



Unreal Estate

Final Report

Clive Chen	z5166040
Emmanuel Kozman	z5120558
Gagandeep Nain	z5137193
Yang Ma	z5100105

Table of Contents

Introduction	2
Context	2
Project Purpose	7
Project Design	8
Overview of Components	8
Diagrams	11
UML (use link for full image)	11
ER Diagram	12
Software Architecture Diagram	13
User Scenarios	14
All Users	14
Renter	15
Advertiser	16
Sequence Diagram	17
Review Contribution	18
Search Contribution	21
Booking Contribution	23
User Account Contribution	25
Conclusion	27
References	28
Appendix	30
Installation manual for windows	30
Languages and Technologies	30
Source code structure	30

Introduction

The Unreal Estate team was largely successful in producing an innovative and creative accommodation platform. The final design is one that leverages the current best practices for accommodation listing and booking as well as providing a new approach to managing and authenticating user reviews. This structure of this report details the culmination of information regarding the initial motivations, intermediate methods and final product of this project.

The motivations and context of the project will be described in the introduction and will present research conducted for the project proposal as well as extended research into user reviews. The exploration into reviews was forecast in our initial proposal scope, however, the degree of focus on this area was increased as the project progressed. After describing the background and history of accommodation platforms, the introduction will include the aim of the Unreal Estate project team which arose as a response to the current status quo.

The main body of the report details the outcome of this aim and the Unreal Estate's final product. This will be achieved by first an overview of the system's architecture, descriptions of the usage scenarios for all types of users and through various diagrammatic representations of the platform. This overview will be followed by a detailed description of the four main contributions of the website, the review component, the search component, the booking component and the user management component. These detailed contribution sections will provide deeper insight into the backend techniques that were used for each component as well as images of the frontend interface.

The report will finally conclude with an evaluation of the overall project. This will describe how the underlying fundamental booking system was completed well and how the review system could be further expanded. Furthermore, the challenges faced, as well as areas for improvement, will be outlined.

Context

Economic Significance of This Project

The tourism industry represents around 10.4% of global GDP with a growth of 3.9% from 2018, second only to manufacturing. The World Travel & Tourism Council's (WTTC) report also outlines the fact that the tourism industry accounts for \$223bn of Australia's GDP. The sheer economic scale and rapid growth of this sector indicates the importance of understanding and exploring the key structures that enable such an industry to operate. One particular key structure is the accommodation industry which allows for the free and convenient movement of people across regions. This report will analyse online accommodation booking in terms of what

makes an effective platform and propose its own online accommodation booking project which attempts to replicate and extend current examples.

Success of Online Travel Agencies

The success of Online Travel Agencies (OTA) is due to several factors including, distribution, availability, removal of barriers to entry and affordability. These factors are all enhanced forms of marketing or unique economic advantages afforded by the tool of the internet. Another key reason for success is the ability for everyone to give peer reviews.

The first and most obvious reason for the success of OTAs and one that translates to any online service is the power of increased distribution. Distribution refers to the specific method of delivery of products and services to customers. It is no surprise then that access to the internet into peoples homes and pockets has allowed a revolution in distribution. Consumers are now able to access the accommodation booking service more often and with greater convenience.

The removal of barriers to entry for both consumer and provider has been another revolution for OTAs. The most recent model for OTAs has been the peer-to-peer model, where individuals are able to become their own agent and be in contact with renters directly. This had led to a much wider adoption and acceptance of the online service. Coupled with the removal of barriers is the increased availability and quantity of accommodation in the market. Availability is also increased by increasing exposure from aggregating accommodation options in order to advertise more completely for a consumer.

Another consequence of increased availability is that it generates more affordable accommodation simply due to increased competition. In addition, the peer-to-peer model introduces unconventional, cheaper forms of accommodation such as a single bedroom in shared apartments, or famously, an airbed in a living room, as was the original idea behind AirBnB.

The final feature of being able to provide peer review has been important in allowing users to widely adopt OTAs. The personalisation and humanisation of the platform leads to customer retention and more trustworthiness.

These factors all need to be considered when analysing and creating a successful OTA platform as they are what distinguish it from traditional forms of accommodation services. In deciding the implementation of a specific online booking system it is important to understand the possible business models available.

History of Online Accommodation Bookings

Historically, the online accommodation booking industry has been around almost as long as the internet itself. The first notable online accommodation service was Travelweb.com, established

in 1994. It started off as simply an online hotel directory but functionality to make bookings added soon after. Many of the other major OTAs were founded within the next decade, with the AirBnb being a unique exception.

Online Tourist Agent	Established	Age (yrs)
Travelweb.com	1994	25
Booking.com	1996	23
Expedia	1996	23
Webjet	1998	21
TripAdvisor, Inc.	2000	19
eDreams	2000	19
Wotif	2000	19
Opodo	2001	18
Trivago	2005	14
HomeAway (stayz.com.au)	2005	14
AirBnB	2008	11

This is clearly still a maturing industry which has seen different business models throughout its history always with area for improvement. There are four major models for OTAs, the advertising model, the agency model, the merchant model and the peer-to-peer model. Each has its own distinct set of advantages and disadvantages and also their own target demographics. The Unreal Estate team will consider the nature of each of these models in determining how to implement our own version.

OTA Business Models

The advertising model is one that assumes the lowest amount of risk. In this model, an OTA will not make any bookings but rather advertises rooms and redirects to other OTAs who handle the booking process using one of the other three models. The advertiser is then paid for the click-through to the site regardless of whether or not a booking is made. The most successful example of this model is Tripadvisor.com.

The agency model is perhaps the most conventional as it is the most similar to non-OTA methods. In this model, an OTA is awarded commissions on the sales that it makes on behalf of the hotels. The prices presented to the consumer are decided by the hotel and the OTA doesn't

take on much risk by simply acting as an intermediary. The most prominent proponents of this model are Booking.com and Expedia.

The merchant model, which is arguably more common, is one where the OTA will purchase the right to rent a set of rooms from a hotel. This kind of wholesale model can be seen in other industries where the OTA will have a contract with the hotel to purchase the rooms at a discounted rate and then has the right to decide on what they will charge to customers. The OTA, in this case, assumes the risk if they are unable to rent out the rooms in exchange for flexibility with pricing. An example of an OTA that uses this model is Webjet.

The final model and perhaps the most disruptive to the industry is the peer-to-peer model. As mentioned earlier, this model allows individuals to act as their own agent for their own properties by connecting directly to renters. The OTA, in this case, acts as an intermediary for advertising, booking and conflict resolution. The original and most successful example is, of course, AirBnB. It will be this model that our project will aim to emulate.

Existing platforms

The current state of the online accommodation booking market capitalises on much of the reasons for success mentioned above. The distribution is inherent with any OTA, while the benefits of removing barriers for entry arise most directly from a peer-to-peer model. The affordability and availability will result from the market forces once a peer-to-peer platform is created.

In terms of the design-flow for existing OTAs, they all follow a similar pattern. The current platforms attempt to make booking as seamless as possible by requesting an input of minimal key search criteria, often being: date of departure, date of arrival, number of guests and location(s). The user is then presented with a list-view or map-view of properties within a small range of the specified location(s) with basic information about price, address and images.

These first few steps are vital for customer conversion, that is, getting the customer from the start of the website to completion of an actual booking. For the majority of websites, the customer does not need to login, they are simply looking for information. If a login was required, most customers would be discouraged from continuing any further thus reducing their website's conversion rate. However, the lack of login means that there is a possibility for missed opportunities for gaining intelligence about customer behaviour when using the website. This might be data about how the customer

Once a customer has made a booking, there is minimal intervention from the website. The renter and host are able to exchange messages but overall there isn't much further interaction. The host usually provides renters with a set of rules and the OTA only acts as conflict resolution. This minimal intervention approach is common among OTAs but again represents a possible source of missed opportunity to provide the renters and hosts with value or service.

