ATM SYSTEM

MANAGEMENT PROJECT

PROGRAMMING LANGUAGE USE: PYTHON

PRESENTED BY:

1.SHREYA LADDHA

2.AJINKYA BHOSLE

3.TANISHQ KHANDELWAL

ABSTRACT

The ATM system is the project which is use to access their bank account in order to make cash withdrawals whenever the user need to make the cash withdrawals, they can enter their PIN(personal identification number) and it will used to display the amount to be withdrawn in the form of 100's ,500's,2000's .Oncer their withdrawn was successful, he amount will be debited in their account.

The ATM system is develop in python. Python is one of the powerful object oriented programming language hence we used this language to develop our project.

The ATM will service one customer at a time a customer will required to enter the ATM card number ,PIN both of which will be send to the database for validation as part of each transaction. The customer will then able to perform one or more transaction . Also the customer must will able to make a balance enquiry of any account linked to the card.

The ATM will communicate each transaction to the database and obtain verification that is allowed by the database. In the case of the cash withdrawal a second message will be send after the transaction have been physically completed if the data base determine the customer's PIN is invalid the customer will be require to re-enter the pin before transaction can complete if a transaction failed for any reason other than any invalid PIN the ATM will display an explanation of the problem and will then ask the customer whether he or she want to do another transaction.

Chapter 1: *Introduction of the Project*

This project is a basic simulation of an Automated Teller Machine (ATM) machine. It uses the pyttsx3 library to produce audio output and the speech_recognition library to recognize input. The program starts by checking the current hour and greets the user with a Good morning, Good afternoon, or Good evening message, depending on the time. Then, the user is prompted to enter their ATM pin and if the pin is correct, the user is presented with various options, such as checking their balance, withdrawing funds, depositing funds, or exiting the program. The balance of the user's account is kept track of as the user performs different transactions.

The given ATM management project has features :

1.voice recognition

2.Hiding password system

3.It shows updated balance after every transaction

Language support: The system can be made multilingual to support users speaking different languages. This can be achieved by using a translation library or by training the language model in multiple languages.

Enhanced security: To make the system more secure, additional security measures can be added. For example, biometric authentication, such as fingerprint scanning, can be added to verify the user's identity.

Additional transactions: The system can be extended to support additional transactions, such as paying bills, checking account statements, and more.

Integration with the bank's database: The system can be integrated with the bank's database to retrieve up-to-date account information and allow real-time transactions. This will also eliminate the need for storing the account information locally

Existing ATM System:

Basic Functionality: The existing ATM system is designed to perform basic functions such as cash withdrawals, balance inquiry, and account transfers.

Limited User Interfaces: The user interface of existing ATMs is often limited, with only a few options available for transactions.

Cash Availability: Cash availability in existing ATMs is limited and may not always be updated in real-time.

Security Concerns: Existing ATMs are vulnerable to security threats such as skimming, card cloning, and hacking.

Proposed ATM System:

Advanced Functionality: The proposed ATM system is designed to offer advanced functions such as bill payments, mobile top-ups, and loan applications

User-Friendly Interfaces: The proposed ATM system has a user-friendly interface, with clear options and instructions for all transactions.

Real-Time Cash Availability: The proposed system have real-time updates on cash availability, ensuring that customers can access cash when they need it.

Enhanced Security: The proposed ATM system will have advanced security features such as encryption, and firewalls to protect against security threats

The following python libraries are used in this project:

1.time

2.datetime

3.pyttsx3

4.os

5.getpass

6.speech_recognition as sr

The installation of this libraries is discussed in detail in the methodology

Chapter 2:Methodology

The given program is a simple simulation of an Automated Teller Machine (ATM). The program uses the pyttsx3 library to provide speech output, and the datetime module to get the current time and greet the user accordingly. The speech recognition feature is not implemented in this program.

When the program runs, it first greets the user based on the time of the day. It then waits for 5 seconds before prompting the user to enter their ATM pin. The user's account balance is set to 5000, and the correct pin is set to 1234. If the user enters the correct pin, the program enters a loop that allows the user to perform various transactions such as checking the balance, withdrawing money, depositing money, and exiting the program.

The user is given a menu of options, which are represented by numbers 1 to 4. The user is asked to enter their choice, and the program performs the corresponding transaction based on the user's input. If the user enters an incorrect pin, the program outputs "Wrong pin. Please try again." and the loop terminates.

Explanation of python libraries used:

*1.**time**: This library provides various functions to work with time and dates. In the code, the sleep function is used to pause the execution of the program for a certain amount of time.*

Installation of time library:

The `time` library is a built-in library in Python, so you don't have to install it. You can use it simply by importing it in your code.

