ATM MACHINE MANAGEMENT SYSTEM

NAME- Kriti Narayanan
Course- Computer Science

Contents

INTRODUCTION

The ATM MACHINE SOFTWARE is program which is same as normal ATM machine . It allows the user to

- Create Account
- Deposit Money
- Withdraw Money
- Check Balance
- Update Pin
- Update Phone Number
- Delete Account

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.

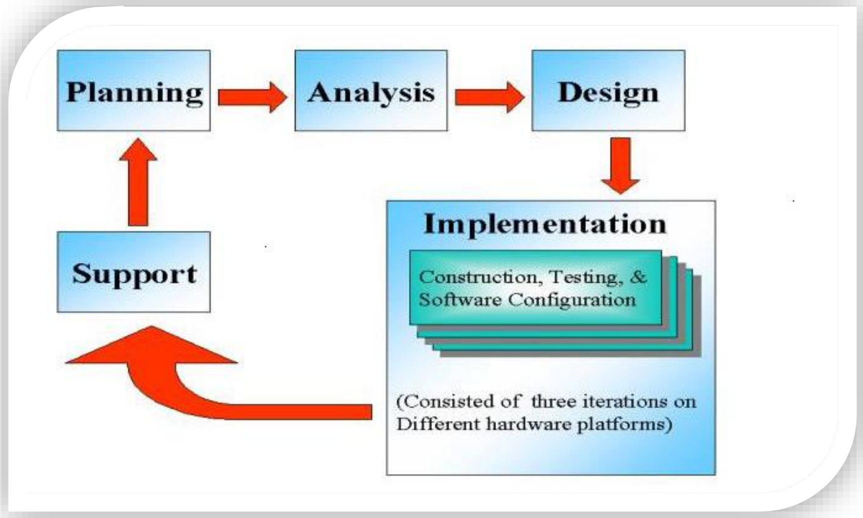
- Write programs utilizing modern software tools.
- Apply object oriented programming principles effectively when developing small to medium sized projects.
- Write effective procedural code to solve small to medium sized problems.
- Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.
- 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beings of be really wants to stand against today's merciless competition where not to wise saying "to err is human" no longer valid, it's outdated to rationalize your mistake.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 had to maintain a lot of ledgers and a lot of paper work has to be done but now software product on this organization has made their work faster and easier. Now only this software has to be loaded on the computer and work can be done. This prevents a lot of time and money. The work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it's an age of computers of and automating such an organization gives the better look.

SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)


Planning
Analysis
Design
Implementation
Construction, Testing, & Software Configuration
(Consisted of three iterations on Different hardware platforms)
Support

The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent 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.

PLANNING PHASE

The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, is necessary to coordinate activities and manage project risks effectively. The depth and formality of project plans should be commensurate with the characteristics and risks of a given project. Project plans refine the information gathered during the initiation phase by further identifying the specific activities and resources required to complete a project.

A critical part of a project manager' sjob is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and network requirements as possible. During this phase, a plan is developed that documents the approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and user input. Personnel assignments, costs, project schedule, and target dates are established.

A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and validation, and systems engineering management planning.

REQUIREMENTS ANALYSIS PHASE

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system performance, security, and maintainability requirements for the system. The requirements are defined in this phase to alevel of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and Evaluation MasterPlan.

The purposes of this phase are to:

- Further define and refine the functional and data requirements and document them in the Requirements Document,
- Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it),
- Develop detailed data and process models (system inputs, outputs, and the process.

- Develop the test and evaluation requirements that will be used to determine acceptable system performance.

DESIGN PHASE

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to scriptprograms during the development phase. Program designs are c onstructed in various ways. Using a top-down approach, designers first identify and link majorprogram components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design. Audit, security, and quality assurance personnel should be involved in the review and approval process. During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the software development, a variety of elements are considered in the design to mitigate risk. These include:

- Identifying potential risks and defining mitigating design features.
- Performing a security risk assessment.
- Developing a conversion plan to migrate current data to the new system.
- Determining the operating environment.
- Defining major subsystems and their inputs and outputs.
- Allocating processes to resources.
- Preparing detailed logic specifications for each software module. The result is a draft System Design Document which captures the preliminary design for the system.
- Everything requiring user input or approval is documented and reviewed by the user. Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system.
- This document receives a rigorous review byAgency technical and functional representatives to ensure that it satisfies the business requirements. Concurrent with the development of the system design, the Agency Project Manager begins development of the

Implementation Plan, Operations and Maintenance Manual, and the Training Plan.

