***Software Requirement and Design***

***Specifications***

***HospitalityHub - Hotel Management System***



***Version: 1.1***

|  |  |
| --- | --- |
| *Course Code* | *CS-3004* |
| *Instructor* | *Mrs Nida Munawwar* |
| *Project Team* | *YUSRA SOHAIL SIDDIQUI 21K -3446*  *SYED ABDUL REHMAN 21K -3156*  *MUHAMMAD XABAD NADEEM 21K -4612* |
| *Submission Date* | *01st December 2023* |

**Table of Contents …………………………………..………………………………………………ii  
Revision History ………………………………………...…………………………………………ii  
1. Introduction………………………………………………….……………………………………**1.1 Purpose ………………………………………….………………………………………………1  
1.2 Intended Audience .……………………………….…..…………………………………………1  
1.3 Document Conventions…………………………….…………...……………………………… 1  
1.4 Intended Audience and Reading Suggestions…….……………….…………………………… 1  
**2. Overall System Description………….………………………….….……….…………………..1**2.1 Project Background...…..….……………………...………………..…...……………………… 2  
2.2 Project Scope……..……….………………………………………….………………………… 2  
2.3 Not In Scope …………….……..……………………..…………………………………………2  
2.4 Project Objectives………..….……………………………………………………….………… 2  
2.5 Stakeholders …………….……..………………………………………………..………………2  
2.6 Operating Environment………………………………..…………..…………………………… 2  
2.7 System Constraints………………………………………………..…………………………… 3  
2.8 Assumptions & Dependencies…………………………………….…………………………… 4  
2.9 Design and implementation constraints …………………………...……………………………4  
2.10 User Documentation ……………………………………………..……………………………5  
**3. External Interface Requirements ……………………………...………………………………4**3.1 User Interfaces ……………………………………………………….…………………………5  
3.2 Hardware Interfaces………………………………………………….………………………… 4  
3.3 Software Interfaces…………………………………………………...………………………… 4  
3.4 Communications Interfaces ………………………………………………..……………………4  
**4. Functional Hierarchy………………………………………...………………………………… 5  
5. Use cases …………………………………………………………………………………………9  
6. Other Nonfunctional Requirements…………………………………….…………………… 11**6.1 Performance Requirements ……………………………………………………………………11  
6.2 Safety Requirements……………………………………………………..…………………… 12  
6.3 Security Requirements……………………………………………………...………………… 12  
6.4 Software Quality Attributes ………………………………………………...…………………12  
6.5 Business Rules ……………………………………………………………………...…………12  
**7. System Architecture …………………………………………………………………..………20**

7.1 System Level Architecture….……………………………………………………….…………21  
**8. Design Strategy ………………………………………………..………………………………21  
9. Detailed System Design ……………………………………….………………………………22**9.1 Database Design ………………………………………………………………………………23  
9.1.1 ER Diagram …………………………………………………………………………………24  
9.2 Application Design ……………………………………………………………………………42  
**10. References …………………………………………………….………………………………52  
11. APPENDIX…………………………………………………...……………………………… 52**

# Introduction

## Purpose

The Hotel Management System (HMS) is a web based application meticulously crafted to simplify and enhance the hotel management process. Tailored for both guests and administrators, the system's core purpose is to provide an **intuitive** and **user friendly** platform. Guests benefit from a seamless reservation experience, enabling them to effortlessly book rooms On the administrative side, HMS empowers hotel staff to **efficiently manage bookings, monitor room availability**, and **engage with user data**. Through its intuitive design and robust functionalities, the HMS seeks to **optimize** the hotel experience, ensuring a smooth and enjoyable stay for all parties involved.

**1.2 Intended audience**

This document is designed for:

- Fellow Students: Potential users and promoters of HOSPITALITYHUB, facilitating their understanding of the platform's features for effective marketing on social media.

- Course Instructors:Serving as a managerial guide, offering insights into HOSPITALITYHUB's development for informed guidance and alignment with educational objectives.

## 1.3 Document Conventions

While writing the **SRS** for  **HOSPITALITYHUB**, the headings for all section andsub sections are written in font ‘**Times New Roman**’ having font size **14**  and the descriptions are written in font ‘**Times New Roman**’ having font size **11**. Apart from this, we have **bolded** out those sentences or words that must beemphasized to be understandable by reader to understand the **complete functionalities** and **features** of our HOSPITALITYHUB website. Also, to make readers **lock** those words and sentences in their minds to get **familiar** with them throughout the documents. Moreover, the reference links are **underlined** having **blue color** to represent it as a **web link**. Furthermore, the important terminologies are written within **single quotes** to make reader notified about **specific terms and conditions** for our website. Lastly, the **tools and technologies** required to make this website are written **capitalized**.

# Overall Description

## Product Background

HOSPITALITYHUB is a revolutionary self-contained website designed to enhance existing hotel management systems. This all-in-one platform focuses on providing comprehensive features and functionalities related to hotel room bookings. The project aims to streamline and simplify the hotel reservation process for both users and administrators.

## Product Scope

The scope of HOSPITALITYHUB encompasses the entire hotel room booking process. Users can register, search for rooms based on preferences, book rooms for any number of guests, request amenities, and manage their bookings. Administrators have the capability to update, insert, and delete bookings and amenity inventory, oversee all bookings made through the platform, configure email notifications, set up promotions, manage seasonal rates, and fine-tune website features and settings for a customized user experience.

## Not InScope

Certain functionalities such as a payment feature fall outside the scope of HOSPITALITYHUB. These exclusions are explicitly defined to manage expectations. Aspects such as external software dependencies, unrelated services, or functionalities not directly related to hotel room bookings are not within the project's scope.

## 2.4 Project Objectives

The primary objectives of HOSPITALITYHUB are to enhance the hotel room booking experience for users and streamline management for administrators. This includes providing a user-friendly interface, efficient booking processes, and robust administration tools to handle reservations, amenities, and configurations effectively.

## 2.5 StakeHolders

The stakeholders involved in the HOSPITALITYHUB project include users, administrators, and developers. Users will interact with the platform to book rooms and manage reservations, administrators will oversee and manage the system, and developers will be responsible for ongoing maintenance and improvements.

**2.6. Operating Environment**

HOSPITALITYHUB is designed to be platform-independent, ensuring accessibility across various operating systems, including Windows. The platform's operating environment allows users to access its features seamlessly, regardless of their preferred operating system.

## 2.7. System Constraints

## In the context of developing a basic hotel booking site, specific constraints guide the project within the scope of simplicity and focused functionality:

## 1. Software Constraints

## The site's software must be streamlined for basic hotel booking functionalities, emphasizing simplicity in user interactions. It should prioritize essential features like room selection, date booking, and user authentication, avoiding unnecessary complexities.

## 2. Hardware Constraints

## Given the basic nature of the site, hardware considerations primarily involve ensuring compatibility with standard web browsers and devices commonly used for online activities. The system should perform optimally on common devices with typical processing power and memory capacities.

## 3. Cultural Constraints

## Cultural constraints are minimal but may involve providing language options for users from diverse backgrounds, ensuring a basic level of inclusivity without complicating the user interface.

