

基于 iOS 的二手车信息平台的设计与实现

摘要

关键词：

装

订

线

The Design and Implementation of Second-hand vehicle information platform based on iOS

Abstract

Key words:

装

订

线

目 录

1 综述.....	1
1.1 选题背景.....	1
1.1.1 中国汽车行业发展现状.....	1
1.1.2 中国二手车行业发展现状.....	1
1.2 国内二手车软件平台发展现状.....	2
1.3 本文的主要工作及意义.....	2
1.4 本文的组织结构.....	3
2 iOS 开发理论基础.....	4
2.1 iOS 系统和 Xcode.....	4
2.1.1 iOS 系统简介.....	4
2.1.2 Xcode 简介.....	4
2.2 Objective-C 介绍.....	6
2.3 MVC 模式.....	6
2.4 本章小结.....	7
3 系统需求分析与概要设计.....	8
3.1 系统需求分析.....	8
3.1.1 功能性需求.....	8
3.1.2 非功能性需求.....	9
3.2 系统概要设计.....	10
3.2.1 系统架构设计.....	10
3.2.2 API 设计.....	11
3.2.3 界面设计.....	13
3.3 本章小结.....	17
4 系统详细设计及实现.....	18
4.1 用户登录模块的详细设计及实现.....	18
4.2 二手车列表模块的详细设计与实现.....	19
4.2.1 车源列表部分.....	19
4.2.2 车源详情部分.....	22
4.3 搜索模块的详细设计与实现.....	22
4.4 车辆求购模块的详细设计与实现.....	24
4.5 车辆发布模块的详细设计与实现.....	26
4.5.1 车辆颜色选择的设计与实现.....	26
4.5.2 上传照片的设计与实现.....	27
4.6 信息管理模块的详细设计与实现.....	29

1 综述

1.1 选题背景

1.1.1 中国汽车行业发展现状

中国是一个汽车行业的庞大市场，据统计，2013 年，中国汽车行业销售量突破 2000 万量【1】，超越美国成为汽车行业的第一大市场。由于汽车的销量在几年之内将一直保持高位的状态，汽车行业的高速发展还将持续很长一段时间。对于国内的汽车行业来说，首先，刚性需求依然旺盛。其次，一二线城市和东部沿海地区的需求依然旺盛，三四线城市的发展潜力强劲，西部的需求增长迅速。另外，政府对汽车行业的支持力度不断加大。

然而，与汽车行业相关的社会问题越来越突出，如停车问题，污染问题等，这些问题在汽车行业发展的同时也给人们提出了思考，使政府以及相关机构不得不采取措施面对这些问题。

1.1.2 中国二手车行业发展现状

对于汽车行业来说，二手车行业是一个重要的组成部分。二手车是指在公安交通管理部门登记注册，在达到国家规定的报废标准之前或在经济实用寿命期内服役，并仍可继续使用的机动车辆【3】。二手车主要有以下特点：

- (1) 价格低廉。和新车相比，二手车的价格较低。而且随着车辆使用年限的增长，车辆的价格还会持续降低。一般来说，使用三年的车辆，价格会比新车降低 50%，这就使二手车的价格充满吸引力。
- (2) 选择主动性强。对于二手车的消费者而言，品牌和车型的选择范围要比新车买家广。二手车市场是车辆的聚集地，品牌琳琅满目，消费者有足够的资源选择到称心如意的车辆。
- (3) 配件多，维修便捷。
- (4) 保值性强，和新车相比，时间对于价格的影响相对较低。

2013 年，中国二手车交易量约为新车的三分之一，而在发达国家，二手车交易量为新车的两到 3 倍。因此，我国的二手车市场仍然有巨大的发展潜力。然而，目前二手车交易市场仍然存在着一些问题，主要表现为【4】：

- (1) 信息不完善，信息来源混杂，真实性难以保证。
- (2) 管理制度不健全，部分二手车市场未与公安部门联网，导致非法车辆流入市场，扰乱了市场秩序。
- (3) 交易手续繁琐，交易过程中需要依靠市场管理人员对交易进行验证，给消费者和销售者带来不便
- (4) 标准不统一，每个二手车市场的收费标准和服务标准各不相同，给税收的征收造成了困难。

1.2 国内二手车软件平台发展现状

随着互联网的发展，诞生了越来越多与二手车相关的软件。而随着智能手机的兴起和移动互联网时代的到来，各种二手车 App 的出现给二手车商及消费者提供了便利。对于二手车 App 来说，想要吸引用户，必须做好以下几点：

- (1) 车源的实时浏览，使用户能看到任一地方即时发布的车源。
- (2) 车辆的快速发布，使二手车商或个人用户能够快速发布自己的车辆。
- (3) 便捷的通讯渠道，使买家和卖家能够即时沟通。
- (4) 信息的便捷管理，使用户能够灵活管理自己发布和关注的车源。
- (5) 工具的内部集成，使用户可以对车辆进行便捷的估价，违章查询等操作。

目前，国内二手车信息平台 App 主要有二手车之家、汽车之家、华夏二手车、中国二手车城等。这些 App 整合自身平台的资源，推出了基本符合上述条件的 App，并且根据自身的优势进行差异化定位，从而吸引到一定规模的用户。然而这些 App 虽然初具规模，但并未脱颖而出，分析下来，大致有以下几点原因【5】：

- (1) 用户粘性不足，设计时没有考虑用户的使用习惯，用户体验较差，导致用户使用一段时间后就放弃使用。
- (2) 设计不合理，无法发挥车商的资源优势，无法使庞大的购买群体快速筛选出自己需要的车型。
- (3) 推广不到位，使某些 App 无法积累庞大的用户群，甚至无法将自己网站的流量导入 App 中。

1.3 本文的主要工作内容及意义

本文的主要任务是设计一款基于 iOS 的二手车信息平台 App，基于百姓网二手车的 Api 完成该 App 代码的编写并上线 App Store。主要工作如下：

- a. 分析当前国内二手车 App 的现状，提出新的二手车信息平台解决方案。
- b. 相关技术概述，主要包括 iOS 系统概述，开发平台概述以及 iOS 开发相关技术的深入描述。
- c. 完成需求分析，并根据需求分析完成概要设计和详细设计。
- d. 根据详细设计划分模块，并完成代码编写。
- e. 对产品进行系统测试和系统性能优化，使产品可交付

本文的主要意义是，通过提出新的二手车平台的设计解决方案，充分发挥了百姓网二手车类目的资源优势以及 iOS 平台高性能的特点，在实时展示资源的基础上实现海量数据的展示。同时，设计了多重筛选的结构，使用户可以在大量的数据中快速找到目标车型。新的聊天工具使买卖双方得以快速沟通。综上，本 App 可以整合优势资源，依靠 iOS 系统的特性使二手车信息展示更加完善，交易更加便捷。

1.4 本文的组织结构

本文共分为六个部分，文章结构如下：

第一章：综述。介绍了课题的研究背景、目前国内二手车应用的现状，阐述了本文的主要工作内容以及意义。

第二章：iOS 开发理论基础。主要从 iOS 系统，XCode 开发工具，Objective-C 语言和 iOS 的高级特性方面介绍了 iOS 开发需要的基础知识。

