

CS5200 Project Report – Hotel Reservation System

Team Members

- ❖ Deepen Mehta
- ❖ Bhavin Vora
- ❖ Mitva Mehta

Link to project on GitHub

<https://github.com/mehtadeepen/HRS/tree/master/HRS>

Problem Description

We are seeking solution to the problem of looking up and reserving hotels online. Users want to reserve hotels for the places they are planning to visit around the world.

They would like to be able to look up hotels in a particular city, for particular dates and within a price range. Users should be able to see the details of the hotel, room-types, facilities, and book the hotel online. They might also need view the past reservations and also cancel a reservation they made. Users must also be able to extend the number of days of a reservation they make.

Once a user visits a hotel he should be able to submit a review about the hotel services and facilities. A like button feature for the hotel can enable users to like the hotel. If a user finds the reviews of another user to be useful then the user should be able to follow other users.

Proposed Solution

As a solution we are providing an online hotel reservation system. Our system will let users search for hotels in a particular city and according to the specified dates it will show the availability of the hotel. They can view details of the hotels including hotel amenities, room types, room rates, availability, ratings, reviews and photos.

Users can reserve a hotel online or cancel an existing reservation. They can also update their ongoing reservation. For making the system better we have provided a feature that enables the users to create their own profile. The users can add hotels to their favorites list. The 'like' button feature enables users to 'like' a hotel.

If a user finds the reviews of another user to be useful then the user should be able to follow other users. The users can also share their experience by providing feedback and

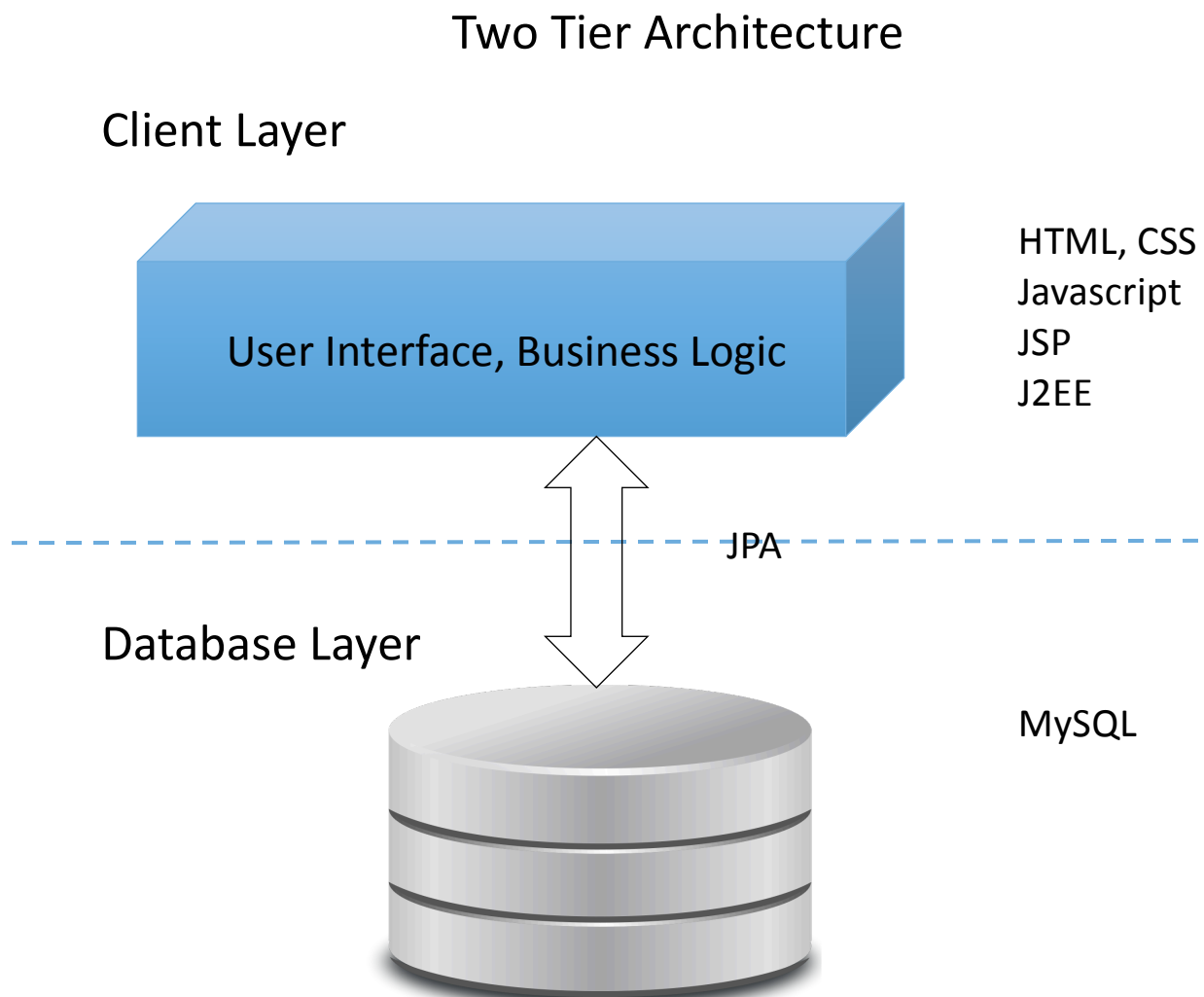
reviews. The rating feature is useful for users trying to book a hotel as it enables them to take better decisions.