## 4. Legal Constraints

## The site must comply with fundamental legal requirements for online transactions and user data protection. While not extensively complex, it should adhere to basic regulations to ensure the security and privacy of user information.

## 5. Environmental Constraints

## In the digital realm, environmental constraints are less pronounced. However, the site should be designed to perform reliably under standard internet conditions, considering users with basic internet connectivity.

## 6. User Constraints

## User constraints revolve around simplicity. The site should cater to users with varying levels of technical proficiency, offering an intuitive interface for straightforward hotel booking processes. Graphic controls may be prioritized over text-heavy interfaces.

## 7. Off-the-Shelf Components Constraints

## If the site incorporates off-the-shelf components, they should align with the site's basic nature, avoiding unnecessary features that could complicate the booking process. Licensing agreements and integration should be straightforward to maintain the site's simplicity.

## 2.8 Assumptions and Dependencies

It is assumed that the user is familiar with an **internet browser** and familiar with handling the **keyboard and mouse**. Since the application is a **web based application** there is a need for the internet browser. It will be assumed that the users will possess decent internet connectivity. While there is **no dependency** of any software **nor does the HOSPITALITYHUB reuse** parts of any other already built project.

## 2.9 Design and Implementation Constraints

The HOSPITALITYHUB website developed until now has a complex code and **full dependency** between different functionalities. For example, an account needs to be created for a user to actually book the room. So, these **restrictions** are complex to handle and modify. Such features are required to **carefully be** looked over by the ‘developers. Also, it is required to be executed with **good internet speed** as there are numerous pictures used on the website that may cause extended retrieval time. Moreover, **large memory** is required to store the database as if we are having many features . Apart from this, only the **citizens within Pakistan** are allowed to utilize HOSPITALITYHUB features as if the features are bounded in such a way. The tools and technologies required to implement this website are ‘**HTML’, ‘CSS’** for front end, ‘**JAVASCRIPT’, ‘PHP’** for back end. ‘**MYSQL’** for database (stored in ‘**PHPMYADMIN’** along with ‘**APACHE’**). **Communication protocol** includes the database MySQL and backend code to make users interact through website with data brought from database to front end through back end. The **security protocols** includes that neither the ‘user’ nor ‘admin’ aere allowed to access any page of website without logging/ signing in except for home page and about us page.

## 2.10 User Documentation

Along with the software, the proper ‘**user manual’** will be delivered to make users understand the proper **functionalities and features** of HOSPITALITYHUB. The user manual will contain ‘**several links’** related to the **similar platforms** so that if user is familiar with those platforms can **easily interact** with HOSPITALITYHUB. Apart from this, the links for ‘**tutorials’** to present a demo of **using** this website will also be added.

# External Interface Requirements

## User Interfaces

The Graphical User Interface provides a **pictorial colorful insight** on every page rendered. Every page contains **font family style** (Rancho Regular,OpenSans Regular). The **grid property** is also applied for aligning the textual content with the graphical content. Hovering effects using **transitions** and **animations** are also added to user interface. The user/admin **fills forms** respective to their respective operations. They can **input data** with the help of keyboard or click with the mouse wherever necessary. The HOSPITALITYHUB system also provides **radio buttons**, **checkboxes** from which the users can select and **links** and **icons** to navigate among the webs pages. **Pushbutton** is also used to save and retrieve content from the database according to the preference selected (Book, Cancel, Perform CURD, etc.).

## Hardware Interfaces

Since the application is web based, it will use **best practice web generation tools** and **tested technologies** such as (Google, Firefox, etc.). **Xammp** local host server for **MySQL** with latest version to provide host address. It is applicable all for **windows browsers**. It will also be supported on all different sized devices because it’s made **responsive** regardless of screen dimensions.

## Software Interfaces

The HOSPITALITYHUB website was developed on **VISUAL CODE STUDIO** version **1.73.1**. The tools and technologies that were required to implement this website is ‘**HTML’, ‘CSS’,’BOOTSTRAP’** for front end, ‘**PHP’**  for back end. ‘**MYSQL’** for database (stored in ‘**PHPMYADMIN’** along with ‘**APACHE’**). For now, it is tested on ‘**windows.’** The available data is integrated with the front end coming from the database as per the user/admins **demands** or **request**. Moreover, user/admins personal data is stored in **database** once they **signup**. Apart from this, the users booked rooms are constantly **updated** in database as soon as they **book** or **cancel**. This database must be accessible by admin only and capable enough to **display** the **requiredinformation** on website and **update** the **information** in database. Apart from this, the ‘**PHPMYADMIN’** along with ‘**APACHE’** is required to access to the **testing** of website and **accessing** database. Whereas the proper **internet connection** is required to utilize the **features and functionalities** of HOSPITALITYHUB. .

## Communications Interfaces

Communications Interfaces

• Different tabs in the applications are communicated internally to send the data between different projects

• Database communications are handled through specialized header to deal with them

# System Features

The **functionalrequirements** for our web application  **HOSPITALITYHUB** illustrates the system **features and services** provided to make sure that the ‘users’ can function as per their desires. The features provided by the ‘system’ are **userfriendly** to ensure that user’s requests are entertained in the best possible way. System features also ensures that any **incorrect information** provided by the user is **pointed** at right time to let the users provide precise and accurate information.

**Admin Features**

**4.1. About Management:**

- Description: Admin can view and update information about the business

- Priority: Medium

- Stimulus/Response: User selects the "About" section.

**- Functional Requirements:**

- View current business information.

- Update and edit business details.

**4. 2. Contact Management:**

- Description: Admin can view and update contact information.

- Priority: High

- Stimulus/Response: User selects the "Contact" section.

**- Functional Requirements**:

- View current contact details.

- Update and edit contact information.

**4. 3. Category Management:**

- Description: Admin can add, view, and manage categories.

- Priority: High

- Stimulus/Response: User selects the "Category" section.

**- Functional Requirements:**

- Add new categories.

- View existing categories.

- Edit and delete categories.

**4.4 Facility and Amenity Management:**

- Description: Admin can add and manage facilities and amenities.

- Priority: High

- Stimulus/Response: User selects the "Facility & Amenity" section.

**- Functional Requirements:**

- Add new facilities and amenities.

- View existing facilities and amenities.

- Edit and delete facilities and amenities.

**4.5 Room Management:**

- Description: Admin can add and manage rooms.

- Priority: High

- Stimulus/Response: User selects the "Room" section.

**- Functional Requirements:**

- Add new rooms.

- View existing rooms.

- Edit and delete rooms.

**4.6 Booking Management:**

- Description: Admin can approve, check, cancel, and manage bookings.

- Priority: Critical

- Stimulus/Response: User selects the "Booking" section.

**- Functional Requirements:**

- Approve booking requests.

- Check current bookings.

- Cancel existing bookings.

- Change booking details.

**4.7 Gallery Image Management:**

- Description: Admin can change gallery images.

- Priority: Medium

- Stimulus/Response: User selects the "Gallery" section.

**- Functional Requirements:**

- Upload and manage gallery images.

