|  |  |  |
| --- | --- | --- |
|  |  | logo |

Project Name: Hotel Paradise Management and Billing System

Project Name: Hotel Paradise Management and Billing System

Date: 8-12-2023

Copyright Notice

This document contains proprietary information of HCL Technologies Ltd. No part of this document may be reproduced, stored, copied, or transmitted in any form or by means of electronic, mechanical, photocopying or otherwise, without the express consent of HCL Technologies. This document is intended for internal circulation only and not meant for external distribution.

Revision History

| Version No | Date | Prepared by / Modified by | Significant Changes |
| --- | --- | --- | --- |
| 1.0 | 28-11-2023 | Yogesh | Provided JDBC backup |
| 1.1 | 03-12-2023 | Yogesh | Fixed some bugs |

**Table of Contents:**

**1. Project Overview**

1.1 Scope

1.2 Out of Scope

1.3 Intended Audience

1.4 High-Level Use Cases

1.5 Use Cases Detailed

1.6 User Interface Modules

1.7 Technical Architecture

1.8 Technical Detailed Description

1.9 Sequence Diagram

1.10 Database Entity Relationship Diagram

1.11 Standards

1.12 Non-Functional Requirements

**2.** **Request for Proposal**

**1. Project Overview:**

The Hotel Paradise Management and Billing System is meticulously crafted to elevate the efficiency of hotel operations, seamlessly managing reservations, guest information, and billing processes. Imagine a system that simplifies the entire experience for both guests and staff. It ensures that checking in is a breeze, with reservations handled effortlessly. Each guest's preferences and special requests are tracked, creating a tailored experience. The billing process becomes straightforward, eliminating any complexities during check-out. The system aims to create a seamless guest experience, from check-in to check-out, providing a reliable and enjoyable stay. Behind the scenes, it acts as a diligent archivist, maintaining a systematic record of all hotel activities. In essence, the Hotel Paradise Management and Billing System is the comprehensive solution for running a hotel smoothly, acting as a reliable friend who takes care of all the details, allowing both guests and staff to focus on what matters most – an exceptional hotel experience.

**1.1. Scope:**

The Hotel Management and Billing System aim to streamline the operations of managing hotel reservations, guest information, and billing. It focuses on providing efficient services to guests and maintaining an organized record of hotel activities.

| **Module** | **Roles** | **Description** |
| --- | --- | --- |
| Booking | User/Admin | To book rooms for new guests and issue them a unique ID. |
| Payment | User/Admin | To make payment by using guests phone number. |
| Record | Admin | Only Admin people can view record by providing valid credentials. |
|  |  |  |

**1.2 Out of Scope:**

* User Interface
* Hardware Interface
* Software Interface

**1.3 Intended Audience:**

Data Carriers:

* Analysts gathering and organizing documentation related to data carriers.
* Questionnaires for efficient responses from a large number of people.
* Personal Interviews to understand user needs.
* Observation for observing ongoing system operations.

**1.4 High-Level Use Cases:**

A. Guest Reservation:

* Actor: Guest
* Description: Guests can create reservations for their stay by providing necessary details such as check-in/out dates, room preferences, and personal information.

B. User Authentication:

* Actor: System
* Description: Users (both guests and administrators) can securely log in to the system using their credentials, ensuring data privacy and access control.

C. View Booking Details:

* Actor: Guest
* Description: Guests can access and view details of their current and past bookings, including room information, reservation dates, and billing summaries.

D. Billing and Invoicing:

* Actor: System
* Description: The system automatically generates bills and invoices for guests based on their stay duration, additional services, and any applicable charges.

E. Room Availability Management:

* Actor: Admin
* Description: Administrators can view and manage the availability of rooms, update room statuses, and handle reservations to optimize room occupancy.

F. Reporting and Analytics:

* Actor: Admin
* Description: Administrators can access comprehensive reports and analytics, including occupancy rates, revenue summaries, and other key performance indicators for strategic decision-making.