DEVELOPMENT PHASE

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements. Programmers use various techniques to develop computer programs. The large transaction oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Effective completion of the previous stages is a key factor in the success of the Development phase. The Development phase consists of:

- Translating the detailed requirements and design into system components.
- Testing individual elements (units) for usability.
- Preparing for integration and testing of the IT system.

INTEGRATION AND TEST PHASE

Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. OIT Security staff assess the system security and issue a security certification and accreditation prior to installation/implementation.

Multiple levels of testing are performed, *including*:

- Testing at the development facility by the contractor and possibly supported by end users
- Testing as a deployed system with end users working together with contract personnel
- Operational testing by the end user alone performing all functions. Requirements are traced throughout testing ,a final Independent Verification & Validation evaluation is performed and all documentation is reviewed and accepted prior to acceptance of the system.

IMPLEMENTATION PHASE

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

OPERATIONS AND MAINTENANCE PHASE

The system operation is ongoing. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization’s needs. When modifications or changes are identified, the system may reenter the planning phase.

The purpose of this phase is to:

- Operate, maintain, and enhance the system.
- Certify that the system can process sensitive information.
- Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.
- Determine when the system needs to be modernized, replaced, or retired.

MODULES USED IN THE PROJECT

MYSQL.CONNECTOR

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249)
MySQL Connector/Python includes support for:

- Almost all features provided by MySQL Server up to and including MySQL Server version 8.0.
- Converting parameter values back and forth between Python and MySQL data types, for example Python datetime and MySQL DATETIME. You can turn automatic conversion on for convenience, or off for optimal performance.
- All MySQL extensions to standard SQL syntax.
- Protocol compression, which enables compressing the data stream between the client and server.
- Connections using TCP/IP sockets and on Unix using Unix sockets.
- Secure TCP/IP connections using SSL.

- Self-contained driver. Connector/Python does not require the MySQL client library or any Python modules outside the standard library.

RANDOM MODULE

The Python random module functions depend on a pseudo-random number generator function random(), which generates the float number between 0.0 and 1.0.

There are different types of functions used in a random module which is given below:

- **random.random()**

This function generates a random float number between 0.0 and 1.0. The function doesn't need any arguments.

- **random.randint()**

This function returns a random integer between the specified integers.

- **random.choice()**

This function returns a randomly selected element from a non-empty sequence. An empty sequence as argument raises an IndexError.

- **random.shuffle()**

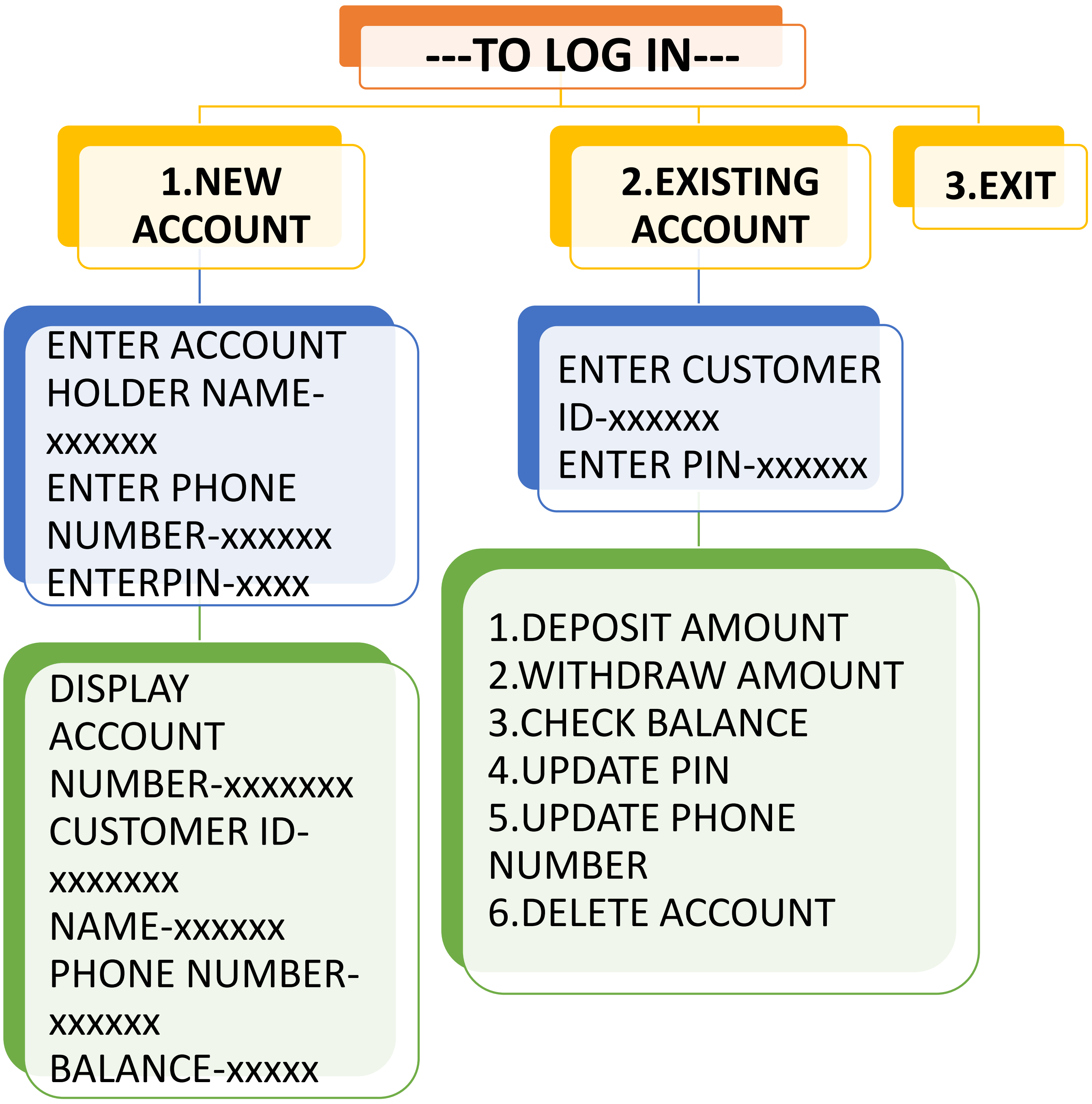
This function randomly reorders the elements in the list.

