

**Contents**

1. Introduction
2. Extreme Programming (XP)
3. Methodology
   1. Requirement Gathering
   2. Iteration Planning
   3. Design and Prototyping
   4. Development and Iteration
   5. Testing and Continuous Integration
   6. Deployment and Feedback
4. Functional and Non-Functional Requirements
5. Work Breakdown Structure (WBS)
6. Schedule and Gantt Chart
7. Diagrams
   1. ERD Diagram
   2. Class Diagram
   3. Sequence Diagram
   4. Use Case Diagram
   5. DFD Diagram
8. Test Cases
9. Conclusion

**Description**

**Prepared by: Haseeb Khan**

This document outlines the design, development, and implementation of the Hotel Reservation System, a web-based application tailored to streamline hotel room bookings for both guests and administrators. The system focuses on delivering a user-friendly interface for customers to search, book, and manage reservations, while empowering hotel staff to efficiently control room availability, pricing, and reporting.

Developed using Extreme Programming (XP) methodology, the project emphasizes iterative development, stakeholder collaboration, and adaptability to evolving hospitality industry needs. The documentation includes functional requirements, system diagrams, test cases, and a detailed work breakdown structure to ensure alignment with user expectations and technical feasibility.

Key highlights:

* Role-based access for guests, staff, and administrators
* Integration with payment gateways for secure transactions
* Real-time updates for room availability and pricing
* Automated reporting for revenue and occupancy analysis

**Introduction**

The Hotel Reservation System is a web-based application designed to manage hotel room bookings, allowing users to search for available rooms, make reservations, and complete transactions. It also provides hotel admins with a panel to manage room availability, pricing, and reservations. Due to the dynamic nature of the hotel industry, where customer feedback and frequent changes in requirements are common, **Extreme Programming (XP)** is chosen for this project. XP focuses on delivering high-quality software through continuous communication, frequent releases, and close collaboration between developers and stakeholders.

**Extreme Programming (XP)**

Extreme Programming is ideal for the Hotel Reservation System because of the following reasons:

* **Frequent Releases and Feedback**: Delivers incremental releases and incorporates stakeholder feedback.
* **High Customer Involvement**: Maintains constant communication with hotel managers, staff, and users.
* **Adaptability**: Supports rapid changes in requirements.
* **Test-Driven Development (TDD)**: Ensures high-quality, bug-free code through automated tests.

**Methodology**

**Requirement Gathering**

* **User Stories**:
* *As a user*, I want to search for available rooms based on travel dates and preferences.
* *As an admin*, I want to manage room availability and pricing.
* **Tools**:
* Interviews and focus groups
* Story Mapping
* Trello/Jira for task tracking

**Iteration Planning**

* Development will be broken into short iterations of 2-3 weeks to allow rapid feedback and frequent releases.
* **Feature Prioritization:** Based on business value, high-priority features will be developed first. The first iterations will focus on essential features such as user registration, room search, and booking.

**Example Iteration Breakdown:**

* Iteration 1: Implement user registration and authentication.
* Iteration 2: Develop room search functionality with filters.
* Iteration 3: Develop the reservation system.
* Iteration 4: Create the admin panel for managing room availability and pricing.
* Iteration 5: Implement payment processing.
* Iteration 6: Finalize the system and conduct user acceptance testing.

**Design & Prototyping**

* **User Stories and Design:** In XP, design evolves alongside development. Initial wireframes or mockups will be developed based on user stories, but the design will be adjusted after each iteration based on user feedback.

**Development**

* **Pair Programming:** Developers will work in pairs, with one writing code and the other reviewing it.
* **Test-Driven Development:** Automated tests will be written before coding.
* **Frequent Refactoring:** Code will be refactored continuously to ensure maintainability.

**Testing & CI/CD**

* **Automated Testing:** Unit tests, integration tests, and UI tests will be implemented.
* **Customer Tests:** Stakeholders will perform user acceptance testing (UAT) during each iteration.
* **Frequent Builds:** Continuous integration tools will be used to test the system after each commit.

**Deployment & Feedback**

* **Frequent Releases:** Working features will be deployed after each iteration.
* **Customer Involvement:** Real-time feedback will be collected to adjust requirements or suggest improvements.
* **Feedback Loops:** New features will be prioritized based on user input.

**Functional and Non-Functional Requirements**

**Functional Requirements**

**Users**:

* Register and log in to the system.
* Search for available rooms based on travel dates, preferences, and price.
* Book rooms and receive confirmation.
* Make payments through an integrated payment gateway

**Admins**:

* Register and log in to the system.
* Add, update, or remove room details, pricing, and availability.
* View and manage reservations.
* Generate reports on bookings and revenue.

