

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 : Pushti Markana M. | Enrollment No.: 23010101160 |
| 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.

PUSHTI MARKANA

Student’s Signature : pushti

Date: 23-12-2024

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

**CERTIFICATE**

This is to certify that the SRS on **“Hotel Management System” has** been satisfactorily prepared by **Pushti Markana** (**23010101160**) 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

**PUSHTI MARKANA**

ABSTRACTION

A **Hotel Management System (HMS)** is a software application designed to streamline and automate the operations of hotels and hospitality establishments. It serves as an integrated solution for managing various aspects of hotel administration, including reservations, check-ins and check-outs, billing, customer service, and overall resource management , ultimately aiming to improve guest experience and optimize overall hotel efficiency by centralizing critical information across different departments. The system enhances efficiency, reduces human error, and improves customer satisfaction by ensuring smooth operation of daily activities.

A Hotel Management System is an essential tool that enhances operational efficiency, improves customer satisfaction, and supports the growth of the hotel business by integrating various functions into a seamless, user-friendly platform.

**Table of Contents**

[List of Figures I](#_Toc192351969)

[List of Tables II](#_Toc192351970)

[1 Introduction 1](#_Toc192351971)

[1.1 Product perspective 1](#_Toc192351972)

[1.2 Product features 1](#_Toc192351973)

[1.2.1 There are four different users who will be using this product: 1](#_Toc192351974)

[1.2.2 The features that are required for the Customer are: 1](#_Toc192351975)

[1.2.3 The features that are required for the Staff are: 1](#_Toc192351976)

[1.2.4 The Features that are required for the Manager are: 1](#_Toc192351977)

[1.3 Functional Requirement 1](#_Toc192351978)

[1.3.1 Manager: 1](#_Toc192351979)

[1.3.2 Customer: 3](#_Toc192351980)

[1.3.3 Staff: 4](#_Toc192351981)

[1.3.4 Transport Member: 5](#_Toc192351982)

[1.4 Non-Functional Requirement 5](#_Toc192351983)

[1.4.1 Usability: 5](#_Toc192351984)

[1.4.2 Accuracy: 5](#_Toc192351985)

[1.4.3 Availability: 5](#_Toc192351986)

[1.4.4 Maintainability: 5](#_Toc192351987)

[2 Design and Implementation Constraints 6](#_Toc192351988)

[2.1 Use case diagram 6](#_Toc192351989)

[2.2 Activity diagram and Swimlane diagram 7](#_Toc192351990)

[2.3 Sequence diagram 8](#_Toc192351991)

[2.4 State diagram 10](#_Toc192351992)

[2.5 Class diagram 10](#_Toc192351993)

[2.6 Data flow diagram 11](#_Toc192351994)

[2.6.1 Context diagram (level-0) 11](#_Toc192351995)

[2.6.2 DFD Level-1 11](#_Toc192351996)

[3 External interface requirement (Screens) 12](#_Toc192351997)

[3.1 Screen-1: Hotel Reservation Form 12](#_Toc192351998)

[3.2 Screen-2: Check-in Form 13](#_Toc192351999)

[3.3 Screen-3: Payment 14](#_Toc192352000)

[3.4 Screen-4: Hotel Feedback Form 15](#_Toc192352001)

[3.5 Screen-5: Staff Application Screen 16](#_Toc192352002)

[3.6 Screen-6: View Profile Screen 17](#_Toc192352003)

[3.7 Screen-7: View Check-in Check-out Screen 18](#_Toc192352004)

[3.8 Screen-8: View Customer requirement Screen for Staff 19](#_Toc192352005)

[3.9 Screen-9: View Customer Location Screen for Transport Member 19](#_Toc192352006)

[4 Database design 20](#_Toc192352007)

[4.1 List of Tables 20](#_Toc192352008)

[5 Stories and Scenario 22](#_Toc192352009)

[5.1 Story-1: Add New Book in Library Catalogue 22](#_Toc192352010)

[5.1.1 Scenario# S1.1 22](#_Toc192352011)

[5.1.2 Scenario# S1.2 22](#_Toc192352012)

[5.1.3 Scenario# S1.3 23](#_Toc192352013)

[5.2 Story-2: Search Book 23](#_Toc192352014)

[5.3 Story-3: Manage due date for borrowed book 23](#_Toc192352015)

[5.4 Story-4: Renew book 23](#_Toc192352016)

[5.5 Story-5: Generate a report on book usage and availability 24](#_Toc192352017)

[6 Test cases 25](#_Toc192352018)

[7 References 30](#_Toc192352019)

# List of Figures

[Figure 2.1‑1 Use case diagram for hotel management system 6](#_Toc192352037)

[Figure 2.2‑1 Activity diagram for check-out Status 7](#_Toc192352038)

[Figure 2.2‑2 Swimlane diagram for check-out status 8](#_Toc192352039)

[Figure 2.3‑1 Sequence diagram for Check-out Status 9](#_Toc192352040)

[Figure 2.4‑1 State diagram of Room 10](#_Toc192352041)

[Figure 2.5‑1 Class diagram for Library management system 10](#_Toc192352042)

[Figure 2.6‑1 Context diagram for Hospital management system 11](#_Toc192352043)

[Figure 2.6‑2 DFD level-1 for Hotel management system 11](#_Toc192352044)

[Figure 3.1‑1 Screen-1: Reservation Form 12](#_Toc192352045)

[Figure 3.2‑1 Screen-2: Check-in Form 13](#_Toc192352046)

[Figure 3.3‑1 Screen-3: Payment 14](#_Toc192352047)

[Figure 3.4‑1 Screen-3: Feedback Form 15](#_Toc192352048)

[Figure 3.5‑1 Screen-5: Staff Application Screen 16](#_Toc192352049)

[Figure 3.6‑1 Screen-6 View Profile Screen 17](#_Toc192352050)

[Figure 3.7‑1 Screen-7 View check-in check-out Status 18](#_Toc192352051)

[Figure 3.8‑1 Screen-8 View Customer requirement screen 19](#_Toc192352052)

[Figure 3.9‑1 Screen-9 View Customer Location screen 20](#_Toc192352053)

# List of Tables

[Table 3.1‑1 Screen element of Reservation form 12](#_Toc192352054)

[Table 3.2‑1 Screen element of Check-in form 13](#_Toc192352055)

[Table 3.3‑1 Screen element of Payment 14](#_Toc192352056)

[Table 3.4‑1 Screen element of Hotel Feedback Form 15](#_Toc192352057)

[Table 3.5‑1 Screen element of Hotel Staff Application Form 16](#_Toc192352058)

[Table 3.6‑1 Screen element of view profile 17](#_Toc192352059)

[Table 3.7‑1 Screen element of check-in check-out Status 18](#_Toc192352060)

[Table 3.8‑1 Screen element of Customer requirement 19](#_Toc192352061)

[Table 3.9‑1 Screen element of Customer location 20](#_Toc192352062)

[Table 4.1‑1 Table: Room 20](#_Toc192352063)

[Table 4.1‑2 Table: Customer 21](#_Toc192352064)

[Table 4.1‑3 Table: Staff 21](#_Toc192352065)

[Table 4.1‑4 Table: Manager 21](#_Toc192352066)

# Introduction

## Product perspective

The project , Hotel Management System is a web based application that allows the hotel manager to handle all activities online . Interactive GUI and the ability to manage various hotel bookings and rooms make this System very flexible and convernient.This application gives manager the power and flexibility to manage the entire System from a single online System. These systems are sophisticated software solutions designed to streamline and automate the day-to-day operations of hotels and other hospitality establishments.

