package com.cg.ibs.cardmanagement.ui;

import java.io.NotActiveException;

import java.math.BigInteger;

import java.util.InputMismatchException;

import java.util.List;

import java.util.Scanner;

import com.cg.ibs.cardmanagement.bean.CaseIdBean;

import com.cg.ibs.cardmanagement.bean.CreditCardBean;

import com.cg.ibs.cardmanagement.bean.CreditCardTransaction;

import com.cg.ibs.cardmanagement.bean.DebitCardBean;

import com.cg.ibs.cardmanagement.bean.DebitCardTransaction;

import com.cg.ibs.cardmanagement.exceptionhandling.IBSException;

import com.cg.ibs.cardmanagement.service.BankService;

import com.cg.ibs.cardmanagement.service.BankServiceClassImpl;

import com.cg.ibs.cardmanagement.service.CustomerService;

import com.cg.ibs.cardmanagement.service.CustomerServiceImpl;

public class CardManagementUI {

static BigInteger accountNumber = null;

static Scanner scan;

static BigInteger debitCardNumber = null;

static BigInteger creditCardNumber = null;

static int userInput = -1;

static int ordinal = -1;

static String type = null;

static String transactionId;

static boolean success = false;

static int myChoice = -1;

static int pin = -1;

boolean check = false;

static String customerReferenceId = null;

static int newCardType = -1;

static int days = 0;

CustomerService customService = new CustomerServiceImpl();

BankService bankService = new BankServiceClassImpl();

public void doIt() throws IBSException {

while (true) {

success = false;

System.out.println("Welcome to card management System");

System.out.println("Enter 1 to login as a customer");

System.out.println("Enter 2 to login as a bank admin");

while (!success) {

try {

userInput = scan.nextInt();

success = true;

} catch (InputMismatchException wrongFormat) {

scan.next();

System.out.println("Enter between 1 or 2");

}

}

if (userInput == 1) {

System.out.println("You are logged in as a customer");

CustomerMenu choice = null;

while (choice != CustomerMenu.CUSTOMER\_LOG\_OUT) {

System.out.println("Menu");

System.out.println("--------------------");

System.out.println("Choice");

System.out.println("--------------------");

for (CustomerMenu mmenu : CustomerMenu.values()) {

System.out.println(mmenu.ordinal() + "\t" + mmenu);

}

System.out.println("Choice");

success = false;

while (!success) {

try {

ordinal = scan.nextInt();

success = true;

} catch (InputMismatchException wrongFormat) {

scan.next();

System.out.println("Choose a valid option");

}

}

if (ordinal >= 0 && ordinal < 16) {

choice = CustomerMenu.values()[ordinal];

BigInteger creditCardNumber = null;

switch (choice) {

case LIST\_EXISTING\_DEBIT\_CARDS:

listExistingDebitCards();

break;

case LIST\_EXISTING\_CREDIT\_CARDS:

listExistingCreditCards();

break;

case APPLY\_NEW\_DEBIT\_CARD:

applyNewDebitCard();

break;

case APPLY\_NEW\_CREDIT\_CARD:

applyNewCreditCard();

break;

case UPGRADE\_EXISTING\_DEBIT\_CARD:

upgradeExistingDebitCard();

break;

case UPGRADE\_EXISTING\_CREDIT\_CARD:

upgradeExistingCreditCard();

break;

case RESET\_DEBIT\_CARD\_PIN:

resetDebitCardPin();

break;

case RESET\_CREDIT\_CARD\_PIN:

resetCreditCardPin();

break;

case REPORT\_DEBIT\_CARD\_LOST:

reportDebitCardLost();

break;

case REPORT\_CREDIT\_CARD\_LOST:

reportCreditCardLost();

break;

case REQUEST\_DEBIT\_CARD\_STATEMENT:

requestDebitCardStatement();

break;

case REQUEST\_CREDIT\_CARD\_STATEMENT:

requestCreditCardStatement();

break;

case REPORT\_DEBITCARD\_STATEMENT\_MISMATCH:

reportDebitStatementMismatch();

break;

case REPORT\_CREDITCARD\_STATEMENT\_MISMATCH:

reportCreditStatementMismatch();

break;

case CUSTOMER\_LOG\_OUT:

