

The Social Software Application Based on iOS

Menglan Ge, Jiao Jin, Yue Zhao, Tingting Chu, Mengqi Li, Dexiang Zhang, Chengcheng Ren*

School of Electrical Engineering and Automation, Anhui University, Hefei Anhui
Email: ^{*}chengcheng_ren@ahu.edu.cn

Received: May 30th, 2016; accepted: Jun. 10th, 2016; published: Jun. 13th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper analyzes the purpose and significance of the project from the development of the iOS system, and introduces the development environment Xcode use of the project, and the development and superiority of the software swift language. At last, the software is tested, which shows the process of the software. Through this App, users can share their own work experience; it is more convenient and fast to communicate with others; and users can make like-minded friends, to enhance the life experience of the user side.

Keywords

App, iOS System, Swift Language

基于iOS的App社交软件

葛孟兰, 金 娇, 赵 悦, 储婷婷, 李梦琪, 张德祥, 任乘乘*

安徽大学电气工程与自动化学院, 安徽 合肥
Email: ^{*}chengcheng_ren@ahu.edu.cn

收稿日期: 2016年5月30日; 录用日期: 2016年6月10日; 发布日期: 2016年6月13日

*通讯作者。

文章引用: 葛孟兰, 金娇, 赵悦, 储婷婷, 李梦琪, 张德祥, 任乘乘. 基于 iOS 的 App 社交软件[J]. 软件工程与应用, 2016, 5(3): 190-197. <http://dx.doi.org/10.12677/sea.2016.53021>

摘要

本论文从iOS系统的发展状况、研究前景分析了项目的目的和意义。介绍了项目的开发环境Xcode的用法,软件swift语言的发展和优越性。最后进行了软件的测试,展示了本软件的使用流程。通过本App,用户可以分享自己的工作生活经验,更加方便快捷的与别人交流,并交到志同道合的朋友,提升了用户方方面面的生活体验。

关键词

App, iOS系统, Swift语言

1. 引言

21世纪信息飞速发展,各种信息通讯工具鳞次栉比、层出不穷,智能手机、手提电脑、平板等等更新换代的频率越来越快,极大地方便并丰富了人们的工作与生活。为顺应时代发展潮流,增强创新能力,展现新时代大学生的人文科技素养和积极向上风貌,本团队开发一款新型软件交流平台。通过此App,用户可以寻找到自己喜爱的书籍,并通过评分和他人推荐发现更多可能喜爱的好书籍,为实现个性化与独立化,此App并提供个人登录账号,用户可以分享自己的读书心得并建立起个人的书库。任何系统的设计开发都是一个不断改进的过程[1]。以此为据,我们随后将在影音、购物、美食、旅游等各个领域建立起相似结构,并进行归纳整合,形成一套完整的个人分享社交平台。

2. 系统可行性分析

2.1. 经济可行性分析

该软件内容充实,使用方便,开发周期短,投资小,功能齐全且实用性价值较大。如今,这类产品在市场上不多,所以很有市场前景,投入运营后提供免费下载,通过植入一些广告来获取一些利润。所以该软件为开发团队带来的效益远超过前期的投入。

2.2. 技术可行性分析

我们的项目是运行在 iPhone 和 iPad 等 iOS 平台上的 App,使用 Apple 最新发布开发语言 Swift 和开发工具 Xcode。iOS 开发只能采用基于苹果操作系统 MAC OS X 的 XCode 开发套件作为开发环境[2]。Xcode 具有统一的用户界面设计,编码、测试、调试都在一个简单的窗口内完成。

Swift 语言是 Apple 在 WWDC2014 所发布的一门编程语言,用来撰写 OSX 和 iOS 应用程序。Swift 语言的特点是:快速、现代、安全、互动,且全面优于 Objective-C 语言。Xcode Playgrounds 功能是提供强大的互动效果,能让 Swift 源代码在撰写过程中能实时显示出其运行结果[3]。

3. 软件设计的要求

掌握 Swift 语言的语法,加深对所学内容的理解,提高解决实际编程问题的能力。理解 iOS 界面开发的相关技术,如故事版 Storyboard, xib, AutoLayout 和 Size Classes 等。理解数据处理和存储方面的知识,会利用 AFNetworking 去请求网络数据并解析成自定义对象。