- **random.randrange(beg,end,step)**

This function is used to generate a number within the range specified in its argument. It accepts three arguments, beginning number, last number, and step, which is used to skip a number in the range. The value of start is 0 by default. Similarly, the value of step is 1 by default.

NOTE: CLASSES ARE ALSO USED TO MAKE THE PROGRAM MORE MANAGEABLE.

FLOW CHART


---TO LOG IN---
1.NEW ACCOUNT
2.EXISTING ACCOUNT
3.EXIT
ENTER ACCOUNT HOLDER NAME-xxxxxx
ENTER PHONE NUMBER-xxxxxx
ENTERPIN-xxxx
ENTER CUSTOMER ID-xxxxxx
ENTER PIN-xxxxxx
DISPLAY ACCOUNT NUMBER-xxxxxxx
CUSTOMER ID-xxxxxxx
NAME-xxxxxx
PHONE NUMBER-xxxxxx
BALANCE-xxxxx
1.DEPOSIT AMOUNT
2.WITHDRAW AMOUNT
3.CHECK BALANCE
4.UPDATE PIN
5.UPDATE PHONE NUMBER
6.DELETE ACCOUNT

PYTHON CODING

```
import mysql.connector as conn
from random import randint
myconn=conn.connect(host='localhost',user='root',passwd
='kriti',database='ATM')
if myconn.is_connected()==False:
    print('...........ERROR...........')
if myconn.is_connected()==True:
    print('...........CONNECTION SUCCESSFUL...........')

cursor=myconn.cursor()
cursor.execute('DROP TABLE IF EXISTS ATM')
cursor.execute('''CREATE TABLE ATM(ACCNO
CHAR(8) NOT NULL PRIMARY KEY,
                        ACC_NM VARCHAR(30) NOT NULL,
                        CUSTOMERID CHAR(5),
                        PHONENO CHAR(10),
                        BALANCE INT,
                       PIN INT)''')
myconn.commit()



# New account
def new_acc():
    name=input('ENTER ACCOUNT HOLDER\'S NAME
:  ')
    phno=int(input('ENTER ACCOUNT HOLDER\'S
PHONE NUMBER :  '))
    pin=int(input('ENTER PIN(4-DIGIT) : '))
    bal=int(input('ENTER OPENING BALANCE: '))
    accno=str(randint(10000000,99999999))
    cusid='SA'+str(accno[-4:-1])

    sql='INSERT INTO ATM
VALUES("{}","{}","{}","{}",{},{})'.format(accno,name,
cusid,phno,bal,pin)
    cursor.execute(sql)
    myconn.commit()
    sql='SELECT * FROM ATM WHERE
ACCNO={}'.format(accno,)
    try:
        cursor.execute(sql)
        record=cursor.fetchall()
        for r in record:
            for i in r:
                print(i)
    except:
        myconn.rollback()

print("=====================================
========================================
===")
```

```
#Existing account
def acc(ID,pin):
    sql='SELECT * FROM ATM WHERE
CUSTOMERID="{}" AND PIN={}'.format(ID,pin)
    try:
