Data Plan

1 Objects to be stored on server:

1.1 User Object Example:

```
var user = {
    userId: 1,
    username: "someUserName",
    password: "secretPassword",

    profile: {
        firstName: "steve",
        lastname: "steve",
        profilePicture: "/images/profilepic.png",
        postcode: 5000,
        mobile: 429109123,
    },

    bookings: [
        booking1,
        booking2
    ]
}
```

1.2 Restaurant Object Example:

```
var resturaunt = {
    resId: 1,
    name: "Maccas",
    aveRatings: 5,
    description: "High class burger joint",
    images: [
        "../test/test.png",
        "../yougetthepoint/ye.png"
    ],
    tables: [
        {
            tableId: 1,
            seats: 4,
            indoor: true,
            available: false,
        },
        {
            tableId: 2,
            seats: 4,
            indoor: false,
            available: true,
        }
    ],
    location: {
        address: "88 Northwest Terrace, Adelaide SA",
        lat: 87.18239,
        lng: 138.78209302019
    },

    reviews: [
        {
            review1,
            review2
        }
    ]
}
```

1.3 Booking Object Example:

```
var booking = {

    bookId: 1,

    time: Date(),

    resId: 1,

    tableId: 1,

}
```

1.4 Review Object Example:

```
var review = {

    rating: 4,

    reviewDescription: "wow I love Maccas",

    userId: 1,

    resId: 1,

}
```

2 Server Routes

2.1 Page Related Requests

Request	**Functionality**	**Server Response**

2.2 User Related Requests

Request	**Functionality**	**Server Response**
GET /user	Retrieve all information related to the user. Including bookings.	Send user object
POST /user/make	Makes a user object and stores on database	Receive new user object and uploads to database.
POST /user/change	Updates existing user object and related database data.	Receive a user object and update database.

2.3 Restaurant Related Requests

Request	Functionality	Server Response
GET /restaurant/get?resId	Retrieve all information related to the restaurant. This will be used any time restaurants are required on the page.	Send restaurant object according to Id.
POST /restaurant/make	Makes a restaurant object and stores on database.	Receive new restaurant object and uploads to database.
POST /restaurant/update	Updates existing restaurant object and related database data.	Receive new restaurant object, replace old restauraunt object and updates all related data on database.
GET /restaurant/delete?resId	Deletes a restaurant.	Delete restaurant object and all related data on database.
GET /restaurant/get?resId&tableId	Retrieve all information related to a restaurant table. Will be used to fill in information related to bookings.	Send table object according to Id and restaurant.
GET /restaurant/search?name	Retrieve list of restaurants that have the name or anything in the description to what was searched	Sort restaurants by similarity to search into list. Send the list of restaurant objects.
GET /restaurant/search?location	Retrieve list of restaurants above 3 stars and close to the location.	Sort restaurants above 3 stars by distance from particular location into list. Send a list of restaurant objects.

2.4 Booking Related Requests

Request	Functionality	Server Response
POST /book/make	Makes a booking, with a date, table Id and a restaurant id.	Receive information about booking. Makes a booking object and stores into database.
GET /book/cancel?bookId	Deletes booking.	Deletes booking object from user.
GET /book/user	Retrieve a list of bookings under the user.	Sends complete list of bookings under user.

2.5 Review Related Requests

Request	Functionality	Server Response
GET /review	Retrieve all information related to the review.	Send user object
POST /review/make	Makes a review object and stores on database	Receive new review object and uploads to database.
POST /review/change	Updates existing review object and related database data.	Receive a review object and update database.