Software Requirements Specification

for

Milton Hotel Website

Version 1.0

Prepared by

Waleed Moursy

Parijat Bandyopadhyay

Vinay Govindan

Sreeraj Punnoli

Rishab Gupta

Simon Achkar

November 14, 2018

Table of Contents

[1. Introduction 1](#_Toc529997931)

[1.1 Purpose 1](#_Toc529997932)

[1.2 Document Conventions 1](#_Toc529997933)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc529997934)

[1.4 Product Scope 2](#_Toc529997935)

[1.5 References 2](#_Toc529997936)

[2. Overall Description 2](#_Toc529997937)

[2.1 Product Perspective 2](#_Toc529997938)

[2.2 Product Functions 3](#_Toc529997939)

[2.3 User Classes and Characteristics 3](#_Toc529997940)

[2.4 Operating Environment 3](#_Toc529997941)

[2.5 Design and Implementation Constraints 4](#_Toc529997942)

[2.6 User Documentation 4](#_Toc529997943)

[2.7 Assumptions and Dependencies 4](#_Toc529997944)

[3. External Interface Requirements 4](#_Toc529997945)

[3.1 User Interfaces 4](#_Toc529997946)

[3.2 Hardware Interfaces 5](#_Toc529997947)

[3.3 Software Interfaces 5](#_Toc529997948)

[3.4 Communications Interfaces 5](#_Toc529997949)

[4. System Features 5](#_Toc529997950)

[4.1 Payment API 5](#_Toc529997951)

[4.2 Payment Webpage 6](#_Toc529997952)

[4.3 Payment info to be stored in Database 6](#_Toc529997953)

[4.4 Payment details webpage 7](#_Toc529997954)

[4.5 User portal to cancel or edit booking 8](#_Toc529997955)

[4.6 API to show all payments on a date 8](#_Toc529997956)

[4.7 Email Notification 9](#_Toc529997957)

[4.8 Admin Dashboard 9](#_Toc529997958)

[4.9 Login using Social Media accounts 10](#_Toc529997959)

[4.10 Tax rates 10](#_Toc529997960)

[4.11 A/B testing 11](#_Toc529997961)

[4.12 Previous booking review 11](#_Toc529997962)

[5. Other Nonfunctional Requirements 11](#_Toc529997963)

[5.1 Performance Requirements 11](#_Toc529997964)

[5.2 Security Requirements 12](#_Toc529997965)

[5.3 Software Quality Attributes 12](#_Toc529997966)

[5.3.1 Availability 12](#_Toc529997967)

[5.3.2 Usability 12](#_Toc529997968)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Refer to cover page | 11/14/2018 | Initial version | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to give a detailed description of the requirements of the hotel website that the hotel’s customers can use to make bookings and access the services provided by the hotel. The system is divided into three subsystems:

* A payment API to process online payments.
* A portal to be used by the hotel’s manager and employees to input customer information as well as financial and administrative data.
* An end-point website to be accessed by the hotel’s customers so they can book a room.

## Document Conventions

The definition, abbreviations and acronyms used in this document:

* USER – A person who interacts with the website to make bookings and access the services provided by the hotel.
* PAYMENT PROVIDER – An organization which offers online services for accepting electronic payments by a variety of payment methods.
* ADMIN – System administrator who is given specific permission for managing and controlling the system
* API - Application Programming Interface
* REST - Representational State Transfer is a software architectural style that defines a set of constraints to be used for creating web services
* SMTP – Simple Mail Transfer Protocol used to send mails.
* JDK – JAVA Development Kit
* HTTP – Hyper Text Transfer Protocol used to transfer web content across the internet.
* FAQ – frequently asked questions

## Intended Audience and Reading Suggestions

This document is intended for the End Users (Milton hotel management and stuff), Marketing team and the Research and Development team which includes the Designers, Software Developers, Testers and Database Developers.

This SRS document contains description of functioning and requirements of the website in technical and non-technical terms which were decided after consulting various stakeholders of the system.

* It is important for the end users and developers to read Section 3 that specify External Interface Requirements which explains interactions between User, Software and the Communications Interface.
* Section 3 and Section 4 are equally important for the end users and developers to read. These sections describe the functional and non-functional requirements of the website respectively.
* It is important for the project managers and marketing managers to read Section 2, i.e. Overall Description which briefly talks about the product’s perspectives, functions, characteristics and its requirements.