System.out.println("LOGGED OUT");

break;

case VIEW\_QUERY\_STATUS:

viewQueryStatus();

break;

}

}

}

} else {

if (userInput == 2) {

System.out.println("You are logged in as a Bank Admin");

BankMenu cchoice = null;

while (cchoice != BankMenu.BANK\_LOG\_OUT) {

System.out.println("Menu");

System.out.println("--------------------");

System.out.println("Choice");

System.out.println("--------------------");

for (BankMenu mmenu : BankMenu.values()) {

System.out.println(mmenu.ordinal() + "\t" + mmenu);

}

System.out.println("Choice");

success = false;

while (!success) {

try {

ordinal = scan.nextInt();

success = true;

} catch (InputMismatchException wrongFormat) {

scan.next();

System.out.println("Enter a valid option");

}

}

if (ordinal >= 0 && ordinal < BankMenu.values().length) {

cchoice = BankMenu.values()[ordinal];

switch (cchoice) {

case LIST\_QUERIES:

listPendingQueries();

break;

case REPLY\_QUERIES:

replyQueries();

break;

case VIEW\_DEBIT\_CARD\_STATEMENT:

viewBankDebitCardStatement();

break;

case VIEW\_CREDIT\_CARD\_STATEMENT:

viewBankCreditCardStatement();

break;

case BANK\_LOG\_OUT:

System.out.println("LOGGED OUT");

break;

}

}

}

} else {

System.out.println("Invalid Option!!");

}

}

}

}

void listExistingDebitCards() {

List<DebitCardBean> debitCardBeans = customService.viewAllDebitCards();

if (debitCardBeans.isEmpty()) {

System.out.println("No Existing Debit Cards");

} else {

for (DebitCardBean debitCardBean : debitCardBeans) {

System.out.println("Debit card type :\t" + debitCardBean.getDebitCardType());

System.out.println("Name on debit card :\t" + debitCardBean.getNameOnDebitCard());

System.out.println("UCI :\t" + debitCardBean.getUCI());

System.out.println("Account number :\t" + debitCardBean.getAccountNumber());

System.out.println("Debit card number :\t" + debitCardBean.getDebitCardNumber());

System.out.println("Debit card date of expiry(yyyy/MM/dd):\t" + debitCardBean.getDebitDateOfExpiry());

System.out.println("......................................................");

}

}

}

void listExistingCreditCards() {

List<CreditCardBean> creditCardBeans = customService.viewAllCreditCards();

if (creditCardBeans.isEmpty()) {

System.out.println("No Existing Credit Cards");

} else {

for (CreditCardBean creditCardBean : creditCardBeans) {

System.out.println("Credit card number:\t" + creditCardBean.getCreditCardNumber());

System.out.println("Credit card status:\t" + creditCardBean.isCreditCardStatus());

System.out.println("Name on Credit card:\t" + creditCardBean.getNameOnCreditCard());

System.out

.println("Credit card date of expiry(yyyy/MM/dd):\t" + creditCardBean.getCreditDateOfExpiry());

System.out.println("UCI:\t" + creditCardBean.getUCI());

System.out.println("Credit card type:\t" + creditCardBean.getCreditCardType());

System.out.println("Credit score:\t" + creditCardBean.getCreditScore());

System.out.println("Credit limit:\t" + creditCardBean.getCreditLimit());

System.out.println("......................................................");

}

}

}

void applyNewDebitCard() {

success = false;

System.out.println("You are applying for a new Debit Card");

while (!success) {

try {

System.out.println("Enter Account Number you want to apply debit card for :");

accountNumber = scan.nextBigInteger();

check = customService.verifyAccountNumber(accountNumber);

success = true;

} catch (InputMismatchException wrongFormat) {

// scan.next();

if (scan.next().equals("x"))

return;

