

**Software Requirements Specification**

for

**Kingfisher Hotel Management System**

Proposal, Technical Project

**Group Name: Redliners**

**ID: G08**

Client: Kingfisher Beach Resort, Yala

Version: 1.0

Prepared by

Redliners

|  |  |
| --- | --- |
| **Member Name** | **Registration Number** |
| A.A. Rifath | SA23089754 |
| H.G.S. Sanchitha | SA23428324 |
| M.M.N.S. Bandara | SA23442016 |
| E.T. Rusiru | SA23438538 |
| H.K. Dilanjan | SA23439450 |
| P.M.C.R. Bandara | SA23514522 |

Table of Contents

[LIST OF TABLES ii](#_Toc15626866)

[LIST OF FIGURES ii](#_Toc15626867)

[Revision History iii](#_Toc15626868)

[1 INTRODUCTION 1](#_Toc15626869)

[1.1 Purpose 1](#_Toc15626870)

[1.2 Intended Audience and Reading Suggestions 1](#_Toc15626871)

[1.3 Product Scope 1](#_Toc15626872)

[2 OVERALL DESCRIPTION 2](#_Toc15626873)

[2.1 Product Perspective 2](#_Toc15626874)

[2.2 Product Functions 2](#_Toc15626875)

[2.3 User Classes and Characteristics 4](#_Toc15626876)

[2.4 Operating Environment 5](#_Toc15626877)

[2.5 Design and Implementation Constraints 5](#_Toc15626878)

[2.6 User Documentation 5](#_Toc15626879)

[2.7 Assumptions and Dependencies 5](#_Toc15626880)

[2.8 User Interfaces 5](#_Toc15626881)

[2.8.1 Customer management page 6](#_Toc15626882)

[2.8.2 Booking management Interface 6](#_Toc15626883)

[2.8.3 Employee management Interface 6](#_Toc15626883)

[2.8.4 Inventory management Interface 6](#_Toc15626883)

[2.8.5 Event management Interface 6](#_Toc15626883)

[2.9 Hardware Interfaces 7](#_Toc15626884)

[2.10 Software Interface 7](#_Toc15626885)

[2.11 Communications Interfaces 8](#_Toc15626886)

[3 System Functional Features 9](#_Toc15626887)

[3.1 Database Requirements 9](#_Toc15626888)

[3.2 Create User 9](#_Toc15626889)

[4 NON-FUNCTIONAL REQUIREMENTS 10](#_Toc15626890)

[4.1 Performance Requirements 10](#_Toc15626891)

[4.2 Safety Requirements 10](#_Toc15626892)

[4.3 Security Requirements 10](#_Toc15626893)

[4.4 Software Quality Attributes 10](#_Toc15626894)

[4.5 Business Rules 11](#_Toc15626895)

[5 References 11](#_Toc15626896)

[6 OTHER REQUIREMENTS 11](#_Toc15626896)

# LIST OF TABLES

[Table 3.1.1: Use Case Description for Create User 9](#_Toc15626897)

[Table 3.1.2: Use Case Description for Create User 9](#_Toc15626897)

# LIST OF FIGURES

[Figure 2.8.1.1: Architecture Diagram of the system [1] 2](#_Toc15626898)

[Figure 2.8.1.1: Dataflow diagram of the system [2] 3](#_Toc15626899)

[Figure 2.8.1.1: User case Diagram for the application 4](#_Toc15626901)

[Figure 2.8.1.1: Customer management Interface 6](#_Toc15626902)

[Figure 2.8.2.1: Booking Interface 7](#_Toc15626903)

Figure 2.8.2.2: Search room Interface………………………………………………8

[Figure 2.8.2.3: Check In UI 7](#_Toc15626903)

[Figure 2.8.2.4: Check Out UI 7](#_Toc15626903)

[Figure 2.8.3.1: Employee management Interface 7](#_Toc15626903)

[Figure 2.8.4.1: Inventory management Interface 7](#_Toc15626903)

[Figure 2.8.5.1: Event management Interface 7](#_Toc15626903)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason | Version |
|  |  |  |  |
|  |  |  |  |

# INTRODUCTION

## Purpose

The purpose of this document is to define the software requirements of the Kingfisher Hotel Management System, which will be developed with Java. This document serves as a detailed guide for developers, stakeholders, and project team members to ensure appropriate understanding of the system requirements. The Kingfisher Hotel Management System will offer automation of simple hotel operations such as Room Reservations, Customer Management, Employee Management, Inventory Management, and Payment Processing. The system aims to enhance business efficiency, reduce errors due to human intervention, and ensure overall customer satisfaction.

This report covers the key system features, technical requirements, user interaction, and quality assurance. It serves as a development guide for developers and provides a clear system construction guide. The first release of the system will manage in-house hotel activities and will be accessible and controllable by receptionists and hotel management staff only. The system is created as an offline program without direct interaction with the guests via the internet.

## Purpose Intended Audience And Reading Suggestions

* **Developers -** To understand the technical structure and functional requirements.
* **Project Manager -** Responsible for overseeing the project, ensuring deadlines are met, and aligning development with business goals.
* **Client Coordinator -** Communicating between the development team and the client, ensuring that the system meets the client’s expectations.
* **Business Analyst -** Ensures that the system aligns with Kingfisher Beach Resort's business goals.
* **Hotel Management (Client) –** Our client Mr. J.B Sehan Madhuka who is the owner of the beach resort.
* **Users (Receptionists & Hotel Management) -** Interact with the system for tasks such as booking rooms, managing customer data, and processing payments.
* **Quality Assurance Engineer (QA) -** Ensures the system meets performance, usability, and security standards through rigorous testing.
* **Technical Writer -** Documents system functionality, creating user manuals and reports to aid system understanding.

## Product Scope

The Kingfisher Hotel Management System is an internal system designed exclusively for hotel staff and management, replacing manual hotel processes with a digital and automated solution. The system will not have online access for customers but will be used internally by hotel staff to manage reservations, customer records, employee schedules, and inventory.

**Key Features and Functionalities:**

* **Room Reservation Management -** Automates room bookings, cancellations, and availability tracking to prevent overbooking.
* **Customer Data Management** - Maintains a centralized database of guest details, booking history, and special preferences.
* **Inventory Management** - Tracks hotel supplies and amenities, preventing shortages through stock level alerts.
* **Employee Management** - CRUD operations and maintains a centralized database of employee details, positions.
* **Reporting & Analytics** - Generates reports on revenue, booking trends, inventory usage, and employee data.
* **Event & Conference Management** - Manages hotel hosted events, tracking schedules, and resources.

**Business Goals & Benefits:**

The Kingfisher Beach Resort's Hotel Management System aims to improve efficiency, reduce manual errors, and streamline daily operations. By automating key tasks, it enhances customer service, optimizes staff scheduling, tracks inventory, and strengthens financial management. With integrated payment processing and real-time data access, the system enables better decision-making, ultimately improving overall service quality and operational performance.

# OVERALL DESCRIPTION

## Product Perspective

This is a new, standalone system for a resort to manage room bookings, inventory, employees, and events. It does not replace any old system. It automates manual work and improves efficiency.

The system will have different modules:

* **Booking system** - Handle room reservations, cancellations, and payments.
* **Inventory system** - Track stock levels, send low-stock alerts.
* **Reservations & payments management -** Managing payments and reservations for rooms etc.
* **Employee management** - Schedule employee jobs, store employee data.
* **Event & conference management** - Add, remove, update events and conferences.
* **Reports & Analytics** - Generate reports on revenue, bookings, and operations.

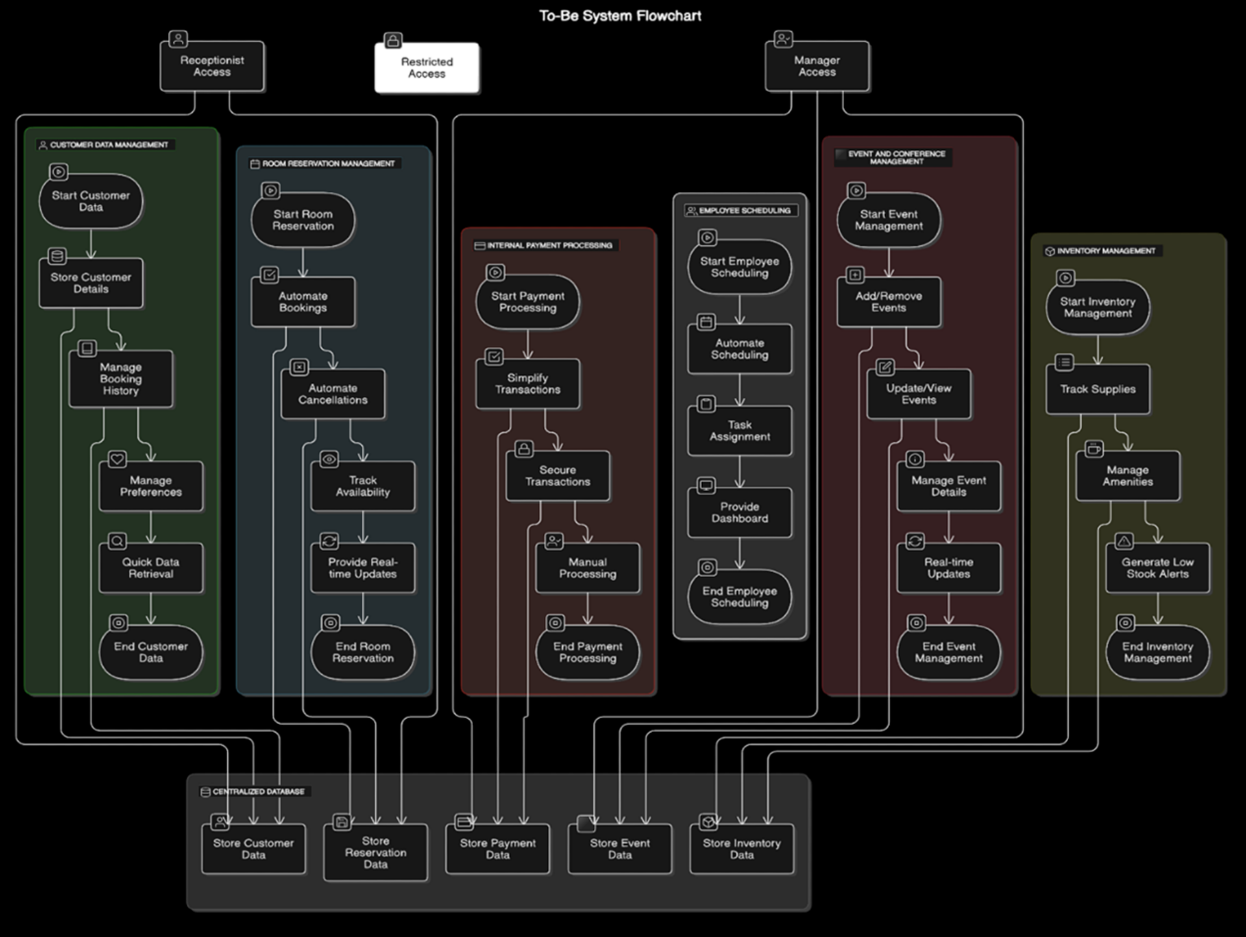


Figure .: Architecture Diagram of the system [1]

