2020-21

Theme: Banking Management System

Made by: Ankur Chanda
Class: XII C
Roll.No: 08

Kendriya Vidyalaya Ballygunge

CERTIFICATE

This is to certify that Mr.Mrs _____, CBSE Roll.NO: _____ has successfully completed the Project Work entitled "BANK MANAGEMENT SYSTEM". In the subject Computer Science (083) laid down in the regulations of CBSE for the purpose of Practical Examination in Class XII to be held in Kendriya Vidyalaya Ballygunge on _____.

Mr. Debjit Biswas Sir
PGT Comp. Science

Examiner:
Name:_____
Signature:
Date:

TABLE OF CONTENTS

ACKNOWLEDGEMENT

Apart from the efforts of me, the success of my project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I express deep sense of gratitude to almighty god for giving me strength for the successful completion of the project.

I express my heartfelt gratitude to my parents for constant encouragement while carrying out this project.

I gratefully acknowledge the contribution of my group members who contributed in bringing the project up to this level, who continues to look after me despite my flaws.

I express my deep sense of gratitude to The Principal, Mrs. Suman Lata madam who has been continuously motivating and extending her helping hand to us.

I express my sincere thanks to Mr. Debjit Biswas, our Computer Science teacher above all who critically reviewed my project and helped in solving every problem that occurred during the implementation of the project.

The guidance and support received from all the members who contributed and who were contributing to this project, was vital for the success of this project. I am grateful for their constant support and help.

INTRODUCTION

"BANK MANAGEMENT SYSTEM" This project is useful for the bank employees as well as the customers to keep a track of account details. The emerging of digital system made information available on finger tips. By automating the transactions, one can view the details as and when required in no time. This project emphases on creation of new customer, managing the existing account holders in the bank, by making digital system one can generate daily reports, monthly reports and annual reports which can enhance the system.

OBJECTIVES OF THE PROJECT

The objective of this project is to let the students apply the programming knowledge into a real- world situation/problem and exposed the students how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.
2. Apply object oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
5. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

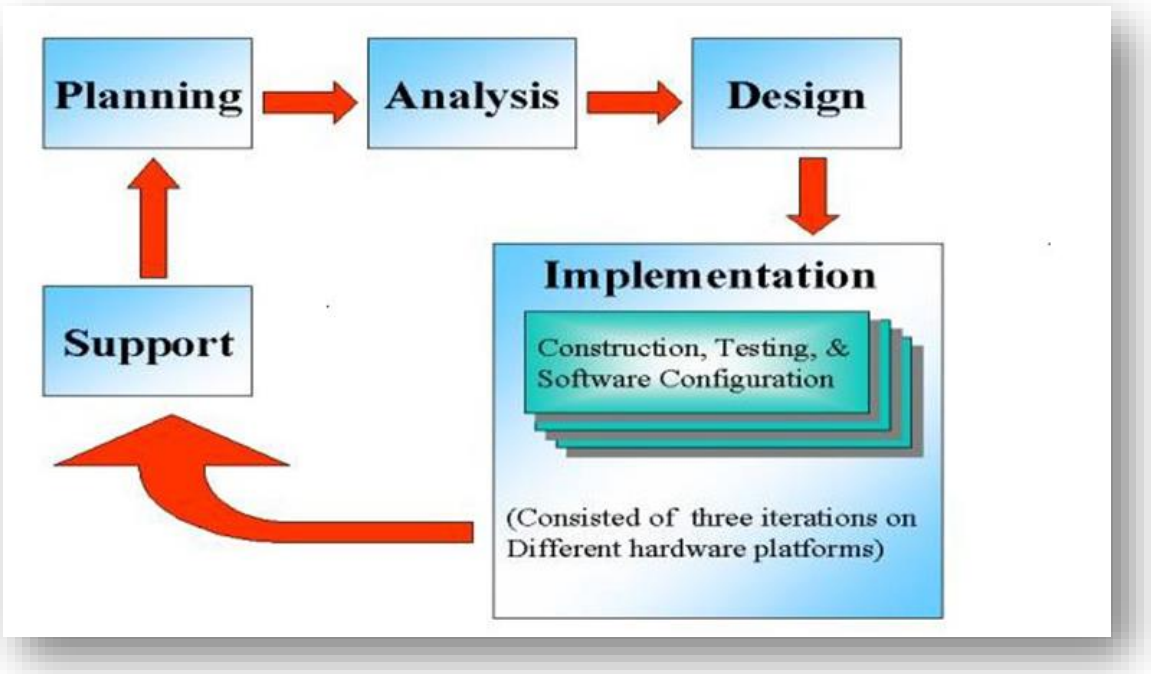
PROPOSED SYSTEM

Today one cannot afford to rely on the fallible human beings of be really wants to stand against today's merciless competition where not to wise saying "to err is human" no longer valid, it's outdated to rationalize your mistake. So, to keep pace with time, to bring about the best result without malfunctioning and greater efficiency so to replace the unending heaps of flies with a much sophisticated hard disk of the computer.

