WDC RESTAURANT WEBSITE REPORT

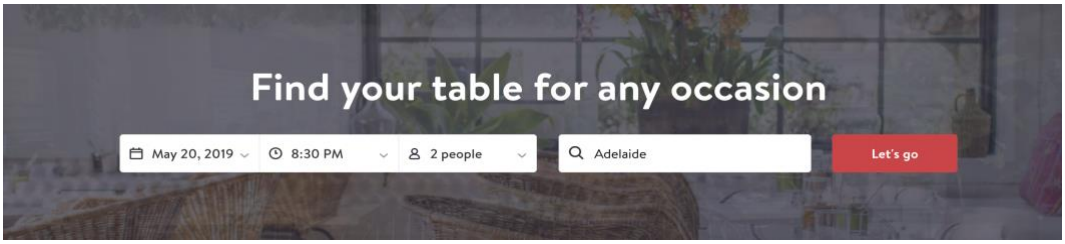
Kien Nguyen (a1735829@student.adelaide.edu.au)
Dat Le (a1730614@student.adelaide.edu.au)
Kien Tang (a1738166@student.adelaide.edu.au)
Zhao Ming Soh (a1751699@student.adelaide.edu.au)

Content

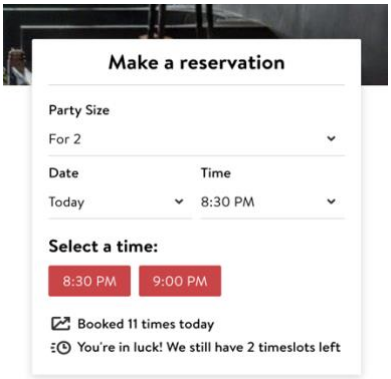
1. Literature Review

Starting off with the project, we first look at what the feature and component that a booking restaurant website should have. After looking through multiple restaurant booking websites, we finally agreed on one website which is called OpenTable that has the most appropriate features and design that matches what we want in our restaurant booking website design.

1. Features wise, there are a few that we kept in the design of our restaurant booking website such as the main search bar that is separated into a few different categories that deal with the selection of time, calendar, number of diners and restaurants. This search bar contains all the important information that are relevant to searching for the most specific restaurant: date, time, number of people and location

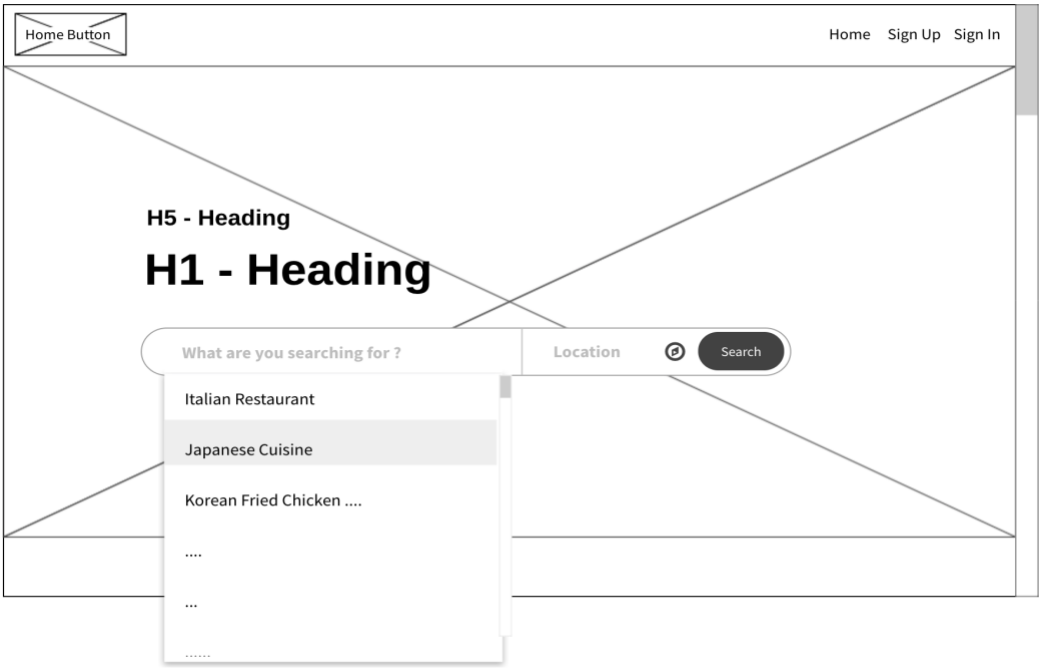


2. Besides, OpenTable also has the common features such as review, sign in and sign up functionality which are packed in a comprehensive design. We believe this is the most suitable design referenced that we can utilised for our page as it is simple and straightforward.
3. We also referenced some of the layout structure designs for the display of a wide range of restaurants upon being searched for. The design layout for the booking function not only contains necessary information but also compress addition of details that customer needs for a successful reservation to be made.



