



Centre for Development of Advanced Computing, Bangalore

Software Requirements Specification

for

Happy Stay

Prepared by

KUNAL SIVACH - 220351920042

AKASH KENDRE - 220350120045

KARAN KHAMKAR - 220351920042

SHUBHAM DONGARE - 220350120030

SAGAR MALIK - 22035132008

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	
1	
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 Assumptions and Dependencies	
3	
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. website Features	4
4.1 Description and Priority	4
4.2 Stimulus/Response Sequence	
4.3 Functional Requirements	
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	
5	
6. Other Requirements	5

Appendix A: Glossary	5
Appendix B: Analysis Models	5

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This software is aimed at facilitating the easy management of rooms for a Hotel named Happy Stay . It will be helpful to the admin and to the customers for service and management.

1.2 Document Conventions

This document uses the following conventions.

DB	Database
ER	Entity Relationship

1.3 Intended Audience and Reading Suggestions

This document will capture all stakeholders' preferences, different conflicts and their resolution. Also, it could be used by potential developers, design engineers, testers, project managers, etc. Eventually this document can be used while preparing user documentation. This document will be proposed to different stakeholders for their approval and can be used as a reference guide in different phases of website development.

1.4 Product Scope

The project, Happy Stay website is a web-based application that allows the resort Admin to handle all resort activities online. Project provides room booking and other necessary hotel management features. The website allows the admin to post available rooms, facility, offers in the website. Customers can view and book room online. Customers can send Enquiry. Other hotel services can also be viewed by the customers.

1.5 Reference

- <https://spring.io/guides>
- Catalogues

- IEEE Recommended Practices for SRS. ANSI / IEEE Std 830 – 1993

2. Overall Description

2.1 Product Perspective

The website considers customer and hotel manager perspectives. Happy Stay specifications – where customers are able to book the rooms and use website functions described in this document, while for the administrative use there is a separate environment, where accounting, logistics and all relevant operations run. Both websites are based on the common database. They are integrated with each other, meaning the relevant data is exchanged. The system “User Interface” is an element of Happy Stay and it represents the web-interface for customer to book rooms online according to their choice.

2.2 Product Functions

User:

- Sign Up – Add user details i.e.- name,email,password,mobile,address,IDproof
- SignIn – User can SignIn using email and password.
- Edit Profile – User can edit user details.
- Change Password – User can change password.
- SignOut – User can SignOut from application.

Customer:

- Book Room – customer can book room for specific days.
- View Rooms – customer can view rooms under various categories.
- Booking Details – customer can view his booking details and status of booking, booking history.
- Invoice – customer can view the Invoice details and take print.
- Enquire – customer can send enquiry with message.

Admin:

- Room category – admin can add/edit/delete room category.
- Facility – admin can add/edit/delete facilities to be provided.
- Rooms – admin can add/edit/delete rooms under room category with specific facilities.
- Booking – admin can view the booking request and can take appropriate action.
- Enquiry – admin can view the enquiry send by customers.
- User registration – admin can add new customer/admin.
- Search – admin can search booking/enquiry by id or between specific date intervals.
- Pages – admin can Edit the details on Contact us/ About us pages.

2.3 User Classes and Characteristics

User Characteristics

There are two user classes in Happy Stay: customer and hotel manager.

1. Customer

Customer is the main stakeholder of the software intensive website. Customer would be able to check available rooms. Customer can select and sort rooms by price. Enter personal and payment details, as well as confirm booking. Customer input is required for making successful reservation.

2. Hotel Manager

Hotel manager is the administrative stakeholder of the software intensive website. Hotel manager would be able to update the information about listed rooms, see reports and check the booking information.

2.4 Operating Environment

Operating environment for this website will be hosted virtually using any cloud service provider.

2.5 Design and Implementation Constraints

1. The website has dependency on the database and the hotel's existing traditional website.
2. Internet connection is a constraint for this website because website is available from cloud therefore customer needs to have good network connection to connect to our web interface.

2.6 Assumptions and Dependencies

1. The requirements for the user interface is developed under the assumption that user interface is a sub website of existing larger website for hotel reservation management, which keeps and processes information about existing, booked and free rooms at the hotel, including the data about room type, number of total rooms at hotel, time of check outs and other factors that influence the room availability at any given time.
2. The process and algorithm for calculating available rooms are assumed to be given prior.

3. External Interface Requirements

3.1 User Interfaces

GUI and Help Issues:

These screens will be user friendly and help will be available at all levels. Help can be any type like the images of the room available, its price and the category it belongs to.

User has to select the proper value from list wherever provided eg. picking the date from the given list.

3.2 Hardware Interfaces

Back-end Server Configuration

- Intel Pentium-IV
- 128 MB RAM
- 1 Raid Controller Card
- 32 bit Ethernet Controller (100 Base T)
- 8 x 2.0 GB Fast SCSI/2 with Raid
- 48 x CD ROM Drive

Front-end Client Configuration

- Intel Pentium-III @ 650 MHz
- 128 MB SD RAM
- 10 GB Hard Disk Drive
- Monitor

3.3 Software Interfaces

Following are the software used for the Happy Stay Website

Software	Description
Operating website	We have chosen Windows operating website for its best support and user friendliness.
Database	To save the user records, user post records we have chosen MySQL + database.
Java	To implement the project, we have chosen Java language for its interactive support.

