THE UNIVERSITY OF NEWSOUTH WALES

COMP9900 Information Technology Project

Project Proposal

Topic3 – Accommodations Web Portal

Team: SkilledDriver

Xinyuan Qian | z5147182@unsw.edu.au | z5147182 | Scrum Master/Developer

Kai Fang | z5137591@ad.unsw.edu.au | z5137591 | Developer

Dan Liu | z5094271@ad.unsw.edu.au | z5094271 | Developer

Yihao Wu | z5154763@ad.unsw.edu.au | z5154763 | Developer

Submission date: 12 AUG 2018

1 Background

Nowadays, most people at some point in their life will have an urge to travel. This may be a one-week holiday to somewhere warm to top up your suntan or this could be a life-changing year, or longer, trip. Everyone has their own reasons that they decide to travel but they all got the same demand, that is, accommodation. Additionally, as most of them are busy with life, the time-consuming and effort-consuming process of finding and reserving the accommodation is sort of roadblock of their travel. Especially when people tend to acquire most of the information online while most of the existing websites (like Airbnb and Booking.com) are overloaded with information and, therefore, more or less opaque and useless. People have a strong desire to find accommodation in a much clearer and more trustworthy website to save their time and effort. On the other side, property providers also need such a clear website to fulfill their holiday units' renting schedule as much as possible. For this reason, providers are forced to present their products or services in an appropriate manner on the web. The most advanced advantage of our web portal is that both providers and visitors will get what they all want on our website with as less noise as possible.

Another drawback of existing accommodation web portal is that once providers uploaded their properties along with corresponding rules, they have no chance to fine-tune the rules to fit visitors' requests. Flexibility is another important factor our web portal will concentrate on. Being more flexible means visitors will have more choices and providers will also be happy to have the advantage to decide whether to seize the "second-class" offer or not.

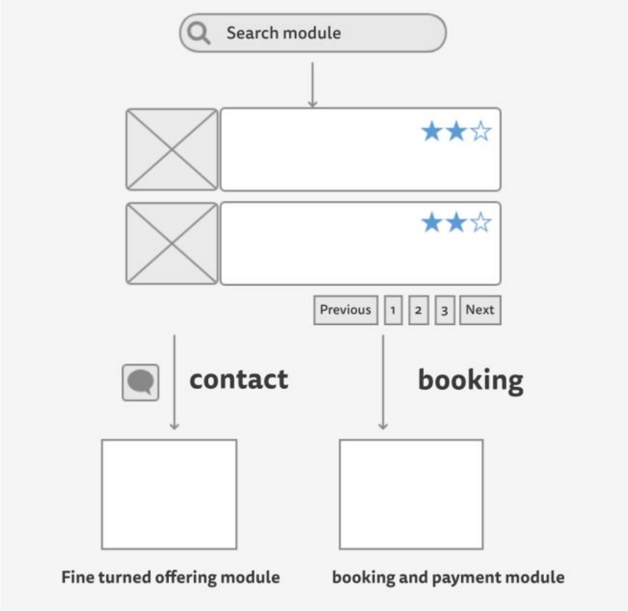
Additionally, apart from accommodation, people nowadays are used to searching almost everything online. Existing websites like Airbnb are sort of too deep into the specific territories, one purpose at all. It is totally a new work to transplant the web portal to another field. In this case, we are going to focus on the creation of a generic, modular architecture for building web portals that can be used for accommodation or other similar services.

2 Methodology

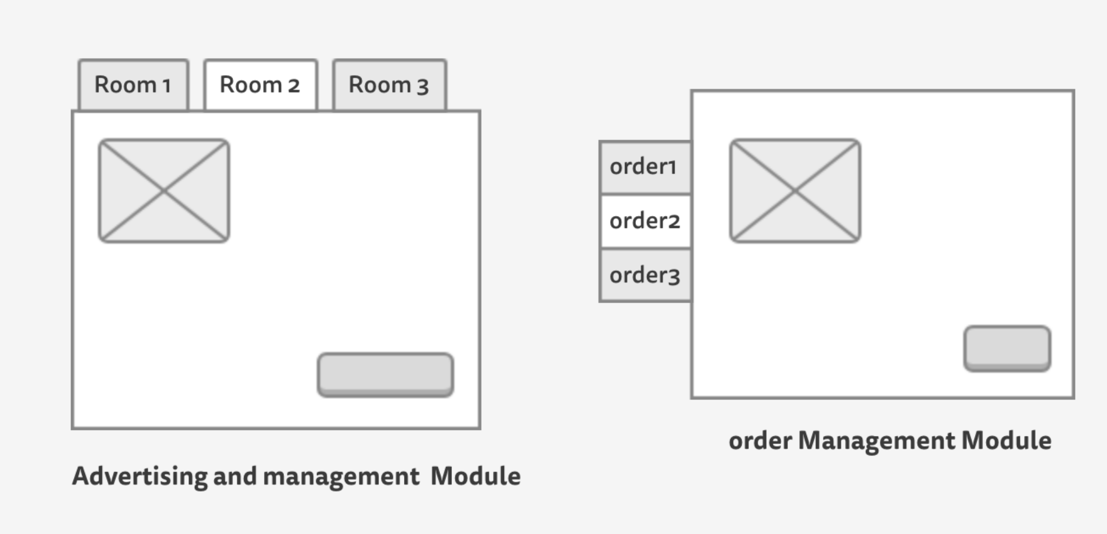
2.1 Blueprint

Both visitor-view and host-view flowchart will be provided in this section to illustrate our project's core solution.