G. Check-in and Check-out:

* Actor: Guest
* Description: Guests can complete the check-in process efficiently upon arrival and check-out seamlessly at the end of their stay, ensuring a smooth experience.

H. User Management:

* Actor: Admin
* Description: Administrators have the authority to manage user accounts, including creating new accounts, updating credentials, and deactivating accounts when necessary.

**1.5 Use Cases Detailed:**

Testing Phases:

Unit Testing in Hotel Paradise Management and Billing System:

Ensuring Every Component Serves Its Purpose.

In our Hotel Management and Billing System, Unit Testing plays a crucial role, much like the meticulous inspection of individual ingredients before creating a delightful dish. Imagine each software component as a unique ingredient, and Unit Testing ensures that each one functions seamlessly within the system.

**1.6 User Interface Modules:**

1. Self-Review: The Chef's Initial Taste Test:

Just as a chef tastes their own creation, our developers initiate the process with a Self-Review. During this phase, the original code creator takes the first bite, meticulously checking for correctness and functionality. It's like ensuring the recipe is followed to perfection.

2. Peer Review: A Collaborative Taste-Testing Session:

Moving to the next stage is the Peer Review, comparable to a collaborative taste-testing session in a kitchen. Another team member, not involved in the initial development, tests the code. This step brings in a fresh perspective, catching any subtleties that might have been missed during the self-review. It's like having a second set of taste buds to identify nuances.

3. SQA Review: The Grand Tasting Event:

The grand finale is the Software Quality Assurance (SQA) Review – the ultimate tasting event. Here, the SQA team rigorously examines the code, leaving no stone unturned to ensure its quality and adherence to standards. It's akin to a panel of expert chefs conducting the final taste test to guarantee a top-notch dish.

4. In Essence: Crafting Harmony in Every Code Unit:

Our Unit Testing approach goes beyond individual code units; it ensures that each piece of code contributes harmoniously to the entire system. Just as a well-coordinated kitchen produces a delightful feast, our approach guarantees the seamless operation of our innovative hospitality solution. Each code unit isn't just standalone; it's an integral part of the culinary masterpiece that is our Hotel Paradise Management and Billing System.

2.1 Integration Testing:

In the world of our Hotel Paradise Management and Billing System, Integration Testing is a pivotal phase ensuring that individual software modules come together seamlessly. This phase combines units that have undergone rigorous Unit Testing, grouping them into larger aggregates for comprehensive testing as defined in our test plan. There are distinct approaches to this integration dance.

2.1.1 Big Bang Approach:

Picture this as the grand ball where all developed modules are elegantly coupled together, forming a complete software system. This collective entity is then subjected to Integration Testing, ensuring the harmony of its components.

2.1.2 Bottom-Up Approach:

Taking a step-by-step approach, this method integrates low-level modules, procedures, or functions first. After testing these integrated components, the next level of modules takes center stage for further Integration Testing.

2.1.3 Top-Down Approach:

In this incremental dance, modules are integrated and tested by moving downward through the control hierarchy. Starting from the top, each level of modules is integrated and rigorously tested, ensuring a solid foundation for the entire system.

2.2 Alpha Testing:

Imagine a sneak peek event for our Hotel Paradise Management and Billing System – Alpha Testing. Users are invited to the development center, exploring the application while developers keenly observe every input and action. Any abnormal behavior noted is swiftly rectified, ensuring a polished and user-friendly system.

2.3 System Testing:

As the Hotel Paradise Management and Billing System prepares for its grand debut, System Testing takes center stage. This phase ensures that the system operates accurately and effectively before live operations commence. Both the logical and physical designs undergo thorough scrutiny on paper to guarantee their seamless implementation.

2.4 Validation Testing:

In the validation gala, every component in our software, crucial for proper functioning, undergoes rigorous testing. Validation Testing ensures that our system is not just operational but perfect for its intended purpose. This meticulous process, occurring at the end of the development journey, takes place after verification, confirming that our system aligns with requirements and fulfills organizational goals and user needs.

