

IT314 Project

Guesthouse booking system

Software Requirements Specification (SRS):

Introduction

Functional and Non-functional requirements

- **Functional requirements:**

1. **User Management:** Admins should be able to manage user accounts, reset passwords, and apply for premium membership.
2. **Hotel/Guesthouse/Property Management:** Property owners should be able to register their property for booking/renting, with the admin able to verify documents and licenses.
3. **View hotel:** The system should filter the property/hotel list based on user filters.
4. **Room Management/service:** Admins should be able to add, edit, and delete rooms and vary room availability and pricing for different dates.
5. **Reservation Management:** Users should be able to apply for waiting lists, make bookings, verify documents, view, approve, and reject bookings, and receive email confirmations.
6. **Cancellation Management:** Cancellation should only be allowed if the user satisfies the hotel/guesthouse policy.
7. **Payment processing:** The system should support multiple payment methods, securely enter and store payment information, and allow admins to view and track payments.
8. **User Support:** Users should be able to file complaints, give feedback, and view and respond to customer support requests.

- **Non-functional requirements:**

1. **Performance:** The system should be quick, responsive, and able to handle high volumes of concurrent users and transactions.
2. **Scalability:** The system should be scalable and handle a large database of users and property details.
3. **Availability:** The system should have a high availability rate, minimize downtime, and have robust disaster recovery mechanisms.
4. **Security:** The system should have strong security measures to protect sensitive information and comply with data protection regulations.
5. **Usability:** The system should be user-friendly, intuitive, and accessible for those with disabilities.
6. **Interoperability:** The system should be compatible with a range of devices and browsers and integrate with other systems.
7. **Reliability:** The system should have a high level of reliability with minimal errors and bugs.
8. **Maintainability:** The system should be easily maintainable and have the capability to be improved over time.