**Project: OOP**

**Submitted to:**

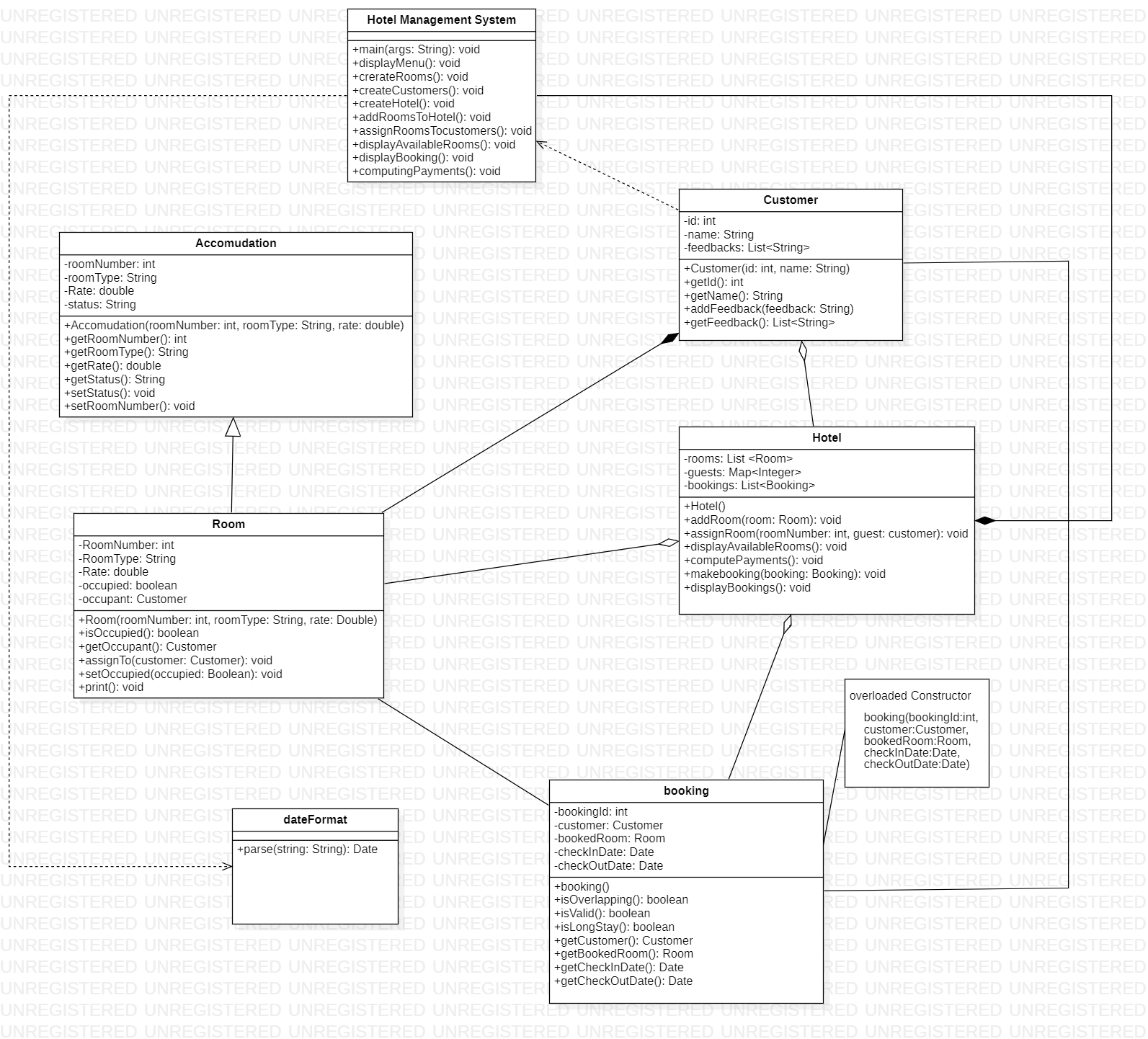
**Sir Arslan Tariq**

**Submitted by:**

**Maida Kosser (221400091)**

**Anika Shahid (221400095)**

**Section: C**

****

**Code:**

**public class Accomodation {**

private int roomNumber;

private String roomtype;

private String status;

private double rate;