*2.**datetime**: This library provides classes for working with dates and times. In the code, the datetime.datetime.now().hour function is used to get the current hour from the system.*

Installation of datetime library:

The `datetime` module is a built-in library in Python, and you can import it in your code using the following statement

syntax: import datetime

*3.**pyttsx3**: This library is used for text-to-speech conversion. It is used to produce spoken outputs for the user.*

Installation of pyttsx3 library:

`pyttsx3` is a third-party library, so you'll need to install it before you can import it in your code. You can install it using `pip`, the Python package manager, by running the following command in your terminal or command prompt

code for installation is as follows:

```
pip install pyttsx3
syntax: import pyttsx3
```

4.os: This library provides a way of using operating system dependent functionality like reading or writing to the file system. In the code, it is not used.

Installation of os library:

The `os` module is a built-in library in Python, and you can import it in your code using the following statement

Syntax: import os

*5.**getpass**: This library provides a secure way of getting password inputs from the user without echoing it to the screen. In the code, it is used to get the ATM pin from the user.*

Installation of getpass library:

The `getpass` module is a built-in library in Python, and you can import it in your code using the following statement

Syntax: import getpass

*6.**speech_recognition**: This library provides easy access to various speech recognition APIs such as Google Speech Recognition, Apple Speech Recognition, etc. In the code, it is imported as sr, but it is not used.*

Installation of speech_recognition library:

`SpeechRecognition` is a third-party library for performing speech recognition in Python, and you'll need to install it before you can import it in your code. You can install it using `pip`, the Python package manager, by running the following command in your terminal or command prompt

code for installation is as follows:

pip install SpeechRecognition

syntax*: import speech_recognition as sr*

Chapter 3:Code for the ATM management project

```
import time
import datetime
import pyttsx3
import os
import getpass
import speech_recognition as sr

engine = pyttsx3.init()
engine.setProperty("rate", 160)
engine.say("Hi Sir")
engine.runAndWait()

def speak(audio):
    engine.say(audio)
    engine.runAndWait()
```

```
if _name_ == "_main_":
    hour = int(datetime.datetime.now().hour)
    if hour < 12:
        speak("Good morning. Please insert your card.")
    elif hour < 18:
        speak("Good afternoon. Please insert your card.")
    else:
        speak("Good evening. Please insert your card.")

    #for card processing
    time.sleep(3)

    password = 1234
    attempts = 3

    while attempts > 0:
        #taking atm pin from user
        speak("Please enter your ATM pin:")
        pin = getpass.getpass(prompt="Enter your ATM pin: ")

        #checking pin is valid or not
        if pin == str(password):
            #user account balance
            balance = 5000

            #loop will run until user gets free
            while True:
```

```
#Showing  info to user
speak("""
   1 = balance
   2 = withdraw balance
   3 = deposit balance
   4 = exit
   """)

try:
   #taking an option from user
   option = int(input("Please enter your choice: "))
except:
   speak("Please enter a valid option")

#for option 1
if option == 1:
   speak(f"Your current balance is {balance}")

if option == 2:
   speak("Please enter the amount you want to withdraw:")
   withdraw_amount = int(input("Please enter withdraw_amount: "))
   if balance >= withdraw_amount:
     balance = balance - withdraw_amount
     speak(f"{withdraw_amount} is debited from your account:")
     speak(f"Your updated balance is {balance}.")
   else:
     speak("Insufficient balance.")
```