- Change existing images.

**4.8 Password Management:**

- Description: Admin can change their login password.

- Priority: High

- Stimulus/Response: User selects the "Change Password" section.

**- Functional Requirements:**

- Enter current password.

- Enter new password.

- Confirm new password.

**4.9 Room Editing:**

- Description: Admin can edit room details.

- Priority: Medium

- Stimulus/Response: User selects the "Edit Room" section.

**- Functional Requirements:**

- Select room to edit.

- Modify room details.

**4.10 View Enquiries:**

- Description: Admin can view customer inquiries.

- Priority: Medium

- Stimulus/Response: User selects the "Enquiries" section.

**- Functional Requirements:**

- View list of inquiries.

- Respond to inquiries.

**4.11 User Management:**

- Description: Admin can view registered users.

- Priority: Medium

- Stimulus/Response: User selects the "Registered Users" section.

**- Functional Requirements:**

- View list of registered users.

- Access user profiles.

**4.12 Booking Search:**

- Description: Admin can search for bookings by booking number.

- Priority: High

- Stimulus/Response: User initiates a search by booking number.

**- Functional Requirements:**

- Enter booking number.

- Retrieve booking details.

# Features for user interface :

**4. 13. User Profile:**

- Description: Users can view and edit their profile information.

- Priority: Medium

- Stimulus/Response: User selects the "Profile" section.

**- Functional Requirements:**

- View current profile information.

- Edit and update profile details.

**4.14 Category Details:**

- Description: Users can view details about available categories.

- Priority: Low

- Stimulus/Response: User selects the "Categories" section.

**- Functional Requirements:**

- View list of available categories.

- Access detailed information about each category.

**4.15 Room Booking:**

- Description: Users can book a room.

- Priority: High

- Stimulus/Response: User selects the "Book a Room" section.

**- Functional Requirements:**

- Select desired room and date.

- Confirm booking details.

- Receive booking confirmation.

**4.16 Change Password:**

- Description: Users can change their login password.

- Priority: Medium

- Stimulus/Response: User selects the "Change Password" section.

**- Functional Requirements:**

- Enter current password.

- Enter new password.

- Confirm new password.

**4.17 Contact the Company:**

- Description: Users can send messages or inquiries to the company.

- Priority: Medium

- Stimulus/Response: User selects the "Contact" section.

- Functional Requirements:

- Enter message details.

- Submit the message.

**4.18 View Gallery:**

- Description: Users can view images of rooms and facilities.

- Priority: Low

- Stimulus/Response: User selects the "Gallery" section.

**- Functional Requirements:**

- Access a gallery of images.

- Navigate through images.

**4.19. View Amenities:**

- Description: Users can view details about available amenities.

- Priority: Low

- Stimulus/Response: User selects the "Amenities" section.

**- Functional Requirements:**

- View list of available amenities.

**4.20. Reset Account Details:**

- Description: Users can request a reset of their account details.

- Priority: Medium

- Stimulus/Response: User selects the "Reset Account" section.

**- Functional Requirements:**

- Request account details reset.

- Follow instructions for reset.

**4.21. View Invoice:**

- Description: Users can view and download invoices for their bookings.

- Priority: Medium

- Stimulus/Response: User selects the "Invoice" section.

**- Functional Requirements:**

- Access a list of invoices.

- Download or view individual invoices.

**4.22 Logout:**

- Description: Users can log out of their account.

- Priority: Medium

- Stimulus/Response: User selects the "Logout" option.

**- Functional Requirements:**

- Confirm logout action.

- Redirect to the login page.

**4.23 Login:**

- Description: Users can log into their accounts.

- Priority: High

- Stimulus/Response: User selects the "Login" option.

**- Functional Requirements:**

- Enter username and password.

- Validate credentials.

- Redirect to the user dashboard upon successful login.

**4.24 Signup:**

- Description: New users can create an account.

- Priority: High

- Stimulus/Response: User selects the "Signup" option.

**- Functional Requirements:**

- Provide necessary registration information.

- Validate and create a new account.

- Redirect to the user dashboard upon successful signup.

**4.25. Check Application Details:**

- Description: Users can view details and information about the application.

- Priority: Low

- Stimulus/Response: User selects the "Application Details" section.

**- Functional Requirements:**

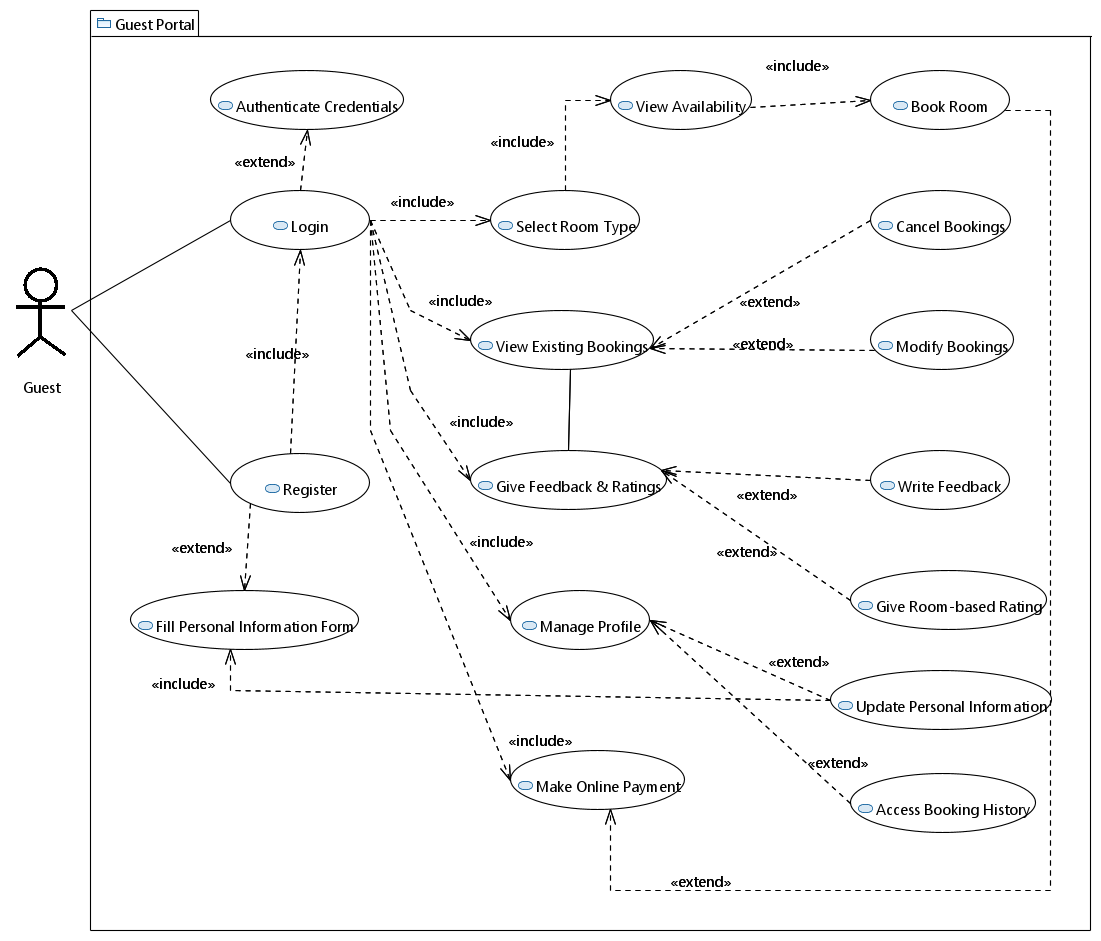
- Access information about the application.

