ATM Simulator Project - Source Code

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

public class ATMSimulator extends JFrame implements ActionListener {
    private CardLayout cardLayout;
    private JPanel mainPanel, loginPanel, menuPanel, transactionPanel;
    private JTextField pinField, amountField;
    private JLabel balanceLabel, messageLabel;
    private double balance = 1000.0; // Initial balance
    private final String correctPIN = "1234"; // sample PIN

    public ATMSimulator() {
        setTitle("ATM Simulator");
        setSize(400, 350);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        setResizable(false);

        cardLayout = new CardLayout();
        mainPanel = new JPanel(cardLayout);

        // Panels
        loginPanel = createLoginPanel();
        menuPanel = createMenuPanel();
        transactionPanel = new JPanel();

        mainPanel.add(loginPanel, "Login");
        mainPanel.add(menuPanel, "Menu");

        add(mainPanel);
        setVisible(true);
    }

    private JPanel createLoginPanel() {
        JPanel panel = new JPanel();
        panel.setLayout(new GridBagLayout());
        panel.setBackground(new Color(224, 255, 255));

        JLabel title = new JLabel("Welcome to ATM Simulator");
        title.setFont(new Font("Segoe UI", Font.BOLD, 18));
        JLabel pinLabel = new JLabel("Enter PIN:");
        pinField = new JPasswordField(10);
        JButton loginBtn = new JButton("Login");

        loginBtn.addActionListener(this);
        loginBtn.setBackground(new Color(0, 153, 153));
        loginBtn.setForeground(Color.WHITE);

        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);

        gbc.gridx = 0; gbc.gridy = 0; gbc.gridwidth = 2;
        panel.add(title, gbc);

        gbc.gridy++;
        gbc.gridwidth = 1;
        panel.add(pinLabel, gbc);

        gbc.gridx = 1;
        panel.add(pinField, gbc);

        gbc.gridy++;
        gbc.gridx = 0; gbc.gridwidth = 2;
        panel.add(loginBtn, gbc);

        return panel;
    }

    private JPanel createMenuPanel() {
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(6, 1, 10, 10));
        panel.setBorder(BorderFactory.createEmptyBorder(20, 60, 20, 60));
```

```
        panel.setBackground(new Color(240, 248, 255));

        JLabel title = new JLabel("ATM Main Menu", SwingConstants.CENTER);
        title.setFont(new Font("Segoe UI", Font.BOLD, 18));

        JButton checkBalanceBtn = new JButton("Check Balance");
        JButton depositBtn = new JButton("Deposit Money");
        JButton withdrawBtn = new JButton("Withdraw Money");
        JButton exitBtn = new JButton("Exit");

        checkBalanceBtn.addActionListener(this);
        depositBtn.addActionListener(this);
        withdrawBtn.addActionListener(this);
        exitBtn.addActionListener(this);

        panel.add(title);
        panel.add(checkBalanceBtn);
        panel.add(depositBtn);
        panel.add(withdrawBtn);
        panel.add(exitBtn);

        return panel;
    }

    private void showTransaction(String type) {
        transactionPanel.removeAll();
        transactionPanel.setLayout(new GridBagLayout());
        transactionPanel.setBackground(new Color(255, 250, 240));

        JLabel label = new JLabel(type + " Amount:");
        label.setFont(new Font("Segoe UI", Font.PLAIN, 16));
        amountField = new JTextField(10);
        JButton submitBtn = new JButton("Submit");
        JButton backBtn = new JButton("Back");
        messageLabel = new JLabel("");
        messageLabel.setFont(new Font("Segoe UI", Font.BOLD, 14));

        submitBtn.addActionListener(e -> handleTransaction(type));
        backBtn.addActionListener(e -> cardLayout.show(mainPanel, "Menu"));

        GridBagConstraints gbc = new GridBagConstraints();
        gbc.insets = new Insets(10, 10, 10, 10);
        gbc.gridx = 0; gbc.gridy = 0; transactionPanel.add(label, gbc);
        gbc.gridx = 1; transactionPanel.add(amountField, gbc);
        gbc.gridx = 0; gbc.gridy = 1; transactionPanel.add(submitBtn, gbc);
        gbc.gridx = 1; transactionPanel.add(backBtn, gbc);
        gbc.gridx = 0; gbc.gridy = 2; gbc.gridwidth = 2; transactionPanel.add(messageLabel, gbc);

        mainPanel.add(transactionPanel, "Transaction");
        cardLayout.show(mainPanel, "Transaction");
        mainPanel.revalidate();
        mainPanel.repaint();
    }

    private void handleTransaction(String type) {
        try {
            double amount = Double.parseDouble(amountField.getText());
            if (amount <= 0) {
                messageLabel.setText("Enter valid amount!");
                messageLabel.setForeground(Color.RED);
                return;
            }

            if (type.equals("Deposit")) {
                balance += amount;
                messageLabel.setText("Deposited ■" + amount + " successfully!");
                messageLabel.setForeground(Color.GREEN);
            } else if (type.equals("Withdraw")) {
                if (amount > balance) {
                    messageLabel.setText("Insufficient balance!");
                    messageLabel.setForeground(Color.RED);
                } else {
                    balance -= amount;
                    messageLabel.setText("Withdrawn ■" + amount + " successfully!");
                    messageLabel.setForeground(Color.GREEN);
                }
            }
```

```
            } catch (NumberFormatException e) {
                messageLabel.setText("Please enter a valid number!");
                messageLabel.setForeground(Color.RED);
            }
        }

        @Override
        public void actionPerformed(ActionEvent e) {
            String action = e.getActionCommand();

            switch (action) {
                case "Login":
                    String enteredPIN = pinField.getText();
                    if (enteredPIN.equals(correctPIN)) {
                        cardLayout.show(mainPanel, "Menu");
                    } else {
                        JOptionPane.showMessageDialog(this, "Invalid PIN! Try 1234", "Error", JOptionPane
                    }
                    break;

                case "Check Balance":
                    JOptionPane.showMessageDialog(this, "Your Balance: ■" + balance, "Balance", JOptionP
                    break;

                case "Deposit Money":
                    showTransaction("Deposit");
                    break;

                case "Withdraw Money":
                    showTransaction("Withdraw");
                    break;

                case "Exit":
                    JOptionPane.showMessageDialog(this, "Thank you for using ATM Simulator!");
                    System.exit(0);
                    break;
            }
        }

        public static void main(String[] args) {
            SwingUtilities.invokeLater(ATMSimulator::new);
        }
    }
```