## Product features

### There are four different users who will be using this product:

* Manager who will be acting as the administrator.
* Customer who will be accessing the Hotel.
* Staff who will be accessing as Employee.
* Transportation Member who will be help Customer in Transport

### The features that are required for the Customer are:

* Allow customers to create profiles with personal details (name, contact information, preferences).
* Maintain a history of past stays, room preferences, special requests (e.g., pillow type, floor preference), and loyalty status.
* Display available rooms in real-time to customers.
* Automatic booking confirmations, reminders, check-in instructions, and promotional offers.
* Customers can order room service directly via an app or room tablet.
* Help customers with information regarding local attractions, booking excursions, arranging transportation, etc.

### The features that are required for the Staff are:

* Staff members should have access to only the features and data relevant to their roles (e.g., front desk staff, housekeeping, management).
* Allow staff to view and update their work schedules, request time off, or swap shifts.
* Keep track of room statuses (e.g., clean, dirty, maintenance) so that staff can efficiently manage guest rooms.

### The Features that are required for the Manager are:

* View and manage real-time room availability and bookings.
* Manage large group bookings, events, and conferences.
* Create invoices for guests and ensure accurate billing for services used.

## Functional Requirement

### Manager:

* Login :

- Input : Manager Code , Contact , Name , Password ,Email .

- Output : Database Record .

* View Profile :

-view our profile and collect more details and information .

* Make Reservation :

-Input: Member Code , Total Person, Check-in date ,Number of Nights.

-Output : Database Record ,Database fully pop-up.

-Work Flow : Validate the given details and record the information in to the given time

* Create Bills :

-Work Flow: Validate the given details and total cost is calculated according to the services gained by the customer.

-Output :printed version of the bill

* Update Guest :

- Input : Member Code ,Contact ,Name ,Email ,Address

-Work Flow : Validate the given details and record the information in to the given database

* Delete Guest :

- Input : Member Code ,Contact ,Name ,Email ,Address

- Output : Member has been Deleted by Manager in database

* Add Staff :

- Input : Employee Code , Contact , Employee Name ,Email ,Address

- Output : Employee has been Record by Manager in database. Database Record.

* Update Staff :

- Input : Employee Code , Contact , Employee Name ,Email ,Address

- Output : Employee has been Updated by Manager in database. Database Record.

* Delete Staff :

- Input : Employee Code , Employee Id, Contact , Employee Name ,Email ,Address

- Output : Employee has been removed in database. Database Record.

* View Staff :

- Input : Employee Code , Employee Id, Contact , Employee Name ,Email ,Address

- Output : View all Employee or staff and also view Specific one Employee details.

* Set Rates :

-Input: Chek-out , Check-in date ,Day ,Number of Guests , price.

-Output : Database Record ,Database fully pop-up.

* Retrive Report :

- Input : Employee Code , Employee Id, Contact , Employee Name , Employee Salary , Address.

-Output : Database Record ,Database fully pop-up

* Staff Payment :

- Input : Employee Code , Employee Id, Contact , Employee Name , Employee Salary , Address.

-Add staff payment and also pay salary of staff.

### Customer:

* View Check-in Check-out Status :

- Input : Customer Code , Contact , Name , Password ,Email .

- Output : Check Customer’s Check-in and Check-out Status.

* Search Room :

- Input : Customer Code , Room Id, Customer Name, Compartment.

- Output : Search Room from book list.

* Add Payment :

- Input : Cheque No. ,Cradit card No , Amount of Payment ,Discount.

- Output : Add Payment of that Customer and record in database .

* View Food Facility :

-view Food Facility and variety of available food and also check it’s freshness.

* View Location :

- Input : Hotel Name and Address.

- Output : View Current Location.

* View Branch : view Available branch Near by Customer.
* View Rating : view Rating of that Hotel given by other Customer.
* Give Rating : Give Rating based on their facility.
* View entertainment Facility : view Entertainment facility for Guests and Customer.

### Staff:

* View Customer requirement :

- Input : Customer Compartment , Servies Room and Centre .

- Output : Database Record . and give Servies based on Customer Requirement.

* Show Salary :

-Input : Employee Code , Contact , Name , Password ,Email .

- Output : Show Salary Status of Employee .

* View Cleaning System :

-View Cleaning System or requirement for Customer Rooms and lobby of Hotel.

### Transport Member:

* View Customer Location :

- Input : Customer Code , Contact , Name ,Id .

- Output : Database Record and View Customer Location for Transport.

## 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 customer and the fines calculated should be correct, consistent, and reliable.

### Availability:

* The System should be available for the duration when the customer stay 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