The Effect of User Reviews

The introduction of user reviewing has had a major impact on the accommodation industry. The equalizing nature of a peer-to-peer system has meant that everyone user on these forums is able to express and evaluate their experiences to the rest of the user population. According to Gu et. al (2009) there is a *significant* relationship between user reviews of hotels and their financial performance as a business. This is perhaps unsurprising as personal experience will inform you that you are heavily influenced by ratings and reviews for products that you view online. Interestingly, however, there is an asymmetry between positive and negative ratings (Avello et al. 2017). If a product has an overall positive rating, then the persuasiveness depends only on if there is a high total of reviews. However, if there is an overall negative rating, then there is no correlation between how effective it is on a user and how many negative reviews there are, that is, only a few are sufficient to dissuade.

The landscape of reviews is complicated further as they are often created as a strong emotional response to an experience. It has been suggested (Gupta et al. 2011) that negative reviews with emotionally charged language are often less compelling as users often attribute the review to an irrational disposition of the reviewer rather than the product. Conversely, positive reviews are taken to apply to the product rather than the emotional character of the reviewer. The informativeness of a review is, therefore, a complicated issue and one suggests that reviews are not taken simply at face value.

Project Purpose

Aim

The Unreal Estate team set out to use what has been learnt and successfully applied by many of the leading accommodation platforms such as Trivago and AirBnB. The goal was to create a peer-to-peer accommodation rental platform with an increased focus on personalisation and user autonomy. Ultimately, the purpose of this project was to explore potential minor but important improvements to the online accommodation industry.

System

The scope of the work was initially to create a strong basic system with extended focus on personalised features. Therefore in terms of the fundamental features of an accommodation system our platform started with the property management component which includes key functions for listing, editing, and deleting properties. There is also the ability for both advertisers and renters to cancel bookings to allow for autonomy over their experiences and properties. Users have the ability to leave reviews and ratings on any properties stayed at which can be viewed by other users. In terms of personalisation, booking users have a route generated from their home to the property on booking to provide them with convenience.

The Unreal Estate project completed the basic system and progressed into extending the review capacity to address the issues mentioned above. The platform allows for advertisers to perform machine learning evaluations on the reviews of their properties in order to combat unfair and inconsistent reviews. There is an overall noise removal and smoothing effect of this machine learning which will be discussed in the following overview and in depth in the review contribution section.

Project Design

Overview of Components

Advertising

Advertising module is designed for advertisers for property listing. After login, the owners will be able to add their properties for rentals. The module will allow them to add or update images, price, address, building type, number of rooms, number of beds, number of bathrooms and the number of guests allowed per room.

Features:

- Advertising page will not available unless the user is logged in
- After login, the owner can navigate the sidebar and be redirected to the advertising page.
- From advertising (owned property listing) page, the user can add new properties by clicking “Add New Property” button at the bottom of the list.
- User is able to fill the form (including upload images) and click “Continue” to be redirected to preview page or return to the previous page by clicking “return” (but the form will not be saved). It may take some time to be redirected to preview page if the image is huge.
- From the preview page, the user is able to add the new property by clicking the “confirm” button
- From the advertising page, the user is able to update property by clicking “Update” beside each property in the list. User will be redirected to the form.
- From the form, property information is preloaded, and the user is able to update the information and submit the form by clicking “Update”. Or the user can return to the previous page by clicking “return” (but the new changes will not be saved).
- From the advertising (owned property listing), user can delete the property by clicking “DELETE” button beside each property in the list.

Review

The user will have the ability to review a property after their period of accommodation has ended. This will be a rating between 1 - 5 in terms of stars in increments of 1. Along with this, the user will get to share their feedback and review in a text box.

This review will be attached to the user, the owner of the accommodation, and the accommodation itself. This means that if another user is viewing the property, they will see all reviews for that property and the average rating of that property.

The user will also be able to visit a page with all their bookings in their account, and for each, if the period of accommodation is over, then the user will be able to submit a review, however, if they already have, then they will only be able to view the review. When leaving a review, the user will be anonymous in their review such that the owner and other users will not be able to see who left the review.

The property manager can go into the 'My properties' page, and then for each property, they can view the reviews on that property. They can then use our inbuilt machine learning functionality in order to detect any misleading reviews where the rating appears to be different from the overall tone of the review text. Using the machine learning network, the property manager can choose to replace any mismatched reviews if the difference in rating between the user and the AI is greater than or equal to 2 stars. However, this needs to be done for all the reviews on the property since we don't want the property manager to be able to pick and choose higher ratings in order to boost the property's overall average rating. The details of this module are explored further in the following section in contribution 1.

Search

The Accommodation Search Module is for users to make detailed searches on properties. The visitors will be able to choose accommodation preferences and receive a list of suggestions.

Features:

- A search bar for the visitor to input locations(street, unit number, suburb, state, postcode), with autocomplete feature that will try to match the input to an address using the Google Maps API.
- User is able to choose the start date and end date of accommodation
- User is able to enter the number of guests that will be staying at the property.
- The result will be a list of properties within 1km of the search destination, the correct number of guests, and availability within the date period. With further information that the user can navigate with.
- The search results are paginated such that only a subset of the results are displayed at first, then the user can choose to view more by scrolling to the bottom and clicking to load more results.
- Search results will show all essential qualities of a property, including the title, an image, the average review score, the suburb, number of bathrooms, bedrooms, and number of guests allowed, price, and the building type.

The information presented to the user throughout the searching process should be simple and relevant in order to assist them to make the correct choice about the properties they would like to view. This module will be described in more detail in the Search Contribution, in the following section of this report.

Booking

Naturally, a fundamental aspect of all accommodation platforms is the ability to make bookings. The Unreal Estate platform allows authenticated users to make bookings with any of the listed properties. Booking a property is done via the search function. Once a property is selected from the search, the user will be presented a detailed summary of the property. If a user is not logged in, there will be a button at the bottom which redirects the user to login. If the user is logged in this button will allow the user to continue to a booking form.

Features:

- Booking form allows the user to modify the originally searched check-in and check-out dates
- Booking form allows the user to modify originally searched number of guests (to a maximum of the properties allowed guests)
- Booking form presents the total price
- Bookings are able to be cancelled by both owners and renters after a booking has been made

This module is detailed in the following section of the report in the Booking Contribution.

User Account Management

The user account management module describes the component of the website which handles all user authentication. This module is key for every other module apart from basic searching (which still depends on users being able to advertise). All users who want to interact with other users need to be associated with an account. Each account includes their personal details as well as their trips, their properties and the bookings associated with each property.