**Non-Functional Requirements**

* **Performance:** The system should handle up to 500 concurrent users without performance degradation.
* **Scalability:** The system must support additional features or increased user demand with minimal modifications.
* **Security:** User data, including payment details, must be encrypted.
* **Usability:** The system should be intuitive and user-friendly for both users and admins.
* **Reliability:** The system should have 99.9% uptime.
* **Compatibility:** The system should be accessible on all major web browsers and mobile devices.

**Work Breakdown Structure (WBS)**

**Level 1: Hotel Reservation System**

**Requirement Analysis**

* Stakeholder Interviews
* User Story Collection
* Requirement Validation

**System Design**

* Database Design
* Wireframes and Mockups
* Architecture Selection

**Development**

**User Module**:

* Registration/Login
* Room Search
* Booking System
* Payment Integration

**Admin Module:**

* Room Management
* Booking Management
* Report Generation

**Testing**

* Unit Testing
* Integration Testing
* User Acceptance Testing

**Deployment**

* Hosting Setup
* Deployment
* Post-Deployment Feedback Collection

**Maintenance and Support**

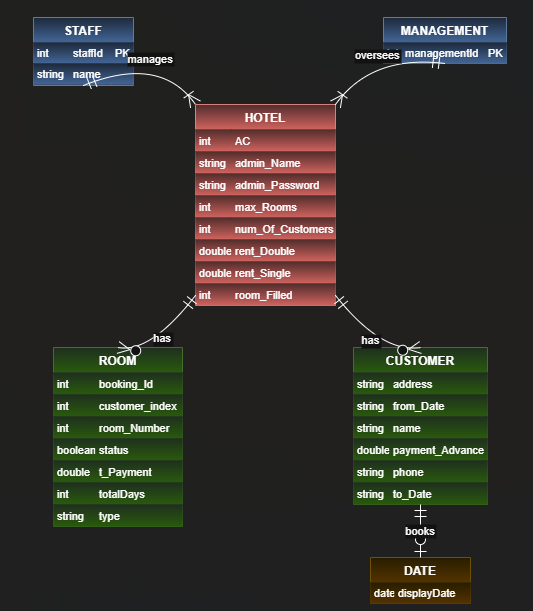
* Bug Fixes
* Feature Enhancements

**Schedule Chart**

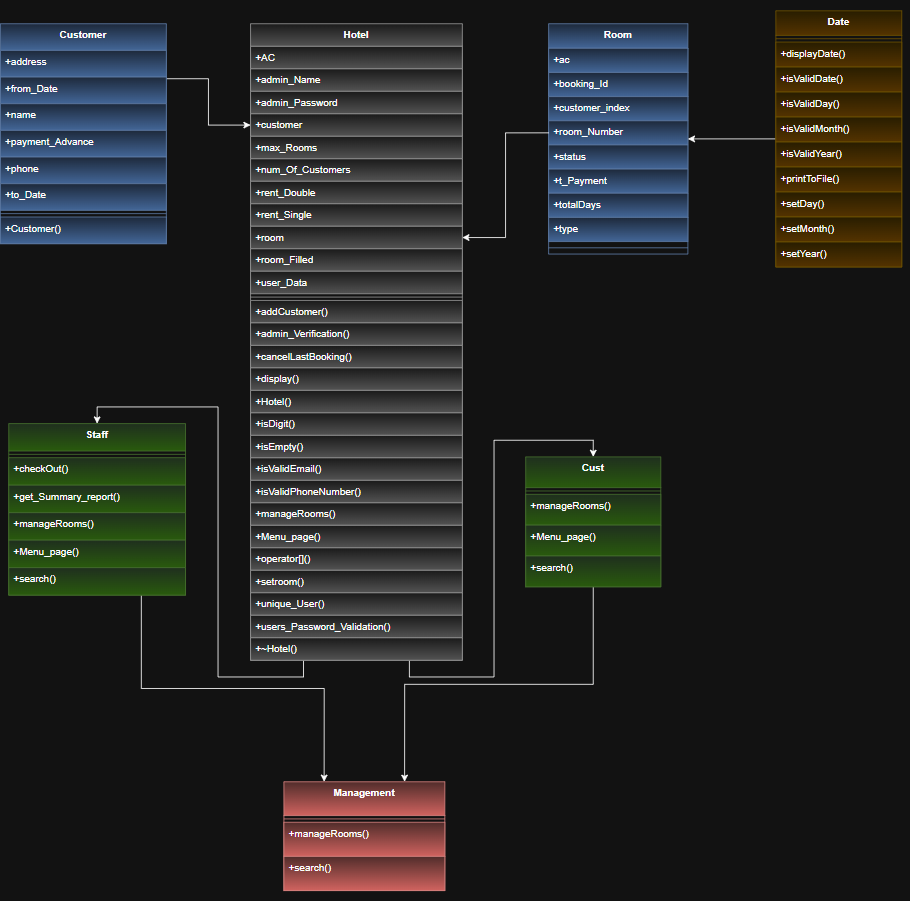
|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start Date** | **End Date** | **Duration** |
| Requirement Analysis | 01/01/2024 | 15/01/2024 | 2 weeks |
| System Design | 16/01/2024 | 30/01/2024 | 2 weeks |
| Development (Iterations) | 01/02/2024 | 31/08/2024 | 7 months |
| Testing | 01/09/2024 | 31/10/2024 | 2 months |
| Deployment | 01/11/2024 | 15/11/2024 | 2 weeks |
| Maintenance and Support | 16/11/2024 | 31/12/2024 | 1.5 months |