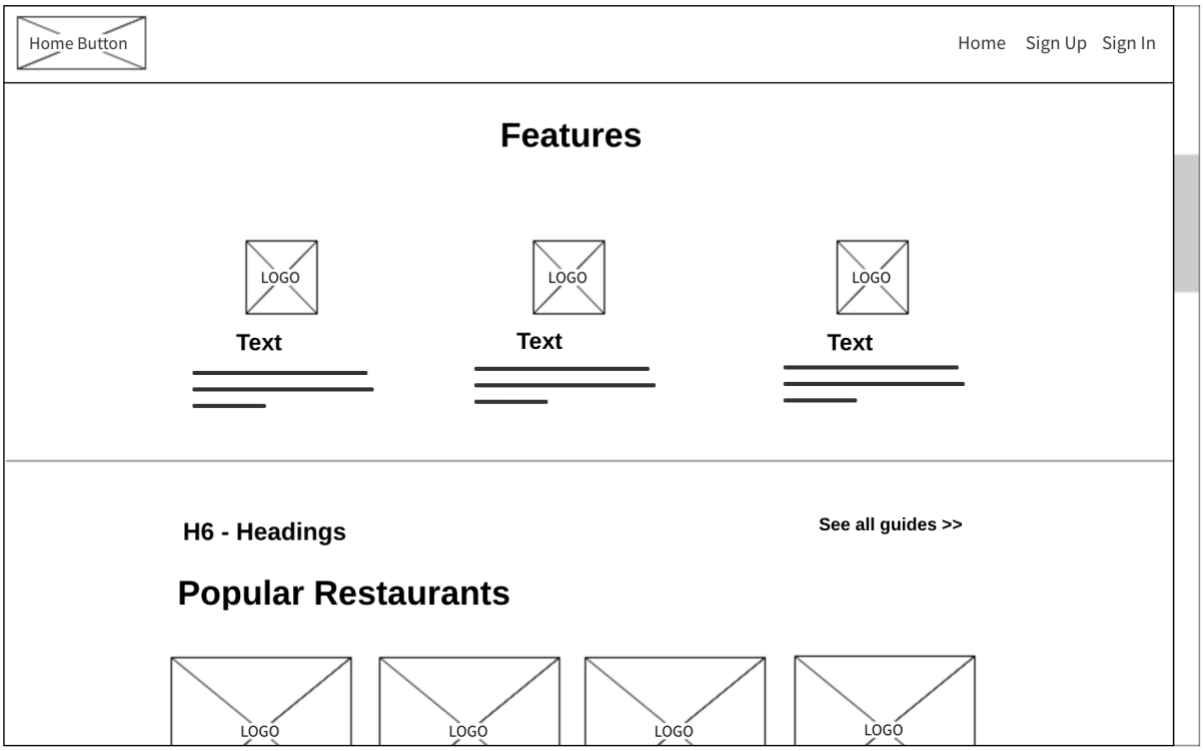
2. Design and Features

2.1 Homepage Design

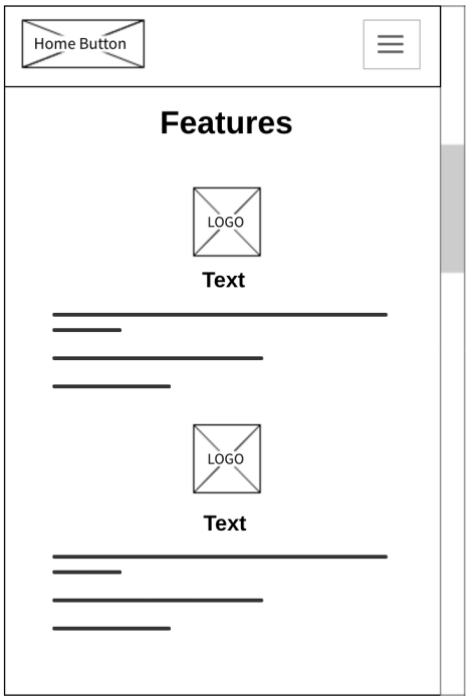


"First part of the Scroll Bar" Desktop Layout

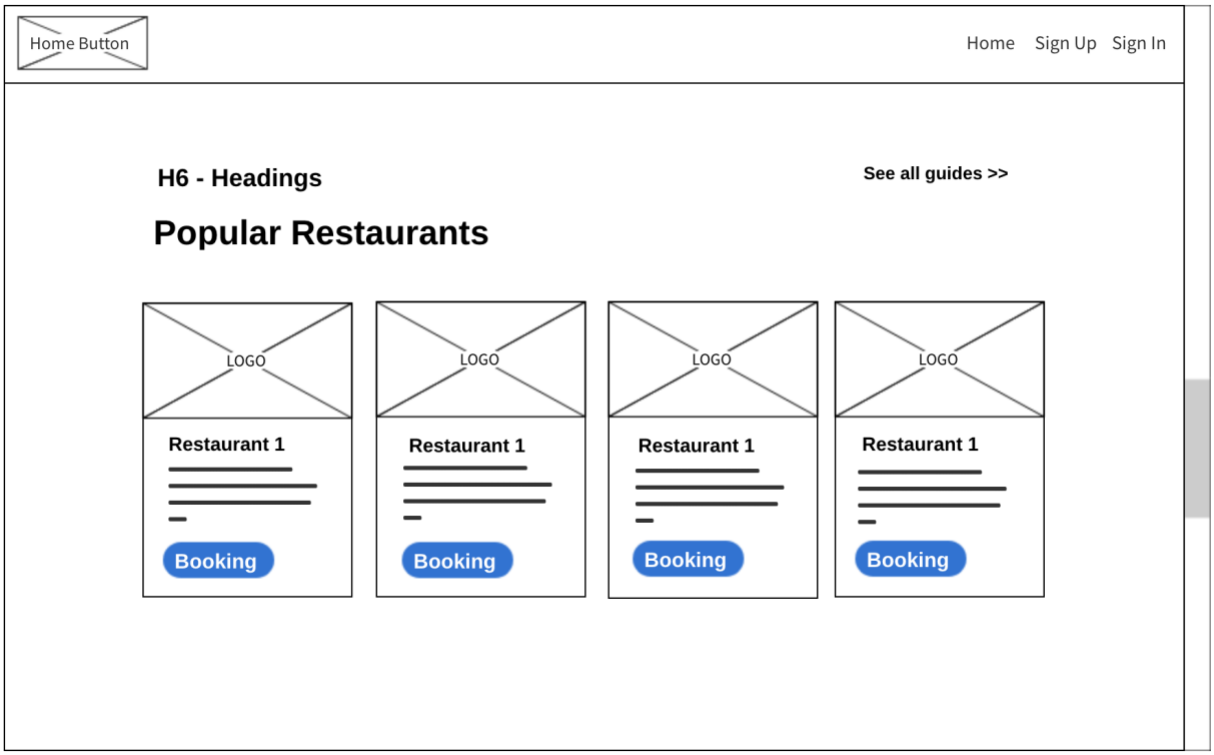
"First Part of the Scroll Bar" Mobile Layout



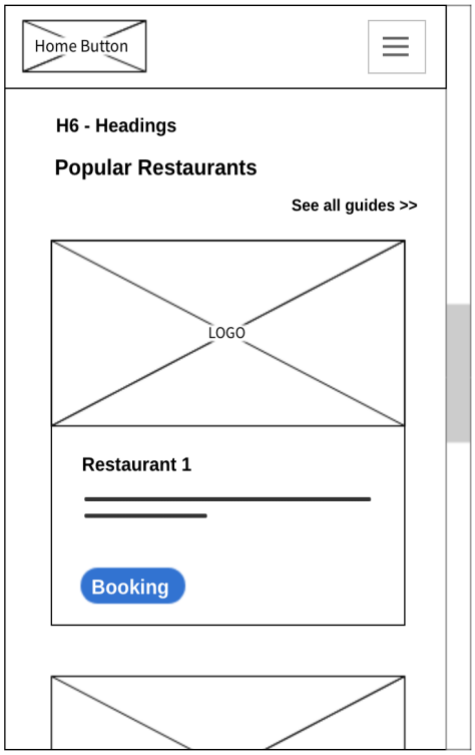
“Second section of the Scroll Bar” Desktop Layout



