

CMP3034M - Mobile Computing - Assessment 1

I. Background

The 'Track my Travel' is an application that will allow users to assist them with there travels. The idea behind this application is to assist and use it as a travel log, making it useful for travellers and bloggers who would like to track all the places they have been without taking too much time. As "there has been a large increase in the number of people out on tours...Tourism is the strongest and largest industry in the global economy world, generating an estimated 11% of the global gross domestic product (GDP)." (Jinendra et al., 2012) Therefore having an application to track and assist travel will be beneficial for a lot people including bloggers, reporters and tourists.

The application will allow users to find their current location, and add it to a locally stored database. Users adding their location allows them to look back at all the places they have been. Having a simple application to add the user's location immediately and quickly is important as according to "researchers often relied upon telephones, pagers, beepers or alarm clocks to either remind participants to keep a diary of the studied events or signal participants to answer a set of questions on a pre-designed questionnaire booklet ... The timeline of reminders and signals were often randomly scheduled and rarely occurred at the times they supposed to occur, i.e., immediately after the study event." (Fan et al., 2012) Meaning that more data is easily collected during or right after the event.

Users are also able to find their next location, by using the Google picker, allowing them to find shops and restaurants of their choice. Once a chosen location of their choice has been chosen an address will be displayed so the user is able to look back at this address.

Another feature on the application is the ability to take photos which are then automatically saved on to the gallery. Once the perfect photo has been taken it can be shared with friends and family using one of the many different sharing applications. This includes Google+, Facebook, email and many more. Sharing images with friends and family is a key feature as a the study by Kang and Schuett shows "that identification and inter-nalization are critical determinants that positively increase actual travel-experience sharing on social media as mediated by perceived enjoyment." (Kang and Schuett, 2013)

Apart from this the user is able to access a database list with a list of all the cities from <https://datahub.io/core/world-cities> . Therefore, finding new cities they can go to. Once a chosen location has been found they are able to book flights, restaurants and accommodation on the Trip Advisor website using the the link provided. "TripAdvisor is a website based on the idea that travellers rely on other travellers' reviews to plan their trips, or at least can be satisfactorily helped in their decisions by them. Presently TripAdvisor contains 10 million travel reviews and opinions and written by 5 million registered members and counts 25 million visitors per month" and " currently covers 212000

hotels, over 30000 destinations, and 74 000 attractions worldwide”. (Miguéns, Baggio and Costa, 2008) Having a button which opens to this link makes it easier for the user to access the website.

II. Implementation

To create this application, I used Android Studio 2.3.3 with a SDK target version of 23. Therefore, when testing the application, I am able to test this on a mobile device rather than on the emulator. This is because there will be issues when opening the camera when running on the emulator.

One of the main parts of the application is to be able to find your current location and add it to a list, which you can access. To be able to find the users location, ‘ACCESS_FINE_LOCATION’ and ‘ACCESS_COARSE_LOCATION’ permissions is required. When asking for fine location it will give permission to use the GPS and Network provider to determine the position, while coarse location only provides permission from the network however it provides a less accurate result than fine location. Therefore, within the ‘Track my Travel’ application if no network is found it will just use the GPS to find an approximation of location by fetching the latitude and longitude coordinates. And when both GPS and network based connection is connected, it will allow a more precise location.

The problem when developing this, the user had to exit the application to then open Google Maps to find the users current location. To resolve this, I added a function which opened google maps every time the user wanted to get their location. This enables them to look at their current location on the map, to make sure it is correct. Once they are happy with this, they are able to go back, and the users location information will be updated on the application.

Once the coordinates are found a geocoder class is needed to translate the latitude and longitude value into an address using ‘reverse geocoding’. This can be done by using the ‘Geocoder’ class within the ‘android.location’ package. This is a backend service which is not included in the Android framework.

Another dangerous permission added within the application is to give access to the camera, allowing users to capture pictures which will display within the application using Image View. These photos then can be shared with friends and family. It is important to ask for permissions as it contains sensitive information.

When opening activities such as TripAdvisor and Place Picker within the Location Information Activity. A network connection will be checked using ‘ACCESS_WIFI_STATE’ and ‘ACCESS_NETWORK_STATE’. TripAdvisor accesses information from a website, and place picker is a Google API that will require access to search for nearby locations which will then display an address. Checking this is important to because if users tries to access these feature without the network, it will not work. Therefore, as a prevention I have added error messages to appear, to make sure the

user turns on the network connection before accessing it. When doing this I had to consider how to present these error messages. I decided to go for an Alert message box errors as this will be unmissable and the user must read it before continuing, it also contains information why they are unable to access certain activity's. If the connection is successful a Toasty message will appear at the bottom of the application which uses a third-party library, the reason for this third-party library is so that it is more user friendly and makes the message it easier to view. The reason I have chosen a Toasty message not is that because it is already successful therefore the user does not need to do anything, therefore having a suitable message at the bottom is enough.

