

OCTANET

Automated teller machine:

Background: -

The Automated Teller Machine (ATM) is a popular banking system utilized by customers to conduct a range of transactions independently, eliminating the requirement for direct assistance from a human teller. Since their inception in the 1960s, ATMs have undergone significant advancements and have become a fundamental component of contemporary banking. These advancements encompass cutting-edge capabilities like wireless connectivity, biometric authentication, and various other features aimed at enhancing the user experience and streamlining banking operations.

Working: -

The functioning of the ATM relies on a microcontroller that oversees the operation of multiple components, including the card reader, keypad, and LCD display. To commence a transaction, the customer inserts their ATM card into the card reader, where the card's data is read and cross-referenced with the information stored in the system's database. If the details match, the customer is prompted to enter their PIN using the keypad. After verifying the PIN, the customer gains entry to the main menu of the system. From there, they can engage in different transactions like depositing funds, withdrawing cash, and checking their account balance.

Code: -

```
from datetime import date, datetime, timedelta

import threading

import time

today = date.today()

lock = threading.Lock()

class CardHolder:

    def __init__(self, cardNumber, pin, firstName, lastName, balance):

        self.cardNumber = cardNumber

        self.pin = pin

        self.firstName = firstName

        self.lastName = lastName

        self.balance = balance

        self.transactions = []
```

```
def get_cardNumber(self):
```

```
    return self.cardNumber
```

```
def get_pin(self):
```

```
    return self.pin
```

```
def get_firstName(self):
```

```
    return self.firstName
```

```
def get_lastName(self):
```

```
    return self.lastName
```

```
def get_balance(self):
```

```
    return self.balance
```

```
def get_transactions(self):
```

```
    return self.transactions
```

```
def set_cardNumber(self, newVal):
```

```
    self.cardNumber = newVal
```

```
def set_pin(self, newVal):
```

```
self.pin = newVal
```

```
def set_firstName(self, newVal):
```

```
    self.firstName = newVal
```

```
def set_lastName(self, newVal):
```

```
    self.lastName = newVal
```

```
def set_balance(self, newVal):
```

```
    self.balance = newVal
```

```
def add_transaction(self, transaction):
```

```
    self.transactions.append(transaction)
```

```
    if len(self.transactions) > 5:
```

```
        self.transactions = self.transactions[-5:]
```

```
def print_out(self):
```

```
    print("Card Number: ", self.cardNumber)
```

```
    print("PIN: ", self.pin)
```

```
    print("First Name: ", self.firstName)
```

```
    print("Last Name: ", self.lastName)
```

```
    print("Balance: ", self.balance)
```

```
def print_boxed(text):
```

```
    lines = text.split('\n')
```

```
    max_length = max(len(line) for line in lines)
```

```
print('┌' + '=' * (max_length + 2) + '┐')
```

```
for line in lines:
```

```
    print('│' + line.ljust(max_length) + '│')
```

```
print('└' + '=' * (max_length + 2) + '┘')
```

```
def print_menu():
```

```
    menu_text = ""
```

```
    THE UBL BANK MENU
```

```
    Please choose one of the following options:
```

1. Deposit
2. Withdraw
3. Fund Transfer
4. Inquire Balance
5. Mini Statement
6. Account Details
7. Change PIN
8. Pay Bill
9. Exit

```
    print_boxed(menu_text)
```

```
def deposit(cardHolder):
```

```
    try:
```

```
        print_boxed("Enter the amount to deposit (in Rps): ")
```

```
        deposit_amount = float(input().strip())
```

```
        if deposit_amount <= 0:
```

```
            print_boxed("Invalid input. Amount must be greater than zero.")
```

```
            return
```

```
        cardHolder.set_balance(cardHolder.get_balance() + deposit_amount)
```

```
        cardHolder.add_transaction(f"Deposit: +{deposit_amount} Rps on {today}")
```

```
        result_text = ""
```

```
    Your new balance is: {cardHolder.get_balance()}
```

```
    Date: {today}
```

```
print_boxed(result_text)

except ValueError:

    print_boxed("Invalid input. Amount must be a number.")
```

```
def withdraw(cardHolder):
```

```
    try:

        print_boxed("Enter the amount to withdraw (in Rps): ")

        withdraw_amount = float(input().strip())

        if withdraw_amount <= 0:

            print_boxed("Invalid input. Amount must be greater than zero.")

            return

        if cardHolder.get_balance() < withdraw_amount:

            print_boxed("Sorry, insufficient balance")

        else:

            cardHolder.set_balance(cardHolder.get_balance() - withdraw_amount)

            cardHolder.add_transaction(f"Withdraw: -{withdraw_amount} Rps on {today}")

            result_text = f"""\
```

```
Successfully withdrawn
```

```
Your current balance is: {cardHolder.get_balance()}"""
```

```
print_boxed(result_text)

except ValueError:

    print_boxed("Invalid input. Amount must be a number.")
```

```
def funds_transfer(cardHolder):
```

```
    recipient_name_text = "Enter the name or debit number of the recipient: "
```

```

print_boxed(recipient_name_text)

recipient_name = input().strip()

if not recipient_name:
    print_boxed("Recipient name or phone number cannot be empty.")
    return

try:
    print_boxed("Enter the amount to send (in Rps): ")
    transfer_amount = float(input().strip())

    if transfer_amount <= 0:
        print_boxed("Invalid input. Amount must be greater than zero.")
        return

    if cardHolder.get_balance() < transfer_amount:
        print_boxed("Sorry, insufficient balance")
    else:
        cardHolder.set_balance(cardHolder.get_balance() - transfer_amount)
        cardHolder.add_transaction(f"Funds Transfer: -{transfer_amount} Rps to {recipient_name} on {today}")

        result_text = f"""\
Name: {recipient_name}
Amount sent: {transfer_amount}
Status: Successfully sent

Date: {today}
Amount remaining: {cardHolder.get_balance()}"""

        print_boxed(result_text)

except ValueError:
    print_boxed("Invalid input. Amount must be a number.")

def balance_inquiry(cardHolder):