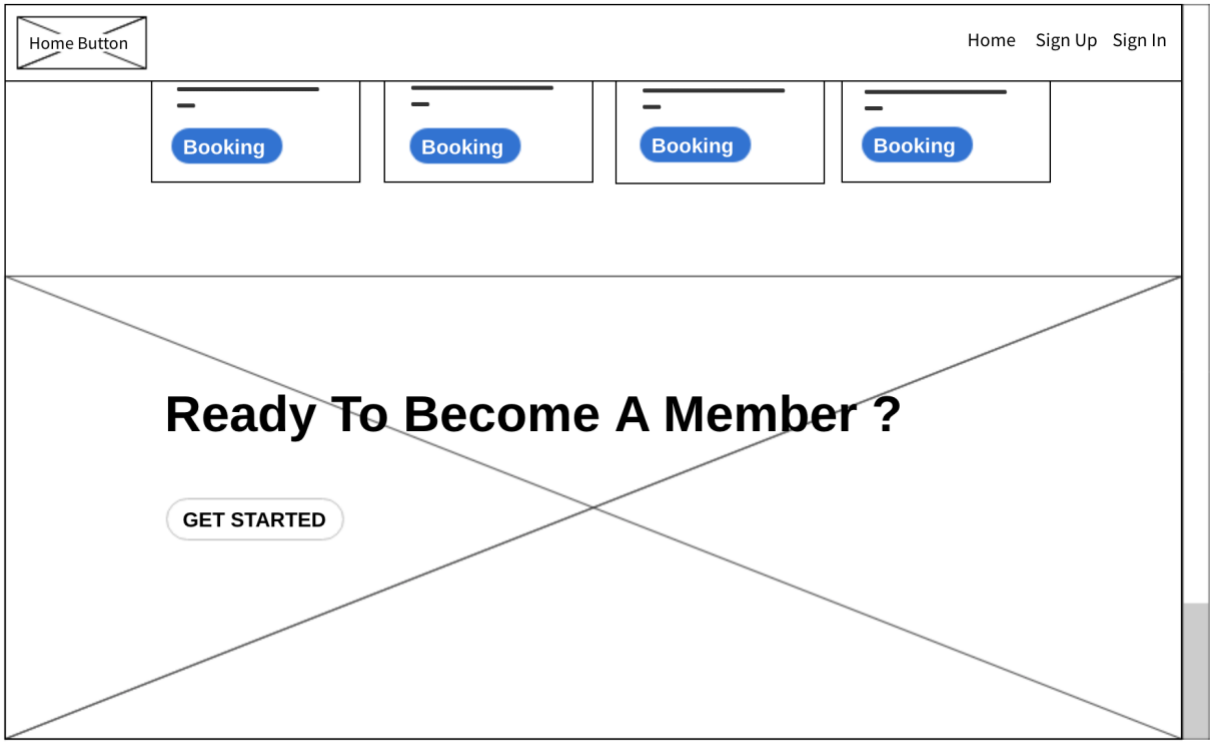
“Second section of the Scroll Bar” Mobile Layout



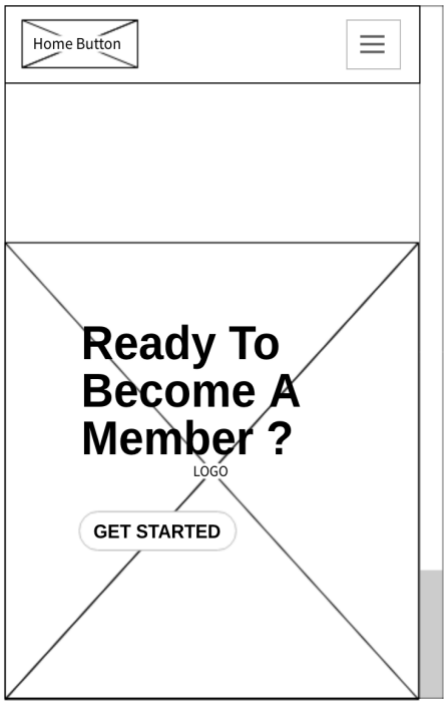
"Third section of the Home Page" Desktop Layout shows users the most recent and popular restaurant near them



"Third Section of the Scroll Bar" Mobile Layout



“Last section of the Home Page” Desktop Layout (note: This is could be the member sign up link for restaurants managers)

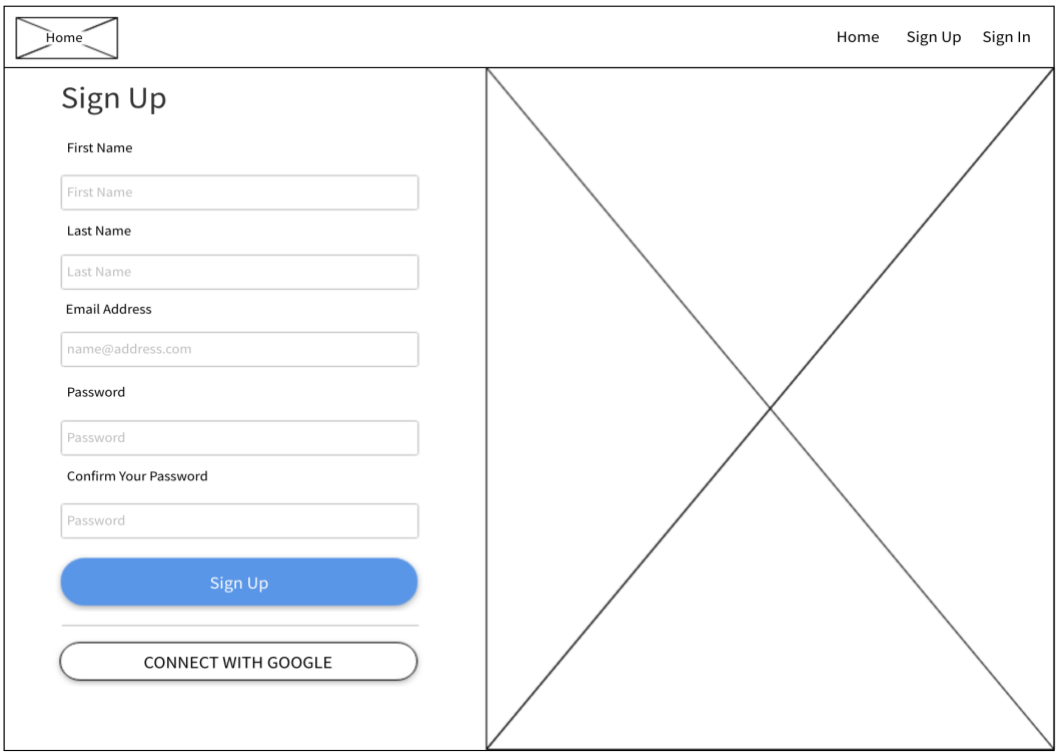


“Third Section of the Scroll Bar” Mobile Layout

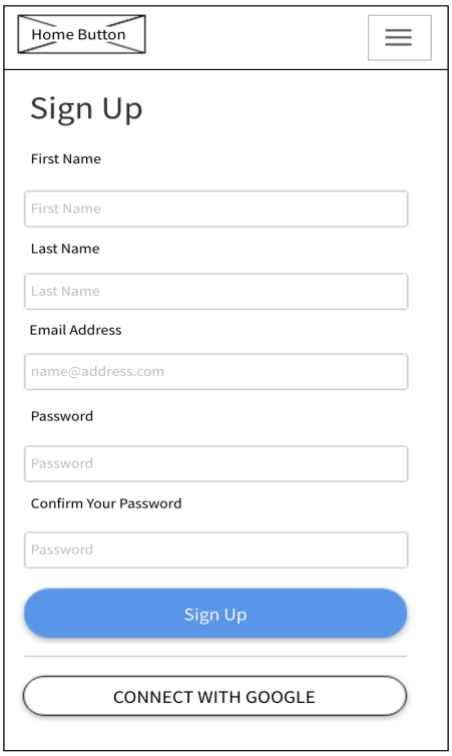
In home page, users are provided with a set of standard features.