Project Architecture

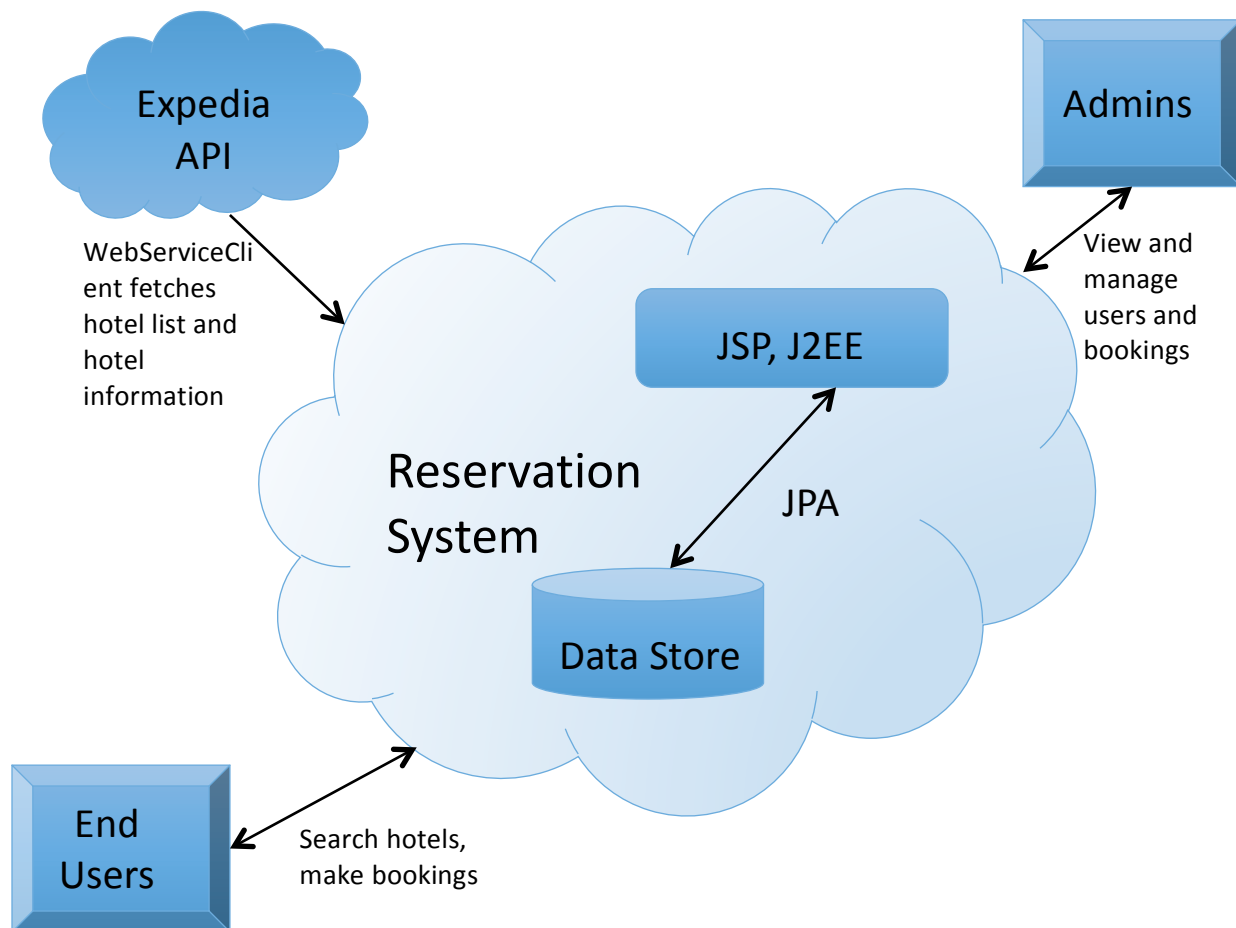
We have used the two-tier architecture for our project.

