Name: Mustafa Khan

Reg No: BCY243029

Project code (Java) of Train booking System

import java.awt.event.\*;

import javax.swing.\*;

import java.awt.\*;

import java.io.\*;

import java.util.\*;

public class TrainBookingGUI {

private JFrame frame;

private CardLayout cardLayout;

private JPanel mainPanel;

// Registration Fields

private JTextField regUsernameField;

private JPasswordField regPasswordField;

private JTextField regAccountField, regAccNameField;

// Login Fields

private JTextField loginUsernameField;

private JPasswordField loginPasswordField;

// Booking Fields

private JComboBox<String> routeBox, ticketBox, scheduleBox;

private JTextField quantityField;

private JTextArea outputArea;

private final String[] routes = {

"Lahore to Karachi", "Rawalpindi to Lahore", "Peshawar to Lahore",

"Lahore to Multan", "Karachi to Quetta", "Islamabad to Faisalabad"

};

private final String[] schedules = {

"10:00 AM", "1:00 PM", "3:00 PM", "5:00 PM", "7:00 PM"

};

public TrainBookingGUI() {

frame = new JFrame("Online Train Booking System");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(600, 500);

frame.setLayout(new BorderLayout());

cardLayout = new CardLayout();

mainPanel = new BackgroundPanel();

mainPanel.setLayout(cardLayout);

// panels

mainPanel.add(createRegistrationPanel(), "Register");

mainPanel.add(createLoginPanel(), "Login");

mainPanel.add(createBookingPanel(), "Book");

frame.add(mainPanel, BorderLayout.CENTER);

frame.add(createNavigationPanel(), BorderLayout.SOUTH);

frame.setVisible(true);

}

private JPanel createNavigationPanel() {

JPanel navPanel = new JPanel();

JButton registerBtn = new JButton("Register");

JButton loginBtn = new JButton("Login");

registerBtn.addActionListener(e -> cardLayout.show(mainPanel, "Register"));

loginBtn.addActionListener(e -> cardLayout.show(mainPanel, "Login"));

navPanel.add(registerBtn);

navPanel.add(loginBtn);

return navPanel;

}

private JPanel createRegistrationPanel() {

JPanel regPanel = new JPanel(new GridLayout(5, 2, 20, 2));

regUsernameField = new JTextField(20);

regPasswordField = new JPasswordField(20);

regAccountField = new JTextField(20);

regAccNameField = new JTextField(20);

regPanel.add(new JLabel("Username:"));

JPanel usernamePanel = new JPanel(new FlowLayout());

usernamePanel.add(regUsernameField);

regPanel.add(usernamePanel);

regPanel.add(new JLabel("Password:"));

JPanel passwordPanel = new JPanel(new FlowLayout());

passwordPanel.add(regPasswordField);

regPanel.add(passwordPanel);

regPanel.add(new JLabel("Account #:"));

JPanel accountPanel = new JPanel(new FlowLayout());

accountPanel.add(regAccountField);

regPanel.add(accountPanel);

regPanel.add(new JLabel("Account Name:"));

JPanel accNamePanel = new JPanel(new FlowLayout());

accNamePanel.add(regAccNameField);

regPanel.add(accNamePanel);

JButton registerBtn = new JButton("Register");

registerBtn.addActionListener(e -> registerUser());

JPanel buttonPanel = new JPanel(new FlowLayout());

buttonPanel.add(registerBtn);

regPanel.add(buttonPanel);

return regPanel;

}

private JPanel createLoginPanel() {

JPanel loginPanel = new JPanel(new GridLayout(3, 2, 20, 5));

loginUsernameField = new JTextField(20);

loginPasswordField = new JPasswordField(20);

loginPanel.add(new JLabel("Username:"));

JPanel usernamePanel = new JPanel(new FlowLayout());

usernamePanel.add(loginUsernameField);

loginPanel.add(usernamePanel);

loginPanel.add(new JLabel("Password:"));

JPanel passwordPanel = new JPanel(new FlowLayout());

passwordPanel.add(loginPasswordField);

loginPanel.add(passwordPanel);

JButton loginBtn = new JButton("Login");

loginBtn.addActionListener(e -> loginUser());

JPanel buttonPanel = new JPanel(new FlowLayout());

buttonPanel.add(loginBtn);

loginPanel.add(buttonPanel);

return loginPanel;

}

private JPanel createBookingPanel() {

JPanel bookingPanel = new JPanel(new GridLayout(6, 2, 20, 5));

quantityField = new JTextField(20);

routeBox = new JComboBox<>(routes);

ticketBox = new JComboBox<>(new String[] { "Standard", "Premium", "VIP" });

scheduleBox = new JComboBox<>(schedules);

bookingPanel.add(new JLabel("Route:"));