学习 CocoaPods 的相关操作,会使用它下载 iOS 开发需要的 framework。学习 UIViewController 的基

本功能，以及 UITableView 的数据源和代理相关功能。

从总体上把握 iOS App 开发的各方面，最终完成这款 App 的开发。

4. 系统设计与分析

该 App 完整的结构层次图如下图 1。

- 1) 主界面：打开软件，主界面有含大量目标对象的列表，其后具评分，供用户进行参考选择。
- 2) 搜索：列表顶部有搜索框可供搜索，可以直接填写搜索条件，回车确认后就从服务器下载搜索结果显示在另一个列表中。
- 3) 登录：主界面右上角含登录按钮，如果需要使用用户相关功能如收藏和评论，需要先登录，当登录成功时会自动打开用户信息页。
- 4) 评论和收藏：登录后返回主界面，选择目标对象后进入，可直接进行评论和收藏操作。

5. 设计详细步骤

5.1. 新建项目

新建一个项目，新建相应的包，User 对应用户操作如登录、评论和收藏的相关包，Detail 是图书详情功能、Index 对应首页，Expand 对应项目的一些扩展功能，Network 存放 App 网络层操作，Search 对应项目搜索功能。

5.2. 设置项目属性

这里我们可以根据需要设置该项目的属性。下面简介图 2 中的设置选项：

- 1) 标识符(Identify)。包含捆绑标识符、软件版本、版本标记、软件设计团队。其中捆绑标识符不可重复，本软件是我们首次开发，目前最新版本 1.0，版本号里的 Build 的 1 说明这个版本是第一次编译的结果，软件设计团队选择 Booking 表示项目由私人团队开发。
- 2) 配置信息(Deployment Info)。包括目标设置、选择设备、主界面、设备方向、状态栏的设置。因为现在的 iOS 设备基本是 iOS8 和 iOS9，所以 DeploymentTarget 选择支持 iOS8.0 以上，本项目使用了 StoryBoard 和 AutoLayout，所以设备可以选择 iPhone 和 iPad，设备方向配置项目启动的 StoryBoard，项目支持 Portrait (竖屏), Landscape Left (横屏向左), Landscape Right (横屏向右), 状态栏我们设置亮度可选。
- 3) 文件命名与尺寸(App Icons and Launch Images)。App Icons Source 应用程序图标源。点击可选择不同程序图标源，这里设置为 Appicon。Launch Images Source 启动图片源，设置为自定义的 Use Asset Catalog。Launch screen file 启动屏幕文件。在导航页面中设置了 Launch screen file 选择，在右侧选项卡中可以选择。这里我们设置默认选项为 LaunchScreen。如下图 2 所示。

5.3. 使用 CocoaPods 下载相关的库

当你开发 iOS 应用时，会经常使用到很多第三方开源类库，比如 JSONKit, AFNetworking 等等。可能某个类库又用到其他类库，所以要使用它，必须得另外下载其他类库，而其他类库又用到其他类库，这也许是比较特殊的情况。另外一种常见情况是，你项目中用到的类库有更新，你必须得重新下载新版本，重新加入到项目中，十分麻烦。因此需要用 CocoaPods。

CocoaPods 应该是 iOS 最常用最有名的类库管理工具了，上述问题，通过 cocoaPods，只需要一行命令就可以完全解决，重要的是，绝大部分有名的开源类库，都支持 CocoaPods。所以，掌握 CocoaPods 的使用是必不可少的基本技能了。

