**To Do List Test App - Documentation**

Developer: Carmen Albu

Xcode: version 6.1.1

iOS: version 8.1

Requirements: I’ve used CocoaPods, and in order for everything to be functional, please run “pod install” in terminal in the directory of the project “ToDoListTest”.

To Do List Test App was developed as a demonstration test, for OnTheBeach company. It is a simple to do list, with the following features:

* add item to list;
* edit item from the list;
* delete item from the list;
* mark item as complete/incomplete
* persist data on device
* sync and persist data to a remote service

I have used Parse as a remote service. Parse is a service to store data and to handle it efficiently. I prepared my app to login/sign up with email and sign in with Facebook.

Parse also has a feature to use cached data before checking the server. If there is any cached data it will show it first and afterwards it will update with what is new from the server.

A better solution for smart caching would be an SQLite local database where we handle our inserting, deleting and updating of to do items and keep it synchronized with the Parse data. (It is noted as a future task)

**Models**:

UserModel – string objectId, string username, string email;

ToDoModel – string objectId, string itemTitle, string itemDescription,

NSDate itemDate, BOOL completed;

**Login Screen/Sign Up Screen**:

Login/Sign Up with email – the fields are mandatory and there is a validation on the email field.

Login with facebook – grabs user’s data such as email, name and creates user on server. Unfortunately, when logging out you cannot login back, because there is an issue from Parse side, where when you call the logout method it keeps Facebook session opened, I found out really late to fix it and manually handle the Facebook login.

**ToDoList Screen:**

It shows a list of To Do items (incomplete) and Completed to do items. If you hold long press on a cell it will update its status to Completed/Incomplete (it depends on its current completed status). You can delete an item if you slide on it.

When grabbing all the to do items from the server, these are ordered descending after updated date. Parse does this automatically: adding a column for created and updated date for each row. Every time you save/update it keeps this columns updated with the correct information.

When selecting a cell it sends as a property to the Detailed View Screen the ToDoModel that is attached to the cell.

The Bar button item on the left calls the logout method. An alert for confirmations is shown.

The Bar button item on the right adds a new to do item to the list.

The list is cached on the device, as said that Parse helps in this process.

**The Detailed To Do Item – Screen:**

It has two TextViews – one for the title, and one for description. It only saves it if these are not empty, or at least one it doesn’t contain the default text.

When leaving this view it saves it automatically.

The bar button on the right marks the item as completed/incomplete.

**Other mentions:**

I keep my arrays for incomplete and complete items in the AppDelegate. So any modification I make at an item, the arrays are updated and also the tables. When I make current modifications in the table view (To Do List ), like changing the status when holding long press on a cell, I do this modifications directly on the table but also in the local arrays and afterwards syncing with the server.

Also when adding/updating existing item in to do list – first I add to the local arrays, afterwards I save it on the server, in the background (not on the main thread), and afterwards synched the cached data as well.

**Unit Tests:**

I developed Test Cases for each view controller. Performance tests as well as simple tests. For this project I’ve only developed white box testing and for demonstration purposes there are also tests with completion expectations for asynchronous methods.

**CocoaPods** – I’ve used CocoaPods to install external libraries for Parse and Reachability.

Thank you.