Visitor-view:



Host-view:



2.2 Back-end frame

On back end of our application, we will use Python and Django. Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so we can focus on writing your app without needing to reinvent the wheel. It’s free and open source. These days, it seems that more and more startups choose the MVP (minimum viable product) model to develop their product initially. Using the proper approach, an MVP model should be scalable for further development. Django is the best solution to creating an MVP that can be further built on because it comes fully featured, right out of the box. This means that it already includes all of the necessary tools for creating any additional features for the product. For instance, an admin panel can be connected through one line of code using Django. As well, Django offers one of the best security levels of the currently available frameworks, ensuring that your project is kept safe. At the same time, Django also can connect to a large number of third-party applications that exist, further accelerating an already quick development process. For example, django-allauth is the best way to provide a "social login" (e.g., Twitter, Facebook, GitHub, etc.) option to users. And the Django Rest Framework is the best way to write REST APIs to connect Django with payment applications like Stripe and others.

Additionally, as a mature accommodation web portal, our database and host will be put on cloud, namely AWS in our plan. The following aspects are the reason why we choose cloud computing.
(a) Cost savings: Once we are on the cloud, easy access to our website's data will save time and money in project start-ups. The pay-as-you-go system also applies to the data storage space needed to service our stakeholders and clients, which means that we'll get exactly as much space as we need, and not be charged for any space that we don't. Taken together, these factors result in lower costs and higher yields.
(b)Security: One major hang up that many organizations have when it comes to adopting a cloud computing solution is the issue of security, especially when the trustworthy environment is what we have discussed as the main feature of our application. As an added security measure, with most cloud-based services, different security settings can be set based on the user.
(c)Insight: As we move ever further into the digital age, it's becoming clearer and clearer that the old adage 'Knowledge is power' has taken on the more modern and accurate form: 'Data is money.' Many cloud-based storage solutions offer integrated cloud analytics for a bird's-eye view of your data. With our information stored in the cloud, we can easily implement tracking mechanisms and build customized reports to analyze information organization-wide.
(d)Disaster recovery: In today's market, even a small amount of unproductive downtime can have a resoundingly negative effect. Downtime in the services leads to lost productivity, revenue, and brand reputation. Cloud-based services provide quick

data recovery for all kinds of emergency scenarios from natural disasters to power outages.

2.3 User Interface

It is important for any web designer to design functional sites that can easily generate interest and online traffic among the internet users. The user interface design of the site plays a vital role in bringing high volume web traffic to it. Therefore, web designers should give due importance to the user interface of the site which is being designed by them.

For success in any online business, a user-friendly website is a must as it will provide an enhanced user experience to the online visitors. Any site that is too complex and difficult will definitely push away online traffic. The use of effective and simple user interface design will be of immense help in achieving the objectives of a website. A good user interface not only increases the site usability but also leads to the smooth completion of any task at hand thereby making everything enjoyable and flexible as per the requirements of users.

Even if any website has amazing graphics with all the bells and whistles, it will definitely fail to generate considerable online traffic and provide enhanced user experience to the visitors in the absence of proper functionality. Websites like Facebook and LinkedIn are good examples of sites that generate high volume traffic mainly due to the enhanced user experience.

In terms of UI, our will project will concentrate on both methodical and well-organized structure and convenient interaction. Users will get only what they request on our website, nothing opaque else. Thus, proven technology like **Google Material Design** is a decent solution for our new flexible accommodation web portal. To be more specific, **MDC-101 Web (Material Components MDC Basics Web)** will be our first-order choice. Under such a frame and standard, the following advantages are not hard to reach:

(a) A sleek, simplified and engaging interface.

An interface designed with the material design requires minimal input process and delivers the most efficient and effective output. Since a lot of people already use Google apps on a daily basis, you can safely assume that they are used to the user experience. The magnetic feel to the tap is quite engaging in my opinion. People also tend to prefer a floating action button. It is unobtrusive and is always there for a user to take an action at any point.

(b)Well documented set of rules. Provides Consistency.

Material design provides a pre-defined set of styles and principles so that you don’t have to worry about explaining the concept details to your designer. The language was developed so that even a non-designer can easily outline, brainstorm and most importantly, comprehend the talking points without much of a learning curve. The concept assumes that all objects are paper cut-outs placed in the real world. It does a good job of providing consistency on all screens and viewport sizes.

