KLE Society's

KLE Technological University



**Open Ended (OE) Assessment Report**

**On *HOTEL MANAGEMNET SYSTEM***

**Object Oriented Programming (20ECSC204)  
Object Oriented Programming Lab (20ECSP203)**

Submitted by

| **Name** | **Roll no** | **SRN** |
| --- | --- | --- |
| Sparsh Kumar | 170 | 01FE21BCS363 |
| Aakash Jain | 144 | 01FE21BCS228 |
| **Team Number:** 3 | | |

Faculty In-charge:

Prof. M.K. Gonal &

Asst. Prof. Savita H.

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

Academic year 2021-22

| **1.** | **Introduction** 2 | |
| --- | --- | --- |
|  | 1.1 | Overview of the problem statement |
|  | 1.2 | Features of Application |
| **2.** | **Design**  6 | |
|  | 2.1 | Class Diagrams |
| **4** | **Implementation** 7 | |
|  | 4.1 | Results |

1. **Introduction**
   1. **Overview of Problem Statement**

Design and implement a Hotel Management System that allows users to manage room bookings, guest details, employee records, and restaurant services in a hotel. The system should provide a user-friendly interface for hotel staff to efficiently handle various operations and tasks.

* 1. **Features of Application**

1. Hotel Management:

Display Available Rooms: Show a list of available rooms in the hotel, along with their room numbers, types, and prices.

Book a Room: Allow users to book a room by providing guest details, including name, Aadhar number, phone number, and the number of days for the stay.

Check-out from a Room: Check-out a guest from a booked room, calculate the total bill, and mark the payment status.

Display Customer Details: Show the reservation details of all guests currently staying at the hotel.

Display Employees: Show the details of employees working at the hotel, such as their names and employee IDs.

Add an Employee: Add new employees to the hotel by specifying their names, employee IDs, and roles (Receptionist, Housekeeping, Manager, Bellboy, or Chef).

Perform Employee Work: Display the tasks performed by each type of employee (Receptionist, Housekeeping, Manager, Bellboy, or Chef).

1. Restaurant Management:

Display Menu: Show the menu items available in the restaurant, along with their item numbers and prices.

Take Order: Allow customers to place orders from the restaurant's menu and calculate the total bill with GST added.

1. Guest and Room Management:

Guest: Store and manage guest details, including names, Aadhar numbers, and phone numbers.

Room: Represent different types of rooms in the hotel, such as Single Rooms, Double Rooms, and Suites, with features and prices.

Reservation: Manage room reservations by associating a guest with a booked room and the duration of stay.

1. Employee Management:

Employee: Represent employees working at the hotel with their names and employee IDs, including Receptionists, Housekeeping staff, Managers, Bellboys, and Chefs.

1. OOPS Concepts Used in Application :

Virtual Function: The work() function is declared as a virtual function in the Employee base class and overridden in derived classes to display specific work details for each employee type.

Classes and Objects: The code is organized using classes as blueprints for entities like Restaurant, Guest, Room, Employee, Reservation, and Bill. Objects of these classes are created to represent individual instances.

Constructor and Destructor: Classes use constructors for object initialization and destructors for resource cleanup. For example, the Hotel class has a constructor to initialize the hotel name and a destructor to manage memory.

Inheritance: Inheritance is used to create specialized classes from a base class, like SingleRoom, DoubleRoom, and Suite inheriting from Room, and Receptionist, Housekeeping, etc., inheriting from Employee.

Encapsulation: Internal data members of classes are kept private, and public methods provide access and modification (getters and setters). For example, Guest class has private data members name and Aadhar number, with public methods to access them.

Dynamic Polymorphism: Virtual functions and pointers/references to the base class are used for dynamic polymorphism. The Hotel class has a virtual function displayCustomers(), overridden in derived classes.

Static Polymorphism (Templates): Template method pattern is used in the function generateWelcomeMessage() to generate a welcome message with different types of hotel names.

Abstraction: Classes hide internal details and expose relevant features. The Room class provides methods to get room details while hiding implementation specifics.

Association and Aggregation: The Hotel class is associated with the Reservation class, and an aggregation relationship exists between the Hotel class and Room class.

Constructor Overloading: Multiple constructors are used in various classes, allowing different ways to initialize objects.

