iOS App Devlopment - Project

SUPINFO
International University

INSTITUTE OF INFORMATION TECHNOLOGY

iOS App Development

Graded Exercise - Project

Objectives

Add Advanced features to an existing app

Version 0.1a
Last update: 19/03/2011
Use: Students

Authors: F. PETIT / S. CUELLA

Index

1. Presentation

You will have to work in pairs.

You've already developed foundations of an iPhone app during the labs. You now have to add additional features to your app.

Remember that you had to programmatically (by writing explicit creation code) create contacts to use the app. The company you are working for has database servers that contain the contacts you need to display in your application. A Web Application developer from the company developed a webservice that retrieve the data you need about contacts.

You will have to perform requests on this webservice and to gather information from it.

Your boss pointed out that your app must be able to display the contacts whether the device connectivity is disabled or the webservice can't be reached. So you will have to persist data between app usages.

2. Webservices

You've already used a webservice in your app, when you were doing requests to Google about how to transform addresses in coordinates.

The webservice you will have to request is located at http://www.labo-mobile-dev.eu/project/export.html or

If you can't reach this page use the export.html bundled with this project, turn on the webserver of your mac, put the file at your webserver root and try to access it at http://localhost/export.html

Now you will have to implement a connection and data retrieval from the server. When your app starts, it has to connect to the webservice and parse the response data to create contacts instances.

Be aware that your app has to manage error (i.e. the webserver is unreachable)

Note: the webservice exports data following the JavaScript Object Notation. (It could have been written in XML or following a custom scheme), this technique is widely used in mobile app development world. So it is quite representative of actual business.

Note: You will probably need a formatter/validator to understand the data returned by the webservice, those 2 are quite popular: http://jsonformatter.curiousconcept.com/ and http://jsonlint.com/

Remember that you already have a JSON parser in your project, so feel free to use it.

3. Persistence

There are many ways to persist data on iOS (raw binary files, XML, and SQLite databases).
Your boss wants you to use SQLite 3. SQLite is a lightweight database engine that works perfectly with mobile devices (it is widely use and works on other popular platforms like ANDROID).
More info about SQLite can be found at the official website, Wikipedia and Apple documentation.

You are now free to choose how to use SQLite persistence:

- Write and hard code your own SQL requests (using SQLite3 framework).
- Use Core Data API (it abstracts data model and persistence to an higher level)

There's a guide about Core Data programming at Apple's Documentation:
http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/CoreData/cdProgrammingGuide.html

If you choose to use Core Data you will have to follow the guide and examples provided by Apple that include to get rid of the old dataclass contact and the creation of a new one using a data model file. (Graphic entity design tool)

Using Core Data for the first time is quite difficult but not impossible, those who managed Core Data will be rewarded with 2 additionnal points (if it works…).

Remember what persistence means: every contact added or fetched from the webservice has to be saved and need to be read when app is next launched.

4. Notation

Feature	Points
Parse JSON and transform it in Contact instances	4
Webservice interaction – notify user if webservice is unreachable	1
Contacts are saved in SQLite database	3
Contacts are loaded from SQLite database	3
Contact Management – Add a new contact	2
Contact Management – Edit a contact	2
Contact Management – Remove a contact	1
User Interface	2
Convention & Coding Quality (Memory management)	2
Bonus – Core Data usage	(+2)
TOTAL	20 (+2)

5. Hand back your project

Create an archive of your project and send it to your trainer by email. The subject must be formatted as follows:

“[2APL] Project by %s(%d),%s(%d)”,name1,id_booster1,name2,id_booster2

Please conform to any additional guideline given by your trainer.