1. The first one is the search bar which is a basic and necessary feature that allows users to search for their preferred restaurants based off of their location.
    a. The search bar has two selective inputs, one is for restaurant particulars and the other one is for tracking the users' location. These two combines would allow the users to search for their preferred restaurants situated close to them.
    b. If any restaurants' name is keyed into the search bar, a list of recommended restaurants will be listed underneath the search bar.
    c. If the users clicked onto the search bar with or without input texts, the users will be directed to a page that has an ordered list of restaurants.
2. The second feature is the standard navigation bar which is set to show up on every single page of the website as it is the only area of navigation that links the entire website to all its different pages that are available. In the mobile window size version, all the navigation details are loaded into a drop-down-list menu located on the top right-hand corner of the webpage.
3. There are three and potentially fourth buttons displayed in names that correspond to their particular functionality which are the “home”, “sign up”, “sign in” and potentially “about”.
4. The “home” button will take users to the homepage, which is set to as the default page of the website.
5. The “signup” button will link users to the Sign-up page for membership registration purposes.
6. The “sign-in” button will link users to the Sign-in page for members access.

2.2 Sign Up Page



Desktop Layout

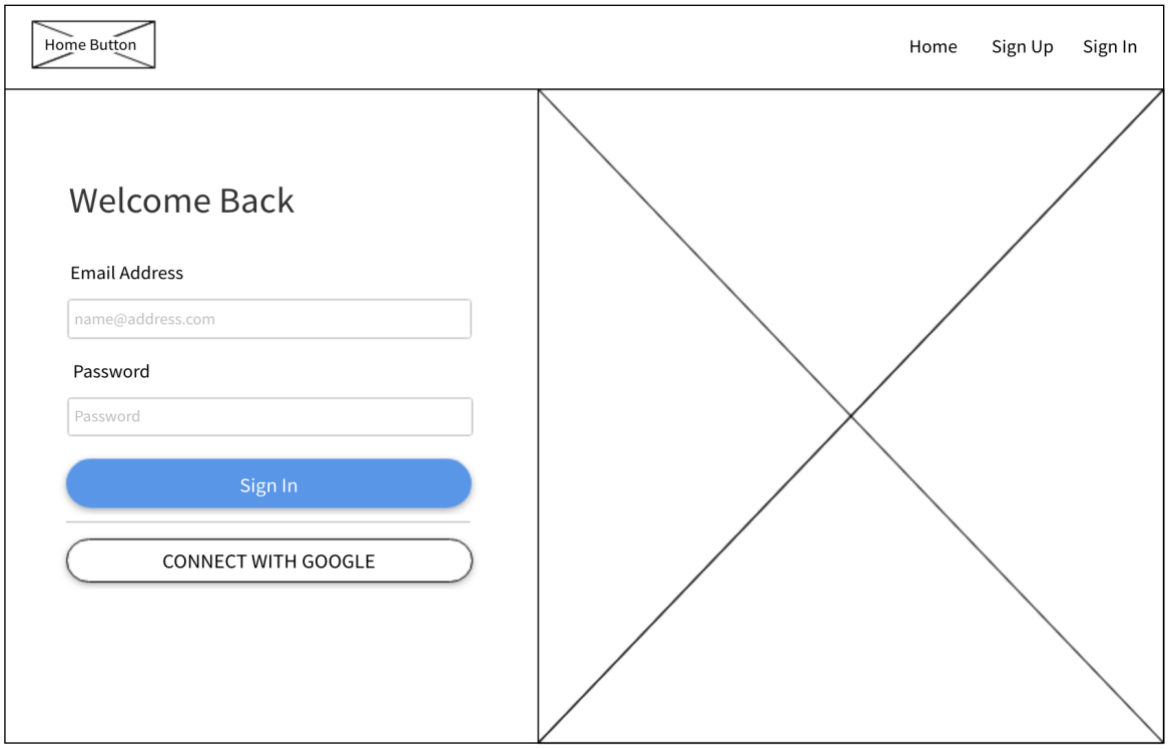


Mobile Layout

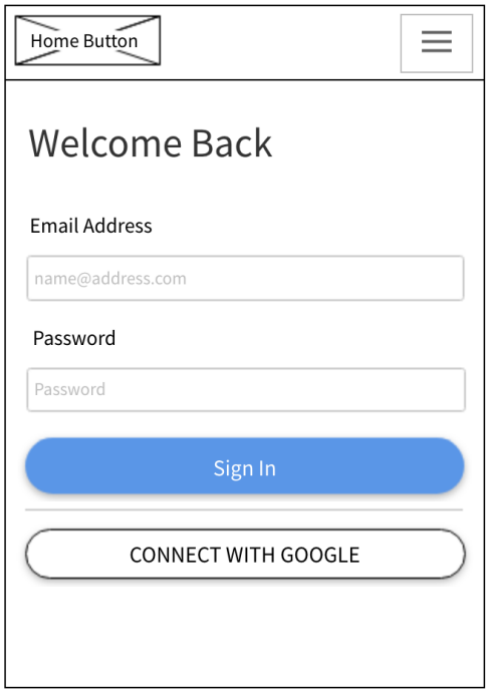
In Sign-Up page, users are given a set of standard user inputs to make a standard membership registration for the users to manage their bookings. As in a conventional reservation system, one would need the user's personal details in order to validate their legitimacy and to make reservation much easier for both the web managers and the restaurant's managers.

1. Starting from the top end to the bottom end of the user inputs, users are prompted to key in 4 kinds of information which include users' first name, last name, email address, and password. If any of the 4 information is invalid, an error message will be presented to ensure that the users provide a correct set of information.
2. Each of the inputs is set to have its own requirements
    a. In the input text of the first name and last name, no special characters and numbers are allowed.
    b. The email address input text only accepts the correct email format that includes the '@'.
    c. The users' initial password keyed into the password input text needs to match the password keyed into the confirmation password input text.
    d. All of this information will be checked for validity in order to complete the sign-up process. Once, the sign-up button is pressed, all of the following information will be stored in our website customer database in our server which will be implemented in our code at a later stage.
3. Alternatively, users can also sign up using existing Google account simply by clicking the button that states their corresponding names at the bottom end through the use of Google's API for sign in purposes.