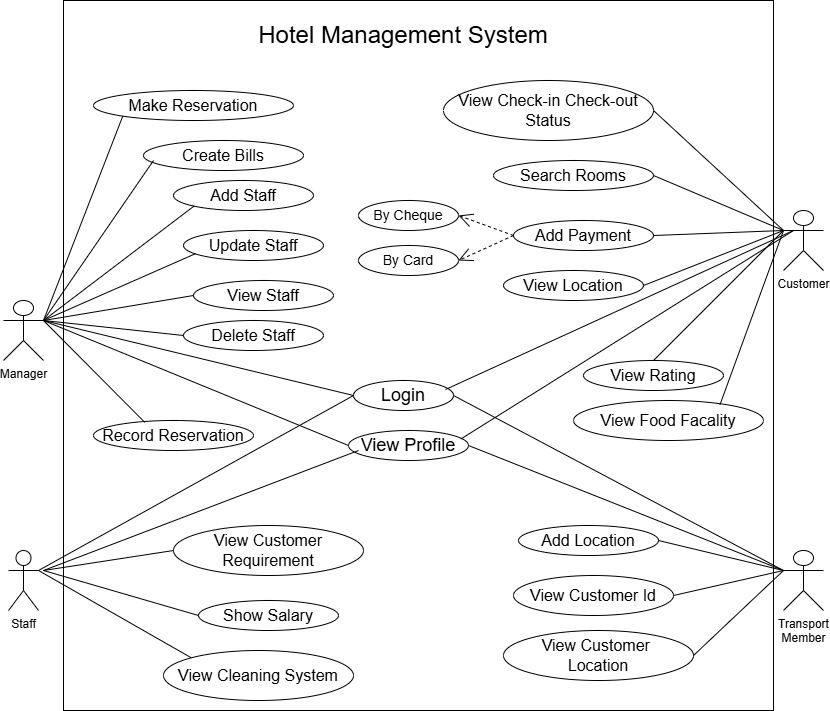


Figure ‑ Use case diagram for hotel management system

## Activity diagram and Swimlane diagram

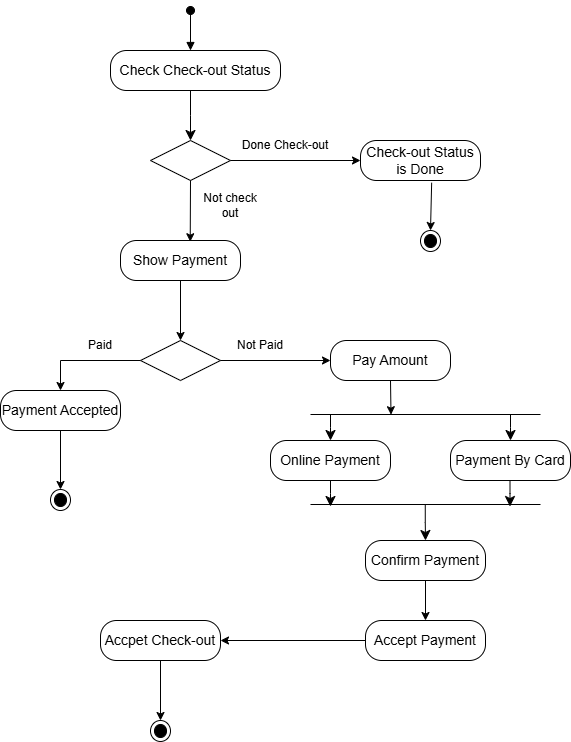


Figure ‑ Activity diagram for check-out Status

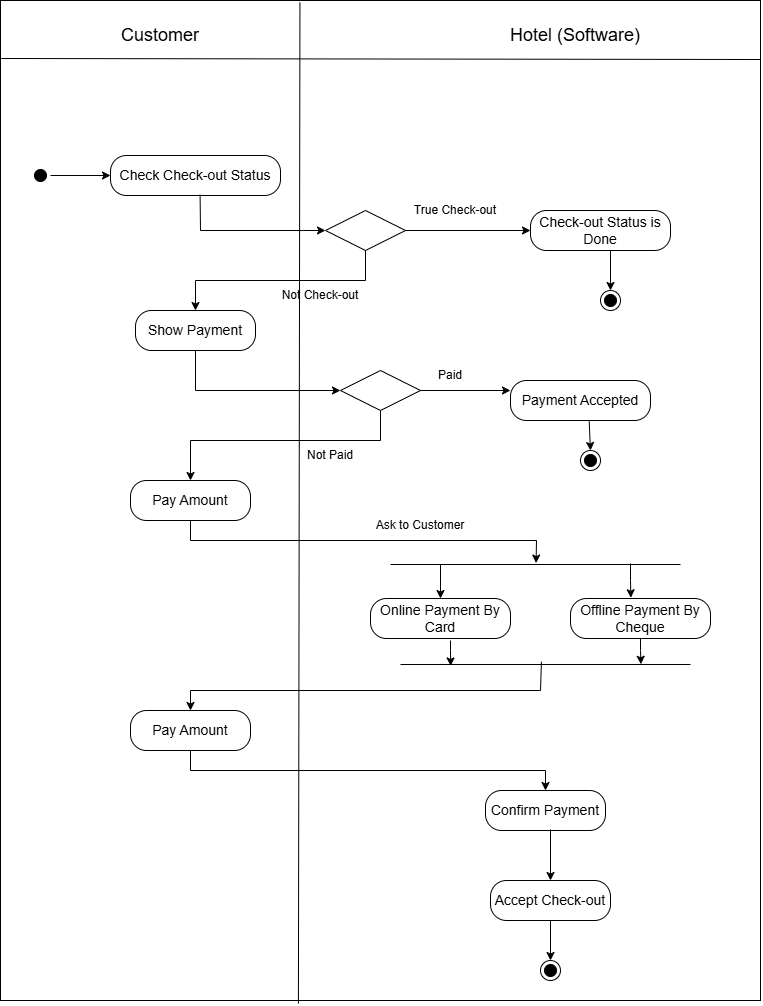


Figure ‑ Swimlane diagram for check-out status

## Sequence diagram

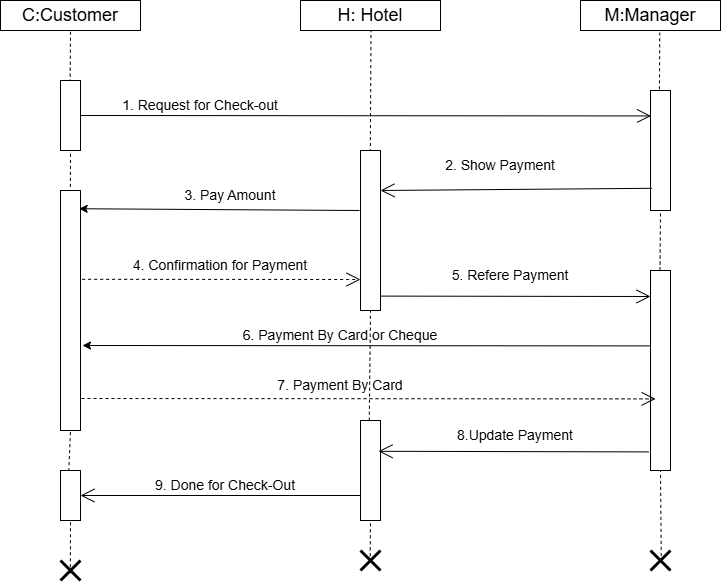


Figure ‑ Sequence diagram for Check-out Status

## State diagram

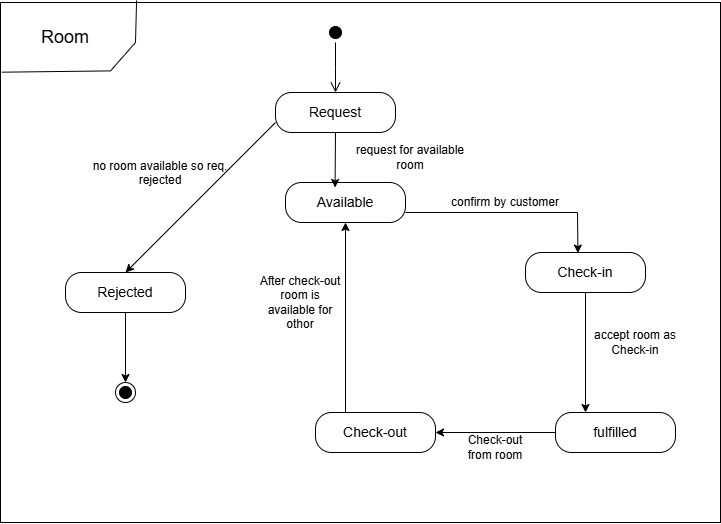


Figure ‑ State diagram of Room

## Class diagram



Figure ‑ Class diagram for Library management system