**Diagrams**

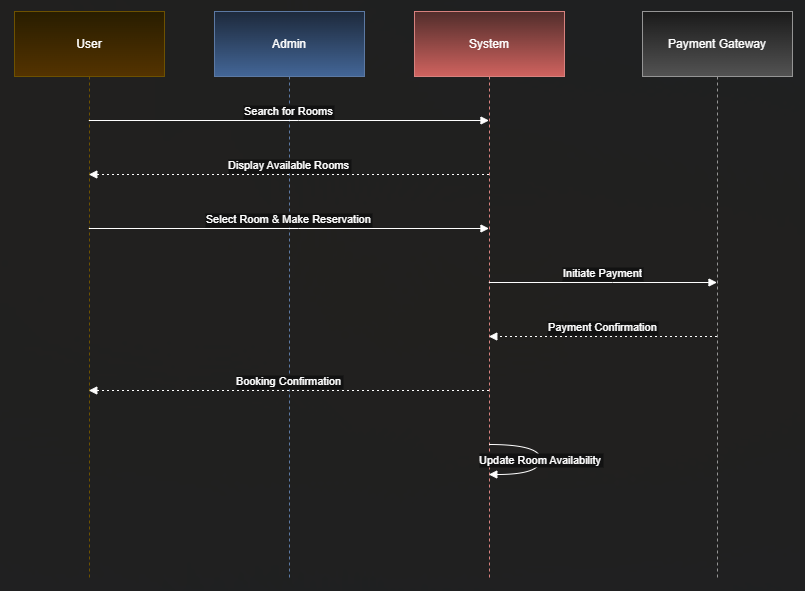
**ERD Diagram:**

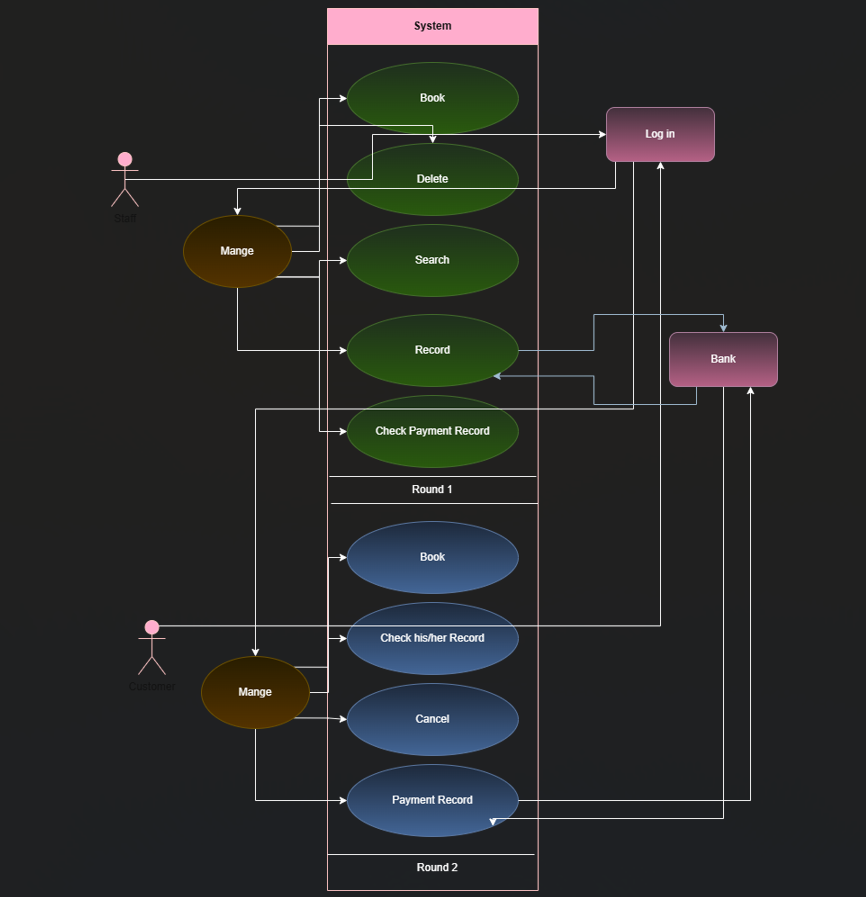
****

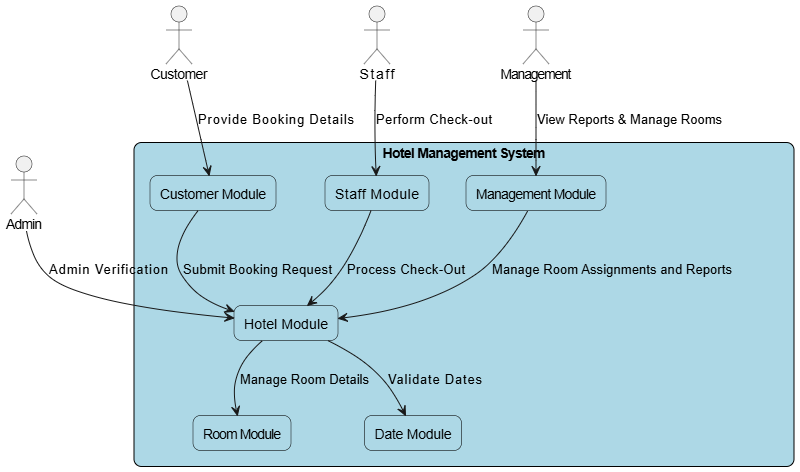
**Class Diagram:**



**Sequence Diagram:**

****

**Use Case Diagram:**

**DFD Diagram:**

**Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Preconditions** | **Test Steps** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| TC-01 | User registration with valid details | None | 1. Open the registration page 2. Fill in valid user details 3. Click "Register" | User is successfully registered and redirected to the login page |  |  |
| TC-02 | Login with valid credentials | User is registered | 1. Open login page 2. Enter valid credentials 3. Click "Login" | User is logged in successfully |  |  |
| TC-03 | Login with invalid password | User is registered | 1. Open login page 2. Enter valid username 3. Enter invalid password 4. Click "Login" | Error message "Invalid password" is shown |  |  |
| TC-04 | Room search based on travel dates | User is logged in | 1. Navigate to the search page 2. Enter valid travel dates 3. Click "Search" | List of available rooms is displayed |  |  |
| TC-05 | Booking a room successfully | User has searched for rooms | 1. Select an available room 2. Click "Book" 3. Confirm booking details 4. Make payment | Booking confirmation is displayed |  |  |
| TC-06 | Booking a room without selecting a room | User is logged in | 1. Navigate to booking page 2. Click "Book" without selecting a room | Error message "Please select a room" is displayed |  |  |