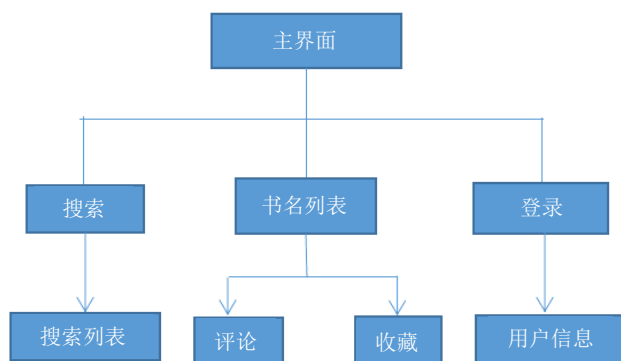


Figure 1. The hierarchical figure of App

图 1. App 结构层次图

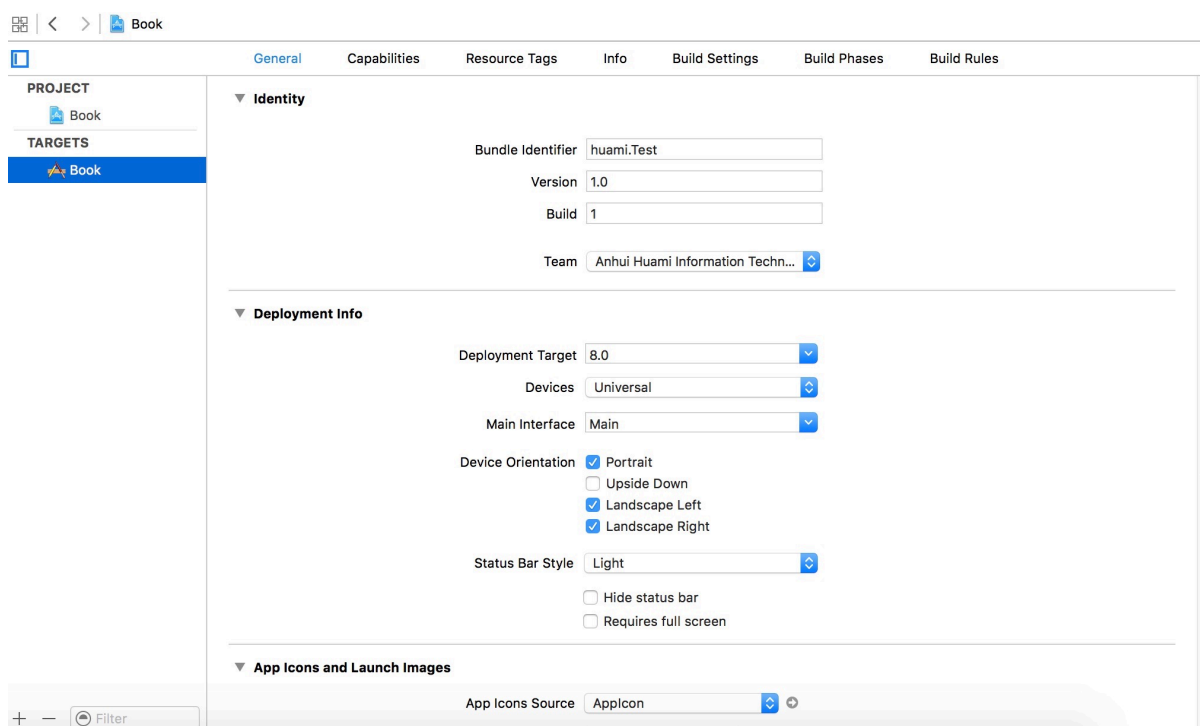


Figure 2. Project properties settings

图 2. 项目属性的设置

5.4. 使用 Storyboard 设计界面

Storyboard 的出现使 UI 间的关系流程一目了然，它其实是 xib 的升级版，将多个 xib 统一管理了。任何事都有双面性，Storyboard 也有它的缺点。Storyboard 中 UIViewController 太多，找到想要的比较困难。这些缺点在结队开发中就会遇见。如果我们用 xib 文件，这样将 UI 最小化分隔开，将后用 code 将这些小单元连接起来，就可以解决这个问题。如果只用 xib 就展现不出 storyboard 的故事情节能力[4]。

设计 StoryBoard 界面，这是已经完成的部分 StoryBoard，App 每个界面对应 StoryBoard 中一个 Scene，除了基本的 Scene 之外，也需要一些额外的 Scene 来支持界面的导航关系，如 UITabBarController Scene，UINavigationController Scene，Scene 之间的线表示页面间的跳转关系，关系可以分为 Push、Modal、Popover 和 Replace 等[5]。

