

Darshan University

A Project Report on

**“Hotel Management System”**

Under the subject

**Software Engineering (2301CS405)**

B. Tech, Semester – IV

Computer Science & Engineering Department

|  |  |
| --- | --- |
| Submitted By | |
| Student Name: Jay Vegad | Enrollment No.:23010101294 |
| Academic Year  (2024-2025) | |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**DECLARATION**

We hereby declare that the SRS, submitted along with the **Software Engineering** **(2301CS405)** for entitled **“Hotel Management System”** submitted in partial fulfilment for the Semester-5 of **Bachelor Technology (B. Tech)** in **Computer Science and Engineering (CSE)** Departmentto Darshan University, Rajkot, is a record of the work carried out at **Darshan University, Rajkot** under the supervision of R. B. Gondaliya and that no part of any of report has been directly copied from any students’ reports, without providing due reference.

(Jay Vegad)

Student’s Signature

Date: \_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
|  | **Computer Science & Engineering Department**  **Darshan University** |

**CERTIFICATE**

This is to certify that the SRS on **“Hotel Management System” has** been satisfactorily prepared by **Jay Vegad** (**23010101294**) under my guidance in the fulfillment of the course **Software Engineering (2301CS405)** work during the academic year 2024-2025.

|  |  |  |
| --- | --- | --- |
| Internal Guide  Prof. R. B. Gondaliya  Darshan University |  | Dean-DIET  Dr. Gopi Sanghani  Darshan University |

**Acknowledgement**

I wish to express my sincere gratitude to my project guide Prof. R. B. Gondaliyaand all the faculty members for helping me through my project by giving me the necessary suggestions and advices along with their valuable co- ordination in completing this work.

I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of my work. In addition to that, I would also like to mention the Darshan University personals who gave me the permission to use and experience the valuable resources required for the project from the University premises.

Thus, in conclusion to the above said, I once again thank the faculties and members of **Darshan University** for their valuable support in completion of the project.

Thanking You

**Jay Vegad**

**ABSTRACT**

The Hotel Management System (HMS) is a comprehensive software solution designed to streamline and enhance the operational efficiency of hotel management. This system integrates various functions including reservations, front desk operations, housekeeping, billing, and reporting into a unified platform. By automating routine tasks and providing real-time data access, HMS aims to improve the guest experience, optimize resource utilization, and increase overall operational productivity.

Key features of the system include an intuitive reservation interface for managing bookings, an automated check-in/check-out process, real-time room availability updates, and a centralized database for guest information. Additionally, the system supports financial management with integrated billing and invoicing capabilities, along with detailed reporting tools for performance analysis.

The HMS is designed with scalability in mind, accommodating both small boutique hotels and large multi-property chains. By leveraging modern technologies such as cloud computing and mobile access, the system offers flexibility and accessibility, enabling hotel staff to manage operations efficiently from various devices. Ultimately, the Hotel Management System aims to enhance customer satisfaction, streamline hotel operations, and provide valuable insights for strategic decision-making.

**Table of Contents**

[List of Figures I](#_Toc146706805)

[List of Tables II](#_Toc146706806)

[1 Introduction 1](#_Toc146706807)

[1.1 Product perspective 1](#_Toc146706808)

[1.2 Product features 1](#_Toc146706809)

[1.2.1 There are three different users who will be using this product: 1](#_Toc146706810)

[1.2.2 The features that are required for the Librarian are: 1](#_Toc146706811)

[1.2.3 The features that are required for the Member are: 1](#_Toc146706812)

[1.3 Functional Requirement 2](#_Toc146706813)

[1.3.1 Librarian **Error! Bookmark not defined.**](#_Toc146706814)

[1.3.2 Member 2](#_Toc146706815)

[1.4 Non-Functional Requirement 5](#_Toc146706816)

[1.4.1 Usability: 5](#_Toc146706817)

[1.4.2 Accuracy: 5](#_Toc146706818)

[1.4.3 Availability: 5](#_Toc146706819)

[1.4.4 Maintainability: 5](#_Toc146706820)

[2 Design and Implementation Constraints 6](#_Toc146706821)

[2.1 Use case diagram 6](#_Toc146706822)

[2.2 Activity diagram and Swimlane diagram 7](#_Toc146706823)

[2.3 Sequence diagram 11](#_Toc146706824)

[2.4 State diagram 12](#_Toc146706825)

[2.5 Class diagram 13](#_Toc146706826)

[2.6 Data flow diagram 14](#_Toc146706827)

[2.6.1 Context diagram (level-0) 14](#_Toc146706828)

[2.6.2 DFD Level-1 14](#_Toc146706829)

[2.6.3 DFD Level-2 15](#_Toc146706830)

[3 External interface requirement (Screens) 16](#_Toc146706831)

[3.1 Screen-1: Registration Form 16](#_Toc146706832)

[3.2 Screen-2: Login Form 17](#_Toc146706833)

[3.3 Screen-3: Add borrower book 18](#_Toc146706834)

[4 Database design 19](#_Toc146706835)

[4.1 List of Tables 19](#_Toc146706836)

[5 Stories and Scenario 20](#_Toc146706837)

[5.1 Story-1: Add New Book in Library Catalogue 20](#_Toc146706838)

[5.1.1 Scenario# S1.1 20](#_Toc146706839)

[5.1.2 Scenario# S1.2 20](#_Toc146706840)

[5.1.3 Scenario# S1.3 20](#_Toc146706841)

[5.2 Story-2: Search Book 21](#_Toc146706842)

[5.3 Story-3: Manage due date for borrowed book 21](#_Toc146706843)

[5.4 Story-4: Renew book 21](#_Toc146706844)

[5.5 Story-5: Generate a report on book usage and availability 21](#_Toc146706845)

[6 Test cases 22](#_Toc146706846)

[7 References 26](#_Toc146706847)

# List of Figures

[Figure 2.1‑1 Use case diagram for library management system 6](#_Toc146706848)

[Figure 2.2‑1 Activity diagram for Book Issue 7](#_Toc146706849)

[Figure 2.2‑2 Swimlane diagram for Book Issue 8](#_Toc146706850)

[Figure 2.3‑1 Sequence diagram for Book Issue 11](#_Toc146706851)

[Figure 2.4‑1 State diagram of Book 12](#_Toc146706852)

[Figure 2.4‑2 State diagram for Librarian 12](#_Toc146706853)

[Figure 2.5‑1 Class diagram for Library management system 13](#_Toc146706854)