## Product Functions

* **Booking System** - Manage room reservations, cancellations, and payments.
* **Inventory System** - Track stock levels and send low stock alerts.
* **Reservations & Payments Management** - Handle payments and reservations for rooms and other services.
* **Employee Management** - Schedule staff shifts and store employee records.
* **Event & Conference Management** - Add, update, and remove events and conferences.
* **Reports & Analytics** - Generate reports on revenue, bookings, and overall operations.

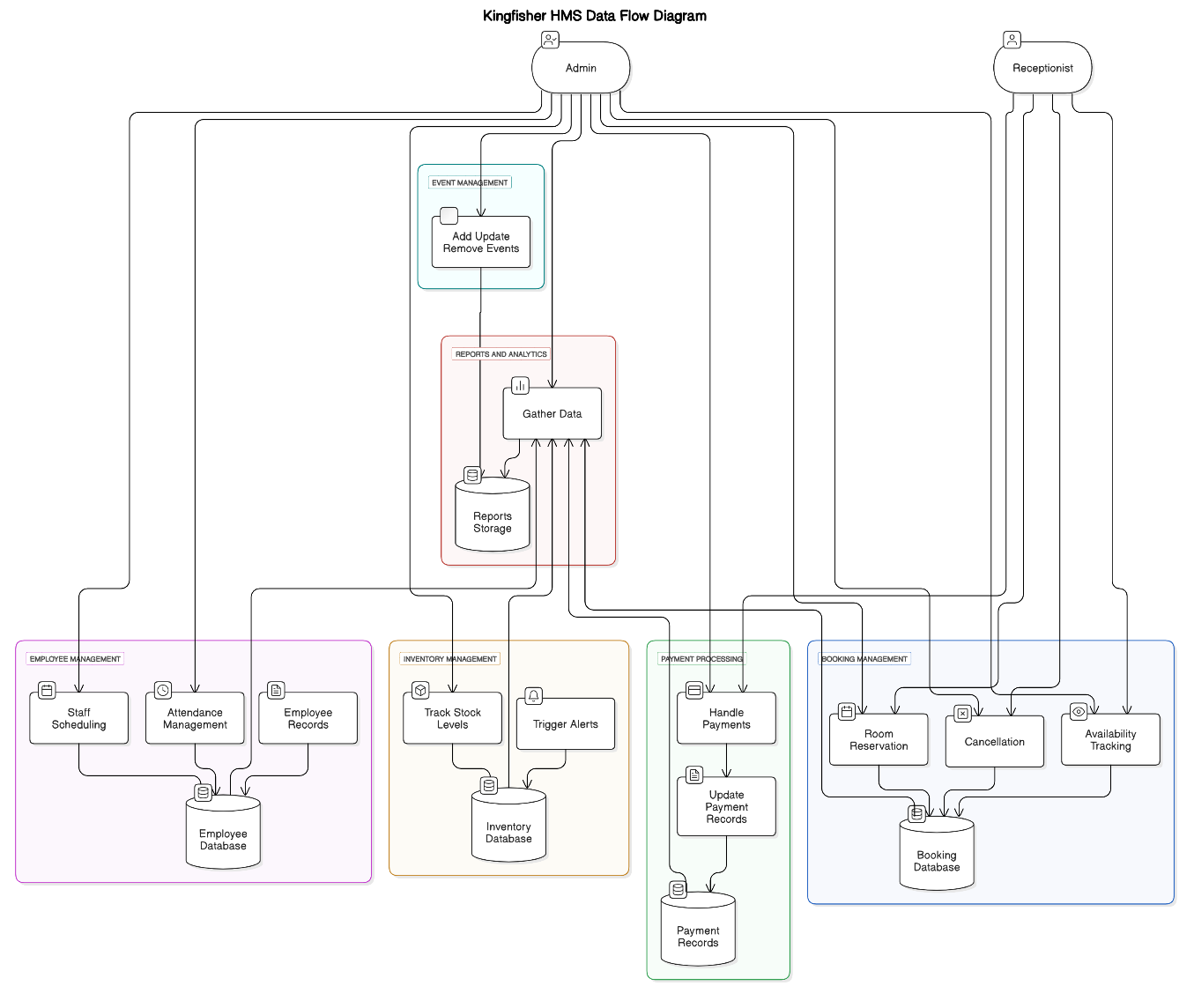


Figure .: Dataflow diagram of the system [2]

## User Classes and Characteristics

**Admin/ Manager:**

* Importance - This is the primary user who oversees the entire system and has the highest level of control.
* Frequency of Use - High
* Subset of Product Functions - Access to all functions, including:
  + Viewing and CRUD operations related to data of reports, inventory, suppliers, orders, employees, customers.
* Technical Expertise - Needs to understand the system functionalities.
* Privilege Levels - Full access.
* Experience - Significant experience in hotel management or similar systems.

**Employee:**

* Importance - High, as they are the operational backbone of the hotel.
* Frequency of Use - High.
* Subset of Product Functions: Viewing and limited CRUD operations related data of all functions.
* Technical Expertise - Medium, depending on role.
* Privilege Levels - Limited.
* Experience - Some experience in hotel management or similar systems.

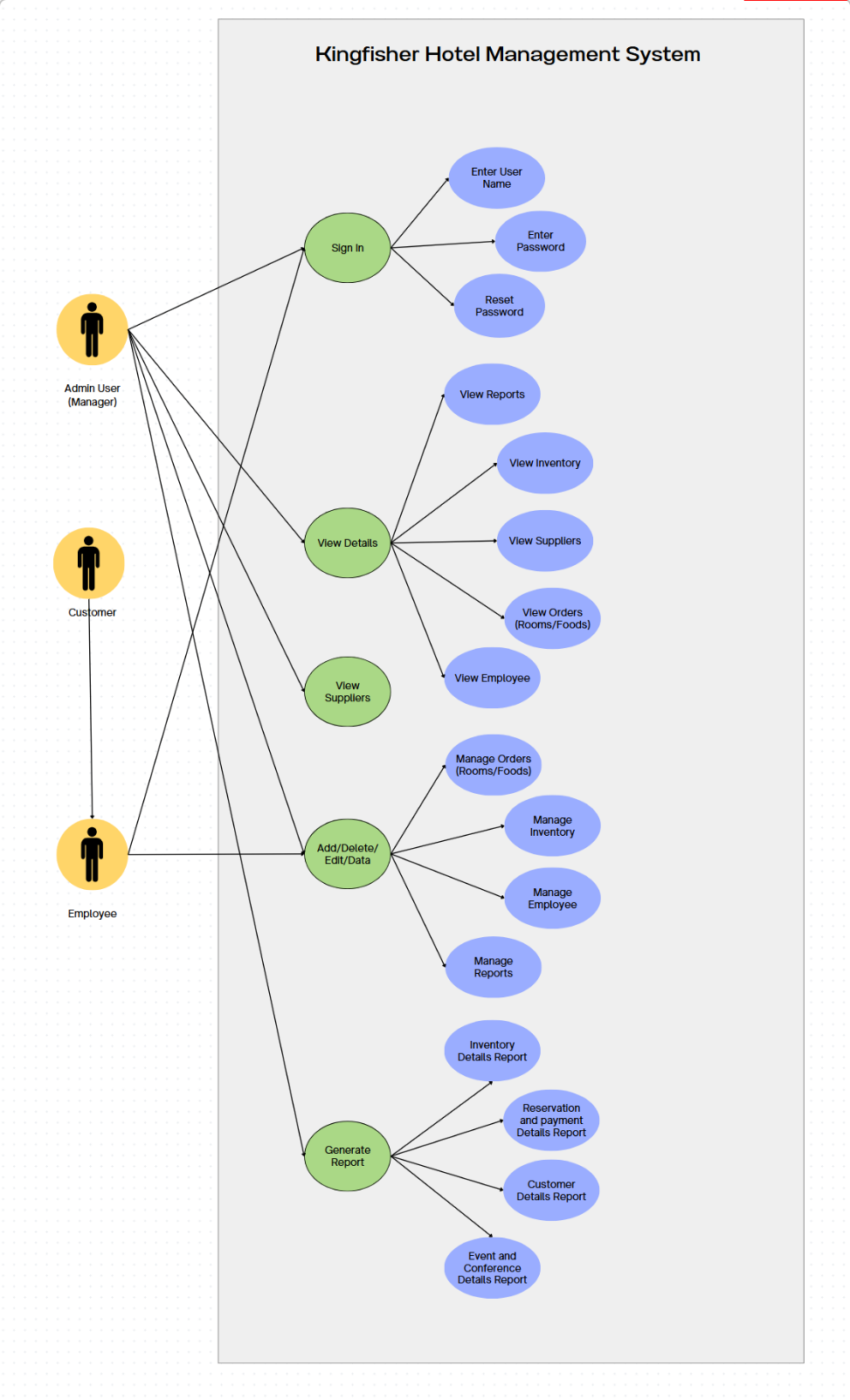


Figure .: User case Diagram for the application

## Operating Environment

* **Hardware** – Runs on local computers at the hotel
* **Operating System** – Compatible with Windows and Linux
* **Database** – Uses MySQL for local data storage.
* **Application Type** – Standalone, not online.
* **User Access** – Limited to hotel admin, receptionist, and authorized employees
* **Dependencies** – Works with local network and printers.

