**Project Name: HOSTEL FINDER**

**Project Member:**

**Mohd Zeya Iqbal Khan 210543181051**

**Pawar Vishal Mahesh 210543181070**

**Abstract:**

This document is meant to delineate the features of Hostel Finder, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other. The Online Hostel Booking System (OHBS) for all India hostels web application is intended to provide complete solutions for vendors as well as customers through a single get way using the internet. It will enable vendors to list their hostel(s), customers to browse through the hostels rooms and book them online without visiting the hostel physically. The administration module will enable a system administrator to approve and reject requests for new hostel and maintain various lists of hostel category.

This system allows the customers to search hostel in a particular city with room availability and price and hostel owner to register their hostel over the internet.

Now a days it is very difficult for an individual especially students and graduates who start their carrier in their respective field to move from their native place to the place where they have to work or complete their studies. There are so many difficulties like lack of knowledge of the respective area, language boundary, locality etc.

So in order to overcome this we have introduced a web based project that can help an individuals to choose a hostel with respect to their preference and convenience prior going to the unknown state or city in India.

**Implementation Technologies:**

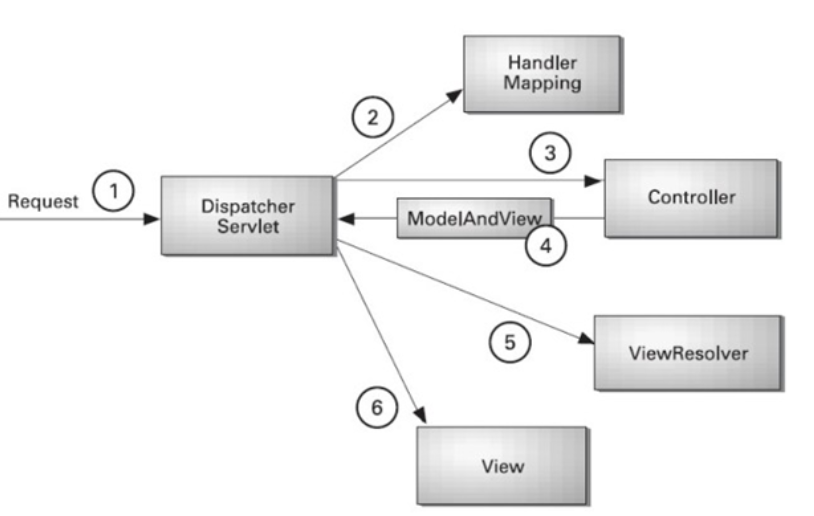
1. **Spring MVC :**

A Spring MVC is a Java framework which is used to build web applications. It follows the Model-View-Controller design pattern. It implements all the basic features of a core spring framework like Inversion of Control, Dependency Injection.

A Spring MVC provides an elegant solution to use MVC in spring framework by the help of DispatcherServlet. Here, DispatcherServlet is a class that receives the incoming request and maps it to the right resource such as controllers, models, and views.

**1.1 Features of Spring Framework:**

* **Model** - A model contains the data of the application. A data can be a single object or a collection of objects.
* **Controller** - A controller contains the business logic of an application. Here, the @Controller annotation is used to mark the class as the controller.
* **View** - A view represents the provided information in a particular format. Generally, JSP+JSTL is used to create a view page. Although spring also supports other view technologies such as Apache Velocity, Thymeleaf and FreeMarker.
* **Front Controller** - In Spring Web MVC, the DispatcherServlet class works as the front controller. It is responsible to manage the flow of the Spring MVC application.



* As displayed in the figure, all the incoming request is intercepted by the DispatcherServlet that works as the front controller.
* The DispatcherServlet gets an entry of handler mapping from the XML file and forwards the request to the controller.
* The controller returns an object of ModelAndView.
* The DispatcherServlet checks the entry of view resolver in the XML file and invokes the specified view component.

**1.2 Advantages of Spring MVC Framework :**

Let's see some of the advantages of Spring MVC Framework:-

* **Separate roles** - The Spring MVC separates each role, where the model object, controller, command object, view resolver, DispatcherServlet, validator, etc. can be fulfilled by a specialized object.
* **Light-weight** - It uses light-weight servlet container to develop and deploy your application.
* **Powerful Configuration** - It provides a robust configuration for both framework and application classes that includes easy referencing across contexts, such as from web controllers to business objects and validators.
* **Rapid development** - The Spring MVC facilitates fast and parallel development.
* **Reusable business code** - Instead of creating new objects, it allows us to use the existing business objects.
* **Easy to test** - In Spring, generally we create JavaBeans classes that enable you to inject test data using the setter methods.
* **Flexible Mapping** - It provides the specific annotations that easily redirect the page.

