



# EAST WES UNIVERSITY

Project Report on

## East West Bank ATM Management System

Course Code : CSE110

Md. Fokrul Akon

Submission Date : 27/12/2023

## Introduction

The “East West ATM Bank” project is a model Internet online ATM site. This site enables the customers to perform the basic banking transactions by sitting at their office or at homes through PC or laptop. The system provides the access to the account, deposit/withdraw the cash from his account, also to view reports of all accounts present. The customers can access the banks website for viewing their Account details and perform the transactions on account as per their requirements. With Internet Banking, the brick and mortar structure of the traditional banking gets converted into a click and portal model, thereby giving a concept of virtual banking a real shape. Thus, today's banking is no longer confined to branches. E-banking facilitates banking transactions by customers round the clock globally.

The primary aim of this “East West ATM Bank” is to provide an improved design methodology, which envisages the future expansion, and modification, which is necessary for a core sector like banking. This necessitates the design to be expandable and modifiable and so a modular approach is used in developing the application software.

Anybody who is an Account holder in this bank can become a member of the East West ATM Bank. He has to have an account in East West Bank. Bank is the place where customers feel the sense of safety for their property. In the bank, customers deposit and withdraw their money. Transaction of money also is a part where customer takes shelter of the bank. Now to keep the belief and trust of customers, there is the positive need for management of the bank, which can handle all this with comfort and ease. Smooth and efficient management affects the satisfaction of the customers and staff members,

indirectly. And of course, it encourages management committee in taking some needed decision for future enhancement of the bank.

Now a day's, managing a bank is tedious job up to certain limit. So software that reduces the work is essential. Also, today's world is a genuine computer world and is getting faster and faster day-by-day. Thus, considering above necessities, the software for ATM system management has become necessary which would be useful in managing the bank more efficiently.

All transactions are carried out online by transferring from accounts in the same Bank or international bank. The software is meant to overcome the drawbacks of the manual system.

## Problem Statement

The Traditional way of maintaining details of a user in a bank was to enter the details and record them. Every time the user needs to perform some transactions he has to go to bank and perform the necessary actions, which may not be so feasible all the time. It may be a hard-hitting task for the users and the bankers too. The project gives real life understanding of Online ATM System and activities performed by various roles in the supply chain. Here, we provide automation for banking system through Internet. Online ATM System project captures activities performed by different roles in real life banking which provides enhanced techniques for maintaining the required information up-to-date, which results in efficiency. The project gives real life understanding of Online ATM System and activities performed by various roles in the supply chain

1. **Main Objective :** Our goal is to develop a software program for managing the entire ATM process related to Administration accounts customer accounts and to keep each every track about their property and their various transaction processes efficiently. Hereby, our main objective is the customer's satisfaction considering today's faster in the world.
2. **Customer Satisfaction:** Client can do his operations comfortably without any risk or losing of his privacy. Our software will perform and fulfill all the tasks that any customer would desire.
3. **Saving Customer Time:** Client doesn't need to go to the bank to do small operation.
4. **Protecting the Customer:** It helps the customer to be satisfied and comfortable in his choices, this protection contains customer's account, money and his privacy.
5. **Transactions Money:** Help client transactions money to/or another bank .

## Key Features and Methods

### 1. Login page():

- Option of entering card number and pin number.
- Login option(can only login if the card and pin number is valid).
- Clear option(can clear the given data and restart).

## **2. Transaction():**

- Deposit Option
- Cash Withdrawal Option
- Balance Enquiry Option
- Pin Change Option
- Exit Option

## **3. MiniStatement():**

- Shows last ten transactions.

## **4. Withdraw1():**

- Withdrawing Money(can successfully withdraw money if the customer has sufficient balance).
- Shows the total amount after withdrawing and also the withdrawal date

## **5. Deposit():**

- Depositing money
- Shows the total amount after depositing.

## **6. Balance Enquiry():**

- Can check the balance of the account.

## Source Code:

```
package bankmanagementsystem;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.io.*;
```

```
import java.util.*;
```

```
import javax.swing.*;
```

```
public class Login_page extends JFrame implements ActionListener {
```

```
    JButton login, clear;
```

```
    JTextField cardTextField;
```

```
    JPasswordField pinTextField;
```

```
    String cardno = "";
```

```
    String pinno;
```

```
    Login_page() {
```

```
        setTitle("AUTOMATED TELLER MACHINE");
```

```
        setLayout(null);
```

```
        ImageIcon i1 = new ImageIcon("/Users/imamhossain/Downloads/East West logo.png");
```

```
        Image i2 = i1.getImage().getScaledInstance(100, 100, Image.SCALE_DEFAULT);
```

```
        ImageIcon i3 = new ImageIcon(i2);
```

```
        JLabel label = new JLabel(i3);
```

```
        label.setBounds(70, 10, 100, 100);
```

```
        add(label);
```

```
        JLabel title = new JLabel("East West ATM");
```

```
        title.setFont(new Font("Oswald", Font.BOLD, 30));
```

```
title.setBounds(200, 30, 400, 40);  
add(title);
```

```
JLabel cardNo = new JLabel("Card No:");  
cardNo.setFont(new Font("Raleway", Font.BOLD, 20));  
cardNo.setBounds(200, 150, 150, 40);  
add(cardNo);
```

```
cardTextField = new JTextField();  
cardTextField.setBounds(300, 162, 250, 20);  
cardTextField.setFont(new Font("Arial", Font.BOLD, 14));  
add(cardTextField);
```

```
JLabel pin = new JLabel("Pin:");  
pin.setFont(new Font("Raleway", Font.BOLD, 20));  
pin.setBounds(200, 180, 400, 40);  
add(pin);
```

```
pinTextField = new JPasswordField();  
pinTextField.setBounds(300, 192, 250, 20);  
pinTextField.setFont(new Font("Arial", Font.BOLD, 14));  
add(pinTextField);
```

```
login = new JButton("Login");  
login.setBounds(350, 250, 100, 25);  
//login.setForeground(Color.WHITE);  
//login.setBackground(Color.black);  
login.addActionListener(this);  
add(login);
```

```

clear = new JButton("clear");
clear.setBounds(350, 280, 100, 25);
//clear.setForeground(Color.WHITE);
//clear.setBackground(Color.black);
clear.addActionListener(this);
add(clear);

getContentPane().setBackground(Color.white);

setSize(800, 400);
setVisible(true);
setLocation(350, 200);

}

public void actionPerformed(ActionEvent ae) {

    if (ae.getSource() == clear) {
        cardTextField.setText("");
        pinTextField.setText("");
    } else if (ae.getSource() == login) {

        try {
            File f1 = new File("/Users/imamhossain/Downloads/imam/Bank.txt");

            BufferedReader r = new BufferedReader(new FileReader(f1));

            PrintWriter w = new PrintWriter(new FileWriter(f1, true));

            ArrayList<String> arr = new ArrayList<>();