The client layer consists of the User Interface and Business logic, and the data layer consists of our database. Object Relational Mapping is used to fetch data from the data layer to be used in the client layer.

The diagram below shows the two-tier architecture of our project:



The diagram below describes how the various components in our system depend on each other:



Our system fetches hotel data from the Expedia API using a WebServiceClient, and displays this data on our system. Further user interactions with our system will save relevant parts of this data into a data store from where we again fetch data to be displayed in certain parts of our system.

End users interact with this system to search hotels for some location, view information of specific hotel, make bookings, write reviews, and follow other users of our system.

Administrators use the system to manage users and to view and manage bookings for users.

APIs Used

In order to obtain a list of hotels for any given location, and detailed hotel information about a particular hotel, we used the Expedia API available [here](#).

Expedia API can return response in both formats, XML as well as JSON. We used JSON as the response format.

Expedia API provides an extensive list of methods such as Hotel List, Room Images, Ping Request, Payment Types, Geo Functions, etc. Out of these, the methods that we used to fetch data for our system are:

1. Hotel List method

This method can be used to return hotels with available rooms for a provided date range, or to return a list of all active properties within the specified location without any availability information

The request parameters that we used to make a request using this method are:

- arrivalDate
- departureDate
- destinationString
- numberOfResults

The response consists of parameters like hotelId, name, address, city, stateProvinceCode, hotelRating, shortDescription, highRate, lowRate, imageUrl, etc

2. Hotel Information method

This method lets you retrieve any combination of hotel and room amenities, property information, suppliers, property images, and more for a single hotelId.

We only need a single request parameter to make a request to this method, that is:

- hotelId

Based on the hotelId that we send in the request, this method will return a response that contains parameters like name, address, city, stateCode, provinceCode, countryCode, hotelRating, checkInTime, checkOutTime, propertyDescription, hotelImages, etc