One has to use the data management software. Software has been an ascent in atomization various organisations. Many software products working are now in markets, which have helped in making the organizations work easier and efficiently. Data management initially hadto maintain a lot of ledgers and a lot of paperwork has to be done but now software producton this organization has made their work fasterand easier. Now only this software has to beloaded on the computer and work can be done.

This prevents a lot of time and money. Thework becomes fully automated and any information regarding the organization can beobtained by clicking the button. Moreover, nowit's an age of computers of and automating such an organization gives the better look.

SYSTEM DEVELOPMENT LIFE CYCLE

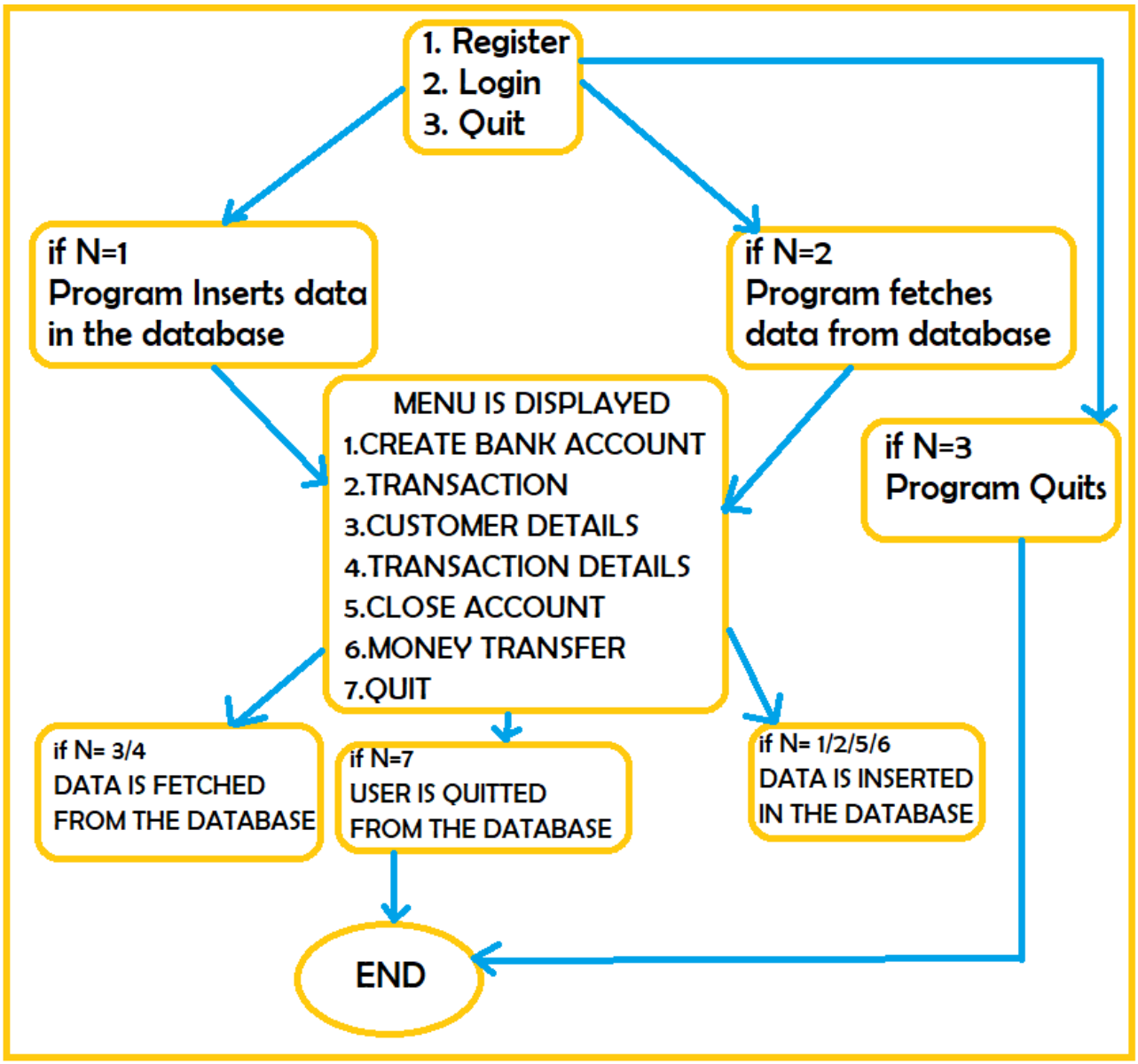


The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases.