## Design and Implementation Constraints

The system must run smoothly on the hotel's existing hardware with minimal resource consumption. It should provide quick response times for booking and inventory updates. Security constraints include restricted access based on user roles. Future maintenance and potential issues from system updates or database modifications are key considerations. The system should also be compatible with any existing hotel policies and regulatory requirements.

## User Documentation

**User Manual**

* A comprehensive guide for hotel staff, covering system navigation, room booking, payment processing, inventory management, and employee scheduling.
* Includes step by step instructions with screenshots.

**Administrator Guide**

* A detailed guide for system administrators, focusing on user management, database maintenance, and system configuration.

**Training Materials**

* PowerPoint presentation.
* User manual.

**FAQs and Troubleshooting Guide**

* A document addressing common issues and their solutions.

## Assumptions and Dependencies

**Assumptions**

* The hotel staff will have basic computer literacy to operate the system.
* The client will provide access to necessary hardware for system installation.
* The internet connection at the hotel will be stable for real time updates and payment processing.
* The client will provide timely feedback during the requirement gathering and testing phases.
* The existing manual processes will remain operational during the transition to the new system.

**Dependencies**

* **MySQL Database:** The system relies on MySQL for data storage and management.
* **Java Runtime Environment (JRE):** The system requires JRE version 8 or higher to run.
* **Client Cooperation:** The project timeline assumes active participation from the client during requirement validation and user acceptance testing.

## User Interfaces

### Customer management page

Allows front desk staff to register new customers and input their details and allocate their rooms.

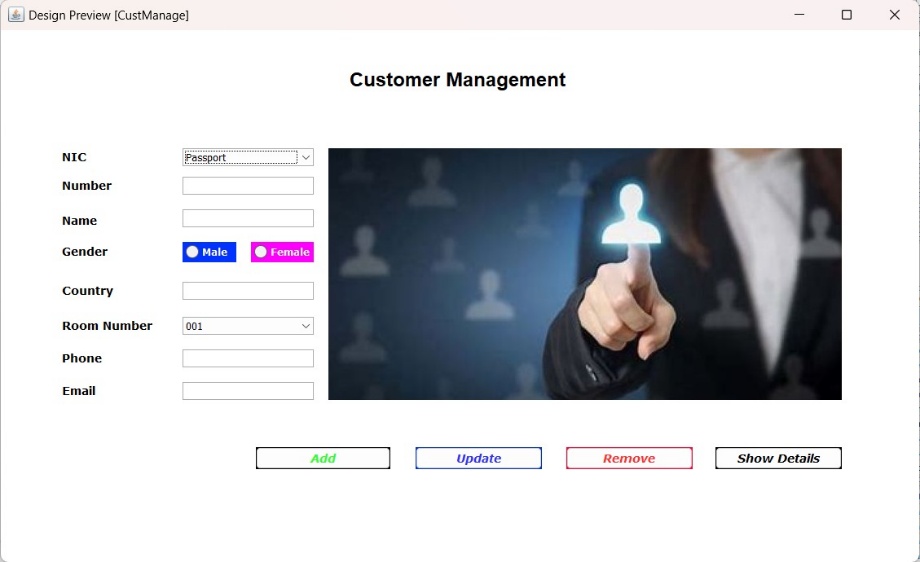


Figure .: Customer management Interface

### Booking management UI

Allows add new room, Staff use this UI for add new room as available room.

