

Book with Me

Online Rental Booking System

By

Prayag Desai,15CEUON030

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

Department of Computer Engg.

Faculty of Technology

Internal Guide:

Prof. Pandav K.Patel

Department of Computer Engg.

Faculty of Technology



**Faculty of Technology Department of Computer Engineering
Dharmsinh Desai University
April 2019**

CERTIFICATE

This is to certify that the project work titled

Book with Me

Online Rental System

is the bonafide

work of

Prayag Desai -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.

Mr. Brijesh S. Bhatt
Assoc. Professor
Dept. of Computer Engg.

Dr. C.K. Bhensdadia
Head of Department
Dept. of Computer Engg.



**Department of Computer Engineering
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad-387001
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 Dharmsinh Desai Institute of Technology, Nadiad for their kind co-operation and encouragement which helped me in completion of this project.

TABLE OF CONTENTS

1	Introduction.....	7
1.1	Purpose.....	8
1.2	Glossary.....	8
2	About the System.....	11
2.1	About BookWithMe.....	12
2.2	Purpose of the System	12
2.3	Scope of the System	12
2.4	Objective of the System.....	13
2.5	Tools and Technologies used.....	13
2.5.1	Why Node.js as backend?	13
2.5.2	Why Express with Node.js?.....	15
2.5.3	Why Bootstrap in front-end?.....	15
2.5.4	Why MongoDB as Database?.....	16
2.5.5	Why EJS as templating engine?.....	17
2.6	Tools used for developing this System.....	18
3	Analysis.....	19
3.1	SYSTEM ANALYSIS	20
3.2	USE CASE DIAGRAM.....	27
3.3	ACTIVITY DIAGRAM.....	29
3.3.1	Activity Diagram for Registered Users	29
3.3.2	Activity Diagram for Non-Registered Users	30
3.3.3	Activity Diagram for Non-Registered Host.....	31
4	Design	32
4.1	SYSTEM ENVIRONMENT	33
4.2	Data Flow Diagram.....	34
4.3	Class Diagram.....	37
4.4	State Diagram.....	40
4.4.1	State diagram for booking property	40
4.4.2	State diagram for cancelling booking.....	41
4.5	Sequence Diagram	42
4.6	Component Diagram.....	44
4.7	Deployment Diagram	45
4.8	Database Design	46
1.1	Application Navigation	50
4.8.1	Navigation for Non-Registered Users	50
4.8.2	Navigation for Registered Users.....	50
4.8.3	Navigation for Registered Hosts.....	51
5	Implementation	53
5.1	PROJECT MANAGEMENT.....	54
5.1.1	Roles AND RESPONSIBILITIES.....	54
5.2	Code Snippets.....	54
6	Test case design and Snapshots	66
6.1	Test Cases.....	67
6.2	Screenshots.....	77
7	Conclusion and Future Extention	94
7.1	Conclusion.....	95
7.2	Discussion	95
7.2.1	Self-Analysis of Project Viabilities.....	95
7.2.2	Problem Encountered while development.....	95
7.2.3	Summary of Project Work.....	95

LIST OF TABLES

Table 1 User Table.....	46
Table 2 Host table.....	47
Table 3 HostProperty table.....	48
Table 4 Booking table.....	49
Table 5 Feedback Table	50
Table 6 Milestones and Deliverables	54
Table 7 Roles and responsibilities	54
Table 8 Test cases for non registered users	67
Table 9 Test Cases for registered users	72
Table 10 Test Cases for registered hosts	76

LIST OF FIGURES

Figure 1 Level 0 DFD	34
Figure 2 Level 1 DFD – host.....	35
Figure 3 Level 1 DFD – registered user	35
Figure 4 Level 2 DFD – registered user	36
Figure 5 Level 2 DFD - host.....	37
Figure 6 Class Diagram.....	38
Figure 7 E-R Diagram.....	39
Figure 8 State Chart Diagram – booking property	40
Figure 9 State Chart Diagram – cancel booking.....	41
Figure 10 Sequence Diagram – property management	42
Figure 11 Sequence Diagram – Login Process	43
Figure 12 Component Diagram	44
Figure 13 Deployment Diagram	45
Figure 14 MVC model	52
Figure 15 Home Page.....	77
Figure 16 User Login	77
Figure 17 User Signup	78
Figure 18 Host Login	78
Figure 19 Host Signup	79
Figure 20 Stripe payment for Registered Hosts	79
Figure 21 Book and Explore properties	80
Figure 22 Manage user credentials	80
Figure 23 Manage bookings.....	81
Figure 24 Previously booked User properties.....	81
Figure 25 Host Registration	82
Figure 26 Host Registration successfull.....	82
Figure 27 Host dashboard	83
Figure 28 Editing Host property	84
Figure 29 Property Details	85
Figure 30 Update host Credentials.....	86
Figure 31 Make property unavailable	86
Figure 32 List properties.....	87
Figure 33 Book property	87
Figure 34 Confirm booking	88
Figure 35 Make Payment for booking property.....	88
Figure 36 Booking Successful	89
Figure 37 User dashboard	89
Figure 38 Ratings and Review.....	90
Figure 39 Review and Booking	90
Figure 40 Making property unavailable	91
Figure 41 Deletion of paid booking	92

CHAPTER 1

INTRODUCTION

1.1 Problem definition

BookWithMe is an online rental booking system for booking of properties by the users rented by the host. The host rents his property for specific period of time for the user to rent. The system manages bookings, property, user and host credentials.

1.2 Glossary

Term	Definition
System / Book With Me	System / Book With Me 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 Motivation

The motivation for developing this system comes from the tedious way in which traditional bookings were made. And also, it becomes very difficult for frequent travelers for renting a hotel room. The tariff for hotel room is very high as compared to the property listed on this System. There are several individuals with some part of the home which the owner wish to rent out but not on contract basis to the renters. The owner wishes to rent a part of their property for some point of time. BookWithMe helps these kind of property owners to rent out their property for specific period of time decided by the host of the property. This system allows the host to create passive income and the renters to save money on hotel tariff.

1.4 Challenges

Major challenge encountered in implementing and designing this System is the management of bookings, especially the automatic deletion of the bookings made by the users. A fair review system was also a challenge for developing the system. The system is developed in such a way that even if the user and the property and also the property gets deleted, the Review on the property is stored and not deleted or altered.

1.5 Outline of the Report

The report consists of seven chapter which briefly discusses the planning and implementation of this project. Chapter-1 gives detailed description about the problem definition, the challenges faced for developing the project and the glossary for terms used throughout the document. Chapter-2 gives he insight on the purpose, scope and objective of this system. Chapter-3 discusses the System analysis In terms of use case and activity flow for the system. Chapter-4 discussed the design of the system in terms of data flow, object-oriented class diagram, sequence diagram, component diagram and deployment of the system in real environment. Chapter-5 gives a brief introduction and quick reference of the tools and technologies used for development of this system. Chapter-6 discusses about the est cases and snapshots of the system. Chapter-7 concludes this project and states the future extensions which needs to be implemented beyond the core functionalities of this system.

CHAPTER 2

ABOUT THE SYSTEM

2.1 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 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.

2.2 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.

2.3 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 how 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 onetime 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.

2.4 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.

2.5 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.

2.5.1 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 Java Script 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.

2.5.2 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.

2.5.3 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.

2.5.4 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.

2.5.5 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 script let 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.

2.6 Tools used for developing this System.

This application is made by using VSCode editor. VSCode is a very powerful editor when it comes to working with multiple technologies. VSCode support multiple plugins which makes the development of front-end and back-end easier by intellisense. Nodemon is the server which is a development dependency used for testing the code and maintaining server while testing the app while production. MongoDB Compass is used for database modelling and visualize and explore data with ad-hoc queries. Debugging is carried out by the Chrome Debugger Tools of the Chrome Web Browser.

CHAPTER 3 ANALYSIS

3.1 SYSTEM ANALYSIS

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

Requirement of the System

i. User Requirements

User requirements does not include many things, but most importantly user must be aware that system works properly with full availability, reliability, security and safety. The user responsibility are as follows:

Should know how to use the application and should adhere to the guidelines and prescribed standards.

ii. System Requirements

A. Functional Requirements

R1: Authentication for Registered Users

Input: Credentials

Description: System gets the credentials of the registered users and according to the credentials of the user system gets the user information from the database.

R1.1: System reads the username and password entered by the User

Input: Correct credentials

Output: User is logged in.

R1.1: System reads the username and password entered by the User

Input: Incorrect credentials

Output: User is not logged in.

R2: Authentication for Registered Hosts

Input: Credentials

Description: System gets the credentials of the registered hosts and according to the credentials of the hosts system gets the host information from the database.

R1.1: System reads the username and password entered by the Host

Input: Correct credentials

Output: User is registered as registered Host. The credentials are stored in the Host database

R1.1: System reads the username and password entered by the Host

Input: Incorrect credentials

Output: User is not registered.

R2: Registration for becoming Registered Hosts

Input: Form credentials entered by non-registered users and/or registered users.

Description: System gets the credentials of the non-registered users and/or registered users. The system checks if there are Registered Hosts already present in the database with the entered credentials

R1.1: If there is host with the same credentials entered

Input: Form credentials entered by non-registered users and/or registered users.

Output: User is asked to enter the form again with new credentials.

R1.1: If there is no host with the same credentials entered

Input: Form credentials entered by non-registered users and/or registered users.

Output: User is registered as Registered Host

R3: User can book and explore properties

Input: Form data entered by the user which contain fields for check-in-date, check-out-date, accommodation per rental and the name of city and state.

Description: Users can book and explore properties in specified city, state and book properties on number of users accommodated in the rental

R3.1 If there are no properties by user specification

Input: Form data entered by the user which contain fields for check-in-date, check-out-date, accommodation per rental and the name of city and state.

Output: No properties

R3.2 If there are properties demanded by the user

Input: Form data entered by the user which contain fields for check-in-date, check-out-date, accommodation per rental and the name of city and state

Output: List of properties demanded by the user along with the property details

R4: User can change their credentials

Input: User's updated credentials

Description: User can change their credentials and update their new credentials

Output: Confirmation of updation of user credentials.

R5. Registered users can delete booking

Input: Request for deleting bookings

Description: User can delete booking made by them earlier. Users have to pay for the cancellation of the paid property and the cancellation of free property is not charged by the system

R5.1 If the cancellation is paid

Input: User id, Payment id and cancellation scheme

Output: The booking is deleted on payment of cancellation fee by the User.

R5.2 If the cancellation is free

Input: User id, Payment id and cancellation scheme

Output: The booking is deleted.

R6. Registered users can view booking details.

Input: Request for details on user's bookings

Description: User can view booking details for the booked properties. Booking details consists of booking transaction id, property details which is booked by the user and when the property is booked by the user

Output: Booking Details.

R7. Registered Users can view properties booked by them earlier

Input: Request for viewing previously booked properties

Description: Registered Users can view properties booked by them earlier

Output: Properties booked by the user earlier. The properties has details on whether the property was successfully vacated by the user or the property was cancelled by the user

R8. Registered Users can view properties booked by them earlier

Input: Request for viewing previously booked properties

Description: Registered Users can view properties booked by them earlier

Output: Properties booked by the user earlier. The properties has details on whether the property was successfully vacated by the user or the property was cancelled by the user

R9. Registered Hosts can host several properties for the Users

Input: property information

Output: Status of the insertion of the property

Description: The host has to give the information about the property at the time of the creation of the property

R10. Registered Hosts can make the property unavailable or the Host to book.

Input: property-id to delete.

Description: The host can make the properties unavailable for the user to book the property. The System checks the status of the property

R10.1 If the property is already occupied by the renters.

Input: property-id to delete

Description: The Host cannot delete the properties that are already occupied by the renters.

Output: Host is not allowed to delete the property.

R10.2 If the property is not occupied by the renters

Input: property-id to delete

Description: The Host can delete the properties that are already occupied by the renters.

Output: Host is allowed to delete the property.

R10.3 Registered Hosts can update the property information

Input: property-id to update

Description: The registered hosts can change the information about the properties.

R10.4 Registered Hosts can only update selected fields when the property is already booked by the other renters.

Input: property-id to update

Description: Registered Hosts cannot update property information such as the rate, cancellation scheme, name and address of the property. The Host cannot update the check-in dates and check-out-date but the Host can extend the date of availability of the property.

Output: Status of updation of the property.

R10.5 Registered Hosts can update all the property information for the property which are not booked by the users

Input: property-id to update

Output: Status of updation of the property details.

B. Non Functional Requirements

- Scalability**

The system should able to scale up and scale down as per the traffic and need to optimize the operational cost.

- Reliability and fault tolerance**

1. The system should up and running all the time with less to zero downtime.
2. The data should persist even after a catastrophic failure occurs.

- Security**

User's private data and system's data should not be leaked or accessed by an unauthorized person.

- **Usability**

Be easy for users to understand and use the product.

- **Performance**

Be able to withstand the changing load and being fast and responsive all the time for users.

3.2 USE CASE DIAGRAM

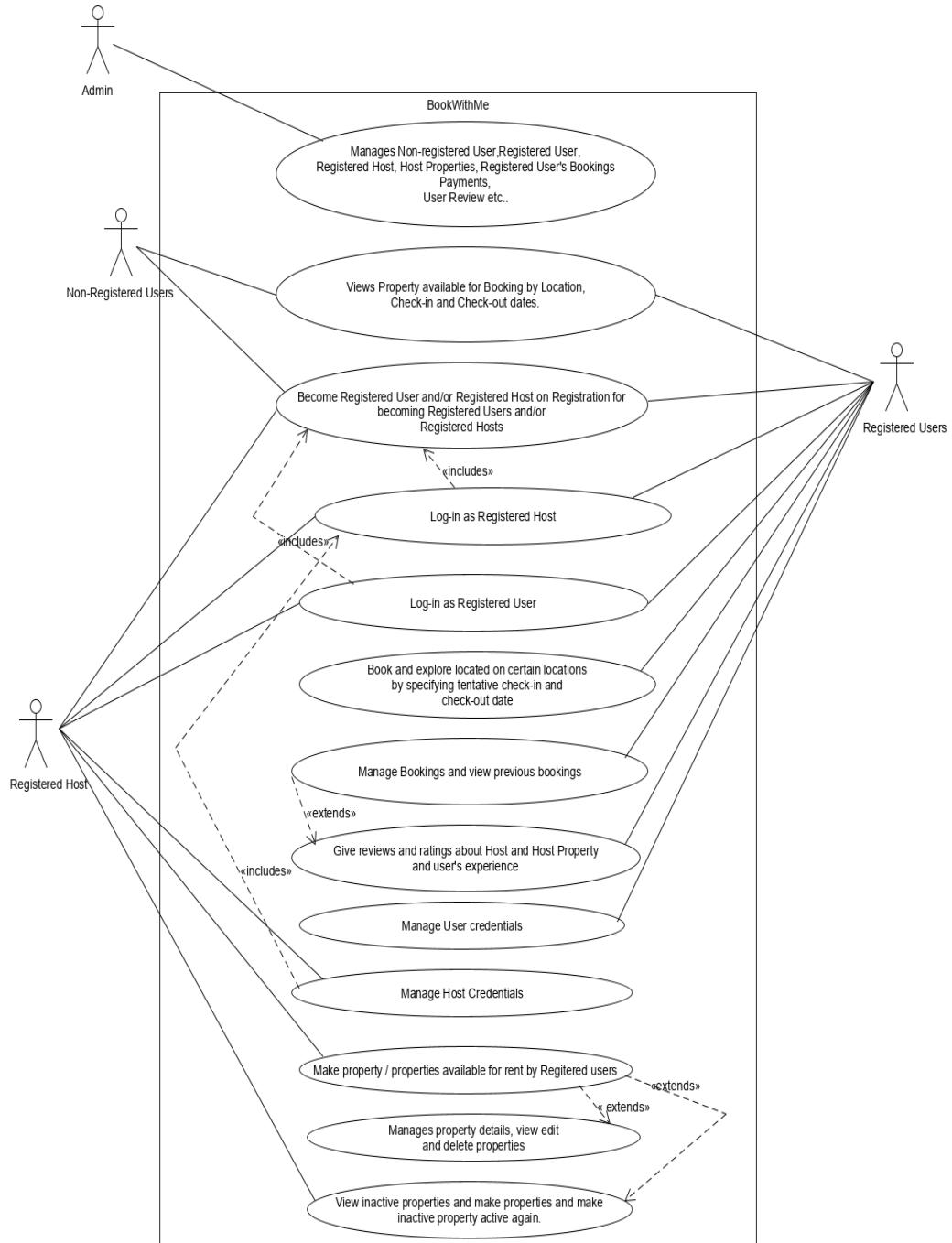


Figure 1 Use Case Diagram

this system is Registered Hosts, Registered Users, Admin and Non registered users.

- Actor

There are 4 actors interacting with this System Admin, Registered Users, Non-Registered User and Registered-Host

- Use cases

1. Admin

- Admin manages bookings, properties, review and ratings, payments, registered users, registered hosts and their credentials.

2. Registered Users

- Registered Users can change their credentials.
- Registered can book and explore properties by location and by location and tentative check-in and check-out dates and accommodation strength per rental
- Registered Users can filter properties by property types and also filter property according to cheapest to costliest rate of rental.
- Registered users can manage Bookings.
- Registered users can view booking details.
- Registered Users can delete the bookings. Registered users have to pay for the property that has paid cancellation and cancel property without payment for property that has free cancellation scheme
- Registered users can view the properties that he/she has deleted or the properties that have been booked by the user previously.
- Registered Users can also become Registered Host by signing up for Registered Host and rent his/her property.
- Registered Users can make rating and give reviews for the properties that User has stated. Users cannot rate or review properties that the user has booked but not stayed still.

3. Non-registered Users

- Non registered users can view properties across various locations based on the check-in date, check-out dates and accommodation strength per rental.
- Non registered users can become Registered Users upon registration.
- Non registered users can become Registered Hosts upon registration.