- Read about features and policies.

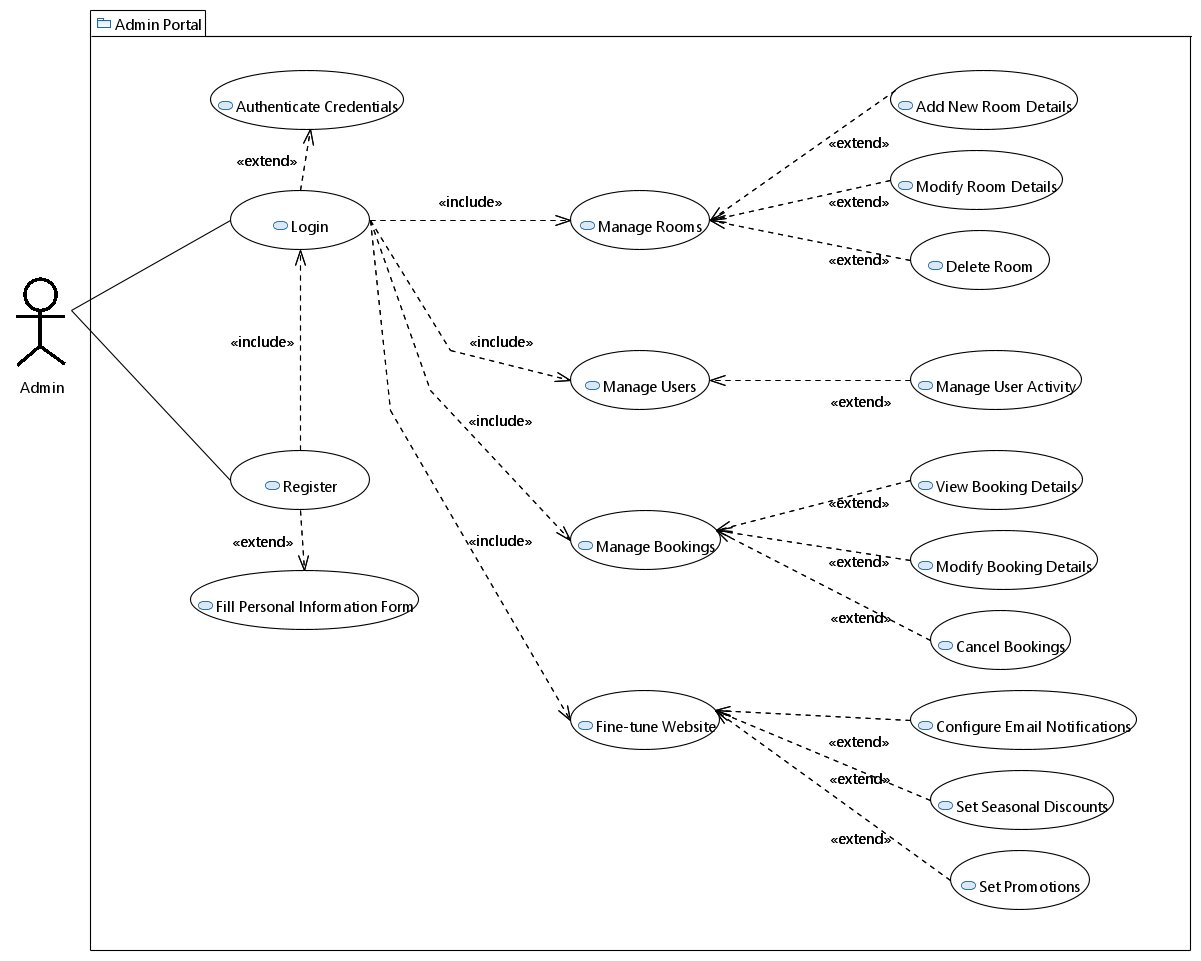
These descriptions provide a comprehensive overview of the functionalities for user profiles. Keep in mind that the priorities and requirements may vary based on the specific goals and focus of your application.

# Use Cases

* 1. **Use Case #01:**



|  |  |
| --- | --- |
| ID: | UC\_1 |
| Use Case Name: | Guest Registration and Management |
| Author: | Syed Abdul Rehman, Xabad Nadeem, Yusra Sohail |
| Last Revision: | 12/3/2023 |
| Description: | Guests can register on the hotel website, providing personal information to create an account. The registration process includes authenticating credentials, filling in a personal information form, and logging in. Once logged in, guests can manage various aspects such as selecting room types, viewing existing room bookings, giving feedback, making online payments, and managing their profile. |
| Primary Actor: | Guest |
| Supporting Actor: | None |
| Pre-Conditions: | Guest is not registered. |
| Post-Conditions: | Guest is registered, logged in, and can manage their profile and bookings. |
| Main Success Scenario: | 1. Guest navigates to the hotel website. 2. Guest selects the option to register. 3. The system authenticates the guest's credentials. 4. Guest fills in a personal information form. 5. The system registers the guest and extends to the login process. 6. Guest logs in using the registered credentials. 7. Upon successful login, the guest can: 8. Select a room type. 9. View room availability. 10. Book a room, extending to make online payment. 11. View existing room bookings, extending to cancel or modify bookings. 12. Give feedback, extend to writing feedback, providing room-based ratings. 13. Manage their profile, extending to updating personal information and accessing booking history. 14. Make online payments for booked rooms. |
| Extensions: | 4a. If registration fails:  The system notifies the guest, and the use case ends.  7a. If room availability cannot be retrieved:  The system notifies the guest, and the use case ends.  7b. If the guest decides to cancel or modify a booking:  Extends to the respective Use Cases (View Existing Bookings, Cancel Bookings, Modify Bookings).  7c. If there are issues with making online payments:  The system logs the issue, notifies support staff, and the use case ends.  7d. If the guest encounters problems updating personal information:  The system logs the issue, notifies support staff, and the use case ends.  7e. If the guest faces difficulties accessing booking history:  The system logs the issue, notifies support staff, and the use case ends.  Note: This use case scenario involves several extensions to handle potential issues or additional functionalities. |