Software development projects typically include initiation, planning, design, development, testing, implementation, and maintenance phases. However, the phases may be divided differently depending on the organization involved.

For example, initial project activities might be designated as request, requirements-definition, and planning phases, or initiation, concept-development, and planning phases. End users of the system under development should be involved in reviewing the output of each phase to ensure the system is being built to deliver the needed functionality.

FLOW CHART


1. Register
2. Login
3. Quit
if N=1
Program Inserts data in the database
if N=2
Program fetches data from database
if N=3
Program Quits
MENU IS DISPLAYED
1.CREATE BANK ACCOUNT
2.TRANSACTION
3.CUSTOMER DETAILS
4.TRANSACTION DETAILS
5.CLOSE ACCOUNT
6.MONEY TRANSFER
7.QUIT
if N= 3/4
DATA IS FETCHED FROM THE DATABASE
if N=7
USER IS QUITTED FROM THE DATABASE
if N= 1/2/5/6
DATA IS INSERTED IN THE DATABASE
END

SOURCE CODE

table.py

```
import  mysql.connector as sql
conn=sql.connect(host='localhost',user='root',passwd='manager',d
atabase='ankur')
if conn.is_connected():
     print('connected succesfully')
cur = conn.cursor()
cur.execute('create table customer_details(acct_no int primary
key,acct_name varchar(25) ,phone_no bigint(25)
check(phone_no>11),address varchar(25),cr_amt float )')
print("TABLE CREATED SUCCESSFULLY")
```

transaction_table.py

```
import  mysql.connector as sql
conn=sql.connect(host='localhost',user='root',passwd='manager',d
atabase='ankur')
cur=conn.cursor()
cur.execute('create table transactions(acct_no int(11),date date
,withdrawal_amt bigint(20),amount_added bigint(20) )')
print("TABLE CREATED")
```

user_table.py

```
import  mysql.connector as sql
conn=sql.connect(host='localhost',user='root',passwd='manager',d
atabase='ankur')
cur=conn.cursor()
cur.execute('create table user_table(username varchar(25)
primary key,passwrd varchar(25) not null )')
```

```
print("TABLE CREATED")
```

