MANIPAL UNIVERSITY JAIPUR

Department of Computer Science & Engineering

III-SEMESTER CSE
CS2131 (OOP Lab)

OOP Lab Assignment#5
Hotel Booking

Name: Prajjwal Pathak
Registration No.: 199301127
Section: CSE-B

Users want to book hotel. For the same they must sign up with Name, age, id, and password which shall be stored in file. As soon as users sign in, their id and password shall be verified first, after successful sign in user selects city (fetch from files) and dates (use Exception handling for wrong dates), which shows him list of all hotels in the city with availability and prices. When user selects any hotel, System asks him for more details like number of rooms, number of persons, name, and age of every person, system also ask that whether he is one of the travelers or not, if yes then his details are automatically included in the traveler list (use inheritance). Finally, system generates bill for the booking with unique booking Id and a discount of Rs. 1000 if net amount exceeds Rs. 10000. when multiple users log in simultaneously (use multiple command prompt), manage all users using Multithreading so that no user would be interrupted.

```
import java.util.*;


import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.text.ParseException;
```

```
import java.text.SimpleDateFormat;


class Customer {
    String name;
    int age;
    String username;
    Vector<Booking> bookings = new Vector<Booking>();
    static String filePath = "D:\\Software\\IntelliJ IDEA Projects\\Raw Java
Programs\\src\\users.txt";


    Customer(String name, int age, String username) {
        this.name = name;
        this.age = age;
        this.username = username;
    }


    public static synchronized void signUp() {
        String name;
        int age;
        String username;


        String password;


        Scanner scan = new Scanner(System.in);
        System.out.println("Enter username: ");
        username = scan.nextLine();
        System.out.println("Checking to see if username exists...");
        BufferedReader bufferedReader;
        BufferedWriter bufferedWriter;
        try {
            bufferedReader = new BufferedReader(new FileReader(filePath));
            String line;
```

```
            boolean usernameExists = false;
            while ((line = bufferedReader.readLine()) != null) {
                line = line.substring(line.indexOf("username:") + 9,
line.indexOf(",", line.indexOf("username:")));
                System.out.println("DEBUG: " + line);
                if (line.equals(username)) {
                    usernameExists = true;
                    break;
                }
            }
            if (usernameExists) {
                System.out.println("[!] Username exists! Please try again.");
            } else {
                System.out.println("Enter name: ");
                name = scan.nextLine();
                System.out.println("Enter age: ");
                age = scan.nextInt();
                System.out.println("Enter password: ");
                password = scan.nextLine();
                password = scan.nextLine();
                bufferedWriter = new BufferedWriter(new FileWriter(filePath,
true));
                bufferedReader = new BufferedReader(new FileReader(filePath));
                if (bufferedReader.readLine() != null) {
                    bufferedWriter.newLine();
                }
                bufferedWriter.write(
                        "username:" + username + "," + "name:" + name + "," +
"age:" + age + "," + "password:" + password + ",");
                System.out.println("[+] User registered successfully");
                bufferedWriter.close();
            }
```

```
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static Customer signIn() {
        String username;
        String password;
        String file_username;
        String file_password;
        int file_age = 0;
        String file_name = "";
        boolean correctPassword = false;
        Scanner scan = new Scanner(System.in);
        BufferedReader bufferedReader;
        System.out.println("Enter username: ");
        username = scan.nextLine();
        System.out.println("Enter password: ");
        password = scan.nextLine();
        try {
            bufferedReader = new BufferedReader(new FileReader(filePath));
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                file_username = line.substring(line.indexOf("username:") + 9,
line.indexOf(",", line.indexOf("username:")));
                file_password = line.substring(line.indexOf("password:") + 9,
line.indexOf(",", line.indexOf("password:")));
                if (username.equals(file_username)) {
                    System.out.println("[#] User Found");
                    file_age = Integer
```

```