```

```
balance_text = f"""\n
Name: {cardHolder.get_firstName()} {cardHolder.get_lastName()}
Your current balance in the account is: {cardHolder.get_balance()}
Date: {today}"""\n
print_boxed(balance_text)
```

```
def mini_statement(cardHolder):
    statement_text = f"""\n
Name: {cardHolder.get_firstName()} {cardHolder.get_lastName()}
Mini Statement: """\n
    transactions = cardHolder.get_transactions()
    if transactions:
        for i, transaction in enumerate(transactions[::-1], start=1):
            statement_text += f"\nTransaction {i}: {transaction}"
    else:
        statement_text += "\nNo transactions yet."
```

```
print_boxed(statement_text)
```

```
def account_details(cardHolder):
    details_text = f"""\n
Debit Card No: {cardHolder.get_cardNumber()}
Name: {cardHolder.get_firstName()} {cardHolder.get_lastName()}
Total Amount: {cardHolder.get_balance()}
Date: {today}"""\n
```



```
print_boxed(details_text)
```

```
def change_pin(cardHolder):
```

```
    pin_text = "Enter your new PIN: "
```

```
    print_boxed(pin_text)
```

```
    new_pin = input().strip()
```

```
    if len(new_pin) != 4 or not new_pin.isdigit():
```

```
        print_boxed("Invalid PIN. Please enter a 4-digit number.")
```

```
        return
```

```
    cardHolder.set_pin(new_pin)
```

```
    print_boxed("PIN successfully changed!")
```

```
def pay_bill(cardHolder):
```

```
    try:
```

```
        print_boxed("Enter the bill amount to pay (in Rps): ")
```

```
        bill_amount = float(input().strip())
```

```
        if bill_amount <= 0:
```

```
            print_boxed("Invalid input. Amount must be greater than zero.")
```

```
            return
```

```
        if cardHolder.get_balance() < bill_amount:
```

```
            print_boxed("Sorry, insufficient balance")
```

```
        else:
```

```
            cardHolder.set_balance(cardHolder.get_balance() - bill_amount)
```

```
            cardHolder.add_transaction(f"Bill Payment: -{bill_amount} Rps on {today}")
```

```
            result_text = f"""\n
```

Bill payment successful

Bill amount: {bill_amount}

Date: {today}

Amount remaining: {cardHolder.get_balance()}""

print_boxed(result_text)

except ValueError:

print_boxed("Invalid input. Amount must be a number.")

def validate_card_number(card_number):

return len(card_number) == 11 and card_number.isdigit()

def validate_pin(pin):

return len(pin) == 4 and pin.isdigit()

if __name__ == "__main__":

current_user = None

list_of_cardHolders = [

CardHolder("02134221017", "2160", "Muhammad", "Faiz Tanveer", 5000),

CardHolder("02134221057", "2161", "Zainab", "Rauf", 10000),

CardHolder("02134221101", "2162", "Hania", "Khan", 600),

CardHolder("02134221077", "2163", "Qasim", "Hassan", 1200)

]

print_boxed("""

WELCOME TO THE UBL BANK

Please insert your card...