        cursor.execute(sql)
        rec=cursor.fetchall()
        if rec==[]:
            print('...........WRONG DATA ENTERED...........')
    except:
        myconn.rollback()

print("====================================
=======================================
===")


def deposit(ID):
    amt=int(input("Enter the money to be deposited: "))

print("====================================
=======================================
===")
    sql='update atm set balance=balance + {} where
CUSTOMERID="{}"'.format(amt,ID)
    cursor.execute(sql)
    myconn.commit()
    print("sucessfully deposited")



def withdraw(ID):
    amt=int(input("Enter the money to withdraw: "))

print("=====================================
=====================================
===")
    ah='select BALANCE from atm where
CUSTOMERID="{}"'.format(ID)
    cursor.execute(ah)
    m=cursor.fetchone()
    if amt<m[0]:
        sql='update atm set balance=balance - {} where
CUSTOMERID="{}"'.format(amt,ID)
        cursor.execute(sql)
        myconn.commit()
        print("Sucessfully updated")

    else:
        print("Your are having less than",amt)
        print("Please try again")

print("=====================================
================")
```

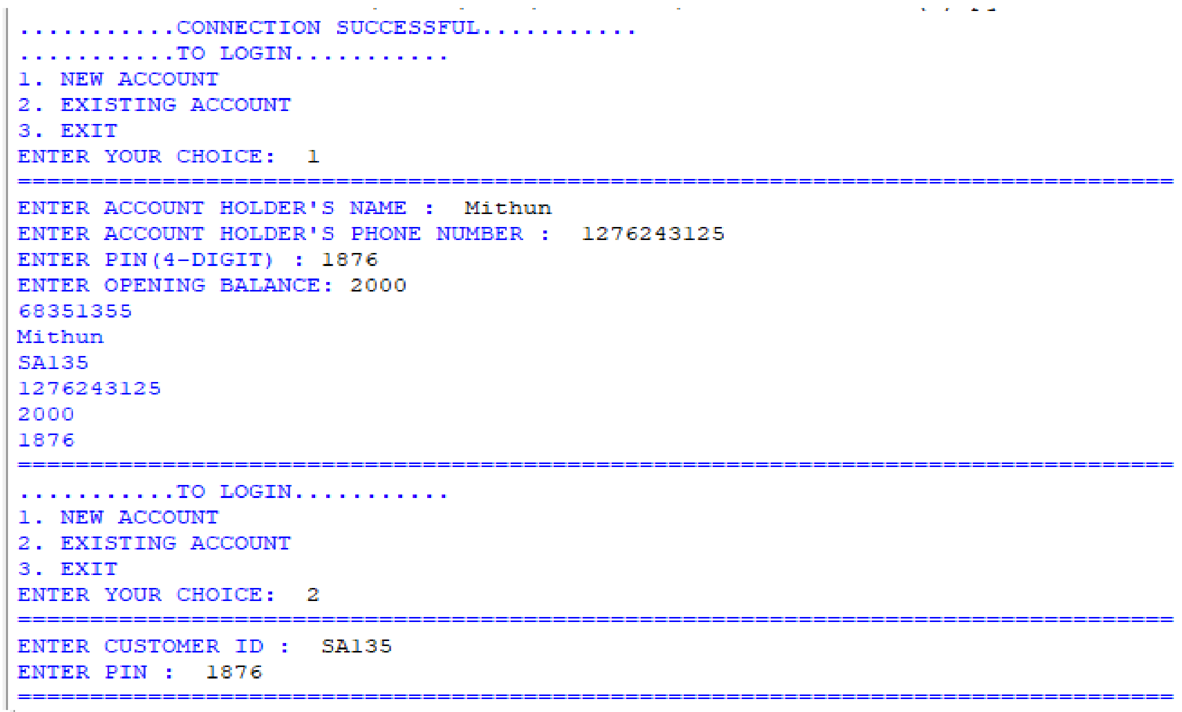
```
def balance(ID):
    ma='select balance from atm where
CUSTOMERID="{}"'.format(ID)
    cursor.execute(ma)
    k=cursor.fetchone()
    print("Balance in your account=",k[0])

print("=====================================
==========================================
===")


def upin(ID):
    pin=int(input("Enter the pin: "))

print("=====================================
==========================================
===")

    sql='update atm set pin={} where
CUSTOMERID="{}"'.format(pin,ID)
    cursor.execute(sql)
    myconn.commit()
    print("PIN updated")
```

```
def upn(ID):
    phn=int(input("Enter the new phone number: "))

print("======================================
=========================================
===")

    sql='update atm set phoneno={} where
CUSTOMERID="{}"'.format(phn,ID)
    cursor.execute(sql)
    myconn.commit()
    print("Phone number updated")


def delete(ID):

print("======================================
=========================================
===")

    sql='delete from atm where
CUSTOMERID="{}"'.format(ID)
    cursor.execute(sql)
    myconn.commit()
    print("Record deleted")
```