[Figure 2.6‑1 Context diagram for Library management system 14](#_Toc146706855)

[Figure 2.6‑2 DFD level-1 for Library management system 14](#_Toc146706856)

[Figure 2.6‑3 DFD level-2 for Issue book 15](#_Toc146706857)

[Figure 3.1‑1 Screen-1: Registration Form 16](#_Toc146706858)

[Figure 3.2‑1 Screen-2: Login Form 17](#_Toc146706859)

[Figure 3.3‑1 Screen-3: Add borrower book 18](#_Toc146706860)

# List of Tables

[Table 3.1‑1 Screen element of Registration form 16](#_Toc146706861)

[Table 3.2‑1 Screen element of Login form 17](#_Toc146706862)

[Table 3.3‑1 Screen element of Add borrower book 18](#_Toc146706863)

[Table 4.1‑1 Table: Book 19](#_Toc146706864)

[Table 4.1‑2 Table: Borrower 19](#_Toc146706865)

[Table 4.1‑3 Table: Staff 19](#_Toc146706866)

[Table 4.1‑4 Table: Student 19](#_Toc146706867)

# Introduction

## Product perspective

The Hotel Management System (HMS) transforms traditional hotel management practices into a modern, internet-based application, facilitating streamlined operations and enhanced guest interactions. The system supports multiple user roles, including Admin, Manager, Receptionist, and Guest, each with tailored functionalities to ensure efficient management and superior guest experiences. By integrating key operational components such as reservations, front desk operations, housekeeping, billing, and reporting into a unified platform, the HMS is designed to meet the needs of various types of hotels, from small boutique establishments to large hotel chains.

## Functional Requirement

### **Admin:-**

### • Basic Report Generation: Users can generate basic reports on occupancy rates and view revenue summaries for specific periods.

### • Profile Management: Users can update their profile information and change their passwords, maintaining up-to-date personal details.

* Login and Authentication: Users can log in with a username and password, reset forgotten passwords, and ensure secure access to the system.

### 

### • Room Availability Check: Users can view room availability for specific dates and filter available rooms by type for easier selection.

### • Basic Reservation: Users can create new reservations and view existing ones, facilitating the booking process for guests.

### • Guest Information Management: Users can add, update, and search for guest information by name or booking reference, maintaining accurate guest records.

### • Basic Check-in/Check-out: Users can check guests in and out, updating room status to occupied or vacant accordingly.

### • Invoice Generation: Users can generate simple invoices for a guest’s stay, and print or email them directly to the guest.

### • Housekeeping Status Update: Users can update the cleaning status of rooms and assign cleaning tasks to housekeeping staff.

### • Guest Feedback Collection: Users can record guest feedback and comments, and view a list of all feedback received.

### **Guest:-**

 **Online Booking:** Guests can search for rooms, check availability, and book online with instant confirmation. They can view room details, amenities, and rates before making a reservation.

 **Booking Management:** Guests can view, modify, or cancel bookings based on hotel policies. They receive notifications for any changes or cancellations.

 **Profile Management:** Guests can create and update their profiles, including contact details and preferences. They can change passwords and manage security settings.

• **Payment Processing:** Guests can make online payments using various payment methods (credit card, debit card, digital wallets). Guests can view their payment history and download invoices. The system provides secure payment processing to protect guest financial information.

• **Feedback & Reviews**: Guests can rate their stay, provide feedback, and receive responses from the hotel. Their suggestions help improve services.

• **Check-in & Check-out**: Online check-in speeds up the arrival process, and digital keys offer seamless room access. Guests can request early check-in or late check-out based on availability.

• **Service Requests**: Guests can request housekeeping, room service, or maintenance and track request status in real-time. Notifications are sent when requests are fulfilled.

• **Loyalty Program**: Guests can enroll in the hotel’s loyalty program, earn points, and redeem them for discounts or special offers. They receive updates on their membership status.

• **Special Offers**: Guests can view and book promotional deals, apply discount codes, and receive notifications about exclusive offers.

• **Support & Communication**: Guests can chat with customer support for assistance with bookings. A FAQ section provides quick answers to common queries.

• **Mobile Accessibility**: The hotel system is available on a mobile app and a mobile-friendly website. Guests can use mobile check-in and digital keys for a smooth experience.

**Manager:-**

* + **User & Role Management:** Managers can create, update, and deactivate staff accounts, assigning roles and permissions as needed.
  + **Room & Rate Management:** Managers can add, update, or remove rooms, set promotional rates, and manage availability.
  + **Reservation Management:** Managers can view, modify, or cancel bookings, analyze trends, and handle overbookings.
  + **Check-in/Check-out Oversight:** Managers monitor guest arrivals and departures, resolve issues, and update room statuses.
  + **Billing & Financial Management:** Managers generate invoices, process payments, review financial reports, and handle refunds or discounts.
  + **Staff Scheduling & Management:** Managers create schedules, track attendance, and evaluate staff performance.

### **Maintenance Management**: Managers log, assign, and track maintenance tasks, ensuring timely issue resolution.

### **Customer Feedback & Complaints**: Managers review guest feedback, address complaints, and improve service quality.

### **Report Generation & Analysis**: Managers create reports, analyze data, and identify operational improvements.

### **Marketing & Promotions**: Managers develop campaigns, track performance, and collaborate to enhance hotel visibility.

### **Receptionist:-**

• **User & Role**: Create, update, and control staff accounts with role-based permissions. Enable or disable accounts to maintain security.

• **Room & Rate**: Add, update, or remove room details, set seasonal rates, and apply discounts. Monitor inventory and ensure real-time availability.

• **Reservation**: View, modify, or cancel bookings, analyze trends, and handle overbookings. Ensure a seamless reservation experience for guests.

• **Check-in/Check-out Oversight**: Monitor arrivals and departures, resolve issues, and update room statuses. Ensure efficient housekeeping coordination and digital check-in.

• **Billing & Financial**: Generate invoices, process payments securely, and review financial reports. Handle discounts, refunds, and guest bill adjustments.

• **Staff Scheduling**: Assign shifts, track attendance, and oversee schedule changes. Evaluate employee performance and ensure smooth operations.

• **Maintenance**: Log, track, and assign maintenance requests while monitoring completion. Maintain records and ensure timely issue resolution.

• **Guest Feedback & Complaint Resolution**: Review and respond to guest feedback, track complaints, and implement improvements. Ensure guest satisfaction with follow-ups.