第三章：系统需求分析与概要设计。主要从功能性和非功能性方面分析了该二手车信息平台的需求。主要从系统架构设计，API 设计，和表现层交互设计方面分析完成了该系统的概要设计。

第四章：系统详细设计及实现。经过概要设计后，将系统合理划分模块，再针对每个模块（二手车列表，搜索，求购，聊天，管理，工具，其他等）完成每个模块的详细设计和实现工作。

第五章：系统测试及优化。介绍了该系统在测试工作中遇到的问题和解决方案，以及该系统在实际使用时遇到的问题和优化方案。

第六章：工作总结及展望。对本文所涉及的工作和成果进行了总结，并对将来的工作进行了展望。

2 iOS 开发理论基础

2.1 iOS 系统和 Xcode

2.1.1 iOS 系统简介

iOS 是苹果公司为移动设备开发的操作系统，于 2007 年 6 月 29 日发布第一个公开版本，现在的版本是 iOS8.3。iOS 支持的设备有 iPhone、iPad、iPod 等。该系统主要有以下特点【6】：

- (1) iOS 系统支持的设备屏幕为触控屏，一般正面只有一个物理按键（Home 键），用户可以通过触控屏的多点触控功能对设备进行控制。此外，设备内建加速器，通过旋转屏幕可以改变屏幕的显示方向。
- (2) iOS 系统使用流畅，由于 iOS 设备的内存限制，在系统中除了后台服务外，只允许有一个活动的 App。其他 App 只允许在后台调用一些支持的系统接口获取数据，使系统运行变得流畅。
- (3) 强大的多点触控功能，支持包括轻按、长按、拖拽、缩放、滑动等多种手势，对于 iPad 还可以使用 4 指或 5 指切换任务。
- (4) iOS 系统通过苹果公司的 iCloud 服务，可以在不同设备之间即时同步数据。

iOS 系统基于 Darwin 内核，由 Free BSD 和 Mach 所改写。分为核心操作系统层，核心服务层，媒体层和触控应用层，如图 2.1 所示。核心操作系统层，主要包含了一些底层组件，包括内存管理，文件系统等服务。通常只有系统框架类能够直接调用这一层的接口；核心服务层，使应用程序可以操作文件，以及使用基本的系统框架，这些框架定义了基本的数据类型。媒体层，主要包括了应用程序需要使用的媒体类框架，用于处理图片、音频和视频。触控应用层，为开发者提供各种封装的框架以及可视化组件，开发者可以利用这些框架设计出不同布局的界面，从而开发出各种优秀的应用。

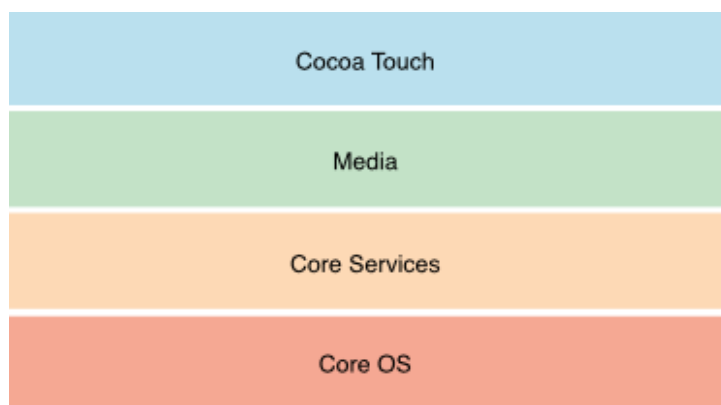


图 2.1 iOS 系统架构

2.1.2 Xcode 简介

XCode 是苹果公司向开发人员提供的集成开发环境【7】，用于开发 Mac 和 iOS 应用程序，

于 2003 年 9 月 28 日推出第一个版本，目前稳定版为 6.3。XCode 的前身是 NeXT 公司的 Project Builder，支持 C、C++、Objective-C 等多种编程语言。从 Xcode3.1 开始，附带 iOS SDK，作为 iOS 的开发环境。XCode 的主界面如图 2.2 所示：

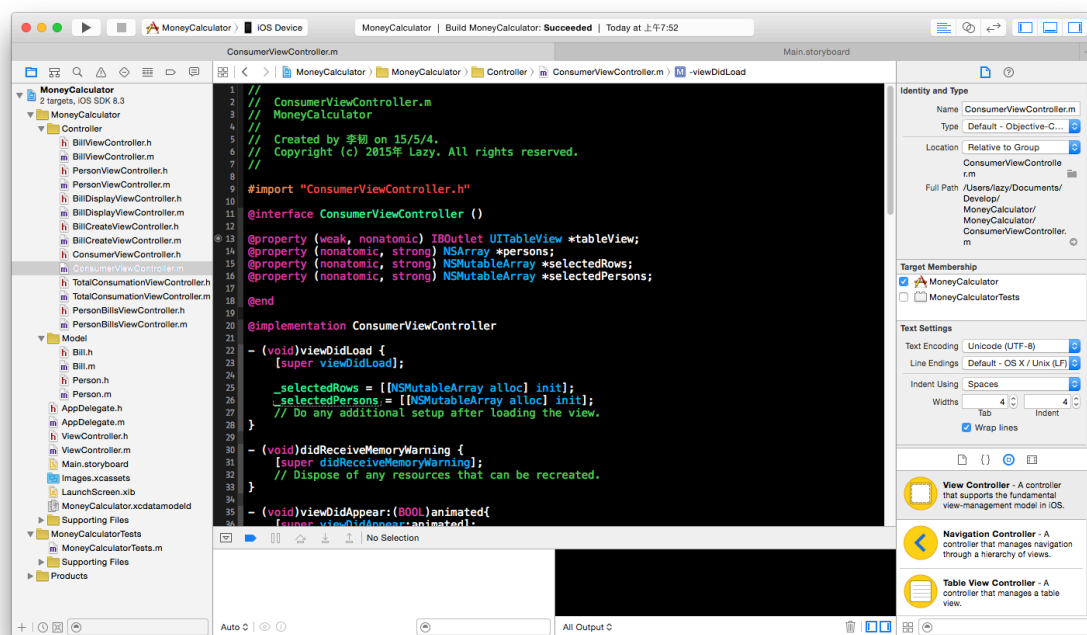


图 2.2 XCode 主界面

和其他集成 IDE 相比，XCode 主要有以下特点【8】：

单窗口界面。单窗口界面使用户无论是编码，UI 设计，测试还是做其他工作，都能专注于某项特定的任务，可以通过建立多个标签使用户在不同的单窗口界面间进行切换。

辅助源码编辑。XCode 通过智能补全功能加快用户的输入。当出现错误时，会高亮代码进行提示。XCode 可以编辑代码段提高工作效率，减少不必要的打字时间。

可视化用户界面设计。通过内建的 Interface Builder 组合窗口、界面、控件、菜单等其他元素、从而搭建应用程序的界面。还可以便捷的将界面上的元素（控件、约束等）链接到代码，使开发者创建界面的方式更加便捷。

集成调试。XCode 支持通过模拟器以及真实地设备进行调试。即使是在真实设备调试时，也可以通过断点随时进行调试操作。通过 Quick Look 功能快速查看特定变量的值。XCode 还可以在调试过程中实时查看网络，内存，CPU 的使用情况。