Features

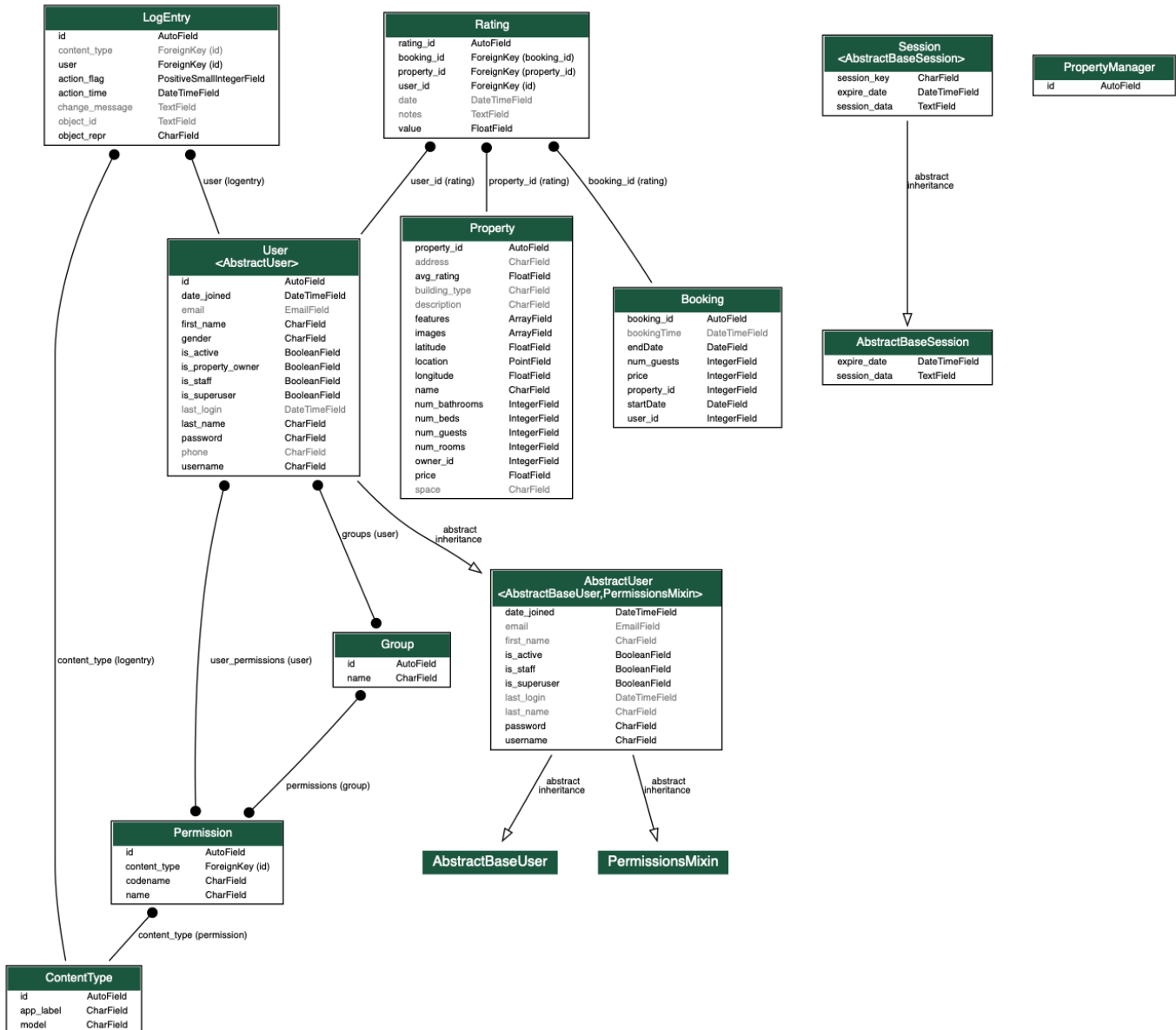
- Users authenticate with a username and password
- The username is equivalent to their email address
- Users have access to past, present and future bookings they make
- Users have access to past, present and future bookings made with their properties
- Users have access to list their own properties
- Users have access to review properties they have booked

This module is explained with images in the User Account Contribution section.

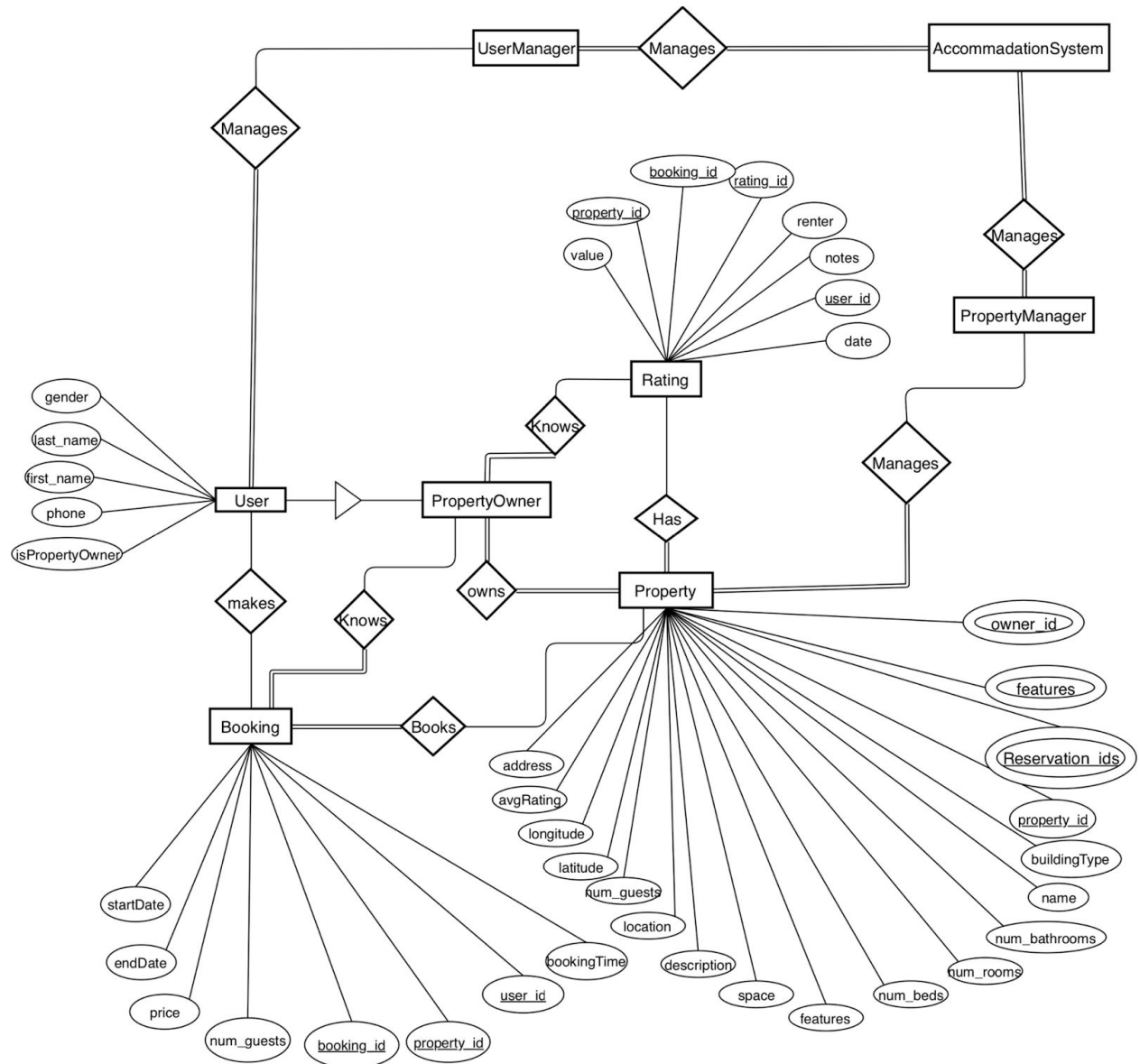
Diagrams

UML (use link for full image)