• **Reports & Analysis**: Generate and analyze reports on hotel operations, trends, and performance. Export reports for business planning and audits.

• **Marketing & Promotions**: Create campaigns, promotions, and special offers. Track effectiveness and enhance hotel **branding.**

## Non-Functional Requirement

### Usability: The UI should be simple enough for everyone to understand and get the relevant information without any special training. Different languages can be provided based on the requirements.

### Accuracy: The data stored about the books and the fines calculated should be correct, consistent, and reliable.

### Availability: The System should be available for the duration when the library operates and must be recovered within an hour or less if it fails. The system should respond to the requests within two seconds or less.

### Maintainability: The software should be easily maintainable and adding new features and making changes to the software must be as simple as possible. In addition to this, the software must also be portable.

# Design and Implementation Constraints

## Use case diagram

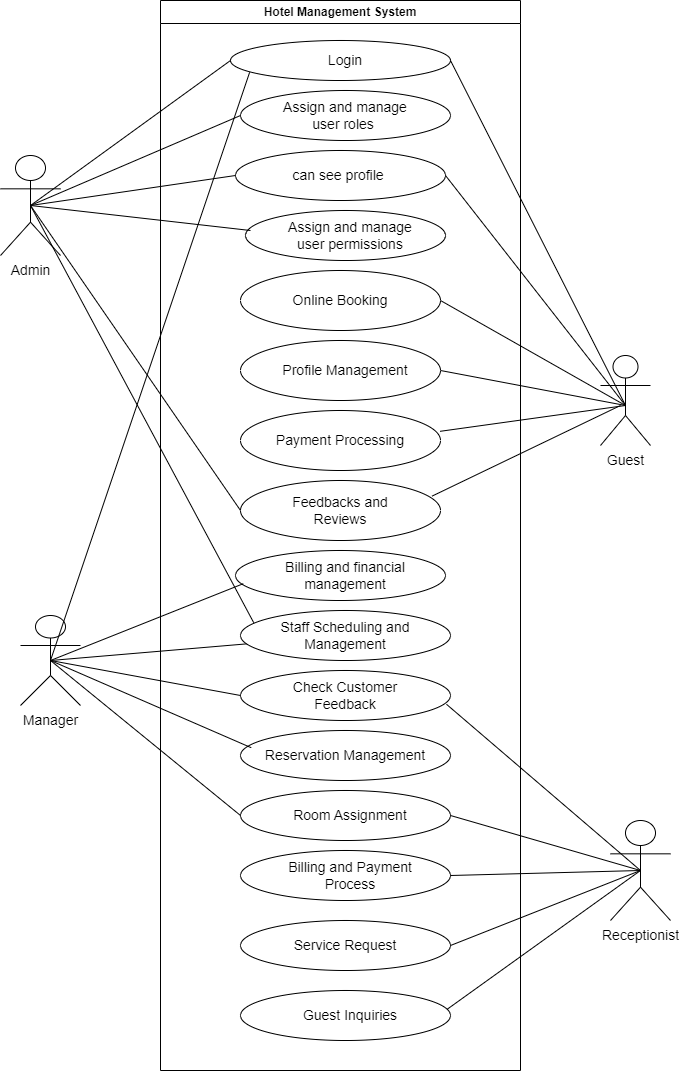


Figure 2.1‑1 Use case diagram for hotel management system

## Activity diagram and Swimlane diagram

### Activity diagram 1

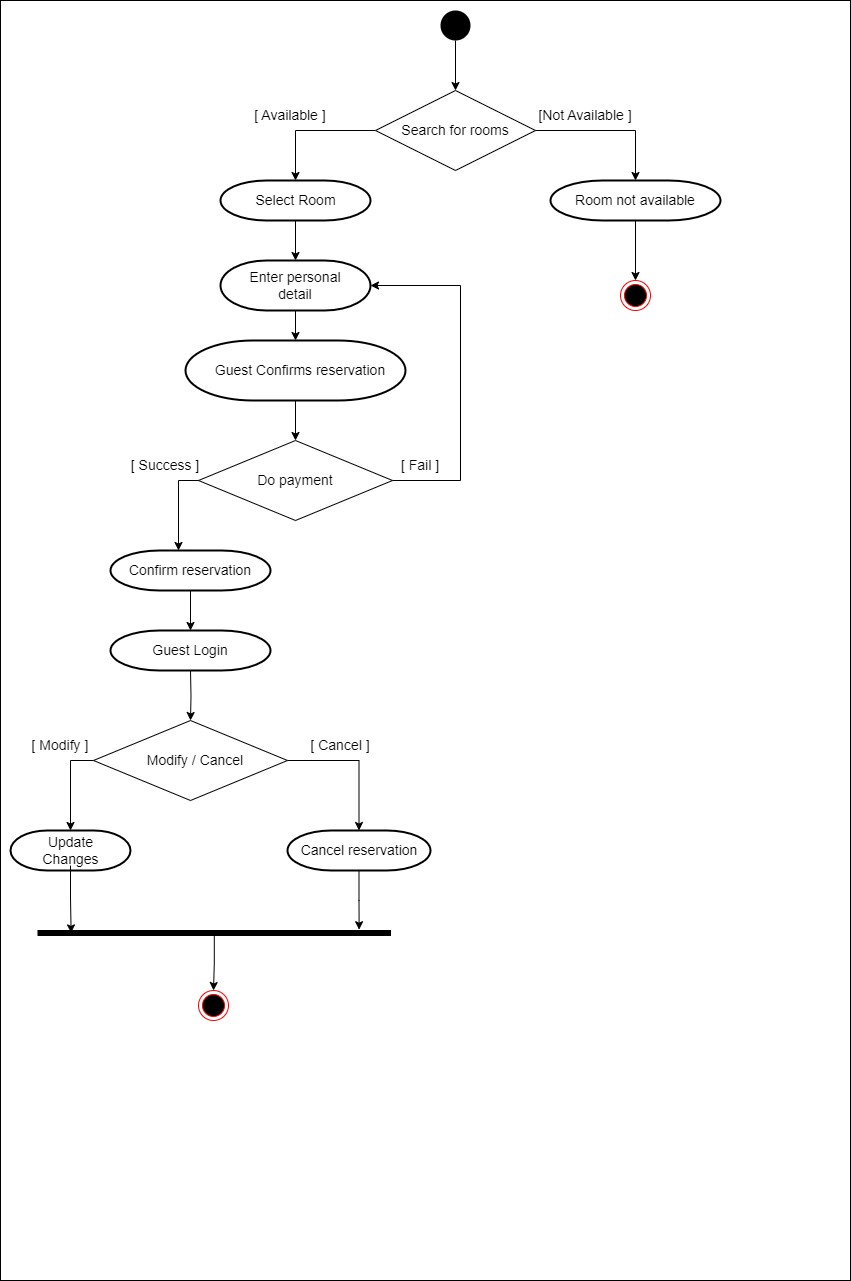


Figure 2.2‑1 Activity diagram for Reserve a room

### Activity diagram 2

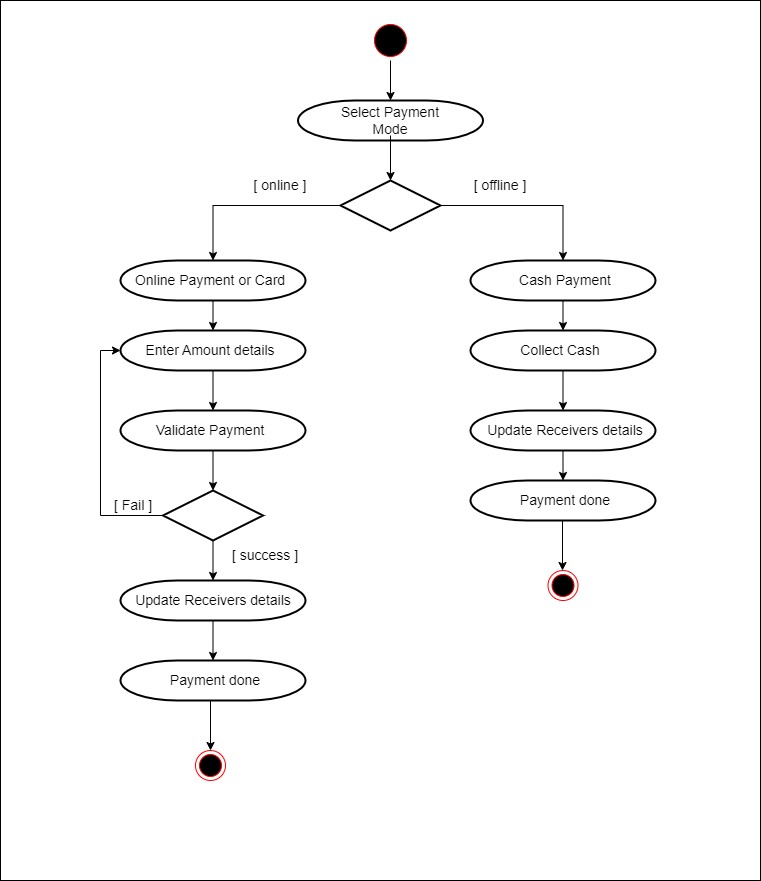
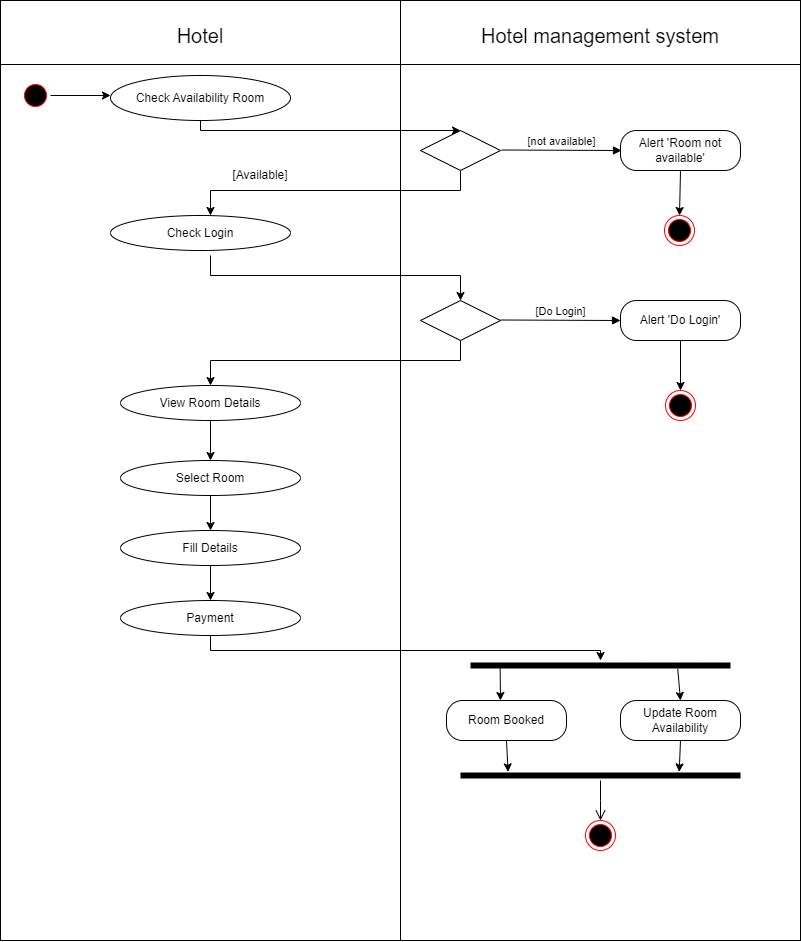


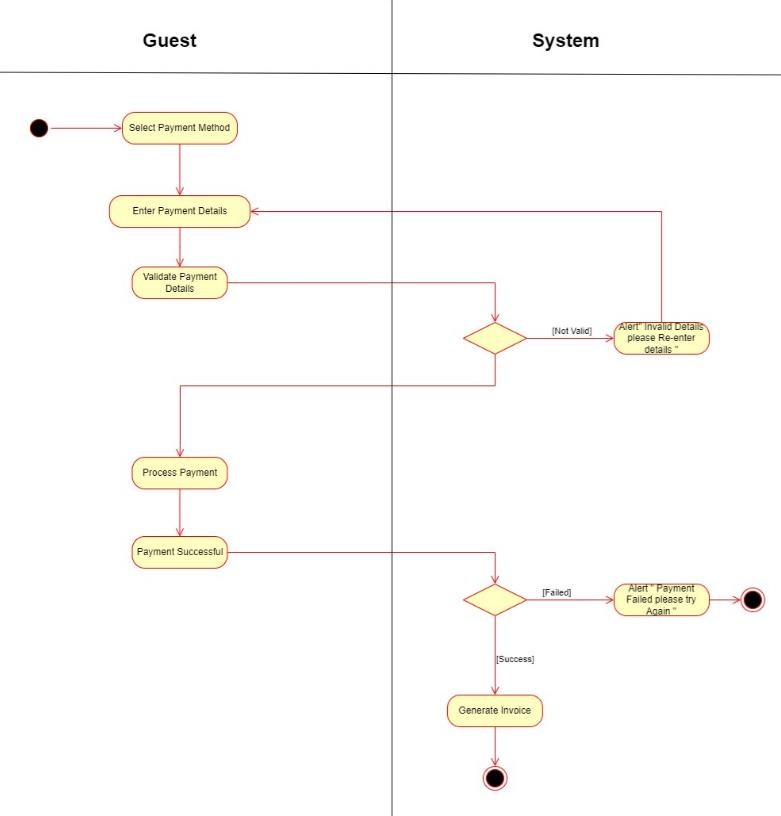
Figure 2.2‑2 Activity diagram for Payment