5.5. 使用 Size Classes, 支持 iPhone 和 iPad 设备

在 iOS8 中, 新增了 Size Classes 特性, 它是对当前所有 iOS 设备尺寸的一个抽象。iOS 尺寸的设备包括: iPhone4-5-6-6plus、iPad、iPad mini、iWatch, 那么如何按照以前那针对种特定设备来编写不同的布局, 是一件麻烦的事。

size classes 菜单是一个九宫格, 我们把屏幕的宽(width)和高(height)分别分成三种情况: (Compact, Regular, Any), 也即紧凑、正常和任意。例如: wCompact/hCompact 则适用于 3.5、4、4.7 英寸的 iPhone 的横屏情形; wAny/hRegular 适用于所有方向是垂直的情形如 iPhone 竖屏、iPad 横屏和竖屏的情形。针对每一种情况, 如果需要的话, 我们可以单独在 storyboard 或 xib 中设置 UIView 的自动布局约束, 甚至某一个 button 是否显示都是能轻松实现[6]。

5.6. 设置 AutoLayout 约束

对于任何一个 UI 组件, 确定了它的(相对于父 view)位置、大小也就确定了它在整个 UI 视图中的展示效果。

Autolayout 是为了解决这些 UI 可视单元或者元素是怎样布局、排列的问题。在过去只有 iPhone4 的时候, 我们可以在代码里将设一个可视单元的位置写死, 这样是没问题的, 但随着 iPhone5、6 的发布, 屏幕尺寸有了越来越多中可能, 未来不排除更多尺寸的 iPhone 发布出来, 这就要求我们的 App 的 UI 元素具有在屏幕尺寸不同的设备上具有一定动态的可调性, 已实现较好的 UI 展示效果。从目前苹果提供的技术来看, 有下、中、上三种实现方法:

下策是, 代码中判断当前设备的尺寸, 对 UI 元素进行手工的调整。

中策是, 通过设置可视单元的 autoresizing 属性, 预设当该 view 所在的环境(父 view)发生变化时它的尺寸和位置应该如何调整。

上策就是结合使用 autolayout 和 sizeclass 对 UI 可视单元的父子关系、兄弟关系进行全方位的调整。

此 App 使用了第三种方法即上策构建 UI。为了适应不同的设备, 采用 autolayout 和 sizeclass 结合的方法, 解决所有的 iOS8 多屏适配问题以及 UI 可视单元或者元素的布局、排列问题而产生的自动布局方法[7]。此种方法调节精度更高: 不仅能确定一个 view 的位置尺寸的变化依据, 还能对这些依据加以不同的优先级(先满足什么条件, 再满足什么条件, 对于重要的位置尺寸可以优先保证), 这样整个 App 就具有极强的动态可调性, 可满足不同设备、不同应用场景下的需求。

Autolayout 的作用非常明确: 帮我们确定在不同设备、不同(父 view)环境下, 同一个可视单元所应具有合适的位置和尺寸, 因此, 当一个 UIView 上所施加的约束能够唯一确定它的 frame(x, y, width, height) 的时候我们的自动布局的使用才是正确的。而新手通常犯的两类错误就是约束不足和约束冲突两种。如果你给出的约束只能够确定这个 view 的大小, 或者位置中的某一个项(比如 x)的时候, 就会出现约束不足的情况, 在 xib 或者 storyboard 中, 会以黄色的警告出现在左侧提示框内; 如果你给出的约束推导出了两个甚至多个互相矛盾的位置尺寸结果的时候, 就产生了布局错误, 在编译的时候直接就 build 不过。

6. 软件测试

6.1. 主界面

按照上面的包的层次进行相应的代码编写, 整个项目采用 MVC 的分层架构设计[8], Model 是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。本项目中设计的模型主要有 User, Book, Review 等。

