COSC 3P97 – Assignment 2

Due: Nov. 21st @ 5:00pm

For this assignment, your task is to create a simple application to create and manage meetings.

The minimal requirements are as follows:

- The ability to schedule (i.e. create) meetings
  - Including *at least* date and time
  - Making reasonable use of widgets
- The ability to have multiple meetings, even for the same day
- The ability to see all meetings for a single day
  - You have a small degree of flexibility here. At the very least, you should have some mechanism for seeing “all meetings today”, and “all meetings tomorrow”
    - This may come in the ability to see a summary of meetings for *any* day, or a week's view, or just a specialized “today” and “tomorrow” feature
    - This is one of the places where a better interface is particularly important (if it isn't easy to see what you have coming up at a glance, you'll simply stop checking)
- Must have **one** of the following:
  1. The ability to add contacts to meetings
     - You can use the built-in contacts list or create a new view to manage your own, just within the application
  2. The ability to add a locations to meetings
     - You must have some mechanism for managing locations
  - Note that if you're creating the mechanism for managing the contact/location, you should use reasonable widgets (I don't need to actually say that, right?)
  - It's perfectly fine to only allow one contact/location for a meeting
  - If, for some reason, you wish to allow for contacts *and* locations for meetings, you won't receive any extra marks for it, but it's still permissible
- Of course, you'll need persistence of meetings/contacts/locations across multiple executions
  - Use an appropriate mechanism for this
- Additional convenience features, including as a minimum:
  - The ability to manage locations/contacts outside of the view for a specific meeting
  - The ability to completely clear *all* meetings
  - The ability to clear all meetings for a single day (it's okay if you limit it to just “today”)
  - The ability to push all of today's meetings to another day
    - Ideally, to either the next weekday, *or* the next weekend day, depending on whether or not today is a weekday
      - e.g. if today is Sunday, it should push my golf game to next Saturday; if it's Tuesday, then push my trial to Wednesday; if it's Friday, then push my Illuminati conference to Monday
  - For whatever options you're including, make sure to use an appropriate UI (e.g. ActionBar)

Note that part of the point of this assignment is for you to start thinking about what makes a useable/reasonable interface, and how to write something that's functional and (reasonably) intuitive. As such, specific design instructions are omitted.

For example, you might decide that you'd like to use a pager for cycling through meetings, or you may prefer a list view (or something else reasonable). But you wouldn't want anything that looks like a 1990s javascript reject, tapping buttons to manually cycle through each entry. Similarly, you're expected to make decent use of space, both for portrait and landscape views (and will likely have those views be at least somewhat different from each other).

Also, I haven't told you whether expired meetings (those that precede today/now) should be deleted automatically or kept for the sake of future reference. This is because it's up to you. Either would be a valid decision. Maybe you'd prefer to split the difference, and have an option to mass-delete expired meetings.

I haven't told you what to do if a meeting has a location, but then that location is deleted. Handle it in a way that you feel is reasonable. (Note: *playing a sad "wah wah wahhh" trombone sound and then crashing* would not qualify as "reasonable")

Submission:
Submit a folder *containing* your project folder. Include within that top-level folder instructions, as well as any other pertinent details, in a .pdf file. If your project is best-tested on a specific API level, be sure to include that in your instructions. You won't be graded on these instructions, but evaluation of your project may not be favourable if you don't make an effort to make it understandable to the marker.

Submit electronically on sandcastle, via the `submit3p97` script.
Be sure to submit this as assignment **#2**.