

Object Oriented Programming

### Project Report

## 

## Submitted to:

Ms. Maheen Nadir

## Submitted by

M. Shakeel Khan

222202011

**Department of Computer Science,**

KICSIT, Kahuta.

Jan 19th, 2022

# **Introduction**

Simple Hotel Management System is based on the concept of managing the hotel room’s bookings and payments. In this system, there are front-desk operations, reservation management, Display room Restaurant & Rooms Feedback, Food Feedback, Different type of rooms. This mini project contains limited but essential features.

# **Objective**

The objective of our project is to create an automated hotel management system to make a fast system to increase efficiency, providing excellent customer satisfaction. Transfer manual work to computerized system.

* Customer
* Room Customer
* Employee
* Dish
* Restaurant Customer
* Restaurant
* Hotel
* Room
* Room Service

# **Problem Description**

The need of this system was that the information was very difficult to retrieve and to find particular information like, e.g. to find out about the customer’s summary, the user has to go through various registers. This results in inconvenience and wastage of time.

Manual calculation are error prone and take a lot of time this may result in incorrect information...e.g. calculations of customers based on room reservations.

The hotel has well furnished rooms with inclusive meals and delicious dishes. The customer requests to book a room by providing the necessary details, a unique ID will be assigned to him.The room will be allocated to the customer depending upon the availability of vacant rooms .If the customer is unhappy with the room conditions he can contact the room service .The details of all the customers will be stored which can be viewed by the manager. The customers can order food from the provided appetizing menu. Before the customer checks out, he will be asked for a feedback.

1. **Features:**

* Pointers
* Arrays
* Classes
* Inheritance
* Polymorphism
* Aggregation
* File Handling
* Exception Handling

# **Implementation Details:**

# **Class: Hotel**

Hotel is the main class which follows singleton design pattern. It consists of Static \*hotel to create an instance of the class. The other attributes include the name and the address of the hotel.

**Functions:**

* generateID() which generates a unique ID for every customer.
* displayAvailable(): displays all the available rooms in the hotel.
* displayMenu(): displays the menu available in the restaurant.
* guestSummary(): It gives the whole summary of the details of the customer.
* askFeedback() : It asks the feedback of the customer about the hotel service.
* vacateRoom(): to vacant the room that has been allotted.
* getCustomerData() : to get the data from customer.
* takeOrder() : to take the order from the customer.

1. **Class: Room**

It has five attributes where roomType describes the type of room i.e., Deluxe, AC, Non-AC, General, Suite. The attribute status tells whether the room is occupied or no.

**Functions:**

* setRoom() : to book a room.
* isVacant() : to know the status whether the room is booked or no.
* displayDetail(): to display the details of all rooms
* vacateRoom(): to vacate the room.
* displayAvailable() : to display the details of the rooms which are available.

1. **Class: Restaurant**

It consists of array all the dishes that are available.

**Functions:**

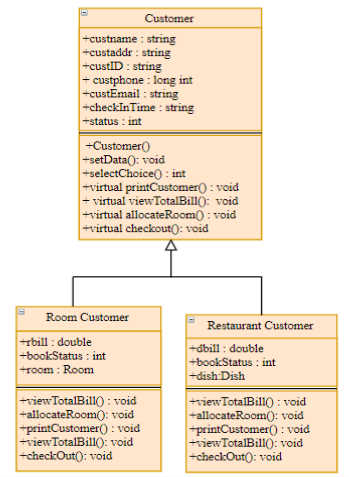
* displayMenu() : to display the menu for all dishes available.
* display() : To display the details of a particular dish
* addDish(): to add a dish into the menu.
* getDish() : to get the dish.

1. **Class: Customer**

The class Customer has two inherited classes restaurantCustomer and roomCustomer. It follows factory design pattern.