* 1. **Use Case #02:**

|  |  |
| --- | --- |
| ID: | UC\_2 |
| Use Case Name: | Admin Management |
| Author: | Syed Abdul Rehman |
| Last Revision: | 12/3/2023 |
| Description: | The administrator can log in to the hotel management system to perform various administrative tasks. The login process involves authenticating the administrator's credentials. Once logged in, the administrator can manage rooms, users, bookings, and fine-tune website settings. |
| Primary Actor: | Admin |
| Supporting Actor: | None |
| Pre-Conditions: | Admin is not logged in. |
| Post-Conditions: | Admin is logged in and can perform administrative tasks. |
| Main Success Scenario: | 1. Admin navigates to the admin login page. 2. Admin logs in, extending to authenticating credentials. 3. Upon successful login, the admin can:  * Register, extending to filling in personal information. * Manage rooms, extending to:   Add a new room.  Modify room details.  Delete a room.   * Manage users, extending to managing user activity. * Manage bookings, extending to:   View booking details.  Modify booking details.  Cancel a booking.   * Fine-tune website, extending to:   Configure email notifications.  Set discounts.  Set promotions. |
| Extensions: | 2a. If admin authentication fails:  The system notifies the admin, and the use case ends.  3a. If admin encounters issues during any administrative task:  The system logs the issue, notifies support staff, and the use case ends.  5a. If there are issues while managing rooms (add, modify, delete):  The system logs the issue, notifies support staff, and the use case ends.  6a. If there are issues while managing users:  The system logs the issue, notifies support staff, and the use case ends.  7a. If there are issues while managing bookings (view, modify, cancel):  The system logs the issue, notifies support staff, and the use case ends.  8a. If there are issues while fine-tuning website settings:  The system logs the issue, notifies support staff, and the use case ends. |

* 1. **Use Case #03:**

A diagram of a company

Description automatically generated

|  |  |
| --- | --- |
| ID: | UC\_3 |
| Use Case Name: | Guest Registration and Management |
| Author: | Syed Abdul Rehman, Xabad Nadeem, Yusra Sohail |
| Last Revision: | 12/3/2023 |
| Description: | Guests can register on the hotel website, providing personal information to create an account. The registration process includes authenticating credentials, filling a personal information form, and logging in. Once logged in, guests can manage various aspects such as selecting room types, viewing existing room bookings, giving feedback, making online payments, and managing their profile. The system interacts with a database to store guest information, and administrators can log in to perform administrative tasks. |
| Primary Actor: | Guest |
| Supporting Actor: | Database, Admin, Room Management System |
| Pre-Conditions: | Guest is not registered. |
| Post-Conditions: | Guest is registered, logged in, and can manage their profile and bookings. |
| Main Success Scenario: | * Guest navigates to the hotel website. * Guest selects the option to register. * The system authenticates the guest's credentials. * Guest fills in a personal information form. * The system registers the guest and extends to the login process. * Guest logs in using the registered credentials.   Upon successful login, the guest can:   * Select a room type. * View room availability. * Book a room, extending to make online payment. * View existing room bookings, extending to cancel or modify bookings. * Give feedback, extend to writing feedback, providing room-based ratings. * Manage their profile, extending to updating personal information and accessing booking history. * Make online payments for booked rooms. * Provide feedback. * View available rooms. * View rates. * Request help/support. * Admin logs in using the admin credentials.   The system authenticates the admin's credentials with the database.  Admin can:   * Register a new admin, extending to login. * Login to the admin dashboard. * Fill in personal information. * Perform administrative tasks. |
| Extensions: | 4a. If registration fails:  The system notifies the guest, and the use case ends.  7a. If room availability cannot be retrieved:  The system notifies the guest, and the use case ends.  7b. If the guest decides to cancel or modify a booking:  Extends to the respective Use Cases (View Existing Bookings, Cancel Bookings, Modify Bookings).  7c. If there are issues with making online payments:  The system logs the issue, notifies support staff, and the use case ends.  7d. If the guest encounters problems updating personal information:  The system logs the issue, notifies support staff, and the use case ends.  7e. If the guest faces difficulties accessing booking history:  The system logs the issue, notifies support staff, and the use case ends.  9a. If admin authentication fails:  The system notifies the admin, and the use case ends.  10a. If admin encounters issues during administrative tasks:  The system logs the issue, notifies support staff, and the use case ends. |

**6. Other Non functional Requirements**

## 6.1 Performance Requirements

Performance requirements for the system describes the **user friendly** provided interface reliability and maintenance. The different functionalities provided by the system require **timely response** to maintain the performance. Performance requirements for the functional requirements are to have a **responsive** and **easy to use** interface which is understandable by user. To maintain different records and retrieval of stored information from the **database** also needs to be **fast** and **reliable** as the stored data must be displayed to users upon **rendering** to web page and the response must be fast.

## 6.2 Safety Requirements

## By Ensuring that only authorized personnel can access the MySQL database, the website prioritizes user data security, instilling user confidence and awareness.

## 6.3 Security Requirements

Using **RBAC** role based access control we have tried **to ensure security requirements** since only admin can access a lot of functionalities that are not even visible to the user.

## 6.4 Software Quality Attributes

Quality attributes that could be added useful to users would be to combine and support traditionalbehaviorand **single page applications**. This will result in maintaining good and reliable user experience. Traditional applications **load** new page every time a user requests are made which slows down the speed of navigation. Single page application does not entirely execute new page reloads which results in **faster** navigation and accordingly a flexible and maintainable user interface. Combining these two would result in more **stable website content** and **interaction based website** functions. Another software quality attribute that could be considered which would help in increasing the maintainability is the use of a more reliable and maintainable **database** platform. Introducing any **cloud based** platform to maintain and use the records would be more maintainable.

## 6.5 Business Rules

Operating principles for the product would be based on the functionalities that **user or admin** can perform. Like a user cannot perform **update**, d**elete**, or a**d**d operations on the product as this is functionality described only for admin and admin can perform this functionality only when he/she **logins into** their account with valid credentials. But if an admin wants to use the services of Hospitality Hub he/she needs to **create account** as a user first and then proceed further with the bookings.

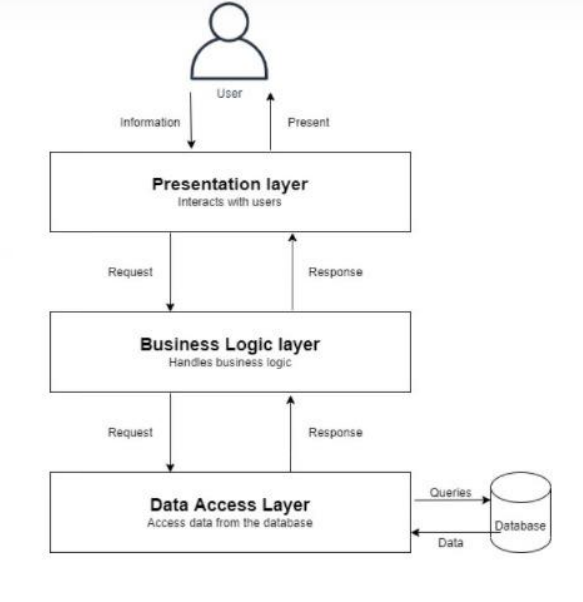
# System Architecture

# Below will be the detailed diagram that will capture the big picture of the project

* 1. **System Level Architecture**

System-level architecture refers to the high-level design and structure of a complex system, encompassing the arrangement and interaction of its components or modules. This type of architecture provides an abstract representation of the system, focusing on the relationships and dependencies between its various elements rather than on the specific details of their implementation