1. **The JDBC Template**

The central class of the Spring JDBC abstraction framework is the **JdbcTemplate** class that includes the most common logic in using the JDBC API to access data, such as handling the creation of connection, statement creation, statement execution, and release of resource. The**Jdbc-Template**class can be found in the **org.springframework.jdbc.core**package.

The **JdbcTemplate** class instances are thread-safe once configured. A single **JdbcTemplate** can be configured and injected into multiple DAOs.

We can use the **JdbcTemplate** to execute the different types of SQL statements. **Data Manipulation Language** (**DML**) is used for inserting, retrieving, updating, and deleting the data in the database such as **SELECT**, **INSERT**, or **UPDATE** statements

**2.1** **MySQL**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

**Features of MySQL:**

* **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

* **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment.

* **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything.

* **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

* **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

1. **Hardware and Software Requirements (Minimum):**

**Hardware:**

1. Intel i3 processor 3rd generation or later / AMD Ryzen 200 2nd generation or later

2. 2 GB ddr3 ram.

3. Windows 7 Home edition or later.

4. 200 GB Sata HDD Space

5. Data Connection 200 kbps

**Software:**

1. Eclipse 4.7 Oxygen
2. MySQL 5.7 with Workbench 8.0
3. Google Chrome version 79.0
4. Apache Tomcat Server 8.5
5. Maven Dependencies
6. **ER Diagram:**

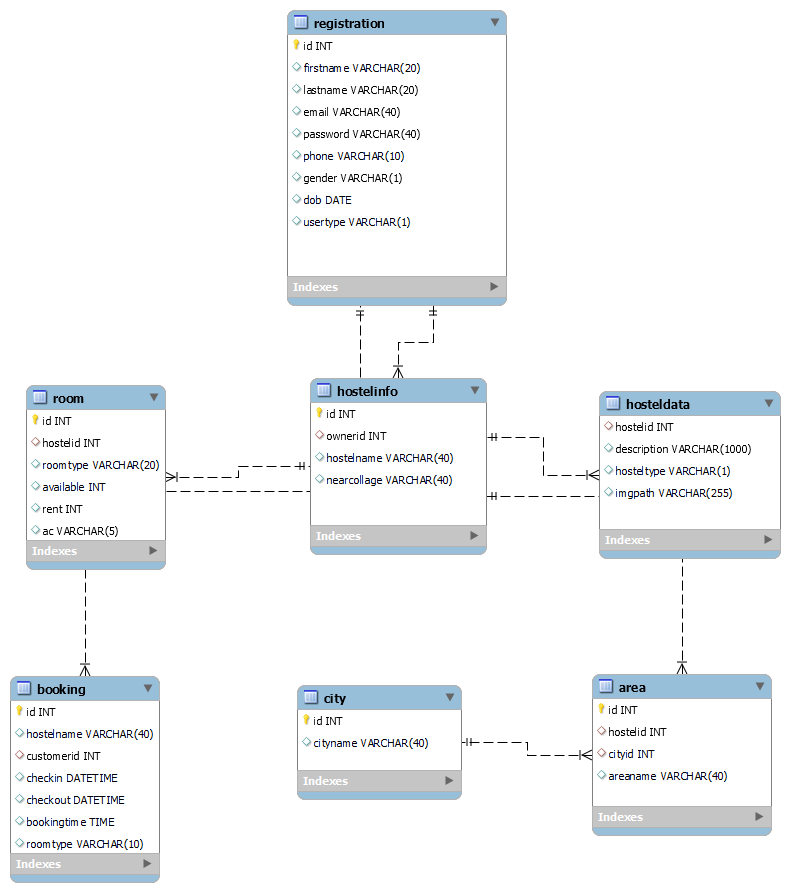


Figure 1: ER Diagram

1. **Table Structures (Database Name : hostelfinder):**
2. **Table name : registration**

**Column name Type**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | int | NO | PRI |  | auto\_increment |
| firstname | varchar(20) | YES |  |  |  |
| lastname | varchar(20) | YES |  |  |  |
| email | varchar(40) | YES |  |  |  |
| password | varchar(40) | YES |  |  |  |
| phone | varchar(10) | YES |  |  |  |
| gender | varchar(1) | YES |  |  |  |
| dob | datetime | YES |  |  |  |
| usertype | varchar(1) | YES |  |  |  |