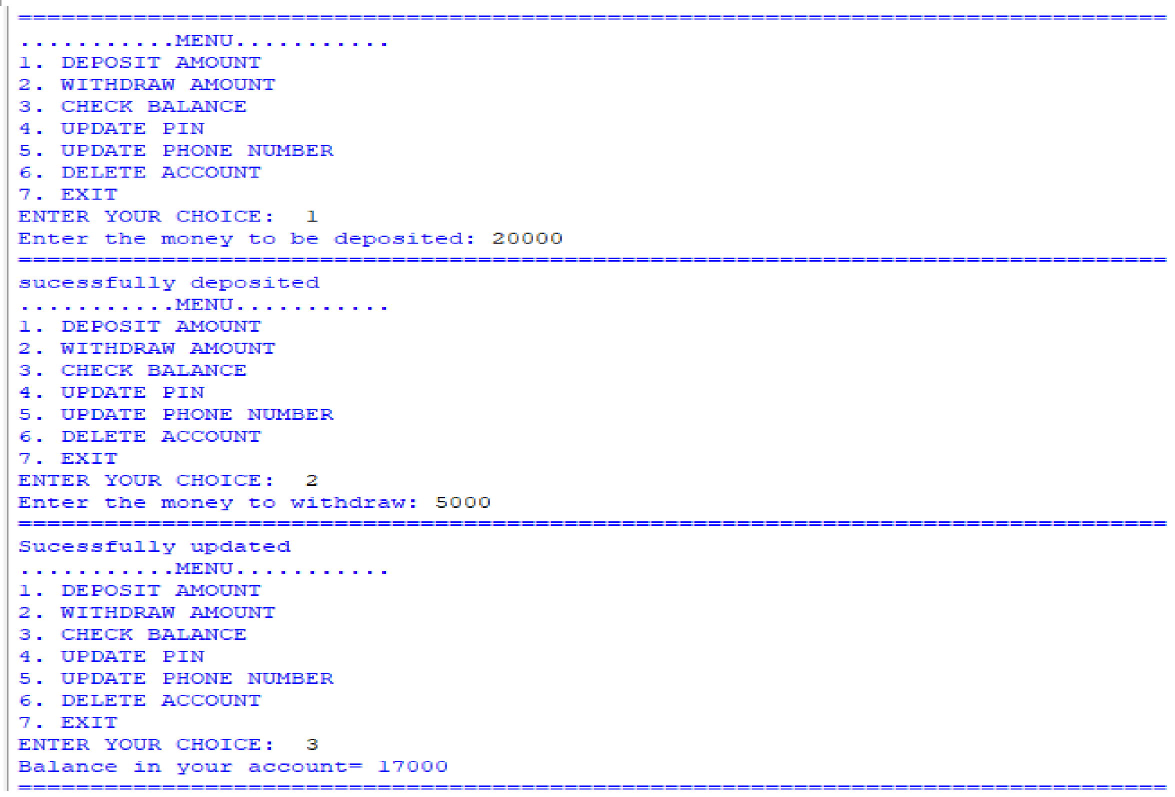
```
def menu(ID):
    while True:
        print('...........MENU...........')
        print('1. DEPOSIT AMOUNT')
        print('2. WITHDRAW AMOUNT')
        print('3. CHECK BALANCE')
        print('4. UPDATE PIN')
        print('5. UPDATE PHONE NUMBER')
        print('6. DELETE ACCOUNT')
        print('7. EXIT')
        ch=int(input('ENTER YOUR CHOICE:  '))
        if ch==1:
            deposit(ID)
        elif ch==2:
            withdraw(ID)
        elif ch==3:
            balance(ID)
        elif ch==4:
            upin(ID)
        elif ch==5:
            upn(ID)
        elif ch==6:
            delete(ID)
        else:
            break
```

```
def main():
    while True:
        print('...........TO LOGIN...........')
        print('1. NEW ACCOUNT')
        print('2. EXISTING ACCOUNT')
        print('3. EXIT')
        ch=int(input('ENTER YOUR CHOICE:  '))

print("======================================
=========================================
===")
        if ch==1:
            new_acc()
        elif ch==2:
            ID=input('ENTER CUSTOMER ID :  ')
            pin=int(input('ENTER PIN :  '))
            acc(ID,pin)
            menu(ID)
        else:
            break

main()
```