2.3 Sign-In Page



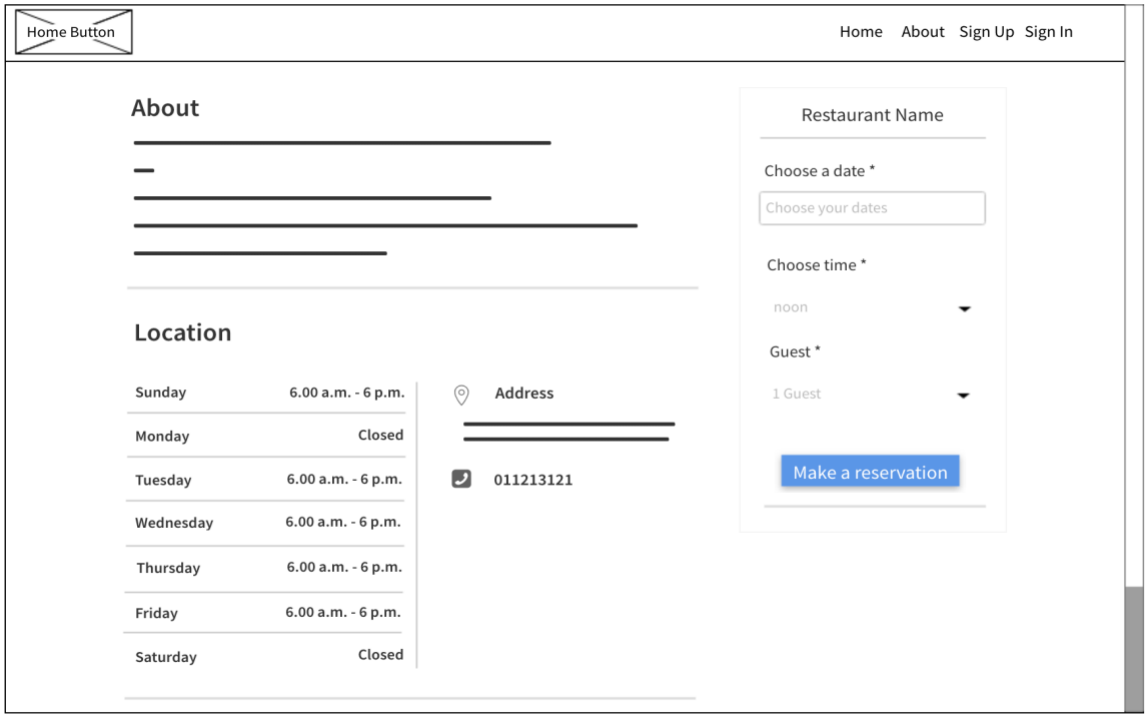
Desktop Layout



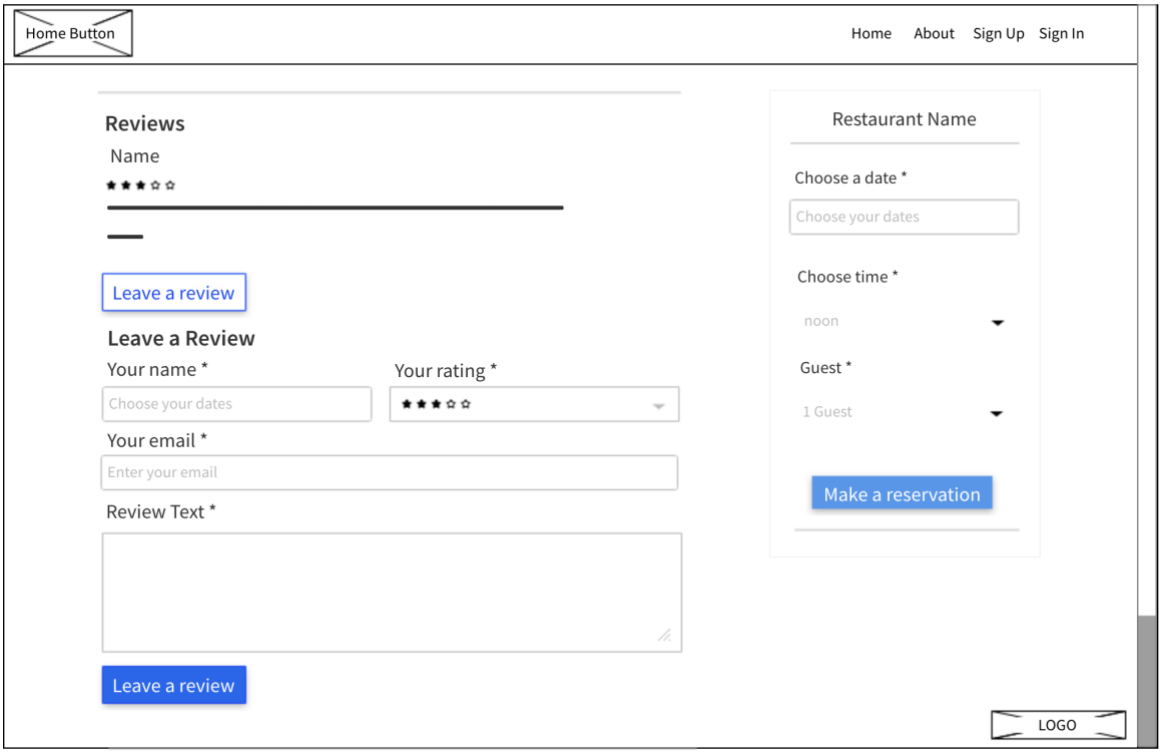
Mobile Layout

In Sign In page, users are given a set of standard login inputs for signing into our website. The sign-in page has the same exact layout as that of the sign-up with only 2 input texts for which the users must provide their registered email address and an appropriate password to log in. Once the sign-in button is pressed, our website will pull data from our server to begin a matching test and upon validation, the users can then login into our restaurant booking page.