## Product Scope

It is a website which enables the customer to make online booking of hotel rooms and meeting halls. The customers can check the room availability and the specifications of rooms which match their budget and need. Additionally, customers have the flexibility to modify or cancel the booking. In case of any queries or enquiries, the customer can use the 24/7 live chat for any kind assistance from trained personnel. The website furnishes its customers with the loyalty points scheme which they can redeem on diverse number of deals. The website will have special offer running at all times to attract more customers.

The hotel owner will have the admin role of the website to administer the system and keep the information accurate. The administrator can, for instance, verify the payments and manage user information.

Furthermore, the customer will have to login to the website with his/her credentials in order to make bookings and see customer specific offers. A payment API will be used to process the payments and rollbacks.

## References

* Gill (2018). Software Requirements Specification (SRS) for Online Tower Plotting S…. [online] Slideshare.net. Available at: https://www.slideshare.net/sukhpalsinghgill/software-requirements-specification-srs-for-online-tower-plotting-system-otps
* Cse.chalmers.se. (2018). [online] Available at: http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs\_example\_2010\_group2.pdf
* Aakashtechsupportdocs.readthedocs.io. (2018). Welcome to SRS for AakashTechSupport’s documentation! — SRS for AakashTechSupport 1.0.1 documentation. [online] Available at: https://aakashtechsupportdocs.readthedocs.io/en/latest/index.html

# Overall Description

## Product Perspective

The hotel website needs to communicate with a payment API to process the online payments for hotel room/hall bookings, modifications and cancellations. The API will be built and tested to integrate with the website. The API would be used to connect with the bank and to get the transaction status.

Owing to the data centric nature of the website, it will need to store the data somewhere. A database is used for that purpose. The database stores all the customer and product related information along with the API communications. A separate table will be maintained for the customers, so they have an option to view their payment status. All the details of the hotel’s events, room availability, etc. will be stored in different tables giving the user the flexibility to fetch the relevant information from the search bar using the filters on the website.

## Product Functions

With the website, the customers will be able to check the room/hall availability for specific dates. The specifications and details will be provided for every room. The website requires the customer to login with his/her credentials either using the sign-up page or using one of their social media accounts to book and make a payment for the room/hall. In case, the customer needs any assistance, he can use the 24/7 live chat feature. The website runs an algorithm to credit loyalty points to the customer's account based on the bookings they make, which they can later redeem to purchase some deals.

Customers can view the ongoing and scheduled events in the hotel and book the extra services that the hotel provides. To list some, the hotel airport shuttle, spa service and restaurant menu. Customers can additionally give feedback on the hotel feedback page and talk about their experience.

The hotel owner would have the admin rights to the website to manage the data and keep the information accurate.

## User Classes and Characteristics

There are three types of users that will interact with this website: the customers, the employees and the administrator. Each of these users have different use of the website so each of them has their own requirements.

* Customers - The customers will use this website to check the room availability and book the room online. They can additionally view and book the extra services provided by the hotel.
* Employees – The employees will use this website to update the hotel information like the events, room availability etc. and assist the customers with their payment or general queries. The employees can also access the public data of the customers like their booking history, loyalty points etc.
* Administrator – The administrator will manage the whole system so there is no inaccurate information. He can additionally give or take specific rights to some employees.

## Operating Environment

Customers and hotel management and stuff need hardware that meets the minimum requirements for commonly used browsers including internet explorer version 9 and above, Chrome version 58 and above. The hardware and software on the customer machine must meet the specific requirement for the version of the browser used.

The website will be hosted on a server that should be capable to support the expected traffic for the website. Modern hosts can scale up and down the assignment of hardware to support the traffic. Similarly, the database will be hosted on MySQL server that can support the traffic for the website. The host for the database can be the same host for the website or different host. There are no direct software or hardware requirements for the website to operate. The website is loaded through HTTP, so an active internet connection is required to access the website.

## Design and Implementation Constraints

It isn’t economically feasible to guarantee the design and implementation will be completable with all browsers available. The design and implementation will be executed to support the below browsers

* Internet explorer
* Chrome
* Firefox

Other browsers might be unable to utilize all functions of the website.

