

Mobile Application Development

COSC2309/2347 Semester 2, 2016

Assignment 2: Social Event Planner (30 marks)

You are to extend the Social Event Planner Application created in Assignment 1 to include a number of new features highlighted below:

Functional Requirements

1. **Context Aware (Location Based) Event Notification:** With people increasingly on the move it is easy to miss an event and nobody wants to be in the wrong place at the wrong time! Therefore your application will provide location based notifications for upcoming events. For example if you are one hour's travel away from the destination of an upcoming event you will be notified one hour before (plus a configurable threshold of say 15 minutes extra). You can use any of the built in Android APIs to support this but **MUST** create your own custom Service to facilitate this functionality.
2. **Google Distance Matrix API.** You can get the current location using the GPS of the device and you already have the locations of all of your events but you need some mechanism to calculate the time to travel between two locations using various modes of transport (you can use the default driving mode for this assignment). Google Distance Matrix API will be used for this purpose.

This API is a REST based web service, which can be queried using standard HTTP Get/Post requests and receiving JSON formatted responses (so as to not tie ourselves solely to using an Android client). Details on how to use this service can be found in [1] and will be discussed further in the lectures and workshops.

3. **Google Maps:** One of the licensing requirements of using the Distance Matrix API is to display locations on a Google Map. Android Studio provides a Google Maps Activity template you can use for this purpose.

Non-Functional Requirements:

1. **Local Database:** The Event model is to be persisted in a local SQLite database. Whilst the database can be simple (in terms of normalisation), it must store all information about Events discussed in Assignment 1. For efficiency you should also maintain an in memory model that is synced with the database and when doing lookups these can be read from memory to avoid a database call (which is only necessary for creation and edits).

Additional Requirements:

- Your application can only perform network related operations (Distance Matrix API) when connectivity is available on the device. As such, you should use a Broadcast Receiver to determine when network connectivity is lost and gained and should perform a location check via Google Distance Matrix API either when a specific time period has elapsed (e.g. 5 minutes) or if the network reconnects and the previous time period has elapsed.

Important:

- You **must** only make use of the libraries provided by the default Android SDK API. You should not make use third-party libraries such as the Google Client Libraries.

Other Requirements

- All non-trivial I/O operations must be performed in a separate worker thread (i.e. Not the UI Thread)
- Your User Interface must support all of the functionality presented under the “functional requirements” section of both assignments (Assignments 1 and 2). As with assignment 1 you should aim for a simple user interface model and can refer to the Android Material Design UI Guidelines for more information [2]
- Your implementation must make efficient use of UI resources through Styles, Themes and XML resources (such as Strings, Dimensions or Colors)
- Your Target Android Version should be API 19 or higher.

Advanced Functionality (Bonus Marks Max 4, Total Capped at 30)

When entering events you perform a distance based comparison of the new event compared to existing events to test whether it is feasible to attend them all based on the selected mode of transport. e.g. it is not possible to be in meetings in Melbourne and Bundoora RMIT campuses that are only half an hour apart when the mode is walking!

(2 Marks)

Perform multiple networks calls in parallel (the number of concurrent instances being configurable by the user) when calling the Distance Matrix API. Note that this is not necessary for this particular API since you can provide multiple locations in a single call, however it is a useful technique to implement since it can be used on other APIs that may benefit from multiple concurrent calls. **(2 Marks)**

Submission Instructions

Your project should be implemented using Android Studio and your project exported as a compressed .zip archive before uploading to Blackboard. **Do not** use any other compression formats - use of other formats (e.g. tar.gz, RAR, etc.) may lead to delays in marking and/or a deduction of assignment marks.

Important Regulations

- You are free to refer to textbooks and notes, and discuss the design issues (and associated general solutions) with your fellow members on Blackboard; however, the assignment should be your own individual work.
- Where you do make use of other references, please cite them in your work. Note that you will only be assessed on your own work so the use of third party designs is discouraged.

This assignment is due at 9:00am on Mon. 10th October 2016. Late marks will be deducted at 10% of the total marks per day up to a maximum of five days.

References:

- [1] *Google Distance Matrix API*
<https://developers.google.com/maps/documentation/distance-matrix/>
- [2] *Up and Running with Material Design.*
<http://developer.android.com/design/index.html>