3.3 ACTIVITY DIAGRAM

3.3.1 Activity Diagram for Registered Users

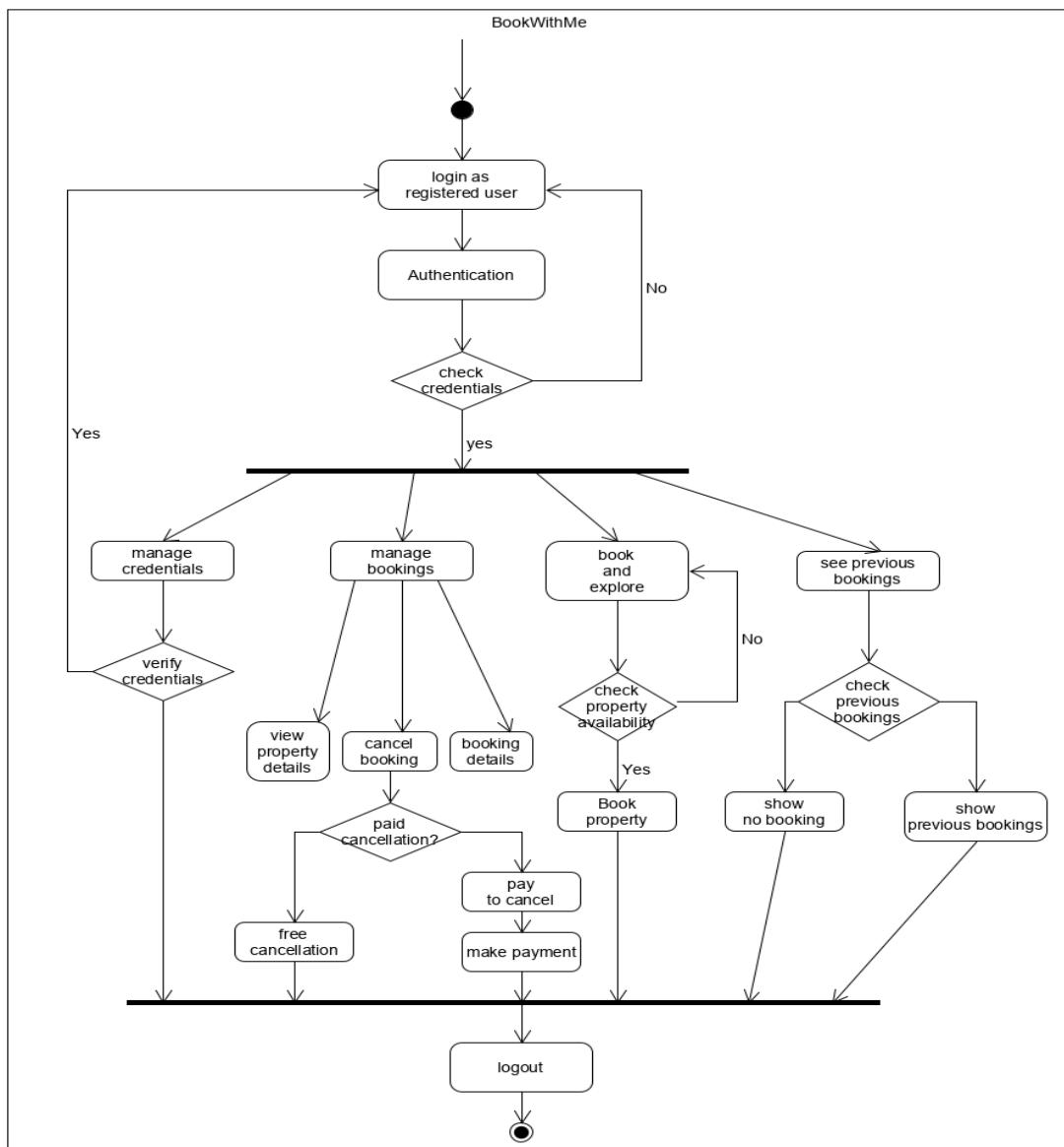


Figure 2 Activity Diagram for registered user

Activity diagram is another important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

3.3.2 Activity Diagram for Non-Registered Users

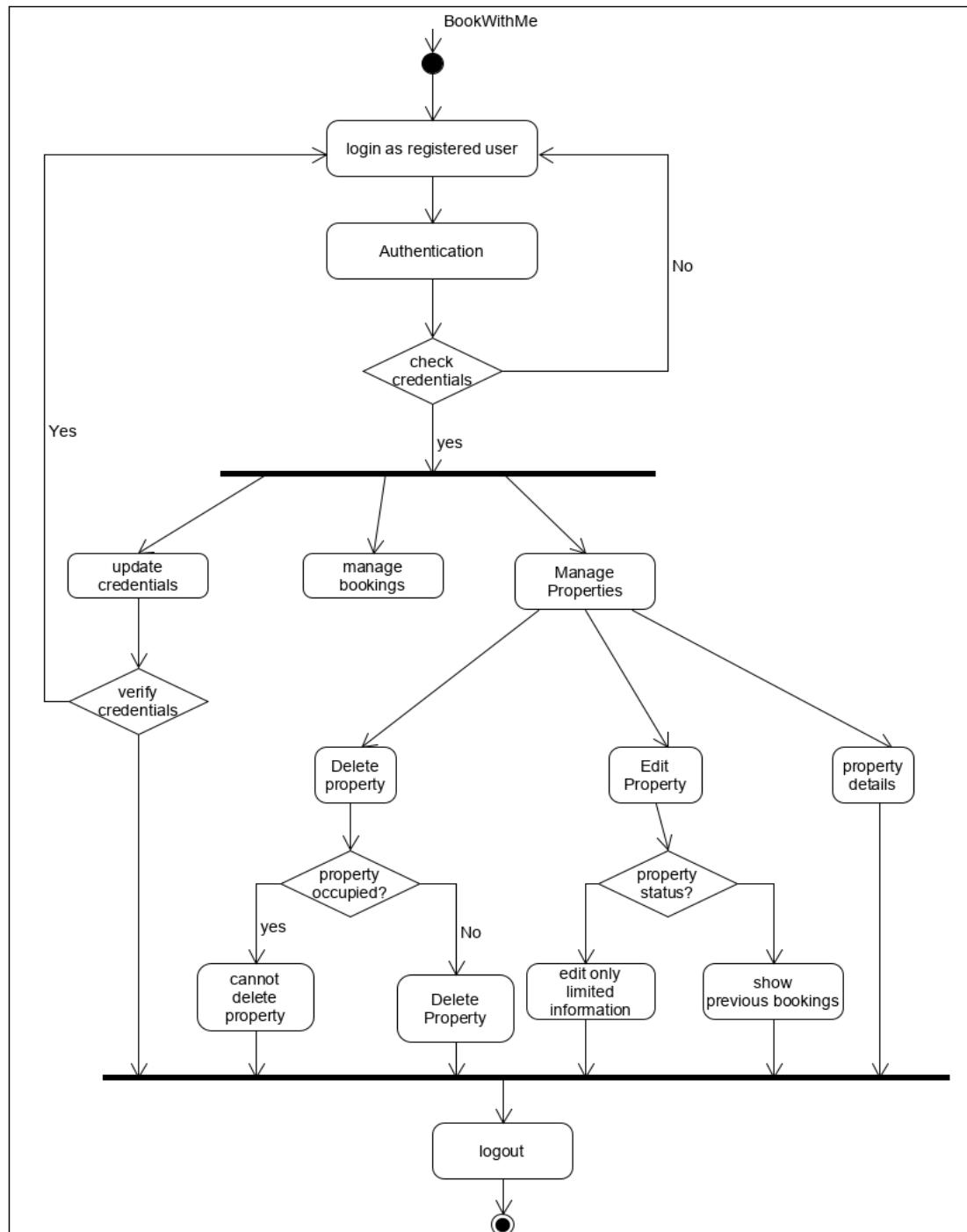


Figure 3 Activity Diagram for Non-Registered Users

3.3.3 Activity Diagram for Non-Registered Host

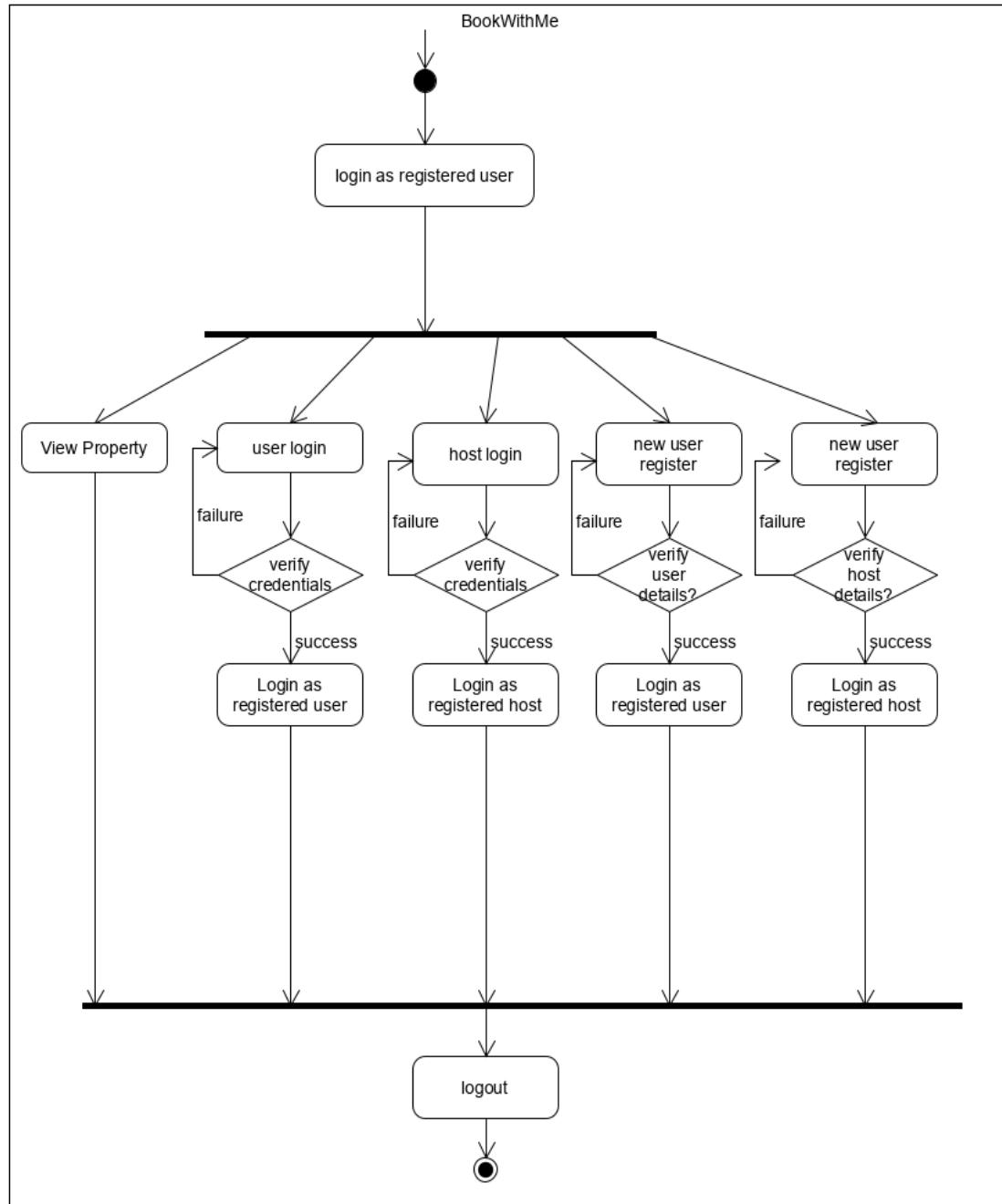
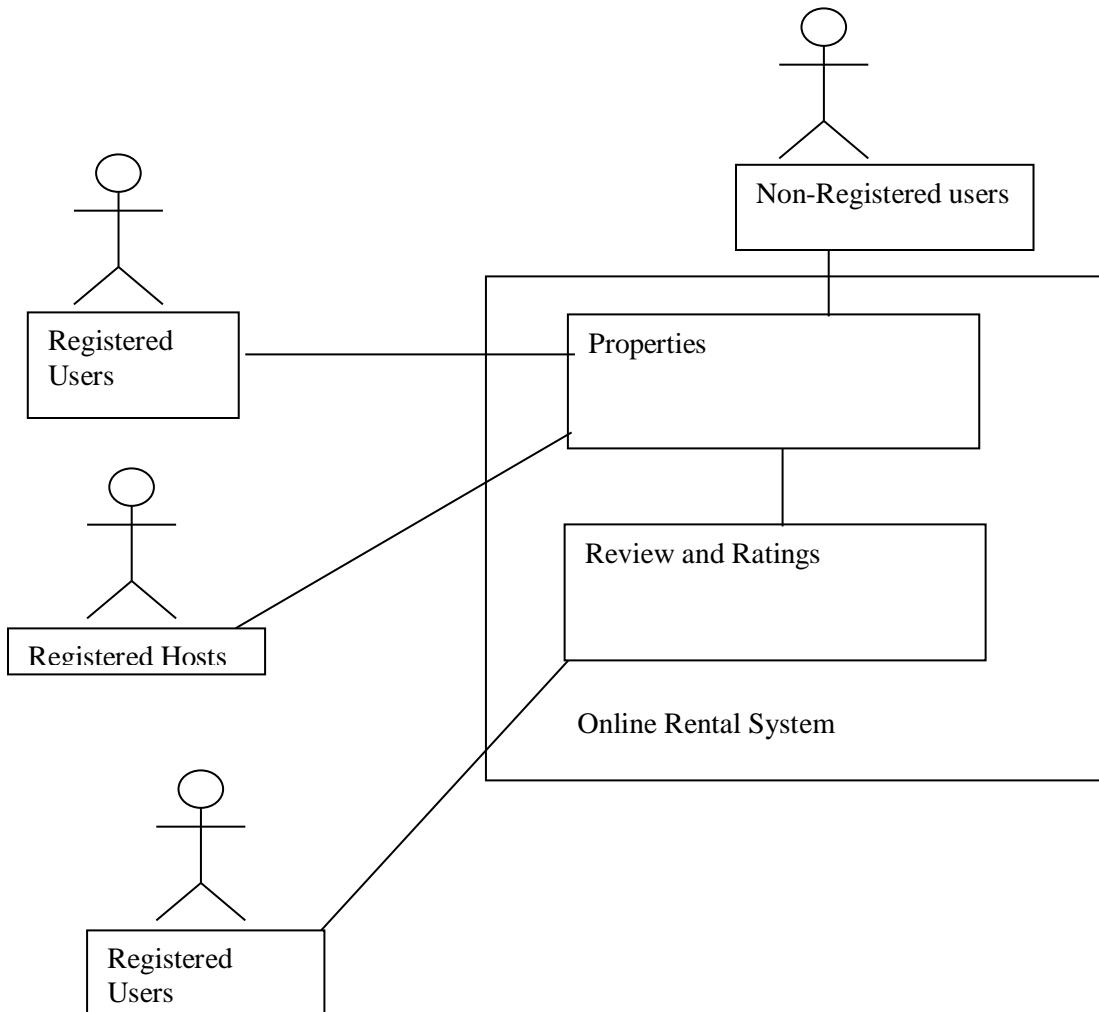


Figure 4 Activity Diagram for Non-Registered Host

CHAPTER 4

DESIGN

4.1 SYSTEM ENVIRONMENT



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.