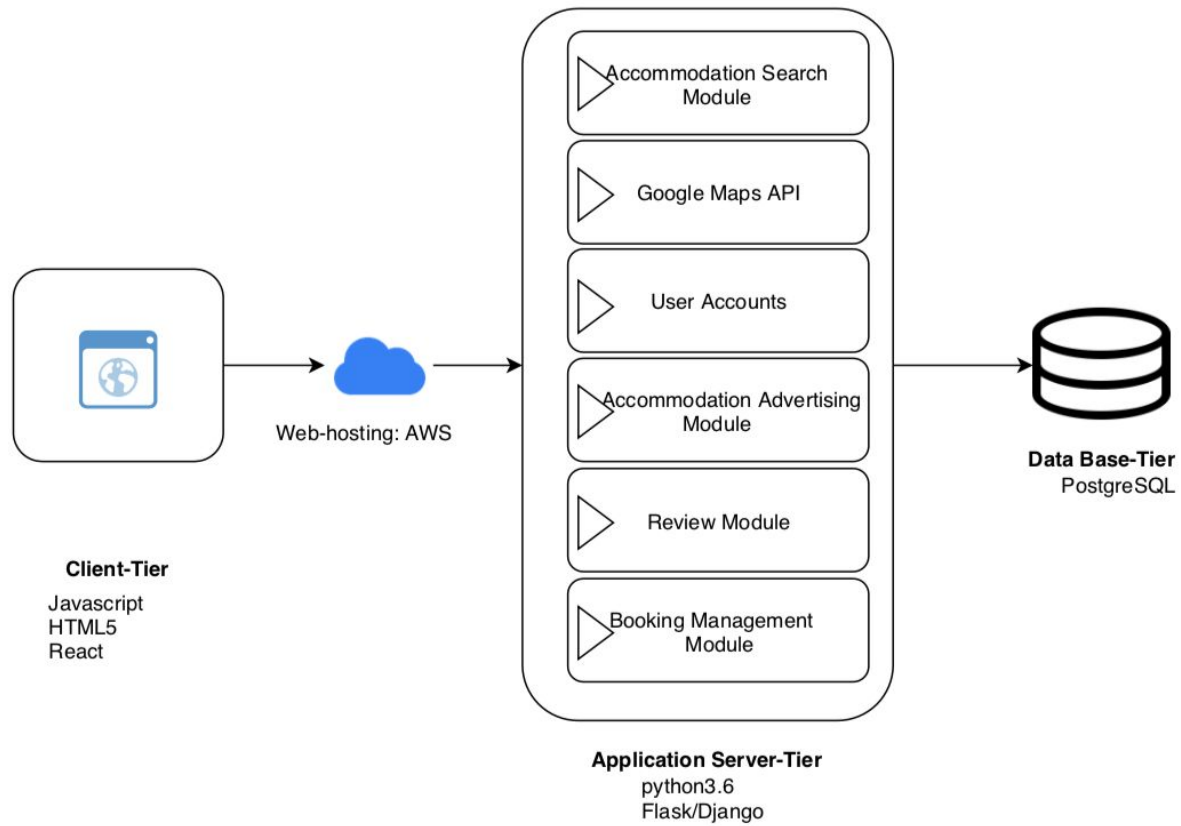
<https://drive.google.com/file/d/1uFgrQEfz3xBNzXeEKdLPzRccEEi1Lbp/view?usp=sharing>



ER Diagram



Software Architecture Diagram



User Scenarios

All Users

The main users are renters and advertisers. Once the homepage is loaded, a request is sent to Google Maps API to fetch the current location of the user based on their IP address. This information is used to present users with default properties on the homepage before a search has been made.

Signup Form

First Name

Last Name

Gender

Male

Phone Number

Email

Password

Confirm Password

CONTINUE

Before logging in, both renters and advertisers need to make *sign-up* and *sign-in* requests. The *sign up* request is made with function *EnterSignUpDetails* to create an account. Then, when the user tries to log in, a request is handled by *CheckUserDetails* function which queries the AWS Cloud Database, and it will respond as either failure or success. Upon a successful response, the user is logged in.

Login

Email

Password

CONTINUE

Renter

After logged in, a renter can fill the search fields on the homepage and make a search. During this process, requests are sent to google maps API for address autocomplete.

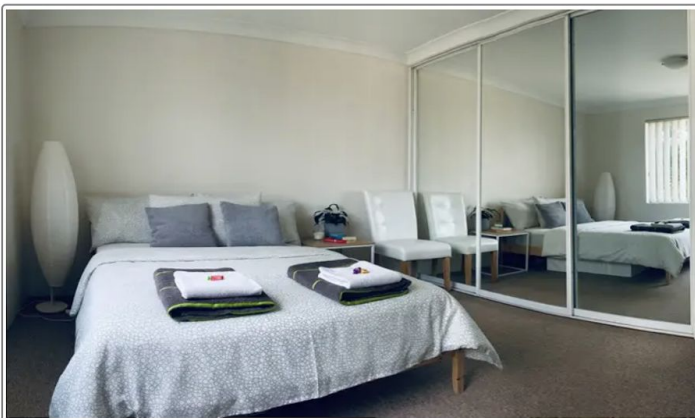
Enter an Address Check In Check Out Number of Guests

- ☐ Kinsale County Cork, Ireland
- ☐ Kinston NC, USA
- ☐ Kinshasa Democratic Republic of the Congo
- ☐ Kinsarvik Norway
- ☐ Kinsmen Park Edmonton, AB, Canada

powered by Google

After the renter clicks *search*, a request is sent to the database, and the *PrimarySearch()* function will filter out the irrelevant results and return a list of properties, which is then displayed on the search results page.

Enter an Address Check In Check Out Number of Guests



Spacious private room near CBD

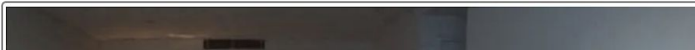
Apartment

☐ 1 ☐ 1 ☐ 3

☐ Kensington, NSW, Australia

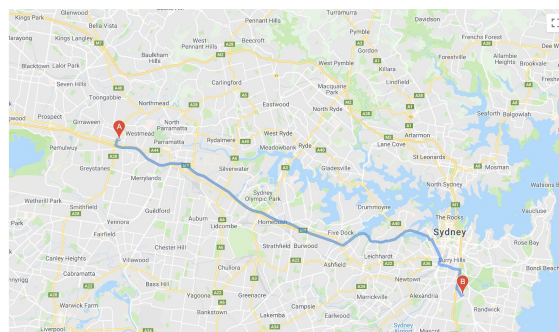
★ 4.4

Price: \$64




All you need for Christmas in Sydney

When the user confirms the booking, a request is sent and the function *BookingFunction()* in the booking model will create a new booking. In the meantime, a request is sent to the Google Maps API to display the route from the current location to the property location on a map.




When a user clicks *Bookings* on the sidebar, a request will be handled by the function *bookingDetailsBID()* to fetch the booking object by booking ID (BID).

Booking Confirmation: 76



Spacious private room near CBD

 Kensington, NSW, Australia

Check In: 2019-08-11

Check Out: 2019-08-12


Total Price: \$64

[CANCEL BOOKING](#) [RETURN TO HOMEPAGE](#)

Advertiser




After logging in, the advertiser can request for the list of owned properties by clicking the “My Properties” button on the sidebar. This request will be handled by function *list_property()* in advertising model, which returns a list of properties or an empty list. Request for deleting, updating and adding new property are handled by *PropertyFunction()* in advertising model.


[ADD NEW PROPERTY](#)



An Oasis in the City !