1. **Table name: hostelinfo**

**Column name Type**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | int | NO | PRI |  | auto\_increment |
| ownerid | int | YES | MUL |  |  |
| hostelname | varchar(40) | YES |  |  |  |
| nearcollage | varchar(40) | YES |  |  |  |

1. **Table name: hosteldata**

**Column name Type**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| hostelid | int | YES |  |  |
| description | varchar(1000) | YES |  |  |
| hosteltype | varchar(1) | YES |  |  |
| imgpath | varchar(255) | YES |  |  |

1. **Table name: area**

**Column name Type**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | int | NO | PRI |  | auto\_increment |
| hostelid | int | YES | MUL |  |  |
| cityid | int | YES | MUL |  |  |
| areaname | varchar(40) | YES |  |  |  |

1. **Table name : city**

**Column name Type**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | int | NO | PRI |  | auto\_increment |
| cityname | varchar(40) | YES |  |  |  |

1. **Table name : room**

**Column name Type**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | int | NO | PRI |  | auto\_increment |
| hostelid | int | YES | MUL |  |  |
| roomtype | varchar(20) | YES |  |  |  |
| available | int | YES |  |  |  |
| rent | int | YES |  |  |  |
| ac | varchar(5) | YES |  |  |  |

1. **Table name : booking**

**Column name Type**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | int | NO | PRI |  | auto\_increment |
| hostelname | varchar(40) | YES |  |  |  |
| customerid | int | YES | MUL |  |  |
| checkin | datetime | YES |  |  |  |
| checkout | datetime | YES |  |  |  |
| bookingtime | time | YES |  |  |  |
| roomtype | varchar(10) | YES |  |  |  |

1. **UML Diagrams:**

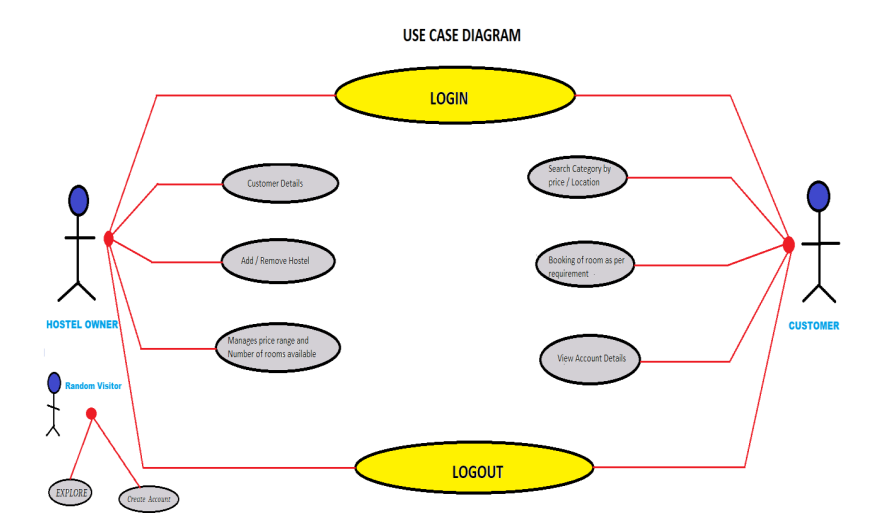


Figure 2: Use Case

1. **End to End Flow of Application:**

**User:**

* 1. User will login to the portal or will have to register if he is not a registered user.
  2. The user can login as a **Customer** or as a **Hostel Owner**. The login page is common for both customer and as well as hostel owner .
  3. On the **Sign Up** page a user can select if he/she is a customer or a hostel owner and fill the sign up form accordingly.
  4. The default user type of the customers is **“1”** and for the hostel owner is **“2”** in the database **hostelfinder** .
  5. As a hostel owner an individual can add the hostel information i.e. hostel name , location , near college , hostel type , room availability , rent , hostel description and hostel image . Once the hostel is added by the hostel owner it will be displayed on the home page and from there a customer can book that particular hostel by clicking **Know More** button . After clicking that button a booking page will be displayed with the hostel description of that particular hostel including **CheckIn** , **CheckOut** , **BookingTime** and **Guests** . The customer can book the hostel according to their desire.
  6. While loging in as customer , all the information of that customer will be displayed .
  7. A user can also access **ContactUs** and **AboutUs** pages .

**Thank You!**