ATM Withdrawal Process – Problem Solving Assignment

Name – Varsha Pal

Course – B.Tech. (Cybersecurity)

Subject– Fundamentals of computer

Teacher's name – Feroz Ahmed

Date – 15-oct-2025

Introduction

The ATM Withdrawal Process is a common banking transaction that allows users to withdraw money using their bank cards and PIN. This system must verify the user's identity, check if there are sufficient funds, and dispense the correct amount of money. Automating this process helps in creating secure and user-friendly banking services.

Essential Elements:

- User inserts card and enters PIN
- System verifies PIN
- User selects "Withdraw"
- User enters amount
- System checks balance
- If sufficient balance, amount is dispensed
- Balance is updated and transaction ends

Ignored Details:

- Card reading hardware details
- Bank backend database implementation
- Receipt printing and multi-language support

Decomposition

1. Insert card.

2. Enter and verify PIN.

3. Choose transaction type (e.g., Withdraw).

4. Enter withdrawal amount.

5. Check if amount <= balance.

6. If yes, dispense cash and update balance.

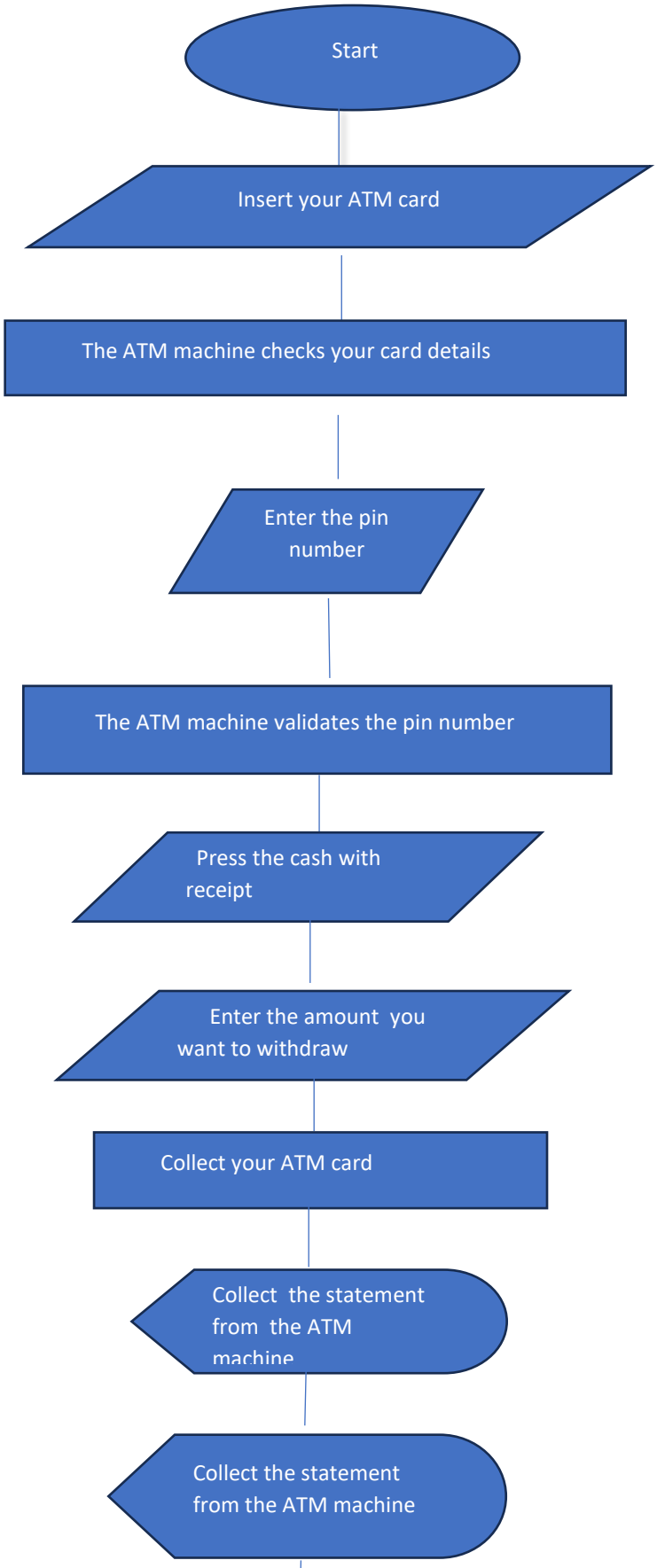
7. Display success message and end session.

8. If no, display error message.

Pattern Recognition

- Repeated use of user input and validation.
- Similar logic to login systems (PIN verification).
- Common condition checks (e.g., sufficient balance).
- Decision making through if-else structures.

FLOW CHART OF ATM WITHDRAWL PROCESS


Start
Insert your ATM card
The ATM machine checks your card details
Enter the pin number
The ATM machine validates the pin number
Press the cash with receipt
Enter the amount you want to withdraw
Collect your ATM card
Collect the statement from the ATM machine
Collect the statement from the ATM machine

Pseudocode

```
BEGIN
  DISPLAY "Insert your card"
  PROMPT "Enter your PIN"
  READ pin
  IF pin is correct THEN
    DISPLAY "Select transaction type"
    IF transaction = "Withdraw" THEN
      PROMPT "Enter amount to withdraw"
      READ amount
      IF amount <= balance THEN
        balance = balance - amount
        DISPENSE cash
        PRINT "Transaction Successful"
        PRINT "Remaining Balance: ", balance
      ELSE
        PRINT "Insufficient Balance"
      END IF
    END IF
  ELSE
    PRINT "Incorrect PIN. Try again."
  END IF
END
```

IMPLEMENTATION

You can select one block, for example: **PIN verification and withdrawal logic**.

```
# ATM Withdrawal - Simple Python Simulation


# Predefined PIN and balance
correct_pin = "1234"
account_balance = 1000.0


# Step 1: PIN verification
```

```
entered_pin = input("Enter your PIN: ")


if entered_pin == correct_pin:
    print("PIN verified.")
    try:
        amount = float(input("Enter amount to withdraw: "))
        if amount <= account_balance:
            account_balance -= amount
            print(f"Please take your cash: ${amount}")
            print(f"Remaining Balance: ${account_balance}")
        else:
            print("Insufficient balance.")
    except ValueError:
        print("Invalid input. Please enter a number.")
else:
    print("Incorrect PIN. Access denied.")
```

OUTPUT

```
PS C:\Users\ADMIN> & C:/Users/ADMIN/AppData/Local/Microsoft/WindowsApps/python3.12.exe
c:/Users/ADMIN/python21.py
Enter your PIN: 1234
PIN verified.
Enter amount to withdraw: 234
Please take your cash: $234.0
Remaining Balance: $766.0
PS C:\Users\ADMIN>
```

IMPLEMENTATION EXPLANATION

The Python code simulates an ATM withdrawal process. It first asks for a PIN and verifies it. If correct, it allows the user to enter an amount to withdraw. It checks if the balance is sufficient and updates the balance if the transaction is successful. If not, it prints an error. It uses conditionals, exception handling, and simple arithmetic.

REFLECTION

Challenges:

- Handling invalid input (e.g., text instead of numbers)

- Simulating real-world ATM steps in simple code

Insights:

- Learned how to break down a system into logical steps

- Used input validation and conditional checks effectively

Improvements:

- Add more security (e.g., limited attempts)

- Simulate full ATM system with deposits, mini statement, etc.

- Use functions and classes for better structure