



**Project Report**

**Data Structures**

Name: Asifa Batool

ID: F2023266753

Section; V-01

Instructor: Muhammad Tahir Sohail

Project Title: Hotel Management System

Submission Due Date: 31 - Jan – 2025

University of Management and technology



# **Report on Hotel Management System Project**

## **Introduction**

This project is the implementation of a Hotel Management System that efficiently handles room allocations, booking requests, and maintains a history of operations for auditing purposes.

## **Objectives**

The primary goals of this project are:

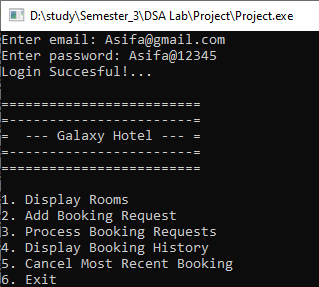
1. To manage hotel rooms and their statuses across multiple floors.
2. To handle and prioritize booking requests, ensuring high-priority requests are processed first.
3. To maintain a rolling 30-day availability status for each room.
4. To implement a batch processing system where booking requests are handled in groups of ten.
5. To maintain a stack-based booking history for auditing and rollback capabilities.
6. To ensure user authentication through email and password validation.

## **Features and Implementation**

### **1. User Authentication**

The system starts with a simple yet effective user authentication mechanism:

* **Email Validation**: Ensures the email contains both @ and . characters.
* **Password Validation**: Ensures the password includes at least one uppercase letter, one lowercase letter, one digit, and one special character.



### **2. Room Management**

The hotel comprises five floors, each containing ten rooms categorized into Single, Double, and Suite types. Each room has:

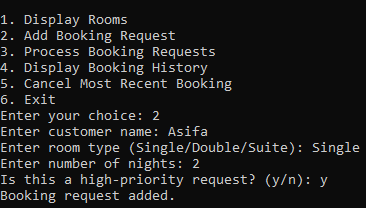
* **ID**: A unique identifier.
* **Status**: Indicates whether the room is Ready, Occupied, Booked, or Unavailable.
* **30-Day Availability**: Tracks whether the room is available for each of the next 30 days.

Rooms are managed using a **linked list(project.cpp)** and **tree(projectTree.cpp),** enabling dynamic allocation and traversal.

### **3. Booking Requests**

Booking requests are handled using a queue. Features include:

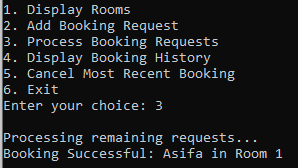
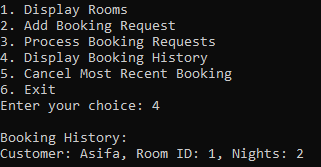
* **High-Priority Requests**: These are added to the front of the queue and processed first.
* **Batch Processing**: Once ten requests accumulate in the queue, they are processed in a single batch.
* **Validation**: Requests exceeding 30 nights are rejected.



### **4. Processing Bookings**

Booking requests are processed as follows:

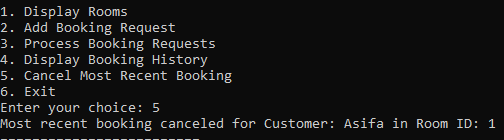
* **Room Search**: Searches for a room of the requested type with sufficient availability for the specified number of nights.
* **Status Update**: Marks the room as Occupied and updates its 30-day availability.
* **Booking History**: Successful bookings are pushed onto a stack for future auditing.



### **5. Booking History and Rollback**

The **stack** data structure is used to:

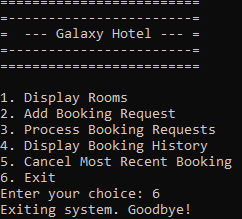
* Record details of successful bookings, including customer name, room ID, and duration of stay.
* Allow rollbacks by canceling the most recent booking and updating the room’s status and availability.



## **Results**

The implemented system successfully:

1. Handles room allocations dynamically and efficiently.
2. Processes booking requests with prioritization and batch handling.
3. Tracks room availability over a rolling 30-day period.
4. Maintains a detailed history of bookings for auditing and rollback.
5. Ensures secure user authentication.



**Conclusion**

This project explains the implementation of a Hotel Management System, focusing on simplicity, efficiency, and scalability. By leveraging fundamental data structures like linked lists, queues, and stacks, the system ensures effective resource management and user satisfaction.