**北京师范大学珠海分校**

本科生毕业论文

|  |  |
| --- | --- |
| **论文题目：** | **基于手机的汽车配件在线商城设计与实现 ——iOS客户端** |

|  |  |  |
| --- | --- | --- |
| 学 院 |  | 信息技术学院 |
| 专 业 |  | 软件工程 |
| 学 号 |  | 1301030090 |
| 学 生 姓 名 |  | 覃红 |
| 指导教师姓名 |  | 林雪海 |
| 指导教师单位 |  | 信息技术学院 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **2017** | **年** | **4** | **月** | **20** | **日** |

**北京师范大学珠海分校学位论文写作声明和使用授权说明**

**学位论文写作声明**

本人郑重声明： 所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名： 日期： 年 月 日

**学位论文使用授权说明**

本人完全了解北京师范大学珠海分校关于收集、保存、使用学位论文的规定，即：按照学校要求提交学位论文的印刷本和电子版本；学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务；学校可以采用影印、缩印、数字化或其它复制手段保存论文；在不以赢利为目的的前提下，学校可以将学位论文编入有关数据库,提供网上服务。（保密论文在解密后遵守此规定）

论文作者签名： 导师签名：

日期： 年 月 日

摘 要

本论文通过研究国内外主流电子商务平台的特点，结合了互联网O2O模式的流行趋势，把传统的汽车配件售卖、汽车美容、零件装配等汽配服务整合到了一起，设计了本款iOS客户端，App的主要的功能为：用户注册登录后，可以在软件内查找、收藏、购买汽车配件，并支持选取指定订单直接在地图界面预约当地的维修点。用户操作起来流畅，方便。

该客户端使用了苹果公司（Apple Inc.）最新发布的编程语言Swift完成编写，它结合了众多现代编程语言的特性，高效强大。本项目同时也使用Xcode提供的Swift与Objective-C混编的技术，使得仍可在现有环境调用Objective-C的库。

**关键词：**Swift，O2O，电子商务，汽配服务，iOS

**ABSTRACT**

This paper analyzes the characteristics of mainstream e-commerce platform at home and abroad, combined with the popular trend of Internet O2O model, the traditional auto parts sales, auto beauty, parts assembly and other auto parts service together to design the iOS client, App The main function is: user registration log, you can find in the software, collection, purchase auto parts, and support the selection of orders directly in the map interface booking local maintenance points. Users operate smoothly and convenient

The client uses Apple Inc.'s latest release of the programming language Swift to complete the preparation, which combines the many modern programming language features, efficient and powerful. The project also uses Xcode's Swift and Objective-C hybrid technology to make it possible to call Objective-C libraries in existing environments.

**Key words:** Swift;O2O;E-business; Auto parts;iOS.

**目 录**

1. 绪论 1

1.1课题研究的背景和现状 1

1.2课题研究的目的和意义 1

1.3本人主要的研究工作 1

1.4论文组织架构 2

2. 课题涉及的相关技术 3

2.1 iOS技术 3

2.1.1 iOS系统介绍 3

2.1.2 Swift与 Xcode简介 3

2.2 Cocoapods 3

2.2.1 Cocoapods简介 3

2.2.2本项目使用到的第三方库 4

2.3 课程设计的指导思想 4

2.3.1软件工程思想 4

2.3.2面向对象思想 4

2.3.3 iOS人机交互规范 4

2.4本章小结 5

3. 系统分析 6

3.1项目规划 6

3.2需求分析 6

3.2.1功能性需求 6

3.2.2非功能性需求 7

3.3开发环境 7

3.3.1硬件环境 7

3.3.2软件环境 7

3.3.3运行需求 7

3.4本章小结 8

4. 客户端的详细设计与实现 9

4.1应用整体设计 9

4.2购物车及结算模块 12

4.3个人中心模块 15

4.4查找配件及产品详情页模块 17

4.5选择车型及添加爱车 19

4.6订单模块 20

4.7预约维修店模块 22

4.8应用改进 24

4.9 本章小结 24

5. 结语 25

5.1 课题工作总结 25

5.2 问题与展望 25

参考文献 26

致谢 27

**图目录**

图 1本人负责的功能点 2

图 2爱车在线功能结构图 9

图 3软件启动流程图 9

图 4项目组织结构图 10

图 5预约维修店视图 11

图 6软件首屏 12

图 7购物车模块用例图 13

图 8跳转前传递信息 15

图 9个人中心用例图 15

图 10个人中心界面 16

图 11查找配件用例图 17

图 12查找配件与商品详情页 18

图 13爱车中心用例图 19

图 14选择车型和爱车列表 20

图 15订单模块用例图 20

图 16订单模块运行图 22

图 17预约维修用例图 22

图 18预约维修店 23

# 绪论

## 1.1课题研究的背景和现状

当前，全球电子商务发展正在进入高速发展和扩张的新阶段，同时伴随着我国国民经济的快速发展及社会信息化的进步，2016年上半年，中国电子商务交易额达10.5万亿元，同比增长37.6%，增幅上升7.2%，电子商务发展拉动着我国的消费需求，促进了新兴行业或是传统行业的升级。汽车配件行业还没有很快的适应当今潮流，在电子商务领域上发展缓慢。特别的，手机购物往往是用户的首选平台，2015年年天猫“双11”创下电子商务领域的盛举，其中68%的交易额都归功于无线端，截至2016年6月，中国网民规模达7.10亿，互联网普及率达到51.7%，而中国的手机网民在上半年达到6.56亿，可见，手机在人们生活中扮演着越来越重要的角色，因为手机的便携性和快速发展的性能，所以人们更乐意在手机浏览内容，购买商品。