| TC-07 | Admin adds a new room successfully | Admin is logged in | 1. Navigate to the admin panel 2. Click "Add Room" 3. Enter room details 4. Click "Save" | Room is successfully added and listed |  |  |
| TC-08 | Payment processing with valid details | User has selected a room | 1. Navigate to payment page 2. Enter valid payment details 3. Click "Pay" | Payment is processed and booking is confirmed |  |  |
| TC-09 | Payment processing with invalid details | User has selected a room | 1. Navigate to payment page 2. Enter invalid payment details 3. Click "Pay" | Error message "Payment failed" is displayed |  |  |
| TC-10 | Viewing booking history | User is logged in and has previous bookings | 1. Navigate to "My Bookings" page | List of past bookings is displayed |  |  |

**Conclusion**

The Extreme Programming (XP) approach provides a solid foundation for developing the Hotel Reservation System with a focus on high-quality software, continuous feedback, and close collaboration with stakeholders. By breaking down the development into small iterations, maintaining flexibility, and prioritizing communication, the XP methodology ensures that the system meets the needs of both users and admins. Regular releases, automated testing, and customer involvement throughout the development process will result in a robust, user-friendly system that is adaptable to future changes and requirements. This approach ensures that the Hotel Reservation System is delivered on time, is of high quality, and meets the expectations of hotel managers, staff, and users.