MAIN.py

```
#BANKING MANAGEMENT SYSTEM MAIN SOURCE CODE
import  mysql.connector as sql
import datetime as dt
import random
conn=sql.connect(host='localhost',user='root',passwd='manager',d
atabase='ankur')
cur = conn.cursor()
a='y'
while a=='y':
     print('=========================WELCOME TO NATIONAL
BANK============================')
     print(dt.datetime.now())
     print('1.REGISTER')
     print()
     print('2.LOGIN')
     print()
     print('3.QUIT')

print('============================================================
====================')
     n=int(input('ENTER YOUR CHOICE='))
     print()
     if n==1:
          name=input('ENTER USERNAME=')
          print()
          passwd=int(input('ENTER A 4 DIGIT PASSWORD='))
          print()
          query="INSERT  INTO user_table
values('{}',{})".format(name,passwd)
```

```
            cur.execute(query)
            conn.commit()
            print()
            print('USER CREATED SUCCESFULLY')
            print("PLEASE LOGIN NOW")
            continue
      if n==2 :
            ne=input('ENTER YOUR USERNAME=')
            print()
            pwd=int(input('ENTER YOUR 4 DIGIT PASSWORD='))
            q1="select * from user_table where username='{}' and
passwrd={}".format(ne,pwd)
            cur.execute(q1)
            if cur.fetchone() is  None:
                  print('INVALID USERNAME/PASSWORD TRY AGAIN')
            else:
                  flag='t'
                  while flag=='t':

print("=========================================================")
                        print("                    NATIONAL BANK")

print("=========================================================")
                        print('1.CREATE BANK ACCOUNT')
                        print()
                        print('2.TRANSACTION')
                        print()
                        print('3.CUSTOMER DETAILS')
                        print()
                        print('4.TRANSACTION DETAILS')
                        print()
                        print('5.CLOSE ACCOUNT')
```

```
                    print()
                    print('6.MONEY TRANSFER')
                    print()
                    print('7.QUIT')

print("=====================================================")
                    g=int(input('ENTER YOUR
CHOICE(1/2/3/4/5/6/7)='))
                    print()
                    if g==1:

print("================================================")
                         print("      NATIONAL BANK CREATE
ACCOUNT SYSTEM")

print("================================================")
                         acc_no=random.randrange(1,99000)
                         nm=input('ENTER YOUR ACCOUNT NAME=')
                         ph=int(input('ENTER YOUR PHONE
NUMBER='))
                         add=input('ENTER YOUR ADDRESS=')
                         cr=int(input('ENTER YOUR CREDIT
AMOUNT='))
                         q2="Select * from customer_details
where acct_no={}".format(acc_no)
                         cur.execute(q2)
                         if cur.fetchone() is None:
                              ind="INSERT  INTO customer_details
values({},'{}',{},'{}',{})".format(acc_no,nm,ph,add,cr)
                              cur.execute(ind)
                              print('ACCOUNT CREATED
SUCCESSFULLY!!!!!')
```

```
                              print("YOUR ACCOUNT NO IS-
",acc_no)
                              conn.commit()
                         else:
                              print("ACCOUNT NUMBER ALREADY
EXISTS! TRY AGAIN")
                    elif g==2:

print("================================================")
                         print("NATIONAL BANK")
                         print("MAKE A TRANSACTION")

print("================================================")
                         ac=int(input('ENTER YOUR ACCOUNT
NUMBER='))
                         cur.execute('select * from
customer_details where acct_no={}'.format(ac))
                         data=cur.fetchall()
                         conn.commit()
                         if cur.rowcount==0:
                              print('ACCOUNT NUMBER INVALID')
                         else:
                              print('1.WITHDRAW AMOUNT')
                              print()
                              print('2.ADD AMOUNT')
                              print()
                              x=int(input('ENTER YOUR CHOICE='))
                              if x==1:
                                   amt=int(input('ENTER
WITHDRAWAL AMOUNT='))
```

```
                                    cur.execute('update
customer_details set cr_amt=cr_amt -{} where
acct_no={}'.format(amt,ac))
                                    q="INSERT  INTO transactions
(acct_no,date,withdrawal_amt) values ({} , '{}' , {})
".format(ac,dt.datetime.today(),amt)
                                    cur.execute(q)
                                    conn.commit()
                                    print('ACCOUNT UPDATED
SUCCESFULLY!!!!!')
                               elif x==2:
                                    amnt=int(input('ENTER AMOUNT
TO BE ADDED='))
                                    cur.execute('update
customer_details set  cr_amt=cr_amt+{}  where
acct_no={}'.format(amnt,ac))
                                    qr="INSERT  INTO transactions
(acct_no,date,amount_added) values ({} , '{}' , {})
".format(ac,dt.datetime.today(),amnt)
                                    cur.execute(qr)
                                    conn.commit()
                                    print('ACCOUNT UPDATED
SUCCESFULLY!!!!!')
                               else:
                                    print("INVALID CHOICE!
SESSION ABORTED!")
                    elif g==3:

print("============================================")
                         print("NATIONAL BANK")
                         print("CUSTOMER DETAILS")
```

```
print("================================================")
                        acc=int(input('ENTER YOUR ACCOUNT
NUMBER='))
                        cur.execute('select * from
customer_details where acct_no={}'.format(acc))
                        if cur.fetchone() is  None:
                             print('INVALID ACCOUNT NUMBER')
                        else:
                             cur.execute('select * from
customer_details where acct_no={}'.format(acc))
                             data=cur.fetchall()
                             print("%10s"%"ACCOUNT
NO","%10s"%"ACCOUNT NAME","%10s"%"PHONE
NUMBER","%20s"%"ADDRESS","%10s"%"cr_amt")
                             for i in data:

print("%10s"%i[0],"%10s"%i[1],"%10s"%i[2],"%20s"%i[3],"%10s"%i[4
])

                    elif g==4:

print("================================================")
                        print("NATIONAL BANK")
                        print("TRANSACTION DETAILS")

print("================================================")
                        acn=int(input('ENTER YOUR ACCOUNT
NUMBER='))
                        cur.execute('select * from
customer_details where acct_no={}'.format(acn))
                        if cur.fetchone() is None:
```

```
                              print('INVALID ACCOUNT NUMBER! ')
                         else:
                              cur.execute('select * from
transactions where acct_no={}'.format(acn))
                              data=cur.fetchall()
                              print("%20s"%"ACCOUNT
NO","%20s"%"DATE","%20s"%"WITHDRAWAL AMOUNT","%20s"%"AMOUNT
ADDED")
                              for i in data:

print("%20s"%i[0],"%20s"%i[1],"%20s"%i[2],"%20s"%i[3])



                    elif g==5:

print("================================================")
                         print("NATIONAL BANK")
                         print("CLOSE ACCOUNT SYSTEM")

print("================================================")
                         print('CLOSE YOUR ACCOUNT')
                         acc_no=int(input('ENTER YOUR ACCOUNT
NUMBER='))
                         cur.execute('select * from
customer_details where acct_no={}'.format(acc_no))
                         data=cur.fetchall()
                         conn.commit()
                         if cur.rowcount==0:
                              print('ACCOUNT NUMBER INVALID')
                         else:
```

```
                                qer='select* from customer_details
where acct_no={}'.format(acc_no)
                                cur.execute(qer)
                                dat=cur.fetchall()
                                print("YOUR ACCOUNT DETAILS ARE:-
")
                                for i in dat:
                                     print("ACCOUNT NO-", i[0])
                                     print()
                                     print("ACCOUNT NAME -",i[1])
                                     print()
                                     print("PHONE NUMBER-", i[2])
                                     print()
                                     print("ADDRESS-",i[3])
                                     print()
                                     print("CREDIT AMOUNT-",i[4])
                                     nb=input("ARE YOU SURE YOU
WANT TO CLOSE YOUR ACCOUNT?(y/n):")
                                     if nb=='y':
                                          cur.execute('delete from
customer_details where acct_no={}'.format(acc_no) )
                                          print('ACCOUNT CLOSED
SUCCESFULLY')
                                     elif nb=='n':
                                          print("ACCOUNT NOT
CLOSED!")
                                          print("PLEASE CONTINUE
WITH YOUR BANKING OPERATIONS")
                                     else:
                                          print("INVALID CHOICE!
SESSION ABORTED!")
```