```

```

String line;
String line1;

cardno = cardTextField.getText();
new Transaction(cardno);

pinno = pinTextField.getText();

while ((line = r.readLine()) != null) {

    String[] cut = line.split(" ");

    if (cut[1].equals(cardno) && cut[2].equals(pinno)) {
        setVisible(false);

        new Transaction().setVisible(true);

        break;
    } else
        JOptionPane.showMessageDialog(null, "Incorrect Card
Number or Pin");

    break;
}
} catch (IOException e) {
    System.out.println(e);
}

}

```



```
}
```

```
public static void main(String[] args) {  
    new Login_page();
```

```
}
```

```
}
```

```
package bankmanagementsystem;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.io.*;
```

```
import java.util.*;
```

```
import javax.swing.*;
```

```
public class Transaction extends JFrame implements ActionListener {
```

```
JLabel l1, l2;
```

```
JButton b1, b2, b4, b5, b6, b7;
```

```
public String cardno;
```

```
Transaction(String cardno) {
```

```
    this.cardno = cardno;
```

```
    new Pin(this.cardno);
```

```
        new Withdrawl(this.cardno);
```

```
        new Deposit(this.cardno);
```

```
        new MiniStatement(this.cardno);
```

```
        new BalanceEnquiry1(this.cardno);
```

```
}
```

```
Transaction() {
```

```
    setLayout(null);
```

```
    ImageIcon i1 = new ImageIcon("/Users/imamhossain/Downloads/atm.png");
```

```
    Image i2 = i1.getImage().getScaledInstance(1000, 1180, Image.SCALE_DEFAULT);
```

```
    ImageIcon i3 = new ImageIcon(i2);
```

```
    l2 = new JLabel(i3);
```

```
    l2.setBounds(0, 0, 960, 1080);
```

```
    add(l2);
```

```
l1 = new JLabel("Please Select Your Transaction");  
l1.setForeground(Color.WHITE);  
l1.setFont(new Font("System", Font.BOLD, 16));  
l1.setBounds(235, 400, 700, 35);  
l2.add(l1);
```

```
b1 = new JButton("DEPOSIT");  
b1.setBounds(170, 498, 150, 35);  
b1.addActionListener(this);  
l2.add(b1);
```

```
b2 = new JButton("CASH WITHDRAWAL");  
b2.setBounds(390, 498, 150, 35);  
b2.addActionListener(this);  
l2.add(b2);
```

```
/*b4 = new JButton("MINI STATEMENT");  
b4.setBounds(170, 542, 150, 35);  
b4.addActionListener(this);  
l2.add(b4);*/
```

```
b5 = new JButton("PIN CHANGE");  
b5.setBounds(390, 542, 150, 35);  
b5.addActionListener(this);  
l2.add(b5);
```

```
b6 = new JButton("BALANCE ENQUIRY");  
b6.setBounds(170, 587, 150, 35);  
b6.addActionListener(this);  
l2.add(b6);
```

```

        b7 = new JButton("EXIT");
        b7.setBounds(390, 587, 150, 35);
        b7.addActionListener(this);
        l2.add(b7);

        setSize(960, 1080);
        setLocation(500, 0);

        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        if (ae.getSource() == b1) {
            setVisible(false);
            new Deposit().setVisible(true);
        } else if (ae.getSource() == b2) {
            setVisible(false);
            new Withdrawl().setVisible(true);

        } else if (ae.getSource() == b5) {
            setVisible(false);
            new Pin().setVisible(true);
        } else if (ae.getSource() == b6) {
            this.setVisible(false);
            new BalanceEnquiry1().setVisible(true);
        } else if (ae.getSource() == b7) {
            System.exit(0);
        }
    }

```

```

    }

    public static void main(String[] args) {

        new Transaction();

    }
}

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;
import javax.swing.*;

public class Withdrawal extends JFrame implements ActionListener {
    private JButton withdrawButton, d500, d1000, d2000, d4000, d5000, d10000, backButton,
clearButton;
    private JTextField amountField;
    private JLabel imageLabel, textLabel;
    private static String cardNumber;

    Withdrawal(String cardNumber) {
        this.cardNumber = cardNumber;
    }
}