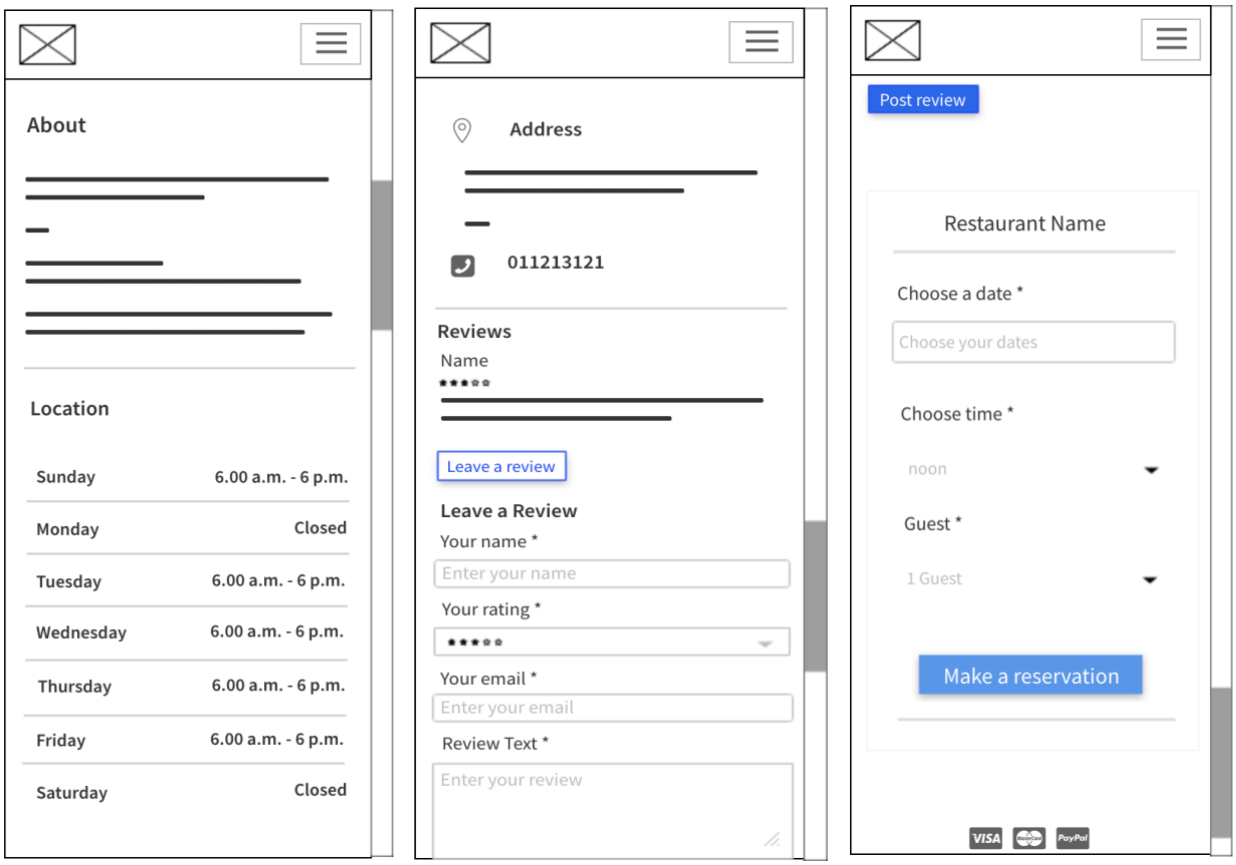
2.4 Restaurant booking page



Desktop Layout (top)



Desktop Layout (bottom)



Mobile Layout

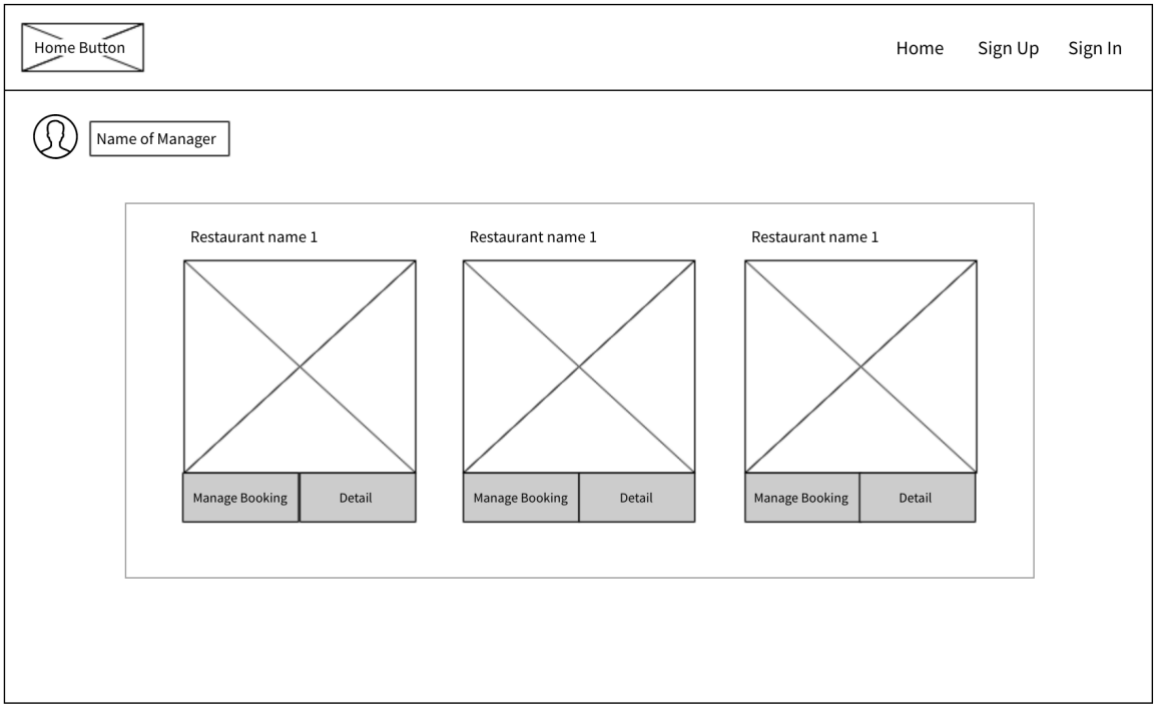
In Restaurant Booking page, users are presented with a scrollable page with multiple user interactive interfaces.

1. At the forefront of the scrolling page, users are given information such as the description of the restaurant, their history and their food based off of the selected restaurants underneath the about text. This changes according to what the users initially search for in the home page where our website will pull the relevant data of that particular restaurant from our database.
2. Also, right below where the information of the restaurant is presented, users are also given the general business details about the selected restaurant located underneath the “Location text”. The business details include the business hours on the left-hand side of the column and the address and contact details on the right-hand side of the column.
3. At the end of the scrolling page, users are given the option to leave a **review** for the restaurant of their liking.