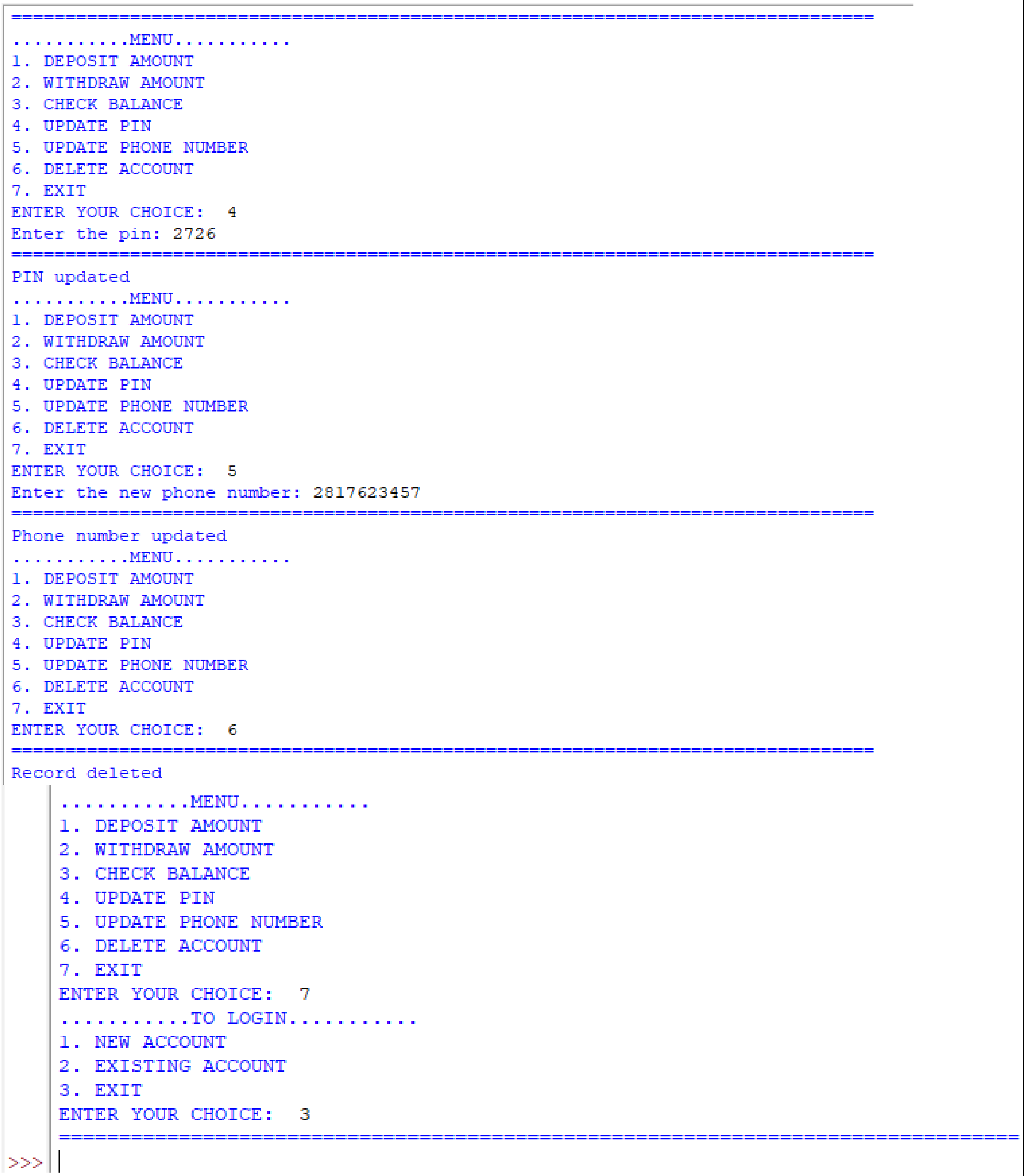
Output of the code



```
...........CONNECTION SUCCESSFUL..........
...........TO LOGIN..........
1. NEW ACCOUNT
2. EXISTING ACCOUNT
3. EXIT
ENTER YOUR CHOICE:  1
================================================================================
ENTER ACCOUNT HOLDER'S NAME :  Mithun
ENTER ACCOUNT HOLDER'S PHONE NUMBER :  1276243125
ENTER PIN(4-DIGIT) : 1876
ENTER OPENING BALANCE: 2000
68351355
Mithun
SA135
1276243125
2000
1876
================================================================================
...........TO LOGIN..........
1. NEW ACCOUNT
2. EXISTING ACCOUNT
3. EXIT
ENTER YOUR CHOICE:  2
================================================================================
ENTER CUSTOMER ID :  SA135
ENTER PIN :  1876
================================================================================
```