测试和持续集成。XCode 内部包含了可以用于功能和性能测试的框架。可以通过 XCTest 等框架编写单元测试代码。XCode 可以通过 bots 自动进行单元测试，打包，分发等操作，当持续集成过程出现错误时，bots 也会发出报告，便于用户分析。

自动保存，工程快照以及源代码控制。XCode 会自动保存对源码和项目的更改，即使 XCode

意外退出或机器意外断点，XCode 再次打开时也可以将工程恢复到之前的状态。XCode 的源码控制系统支持 Git 和 SVN 两种代码。

完善的文档。XCode 为开发者提供详细的技术信息，通过 Quick Help 可以在视图内查看简明的 API 信息。XCode 的文档包括 API 说明、编程指南、教程、示例代码以及演示视频等，所有的资源都可以在 XCode 文件查看器中浏览并离线下载。

将应用程序分发给测试者或者提交到 App Store。XCode 可以便捷的分发打包好的 App，或是直接提交到 App Store 审核。

2.2 Objective-C 介绍

Objective-C 是苹果公司开发 iOS 和 Mac 应用的主要编程语言。该语言发行于 1983 年【9】，目前版本为 2.1。Objective-C 将 Smalltalk 的消息传递机制与 ANSI C 融合，是 C 语言的超集。任何 C 语言代码都可以直接通过 Objective-C 的编译器。

Objective-C 和 C 相比加入了面向对象的特性，最大的特点是消息传递机制，该机制与 C 语言中的函数调用的机制有很大的不同，在代码中向一个类的实例传递消息时，即使该类没有处理消息的方法，也可以编译通过，但是在运行时会抛出异常。而 C 语言中，调用一个类的方法时，如果该类没有该方法，则编译时就会报错。

Objective-C 的其他特性包括引入了协议、动态类型、转发、类别等概念，而对于内存管理，目前的 Objective-C 实现了自动的垃圾回收机制，即 ARC（Automatic Reference Counting）【10】。通过该方式，开发者无需手动管理内存，这大大减轻了开发者的负担，使开发者可以专注于功能的实现。

2.3 MVC 模式

MVC（Model-View-Controller）模式是 iOS 应用开发的基本模式，即模型-视图-控制器。最早于 1978 年为 Smalltalk 提出。MVC 模式将应用划分为模型，视图和控制器。MVC 基本模型如图 2.3 所示：

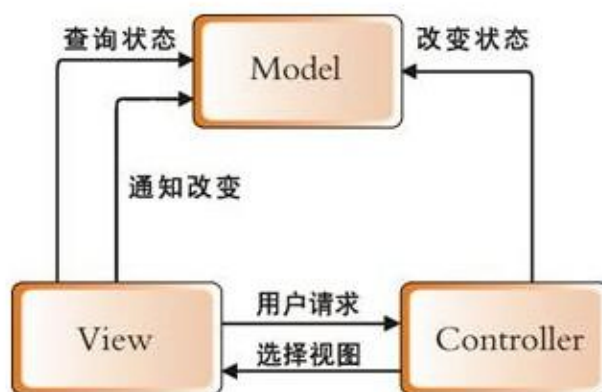


图 2.3 MVC 基本模型

模型（Model）,主要用于处理数据模型及业务逻辑，可以对数据进行直接访问，以及在其中处理业务逻辑。

视图（View）,能够使数据有目的的显示出来，视图一般不处理逻辑上的功能，只负责接收需要展示的内容并展示出来。

控制器（Controller），起到不同层面的组织作用，是模型和视图的桥梁。控制器将用户的操作传递给数据模型，模型处理逻辑后再通过控制器传给视图。

一般来说，模型和视图各自独立，通过控制器相连接。这种设计模式可以方便的对应用划分模块，具有高内聚、低耦合的可复用性和可维护性，使代码和用户界面的交互变得清晰。为开发者创建应用提供了基础。

2.4 本章小结

本章主要介绍了 iOS 开发的基本技术，包括 iOS 系统的简要介绍，以及开发 iOS 所使用的集成开发环境 XCode 的介绍。之后介绍了开发 iOS 所使用的 Objective-C 语言的基本特点，以及开发 iOS 应用常用的 MVC 框架。了解关于 iOS 基本技术的内容对于设计并开发一款 App 是必要的，可以使开发者在设计应用时兼顾 iOS 的特性，设计出能充分发挥 iOS 系统特点的应用。

3 系统需求分析与概要设计

本章主要对系统的需求分析和概要设计进行描述。对于需求分析，主要从功能性需求分析和非功能性需求分析两方面进行描述。对于概要设计，主要从系统架构设计，API 设计，界面设计三方面进行描述。

3.1 系统需求分析

系统需求分析是系统开发中的重要组成部分，对于系统需求的分析可以有效地了解业务需求，合理划分模块。对于需求的分析包括功能性需求分析和非功能性需求分析。功能性需求分析是指按功能列出系统的特定结果，而非功能性需求会列出系统的一些整体特性。

3.1.1 功能性需求

本系统的目的为设计并实现一个二手车信息平台，该平台的主要功能为：查询本地车源、个人车源、全国车源，筛选符合特定条件的车源，发布车辆，用户间实时聊天，车辆求购，信息管理，车辆估价，违章查询等。本应用的主要工作形式为，客户端调用百姓网后台的接口获取信息，然后加工信息后在客户端进行展示。

据此，可以将本系统划分为以下模块，如图 3.1 所示：

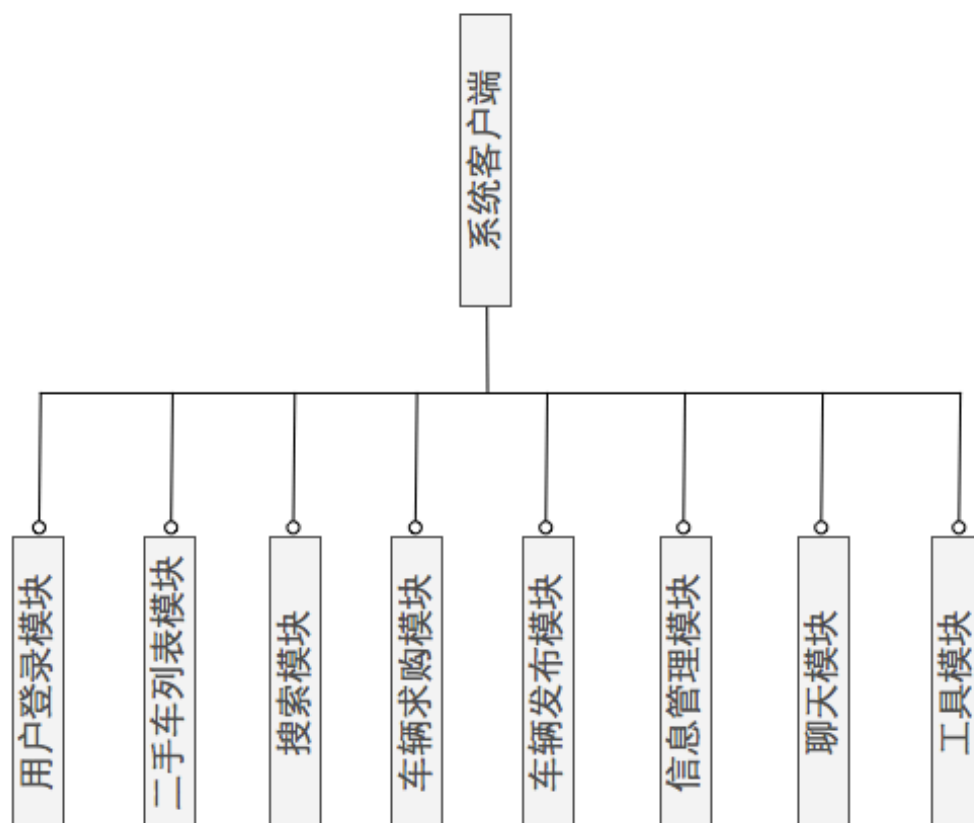


图 3.1 客户端功能模块

系统主要划分为：用户登录模块、二手车列表模块、搜索模块、车辆求购模块、车辆发布模块、信息管理模块、聊天模块、工具模块等主要功能模块。下面具体介绍各模块的功能性需求。