In alignment with the Capability Maturity Model (CMMI-SW v1.1), validation becomes a refined process, evaluating our software to determine its satisfaction of specified requirements.

**1.8 Technical Architecture:**

The heartbeat of our Hotel Paradise Management and Billing System lies in its intricate technical architecture. Let's delve into the design intricacies that pave the way for the creation of this innovative system.

1. System Design Approach:

Our system design is not just a solution; it's an artful approach to crafting a new paradigm in hotel management. The design phase furnishes the profound understanding and procedural details essential for implementing the system, as recommended during the comprehensive system study.

2. Logical and Physical Design Phases:

As our system evolves, it traverses two crucial developmental phases: logical design and physical design. The logical design orchestrates the flow of our system, defining its boundaries through detailed Data Flow Diagrams (DFD). On the other hand, the physical design scrutinizes the existing physical system, preparing output specifications, editing controls, security specifications, and an implementation plan. This phase involves a meticulous walkthrough of the information flow, reviewing benefits, costs, target data, and system constraints.

3. Creative Essence of Software Design:

In the realm of software design, creativity takes center stage. A well-designed system correlates directly with the quality and accuracy of the software developed. The design of our information system meticulously outlines how our system meets the identified requirements from the early stages. This phase, often referred to as logical design by system specialists, determines how outputs are produced and in what format. The design shapes input data and master files to meet the proposed output requirements.

4. Phases in System Design:

The design phase encompasses critical steps:

* + Review of the current physical system.
  + Preparation of output specifications, determining formats, content, and frequency of reports.
  + Preparation of input specifications.
  + Development of edit control and security specifications.
  + Specification of the implementation plan.
  + Conducting a logical design walkthrough of information flow, output, input controls, and the implementation plan.

5. Input Design:

Input design transforms user-oriented input into a computer-based format, ensuring error-free data entry operations. This pivotal stage determines how data is fed into the system, ensuring specificity and accuracy.

6. Output Design:

As the most direct information source to users, computer output design is a meticulous process. It involves designing outputs, such as reports, aligned with user requirements. A well-crafted output design enhances the system's relationship with users and facilitates decision-making for management.

7. Database Design:

In the heartbeat of our system design, database design takes a prominent role. Common data, shared across applications, is managed as an entity by our database software. The primary objectives here are fast response times to inquiries, maximizing information availability at a low cost, and controlling redundancy.

Our technical architecture lays the foundation for a Hotel Management and Billing System that not only meets but exceeds expectations, setting new standards in efficiency and innovation.

**1.9 Technical Detailed Description:**

The Technical Detailed Description encapsulates the robust framework and functionality of our Hotel Paradise Management and Billing System, a dynamic and integrated solution tailored for seamless hotel operations. Let's explore the intricacies of this technical landscape.

1. System Solution and Approach:

The design philosophy behind our Hotel Management and Billing System is rooted in providing a comprehensive solution for the hotel industry. This approach entails understanding the unique needs of hotel operations and translating them into a system that streamlines processes and enhances overall efficiency.

2. Logical and Physical System Design:

Our system undergoes a meticulous two-phase development, starting with logical design and progressing to physical design. The logical design delineates the system's flow through detailed diagrams, defining its operational boundaries. On the other hand, the physical design involves an in-depth review of the existing physical system, creating output specifications, configuring edit controls and security measures, and specifying the implementation plan.

3. Creative Aspect of Software Design:

Creativity takes center stage in our software design, influencing the quality and accuracy of the final product. This design phase determines how the system meets identified requirements, shaping input data and master files to align with proposed output needs.

4. System Development Phases:

* Review of Current System:

A comprehensive review of the existing physical system sets the stage for improvements and enhancements.

* Output Specification:

Detailed output specifications are prepared, defining the format, content, and frequency of reports, including terminal specifications.

* Input Specification:

Input specifications are crafted to ensure error-free data entry operations and meet the specific needs of the proposed output.

