Software Requirement Specification

for

Hotel Management System

Version 1.0

Prepared by Bhagya Shree

Bhavya Ahuja

Maninder Kaur Baria

Ravneet Singh Oberoi

Santhamohan Manivannan

Vivekanand Boopathy

MSCDA5510 Software Development in Business Environment

Table of Contents

Table of Contents 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 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 Assumptions and Dependencies 4

3. External Interface Requirements 5

3.1 User Interfaces 5

3.2 Hardware Interfaces 5

3.3 Software Interfaces 5

3.4 Communications Interfaces 5

4. System Features 6

5. Other Nonfunctional Requirements 9

5.1 Performance Requirements 9

5.2 Safety Requirements 9

5.3 Security Requirements 9

5.4 Software Quality Attributes 9

5.5 Business Rules 10

6. Other Requirement 10

6.1 Database Requirement 10

6.1.1 MongoDB disk and memory requirements 10

Disk requirements*:* 10

MongoDB cleanup routine 11

Memory requirements 11

6.2 Legal Requirements 12

# Introduction

## Purpose

The purpose of this document is to present a detailed description of the Online Hotel Management System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

## Document Conventions

This document follows MLA Format. Bold-faced text has been used to emphasize section and sub-section headings. Highlighting is to point out words in the glossary and italicized text is used to label and recognize diagrams.

## Intended Audience and Reading Suggestions

This document is intended for gathering requirements from the hotel owner and staffs. This document will be useful for Developers, Testers and Architects for designing the project’s design and flow.

Next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety but are intended for different audiences and thus use different language.

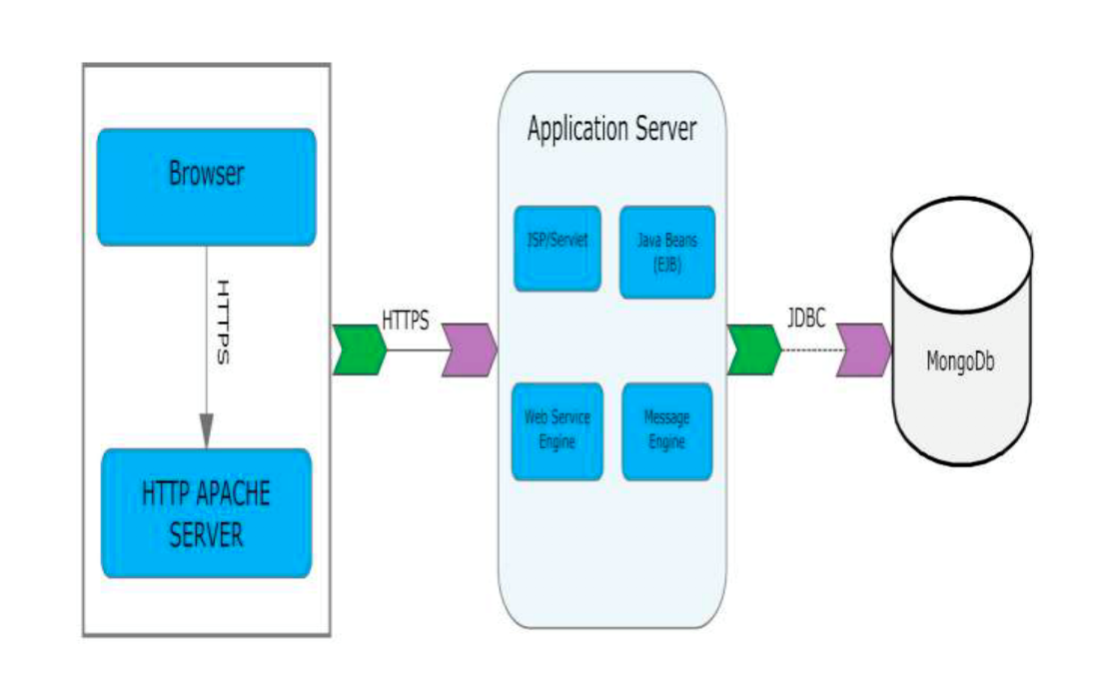
## Product Scope

This software is a web application which is designed for the efficient booking of hotel rooms for a reputed hotel chain. The web application is not only responsible for providing efficient hotel reservation system but also for providing appropriate discount coupons and offers to the customers as and when released by the management staff. The web application also updates the database in the real time regarding the latest reservation and the cancelation which are happening. The application eases the way hotel reservation are done, thereby helping the reservation department as well as making the entire reservation process much simpler for the customer.

