package src.bank.management.system;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.ResultSet;

import java.util.Date;

/\*\*

\* FastCash class to handle fast cash withdrawal operations in the bank management system.

\*/

public class FastCash extends JFrame implements ActionListener {

private JButton withdraw100Button, withdraw500Button, withdraw1000Button, withdraw2000Button, withdraw5000Button, withdraw10000Button, backButton;

private String pin;

/\*\*

\* Constructor to initialize the fast cash interface.

\*

\* @param pin User's PIN for authentication.

\*/

public FastCash(String pin) {

this.pin = pin;

// Setting up the background image

ImageIcon backgroundIcon = new ImageIcon(ClassLoader.getSystemResource("icon/atm2.png"));

Image scaledImage = backgroundIcon.getImage().getScaledInstance(1550, 830, Image.SCALE\_DEFAULT);

ImageIcon scaledIcon = new ImageIcon(scaledImage);

JLabel backgroundLabel = new JLabel(scaledIcon);

backgroundLabel.setBounds(0, 0, 1550, 830);

add(backgroundLabel);

// Adding labels

JLabel selectAmountLabel = new JLabel("SELECT WITHDRAWAL AMOUNT");

selectAmountLabel.setBounds(445, 180, 700, 35);

selectAmountLabel.setForeground(Color.WHITE);

selectAmountLabel.setFont(new Font("System", Font.BOLD, 23));

backgroundLabel.add(selectAmountLabel);

// Adding buttons for fast cash options

withdraw100Button = createButton("Rs. 100", 410, 274);

backgroundLabel.add(withdraw100Button);

withdraw500Button = createButton("Rs. 500", 700, 274);

backgroundLabel.add(withdraw500Button);

withdraw1000Button = createButton("Rs. 1000", 410, 318);

backgroundLabel.add(withdraw1000Button);

withdraw2000Button = createButton("Rs. 2000", 700, 318);

backgroundLabel.add(withdraw2000Button);

withdraw5000Button = createButton("Rs. 5000", 410, 362);

backgroundLabel.add(withdraw5000Button);

withdraw10000Button = createButton("Rs. 10000", 700, 362);

backgroundLabel.add(withdraw10000Button);

backButton = createButton("BACK", 700, 406);

backgroundLabel.add(backButton);

// JFrame settings

setLayout(null);

setSize(1550, 1080);

setLocation(0, 0);

setVisible(true);

}

/\*\*

\* Creates a JButton with specified properties.

\*

\* @param text The text to be displayed on the button.

\* @param x The x-coordinate of the button.

\* @param y The y-coordinate of the button.

\* @return The created JButton.

\*/

private JButton createButton(String text, int x, int y) {

JButton button = new JButton(text);

button.setForeground(Color.WHITE);

button.setBackground(new Color(65, 125, 128));

button.setBounds(x, y, 150, 35);

button.addActionListener(this);

return button;

}

/\*\*

\* Handles button click events.

\*

\* @param e Action event.

\*/

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == backButton) {

setVisible(false);

new MainClass(pin);

} else {

String amount = ((JButton) e.getSource()).getText().substring(4);

Connn conn = new Connn();

Date date = new Date();

try {

ResultSet resultSet = conn.statement.executeQuery("SELECT \* FROM bank WHERE pin = '" + pin + "'");

int balance = 0;

// Calculate current balance

while (resultSet.next()) {

if ("Deposit".equals(resultSet.getString("type"))) {

balance += Integer.parseInt(resultSet.getString("amount"));

} else {

balance -= Integer.parseInt(resultSet.getString("amount"));

}

}

// Check if balance is sufficient

if (balance < Integer.parseInt(amount)) {

JOptionPane.showMessageDialog(null, "Insufficient Balance");

return;

}

// Update the balance in the database

conn.statement.executeUpdate("INSERT INTO bank VALUES('" + pin + "', '" + date + "', 'Withdrawal', '" + amount + "')");

JOptionPane.showMessageDialog(null, "Rs. " + amount + " Debited Successfully");

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, "An error occurred: " + ex.getMessage());

} finally {

setVisible(false);

new MainClass(pin);

}

}

}

public static void main(String[] args) {

new FastCash("");

}

}