JPanel routePanel = new JPanel(new FlowLayout());

routePanel.add(routeBox);

bookingPanel.add(routePanel);

bookingPanel.add(new JLabel("Ticket Type:"));

JPanel ticketPanel = new JPanel(new FlowLayout());

ticketPanel.add(ticketBox);

bookingPanel.add(ticketPanel);

bookingPanel.add(new JLabel("Schedule:"));

JPanel schedulePanel = new JPanel(new FlowLayout());

schedulePanel.add(scheduleBox);

bookingPanel.add(schedulePanel);

bookingPanel.add(new JLabel("Quantity:"));

JPanel quantityPanel = new JPanel(new FlowLayout());

quantityPanel.add(quantityField);

bookingPanel.add(quantityPanel);

JButton bookBtn = new JButton("Book Ticket");

bookBtn.addActionListener(e -> bookTicket());

JPanel buttonPanel = new JPanel(new FlowLayout());

buttonPanel.add(bookBtn);

bookingPanel.add(buttonPanel);

outputArea = new JTextArea();

outputArea.setEditable(false);

bookingPanel.add(new JScrollPane(outputArea));

return bookingPanel;

}

class BackgroundPanel extends JPanel {

private Image backgroundImage;

public BackgroundPanel() {

this.backgroundImage = new ImageIcon(getClass().getResource("/resources/hq720.jpg")).getImage();

if (backgroundImage == null) {

System.out.println("Image not found!");

}

}

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

if (backgroundImage != null) {

g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);

}

}

}

private void registerUser() {

String username = regUsernameField.getText();

String password = new String(regPasswordField.getPassword());

String account = regAccountField.getText();

String accName = regAccNameField.getText();

try (BufferedWriter writer = new BufferedWriter(new FileWriter("users.txt", true))) {

writer.write(username + "," + password + "," + account + "," + accName);

writer.newLine();

JOptionPane.showMessageDialog(frame, "Registration Successful!");

clearRegistrationFields();

cardLayout.show(mainPanel, "Login");

} catch (IOException e) {

JOptionPane.showMessageDialog(frame, "Error saving user data.");

}

}

private void clearRegistrationFields() {

regUsernameField.setText("");

regPasswordField.setText("");

regAccountField.setText("");

regAccNameField.setText("");

}

private void loginUser() {

String username = loginUsernameField.getText();

String password = new String(loginPasswordField.getPassword());

try (BufferedReader reader = new BufferedReader(new FileReader("users.txt"))) {

String line;

boolean loggedIn = false;

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

String[] userData = line.split(",");

if (username.equals(userData[0]) && password.equals(userData[1])) {

loggedIn = true;

break;

}

}

if (loggedIn) {

JOptionPane.showMessageDialog(frame, "Login Successful!");

cardLayout.show(mainPanel, "Book");

} else {

JOptionPane.showMessageDialog(frame, "Invalid username or password.");

}

} catch (IOException e) {

JOptionPane.showMessageDialog(frame, "Error reading user data.");

}

}

private void bookTicket() {

String route = (String) routeBox.getSelectedItem();

String type = (String) ticketBox.getSelectedItem();

String bookingTime = (String) scheduleBox.getSelectedItem();

int qty;

try {

qty = Integer.parseInt(quantityField.getText());

if (qty <= 0)

throw new NumberFormatException();

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(frame, "Enter a valid quantity.");

return;

}

double basePrice = 1500 + new Random().nextInt(1000);

double surcharge;

switch (type) {

case "Premium":

surcharge = 500;

break;

case "VIP":

surcharge = 1000;

break;

default:

surcharge = 0;

break;

}

double total = (basePrice + surcharge) \* qty \* 0.8;

Ticket ticket = new Ticket(route, type, qty, total, bookingTime);

JOptionPane.showMessageDialog(frame, ticket.toString());

outputArea.setText(ticket.toString());

}

public static void main(String[] args) {

SwingUtilities.invokeLater(TrainBookingGUI::new);

}

// Ticket class

class Ticket {

private String route;

private String type;

private int quantity;

private double totalPrice;

private String bookingTime;

public Ticket(String route, String type, int quantity, double totalPrice, String bookingTime) {

this.route = route;

this.type = type;

this.quantity = quantity;

this.totalPrice = totalPrice;

this.bookingTime = bookingTime;

}

@Override

public String toString() {

return String.format(

"Ticket Details:\nRoute: %s\nTicket Type: %s\nQuantity: %d\nTotal Price: %.2f PKR\nBooking Time: %s",

route, type, quantity, totalPrice, bookingTime);

}

}

}