# Overall Description

## Product Perspective

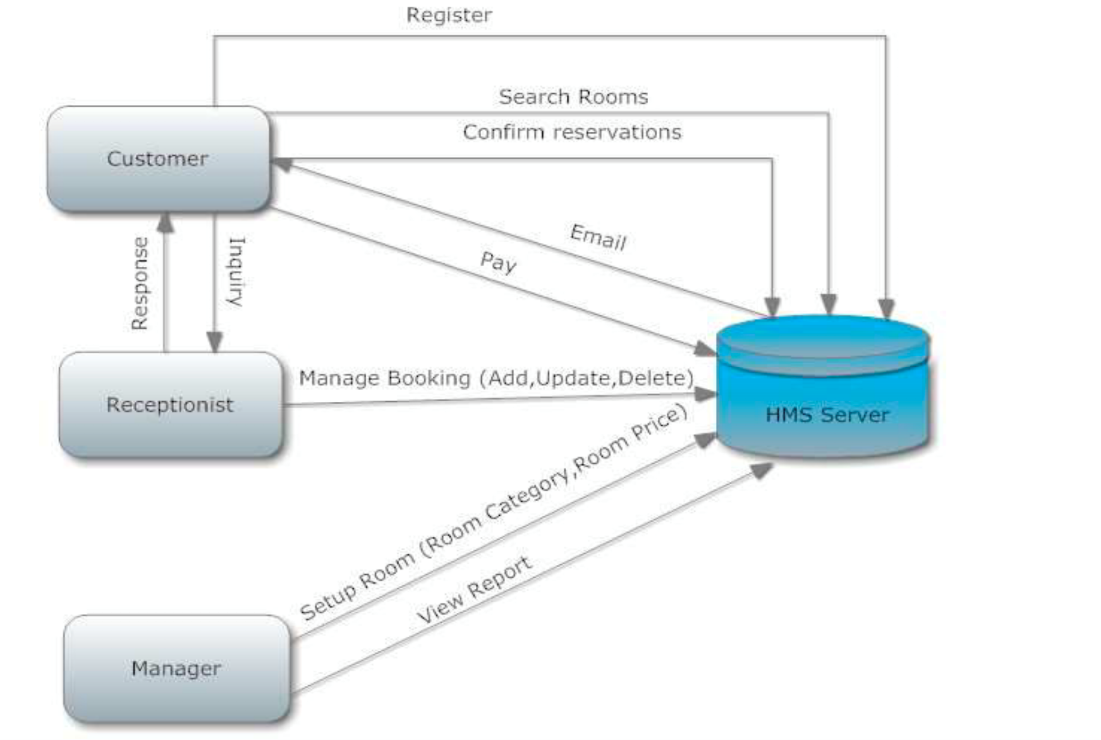
The Hotel Management System is a new self-contained software product which will be produced by the project team in order to overcome the problems that have occurred due to the current manual system. The newly introduced system will provide an easy access to the system and it will contain user friendly functions with attractive interfaces. The system will give better options for the problem of handling large scale of physical file system, for the errors occurring in calculations and all the other required tasks that has been specified by the client. The final outcome of this project will increase the efficiency of almost all the tasks done at the Hotel in a much convenient manner.



## 

## Product Functions

* Our Product General functions are:
* Customer Registration
* Check for Availability of Rooms
* Display the Rate
* Confirmation of Booking
* Email Notification
* Payment
* Set Room Details
* Manage Booking Details
* Generate Report
* Customer Service



## User Classes and Characteristics

There are 3 user Levels in our Hotel Management System:

A. Hotel Manager

B. Receptionist

C. Customers

**Hotel Manager**

* Manager have every access to the hotel system.
* Manager is solely responsible for managing hotel resources and staffs.
* Manager can view any report such as financial report, customer information, booking information, and room information, analyze them and take the decision accordingly.
* Manger is required to have experience on managing hotel previously, and have base knowledge of database and application server.

**Receptionist**

* Hotel Receptionist sole purpose is to provide the quality customer service.
* They have least access than manager.
* They can manage the booking details.
* They can search for availability of rooms, add the customer, confirm the booking, and update the booking details.
* Manager of hotel would probably want the receptionist who have good communication skills and command over English language.
* They should have basic IT Knowledge.

**Customer**

* Customer is vital part of the system. Customer has access to view the vacant room information and price range.
* They should be able to confirm the booking and cancel it if necessary.
* Customers have access to customer service desk portal to forward their inquiry.
* Customer should at least be capable to use the web UI interface.

## Operating Environment

Operating environment for the Hotel Management System is as listed below. Distributed Database Client/Server system Operating system: Windows/Mac Database: MongoDB Platform: Java, Java Scripts

## Design and Implementation Constraints

* Memory: System will have only 10GB space of data server.
* Language Requirement: Software must be only in English.
* Budget Constraint: Due to limited budget, HMS is intended to very simple and just for basic functionalities. UI is going to be very simple.