Please wait while your card is being processed!

```
""")
```

```
time.sleep(2)
```

```
while True:
```

```
    card_number_text = "Please enter your debit card number: "
```

```
    print_boxed(card_number_text)
```

```
    card_number = input().strip()
```

```
    pin1_text = "Please enter your PIN: "
```

```
    print_boxed(pin1_text)
```

```
    pin1 = input().strip()
```

```
    if not validate_card_number(card_number) or not validate_pin(pin1):
```

```
        print_boxed("Invalid card number or PIN format. Please try again.")
```

```
        continue
```

```
    for cardHolder in list_of_cardHolders:
```

```
        if cardHolder.get_cardNumber() == card_number and cardHolder.get_pin() == pin1:
```

```
            current_user = cardHolder
```

```
            break
```

```
    if current_user is None:
```

```
        print_boxed("Card number or PIN not recognized. Please try again.")
```

```
    else:
```

```
        welcome_text = f"""
```

```
*****
```

```
Welcome, {current_user.get_firstName()} {current_user.get_lastName()}!
```

```
*****
```

```
'''
```

```
    print_boxed(welcome_text)
```

```
    break
```

```
while True:
```

```
    print_menu()
```

```
    option = input("Enter your choice (1-9): ").strip()
```

```
    if option == "1":
```

```
        deposit(current_user)
```

```
    elif option == "2":
```

```
        withdraw(current_user)
```

```
    elif option == "3":
```

```
        funds_transfer(current_user)
```

```
    elif option == "4":
```

```
        balance_inquiry(current_user)
```

```
    elif option == "5":
```

```
        mini_statement(current_user)
```

```
    elif option == "6":
```

```
        account_details(current_user)
```

```
    elif option == "7":
```

```
        change_pin(current_user)
```

```
    elif option == "8":
```

```
        pay_bill(current_user)
```

```
    elif option == "9":
```

```
        print_boxed(''''
```

```
*****
```

```
Thank you for using THE UBL BANK!
```

```
*****
```

```
""")
```

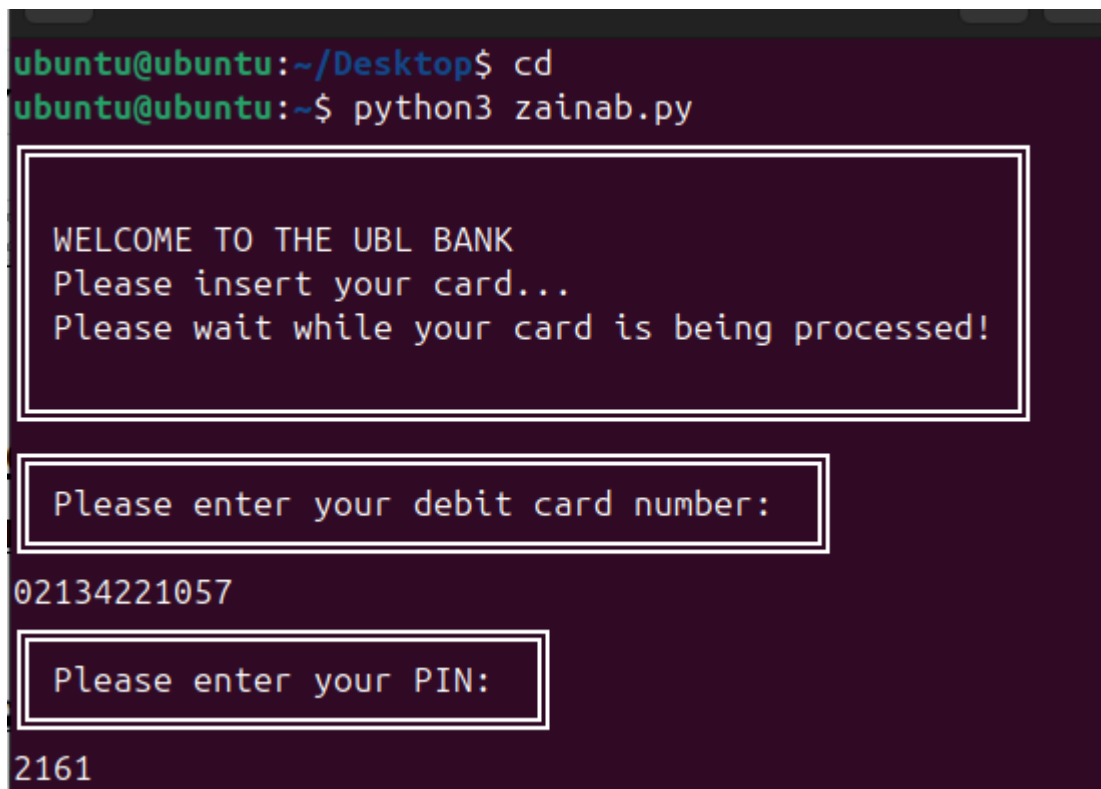
```
break
```

```
else:
```

```
print_boxed("Invalid option. Please choose a valid option (1-9).")
```