**7.2 Software Architecture**

****

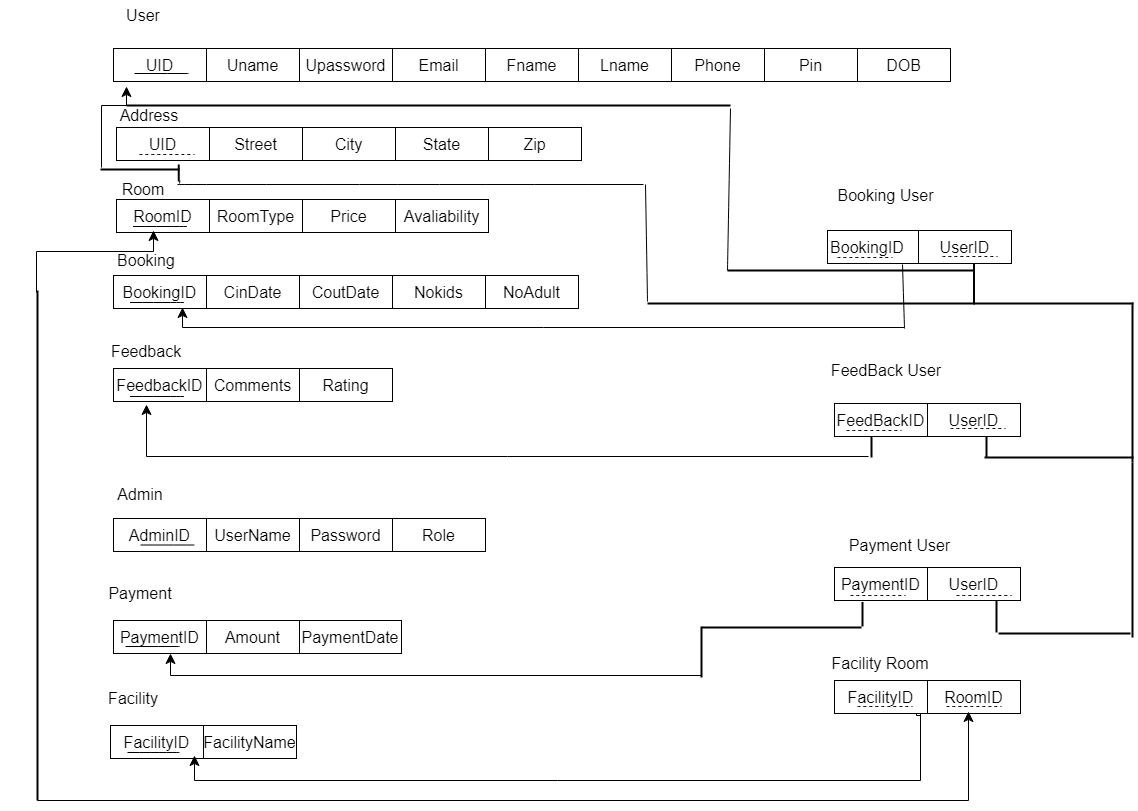
1. **Design Strategy**

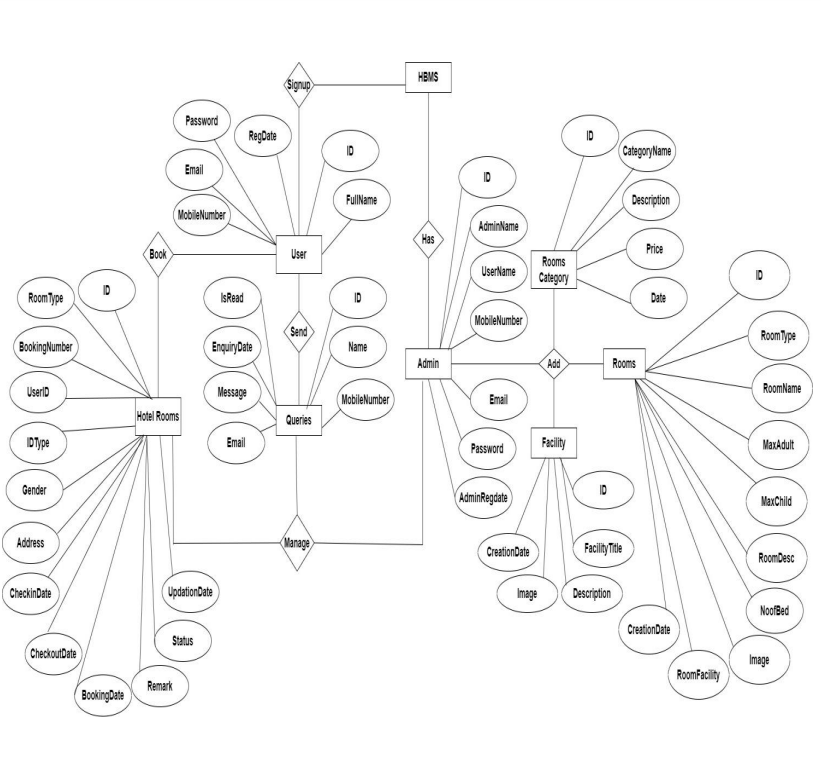
The design strategy focuses on modularity for future enhancements, reusable components, a minimalist user interface for simplicity, a relational database for efficient data management, careful synchronization for concurrency, and acknowledges trade-offs while prioritizing user-centric principles.

**9. Detailed System Design**

**UML DIAGRAM:**

**9.1 Database Design**

****

** 9.1.1 ER Diagram**

**9.1.2 DataDictionary**

**DATA DICTIONARY 1**

< Data Dictionary 1: tbladmin >

Name: tbladmin

Alias: None

Where-used/how-used:

Processes:

Admin Registration

Admin Login

Usage:

Input for Admin Registration process

Output from Admin Login process

Store for admin-related data

Content Description:

Notation: This table stores information about administrators.

Column Name:

ID

AdminName

UserName

MobileNumber

Email

Password

AdminRegdate

Column Descriptions:

ID:

Type: int(10)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: PK

AdminName:

Type: varchar(120)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

UserName:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

MobileNumber:

Type: bigint(10)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Email:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Password:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

AdminRegdate:

Type: timestamp

Length: Not applicable

Nullable: Yes

Default Value: current\_timestamp()

Key Type: None

**DATA DICTIONARY 2**

< Data Dictionary 2: tblbooking >

Name: tblbooking

Alias: None

Where-used/how-used:

Processes:

Room Booking

Booking Modification

Booking Cancellation

Usage:

Output from Room Booking process

Input for Booking Modification and Cancellation processes

Store for booking-related data

Content Description:

Notation: This table stores information about hotel room bookings.