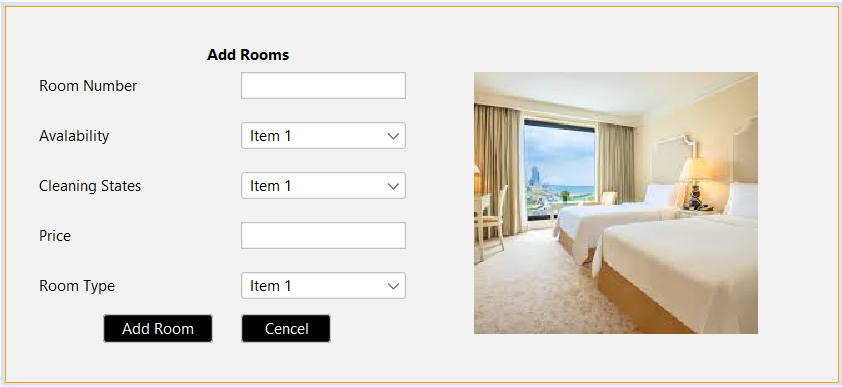


Figure .: Booking Interface

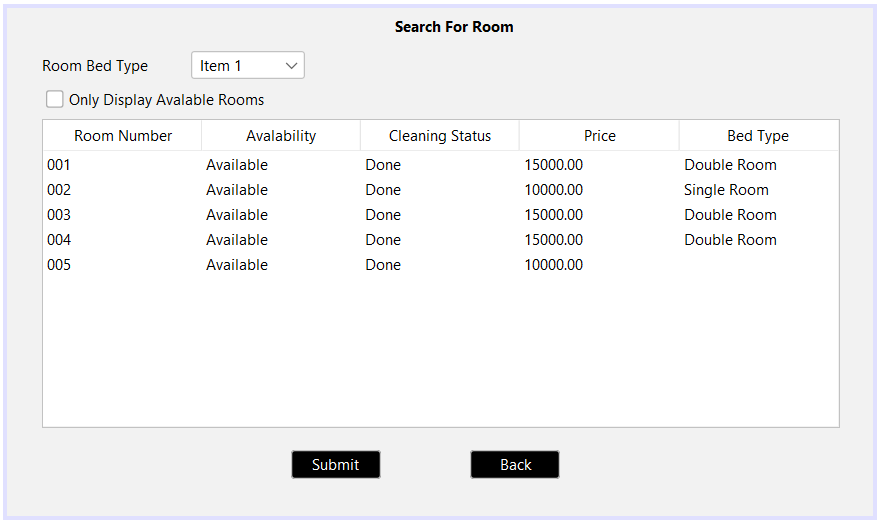


Figure .2: Search room Interface

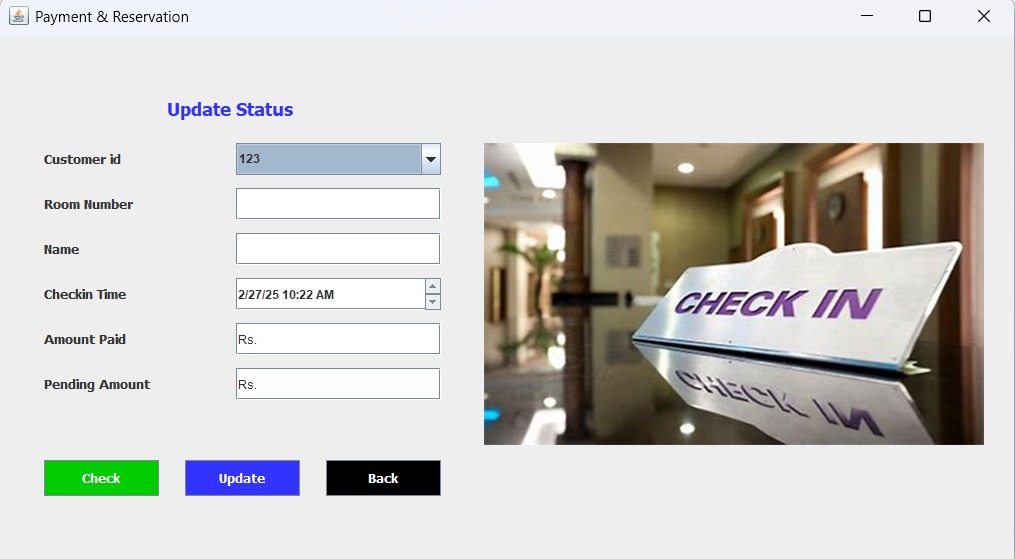


Figure .3: Check In UI

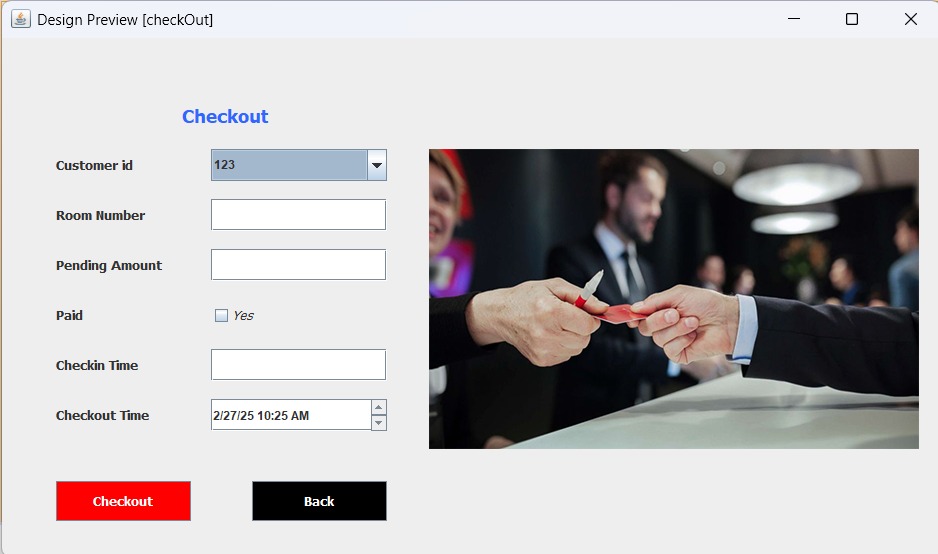


Figure .4: Check Out UI

### Employee management UI

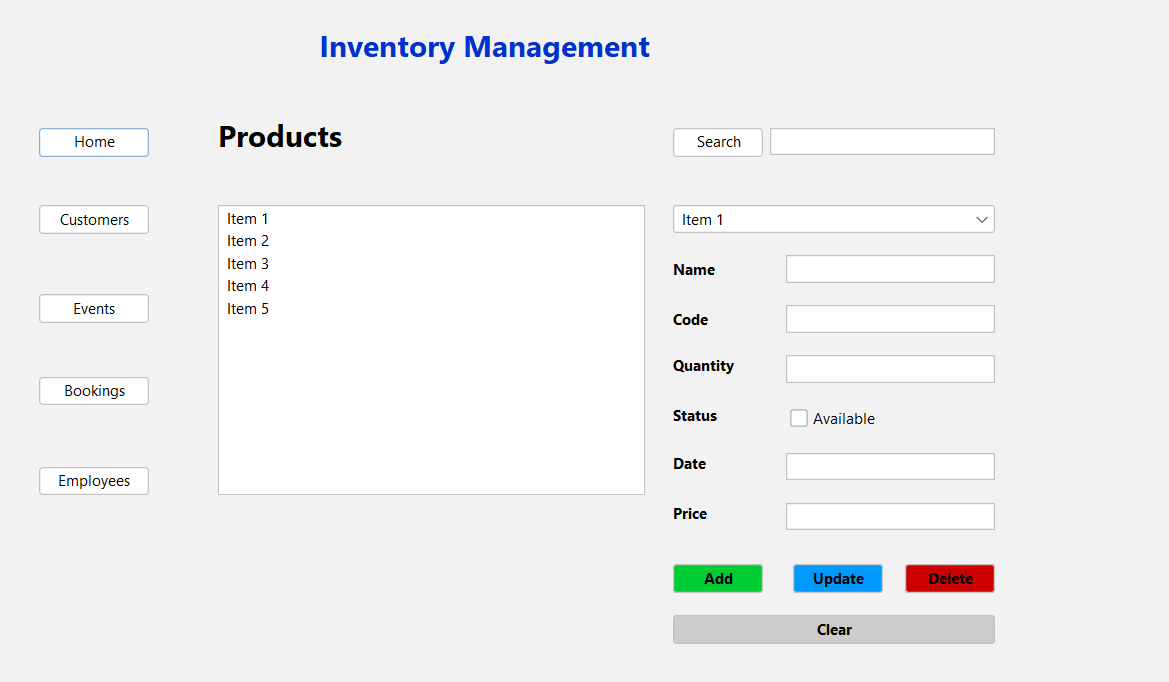
Allows to add, edit, remove and view employee data and assign job roles.



*Figure 2.8.3.1: Employee management Interface*

### Inventory management UI

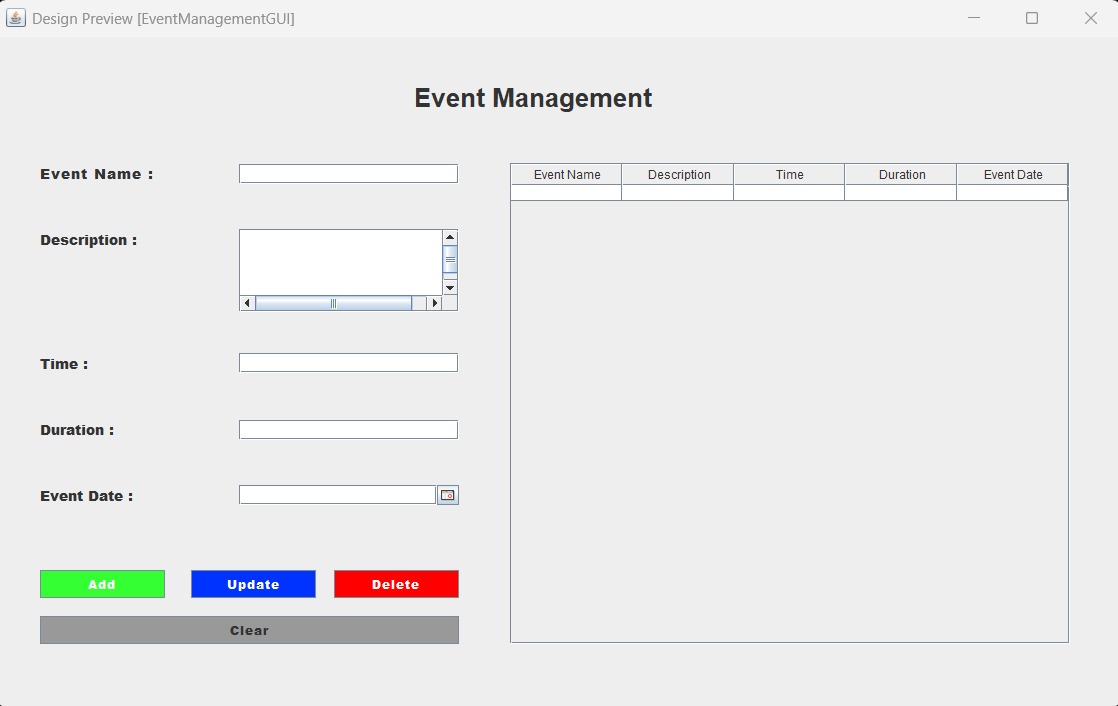
Allows to add, update, delete and view inventory items.



*Figure 2.8.4.1: Inventory management Interface*

### Event management UI

Allows to add, remove, update and view events hosted in the resort.



*Figure 2.8.5.1: Event management Interface*

## Hardware Interfaces

* Laptop/Desktop
* 1.8 GHz or faster processor. Quad-core or better recommended
* RAM needed: 8 GB or 16GB of RAM recommended.
* Hard disk space: Minimum of 800MB up to 100 GB of available space, depending on features installed.
* Wi-Fi router - Active Internet connection.

## Software Interface

The system will integrate with the following software components:

**MySQL Database**:

* Used for storing customer data, booking records, and inventory details.

**Java Runtime Environment (JRE)**:

* Required to run the system. Version 8 or higher is recommended.

**Operating System**:

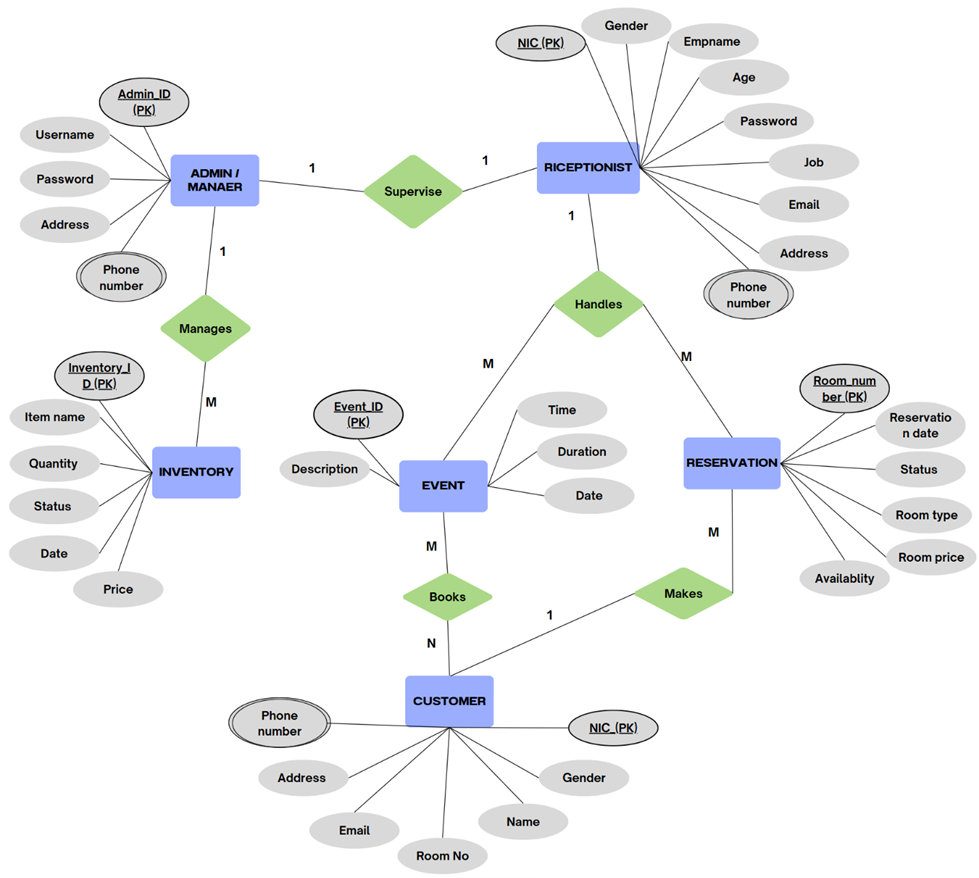
* Compatible with Windows and Linux operating systems.

## Communications Interfaces

The system operates as a standalone local application used only by hotel staff. Internet connectivity is required for some features such as database connectivity and email sending. All data processing happens within the system, ensuring secure and controlled access. The system will store and retrieve data from a local database, with role based access for the admin and receptionist.