Apartment

 1  1  1

 Potts Point, NSW, Australia

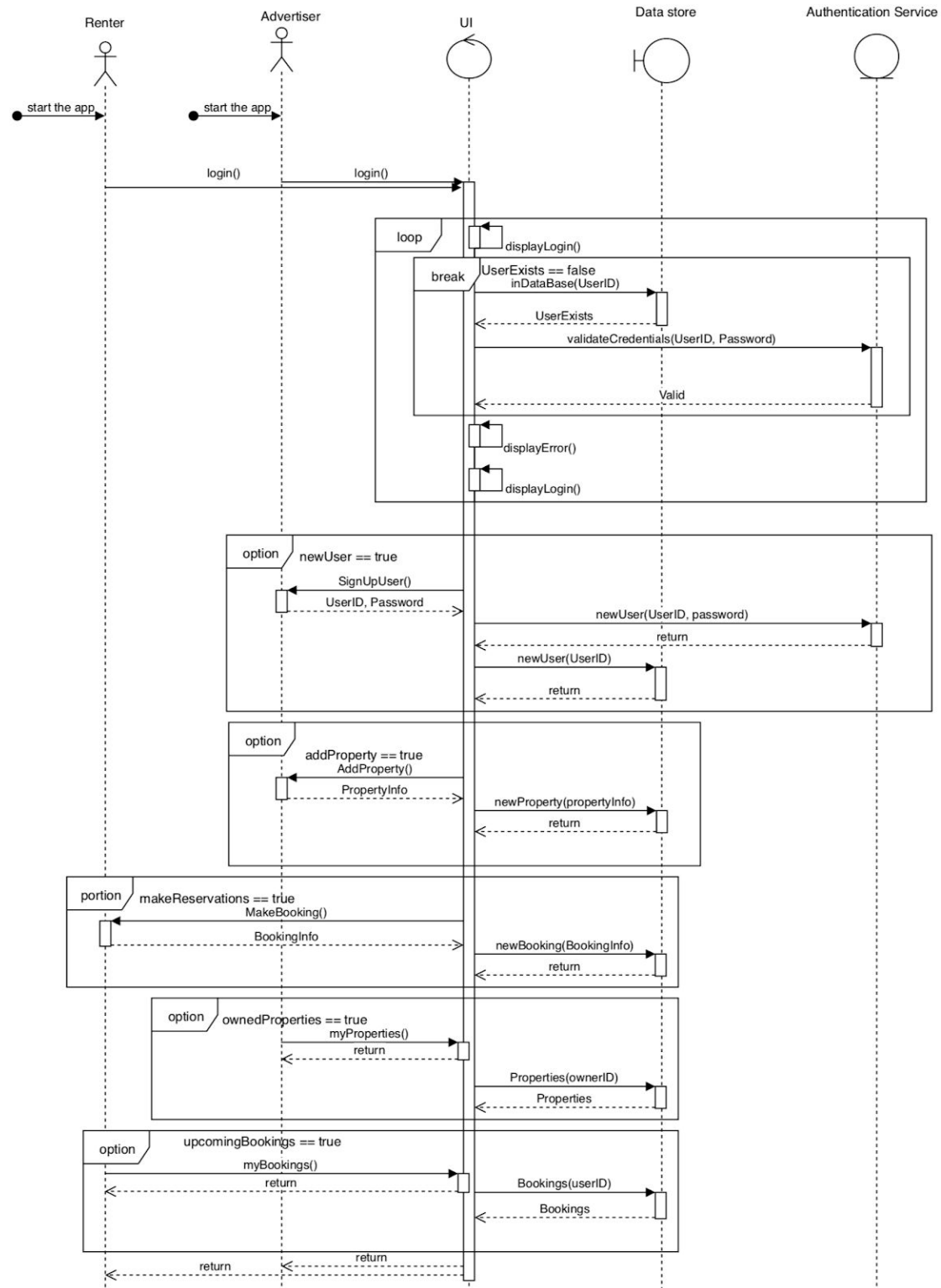
★ 4.7

Price: \$100

[RESERVATIONS](#) [REVIEWS](#)

[EDIT](#) [DELETE](#)

Sequence Diagram

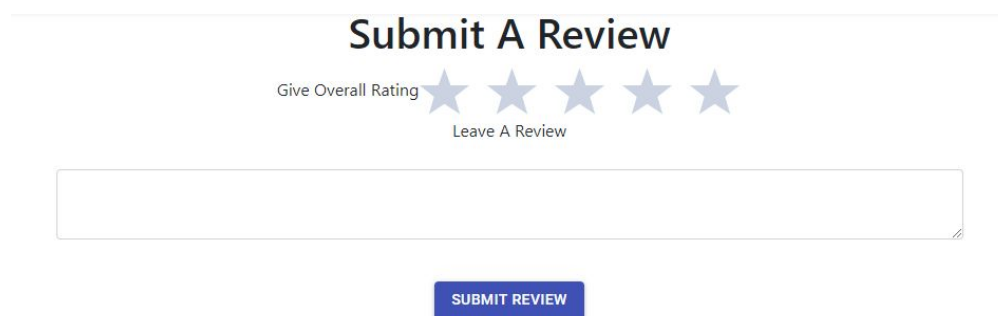


Review Contribution

There are two parts to this component

- 1) Adding a review to a booking
- 2) Classifying a rating based on the text in the review

Adding a review is done by first having rented some accommodation, and then after the rental is over, the user will be able to review their stay from the Bookings page (accessible from the side menu). This review is made up of a rating from 1 -5 stars (increments of 1) and a text field, where the user is able to share more about their experience as shown in the image.



This input will be added to the database along with the booking_id, property_id, and user_id it is associated with as a review. These reviews can then be seen on the Property page, as well as the overall 'average' of the reviews.

If a user wants to view their review after leaving it, they can do so by clicking 'View Review' from Bookings (this will change from Write Review after a review has been left). The Property Manager can also view all reviews for any of their properties in order to keep up to date with the user's experience with their property.

The other part is that the property owner can also utilise our inbuilt machine learning algorithm to detect reviews that have ratings that are contradictory to the overall sentiment left in the text review. After this detection the property manager can choose to overwrite the User rating with the Neural Network's rating however, this will only be done on Reviews that have a difference greater than or equal to 2 stars between the User and AI rating. Also, this can not be done selectively i.e per review and must be done to all reviews on the property as to prevent the property manager from just using this feature selectively when it increases their properties average rating.