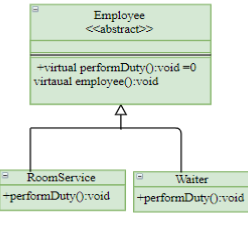
**Functions:**

* Setdata() : to enter all the customer details.
* selectChoice() : To select the choice between booking a room and placing an order for a dish.
* callRoomService(): To call the room service.
* viewTotalBill() : to view the bill of the service service provided to him.
* Checkout (): to checkout from the hotel.
* printCustomer() : to print the customer details who is in the room.



1. **Class: Employee**

performDuty() : To display all the employees working in the hotel.



The class employee inherits two classes namely RoomService and Waiter who perform their respective duties on customer demand. It follows strategy design pattern.

1. **Class: Dish**

This class has attributes such as dishname, price and dish type. The dishtype specifies the type to which the given dish belongs i.e., main, pulao, tea, sweet etc. The parameterized constructor creates new dishes.

1. **Class: Exception**

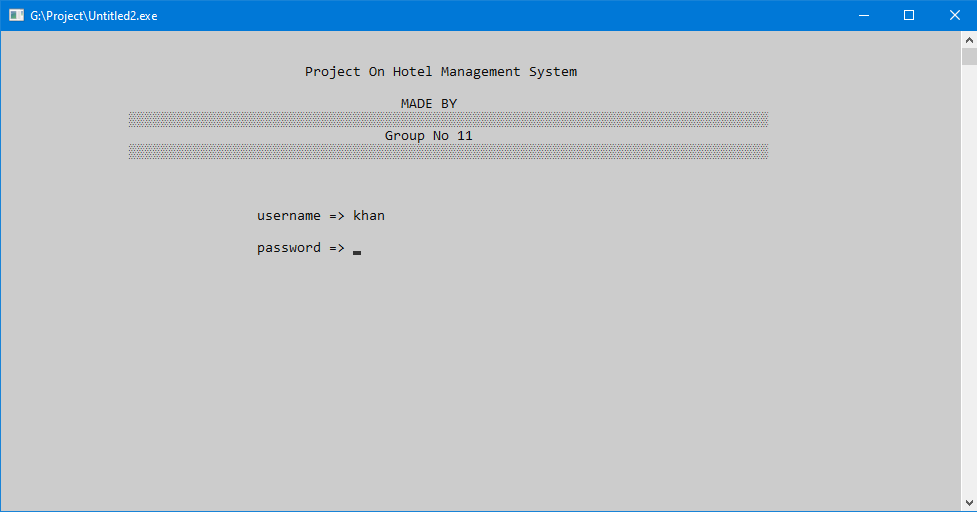
The following exception class comes into picture if

* Any customer asks for a room which is already occupied.
* Any customer orders for a dish which is not available at that instance of time.

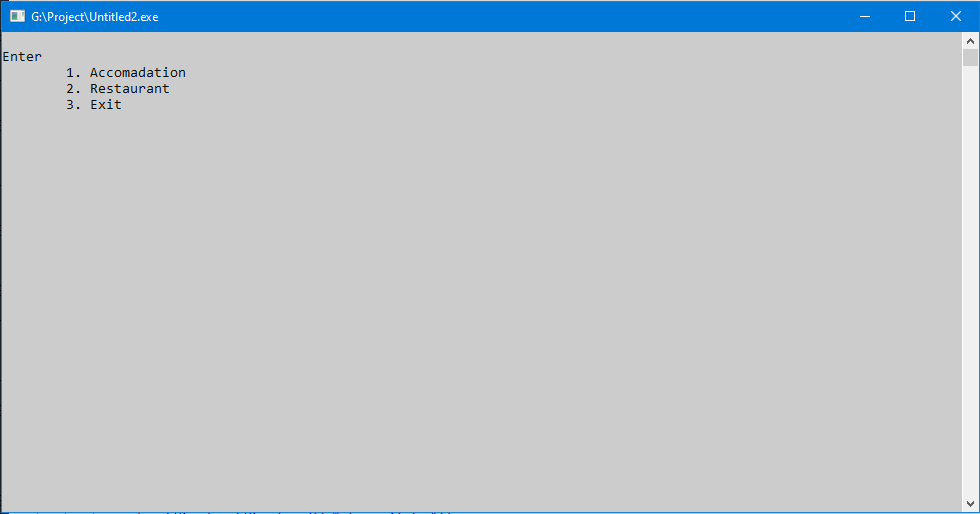
In the above cases Exception Handler will throw an appropriate error.