// overloaded constructor

public Accomodation(int roomNumber, String roomtype, String status, double rate) {

this.roomtype = roomtype;

this.roomNumber = roomNumber;

this.rate = rate;

this.status = status;

}// Accomodation

// getter and setter Method

public int getRoomNumber() {

return roomNumber;

}// getroomnumber

public void setRoomNumber(int roomNumber) {

this.roomNumber = roomNumber;

}// getroomnumber

public String getRoomType() {

return roomtype;

}// get room type

public String getStatus() {

return status;

}// getStatus

public double getRate() {

return rate;

}// getrate

public void setStatus(String status) {

this.status = status;

}// setstatus

}

**public class Room extends Accomodation {**

private Boolean occupied;

private Customer occupant;

// Default constructor

public Room(int roomNumber, String roomtype, String status, double rate) {

super(roomNumber, roomtype, status, rate);

this.occupied = false;

this.occupant = new Customer(0, "");

}

// Getters and setters

public Boolean isOccupied() {

return occupied;

}

public void setOccupied(Boolean occupied) {

this.occupied = occupied;

}

public Customer getOccupant() {

return occupant;

}

public void AssignTo(Customer customer) {

this.occupant = customer; // customers assigns the room

this.occupied = true;

}// assignto

public void print() {

System.out.println();

System.out.println("occupied:" + occupied);

System.out.println("occupant: " + occupant.getName());

}

}

import java.util.ArrayList;

import java.util.List;

**public class Customer {**

private int id;

private String name;

private List<String> feedbacks;

public Customer(int id, String name) {

this.id = id;

this.name = name;

this.feedbacks = new ArrayList<>();

}

public int getID() { // customers id and name

return id;

}

public String getName() {

return name;

}

public void addFeedBack(String feedback) {

feedbacks.add(feedback); // customers get the feedback

}

public List<String> getFeedBack() {

return feedbacks;

}

public void put(int id2, Customer guest) {

}

}

import java.util.Date;

**public class Booking {**

private int bookingId;

private Customer customer;

private Room bookedRoom;

private Date checkInDate;

private Date checkOutDate;

public Booking(int bookingId, Customer customer, Room r, Date checkInDate, Date checkOutDate) {

this.bookingId = bookingId;

this.customer = customer;

this.bookedRoom = r;

this.checkInDate = checkInDate;

this.checkOutDate = checkOutDate;

}

public boolean isOverlapping(Booking otherBooking) {

return checkInDate.before(otherBooking.getCheckOutDate()) &&

otherBooking.getCheckInDate().before(checkOutDate);

}

public boolean isValid() {

Date currentDate = new Date();

return currentDate.before(checkInDate);

}

public boolean isLongStay() {

long duration = checkOutDate.getTime() - checkInDate.getTime();

return duration >= 7 \* 24 \* 60 \* 60 \* 1000;

}

public Customer getCustomer() {

return customer;

}

public Room getBookedRoom() { // customer booked the room

return bookedRoom;

}

public Date getCheckInDate() {

return checkInDate;

}

public Date getCheckOutDate() {

return checkOutDate;

}

}

import java.util.Date;

**public enum dateFormat {**

public static Date parse(String string) {

return null;

}

}

import java.util.Date;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

**public class Hotel {**

private List<Room> rooms;

private Map<Integer, Customer> guest;

private List<Booking> bookings;

public Hotel() {

rooms = new ArrayList<>();

guest = new HashMap<>();

bookings = new ArrayList<>();