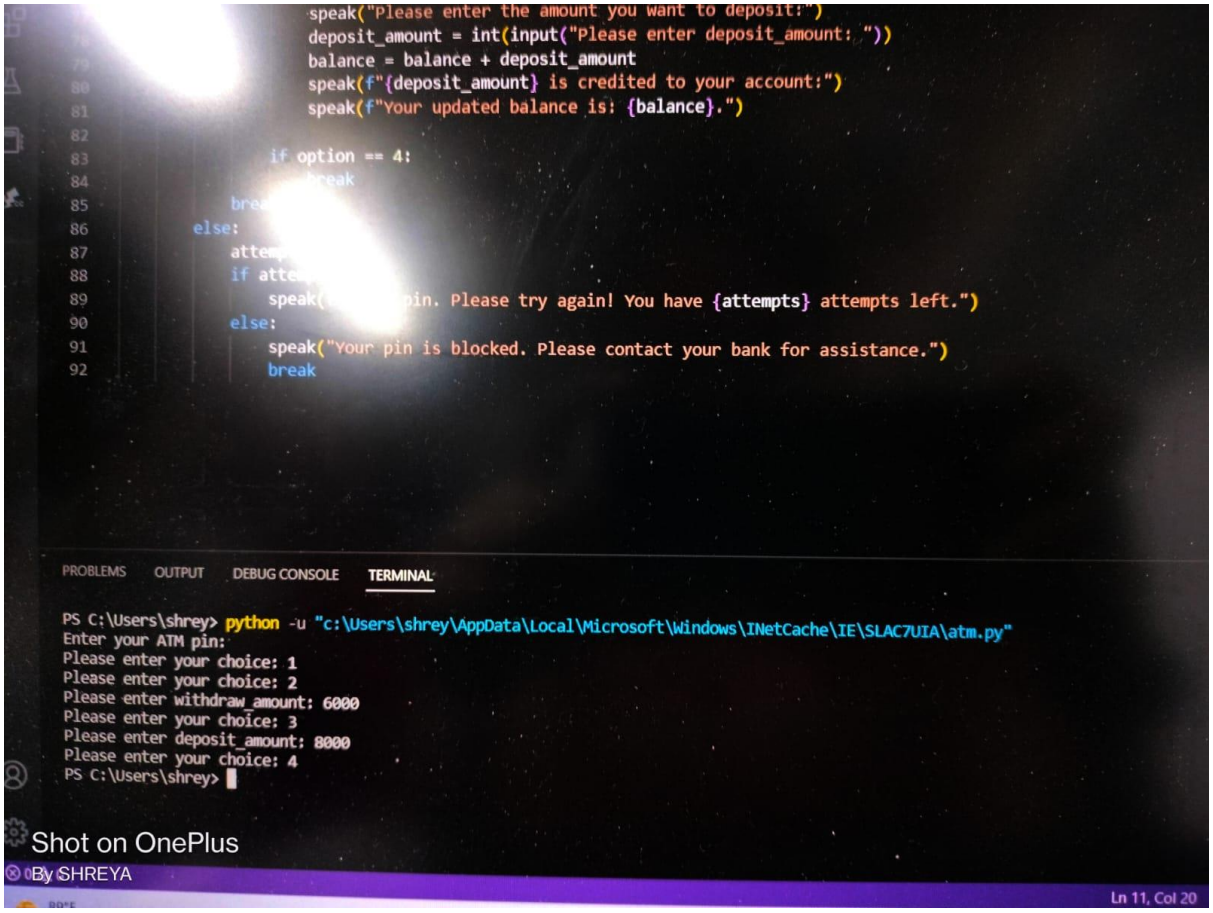
```
            if option == 3:
                speak("Please enter the amount you want to deposit:")
                deposit_amount = int(input("Please enter deposit_amount: "))
                balance = balance + deposit_amount
                speak(f"{deposit_amount} is credited to your account:")
                speak(f"Your updated balance is: {balance}.")


            if option == 4:
                break
        break
    else:
        attempts -= 1
        if attempts > 0:
            speak(f"Wrong pin. Please try again! You have {attempts} attempts
left.")
        else:
            speak("Your pin is blocked. Please contact your bank for assistance.")
            break
```