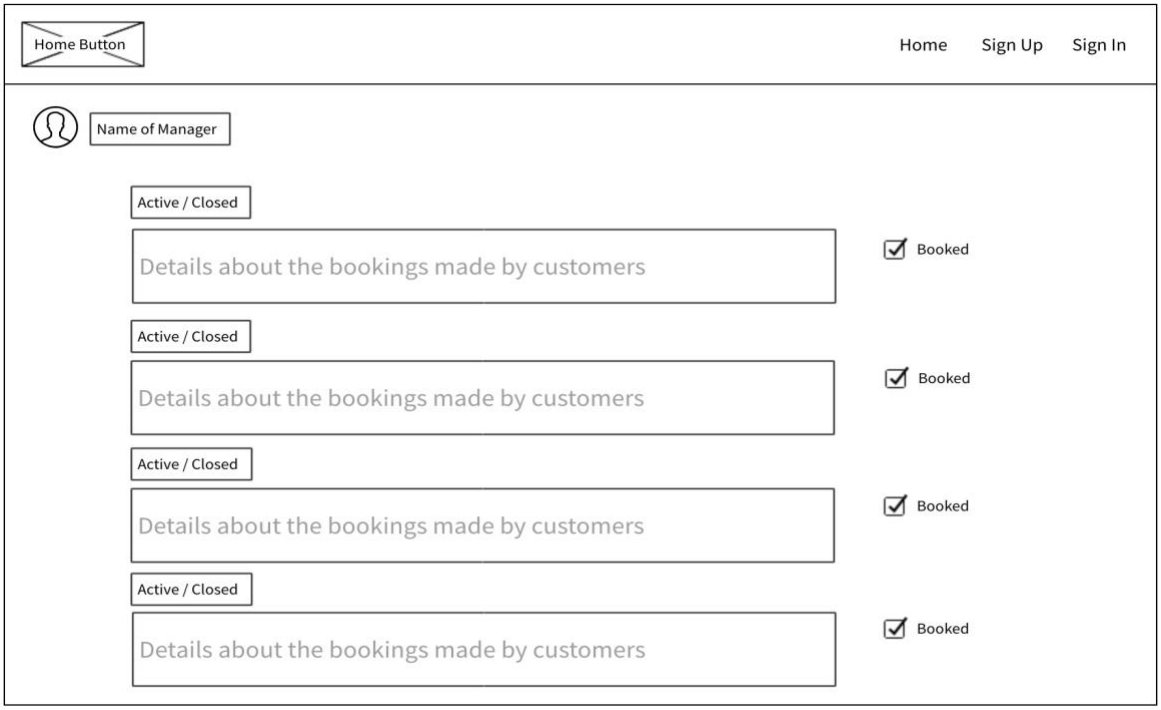
a. The review system in our website includes a rating system that shows users degree of liking or disliking based off of 5 stars. In order to leave a review, the users are required to provide their names, email address, ratings and the account of their review.
b. The name and email address inputs are placed with some rules such as the name input text cannot have numbers and special characters whereas the email address input text must have the correct format keyed in with a '@'. After all of this are verified, the users need to press the "leave a review" buttons so that all of the information above can be stored into our database and regurgitate into a form of text model that is display in the area above the review model.

4. At the right-hand side of the web page, we have a fixed box model that handles the reservation of the selected restaurant with an input text and two input options given.
    a. In the box model, users have to select the date of their reservation, the time of their reservation and the number of people for the reservation. The date input text only takes in the correct date format and that format looks like this, "Days/months/years" where each input must be in a pair of numbers. As for the other two input options, users are only required to choose the appropriate choice of numbers which are already listed in the input options list. Of course, all of this information will also be directed to our server database so that the receiving end can work with the information that we provide them with for the reservation which are the restaurant managers.

2.5 Manager Page



The profile page for the managers that have signed up to our website membership



The page where managers can manage the booking requests made by the customers

This is an integral part of our website because for any appearance of restaurants on our page requires managers of the specified restaurant to join our website by signing up with us.

1. In the managers' profile page, managers are given the feature of adding and cancelling the number of restaurants that will be utilised in the search engine of our web page.
2. Upon clicking the "Manage Booking" button, managers are directed to a page that enables managers to review and modify the existing bookings that are made by the customers. The managers can change the status about a specific booking session as active or closed by clicking the active/closed button on the top of each booking.

3. Review

3.1 Kinematic loads and Cognitive loads

In order to minimize kinematic and cognitive loads, we design our web page layout as simple and straightforward as possible.

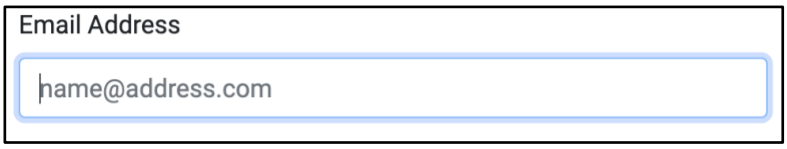
1. In our home page, we have a very apparent search bar that is situated in the middle of the page with a very clear label of what it is used for. Therefore, users don't have to waste any effort in searching for their favourite restaurants.
2. Another key design is the navigation bar where we fix it at the top of every web pages to ensure users can easily navigate to the page of their liking. We also use concise and understandable wordings for every navigation button in every single web page in order to convey quick and clear use of each of the buttons that are present.
    a. Users also don't have to worry about getting lost in navigating through our web pages as every web page are given clear texts and signs of their respective functionalities.
    b. Users also don't need to work hard at moving about for navigation purposes as everything that has close relationship with one another are placed in an encapsulated area that is respective of their functionalities such as the reservation box model where users can key in everything that is needed for a reservation to be made in such close proximity without any wasted motion at all.

3.2 Does it meet the standards and heuristics?

After reviewing our web pages, we strongly believe that our user interface design for our web site does meet the standards and heuristics of a web site.

1. Our design has a feasible visibility of system status. In terms of navigation around our website, there are definitely clear feedbacks from the buttons that are provided in every section ranging from the navigation menu to users inputs area such that when clicking on any of the texts in the navigation menu, one would be immediately sent to the appropriate pages specific to what the texts are stating. In user input

texts domain, one would be prompted with messages once something is done such as an error message would be displayed if the users keyed in characters that are invalid to our text input requirements.



Email Address Input Text

Effects of hovering mouse cursor to any of the buttons will turn dark in colour

2. Our system matches the real-world system. Since we design our website based on a real website called OpenTable.com.au. Although not every aspect of our system is going to match exactly like that of an official website because we are still rookies when it comes to design.
3. Our system provides users with control and freedom. We aren't too sure about users freedom but in terms of control we definitely have a clear path as to how users can recover from mistakes such as the users will be given a line of short text below the email address input text upon entering a wrong email address or the entire outline of the box turns red which is a universal indicator that something is wrong. This also applies to other forms of user input texts.
4. Our website has good Error prevention. The user interfaces of our website are mostly user-based inputs, we have to set requirements for each user inputs in order to prevent users error inputs in the first place. This is visible when users provide the wrong details to users input texts such as the email address input text.