The design and implementation will support credit cards currently accepted by the Admin API Currently accepted credit cards are

* Visa
* Master card
* American express

## User Documentation

The website will have a dedicated “Help” page where the customers can find answers to some general queries. If the FAQs does not resolve the matter, the customer can use the 24/7 live chat for further assistance.

## Assumptions and Dependencies

The software will depend on the Bank API defined parameters and datatypes.

The software will assume the database will hosted on MySQL server.

# External Interface Requirements

## User Interfaces

Initial interface should be a login page where the customer can input the username and password to login or a link to register if he/she is a new user. New customer can either fill up a form to register or use the quick registration through social media. If a returning user enters invalid credentials, user would be prompted to enter the right credentials or to use the forgot password link to reset the password. If the user enters the right credentials, user will be redirected to his dash board where user has links to either to edit/make a booking, view previous bookings and payments or to view the current status of the bookings/payment.

The booking page allows the user to either book a room by browsing through the list of room types available or cancel an existing booking. After booking the room the bookings page allows the user to make the payment online and confirm the booking.

## Hardware Interfaces

The Website doesn't have any direct hardware interface.

## Software Interfaces

The website will interact with bank through API in addition to interface to MySQL server.

## Communications Interfaces

The website is compatible with browsers listed in section 2.5 and is loaded through HTTP. Upon payment/transaction the website uses a REST API call over the HTTP to the bank in order to process and approve the payment from the bank’s end. The bank in turn returns the status of the payment to the website via another REST API call making the payment process asynchronous. Upon successful transaction the website mails the details of the booking to the user using SMTP. The user can also signup using their preferred social media/mail which in turn invokes an API call to the corresponding site querying for the details required.

# System Features

## Payment API

4.1.1 Description and Priority

The customer must be capable of making a secure online payment via an API call to the bank.

4.1.2 Stimulus/Response Sequences

The user makes the payment by entering the payment details which is securely transferred to the bank via an API call. The bank validates and processes the payment and returns the transaction details as a response to another API call.

4.1.3 Functional Requirements

**REQ.4.1.1:** Transaction table: A table is to be created to track the details of the transaction

**REQ.4.1.2**: Payment API: Secure API to call to transfer the payment details to the bank to process.

**REQ.4.1.3:** Transaction details API call: Secure API to receive the Transaction status.

**REQ.4.1.4:** Transaction is to be tracked and closed if the response times out.

## Payment Webpage

4.2.1 Description and Priority

The customer must be capable of making a secure online payment by entering the bank details in the online payment form.

4.2.2 Stimulus/Response Sequences

The user makes the payment by entering the payment details which is securely transferred to the bank via an API call. The bank validates and processes the payment and returns the transaction details as a response to another API call. Upon a successful transaction the details are noted, and the payment is marked against the customer.

4.2.3 Functional Requirements

**REQ.4.2.1:** Payment History table: A table is to be created to track the details of the Payment made by the customer

**REQ.4.2.2:** Payment Form: A form to get the bank details like Payment mode, Bank name, card/account details and secure PIN.

**REQ.4.2.2.1:** Complete Form: The form should validate if the user has entered all the details required before making the payment.

**REQ.4.2.3:** Online Banking: If the customer selects the mode of payment to be online banking the form should prompt account number/User name and key.

**REQ.4.2.4:** Card Payment: If the customer selects the mode of payment to be credit/debit card the form should prompt card number, CVV code and the secure pin of the user.

**REQ.4.2.4.1:** Card Validation: The entered card number should be in decimal format and of 16 digits.

**REQ.4.2.4.2:** Expiry Dated: The cards expiry date should be greater than or equal to the current date.

**REQ.4.2.5:** Payment Details: The user should be given an option to add/save the preferred payment method and details.

**REQ.4.2.6:** Successful Payment: Upon successful transaction the payment must be marked against the user in the Payment History and Bookings Table and the user should be notified of the payment success and redirected to the website.

**REQ.4.2.7:** Payment Failure: The user should be notified with payment failure details and should be redirected to the website.

**REQ.4.2.8**: Email Notification: Upon successful payment the user should be mailed with the Payment and Booking details as an acknowledgment (handled by functional requirement 4.7)

## Payment info to be stored in Database

Create feature to store all the payment information in the database for the admin.

