```java
import java.util.*;



import java.io.*;

class User {
    private String name;
    private String cardNumber;
    private double balance;
    private int pin;

    User(String name, String cardNumber, double balance, int pin) {
        this.name = name;
        this.cardNumber = cardNumber;
        this.balance = balance;
        this.pin = pin;
    }

    public String getName() {
        return name;
    }

    public String getCardNumber() {
        return cardNumber;
    }

    public double getBalance() {
        return balance;
    }

    public int getPin() {
        return pin;
    }

    public void deposit(double amount) {
        balance = balance + amount;
    }

    public void withdraw(double amount) {
        balance = balance - amount;
    }
}

class ATM {
```

```
    private User u;
    private User[] users = new User[100];
    private String adminPassword;
    private int numberOfUsers = 0;
    private File file;

    ATM(String adminPassword) throws IOException{
        this.adminPassword = adminPassword;
        file = new File("users.txt");
        if (!file.exists()) {
            file.createNewFile();
        } else {

            try (Scanner scanner = new Scanner(file)) {
                while (scanner.hasNextLine()) {
                    String line = scanner.nextLine();
                    String[] parts = line.split(",");
                    String name = parts[0];
                    String cardNumber = parts[1];
                    double balance = Double.parseDouble(parts[2]);
                    int pin = Integer.parseInt(parts[3]);
                    users[numberOfUsers++] = new User(name, cardNumber, balance,
pin);

                }
            }
        }
    }

    public User getUser(String cardNumber) {
        for (int i = 0; i < numberOfUsers; i++) {
            if (users[i].getCardNumber().equals(cardNumber)) {
                return users[i];
            }
        }
        return null;
    }



    public boolean authenticatePin(String cardNumber, int pin) {
        User user = getUser(cardNumber);
        if (user != null && user.getPin() == pin) {
            return true;
        }
```

```
        return false;
    }


    public void deposit(String cardNumber, double amount) {
        User user = getUser(cardNumber);
        if (user != null) {
            user.deposit(amount);
        }
    }

    public void withdraw(String cardNumber, double amount) {
        User user = getUser(cardNumber);
        if (user != null) {
            user.withdraw(amount);
        }
    }

    public boolean isAdmin(String password) {
        return password.equals(adminPassword);
    }

    public void createUser(String name, String cardNumber, double amount, int
pin) throws IOException {
        if(numberOfUsers<100){
            users[numberOfUsers] = new User(name, cardNumber, amount, pin);
            numberOfUsers++;


            try (FileWriter writer = new FileWriter(file, true)) {
                writer.write(name + "," + cardNumber + "," + amount + ","
                        + pin + "\n");
                writer.close();
            }

        }

    }

    public double checkBalance(String cardNumber) {
        User user = getUser(cardNumber);
        if (user != null) {
            user.getBalance();
        }
        return user.getBalance();
```

```
    }

    public void deleteUser(String cardNumber)throws IOException {
        for (int i = 0; i < numberOfUsers; i++) {
            if (users[i].getCardNumber().equals(cardNumber)) {
              users[i] = null;
              for (int j = i; j < numberOfUsers - 1; j++) {
                users[j] = users[j + 1];
              }
              users[numberOfUsers - 1] = null;
              numberOfUsers--;
              saveUserData();
              break;
            }
          }
    }

    public User[] getUsers() {
        return Arrays.copyOf(users, numberOfUsers);
    }
    public void saveUserData()throws IOException{
        FileWriter writer = new FileWriter("users.txt", false);
        for(int i= 0;i<numberOfUsers;i++){
            User user = users[i];
            if(user != null){
                writer.write(user.getName()+","+user.getCardNumber()+","+user.get
Balance()+","+user.getPin()+"\n");
            }
        }
        writer.close();
    }
}

class Admin {
    private ATM atm;

    Admin(ATM atm) {
        this.atm = atm;
    }

    public void createNewUser(String name, String cardNumber, double amount, int
pin)throws IOException {
        atm.createUser(name, cardNumber, amount, pin);
    }
```

```
    public void deleteUser(String cardNumber)throws IOException {
        atm.deleteUser(cardNumber);
    }

    public User[] getUsers() {
        return atm.getUsers();
    }
}

public class AtmSimulationSystem {
    public static void main(String[] args)throws IOException {
        ATM atm = new ATM("admin2911");
        Scanner ab = new Scanner(System.in);
        Admin admin = new Admin(atm);


        System.out.println("***********Welcome To the ATM Simulation
System**********");
        System.out.println("What do You Want to select .....");
        System.out.println("1.Admin Mode");
        System.out.println("2.User Mode");
        System.out.println("Press A for Admin Mode and U for User Mode");
        String mode = ab.nextLine();
        if(mode.equalsIgnoreCase("A")){
            System.out.println("You have Chosen Admin Mode:");
            System.out.println("To continue You have to Enter Your password:");
            String adminPassword = ab.nextLine();
            if(atm.isAdmin(adminPassword)){
                while(true){

                    System.out.println("What Would You Want to do.");
                    System.out.println("1.Create New User");
                    System.out.println("2.Delete User");
                    System.out.println("3.Veiw all Users");
                    System.out.println("4.Check Balance");
                    System.out.println("5.LogOut");
                    System.out.println();
                    System.out.print("Enter Your Choice:");