                            .parseInt(line.substring(line.indexOf("age:") + 4,
line.indexOf(",", line.indexOf("age:"))));
                    file_name  =  line.substring(line.indexOf("name:")  +  5,
line.indexOf(",", line.indexOf("name:")));
                    while (true) {
                        if (password.equals(file_password)) {
                            correctPassword = true;
                            System.out.println("[.] Logging In");
                            break;
                        }
                        System.out.println("[X]    Please    enter    correct
password!");
                        password = scan.nextLine();
                    }
                }
            }
            bufferedReader.close();
            if (!correctPassword) {
                System.out.println("[!] User Not Found");
                return null;
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        if (correctPassword) {
            return new Customer(file_name, file_age, username);
        } else {
            return null;
        }
    }
```

```
    public static Customer getCustomer(String username) {
        int age = 0;
        String name = "";
        String file_username;
        BufferedReader bufferedReader;
        Customer customer = null;
        try {
            bufferedReader = new BufferedReader(new FileReader(filePath));
            String line;
            while ((line = bufferedReader.readLine()) != null) {
                file_username = line.substring(line.indexOf("username:") + 9,
line.indexOf(",", line.indexOf("username:")));
                if (username.equals(file_username)) {
                    age = Integer.parseInt(line.substring(line.indexOf("age:")
+ 4, line.indexOf(",", line.indexOf("age:"))));
                    name   =   line.substring(line.indexOf("name:")   +   5,
line.indexOf(",", line.indexOf("name:")));
                }
                customer = new Customer(name, age, username);
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return customer;
    }
}

class Hotel {
    String name;
    String city;
    int price; // per room
```

```
    int totalRooms;
    int vacantRooms;
    Vector<Customer> customers = new Vector<Customer>();


    public static Vector<Hotel> initHotels() {
        Vector<Hotel> hotels = new Vector<Hotel>();
        BufferedReader bufferedReader;
        String filePath = "D:\\Software\\IntelliJ IDEA Projects\\Raw Java
Programs\\src\\hotels.txt";
        String line;
        Hotel hotel;
        try {
            bufferedReader = new BufferedReader(new FileReader(filePath));
            while ((line = bufferedReader.readLine()) != null) {
                hotel = new Hotel();
                hotel.city = line;
                line = bufferedReader.readLine();
                hotel.name = line;
                line = bufferedReader.readLine();
                hotel.price = Integer.parseInt(line);
                line = bufferedReader.readLine();
                hotel.totalRooms = Integer.parseInt(line);
                line = bufferedReader.readLine();
                hotel.vacantRooms = Integer.parseInt(line);
                while (!(line = bufferedReader.readLine()).equals("-")) {
                    hotel.customers.add(Customer.getCustomer(line));
                }
                hotels.add(hotel);
            }
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
```

```
            e.printStackTrace();
        }
        return hotels;
    }
}

class Booking {
    String id;
    int numOfRooms;
    int numOfPeople;
    Date startDate;
    Date endDate;
    Hotel hotel;

    public static synchronized Booking setBooking(Hotel hotel) {
        Scanner scan = new Scanner(System.in);
        SimpleDateFormat dateFormat = new SimpleDateFormat("dd-MM-yyyy");
        String inputDate;
        boolean error = false;
        Booking booking = new Booking();
        booking.hotel = hotel;
        System.out.println("Enter number of rooms: ");
        booking.numOfRooms = scan.nextInt();
        if (hotel.vacantRooms < booking.numOfRooms) {
            System.out.println("[X] Rooms not available");
            return null;
        }
        System.out.println("Enter number of people: ");
        booking.numOfPeople = scan.nextInt();

        try {
            System.out.println("Enter start date (dd-MM-yyyy): ");
```

```
            inputDate = scan.nextLine();
            inputDate = scan.nextLine();
            booking.startDate = dateFormat.parse(inputDate);
            System.out.println("Enter end date (dd-MM-yyyy): ");
            inputDate = scan.nextLine();
            booking.endDate = dateFormat.parse(inputDate);
        } catch (ParseException e) {
            error = true;
            e.printStackTrace();
        }
