**Hướng dẫn sử dụng SetaBase**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Notes |
| 2013-01-23 | 1.0 | ThanhLV |  |

**I. Tạo Project từ SetaBase**

Bước 1: Copy toàn bộ thư mục SetaBase sang một thư mục khác.

Bước 2: Chuyển sang thư mục vừa copy, đổi tên thư mục SetaBase thành tên Project chuẩn bị làm.

Bước 3: Mở file Project và đổi tên Project trong Xcode.

Bước 4: Thay đổi Scheme theo tên Project

Bước 5: Bổ sung các thư viện cần thiết (tìm trong Other Libs) và thiết lập các thuộc tính của Project mới. Bổ sung Unit Tests nếu cần.

**II. Cấu trúc Project**

1. **ThirdParty**

Tập trung các thư viện opensource / closedsource.

SetaBase đã tích hợp sẵn các thư viện thường dùng:

* AFNetworking: sử dụng để kết nối API
* JSONKit: thư viện xử lý JSON
* AsyncImageView: hiển thị ảnh bất đồng bộ
* MagicalRecord (+mod): thao tác với CoreData theo ActiveRecord
* MBProgressHUD: hiển thị trạng thái loading, notify.

Các thư viện khác có thể tìm trong thư mục Other Libs và file Open Source Libraries.docx

1. **Resources**

Tập trung các resources của Project, phân loại theo dạng resource như ảnh, HTML, fonts,…

Khi kéo resources vào project phải chia nhỏ group theo chức năng/màn hình.

1. **Classes**

Tập trung các Class xử lý. Các Class dạng Manager phải tạo group mới.

* *Helpers:* một số class tiện ích hay dùng. Các function bổ sung viết thêm vào class Util.
* *Categories:* các function bổ sung cho các class cơ sở của iOS.
* *Enums:* định nghĩa các enum của project
* *Model:* chứa data model, các class NSManagedObject (generated by Xcode), các class object (như DTO).
* *CustomControls:* chứa các Control viết riêng cho Project.
* *DataManager:* class xử lý các tác vụ liên quan đến Database với các model / class object từ group Model. Sử dụng ActiveRecord Design Pattern dùng lib MagicalRecord.
* *APIClient:* class xử lý các tác vụ liên quan đến API
* *Common:* chứa các define chung của Project.
  + Session: class xử lý các tác vụ liên quan đến Session của App, hỗ trợ sẵn các property thường dùng và thao tác load/save Session.
  + Macro.h: định nghĩa các macro hay dùng
  + Global.h: định nghĩa các Constant và các thành phần dùng toàn cục.
  + AppDelegate.h: AppDelegate của App.
* *Views*

Tập trung các Views / ViewControllers của App, chia group theo chức năng / màn hình của App. 3 file .h, .m, .xib phải để cùng một group.

* + BaseViewController: ViewController cơ sở, các ViewController tạo mới phải kế thừa từ BaseViewController.

**III. Một số lưu ý**

* Support iOS >= 5.0
* Sử dụng ARC
* Sử dụng Xcode 4.5 trở lên
* Các class dạng Manager viết theo Singleton Design Pattern, cách thức giống class DataManager.
* Để log sử dụng DLog(@””)
* Ưu tiên sử dụng các Macro (được định nghĩa trong file Macro.h), ví dụ NEW\_VC()
* Các function mở rộng cho các class cơ sở tìm trong mục Categories trước khi bổ sung từ nguồn khác. Các function tiện ích tìm và bổ sung ở class Util.

**IV. Link cho các lib được sử dụng chính**

\* MagicalRecord:

<https://github.com/magicalpanda/MagicalRecord>

Mod: ObjectiveRecord

<https://github.com/mneorr/Objective-Record>

\* AFNetworking:

<https://github.com/AFNetworking/AFNetworking>

\* AsyncImageView:

<https://github.com/nicklockwood/AsyncImageView>

\* JSONKit:

<https://github.com/johnezang/JSONKit>

\* MBProgressHUD

<https://github.com/jdg/MBProgressHUD>

\* iphonelib

<https://github.com/meinside/iphonelib>

Các libs khác tham khảo Other Libs/Open Source Libraries.docx

**V. Một số đoạn code tham khảo**

**\* Thao tác Session:**

self.session.userID = nil;

self.session.isAutoLoggedIn = NO;

[self.session save];

**\* Thao tác DB**

//Query

NSPredicate \*shopTypePredicate = [NSPredicate predicateWithFormat:@"%K like %@", kName, keyword];

NSArray \*shopTypeList = [ShopType findAllWithPredicate:shopTypePredicate];

//Create

City \*city = [City create];

city.item\_id = @([[dict objectForKey:kID] intValue]);

city.name = [dict objectForKey:kName];

city.prefecture\_id = @([[dict objectForKey:kPrefectureID] intValue]);

[city save];

//Delete

NSArray \*favorites = [Favorites where:@{@"item\_id":@(itemID),@"isShop":@(isShop)}];

if (favorites && [favorites count] > 0) {

[favorites each:^(id object) {

[(Favorites\*)object deleteInContext:[NSManagedObjectContext defaultContext]];

}];

}

[[NSManagedObjectContext defaultContext] save];