## 1.2课题研究的目的和意义

我们想让车主能放心的在网上采购汽车配件，借助互联网的传播性和便捷性来售卖配件产品，在保证配件质量的前提下同时提供优质的配件拆装及售后服务。

本项目基于这种需求，设计一款基于iOS平台的客户端 App。整合汽车零配件、维修店及汽车美容店，通过O2O(线上到线下)模式，实现车主线上自主选购配件，并通过地图功能查找及预约指定的线下实体店去拆装维修。使用过程简单直观，人性化，适合大多数想要DIY汽车配件的客户，给他们带来便利。

## 1.3本人主要的研究工作

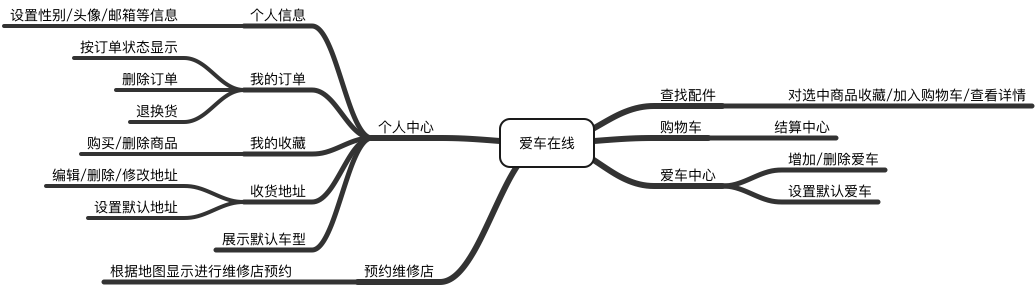
本人在这个项目中主要负责该App核心功能模块框架的设计及代码编写，界面设计。

图 1本人负责的功能点

以上内容包括了购物车、个人中心、订单列表、预约维修等功能模块的设计与编写，涉及到了网络请求、图片处理等功能点。持续的进行测试，完善优化已有的功能模块。

如图1所示，以下是本人负责的所有功能开发。

## 1.4论文组织架构

第一章论述了该课题的现状和发展意义以及本人的主要工作。

第二章介绍了项目中使用到的相应技术和一些软件设计思想。

第三章进行了项目的系统分析，需求分析以及介绍了开发环境。

第四章详细的介绍了主要的功能模块的设计与实现和应用的改进建议。

第五章对该项目进行了总结与展望。

# 课题涉及的相关技术

## 2.1 iOS技术

### 2.1.1 iOS系统介绍

iOS是苹果公司为移动设备所开发的专有移动操作系统，它所支持的设备包括 iPhone、iPod touch 和 iPad。与谷歌公司的 Android 不同，iOS 不支持任何非苹果的硬件设备。iOS的简洁流畅稳定是人们偏好它的原因，iOS中的App Store下载量已经达到了 1400 亿次，这奠定了越来越多的开发者将投身之中。

本项目中软件就是基本iOS平台开发的，且使用了众多iOS特性，例如常用的UITableView, UIView, StoryBoard。

### 2.1.2 Swift与 Xcode简介

Swift是一种支持多编程范式和[编译](https://zh.wikipedia.org/wiki/%E7%BC%96%E8%AF%91" \o "编译)式的编程语言，是用来撰写macOS/OS X、iOS、watch OS和tvOS的语言之一，其外是Objective-C。 2014年，其在[苹果开发者年会](https://zh.wikipedia.org/wiki/WWDC)（WWDC）发布。Swift取消了Objective-C的指针及其他不安全访问的使用。同许多脚本语言一样，Swift可以推断变量类型。同时，它提供了类似C++、C#的名字空间（namespace）、泛型（generic）、运算对象重载（operator overloading）。Swift被形容为 “没有C的Objective-C”。

Xcode是苹果公司向开发人员提供的集成开发环境，用于开发macOS、iOS、WatchOS和tvOS的应用程序。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。

## 2.2 Cocoapods

### 2.2.1 Cocoapods简介

CocoaPods项目的源码 在 Github 上管理。该项目开始于 2011 年 8 月 12 日，经过多年发展，现在已经成为 iOS 开发事实上的依赖管理标准工具。开发 iOS 项目不可避免地要使用第三方开源库，CocoaPods 的出现使得本人可以节省设置和更新第三方开源库的时间。

本App使用了CocoaPods来统一管理部分第三方库，所以可以方便的的添加、修改、更新、删除。大大缩短了配置时间。本人只需要将用到的第三方开源库放到一个名为 “Podfile” 的文件中，执行一行代码CocoaPods 就会自动将这些第三方开源库的源码下载下来，并且为本人的工程设置好相应的系统依赖和编译参数。

### 2.2.2本项目使用到的第三方库

项目中使用的第三方库：JSONNeverDie、AlipaySDK、BaiduMap\_IOSSDK、Charts、fmdb、IQKeyboardManager、Alamofire、Kingfisher、NVActivityIndicatorView、WeChatSDK。