- a) 用户登录模块，是本系统的基础性功能模块，用户必须通过登录模块登录成功后才可以使用系统的其他功能。对于登录模块，如果用户已在百姓网注册过，则可以通过输入密码或接收手机验证码两种方式登录。如果用户未在百姓网注册，则只能通过手机号接收验证码登录。系统登录时需要连接网络，根据返回的参数判断登录结果。如果用户已经登录过系统，则无需再次登录。本系统不提供注册功能。
- b) 二手车列表模块，该模块的主要功能是展示最新发布的二手车源，包括个人车源，本地车源和全国车源。模块按列表的形式展示车源内容，每次展示一定的数量，用户下拉时会刷新列表，上拉时可以加载更多内容，在模块内部可以按一定的条件筛选信息，包括城市、车型、价格、行驶里程等。点击一条信息后会展示车辆详情。该模块是本系统的核心模块。
- c) 搜索模块，该模块的主要功能是按关键词搜索车源，搜索的范围是二手车列表中展示的车源。如，二手车列表中展示的是本地车源，则只能在这些车源内按关键字搜索结果。搜索的方式有文字输入和语音输入两种。搜索记录要求本地化存储，下次打开应用时直接展示出来。
- d) 车辆求购模块，该模块的主要功能为展示全部车辆求购的内容，发布以及管理自己的车辆求购信息，在该模块内支持按收车区域、收车价格等方式筛选。
- e) 车辆发布模块，该模块的主要功能是发布车源信息，发布的信息包括车型、行驶里程、车辆颜色、价格、车辆照片、车辆描述等内容，发布信息时需要连接网络，根据服务器返回的结果判断发布成功或失败。
- f) 信息管理模块，该模块的主要负责用户对自己的个人信息的管理，包括用户昵称，已发布车辆管理和已收藏车辆管理。对于已发布和已收藏的车辆，支持从列表中删除，然后发送信息给服务器，在数据库中删除对应的信息。
- g) 聊天模块，该模块利用融云的第三方实时通讯框架在应用内部实现了用户聊天的功能。用户可以发送车辆信息，文字，音频，也可以通过语音电话发起和另一位用户的会话。该模块分为全国聊天室，本地聊天室，个人聊天三种形式。
- h) 工具模块，该模块主要为用户提供各种便捷的工具，如车辆估价、违章查询、限迁查询等，这些工具的表现形式皆为第三方网页，在系统内点击相应地按钮直接打开，本系统使用这些工具时已得到工具作者的授权。

3.1.2 非功能性需求

除了必要的功能性需求外，应用还需要关注非功能性需求，非功能性需求定义了系统的特性，解决的是“如何使系统在真实环境中运行”的问题，在系统开发中常常被轻视，如果开发者在需求分析中没有考虑非功能性需求，那么设计的系统将很难取得实效。本文在设计系统时考虑了以下非功能性需求：

- 系统的健壮性。本应用在设计时充分考虑了系统的健壮性问题，在用户输入的地方，如密码输入，车辆价格输入时加入了错误处理功能。防止由用户误操作带来的系统意外崩溃。
- 网络环境的处理。由于本系统的各项功能是在网络环境下正常实现的，在不同的网络环境下，如弱 Wi-Fi 环境，2G/3G/4G 环境下，系统都能做出合理地响应，即响应时间或是阻塞主线程的时间不应太长。
- 设计风格以及用户体验的一致性，在设计应用时，符合苹果官方发布的《iOS Human Interface Guidelines》上的设计规范，即界面风格保持一致性，字体，图标尺寸合理，界面简洁。同时，注重用户体验，充分符合 iOS 用户的操作习惯。例如，当“返回”按钮位于左上角时，界面支持右划返回，当“返回”按钮位于右上角时，文字用“取消”代替，点击时界面向下收出屏幕。保持良好的用户体验可以使用户在使用时更加便捷，专注于应用的功能。

3.2 系统概要设计

系统概要设计是指系统的总体实现方案，确定系统的总体布局，各个子模块的功能和模块间的相互关系，以及与外部系统的关系。概要设计从整体上说明了软件的实现思路。本系统从系统架构设计，API 设计和界面设计三方面阐述系统的概要设计。

3.2.1 系统架构设计

根据需求分析，将系统划分成了若干模块，从而可以设计出如下的系统架构，如图 3.2 所示：

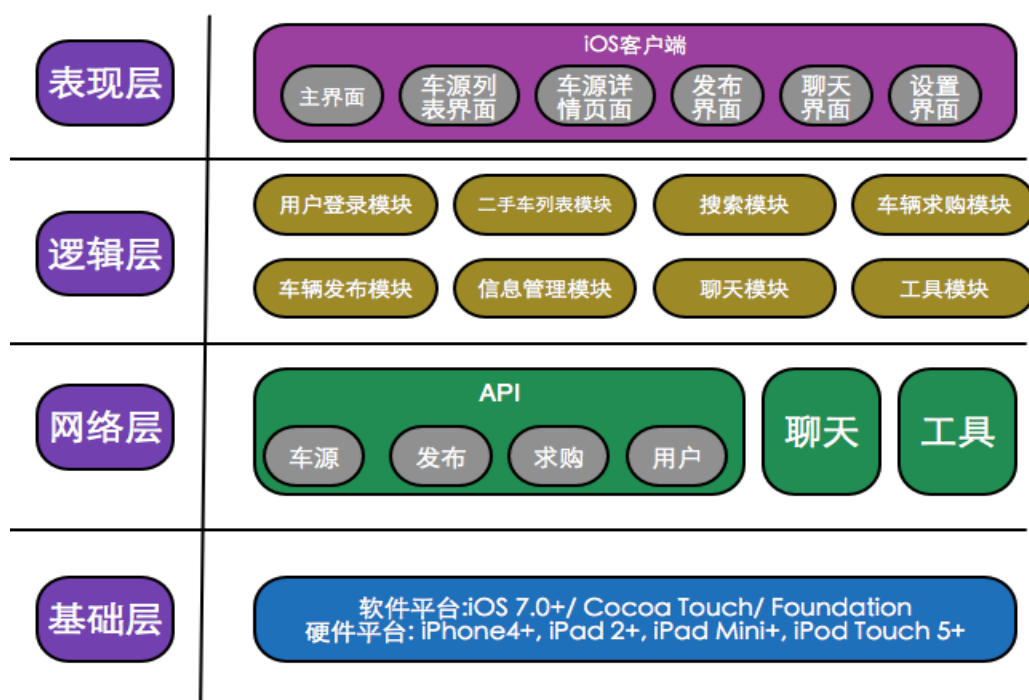


图 3.2 系统架构图

基础层：基础层是本系统所依赖的硬件和软件标准。本系统需要在 iOS7.0 以上的系统上运行，同时需要 Cocoa Touch, Foundation 等系统框架作为支撑。需要 iPhone 4 以上的 iPhone 手机，iPad 2 或 iPad mini 以上的 iPad 设备或是 iPod Touch 5 以上的 iPod 设备运行。

