ECE 1778 - Creativity and Programming for Mobile Devices
February 2013
Programming Assignment #P4, for Programmers

Threads and Databases

The goal of this assignment is to learn the how to offload non-user interface tasks into separate threads, and to get a handle on the basics of a simple on-device database.

1 Reading

Read the following sections from the course texts, if you are developing on Android:

i. Pages 201-211 ("The WebView Widget") of the **The Busy Coder's Guide to Android Development**, version 4.4.
ii. Pages 367-369 ("Dealing with Threads") of the **The Busy Coder's Guide to Android Development**, version 4.4.
iii. Pages 467-505 ("SQL Lite Databases") of the **The Busy Coder's Guide to Android Development**, version 4.4.

For iPhone: from **Beginning iPhone 6 Development Exploring the iOS SDK** by Mark, Nutting, LaMarche and Olsson and the web**,** read:

i. See: http://www.iphonesdkarticles.com/2008/08/uiwebview-tutorial.html on how to launch a browser within your application.
ii. Lookup the method initWithContentsOfURL in the iPhone Documentation.
iii. Chapter 15 ("Grand Central Dispatch, Background Processing, and You").
iv. Chapter 13 ("Basic Data Persistence"), the section on Using SQLite3.

2 Assignment

NOTE: As in previous assignments, when writing your code for this assignment, please be sure to follow 'Braiden Brousseau's Guide To Quality Apps' that was given as part of Assignment P1. Part of your grade will be assigned for fulfilling them.

You are to write an application that creates a small SQL database that is populated from a file (that contains a list of names) located at a specified URL on the Internet. It will then search for those names on the Internet, and display the results. The application will do two different pieces of work (populating the database and searching) by spawning two separate threads from the main UI activity. Here is the specification in more detail:

When the application launches, it should present the user with three interface widgets:

1. A text field (call it DBURL) that defaults to the following string: 'http://www.eecg.utoronto.ca/~jayar/PeopleList' This string should be changeable by the user.
2. A button that is labeled 'Populate' that is initially active.
3. A button that is labeled 'Search' that is initially **inactive**. (i.e. touching it causes nothing to happen).

When the *Populate* button is touched, the App should spawn a thread that goes to the DBURL (which is a simple text file of names, one per line) and populate the database with the list of names there. While this is happening, the UI should display some kind of 'I know you're waiting' visual – such as a progress bar or spinning wheel. At this point it should not be possible to activate the *Search* button.

When the database is populated the first thread should send a message to the main UI process that it has finished (and then it should terminate), and the UI should both stop displaying the wait motif, and should emit a 'toast' (Android) or an 'alert' (iPhone) that indicates the database is loaded.

At this point the second button ('Search') should become active. When it is touched, a second thread should be spawned that walks through each of the names in the database, and emits a query to Google on each of the names in the database, and display the result as a web page, using the WebKit browser (or just the UIWebView on the iPhone). Each display of the results should stay on the screen until the user pushes the 'back' button, and then the result for the next name displayed.

Due: February 26th, at 6pm, 0.5 marks off every hour late. Submit your solution through Blackboard associated with assignment P4.

What to submit: a zip file containing your complete project, runnable from Eclipse, or runnable from Xcode if on iPhone.