以上库的主要功能包括了：图片下载、网络请求、地图、微信登录、JSON解析、进度指示器、支付，键盘管理等。

## 2.3 课程设计的指导思想

### 2.3.1软件工程思想

以系统性的、规范化的、可定量的过程化方法去开发和维护软件，以及如何把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来。

### 2.3.2面向对象思想

面向对象是种具有对象概念的程序编程范型，同时也是一种程序开发的抽象方针。它可能包含数据、属性、代码与方法。对象则指的是类的实例。它将对象作为程序的基本单元，将程序和数据封装其中，以提高软件的重用性、灵活性和扩展性，对象里的程序可以访问及经常修改对象相关连的数据。

本项目中使用的Swift，是一门新兴的重要的面向对象编程语言。

### 2.3.3 iOS人机交互规范

本项目遵守着苹果官方发布的《iOS人机交互指南》，软件正确使用了模态视图和子控制器视图，用户知道自己是从哪个界面来，也知道怎么回到上个界面，大量界面支持手指右滑返回，交互直观自然。

## 2.4本章小结

本章着重对项目中使用到的技术作了详细的介绍，也同时介绍了项目中使用到的第三方库，它们都是开发此应用不可或缺的东西。

# 系统分析

## 3.1项目规划

本项目的用户大多是有车人士，且已经有了智能手机，他们都有维护自己爱车的经验，但是作为这个互联网时代的消费者，他们自己购买配件和去线下寻找维修点维护汽车的成本都太高了。这也是这个项目诞生的原因，软件用清晰的分类在线上展示配件，用户可以点击购买，也可以收藏，当用户购买配件之后，本应用会在地图上标记附近所有的维修店详情，用户自行选择预约线下的维修店并提交订单信息给商家，最终完成了爱车的维修与保养，无缝连接了用户从挑选配件到更换配件的整个过程。

iOS目前是所有优秀软件都必须要登陆的平台，该项目毫无犹豫的需要在iOS 端的开发。

## 3.2需求分析

### 3.2.1功能性需求

1) 附近维修店，可以根据用户的地理位置在地图上显示附近的维修店。

2) 查找配件，可以根据用户选择的车型、关键字为用户筛选出合适的配件。

3) 我的爱车，用户可以自行添加不同的车型，也可设置默认车型。

4) 我的订单，显示用户的订单列表，包括可以按待发货、待支付等条件筛选订单。

5) 我的收藏，用户可以收藏商品，然后在此界面统一管理收藏，可直接购买。

6) 预约维修店，用户可以在附近的维修店列表直接预约，并发送给店家订单信息。

7) 退换货管理，用户可以操作退货。

8) 购物车，用户可将商品添加在购物车中，来批量购买商品。

9) 注册与登录，手机号可注册，用户设置好密码后可在之后登陆。

10) 个人中心，包含了订单、收藏、收货地址等入口，也是个人详情也在此展示。

11) 商品详情页，包括了商品的价格信息、收藏/购买按钮，商品的详情介绍。

12) 确认订单，用户可以自定义支付方式，收货地址，也可在此页面查看购买的商品列表。

13) 个人详情，用户可自定义头像，性别，邮箱等信息。

### 3.2.2非功能性需求

1) 系统的性能，需要同时满足多人使用。

2) 可靠性，用户在公测系统上使用保证软件不会Crash。

3) 可维护性，方便新加入的伙伴持续的维护该App。

4) 兼容性，保证兼容近两年的iOS系统。

## 3.3开发环境

### 3.3.1硬件环境

开发机型：Macbook Air mid 2013, 1.3 GHz Intel Core i5, 4 GB 1600 MHz DDR3

测试机型：iPhone 7，且本论文所有截图均来自iPhone 7。

### 3.3.2软件环境

|  |  |  |
| --- | --- | --- |
|  | 软件名称 | 版本 |
| 电脑系统 | macOS | 10.12 |
| 测试系统 | iOS | 10.2.1 |
| 开发工具 | Xcode | 8.2.1 |
| 版本控制器 | Cornerstone | 2.7.10 |
| 设计工具 | Sketch | 40.3 |
| MindNode | 2.4.2 |
| OmniGraffle | 7.0.2 |

表 1软件环境

### 3.3.3运行需求

软件支持在iOS 8 以上的系统运行，已测试iOS 10.3 Public beta 系统下运行正常。

屏幕仅适配了4.7英寸，5.5英寸的iPhone。iPad和3.5英寸设备暂未测试。

## 3.4本章小结

本章对项该项目做了系统分析，包括了对整个项目的规划，需求分析，以及对运行环境的要求。

# 客户端的详细设计与实现

## 4.1应用整体设计



图 2爱车在线功能结构图

本项目的整体功能结构图如图2所示，以上包括了个人中心、查找配件、我的爱车、购物车、订单管理、附近维修店、预约维修店、地址管理等功能点。



图 3软件启动流程图

用户打开并正常使用该App的流程设计如图3所示，除非用户点击退出登录，软件会在本地储存用户的登录信息。



图 4项目组织结构图

本项目基于Xcode编写，项目的组织结构图如图4所示。

项目使用CocoaPods来统一管理第三方的库，但是有些不支持CocoaPods的库可以选择直接将整个项目置入到该项目中，如微信登录的SDK，此时根据第三方库的正常运营需求在Xcode手动添加必要的Frameworks，值得值得注意的是，若第三方库使用的Objective-C编写，则需要在本项目建立桥接文件，使得Swift项目也能够调用Objective-C的接口，实现Objective-C与Swift混编。该桥接文件储存在Others文件夹。

