**BookWithMe**

**Online Rental Booking System**

#### By

**Prayag Desai**

**A project submitted**

**In**

**partial fulfillment of the requirements for the degree of**

**BACHELOR OF TECHNOLOGY**

**in Computer Engineering**

**Internal Guide**

*Prof.: Brijesh S. Bhatt.*

*Associate Professor*

Dept. of Comp. Engg.



**Faculty of Technology Department of Computer Engineering**

**Dharmsinh Desai University**

#### April 2019

CERTIFICATE

This is to certify that the project work titled

## BookWithMe

## Online Rental System

is the bonafide work of

Prayag Desai (Id no. 15CEUON030)

carried out in the partial fulfillment of the degree of Bachelor of Technology in Computer Engineering at Dharmsinh Desai University in the academic session

December 2018 to April 2019.

Brijesh S. Bhatt Dr. C. K. Bhensdadia

Assoc. Prof. Head,

Dept. of Computer Engg. Dept. of Computer Engg.



### Faculty of Technology Department of Computer Engineering

**Dharmsinh Desai University April 2019**

### Acknowledgements

I have taken efforts in this Project. However, it would not have been possible without the kind support and help of many individuals and organization. I would like to extend my sincere thanks to all of them.

I am highly indebted to Professor. **Brijesh S. Bhatt , Associate professor** and Professor. **Pandav Patel, Assistant Professor** of Department of Computer Engineering, Dharmsinh Desai institute of Technology Nadiad, for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express to my gratitude towards members of Dharamsinh Desai Institute of Technology, Nadiad for their kind co-operation and encouragement which helped me in completion of this project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| Chapter |  | Page |
| I. | **Introduction**  Purpose of the document  Glossary  References | 5  5  5 |
| II. | **About the System** | 6 |
|  | About BookWithMe  Purpose of the System  Scope of this System  Objective of the System  Tools and Technologies | 6  10  12 |
| III. | **Design and Analysis**  Workflow | 13  13 |
| IV. | **Implementation**  Code snippet with snapshots | 14  14 |
| V. | **Conclusion and Future Extension** | 18 |

**Chapter 1**

**Introduction**

## 1.1. Purpose

The purpose of this document is to present a detailed description of BookWithMe the Online Rental Booking 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 and will be proposed to the External and the internal faculty for its approval.

## 1.2. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| System / BookWithMe | System / BookWithMe refers to the online web platform for booking rental at specifies place and time by its registered users, those properties hosted by the registered hosts. |
| Registered User | Person who is a member of the service provided by the System for booking rentals which are hosted by the registered hosts. |
| Database | Collection of all the information monitored by this system. |
| Admin | Person who overlooks and manages the registered users, registered hosts and non-registered users, host properties, user and host credentials and bookings. |
| Registered Hosts | Person who is a member of service provided by the System to declare his/her property for rent for specific period of time for registered users to stay. |
| Non Registered Users | Person who is not an active member of the services provided by the System either for booking rentals or hosting one. The non registered users can only view and look for available properties in particular location at particular time. Non registered users can become either registered users, registered hosts or both. |
| Review | A written recommendation made by the registered users who have stayed in the property, about the host property, host and his / her experience in the property. |
| Reviewer | A person that has stayed in the property hosted by registered hosts. |
| Software Requirements Specification (SRS) | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |

## 1.3. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

**Chapter 2**

About the System

* 1. **About BookWithMe**

BookWithMe is an online platform for renting properties. BookWithMe helps homeowners to rent their spare rooms or entire property for about just days, weeks or months. ***BookWithMe*** will benefit homeowners to generate passive income on the their property which they no longer use. ***BookWithMe*** allows ***Non Registered Users*** to explore properties which are hosts by the ***Registered Hosts.*** The ***Registered Users*** can book properties for pre-specified period of time on location provided where ***BookWithMe*** has properties registered by ***Registered Hosts***. ***BookWithMe*** benefits the users as it is cheap compared to booking a motel and the service is excellent.