## Data flow diagram

### Context diagram (level-0)

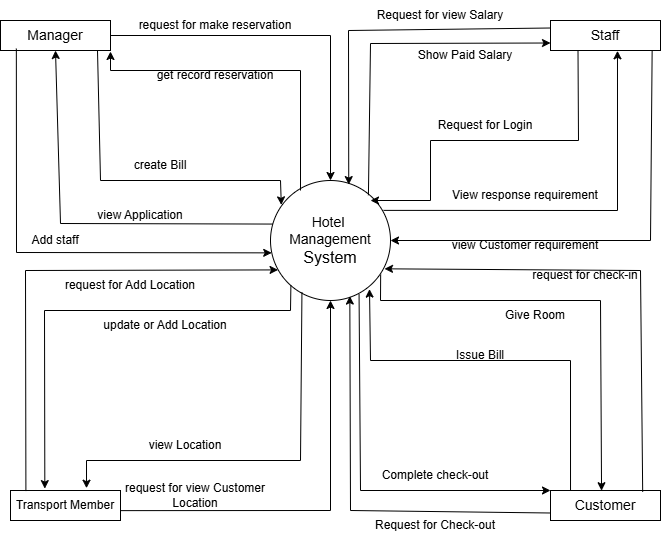


Figure ‑ Context diagram for Hospital management system

### DFD Level-1

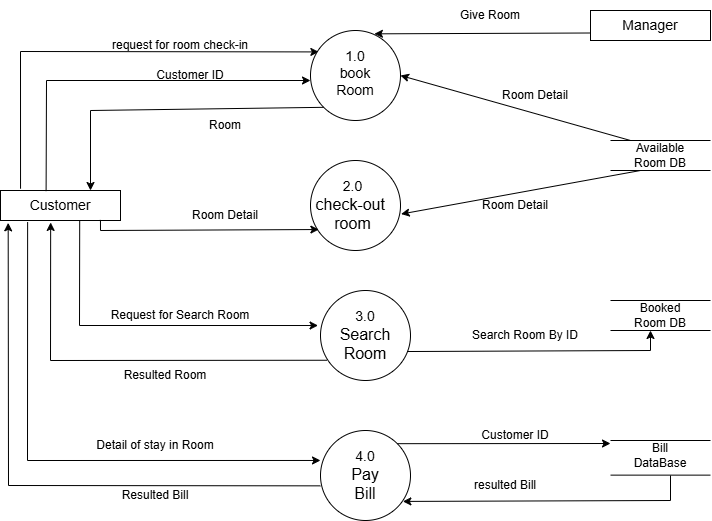


Figure ‑ DFD level-1 for Hotel management system

# External interface requirement (Screens)

## Screen-1: Hotel Reservation Form

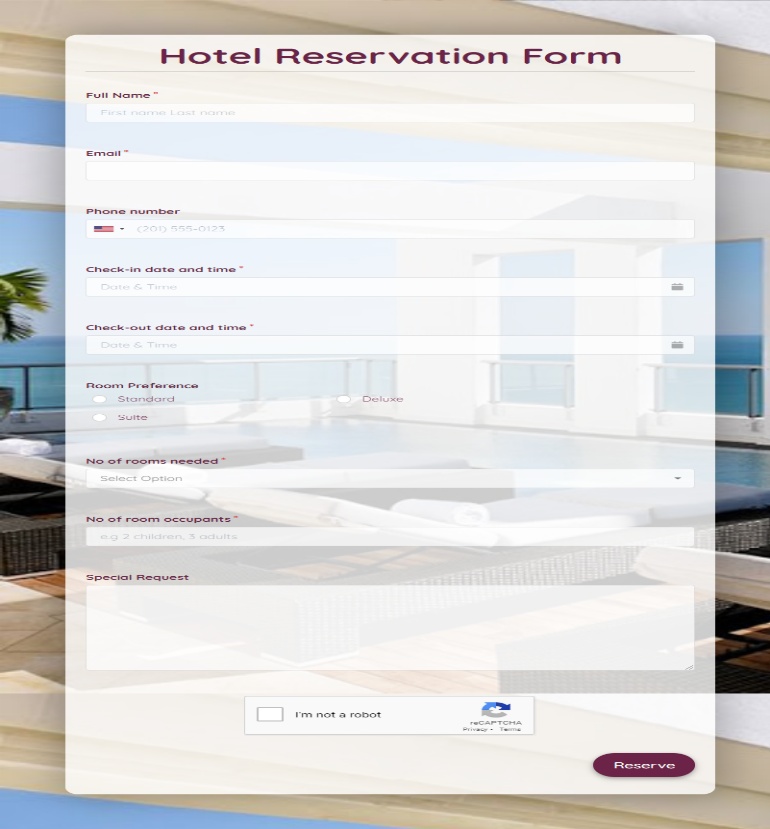


Figure ‑ Screen-1: Reservation Form

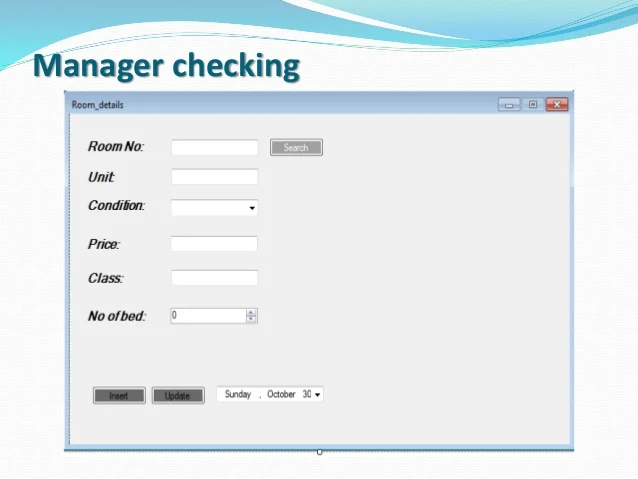
**Purpose:** This form will allow the target end-users to reservation in the system. To register , the following information will be encoded in the system.

