

ECE 1778 - Creativity and Programming for Mobile Devices
January 2011
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. Chapter 17 (“Embedding the WebKit Browser”) of the **The Busy Coder's Guide to Android Development**.
- ii. Chapter 21 (“Dealing with Threads”) of the **The Busy Coder's Guide to Android Development**.
- iii. Chapter 27 (“Managing and Accessing Local Databases”) of the **The Busy Coder's Guide to Android Development**.

The equivalent from Mark, Nutting and LaMarche, **Beginning iPhone 4 development**, can be found in:

- 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 13 (“Grand Central Dispatch, Background Processing, and You”).
- iv. Chapter 12 (“Basic Data Persistence”), see the section on Using iOS’s Embedded SQLite3.

2 Assignment

Write an Android application that creates a small SQL database, and spawns two threads. The first thread should open the following file from the network – <http://www.eecg.utoronto.ca/~jayar/file1> - and populate the database with the list of names there. Once done, that thread should send a message to the main UI process that it has finished (and then it should terminate); the main UI thread should emit a ‘toast’ (Android) or an ‘alert’ (iPhone) that indicates this is done.

The second thread should wait for the first thread to be finished, and then walk through each of the names in the data base, and emit 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.

Due date: March 1st, at 9am. Submit your solution through Blackboard.

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