Mass Transit – Assignment 4

Due: April 7, 2014
Spring 2014 Computer Science 411
Mobile Device Application Programming

Instructions: Please submit your solution to the following exercise by 23:59 on April 7, 2014 via «`http://dropbox.ecs.fullerton.edu/`». Your submission must, at a minimum, include a plain ASCII text file `README.txt`, all necessary source files, libraries, and build configuration files to allow the submission to be built and run independently by the instructor. **All files** must include a header identifying the author, author's contact information, and a brief description of the file. The `README.txt` file must describe any bugs and features along with a description of what was or was not completed in the solution. The `README.txt` must also describe the use of the program and if there are any external dependencies. **Do not include any object files, binary executables, or other superfluous files.**

Place your submission in a folder and name it *lastname.firstname_asgt4*. **Zip** the folder and submit the zipped folder, named *lastname.firstname_asgt4.zip*, via «`http://dropbox.ecs.fullerton.edu/`». For example, if your name was Tilly Titan, then your file would be named *Titan.Tilly_asgt4.zip*. **Do not use any other archiving method other than zip.**

Failure to follow these instructions will detrimentally effect your assignment's score.

Plagiarism and academic dishonesty is not tolerated. Correctly and properly attribute all third party material and references.

Do not make calls to `system( )`, `exec( )` or similar system calls which trigger the operating system to execute an external program.

Assignment

Objective: The objective of this assignment is to gain experience using different views such as table views, scroll views, and web views and relational databases. Additionally, the objective is to gain experience subclassing such views to behave appropriately for the student's application. This application will also use the Tab Bar Controller. This assignment results in creating an iPhone application called *MassTransit* which is a utility for finding mass transit maps and time tables.
Rubric: The assignment is out of 100 points.

- 75 points for a correctly executing application
- 15 points for compiling with no errors or warnings
- 10 points for using the correct name for files, the program and documenting your code.

Not including a `README.txt` forfeits 25 points. Your code is expected to be well organized and well documented[1]. Points will be deducted for poorly structured or undocumented code.
Prerequisites: In order to complete the exercise successfully, you will need to have the following:

- Apple Xcode
- Familiarity with the bus schedules and maps from OCTA, «`http://www.octa.net/bus-routes-and-schedules.aspx`»
- Familiarity with the train schedules and maps from Metrolink, «`http://metrolinktrains.com/schedules/`»
- Familiarity with the bus and rail schedules and maps from MTA, «`http://www.metro.net/riding/maps/`»
- SQLite3 format of Google Transit Feed data for each system operator «`https://gamble.ecs.fullerton.edu/teaching/spring14/cs411/sample-code`»
- Transit agency system maps «`https://gamble.ecs.fullerton.edu/teaching/spring14/cs411/sample-code`»

Requirements: Students first select two transit agencies from the following list:

- Los Angeles MTA

[1] Well organized and well documented means sufficient comments to explain what all the functions do and even what is going on within a function

- OCTA
- Metrolink

The application must use a Tab Bar Controller. The tabs on the bottom, from left to right are *transit agency A*, *transit agency B*, *transit agency A map*, and *transit agency B map*. For example, if you select OCTA and Metrolink then the tabs along the bottom are *OCTA*, *Metrolink*, *OCTA map*, and *Metrolink map*.

The first two tabs perform identical actions. They present the user with a table view of all the routes the transit agency provides. Tapping on a route, shows another table view which is an ordered list of all the stops along the route. Tapping on a stop shows a detail view of the stop which includes any relevent strings from the database and a map of the stop location.

The last two tabs perform identical actions as well. They present the user with a pan and zoom interface to the transit agency's PDF system map.

Please use your imagination and your own experience using mass transit to devise an application that you would find useful and meaningful. This will undoubtedly mean going beyond the bare minimum of the assignment's requirements.

Consider that for your application to be more useful than a printed schedule, the schedule and map data will require to be put into some sort of data structure that enables the programmer to ask meaningful questions and receiving meaningful answers. For example, if a bus schedule is encoded into an object that answers a question such as *when is the next bus, given the current time?*, then the bus schedule must be encoded in such a way that is more than just a naïve look up table.

This application may not use any network services such as fetching data via http except for geocoding and map data.

Google Transit Data Feed: This is a specification published by Google that many transit operators publish their data in. Details about the specification can be found at «`https://developers.google.com/transit/`». A direct link to the reference at «`https://developers.google.com/transit/gtfs/reference`». This assignment requires mastery of the static transit feed data.

A list of public feeds is available at «`http://code.google.com/p/googletransitdatafeed/wiki/PublicFeeds`». The following are the links used to retrieve the raw data used to create the SQLite databases:

- «`http://www.octa.net/current/google_transit.zip`»
- «`http://developer.metro.net/gtfs/google_transit.zip`»
- «`http://www.metrolinktrains.com/content/google_transit.zip`»

Suggested Improvements: The following are a few suggested improvements to the project. Do not attempt these until you have completed the base requirements.

- Add an element where one can follow the twitter feed of the transit agencies to learn of delays and service changes.
- Using the current time and the schedule, calculate the next expected train or bus.
- Switch between daytime, evening and weekend schedules by calculating the current time.
- Use the GPS to figure out where the closest bus stop or rail station is.
- Incorporate the GTFS-realtime, «`https://developers.google.com/transit/gtfs-realtime/`».

$Id: masstransit_app.tex 4874 2014-03-19 04:33:46Z mshafae $