System.out.println("Renter 10 digit account number");

continue;

} catch (IBSException notFound) {

System.out.println(notFound.getMessage());

continue;

}

}

if (check) {

success = false;

while (!success) {

try {

System.out.println("We offer three kinds of Debit Cards:");

System.out.println(".....................................");

System.out.println("1 for Platinum");

System.out.println("2 for Gold");

System.out.println("3 for Silver");

System.out.println("Choose between 1 to 3");

newCardType = scan.nextInt();

System.out.println("You have applied for: " + customService.getNewCardtype(newCardType));

customerReferenceId = customService.applyNewDebitCard(accountNumber,

customService.getNewCardtype(newCardType));

System.out.println("Application for new debit card success!!");

System.out.println("Your reference Id is " + customerReferenceId);

success = true;

} catch (InputMismatchException cardNew) {

if (scan.next().equals("x"))

return;

System.out.println("Enter 1/2/3");

} catch (IBSException cardNew) {

System.out.println(cardNew.getMessage());

continue;

}

}

}

}

void applyNewCreditCard() {

success = false;

System.out.println("You are applying for a new Credit Card");

while (!success) {

try {

System.out.println("We offer three kinds of Credit Cards:");

System.out.println(".....................................");

System.out.println("1 for Platinum");

System.out.println("2 for Gold");

System.out.println("3 for Silver");

System.out.println("Choose between 1 to 3");

newCardType = scan.nextInt();

System.out.println("You have applied for: " + customService.getNewCardtype(newCardType));

customerReferenceId = customService.applyNewCreditCard(customService.getNewCardtype(newCardType));

System.out.println("Application for new Credit card success!!");

System.out.println("Your reference Id is " + customerReferenceId);

success = true;

} catch (InputMismatchException cardNew) {

if (scan.next().equals("x"))

return;

System.out.println("Enter 1/2/3");

} catch (IBSException cardNew) {

System.out.println(cardNew.getMessage());

continue;

}

}

}

void upgradeExistingDebitCard() {

success = false;

System.out.println("Enter your Debit Card Number: ");

while (!success) {

try {

debitCardNumber = scan.nextBigInteger();

check = customService.verifyDebitCardNumber(debitCardNumber);

type = customService.verifyDebitcardType(debitCardNumber);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Enter a valid Debit card number");

continue;

} catch (IBSException noCard) {

System.out.println(noCard.getMessage());

continue;

}

catch (NullPointerException notFound) {

System.out.println("Debit Card not Found");

continue;

}

if (check) {

if (type.equals("Silver")) {

System.out.println("Choose 1 to upgrade to Gold");

System.out.println("Choose 2 to upgrade to Platinum");

String mString = null;

success = false;

while (!success) {

try {

myChoice = scan.nextInt();

mString = customService.checkMyChoice(myChoice);

System.out.println("You have applied for " + mString);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Choose between 1 or 2");

continue;

} catch (IBSException e) {

System.out.println(e.getMessage());

}

}

System.out.println(customService.requestDebitCardUpgrade(debitCardNumber, mString));

} else if (type.equals("Gold")) {

System.out.println("Choose 2 to upgrade to Platinum");

success = false;

String mString = null;

while (!success) {

try {

myChoice = scan.nextInt();

System.out.println(customService.checkMyChoiceGold(myChoice));

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Enter 2 to upgrade");

continue;

} catch (IBSException e) {

System.out.println(e.getMessage());

}

}

System.out.println(customService.requestDebitCardUpgrade(creditCardNumber, mString));

} else {

System.out.println("You already have a Platinum Card");

}

}

}

}

void upgradeExistingCreditCard() {

success = false;

System.out.println("Enter your Credit Card Number: ");

while (!success) {

try {

creditCardNumber = scan.nextBigInteger();

check = customService.verifyCreditCardNumber(creditCardNumber);

type = customService.verifyCreditcardType(creditCardNumber);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Enter a valid Credit card number");

continue;

