# Annotated Bibliography

1. Google Places API. Various: <https://developers.google.com/places/documentation/>

The places search page was used to understand how nearby search request URLs needed to be formed, using the appropriate required and optional parameters. This page also improved understanding of the output in a search response, how it could be used in the application. The place details again offered the above but for detailed requests. Furthermore the place photo’s page was again evaluated and a lot was learned, how to get a photo reference and use it to make a Place photo request.

1. Google Maps Android API v2. Various: <https://developers.google.com/maps/documentation/android/>

The Google Maps for Android API was used extensively throughout the project. The getting started page was really helpful in setting up the environment needed and obtaining an API key. The map objects pages as well as the sample code the pages linked to were really useful when developing the mapping elements of the application.

1. Fraser Coast Chronicle Newspaper and East Riding of Yorkshire Council. Some recent evidence to support the theory that tourist information centre visitor numbers are in decline. <http://www.frasercoastchronicle.com.au/news/visitors-at-centres-in-decline/1615621/> . Further evidence that tourist information centres visitor numbers have been declining for some time, a council report. <http://www.eastriding.gov.uk/public_reports/TheCabinet/16March2010/Tourist%20Information%20Centres%20-%20Hornsea%20and%20Withernsea.pdf>

Both these articles motivated me to create and develop an app that would offer the tourist information once provided by these centres in a modern format that could be used in the ‘technology age’.

1. CNET. An article which discusses how the Google Play Store is rapidly catching up with number of apps available on the apple app Store and how it could overtake it in the near future. <http://news.cnet.com/8301-1035_3-57521252-94/can-apples-app-store-maintain-its-lead-over-google-play/>

This article helped make the decision to develop for the emerging mobile platform that is Android rather than the iOS operating system

1. Google Play Store. Manchester Tourist Guide Android Application. <https://play.google.com/store/apps/details?id=com.mymobilemanchester&feature=search_result>

This application was evaluated when researching related works to the application being created.

1. Google Play Store. Pocket Britain Android Application. <https://play.google.com/store/apps/details?id=com.phonegap.pocketbritain&feature=search_result>

Another application that was evaluated when researching related works to the application being created.

1. Google Play Store. Yell Android Application. <https://play.google.com/store/apps/details?id=com.yell.launcher2&feature=top-free>

The last application that was evaluated when researching related works to the application being created.

1. GSM Arena. Mobile Device operating system market share figures for the 3rd quarter of 2012, showing iOS’ and particularly Androids dominance. <http://www.gsmarena.com/android_and_samsung_build_on_their_market_shares_in_q3_2012-news-5082.php>

This article helped make the decision to develop for the Android operating system.

1. Apple Developers. Shows that you can only develop for iOS on an Intel based Mac. [http://developer.apple.com/library/ios/#documentation/Xcode/Conceptual/ios\_development\_workflow/45-iOS\_Development\_FAQ/faq.html#//apple\_ref/doc/uid/TP40007959-CH12-SW1](http://developer.apple.com/library/ios/#documentation/Xcode/Conceptual/ios_development_workflow/45-iOS_Development_FAQ/faq.html)

This resource again helped make the decision to develop for the Android operating system due to the restriction of having to use a Mac computer.

1. Wikipedia. Reveals methods of using Mac OS X on a non-Apple PC. <http://en.wikipedia.org/wiki/OSx86>

This web page helped describe how Mac OS X can be used on a non-Apple PC. This provided more evidence about why this should be something that should be avoided if possible and further supported the decision to develop for Android.

1. BBC. An article about the negative feedback and complaints made about Apple’s new Map app. <http://www.bbc.co.uk/news/technology-19659736>

This article is from the news and confirmed the notion that Apple had switched to using their own maps which were perceived as being inferior to Google Maps. This gave another reason to avoid developing for iOS as the future direction of the company with regards to mapping could not be confirmed. It was also unclear whether API for Apple’s maps would be made available, how good it would be and whether it would meet the needs of the project.

1. PC Mag. News article stating the possibility of a new Google Map app being launched for iOS 6. <http://www.pcmag.com/article2/0,2817,2412213,00.asp>

Background information on how Google Maps may still be supported on Apple devices by the use of an external app. Although the risk of developing for the iOS operating system with the hope that Google Maps would still be supported was too great.