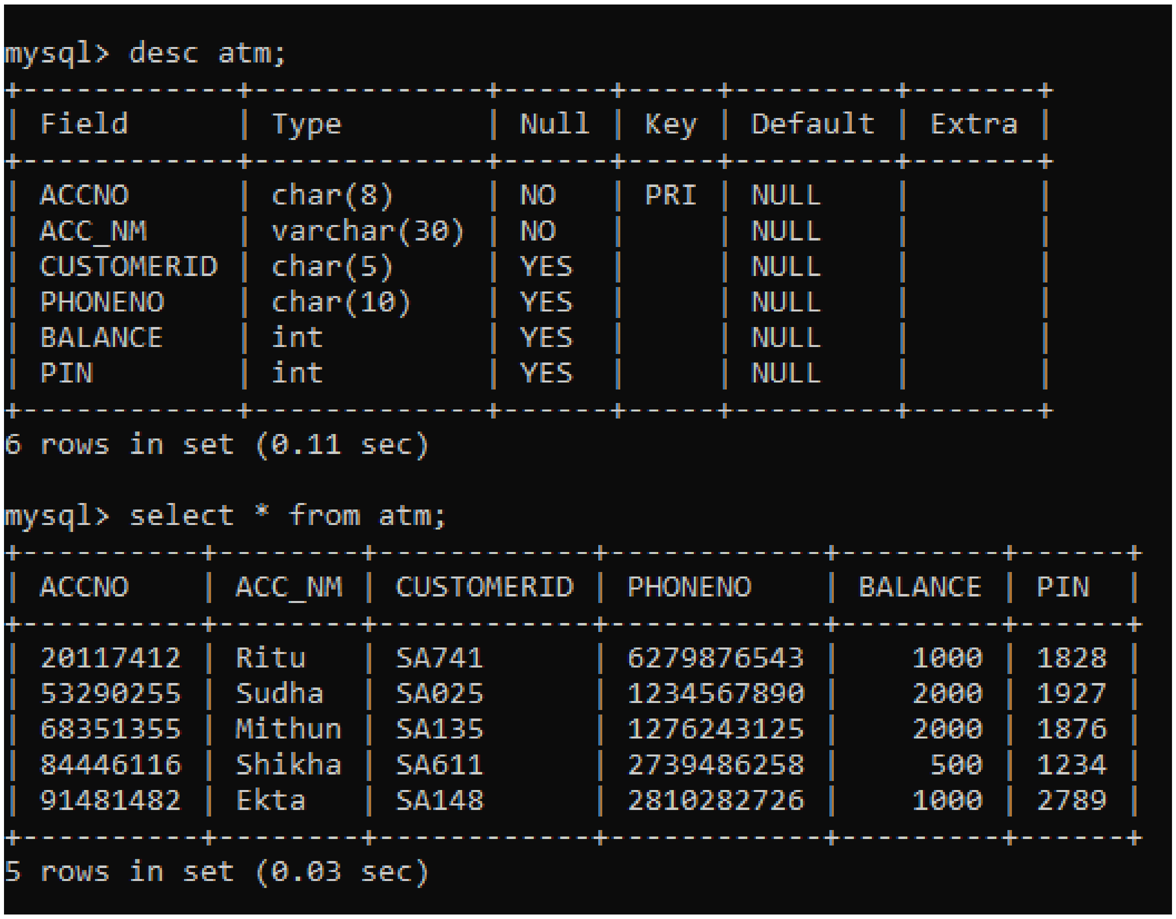


```
================================================================================
...........MENU..........
1. DEPOSIT AMOUNT
2. WITHDRAW AMOUNT
3. CHECK BALANCE
4. UPDATE PIN
5. UPDATE PHONE NUMBER
6. DELETE ACCOUNT
7. EXIT
ENTER YOUR CHOICE:  1
Enter the money to be deposited: 20000
================================================================================
sucessfully deposited
...........MENU..........
1. DEPOSIT AMOUNT
2. WITHDRAW AMOUNT
3. CHECK BALANCE
4. UPDATE PIN
5. UPDATE PHONE NUMBER
6. DELETE ACCOUNT
7. EXIT
ENTER YOUR CHOICE:  2
Enter the money to withdraw: 5000
================================================================================
Sucessfully updated
...........MENU..........
1. DEPOSIT AMOUNT
2. WITHDRAW AMOUNT
3. CHECK BALANCE
4. UPDATE PIN
5. UPDATE PHONE NUMBER
6. DELETE ACCOUNT
7. EXIT
ENTER YOUR CHOICE:  3
Balance in your account= 17000
================================================================================
```



```
================================================================================
..........MENU..........
1. DEPOSIT AMOUNT
2. WITHDRAW AMOUNT
3. CHECK BALANCE
4. UPDATE PIN
5. UPDATE PHONE NUMBER
6. DELETE ACCOUNT
7. EXIT
ENTER YOUR CHOICE:  4
Enter the pin: 2726
================================================================================
PIN updated
..........MENU..........
1. DEPOSIT AMOUNT
2. WITHDRAW AMOUNT
3. CHECK BALANCE
4. UPDATE PIN
5. UPDATE PHONE NUMBER
6. DELETE ACCOUNT
7. EXIT
ENTER YOUR CHOICE:  5
Enter the new phone number: 2817623457
================================================================================
Phone number updated
..........MENU..........
1. DEPOSIT AMOUNT
2. WITHDRAW AMOUNT
3. CHECK BALANCE
4. UPDATE PIN
5. UPDATE PHONE NUMBER
6. DELETE ACCOUNT
7. EXIT
ENTER YOUR CHOICE:  6
================================================================================
Record deleted
..........MENU..........
1. DEPOSIT AMOUNT
2. WITHDRAW AMOUNT
3. CHECK BALANCE
4. UPDATE PIN
5. UPDATE PHONE NUMBER
6. DELETE ACCOUNT
7. EXIT
ENTER YOUR CHOICE:  7
..........TO LOGIN..........
1. NEW ACCOUNT
2. EXISTING ACCOUNT
3. EXIT
ENTER YOUR CHOICE:  3
================================================================================
>>>
```