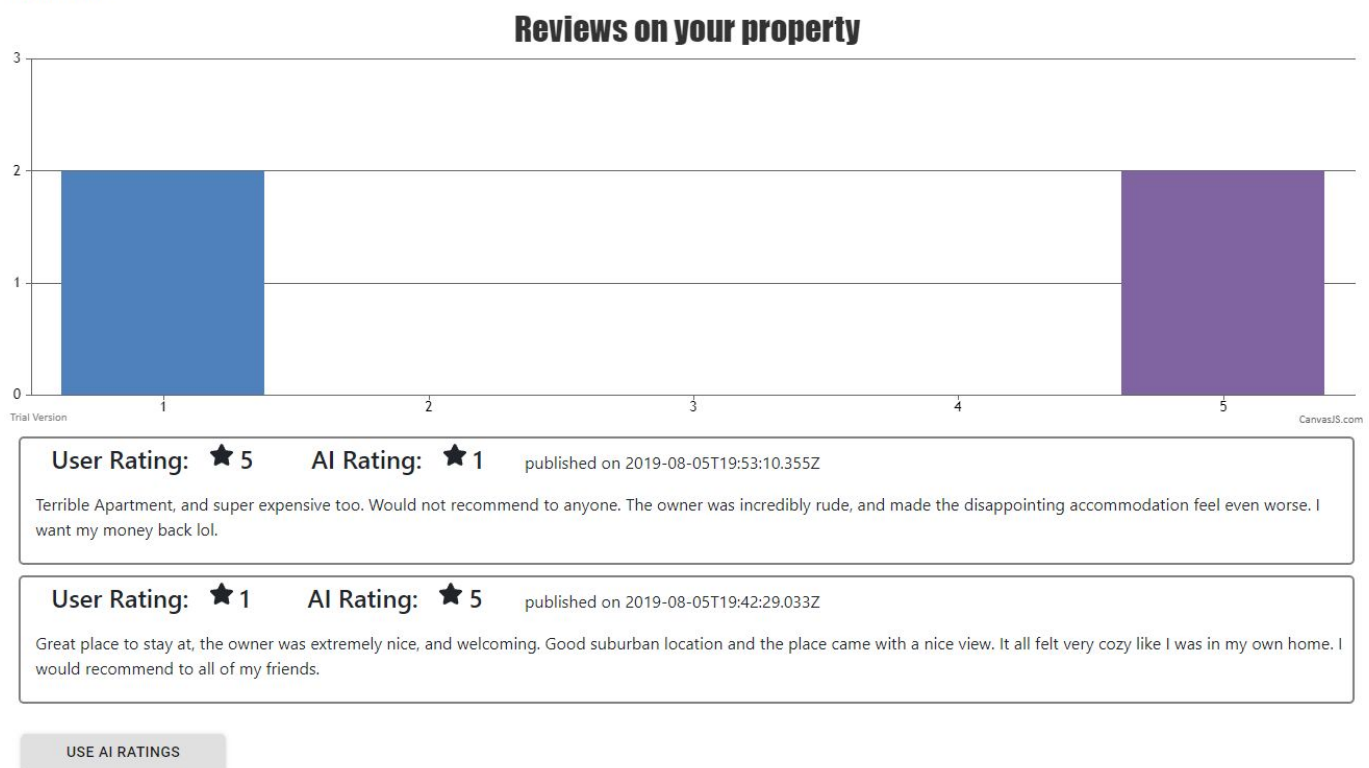
This is done by retrieving the property_id of the selected property, then the backend will retrieve all reviews on that property, and feed each review into the neural network classifier. The classifier will make predictions on each review based on the context of the text, and return a

rating between 1-5 stars. We can then return this to the frontend in order to display the neural network's ratings, where the property manager can then make the decision whether or not to use the user or network's ratings.

Here is an example below of two very misleading reviews, where in the first case, the User gives a 5-star rating, however, proceeds to write very negative feedback based on their experience. The trained neural network is able to detect that this should indeed be classified as a 1-star rating. In the second case, the User gives a 1-star rating, however, proceeds to write very positive feedback based on their experience. Again, our neural network is able to detect that this review should have been classified as a 5-star rating.

Now the property manager is able to use the button to use the AI's ratings instead, and while this may not increase the property's average rating, it will lead to more consistent reviews, and filtering out fake reviews.

Reviews



Another change that was made to reviews in order to increase the reliability of an average rating, was to use a 'trimmed mean' in which we filter/trim out the top and bottom 10% of ratings such that any outliers would not be accounted for in the average. This turned out to be a good method as not many properties have more than 10 reviews, and hence most properties have a real average, however, the properties that do have more than 10 reviews have a much more consistent average if there is a case of outliers.

Both of these features are implemented in order to help both renters and owners to have more consistent reviews at a glance, which in turn may persuade/dissuade users accordingly, and help owners get the feedback they need in order to improve their properties/service.

The Neural Network model was made using TensorFlow. The model is a bidirectional recurrent neural network, where each directional layer is multilayered, and made of two LSTM layers. Both directional layers are the same in that they have a layer of 50 LSTM cells, feeding into 100 LSTM cells. Some training choices include training in batches of 100 reviews, using dropout of 0.6 for a more flexible network, a small learning rate of 0.005 such that learning doesn't blow up.

We trained the model on an unofficial dataset of TripAdvisor reviews, which was scraped from their website, and can be found here (<https://www.cs.cmu.edu/~jiweil/html/hotel-review.html>). This did involve some challenges such as reviews in different languages, however, this was avoided by filtering out reviews in languages other than English. We ended up with 45000 total reviews per star from 1-5 and then split these 45000 into an 80/20 split for training/testing.

We also used the word to vector embeddings by GloVe, this will be automatically downloaded the first time you use the classifier on the website (as the file was too big to push to Github). For more information, you can visit (<https://nlp.stanford.edu/projects/glove/>).

Our neural network ended up with an accuracy of ~63% as shown in the figure below, which is better than some other models that have been used to do this task, such as the following: <http://www.developintelligence.com/blog/2017/03/predicting-yelp-star-ratings-review-text-python/>

```
→ ml git:(master) * python3 runner.py eval
Evaluation run
Local Embeddings pickle found, loading...
Loading Review Data...
Parsing 45000 files
Loaded and preprocessed 45000 samples for evaluation
2019-08-10 10:10:33.070164: I tensorflow/core/platform/cpu
FINAL ACC: 0.6298444430033366
```

Search Contribution

The Accommodation Search module was built so that the user can efficiently and intuitively search for their desired accommodation without having to login/signup such that they can simply check what properties are available without the hassle.

We have added some additional features on the frontend to make searching as effortless as possible, such as the automatic translation of the user's input to an address as shown in the image below. Once the user begins to type a location, we have incorporated the Google Places API in order to translate this text, to some popular places that match the input so far. As can be seen in the image, the frontend will display 5 potential places that are closest to your input as determined using the API, after which, the user may simply select one, and the address will be auto-filled. This makes it much easier to input an address for the user, especially if it is a popular location.

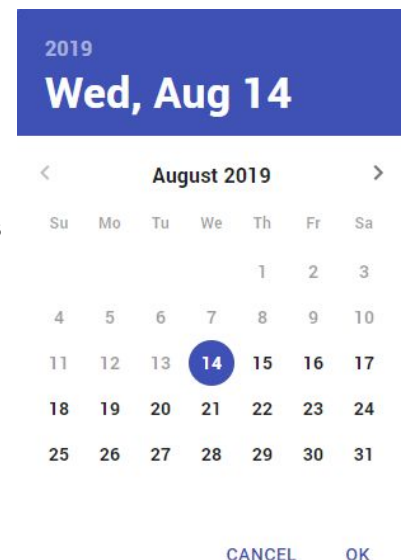
Enter an Address

Q University of New South

- University of New South Wales, Randwick Campus Randwick NSW, Australia
- The University of New South Wales Sydney NSW, Australia
- University of New South Wales at the Australian Defence Force Academy (UNSW Canberra) Northcott ...
- University of New South Wales, Cliffbrook Campus Coogee NSW, Australia
- University of New South Wales, Canberra Campus Tobruk Road, Campbell ACT, Australia

powered by Google

When selecting the check-in and check-out dates on the search page, they have been restricted such that the user is not able to input invalid data, so that it is as easy as possible for users to enter the correct dates. Some of these restrictions include, not being able to choose a check-in date previous to the current date, and not being able to choose a check-out date before the check-in date. An example of this is shown in the image on the right, where a user is trying to select a check-out date after selecting the check-in date as the 13th August. Hence, all check-out dates before the 14th of August have been greyed out, and unselectable.