网络层：系统通过网络层与外部进行通信，网络层分为三个部分：①API，是系统获取车型资源，发布车辆，处理用户登录以及其他核心操作所调用的百姓网 API；②聊天，是系统所使用的融云第三方即时通讯框架，通过连接融云的服务器实现用户间的实时聊天功能；③工具，是系统所使用的第三方工具 URL，系统通过直接打开 UIWebView 加载这些第三方工具页面。

逻辑层：系统通过逻辑层处理核心事物逻辑，逻辑层直接与网络层和表现层通信，接收网络层传来的数据，处理后发送给表现层。逻辑层按模块划分，各模块的功能描述见 3.1.1 节。

表现层：表现层主要负责界面的展示，接收用户的触摸和输入事件等。对于表现层的设计见 3.2.3 节。

3.2.2 API 设计

本系统所调用的 API 包括：获取车源的 API，发车的 API，求购的 API，获取用户信息的 API，登录的 API 等，通过分析功能性需求，对需要的 API 设计如下(由于 API 数量过多，下面只列出核心功能的 API)：

表 3.1 获取车源 API

API 名称	http://www.baixing.com/api/mobile/ershouqiche/ad		
API 参数及说明	参数	类型	说明
	apiFormatter	String	可选，有此项则返回获取的总数
	Koala	Int	可选，有此项且设为 1 为获取个人车源
	City	String	可选，无此项则获取全国车源
	From	int	可选，从第几项开始
	Size	Int	可选，返回的数目
	sort	String	可选，指定的排序方式
	车品牌	String	可选，指定的车品牌
	价格	Dictionary	可选，指定的价格范围
	年份	Dictionary	可选，指定的车龄范围

表 3.2 发布车源 API

API 名称	http://www.baixing.com/api/mobile/ad.insert/		
API 参数及说明	参数	类型	说明
	title	String	必填，标题
	Content	String	必填，车辆描述
	Contact	String	必填，联系电话

续表 3.2

API 参数及说明	参数	类型	说明
	Images	Array	选填，上传的图片，用 NSData 表示
	车系列	String	必填，车辆类型
	价格	String	必填，车辆价格
	年份	String	必填，车龄
	车辆里程	String	必填，车辆里程

表 3.3 获取求购信息 API

API 名称	http://www.baixing.com/api/mobile/CartierWantAd.getAds		
API 参数及说明	参数	类型	说明
	City	String	可选，无此项则获取全国求购车源
	From	String	可选，从第几项开始
	Size	String	可选，返回的数目
	sort	String	可选，指定的排序方式
	车品牌	String	可选，指定的车品牌
	价格	Dictionary	可选，指定的价格范围
	年份	Dictionary	可选，指定的车龄范围

表 3.4 获取用户已发的求购列表 API

API 名称	http://www.baixing.com/api/mobile/CartierWantAd.getMyAds		
API 参数及说明	参数	类型	说明
	City	String	可选，无此项则获取全国求购车源
	From	String	可选，从第几项开始
	Size	String	可选，返回的数目
	sort	String	可选，指定的排序方式
	车品牌	String	可选，指定的车品牌
	价格	Dictionary	可选，指定的价格范围
	年份	Dictionary	可选，指定的车龄范围
	id	String	必填，用户 id

表 3.5 获取用户已发车源 API

API 名称	http://www.baixing.com/api/mobile/user.importantAds		
API 参数及说明	参数	类型	说明
	id	String	必填，用户 id

表 3.6 用户登录 API

API 名称	http://www.baixing.com/api/mobile/user.login		
API 参数及说明	参数	类型	说明
	type	String	必填，用户登录类型，密码或验证码登录
	Identity	String	必填，用户名
	Password	String	必填，用户密码

表 3.7 用户删除已发车源 API

API 名称	http://www.baixing.com/api/mobile/ad.delete		
API 参数及说明	参数	类型	说明
	id	String	必填，用户 id

3.2.3 界面设计

根据功能需求以及非功能性需求的分析，同时根据《iOS Human Interface Guidelines》的设计规范，系统设计了如下界面（限于篇幅，仅介绍设计核心功能的界面）：

登录界面，如图 3.3 所示：



图 3.3 登录界面

主界面，如图 3.4 所示：



图 3.4 主界面

车源列表界面，如图 3.5 所示：



3.5 车源列表界面

车源详情界面，如图 3.6 所示：



图 3.6 车源详情页面

搜索界面，如图 3.7 所示：



图 3.7 搜索界面

发布界面，如图 3.8 所示：



图 3.8 发布界面

求购界面，如图 3.9 所示：



图 3.9 求购界面

信息管理界面，如图 3.10 所示：



3.10 信息管理界面

3.3 本章小结

本章主要介绍了对本系统的需求分析和概要设计。需求分析主要从功能性需求好非功能性需求两个方面进行了分析，在功能性需求分析完成后，将系统合理地划分成了若干模块；非功能性需求主要从系统的健壮性，网络环境的处理，设计风格以及用户体验的一致性三个方面进行了分析。在需求分析完成后，对系统进行了概要设计，概要设计由系统架构设计，网络 API 设计和界面设计三部分组成。对系统的需求分析和概要设计为系统的详细设计和具体功能的实现提供了基础。

4 系统详细设计及实现

4.1 用户登录模块的详细设计及实现

用户登录模块是用户登录系统后第一个访问的模块，它的主要功能是向服务器发送用户的用户名密码，并返回登录结果。如果登录成功，则进入主界面，否则，将停留在登录模块。登录模块的具体流程图如图 4.1 所示：