(c)The virtual feel of the objects.
The 'materials' layered on the web page respond to frequent actions like mouse over and tap. Such responses tell the user indirectly that they are interacting with a virtual object and prompts them to take a natural course of action.
(d)Borrows the best features from skeuomorphic and flat design.
One of the biggest differences between flat design and material design is that the flat design entirely ignores the "real world" physics and lighting. Digital elements are made however the designer wishes to make them with no concern on what "reality" would do to the element if the object were physical. If you are not a fan of this approach, material design solves this issue by borrowing some items from the skeuomorphic approach.

2.4 Team Skill Set

IDE: Pycharm

Code management: BitBucket

SCRUM: Trello

3 Epics

In order to provide a clearer views , the epics will be split to different parts according to different modules including account management module, core business module, cloud host/database. Although the detailed contents and functions have already been discussed, the real user interfaces are not yet being decided, so the thumbnails view will be displayed to clarify our designs and ideas.

3.1 Account management module

Size: small

The user module allows users to register, log in, and log out.

Each user is assigned one or more roles. By default, there are three roles: anonymous (a user there are not logged in) and authenticated (a user who has signed up and been authorized), and administrator (a signed in user who will be assigned site administrator permissions)

Additionally, users are allowed various permissions with their account. For anonymous, they can search and browse housing listings. For authenticated, they can fine tune some personal configuration settings and private housing information through their individual ‘my account’ page. Each log-in user also has a personal order page to view historical orders and current order status. Order management, hotel management and user management are typically available only to administrators, who can operate directly on the database, such as deleting information about a property or users.

3.2 Core Modules (Accommodation advertising module, Accommodation Search module, Visitor request module)

Size: medium

- In advertising module, housing providers are allowed to advertise properties and maintain their advertisements. Providers can add new resources and provide the details by fulfill a form and upload photos as well as essential documents.
- In search module, users are able to do a general search considering only key words such as location. This search module will filter out inappropriate ones. A detailed and dedicated search page will be displayed after that.
- In request module, let the visitor see the results of the search module. The visitor can do more specific search and see the detail of selected accommodations. It is a more specific search bar that allows visitors filter from many different aspects. The results will be presented in a list, sorted by user searching needs (such as check-in/out time, location, price, etc.)

3.3 Fine-turned offering module

Size: medium

This module provides renters and providers negotiate with each other for more requests. For example, if a host see a request which is similar to those provided by him/her, he/she may compromise then and the host can modify the accommodation advertisement and inform the visitor.

3.4 Booking and Payment module

Size: medium
This module allows the users booking and paying on-line. Providers and renters can scan and check the order lists. Additionally, we design that renters can make payments through third-party payment platforms (such as Apple pay, PayPal, Credit or Debit Card) and choose the payment currency, when the user selects the appropriate service, enter the confirmation order page. In this way, users can intuitively see the service they choose and the amount they need to pay.

3.5 Accommodation review module

Size: small
This module gives renters a place to evaluate their accommodation experience. For example, they can rate and write about the experience. These comments are visible to anyone so as to be used to advise other users on renting.

3.6 Database design

Size: large
The core system of this application is Database Management System, which supports real-time transaction and daily analysis works.
The design of the database is related to the operational efficiency of the entire application system. A good database design should have **faster running efficiency**, **reduced data query and response time**, **improved data storage and transmission speed**, and it must **guarantee data integrity and security**. we will compare exists RDBMS application (e.g. mysql, postgresql, redis) and choose a proven solution based on four main aspects above.
According to system requirements and analysis, we will design the following data structure and table mainly in four aspects as follow:
1. Administrator information table
2 . User Information table
2. Housing information table
4 . order information table

3.7 Cloud-based architectures

Size: medium

Based on our requirements, we will not adopt traditional servers solutions.

We will use **Cloud-based servers (AWS)** to deploy DBMS storage and computing services.

3.8 Mobile Application cross-platform

Size: large

Mobile platform is one of the fastest-growing interactive platform. We decide to convert web application to a mobile app.

For some reasons, the screen size of the mobile phone device is different. To do the mobile web page, you need to consider compatibility on various sizes of Android/IOS devices. Therefore, we will prefer related page elements which **compatible with mobile platforms** like device pixel ratio when we develp HTML/CSS application.

4 plan to do

We will try to finish epics in small scale and medium scale, and set large scale epic as optional part.

To be more specific, here is a table list

What we will do	**What we may not do**
Account management module	Mobile Application cross-platform
Accommodation advertising module	
Accommodation Search module	
Visitor request module	
Fine-turned offering module	
Booking and Payment module	
Accommodation review module	
Database design	
Cloud-based architectures	

5 Sprint Schedule

According to the guide of Agile development, our team decide to set up daily discussion and weekly code review using Trello and git.

Schedule:

- Week 1/2 Project Select and Literature Review
- Week 3 Project Proposal
- Week 4/5 Web Back-end Application Structure and Web UI Design
- Week 6/7 Web Back-end Development
- Week 8/9 Web front-end Development
- Week 10/11 Beta Testing and Performance Testing
- Week 12/13 Final Project Demo

- Weekly Meeting Times
- Friday, Postgraduate Study Space, 14:00-18:00
- Sunday, Postgraduate Study Space, 14:00-18:00