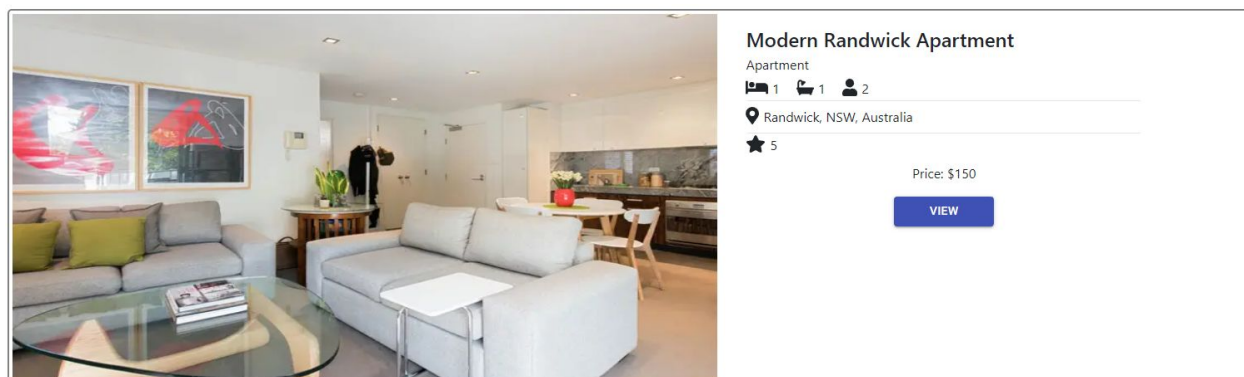
The number of guests to be selected is defaulted at 1 and is a drop-down menu between 1 and 10 as we don't allow for more guests than 10 at any accommodation.

Once the search button has been clicked, the frontend will send the input in the search form to the backend, the backend will retrieve all properties that are available throughout the check-in and check-out period, properties within 1 kilometre of the inputted address, and then properties with the number of guests in the input.

This is then sent back to the frontend, where it displays the list of results as shown in the image below. The list is paginated as to increase the efficiency in rendering the results, and to also not overwhelm the user with so many results. The user can then load in more results if they scroll to the bottom and click on the button that says "Show More Properties".

To obtain all properties within a certain distance is done by taking the address that was inputted and using the Google Maps API to geocode the address into a latitude and longitude. This is then created as a Point using Django, which can be used to lookup in the database to get locations within a certain distance.

The results appear as a list of properties, where each property is displayed in a tile as shown in the image below. As shown in the image, the results will include the title of the property, the number of bedrooms, bathrooms, and guests available in the property, the suburb of the property, the average rating, and the price per day.



After the user has found the property they are interested in, they can click the View button in order to be redirected to a page where the user will be able to view more details about the property including a more accurate description of the space, and the reviews left for the property.

Booking Contribution

The booking module as mentioned in the overview is a key component of any accommodation platform. For our peer-to-peer system, this removes a key barrier of entry into the renting and owning market. Through this module, the two user groups are directly connected and transactions can be completed without the use of an agent.

The booking model comprises a primary booking id, the renter id, the property id, the check-in date, the check out date, the number of guests, the total cost and the time the booking was made. This is the minimal set of fields required to uniquely and accurately identify a booking.

Bookings can be submitted after selecting a property from the search results. If a user is not logged in the button will be displayed as *Login to book* however if they haven't logged in yet the button will be displayed as *Book*.

needed. Contact: May

Price per night: \$59

LOGIN TO BOOK

Reviews

published on 08/12/2017

needed. Contact: May

Price per night: \$59

BOOK

Reviews

published on 08/12/2017

Once a logged-in user clicks the book button they are redirected to a booking page with a small summary of the property above and a form below which automatically populates with their original search parameters.

3 1 4

Randwick, NSW, Australia

★ x.x

Price: \$140

1

2

3

4

Check In August 29th

Check Out August 30th

Numt Gue

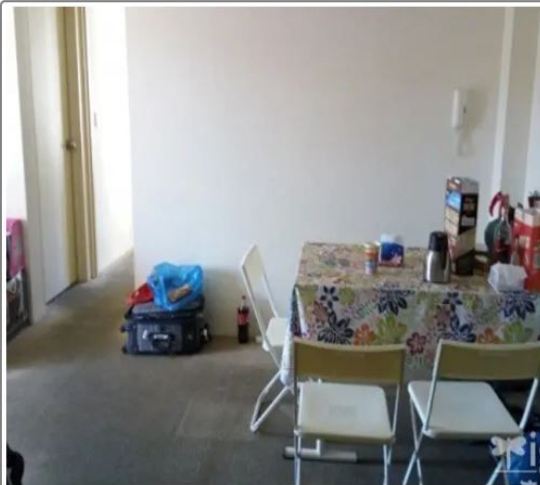
Confirm

Total Price: \$140


The form contains fields for check-in date, check out date and number of guests and can be edited without exceeding the limitation on the number of guests. The form also automatically updates the total price based on the number of nights.

Once this form is submitted the booking is sent to the database on the AWS server. The response from the database is returned as a confirmation page with the booking id and all of the details of the booking. The confirmation page has the option to immediately cancel the booking, in case a mistake had been made or also to return to the homepage. Below the confirmation is a personalisation feature which gives a driving route from the user to their property using the Google Maps API as mentioned above in the renter user scenario.

Booking Confirmation: 83

A photograph showing the interior of a property. It features a table covered with a floral tablecloth, several chairs, and some items on the table like a coffee maker and bottles. A doorway is visible in the background.

AUD58/9t, direct bus to city, opera house

 Kingsford, NSW, Australia

Check In: 2019-08-20

Check Out: 2019-08-27

Total Price: \$413

[CANCEL BOOKING](#)

[RETURN TO HOMEPAGE](#)



After the booking process is complete, the renter can view all their bookings using the Bookings tab in the side navigation bar. On this page users are able to cancel their bookings, write reviews and view previously submitted reviews. For owners of a property they are also able to view the bookings on their properties by clicking on My Properties in the side navigation bar. The bookings are connected to each property and can be cancelled from this page as well.

User Account Contribution

User account management module provides three main functional components: user login, user sign up and their personalised navigation bar. The main navigational component is the navigation bar, which helps to redirect the user to their profile, properties and bookings. Any new user can sign up by clicking the “Signup” button at the top of the webpage. Then the user will be redirected to the sign-up form, and the user can complete the form by filling first name, last name, email and password which are the fields in the database representation of a user.

Signup Form

First Name

Last Name

Gender

Phone Number

Email

Password

Confirm Password

CONTINUE

Sign up validation is handled by both frontend and backend. If the fields are valid, the user will successfully sign up and be redirected back to the home-page. Then, from the navigation bar on top, the user can be redirected to the login page by clicking “login”. The user is logged in and authenticated via their username and password.

Login

Email

sample@sample.com

Password

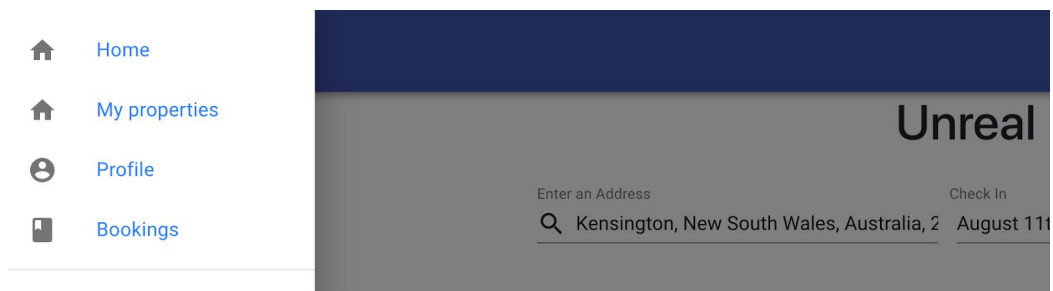
...

CONTINUE

The backend provides validation on the user credentials. On success, the user is redirected to the home-page and is granted access to more functionalities. On failure, a toast will come up notifying the user what goes wrong.

