# SIT 305 Assignment 2 Mobile app

## Overview

In this assignment, I am going to develop an island travel booking app, which allow users to book a travel trip and join the beautiful journey.

## Target audience

Users who are interested in traveling to island and beach, who enjoy spending times in a fantastic island.

## Reasoning of how your project demonstrates creativity

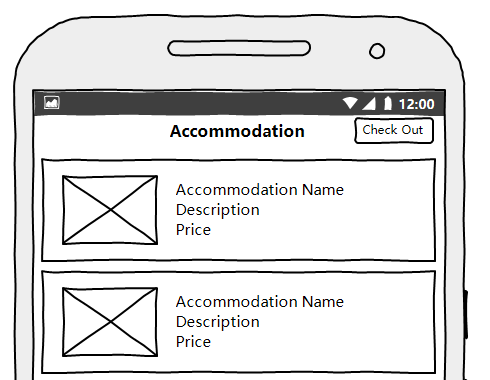
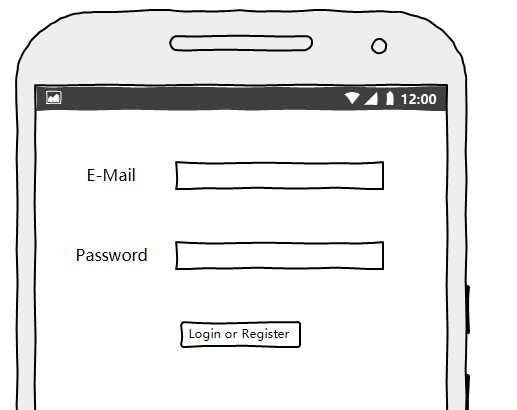
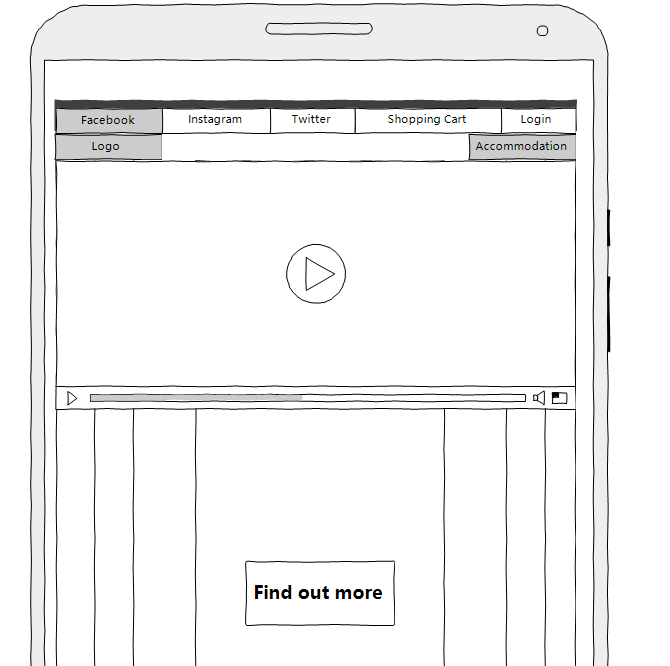
I am also a big fan of beach and island life; I have traveled to Hamilton island myself. So, in my home page design, I have a video to let users/viewer to watch Hamilton beautiful views. I decide to develop a booking system and being able to send booking confirmation to user’s email address.

# Features

1. Sign up using e-mail and password
2. Sign in using e-mail and password
3. Select interested accommodation and add to shopping cart
4. If any items in the shopping cart is unpaid, it will be stored into Shared Preference and will be retrieved back when the user is navigated to Shopping Cart Activity
5. Once the order is paid, the user will receive an e-mail containing detailed information of the order, including accommodation name, unit price, check-in date, check-out date, and total amount of the order.
6. Locally persisted order data will be cleared once paid.

# Design

## Wireframe





## UI Color choice

UI Style: white and blue as main colour choice represent sky and ocean and take Hamilton Island official website as inspiration.

