**GitHub Username**: ruiguo11

# POI finder

## Description

Looking for a good restaurant, an interesting place, or a nice clean hotel, POI finder will help you sort it out, and also tell you how to get there. And you also can save an interesting place to your own favourites list, so you can get the information without internet connection. And also sharing interesting place with your friends.

## Intended User

The intended user are the travellers, and people want to explore new places.

## Features

- The app will all allow the user to select several places which include attractions, restaurants, hotels
- Allow the user to save their favourite places, and can read the details of the place offline, or share with other via email.
- Get directions to how to go to a place of interesting.
- This app will show point of interest nearby or search in a city.

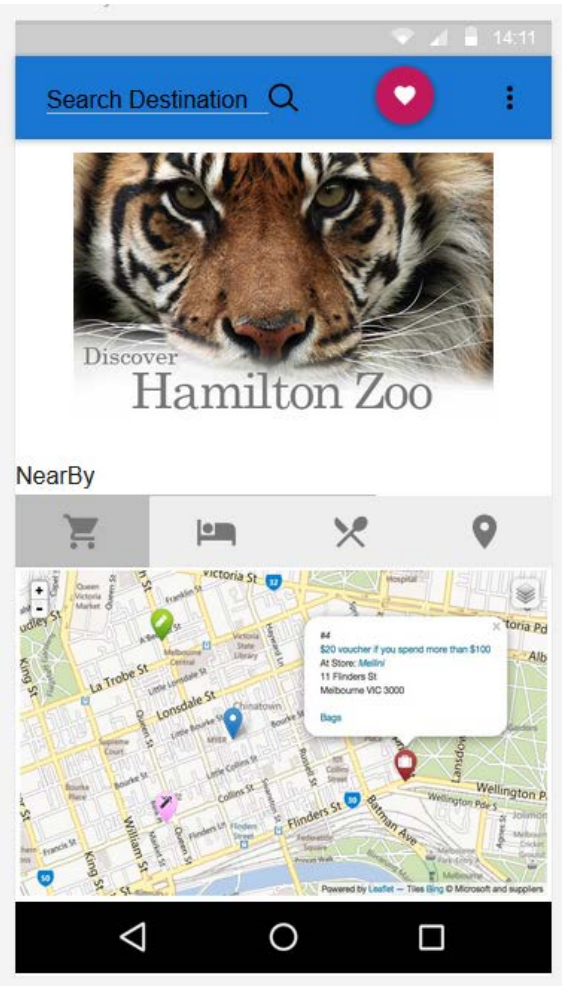
List the main features:

- Saves information
- Share information
- Route planning
- Searching information

# User Interface Mocks

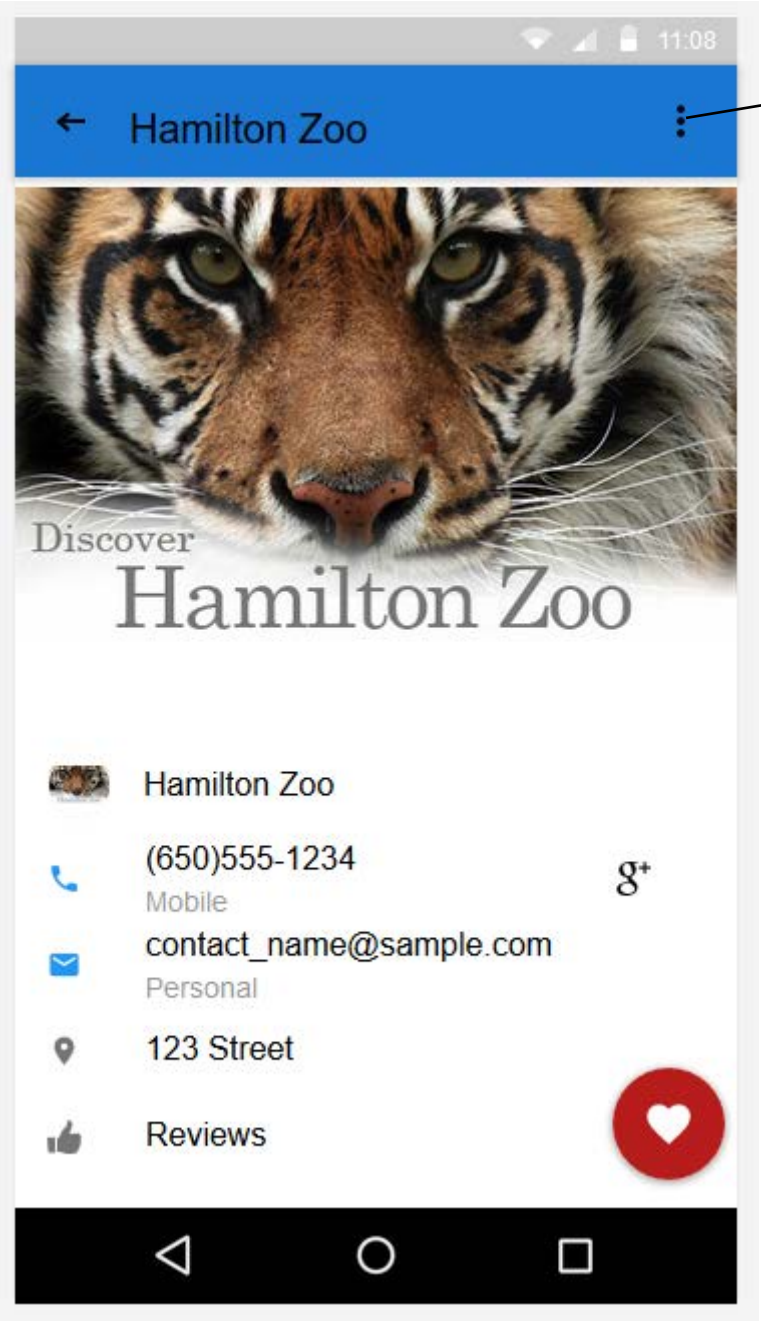
## Screen 1 (Main Activity)

This is the main activity screen. When the app launch, the map will display the nearby shopping centres, attractions, restaurants, hotels. The user can click the mark on the map to get details of the selected point of interested. (see screen 2)



## Screen 2 (Detail Activity)

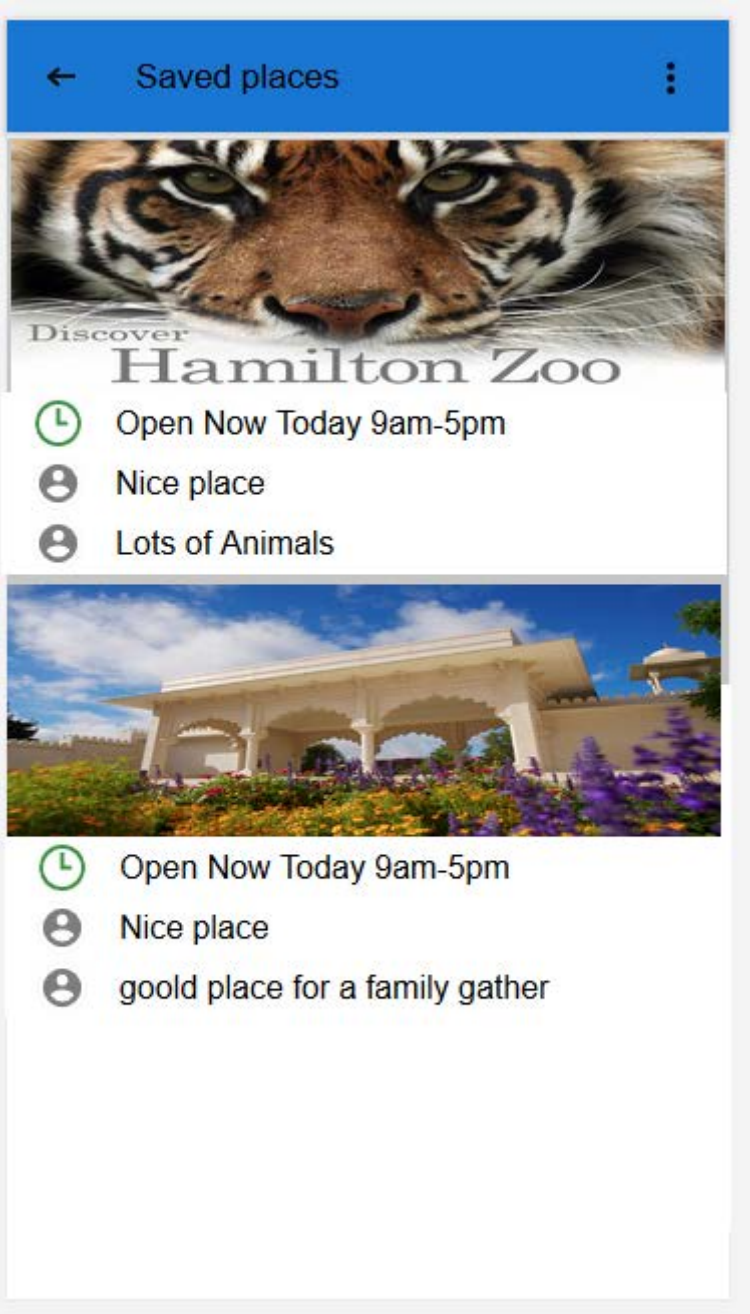
This is the details screen. When the user clicks the address or the icon by the address, it will display the location and the direction to go there. ()
The FAB will allow the user to add/remove the place to his favourites list, the user can view offline.
This screen will use CollapsingToolBar Layout, and NestedScollView.
By press the back button, it will back to main screen (Screen 1), and if the user swipes the screen it will display next interesting place.