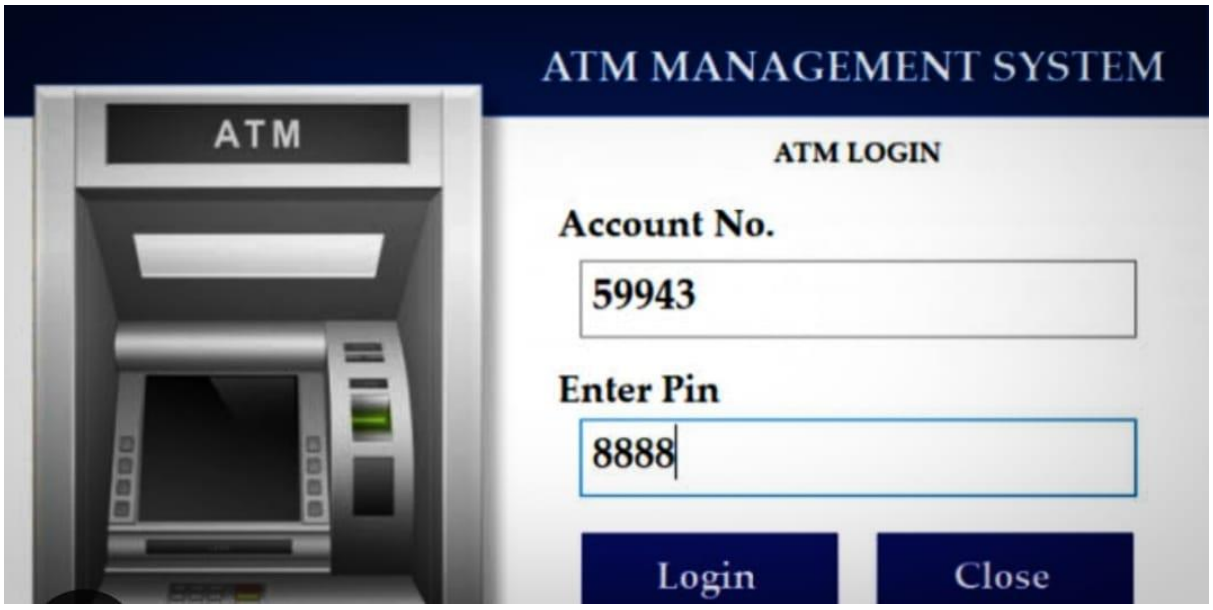
Chapter 4:Result and Discussion

Following are the result and images of output and ATM Management System:

WhatsApp Video 2023-02-10 at 18.13.10.mp4


speak("Please enter the amount you want to deposit:")
deposit_amount = int(input("Please enter deposit_amount: "))
balance = balance + deposit_amount
speak(f"{deposit_amount} is credited to your account:")
speak(f"Your updated balance is: {balance}.")
if option == 4:
break
else:
if atte
speak(pin. Please try again! You have {attempts} attempts left.")
else:
speak("Your pin is blocked. Please contact your bank for assistance.")
break
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PS C:\Users\shrey> python -u "c:\Users\shrey\AppData\Local\Microsoft\Windows\INetCache\IE\SLAC7UIA\atm.py"
Enter your ATM pin:
Please enter your choice: 1
Please enter your choice: 2
Please enter withdraw_amount: 6000
Please enter your choice: 3
Please enter deposit_amount: 8000
Please enter your choice: 4
PS C:\Users\shrey>
Shot on OnePlus
By SHREYA
Ln 11, Col 20


CARD


ATM MANAGEMENT SYSTEM
ATM
ATM LOGIN
Account No.
59943
Enter Pin
8888
Login
Close

Code explanation:

This code implements a basic ATM machine in Python. When the code is executed, it greets the user with a "Good morning/afternoon/evening" message and then requests the user to insert their ATM card. After a 5 second sleep, the code prompts the user to enter their ATM pin. The code has a fixed password, which is 1234. If the entered pin is correct, the code presents the user with a menu of options to check the balance, withdraw money, deposit money, or exit.

When the user selects an option, the code performs the corresponding action, such as updating the balance after a withdrawal or deposit. If the user enters an invalid option, the code prompts them to enter a valid option. If the user enters an incorrect pin, the code informs them that the pin is wrong and asks them to try again.

Overall, the code provides a basic functionality of an ATM machine. However, it can be improved in many ways. For example, it can be made more secure by implementing multiple levels of authentication, such as password and a fingerprint scan. It can also be made more user-friendly by adding error handling for invalid inputs, such as a non-integer input for the pin or the amount. Additionally, the code can be made more efficient by implementing a database to store user information, instead of hard-coding the values in the code.

CHAPTER 5: CONCLUSION

The Above code is a basic ATM management system in python. It uses text-to-speech conversion to communicate with the user and get their inputs through the terminal. The system supports basic operations such as checking the balance, withdrawing money, depositing money, and exiting the system.

However, it is a very basic implementation and there is room for improvement. For example, it does not have any security measures in place for protecting the user's data. Additionally, it does not have any error handling mechanism for invalid inputs, and the design could be made more user-friendly.

In conclusion, the code provides a basic foundation for an ATM management system but it should be improved to make it more robust and secure.

CHAPTER 6 : Future Scope

The future scope of the ***ATM MANAGEMENT SYSTEM*** can be enhanced in the following ways:

User authentication: The code should implement a proper user authentication mechanism such as username-password login to secure the user's data.

Error handling: The code should implement error handling mechanisms to handle invalid inputs and exceptions that may occur during execution.

User interface: The code can be made more user-friendly by implementing a graphical user interface (GUI) using libraries such as PyQt or tkinter.

Speech recognition: The code can be enhanced to use speech recognition APIs to enable voice-based interaction with the user.

Data storage: The code should implement a mechanism for storing the user's data securely, for example by using a database management system.

Additional functionalities: More functionalities can be added to the system, such as the ability to transfer funds between accounts, view transaction history, change the pin, etc.

In conclusion, the code provides a basic foundation for an ATM management system, but there is a lot of scope for improvement to make it more robust, secure, and user-friendly

<<<<<END>>>>>