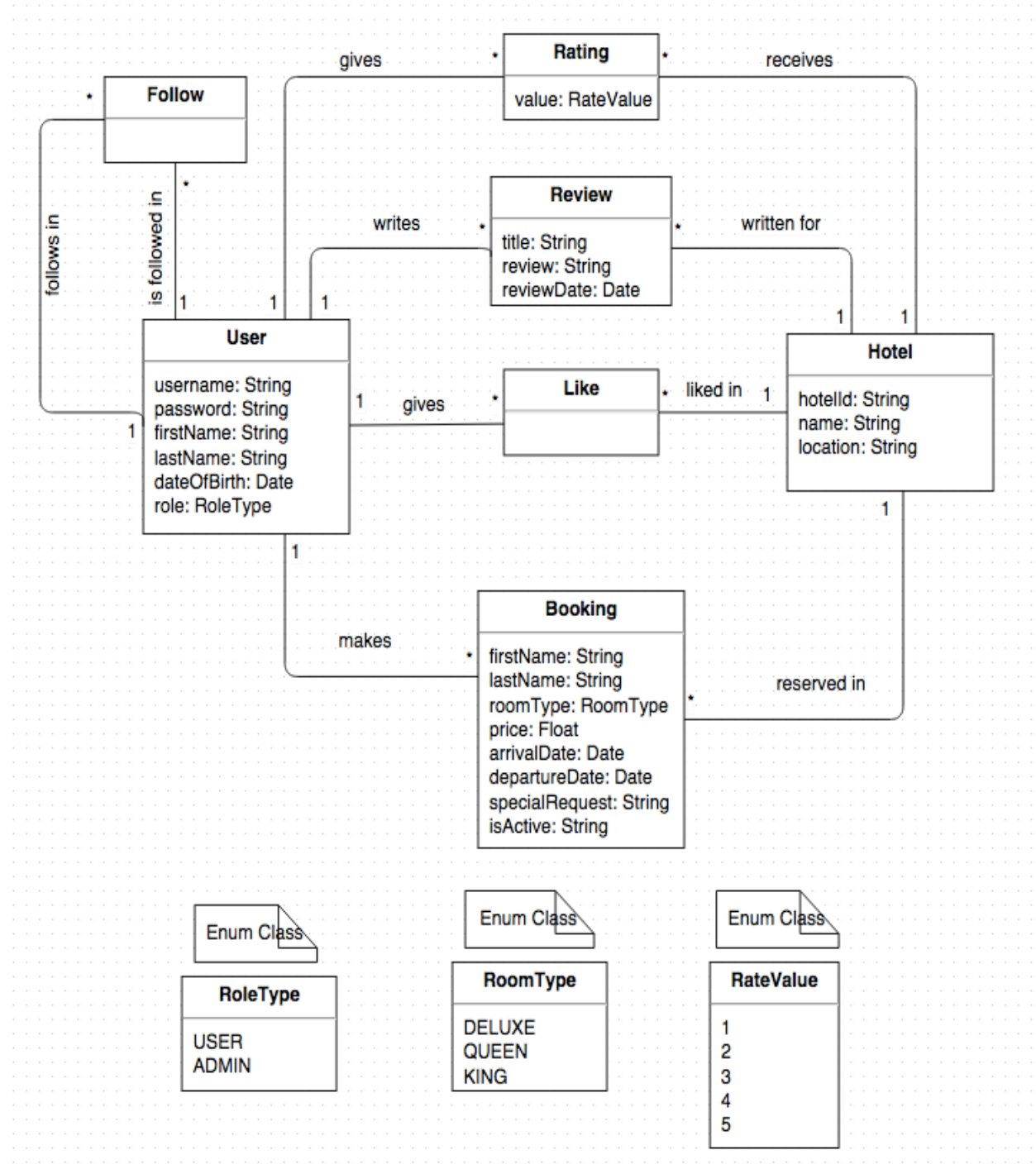
Technology stack

We have used the following technologies in the project.

- J2EE
- JSP
- MySQL
- JPA
- BootStrap for CSS
- jQuery
- javascript
- AJAX
- CSS
- HTML

(UML diagram on next page)

UML Class Diagram



USE CASE ANALYSIS

Actor: Customer

Use Case: Create a new account

Description: A new customer wants to register with the system

Use Case: User Login

Description: Customer logs in into the system

Use Case: Search Hotel

Description: Customer can search for hotel for any city around the world.

Use Case: View Hotel Details

Description: Customer can view hotel details

Use Case: Add a review for a hotel

Description: Customer wants to share his experience for a hotel he stayed in

Use Case: View Reviews

Description: Customer can look for reviews for hotel

Use Case: View Personal Profile

Description: Customer can view personal details

Use Case: Update profile information

Description: Customer wants to update his profile information

Use Case: Like a Hotel

Description: Customer wants like a hotel

Use Case: Follow another User

Description: Customer wants to follow another User

Use Case: View Booking History

Description: Customer wants to view history of his bookings and cancellations

Use Case: Extend existing booking

Description: Customer wants extend his existing booking

Use Case: Book Hotel

Description: Customer can book hotel

Use Case: Cancel Booking

Description: Customer can cancel reservation made at hotel

Actor: Administrator

Use-Case: View booking for user

Description: View booking for particular user

Use-Case: Create Admin Account

Description: Creation of administrator account

Use-Case: View all bookings

Description: View all bookings for all users

Use-Case: View booking for user

Description: View booking for particular user.

Use-Case: Book hotel for a customer

Description: Complete booking for a customer.

Use-Case: Cancel a booking for a customer

Description: Cancel reservation for particular user.

Use-Case: User Maintenance

Description: Perform maintenance for particular user.

Future Scope

We have thought of the following possible enhancements that we would like to make to our project in the future:

- We could add the functionality for handling different types and methods of payment. For now we have assumed that payment will be made upon check out from the hotel, and our system does not accept any kind of payments.
- We can expand the project to provide the users with the functionality of booking flights from our system, as users mostly make hotel reservations depending on their flight reservations.
- A price comparator can be added, by which we can provide the users with a comparison of hotel prices that our system provides versus the prices that other similar systems provide.