} catch (NullPointerException notFound) {

System.out.println(notFound.getMessage());

continue;

} catch (IBSException notFound) {

System.out.println(notFound.getMessage());

continue;

}

if (check) {

if (type.equals("Silver")) {

System.out.println("Choose 1 to upgrade to Gold");

System.out.println("Choose 2 to upgrade to Platinum");

String mString = null;

success = false;

while (!success) {

try {

myChoice = scan.nextInt();

mString = customService.checkMyChoice(myChoice);

System.out.println("You have applied for " + mString);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Choose between 1 or 2");

continue;

} catch (IBSException e) {

System.out.println(e.getMessage());

}

}

System.out.println(customService.requestDebitCardUpgrade(creditCardNumber, mString));

} else if (type.equals("Gold")) {

System.out.println("Choose 2 to upgrade to Platinum");

success = false;

String mString = null;

while (!success) {

try {

myChoice = scan.nextInt();

System.out.println(customService.checkMyChoiceGold(myChoice));

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Enter 2 to upgrade");

continue;

} catch (IBSException e) {

System.out.println(e.getMessage());

}

}

System.out.println(customService.requestDebitCardUpgrade(creditCardNumber, mString));

} else {

System.out.println("You already have a Platinum Card");

}

}

}

}

void resetDebitCardPin() {

success = false;

System.out.println("Enter your Debit Card Number: ");

while (!success) {

try {

debitCardNumber = scan.nextBigInteger();

check = customService.verifyDebitCardNumber(debitCardNumber);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Enter a valid debit card number");

} catch (IBSException newException) {

System.out.println(newException.getMessage());

}

}

if (check) {

System.out.println("Enter your existing pin:");

success = false;

while (!success) {

try {

pin = scan.nextInt();

if (customService.getPinLength(pin) == 4) {

if (customService.verifyDebitPin(pin, debitCardNumber)) {

System.out.println("Enter new pin");

success = false;

while (!success) {

try {

pin = scan.nextInt();

if (customService.getPinLength(pin) != 4)

throw new IBSException("Incorrect Length of pin ");

System.out.println(customService.resetDebitPin(debitCardNumber, pin));

success = true;

} catch (InputMismatchException wrongFormat) {

System.out.println("Enter a valid 4 digit pin");

if (scan.next().equals("x"))

return;

continue;

} catch (IBSException ExceptionObj) {

System.out.println(ExceptionObj.getMessage());

continue;

}

}

} else {

System.out.println("You have entered wrong pin ");

System.out.println("Try again");

}

}

success = true;

} catch (InputMismatchException wrongFormat) {

System.out.println("Enter a valid 4 digit pin");

if (scan.next().equals("x"))

return;

continue;

} catch (IBSException ExceptionObj) {

System.out.println(ExceptionObj.getMessage());

continue;

}

}

}

}

void resetCreditCardPin() {

success = false;

System.out.println("Enter your Credit Card Number: ");

while (!success) {

try {

creditCardNumber = scan.nextBigInteger();

check = customService.verifyCreditCardNumber(creditCardNumber);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Enter a valid credit card number");

} catch (IBSException newException) {

System.out.println(newException.getMessage());

}

}

if (check) {

System.out.println("Enter your existing pin:");

success = false;

while (!success) {

try {

int pin = scan.nextInt();

if (customService.getPinLength(pin) == 4)

if (customService.verifyCreditPin(pin, creditCardNumber)) {

System.out.println("Enter new pin");

success = false;

while (!success) {

try {

pin = scan.nextInt();

if (customService.getPinLength(pin) != 4)

throw new IBSException("Incorrect Length of pin ");

System.out.println(customService.resetCreditPin(creditCardNumber, pin));

success = true;

} catch (InputMismatchException wrongFormat) {

System.out.println("Enter a valid 4 digit pin");

if (scan.next().equals("x"))

return;

continue;

} catch (IBSException ExceptionObj) {

System.out.println(ExceptionObj.getMessage());

continue;

}

}

} else {

System.out.println("You have entered wrong pin ");