4.2 Data Flow Diagram

A **data-flow diagram** (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

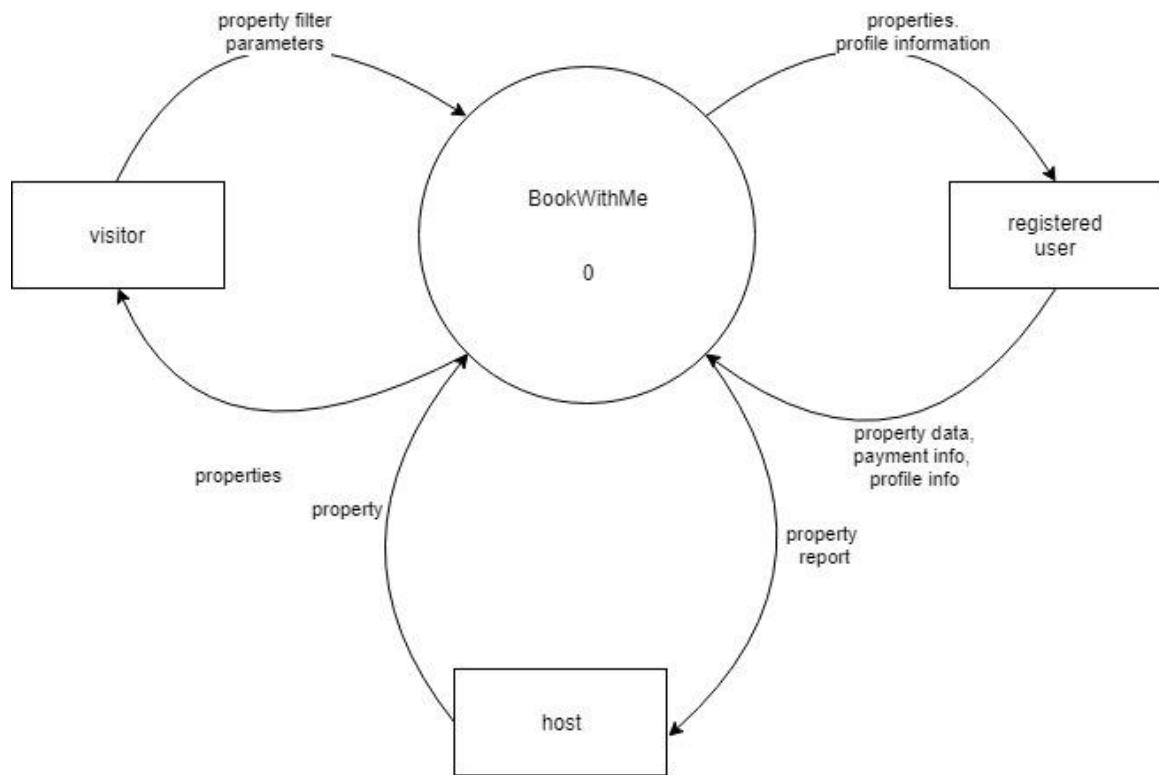


Figure 5 Level 0 DFD

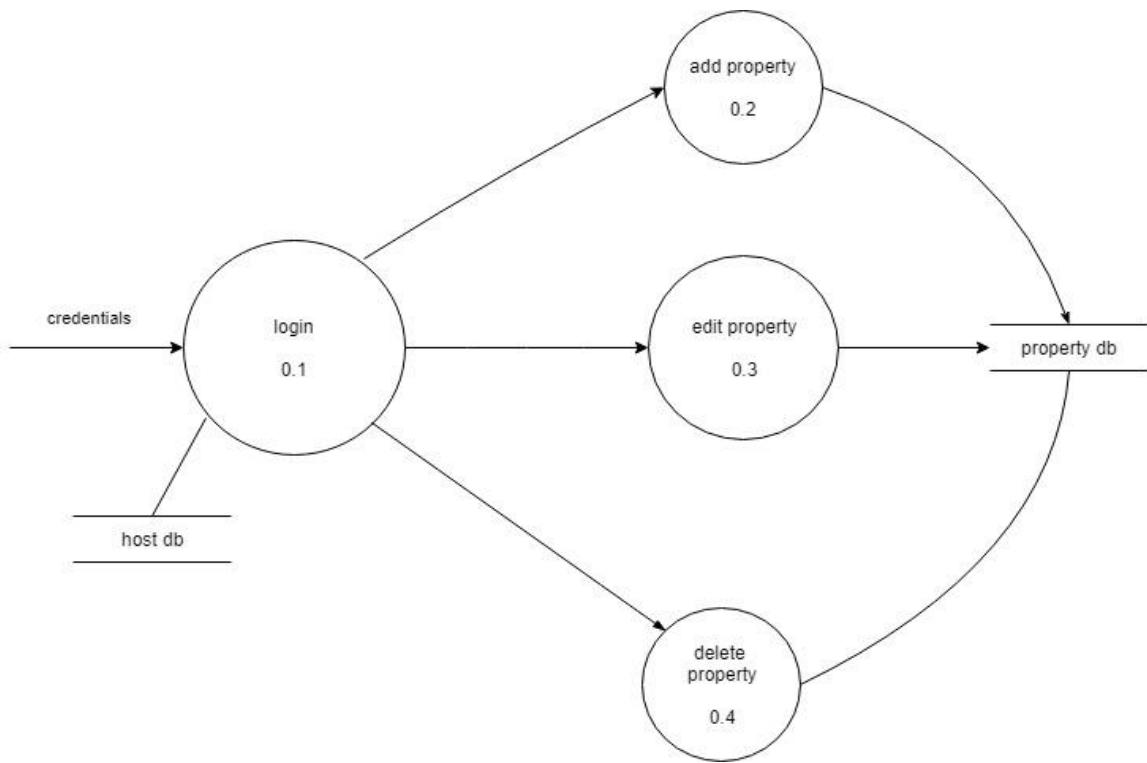


Figure 6 Level 1 DFD – host

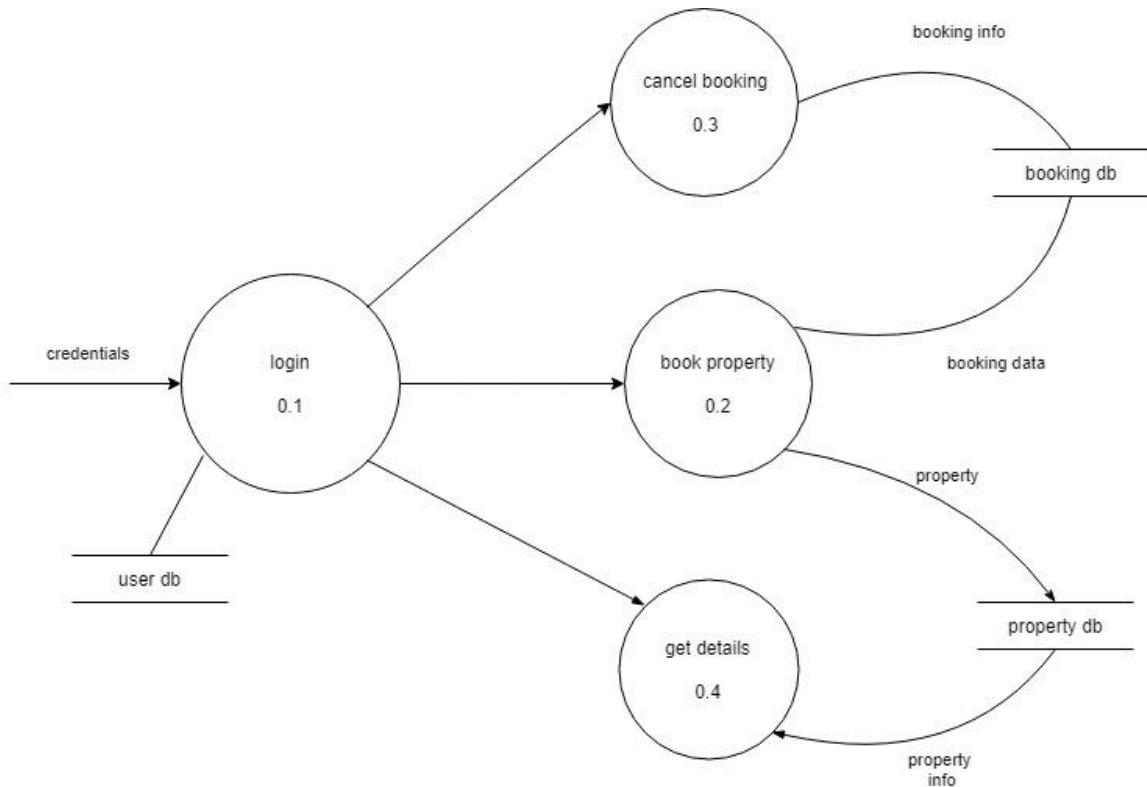


Figure 7 Level 1 DFD – registered user

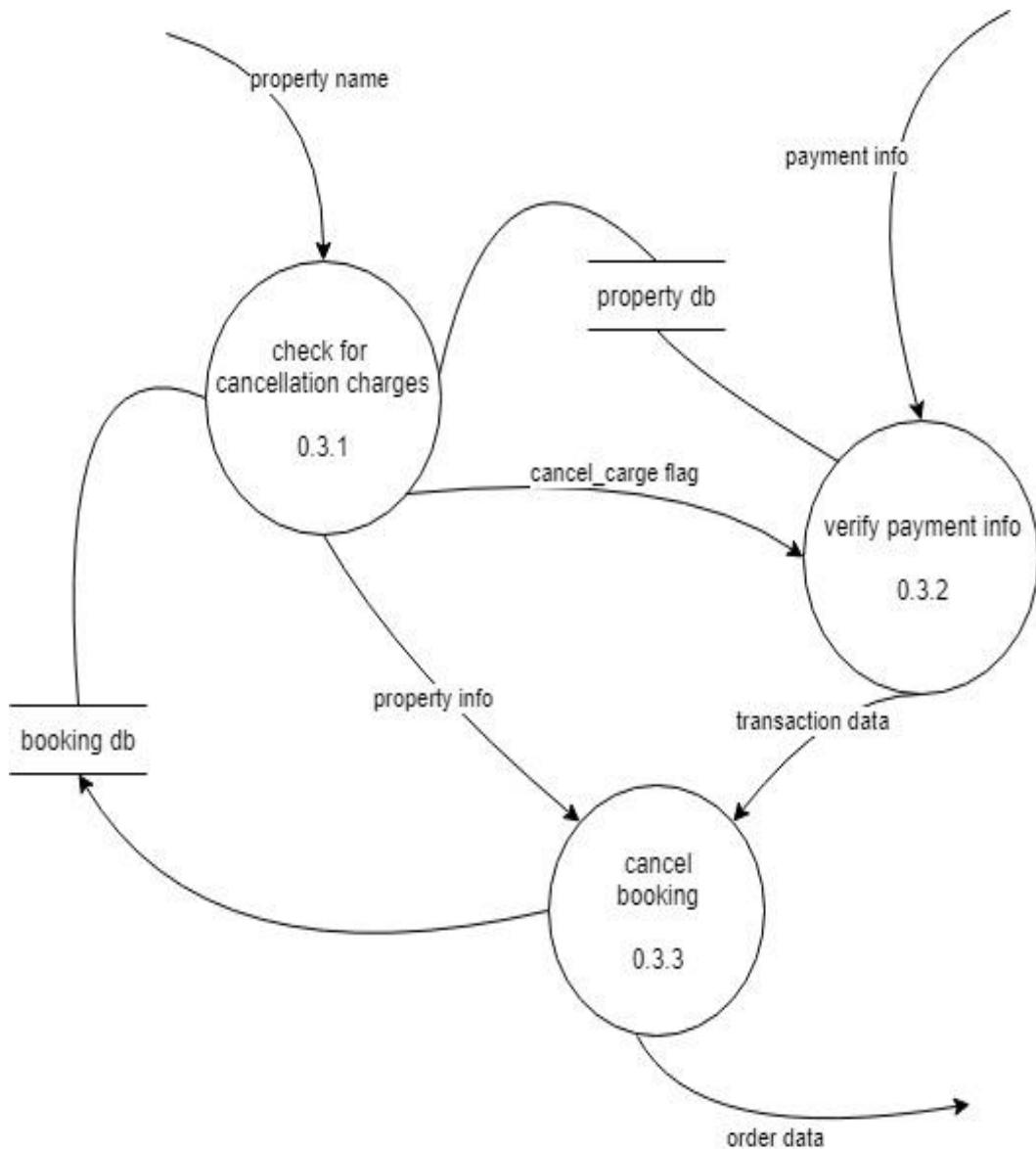


Figure 8 Level 2 DFD – registered user

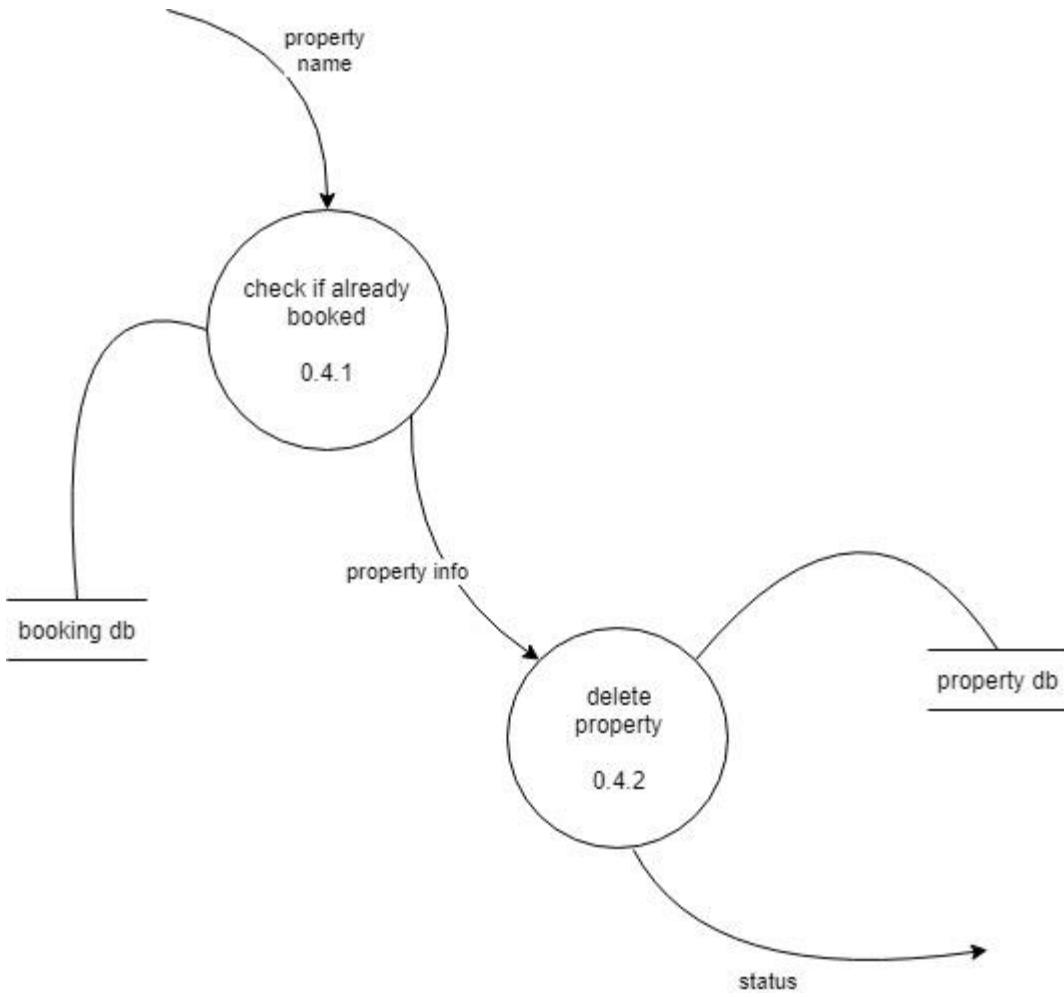


Figure 9 Level 2 DFD - host

4.3 Class Diagram

In software engineering, a **class diagram** in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

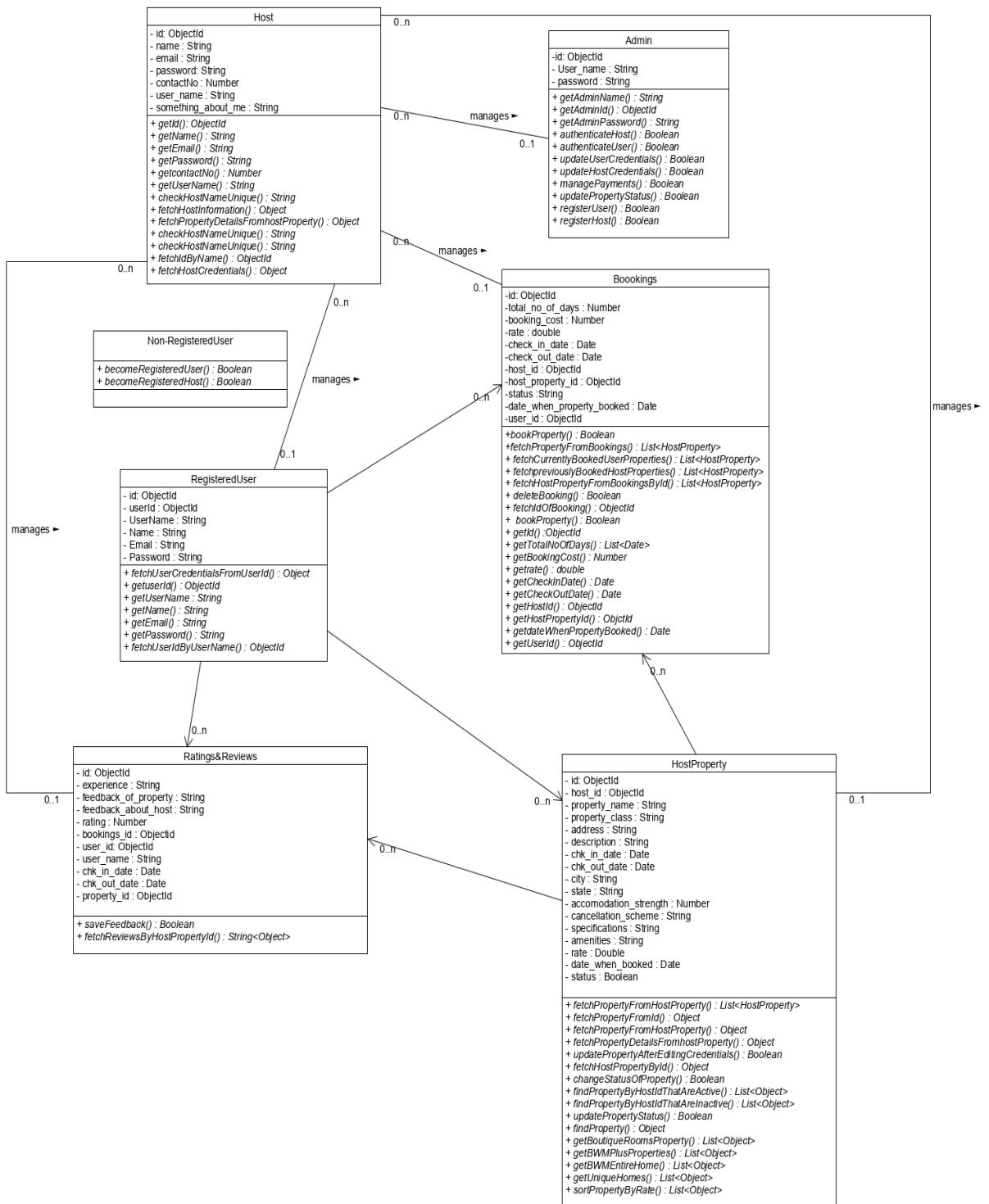


Figure 10 Class Diagram

4.4 Entity Relationship Diagram

An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database.

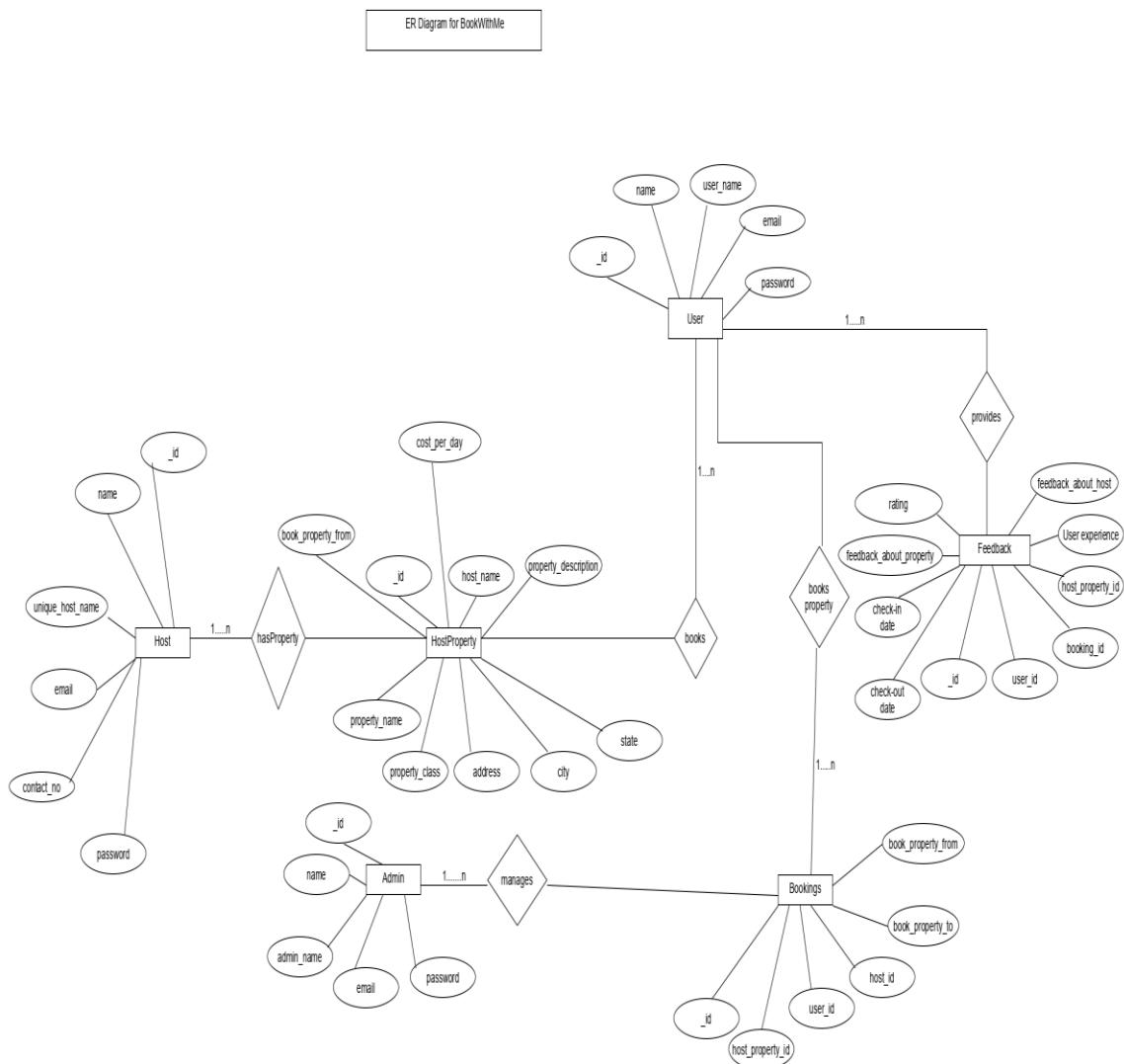


Figure 11 E-R Diagram

4.5 State Diagram

A **state diagram** is used to represent the condition of the system or part of the system at finite instances of time. It's a **behavioral** diagram and it represents the behavior using finite state transitions. It is used to state the events responsible for change in state (we do not show what processes cause those events).

