

Mobile Application Development

COSC2309/2347 Semester 1, 2013

Social Organiser/MeetUp App

Assignment 2 (30 marks)

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

Functional Requirements

1. **Notification of Event:** Users are to be notified **15 minutes** before the due date/time of an *Event*. Such notifications should be performed **irrespective of whether the application is running at the time an event is due**. You can use any of the built in Android APIs to support this but **MUST create your own custom Service** to facilitate this functionality.

Non-Functional Requirements:

1. **Local Database:** The Event **model** is to be persisted in a **local SQLite database**. Whilst the database can be simple, it must store all information about Events discussed in Assignment 1.
2. **Google Tasks API:** Your application must integrate with the Google Tasks API, in order to offer **persistence of event data in the cloud**. The Google Tasks API is a **REST based web service**, which allows for the **storage and management of categorically organized To-Do Lists**, and while it is **not designed exactly for scheduling events** it provides a mechanism where the **mapping should be straight-forward**. Interaction with this service involves performing **HTTP Get/Post requests** and **receiving JSON formatted responses**. Details on how to use this service can be found in [1] [2] [3] [4] and will be discussed further in the lectures and workshops.

Additional Requirements:

- For simplicity, it is assumed that the **local database is the primary data source of your application** and as such, **all changes propagate from the client application to the remote server only** i.e. You do not have to detect changes performed on the remote data by any third-party application such as a web interface. However, **a manual means (button, action item etc.) of retrieving all data on the server and overriding the local database should be presented to the user.**
- Your application should perform **network related operations only when connectivity is available** on the device. As such, **network requests generated as a result of changes to the local database must be queued until such time as network connectivity is available.**

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 functionalities presented under the “functional requirements” section of the assignment (Assignments 1 and 2) and offer an intuitive (and efficient) means of interaction with the user. Refer to the Android UI Guidelines for more information [5]
- 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 17 (see additional information below).

Advanced Functionality (Bonus Marks: 4 Points)

Perform full syncing with the Cloud resource allowing a service to fetch changes as they occur on the Google Tasks Cloud Service, as manipulated through its web interface (in Gmail).

Additional Information:

In order to use your Google account (for integration with the Google Tasks API), you will need to create an emulator with a “Google APIs, (Google Inc.)- API Level 17” target.

Note also that by default the ATOM x86 AVD images do not support the Google APIs and you should use a standard ARM image with Google API support.

We will post an update if we are able to find a simple way to get Google APIs to work with the x86 emulator (since it appears to be possible with workarounds).

- To allow full Internet connectivity from the Android emulators in the lab machines

Go to: Settings > Wireless & Networks > Mobile networks > Access Point Names > T-Mobile US .. then enter proxy details (proxy.rmit.edu.au and port 8080)

NOTE: We were not able to get the API Level 10 to access Google Accounts due to firewall issues so we are limited to using API 17.

Submission Instructions

Your project should be implemented using **eclipse** and your project exported as a compressed **.zip** archive before uploading to **Weblearn**. **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.

Resources:

- [1] Google Inc. *Google Tasks API*. Available:
<https://developers.google.com/google-apps/tasks/>
- [2] Google Inc. *Google Tasks API Reference*. Available:
<https://developers.google.com/google-apps/tasks/v1/reference/>
- [3] Google Inc. *Authenticating to OAuth2 Services*. Available:
<http://developer.android.com/training/id-auth/authenticate.html>
- [4] Google Inc. *Write Your First App: Prerequisites (Google Tasks API)*. Available: <https://developers.google.com/google-apps/tasks/firstapp>
- [5] Google Inc. *Android Design*. Available:
<http://developer.android.com/design/index.html>