

SRS - Software Requirements Specification

1) Hotel management system

* Problem Statement:

The currently used hotel management is inefficient, it may have double bookings, poor coordination between the waiter & the chef, no online bookings. This may create customer dissatisfaction. So we need a ^{centralized} proper and efficient ~~hotel~~ ~~man~~ system to perform ~~meet~~ these all operations.

~~Read~~ SRS document:

Introduction: It is a tool to book hotel rooms through online.

* Keepers of this document:

The main purpose of this document is to provide foundation of the project. It gives the flow of the ~~the~~ like 'How the system works' & 'what are the user expectations?'

* Scope of the document:

The scope of hotel management is evolved now-a-days. It's not only about booking rooms in hotels or ordering foods, but it has gone deep into tourism, airlines, clubs, catering etc.

* Overview:

It is intended to booking of rooms through an online platform

- Receptionist - Access to update booking details
- Manager - Able to view financial reports
- Customer - Able to check room availability

* General description:

- users: customers, receptionists, Hotel manager, staff
- functions: Booking, Billing, staff allocation, reports, feedback
- System environment: web app + database + payment gateway

* Functional requirements:

- Customer: Register/Login, view rooms, book/cancel, pay online, feedback
- Receptionist: check in/out, manage reservations, generate invoices
- Manager: view reports, manage staff, monitor occupancy
- staff: view tasks, update task status

* Interface requirements:

- UI: simple, responsive dashboards
- Hardware: PC / laptop / tablet with internet
- Software: MySQL DB, Node.js backend, payment gateway integration

* Performance requirements:

- Data of user should be updated immediately after he/she registers
- Query should be solved within ^{5 minutes} ~~no time~~
- Login should be done in 3 seconds
- 24/7 availability

* Design constraints:

- Role based access control
- Budget limitation on the high end servers
- payment must be securely received
- data privacy

* Non functional attributes:

- Reliability - Able to satisfy all operations to meet end user requirements
- Portability - Should be available on all windows based environments
- Security - users data privacy, secure payments
- Maintainability - It should be maintainable
- Flexibility - Flexible to add features & handle them
- Availability - Available on specific date & time

* Preliminary Schedule & Budget

- Requirements analysis - 2 weeks
 - Design - 3 weeks
 - Development - 6 weeks
 - Testing - 2 weeks
 - Deployment - 1 week
 - Budget estimate : ~~approx~~ 8-10 lacs
- 