4.5.1 State diagram for booking property

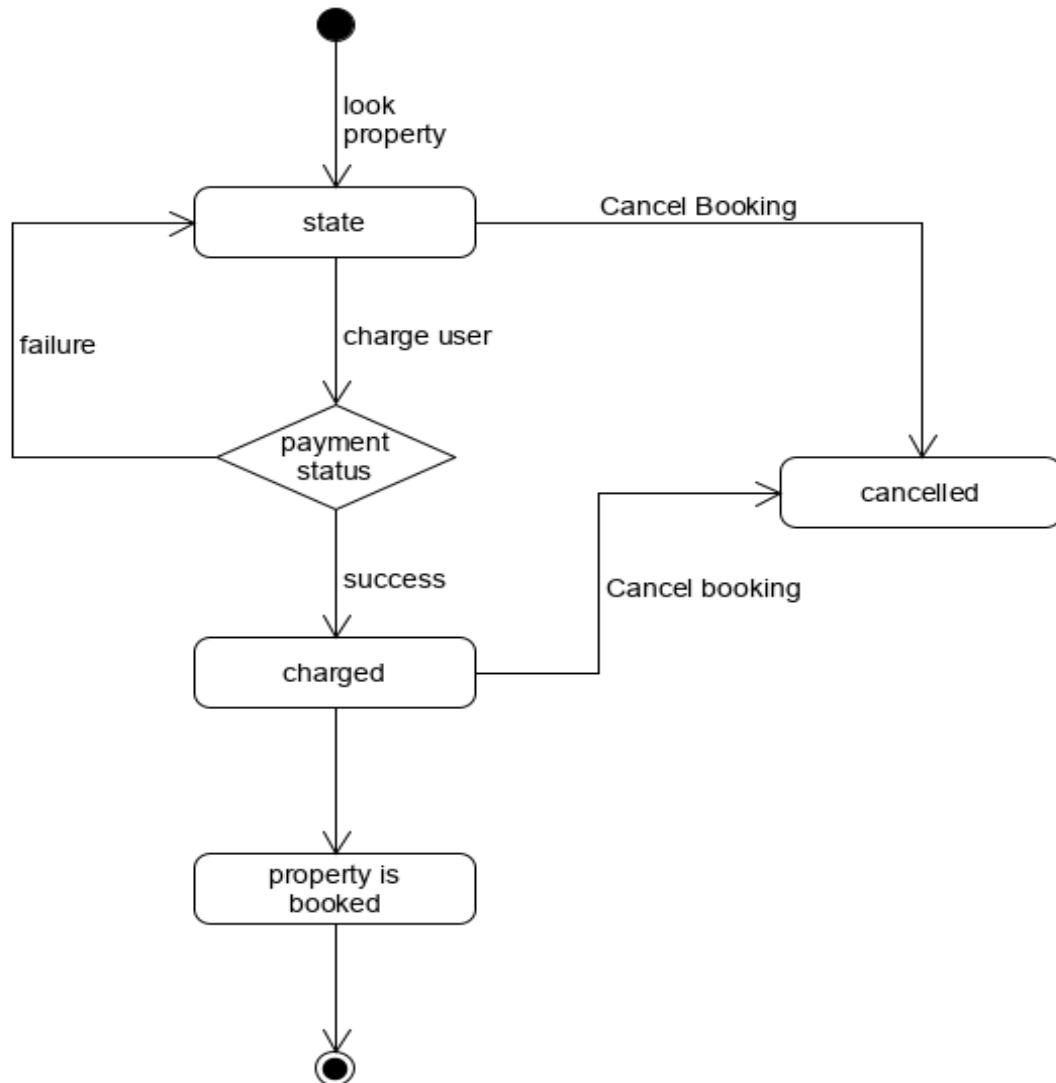


Figure 12 State Chart Diagram – booking property

4.5.2 State diagram for cancelling booking

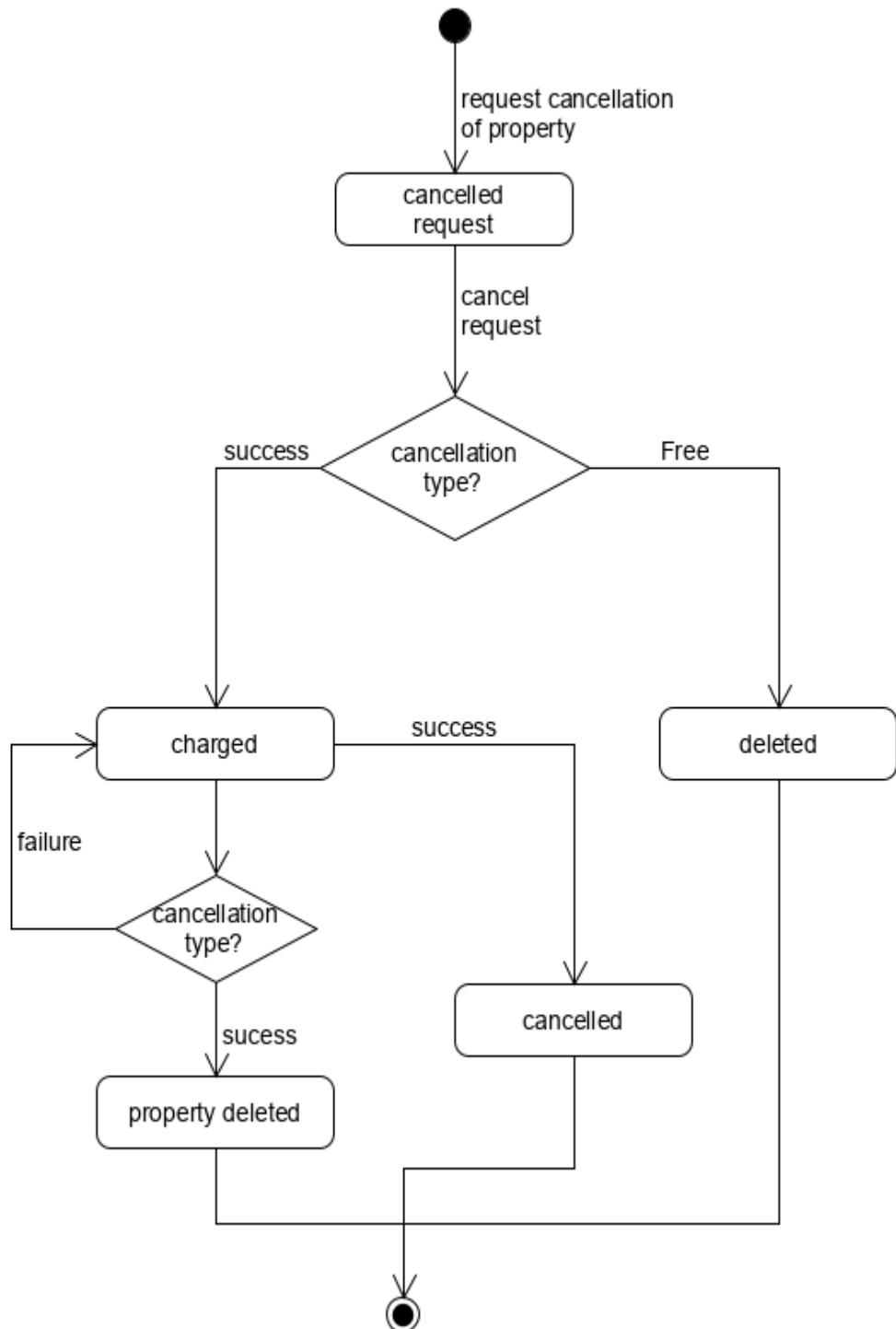


Figure 13 State Chart Diagram – cancel booking

4.6 Sequence Diagram

Sequence diagram is the most common kind of interaction diagram, which focuses on the message interchange between a number of lifelines. Sequence diagram describes an interaction by focusing on the sequence of messages that are exchanged, along with their corresponding occurrence specifications on the lifelines.

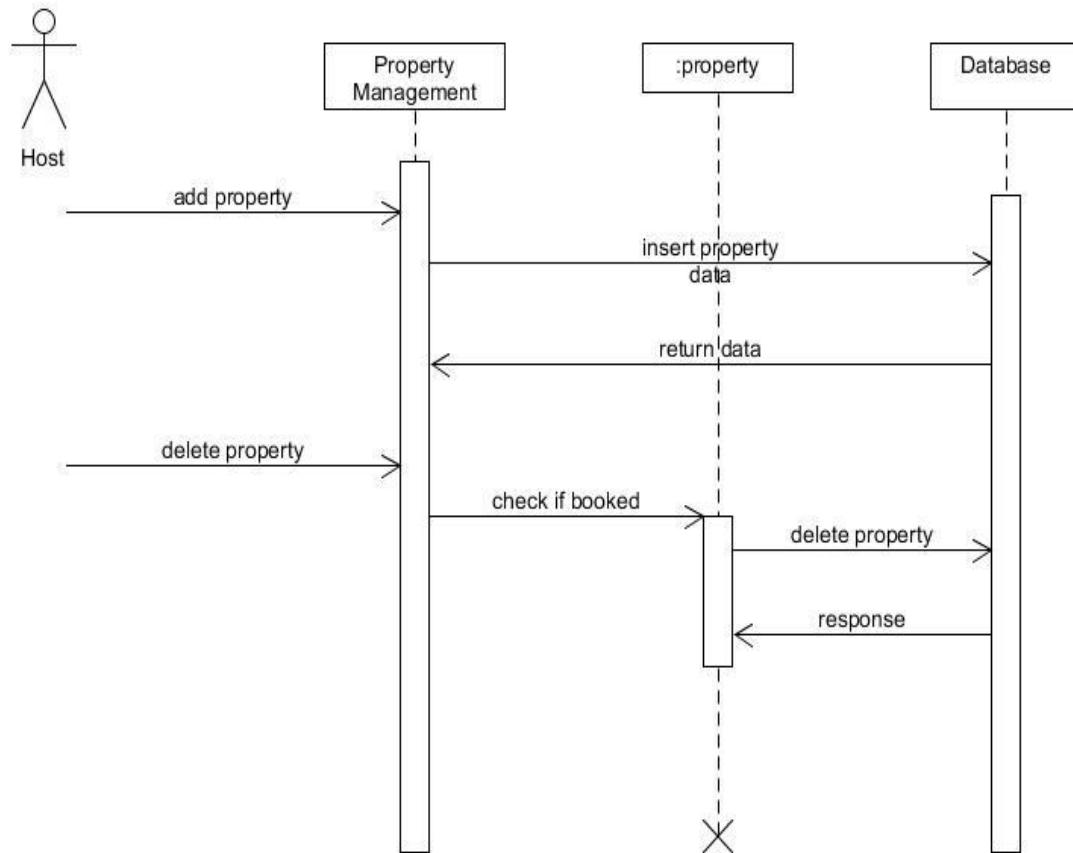


Figure 14 Sequence Diagram – property management

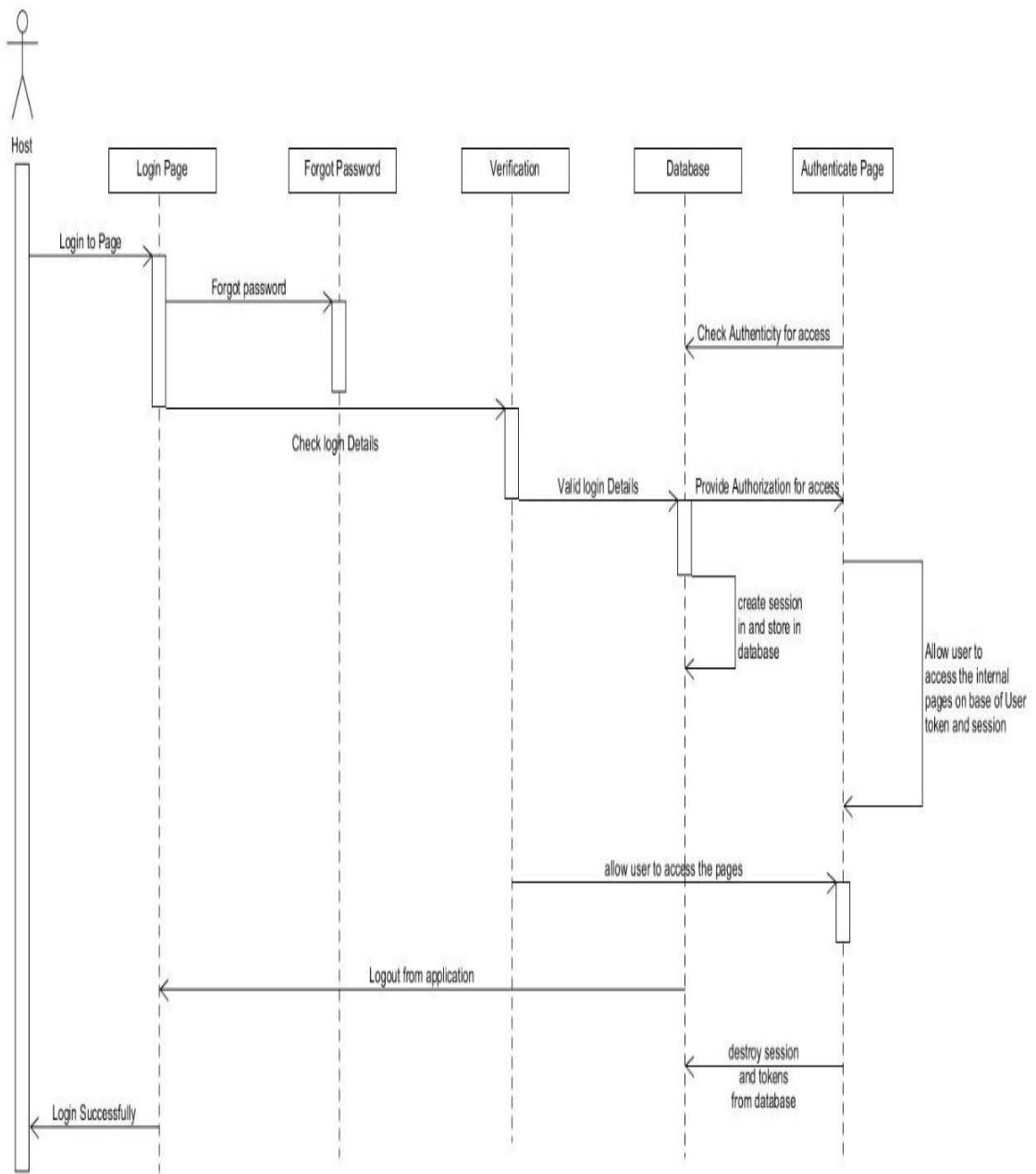


Figure 15 Sequence Diagram – Login Process

4.7 Component Diagram

Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

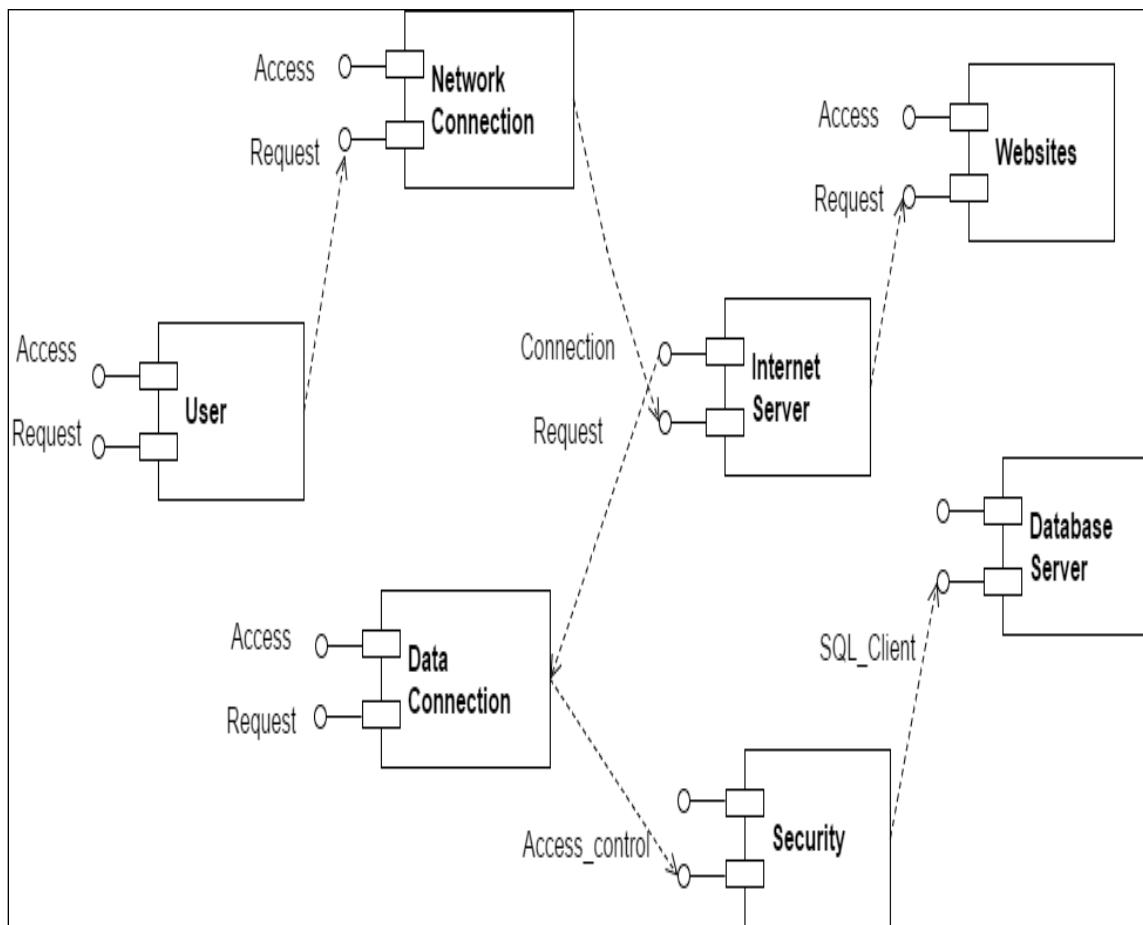


Figure 16 Component Diagram

4.8 Deployment Diagram

A UML deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them. Deployment diagrams is a kind of structure diagram used in modeling the physical aspects of an object-oriented system. They are often be used to model the static deployment view of a system (topology of the hardware).