```

```
Withdrawal() {  
    setLayout(null);  
  
    ImageIcon atmIcon = new ImageIcon("/Users/imamhossain/Downloads/atm.png");  
    Image scaledImage = atmIcon.getImage().getScaledInstance(1000, 1180,  
Image.SCALE_DEFAULT);  
    ImageIcon scaledIcon = new ImageIcon(scaledImage);  
    imageLabel = new JLabel(scaledIcon);  
    imageLabel.setBounds(0, 0, 960, 1080);  
    add(imageLabel);  
  
    textLabel = new JLabel("Enter the amount you want to withdraw");  
    textLabel.setFont(new Font("System", Font.BOLD, 16));  
    textLabel.setBounds(170, 350, 400, 20);  
    textLabel.setForeground(Color.WHITE);  
    textLabel.setBackground(Color.BLACK);  
    imageLabel.add(textLabel);  
  
    amountField = new JTextField();  
    amountField.setFont(new Font("Raleway", Font.BOLD, 22));  
    amountField.setBounds(170, 400, 150, 30);  
    imageLabel.add(amountField);  
  
    withdrawButton = new JButton("Withdraw");  
    withdrawButton.setBounds(150, 635, 150, 30);  
    withdrawButton.setFont(new Font("Raleway", Font.BOLD, 16));  
    withdrawButton.addActionListener(this);  
    imageLabel.add(withdrawButton);
```

```

d500 = new JButton("500");
d500.setBounds(150, 502, 150, 30);
d500.setFont(new Font("Raleway", Font.BOLD, 16));
d500.addActionListener(this);
imageLabel.add(d500);

// Add other buttons and components...

backButton = new JButton("Back");
backButton.setBounds(400, 635, 150, 30);
backButton.setFont(new Font("Raleway", Font.BOLD, 16));
backButton.addActionListener(this);
imageLabel.add(backButton);

clearButton = new JButton("Clear");
clearButton.setBounds(350, 400, 130, 25);
clearButton.setFont(new Font("Raleway", Font.BOLD, 16));
clearButton.addActionListener(this);
imageLabel.add(clearButton);

setSize(900, 900);
setLocation(300, 0);
setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    try {
        // Handle actions...

    } catch (Exception e) {

```

```
        e.printStackTrace();
    }
}

public static void main(String[] args) {
    new Withdrawal();
}
}

package bankmanagementsystem;

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;
import javax.swing.*;

public class Pin extends JFrame implements ActionListener {
    private JPasswordField newPasswordField, reEnterPasswordField;
    private JButton changeButton, backButton;
    private JLabel titleLabel, newPinLabel, reEnterPinLabel, imageLabel;
    private static String cardNumber;

    Pin(String cardNumber) {
        this.cardNumber = cardNumber;
    }
}
```



```
}
```

```
Pin() {
```

```
    setLayout(null);
```

```
    ImageIcon atmIcon = new ImageIcon("/Users/imamhossain/Downloads/atm.png");
```

```
    Image scaledImage = atmIcon.getImage().getScaledInstance(1000, 1180,  
Image.SCALE_DEFAULT);
```

```
    ImageIcon scaledIcon = new ImageIcon(scaledImage);
```

```
    imageLabel = new JLabel(scaledIcon);
```

```
    imageLabel.setBounds(0, 0, 960, 1080);
```

```
    add(imageLabel);
```

```
    titleLabel = new JLabel("CHANGE YOUR PIN");
```

```
    titleLabel.setFont(new Font("System", Font.BOLD, 16));
```

```
    titleLabel.setForeground(Color.WHITE);
```

```
    titleLabel.setBounds(280, 330, 800, 35);
```

```
    imageLabel.add(titleLabel);
```

```
    newPinLabel = new JLabel("New PIN:");
```

```
    newPinLabel.setFont(new Font("System", Font.BOLD, 16));
```

```
    newPinLabel.setBounds(180, 385, 200, 35);
```

```
    newPinLabel.setForeground(Color.WHITE);
```

```
    imageLabel.add(newPinLabel);
```

```
    newPasswordField = new JPasswordField();
```

```
    newPasswordField.setFont(new Font("Raleway", Font.BOLD, 25));
```

```
    newPasswordField.setBounds(350, 390, 180, 25);
```

```
    imageLabel.add(newPasswordField);
```

```

reEnterPinLabel = new JLabel("Re-Enter New PIN:");
reEnterPinLabel.setFont(new Font("System", Font.BOLD, 16));
reEnterPinLabel.setForeground(Color.WHITE);
reEnterPinLabel.setBounds(180, 440, 200, 35);
imageLabel.add(reEnterPinLabel);

reEnterPasswordField = new JPasswordField();
reEnterPasswordField.setFont(new Font("Raleway", Font.BOLD, 25));
reEnterPasswordField.setBounds(350, 440, 180, 25);
imageLabel.add(reEnterPasswordField);

changeButton = new JButton("CHANGE");
changeButton.setBounds(390, 588, 150, 35);
changeButton.addActionListener(this);
imageLabel.add(changeButton);

backButton = new JButton("BACK");
backButton.setBounds(390, 633, 150, 35);
backButton.addActionListener(this);
imageLabel.add(backButton);

setSize(960, 1080);
setLocation(500, 0);
setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    try {
        String newPassword = newPasswordField.getText();
        String reEnterPassword = reEnterPasswordField.getText();

```

```

if (!newPassword.equals(reEnterPassword)) {
    JOptionPane.showMessageDialog(null, "Entered PIN does not match");
    return;
}

if (ae.getSource() == changeButton) {
    if (newPassword.equals("")) {
        JOptionPane.showMessageDialog(null, "Enter New PIN");
    } else if (reEnterPassword.equals("")) {
        JOptionPane.showMessageDialog(null, "Re-Enter new PIN");
    } else {
        File file = new File("/Users/imamhossain/Downloads/imam/Bank.txt");
        BufferedReader reader = new BufferedReader(new FileReader(file));
        PrintWriter writer = new PrintWriter(new FileWriter(file, true));
        ArrayList<String> lines = new ArrayList<>();

        String line;
        while ((line = reader.readLine()) != null) {
            String[] parts = line.split(" ");
            if (parts[1].equals(cardNumber)) {
                parts[2] = newPassword;
                line = parts[0] + " " + parts[1] + " " + parts[2] + " " + parts[3];
                lines.add(line);
            } else {
                lines.add(line);
            }
        }
    }

    PrintWriter writer2 = new PrintWriter(file);