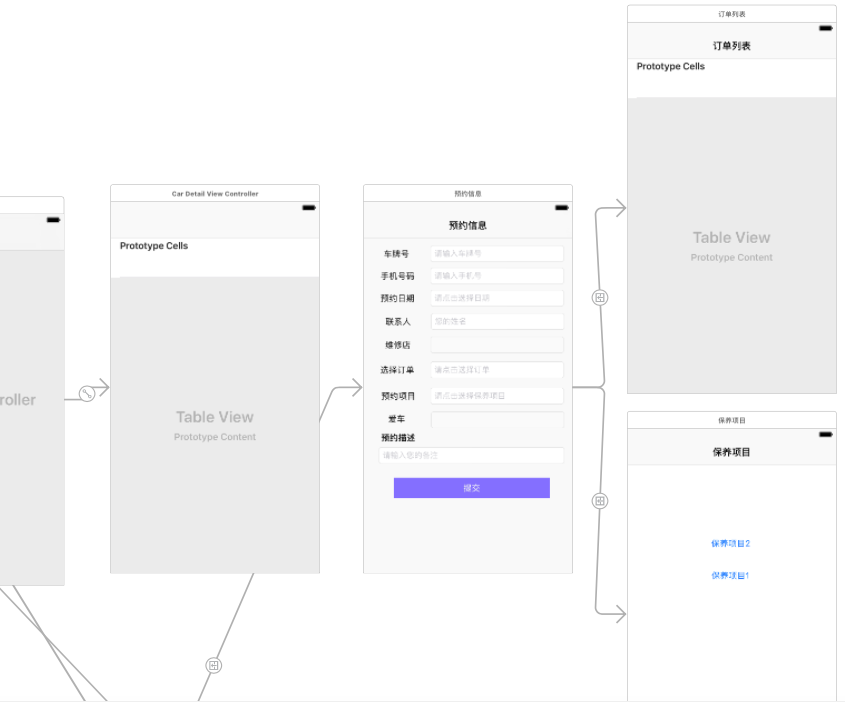


图 5预约维修店视图

本项目使用了模型 - 视图 - 控制器架构。Model文件夹则储存了各个数据模型，如从服务器获取的车型数据、维修店数据模型、产品详情数据模型等。视图控制器文件夹包括了项目中所有的视图控制器，视图则在故事版里面，项目高度依赖故事版，绝大多数的软件视图都在故事版里，通过Xcode的Size Class技术可实时模拟真机尺寸检查屏幕适配情况。当然，每个故事版中的界面都绑定着相应的ViewController，ViewController即视图的控制器，负责操作视图的表现。本项目的故事版部分截图如图5所示。

图4中的尾部有一个Pods的项目，是因为CocoaPods会将所有的第三方库以target的方式组成一个名为Pods的工程，每个库都会对应一个Target，Tgarget即是一个包含了相应的文件信息资源仓库/积木，可以由开发者自由调用。一个项目可以包含多个TARGET的。整个第三方库工程会生成一个名称为libPods.a的静态库,然后COCOAPODS会把PODS工程和你自己的工程生成到一个工作空间方便管理。

本项目使用USERFAULTS（键值对）来在本地储存简单的常用数据，如用户的登录信息。



图 6软件首屏

进入App后首页的4个Tab界面如上所示。

以上对项目的文件结构和整体界面及功能设计进行了介绍，下面是软件核心模块的设计与实现，将会有对重要代码的说明与解释。

## 4.2购物车及结算模块

目前购物车仅有一个入口即可在主页点击“查找配件”后，购物车图标则显示在右上角。用户在“查找配件”的页面可以方便的添加配件，然后可以直接点击购物车图标准备结算。

购物车支持用户多选，即可以多选删除，多选付款，通知支持更改单个商品的数量，以上操作用户均可在界面底部实时查看价格更新。选择好需要的商品点击下一步即进入结算界面。

结算界面用户可定义的是收货地址、配送方式、支付方式、提货方式、发票信息、优惠券使用，用户也可以在此界面查看商品清单，并可在页面顶部查看系统根据用户选择的地址的预估送达时间。当用户点击提交按钮时会调用支付宝进行支付操作。

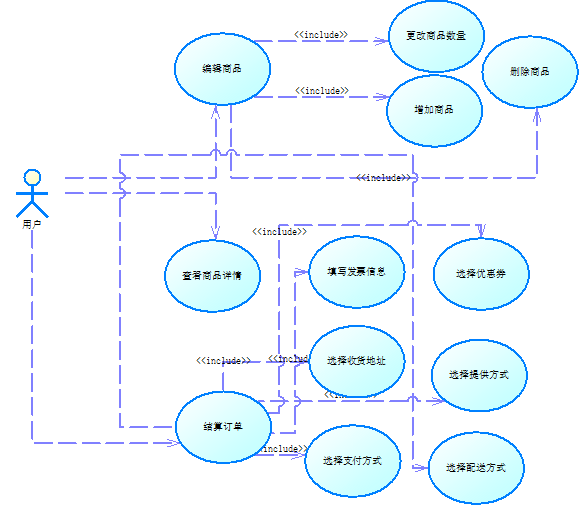


图 7购物车模块用例图

该模块的用例图如图7所示。

当软件从服务器获取数据后，通过UITableView的接口可以快速配置单元格的显示内容，即控制购物车中商品列表中的显示方式。

|  |
| --- |
| //配置要传入的参数  let parameters = ["UserID":userID,"PageNO":"1","IsShopper":0]  //获取网络数据  Alamofire.request(.POST, getShoppintCarURL , parameters:parameters as? [String : AnyObject])                  .response { request ,response ,data , eror in  //解析获得的数据为JSON    let json = JSONND.initWithData(data!)    let result = json.arrayValue    //解析特定的数据并储存到本地  for i in 0..<result.count {  ……    }    //若购物车中没有数据    if result.count == 0 {      ……     }    //刷新数据    self.tableView.reloadData()  } |