View 是应用程序中处理数据显示的部分, 通常视图是依据模型数据创建的。存放视图, Xib, Cell,

Storyboard。Controller 是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。因为我们这个是网络应用，所以需要先去下载网络的图书数据，使用 AFNetworking 这个 iOS 最流行的网络请求库，并且使用 Get 方法去请求数据。当请求到数据后，加入数组并刷新 UITableView 的数据源，然后会执行 tableView 的数据源和代理方法，来显示数据[5]。如下图 3 所示。

当选中列表顶部的搜索框的时候可以填写搜索条件，回车确认后就从服务器下载搜索结果显示在一个列表中，效果如下图 4 所示。

6.2. 子界面

当点击图书列表的某一项的时候，触发 UITableView 的代理方法来打开图书详情页面，在详情页面也是请求数据并展示在 UITableView 上[9]。如下图 5。

如果需要使用用户相关功能如收藏和评论，需要先登录，当登录成功打开用户信息页，如下图 6 所示。



Figure 3. Main interface

图 3. 主界面

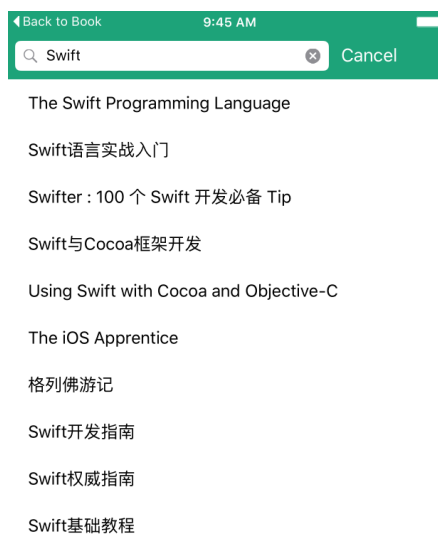


Figure 4. The search interface of the main nterface

图 4. 主界面的搜索界面



Figure 5. Sub interface

图 5. 子界面



Figure 6. User login interface

图 6. 用户登陆界面

7. 总结

传统的开放式社交平台如 Twitter、微博等已经逐渐吸引不了更多的用户，越来越多的用户，开始关注隐私或者转向别的社交形式[10]。通过此 App，用户可以查看和搜索各种图书的信息，自由发表有关书籍的评论，收藏喜欢的图书。当用户为看哪本书拿不定主意时，可以打开此 App，查评分看评论。

推广后，通过该 App 还可以分享音乐，电影，旅游，工作生活经验等方方面面，更加方便快捷的与别人交流，并交到志同道合的朋友。此 App 让用户更大胆自信的展现自我，丰富朋友圈，让生活更加多姿多彩。该 App 主题是崇尚自由，任何人下载此软件都可以发表自己的文字和想说的话，随着时间的流逝大家只会更加喜欢此软件。

基金项目

安徽大学学生创新创业训练计划(201510357242)，安徽大学教育质量提升计划项目(xjyxm14048)。

参考文献 (References)

- [1] 段学东, 何九周. 基于敏捷思想的 iOS 平台软件的设计与实现[J]. 计算机技术与发展, 2012, 22(9): 53-58.
- [2] XCode [EB/OL]. <http://en.wikipedia.org/wiki/XCode>, 2012-07-29.
- [3] 百度文档. http://wenku.it168.com/d_000229590.shtml
- [4] 王巍. Swifter: 100 个 swift 开发必备 Tip [M]. 北京: 电子工业出版社, 2015.
- [5] 龙马. Swift 从入门到精通[M]. 北京: 人民邮电出版社, 2015.
- [6] 关东升. iOS 开发指南从零基础到 App Store 上架[M]. 北京: 人民邮电出版社, 2013.
- [7] 李岚, 李可嘉. iOS Autolayout 中的约束[J]. 科技创新与应用, 2015(10): 72-73.
- [8] 李春红, 高建华. 使用分层模型改进 MVC 设计架构[J]. 计算机工程与设计, 2007, 28(4): 766-769.
- [9] 朱元波, 陈小玉. iOS 8 案例开发大全[M]. 北京: 清华大学出版社, 2015.
- [10] 张利, 王欢. 我国移动社交网络服务的发展现状与方向研究[J]. 现代情报, 2012, 32(11): 68-77.