# Data

The app data is stored both locally and remotely. Accommodation and orders data is stored locally using JSON string. While user e-mail and password pairs are stored in Google Firebase because the app takes use of Firebase Authentication services to achieve Login and Register functions.

JSON is adopted to make it easier to write and retrieve locally persisted data. That is, all instances of class Item (accommodation) and class Order are represented by a JSON string. Otherwise it could be more difficult and time-consuming to compose and parse the serialized data. By using JSON, we need just to let classes implement the Serializable interface and override the ToString function to make persist data easier. For read data back, we just need to make a constructor which takes a String as argument, which is actually the JSON string.

## API/Class Structure

## Features and APIs used in the app

1. Google Firebase Authentication

Achieved user sign up and sign in using Google Firebase Authentication. By using this API, there is no need to write code to manage the user e-mail and password pairs using Database or Shared Preference.

By using Google Firebase APIs, it will be easier to accommodate remote data persistence using Google Firebase Storage in future updates of the app.

1. Shared Preference

It is a light-weighted local data persistence mechanism. It is easy to manage smaller-sized data locally using this mechanism. In this app, the data of accommodation and orders is not that large. It becomes most suitable to choose this mechanism to achieve local data persistence at this stage and scale. However, its efficiency could be a disaster if there are thousands accommodations information. In this case, a better solution must be used, for example, SQLite or Google Firebase Storage.

1. YouTube Video API (Third-party)

It could be important to achieve better user experience to have some video or audio information in the app. YouTube is a common used video provider across the world. It is important for the app to be able to utilize video or audio materials on YouTube.

1. Object Serialization

To pass objects between multiple Activities in Android using intent, it is necessary to make the object to be Serializable as Intent only accepts Serializable extras. To achieve this, we need to implement the Serializable interface and overwrite the ToString method.

1. JSON

Accommodation and order classes have many fields, it could be much more difficult to compose and store these fields by myself than using JSON. To make it easier to use JSON, I make all these fields to be of generic data types, because most generic data types are Serializable. For example, String, Integer, Boolean.

1. Recycler View

Reduce resources required to run the app by RECYCLE views. This is significantly useful when there are a huge amount of items to be displayed on the screen. For example, if there are 1 million items to be displayed, the app will need 1 million times of resources. By using recycler view, only several times of resources are needed because once a view is slides out of the screen, the resource is recycled to be used by another view.

1. Sun. Mail API

It could be very important to let users know what happened for any apps. In this app, if a user paid for something, he or she obviously wants to know what he or she has purchased. Consequently, I accommodated e-mail service in this app. It is achieve by using the Sun.Mail API and my Gmail account.

One thing worth noting is that such a way to use Gmail is deemed as insecure by Google. So it is necessary to configure the Gmail account to accept lower security level applications. Otherwise, the app will fail to access Gmail services and hence the e-mail service of the app would not work.

## Significant method signatures

1. static void WriteSharedPref(Context context, String key, String value)

static String ReadSharedPref (Context context, String key)

This pair of functions are to write and read the String value of the key. As mentioned above, data persistence of accommodation and order instances are achieved using shared preference. Consequently, read and write of the shared preference are frequently used in the app. If we arrange the two functions as static functions, the code will looks more clean and readable.

1. static void sendMail(String recipient, String subject, String content)

MailT

As we all know, sending an e-mail is obviously a network activity and deemed as time-consuming. It is a common sense to put time-consuming or other resource intensive activities in a child thread rather than the UI thread. Consequently, it could be reasonable to send e-mail in a child thread of this app. MailT is actually a subclass of Thread. I put the code snippet which actually sends the e-mail in its run() function. By this way, the e-mail sending would not block the UI thread.

## Significant classes

1. MailT

As mentioned above, MailT is a subclass of Thread and is used to avoid blocking issues when sending e-mails.

1. Order

The class which represents selected but not paid accommodations.

1. Item

The class which represents accommodations.

1. IslandPagerAdapter

The adapter class to implement the image pager in MainAcitivy.