}

public void addRoom(Room room) {

rooms.add(room);

System.out.println("Room " + room.getRoomNumber() + " has been added in the hotel.");

}

public void assignRoom(int roomNumber, Customer guest) {

for (Room room : rooms) {

if (room.getRoomNumber() == roomNumber) {

if (!room.isOccupied()) {

room.AssignTo(guest);

guest.put(guest.getID(), guest);

System.out.println(

"Room " + roomNumber + "(" + room.getRoomType() + ") " + "assigned to " + guest.getName()

+ ".");

} else {

System.out.println("Room " + roomNumber + " is already booked.");

}

return;

}

}

System.out.println("Room " + roomNumber + "does not exist in te Hotel.");

}

public void displayAvailableRoooms() {

System.out.println("\nAvailable Rooms are:");

for (Room room : rooms) {

if (!room.isOccupied()) {

System.out.println(

"Room " + room.getRoomNumber() + " (" + room.getRoomType() + ")" + " for $" + room.getRate());

}

}

}

public void computerPayements() {

System.out.println("\n\npayments pending: ");

for (Booking booking : bookings) {

Customer customer = booking.getCustomer();

Room room = booking.getBookedRoom();

double rate = room.getRate();

Date CheckInDate = booking.getCheckInDate();

Date CheckOutDate = booking.getCheckOutDate();

Long duration = CheckOutDate.getTime() - CheckInDate.getTime();

duration = duration / (1000 \* 60 \* 60 \* 24);

double totalPayement = duration \* rate;

System.out.println(customer.getName() + " has to pay $" + totalPayement + " (The rent for " + duration

+ " days for room " + room.getRoomType() + ")");

}

}

public void makeBooking(Booking booking) {

bookings.add(booking);

}

public void displayBookings() {

System.out.println();

System.out.println("Bookings :");

SimpleDateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd");

for (Booking booking : bookings) {

Customer customer = booking.getCustomer();

Room room = booking.getBookedRoom();

String checkInDate = dateformat.format(booking.getCheckInDate());

String checkOutDate = dateformat.format(booking.getCheckOutDate());

System.out.println(customer.getName() + " has booked room "

+ room.getRoomNumber() + " from " + checkInDate + " to " + checkOutDate + ".");

}

System.out.println();

}

}

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

**public class HotelManagementSystem {**

public static void main(String[] args) throws ParseException {

System.out.println("\n\_\_\_\_\_\_\_\_Welcome to Hotel Management\_\_\_\_\_\_\_\_\n");

Customer customer1 = new Customer(1, "Jenny");

Customer customer2 = new Customer(2, "izlan");

Customer customer3 = new Customer(2, "izlan");

Room room101 = new Room(101, "Standard", "LowerClass", 50.0);

Room room102 = new Room(102, "Luxury", "MiddleClass", 100.0);

Room room103 = new Room(103, "Delux", "HighClass", 150.0);

Hotel hotel = new Hotel();

hotel.addRoom(room101);

hotel.addRoom(room102);

hotel.addRoom(room103);

hotel.displayAvailableRoooms();

SimpleDateFormat daterformat = new SimpleDateFormat("dd-mm-yyyy");

Date CheckInDate1 = daterformat.parse("21-08-2023");

Date CheckOutDate1 = daterformat.parse("25-08-2023");

Date CheckInDate2 = daterformat.parse("13-08-2023");

Date CheckOutDate2 = daterformat.parse("16-08-2023");

Booking booking1 = new Booking(1, customer1, room101, CheckInDate1, CheckOutDate1);

Booking booking2 = new Booking(2, customer2, room102, CheckInDate2, CheckOutDate2);

hotel.makeBooking(booking1);

hotel.makeBooking(booking2);

hotel.displayBookings();

hotel.assignRoom(101, customer1);

hotel.assignRoom(103, customer2);

hotel.computerPayements();

hotel.displayAvailableRoooms();

}

}