上述代码表示了使用Alamofire库来获取服务端的数据，事实上本项目所有的网络请求均调用了此库，它是iOS开发中热门的网络基础框架。当从服务器获取到了源数据以后，本人使用JSONNeverDie库来解析此数据，然后存储在相应的数据模型，供UITableView的接口使用，最后展示出图形界面。

|  |
| --- |
| //tableView中监听被选中的行的系统函数  override func tableView(tableView: UITableView, didSelectRowAtIndexPath indexPath: NSIndexPath) {  //根据indexPath.section识别不同行，程序作出不同反应    switch indexPath.section {    case 1:        let alterV = UIAlertView(title: "", message: "请选择支付方式", delegate: self, cancelButtonTitle: "取消", otherButtonTitles: "支付宝","易付宝")          alterV.tag = 22          alterV.show()             ……     }  } |

在结算界面，为了让用户可以自定义收货地址、查看商品清单等，所以在故事板中结算视图使用Segue技术直接“连线”到即将跳转的视图，同样的，这也是iOS开发中常用的方式。

|  |
| --- |
| //转跳时传递相应数据，根据不同identifier作出相应动作,此处为跳转到不同的界面      override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {          //identifier在故事版中已定义好          if segue.identifier == "MakeOrderToAddress" {  ……          }          else if segue.identifier == "MakeOrderToItsGoodList" {  ……          }      } |

如上所示，当程序执行performSegueWithIdentifier函数时，通过覆盖prepareForSegue接口来实现提前把信息传递至下一个界面。而识别符即在故事板中“连线”属性中的“segue.identifier”。



图 8跳转前传递信息

此模块运行界面如图8所示

## 4.3个人中心模块

个人中心作为的软件的四大首页之一，包含了非常多的核心功能。用户可以在启动软件后的底部直接点击切换到个人中心。这里包含了多个功能的入口，包括我的订单、个人资料、我的收藏、收货地址。同时也显示了用户的默认车型，在未来软件迭代也会上线积分、余额的展示。界面的上方使用了清新明亮的色块展示用户资料。

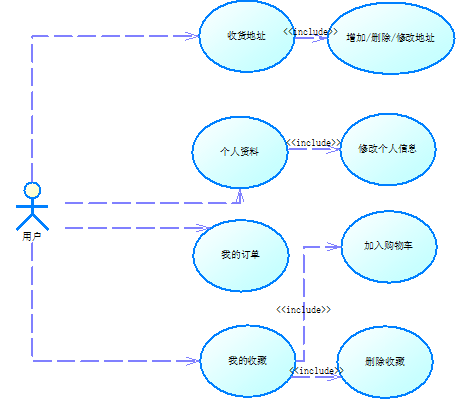


图 9个人中心用例图

该模块的用例图如图9所示。

同样的，本人使用Segue技术，在在跳转界面前使用prepareForSegue方法来传递数据。根据每条Segue线的标识符不同，来往不同的界面传递数据。即提前给准备跳转下一个页面中的变量赋值。

本项目大量使用了UITableViewController，这可以方便快速的进行精美的布局，类似于Android开发中的listView。本项目高度自定义了该控制器中的接口，每一行数据iOS中称为Cell，实质是一个View，TableView的核心就是复用Cell，适用于每行数据类型和展示方式一致，只是数据不同的情况。

覆盖此方法就能进行每行Cell内容的定义，这也是iOS开发中主流的方式：

|  |
| --- |
| override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell{  //自定义Cell  return cell  } |



图 10个人中心界面

界面展示如图10。

## 4.4查找配件及产品详情页模块

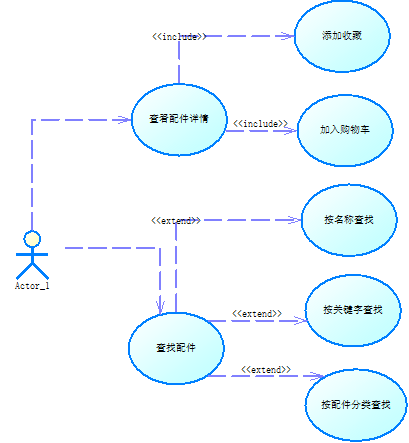


图 11查找配件用例图

查找配件是本项目一个重要的功能，在软件首屏。用户可以直接根据配件名称或配件编号来检索，也能通过选择配件分类的筛选的方式来浏览配件。软件按条件筛选出相关的配件列表供用户挑选时，用户可以直接在右边的按钮添加到购物车，当然，用户也可以点击，进入商品的详情页再决定是否购买。在产品详情页在代码的优化上，可以非常快速的弹出界面展示详情。该模块的用例图如图11所示。

|  |
| --- |
| //加入购物车按钮的监听函数  func AddCarbtnClicked (btn:UIButton) {          //配置参数          let parameters = ["UserID":userID, "ProductID":"[\(goodID)]", "Quantity": "[\(quantity)]"]          //向服务器提交商品数据          Alamofire.request(.POST, addShoppingCarURL, parameters:parameters as? [String : AnyObject])              .response { (request, response, data, error) in          }          //提交完成后安卓总Toast的提示          self.navigationController?.view.makeToast("成功加入购物车！")   } |