Code Output (ATM) :-

CHECKING DEBIT CARD NO AND PIN:



```
ubuntu@ubuntu:~/Desktop$ cd
ubuntu@ubuntu:~$ python3 zainab.py

WELCOME TO THE UBL BANK
Please insert your card...
Please wait while your card is being processed!

Please enter your debit card number:
02134221057

Please enter your PIN:
2161
```

WRONG CARD/PIN:

```
ubuntu@ubuntu:~/Desktop$ cd
ubuntu@ubuntu:~$ python3 zainab.py

WELCOME TO THE UBL BANK
Please insert your card...
Please wait while your card is being processed!

Please enter your debit card number:
36287387263

Please enter your PIN:
2722

Card number or PIN not recognized. Please try again.

Please enter your debit card number:
```

MAIN MENU:

```
*****
Welcome, Zainab Rauf!
*****

THE UBL BANK MENU
Please choose one of the following options:
1. Deposit
2. Withdraw
3. Fund Transfer
4. Inquire Balance
5. Mini Statement
6. Account Details
7. Change PIN
8. Pay Bill
9. Exit

Enter your choice (1-9): █
```

DEPOSIT AMOUNT:

```
Enter your choice (1-9): 1

Enter the amount to deposit (in Rps):

2000

Your new balance is: 12000.0
Date: 2024-06-12

THE UBL BANK MENU
Please choose one of the following options:
1. Deposit
2. Withdraw
3. Fund Transfer
4. Inquire Balance
5. Mini Statement
6. Account Details
7. Change PIN
8. Pay Bill
9. Exit

Enter your choice (1-9):
```

WITHDRAW AMOUNT:

```
Enter your choice (1-9): 2

Enter the amount to withdraw (in Rps):

500

Successfully withdrawn
Your current balance is: 11500.0

THE UBL BANK MENU
Please choose one of the following options:
1. Deposit
2. Withdraw
3. Fund Transfer
4. Inquire Balance
5. Mini Statement
6. Account Details
7. Change PIN
8. Pay Bill
9. Exit

Enter your choice (1-9):
```

INSUFFICIENT BALANCE, CAN'T WITHDRAW:

Enter your choice (1-9): 2

Enter the amount to withdraw (in Rps):

1000000000000

Sorry, insufficient balance

THE UBL BANK MENU
Please choose one of the following options:

1. Deposit
2. Withdraw
3. Fund Transfer
4. Inquire Balance
5. Mini Statement
6. Account Details
7. Change PIN
8. Pay Bill
9. Exit

Enter your choice (1-9):

FUNDS TRANSFER:

Enter your choice (1-9): 3

Enter the name or debit number of the recipient:

faiz tanveer

Enter the amount to send (in Rps):

300

Name: faiz tanveer
Amount sent: 300.0
Status: Successfully sent
Date: 2024-06-12
Amount remaining: 4200.0

INQUIRE BALANCE:

Enter your choice (1-9): 4

Name: Zainab Rauf
Your current balance in the account is: 4200.0
Date: 2024-06-12

MINI STATEMENT:

Enter your choice (1-9): 5

Name: Zainab Rauf
Mini Statement:
Transaction 1: Funds Transfer: -300.0 Rps to faiz tanveer on 2024-06-12
Transaction 2: Withdraw: -7000.0 Rps on 2024-06-12
Transaction 3: Withdraw: -500.0 Rps on 2024-06-12
Transaction 4: Deposit: +2000.0 Rps on 2024-06-12

ACCOUNT DETAILS:

Enter your choice (1-9): 6

Debit Card No: 02134221057
Name: Zainab Rauf
Total Amount: 4200.0
Date: 2024-06-12

CHANGE PIN:

Enter your choice (1-9): 7

Enter your new PIN:

2163

PIN successfully changed!

BILL PAYMENT:

Enter your choice (1-9): 8

Enter the bill amount to pay (in Rps):

250

Bill payment successful
Bill amount: 250.0
Date: 2024-06-12
Amount remaining: 3950.0

EXIT:

Enter your choice (1-9): 9

Thank you for using THE UBL BANK!

❖ Conclusion: -

In conclusion, the ATM system is a reliable and efficient way for customers to perform various banking transactions. The system is user- friendly and secure, with multiple layers of security to protect against fraud. The system is able to perform various transactions such as depositing money, withdrawing cash, and checking account balance. With the increasing demand for banking services, the ATM system will continue to play a vital role in the banking industry for many years to come.

From this presentation, one can observe that an ATM system is associated with the bank transactions of the consumers. Majorly, the ATM system is utilized for the money associated transactions from the consumers. Consumers make major use of ATM to withdraw money from their bank account. It is a fast way to get money out of your account, especially when on the go or during a trip.