With an understanding of servers and their relationship with the client side, several features can now be made persistent and extendable beyond a single session. These features include the search function, login/signup, storing of personal bookings and favourites as well as adding new hotels.

# 1 Search Function

The user will search by location or hotel name using the search bars available on every page. The client will send a request to the server for hotel's that are near the users search specifications. The server will send back the relevant hotels and populate the hotels available page.

| | |
|---|---|
| Client | Location or hotel name and dates |
| Server | Location (lat,lng), name, star rating, pictures |

*Table 1.1: Data Location – Search Function*

## 1.1 Search Timeline

- ⇒ User inputs a location or hotel name into search bar and clicks 'GO'.
- ⇒ A GET request is sent to the server.
- ⇒ The server obtains the search parameters and searches the file containing the hotel information for matches.
- ⇒ Server responds with JSON data of hotels that match the search.
- ⇒ The Hotels Available page is rendered with relevant information.

# 2 Log In/Profile

Currently, the website includes a functioning implementation of a sign up feature. However, to make this persistent across sessions and to include a log in feature, the server will need to store the following information about each user:

| | |
|---|---|
| Client | Email, password |
| Server | User ID, email, first name, last name, E-mail address, home address, city, password, admin status |

*Table 2.1: Data Location – Log In*

A user will submit an email and password. This information will be sent to the server which will check if these match a user already stored on the server. If it matches, the user will be logged in.

For every user, the following will be stored in different JSON object arrays to keep track of which hotels have been booked and are favourites of the user.

| | |
|---|---|
| Server | Booked hotels, favourite hotels |

*Table 2.2: Data Location – Booked and favourite hotels*

### 2.1 Login function timeline

- ⇒ Username and password is input by user.
- ⇒ Ajax request is sent to server which searches relevant JSON file for existence of user.
- ⇒ The corresponding *user identifier* is used to gather other relevant information including current bookings and favourites.
- ⇒ The response to request is sent back to the client with the user's information.
- ⇒ The webpage is updated to display information relevant to the user.

## 3 Creating a Profile

In order to create a profile, the user will need to submit the existing form, detailing their personal information.

| | |
|---|---|
| Client | Email, password, name, address, city, password, admin status |
| Server | *Identical to above* with the addition of a user ID |

*Table 3.1: Data Location – Creating a profile*

### 3.1 Creating a Profile timeline

- ⇒ User fills out relevant information in login page.
- ⇒ Post request is sent to server.
- ⇒ The information supplied is distributed to relevant JSON file.
- ⇒ The user is issued a *user identifier.*
- ⇒ The server sends information back to the client and the user appears to be logged in.

## 4 Maps

The maps function of our web page will require a Hotel Name, latitude value and longitude value. Due to the refine feature of our webpage the client will need to know which websites are displayed and get the relevant hotel information from the hotel's database.

### 4.1 Map population timeline

- ⇒ User "actions a trigger" to see hotel on map.
- ⇒ A GET request is sent to the server with the *hotel identifier.*
- ⇒ In the server the hotel's JSON array is queried using the *hotel identifier.*
- ⇒ Latitude and longitude variables are sent back and displayed on the map.

# 5 Uploading a hotel

Much like creating a profile, creating a hotel will require the user to supply all of the information. This information will be sent to the server to store in the correct files and format. It is noted that only users that have admin status (chosen upon sign up) can upload a hotel.

| | |
|---|---|
| Client | User identifier, hotel name, location(lat,lng), pictures, description, price, amenities |
| Server | *Identical to above* - store in an object array |

*Table 5.1: Data Location – Uploading a hotel*

## 5.1 Creating a Hotel timeline

- ⇒ User inputs all relevant information into upload hotel page.
- ⇒ POST request is made to server with hotel information.
- ⇒ Server creates object and stores information in relevant file and format.
- ⇒ The server now contains the new hotel, ready to be sent to the client when requested.

# 6 SUMMARY

| Potential Object Arrays | |
|---|---|
| Users | First name, last name, E-mail address, home address, city, password, *user identifier,* admin status |
| User favourite/booked | User identifier, hotels booked, favourite hotels |
| Hotel | Identification number, hotel name, location(lat,lng), city, country, star-rating, amenities |

**User/hotel identifier* – In order to avoid confusion, each user and hotel will be allocated a unique identifier on the server side. These identifiers will not change in order to avoid confusion or conflicts. E.g. two people having the same name or a hotel changing its name.