|  |
| --- |
|  |
| HOTEL RESERVATION |
| Web delight |

|  |
| --- |
|  |

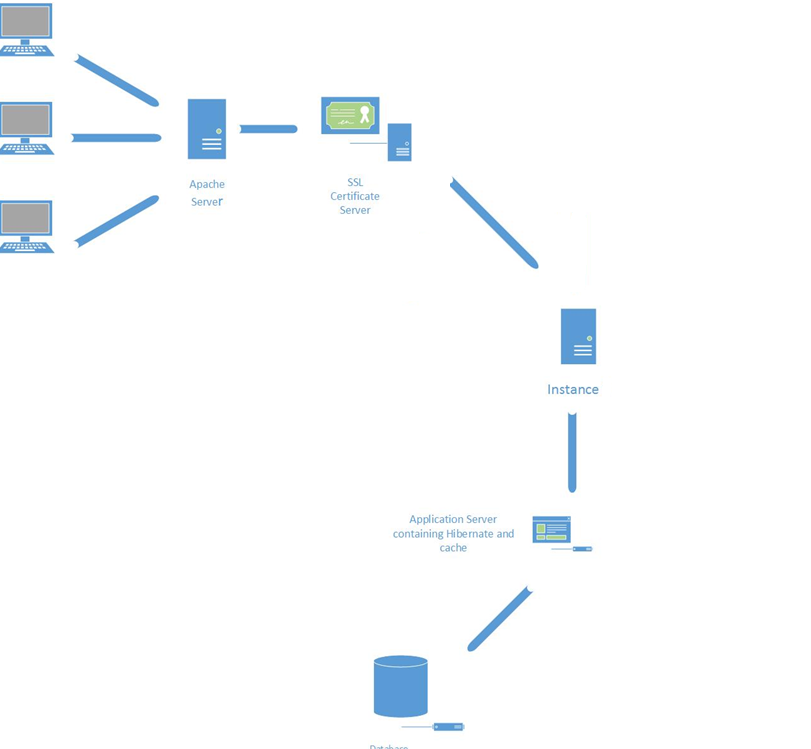
# Architectural Diagram:

DAYANA

SMITHA

SUSHMA

Hamsa



# Modules

## Apache Tomcat Server

Apache Tomcat Server installed and configured for local host machine. Apache Tomcat Server hosts at port 80. We have configured apache tomcat server with two other modules for SSL Encryption.

SSL Certificates are small data files that digitally bind a cryptographic key to an organization’s details. When installed on a web server, it activates the padlock and the https protocol (over port 443) and allows secure connections from a web server to a browser.

## SSL Encryption

Apache Tomcat server is configured to use self-generated SSL Certificate on port 443. Any https:// request will be served on port 443. Configured httpd-ssl.conf file with certificate and key file for SSL encryption. Then in Virtual Host definition for SSL, we have added every JKMount definition related to two different instances. This will redirect 443 port to Hotel Reservation folder in different instances.

This means SSL connections will be handled by Apache, then punted over to Tomcat with AJP. This greatly simplifies configuration, as all you need in your Tomcat config is the AJP connector.

## Instances

Apache is configured with two instances of tomcat with AJP connector on port 8009 and port 8010. This is used to have a non SSL connection between apache and multiple instances. This two instances are having same copy of web application in both webapps folder. The configuration is made in such a way that two instances are workers of apache server and the request will be sent to any of server according to round robin algorithm.

## Application Server

* **Frontend**: Plain CSS, HTML, Ajax, and JavaScript
* **ORM tool**: Hibernate 4
* **Distributed Cache**: Memcache
* **Web services**: Jersey 2.17 Restful
* **IDE**: Eclipse
* **Compression**: HTTP Compression

We have used Hibernate 4 as object relational mapping which is JAVA framework. Hibernate is concerned with data persistence as it applies to relational databases (via JDBC).

We have used Restful web services for POST request .

HTTP compression is a capability to provide greater transmission speeds between the web clients and web servers and make better use of available bandwidth. We have implemented in our project as one of the core feature which can increase the transmission speed.

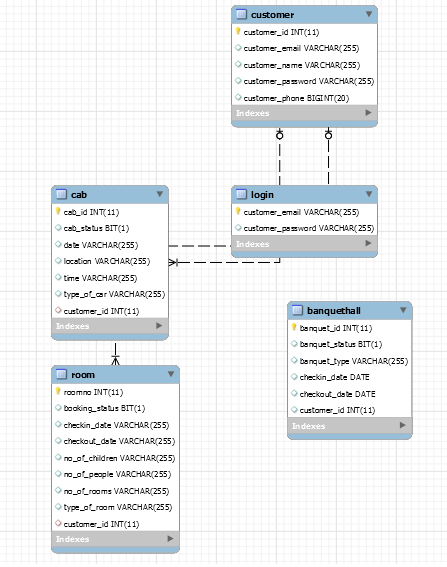
Memcache is used as our distributed cache for web application. Memcache is Free & open source, high-performance, distributed memory object caching system intended for use in speeding up dynamic web applications by alleviating database load.

Front end is developed using HTML and simple CSS file. We have used some HTML 5 tags in our website to make the front end looks more elegant in design. We have downloaded some templates from internet for a better CSS. Then we have modified the CSS according to website requirement. JavaScript as well as jQuery is used as client side programing to accomplish some functionalities of webpages.

Jersey RESTful Web Services framework is open source, production quality, framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) JAX-RS is a very well designed API that makes mapping HTTP requests to methods, extracting parameters from various parts of a HTTP request, handling content negotiating, and many other low level tasks very easy.

## Database

**Tool Used**: MySQL



# Mandatory Services:

## Registration

A User can register to our Hotel. To Register he has to provide all his details like name email id, Phone number, password. We are inserting his detail into database and providing him with a unique customer id.

## **2.Room Booking**

A Customer can book a room. Here we are inserting data into database. When user submit the information about date of check-in and checkout , number of adults and children . We are submitting our data with POST request and allocating him a room.

## 3.Banquet Hall Booking

A Customer can book a Banquet Hall. Here we are inserting data into database. When user submit the information about date of check-in and checkout and type of hall. We are submitting our data with POST request and allocating him a Banquet Hall.

## 4. Cab Booking

A Customer can book a Cab. Here we are inserting data into database. When user submit the Pick-up location and time, we are submitting our data with POST request and allocating him a Cab.

# Functionalities:

1. **Dining Information**: Hotel dining information is provided in this page .All the details about Breakfast , Lunch, Dinner has been provided.

2. **Sight Seeing:** A user can see what places he can visit in Dallas. All the necessary information about the places has been provided**.**

3. **Search:** Search page shows the customer details residing in our hotel.

# Problems and how we solved it

1.Hibernate only works when there is exact mapping from the DB table to a class. It does not allow new properties in the class. DB Table structures had to be changed to fit this.

2.We tried to have SSL certificate to both the instances but AJP port was not forwarding the request from Apache to SSL port of tomcat so we installed the encryption to Apache server which will connect to tomcat instances.

3.Keeping track of standard design for all the pages to preserve the similarity between them was a challenge. Constant check with the page structure (tags and links) as the length grows was a difficult task but not impossible.

4.Ideas kept changing for better outcome. Developing a dynamic prototype at each step was challenging but it paid off.