```

```

        for (String x : lines) {
            writer2.println(x);
            writer2.flush();
        }

        JOptionPane.showMessageDialog(null, "PIN changed successfully");
        setVisible(false);
        new Transaction().setVisible(true);
    }
} else if (ae.getSource() == backButton) {
    new Transaction().setVisible(true);
    setVisible(false);
}
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    new Pin();
}
}

```

```

package bankmanagementsystem;

```

```
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;
import javax.swing.*;

public class Deposit extends JFrame implements ActionListener {
    private JButton depositButton, d500, d1000, d2000, d4000, d5000, d10000, backButton,
    clearButton;
    private JTextField amountField;
    private JLabel imageLabel, textLabel;
    private int totalAmount;
    private static String cardNumber;

    Deposit(String cardNumber) {
        this.cardNumber = cardNumber;
    }

    Deposit() {
        setLayout(null);

        ImageIcon atmIcon = new ImageIcon("/Users/imamhossain/Downloads/atm.png");
        Image scaledImage = atmIcon.getImage().getScaledInstance(1000, 1180,
        Image.SCALE_DEFAULT);
        ImageIcon scaledIcon = new ImageIcon(scaledImage);
        imageLabel = new JLabel(scaledIcon);
        imageLabel.setBounds(0, 0, 960, 1080);
        add(imageLabel);

        textLabel = new JLabel("Enter the amount you want to deposit");
```

```
textLabel.setFont(new Font("System", Font.BOLD, 16));
textLabel.setBounds(170, 350, 400, 20);
imageLabel.add(textLabel);
```

```
amountField = new JTextField();
amountField.setFont(new Font("Raleway", Font.BOLD, 22));
amountField.setBounds(170, 400, 150, 30);
imageLabel.add(amountField);
```

```
depositButton = new JButton("Deposit");
depositButton.setBounds(150, 635, 150, 30);
depositButton.setFont(new Font("Raleway", Font.BOLD, 16));
depositButton.addActionListener(this);
imageLabel.add(depositButton);
```

```
d500 = new JButton("500");
d500.setBounds(150, 502, 150, 30);
d500.setFont(new Font("Raleway", Font.BOLD, 16));
d500.addActionListener(this);
imageLabel.add(d500);
```

```
// Add other buttons and components...
```

```
backButton = new JButton("Back");
backButton.setBounds(400, 635, 150, 30);
backButton.addActionListener(this);
imageLabel.add(backButton);
```

```
clearButton = new JButton("Clear");
clearButton.setBounds(350, 400, 130, 25);
```

```

clearButton.setFont(new Font("Raleway", Font.BOLD, 16));
clearButton.addActionListener(this);
imageLabel.add(clearButton);

setSize(900, 900);
setLocation(300, 0);
setVisible(true);
}

public void actionPerformed(ActionEvent ae) {
    try {
        if (ae.getSource() == depositButton) {
            String amt = amountField.getText();
            Date date = new Date();
            if (amt.equals("")) {
                JOptionPane.showMessageDialog(null, "Please Enter the amount you want to
deposit");
            } else {
                int amt1 = Integer.valueOf(amt);

                File file = new File("/Users/imamhossain/Downloads/imam/Bank.txt");
                BufferedReader reader = new BufferedReader(new FileReader(file));
                PrintWriter writer = new PrintWriter(new FileWriter(file, true));
                ArrayList<String> lines = new ArrayList<>();

                String line;
                while ((line = reader.readLine()) != null) {
                    String[] parts = line.split(" ");
                    if (parts[1].equals(cardNumber)) {
                        int amt2 = Integer.valueOf(parts[3]);

```

```

        totalAmount = amt1 + amt2;
        parts[3] = "" + totalAmount;
        line = parts[0] + " " + parts[1] + " " + parts[2] + " " + parts[3];
        lines.add(line);
    } else {
        lines.add(line);
    }
}

```

```

PrintWriter writer2 = new PrintWriter(file);
for (String x : lines) {
    writer2.println(x);
    writer2.flush();
}

```

```

JOptionPane.showMessageDialog(null,
    "Tk: " + amt + " Successfully deposited." + "\nDate:" + date);
setVisible(false);
MiniStatement.addTransaction("Deposit of " + amt1 + " Total:" + totalAmount);
new MiniStatement(amt1, totalAmount).setVisible(true);
new Transaction().setVisible(true);
}

```

```

} else if (ae.getSource() == d500) {
    amountField.setText("500");
} else if (ae.getSource() == d1000) {
    amountField.setText("1000");
} else if (ae.getSource() == clearButton) {
    amountField.setText("");
} else if (ae.getSource() == backButton) {

```



```

        setVisible(false);
        new Transaction().setVisible(true);
    }
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void main(String[] args) {
    new Deposit();
}
}

```

```

package bankmanagementsystem;

```

```

import java.awt.*;
import java.io.*;
import java.text.*;
import java.util.*;
import javax.swing.*;

```

```

public class MiniStatement extends JFrame {
    private static String cardNumber;
    private static int depositAmount, totalAmount;
    private static ArrayList<String> lastTenTransactions = new ArrayList<>();
}