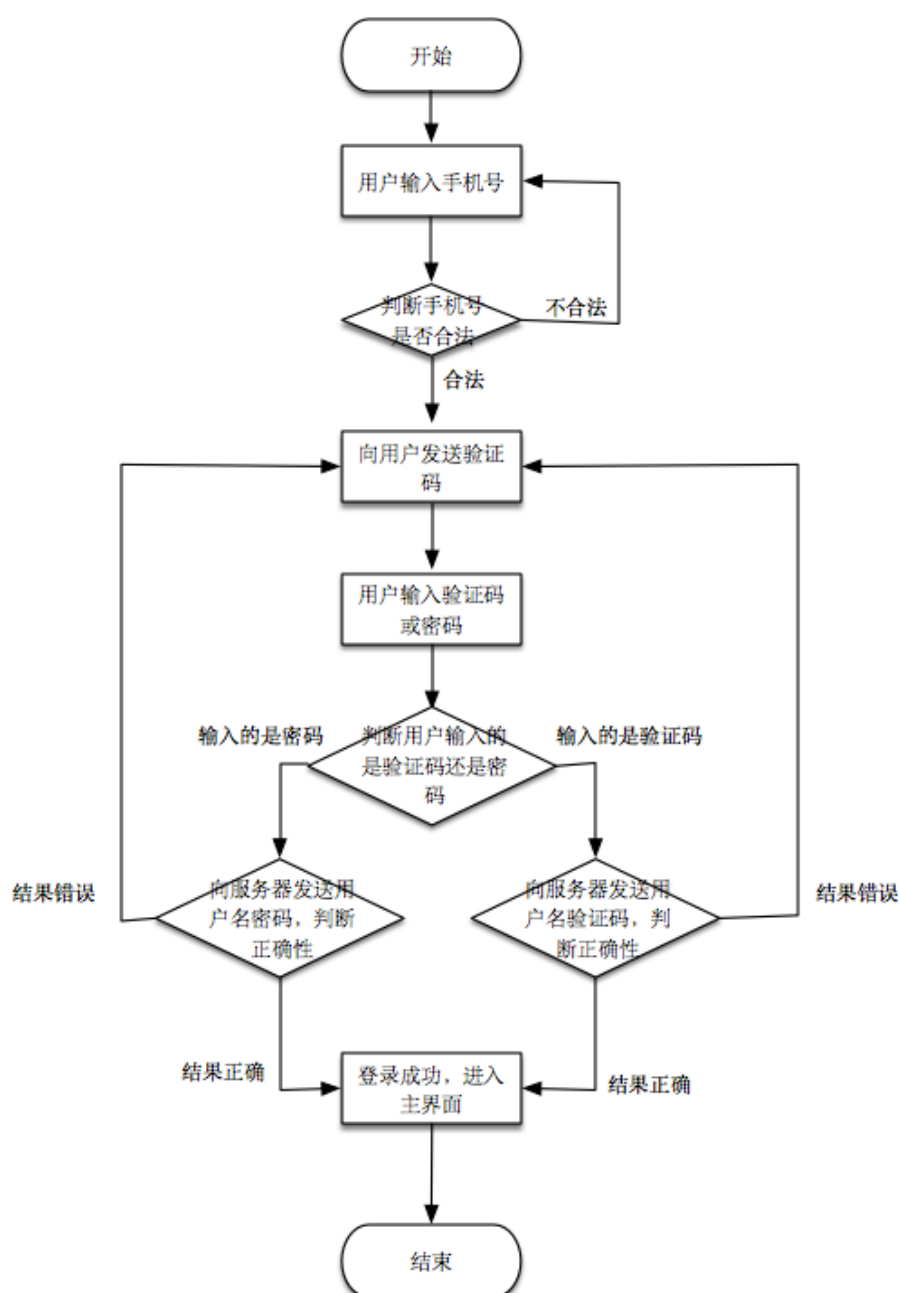


图 4.1 登录流程图

整个流程由 CartierLoginController 和 CartierVerifyController 两个类负责，其中，CartierLoginController 主要负责手机号的输入和验证，CartierVerifyController 主要负责处理用户输入的密码或验证码，以及向服务器发送登录请求并接收结果的工作。

在 CartierLoginController 中，核心逻辑是手机号的验证，即判断用户输入的号码是否为真实存在的手机号，这个工作主要利用正则表达式完成，匹配的模式为：

```
NSString *phoneRegex = @"^((13[0-9])|(15[^4,\D])|(18[0,0-9]))\d{8}$";
```

用户正确输入手机号后，系统将自动向用户发送验证码短信，如果用户没有在百姓网注册过，则只能选择这一种方式输入登录凭证，如果用户已经在百姓网上注册过，则可以选择输入验证码或是密码。当用户输入完成后并点击登录按钮后，系统将向服务器发送登录请求，请求 API 见表 3.6。在本系统中，发送请求采用的是第三方的 AFNetworking 框架，AFNetworking 构建在 NSURLConnection 以及 NSOperation 之上，同时利用了 Foundation 框架的特性，拥有良好的架构以及丰富的 Api，可以使开发者更方便的发送网络请求。同时本系统多采用 Block 方式发送网络请求。Block 是 iOS4 之后引入的新语法，类似于 Java 中的匿名类，是一个匿名的内联代码集合。通过使用 Block 可以实现异步调用的方式，即，向一个对象发送消息后，当该对象完成相应地处理后，再通过 block 回调。本系统在登录时发送密码的具体操作如下：

```
[BaixingAuth loginWithMobile:self.loginName andPassword:password thenCallback:
```

```
^(NSDictionary *dic, NSError *err){
    if (err) {
        [self showFailure];
    } else {
        [self loginSuccess];
    }
}];
```

该消息包含三个参数，用户名、密码和回调方法。回调时，会根据结果执行相应地操作。发送验证码的方法与该方法类似。当登录成功后，会加载并进入主界面。

4.2 二手车列表模块的详细设计与实现

二手车列表模块是本系统最为核心的模块，主要功能是二手车源的展示，分类的筛选以及车源详情的查看。二手车源的展示分为三种，个人车源，本地车源及全国车源，本文以全国车源为例介绍这个模块的具体设计实现，分为车源列表部分和车源详情部分。

4.2.1 车源列表部分

该模块的主界面如图 3.5 所示，分为导航栏、筛选栏以及车源列表。车源列表采用的是 UITableView，列表支持下拉刷新，以及上拉加载更多，列表的 cell 为自定义风格，全部元素由代码搭建而成。导航栏的左侧为返回按钮，中间为标题栏，右侧为搜索按钮，点击搜索按钮后将进入搜索界面。中间的筛选栏（如图 4.2 所示）视车源列表的类型略有不同，下面作重点介绍。

不限品牌 | 不限价格 | 城市 | 默认排序 | 精准筛选

图 4.2 筛选栏

中间的筛选栏由 `UIFilterView` 类负责，该类继承于 `UIView`，属于 `View` 层，该类的核心方法是 `-(void)setFilters`，这个方法的工作步骤如下：

- 配置不同属性的条目事件，如城市条目，点击后进入城市选项列表。
- 读取车源展示类型，从而判断筛选栏条目的个数（如果为全国车源，则加入城市一栏）。
- 根据条目的个数均等计算每个条目所占的宽度，每个条目都是一个 `UIButton`。
- 将实例化的 `UIButton` 加入 `View` 中，并根据配置赋予不同的事件。

筛选栏分为五种，其中品牌、价格、城市、精准筛选都会进入新的 `Controller` 中进行选择，默认排序将直接在当前视图下动画弹出下拉选择框，并进行背景的高斯模糊处理，效果见图 4.3 所示。



图 4.3 默认排序下拉选择菜单

其中，下拉菜单是一个 UITableView，当点击默认排序时，直接由位置移动动画加以呈现，下拉菜单的每一项都对应不同的参数，与表 3.1 的 sort 字段相对应，根据参数的不同返回不同的结果。