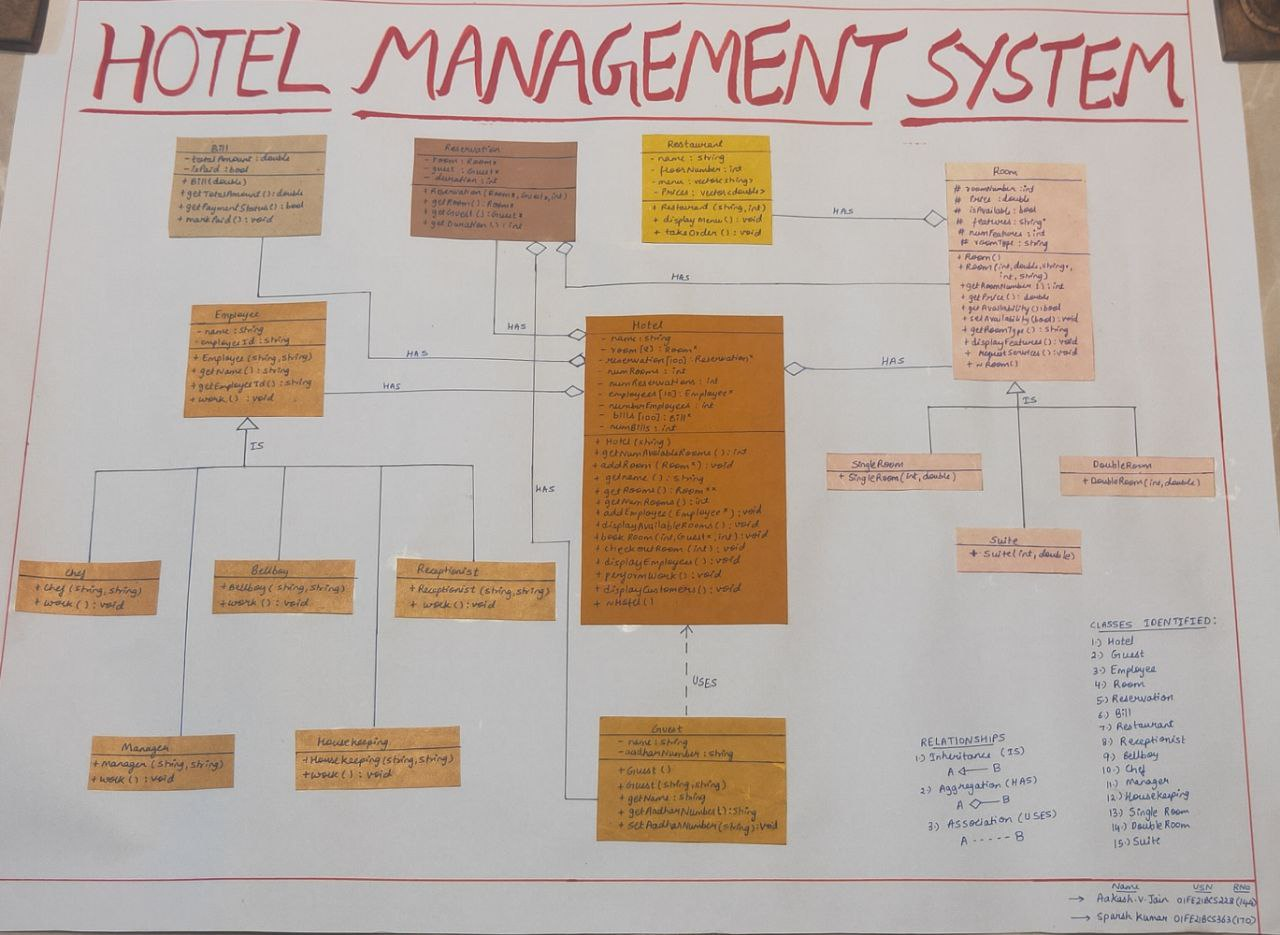
Vectors: Vectors are used to manage restaurant menu items and prices, providing dynamic resizing and convenience for managing data.

Virtual Functions: The Employee class has a virtual function work(), allowing different employee types to implement their specific work tasks.

Exception Handling: Input validation for Aadhar number in the Guest class constructor is handled using exception handling to ensure valid data entry.