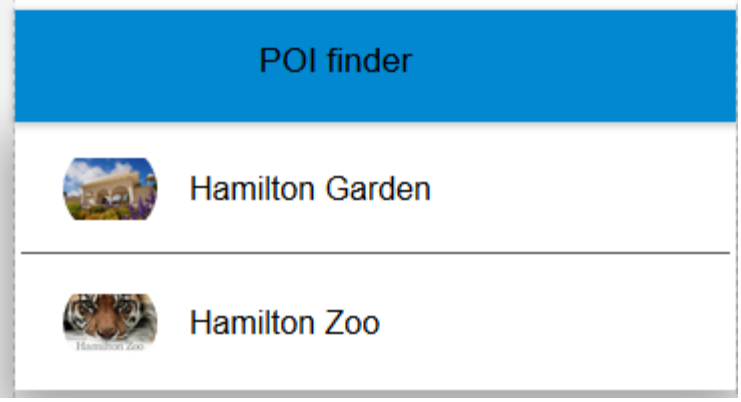

When the user clicks her, it will show up the share preference, so the user can share this place

## Screen 3(SavedPlaces Activity)

This screen will show a list of saved places. When the user clicks a place, it will show the place information in the Detail Activity (Screen 2).



### Screen 4 (Widget)




When the user clicks the top title bar, it will launch the app. When the user clicks the item in the list, it will launch the details activity of that place.

## Key Considerations

#### How will your app handle data persistence?

I will build a Content Provider to handle data, and save data to SQLite database so the use can get information offline.

#### Describe any corner cases in the UX.

Haven't got any corner cases in mind so far

#### Describe any libraries you'll be using and share your reasoning for including them.

Use Picasso to handle the loading and caching of images.

#### Describe how you will implement Google Play Services.

Using Google location & Content API
Google Map, Direction, Place API

## Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

## Task 1: Project Setup

- Configure support libraries
- Set up project in Google developer console and enable selected google APIs

## Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for DetailsActivity
- Build UI for SavedPlaceActivity
- Add material design to each activities

## Task 3: Handling data

- Implement Google Play Services
- Build Content Provider
- Implement AsyncTask to get data from required Google API
- Create SQLite database to save information
- Implement CursorLoader

## Task 4: Add Notifications and Widgets

Describe the next task. List the subtasks. For example:

- Create notifications
- Create Collection widget

## Task 5: Testing and error handle

- Build unit testing
- Handle errors

**Appendix:**

http://www.pedfieldcountryhouse.co.nz/uploads/100922/images/213227/HamiltonZoo.jpg

https://en.wikipedia.org/wiki/Hamilton_Gardens