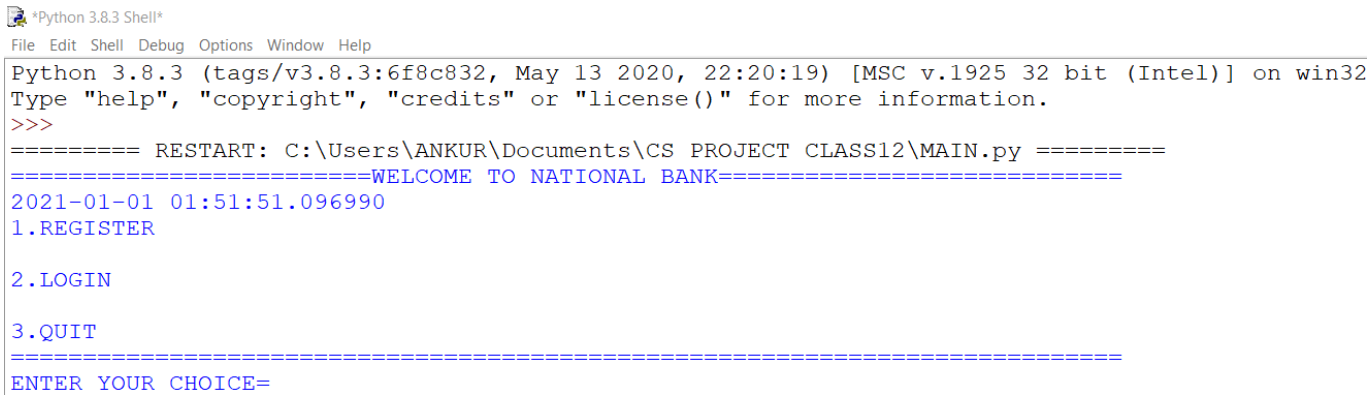
```
                    elif g == 6:

print("================================================")
                        print("NATIONAL BANK")
                        print("MONEY TRANSFER SYSTEM")
                        print("1.INTRA BANK MONEY TRANSFER")
                        print("2.INTER BANK MONEY TRANSFER")

print("================================================")
                        t=int(input("ENTER YOUR CHOICE(1/2):"))
                        if t==1:
                             a4=int(input("ENTER YOUR ACCOUNT
NUMBER:"))
                             a5=int(input("ENTER THE ACCOUNT
NUMBER WHERE MONEY IS TO BE TRANSFERRED:"))
                             q3="SELECT* FROM CUSTOMER_DETAILS
WHERE ACCT_NO={}".format(a5)
                             cur.execute(q3)
                             if cur.fetchone() is None:
                                  print("ACCOUNT NUMBER DOES
NOT EXISTS!")
                             else:
                                  ame=int(input("ENTER THE
AMOUNT TO BE TRANSFERRED:"))
                                  q5="UPDATE CUSTOMER_DETAILS
SET cr_amt= cr_amt-{} WHERE ACCT_NO={}".format(ame,a4)
                                  q4="UPDATE CUSTOMER_DETAILS
SET cr_amt=cr_amt+{} WHERE ACCT_NO={}".format(ame,a5)
                                  cur.execute(q4)
                                  cur.execute(q5)
                                  conn.commit()
```