Table ‑ Screen element of Reservation form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Full Name | Textbox | M | 1 | Username field should be editable and accept the Username. |
| 2 | Email | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| 3 | Phone Number | Textbox | M | 1 | Phone No. field should be editable and accept the Phone Number and display. |
| 4 | Check-in Date | Date | M | 1 | Check-in Date field should be editable and accept the Date and display as Date format. |
| 5 | Check-out Date | Date | M | 1 | Check-out Date field should be editable and accept the Date and display as Date format. |
| 6 | No Room Need | Drop Down | M | 1 | No. Room Needed field Should be editable and accept Number that select Drop Down. |
| 7 | I agree to the terms | Checkbox | M | 1 | Checkbox for accepting terms and condition. |
| 8 | Reserve | Button | ------ | ------ | Reserve is a button for store the entered data into database. |

## Screen-2: Check-in Form

Figure ‑ Screen-2: Check-in Form



**Purpose:** This form will be used by the system’s users to access records and features of the system. The users will input the correct combination of their room Detail to be able to access room that available to the system.

Table ‑ Screen element of Check-in form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Room No | Textbox | M | 1 | Room No field should be editable and accept the ID for that Room. |
| 2 | Unit | Textbox | M | 1 | Unit field should be editable and accept the unit of that Room . |
| 3 | Price | Textbox | M | 1 | Price field should be editable and accept that true price of particular Room. |
| 4 | Class | Textbox | M | 1 | Class field should be editable and accept that class (Apartment ) of that Room. |
| 5 | No. of Bed | Drop Down | M | 1 | No. of Bed is Drop Down list that refers No. of Bed required by Customer. |
| 6 | Date | Drop Down | M | 1 | Date is Drop Down list that refers Date Chosen by Customer. |
| 7 | Check in | Button | ------ | ------ | Check-in button navigates to another page even if valid Check-in credentials. |

## Screen-3: Payment

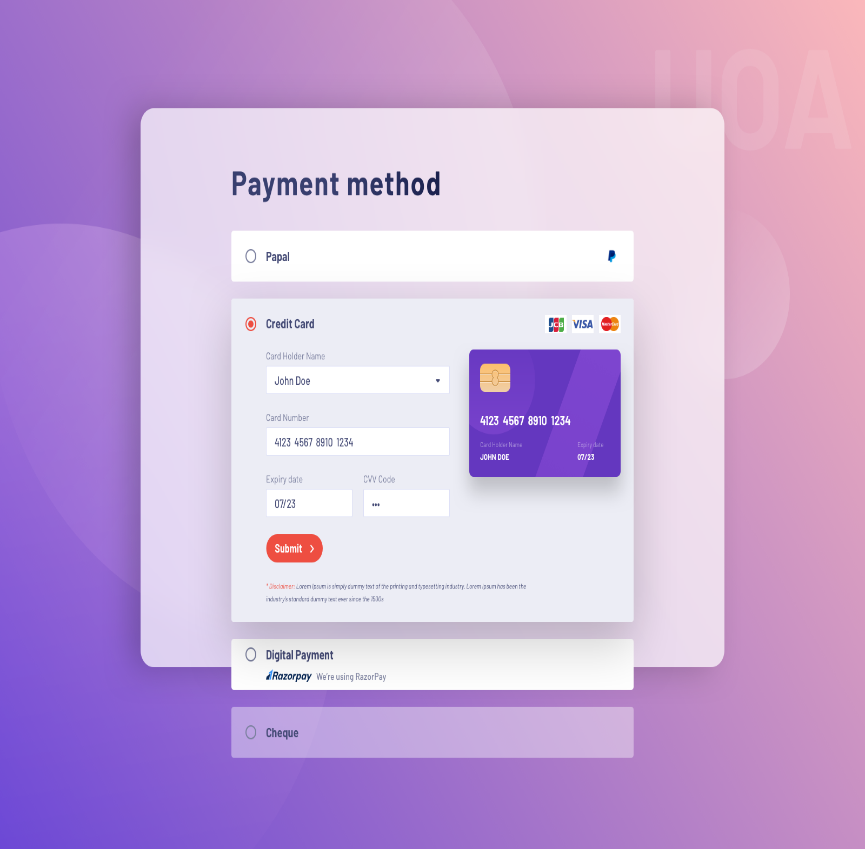


Figure ‑ Screen-3: Payment

**Purpose:** This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

Table ‑ Screen element of Payment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Chose Card | Selected | M | 1 | Chose Card field should be editable and fill the Card Detail. |
| 2 | Card Holder Name | Textbox | M | 1 | Card Holder Name field should be editable and accept The Card Holder Name. |
| 3 | Card Number | Textbox | M | 1 | Card Number field should be editable and accept only Number as 16 Digit Number. |
| 4 | Expiry Date | Date | M | 1 | Expiry Date field should be editable and accept only Date as Last Date. |
| 5 | CVV Code | Textbox | M | 1 | CVV Code field should be editable and accept only Number as 3 Digit Number. |
| 6 | Submit | Button | ----- | ----- | Submit is a button for store the entered data into database. |
| 7 | Cancel | Button | ----- | ----- | Cancel is a button for close add borrower dialog. |

## Screen-4: Hotel Feedback Form

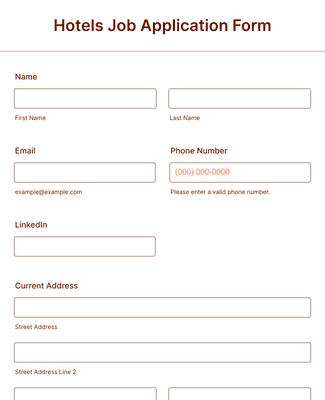
Figure 3.4‑ Screen-3: Feedback Form

**Purpose:** This module will allow the system administrator to add, edit, update or delete borrowers of book. The admin can add borrower information and manage it.