### Swimlane diagram 1



*Figure 2.2-2 Swimlane diagram for Room booking*

### Swimlane diagram 2



*Figure 2.2-3 Swimlane diagram for Payment*

## Sequence diagram



Figure 2.3‑1 Sequence diagram for Login request

## State diagram



Figure 2.4‑1 State diagram of Room



Figure 2.4‑2 State diagram for Librarian

## Class diagram



Figure 2.5‑1 Class diagram for Library management system

## Data flow diagram

### Context diagram (level-0)



Figure 2.6‑1 Context diagram for Library management system

### DFD Level-1



Figure 2.6‑2 DFD level-1 for Library management system

### DFD Level-2



Figure 2.6‑3 DFD level-2 for Issue book

# External interface requirement (Screens)

## Screen-1: Generate Report

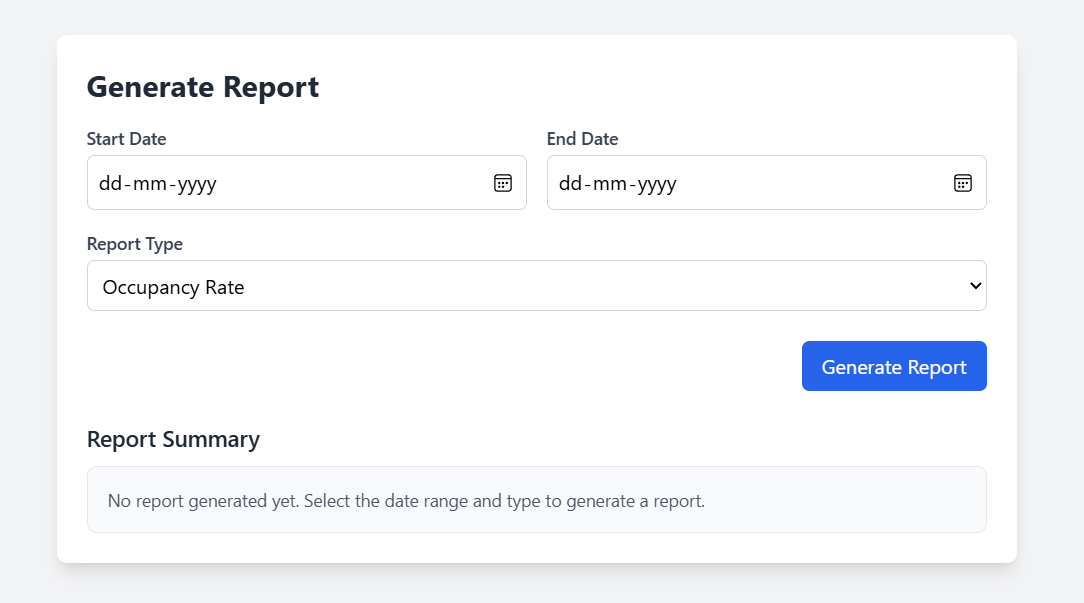


Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Start Date | Datepicker | M | 1 | Selectable date field for report start date. |
| 2 | End Date | Datepicker | M | 1 | Selectable date field for report end date. |
| 3 | Report Type | Dropdown | M | 1 | Allows the user to select a report type (e.g., Occupancy Rate). |
| 4 | Generate Report | Button | --- | --- | Triggers report generation based on selected inputs. |
| 5 | Report Summary | Display | O | 1 | Shows the generated report details or a message if no report is available. |

## Screen-1: Room Availability Check

A screenshot of a computer

Description automatically generated

Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Check-in Date | Datepicker | M | 1 | Selectable date field for check-in date. |
| 2 | Check-out Date | Datepicker | M | 1 | Selectable date field for check-out date. |
| 3 | |  | | --- | |  |  |  | | --- | | Room Type | | Dropdown | M | 1 | Allows the user to select a room type (e.g., All). |
| 4 | Number of Guests | Input Field | M | --- | Allows the user to enter the number of guests. |
| 5 | Special Requests | Textarea | O | 1 | Allows users to enter any special requests for the booking. |
| 6 | Check Availability | Button | --- | --- | Triggers room availability check based on selected inputs. |
| 7 | Available Rooms | Display | O | 1 | Shows the available rooms or a message if no rooms are available. |