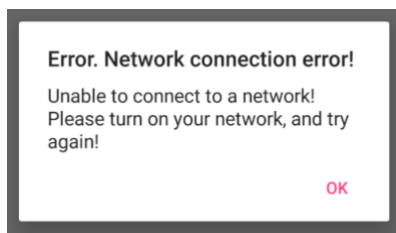


Figure 1 - Network Connection Alert

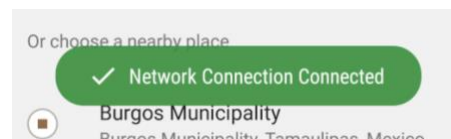


Figure 2 - Network Connection Toasty

```
allprojects {
    repositories {
        jcenter()
        maven {
            url 'https://maven.google.com/'
            name 'Google'
        }
        maven { url "https://jitpack.io" }
    }
}
```

Figure 3 - Build gradle repositories

Having these error handling is important to avoid errors within the application, and to make sure the user is aware why certain parts of the activity is not working, I have also used error handling when asking for permissions.

```
//If no network is found an error message will appear.
AlertDialog.Builder alert = new AlertDialog.Builder(locationActivity.this);
alert.setTitle("Unable to connect to a network!");
alert.setMessage("Network connection is needed for this activity. Please turn on your network, and try again!");
alert.setNegativeButton("Ok", (dialog, id) -> {
    //Alert will not do anything once it has been clicked.
})
```

Figure 4 - An example of code where an alert message has been used

Another main part of the application is having a database storing all the places, date and time the user has been at what specific time. This is so users are able to look at this information and recall where and when they have been to which specific place. The reason I have chosen to store data using SQLite, so users are able to store and remove information logically. It stores structured data within a private

database and is stored within the application and not in the internal or external storage of the device. Meaning that the database is only accessible within the application. As it is not stored within the network server it means that users are able to access the database without having access to network. It requires an ‘android.database.sqlite’ API package. The reason for this it is lightweight and can store a lot of data. The difficulties with SQLite is if the user would like to access this information or database outside the application it would be very difficult.

Another storage method within the application is saving the images to the gallery. This saves images internally within the device, so that users are also able to share these images with their friends. Having images saved on the device allows the user to access these whenever they would like. The reason that I have chosen to save the images to the gallery is because this is an application which you can not uninstall. Which is important as pictures of memories should not be removed.



Figure 5 - Search Filter for Cities



Figure 6 - SQLite Database

Within the application there is also list of cities, to be able to do this I had to look for a database containing all of this information (<https://datahub.io/core/world-cities>). This data is a JSON file, which is an alternative method to XML as it is used to transmit data between a web application and a server. Therefore, to display all this data I had to use parse JSON data and display it with the application. At first this was displayed as a text view, to present this better I have chosen to display the city name, sub-country and country information. However as there is a lot of data, I have decided to place a search button so that users are able to filter information easier. However, to improve this in the future I can also add a sorting or filtering method to find specific cities. Another additional feature that can be added is adding a tick list for each city, so that users are able to select which cities they have been to. Therefore, allowing them to keep track of which cities they have and have not been to.

III. Table of References

Fan, Y. *et al.* (2012) ‘Smartphone-Based Travel Experience Sampling and Behavior Intervention among Young Adults’, *Security*. Available at: <http://conservancy.umn.edu/handle/11299/132726>.

Jinendra, D. *et al.* (2012) ‘Smart Travel Guide: Application for Android Mobile’, Special Issue of International Journal of electronics Communication & Soft Computing Science & Engineering, pp. 115–120.

Kang, M. and Schuett, M. A. (2013) ‘Determinants of Sharing Travel Experiences in Social Media’, *Journal of Travel and Tourism Marketing*, 30(1–2), pp. 93–107. doi: 10.1080/10548408.2013.751237.

Miguéns, J., Baggio, R. and Costa, C. (2008) ‘Social media and Tourism Destinations: TripAdvisor Case Study’, *Advances in Tourism Research*, 26(28), pp. 26–28. doi: 10.1088/1751-8113/44/8/085201

IV. Resources

- Code within workshops
- <https://www.androidauthority.com/use-sqlite-store-data-app-599743/>
- <http://www.theappguruz.com/blog/android-using-sqlite-database>
- <https://www.android-examples.com/change-textview-text-programmatically-in-android/>
- <https://www.youtube.com/watch?v=gnEnOMCwND0>
- <https://blog.reigndesign.com/blog/using-your-own-sqlite-database-in-android-applications/>
- <http://www.viralandroid.com/2015/12/how-to-capture-image-from-android-camera-and-display-it-programmatically.html>
- <https://developer.android.com/training/camera/photobasics.html#TaskScalePhoto>
- <https://stuff.mit.edu/afs/sipb/project/android/docs/training/camera/photobasics.html>
- <https://developer.android.com/reference/android/provider/MediaStore.Images.Media.html>
- https://www.youtube.com/watch?annotation_id=annotation_3065812653&feature=iv&src_vid=SK98ayjkhk1E&v=aQAIMY-HzL8
- <http://www.mysamplecode.com/2012/06/android-internal-external-storage.html>
- <https://www.androidhive.info/2012/01/android-json-parsing-tutorial/>
- <https://developer.android.com/training/articles/security-tips.html>
- https://www.tutorialspoint.com/android/android_webview_layout.htm
- <https://datahub.io/core/world-cities>
- <http://android-coffee.com/get-data-from-website/>
- <https://developers.google.com/places/android-api/>
- <http://en.proft.me/2017/05/18/android-google-places-api-tutorial/>
- https://www.tutorialspoint.com/android/android_network_connection.htm
- <http://www.gettyimages.co.uk/photos/travel?sort=mostpopular&mediatype=photography&phrase=travel>
- <https://www.thecrazyprogrammer.com/2016/01/android-simple-listview-with-search-functionality-example.html>