System.out.println("Try again");

}

success = true;

} catch (InputMismatchException wrongFormat) {

System.out.println("Enter a valid 4 digit pin");

if (scan.next().equals("x"))

return;

continue;

} catch (IBSException ExceptionObj) {

System.out.println(ExceptionObj.getMessage());

continue;

}

}

}

}

void reportDebitCardLost() {

success = false;

while (!success) {

try {

System.out.println("Enter your Debit Card Number: ");

debitCardNumber = scan.nextBigInteger();

check = customService.verifyDebitCardNumber(debitCardNumber);

if (check) {

System.out.println(customService.requestDebitCardLost(debitCardNumber));

success = true;

}

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Not a valid format");

continue;

} catch (IBSException newException) {

System.out.println(newException.getMessage());

continue;

}

}

}

void reportCreditCardLost() {

success = false;

while (!success) {

try {

System.out.println("Enter your Credit Card Number: ");

creditCardNumber = scan.nextBigInteger();

check = customService.verifyCreditCardNumber(creditCardNumber);

if (check) {

System.out.println(customService.requestCreditCardLost(creditCardNumber));

success = true;

}

} catch (InputMismatchException wrongFormat) {

System.out.println("Not a valid format");

if (scan.next().equals("x"))

return;

continue;

} catch (IBSException newException) {

System.out.println(newException.getMessage());

continue;

}

}

}

void requestDebitCardStatement() {

success = false;

while (!success) {

try {

System.out.println("Enter your Debit Card Number: ");

debitCardNumber = scan.nextBigInteger();

check = customService.verifyDebitCardNumber(debitCardNumber);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Not a valid format");

} catch (IBSException newException) {

System.out.println(newException.getMessage());

continue;

}

}

success = false;

if (check) {

while (!success) {

try {

System.out.println("enter days : ");

days = scan.nextInt();

customService.checkDays(days);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Not a valid format");

} catch (IBSException newException) {

System.out.println(newException.getMessage());

continue;

}

}

try {

List<DebitCardTransaction> debitCardBeanTrns = customService.getDebitTransactions(days,

debitCardNumber);

for (DebitCardTransaction debitCardTrns : debitCardBeanTrns)

System.out.println(debitCardTrns.toString());

}

catch (IBSException newException) {

System.out.println(newException.getMessage());

}

}

}

void requestCreditCardStatement() {

success = false;

while (!success) {

try {

System.out.println("Enter your Credit Card Number: ");

creditCardNumber = scan.nextBigInteger();

check = customService.verifyCreditCardNumber(creditCardNumber);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Not a valid format");

continue;

} catch (IBSException newException) {

System.out.println(newException.getMessage());

continue;

}

}

success = false;

if (check) {

while (!success) {

try {

System.out.println("enter days : ");

days = scan.nextInt();

customService.checkDays(days);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Not a valid format");

continue;

} catch (IBSException e) {

System.out.println(e.getMessage());

continue;

}

try {

List<CreditCardTransaction> creditCardBeanTrns = customService.getCreditTrans(days,

creditCardNumber);

for (CreditCardTransaction creditCardTrns : creditCardBeanTrns)

System.out.println(creditCardTrns.toString());

}

catch (IBSException e) {

System.out.println(e.getMessage());

continue;

}

}

}

}

void reportDebitStatementMismatch() {

success = false;

while (!success) {

try {

System.out.println("Enter your transaction id");

if ((transactionId = scan.next()).equals("x"))

return;

check = customService.checkDebitTransactionId(transactionId);

if (check) {

System.out.println(customService.raiseDebitMismatchTicket(transactionId));

success = true;

}

} catch (InputMismatchException wrongFormat) {

scan.next();

System.out.println("Not a valid format");

continue;

} catch (IBSException e) {

System.out.println(e.getMessage());

continue;

}

}

}

void reportCreditStatementMismatch() {

success = false;

while (!success) {

try {

System.out.println("Enter your transaction id");

if ((transactionId = scan.next()).equals("x"))

return;

check = customService.verifyCreditCardTransactionId(transactionId);

if (check) {

System.out.println(customService.raiseCreditMismatchTicket(transactionId));

success = true;

