**Introduction**

**.Background**

Nowadays, travelling has become a popular lifestyle, as it makes people be relaxed and close to nature. It also a good way to experience and learn new thing from different countries and cultures.

While the demands of accommodation increasing, the online service for accommodation booking by website and application has become available and easier, as the software technologies provider convenient and fast development methods and frames.

In this way, the travelling accommodation marketing is the great demand, but, fulfilling it, is potential and feasible by quickly and efficiently website development. The website will play an important role to generate profit for accommodation agencies, but also offers to reserve requirement for customers with useful and valuable information.

**.Existing system and drawback**

The travelling-accommodation industry generating great potential marketing demands advance the number of relevant websites.

However, there are a few common drawbacks among these websites, which could affect the customers’ user experience and cause low efficiency when search some destination or target accommodation.

For example, Trivago, as a popular small-size website, however, the complicated search function at homepage could cause the bad user experience for the new users, as different operations may make people confused and lose the patient.

In addition, Airbnb, as one of the most popular website, the oversample search page without filter options could also make user unsatisfied with it, as users may spend time on waiting for the jump to a further page to utilise filter functions.

In conclusion, current websites are overloaded or oversample with information, therefore, more or less opaque and useless.

**.Problem we have to solve**

If we want to design a high-quality website and offer better service for users, it would be necessary to implement these issues:

• The layout of the portal should be easy

• The number of information on each page should not be too large

• Each page should only offer the information that the visitor requests

**Epics:**

**.Aim:**

The main aim of the project is to create a Web application that will allow providers to present their holiday units, and searchers to review the offers and reserve or rent a property. The application should be limited to properties located in NSW.

This website will satisfy both user(customers) and host demands. The customers could use it to search and reserve the accommodation or holiday units based on the location they typed. Also, the customers could extend the function to be hosts and post properties for others to review and reserve.

These are the main epics for our project:

**.The three Epics:**

The main epics are defined in the service. Our website(miniairbnb) will be designed by 3 core service, which are database, users and hosts, respectively. For each epic, it will be achieved by separating models.

**1.Database service:**

**.Database model(implement by PostgreSQL):**

PostgreSQL[1], often simply Postgres, is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance. It can handle workloads ranging from small single-machine applications to large Internet-facing applications (or for data warehousing) with many concurrent users; on macOS Server, PostgreSQL is the default database; and it is also available for Microsoft Windows and Linux (supplied in most distributions). Such an open source, high efficiency and the quality database will support our data, and it could be easy to assign to the Linux server for maintaining the website working.

Our database model will be designed for terminating users and hosts information, which include the personal profile, properties information etc. In addition, the established database will mainly focus on manage user and hosts and related data to support the users' operation, like search, edit, delete and add. The detail about the "miniairbnb" database will be indicated by others models, which we will show below.

**2.Service for customer**

This is the main model and also the most vital model for this website. This model will fulfil a list of services which based on the logical behaviour of users. For example, the searching, filter, check, booking, generating order, accepting or not could be the logically whole service processing when the user utilises our website. In this way, our customer model will implement specific functions to satisfy each step of this processing.

**a. Login and register model**

In this model, it will only be triggered or detected when users want to utilize the booking function. It means that if users just overview the resources of properties which are available, they will be allowed without any limitation. However, if users want to book the property they prefer to choose, the website will check users status to distinguish whether they are login or not. If they are not log in, the website will jump to login in page, then come back. If they are the first time to try our website book service, the website will guide them to register page first.

Anyway, our database will always provide the ability to identify the user and manage their behaviours.

For the register, we need Username, Email, First Name Last Name, DoB, Password to register as a guest by default. If users finished this processing, the email will be the key to recognise users.

**b. Search module - finds properties and filters out inappropriate ones**

For searching, our design is giving users a search form at home page, which could directly type the destination address without other options. Then, the website will jump to a new page and it will show the searching results list, which includes all properties located in the specific area typed by the user. If users want to filter out inappropriate objectives, they could use filter functions to achieve it at this page. Actually, the filter options will still return a results list, but it could be closer to users’ demands.

The search function in home page should only contain some basic search content: location, Check in/out Date, Num of Guest. Further, the filter could narrow by price, rating, distance, etc. Anyway, the result will return by a sort algorithm which based on some digital data.

**c. Visitor request module – for users to:**

If the user is satisfied with the property they checked, they could do the booking operation immediately.

This operation will modify some attributes of the property, for example, the available time would be changed as the user occupy some days on it. It will also post a message to host to reminder host accepting or rejecting the booking request. Further, it generates an order list for the user to indicate the order status, which may be pending, accepted or rejected.

All the points mentioned before will be triggered by users, but implement and maintain in the database.

It means that these modified operations all reverent a specific SQL operations.

**d. comment model:**

The feedback from the user for a property could be achieved by a comment related to the specific accommodation the user has experienced. It also a part of the server for users.

**3.Service for host**

**a. property manage model:**

The host module actually is the extent of normal users’ functions, which means a host could search and reserve the holiday units but also post their own properties. We will provide a button on the navigation bar to guide the user upgrading to host if they want to publish properties. When a user successfully become a host, the additional functions, like, release the properties, manage properties, accept or deny the order, will open for him or her.

About the booking request from users, the request will be stored in user model and it will trigger the message reminder in host model, which means that host could notice the booking message at their account and do accepting or denying operations later, as it is a function or service offered to host.

We will design a specific part of the database to implement this processing between use and host.

**b. Publish(advertising) model:**

In this model, when the host provides his property, we plan to provide the advertising properties(items) on the website page. To be clear, if the user types any useful search keyword, we will return a list of properties based on his demands, these returned results could be called advertising. Once the user clicks the specific item, we will demonstrate the property’s information edited and offered by the host. In this way, we implement the advertising model by users selecting and database feedback.

**Project methodology:**

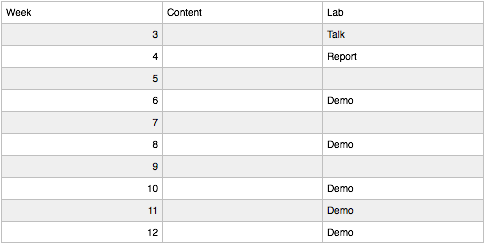
Development environment

scrum Master:

production owner: scrum team

developer: All scrum team

Schedule(Table):



[1] https://en.wikipedia.org/wiki/PostgreSQL