## Screen-3: Book your stay

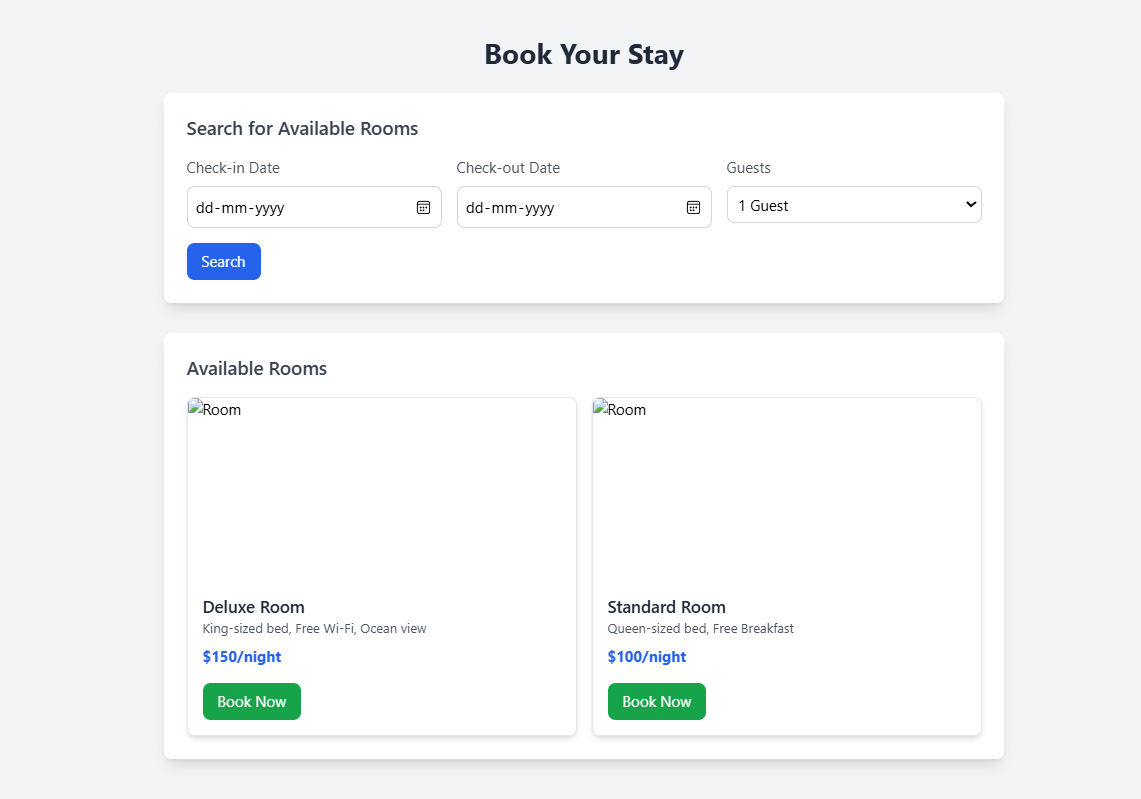


Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Check-in Date | Datepicker | M | 1 | Selectable date field for check-in date. |
| 2 | Check-out Date | Datepicker | M | 1 | Selectable date field for check-out date. |
| 3 | Room Type | Dropdown | M | 1 | Allows the user to select a room type (e.g., All). |
| 4 | Number of Guests | Input Field | M | — | Allows the user to enter the number of guests. |
| 5 | Special Requests | Textarea | O | 1 | Allows users to enter any special requests for the booking. |
| 6 | Check Availability | Button | — | — | Triggers room availability check based on selected inputs. |
| 7 | Available Rooms | Display | O | 1 | Shows the available rooms or a message if no rooms are available. |

## Screen-1: Generate Report

A screenshot of a computer

Description automatically generated

Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | First Name | Input Field | M | 1 | Allows the user to enter their first name. |
| 2 | Last Name | Input Field | M | 1 | Allows the user to enter their last name. |
| 3 | Email Address | Input Field | M | 1 | Allows the user to enter their email address. |
| 4 | Preferences & Requests | Textarea | O | 1 | Allows users to enter preferences or special requests. |
| 5 | New Password | Password | M | 1 | Allows users to enter a new password. |
| 6 | Confirm Password | Password | M | 1 | Requires users to confirm their new password. |
| 7 | Save Changes | Button | — | — | Saves the changes made to the profile. |

## Screen-1: Billing & financial management

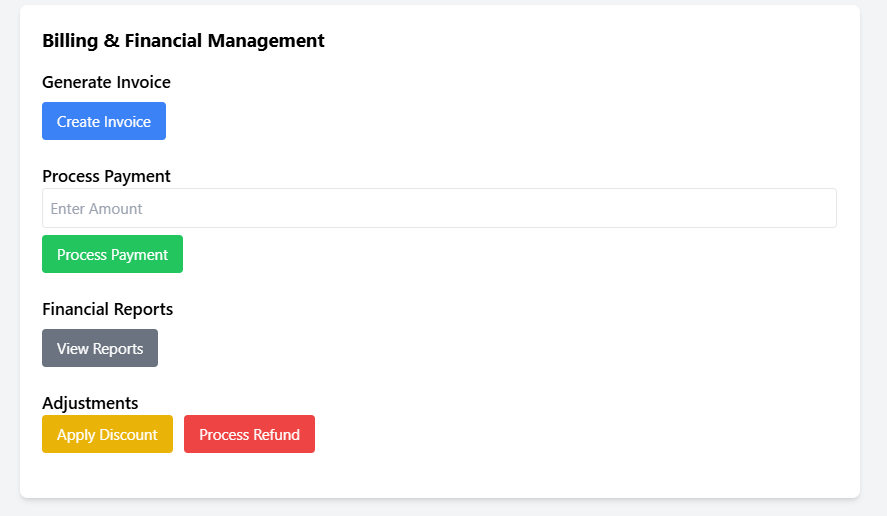


Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Create Invoice | Button | M | 1 | Generates a new invoice. |
| 2 | Enter Amount | Input Field | M | 1 | Allows users to enter the payment amount. |
| 3 | Process Payment | Button | M | 1 | Processes the entered payment amount. |
| 4 | View Reports | Button | O | 1 | Opens financial reports. |
| 5 | Apply Discount | Button | O | 1 | Applies a discount to a transaction. |
| 6 | Process Refund | Button | O | 1 | Processes a refund for a payment. |

## Screen-1: Staff scheduling

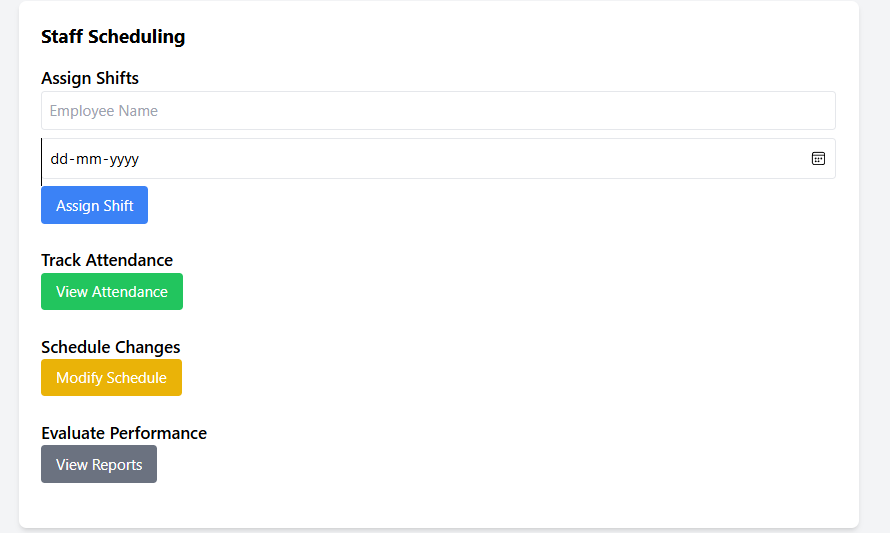


Table 3.3‑1 Screen element of Add borrower book

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Employee Name | Input Field | M | 1 | Allows users to enter the employee's name. |
| 2 | Date Picker | Input Field | M | 1 | Allows users to select a date for shift assignment. |
| 3 | Assign Shift | Button | M | 1 | Assigns a shift to the selected employee. |
| 4 | View Attendance | Button | O | 1 | Opens the attendance tracking section. |
| 5 | Modify Schedule | Button | O | 1 | Allows users to make changes to the schedule. |
| 6 | View Reports | Button | O | 1 | Displays performance evaluation reports. |