                    int choice = ab.nextInt();
                    switch(choice){
                        case 1:
                        System.out.println("Enter Name: ");
                        String name = ab.next();
                        System.out.println("Allocated Card Number: ");
                        String cardNumber = ab.next();
```

```
                        System.out.println("Enter Balance: ");
                        double balance = ab.nextDouble();
                        System.out.println("Set Pin: ");
                        int pin = ab.nextInt();
                        try{
                            atm.createUser(name, cardNumber, balance, pin);
                            System.out.println("User created successfully");
                        } catch (IOException e) {
                            System.out.println("Error: " + e.getMessage());
                        }
                        break;
                        case 2:
                        ab.nextLine();
                        System.out.println("Enter the Card Number You want to
Delete");
                        cardNumber = ab.nextLine();
                        try{
                            atm.deleteUser(cardNumber);
                            System.out.println("User Deleted Successfully");
                        }
                        catch(IOException e){
                            System.out.println("Error: "+e);
                        }
                        break;
                        case 3:
                        User[] users = atm.getUsers();
                        for (User user:users) {
                            if (users != null) {
                              System.out.println("Name: " + user.getName());
                              System.out.println("Card Number: " +
user.getCardNumber());
                              System.out.println("Balance: " +
user.getBalance());
                              System.out.println("PIN: " + user.getPin());
                              System.out.println();
                            }
                          }
                          break;
                        case 4:
                        ab.nextLine();
                        System.out.print("Enter Card Number: ");
                        cardNumber = ab.nextLine();
                        balance = atm.checkBalance(cardNumber);

                        System.out.println("Current balance: " + balance);
```

```
                    System.out.println();
                    break;
                    case 5:
                    System.out.println("Logging Out.....");

                    return;
                    default:
                    System.out.println("Invalid Option......");

                }

            }
        }else {
            System.out.println("Incorrect Password . Please Try Again...");
        }
    }else if(mode.equalsIgnoreCase("U")){
        System.out.println("You have Chosen User Mode");
        while(true){

        System.out.println("What Would you want to do");
        System.out.println("1.Withdraw Funds");
        System.out.println("2.Deposit Funds");
        System.out.println("3.Check Current Balance");
        System.out.println("4.Exit");
        System.out.println();
        System.out.print("Enter Choice : ");
        int choice1 = ab.nextInt();
        System.out.println();
        switch(choice1){
            case 1:
            ab.nextLine();
            System.out.println("Enter your Card Number");
            String cardNumber = ab.next();
            System.out.println("Enter Your Pin");
            int pin = ab.nextInt();
            if(atm.authenticatePin(cardNumber, pin)){
                while(true){
                    System.out.print("Enter amount You want to Withdraw: ");
                    double amount = ab.nextDouble();

                    atm.withdraw(cardNumber,amount);
                    System.out.println("Withdrawal Successfully Done");
                    atm.saveUserData();

                    break;
```

```
                }
            }
            else{
                System.out.println("Invalid Pin");
            }
            break;
            case 2:
            ab.nextLine();
            System.out.println("Enter the Card Number");
            cardNumber = ab.nextLine();
            System.out.println("Enter Your Pin");
            pin = ab.nextInt();
            if(atm.authenticatePin(cardNumber, pin)){
                while(true){
                    System.out.println("Enter amount you Want to Deposit");
                    double amount = ab.nextDouble();
                    atm.deposit(cardNumber, amount);
                    System.out.println("Amount Deposited Successfully");
                    atm.saveUserData();
                    break;
                }
            }
            else{
                System.out.println("Invalid Pin");
            }
            break;
            case 3:
            ab.nextLine();
            System.out.println("Enter Your card Number");
            cardNumber = ab.nextLine();
            System.out.println("Enter Your Pin");
            pin = ab.nextInt();
            if(atm.authenticatePin(cardNumber, pin)){
                while(true){
                    double balance = atm.checkBalance(cardNumber);
                    System.out.println("Current Balance :"+balance);
                    System.out.println();
                    break;
                }
            }
            else{
                System.out.println("Invalid Pin");
            }
            break;
            case 4:
```

```
                System.out.println("Exiting The User Mode.....");
                System.out.println("*******Thanks For Using the ATM*******");
                return;
                default :
                System.out.println("Invalid Option");
                break;



            }
        }

    }else {
        System.out.println("Invalid Mode Entered ....");
    }
    ab.close();
    }
}
```