A warning message is displayed when users keyed in the wrong email format

5. Our website has good consistency and standards as we have already mentioned in kinematic and cognitive loads section.
6. Our website accommodates recognition rather than recall. Our designs are very minimalistic, all of our pages contained only a minimal amount of words and our

website also has a straightforward design language which should prompt users to navigate through our website only by recognition, not by recollection.

7. Our website has good flexibility and efficiency of use. The design language used in our website is made for both experienced and inexperienced users as again our design is focused on the minimalistic side that only has functionalities that are necessary. Therefore, we are confident that inexperienced users can definitely navigate through our website effortlessly.

4. Data Plan

4.1 Server Routes

Method	Purpose	Request/Response
GET /search	Retrieve a list of restaurants nearest to the detected location	Response with an array of restaurant objects
GET /search?q=	Retrieve a list of restaurants based on search text & nearest to the provided location	Response with an array of restaurant objects
GET /search/restaurants/restaurant_id	Retrive the restaurant details page	Response with a restaurant object which included all its details
POST /search/restaurants/restaurant_id/reviews	Create a new review	Send a review object to be added into database
POST /booking/new	Create a new booking for the user	Send a booking object to be added into database
POST /booking/booking_id	Update an existing booking for the user	Send a booking object to be update into database
GET /manager/manager_id	Retrive the manager home page	Response with an array of restaurant objects which are managed by manager
GET /manager/restaurant_id/manage	Retrive the restaurant manager page	Response with a restaurant object defined by id

GET /manager/restaurant_id/booking	Retrieve a list of active booking of the restaurant defined by id	Response with an array of booking objects of the restaurant defined by id
POST /manager/restaurant_id/booking/booking_id	Update/modify a booking	Send a booking object to update booking details for the restaurant defined by id
GET /user/user_id	Retrieve the information about the user	Response with a user object which included all its details & related active booking
POST /signup	Create a new user	Send a user object to be added into database
GET /signin	Log user in	Response with a session for user

4.2 Object Examples

1. Restaurant Object:

```
{
"restaurant_id": "A001",
 "restaurant_name": "Sri Vasista's Family Restaurant",
 "restaurant_phone": "+61870731231",
 "restaurant_address": "Godi La Vita 162B King William Rd",
 "restaurant_latitude": "-34.954950",
 "restaurant_longitude": "138.599780",
 "restaurant_capacity": 100,
 "restaurant_description": "For modern Italian cuisine in Adelaide, look no further than Godi La Vita..."
}
```

2. Booking Object:

```
{
 "customer_id": "1",
 "restaurant_id": "A001",
 "start_time": "19:30",
 "no_of_people": 2,
 "status": "active"
}
```

3. User Object:

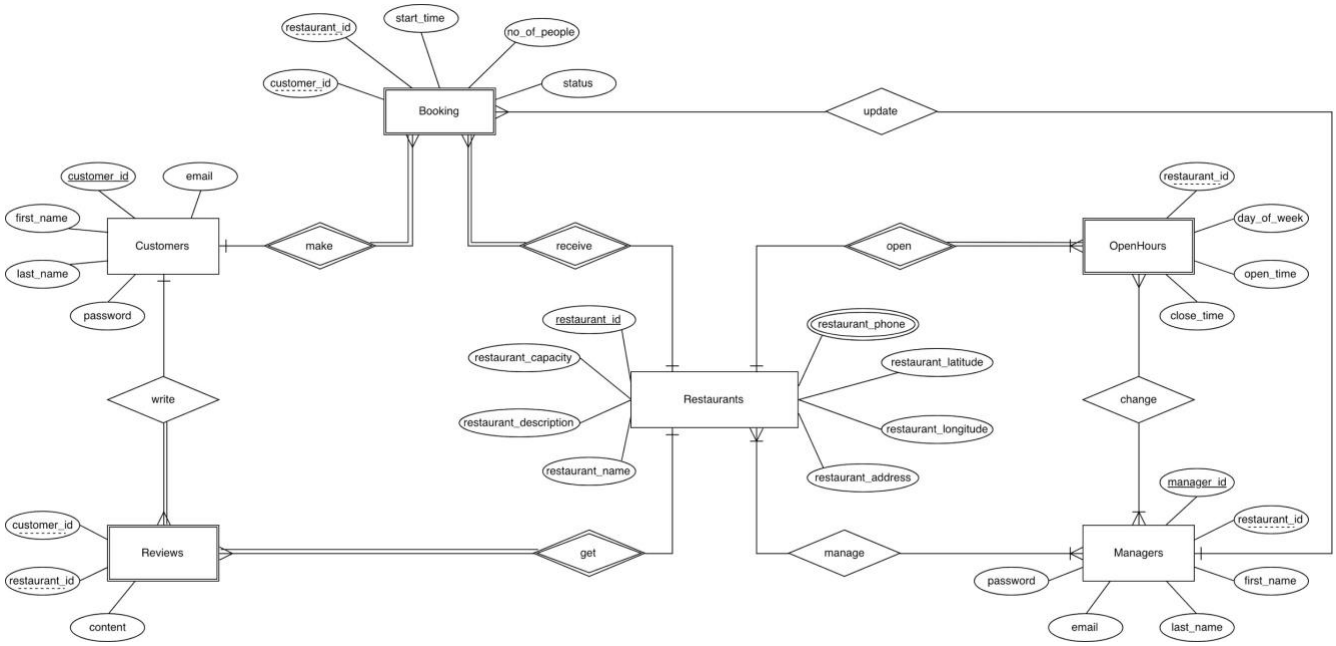
```
{
 "customer_id": "1",
 "first_name": "Ian",
 "last_name": "Knight",
 "email": "ian.knight@adelaide.edu.au",
 "password": "ianverySecretPassw0rd"
}
```

4.3 Client/Server Interactions

Page	Interaction
ALL	Pages are served up statically by the server
Home	Client send GET request to server, asking for 4 nearest restaurants
Search	Client sends GET request to when a search is made and the server returns a list of restaurant based on keywords and detected location
Restaurant	Client send GET request to server, asking for the restaurant details using id and the server returns a restaurant object
Booking	Client send POST request to server, create new booking and the server returns a success message

Manage Booking	- Client send GET request to server, asking for the list of booking related to current restaurant - Client send POST request to server, asking for updating booking using booking id
Manage Restaurant	- Client send GET request to server, asking for the details of current restaurant - Client send POST request to server, asking for updating current restaurant details
User	- Client send GET request to server, asking for the details of current user - Client send GET request to server, asking for a list of booking belong to current user - Client send POST request to server, asking for updating booking using booking id - Client send POST request to server, asking for updating current user details
Sign Up	Client send POST request to server, asking for creation of new user
Sign In	Client send GET request to server, asking for user authentication

4.4 ER-Model Diagram



(a higher resolution version of this ER-Model can be found in docs folder)