品牌筛选，城市筛选，价格筛选的工作原理类似，以品牌筛选为例。品牌筛选由 TreeViewController 类负责，由于车型众多，所以引入了一个车型数据库 Local.db，这个数据库是一个小型的 sqlite 数据库，在程序运行时直接读入本系统中，系统由 FMBasedObject 类负责读取数据库，该类引用了 FMDB 的第三方库，FMDB 是基于 SQLite 构建的一个第三方库，提供了便于开发者使用的 API。同时，以 FMBasedObject 为基类，新建了子类 CarType，并提供一些简单的 API，从而方便的按级调取车型。品牌筛选的界面如图 4.4 所示。



图 4.4 品牌筛选界面

当筛选完成后，调用 UIFilterView 类的 - (void)onChangeParams:(NSDictionary *)params 方法，该方法会通知其所属的 View Controller 刷新车源列表，并对其传递筛选完成后的参数。

筛选栏的最后一项是精准筛选，精准筛选是对前面几项筛选功能的补充，可以对里程、车龄等类目进行精确地筛选，精准筛选的界面中采用了自行开发的两点拖动条，可以根据手指的位置实时更新箭头的位置，并换算成箭头当前代表的数值，方便的设置数值的起点和终点。精准筛选的界面如图 4.5 所示。

二手车列表模块在加载时与登录类似，采用的是异步获取数据的方式。同时，取得数据后，对每一个 cell 的刷新也是采用异步方式，这样做的优点是在获取数据时保证主线程不被阻塞，图片加载工作在后台进行，从而提高了用户体验。

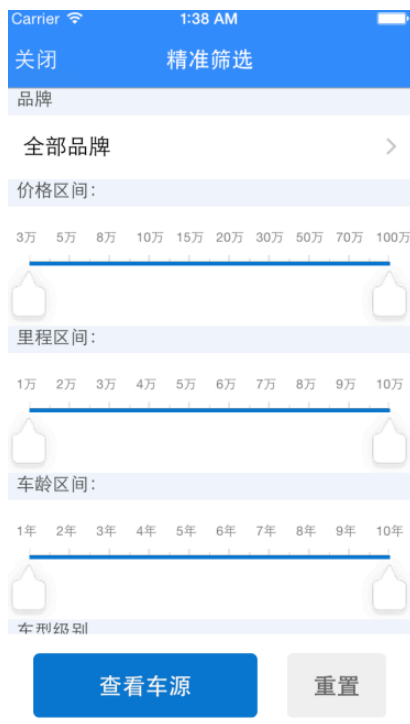


图 4.5 精准筛选界面

4.2.2 车源详情部分

在车源列表中点击任意一个车源会进入车源详情页面，车源详情页面如图 3.6 所示，主要处理的是界面的搭建以及内容的展示工作。在该页面上还可以进行收藏，分享到微信等操作。点击收藏时，会调用网络 api 将用户信息和该车源的 id 作为参数传递，而分享到微信使用微信的分享框架，支持分享到朋友圈和发送给好友。

车源详情页面同时支持私信和给卖家拨打电话，私信功能的介绍见 4.7 节，给卖家拨打电话调用 UIApplication 类的 openURL 方法，其中 telprompt 为 iOS 直接拨号的固定 URL：

```
UIApplication sharedApplication] openURL:[NSURL URLWithString:[NSString
stringWithFormat:@"telprompt://%@", self.ad.contact]]];
```

4.3 搜索模块的详细设计与实现

搜索模块负责的是按关键词搜索一定范围内的车源，搜索模块只有通过二手车列表模块才可以进入，搜索的范围仅限定于之前二手车列表出现过的车辆。该模块是系统中一个较小的模块，为了提供较好的用户体验，根据功能性需求分析，为该模块设计了如下的功能要点：

- 进入搜索界面（见图 3.7）后直接呼出输入键盘。
- 搜索完毕后监听输入法的回车事件，立即展开搜索。
- 搜索结果须持久化保存，以便下次打开系统时直接使用。
- 搜索支持语音输入，语音输入时可以随时暂停，语音输入后立刻将语音输入结果反馈到

搜索栏，并呼出输入键盘，方便用户对输入结果进行修正。

下面逐个分析上述功能要点的实现。

直接呼出输入键盘，采用的是 iOS 中抢夺第一响应者的方式。在 iOS 的视图界面中有“响应链（Response-Chain）”的概念，即如果子视图无法对事件做出相应，那么事件会传递给它的父视图，直到有视图对其做出响应为止。当搜索框作为第一响应者时（First-Responder），输入键盘会被弹出，但是进入该页面时不存在第一响应者，所以采取下面的方式让搜索框成为第一响应者：

```
[_searchBar becomeFirstResponder];
```

_searchBar 为 UISearchBar 的实例化对象，becomeFirstResponder 为 UIView 的实例方法，即使该 view 成为第一响应者。

搜索完毕后监听输入法的回车事件是 UISearchBar 的代理事件，即，UISearchBar 会自动检测用户是否按下回车，然后回调该方法。

对于搜索结果的持久化保存是由 CartierSearchManager 类负责，搜索结果的集合是一个 NSArray，这个类封装了一些常用的方法处理搜索结果，如添加记录，删除记录。这些方法执行完毕后会自动保存结果到本地，方法是调用 NSArray 类的 writeToFile 方法以文件的形式保存到本地，下次打开应用时再读取。

搜索模块的语音输入采用的是科大讯飞的第三方库，识别时会调用该库的一系列方法，上传语音后返回结果，为了方便用户使用，设计 UI 如图 4.6 所示。



图 4.6 搜索界面的语音识别 UI

该 UI 实现的难点是向内添加“点击停止录音”的按钮，并使其上边直形化，下边圆角化，本系统的实现方案是将此按钮作为直角矩形添加，然后在其下方添加一个圆角遮罩，实现方案如下：

```
UIBezierPath *maskPath = [UIBezierPath bezierPathWithRoundedRect:_stopButton.bounds
byRoundingCorners:(UIRectCornerBottomLeft | UIRectCornerBottomRight)
cornerRadii:CGSizeMake(10.0, 10.0)];
```

添加完遮罩后，实际上放大了用户的可触摸部分，图 4.7 中的灰色和绿色部分即为可触摸部分，但绿色部分是隐藏的。由于实际使用时绿色部分面积过小，用户基本无法察觉，所以这种遮罩方案是可行的。

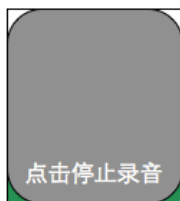


图 4.7 语音输入 UI 草图

用户执行搜索后，系统跳转到二手车列表模块展示搜索结果，该界面与之前界面的不同点是，导航栏右侧不再有搜索按钮。

4.4 车辆求购模块的详细设计与实现

车辆求购模块是供用户发布求购信息的地方。粗略来说，车辆求购模块与二手车列表模块有一定的相似之处，它们都展示的是一个车源列表。但由于目的不同，所以在设计时应区分它们的不同点。该模块的核心依然是一个列表界面，如图 3.9 所示，分为导航栏，筛选栏和车源列表栏，发布求购四项，导航栏的右侧按钮为“我的求购”，点击可进入我的求购的管理。筛选栏复用了 4.2 中提到的筛选栏，根据求购的设置初始化品牌、价格、区域三个维度的筛选。发布求购是一个橙色按钮，固定在屏幕底端。下面重点介绍车源列表栏。