Implementation Constraint:

* Application should be based on Java only.
* Reliability Requirements: System should sync frequently to backup server in order to avoid the data loss during failure, so it can be recovered.

**2.5.1 Hardware**

* Operating System Supports all known operating systems, such as Windows,
* Linux
* Computer 512MB+ RAM, monitor with minimum resolution of 1024x768,
* keyboard, and mouse
* Hard Drive should be in NTFS file-system formatted with minimum 10 GB of free
* space
* A Laser printer will need to be used to print these reports and notes

**2.5.2 Software**

* Software is designed to run on any platform above Microsoft Windows 7 (32bit).

**2.5.3 High level Language**

* MongoDB (backend)
* JSP, HTML, CSS, JavaScript (front end)

## Assumptions and Dependencies

It is assumed that system developed will work perfectly that’s going to be developed under the Windows OS, and Apache Server with Mongo DB database. If incase of any difficulties, SRS should be flexible enough to change accordingly.

# External Interface Requirements

## User Interfaces

There are basically 3 types of users: Customer, Hotel Owners , 3rd party Owners.

For Customers, Website will be the interface where they can use the functionalities.

For Hotel Owners, the User Interface will be the Website and the Database where all the entries will be stored.

For 3rd Party Owners, the user interface will be their own application and the database where they will receive all the entries made by the user.

The Website is Compatible with any browser such as Internet Explorer, Mozilla, Chrome, Safari etc.

## Hardware Interfaces

For Customers: For the Website to run, you need a mobile device or a computing device with a web browser like chrome, Firefox, safari etc. and any operating system such as Windows, Mac, Linux etc.

For Hotel Owners: For all the maintenance in the website, they would be needing a computer system for all the backend operations to add or delete the offers and to maintain the room availability etc.

## Software Interfaces

It doesn’t need any specific software to run. The website would be able to run on any system with a well-run operating system.

For Hotel Owners: They would also need a database system to maintain their records as well as the records of their customers.

## Communications Interfaces

A good Internet connection is mandatory to run a website at both client side and server side. Communication between client and server should be served over HTTP. This is an interface which separated the server roles with that of the client roles. Apart from that it doesn’t have any unique communication interface.

# System Features

4.1 Functional Requirements

Functional requirements define the fundamental actions that system must perform. The functional requirements for the system are divided into the below main categories, Reservation/Booking, Food, Payment, Customer Support and Management. For further details, refer to the use cases.

1. **Reservation/Member Booking**
   1. The system shall provide information for reservations online or offline.
   2. The system shall record the customer’s details. ( Name ,Phone Number and Address)
   3. The system should allow users to login
   4. The system shall record the booking details.
   5. The system shall display the default room rate.
      1. The system shall display the special discount on the booking.
   6. The system shall display Booking confirmation whether or not the room is guaranteed.
   7. The system shall allow the customer to cancel the Booking.
   8. The system shall allow the customer to edit the Booking.
   9. The system shall check-in customers.
   10. The system shall record customer feedback.
   11. The system shall allow customers to redeem the loyalty points for booking.
   12. The system shall support Notification to be pushed once booking is confirmed.

**2.Food**

* 1. The system shall display the food menu on website, for users to view the cuisines available
  2. The system shall track all meals purchased in the hotel (restaurant and room service).
  3. The system shall record payment and payment type for meals.
  4. The system shall bill the current room if payment is not made at time of service.
  5. The system shall accept reservations for the restaurant and room service.

**3.Management**

* 1. The system shall display the hotel occupancy for a specified period of time (days; including past, present, and future dates).
  2. The system should allow customers to avail discounts
  3. The system shall display projected occupancy for a period of time (days).
  4. The system shall display room revenue for a specified period of time (days).
  5. The system shall display food revenue for a specified period of time (days).
  6. The system shall display an exception report, showing where default room and food prices have been overridden.
  7. The system shall allow for the addition of information, regarding rooms, rates, menu items, prices, and user profiles.
  8. The system shall allow for the deletion of information, regarding rooms, rates, menu items, prices, and user profiles.
  9. The system shall allow for the modification of information, regarding rooms, rates, Food menu items, prices, and user profiles.

**4.Payment Providers**

* 1. The Payment provider shall setup Payment API to allow online services
  2. The Payment API shall handle different tax rates.
  3. The payment info shall be recorded in data base so that the provider can pull the report.

**4.Customer Support**

* 1. The System shall implement online support system to assist the customers online.