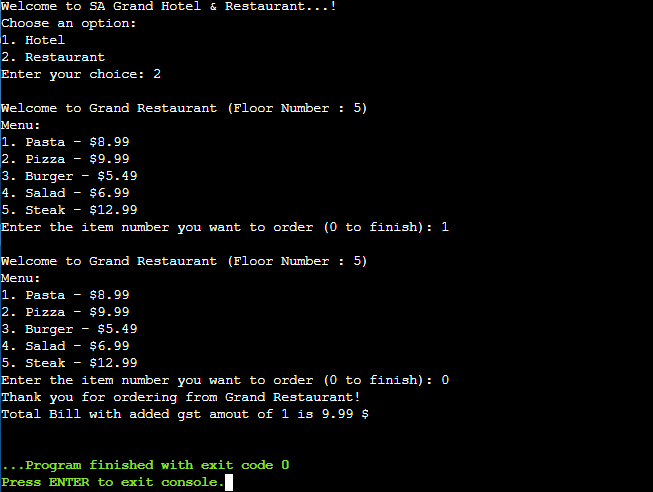
In summary, the Hotel Management System with Restaurant functionalities is designed using OOP concepts, templates, vectors, and virtual functions to create a flexible, reusable, and efficient application for managing hotels, guests, rooms, employees, and restaurant orders.

1. **Design**
   1. Class Diagram

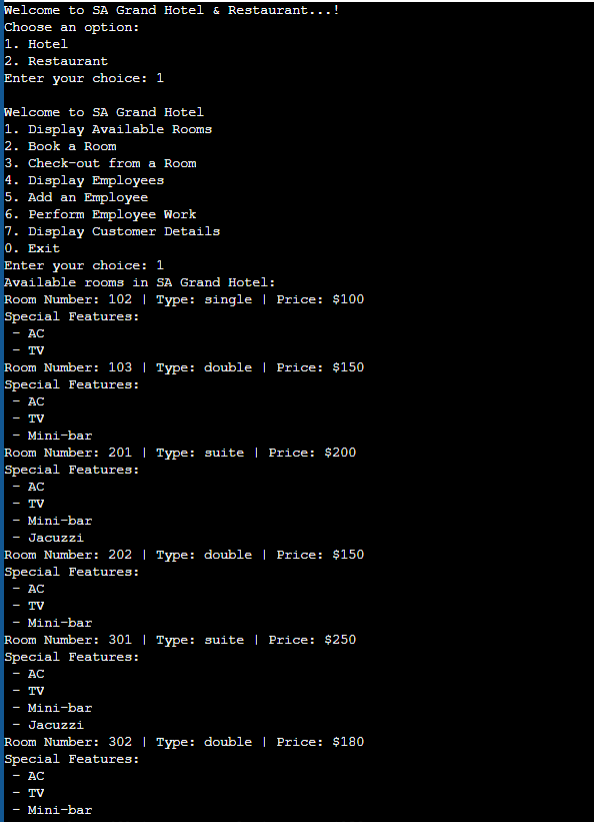
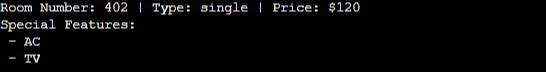