上文提到，这是自定义Cell的函数里面，本人向每行单元格的右侧添加一个按钮，即“加入购物车”。并为它指定Target到AddCarbtnClicked方法里面去，用Java开发中的话语则意为在AddCarbtnClicked方法里面监听这个button的点击事件，如上代码所示。

在商品详情页里面，除了常规的使用Alamofire来获取数据外，本项目巧妙的使用NSTimer来刷新TableView的数据，使得一旦网络传回数据，TableView立即reloadDate，这大大减少了“卡顿”的现象。代码如下:

|  |
| --- |
| \_ = NSTimer.scheduledTimerWithTimeInterval(0.1, target: self, selector: #selector(GoodDetailTableViewController.iii), userInfo: nil, repeats: false)  //意味上述计时器会每隔0.1s调用下面函数一次，   func iii() {  //刷新表格数据          self.tableView.reloadData()  } |

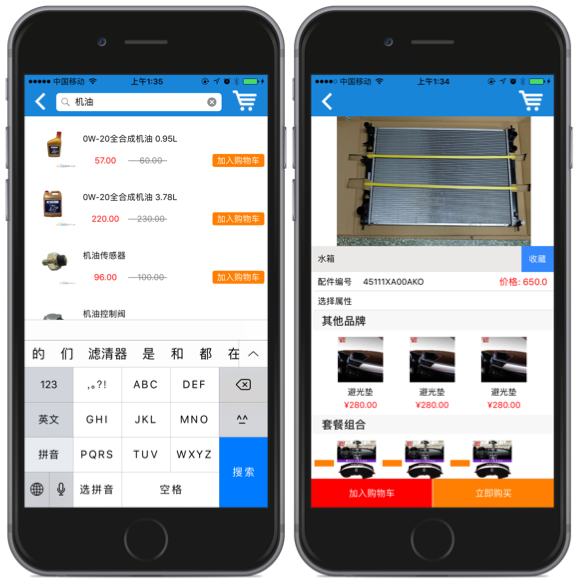


图 12查找配件与商品详情页

运行截图如上图12。

## 4.5选择车型及添加爱车

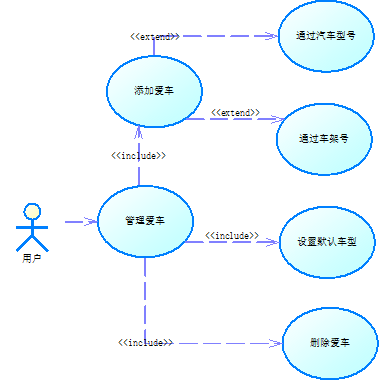


图 13爱车中心用例图

每个用户都需要设置好车型以后软件就能提供相应的服务。用户添加爱车的方式有两种，一是通过车架号，车架号是制造厂为了识别而给一辆车指定的一组字码；二是通过选择车型来添加。爱车页面可在进入首页后直接在底部进入，用户可以看到添加的所有爱车，并且可以删除、设置默认车型，当然这也是使用了UITableViewController实现的，车辆列表也是通过自定义Cell实现的。该模块用例图如图13所示。

|  |
| --- |
| //将TextField的输入源设置为PickerView      func initTextField(){          //为组件设置委托和数据源          brandPick.delegate = self          brandPick.dataSource = self          brandTextField.inputView = brandPick  //为TextField结束编辑的时候增加事件方法，即实现委托的方法          brandTextField.addTarget(self, action: #selector(ChooseCarViewController.brandsSelect), forControlEvents: UIControlEvents.EditingDidEnd) |

特别的，当用户按车型来添加汽车时，录入数据使用的TextField组件，每当用户结束这次输入，则可做出相应动作，在本项目的应用则是每当用户选择一个条件，则下一个可选条件会随上一个条件而改变。如上代码所示。

委托和数据源，这类似于Java的接口，使TextField等组件有了新的能力。这里的TextField会在用户每次结束编辑时触发相应函数，可以通过覆盖委托方法就可以调用系统级的函数。