# Other Non-functional Requirements

## Performance Requirements

* Any page of the application should not take more than 5 seconds to load on a DSL broadband connection.
* The system may be throttled or slowed down on heavy loads to ensure service for everybody. By throttling is meant that certain functionality may be unavailable during heavy server load.
* The application should be able to support 1000 concurrent users without any performance degradation.
* Although striving to have a 100% uptime, unless during a scheduled maintenance period, which will be relayed to the users of the site well in advance, problems may occur.
* Should be using the traffic metrics to increase the servers on demands and balance the load among the servers.

## Safety Requirements

* The hotel should also a well-documented and test continuity business plans that addresses all aspects of the hotel’s business.
* Both the data and the software have to backed up periodically.
* The data should be backed up daily andbe stored in a Storage service and a complete backup is needed before in cases of maintenance schedules.All backups that are greater than a year are pushed to cheaper storage such as Glacier service.

## Security Requirements

* There needs to be clearly defined roles of the users. These roles are 'customer’ and 'administrator'. Each person that goes to the system's website will be required to register if they want to do more than just read / browse site content.
* A secure server will be required to ensure confidentiality of customer’s credit card and other details
* Because of the different roles, passwords and user accounts must be implemented properly. It should be difficult to gain access to the site in an illegal manner.

## Software Quality Attributes

### Scalability Requirements

* The system should be able to scale up to 10000 concurrent users (if there is a need in the future) by installing additional hardware components.

### Availability Requirements

* The system has to be online 24 hours a day, 7 days a week. There is no place for an extended downtime, especially when the project goes International, and time zones will control the traffic load.
* The MTBF (if any) should not be less than 2 months.
* In case of a failure that leads to a system outrage, the MTTR should not be more than 2 hours.

## Business Rules

* Each customer can make one or more reservations.
* Each reservation can only be made by one and only one customer.
* Each room can only belong to zero or more room reservations.
* Each room reservation can belong to one and only one payment.
* Each payment can belong to one or more room reservations.
* Rooms cannot be assigned to more than one room reservation in the same time frame.
* Rooms can be cancelled only if Customer has at least one reservation.
* Change in Room booking can be done only if user at least one reservation.

# Other Requirement

# Database Requirement

# MongoDB disk and memory requirements

## Disk requirements*:*

The amount of disk space required by MongoDB depends entirely on the number of assets in the system. If the number of assets is stable, or if there is a steady influx / deletion rate in the archive so that the number of assets remains at a certain level, the instance will not grow.

Each asset remains approximately 10 days in MongoDB after deletion. Hence, the approximate formula for calculating the number of assets can be expressed as follows:

N = i \* (t + 10) + p

p: Number of assets that are permanently in the system

t: Max number of days an asset is kept in the system after ingestion

i: Number of assets ingested per day (24h)

N: Total number of assets in the system at any given time.

Each asset requires approximately 10Kb of space. The formula to calculate the disk space required for assets in MongoDB goes like this:

Disk space (Megabytes) = N \* 0,01

(N is the total number of assets, as explained above.)

All indexed files are pushed to MongoDB

When an Index Manager is configured to push data to server's MongoDB instance, all the indexed content from that Index Manager server is pushed to MongoDB, even if only some of the indexes are used as archives in Hotel Management System. For example, there may be a case where Hotel, hosts only a few archives from the Index Manager server while other indexes are used. In this case, data from all indexes is pushed to the MongoDB server, since Index Manager is considered a "slave" of the Hotel’s server.

## MongoDB cleanup routine

MongoDB does not free up disk space when data is deleted. However, already allocated disk space can be reused by new data. There are MongoDB admin commands that can be used to shrink and defragment the database files. However, Hotel Web does not automatically run such commands on MongoDB.

Hotel Web uses the “small files” option of MongoDB to ensure that disk space is allocated in small chunks rather than huge blocks with exponentially increasing sizes. Older versions of Hotel Web did not do this, so older installations may have large MongoDB data files which are mostly unused.

Important: Never delete the database files

MongoDB is NOT a cache - It is a database. Valuable data such as albums, CMS exports and (starting with Feature Release 4) users and groups are stored in MongoDB.

Deleting database files causes loss of data that can only be restored from a backup.

Storing the MongoDB database files in an alternative location

The location of the MongoDB data files is not configurable per se. However, you can create a file system junction point to place the folder elsewhere.

## Memory requirements

MongoDB requires approximately 1GB of RAM per 100.000 assets. If the system has to start swapping memory to disk, this will have a severely negative impact on performance, and should be avoided.

# Legal Requirements

Copyright laws and license agreements must be respected for any third-party software used in the creation of this system.