Table 3.4‑ Screen element of Hotel Feedback Form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Rates | Rates | M | 1 | Rates field should be editable and fill the star with review as form of point. |
| 2 | Suggestion | Textbox | O | 1 | Suggestion field should be editable and fill the Detail and Suggestion given by Customer. |
| 3 | Email | Textbox | M | 1 | Email field should be editable and accept The Email Id of Customer. |
| 4 | Submit | Button | ----- | ----- | Submit is a button for store the entered data into database. |

## Screen-5: Staff Application Screen

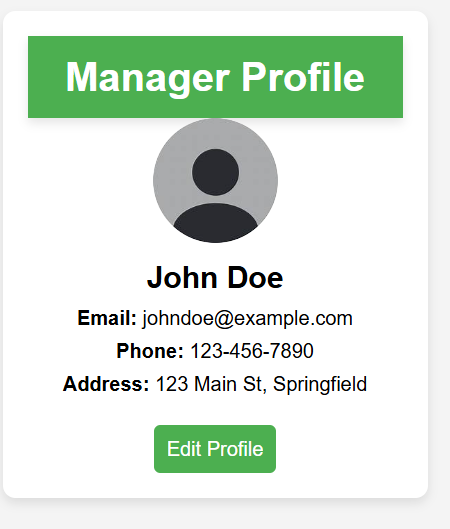
Figure 3.5‑ Screen-5: Staff Application Screen

**Purpose:** This module will allow the system administrator to application for Staff or Employee. The admin can add staff information and manage it.

Table 3.5‑ Screen element of Hotel Staff Application Form

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | First Name | Textbox | M | 1 | First name field should be editable and accept the Username. |
| 2 | Last Name | Textbox | M | 1 | Last name field should be editable and accept the Username. |
| 3 | Email | Textbox | M | 1 | Email field should be editable and accept the email with proper format. |
| 4 | Phone Number | Textbox | M | 1 | Phone No. field should be editable and accept the Phone Number and display. |
| 5 | Current Address | Textbox | M | n | Address field should be editable and accept the Address as String. |
| 6 | Submit | Button | ----- | ----- | Submit is a button for store the entered data into database. |

## Screen-6: View Profile Screen

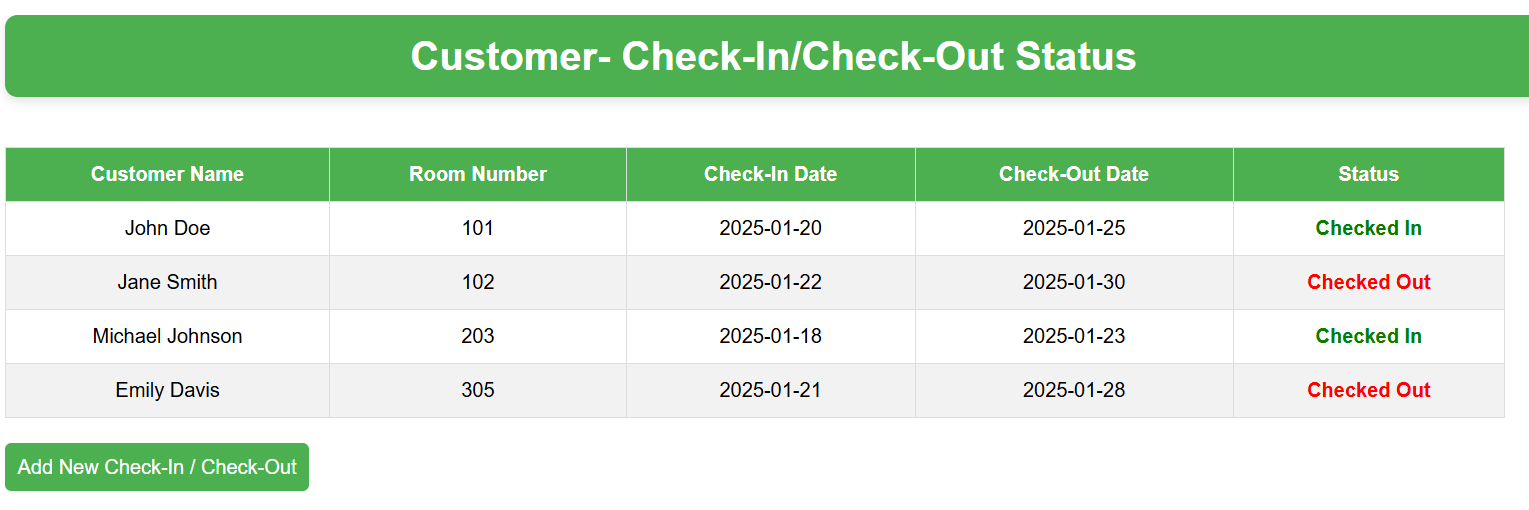
Figure 3.6‑ Screen-6 View Profile Screen

**Purpose:** This module will allow the system administrator to View Profile for Manager. The admin can view manager profile also.

Table 3.6‑ Screen element of view profile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Manager Profile | Header | M | 1 | Title of the profile section. |
| 2 | Profile Picture | Image | M | 1 | Displays a placeholder profile picture. |
| 3 | Manager Name | Textbox (Read-only) | M | 1 | Displays the manager's full name. |
| 4 | Email | Textbox (Read-only) | M | 1 | Displays the manager's email address. |
| 5 | Phone | Textbox (Read-only) | M | 1 | Displays the manager's phone number. |
| 6 | Current Address | Textbox | M | n | Address field should be editable and accept the Address as String. |
| 7 | Submit | Button | ----- | ----- | Submit is a button for store the entered data into database. |

## Screen-7: View Check-in Check-out Screen

****Figure 3.7‑ Screen-7 View check-in check-out Status

**Purpose:** This module will allow the system administrator to Customer for check their Status. The Customer

can add new Check-in and Check-out Status and manage it

Table 3.7‑ Screen element of check-in check-out Status

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Guest Name | Textbox | M | 1 | Field to enter the guest's name for search. |
| 2 | Room Number | Textbox | O | 1 | Field to enter the room number for search. (Optional) |
| 3 | Check-in Date | Datepicker | O | 1 | Field to select the check-in date for search. (Optional) |
| 4 | Check-out Date | Datepicker | O | 1 | Field to select the check-out date for search. (Optional) |
| 5 | Status | TextBox | M | 1 | Field to initiate the search based on entered criteria. |
| 6 | Add new Chececk-in Button | Button | O/M | 1/N | Field that new check-in or check-out for customer |