当用户输入完成后，监听函数则把下一个文本框改为可用（enable=true）。

在爱车界面也是通过Alamofire像服务器获取数据后，存储到数据模型后供TableVeiw的Cell使用，从而显示出每辆车的信息。



图 14选择车型和爱车列表

运行界面如图14所示。

## 4.6订单模块

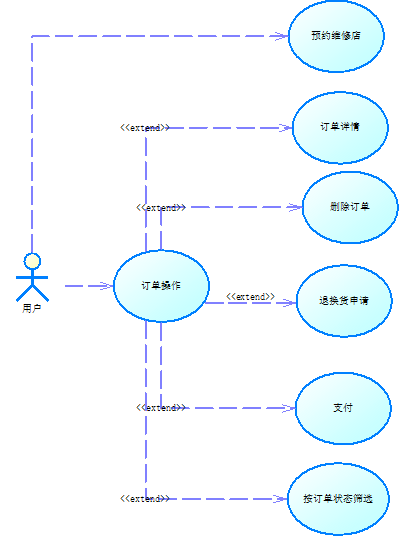


图 15订单模块用例图

订单模块存在个人中心，用户点击我的订单则可以看到全部的订单，用户也可手动的在“待付款”“待发货”“已发货”等Tabs查看对应的订单状态下的所有订单。用户仍可以继续支付在结算界面未支付的订单，也可以选择退换货、查看物流信息。为了更加人性化，每个订单均可以点击跳转到订单详情，订单详情包括了所购买的商品、订单状态、总价、订单信息和收货信息，甚至仍可以点击商品小图跳转到商品详情。本模块用例图如图15。

|  |
| --- |
| //跳转前传递数据      override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {          //根据identifier判断不同动作          if segue.identifier == "orderToDetail" {              //获取即将跳转页面的对象，传递数据              let receive = segue.destinationViewController as! orderDetailViewController              receive.data = data[tempIndex!.section]              receive.price = "￥" + "\(priceArr[(tempIndex?.section)!])"          }          if segue.identifier == "MyOrderToReturnedPurchase" {            ……          }      } |

本项目大多数跳转方式一样，在订单详情页可以使用手势右滑反馈到所有订单，因为它们都是同一个导航控制器。用户也可选择删除订单。同样的，本项目也是用Alamofire库来获取数据，然后使用TableView来展示订单。相关代码如上所示，这是点击每个订单中任意商品跳转至商品详情页的核心代码。这里覆盖了UITableView的委托方法，即didSelectRowAtIndexPath，意为监听每行数据被点击的事件，indexPath是iOS系统反馈来的行位置，可监听具体点到哪一个Cell，这里没有作此判断，因为需要点击任意商品都跳转至商品详情，同样的，在跳转前把相应数据传递过去，这样订单详情页就能根据不同的订单显示不同的数据。



图 16订单模块运行图

运行截图如上。

## 4.7预约维修店模块

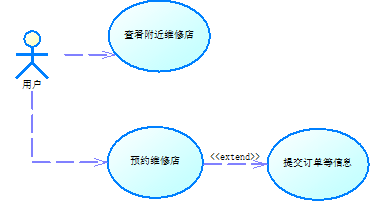


图 17预约维修用例图

这是本项目最特别的模块，用户可在首页和订单界面点击“维修店”或“预约维修店”进入。用户可以直接在地图界面查看附近的所有维修店，点击图钉，会弹出对应维修店的详情并能进入预约界面，预约界面需要用户填写手机号码、预约日期、车牌号等信息，其中维修店和爱车（默认车型）不能修改，让用户点击选择订单和预约项目的输入框会跳转到下一个界面，用户在此界面选择某个订单/某个保养项目，选择的数据返回至上一个界面，点击提交即完成了预约流程。用例图如上。

|  |
| --- |
| //注册通知 for order          NSNotificationCenter.defaultCenter().addObserver(self, selector: #selector(bookStoreViewController.disPlayMsg(\_:)), name:"NotificationIdentifier", object: nil)    //收到通知后触发      func disPlayMsg(notification:NSNotification){          chooseOrder.text = "订单号 \(notification.object as! String)"      } |

本模块传递数据的方式教之前不一样，通过“通知”来传递。即运用了观察者模式。

首先，在预约界面注册通知，并实现该界面收到通知后的函数，此处为用户在填写预约界面时点击选择订单和点击选择维修项目时的通知，如上代码。此处的响应函数disPlayMSG的作用是得到下个界面的传回的值，然后在预约界面里显示出来。

|  |
| --- |
| //使用通知将订单号呈现至上一次页面          NSNotificationCenter.defaultCenter().postNotificationName("NotificationIdentifier", object: data[section][0][0][0])          self.navigationController?.popViewControllerAnimated(true) |

上面描述下一个界面怎么利用通知传回去的。如上，在订单选择界面，当用户点击订单下面的按钮，则程序会将该订单的订单号传回去让预约信息界面来显示。保养项目也是如此。此页面的代码原理和订单模块的原理相似。

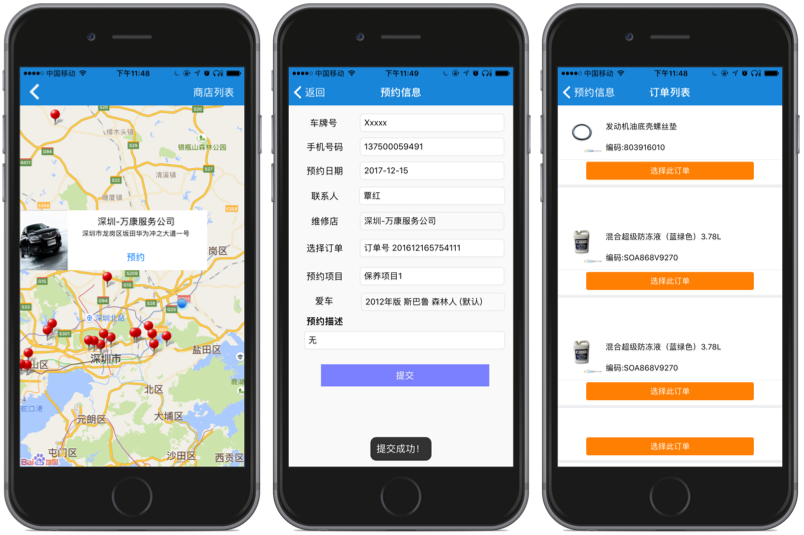


图 18预约维修店

值得一提的是，本模块使用时百度地图提供的SDK，配置方式非常简单，使用前文提及的CocoaPods，可以在控制台一行代码将此地图SDK添加至项目中并自动配置，然后参考文档直接可上手。运行截图如上所示。