# Database design

## List of Tables

* Guests
* Rooms
* Reservations
* Employees
* Admin

Table 4.1‑1 Table: Book

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| GuestID | int | NN | PK (Auto Increment) | Unique ID for the guest |
| First Name | varchar(100) | NN |  |  |
| Last Name | varchar(100) | NN |  |  |
| Email | email | AN | UNIQUE |  |
| Phone Number | varchar(50) | AN |  |  |

Table 4.1‑2 Table: Borrower

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| Room ID | int | NN | PK (Auto Increment) | Unique ID for the room |
| Room Number | varchar(100) | NN | UNIQUE |  |
| RoomType | VARCHAR(100) | NN |  |  |
| PricePerNight | DECIMAL(10,2) | NN |  |  |
| Status | bit | NN |  | 'Available' |
| Floor Number | int | NN |  |  |

Table 4.1‑3 Table: Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| Reservation ID | int | NN | PK (Auto Increment) | Unique ID for the reservation |
| guest ID | int | NN | FOREIGN KEY | References of Guests table |
| Room ID | int | NN | FOREIGN KEY | References of Rooms table |
| CheckInDate | date time | NN |  |  |

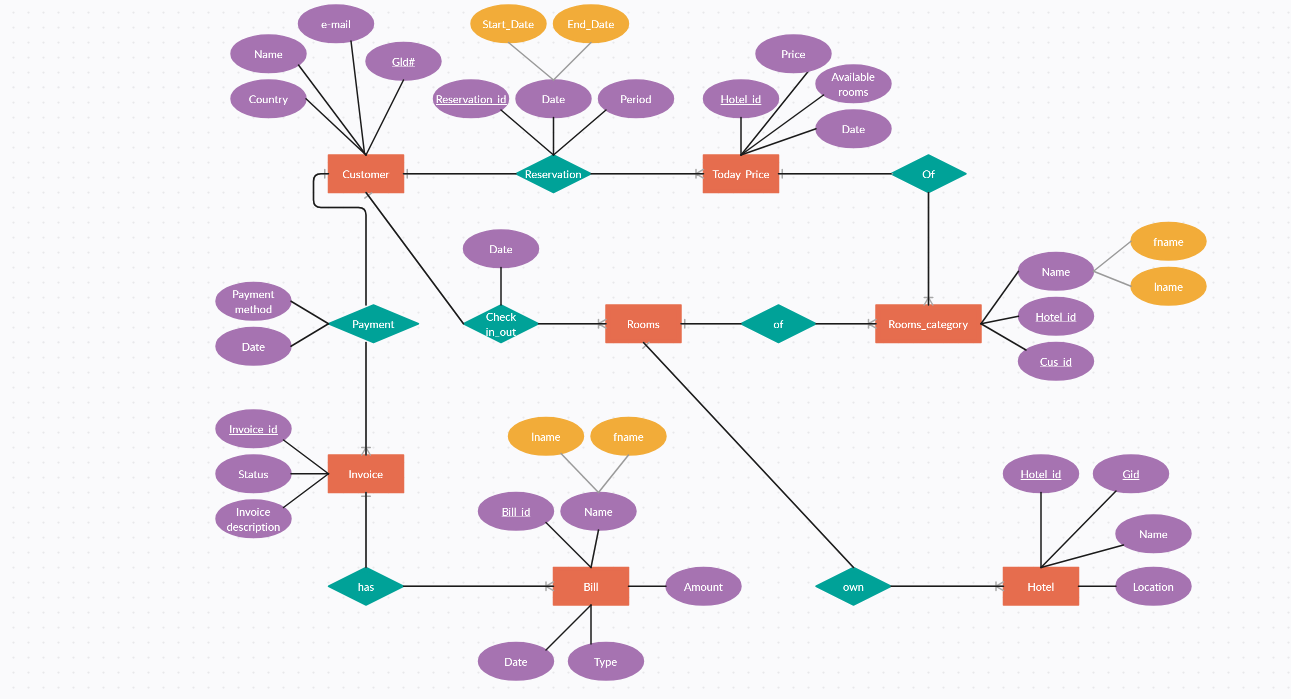
Table 4.1‑4 Table: Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| Employee ID | int | NN | PK (Auto Increment) | Unique ID for the employee |
| First Name | varchar(100) | NN |  |  |
| Last Name | varchar(100) | NN |  |  |
| DOB | Date Time | NN |  |  |
| Department | varchar(100) | NN |  |  |
| Contact | varchar(10) | NN |  |  |

Table 4.1‑5 Table: *Admin*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| AdminId | int | NN | PK (Auto Increment) | Unique ID for the Admin |
| First Name | varchar(100) | NN |  |  |
| Last Name | varchar(100) | NN |  |  |
| DOB | Date Time | NN |  |  |
| Department | varchar(100) | NN |  |  |
| Contact | varchar(10) | NN |  |  |
| Email | Varchar(100) | NN | UNIQUE | Unique Email for the Admin |

**ER DIAGRAM**



# Stories and Scenario

## Story-1: : Book a room in hotel

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Customer,  I want to Book a new room  So that we can live stay there. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | So can customer book room and stay in our hotel. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Adding a customer for stay. |
| Prerequisite | **:** | Customer is logged in to our hotel management system. |
| Acceptance Criteria | **:** | **Given:**  The Customer is navigated to the room catalog management page. Valid room information, including size, bed size, facilities, and other relevant details is added.  **When:**  The Customer selects the "Book A Room" option  And The Customer enters valid room details  The customer clicks the "Save" button to Book a room.  **Then t**he system successfully books room for customer. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New room with Invalid Information. |
| Prerequisite | **:** | The customer is logged into the hotel management system. |
| Acceptance Criteria | **:** | **Given:** The customer is on the hotel catalogue management page  **When:** The customer selects the "Book A Room" option and the customer enters an incomplete or incorrect room details and customer clicks the "Save" button to book a new room to the catalogue.  **Then t**he system displays error messages for the incorrect or missing information and the room is not booked. |