## Screen-8: View Customer requirement Screen for Staff

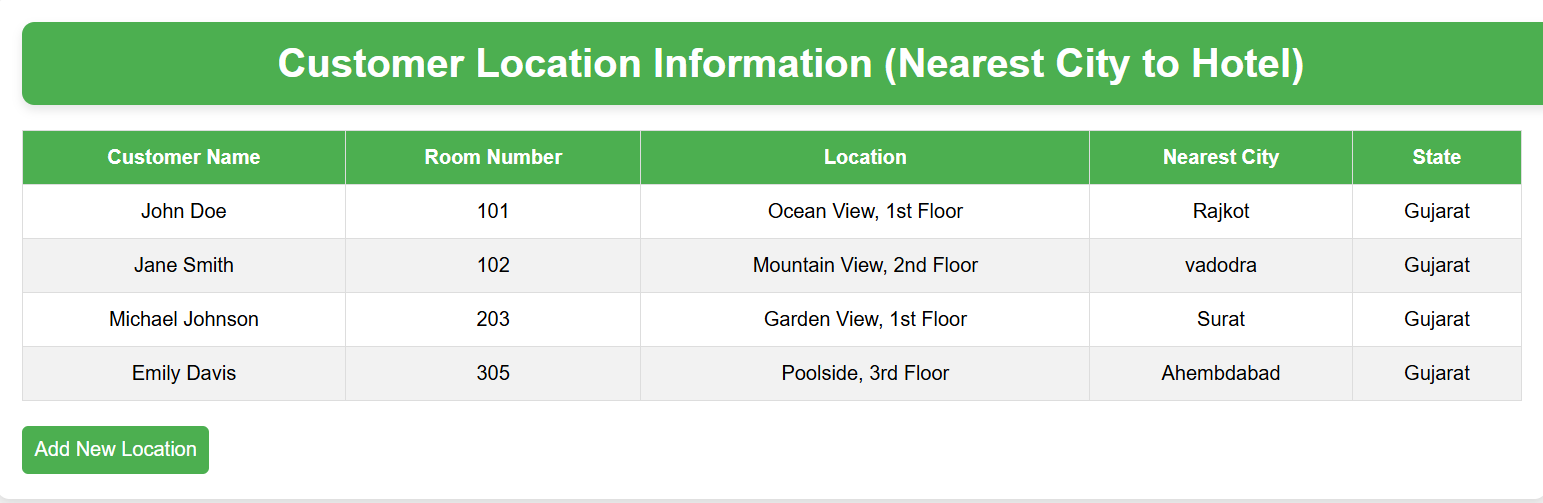
Figure 3.8‑ Screen-8 View Customer requirement screen

**Purpose:** This module will allow the system administrator to Customer requirement to complete by staff. The staff can add customer information and manage it

Table 3.8‑ Screen element of Customer requirement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Customer Name | Textbox (Read-only) | M | 1 | Displays the name of the customer who submitted the requirement. |
| 2 | Room Number | Textbox (Read-only) | M | 1 | Displays the room number associated with the requirement. |
| 3 | Requirement | Textbox (Read-only) | M | 1 | Displays the specific requirement submitted by the customer. |
| 4 | Status | Label (with color-coding) | M | 1 | Displays the current status of the requirement (e.g., Pending, Completed, Canceled). |
| 5 | Actions | Button | M | 1 | Button to allow staff to update the status or take further action on the requirement. |
| 6 | Add Requirement | Button | o | 1 | Button to allow customer for add new Requirement for them. |

## Screen-9: View Customer Location Screen for Transport Member

 Figure 3.9‑ Screen-9 View Customer Location screen

**Purpose:** This module will allow the system Transport Member to vie customer location. The member can add customer information and manage it

Table 3.9‑ Screen element of Customer location

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr. | Screen Element | Input Type | O/M | 1/N | Description |
| 1 | Customer Name | Textbox (Read-only) | M | 1 | Displays the name of the customer. |
| 2 | Room Number | Textbox (Read-only) | M | 1 | Displays the room number assigned to the customer. |
| 3 | Location | Textbox (Read-only) | M | 1 | Displays the specific location of the room (e.g., Ocean View, 1st Floor). |
| 4 | Nearest City | Textbox (Read-only) | M | 1 | Displays the nearest city to the hotel. |
| 5 | State | Textbox (Read-only) | M | 1 | Displays the state where the hotel is located. |

# Database design

## List of Tables

* Room
* Customer
* Staff
* Manager
* Payment
* Bill

Table ‑ Table: Room

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| RoomID | int | NN | PK (Auto Increment) |  |
| ApparmentName | varchar(100) | NN |  |  |
| FloorID | int | NN |  |  |
| NoOfBed | int | AN |  |  |
| AC/NoneAC | Boolean | AN |  |  |

Table ‑ Table: Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| CustomerID | int | NN | PK (Auto Increment) |  |
| RoomID | int | NN | FK | Reference of Room Table |
| Check-InDate | DateTime | NN |  |  |
| Check-OutDate | DateTime | AN |  |  |
| CustomerName | Varchar(100) | NN |  |  |
| Email | Varchar(100) | AN |  |  |
| Contact | number(10,0) | NN |  |  |

Table ‑ Table: Staff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| StaffID | int | NN | PK (Auto Increment) |  |
| StaffName | varchar(100) | NN |  |  |
| IsManager | Boolean | AN |  |  |
| Designation | varchar(100) | AN |  |  |
| Email | Varchar(100) | AN |  |  |
| Salary | Decimal(10,4) | AN |  |  |

Table ‑ Table: Manager

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| ManagerID | int | NN | PK (Auto Increment) |  |
| ManagerName | varchar(100) | AN |  |  |
| Gender | varchar(100) | AN |  |  |
| DOB | DateTime | AN |  |  |
| Email | Varchar(100) | AN |  |  |
| Department | varchar(100) | AN |  |  |
| Contact | number(10,0) | AN |  |  |
| Decignation | Varchar(100) | AN |  |  |

Table ‑5 Table: payment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| CardID | int | NN | PK (Auto Increment) |  |
| CustomerID | int | NN | FK | Reference of Customer Table |
| CardHolderName | varchar(100) | AN |  |  |
| Amount | decimal(20,4) | NN |  |  |
| CVV Number | int | NN |  |  |
| PaymentDate | DateTime | AN |  |  |
| BillID | int | NN |  |  |

Table ‑6 Table: Bill

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column | Data Type | Null | Keys & Constrains | Default Value & Description |
| BillID | int | NN | PK (Auto Increment) |  |
| CustomerID | int | NN | FK | Reference of Customer Table |
| CustomerName | varchar(100) | AN |  |  |
| Amount | decimal(20,4) | NN |  |  |
| Check-inDate | DateTime | AN |  |  |
| Check-OutDate | DateTime | AN |  |  |

# Stories and Scenario

## Story-1: Book a Room for Customer in Hotel

|  |  |  |
| --- | --- | --- |
| *Story # S1* | : | As a Customer,  I want to book a new room in hotel  So that I can easily find and reserve it. |
| Priority | **:** | High |
| Estimate | **:** | XL |
| Reason | **:** | The ability to book a hotel room is crucial for ensuring seamless customer reservations and an efficient hotel management system. |