## 4.8应用改进

适配新版本的Swift。本项目已经适配了iOS 10，但仍使用Swift 2.3编写，Swift的开源加上Apple的影响力，使得Swift语言得以高速发展，现已更新至3.0版本，项目需要面对的问题也在于此，开发者需尽快学习Swift的新特性，迎接每一次更新。

充分利用新技术。应该充分的利用iOS的新技术。Apple 在iOS 9为iPhone带来了3D-Touch技术，在iOS 10中通知中心增加了Widget，所以该项目应该适配此技术，然用户可以利用新iPhone的压感技术在桌面重按爱车在线icon直接进入常用功能界面，也可以在下拉通知中心直接获取新配件的上新推荐。

用户目前在没有登录的情况下是无法操作的。未来需要开放权限，让即使未登录的用户也能浏览商品，否则是不合理的，强制让用户注册的时候，也许会使用户不得不关掉软件。

软件还没有充分的测试，例如在弱网/无网环境的提示是否友好，绝对杜绝软件Crash的情况，用户体验非常不好。

## 4.9 本章小结

本章详细介绍了项目核心模块的功能、实现方法、关键代码、涉及到的iOS技术，每个模块都有最后的运行图。在最后对于软件还未实现的功能做出了总结。

# 结语

## 课题工作总结

本软件非常注重用户的体验，如用户在登录界面时，对账户和密码字符的输入状态监控得非常人性化，如当用户没有输入账户密码时，“登录按钮”是灰色不可点击的，类似的考虑到了多种场景来让一个按钮该展示时才展示，不干扰用户同时也起到了指导作用。此外，登录的进度动画的持续时间经过了多次修改，只为让该加载动画看起来顺滑不拖沓。

在设计软件的过程中参考了非常多同类的优秀软件，比如个人资料页的展示方式，参考了大名鼎鼎的微信，并调整至适合本软件，且点击每行资料后跳转的编辑界面是可以复用的，巧妙地根据了数据类型的特点来在同一界面出现前加载不同的样式，节省了系统资源。同时软件开启时并不会加载过多数据，这也成就了本人的项目的启动时间在0.6秒左右（iPhone7运行，计时器手动计时），体验非常好。

项目的业务没有淘宝复杂，所以在项目上线初期各种展示方式争取做到极简，直观。在开发的过程中，除了官方的开发文档，《iOS人机交互指南》也是必须参考的，iOS上的软件就应该符合iOS的设计规范！

最后在项目开发过程中，本人把每一次改动，每一次修复的bug，每一次新增的功能均记录在文档里面，这不但有助于后期的测试和开发人员，也是对自己的工作成果的一个总结。

## 5.2 问题与展望

本程序依然还只是一个完成度高的框架，对于一个电商平台，还有很多的模块没有完善，很多个功能没有测试，加上使用着全新的Swift语言，开发语言迭代也非常快，希望爱车在线iOS端能和Swift一起进步。

我们相信未来的用户是更偏好移动的，而如今正是移动互联网大潮的来临，对于一个传统的企业，在这风口上若转型成功，拥抱互联网，未来的发展应该相当可观。

# 参考文献

[1]贾慕星. 基于iOS的化妆品电子商务APP的设计与实现[D].北京邮电大学,2015

[2]崔维,李晖,刘佳佳,王艳娟. 基于IOS的企业APP客户端的开发与实现[J]. 科技创新与应用,2014,(24):30-31.

[3]王猛.基于iOS平台的企业级电商交易平台客户端的设计与实现[D].北京交通大学,2015.

[4]尚志会,张建伟,马琳琳,蔡增玉,陈彦伟. 基于iOS平台的网购应用客户端的设计与实现[J]. 湖北民族学院学报(自然科学版),2016,02:210-213.

[5] 孙桥. 基于移动设备的O2O电子商务平台的设计与实现[D].大连理工大学,2014

[6]王汉肖. 手机购物方案生成系统的设计与实现[D].山东大学,2015.

[7]段艳艳,赵启升. 基于iOS的O2O商务平台设计与实现[J]. 泰州职业技术学院学报,2014,(03):55-57.

[8]李岚,李可嘉. iOS Autolayout中的约束[J]. 科技创新与应用,2015,(10):80.

[9]Apple Inc. .Using Swift with Cocoa and Objective-C［EB/OL］. <https://itunes.apple.com/hk/book/using-swift-cocoa-objective/id888894773?l=zh&mt=11>, 2014-06-02

# 致谢

非常感谢林学海老师的耐心指导和对本人严格的要求，无论在学习还是生活中，使本人可以在短时间内通过自己的能力得到了许多收获。

也感谢项目组同学们的陪伴，他们都非常厉害，工作期间也非常的认真负责，整个团队的融洽氛围也是完成此次毕业论文必不可少的条件。