Column Name:

ID

RoomId

BookingNumber

UserID

IDType

Gender

Address

CheckinDate

CheckoutDate

BookingDate

Remark

Status

UpdationDate

Column Descriptions:

ID:

Type: int(10)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: PK

RoomId:

Type: int(5)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

BookingNumber:

Type: varchar(120)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

UserID:

Type: int(5)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: None

IDType:

Type: varchar(120)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Gender:

Type: varchar(50)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Address:

Type: mediumtext

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

CheckinDate:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

CheckoutDate:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

BookingDate:

Type: timestamp

Length: Not applicable

Nullable: Yes

Default Value: current\_timestamp()

Key Type: None

Remark:

Type: varchar(50)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Status:

Type: varchar(50)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

UpdationDate:

Type: timestamp

Length: Not applicable

Nullable: Yes

Default Value: None ON UPDATE current\_timestamp()

Key Type: None

**DATA DICTIONARY 3**

< Data Dictionary 3: tblcategory >

Name: tblcategory

Alias: None

Where-used/how-used:

Processes:

Room Category Management

Price Configuration

Usage:

Input for Room Category Management and Price Configuration processes

Store for category-related data

Content Description:

Notation: This table stores information about hotel room categories.

Column Name:

ID

CategoryName

Description

Price

Date

Column Descriptions:

ID:

Type: int(10)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: PK

CategoryName:

Type: varchar(120)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Description:

Type: mediumtext

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Price:

Type: int(5)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: None

Date:

Type: timestamp

Length: Not applicable

Nullable: Yes

Default Value: current\_timestamp()

Key Type: None

**DATA DICTIONARY 4**

< Data Dictionary 8: tbluser >

Name: tbluser

Alias: None

Where-used/how-used:

Processes:

User Registration

User Login

Usage:

Input for User Registration and Login processes

Store for user-related data

Content Description:

Notation: This table stores information about users.

Column Name:

ID

FullName

MobileNumber

Email

Password

RegDate

Column Descriptions:

ID:

Type: int(10)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: PK

FullName:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

MobileNumber:

Type: bigint(10)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Email:

Type: varchar(120)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Password:

Type: varchar(120)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

RegDate:

Type: timestamp

Length: Not applicable

Nullable: Yes

Default Value: current\_timestamp()

Key Type: None

**DATA DICTIONARY 5:**

< Data Dictionary 7: tblroom >

Name: tblroom

Alias: None

Where-used/how-used:

Processes:

Room Management

Usage:

Input for Room Management process

Store for room-related data

Content Description:

Notation: This table stores information about hotel rooms.

Column Name:

ID

RoomType

RoomName

MaxAdult

MaxChild

RoomDesc

NoofBed

Image

RoomFacility

CreationDate

Column Descriptions:

ID:

Type: int(10)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: PK

RoomType:

Type: int(10)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

RoomName:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

MaxAdult:

Type: int(5)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

MaxChild:

Type: int(5)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

RoomDesc:

Type: mediumtext

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

NoofBed:

Type: int(5)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Image:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

RoomFacility:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

CreationDate:

Type: timestamp

Length: Not applicable

Nullable: Yes

Default Value: current\_timestamp()

Key Type: None

**DATA DICTIONARY 6:**

< Data Dictionary 6: tblpage >

Name: tblpage

Alias: None

Where-used/how-used:

Processes:

Website Page Management

Usage:

Input for Website Page Management process

Store for page-related data

Content Description:

Notation: This table stores information about website pages.

Column Name:

ID

PageType

PageTitle

PageDescription

Email

MobileNumber

UpdationDate

Column Descriptions:

ID:

Type: int(10)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: PK

PageType:

Type: varchar(120)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

PageTitle:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

PageDescription:

Type: mediumtext

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Email:

Type: varchar(120)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

MobileNumber:

Type: bigint(10)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

UpdationDate:

Type: timestamp

Length: Not applicable

Nullable: Yes

Default Value: None ON UPDATE current\_timestamp()

Key Type: None

**DATA DICTIONARY 7:**

< Data Dictionary 5: tblfacility >

Name: tblfacility

Alias: None

Where-used/how-used:

Processes:

Facility Management

Usage:

Input for Facility Management process

Store for facility-related data

Content Description:

Notation: This table stores information about hotel facilities.

Column Name:

ID

FacilityTitle

Description

Image

CreationDate

Column Descriptions:

ID:

Type: int(10)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: PK

FacilityTitle:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Description:

Type: mediumtext

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Image:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

CreationDate:

Type: timestamp

Length: Not applicable

Nullable: Yes

Default Value: current\_timestamp()

Key Type: None

**DATA DICTIONARY 8:**

< Data Dictionary 9: tblcontact >

Name: tblcontact

Alias: None

Where-used/how-used:

Processes:

Contact Inquiry Management

Usage:

Input for Contact Inquiry Management process

Store for contact-related inquiries

Content Description:

Notation: This table stores information about contact inquiries.

Column Name:

ID

Name

MobileNumber

Email

Message

EnquiryDate

IsRead

Column Descriptions:

ID:

Type: int(10)

Length: Not applicable

Nullable: No

Default Value: None

Key Type: PK

Name:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

MobileNumber:

Type: bigint(10)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Email:

Type: varchar(200)

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

Message:

Type: mediumtext

Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

EnquiryDate:

Type: timestamp

Length: Not applicable

Nullable: Yes

Default Value: current\_timestamp()

Key Type: None

IsRead:

Type: int(5)