4.3.1 Description and Priority

Admin needs to see the report of all the payments. This feature enables to store the information in the database and thus formation of the report can be made with ease. This feature has high priority.

4.3.2 Stimulus/Response Sequences

When the customer does a payment, the information like username, transaction type, total amount, date of payment, description of payment, payment method etc. should be stored in the database.

4.3.3 Functional Requirements

**REQ.4.3.1:** Create database schema to store the payment info.

**REQ.4.3.2:** Define entities and relationships

**REQ.4.3.2.1:** Map entities and relationships with database

**REQ.4.3.3:** Consume API and store the payment details in the database

## Payment details webpage

Create a webpage to view the payment status for the customers

4.4.1 Description and Priority

Customer needs to see the payment status after each payment. This feature will help customer to identify whether the payment is successful or not. This feature has a moderate priority.

4.4.2 Stimulus/Response Sequences

When the customer completes a transaction, the page will be redirected to this payment status page and it shows the status of payment. Payment status page should show the status, details of the payment like room no, booking start date, booking end date, total price, date of booking etc.

4.4.3 Functional Requirements

**REQ4.4.1:** Design a login and signup page

**REQ4.4.2:** Validate username and password

**REQ4.4.3**: Create a forget password page

**REQ4.4.4:** Send email to reset password

**REQ4.4.5:** Create a payment page

**REQ4.4.6:** Create a payment status page

**REQ4.4.6.1:** Read the status from the database

**REQ4.4.7:** Add log out button

**REQ4.4.7.1:** Handle logout

## User Dashboard to cancel or edit booking

Create a webpage to cancel or edit the booking for the customer

4.5.1 Description and Priority

Customer needs to cancel or edit the booking done before the date. This will help customers to modify their booking and this will bring in flexibility to the customers. This feature has moderate priority.

4.5.2 Stimulus/Response Sequences

Each booking detail should be displayed in this page and there should be an option to edit or cancel the booking. Repayment back to the account should be incorporated with the Cancel option. Edit option helps to change the booked dates.

4.5.3 Functional Requirements

**REQ.4.5.1:** Create database schema to store the booking details

**REQ4.5.2:** Design a dashboard webpage to View the booking with options to Edit or Cancel the booking

**REQ4.5.3:** Create CRUD queries

**REQ4.5.4:** Provide connection with the database

**REQ4.5.5:** Link this page with menu page

## API to show all payments on a date

Create an API to get and show all the payments on a specific date for the admin.

4.6.1 Description and Priority

Admin needs to see all the payments received per date. This will help the admin to analyze the payments received per date. This feature has a moderate priority.

4.6.2 Stimulus/Response Sequences

If the admin wants to see the payment details for a date, he can use this API to get all the payment received per date by just inputting the date. API should return a list of all the payments and their details including the time, total price, discount details if any, username, etc.

4.6.3 Functional Requirements

**REQ.4.6.1:** Define the API call

**REQ.4.6.2:** Create a server to host that API

**REQ.4.6.3:** Create connection with the database and the server

## Email Notification

4.7.1 Description and Priority

Every time a customer makes a payment for the hotel, either to book a room or to buy a specific service. A notification must be sent to the customer’s emailing address informing them of the payment. ( Priority: 8)

4.7.2 Stimulus/Response Sequences

A customer makes a payment via the website, after finishing the checkout phase and entering all their payment information, a payment is processed. While the payment API deals with transferring the money from the customer’s account to the client’s account, an email notification must be sent to the customer’s email. A 3rd party mailing system will be used for this, the system will be responsible for processing the email. A copy of the email will be stored on the hotel’s database, this way the hotel can keep track of all the email they sent to their customers.

4.7.3 Functional Requirements

**REQ.4.7.1:** A new table must be created in the database in order to store all the emails sent.

**REQ.4.7.2:** An email template must be created in order to use it for all the emails, the template should contain the logo of the hotel, mailing address and contact information, the customer information, and the payment information.

**REQ.4.7.3:** 3rd party system for email processing. TBD later.

## Admin Dashboard

4.8.1 Description and Priority

An admin panel / dashboard for the hotel manager or employee with admin privileges. The user of this portal will see a list of all the customers and will have access to all of their public data including their booking history, their payment history, and their order history. This way the admin can help the customer more efficiently if a problem occurs. ( Priority: 6)

4.8.2 Stimulus/Response Sequences

* Admin login (with username and password).
* After authorization, admin will be redirected to the dashboard.
* The dashboard contains a list of the users, list of the most recent bookings and services.
* If the admin clicks on one of the users, they will be redirected to the user’s profile where they can see all the user’s public data including bookings and services provided.

