Mobile Application Development
COSC2309/2347 Semester 2, 2016
Social Event Planner

Assignment 1 (20 marks)

You are to implement a simple Social Event Planner app to create and schedule events and invite attendees. In this first assignment, you will implement the basic user interface for event creation, modification and calendar viewing. This will be extended in Assignment 2 to provide a full application complete with storage and networking/cloud service behaviour (including a location-based alarm that will trigger based on your distance from your current location to that of the event).

At the core of the functional requirements of this application is the following entity:

Event: An *event* is an activity that occurs at a specific date, time and location and has a number of attendees. It must (at a minimum) maintain the following information:

- **Title**: The title of the event (e.g. “Video game night”)
- **Start Date:** The start date and time of an Event (e.g. July 7th, 2016, 7.30PM)
- **End Date:** The end date and time of an Event (must be later than start date)
- **Venue:** The location of the event (e.g. “K’s House, 13 Node.js”)
- **Location:** A String representation of geographical latitude and longitude of the venue (e.g. -37.805631, 144.963053), you can get this from Google Maps web page directly (outside of your app).
- **Note**: Additional information about the event (e.g. “BYOD, costume and snacks!”)
- **Attendees**: A list of individuals who are invited to this event (as picked from the user’s contacts list).
- **Id**: A randomly generated combination of numbers and letters, which uniquely identifies an event (not visible to the user).

Functional Requirements
Your application must provide the following functionalities and meet the non-functional requirements stated under the “Other Requirements” section below.

- **Schedule and Unschedule an Event:** The application should allow the creation of an unbounded set of events. For simplicity, you can ignore duplicate entries or events with overlapping times (we will make sure test data does not overlap when assessing your assignment).

- **Edit Event Details:** Users should be able to edit the event details as well as add or remove attendees (using appropriate system activities and APIs).
- **View Events:** Users should be able to display a list of events where each element in the list will be a synoptic view (summary) of the Event e.g. title, date and number of attendees. The list should be sorted according to date (user can toggle ascending/descending order).
- **View Calendar:** Users should be able to view the entries in a calendar style layout, based on either* a week or month view (you can look at the calendar app on an Android device or emulator for ideas but are free to be creative with your layout). **IMPORTANT NOTE:** You must create your own UI using standard layouts i.e. you cannot just use a standard or third party Calendar widget.
- **Selection/Editing:** In the two view modes described above, users should be able to add or edit items via direct manipulation (e.g. long press or gesture).

Other Requirements:

- In assignment 1 you are not expected to persist data however your data must not be limited to the lifetime of any specific activity (i.e. must have application scope to facilitate testing). For simplicity you can also hard code test data (but put it in a separate class for cohesion and code readability).
- Your User Interface must support all of the functionalities presented under “functional requirements” above.
- Your implementation must make efficient use of common values (such as Strings, Dimensions or Colors) by (re)using values from the appropriate XML resource file. i.e. do not hardcode such values into your layout files or your application.
- Your Target Android Version should be API Level 19 or higher, however, for students with old devices who REALLY want to target lower levels you are allowed to do so.
- You can write your application in the single Activity per screen phone style however you may choose to write a tablet version using Fragments.
- You should use the Model View Controller approach to assist in writing modular and cohesive code.
- You may want to consider branding, business models, distribution frameworks, 3rd party integration (maps, navigation, social media sites) etc. and how this impacts on your app design although you will not be assessed on these features (these will be covered in assignment part 2).

* You only have to choose one or the other but if you are having fun with layouts you are welcome to create both so that you have a total of three different views!

Advanced Functionality (Bonus Marks - Max 4 marks and Total Assignment Marks are Capped at 20/20)

To receive these bonus marks you can provide any of the following additional functionalities at two marks each (up to a maximum of four bonus marks):

These are listed from easiest to hardest :)

1) Attach a recorded voice note as an additional data item for each event.

2) SMS a newly created Event to all of the attendees. Your app should also intercept incoming SMS's to determine whether it is a new event from another person. If so, your application should allow the user to either decline or accept and add the event accordingly.

3) Social media integration for attendee management.

Submission Instructions

Your project should be implemented using Android Studio and your project exported as a single compressed .zip archive before submission. **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 and APIs (beyond the standard Android libraries) is discouraged.

This assignment is due at 11.59pm on Fri. 26th August 2016