# System Functional Features

## Database Requirements



## Create User

Table .: Use Case Description for Sign in

|  |  |  |
| --- | --- | --- |
| Use case ID | 01 | |
| Use case Name: | Sign in | |
| Actors: | Receptionist or Owner/Admin | |
| Pre- Conditions | Registered Receptionist or Owner should have a valid user account. | |
| Post Condition | User is successfully logged in. | |
|  | | |
|  | Action | System Response |
| Success Path | 1) User clicks on sign in option.  3) User enter details.  4) User clicks on submit button. | 2) Sign in page is displayed to user.  5) System validates the details.  6) System signs the user in. |
| Alternate Path |  | If user enters incorrect details on the application, it will show an error message. |
| Exception Path |  | In step No.5, if the data couldn’t be validated ask user to enter the correct details again. |
| Special Requirements |  | |

**Employee Management**

Table .2: Use Case Description for Add Employee

|  |  |  |
| --- | --- | --- |
| Use case ID | 02 | |
| Use case Name: | Add Employee | |
| Actors: | Owner | |
| Pre- Conditions | Owner should be logged into the system. | |
| Post Condition | Successfully add new Employee to the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1)Owner clicks on Employee button.  3) Owner enter the Employee details to add.  4) Owner clicks on Add Employee button. | 2) Displays the Employee Management UI  5) Validate Employee details.  6)Display Employee added successfully message. |
| Alternate Path |  | If the user enters wrong credentials, System shows an error message. |
| Exception Path | Error displayed fields are empty. |  |
| Special Requirements | Only owner can add new Employee to the system. | |

Table .3: Use Case Description for Update Employee

|  |  |  |
| --- | --- | --- |
| Use case ID | 03 | |
| Use case Name: | Update Employee | |
| Actors: | Owner | |
| Pre- Conditions | Owner should be logged into the system. | |
| Post Condition | Successfully update Employee to the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1)Owner clicks on Employee button.  3)Owner enter the new Employee details to add.  4) Owner clicks on Update Employee button. | 2) Displays the Employee Management UI.  5) Validate new Employee details.  6)Display update successfully message. |
| Alternate Path |  | If the user enters wrong credentials, System shows an error message. |
| Exception Path | Error displayed fields are empty. |  |
| Special Requirements | Only owner can update Employees in the system. | |

Table .4: Use Case Description for Delete Employee

|  |  |  |
| --- | --- | --- |
| Use case ID | 04 | |
| Use case Name: | Delete Employee | |
| Actors: | Owner | |
| Pre- Conditions | Owner should be logged into the system. | |
| Post Condition | Successfully delete user from the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1)Owner clicks on Employee button.  3)Owner enter NIC of Employee who is need to be delete.  4) Owner clicks on Delete Employee button. | 2) Displays the Employee Management UI.  5)Display delete successfully message. |
| Alternate Path |  |  |
| Exception Path |  |  |
| Special Requirements | Only Owner can delete users in the system. | |

Table .5: Use Case Description for View Employee

|  |  |  |
| --- | --- | --- |
| Use case ID | 05 | |
| Use case Name: | View Employee | |
| Actors: | Owner | |
| Pre- Conditions | Owner should be logged into the system. | |
| Post Condition | Successfully view Employee details from the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1)Owner clicks on Employee button.  3)Owner enter NIC of Employee who is need to be view. | 2) Displays the Employee Management UI.  4)Display Employee details successfully. |
| Alternate Path |  |  |
| Exception Path |  |  |
| Special Requirements | Only owner can view Employee details in the system. | |

**Customer Management**

Table .6: Use Case Description for Add Customer

|  |  |  |
| --- | --- | --- |
| Use case ID | 06 | |
| Use case Name: | Add Customer | |
| Actors: | Receptionist | |
| Pre- Conditions | Receptionist should be logged into the system. | |
| Post Condition | Successfully add new Customer to the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1) Receptionist clicks on Customer button.  3) Receptionist enter the Customer details to add.  4) Receptionist clicks on Add Customer button. | 2) Displays the Customer Management UI  5) Validate Customer details.  6)Display Customer added successfully message. |
| Alternate Path |  | If the Receptionist enters wrong credentials, System shows an error message. |
| Exception Path |  | Error displayed fields are empty. |
| Special Requirements | Only owner can add new Employee to the system. | |

Table .7: Use Case Description for Update Customer

|  |  |  |
| --- | --- | --- |
| Use case ID | 07 | |
| Use case Name: | Update Customer | |
| Actors: | Receptionist | |
| Pre- Conditions | Receptionist should be logged into the system. | |
| Post Condition | Successfully update Customer to the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1) Receptionist clicks on Customer button.  3) Receptionist enter the new Customer details to add.  4) Receptionist clicks on Update Customer button. | 2) Displays the Customer Management UI.  5) Validate new Customer details.  6)Display update successfully message. |
| Alternate Path |  | If the Receptionist enters wrong credentials, System shows an error message. |
| Exception Path |  | Error displayed fields are empty. |
| Special Requirements | Only Receptionist can update Customer in the system. | |

Table .8: Use Case Description for Delete Customer

|  |  |  |
| --- | --- | --- |
| Use case ID | 08 | |
| Use case Name: | Delete Customer | |
| Actors: | Receptionist | |
| Pre- Conditions | Receptionist should be logged into the system. | |
| Post Condition | Successfully delete Customer from the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1) Receptionist clicks on Customer button.  3) Receptionist enter NIC of Customer who is need to be delete.  4) Receptionist clicks on Delete Customer button. | 2) Displays the Customer Management UI.  5)Display delete successfully message. |
| Alternate Path |  |  |
| Exception Path |  |  |
| Special Requirements | Only Receptionist can delete Customer in the system. | |

Table .9: Use Case Description for View Customer

|  |  |  |
| --- | --- | --- |
| Use case ID | 09 | |
| Use case Name: | View Customer | |
| Actors: | Receptionist | |
| Pre- Conditions | Receptionist should be logged into the system. | |
| Post Condition | Successfully view Customer details from the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1) Receptionist clicks on Customer button.  3) Receptionist enter NIC of Customer who is need to be view. | 2) Displays the Customer Management UI.  4)Display Customer details successfully. |
| Alternate Path |  |  |
| Exception Path |  |  |
| Special Requirements | Both owner and receptionist can view customer details in the system. | |

**Booking Management**

Table .10: Use Case Description for Add Room

|  |  |  |
| --- | --- | --- |
| Use case ID | 10 | |
| Use case Name: | Add Room | |
| Actors: | Receptionist | |
| Pre- Conditions | Receptionist should be logged into the system. | |
| Post Condition | Successfully add new Room to the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1) Receptionist clicks on Rooms button.  3) Receptionist enter the Room details to add.  4) Receptionist clicks on Add Room button. | 2) Displays the Rooms Management UI  5) Validate Room details.  6)Display Room added successfully message. |
| Alternate Path |  | If the Receptionist enters wrong credentials, System shows an error message. |
| Exception Path |  | Error displayed fields are empty. |
| Special Requirements | Both owner and Receptionist can add new Rooms to the system. | |

Table .11: Use Case Description for Update Room

|  |  |  |
| --- | --- | --- |
| Use case ID | 11 | |
| Use case Name: | Update Room | |
| Actors: | Receptionist | |
| Pre- Conditions | Receptionist should be logged into the system. | |
| Post Condition | Successfully update Customer to the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1) Receptionist clicks on Rooms button.  3) Receptionist enter the new Room details to add.  4) Receptionist clicks on Update Room button. | 2) Displays the Rooms Management UI.  5) Validate new Room details.  6)Display update successfully message. |
| Alternate Path |  | If the Receptionist enters wrong credentials, System shows an error message. |
| Exception Path |  | Error displayed fields are empty. |
| Special Requirements | Both owner and Receptionist can add new Rooms to the system. | |

Table .12: Use Case Description for View Room

|  |  |  |
| --- | --- | --- |
| Use case ID | 12 | |
| Use case Name: | View Room | |
| Actors: | Receptionist | |
| Pre- Conditions | Receptionist should be logged into the system. | |
| Post Condition | Successfully view Customer details from the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1) Receptionist clicks on Room button.  3) Receptionist enter Room Number of Room which is need to be view. | 2) Displays the Room Management UI.  4)Display Room details successfully. |
| Alternate Path |  |  |
| Exception Path |  |  |
| Special Requirements | Both owner and receptionist can view room details in the system. | |

**Generate Reports**

Table .13: Use Case Description for Generate Report

|  |  |  |
| --- | --- | --- |
| Use case ID | 13 | |
| Use case Name: | Generate Report | |
| Actors: | Owner | |
| Pre- Conditions | Owner should be logged into the system. | |
| Post Condition | Successfully generate reports from the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1)Clicks on report generate button.  3)User choose the criteria. | 2)Show the window with selectable criteria with the report is based on.  4)Show report according to the select criteria. |
| Alternate Path |  |  |
| Exception Path |  |  |
| Special Requirements | Only owner can generate reports in the system. | |

**Check in and Check out**

*Table 3.1.12: Use Case Description for Check in*

|  |  |  |
| --- | --- | --- |
| Use case ID | 12 | |
| Use case Name: | Check in | |
| Actors: | Receptionist | |
| Pre- Conditions | Receptionist should be logged into the system. | |
| Post Condition | Successfully update Customer to the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist selects a customer ID from the dropdown list. 2. The receptionist enters or verifies the room number and customer name. 3. The receptionist updates the check-in time (if needed). 4. The receptionist enters the amount paid. 5. The receptionist clicks "Update". | 1. The system fetches the customer's details. 2. The system validates the input. 3. The system records the check-in time. 4. The system calculates the pending amount. 5. The system saves all changes and confirms the update. |
| Alternate Path |  | If the customer is checking in for the first time, the system may require additional verification. |
| Exception Path |  | 1. If an invalid customer ID is selected, the system should display an error. 2. If a required field (e.g., room number) is left empty, the system prompts the user to complete it. 3. If the amount paid exceeds the total due, the system warns about an overpayment. |
| Special  Requirements | 1. The system should have a user-friendly interface for hotel staff. 2. Data should be stored securely. 3. The system should support multiple users accessing it simultaneously. | |