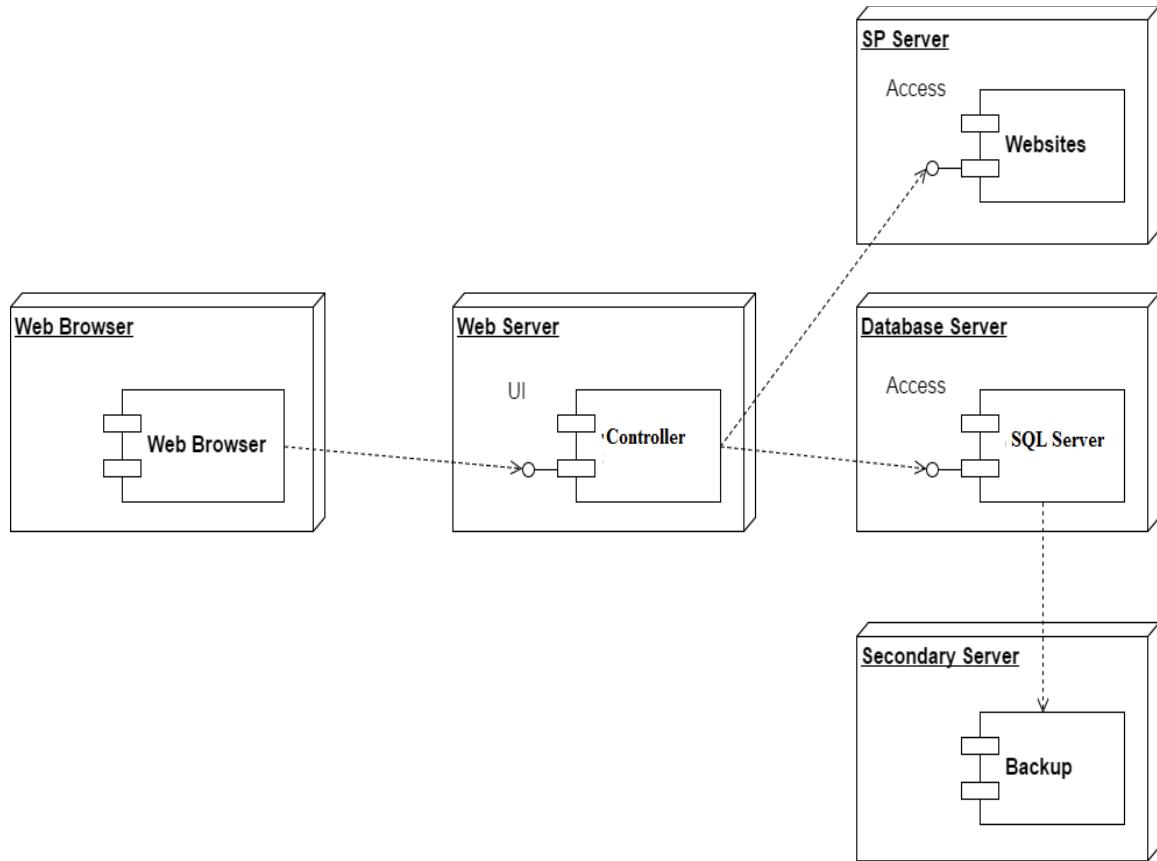


Figure 17 Deployment Diagram

4.9 Database Design

The following tables are the part of the BookWithMe Database

1. User
2. Host
3. Admin
4. Feedback
5. Host Property
6. Booking

4.9.1 Data Dictionary

A data dictionary, or Meta data repository, as defined in the IBM Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format. The term may have one of several closely related meanings pertaining to databases and database management systems (DBMS):

- A document describing a database or collection of databases.
- An integral component of a DBMS that is required to determine its structure.
- A piece of middle ware that extends or supplants the native data dictionary of DBMS.

COLUMN NAME	DATA TYPE	SPECIFICATION	DESCRIPTION
_id	ObjectId	Primary Key	User Unique Id.
Name	String		Name of the User
Username	String	-	Unique Username of the User
E-Mail	String	-	E-Mail id of the user
Password	String	-	Password for the User to log-in to the System.

Table 1 User Table

COLUMN NAME	DATA TYPE	SPECIFICATION	DESCRIPTION
_id	ObjectId	Primary Key	Host Unique Id.
Name	String		Name of the Host
Username	String	-	Unique Username of the Host
E-Mail	String	-	E-Mail id of the Host
Password	String	-	Password for the Host to log-in to the System.
Info	String	-	The info field stores the general introduction the Host gives to the Users which Users can see at the time of booking the property

Table 2 Host table

COLUMN NAME	DATA TYPE	SPECIFICATION	DESCRIPTION
_id	Integer	Primary Key	Orders Unique Id.
HostId	ObjectId	Foreign Key	Id of the Host of the property.
PropertyName	String	-	Name of the property
PropertyClass	String	-	Describes the type of the property
Address	String	-	Address where property is located
Available-from	Date	-	Date from which the user can book this property
Available-to	Date	-	Date to which the user can book the property
State	String	-	Name of the State
City	String	-	Name of the City
Accomodation	Number	-	The maximum number

Strength			of guests allowed per rental property.
CancellationScheme	String	-	Specifies if the cancellation is Paid or Free cancellation of the property
Amenities	String	-	Specifies what amenities are provided on booking this rental property
Specifications	String	-	What is the most peculiar thing about the property
Rate	Float	-	Rate of the rental property per night.
Status	String		Shows if the property is open or not for the booking by the users
DateUpdated	Date		This shows when the property information was update by the Host of this property

Table 3 HostProperty table

COLUMN NAME	DATA TYPE	SPECIFICATION	DISCRIPTION
_id	ObjectId	Primary Key	Unique Booking Id.
TotalDays	Integer		Total no of days for which the booking is made

Amount	Float	-	Booking total amount
Check-in-Date	Date	-	Users Check-in-Date
Check-out-Date	Date	-	Users Check-out-Date
HostPropertyId	ObjectId	Secondary Key	Property Id of which booking is made
HostId	ObjectId	Secondary Key	Id of the Host to whom the property belong which is being booked
PropertyIsBooked	Boolean	-	Shows the status of the booking. Shows if this Booking is active or not
DateOfBooking	Date	-	Date when the booking is made.
UserId	ObjectId	-	Id of the User who made the booking.

Table 4 Booking table

COLUMN NAME	DATA TYPE	SPECIFICATION	DISCRIPTION
_id	ObjectId	Primary Key	Unique feedback Id.
Experience	String	-	Overall experience about the host service and the Host property
PropertyFeedback	String	-	Feedback about the Property
HostFeedback	String	-	Feedback about the Host
BookingId	ObjectId	Secondary Key	Id of Booking to which the reviews and ratings are given

Rating	Integer	-	Overall ratings in the form of stars are given to the property
UserId	ObjectId	-	Id of the User who gave the ratings and review.
PropertyId	ObjectId	-	The id of the property to which the review and ratings are given

Table 5 Feedback Table

4.10 Application Navigation

The application is broken down into three Pieces. The Registered Users, Registered Hosts and The Non-Registered Users. Each one of them have different Navigations. All the navigations are handled by the Navigation Bars in the System. Others special navigations are handled by the link buttons.

4.10.1 Navigation for Non-Registered Users

Non Registered Users can navigate to perform following actions

- User log-in
- User-Signup
- Host-Login
- Become Certified Host
- Help

4.10.2 Navigation for Registered Users

Registered Users can navigate to perform following actions

- Manage your Credentials
- Manage Bookings
- Book and Explore properties

- Show my previous Bookings
- User Dashboard

4.10.3 Navigation for Registered Hosts

Registered Hosts can navigate to perform following actions

- Navigate to Host Dashboard
- Manage Credentials
- Create Rentals
- Manage Properties
- View Inactive properties

4.11 Architectural Designing of the System

This system is implemented by using MVC Architecture.

Why MVC?

Model–View–Controller (MVC) is an architecture that separates the representation of information from the user's interaction with it. It is a software architectural pattern for implementing user interfaces on computers. It divides a given application into three interconnected parts in order to separate internal representations of information from the ways that information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development

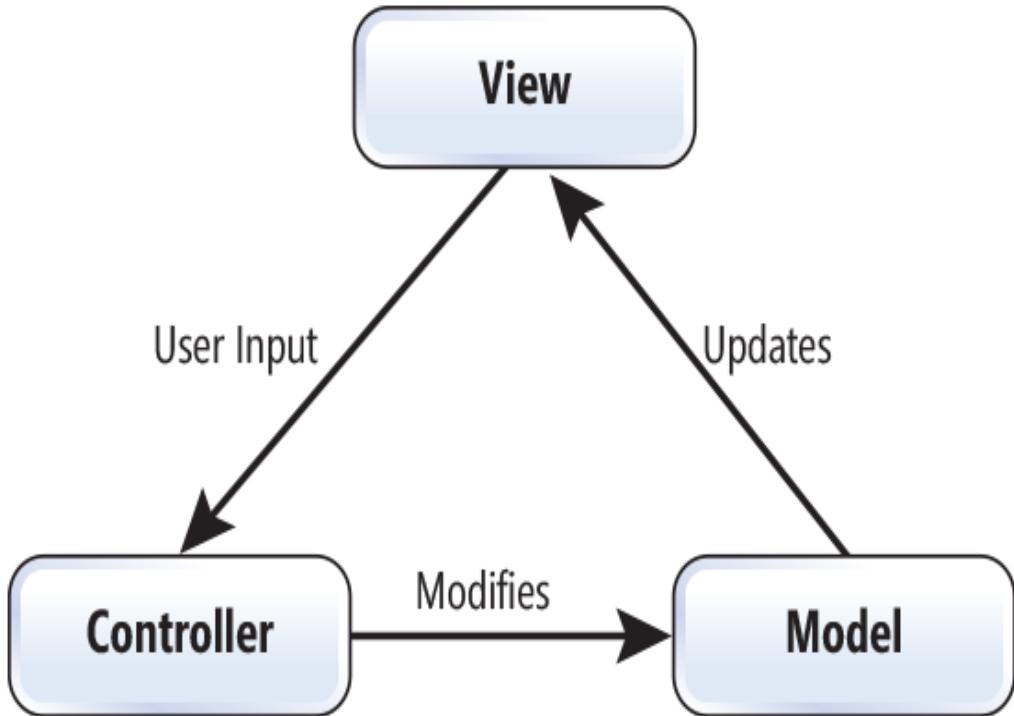


Figure 18 MVC model

A. Model

Model code is written in some high-level programming language such as Java, Python, or C++. Usually the model code interfaces with a database using the Standard Query Language (SQL). In our system the model is implemented by Node.js

B. View

View represents the visualization of the data that model contains. The views in the System are EJS views.

C. Controller

The controller part is handled by Node.js at the back end. Node.js manages to control the business logic and is a server code that gets invoked whenever a user clicks on any link in the web pages or interacts enters data and submits a form.

CHAPTER 5

IMPLEMENTATION

5.1 PROJECT MANAGEMENT

PHASE	DELIVERABLES	PURPOSE
System Requirement and Analysis	<ul style="list-style-type: none"> Requirement Gathering and analysis. Functional Specifications Non Functional Specifications 	It gives the exact understanding of the user's requirements.
System Design	<ul style="list-style-type: none"> E-R diagram Data flow diagram Use case diagram Class diagram Activity diagram 	It gives the logical structure that describes the system.
Implementation and Testing	<ul style="list-style-type: none"> The output obtained for the required functionality after implementing and doing various types of testing 	It gives the required module

Table 6 Milestones and Deliverables

5.1.1 Roles AND RESPONSIBILITIES

Name	Role				
	Analysis	Designing	Coding	Testing	Documentation
Prayag Desai	✓	✓	✓	✓	✓

Table 7 Roles and responsibilities

5.2 Code Snippets

This System is Implemented using MVC Architectural pattern. The MVC modules are not hard coded in Node.js. Node.js Express app contains app.js file which contains the configurations on how to use middleware and all the third-party packages

In the *app.js* file

```
const express = require('express');
```

1. This is how third party packages are used in NodeJS project.

```
const app = express();
```

2. To use the express in the app use a constant that remains the same in the application so that there are no problems in the naming conventions

```
app.set('view engine','ejs');  
app.set('views','views');
```

3. This how templating engine is set in NodeJS express project

```
app.use(express.static(path.join(__dirname, 'public')));
```

4. Set the static path in the express projects. This code creates a folder public in the project folder. Public folder is used to keep the css javascript and all the static files needed for the project

```
app.use(bodyParser.urlencoded({ extended: false }));  
app.use(cookieParser());  
app.use(session({secret : 'prayagdesai'}));
```

5. The above code sets the body-parser which is used in Express to maintain cookies and sessions.

```
const adminRouter = require('./routers/admin');
```

6. This code is used to load the controller to the app.js. Every request made to the server is handled by the middleware in sequence. The middleware then executes the request and renders the page as requested

```
app.use(nonRegisteredUsersRouter);  
app.use(adminRouter);  
app.use(registeredHostsRouter);  
app.use(registeredUsersRouter);  
app.use(errorRouter);
```

7. These are the middlewares that handles the incoming request to the server.

The *database.js* file in the utility folder has the connection to the database. The database server will get connected once the back-end server gets connection.

The database will be created if there is no database or if there is a database, then the operation on the database will be performed.

Database.js

```
const mongodb = require('mongodb')  
const mongoClient = mongodb.MongoClient; // calling the mongodb client constructor  
  
let url = "mongodb://localhost:27017/demodbTest";
```

```

let _db // this underscore variable is used in this file only to fetch the type of database schema

const mongoconnect = (callback) => { // callback function passed on creating and connecting
application to the database server
    mongoClient.connect(url,{useNewUrlParser:true}).
    then(client =>{
        console.log('Connected');
        _db = client.db()
        callback(client);
    })
    .catch(err =>{
        console.log('Error in connecting database');
        throw err;
    })
}

// this function connects to the database _db or else returns error on not finding database
const getDb = () =>{
    if(_db)
    {
        return _db;
    }
    else{
        throw 'No database found';
    }
}

exports.mongoconnect = mongoconnect;
exports.getDb = getDb;

```

Every incoming request is in the form of URL and the Router matches the URL and subsequently the Controller handles the URL

There are separate routers for Registered Hosts, Registered Users, Non-Registered Users and Admin

```

const express = require('express');
const router = express.Router();

const adminController = require('../controllers/admin_c');

router.post('/registerhost',adminController.postbecomeHost);
router.post('/user-registration-successfull',adminController.postbecomeUser);
router.post('/user-auth',adminController.postAuthenticateUser);
router.post('/host-auth',adminController.postAuthenticateHost);

```

```
module.exports = router;
```

Routing is carried away by Router() of the Express Library. The routing request can either be get request or post request. This can be handled by the router by specifying the method. The router method takes two arguments 1. URL and the 2. The controller that handles the request.

There are separate controllers for Registered Hosts, Registered Users, Non-Registered Users and Admin. Each controller contains a middleware implementation which is configured in the routers. The Business logic is implemented in the controller. The controller communicates with the Model. The Model manipulates the data and gives the result to the controllers and the controllers then passes the data to View to render the data. This separation of concern is demonstrated below

One such example of Controller is. The postPaidCancellation Controller takes three arguments. The request, response and next object. This controller handles the paid cancellation of the booking by the Registered User.

```
exports.postPaidCancellation = (req, res, next) =>{
  let sess = req.session;
  const booking_id = req.body.booking_id;
  console.log(booking_id);
  const cancelation_charge = req.body.cancellation_charge;
  Bookings.deleteBooking(booking_id).then(result =>{
    const user_name = sess.userCredentials.user_name;
    const user_id = sess.userCredentials._id; // this is of type string
    let aux_array = [];

    // Set your secret key: remember to change this to your live secret key in production
    // See your keys here: https://dashboard.stripe.com/account/apikeys
    var stripe = require("stripe")("sk_test_6SY2LcDXwkLcXil0ICEIFvXq005Xa3W26F");

    // Token is created using Checkout or Elements!
    // Get the payment token ID submitted by the form:
    const token = req.body.stripeToken; // Using Express
```

```

(async () => {
  const charge = await stripe.charges.create({
    amount: req.body.cancellation_charge * 100,
    currency: 'usd',
    description: 'Cancellation Charge',
    source: token,
    metadata : {Booking_id_canceled : req.body.booking_id.toString(),
    User_who_canceled_this_property : sess.userCredentials.user_name}
  }) // charge ends
  Bookings.fetchCurrentlyBookedUserProperties(user_id).then(currentlyBookedProperties =>{
    if(currentlyBookedProperties.length === 0){
      // if there are no bookings of this user in the bookings table
      res.render('registered-users/view-booked-properties',{
        a : [],
        b : [],
        username : user_name,
        message : false,
        delete_message : true
      });
    } // render ends here
  }) // if ends here
  else {
    currentlyBookedProperties.forEach( (element) =>{
      Bookings.fetchPropertyDetailsFromhostProperty(element.host_property_id).then(ans =>{
        if(aux_array.length < currentlyBookedProperties.length){
          aux_array.push(ans)
        }
        if(aux_array.length === currentlyBookedProperties.length){
          //console.log('AUX ARRAY IS ')
          //console.log(aux_array);
          //console.log('B IS ')
          //console.log(aux_array);
          res.render('registered-users/view-booked-properties',{
            a : aux_array,
            b : currentlyBookedProperties,
            username : user_name,
            message : false,
            delete_message : true
          });
        }
      }) // render ends
    }) // inner if ends
  } // if ends
}) // forEach ends
}).catch(err =>{
  console.log(err);
}) // Bookings.fetchPropertyDetailsFromhostProperty promise ends
}) // forEach ends

```

```

} // else ends here

}).catch(err =>{
  console.log('error in fetching host properties from bookings')
}) // Bookings.fetchCurrentlyBookedHostProperties promise over
})();

}).catch(err =>{
  console.log(err);
})
}

```

The res.render() renders the view. This method takes two arguments. The path of the EJS template which is to be rendered and the javascript object which contains the data to be passed to the templating engine (EJS View).

Inside the EJS view, the passed data can be embedded to the html by the wrapping the scriptlet tags <% = objectKey%>.