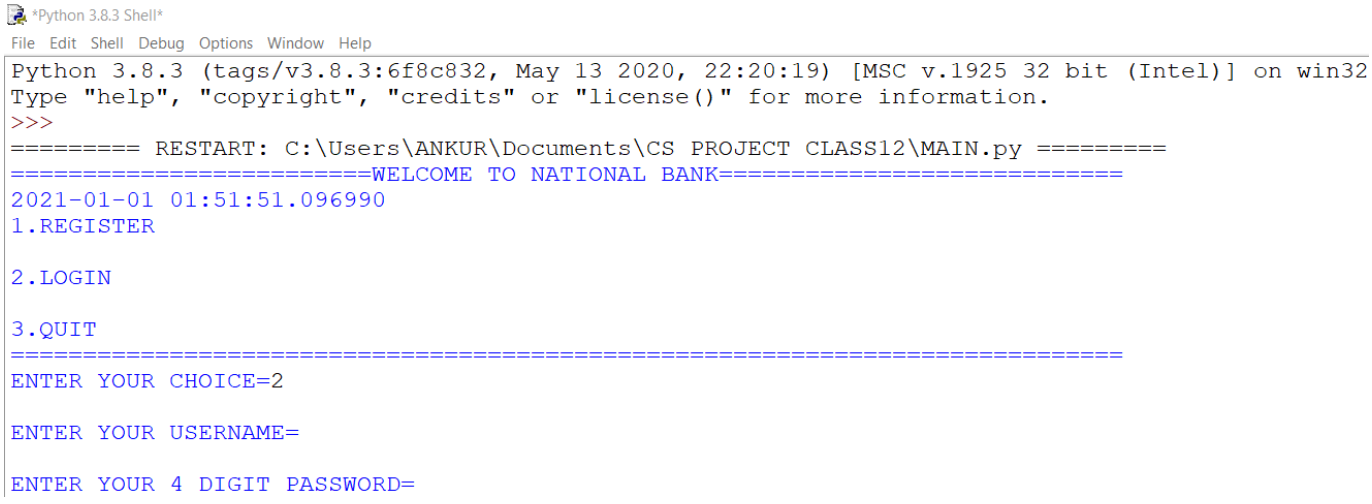
```
                                        print("AMOUNT TRANSFERRED
SUCCESFULLY")
                              elif t==2:
                                        bank=input("ENTER THE NAME OF THE
BANK:")
                                        ax=int(input("ENTER YOUR ACCOUNT
NUMBER:"))
                                        q7="SELECT* FROM CUSTOMER_DETAILS
WHERE ACCT_NO={}".format(ax)
                                        cur.execute(q7)
                                        if cur.fetchone() is None:
                                             print("ACCOUNT NUMBER DOES
NOT EXIST:")
                                        else:
                                             acno=int(input("ENTER THE
ACCOUNT NUMBER WHERE MONEY TO BE TRANSFERRED:"))
                                             money=int(input("ENTER AMOUNT
TO BE TRANSFERRED:"))
                                             q8="UPDATE CUSTOMER_DETAILS
SET cr_amt=cr_amt-{} WHERE ACCT_NO={}".format(money,ax)
                                             cur.execute(q8)
                                             conn.commit()
                                             print("YOUR REQUEST WILL BE
PROCESSED SHORTLY!")
                              else:
                                        print("INVALID CHOICE!")
                         elif g==7:

print("================================================")
                              print("NATIONAL BANK")
                              print("THANK YOU FOR USING OUR
SERVICES")
```

```
                    print("HOPE TO SEE YOU AGAIN")

print("================================================")
                    break
                else:
                    print("INVALID CHOICE! TRY AGAIN")
    break
    if n==3:
        print("THANK YOU")
        print("QUITTING..")
        break
    else:
        print("INVALID CHOICE! TRY AGAIN")
```