*Table 3.1.13: Use Case Description for Check out*

|  |  |  |
| --- | --- | --- |
| Use case ID | 13 | |
| Use case Name: | Check out | |
| Actors: | Receptionist | |
| Pre- Conditions | Guest must be checked into the hotel. | |
| Post Condition | Guest is checked out successfully. | |
|  | | |
|  | Action | System Response |
| Success Path | 1. Guest provides details for checkout. 2. Receptionist confirms pending amount. 3. Guest clears dues (if any). 4. Receptionist finalizes checkout. 5. System generates invoice. | 1. System fetches guest details. 2. System displays pending amount. 3. System updates payment status. 4. System marks guest as checked out. 5. Invoice is displayed and saved. |
| Alternate Path | Guest has already paid the amount. | System automatically marks payment as complete. |
| Exception Path | Guest has pending dues and cannot pay. | System prevents checkout until dues are cleared. |
| Special  Requirements | 1. System should provide a printable invoice. 2. Checkout must update room availability. 3. System should log checkout details for record-keeping. | |

*Table 3.1.14: Use Case Description for Delete Reservation*

|  |  |  |
| --- | --- | --- |
| Use case ID | 14 | |
| Use case Name: | Delete Reservation | |
| Actors: | Receptionist | |
| Pre- Conditions | 1. A reservation must exist in the system. 2. The receptionist must have the necessary permissions to delete a reservation. 3. The customer ID and reservation details must be available. | |
| Post Condition | 1. The reservation is removed from the system. 2. The room is marked as available. 3. Any associated payments or refunds are processed (if applicable). | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist selects a customer ID from the dropdown list. 2. The receptionist verifies the reservation details. 3. The receptionist clicks the "Delete" button. | 1. The system retrieves the reservation details. 2. The system displays the relevant information. 3. The system prompts for confirmation. 4. The system removes the reservation, |

|  |  |  |
| --- | --- | --- |
|  | 4. The receptionist confirms the deletion. | updates room availability, and processes any pending payments or refunds. |
| Alternate Path |  | 1. If the customer requests to modify instead of deleting the reservation, the receptionist can navigate to the update screen instead. 2. If the customer decides to reschedule, the system redirects them to the reservation modification page. |
| Exception Path |  | 1. If the selected reservation does not exist, the system displays an error message. 2. If the reservation has associated payments, the system prompts the receptionist to process a refund (if applicable). 3. If there is a system failure, the reservation remains unchanged, |
|  |  | and the receptionist is notified of the error. |
| Special  Requirements | 1. The system should log all deletion activities for security and audit purposes. 2. Only authorized staff should be able to delete reservations. 3. The system should provide an option to restore accidentally deleted reservations within a specified period. | |

**Inventory Management**

*Table 3.1.15: Use Case Description for Add item*

|  |  |  |
| --- | --- | --- |
| Use case ID | 15 | |
| Use case Name: | Add item | |
| Actors: | Receptionist | |
| Pre- Conditions | The receptionist must be logged into the system. | |
| Post Condition | Employee records are successfully updated (added, edited, removed, or displayed). | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist enters item details in the provided fields. | 1. The system captures the input data. 2. he system processes the request. |

|  |  |  |
| --- | --- | --- |
|  | 1. The receptionist selects the desired action (Add, Edit,   Remove, Show  Details).   1. The system validates the entered details. 2. The system updates the item records accordingly. 3. A confirmation message is displayed to the receptionist. | 1. If valid, proceeds to the next step; otherwise, an error message is displayed. 2. The system stores the new or modified information. 3. The system confirms the operation success. |
| Alternate Path | The receptionist selects  "Show Details." | The system retrieves and displays the item’s information. |
| Exception Path | 1. Required fields are left empty. 2. Invalid NIC or phone number is entered. 3. The system is offline. | 1. The system prompts   the receptionist to fill in missing fields.   1. The system displays an error message indicating invalid input. 2. The system shows an error message indicating connectivity issues. |
| Special  Requirements | 1. The system should have a user-friendly interface. 2. Item passwords must be securely stored. | |
|  | 1. Data validation should be enforced for correct input formats. | |

*Table 3.1.16: Use Case Description for Update item*

|  |  |  |
| --- | --- | --- |
| Use case ID | 16 | |
| Use case Name: | Update item | |
| Actors: | Receptionist | |
| Pre- Conditions | The receptionist must be logged into the system. | |
| Post Condition | The employee details are successfully updated in the database. | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist selects an existing item from the system. 2. The receptionist edits the necessary fields (e.g., address, job title, phone number). 3. The receptionist clicks the "Edit" button. 4. If validation is successful, the system updates the database. | 1. The system retrieves and displays the current details. 2. The system captures the updated input data. 3. The system validates the modified details. 4. The system confirms the update. |
| Alternate Path | A success message is displayed to the receptionist. | The system provides confirmation of the changes. |
| Exception Path | 1. The receptionist does not select an existing item. 2. Invalid data (e.g., incorrect NIC, phone number, missing fields) is entered. 3. The system is offline. | 1. The system prompts   the receptionist to choose an item before editing.   1. The system displays an error message and prevents updating. 2. The system displays a message indicating connection issues and prevents the update. |
| Special  Requirements | 1. The system should allow easy navigation between item records. 2. A log of updates should be maintained for tracking changes. 3. The system should validate input fields before allowing updates. | |

*Table 3.1.17: Use Case Description for Delete item*

|  |  |  |
| --- | --- | --- |
| Use case ID | 17 | |
| Use case Name: | Delete item | |
| Actors: | Receptionist | |
| Pre- Conditions | The receptionist must be logged into the system. | |
| Post Condition | The selected item’s record is permanently removed from the database. | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist selects an item from the system. 2. The receptionist clicks the "Remove" button. 3. The receptionist confirms the deletion. 4. The system updates the database and removes the item entry. | 1. The system retrieves and displays the current details. 2. The system asks for confirmation before deletion. 3. The system validates the request and removes the item’s record. 4. The system displays a success message confirming deletion. |
| Alternate Path |  |  |
| Exception Path | 1. The receptionist does not select an item. | 1) The system displays an error message asking for a selection. |
|  | 1. The receptionist cancels the delete request. 2. The system is offline. | 1. The system does not remove the item record and returns to the previous screen. 2. The system displays an error message indicating connection issues. |
| Special  Requirements | 1. The system should ask for confirmation before deleting a record. 2. Deleted records cannot be recovered unless an "undo" function is implemented. 3. The system should log deletions for security and audit purposes. | |

*Table 3.1.18: Use Case Description for Search item*

|  |  |  |
| --- | --- | --- |
| Use case ID | 18 | |
| Use case Name: | Search item | |
| Actors: | Receptionist | |
| Pre- Conditions | The receptionist must be logged into the system. | |
| Post Condition | The system displays the searched item details if available. | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist enters search criteria (e.g., name, NIC, job title, phone number). 2. The system displays a list of matching employees. 3. The receptionist clicks on an item record. | 1. The system retrieves item records that match the criteria. 2. The receptionist can select an item to view full details. 3. The system displays the selected item’s full details. |
| Alternate Path |  |  |
| Exception Path | 1. No matching item is found. 2. The search query is invalid or empty. 3. The system is offline. | 1. The system displays a message: "No matching records found." 2. The system prompts   the receptionist to enter valid search criteria.   1. The system displays an error message indicating connection issues. |
| Special  Requirements | 1. The system should support multiple search criteria (e.g., name, NIC, job title, phone number). 2. The search should be optimized for speed and accuracy. 3. The system should allow filtering and sorting of results. | |

**Event & Conference Management**

*Table 3.1.19: Use Case Description for Add event*

|  |  |  |
| --- | --- | --- |
| Use case ID | 19 | |
| Use case Name: | Add event | |
| Actors: | Receptionist | |
| Pre- Conditions | The receptionist must be logged into the system. | |
| Post Condition | A new event is successfully added to the event list. | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist enters event details (name, description, time, duration, date). 2. The receptionist clicks on the "Add" button. 3. The input data is valid. 4. The system confirms the addition. | 1. The system opens the event creation form. 2. The system allows input and verifies the format. 3. The system validates the input data. 4. The system saves the event in the database. 5. A confirmation message is displayed: "Event added successfully." |
| Alternate Path | 1. The receptionist clicks "Add" but does not enter any details. | The system does not save the event and remains on the form. |
| Exception Path | 1. The receptionist enters incomplete or invalid data (e.g., missing event name, incorrect date format). 2. The receptionist corrects the data and retries. | 1. The system displays an error message highlighting the incorrect fields. 2. The system revalidates the input and proceeds with adding the event if valid. |
| Special Requirements | 1. The system should prevent duplicate event names for the same date. 2. The system should allow optional fields like "Description" while requiring essential fields like "Event Name" and "Date." | |

*Table 3.1.20: Use Case Description for Update event*

|  |  |  |
| --- | --- | --- |
| Use case ID | 20 | |
| Use case Name: | Update event | |
| Actors: | Receptionist | |
| Pre- Conditions | The receptionist must be logged into the system. | |
| Post Condition | The event details are successfully updated in the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist selects an event from the list. 2. The receptionist modifies event details in the event table (name, description, time, duration, date). 3. The receptionist clicks the "Update" button. 4. The input data is valid. 5. The system confirms the update. | 1. The system retrieves and displays the event details. 2. The system allows changes in the input fields. 3. The system validates the input data. 4. The system updates the event in the database. 5. A confirmation message is displayed: "Event updated successfully." |
| Alternate Path | 1. The receptionist selects an event but does not modify any details. 2. The receptionist clicks the "Update" button. | 1. The system detects no changes. 2. The system does not make any changes to table. |
| Exception Path | 1. The receptionist enters invalid data (e.g., missing event name, incorrect time format). 2. The receptionist corrects the data and retries. | 1. The system displays an error message. 2. The system revalidates the input and proceeds with the update if valid. |
| Special Requirements | 1. The system should allow undoing changes before confirming the update. 2. The system should log all update activities for auditing purposes. | |

