Author & Email:

Ayling, Robin ([r.ayling267@canterbury.ac.uk](mailto:r.ayling267@canterbury.ac.uk))

Assignment 1 Report

**ASSIGNMENT NAME:** Mobile Computing Assignment 1

**COURSE:** MC OMD3MBC

**YEAR:** 3

**TUTOR:** Joseph Williams

**DUE DATE:** 7/5/2018

**DATE HANDED IN:** 4/5/2018

**NAME:** Robin Ayling

Contents

[Introduction 2](#_Toc513210692)

[Features 2](#_Toc513210693)

[Limitations and Reflection 2](#_Toc513210694)

[User Guide 3](#_Toc513210695)

[Start-up 3](#_Toc513210696)

[Advanced search 4](#_Toc513210697)

[Results – list view 4](#_Toc513210698)

[Results – map view 5](#_Toc513210699)

[Restaurant details 5](#_Toc513210700)

[Favourites 6](#_Toc513210701)

[Marking 6](#_Toc513210702)

[Fragments 6](#_Toc513210703)

[SQLite Database 6](#_Toc513210704)

[AsyncTask 6](#_Toc513210705)

[Third-party Libraries and APIs 6](#_Toc513210706)

[References 6](#_Toc513210707)

# Introduction

For this assignment I was tasked to create an android app from one of three ideas. I chose the Happy Eatery app. This app is to provide functionality for the user to search for restaurants within a certain distance of an address or the user’s current location. These results needed to be filtered by rating and ordered by distance. These results should be displayed in a list and optionally as markers on a map. The user needs to be able to save/remove favourite restaurants to persistent storage.

# Features

* Simple landing page containing location input and a button. Keeping this simple allows the user to jump straight into results without having to input multiple settings.
* List view that displays all search results ordered by distance, displaying restaurant name, type, rating, and distance from user’s location/the input address.
* Map view which displays search results as markers on a map. There is a seek-bar to change which results are visible to make the map more user-readable.
* Advanced search activity which lets the user refine their search. There options to change the distance in miles between 1 and 100, input a new address or current location, or select a rating score as well as comparator.
* Favourites list that shows all restaurants saved to favourites and their distance from the given location. If no location has been given then the default is a longitude of 53 and latitude of -3 and the distance is calculated from there.
* Clicking an item in either list or clicking the description on a map marker will take the user to the restaurant page. This page displays the details of the restaurant, distance from the location (with the same details as above), and a map at the top showing the restaurant’s location.

# Limitations and Reflection

This app is mostly limited by the FHRS API[1]. One of the issues is that results are slow, sometimes even timing out. In this case a message is displayed telling the user that the app failed to load the results. This can be rectified by clicking the ‘more results’ button. Another limitation with the API is that the locations aren’t always accurate, sometimes showing results up to a mile away from where they should be.

This app relies on location services to get the last known location and geocoding to parse the postcode into latitude and longitude. If the user disables location services and geocoding is not present then the search function will be rendered unusable.

Occasionally the map view will not display the first batch of results due to loading times. This can be fixed by clicking the ‘more results’ button.

On reflection I found android studio somewhat difficult to use. Playing around with the app theme caused my manifest file to mess up requiring me to start a new project 3 times. Had this not happened I may have had more time to add features like filtering the search by type or changing the amount of results displayed.

Originally, I was storing the distance of the restaurant in the database until I realised the distance would change depending on the input location. To fix this I found a method to calculate distance from latitude and longitude[2].

I designed all of the activities to be displayed in portrait mode so I added “android:screenOrientation="portrait"” to each activity in the manifest file. I also added “android:launchMode="singleTask"” to the search activity. This made it so that when the user makes multiple searches they don’t have to press the back button for every time they made a search.

# C:\Users\Robin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20180504-143524.pngUser Guide

## Start-up

Download and open the HappyEatery app. Here you will be greeted by the following screen. Please select to either use your current location or enter a postcode below. If both are selected/entered then current location will take priority. You can also access your favourites from here by clicking the favourites button at the bottom or do a more advanced search by clicking the advanced button.

## C:\Users\Robin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20180504-143547.pngAdvanced search

Tapping advanced search will take you to the following screen. Here you can select options to change the range of your search up to 100 miles, change your location, or select a rating and a comparator (the default comparator is Greater than or equal).

## C:\Users\Robin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20180504-143531.pngResults – list view

After tapping the search button, you will be shown the first twenty results of your search in a list view. Each item in the list will show the restaurant name, type, rating, and distance from the selected location. Tapping an item will take you to the restaurant details pages where you can see further details. The more results button will load a further 20 results into the list. There are also buttons to take you to your favourites or the advanced search at the bottom. Swiping the screen left will show the map view. Occasionally the app will fail to load results. If this happens simply press the more results button and it will attempt to load the results again.

## C:\Users\Robin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20180504-143536.pngResults – map view

The map view will display the results as markers on a map, twenty at a time. Using the slider at the bottom will change which twenty results are shown, the slider being all the way to the left showing only the first twenty results and the right showing the last twenty. As with the previous view this one will also occasionally not load results. The more results button fixes this. Tapping a marker will display an info box with details of the restaurant in it. Tapping this info box will take you to the restaurant details page. There are also buttons to take you to your favourites or the advanced search at the bottom.

## C:\Users\Robin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20180504-143605.pngRestaurant details

This page displays the details of the selected restaurant, including the name, type, rating, distance away, and address. It also displays a map with a marker for the location of the restaurant and a button to add the restaurant to your favourites list. This button will display ‘Fave’ when the restaurant is not in your favourites and ‘Unfave’ when it is. Tapping the unfave button will remove the restaurant from your database.

## C:\Users\Robin\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20180504-143556.pngFavourites

This page shows a list of all the restaurants you have saved to your favourites. Tapping an item will take you to the details page where you can view the details and remove it from your favourite if you wish. To navigate away from this page just press the back button.

# Marking

## Fragments

I used fragments for the results activity so that the user and swipe between the list and map views.

## SQLite Database

I used an SQLite database to persistently store data about the user’s favourite restaurants.

## AsyncTask

AsyncTask was used to load entries from the database and populate the UI in the favourites activity. This is in case the user has stored many entries in the database so the UI wont crash when it tries to load them all.

## Third-party Libraries and APIs

I used the FHRS API to get the data for the restaurants and the Google Play Services[3] to load the maps. I also used the Volley library to get the data from the FHRS API.

# References

1. FHRS API <http://api.ratings.food.gov.uk/help>
2. Distance calculation method <https://stackoverflow.com/questions/3694380/calculating-distance-between-two-points-using-latitude-longitude-what-am-i-doi?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>
3. Google Play Services <https://developers.google.com/android/guides/overview>