## Purpose of the System

The purpose of this System is to enable users to book properties which are made available by hosts through leveraging services provided by the System.

## Scope of the System

This software system will be an Online rental System. This system will be designed to minimize the renter’s tariff per rental booked as compared to the booking rates of motels and to maximize the earnings of the property owner who allow Registered Users to book their property for some time. The system does this by automating the management of property, bookings and credentials of hosts, users and properties which would otherwise have to be performed manually. The ***registered users*** are allowed to book the property made available by the ***registered hosts***, manage bookings (viewing booking details, deleting booking and managing bookings).The ***registered users*** can also manage their credentials, view property booked by them in the near past, make new bookings and view the specifications of the property booked by them or the property they intend to book. ***Registered users*** can also filter the property they search on the basis of hoe good a property is rated by others (filter properties by popularity), filter properties by certain class of property provided by the hosts (BookWithMe plus, entire home, boutique rooms and Unique homes.) and also filter property by the ascending order on the rates of rental per night. BookWithMe also implements a Review system which stores reviews and ratings given by the ***registered users*** to the property and the service of the ***registered host*** who host the property. This reviews made by the ***registered users*** are never deleted by the user and hosts. The review system is effective for honest feedback by the registered users. ***Registered hosts*** can make their property available for pre-specified period of time for the ***registered users*** to rent. ***Registered hosts*** can host property for booking, manage credentials, manage properties, extend/ delete bookings and make their properties unavailable for the renters. ***Non registered*** users can become registered users and registered hosts on successfully registering themselves for the same. ***Non registered users*** can only explore the properties available to rent but cannot book one. ***BookWithMe*** is a paid service. There is a one time fee for becoming registered users and registered hosts. The registered hosts have to pay pre-specified amount to the admin on every booking a registered user makes by using this platform. Also there is a booking cancellation fee for paid booking. This responsive application is made to run for both computers and mobile regarding any kind of operation system.

## Objective of the System

The objective of this System is to design and implement such a System which can run on multiple platform and browsers as the demand of online rental applications are certainly increasing. This software is made extensive user friendly and simple yet decent and powerful UI and feature rich which makes it very effective to use.

## Tools and Technologies used.

The System is implemented using Node.js and Express in the backend, html, CSS, JavaScript, jQuery and Bootstrap. No-SQL database MongoDB is database used by the System.

* + 1. **Why Node.js as backend?**

There are tons of alternatives to Node.js when building software but here are some reasons to use Node.js as a backend language.

* **JavaScript is probably the most popular programming language in the world**. It is the only language that can run inside a Web Browser, which gives it a unique point of advantage over any other language. With the rise of React Native, Electron and similar solutions, it’s also the only language which you can use to create web apps, frontend and backend, mobile apps, desktop apps. It’s pretty universal, and once you master the JavScript language in one area, your skills can be easily ported into another area and you just need to learn the environment differences (for example, how to use React Native rather than Electron, or the DOM).
* **Node.js is event driven and single threaded** What does this mean? In short, it can handle heavy load very well. Every single Node.js program can manage a lot of concurrent connections, and still be very fast because of its non-blocking I/O nature. This means that you have all the tools in place to build a very performant system, out of the box.
* **Node.js is hugely popular** This has a few side effects. First, you’ll find the solution to any kind of problem discussed online. **The community is huge** and very helpful. It’s easy to get started, and there are many resources to learn from. Second, there is a library for everything. The npm repository has a huge set of ready-to-use libraries, available with a simple npm install command. Thanks to this popularity, a vicious cycle is set up: your favorite API might only offer an official Node.js library, and other languages are not even considered, left for unofficial packages (with varying level of quality).
* **You can easily deploy Node.js apps** Some programming environments require a dedicated setup, even for simple projects, because no one has built an ecosystem around them.