**OUTPUT:**

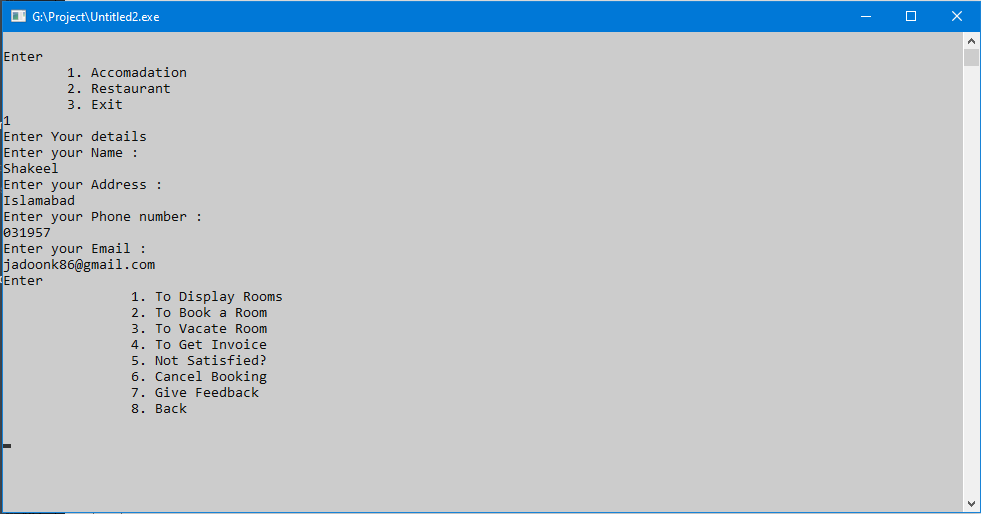
User Enter the username and password to enter in Hotel Management system.



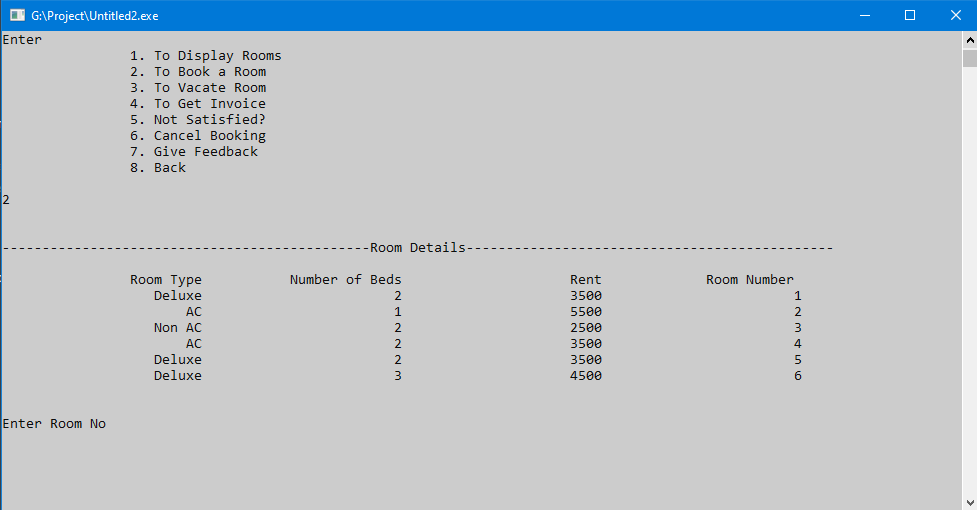
After this, there are two options one for restaurant and one for booking of rooms.



Select an option to book room or want to see room types etc.



Here are the types of room you can see or want to book room which you want.



Here is the second option i.e. Restaurant in which u can see Menu of hotel and chose Dish which you want. Also a Feedback option to give feedback about meal or staff respectively.