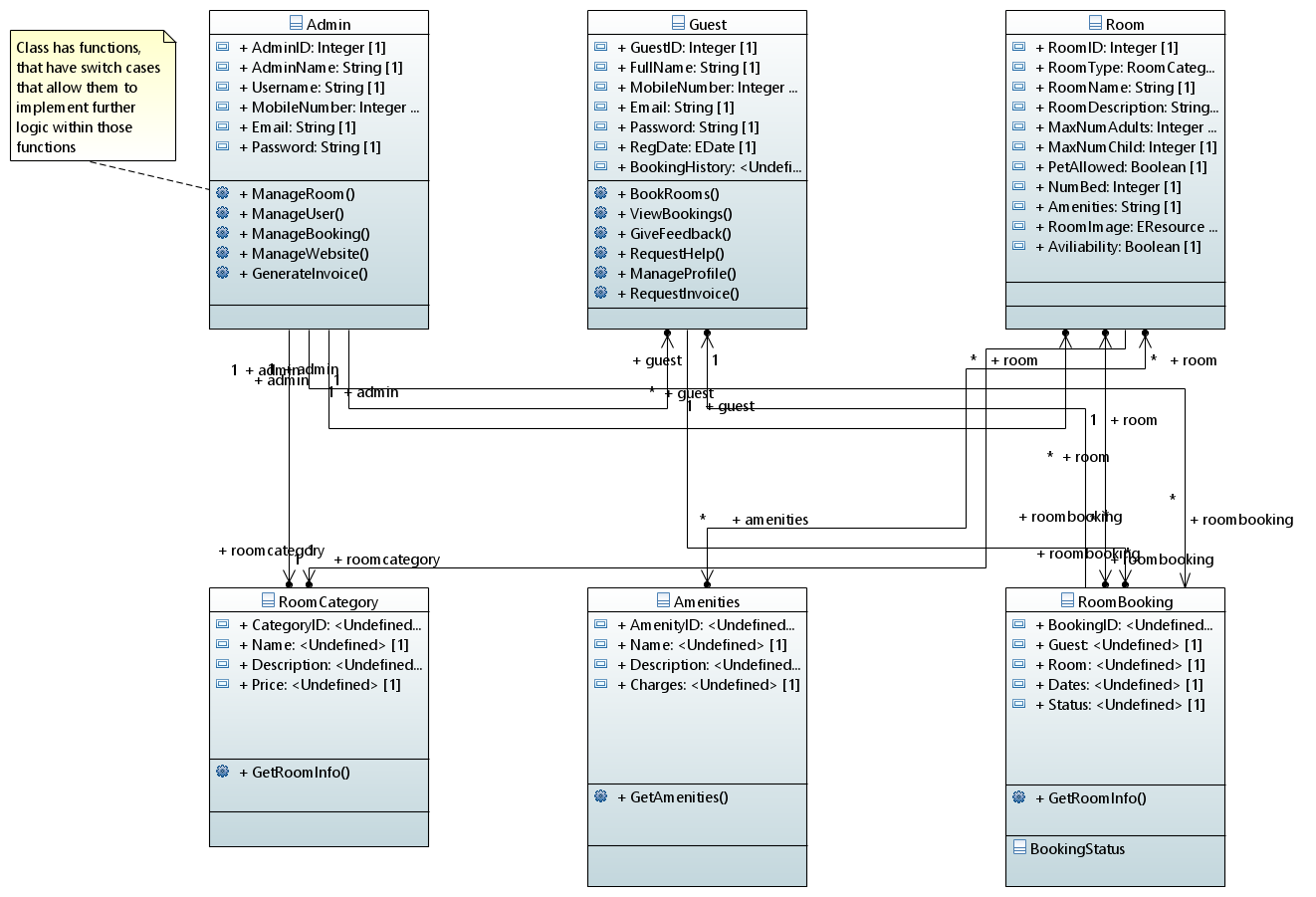
Length: Not applicable

Nullable: Yes

Default Value: None

Key Type: None

**9.2 Application Design**



**A screenshot of a computer

Description automatically generated9.2.1 Activity Diagrams**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A diagram of a company

Description automatically generated

A diagram of a company

Description automatically generated

A diagram of a company

Description automatically generated

* **COMPONENT DIAGRAM:**

A diagram of a computer program

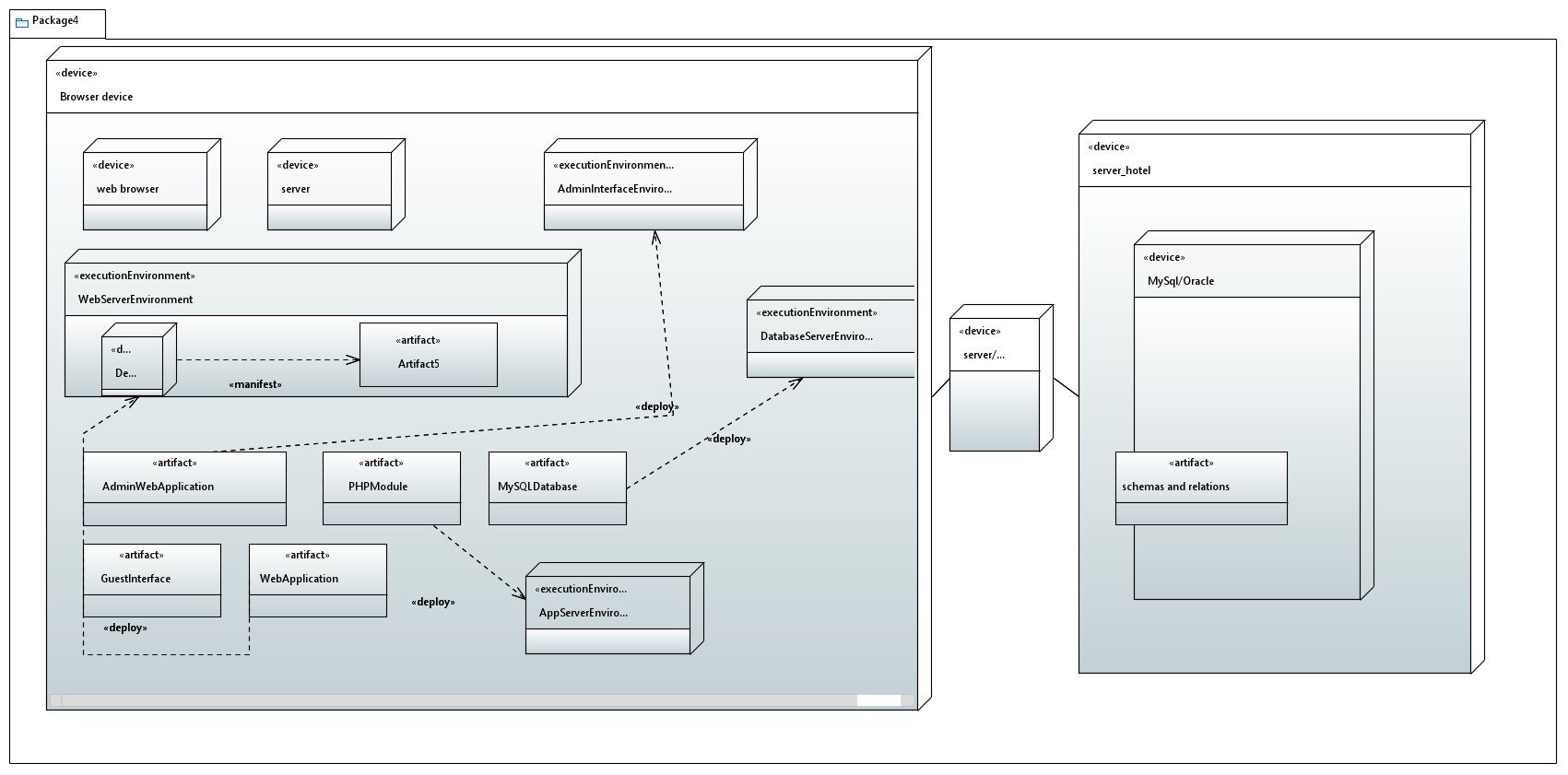
Description automatically generated

**SEQUENCE DIAGRAM**

**A diagram of a flowchart

Description automatically generated**

* **DEPLOYMENT DIAGRAM:**



* **STATE CHART DIAGRAM:**

A diagram of a company

Description automatically generated

**11. References**

1. <https://www.travelperk.com/blog/best-websites-for-booking-hotels/>
2. <https://99designs.com/inspiration/websites/hotel>
3. <https://www.thirdrocktechkno.com/blog/software-design-document-vs-software-requirement-specification/>
4. <https://app.diagrams.net/>
5. <https://www.protel.net/hotel-management-system/>

**Appendix B: To Be Determined List**

* Our project is **limited** to the **citizens of Pakistan** and services are only provided within Pakistan. Although, we are trying to **extend** HOSPITALITYHUB **outside** Pakistan. Once we get **successful**, this SRS will be again referred to make some **changes** in **functional**, **nonfunctionalrequirements** along with specifying the **stakeholders**.