* + 1. **Why Express with Node.js?**

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework −

* Allows to set up middleware to respond to HTTP Requests.
* Defines a routing table which is used to perform different actions based on HTTP Method and URL.
* Allows to dynamically render HTML Pages based on passing arguments to templates.
  + 1. **Why Bootstrap in front-end?**
* **Saves time and is easy to use** Using Bootstrap, you can save a lot of time. You don’t have to spend time writing code, you can just use the Bootstrap predefined design templates and classes and put it exactly where they fit. That’s why it’s simple to use Bootstrap. If you have the basic working knowledge of HTML and CSS, you can start development with Bootstrap.
* **Customizable** An advantage of Bootstrap is that it offers many ways to be customized so that you can make it your own. You can pick and choose what is needed and toss what is not. Bootstrap can be adjusted exactly as you want it and as your project requires. This is simply accomplished using the Bootstrap customize page.
* **Great grid system** Creating page layouts needs a good grid. Bootstrap has this benefit: one of the best responsive, mobile grid system. It’s really easy to use and if you need to work through columns, then you’re in the right place using Bootstrap. Very handy when you want to hide some content based on screen size. Adding a class such as .visible-desktop to an element will make it visible only for desktop users. There are similar classes for tablets and phones
* **Consistency**
* **Responsiveness** The need to have a responsive website is very important. Creating mobile-ready websites is a breeze with Bootstrap thanks to the fluid grid layout that dynamically adjusts to the proper screen resolution. If you shift from a laptop to an iPad, you won’t have to worry over your work. Bootstrap adapts to the change in platforms.
* **Compatibility** Bootstrap is compatible with all modern browsers and Internet Explorer versions. Bootstrap is equipped with elements that are being considered the future of design itself. For example, both HTML5 and CSS3 are things that are going to be big in the future. Plugins like HTML5Shiv and Respond.js come as part of Bootstrap’s default template. These help in porting HTML5 elements into older non-HTML5 browsers.
  + 1. **Why MongoDB as Database?**

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. This tutorial will give you great understanding on MongoDB concepts needed to create and deploy a highly scalable and performance-oriented database.

* MongoDB provides aggregation feature to use it in an efficient manner. For batch processing of data and aggregation operations, MapReduce can be used. MapReduce is nothing but an associated implementation for processing and generating big data sets with the parallel, distributed algorithm on a cluster.
* **MongoDB uses BSON format** One of the key features of MongoDB is that it uses BSON format. BSON is a JSON-like storage format. BSON stands for Binary JSON which is a binary-encoded serialization of JSON-like documents that MongoDB uses when storing documents in collections. It adds support for data types like Date and binary that aren’t supported in JSON
* **MongoDB Ad hoc queries** MongoDB supports field, range queries, regular expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions. MongoDB is able to support ad hoc queries by indexing BSON documents and using a unique query language
* **MongoDB is Schema – Less** MongoDB is a schema-less database (written in C++) because of which is much more flexible than traditional database tables. The benefit is the lack of setup and the reduced friction with OOP. So, in order to save an object, you just have to serialize it to JSON and send it to MongoDB. There is no need for type mapping which removes an additional burden.
* **MongoDB Indexing** Indexes are created to improve the performance of searches. The good thing is that any field in a MongoDB document can be indexed with primary and secondary indices. It enables the database engine to efficiently resolve queries which make it one of the best key features of MongoDB.
  + 1. **Why EJS as templating engine?**

EJS is the most popular templating engine for Node.js Express apps. Other popular options include PUG and handlebars.

* Use plain JavaScript We love JavaScript. It's a totally friendly language. All templating languages grow to be Turing-complete. Just cut out the middle-man, and use JS!
* Fast development time Don't waste time and attention figuring out arcane new syntax because 'elegance' — or how to preprocess your data so it will actually render right.
* Simple syntax JavaScript code in simple, straightforward scriptlet tags. Just write JavaScript that emits the HTML you want, and get the job done!
* Speedy execution We all know how fast V8 and the other JavaScript runtimes have gotten. EJS caches the intermediate JS functions for fast execution.
* Easy debugging It's easy to debug EJS errors: your errors are plain JavaScript exceptions, with template line-numbers included.
* Active development EJS has a large community of active users, and the library is under active development. We're happy to answer your questions or give you help.