}

} catch (InputMismatchException wrongFormat) {

scan.next();

System.out.println("Not a valid format");

} catch (IBSException e) {

System.out.println(e.getMessage());

continue;

}

}

}

void viewQueryStatus() {

success = false;

while (!success) {

System.out.println("Enter your Unique Reference ID");

try {

if ((customerReferenceId= scan.next()).equals("x"))

return;

System.out.println(customService.viewQueryStatus(customerReferenceId));

success = true;

} catch (IBSException e) {

System.out.println(e.getMessage());

} catch (NullPointerException n) {

System.out.println("d");

continue;

}

}

}

void listPendingQueries() {

List<CaseIdBean> caseBeans = bankService.viewQueries();

if (caseBeans.isEmpty()) {

System.out.println("No Existing Queries");

} else {

for (CaseIdBean caseId : caseBeans) {

System.out.println(caseId.toString());

}

}

}

void replyQueries() {String queryId = null;

String newStatus = null;

success = false;

while (!success) {

try {

System.out.println("Enter query ID ");

if (( queryId= scan.next()).equals("x"))

return;

check = bankService.verifyQueryId(queryId);

success = true;

} catch (InputMismatchException wrongFormat) {

scan.next();

System.out.println("Not a valid format");

continue;

} catch (IBSException ibs) {

System.out.println(ibs.getMessage());

continue;

}

}

if (check) {

System.out.println("Enter new Status");

if (( newStatus= scan.next()).equals("x"))

return;

bankService.setQueryStatus(queryId, newStatus);

}

else {

System.out.println("Invalid query id");

}

}

void viewBankDebitCardStatement() {

success = false;

while (!success) {

try {

System.out.println("Enter your Debit Card Number: ");

debitCardNumber = scan.nextBigInteger();

check = bankService.verifyDebitCardNumber(debitCardNumber);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Not a valid format");

} catch (IBSException newException) {

System.out.println(newException.getMessage());

continue;

}

}

success = false;

if (check) {

while (!success) {

try {

System.out.println("enter days : ");

days = scan.nextInt();

bankService.checkDays(days);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Not a valid format");

} catch (IBSException newException) {

System.out.println(newException.getMessage());

continue;

}

}

try {

List<DebitCardTransaction> debitCardBeanTrns = bankService

.getDebitTransactions(days, debitCardNumber);

for (DebitCardTransaction debitCardTrns : debitCardBeanTrns)

System.out.println(debitCardTrns.toString());

}

catch (IBSException newException) {

System.out.println(newException.getMessage());

}

}

}

void viewBankCreditCardStatement() {

success = false;

while (!success) {

try {

System.out.println("Enter your Credit Card Number: ");

creditCardNumber = scan.nextBigInteger();

check = bankService.verifyCreditCardNumber(creditCardNumber);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Not a valid format");

continue;

} catch (IBSException newException) {

System.out.println(newException.getMessage());

continue;

}

}

success = false;

if (check) {

while (!success) {

try {

System.out.println("enter days : ");

days = scan.nextInt();

bankService.checkDays(days);

success = true;

} catch (InputMismatchException wrongFormat) {

if (scan.next().equals("x"))

return;

System.out.println("Not a valid format");

continue;

} catch (IBSException e) {

System.out.println(e.getMessage());

continue;

}

try {

List<CreditCardTransaction> creditCardBeanTrns = bankService

.getCreditTrans(days, creditCardNumber);

for (CreditCardTransaction creditCardTrns : creditCardBeanTrns)

System.out.println(creditCardTrns.toString());

}

catch (IBSException e) {

System.out.println(e.getMessage());

continue;

}

}

}}

public static void main(String args[]) throws Exception {

scan = new Scanner(System.in);

CardManagementUI obj = new CardManagementUI();

obj.doIt();

System.out.println("Program End");

obj.scan.close();

}

}