### Scenario# S1.1

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.1* | : | Booking a New Room with Valid Information |
| Prerequisite | **:** | The hotel receptionist or the customer is logged into the hotel management system. |
| Acceptance Criteria | **:** | **Given:**  The receptionist/customer is navigated to the room booking page. Valid customer and booking details (such as check-in date, check-out date, room type, and payment details) are entered.  **When:**  The user selects the "Book Room" option.  The user enters valid booking details.  The user clicks the "Confirm Booking" button to finalize the reservation.  **Then :**  The system successfully books the room. A confirmation message is displayed with the booking reference number. The room status is updated to "Reserved" in the system. |

### Scenario# S1.2

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.2* | : | Booking a Room with Invalid Information |
| Prerequisite | **:** | The hotel receptionist or the customer is logged into the hotel management system. |
| Acceptance Criteria | **:** | **Given:** The receptionist/customer is on the room booking page.  **When:**  The user selects the "Book Room" option.  The user enters incomplete or incorrect booking details (e.g., missing check-in date, invalid payment method).  The user clicks the "Confirm Booking" button.  **Then :** The system displays appropriate error messages for incorrect or missing information. The room is not booked. |

### Scenario# S1.3

|  |  |  |
| --- | --- | --- |
| *Scenario# S1.3* | : | Attempting to Book a Room That is Already Reserved. |
| Prerequisite | **:** | The hotel receptionist or the customer is logged into the hotel management system and is on the room booking page. |
| Acceptance Criteria | **:** | **Given**: The room information (room type, availability status, and other relevant details) is available, and the room is already booked by another customer.  **When**: The user selects a room that is already reserved.  The user attempts to confirm the booking.  **Then**: The system prevents double booking and displays a message indicating that the selected room is unavailable. |

## Story-2: Search Book

|  |  |  |
| --- | --- | --- |
| *Story # S2* | : | As a Librarian or member,  I want to search for books by title, author, or keyword,  So that I can quickly find books that match my interests. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Implementing a search functionality is essential for enhancing the user experience, as it allows librarian and member to efficiently discover and access the library's resources. |

## Story-3: Manage due date for borrowed book

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As Librarian,  I want to manage due dates for borrowed books  So that I can ensure that books are returned on time and avoid overdue fines. |
| Priority | **:** | High |
| Estimate | **:** | M |
| Reason | **:** | Proper due date management is crucial for maintaining the library's collection and ensuring that books are available for all members. |

## Story-4: Renew book

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As Librarian,  I want to renew a book that I have borrowed  So that I can extend my borrowing period if needed. |
| Priority | **:** | Medium |
| Estimate | **:** | M |
| Reason | **:** | Book renewal functionality is a convenience feature for librarian, allowing them to keep a book for an extended period if no one else has requested it. |

## Story-5: Generate a report on book usage and availability

|  |  |  |
| --- | --- | --- |
| *Story # S3* | : | As Librarian,  I want to generate reports on library usage and book availability  So that I can make informed decisions about library’s books. |
| Priority | **:** | Medium |
| Estimate | **:** | L |
| Reason | **:** | Reporting functionality helps librarians track the usage of library resources and make data-driven decisions to improve services and collections. |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Hotel Management | Test Designed by: | PUSHTI MARKANA |
| Module Name: | **View Profile** | **Test Designed date:** | 12-02-2025 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 15-03-2025 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: Web site should be accessible | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | Login to page with valid credential | Functional | Login to Hotel management system page through valid credential | TC\_001 |
| TC\_002 | Login to page with invalid credential | Functional | Login to Hotel management system page through invalid credential | TC\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Profile Page with valid credential |
| **Test Type** | Functional |
| **Test Priority** | High |
| **Pre-condition** | User should be Properly login |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **BUG ID** |
| 1 | Access to View Profile Page | The Page launched properly | Page launched successfully | Pass |  | Page in website |  |
| 2 | Enter valid Username in username field | Username field should be editable and accept the Username | Username input accepted | Pass |  | Username: PushtiMarkana |  |
| 3 | Enter valid Email id | Email id field should be editable and accept the email id | Email Address input accepted | pass |  | Email: pushti@gmail.com |  |
| 4 | Enter Phone | Phone field should be editable and accept the phone | Phone 10 digit input is accepted | pass |  | Phone:1234567890 |  |
| 5 | 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 |  |
| 6 | Click on Submit button | User should view to profile and navigated to Profile | User navigated to profile and username should br display in top of the right side. | pass |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | View Profile page with invalid credential |
| **Test Type** | Functional |
| **Test Priority** | Medium |
| **Pre-condition** | View Profile page can acceible |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Step** | **Test Case Description** | **Expected Result** | **Actual Result** | **Status** | **Comment** | **Data** | **Bug ID** |
| 1 | Verify that User is not able to profile 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 Emailid | Should be display an error message enter wrong email | Display an error of worng Email | fail | performa a validation function fix it |  |  |
| 4 | 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 |  |  |  |
| 5 | 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 |

------2-----------

|  |  |  |  |
| --- | --- | --- | --- |
| Project Name: | Hotel Management | Test Designed by: | PUSHTI MARKANA |
| Module Name: | **Reservation** | **Test Designed date:** | 12-02-2025 |
| Release Version: | **1.0** | **Test Executed by:** | **R. B. Gondaliya** |
|  |  | **Test Execution date:** | 15-03-2025 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pre-condition: Web site should be accessible | | | | |
| Test Case ID | **Test Title** | **Test Type** | **Description** | **Test Case ID** |
| TC\_001 | Reservation with valid credential | Functional | Reservation to Hotel management system page through valid credential | TC\_001 |
| TC\_002 | Reservation with invalid credential | Functional | Reservation to Hotel management system page through invalid credential | TC\_002 |

|  |  |
| --- | --- |
| **Test Case Title** | Reservation with valid cradital |
| **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 | Launch application with the given url | The site launched properly | Site launched successfully | Pass |  | https://accounts.google.com/Servicereservation |  |
| 2 | Verify that the login screen and verify that any room is available in Hotel | All listed control displayed properly on the page and room is Available for Customer | Reservation page loaded successfully and show the rooms id. | Pass |  |  |  |
| 3 | Verify is customer is already login | All Data about Customer like user and pass is correct | Cursor focus in Room id | Pass |  |  |  |
| 4 | Click Button and Verify Email, Password ,Username and id and other Information | Make the reservation of room for customer and he book the room | Customer is book the room | Fail |  |  |  |

|  |  |
| --- | --- |
| **Test Case Title** | Reservation with valid cradital |
| **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 reservation in Hotel | Should be display an error message enter | Display an error of wrong for Do not Reservation | 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 login in websit for Reservation | Should be display an error message enter Please login before Reservation | Display an error of login | fail |  |  |  |
| 4 | Verify that Email Address isvalid | Should be dispay an error message enter Valid Email | Display an error for Valid Email | Pass |  |  | B2\_002 |

# References

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