## Tools used for developing this System.

Active development EJS has a large community of active users, and the library is under active development. We're happy to answer your questions or give you help.

**Chapter 3**

Analysis

# Table of Contents

[Table of Contents i](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487619)

[List of Figures ii](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487620)

[1.0. Overall Description 4](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487627)

[1.1 System Environment 4](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487628)

[1.2 Functional Requirements Specification 5](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487629)

[1.2.1 Reader Use Case 5](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487630)

[Use case: Search Article 5](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487631)

[1.2.2 Author Use Case 6](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487632)

[Use case: Submit Article 6](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487633)

[1.2.3 Reviewer Use Case 7](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487634)

[Use case: Submit Review 7](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487635)

[1.2.4 Editor Use Cases 8](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487636)

[Use case: Update Author 8](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487637)

[Use case: Update Reviewer 9](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487638)

[Use case: Update Article 9](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487639)

[Use case: Receive Article 10](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487640)

[Use case: Assign Reviewer 11](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487641)

[Use case: Receive Review 11](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487642)

[Use case: Check Status 12](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487643)

[Use case: Send Response 12](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487644)

[Use case: Send Copyright 13](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487645)

[Use case: Remove Article 14](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487646)

[Use case: Publish Article 14](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487647)

[1.3 User Characteristics 15](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487648)

[1.4 Non-Functional Requirements 15](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487649)

[3.0. Requirements Specification 17](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487650)

[3.1 External Interface Requirements 17](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487651)

[3.2 Functional Requirements 17](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487652)

[3.2.1 Search Article 17](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487653)

[3.2.2 Communicate 18](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487654)

[3.2.3 Add Author 18](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487655)

[3.2.4 Add Reviewer 19](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487656)

[3.2.5 Update Person 19](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487657)

[3.2.6 Update Article Status 20](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487658)

[3.2.7 Enter Communication 20](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487659)

[3.2.8 Assign Reviewer 21](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487660)

[3.2.9 Check Status 21](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487661)

[3.2.10 Send Communication 22](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487662)

[3.2.11 Publish Article 22](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487663)

[3.2.12 Remove Article 23](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487664)

[3.3 Detailed Non-Functional Requirements 23](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487665)

[3.3.1 Logical Structure of the Data 23](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487666)

[3.3.2 Security 25](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487667)

[Index 26](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487668)

# List of Figures

[Figure 1 - System Environment 4](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487669)

[Figure 2 - Article Submission Process 6](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487670)

[Figure 3 - Editor Use Cases 8](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487671)

[Figure 4 - Logical Structure of the Article Manager Data 23](file:///C:\Users\Hp\Downloads\SRS%20Example%20(2).doc#_Toc77487672)

## 3.1 System Environment

Author

Reader

Editor

HS DB

Online Journal

Article Manager

Web Publishing System

Reviewer

BookWithMe an Online Rental System has four active actors and one cooperating system.

The Author, Reader, or Reviewer accesses the Online Journal through the Internet. Any Author or Reviewer communication with the system is through email. The Editor accesses the entire system directly. There is a link to the (existing) Historical Society.

## 3.2 Functional Requirements Specification

## Use Case Diagram

## Figure-2. Use case diagram for BookWithMe

## 

This section outlines the use cases for each of the active actors separately. The actors for this system are Registered Hosts, Registered Users, Admin and Non registered users.

### 2.2.1 Admin use case

#### Use case: Manage registered users, registered hosts, properties and bookings

**Brief Description**

The admin manages the registered users, registered hosts, properties hosted by the registered hosts and the bookings made by the registered users. The admin also manages the payments by registered hosts and registered users for becoming member and leveraging the service provide by the system, payments made by the registered user on every bookings and payments on cancellation of the bookings by the registered users.

**Additional Description**

Before this use case can be initiated, the Admin has full access the credentials of the registered users and registered hosts. Admin has also access to the payments made by both the registered users and registered hosts.