Import javax.swing.\*;

Import java.awt.\*;

Import java.awt.event.\*;

Import java.util.HashMap;

Class HotelManagementSystemWithPayment extends JFrame {

Private HashMap<String, Double> roomPrices = new HashMap<>(); // Room price storage

Private HashMap<String, String> bookings = new HashMap<>(); // Room to customer mapping

Public HotelManagementSystemWithPayment() { setTitle(“Hotel Management System with

Billing & Payment”); setSize(700, 500);

setDefaultCloseOperation(EXIT\_ON\_CLOSE); setLayout(new BorderLayout());

JLabel titleLabel = new JLabel(“Hotel Management System with Billing & Payment”,

JLabel.CENTER); titleLabel.setFont(new Font(“Arial”, Font.BOLD, 20)); add(titleLabel,

BorderLayout.NORTH);

JTabbedPane tabs = new JTabbedPane();

JPanel bookingPanel = new JPanel(new GridLayout(5, 2, 10, 10));

bookingPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

JLabel roomNumberLabel = new JLabel(“Room Number:”);

JTextField roomNumberField = new JTextField();

JLabel roomPriceLabel = new JLabel(“Room Price (per day):”);

JTextField roomPriceField = new JTextField();

JLabel customerNameLabel = new JLabel(“Customer Name:”);

JTextField customerNameField = new JTextField();

JButton bookButton = new JButton(“Book Room”);

JTextArea bookingLog = new JTextArea(5, 20);

bookingLog.setEditable(false);

bookingPanel.add(roomNumberLabel);

bookingPanel.add(roomNumberField);

bookingPanel.add(roomPriceLabel);

bookingPanel.add(roomPriceField);

bookingPanel.add(customerNameLabel);

bookingPanel.add(customerNameField);

bookingPanel.add(new JLabel());

bookingPanel.add(bookButton); bookingPanel.add(new

JLabel(“Booking Log:”)); bookingPanel.add(new

JScrollPane(bookingLog)); tabs.add(“Room Booking”,

bookingPanel);

JPanel billingPanel = new JPanel(new GridLayout(5, 2, 10, 10));

billingPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

JLabel billingRoomLabel = new JLabel(“Room Number:”);

JTextField billingRoomField = new JTextField();

JButton checkOutButton = new JButton(“Check-Out & Generate Bill”); JTextArea

billingLog = new JTextArea(5, 20);

billingLog.setEditable(false); billingPanel.add(billingRoomLabel);

billingPanel.add(billingRoomField); billingPanel.add(new JLabel());

billingPanel.add(checkOutButton); billingPanel.add(new

JLabel(“Billing Log:”)); billingPanel.add(new);

JScrollPane(billingLog)); tabs.add(“Billing”, billingPanel); add(tabs,

BorderLayout.CENTER); bookButton.addActionListener(e -> { String room

= roomNumberField.getText();

String customer = customerNameField.getText();

String priceText = roomPriceField.getText();

If (!room.isEmpty() && !customer.isEmpty() && !priceText.isEmpty()) { Try

{

Double price = Double.parseDouble(priceText);

roomPrices.put(room, price); bookings.put(room, customer);

bookingLog.append(“Room “ + room

+ “ booked by “ + customer + “ at $” + price

+ “ per day.\n”); roomNumberField.setText(“”);

customerNameField.setText(“”);

roomPriceField.setText(“”); } catch

(NumberFormatException ex) {

JOptionPane.showMessageDialog(this, “Invalid price entered!”, “Error”,

JOptionPane.ERROR\_MESSAGE)**;**

**}**

} else {

JOptionPane.showMessageDialog(this, “Please fill all fields!”, “Error”,

JOptionPane.ERROR\_MESSAGE);

}

});

checkOutButton.addActionListener(e -> { String

room = billingRoomField.getText();

If (bookings.containsKey(room)) {

String customer = bookings.get(room);

Double roomPrice = roomPrices.get(room);

String daysStayedText = JOptionPane.showInputDialog(this, “Enter the number of days stayed:”);

If (daysStayedText != null && !daysStayedText.isEmpty()) {

Try {

Int daysStayed = Integer.parseInt(daysStayedText);

Double totalBill = roomPrice \* daysStayed;

String creditCardNumber = JOptionPane.showInputDialog(this, “Enter Credit

Card Number:”);

String password = JOptionPane.showInputDialog(this, “Enter Credit Card

Password:”);

If (creditCardNumber != null && password != null &&

!creditCardNumber.isEmpty() && !password.isEmpty()) {

billingLog.append(“Room “ + room + “ checked out. Bill for “ + customer + “:

$” + totalBill + “\n”); billingLog.append(“Payment successful for “ + customer

+ “ using card ending in “ +

creditCardNumber.substring(creditCardNumber.length() – 4) + “\n”); Bookings.remove(room);

roomPrices.remove(room);

} else {

JOptionPane.showMessageDialog(this, “Payment failed. Please enter valid payment details!”,

“Error”, JOptionPane.ERROR\_MESSAGE);

}

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(this, “Invalid number of days entered!”,

“Error”, JOptionPane.ERROR\_MESSAGE);

}

}

14

} else {

JOptionPane.showMessageDialog(this, “Room not found or not booked!”, “Error”,

JOptionPane.ERROR\_MESSAGE);

} billingRoomField.setText(“”);

});

}

Public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

HotelManagementSystemWithPayment hms = new

HotelManagementSystemWithPayment();

Hms.setVisible(true) ;

}

}

}