*Table 3.1.21: Use Case Description for View Event*

|  |  |  |
| --- | --- | --- |
| Use case ID | 21 | |
| Use case Name: | View Event | |
| Actors: | Receptionist | |
| Pre- Conditions | The receptionist must be logged into the system. | |
| Post Condition | The system displays the event details. | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist opens the event management system. 2. The receptionist selects or navigates to the event list. 3. The system displays the event list in a table format. | 1. The system loads the event management interface. 2. The system retrieves all stored events from the database. 3. The table shows event details such as Name, Description, Time, Duration, and Date. |
| Alternate Path | The event list is empty (no events added). | The system displays empty table." |
| Exception Path | The system fails to retrieve event data due to a database error. | The system displays an error message: "Error retrieving events. Please try again later." |
| Special Requirements | 1. The system should allow sorting and filtering of events by date, name, or duration. 2. The system should refresh the event list automatically when new events are added. | |

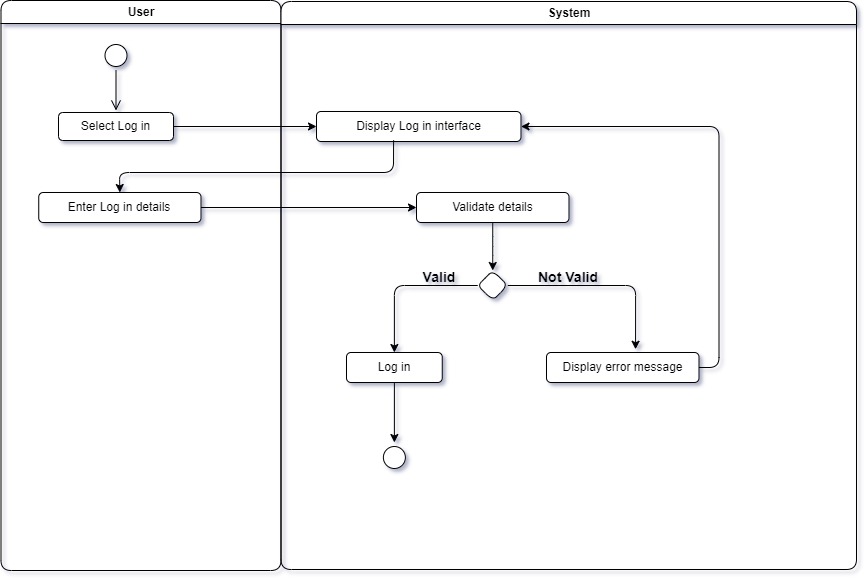
*Table 3.1.22: Use Case Description for Delete event*

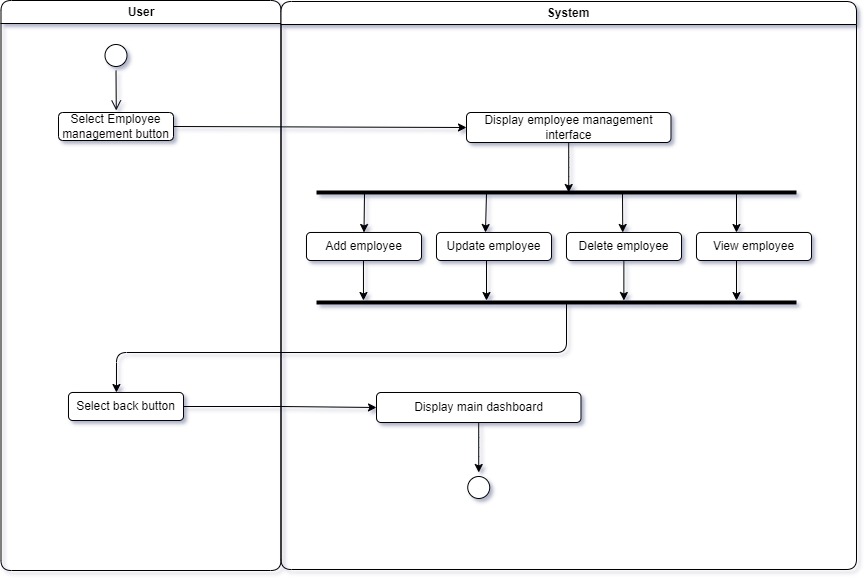
|  |  |  |
| --- | --- | --- |
| Use case ID | 22 | |
| Use case Name: | Delete event | |
| Actors: | Receptionist | |
| Pre- Conditions | The receptionist must be logged into the system. | |
| Post Condition | The selected event is removed from the system. | |
|  | | |
|  | Action | System Response |
| Success Path | 1. The receptionist opens the event management system. 2. The receptionist selects an event from the event list. 3. The receptionist clicks the "Delete" button. 4. The receptionist confirms the deletion. 5. The system updates the event list. | 1. The system loads the event management interface. 2. The system highlights the selected event. 3. The system prompts a confirmation message: "Are you sure you want to delete this event?" 4. The system removes the event from the database. 5. The table refreshes and no longer displays the deleted event. A success message appears: "Event deleted successfully." |
| Alternate Path | The receptionist selects an event but cancels the deletion when prompted. | The system does not delete the event and returns to the event list. |
| Exception Path | 1. The receptionist tries to delete an event, but the system encounters a database error. 2. The receptionist tries to delete an event but did not select any row to delete. | 1. The system displays an error message: "Error deleting event. Please try again later." 2. The system notifies: "No row is selected! Please select one row." |
| Special Requirements | 1. The system should ask for confirmation before deleting an event. 2. The system should allow only authorized users (receptionists) to delete events. 3. The system should log all deletion activities for auditing purposes. | |

# **NON-FUNCTIONAL REQUIREMENTS**

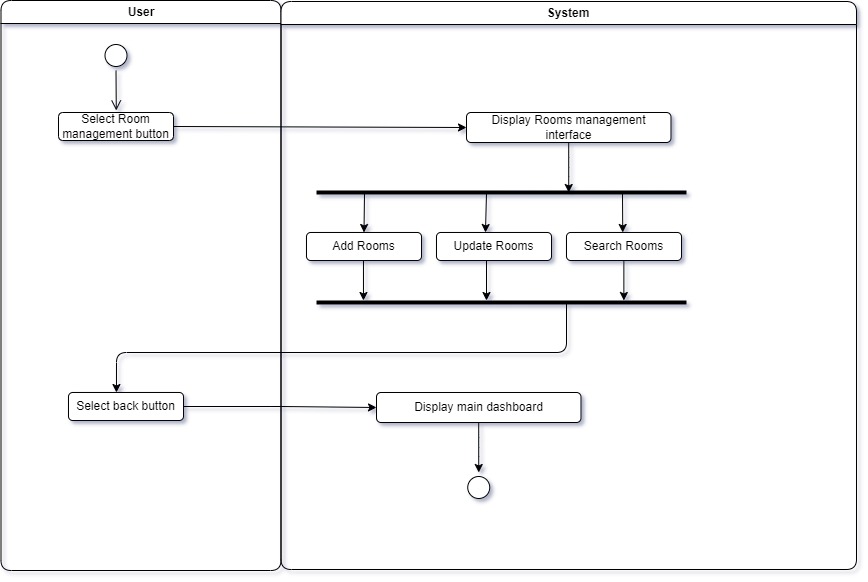
## Performance Requirements

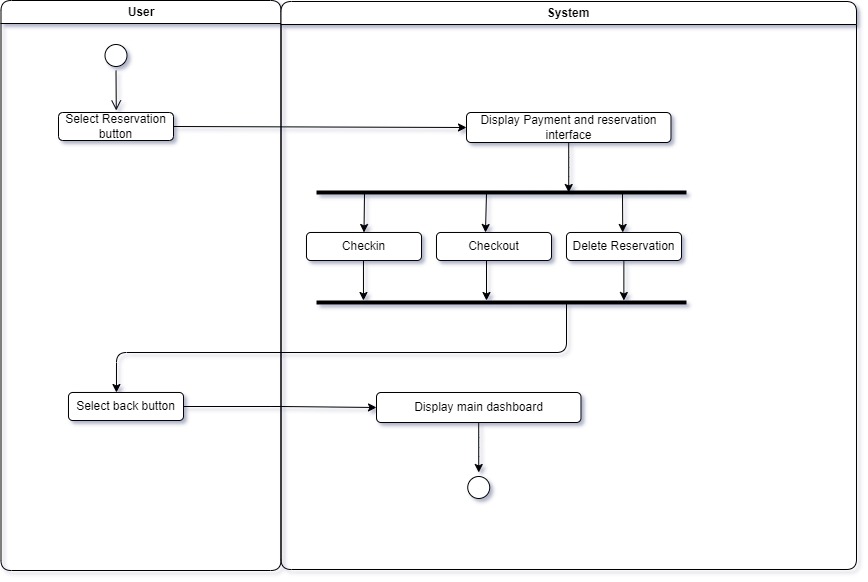
You are exposed to concurrent access by the receptionist and manager, and the system needs to manage that without getting into performance problems. The response time for key operations like room booking, payment amount entry, and inventory updates should be less than 4 seconds to maintain the flow of work. Moreover, the database must be able to store a minimum of 10,000 records (customer bookings, recorded payments and inventory logs), without compromising on system performance. For a reliable and minimal disruption experience, the system must achieve 99% uptime to ensure that daily operations occur seamlessly.