Either email and password don't match ✕

After successfully logged in, the “Profile” button will lead the users to profile page where users can edit their own profile.



In terms of the non-functional features for the account module, the backend monitors user activity using JWT Authentication. This REST framework passes authentication tokens for the user through the request headers from the frontend. The token is then used for all backend requests to associate a user with the request. These are the key aspects of the user management contribution which underpins the entire peer-to-peer platform.

Conclusion

The overall Unreal Estate platform was successful with all of the planned epics finished and the scope was extended beyond what was first anticipated. The major components of User Account Management, Advertising, Searching, Booking and Reviewing were all completed. The project provided every member with challenges from which they all learnt from and progressed. Initially, each member had a different foundation of skills and this was something that needed to be taken into consideration at every stage.

Reflecting on the project, the things that were accomplished well were all of the major functionalities. Each component was completed and integrated cleanly into the whole system. There are no major bugs with any of these components. On the other hand, the challenges with the project were mainly to do with the learning curve associated with the new frameworks as well as the asynchronous calls to the database. The frontend framework could be improved with minor adjustments which would enable more interactions. For instance, it would have been preferable to include a map on the search results page, either individual maps for each property or one overall map with all properties included. This would have correlated with our desire to create a more personalised platform.

The major area for further exploration would be with the review system. The machine learning which was applied was an effective proof of concept which demonstrates that there is potential to improve future rating systems. One avenue for further research could go into purely statistical methods for smoothing reviews and removing noise which would have a similar effect to what our neural network model accomplished. Another possibility would be to use the model to identify particularly emotional language which might not be representative of informative or productive feedback. Overall, however, the project was successful in complementing the current state of the online accommodation industry as well as suggesting further improvements.

References

https://docs.rightscale.com/cm/designers_guide/cm-cloud-computing-system-architecture-diagrams.html

Docs.rightscale.com. (2019). *Cloud Computing System Architecture Diagrams*. [online] Available at: https://docs.rightscale.com/cm/designers_guide/cm-cloud-computing-system-architecture-diagrams.html [Accessed 20 Jun. 2019].

<https://www.wttc.org/-/media/files/reports/economic-impact-research/regions-2019/world2019.pdf>

Wttc.org. (2019). [online] Available at: <https://www.wttc.org/-/media/files/reports/economic-impact-research/regions-2019/world2019.pdf> [Accessed 20 Jun. 2019].

<https://www.tandfonline.com/doi/full/10.1080/13683500.2013.827159>

Taylor & Francis. (2019). *Airbnb: disruptive innovation and the rise of an informal tourism accommodation sector*. [online] Available at: <https://www.tandfonline.com/doi/full/10.1080/13683500.2013.827159> [Accessed 20 Jun. 2019].

<http://stec.univ-ovidius.ro/html/anale/RO/2016/2016-II-full/s3/21.pdf>

Pentescu, A. (2019). *Millennials, Peer-to-Peer Accommodation and the Hotel Industry*. [online] Stec.univ-ovidius.ro. Available at: <http://stec.univ-ovidius.ro/html/anale/RO/2016/2016-II-full/s3/21.pdf> [Accessed 20 Jun. 2019].

https://www.researchgate.net/publication/313448914_A_comparison_of_the_merchant_and_agency_models_in_the_hotel_industry_P_Liao_et_al

Liao, Peng & Ye, Fei & Wu, Xiaoli. (2017). A comparison of the merchant and agency models in the hotel industry: P. Liao et al.. *International Transactions in Operational Research*. 10.1111/itor.12365.

<https://www.sciencedirect.com/science/article/pii/S0278431917304401>

Martin-Fuentes, E., Fernandez, C., Mateu, C. and Marine-Roig, E. (2018). Modelling a grading scheme for peer-to-peer accommodation: Stars for Airbnb. *International Journal of Hospitality Management*, 69, pp.75-83.

<https://journals.sagepub.com/doi/full/10.1177/0047287513481274>

Filieri, R. and McLeay, F. (2013). E-WOM and Accommodation. *Journal of Travel Research*, 53(1), pp.44-57.

<https://www.tandfonline.com/doi/full/10.1080/02642069.2010.485640>

Smithson, S., Devece, C. and Lapiedra, R. (2011). Online visibility as a source of competitive advantage for small- and medium-sized tourism accommodation enterprises. *The Service Industries Journal*, 31(10), pp.1573-1587.

<https://www.jinfont.com/resources/bi-defined/3-tier-architecture-complete-overview/>

- JReport. (2019). *3-Tier Architecture: A Complete Overview - JReport*. [online] Available at: <https://www.jinfony.com/resources/bi-defined/3-tier-architecture-complete-overview/> [Accessed 20 Jun. 2019].
<https://codeburst.io/software-architecture-the-difference-between-architecture-and-design-7936abdd5830>
- Aladdin, M. (2019). *Software Architecture - The Difference Between Architecture and Design*. [online] codeburst. Available at: <https://codeburst.io/software-architecture-the-difference-between-architecture-and-design-7936abdd5830> [Accessed 20 Jun. 2019].
- <https://webflow.com/blog/the-web-design-process-in-7-simple-steps>
- Meazey, M. (2019). *The web design process in 7 simple steps* | *Webflow Blog*. [online] Webflow.com. Available at: <https://webflow.com/blog/the-web-design-process-in-7-simple-steps> [Accessed 20 Jun. 2019].
- <https://www.sciencedirect.com/science/article/pii/S0278431908000546>
- Ye, Q., Law, R. and Gu, B. (2009). The impact of online user reviews on hotel room sales. *International Journal of Hospitality Management*, 28(1), pp.180-182.
- <https://www.sciencedirect.com/science/article/pii/S0261517717302376>
- Gavilan, D., Avello, M. and Martinez-Navarro, G. (2018). The influence of online ratings and reviews on hotel booking consideration. *Tourism Management*, 66, pp.53-61.
- <https://www.sciencedirect.com/science/article/abs/pii/S0148296311001305>
- Kim, J. and Gupta, P. (2012). Emotional expressions in online user reviews: How they influence consumers' product evaluations. *Journal of Business Research*, 65(7), pp.985-992.
- <https://www.sciencedirect.com/science/article/pii/S027843191831168X>
- Li, H., Xie, K. and Zhang, Z. (2019). The effects of consumer experience and disconfirmation on the timing of online review: Field evidence from the restaurant business. *International Journal of Hospitality Management*, 84, p.102344.

Appendix

Installation manual for windows

Prerequisites to run software are:-

- Python3
- Anaconda
- NPM
- Node

Please follow the commands to run the server :-

1. Open PowerShell terminal and go to directory capstone-project-unreal-estate
2. Run the command “./installFiles.ps1”
3. Run the command “conda install gdal” (Installation with anaconda is required as pip does not configure it properly.
4. Run the server by “python3 ./unreal_estate/manage.py runserver”
5. Open the browser and go to “<http://127.0.0.1:8000>”

Languages and Technologies

- Python
- HTML
- CSS
- Javascript
- React
- Django
- PostgreSQL
- Google API
- Cloudinary API
- TensorFlow
- JWT

Source code structure

The code is divided into the backend and frontend. The backend is based on the Django framework and the frontend is based on the React framework. The Django project is divided into smaller apps which correspond to the modules mentioned in the overview.

The backend code is written in different apps these apps represent different modules of the project. The backend project dependencies are listed in the “requirements.txt” file.

All the frontend code is written in the client app. The frontend dependencies are listed in the file “package.json”. All javascript code is written in src/components folder in the client app. All CSS files are in folder src/css in the client app.