## Story-2: Search Room in hotel management system

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Customer,  I want to search for room by floor, bed size, or keyword,  So that I can quickly find room that match my choice. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Implementing a search functionality is essential for enhancing the customer experience, as it allows customer to efficiently discover and access the hotel’s resources. |

* + 1. Scenario# S2.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Search a room for stay. |
| Prerequisite | **:** | Customer is logged into the hotel management system and is on the room search page. |
| Acceptance Criteria | **:** | **Given:** The customer is on the search page and can view available filter options.  **When:**  The customer selects a floor (e.g., "First Floor", "Second Floor") from the dropdown or filter options.  The customer clicks the "Search" button.  **Then t**he system displays a list of rooms available on the selected floor. |

### Scenario# S2.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New room with Invalid Information. |
| Prerequisite | **:** | Customer is logged into the hotel management system and is on the room search page. |
| Acceptance Criteria | **:** | **Given:** The customer is on the search page and has entered filter criteria.  **When:** The customer selects filters (e.g., floor, bed size, keyword) that do not match any available rooms.  The customer clicks the "Search" button.  **Then t**he system displays a message: "No rooms found matching your criteria. |

## Story-3: Check-In and Check-Out Process

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As a front desk clerk,  I want to process guest check-ins and check-outs efficiently,  So that I can ensure a smooth and accurate transition for guests staying at the hotel. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Streamlining the check-in and check-out process is vital for providing a positive guest experience and managing room availability effectively. |

* + 1. Scenario# S3.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Search a room for stay. |
| Prerequisite | **:** | The front desk clerk is logged into the hotel management system, and the guest has a reservation. |
| Acceptance Criteria | **:** | **Given:** The front desk clerk accesses the guest check-in page.  **When:** The clerk searches for the guest’s reservation using the guest’s name or reservation ID.  The system displays the guest's reservation details, including room number and check-in date.  The clerk verifies the details with the guest and confirms any special requests or preferences  The clerk processes the check-in, assigning the room and updating the system.  The system generates a room key and provides check-in details (e.g., check-in time, room number).  **Then** the guest is checked in, and the room availability is updated in the system.  The system shows the guest's new status as "Checked In." |

* + 1. Scenario# S3.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Adding a New room with Invalid Information. |
| Prerequisite | **:** | The front desk clerk is logged into the hotel management system, and the guest has not shown up for check-in. |
| Acceptance Criteria | **:** | **Given:** The front desk clerk accesses the reservation system and finds a guest’s reservation marked as No-Show.  **When:** The system indicates that the guest did not arrive for check-in by the scheduled check-in time.  The clerk confirms the no-show status and updates the reservation to reflect this.  **Then t**he room is marked as available, and the system automatically adjusts the booking status accordingly. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | EMI Calculator | Test Designed by: | P. U. Jadeja |
| Module Name: | **Login** | **Test Designed date:** | 01-10-2023 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 15-01-2023 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: Web application should be accessible | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | Login to web application with valid credential | Functional | Login to Library management system web application through valid credential | TC\_001 |
| TC\_002 | Login to web application with invalid credential | Functional | Login to Library management system web application through invalid credential | TC\_002 |
| TC\_003 | Varify login page elements | GUI | varify that all elements are availabe on login page | TC\_003 |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Access Web application URL | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Enter valid Username in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username:  Rbgondaliya@gmail.com |  |
| 3 | Enter valid Password in Password field | Password field should be editable and accept the password and display as star or dot | Password input displayed in dot and accepted | pass |  | Password: rbgondaliya |  |
| 4 | Enter valid captcha code in captch field | Captch field should editable and accept captcha and captcha is case sensitive | Captcha input accepted | Pass | Step required when human action validation perform | get captcha from image which is near by captcha field |  |
| 5 | Click on login button | User should login into site and navigated to dashboard | User navigated to dashboard and username should br display in top of the right side. | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Login to web application with invalid credential |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Verify that User is not able to Login with invalid Username and invalid Password | Should be display an error message enter wrong username or password | Display an error of wrong username and password | Pass |  |  |  |
| 2 | Verify that User is not able to Login with Valid Username and invalid Password | Should be display an error message enter wrong password | Display an error of wrong password | Pass |  |  |  |
| 3 | Verify that User is not able to Login with invalid Username and Valid Password | Should be display an error message User not found | Display an error Username not found | Pass |  |  |  |
| 4 | Verify that User is not able to Login with blank Username or Password | Set required field validation message for Username and Password | Display an error of wrong username and password | Fail | Not performa a validation function fix it |  | Bug\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Varify login page elements |
| **Test Type** | GUI |
| **Test Priority** | Medium |
| **Pre-condition** | Web application should be accessible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Launch application with the given url | The site launched properly | Site launched successfully | Pass |  | <https://accounts.google.com/ServiceLogin> |  |
| 2 | Verify that the login screen contains elements such as Username, Password, Sign in button, Remember password check box, Forgot password link, and Create an account link. | All listed control displayed properly on the page | Login page loaded successfully | Pass |  |  |  |
| 3 | Verify that cursor is focused on “Username” text box on the page load | Cursor is focused in Username textbox | Cursor focus in Username textbox | Pass |  |  |  |
| 4 | Verify that tab functionality is working properly or not | When tab pressed cursor move in next control | Cursor moving in next control | Pass |  |  |  |
| 5 | Verify that all the fields such as Username, Password has a valid placeholder | All text fields have proper placeholder | All text fields have proper placeholder | Pass |  |  |  |
| 6 | Verify that the labels float upward when the text field is in focus or filled (In case of floating label) | When field is focused or filled, label display on top of the filled | When field is focus or filled, label display on top of the filled | Pass | step required when fields with floating label |  |  |
| 7 | verify that forgot password link working properly | when click on forgot password load forgot passworg page | forgot password link not working | Fail |  |  |  |

# References

* http://www.w3schools.com/html/html\_intro.asp
* https://www.w3schools.com/php/default.asp
* https://www.javatpoint.com/uml