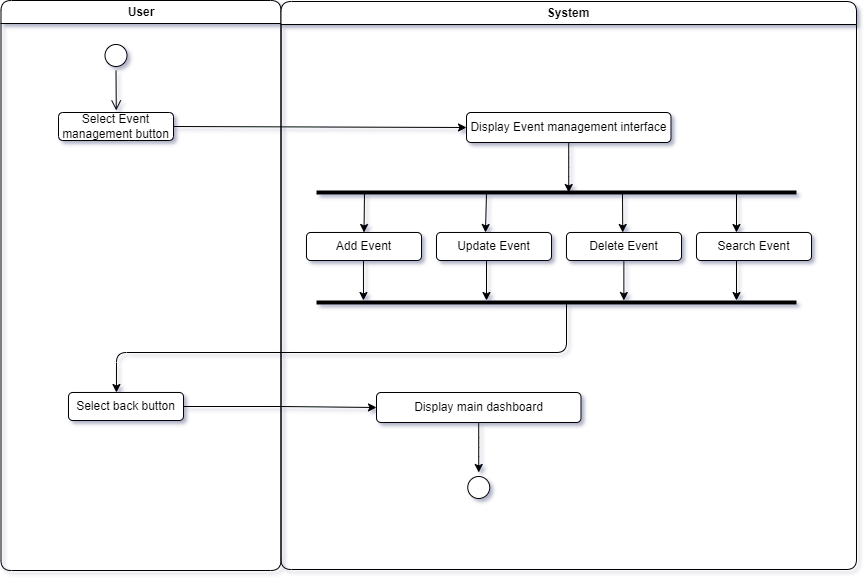
**User Login**

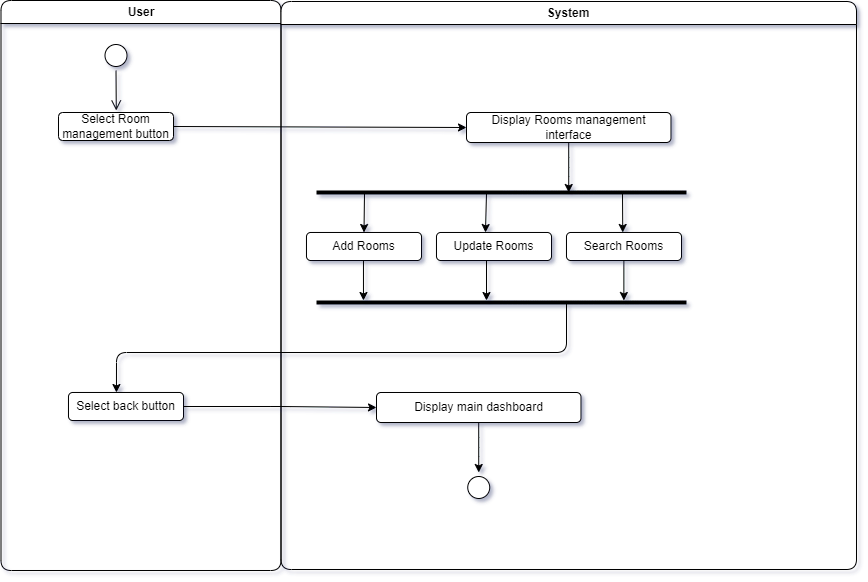
**Employee Management**

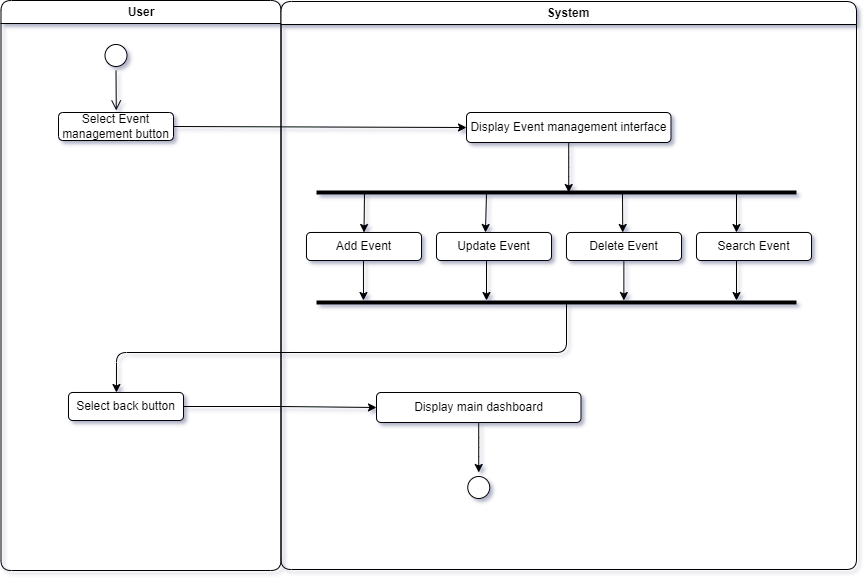
**Room Management**

****

**Payment & Reservation**

**Event Management**

**Customer Management**

**Inventory Management**

## Safety Requirements

Regular automated backups have to take place on the database so that no data is lost and there is a swift recovery possible after system failure, growing the trust in the systems reliability. In event of system failure, the problem must be quickly diagnosed and fixed to minimize down time. The software relies on the hardware infrastructure, which means a hardware breaking down can cause system crashes. So, Hardware health needs to be monitored continuously to avoid any downtime and keep the operations running smoothly.

## Security Requirements

To guarantee security, the system uses stringent authentication, data integrity, and access control procedures. A distinct username is given to each user (manager and receptionist) for identification and authentication purposes. When creating an account, the user is given a predetermined password that they can modify at a later time. To avoid unwanted access, every password is safely encrypted. In addition, incorrect password attempts will be recorded, and for security reasons, the account will be temporarily disabled after several unsuccessful attempts. Sensitive inventory data and financial records can only be viewed and altered by the manager.

## Software Quality Attributes

#### **Usability**

The system is designed for **two internal users**:

1. **Receptionist** - Manages room bookings, customer records, and enters physical payment amounts.
2. **Manager** - Oversees financial records, inventory, employee schedules, and system settings. The application must have a **user friendly UI**, ensuring easy navigation and minimal learning curve for hotel staff.

#### **Testability**

To ensure the system functions correctly, the following **testing methodologies** will be applied:

* **Unit Testing** - Individual components are tested to verify their functionality.
* **Integration Testing** - Different system modules are combined and tested as a whole to ensure smooth interactions.
* **System Testing** - The system is tested against defined performance and functional requirements.
* **Acceptance Testing** - Final validation is conducted to ensure that the system meets the needs of the hotel staff and is ready for deployment.

## Business Rules

* Only the manager has the authority to add or remove system users.
* Only the manager can access financial reports and monitor overall system performance.
* The receptionist is responsible for booking management, check in, and check out processes.
* The receptionist can enter only the physical payment amounts received from customers, no online transactions or automated payment processing occurs.
* All data modifications and transactions must be logged to maintain accountability.

# References

|  |  |
| --- | --- |
| [1] | Lutkevich, B. (2021). *What is project scope?* [online] TechTarget. Available at: https://www.techtarget.com/searchcio/definition/project-scope.  [Accessed 28 February 2025]. |

|  |  |
| --- | --- |
| [2] | Rosencrance, L. (2019). *Software Requirements Specification (SRS)*. [online] SearchSoftwareQuality. Available at: https://www.techtarget.com/searchsoftwarequality/definition/software-requirements-specification.  [Accessed 28 February 2025]. |

# OTHER REQUIREMENTS

Appendix D: Personal Contribution

Name: A.A. Rifath

SA number: SA23089754

Group name & ID: Redliners G08

Role: Project Manager

Main contributions:

* Divided the tasks for the project among group members.
* Database design for the system.
* Preparing presentations, reports and logbooks.
* Developing the Inventory management function and integrating the final system in progress.
* Wrote the Product Perspective, Product Functions, User Classes and Characteristics, Operating Environment and Design and Implementation Constraints parts of the SRS report.
* Created the Architecture Diagram, DFD and user case diagram for the report.
* Coordinating and overseeing the development of the team.

Name: H.G.S. Sanchitha

SA number: SA23428324

Group name & ID: Redliners G08

Role: Client Coordinator

Main contributions:

* Analysis Implementation, testing and created activity diagrams for SRS report.
* Wrote the Non-Functional Requirements section of the SRS report.
* Assist preparing presentations, reports.
* Finding a client for this project.
* Requirement gathering and coordinating with the client.
* Implementation of the reservations and payment management function of the system in progress.
* Assist in debugging, testing and final integration of the system.

Name: M.M.N.S. Bandara

SA number: SA23442016

Group name & ID: Redliners G08

Role: Technical Writer

Main contributions:

* Wrote the System Functional Features part of the SRS report.
* Created user case description tables for the SRS report.
* Assist in creating presentations, reports and docs as Technical Writer.
* Assist in finding a client for the project.
* Prepared project schedule.
* Implementation of the customer management function of the system in progress.

Name: E.T. Rusiru

SA number: SA23438538

Group name & ID: Redliners G08

Role: Quality Assurance

Main contributions:

* Covered the Product Introduction, Intended Audience and Reading Suggestions section, Product Purpose, Goals and Benefits of the SRS report.
* Assist preparing presentations, reports.
* Assist in requirement gathering.
* Implementation of the System's Event & Conference Management function in progress.
* Quality Assurance role with testing the system and find bugs.

Name: H.K. Dilanjan

SA number: SA23439450

Group name & ID: Redliners G08

Role: UI/UX Designer

Main contributions:

* Wrote the System Functional Features, Database Requirements parts of the SRS report.
* Created the user case description tables, ER diagram, user case diagram for the SRS report.
* Creating all diagrams for the presentation and assist in preparing presentations and docs.
* Managing UI/UX development of the system with the project manager.
* Assist in database design.
* Implementation of the employee management and job scheduling function of the system in progress.

Name: P.M.C.R. Bandara

SA number: SA23514522

Group name & ID: Redliners G08

Role: Business Analyst

Main contributions:

* Wrote the User Documentation, Assumptions and Dependencies, User Interfaces, Hardware Interfaces, Software Interface, Communications Interfaces parts of the SRS report.
* Creating and overseeing the budget of this project.
* Communicating with the client as the Business Analyst.
* Assist in preparing presentations and documentation.
* Implementation of the Booking management function of the system in progress.