Name : Siddabathula Jagadeesh Babu

Email :siddabathulajagadeeshbabu@gmail.com

Task No : 2

Title: Design and Implementation Report for ATM Interface Program in Java

Introduction:

This report presents the design and implementation details of an ATM interface program developed in Java. The program aims to provide users with basic banking operations such as checking balance, withdrawing money, and depositing money, while ensuring security through user authentication.

Functionalities:

User Class:

The User class represents each user of the ATM and includes the following attributes:

userID: Represents the unique identifier for each user.

userPIN: Represents the Personal Identification Number (PIN) associated with the user.

accountBalance: Represents the current balance of the user's account.

```
public class User {
    private String userID;
    private int userPIN;
    private double accountBalance;

    // Constructor
    public User(String userID, int userPIN, double accountBalance) {
        this.userID = userID;
        this.userPIN = userPIN;
        this.accountBalance = accountBalance;
    }

    // Getters and Setters
    // ...
```

}

ATM Class:

The ATM class encapsulates ATM functionalities and includes methods for performing operations like checking balance, withdrawing money, and depositing money.

```
public class ATM {
    // Method to check balance
    public double checkBalance(User user) {
        return user.getAccountBalance();
    }

    // Method to withdraw money
    public void withdraw(User user, double amount) {
        double balance = user.getAccountBalance();
        if (balance >= amount) {
            balance -= amount;
            user.setAccountBalance(balance);
            System.out.println("Amount withdrawn: " + amount);
        } else {
            System.out.println("Insufficient balance.");
        }
    }

    // Method to deposit money
    public void deposit(User user, double amount) {
        double balance = user.getAccountBalance();
        balance += amount;
        user.setAccountBalance(balance);
        System.out.println("Amount deposited: " + amount);
    }
```

}

User Authentication:

The program prompts users to enter their user ID and PIN upon startup and validates the entered credentials against stored user data.

Input/Output Handling:

Java's input/output functionalities are utilized to interact with users. Messages and prompts are displayed to guide users through the ATM interface.

Error Handling:

Error handling mechanisms are implemented to deal with invalid user input, insufficient funds, etc. Informative error messages are provided to assist users in correcting their mistakes.

Testing and Debugging:

The program is thoroughly tested with different scenarios to ensure functionality and reliability. Any issues encountered during testing are debugged, and necessary adjustments are made to the code.

Documentation:

The code is documented using comments to explain the purpose of each class, method, and significant block of code. A README file is included with instructions on running the program and other relevant information.

Conclusion:

The ATM interface program developed in Java incorporates essential functionalities such as user management, ATM operations, user authentication, input/output handling, error handling, testing, debugging, and documentation. This design ensures usability, security, functionality, and reliability for the ATM system.