Table Structure



```
mysql> desc atm;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| ACCNO      | char(8)     | NO   | PRI | NULL    |       |
| ACC_NM     | varchar(30) | NO   |     | NULL    |       |
| CUSTOMERID | char(5)     | YES  |     | NULL    |       |
| PHONENO    | char(10)    | YES  |     | NULL    |       |
| BALANCE    | int         | YES  |     | NULL    |       |
| PIN        | int         | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
6 rows in set (0.11 sec)

mysql> select * from atm;
+----------+--------+------------+------------+---------+------+
| ACCNO    | ACC_NM | CUSTOMERID | PHONENO    | BALANCE | PIN  |
+----------+--------+------------+------------+---------+------+
| 20117412 | Ritu   | SA741      | 6279876543 |    1000 | 1828 |
| 53290255 | Sudha  | SA025      | 1234567890 |    2000 | 1927 |
| 68351355 | Mithun | SA135      | 1276243125 |    2000 | 1876 |
| 84446116 | Shikha | SA611      | 2739486258 |     500 | 1234 |
| 91481482 | Ekta   | SA148      | 2810282726 |    1000 | 2789 |
+----------+--------+------------+------------+---------+------+
5 rows in set (0.03 sec)
```

BIBLIOGRAPHY

- https://drive.google.com/drive/folders/1c4jl86t0lDbFDGofpz6pFOff9a3Fh9kJ
- COMPUTER SCIENCE WITH PYTHON BY PREETI ARORA (2021-EDITION)
- https://www.javatpoint.com
- https://www.geeksforgeeks.org
- https://www.w3resource.com