* Edit Control and Security Specification:

Measures are defined to ensure the integrity and security of the system, including controls for editing and securing data.

* Implementation Plan:

A detailed plan outlines the implementation process, addressing the practical aspects of deploying and integrating the system.

* Logical Design Walkthrough:

A walkthrough of the logical design covers information flow, output, input controls, and the implementation plan, ensuring a comprehensive understanding of the system's workings.

* Input Design:

This phase focuses on converting user-oriented input into a computer-based format, ensuring error-free data entry operations and specifying how data is fed into the system.

* Output Design:

Computer output, a crucial information source for users, is designed meticulously to align with user requirements. This enhances the system's relationship with users and aids decision-making for management.

* Database Design:

Central to our system, database design ensures efficient data management. Shared across applications, the database handles data as an entity, aiming for fast response times, information availability, and redundancy control.

Our Technical Detailed Description sets the stage for a Hotel Management and Billing System that not only meets but exceeds industry standards, bringing innovation and efficiency to the forefront of hotel operations.

**1.9 Sequence Diagram:**

...

**1.10 Database Entity Relationship Diagram:**

**1.11 Standards:**

1. User Interface Standards:

* + Consistent color scheme and layout across all screens.
  + Intuitive navigation for users of varying technical expertise.
  + Accessibility standards compliant with WCAG guidelines.

2. Coding Standards (Java):

* + Follow Java Naming Conventions for variable and method naming.
  + Implement proper exception handling to ensure robustness.
  + Adhere to SOLID principles for object-oriented design.

3. Database Design Standards:

* + Normalize the database to reduce redundancy and ensure data integrity.
  + Follow ACID properties to maintain consistency in database transactions.

4. Security Standards:

* + Implement encryption for sensitive data, such as user passwords.
  + Apply role-based access control to restrict system access based on user roles.

5. Performance Standards:

* + Optimize database queries for efficient data retrieval.
  + Implement caching mechanisms to enhance system performance.

6. Documentation Standards:

* + Maintain comprehensive and up-to-date documentation for code, database schema, and system architecture.
  + Use Javadoc for inline code documentation.

7. Testing Standards:

* + Adopt Test-Driven Development (TDD) practices for code reliability.
  + Perform regular regression testing after system updates.

8. Compliance Standards:

* + Adhere to relevant industry regulations and compliance standards.
  + Ensure GDPR compliance for handling user data.

**1.12 Non-Functional Requirements:**

1. Performance:

* + The system should handle a minimum of 1000 concurrent users during peak hours.
  + Response time for basic operations (e.g., room reservation, billing) should be within 2 seconds.

2. Scalability:

* + The system should be scalable to accommodate a 20% increase in users annually.
  + Database capacity should scale to handle at least 5 years of historical data.

3. Reliability:

* + The system should have a 99.99% uptime, allowing for scheduled maintenance windows.
  + In the event of a system failure, data recovery should be achieved within 2 hours.

4. Security:

* + User authentication should follow industry-standard encryption protocols.
  + Access to sensitive information, such as financial transactions, should be restricted based on user roles.

5. Usability:

* + The user interface should be intuitive, requiring minimal training for hotel staff.
  + The system should support multiple languages to cater to a diverse user base.

6. Compatibility:

* + The system should be compatible with popular web browsers (Chrome, Firefox, Safari, Edge).
  + Mobile responsiveness is required to support users accessing the system from various devices.

7. Maintainability:

* + Code should be well-documented, following industry standards.
  + The system should support seamless updates and patches without disrupting ongoing operations.

8. Compliance:

* + The system should adhere to data protection regulations such as GDPR.
  + Payment processing should comply with Payment Card Industry Data Security Standard (PCI DSS).

9. Auditability:

* + All user actions, especially those related to financial transactions, should be logged for auditing purposes.
  + Logs should be retained for a minimum of 12 months.

|  |  |  |
| --- | --- | --- |
|  | Request for Proposal | logo |