One such demo EJS view is view-booked-properties.ejs which displays the currently booked properties by the User.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">

  <title>Document</title>
</head>
<body>
  <% if (delete_message === true) { %>
  <div align="center">
    <div class="alert alert-success">
      <strong>Booking Successfully Deleted!</strong>
    </div>
  </div>
<% } %>

```

```

</div>
<% } %>
<nav class="navbar navbar-expand-sm navbar-light bg-light">

    <a class="navbar-brand" href="/">BookWithMe</a>
    <div class='container'>
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="manage-credentials">Manage your Credentials
                    <span class="sr-only">(current)</span>
                </a>
            </li>

            <li class="nav-item active">
                <a class="nav-link" href="/manage-booking">Manage Bookings
                    <span class="sr-only">(current)</span>
                </a>
            </li>

            <li class="nav-item active">
                <a class="nav-link" href="/book-properties-nav">Explore and Book Properties
                    <span class="sr-only">(current)</span>
                </a>
            </li>
            <li class="nav-item active">
                <a class="nav-link" href="/previous-bookings">Show My Previous Bookings
                    <span class="sr-only">(current)</span>
                </a>
            </li>
        </ul>
    </div>

</nav>

<div class="jumbotron" align="center">

    <h1 class="display-4">Hello, <%= username %></h1>
    <p class="lead">View properties booked by you.</p>
    <hr class="my-4">
    <p>We have your currently booked properties and also maintain properties booked by you in near past.</p>

```

```

<p class="lead">

</p>
</div>
<br>
<br>

</div>

<div class="container" align='center'>
<div align='center'>
<p class='lead display-4'>
    Currently Booked Properties are
</p>

<br>
<br>

</div>
<%if (a.length === 0) { %>
<div class="container">
<p class="lead">
    You have not booked any properties.
</p>
</div>
<% } else { %>

<% for(var i = 0; i < a.length; i++ ) { %>
<div class="card bg-secondary text-white" align="center">
    <div class="card-header" id="id1">
        <p class="h3">
            <%= a[i].property_name%>
        </p>
    </div>

    <div class="card-body">
        <h5 class="card-title"><%= a[i].property_name%> %></h5>
        <p class="card-text"><%= a[i].address%></p>
    </div>
</div>

```

```

<p class="display-5"> You booked this property on <%=
b[i].date_when_property_booked %> from <%= b[i].check_in_date %> to <%= b[i].check_out_date
%></p>
<div class="center">
<div class="container">
<div class="row">
<div class="col-sm">
<form action="/property-details" method="POST">
<input type="hidden" name="property_id" value="<% a[i]._id %>">

<button type="submit" class="btn btn-info">View Property Details</button>
</form>
</div>
<div class="col-sm">
<form action="/rate-property" method="POST">
<input type="hidden" value="<% a[i]._id %>" name="property_id">
<input type="hidden" name="chk_in_date_from_booking" value="<%=
b[i].check_in_date %>">
<input type="hidden" name="chk_out_date_from_booking" value="<%=
b[i].check_out_date %>">
<input type="hidden" name="bookings_database_id" value="<%= b[i]._id %>">
<button type="submit" class="btn btn-info">Rate this property</button>

</form>
</div>
</div>
</div>
</div>
</div>

</div>
<br>
<br>
<% } %>

<% } %>

```

```

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzOOrT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"
integrity="sha384-
ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js"
integrity="sha384-
smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQp0Mk8ck+5T"
crossorigin="anonymous"></script>
</body>

</body>
</html>

```

The **.gitignore** file contains the folder that are not to be tracked by the git version controlling system.

The **package.json** contains the metadata for the project package. The package.json file also contains the information about the ownership and license of the project. The runtime configuration scripts are specified in the package.json file. Also the file contain the version of all the third party packages used in the project. The package.json is generate by the Node's package manager. The package.json file for this project is

```
{
  "name": "bookwithme",
  "version": "1.0.0",
  "description": "bookwothme is an online rental service",
  "main": "app.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1",
    "start": "nodemon app.js"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/PrayagDesaiweb/BookWithMe.git"
  }
},
```

```
"author": "Prayag Desai",
"license": "ISC",
"bugs": {
  "url": "https://github.com/PrayagDesaiweb/BookWithMe/issues"
},
"homepage": "https://github.com/PrayagDesaiweb/BookWithMe#readme",
"dependencies": {
  "body-parser": "^1.18.3",
  "bootstrap": "^4.2.1",
  "cookie-parser": "^1.4.3",
  "ejs": "^2.6.1",
  "express": "^4.16.4",
  "express-session": "^1.15.6",
  "jquery": "^3.3.1",
  "mongodb": "^3.1.11",
  "node-schedule": "^1.3.2",
  "nodemon": "^1.18.9",
  "popper.js": "^1.14.7",
  "stripe": "^6.28.0"
}
}
```

In this project following third-party Node Packages are used