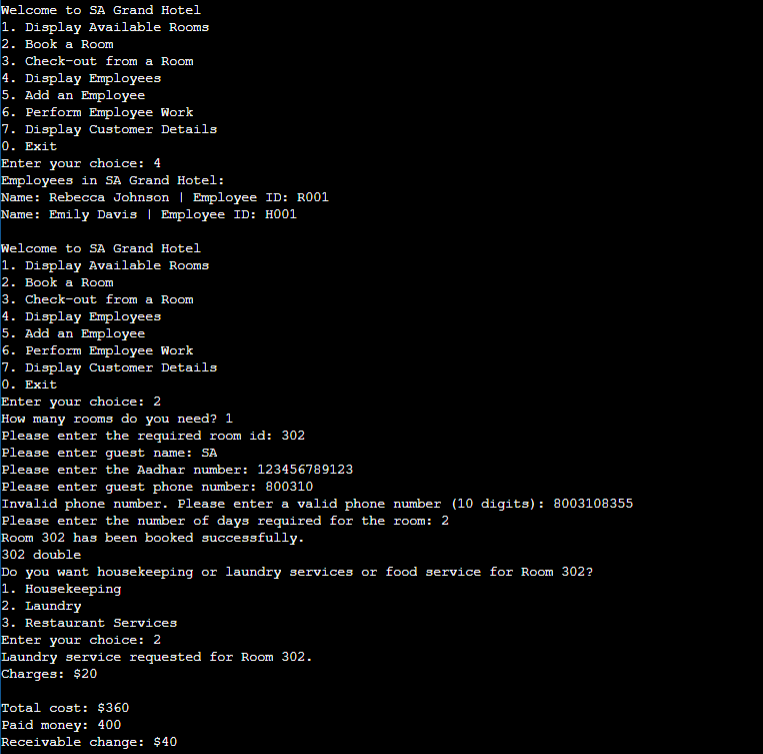
1. **Implementation**
   1. Results

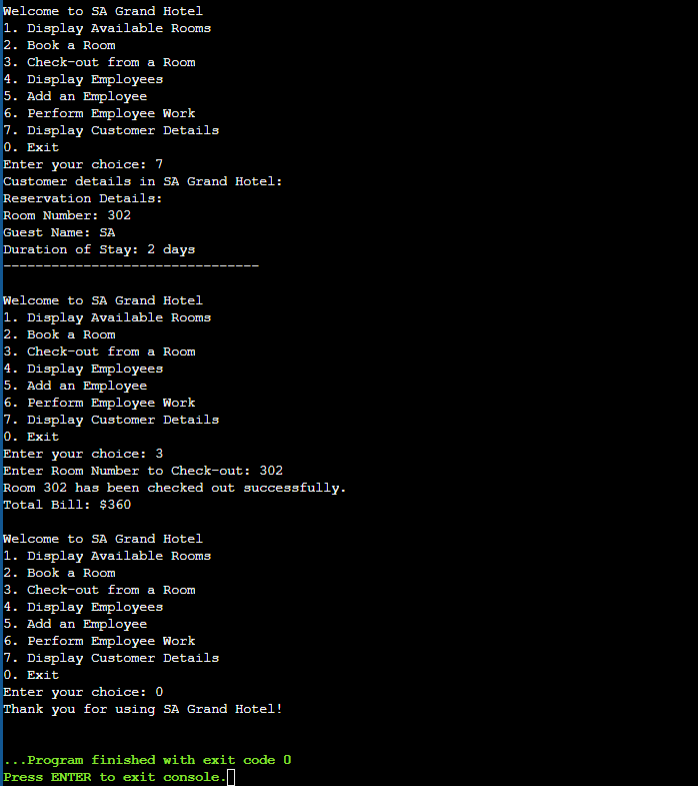
Snapshot 1



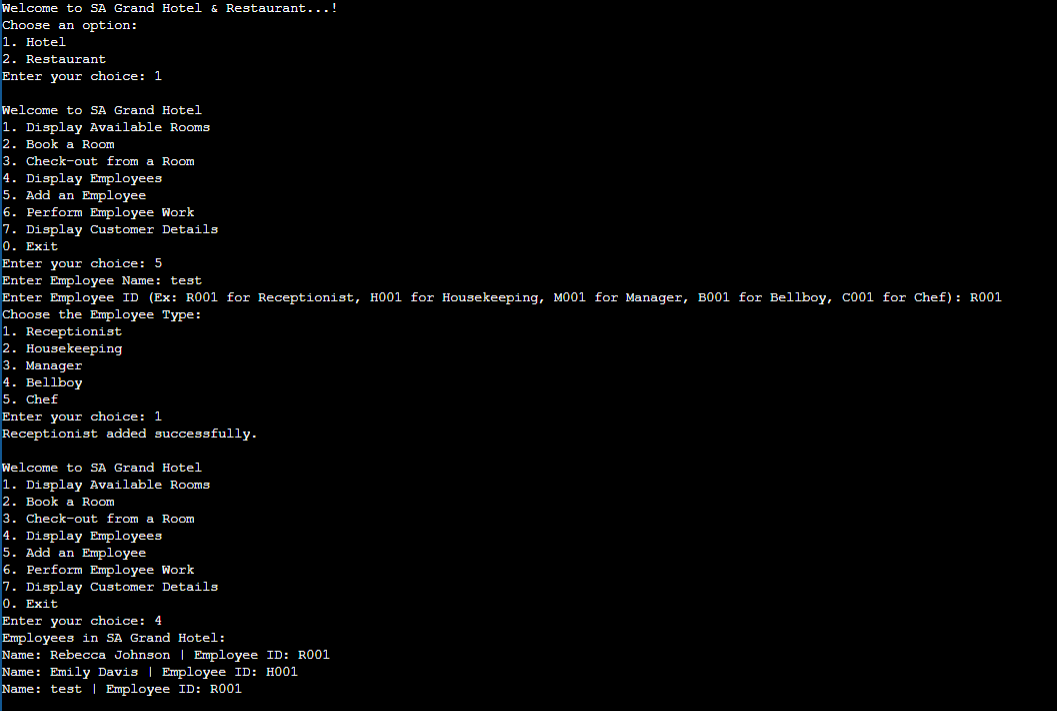
Snapshot 2

**** 





Snapshot 3



Snapshot 4 (With errors or invalid inputs)