        booking.id = UUID.randomUUID().toString();
        if (error) {
            return null;
        } else {
            BufferedWriter bufferedWriter;
            String filePath = "D:\\Software\\IntelliJ IDEA Projects\\Raw Java
Programs\\src\\bookings.txt";
            try {
                bufferedWriter = new BufferedWriter(new FileWriter(filePath,
true));
                bufferedWriter.write(booking.id);
                bufferedWriter.newLine();
                bufferedWriter.write(Integer.toString(booking.numOfRooms));
                bufferedWriter.newLine();
                bufferedWriter.write(Integer.toString(booking.numOfPeople));
                bufferedWriter.newLine();
                bufferedWriter.write(booking.startDate.toString());
                bufferedWriter.newLine();
                bufferedWriter.write(booking.endDate.toString());
                bufferedWriter.newLine();
                bufferedWriter.write(booking.hotel.name);
                bufferedWriter.newLine();
                bufferedWriter.write("-");
```

```
                bufferedWriter.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return booking;
        }
    }
}

public class BookingSystem {
    static Vector<Hotel> hotels = new Vector<Hotel>();
    static Customer currentUser;

    public static void clearScreen() {
        // Clears Screen in java
        try {
            if (System.getProperty("os.name").contains("Windows"))
                new                ProcessBuilder("cmd",                "/c",
"cls").inheritIO().start().waitFor();
            else
                Runtime.getRuntime().exec("clear");
        } catch (IOException | InterruptedException ex) {
        }
    }

    public static void showHotels() {
        for (int i = 0; i < hotels.size(); i++) {
            System.out.println((i+1) + ". " + hotels.get(i).city + " - " +
hotels.get(i).name);
        }
    }

    public static void bookRooms() {
```

```
        Scanner scan = new Scanner(System.in);
        showHotels();
        int choiceInt = scan.nextInt();
        if(choiceInt > hotels.size() || choiceInt < 1) {
            System.out.println("[!] Invalid Input");
            System.exit(0);
        }
        Hotel selectedHotel = hotels.get(choiceInt - 1);
        clearScreen();
        System.out.println("------------------------");
        System.out.println("City: " + selectedHotel.city);
        System.out.println("Name: " + selectedHotel.name);
        System.out.println("Price of a Room (per day): " +
selectedHotel.price);
        System.out.println("Total Rooms: " + selectedHotel.totalRooms);
        System.out.println("Vacant Rooms: " + selectedHotel.vacantRooms);
        System.out.println("------------------------");
        System.out.println("Do you want to book rooms? [Y/N]");
        String choice = scan.nextLine();
        choice = scan.nextLine();
        if(!choice.equals("y") && !choice.equals("Y")) {
            return;
        }
        currentUser.bookings.add(Booking.setBooking(selectedHotel));
        System.err.println("[+] Room Booked");
    }

    public static void main(String[] args) {
        String choice;
        int choiceInt;
        Scanner scan = new Scanner(System.in);
        hotels = Hotel.initHotels();
```

```
        System.out.println("1. Sign In\n2. Sign Up");
        choice = scan.nextLine();
        if (choice.equals("2")) {
            Customer.signUp();
            currentUser = Customer.signIn();
        } else if (choice.equals("1")) {
            currentUser = Customer.signIn();
        } else {
            System.out.println("[!] Invalid Input");
            System.exit(0);
        }
        System.out.println("Select Hotel" );
        System.out.println("--------------------------");
        bookRooms();
    }
}
```

```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe"
1. Sign In
2. Sign Up
2
Enter username:
prajjwal
Checking to see if username exists...
DEBUG: admin
DEBUG: shree
Enter name:
Prajjwal Pathak
Enter age:
21
Enter password:
topper
[+] User registered successfully
Enter username:
prajjwal
Enter password:
topper
[#] User Found
[.] Logging In
Select Hotel
-------------------------
1. Mumbai - Taj Hotel
2. Mumbai - JK Hotel
3. Mumbai - Sea Inn
4. Jaipur - Highway King
5. Jaipur - City Hotel
```

```
1. Mumbai - Taj Hotel
2. Mumbai - JK Hotel
3. Mumbai - Sea Inn
4. Jaipur - Highway King
5. Jaipur - City Hotel
4
0------------------------
City: Jaipur
Name: Highway King
Price of a Room (per day): 1200
Total Rooms: 50
Vacant Rooms: 50
------------------------
Do you want to book rooms? [Y/N]
y
Enter number of rooms:
1
Enter number of people:
1
Enter start date (dd-MM-yyyy):
11-05-2021
Enter end date (dd-MM-yyyy):
16-05-2021
[+] Room Booked

Process finished with exit code 0
```