```
"dependencies": {
  "body-parser": "^1.18.3",
  "bootstrap": "^4.2.1",
  "cookie-parser": "^1.4.3",
  "ejs": "^2.6.1",
  "express": "^4.16.4",
  "express-session": "^1.15.6",
  "jquery": "^3.3.1",
  "mongodb": "^3.1.11",
  "node-schedule": "^1.3.2",
  "nodemon": "^1.18.9",
  "popper.js": "^1.14.7",
  "stripe": "^6.28.0"
}
```

The projects needs this packages for development of the application. All this dependency packages are located inside the **node modules** folder.

5.3 Coding Standards

Coding standards for development for Nodejs Express are strictly followed. The naming conventions are strictly adhered by. All the code are check for exceptions and all the exceptions are handled. Async code is handled by promises so that the code is made error free, accurate as much as possible. Latest JavaScript ES6 features are implemented in the project. Also the code contains relevant documentation.

CHAPTER 6

TEST CASE DESIGN AND SNAPSHOTS

6.1 Test Cases

This system is a conglomeration of three main component. Registered Hosts, Registered Users and Non-Registered-Users. Each of these test cases are discussed.

Test case Id	Test Scenario	Test Steps	Test Data	Expected Result	Expected Result	Pass / Fail
NRU -1	Check navigation to user-login	Click user-login in the navbar`	-	Directed to the user-login page	Directed to the user-login page	Pass
NRU -2	Check navigation to host login	Click host-login in the navbar`	-	Directed to the host-login page	Directed to the host-login page	Pass
NRU -3	Check navigation to user-signup	Click user-signup in the navbar`	-	Directed to the user-signup page	Directed to the user-signup page	Pass
NRU -4	Check navigation to host-signup	Click host-signup in the navbar`	-	Directed to the host-signup page	Directed to the host-signup page	Pass
NRU -5	Check accommodation Or explore properties as per location and check-in credentials	Fill in the information on where and when you want to view properties`	City= Nadiad State=Gujarat Check-in-date: 28-4-19 Check-out-date : 1- 3 - 19	Page shows properties in Nadiad according to the data entered but the non-registered users	Page shows properties in Nadiad according to the data entered but the non-registered users	Pass

Table 8 Test cases for non registered users

Test case	Test Scenario	Test Steps	Test Data	Expected Result	Expected Result	Pass/ Fail
RU-1	Log-in as Registered user	Enter the username and password	Username = Prayagdesai, Password = Password	Directed to the User dashboard	Directed to host dashBoard	Pass
RU-2	Log-in as Registered User	Enter the username and password	- Username = Prayagdesai, Password = Password12	Redirected to the user-login page. User Credentials are incorrect.	Redirected to the user-login page.	Pass
RU-3	Update credentials	Enter the changed credentials	Username = Prayagdesai1234	Redirected to log-in page to log-in with the new credentials	Redirected to log-in page to log-in with the new credentials	Pass
RU-4	Manage Bookings	View Rental details, Cancel this booking or view booking details	-	If current bookings are made then the bookings will be shown. Else System will show no bookings	If current bookings are made then the bookings will be shown. Else System will show no bookings	Pass
RU-5	View property details	Navigate to the Manage Bookings and click on the view property detail link button of the booking.	-	The page shows the property details and all the information regarding the property	The page shows the property details and all the information regarding the property	Pass

RU -6	Cancel this booking	Navigate to the Manage Bookings and click on the view cancel this booking button of the booking.	-	If the booking is under free-cancellation scheme, The booking is deleted. If the booking is under paid cancellation, User is Directed to payment page to pay the cancellation charge	If the booking is under free-cancellation scheme, The booking is deleted. If the booking is under paid cancellation, User is Directed to payment page to pay the cancellation charge	Pass
RU -7	Delete paid booking	Navigate to the Manage Bookings and click on the view cancel this booking button of the booking. If the booking is under paid cancellation scheme, user will redirect to the delete-booking page	Email = Prd243@gmail.com Card Number = 42424242424242 Month and date = 09/19 CVC Number = 1203	On successful payment of cancellation fee, User is redirected to the host-dashboard page. The user will see the deleted property is removed from the user dashboard.	On successful payment of cancellation fee, User is redirected to the host-dashboard page. The user will see the deleted property is removed from the user dashboard	Pass
RU -8	View Booking details	Navigate to the Manage Bookings and click	-	The booking related information and	The booking related information and	Pass

		on the view property detail link button of the booking.		detail are displayed to the User	detail are displayed to The User	
RU -9	Explore Properties by location	Navigate to Book and Explore properties from the User Dashboard	City = Dallas State = Texas	The System lists all the property available for rent in Dallas, Texas. The system will show no property if there are no property in the specifies city and state	The System lists all the property available for rent in Dallas, Texas The system will show no property if there are no property in the specifies city and state	Pas s
RU -10	Explore property by location and check-in and check-out date and number of accommodation per rental	Navigate to the Manage Bookings and click on the view property detail link button of the property.	City= Dallas State = Texas Check-in-date = 12-10-2019 Check-out-date = 22-10-2019 Accommodation per rental = 5	The System lists all the property available for rent in Dallas, Texas. The system will show no property if there are no property in the specifies city and state	The System lists all the property available for rent in Dallas, Texas. The system will show no property if there are no property in the specifies city and state	Pas s

RU -11	View only BookWithMe properties	Navigate to the Manage Bookings and click on the view property detail link button of the booking. If there are properties in the specified location, Click on the View only BookWithMe plus properties	-	The System will show only BookWithMe plus properties to the User. If there are no such properties, the system will show no properties to the user	The System will show only BookWithMe plus properties to the User. If there are no such properties, the system will show no properties to the user	Pass
RU -12	View only BookWithMe Entire Homes	Navigate to the Manage Bookings and click on the view property detail link button of the booking. If there are properties in the specified location, Click on the View only BookWithMe entire Homes	-	The System will show only BookWithMe Entire Homes to the User. If there are no such properties, the system will show no properties to the user	The System will show only BookWithMe Entire Home to the User. If there are no such properties, the system will show no properties to the user	Pass
RU -13	View only BookWithMe Boutique Rooms	Navigate to the Manage Bookings and click on the view property detail link button of the booking. If there are	-	The System will show only BookWithMe Boutique rooms properties to the User. If there are no such	The System will show only BookWithMe boutique rooms properties to the User. If there are no such	Pass

		properties in the specified location, Click on the View only BookWithMe Boutique rooms		properties, the system will show no properties to the user	properties, the system will show no properties to the user	
RU -14	Sort properties by decreasing order of rates of the rental					
RU -15	Rate the property	Navigate to user dashboard and select the property you want to give rating to	Enter the details about your experience in the property, Add review about the property and the Host's hospitality . Also give final rating.	The system will submit the rating given bu the user. If the user has not stayed in the property after the booking. The User then cannot rate the property	The system will submit the rating given bu the user. If the user has not stayed in the property after the booking. The User then cannot rate the property	Pas s
RU -16	View Previously booked properties.	Navigate to View previously booked properties	-	The system will show you all your previous bookings if you have any. Or else the system will show you no previous bookings made by the User	The system will show you all your previous bookings if you have any. Or else the system will show you no previous bookings made by the User	Pas s

Table 9 Test Cases for registered users

Test case Id	Test Scenario	Test Steps	Test Data	Expected Result	Expected Result	Pass / Fail
RH -1	Log-in as Registered host	Enter the username and password	Username = Prayagdesai, Password = Password	Directed to the User dashboard	Directed to the User dashboard	Pass
RH -2	Log-in as Registered Host	Enter the username and password	Username = Prayagdesaiqwe, Password = Password	Redirected to the host-login page. User Credentials are incorrect.	Redirected to the host-login page. User Credentials are incorrect.	Pass
RH -3	Host dashboard	Visit the host dashboard	-	Registered Host is directed to the host dashboard	Registered Host is directed to the host dashboard	Pass
RH -4	Update Host Credentials	Navigate from Host Dashboard to Update Credentials link button.	Previous username = Prayagdesai Updated Username = Prayagdesai1234	After successful updation of host credentials the host is redirected to the host dashboard	After successful updation of host credentials the host is redirected to the host dashboard	Pass
RH -5	Host can make their properties available to registered users for rent	Navigate from host Dashboard to Create Renal Property link button	Enter the information about the property.	After the property is created, the host is redirected to the Host Dashboard.	After the property is created, the host is redirected to the Host Dashboard	Pass
RH -6	View your deleted properties	Navigate from host dashboard to the View your inactive properties by clicking the link button	-	This section shows the properties deleted by you which you no longer wish to rent. You can enable this	This section shows the properties deleted by you which you no longer wish to rent. You can enable this	Pass

				properties.	properties	
RH -7	Host can make their inactive properties active again	Navigate from host dashboard to the View your inactive properties by clicking the link button. Click make property active link button	Enter the information about the property.	Upon successful updation of property details, the property is made active from check-in date to check-out date.	Upon successful updation of property details, the property is made active from check-in date to check-out date.	Pass
RH -8	Host can edit information of property - 1	From the host dashboard. From all the properties that has been declared by the Host for rent, click the link button for Edit property for the property host wants to edit	Enter the information about the property.	If the property is already booked by some user, the host cannot edit information such as rates, payment scheme, name of the property, check-in and check-out dates. Upon successful updation of the details, the host is redirected to the host dashboard	If the property is already booked by some user, the host cannot edit information such as rates, payment scheme, name of the property, check-in and check-out dates. Upon successful updation of the details, the host is redirected to the host dashboard	Pass
RH -9	Host can edit information of	From the host dashboard. From all	Enter the information about the property.	If the property is not booked by some	If the property is not booked by some	Pass

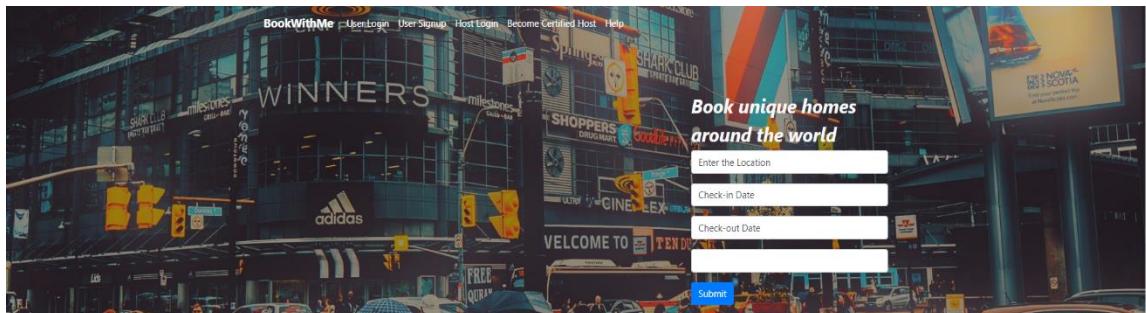
	property – 2	the properties that has been declared by the Host for rent, click the link button for Edit property for the property host wants to edit		user, the host can edit any property information	user, the host can edit any information	
RH -10	The Host can view the bookings made by the users on their property	From the host dashboard. From all the properties that has been declared by the Host for rent, click the link button for View property details.	-	The Host will see the bookings made by the users on that particular property along with the information of the property	The Host will see the bookings made by the users on that particular property along with the information of the property	Pass
RH -11	Hosts can delete the property - 1	From the host dashboard. From all the properties that has been declared by the Host for rent, click the link	-	The host cannot delete the properties that are already booked by some users. The Host is redirected to the Host dashboard or host is	The host cannot delete the properties that are already booked by some users. The Host is redirected to the Host dashboard or host is	Pass

		button for View property details.		able to navigate anywhere	able to navigate anywhere	
RH -12	Host can delete the property – 2	From the host dashboard. From all the properties that has been declared by the Host for rent, click the link button for View property details.	-	The host can delete the property if no user has booked the property. The user is then redirected to the View Inactive properties page	The host can delete the property if no user has booked the property. The user is then redirected to the View Inactive properties page	Pass

Table 10 Test Cases for registered hosts

6.2 Screenshots

1. BookWithMe home page



Welcome to BookWithMe !!

Letting complete strangers into your home might not sound like a good idea to everyone. Neither, on the flip side, is letting yourself into the home of a complete stranger. But for the many millions of users of accommodation-sharing site BookWithMe, the idea isn't just good, it's brilliant.

BookWithMe is an online marketplace which lets people rent out their properties or spare rooms to guests.

We Love Travellers and Explorers

Hotel stays are boring. Enjoy our Host's hospitality. We make you feel like home even though you're not at home. You have thousands of properties and hosts to choose from. We aim to make your stay memorable and comfortable.

Figure 19 Home Page

2. User log-in

A screenshot of the BookWithMe user login page. The top navigation bar is identical to the home page. The main area has a light gray background with the word "Log in" in large, bold letters. Below it is a sub-headline: "User Log in for exploring and booking rentals anywhere and everywhere with us". At the bottom, there are two input fields: "User Name" and "Password", each with a placeholder text ("Enter your username" and "Enter the password that you had used for registration"). Between the fields is a small note: "Forgot the username that you had used for registration?". A "Submit" button is located below the password field.

Figure 20 User Login

3. Become certified Users

The screenshot shows the 'User Signup' page for BookWithMe. At the top, there's a navigation bar with links for 'Host Login', 'User Login', 'Become certified Host', and 'Help'. Below the navigation, a greeting 'Hello Guest!!' is displayed. A sub-greeting 'Become a part of BookWithMe and explore great rentals available wherever and whenever you go.' follows. A promotional message 'You are just a click away! With one time fee of \$45 we offer you exotic properties across the globe' is shown, with a 'Learn more' button. The main form area contains fields for 'UserName' (with placeholder 'Enter your name'), 'Name' (placeholder 'Enter your name'), 'Email address' (placeholder 'Enter email'), and 'Password' (placeholder 'Password'). A 'Pay with Card' button is located at the bottom right of the form. A progress bar at the bottom left indicates 'Waiting for qstripe.com...'.

Figure 21 User Signup

4. Host log-in

The screenshot shows the 'Host Login' page for BookWithMe. At the top, there's a navigation bar with links for 'BookWithMe', 'User Signup', 'User Login', 'Become certified Host', and 'Help'. Below the navigation, a greeting 'Hello, Guest!' is displayed. A sub-greeting 'Sign-in with BookWithMe to start renting your property on your convenience. Maintain existing properties and Accessing details about your current property on rent' follows. A 'Learn more about Registered hosts' button is present. The main form area features a purple square icon with a white letter 'B' and the text 'Host Login'. Below this, a note says 'Keep properties on rent by Logging in First. If you are not Registered Host, Click here to become one.' There are two input fields: 'Enter Username' and 'Password', and a 'Log in' button.

Figure 22 Host Login

5. User sign-up

The screenshot shows the 'BookWithMe' host sign-up interface. At the top, there's a navigation bar with links for 'Host Login', 'User Login', 'Become Registered User', and 'Help'. The main heading is 'Hello, Guest!'. Below it, a sub-headline says 'You are just a click away from registering yourself.' A note states 'Register here. There is a one time \$150 fee for registering as our certified host.' with a 'Learn more' button. The registration form starts with 'Register Here. We only accept Credit Card Payments.' and includes fields for 'Username', 'Name', 'Email address', 'Password', 'Contact No.', and 'Enter valid phone number'. It also has sections for 'Tell me something about yourself!' and 'Describe yourself. Your profession and more...'. A note at the bottom right of the form area says 'This helps users who book this property to know you much better.' A 'Pay with Card' button is located at the bottom right of the form.

Welcome

Figure 23 Host Signup

6. Payment service for becoming certified User

The screenshot shows a Stripe payment interface overlaid on a 'BookWithMe' page. The background shows a 'Hello Guest!!' message and a 'Become Certified User' button. The Stripe modal is titled 'Become Certified User' and includes a note 'One time membership charge'. It has fields for email ('prayagdeval@gmail.com'), card number ('4242 4242 4242 4242'), expiration ('09 / 19'), and CVV ('123'). It also has a 'Remember me' checkbox and a 'Pay \$45.00' button. The background page has fields for 'Choose your unique username' ('Prayag Deval') and 'Your Email' ('prayagdeval777@gmail.com'). A note at the bottom left of the stripe modal says 'We'll never share your email with anyone else.' and a note at the bottom right says 'This helps users who book this property to know you much better.'