OUTPUT

Main page





Menu page

```
=========================WELCOME TO NATIONAL BANK=============================
2021-01-01 01:55:58.404799
1.REGISTER

2.LOGIN

3.QUIT
================================================================================
ENTER YOUR CHOICE=2

ENTER YOUR USERNAME=Rajiv98

ENTER YOUR 4 DIGIT PASSWORD=2298
=========================================================
                    NATIONAL BANK
=========================================================
1.CREATE BANK ACCOUNT

2.TRANSACTION

3.CUSTOMER DETAILS

4.TRANSACTION DETAILS

5.CLOSE ACCOUNT

6.MONEY TRANSFER

7.QUIT
=========================================================
ENTER YOUR CHOICE(1/2/3/4/5/6/7)=
```

Create Bank Account

```
==========================================================
                    NATIONAL BANK
==========================================================
1.CREATE BANK ACCOUNT

2.TRANSACTION

3.CUSTOMER DETAILS

4.TRANSACTION DETAILS

5.CLOSE ACCOUNT

6.MONEY TRANSFER

7.QUIT
==========================================================
ENTER YOUR CHOICE(1/2/3/4/5/6/7)=1

================================================
      NATIONAL BANK CREATE ACCOUNT SYSTEM
================================================
ENTER YOUR ACCOUNT NAME=Rajiv Gupta
ENTER YOUR PHONE NUMBER=9965433211
ENTER YOUR ADDRESS=Florida
ENTER YOUR CREDIT AMOUNT=10000
ACCOUNT CREATED SUCCESSFULLY!!!!!
YOUR ACCOUNT NO IS- 2444
```

Transaction

```
==========================================================
                    NATIONAL BANK
==========================================================
1.CREATE BANK ACCOUNT

2.TRANSACTION

3.CUSTOMER DETAILS

4.TRANSACTION DETAILS

5.CLOSE ACCOUNT

6.MONEY TRANSFER

7.QUIT
==========================================================
ENTER YOUR CHOICE(1/2/3/4/5/6/7)=2

================================================
NATIONAL BANK
MAKE A TRANSACTION
================================================
ENTER YOUR ACCOUNT NUMBER=2444
1.WITHDRAW AMOUNT

2.ADD AMOUNT

ENTER YOUR CHOICE=1
ENTER WITHDRAWAL AMOUNT=1000
ACCOUNT UPDATED SUCCESFULLY!!!!!
```

```
==========================================================
                    NATIONAL BANK
==========================================================
1.CREATE BANK ACCOUNT

2.TRANSACTION

3.CUSTOMER DETAILS

4.TRANSACTION DETAILS

5.CLOSE ACCOUNT

6.MONEY TRANSFER

7.QUIT
==========================================================
ENTER YOUR CHOICE(1/2/3/4/5/6/7)=2

================================================
NATIONAL BANK
MAKE A TRANSACTION
================================================
ENTER YOUR ACCOUNT NUMBER=2444
1.WITHDRAW AMOUNT

2.ADD AMOUNT

ENTER YOUR CHOICE=2
ENTER AMOUNT TO BE ADDED=3000
ACCOUNT UPDATED SUCCESFULLY!!!!!
```

Customer Details

```
=======================================================
                    NATIONAL BANK
=======================================================
1.CREATE BANK ACCOUNT

2.TRANSACTION

3.CUSTOMER DETAILS

4.TRANSACTION DETAILS

5.CLOSE ACCOUNT

6.MONEY TRANSFER

7.QUIT
=======================================================
ENTER YOUR CHOICE(1/2/3/4/5/6/7)=3

================================================
NATIONAL BANK
CUSTOMER DETAILS
================================================
ENTER YOUR ACCOUNT NUMBER=2444
ACCOUNT NO ACCOUNT NAME PHONE NUMBER              ADDRESS     cr_amt
      2444 Rajiv Gupta 9965433211              Florida    12000.0
```

Transaction Details

```
=======================================================
                    NATIONAL BANK
=======================================================
1.CREATE BANK ACCOUNT

2.TRANSACTION

3.CUSTOMER DETAILS

4.TRANSACTION DETAILS

5.CLOSE ACCOUNT

6.MONEY TRANSFER

7.QUIT
=======================================================
ENTER YOUR CHOICE(1/2/3/4/5/6/7)=4

================================================
NATIONAL BANK
TRANSACTION DETAILS
================================================
ENTER YOUR ACCOUNT NUMBER=2444
          ACCOUNT NO                 DATE    WITHDRAWAL AMOUNT         AMOUNT ADDED
                2444          2021-01-01                 1000                 None
                2444          2021-01-01                 None                 3000
```