3.4 Communications Interfaces

Communication between front end and back end will be done using rest API with HTTP request protocol and images and file will be saved through FTP servers.

4. Website Features

4.1 Description and Priority

The Room Booking website maintains information about types of rooms, available rooms, rate of each type of room, any offers on room in the hotel for a customer. It creates database of customer who has visited the website for enquiry and also of the customers who booked room by signing up. Also it has database of admin who is operating the website. Of course, this project has a high priority because of which more number of customers can visit the website for booking and inquiry process.

4.2 Stimulus / Response Sequences

- Website will create the post about room information.
- Post will create description of rooms, prices & images.
- Customer will book room as per his requirement.
- Doubts will clear with help menu.
- All the information about customers and bookings will be saved in database.

4.3 Functional Requirements

Other functional requirement as follows-

- ☐ Register and Sign-Up.
- ☐ Adding, editing user profile.

- ☐ Adding, editing room and discount posts.
- ☐ Editing admin profiles.

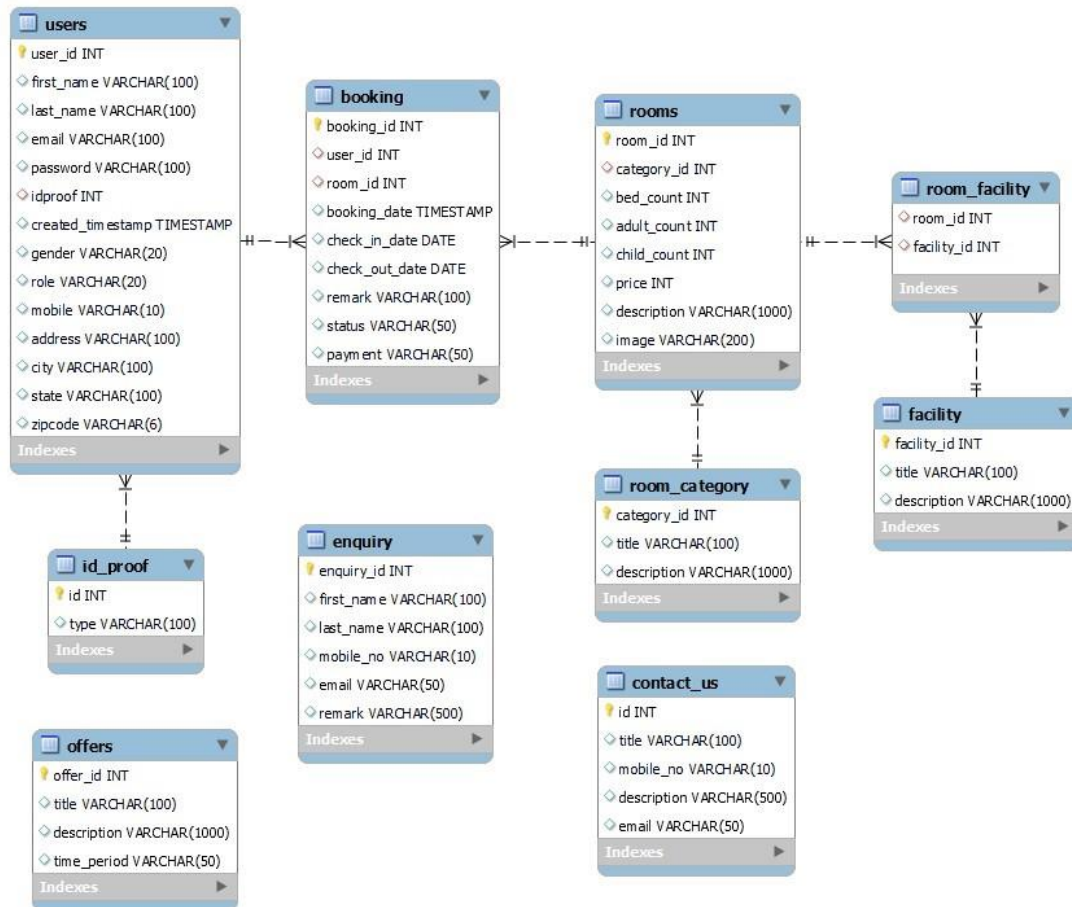
5. Other Nonfunctional Requirements

5.1 Performance Requirements

A) E-R Diagram

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- **ENTITIES:** This specifies distinct real-world items in an application.
- **PROPERTIES/ATTRIBUTES:** This specifies properties of an entity and relationships.
- **RELATIONSHIPS:** This connects entities and represents meaningful dependencies between them.



ER- Diagram

B) Normalization:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

5.3 Security Requirements

- Users of the website will have secure storage of data and their logins and passwords are safely stored.
- All the website users will have certain privileges, which vary according to their website role.

5.4 Software Quality Attributes

- AVAILABILITY: Weekly Backups and Monthly backups by Administration.
- CORRECTNESS: The availability of a room will be automatically updated once the room is booked for a specific date.
- MAINTAINABILITY: Regular trouble shooting is done and website is maintained to see that the functionality of the website is not disturbed at any instant of time.

Appendix A: Glossary

GUI: Graphical User Interface

HTML: Hypertext Markup Language

SRS: Software Requirements Specification

CSS: Cascading Style Sheets

HTTP: Hyper Text Transfer Protocol

FTP: File Transfer Protocol

Appendix B: Analysis Model