Figure 24 Stripe payment for Registered Hosts

7. Book and explore properties page

The screenshot shows a user interface for booking properties. At the top, there are navigation links: BookWithMe, Manage your Credentials, View your bookings, Manage Bookings, and Show My Previous Bookings. Below this, a greeting "Hello, Prayag Desai !! Start booking properties with us." is displayed. A sub-instruction "Book properties according to availability and also view properties and check it's availability. Explore rental availability and much more..." follows. A main heading "Explore all the properties by location" is centered. Below it, there are two input fields: "City" (with placeholder "Enter the name of City") and "State" (with placeholder "Enter the name of State"). A green "View All properties" button is located below these fields. Further down, a section titled "Explore properties by Check-in and Check-out date and Location" is shown. It includes fields for "City" (placeholder "Enter city name"), "State" (placeholder "Enter the state"), and "Password" (placeholder "Enter the name of the State where you want to search rental"). Below these are three sub-sections: "Check-in Information" (placeholder "MM/DD/YYYY"), "Check-out Information" (placeholder "Enter the check-in date"), and "Accommodation strength per rental" (placeholder "Enter the accommodation strength i.e how many people can be accommodated in your rental property"). A blue "Search for properties" button is at the bottom of this section.

Figure 25 Book and Explore properties

8. Manage User Credentials

The screenshot shows a user interface for managing user credentials. At the top, there are navigation links: BookWithMe, View your bookings, Manage Bookings, Explore and Book Properties, and Show My Previous Bookings. Below this, a heading "Manage your Credentials here.." is displayed. A sub-instruction "Once you change your credentials, you are suppose to log-in again" follows. The form consists of several input fields: "Username" (placeholder "prayagdesai", feedback "Your username is prayagdesai"), "Name" (placeholder "Prayag Desai", feedback "Your name is Prayag Desai."), "Email address" (placeholder "prayagdesai777@gmail.com", feedback "Your email is prayagdesai777@gmail.com."), and "Password" (placeholder "password", feedback "Your password is password."). A blue "Submit" button is located at the bottom right of the form.

Figure 26 Manage user credentials

9. Manage bookings if there are no bookings

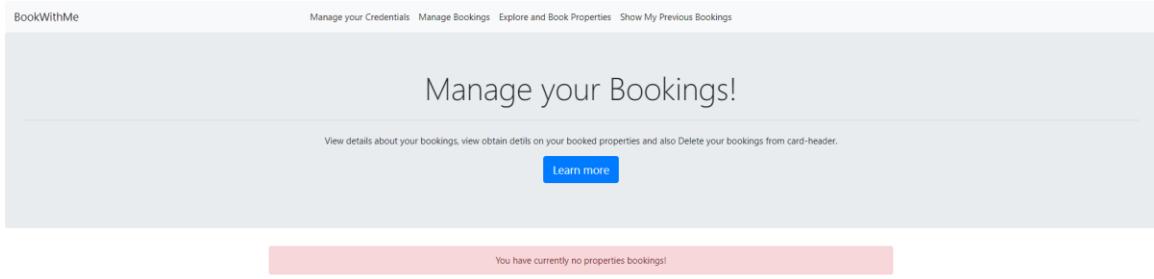
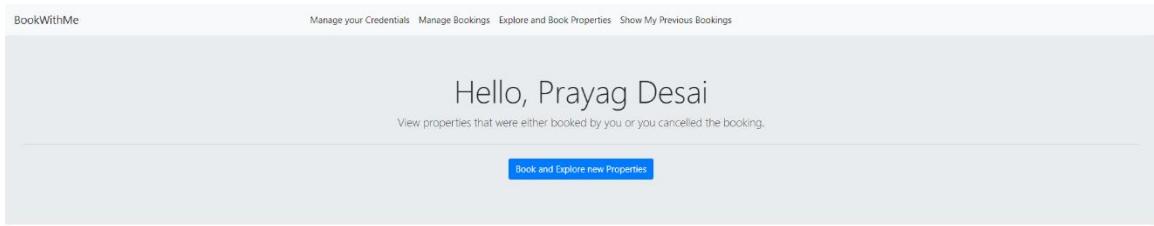


Figure 27 Manage bookings.

10. Previously booked properties of the User



Previously Booked Properties are

You have not booked any properties in the past.
To Book and Explore new properties click here

Figure 28 Previously booked User properties

11. Host registration

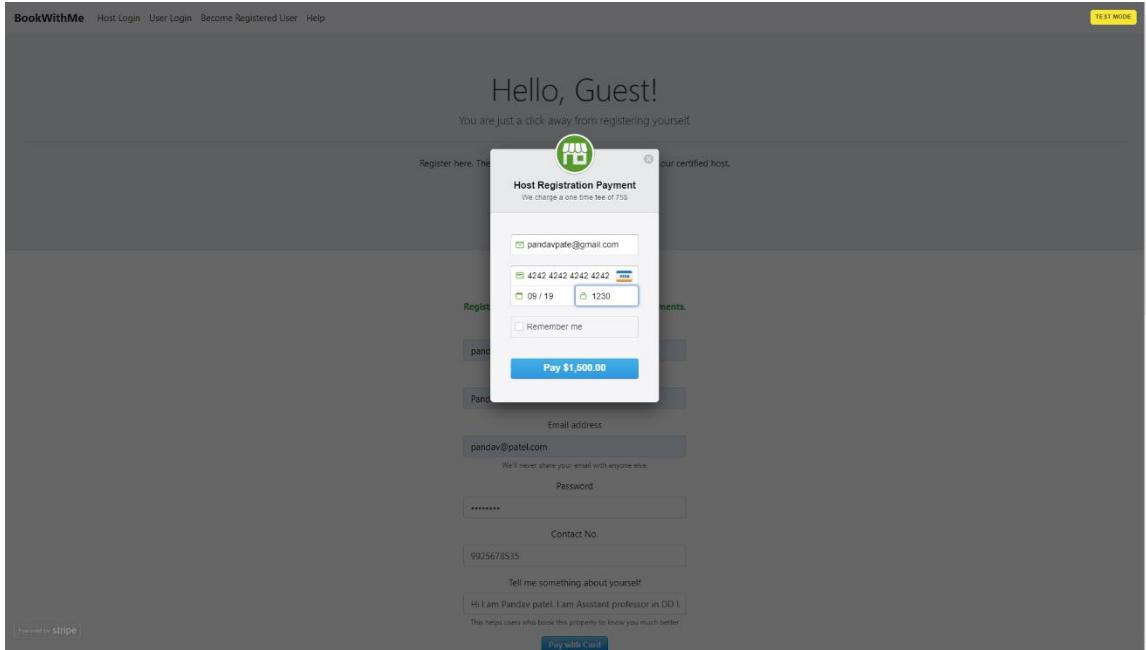


Figure 29 Host Registration

12. Welcome message appears only for the first time after the Host registration

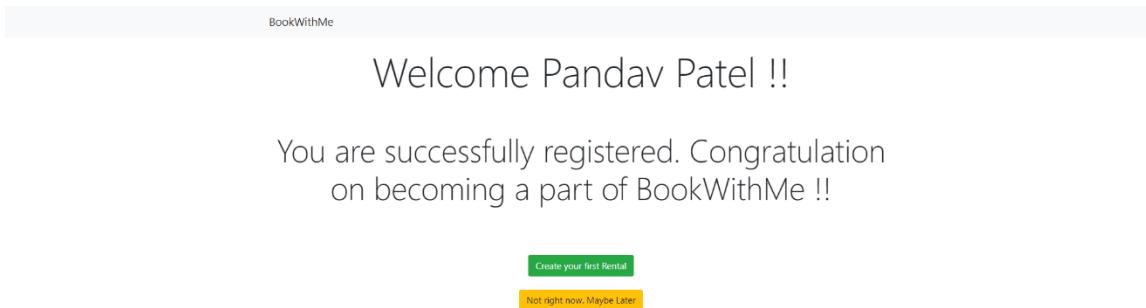


Figure 30 Host Registration successfull

13. Host DashBoard

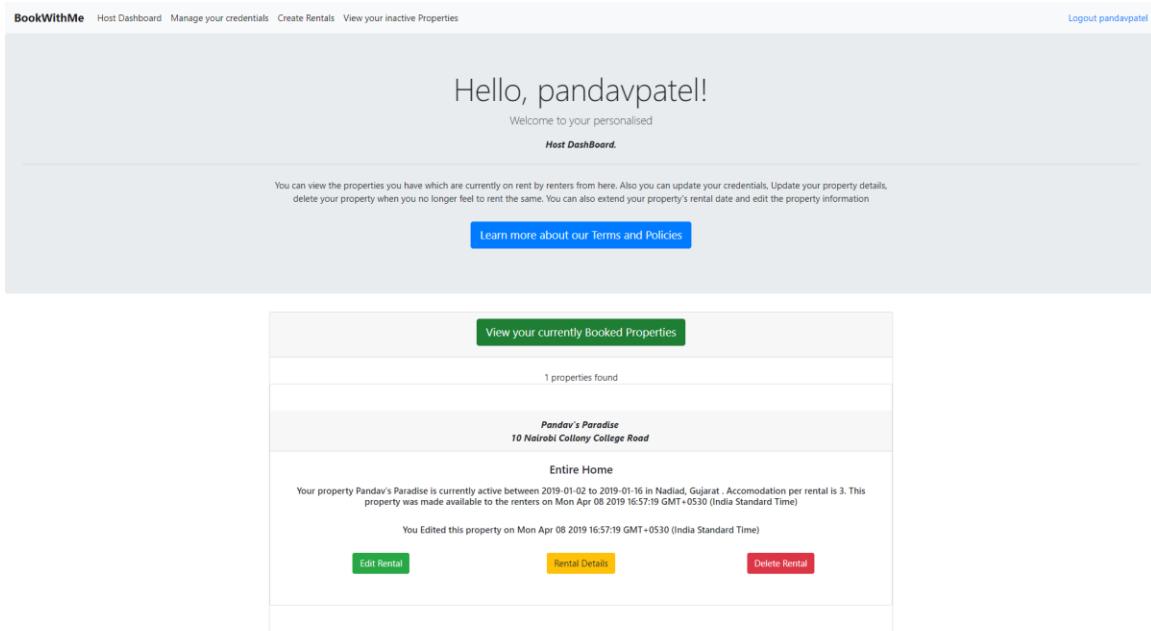


Figure 31 Host dashboard

14. Edit properties. The host can edit all the property information if no user has booked his property

Address

Enter the address where your property is located

City

State

Enter the city name.

Enter the state where your property is located

Property description

Describe your property in brief!

Property specifications

The property is very near to Dharamsingh Desai University. Also fine restaurants and Amusement places are in a very close proximity. This place can be eas

Describe why your property is special to our customers

Amenities

What Amenities does your property have?

Rate per Night of your property

Describe the rate per night for your property. Remember the reasonable your rate is, the more renters will be interested to book this property

Select the type of property

Entire Home
 BookWithMe plus
 Boutique Rooms
 Unique Homes

Select the rental cancellation scheme

free cancellation
 Paid cancellation

Rent your property from

Rent your property to

Accommodation strength per rental

Enter the accommodation strength i.e how many people can be accommodated in your rental property

Figure 32 Editing Host property

15. Property details of the property that has no bookings made by the user and no reviews given

BookWithMe Host Dashboard Manage your credentials Create Rentals Logout

Property Name
Pandav's Paradise

Property description
Fully integrated township. Residential Plots, Villas and more. All the comforts of a city place. Club House, Hospital, School, Yoga and Meditation centre, Banquet Hall, Gymnasium, Swimming pool, etc. Located in a very safe community 8km long boundary wall with 24x7 security. Oasis of greenery and open space. More than 10,000 trees planted. Right next to Mahindra S2Z. Get in touch with your roots. Explore Gujarat with base in Hadhad. Dedicated Concierge Service for out of town residents.



Property Specifications
The property is very near to Dharamsingh Desai University. Also fine restaurants and Amusement places are in a very close proximity. This place can be easily located from road.

BookWithMe Entire Homes



BookWithMe Unique Homes is a new selection of only the highest quality homes with hosts known for great reviews and attention to detail. Every Unique home is visited in person for a 100+ point quality inspection to ensure your comfort. Just look for the Unique Homes badge. Beautiful homes. Exceptional hosts. Verified for quality.

Beautiful homes Every Unique Homes Room is one-of-a-kind, thoughtfully designed, and equipped with a standard set of amenities—whether you're in a private room or have the entire place to yourself. Verified for quality All BookWithMe Unique Homes are visited in person to ensure comfort, consistency, and design. They are checked for 100+ things that guests told us they love, from must-have amenities to the art on the walls.

Rent/Night for Pandav's Paradise is 100

Property Amenities
Aminities

King size Rooms and Bathtub. The property also has fireplace and Swimming Pool. There is also a Kitchen provided to the renter. Enjoy beautiful view from the Balcony with a cup of coffee

Reviews

There are no reviews for this property

Bookings

This property is available from 2019-01-02 to 2019-01-02

Nobody has booked this property

Rent/Night for Pandav's Paradise is 100

Figure 33 Property Details

16. Update host credentials

Update your credentials
 Update your credentials anytime. Once you click the edit button changes will be made..

Your original credentials will be displayed and then you can subsequently change your credentials as you want.

When your credentials are changed you are redirected to the login page. Kindly Log-in with your new credentials

Username	
pandavpatel	
Your username is pandavpatel	
Name	
Pandav Patel	
Your Name is Pandav Patel	
Email address	
pandav@patel.com	
Your email address is pandav@patel.com	
Password	

Your password is password.	
Contact No.	
9925678535	
Your phone no is 9925678535.	
Update	

Figure 34 Update host Credentials

17. Make property unavailable by Host. The host can delete property which is not booked by any other user

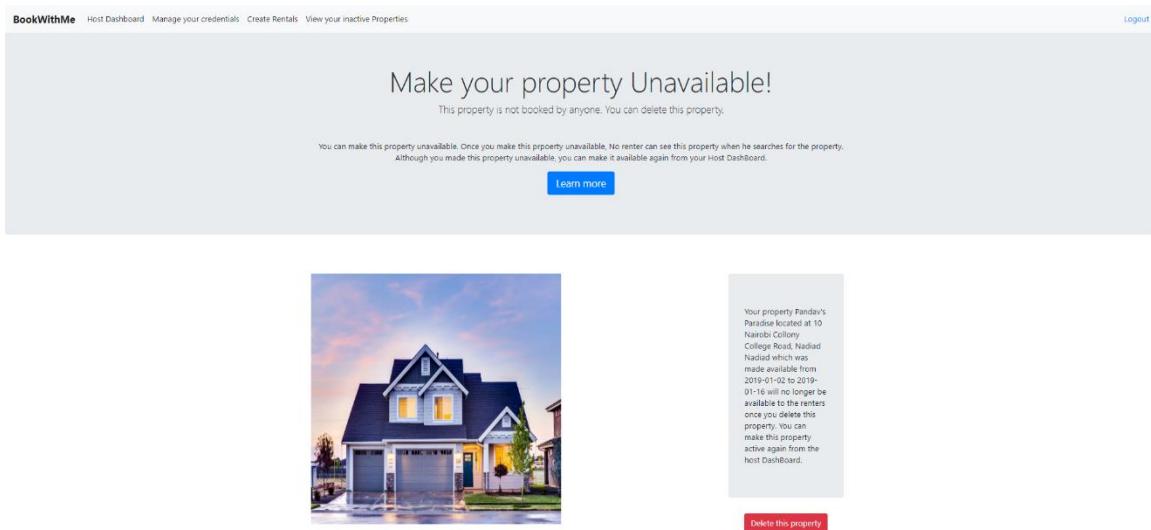


Figure 35 Make property unavailable

18. Properties available based on check-in, check-out dates, accommodation per rental and location

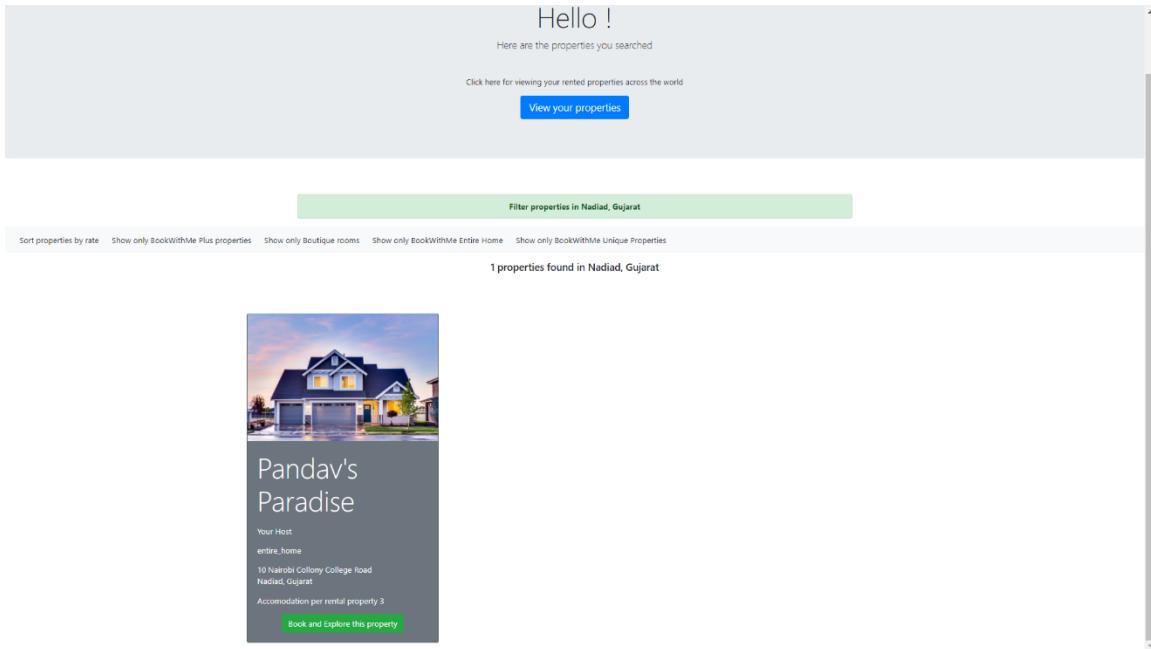


Figure 36 List properties.

19. Book property

The screenshot shows the booking interface for 'Pandav's Paradise'. It starts with a section titled 'BookWithMe Entire Homes' featuring the 'BookWithMe House' logo and a brief description of the service. Below this is a section titled 'Aminities' with a note about king-size rooms, bathtubs, fireplaces, and swimming pools. There is a 'Book this property' button with date input fields for check-in (2019-01-03) and check-out (2019-01-05). Further down are links for 'Directions' and 'Reviews'. At the bottom, it says 'There are no reviews for this property' and 'Rent/Night for Pandav's Paradise is 100'.

Figure 37 Book property

20. Confirm booking

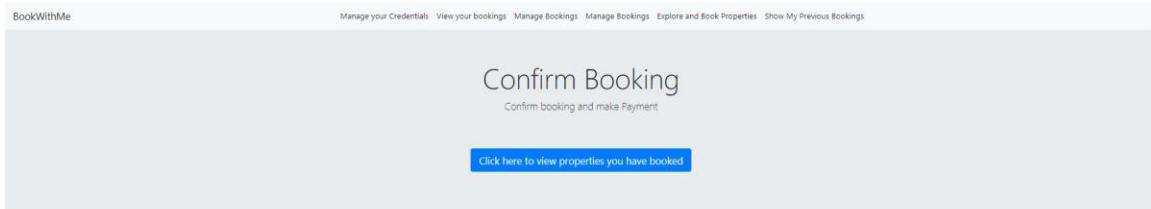


Figure 38 Confirm booking

21. Make payment

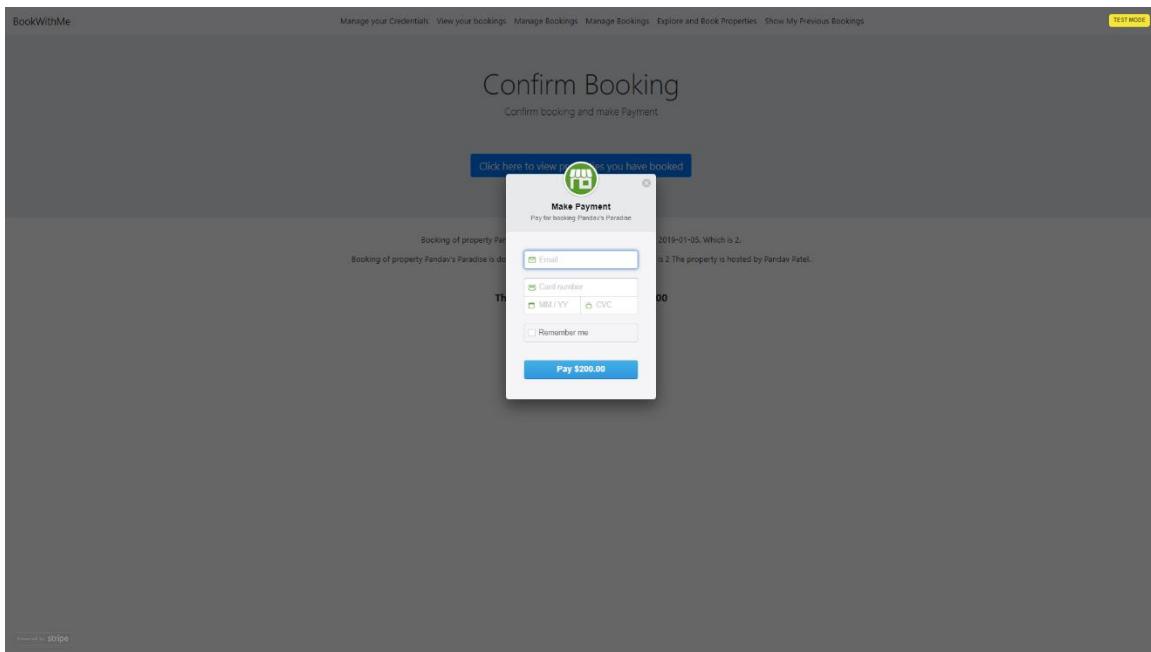


Figure 39 Make Payment for booking property

22. Booking successful

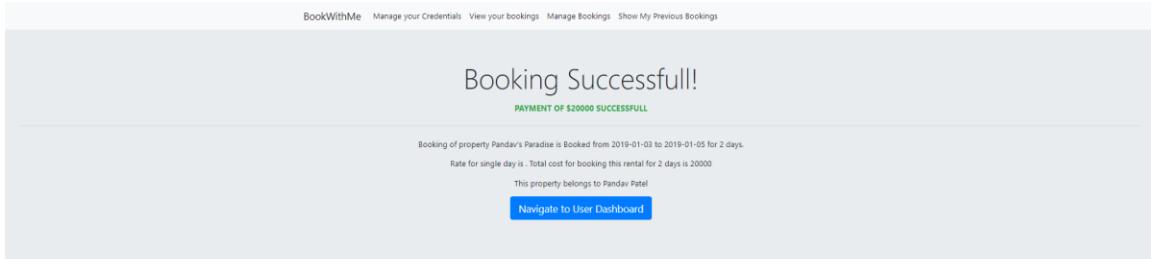


Figure 40 Booking Successful

23. User Dashboard shows the property just booked by the user

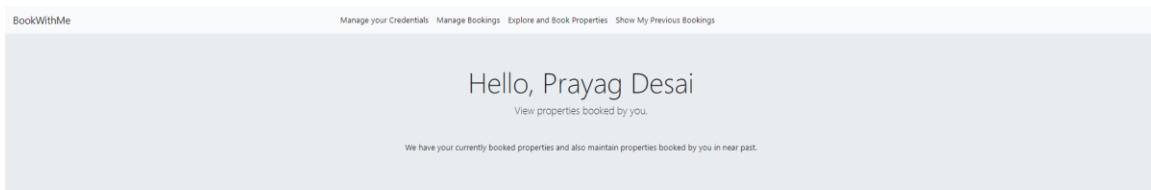


Figure 41 User dashboard

24. User can book rate property if and only if the user has stayed in the property after he made booking

BookWithMe

Manage your Credentials Manage Bookings /user-dashboard User Dashboard Explore and Book Properties Show My Previous Bookings

It's time for Feedback, Prayag Desai

You can always help our hosts by giving honest feedback about your experiences

Give a detailed feedback about the property and hosts. Describe your experience in brief and also provide our hosts insight on how they can help make your stay a memorable one. Also this reviews will help fellow hosts to select properties on rent using your valuable reviews.

We really appreciate your honest feedback and ratings!

[Learn more](#)

Describe your experience in brief

Experience was awesome and I really enjoyed the hospitality

Tell us about your experience about this property.

General Feedback about the property

Property was okay, it can still be improved. Light bulbs were not working properly in bathroom

Giving us insight of how you feel about the property. Was the property good? If not also specify what you didn't like about property

Feedback about the Host

Host was great, had excellent time with him

Tell us about your experience with host of this property.

How would you overall rate your experience about the rental and the host combined?

4 / 5

[Submit your Ratings and Feedback](#)

Figure 42 Ratings and Review

25. Review made and by the user is reflected in the property details and not deleted by Host

BookWithMe Unique Homes is a new selection of only the highest quality homes with hosts known for great reviews and attention to detail. Every Unique home is visited in person for a 100% quality audit prior to receive your comfort. Just look for the Unique Homes badge.

Beautiful homes Every Unique Homes Room is of a+ kind, thoughtfully designed, and equipped with a standard set of amenities—whether you're in a private room or have the entire place to yourself.

Verified for quality All BookWithMe Unique Homes are visited in person to ensure comfort, consistency, and design. They are checked for 100+ things that guests told us they love, from must-have amenities to the art on the walls.

Property Amenities

Aminities

King size Rooms and Bathtub. The property also has fireplace and Swimming Pool. There is also a Kitchen provided to the renter. Enjoy beautiful view from the Balcony with a cup of coffee

Reviews

Experience was awesome and I really enjoyed the hospitality. Property was okay. It can still be improved. Light bulbs were not working properly in bathroom. Host was great, had excellent time with him

, Prayag Desai rates this property 4

— In this property between Thu Jan 03 2019 05:30:00 GMT+0530 (India Standard Time) and Sat Jan 05 2019 05:30:00 GMT+0530 (India Standard Time)

Bookings

This property is available from 2019-01-02 to 2019-01-16

Check-In Date	Check-Out Date	User Name	Date when Property is Booked
2019-01-03	2019-01-05	Prayag Desai	Mon Apr 08 2019 17:27:45 GMT+0530 (India Standard Time)

Rent/Night for Panday's Paradise is 100

Figure 43 Review and Booking

25. Host cannot make property unavailable if the host has not vacated the property

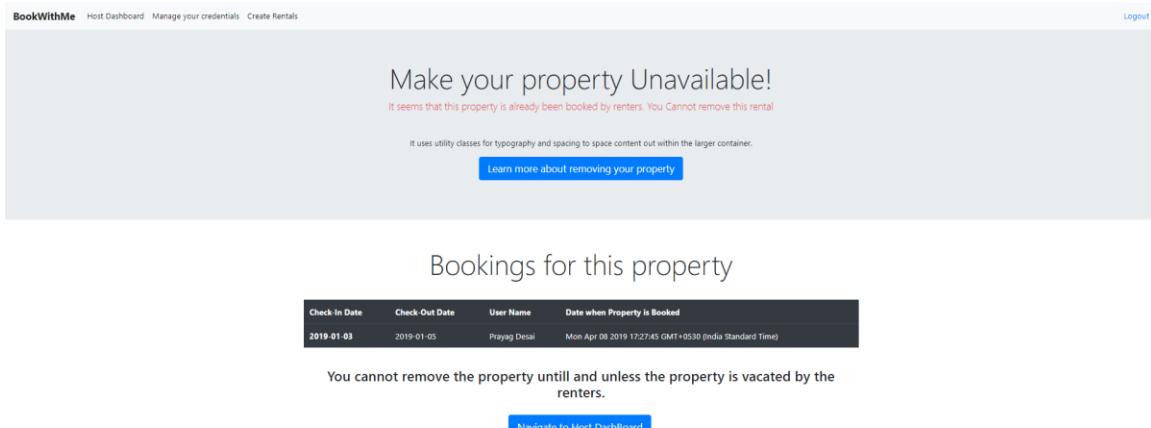
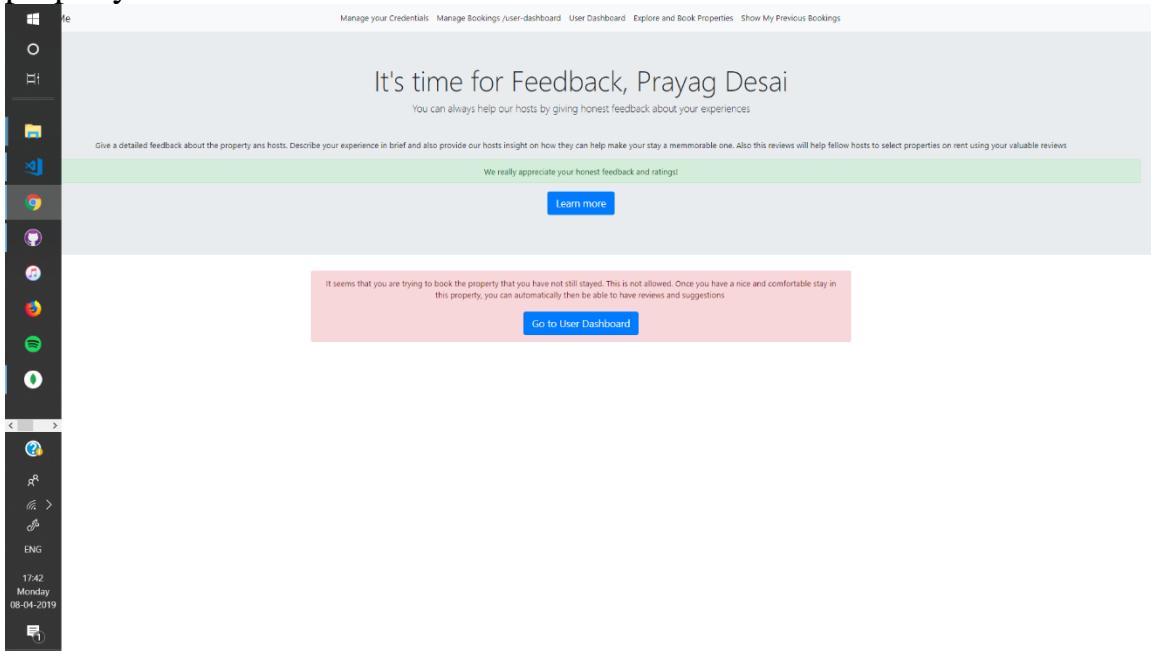


Figure 44 Making property unavailable

26. User cannot rate the property after booking if user has not stayed in the property



27. Deleting Paid booking

BookWithMe [Manage your Credentials](#) [View your bookings](#) [Cancel Booking](#) [Explore and Book Properties](#) [Show My Previous Bookings](#)

Rental Booking Details

Paid Cancellation.

This property can be cancelled only on paying of cancellation fee. The standard Cancellation fee is 25% of the Rate of the rental. Once you cancel this booking, you will no longer be able to stay in this property. You can still book this property again if it is available.

Nihal's Paradise

6 Nairobi Colony College Road
Nadiad, Gujarat

Rate per night of this rental is 120

This property is made available from 2019-04-11 to 2019-05-09
You booked this property from to 2019-01-03 on Mon Apr 08 2019 17:27:45 GMT+0530 (India Standard Time)

The transaction id for this booking is 5ca837394943d5413a00eb229

The cancellation charge for this booking is 30

[Navigate me to User Dashboard](#)

[Manage Bookings](#)

[Pay with Card](#)

BookWithMe [Manage your Credentials](#) [View your bookings](#) [Cancel Booking](#) [Explore and Book Properties](#) [Show My Previous Bookings](#) TEST MODE

Rental Booking Details

Paid Cancellation.

This property can be cancelled only on paying of cancellation fee. The standard Cancellation fee is 25% of the Rate of the rental. Once you cancel this booking, you will no longer be able to stay in this property. You can still book this property again if it is available.

Cancel Booking
Cancel this booking

Name: Nihal's Paradise

Rate: ₹ 120

This property is made available from 2019-04-11 to 2019-05-09
You booked this property from to 2019-01-03 on Mon Apr 08 2019 17:27:45 GMT+0530 (India Standard Time)

The transaction id for this booking is 5ca837394943d5413a00eb229
The cancellation charge for this booking is 30

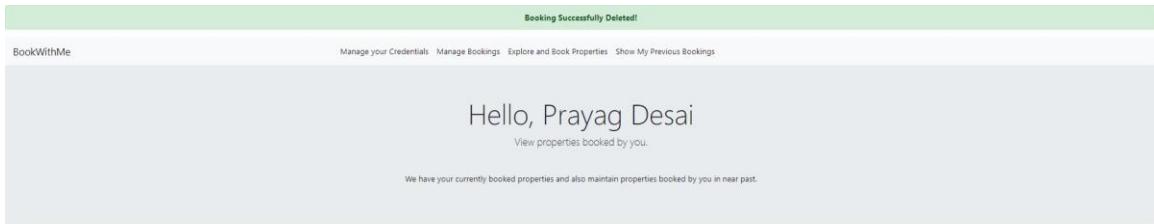
[Pay \\$30.00](#)

[Navigate me to User Dashboard](#)

[Manage Bookings](#)

[Pay with Card](#)

Powered by 



Currently Booked Properties are

Nihal's Paradise

Nihal's Paradise
6 Nairobi Colony College Road

You booked this property on Mon Apr 08 2019 17:41:31 GMT+0530 (India Standard Time) from 2019-04-11 to 2019-04-13

[View Property Details](#) [Rate this property](#)

CHAPTER 7

CONCLUSION AND FUTURE EXTENSION

7.1 Conclusion

This web application provides a very clean and effective solutions for user to book properties and host to rent their properties using the platform. This application also targets clean design of displaying and booking properties to its users, and the UI is design such that the users will not feel hesitated or confused to use this application. Also, there is a powerful and feature rich UI is designed and functionalities are implemented so that the hosts are easily and flexibly able to rent their properties to the users without any hustle. Bookings and Check-in and Check-out is implemented in such an efficient fashion which makes this application very effective to use.

7.2 Discussion

7.2.1 Self-Analysis of Project Viabilities

According to us, this project is completed with the primary functionalities as specified earlier but then again there is lot more than this which can be done. The project is well capable to handle the given job for some particular task but not all of them. So then it is a challenge to further develop it in to a well flagged software as it was challenge to develop up to this stage.

7.2.2 Problem Encountered while development

There were many problems encountered during the design and the development phase of the project.

- Problem regarding web application layout.
- Problem to maintain database.
- Problem regarding multiple synchronous requests to the server.
- Problem regarding handling Async code in Node.js
- Problem regarding parsing some JavaScript content in EJS templating engine.
- Problem in deciding the deadline to implement certain features in the project

7.2.3 Summary of Project Work

I have completed our project work using software engineering and system analysis and design approach. We have considered all the suggestions given by my internal guide and

have also followed the desired instructions on documentation and coding standards stated by the project guides. I have done work with pre-planned scheduling related with time constraints and result oriented progress in project development.

7.3. Limitations

The System lack Authorization of the admin, hosts and users. Also, the System lacks feature for uploading images by the host of their property. The system lacks mechanism to suggest user new properties to book and perform analytics on the user generated data.

7.4 Future Extension

There are some functionalities that can be implemented later which can be very important besides core functionalities that are implemented in this project. This project can be further extended by adding following functionalities

- Implementing Recommendation System which recommends user the property that suits the taste for the user based on the user's previous bookings.
- Implementation of google analytics so that the platform come to knows the trends for booking and hosting of the properties across several locations.
- Image uploading feature by Registered Hosts so that the users can have a view of the property that they have to book.
- Enhancement in the front-end and UX and UI design

7.5 Bibliography

- **JavaScript:** <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- **Node.js:** <https://nodejs.org/en/docs/>
- **Express:** <https://expressjs.com/en/api.html>
- **Bootstrap:** <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
- **CSS:** <https://www.w3schools.com/w3css/>
- **jQuery:** <https://api.jquery.com/>
- **EJS:** <https://ejs.co/>
- **Stripe Online Payment Gateway:** <https://stripe.com/docs/quickstart>
- **MongoDB:** <https://docs.mongodb.com/>
- **Coding Practices for Nodejs Express project** <https://blog.risingstack.com/node-js-best-practices/>