Money Transfer

```
============================================================
                    NATIONAL BANK
============================================================
1.CREATE BANK ACCOUNT

2.TRANSACTION

3.CUSTOMER DETAILS

4.TRANSACTION DETAILS

5.CLOSE ACCOUNT

6.MONEY TRANSFER

7.QUIT
============================================================
ENTER YOUR CHOICE(1/2/3/4/5/6/7)=6

================================================
NATIONAL BANK
MONEY TRANSFER SYSTEM
1.INTRA BANK MONEY TRANSFER
2.INTER BANK MONEY TRANSFER
================================================
ENTER YOUR CHOICE(1/2):1
ENTER YOUR ACCOUNT NUMBER:2444
ENTER THE ACCOUNT NUMBER WHERE MONEY IS TO BE TRANSFERRED:48802
ENTER THE AMOUNT TO BE TRANSFERRED:2000
AMOUNT TRANSFERRED SUCCESFULLY
```

Close Account

```
==========================================================
                    NATIONAL BANK
==========================================================
1.CREATE BANK ACCOUNT

2.TRANSACTION

3.CUSTOMER DETAILS

4.TRANSACTION DETAILS

5.CLOSE ACCOUNT

6.MONEY TRANSFER

7.QUIT
==========================================================
ENTER YOUR CHOICE(1/2/3/4/5/6/7)=5

==================================================
NATIONAL BANK
CLOSE ACCOUNT SYSTEM
==================================================
CLOSE YOUR ACCOUNT
ENTER YOUR ACCOUNT NUMBER=2444
YOUR ACCOUNT DETAILS ARE:-
ACCOUNT NO- 2444

ACCOUNT NAME - Rajiv Gupta

PHONE NUMBER- 9965433211

ADDRESS- Florida

CREDIT AMOUNT- 10000.0
ARE YOU SURE YOU WANT TO CLOSE YOUR ACCOUNT?(y/n):y
ACCOUNT CLOSED SUCCESFULLY
```

TESTING

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

TESTING METHODS

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

BLACK BOX TESTING

Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

SPECIFICATION-BASED TESTING

Specification-based testing aims to test the functionality of software according to the applicable requirements.[16] Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the

test case. Specification-based testing is necessary, but it is insufficient to guard against certain risks

ADVANTAGES AND DISADVANTAGES

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code must have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers don't. But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed.

That's why there are situations when (1) a black box tester writes many test cases to check something that can be tested by only one test case, and/or (2) some parts of the back end are not tested at all. Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

WHITE BOX TESTING

White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these)

Types of white box testing:-

The following types of white box testing exist:

- api testing - Testing of the application using Public and Private APIs.
- Code coverage - creating tests to satisfy some criteria of code coverage.

For example, the test designer can create tests to cause all statements in the program to be executed at least once.

- fault injection methods.
- mutation testingmethods.
- static testing - White box testing includes all static testing.

CODE COMPLETENESS EVALUATION

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

- FunctionCoverage: Which reports on functions executed and
- StatementCoverage: Which reports on the number of lines executed to complete the test.

They both return coverage metric, measured as a percentage

HARDWARE AND SOFTWARE REQUIREMENTS

I.OPERATING SYSTEM	:	WINDOWS 7 AND ABOVE
II. PROCESSOR	:	PENTIUM(ANY) OR AMD ATHALON(3800+- 4200+ DUALCORE)
III. MOTHERBOARD	:	1.845 OR 915,995 FOR PENTIUM OR MSI K9MM-V VIAK8M800+8237R PLUS CHIPSET FOR AMD ATHALON
IV. RAM	:	512MB+
V. Hard disk	:	SATA 40 GB OR ABOVE
VI. CD/DVD r/w multi drive combo:		(If back up required)
VII. FLOPPY DRIVE 1.44 MB	:	(If Backup required)
VIII. MONITOR 14.1 or 15 -17 inch		
IX. Key board and mouse		
X. Printer	:	(if print is required – [Hard copy])

SOFTWARE REQUIREMENTS:

I. Windows OS
II. Python

INSTALLATION PROCEDURE

- Install Python and MySQL from the web.
- Open MySQL and create a database
- Run the table.py file
- Run the user_table.py file
- Run the transaction_table.py file
- Open the MAIN.py file
- Run the file
- Register your account

BIBLIOGRAPHY

1. Computer Science with Python-Class XI by: Sumita Arora
2. Website:https://www.w3schools.com/python/python_mysql_getstarted.asp

----------X----------X----------X----------