1. The Android plugin for Net Beans. <http://kenai.com/projects/nbandroid>

The web page for an IDE that was researched when looking at development tools to use for the project.

1. IntelliJ Idea IDE with support for Android. <http://www.jetbrains.com/idea/features/android.html>

The web page for an IDE that was researched when looking at development tools to use for the project.

1. Android Developers. The Eclipse IDE with ADT was used throughout the development project. The download page that enables you to get the Eclipse IDE, Android SDK and ADT plugin: <http://developer.android.com/sdk/index.html>

This allows you to download all tools needed for Android Development in the Eclipse IDE. For this project the whole bundle wasn’t needed as the latest version of Eclipse had already been installed.

1. Android Developers. Installing the Eclipse ADT plugin. <http://developer.android.com/sdk/installing/installing-adt.html>

This tutorial shows how you can install the ADT plugin into Eclipse if you already have Eclipse installed.

1. Android Developers. ‘Building Your First App’ Android tutorial. <http://developer.android.com/training/basics/firstapp/index.html>

This resource from the Android developer’s website is a tutorial which helps create a ‘Hello World’ application and was the first app created on the project and helped show how Android projects and apps can be set up and structured.

1. Aberystwyth University. Android worksheets and presentations from workshops held by Chris Loftus and Andy Starr. <http://users.aber.ac.uk/cwl/workshop/>

These worksheets built on top of the very basic features learnt in the ‘building your first app’ tutorial from Google so that small applications could be built on the Android platform. More information on how different components work and were used together was understood.

1. GitHub. The homepage of the online project hosting service where a free repository can be set up. <https://github.com/>

One of GitHub’s free repositories was used to quickly and easily back up project work externally in the cloud.

1. Object Aid. A tool used to generate class diagrams automatically from source code in Eclipse. <http://www.objectaid.com/class-diagram>

This tool was used to create the class diagram for the application from the source code written for the project and can be seen in the appendix.

1. Testdroid. The tool used to automatically test the user interface of the TourInf application. <http://testdroid.com/product/testdroid-recorder#0>

This tool allowed automatic user interface tests to be created by recordings of tests being made and then test code being automatically generated from the recording. This allowed the user interface to be tested more rigorously and independent of user opinion and error when testing.

1. Android developers. This web page details the fundamental Android application components. <http://developer.android.com/guide/components/fundamentals.html>

The page was used in initial understanding of Android development and its components. It was then further used to provide accurate definitions and explanations of components within this report.

1. Maki Icons. An open source project by MapBox. Maki is a clean point of interest icon set made for web cartography. <http://mapbox.com/maki/>

This icon set provided the images to be used to produce the main menu of the application. This set was used as the icons provided have a professional appearance, were available for free and no licensing. The images were also provided in vector format so they were easy to resize.

1. Ed Burnette, Hello, Android: Introducing Google’s Mobile Development Platform, Third Edition, for Android 2. Pragmatic bookshelf 2011.

This book was lent from a friend to research programming for Mobile Devices. Chapter 2 Key Concepts was really useful in understanding the structure and nature of Android programs. Parts of chapter 3 designing for the user interface, were useful in helping to create the user interface and helping to create it for different orientations and screen sizes (section 13.5 also helped with this). Chapter 7 was read to find out about how intents can be used to launch web pages in a browser and how web services can be used in Android. Chapter 8 locating and sensing was also researched to find out about using devices’ inbuilt GPS. In section 8.3, Bird’s Eye View was also read at the start of the project for mapping. Although the examples that were provided were using the old map API, the information provided still helped with learning the essentials and theory behind their use. Section 13.2 building for multiple versions also helped understand how features and methods should be checked to make sure they will be supported by multiple versions of the Android operating system to support as many devices as possible.

1. Android Developers. The strategy and theory of using Android’s Network Location Provider together with GPS to determine a user’s location. <http://developer.android.com/guide/topics/location/strategies.html>

The resource helped gain an insight into how the GPS part of the application could be built and what methods may be needed.