```

```
MiniStatement(String cardNumber) {  
    this.cardNumber = cardNumber;  
}
```

```
MiniStatement(int depositAmount, int totalAmount) {  
    setTitle("Mini Statement");  
    setLayout(null);
```

```
    ImageIcon bankLogo = new ImageIcon("/Users/imamhossain/Downloads/East West  
logo.png");
```

```
    Image scaledImage = bankLogo.getImage().getScaledInstance(100, 100,  
Image.SCALE_DEFAULT);
```

```
    ImageIcon scaledIcon = new ImageIcon(scaledImage);
```

```
    JLabel logoLabel = new JLabel(scaledIcon);
```

```
    logoLabel.setBounds(30, 10, 100, 80);
```

```
    add(logoLabel);
```

```
    JLabel bankLabel = new JLabel("East West Bank");
```

```
    bankLabel.setBounds(220, 30, 300, 20);
```

```
    bankLabel.setFont(new Font("Raleway", Font.BOLD, 20));
```

```
    add(bankLabel);
```

```
    SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy");
```

```
    String todayDate = dateFormat.format(new Date());
```

```
    JLabel dateLabel = new JLabel("Date: " + todayDate);
```

```
    dateLabel.setBounds(440, 120, 150, 20);
```

```
    add(dateLabel);
```

```
    JLabel cardLabel = new JLabel("Card NO: " + cardNumber);
```

```
cardLabel.setBounds(20, 120, 400, 20);  
cardLabel.setFont(new Font("Raleway", Font.BOLD, 20));  
add(cardLabel);
```

```
try {  
    BufferedReader reader = new BufferedReader(new  
FileReader("/Users/imamhossain/Downloads/imam/Transaction.txt"));  
    String line;  
    int i = 0;  
    while ((line = reader.readLine()) != null && i < 10) {  
        JLabel transactionLabel = new JLabel(line);  
        transactionLabel.setFont(new Font("Arial", Font.BOLD, 15));  
        transactionLabel.setBounds(150, 200 + i * 50, 300, 60);  
        add(transactionLabel);  
        i++;  
    }  
    reader.close();  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

```
setSize(600, 600);  
setLocation(20, 20);  
getContentPane().setBackground(Color.WHITE);  
setVisible(true);  
}
```

```
public static void addTransaction(String transaction) {  
    SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy");  
    String todayDate = dateFormat.format(new Date());
```

```

lastTenTransactions.add(todayDate + ": " + transaction);

if (lastTenTransactions.size() > 10) {
    lastTenTransactions.remove(9);
}

try (BufferedWriter writer = new BufferedWriter(new
FileWriter("/Users/imamhossain/Downloads/imam/Transaction.txt"))) {
    for (String trans : lastTenTransactions) {
        writer.write(trans);
        writer.newLine();
    }
} catch (IOException e) {
    e.printStackTrace();
}

}

public static void main(String[] args) {
    new MiniStatement(depositAmount, totalAmount);
}
}

```

```

package bankmanagementsystem;

```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import java.io.*;
```

```
import java.util.*;
```

```
import javax.swing.*;
```

```
public class BalanceEnquiry1 extends JFrame implements ActionListener {
```

```
    private JButton backButton;
```

```
    private int accountBalance;
```

```
    private static String cardNumber;
```

```
    BalanceEnquiry1(String cardNumber) {
```

```
        this.cardNumber = cardNumber;
```

```
        fetchAccountBalance();
```

```
        initializeUI();
```

```
    }
```

```
    BalanceEnquiry1() {
```

```
        fetchAccountBalance();
```

```
        initializeUI();
```

```
    }
```

```
    private void fetchAccountBalance() {
```

```
        try {
```

```
            File file = new File("/Users/imamhossain/Downloads/imam/Bank.txt");
```

```
            BufferedReader reader = new BufferedReader(new FileReader(file));
```

```
            ArrayList<String> lines = new ArrayList<>();
```

```
            String line;
```

```

while ((line = reader.readLine()) != null) {
    String[] parts = line.split(" ");
    if (parts[1].equals(cardNumber)) {
        accountBalance = Integer.parseInt(parts[3]);
    }
}

reader.close();
} catch (IOException e) {
    e.printStackTrace();
}
}

private void initializeUI() {
    setLayout(null);

    ImageIcon atmIcon = new ImageIcon("/Users/imamhossain/Downloads/atm.png");
    Image scaledImage = atmIcon.getImage().getScaledInstance(900, 900,
Image.SCALE_DEFAULT);
    ImageIcon scaledIcon = new ImageIcon(scaledImage);
    JLabel imageLabel = new JLabel(scaledIcon);
    imageLabel.setBounds(0, 0, 900, 900);
    add(imageLabel);

    backButton = new JButton("Back");
    backButton.setBounds(355, 520, 150, 30);
    backButton.addActionListener(this);
    imageLabel.add(backButton);

```

```

        JLabel balanceLabel = new JLabel("Your Current Account Balance is TK " +
accountBalance);

        balanceLabel.setForeground(Color.WHITE);

        balanceLabel.setBounds(170, 300, 400, 30);

        imageLabel.add(balanceLabel);


        setSize(900, 900);

        setLocation(300, 0);

        setUndecorated(true);

        setVisible(true);
    }


    @Override
    public void actionPerformed(ActionEvent ae) {
        setVisible(false);


        new Transaction().setVisible(true);
    }


    public static void main(String[] args) {
        new BalanceEnquiry1();
    }
}

```

output:

AUTOMATED TELLER MACHINE



# East West ATM


Card No:

Pin:

Login


clear

AUTOMATED TELLER MACHINE



# East West ATM

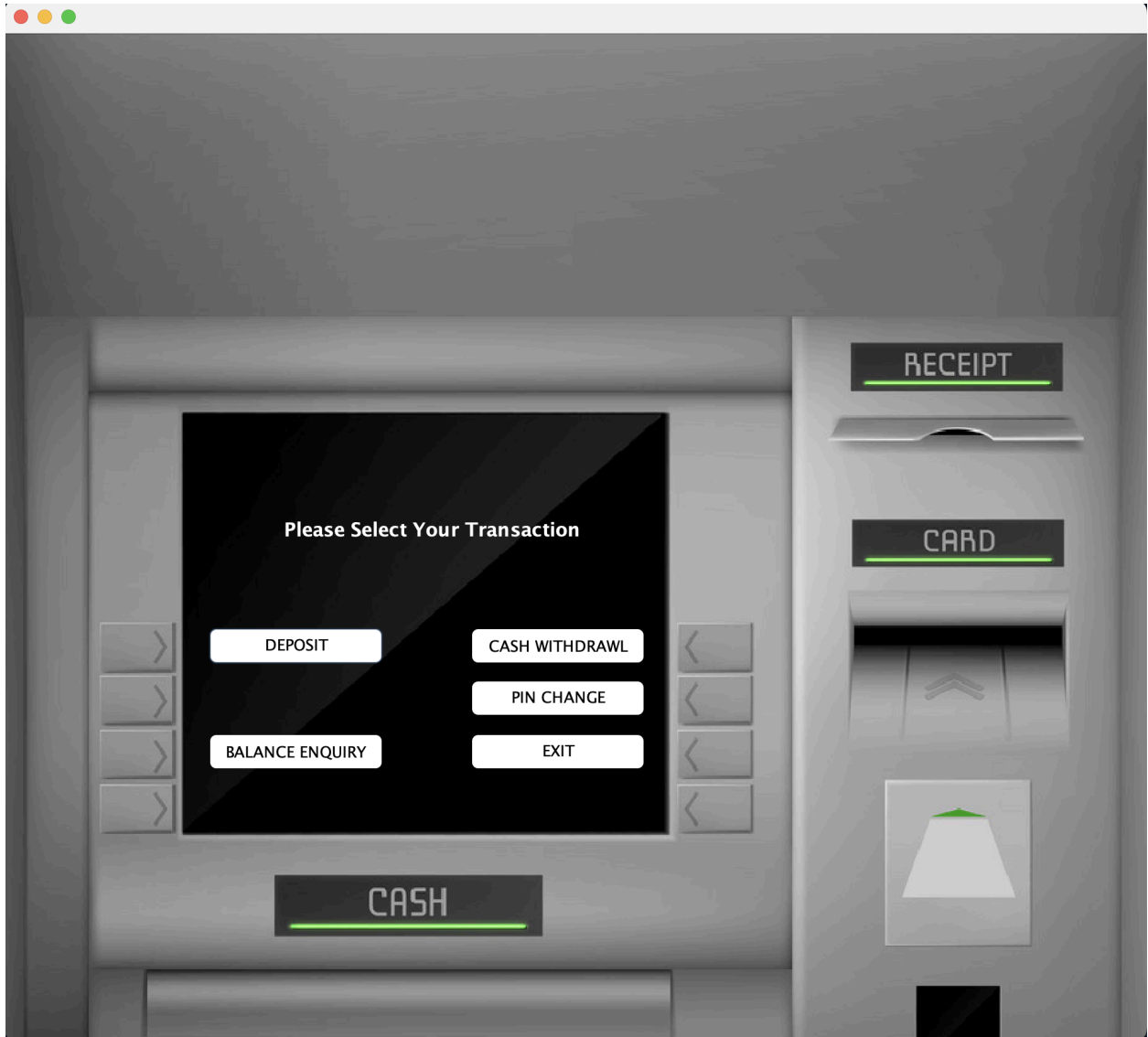
Message



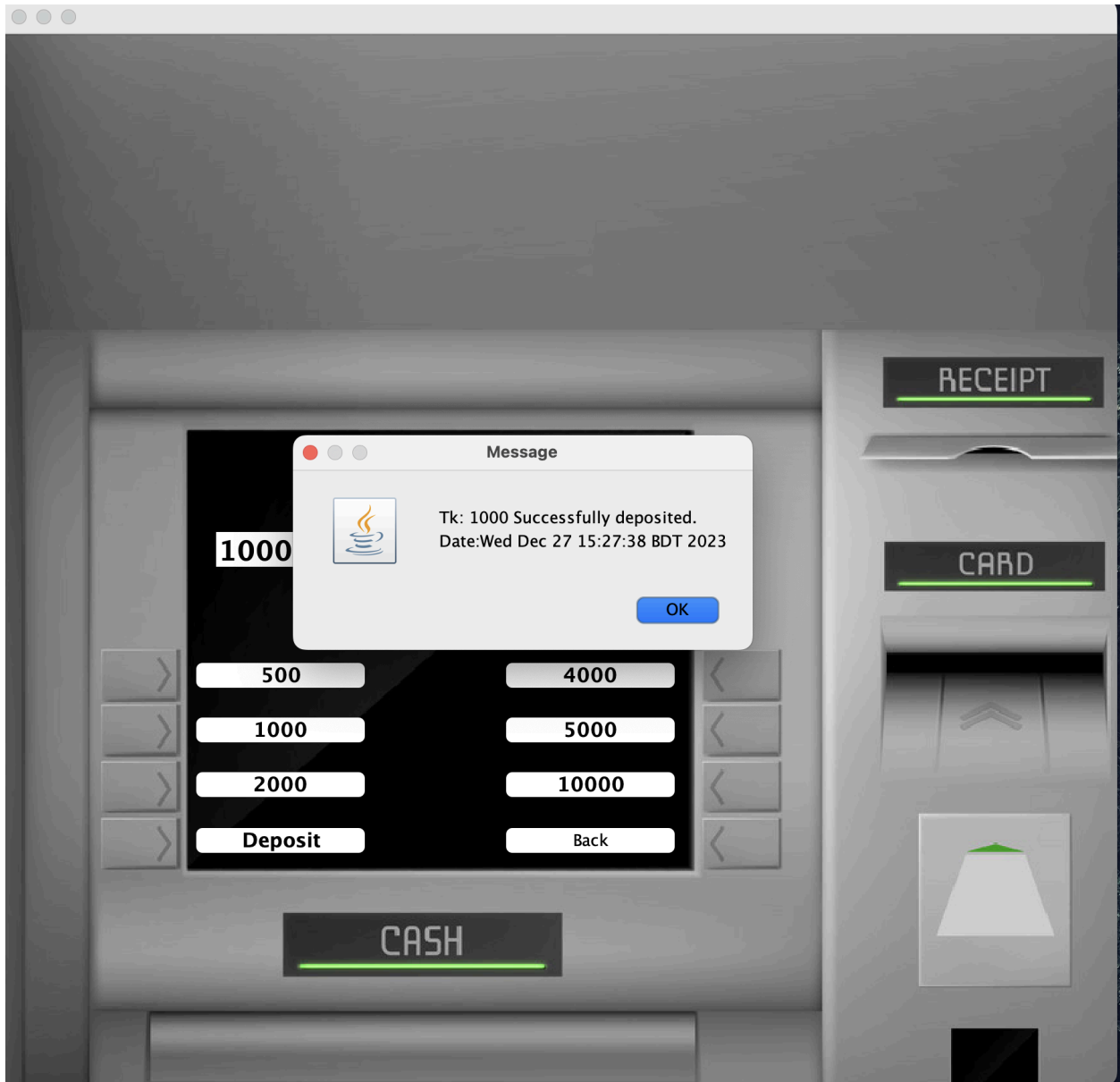
Incorrect Card Number or Pin

OK











## Mini Statement



# East West Bank

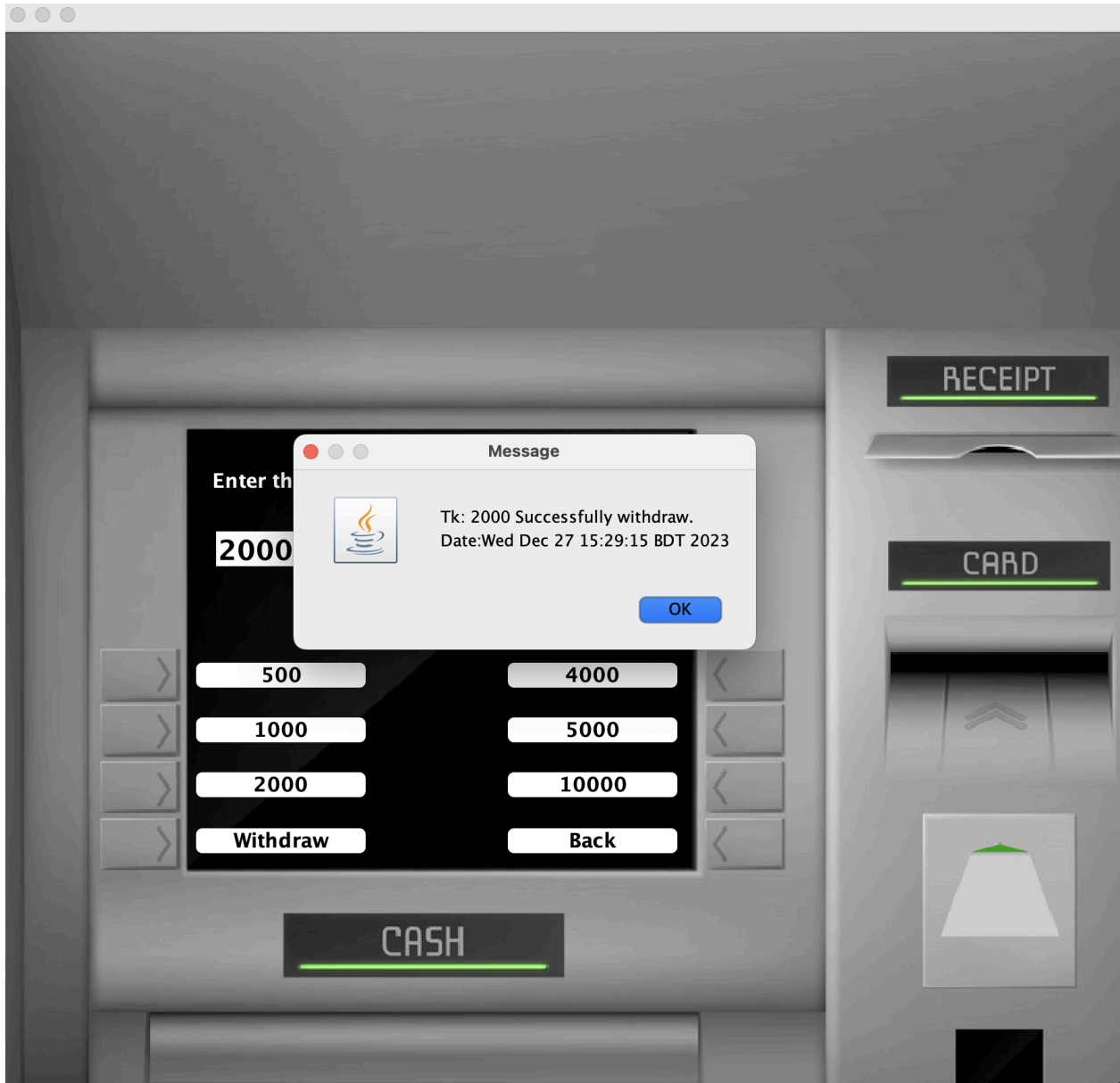
**Card NO :1234567890**

Date: 27-12-2023

**27-12-2023: Deposit of 1000Total:100298**









## Mini Statement



# East West Bank

**Card NO :1234567890**

Date: 27-12-2023

**27-12-2023: Deposit of 1000Total:100298**

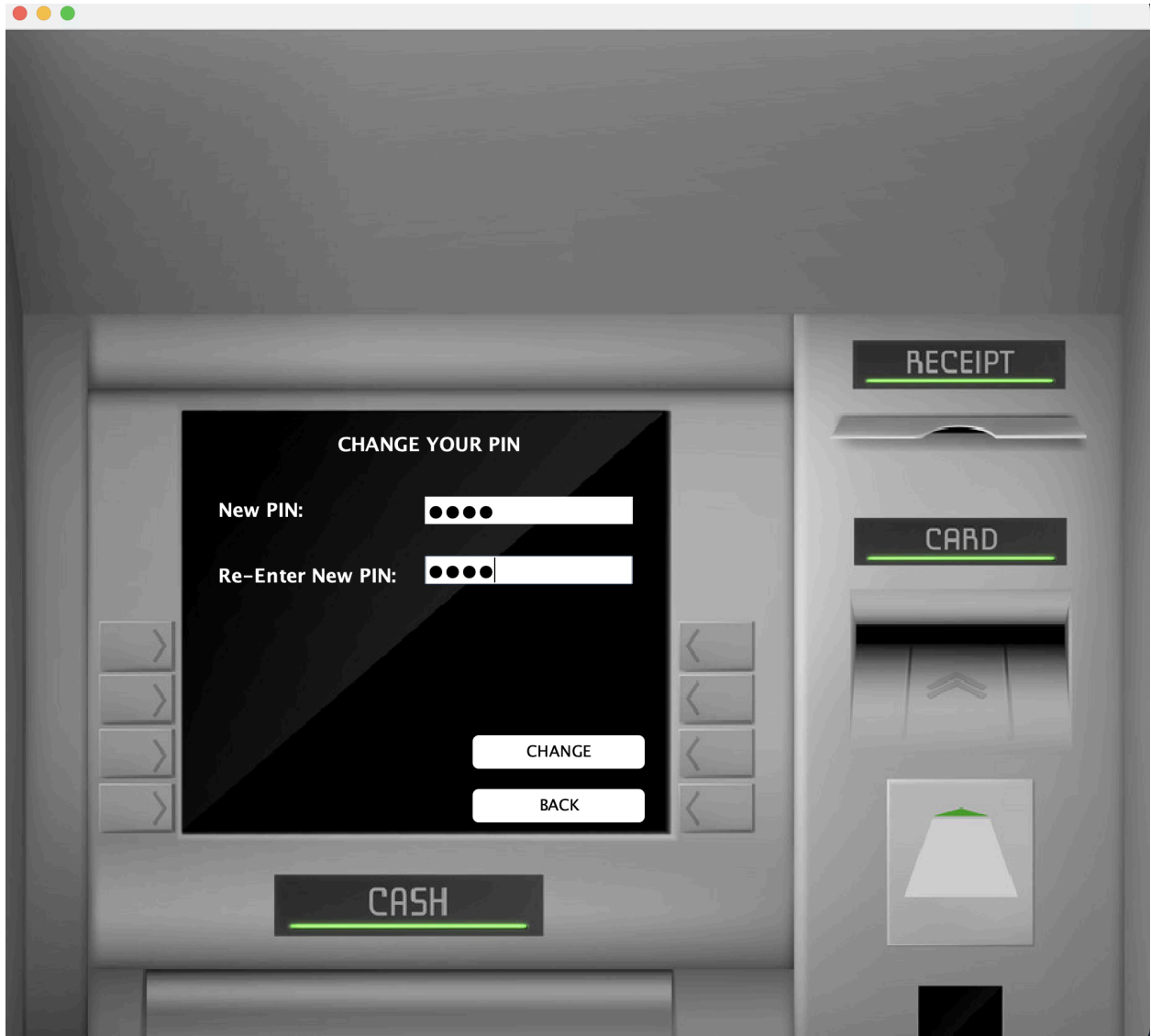
**27-12-2023: Withdrawal of 2000 Total:98298**

**27-12-2023: Withdrawal of 2000 Total:96298**









## Conclusion

This project is developed to nurture the needs of a user in a banking sector by embedding all the tasks of transactions taking place in a bank. Future version of this project will still be much enhanced than the current version. Writing and depositing checks are perhaps the most fundamental ways to move money in and out of a checking account, but advancements in technology have added ATM and debit card transactions. All banks have rules about how long it takes to access your deposits, how many debit card transactions you're allowed in a day, and how much cash you can withdraw from an ATM. Access to the balance in your checking account can also be limited by businesses that place holds on your funds.

Banks are providing internet banking services also so that the customers can be attracted. By asking the bank employs we came to know that maximum numbers of internet bank account holders are youth and business man. Online banking is an innovative tool that is fast becoming a necessity. It is a successful strategic weapon for banks to remain profitable in a volatile and competitive marketplace of today. If proper training should be given to customer by the bank employs to open an account will be beneficial secondly the website should be made friendlier from where the customers can directly make and access their accounts.

Thus, the ATM Management System is developed and executed successfully.