4.8.3 Functional Requirements

**REQ.4.8.1:** A database table to hold information about the admin for authorization.

**REQ.4.8.2:** An interface for the dashboard, containing:

**REQ.4.8.3:** A menu where the admin chooses their required option.

**REQ.4.8.3.1:** Create interface to update Room Availability

**REQ.4.8.3.2:** Create chat-box to assist customers

**REQ.4.8.4:** Lists of all the users, all bookings, and all services ordered by the users.

**REQ.4.8.2.3:** A user’s profile containing all the user’s information.

## Login using Social Media accounts

4.9.1 Description and Priority

The hotel customer would like to use his social media account to log in into his/her hotel account. This enables the customer to reuse his social media username and password instead of creating new login credentials.

4.9.2 Stimulus/Response Sequences

* Customer opens the hotel website and navigate to the login page
* Customer selects the login using social media account
* Customer enters social media credentials.
* The website will verify the social media credentials are valid and store the user name in the database.

4.9.3 Functional Requirements

**REQ.4.9.1:** An API to receive credentials from social networks.

**REQ.4.9.2:** Parse API json.

**REQ.4.9.3:** Verify & Store credentials into database.

## Tax rates

4.10.1 Description and Priority

The website and Payment API shall be able to handle different tax rates for different customers based on Tax law.

4.10.2 Stimulus/Response Sequences

* Customer enters the details related to the Tax rate to be used (exempt, local tax rate)
* The API will apply the correct tax rate on top the booking cost and charge the correct total cost

4.10.3 Functional Requirements

**REQ.4.10.1:** Create database schema for tax rates.

**REQ.4.10.2:** Integrate taxes with payment API.

## A/B testing

4.11.1 Description and Priority

The website should support A/B testing in order to support hotel owner in deciding the best view for the website in generating revenue from ads.

4.11.2 Stimulus/Response Sequences

* Different hotel customer will be directed to one of the two versions of the hotel website.
* A script is tracking the most viewed version.
* Owner will decide the view to be permanently used.

4.11.3 Functional Requirements

**REQ.4.11.1:** Create multiple views for website.

**REQ.4.11.2:** Edit website to switch between views (ads/no ads).

**REQ.4.11.3:** Create a script to track the most viewed.

**REQ.4.11.4:** Create database schema for ads and designs.

## Previous booking review

4.11.1 Description and Priority

Hotel customer needs to be able to review previous booking including dates and payments

4.11.2 Stimulus/Response Sequences

* Customer login to the hotel account
* Customer selects previous bookings
* Website shows all details from previous bookings

4.11.3 Functional Requirements

**REQ.4.12.1:** Fetch payments with dates from database.

**REQ.4.12.2:** List details in a webpage.

**REQ.4.12.3:** Integrate webpage to dashboard.

# Other Nonfunctional Requirements

## Performance Requirements

The system must be interactive, and the delays involved must be minimum. There should not be any immediate delay in every action-response of the system. In case of opening windows forms, popping error messages, and saving the settings or sessions there should be a of delay much below 2 second. In case of accessing databases, sorting questions and evaluation there should be no delays and the operation should get executed in less than 2 seconds for opening, sorting, computing, and posting (> 95%) of the files. Also, when connecting to the server the delay should be based on the distance of the two systems and the configuration between them so that there should be a high probability that there will be (or will not be) a successful connection in less than 10 seconds for sake of good communication.

## Security Requirements

Information should be securely transmitted to server without any changes in information.

Users accounts must be secured from unauthorized access hence proper login mechanism should be used to avoid hacking. Login using OTP to the registered mobile number can be used to check spam for increasing the security. Also, security to be provided from unwanted use of recognition software. Proper encryption methodology should be incorporated, and system should prevent security leaks.

## Software Quality Attributes

### Availability

If the internet service gets disrupted while sending information to the server, the information can be sent again for verification.

### Usability

The system should be easy to handle and should navigate in the most expected way with no delays and the system program should react accordingly and transverse quickly between its states.