```java
package com.beginning_java;

import java.util.*;
import java.util.concurrent.TimeUnit;

class User{
    String name;
    Account account;
    Card card;
    User(String name,Account account, Card card){
        this.name = name;
        this.account = account;
        this.card = card;
    }
}
class Account{
    int balance;
    Account(int balance){
        this.balance = balance;
    }
}
class Card{
    HashMap<Integer,Integer> idAndPin =new HashMap<>();
    Card(int id,int pin){
        idAndPin.put(id,pin);
    }
}
```

```java

class ATM{
    Scanner sc = new Scanner(System.in);

    final int STARTING_BALANCE = 10000;
    int machineBalance = STARTING_BALANCE;

    User user;
    ATM(User user){
        this.user = user;
    }

    void startMachine(){
        System.out.println("Welcome");
        try{
            TimeUnit.MILLISECONDS.sleep(500);
        }catch (Exception ignored){}
        System.out.println("Insert your card and enter your PIN");
        authenticatePin();
    }
    private void authenticatePin(){
        int enteredPin = sc.nextInt();
        if(user.card.idAndPin.get(1)==enteredPin){
            System.out.println("PIN validated");
            try{
                TimeUnit.SECONDS.sleep(1);
            }catch (Exception ignored){}
            selectTransaction();
        }
        else{
            System.out.println("ERROR : Invalid PIN");
            try{
                TimeUnit.SECONDS.sleep(3);
            }catch (Exception ignored){}
            System.out.println("");
            startMachine();
        }
    }
    private void selectTransaction(){
        System.out.println("Hello Mr. "+user.name);
        try{
            TimeUnit.SECONDS.sleep(1);
```

```java
        }catch (Exception ignored){}
        System.out.println("Select transaction");
        System.out.println("Cash Withdraw(Enter 1) Cash Deposit(Enter 2) View Balance(Enter 3)");
        int selectedTransaction = sc.nextInt();
        if(selectedTransaction==1){
            withdraw();
        }
        else if(selectedTransaction==2){
            deposit();
        }
        else if(selectedTransaction==3){
            viewBalance();
        }
        else{
            System.out.println("Invalid Option, kindly re-select the transaction");
            selectTransaction();
        }
    }
    private void withdraw(){
        System.out.println("Enter the amount to be withdrawn");
        int withdrawlAmount = sc.nextInt();
        if(withdrawlAmount % 100 != 0){
            System.out.println("ERROR : Only 100 and 500 denominations available in machine");
            try{
                TimeUnit.SECONDS.sleep(2);
            }catch (Exception ignored){}
            System.out.println("");
            startMachine();
        }
        else{
            if(user.account.balance<withdrawlAmount){
                System.out.println("ERROR : Insufficient Balance in Account");
                try{
                    TimeUnit.SECONDS.sleep(2);
                }catch (Exception ignored){}
                System.out.println("");
                startMachine();
            }
            else if(machineBalance< withdrawlAmount){
                System.out.println(" ERROR : Insufficient Balance in Machine");
                machineBalance += STARTING_BALANCE;
                try{
                    TimeUnit.SECONDS.sleep(2);
                }catch (Exception ignored){}
                System.out.println("");
                startMachine();
            }
            else{
                user.account.balance -= withdrawlAmount;
                machineBalance -= withdrawlAmount;
                try{
                    TimeUnit.SECONDS.sleep(1);
                }catch (Exception ignored){}
                System.out.println("
Please collect your cash");
                try{
                    TimeUnit.SECONDS.sleep(1);
                }catch (Exception ignored){}
                System.out.println("Transaction is Successful");
                try{
                    TimeUnit.SECONDS.sleep(3);
                }catch (Exception ignored){}
                System.out.println("");
                startMachine();
            }
        }

    }
    private void deposit(){
        System.out.println("Enter the amount to be deposited");
        int depositAmount = sc.nextInt();
        System.out.println("Put the "+depositAmount+" cash in the slot");
        try{
```

```java
            TimeUnit.SECONDS.sleep(2);
        }catch (Exception ignored){}
        user.account.balance += depositAmount;
        System.out.println("Transaction is Successful");
        try{
            TimeUnit.SECONDS.sleep(3);
        }catch (Exception ignored){}
        System.out.println("");
        startMachine();
    }
    private void viewBalance(){
        System.out.println("Current Account Balance : "+ user.account.balance);
        try{
            TimeUnit.SECONDS.sleep(1);
        }catch (Exception ignored){}
        System.out.println("Transaction is Successful");
        try{
            TimeUnit.SECONDS.sleep(3);
        }catch (Exception ignored){}
        System.out.println("");
        startMachine();
    }
}

public class Main {

    public static void main(String[] args) {

        Account accountA = new Account(5000);
        Card cardA = new Card(1,1234);
        User userA = new User("Deepak",accountA,cardA);

        ATM atm = new ATM(userA);
        atm.startMachine();

    }
}

```