Mobile Application Development
COSC2309/2347 Semester 1, 2019

Assignment 2: Movie Night Planner App Extended (30 marks)

You are to extend the Movie Night Planner Application created in Assignment 1 to include a number of new features described below:

Functional Requirements

1. **Local Database:** The complete model (all events and movie data) 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 (including attendees) and Movies as discussed in Assignment 1. On application startup it should read the contents of `events.txt` and `movies.txt` and populate the database if not already present. For efficiency you should maintain your original in memory model which is synced (loaded) in `onStart()` and saved/persisted in `onStop()`[1]. For cohesion, database functionality should reside in classes separate from the model, view and controller classes.

2. **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 movie events. For example, if you are one hour's travel away from the destination of an upcoming event you should be notified one hour before (plus a **configurable *notification threshold*** of say 15 minutes extra). The notification should be displayed in the Notification Area and provide Dismiss/Cancel/Remind in 'n' minutes options (where 'n' is **another user configurable setting i.e. *remind again duration***). NOTE: Dismiss means just clear the notification and do not produce any further notifications, whereas Cancel means actually cancel and remove the event.

   When the user selects Cancel, a confirmation dialog box should be displayed showing a summary of the event details so that the user can choose to confirm the permanent removal of the event. If the user does not confirm the deletion, then the notification is simply dismissed and the user will be reminded again in 'n' minutes.

   You can use any of the built in Android APIs to support this but MUST create your own custom Service to facilitate this functionality e.g. create either a `JobService` or use the `AlarmManager` to launch your own

[1] Alternatively you may wish to do this in the background for additional protection against data loss but this is not required for this assignment

started Service. A ***notification period*** with which location and distance is checked should be the **third user configurable setting** for this assignment.

Google Distance Matrix API. To implement (2) 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 mechanisms to calculate the time to travel between two locations using various modes of transport (you can just use the 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 receives JSON formatted responses (so as to not unnecessarily tie ourselves to Android APIs). Details on how to use this service can be found in [1] and will be discussed further in class.

3. **Display Event Location information on a Google Map:** 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. You should use the Map to display the location of the three soonest (i.e. closest in time in the future to the current time) event locations.

4. **Network Monitoring:** Since your application can only perform network related operations (Distance Matrix API) when connectivity is available on the device you should have a `BroadcastRecever` that listens for network state. You should perform an Event Notification check (requirement 2) whenever network connectivity is re-established.

Additional Implementation Requirements

- All non-trivial I/O operations must be performed in a separate worker thread (i.e. Not the UI Thread). Your implementation must have at least one `AsyncTask` and one started `Service` (which can be an `IntentService`).
- In addition to the database functionality specified above all user settings (*notification threshold, remind again duration*, and *notification period*) should be saved using a mechanism other than SQLite e.g. `SharedPreferences` or a file.
- Your User Interface must support all of the functionalities presented under the “functional requirements” section of both assignments (Assignments 1 and 2). As with assignment 1 you should aim for a simple/efficient user interface design 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 25 or higher **(you must implement runtime permission checking but can use Caspar's supplied code[2]).**

Code Quality

- You will be assessed on code quality using the same guidelines from assignment 1.

Submission Instructions

- You MUST make use of the libraries provided by the default Android SDK API. You must NOT make use of third-party libraries such as the Google Client Libraries.
- Your project should be implemented using Android Studio and your project exported as a compressed .zip archive before uploading to Canvas. **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 students on Canvas; however, the assignment should be your own individual work (or as a student pair as per assignment 1 rules).
- 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 **code** is prohibited.

This assignment is due 6:00PM Fri. 24th May 2019 (end of week 11) and is worth 30%. Late marks will be deducted at 10% of the total marks per day up to a maximum of five days (after which assignments will no longer be marked unless special consideration applies as per course guide).

References:

[1] *Google Distance Matrix API*
https://developers.google.com/maps/documentation/distance-matrix/

[2] *Up and Running with Material Design.*
https://developer.android.com/design/index.html

[2] This is the one exception of where you are allowed to use other/3rd party code in this assignment!