1. Android Developers. A training article that includes a button to download a sample location app. LocationAware.zip. <http://developer.android.com/training/basics/location/currentlocation.html>

The above sample project was downloaded to see what it produced and how it had been implemented, giving ideas to how these elements may be created in the final application.

1. Android Developers. An overview of what Grid View is on the Android Platform and how you can use it. <http://developer.android.com/guide/topics/ui/layout/gridview.html>

This resource was really helpful in setting up the basic initial grid view in the application, which formed the basis for the main menu.

1. Google Maps Android v1 API library. The original Android maps API. <https://developers.google.com/maps/documentation/android/v1/>

The news that this API had become deprecated and was being replaced by a new version was announced and seen on the above website.

1. Google Maps Android v2 API. How to install the maps API SDK and obtain a Maps API key. <https://developers.google.com/maps/documentation/android/start#installing_the_google_maps_android_v2_api>

This guide helped greatly in installing the required SDK and obtaining the Maps API key needed.

1. Google API Console. Where settings can be allocated for Google APIs.

<https://code.google.com/apis/console/>

The Google API console was used to obtain an API key and request access to using the Google Maps Android v2 and Google Places APIs.

1. Google Map API Documentation. How to get sample code of the new maps API being used.

<https://developers.google.com/maps/documentation/android/intro#sample_code>

These instructions were used to download the sample app for the new Google Maps API.

1. Android Hive. Android Google Places and Maps tutorial.

<http://www.androidhive.info/2012/08/android-working-with-google-places-and-maps-tutorial/>

This tutorial was followed and the majority of this 3rd party code was used to develop the Google Places aspect of the application which was then modified in areas for the finished application.

1. Google Places API documentation. The list of supported place types.

<https://developers.google.com/places/documentation/supported_types>

This list of supported types was used to know which categories could be used in the application. These type strings were then used in the application to make requests to the Google Places API, using the type associated with the category selected.

1. Android Developers. The reference page for the Geocoder API class.

<http://developer.android.com/reference/android/location/Geocoder.html>

Methods from this class were used to convert town/city names and postcodes into lat/long values inside the app so those values could be used to make a Google Places request.

1. Stack Overflow. This answer to question regarding whether directions could be got from using the standard Google Maps for Android v2 API

<http://stackoverflow.com/questions/14495030/google-map-api-v2-get-driving-directions>

The last answer to the question asked provides a solution to getting directions to be displayed on a Google map. However this solution was very complicated and would not have produced a professional output and was therefore ignored.

1. Google Developers. The terms of service for Google Maps and Earth.

<https://developers.google.com/maps/terms>

After researching methods to get directions displayed on Android Maps it was found that there may be the possibility that Google’s terms of service would be broken by altering their maps to get the functionality programmatically. This web page was then found and read to confirm that it would indeed break their terms of service.

1. Google Developers. The terms list of intents that can be used on Android devices.

<http://developer.android.com/guide/appendix/g-app-intents.html>

This resource was read to investigate what intents can be used and sent by an application. It was then realised that the VIEW intent action for the browser could be used to show directions in a browser of a devices map app.

1. Stack Overflow. Information found on how to detect if a device can make a phone call.

<http://stackoverflow.com/questions/7429722/how-to-detect-if-device-is-capable-of-calling-and-messaging>

This post showed how to detect if a device is capable of making a phone call. By using this method, the button to make a call in the app would only be displayed if the device it was being run on was capable of making a call.

1. Android Developers. A training guide on how bitmaps should be processed on a separate thread to the user interface.

<http://developer.android.com/training/displaying-bitmaps/process-bitmap.html>

This training example was seen as an example of how to process bitmaps in a background thread using AsyncTask. It reminded the developer that BitmapFactory.decode methods should not run on the UI thread due to unpredictability in the time taken for an image to be read and loaded over a network.

1. Dibbus. A tutorial on how to change the colour of Android buttons by using 9 patch images.

<http://www.dibbus.com/2011/03/9patch-images-in-android/>

The tutorial was read and parts used. The selector XML file was used in the project, as were the blue and white matte images that were offered as no license was required.