车源列表栏的实现细节与二手车列表模块的列表栏类似，但根据求购模块的功能分析，对 UITableViewCell 的设计有所区别，求购列表的 Cell 设计如 4.8 所示。



图 4.8 求购列表的 cell 设计

图 4.8 中间的灰色部分展示的是需要显示的求购内容，其中横线上方的部分为求购描述，这种设计的问题是由于求购描述的长度不同，造成了每个 cell 的高度都不尽相同。而 UITableView 的 cell 高度是默认相同的，需要采取一定的方案使每个 cell 都完全展示自己的内容。

本系统采用的方案是先读取每个 cell 展示的文本内容，然后在一定宽度下计算展示该文本内容需要的高度，并将其保存，然后调用 UITableView 的 reloadData 方法，在 reload 时 UITableView 的 -(CGFloat)tableView:(UITableView *)tableView heightForRowAtIndexPath: (NSIndexPath*) indexPath 方法，该方法会按 indexPath 读取相应地高度。由于每次加载的 cell 为 20 个，所以该方法的性能开销不大。该方法的流程图如图 4.9 所示。

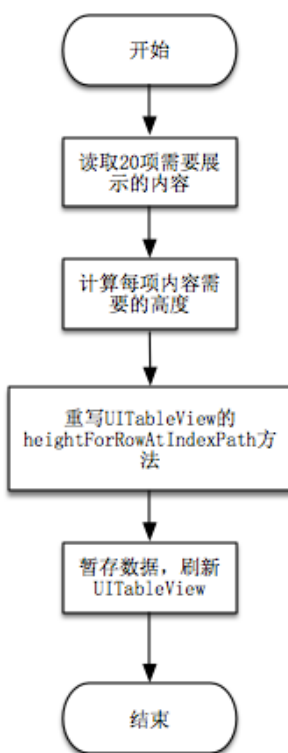


图 4.9 动态加载 UITableViewCell 的过程

求购模块的第二部分是我的求购板块，用户可以在该部分管理自己已发布的求购内容，这一部分也是由 UITableView 组成的列表，和之前的车源列表不同的是，该列表的每个 cell 都配置了删除按钮，点击删除按钮后，会提示是否要删除该求购信息。当删除信息时，要先删除数据源，即保存在 NSArray 中的信息，再调用 UITableView 的删除方法删除一个 cell，否则，系统会报错。原因是删除 cell 时 UITableView 会进行刷新，此时 cell 的数目和数据源中的数目并不相符，违反了 UITableView 的规定，因此会报错。

求购模块的第三部分是发布求购信息，此部分和 4.5 节中介绍的车辆发布模块类似，请参见 4.5 节的内容。

4.5 车辆发布模块的详细设计与实现

车辆发布模块负责的是二手车源的发布，对于二手车源的发布，根据功能性需求分析，主要设计了车品牌、价格、行驶里程、车辆颜色，是否包含过户费，车辆详情以及车辆照片等字段。每个字段的具体填写要求如表 4.1 所示。

表 4.1 车辆发布各个字段的具体设计

字段名称	字段具体设计
价格	文本框填写
行驶里程	文本框填写
车型品牌	跳转 TreeViewController 进行筛选
车辆颜色	弹出 PopOver 式 View 进行选择
是否包含过户费	Radio Button 进行单选
上传照片	弹出 UIActionSheet 供用户选择拍照或是从相册中选择
车辆详情	文本区域输入，支持语音输入，类似 4.3 节搜索的语音输入

下面详细介绍车辆颜色选择和上传照片两个部分的设计与实现。

4.5.1 车辆颜色选择的设计与实现

车辆颜色的选择采用的是类似弹出式窗口的方式。如图 4.10 所示，



图 4.10 车辆颜色选择的界面

单击选择车辆颜色的字段时，该 view 移到视图的最上层，同时视图的其他部分做变暗处理。该 view 的布局为 18 个按钮，经过读取颜色的配置，形成不同的颜色按钮，当单击任意一个按钮时，该页面会隐藏到视图最底层。

4.5.2 上传照片的设计与实现

在发布车辆界面单击“点击上传车辆照片”后，会弹出一个 UIActionSheet 似的选择视图。UIActionSheet 是 UIKit 中的选择按钮项，可以添加若干项，并为每一项添加事件。本系统中弹出的视图作用与 UIActionSheet 类似，但为了更直观地说明每一项的内容，采用封装 UIView 的方式，如图 4.11 所示。



图 4.11 上传照片的选择视图

用户可以选择拍照和相册选择两种方式，支持添加总计 12 张照片。当选择拍照模式时，跳转到 CartierImagePicker- Controller 类，该类是 UIImagePickerController 的子类，用于从照相机或相册中获取图片，由于选择的是拍照模式，所以对该类的设置如下：

`imagePicker.sourceType = UIImagePickerControllerSourceTypeCamera;`

除此之外，还应设置代理，以便执行回调函数，拍摄完毕后，将回调的照片加入待上传照片集（NSArray）中。

当用户选择从相册选择时，界面跳转到 BXImagePickerController 类，该类封装了 UIImagePickerController，加入了多选按钮，见图 4.12 所示。

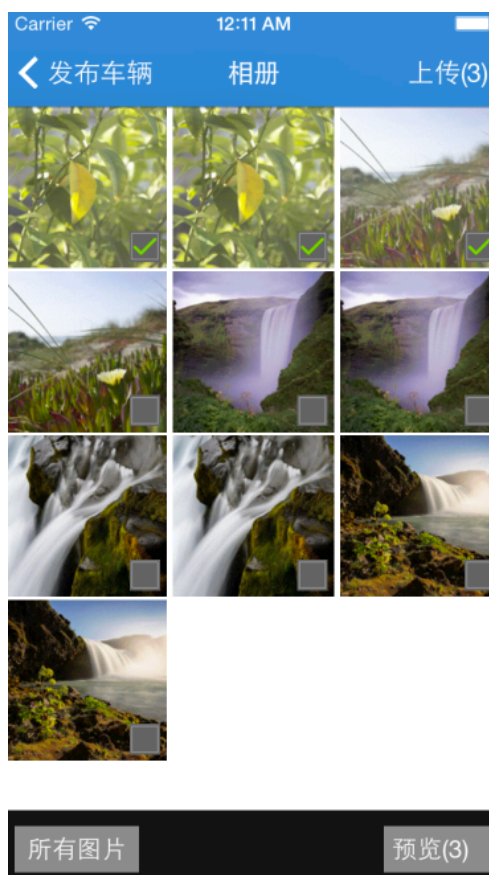


图 4.12 从相册选择照片

值得注意的是，用户有可能选择拍摄一部分照片，而另一部分照片从相册中选择。此时，一种动作结束后，应当记录目前照片的总数，在跳转到另一种动作时传递当前照片的总数，防止照片的数量溢出。

用户单击图 4.12 右下角的预览可以进入到已选照片的预览界面，支持滑动预览单张照片以及删除当前的照片，删除后会通知相关的类做好记录工作。

点击上传按钮后，照片将开始上传，并在发车模块的主界面显示已选择的照片，在该界面下点击任意照片同样会进入已选照片的预览界面。

4.6 信息管理模块的详细设计与实现

信息管理模块是负责管理用户信息的模块，包括管理正在出售的车源，已经收藏的车源，以及管理用户信息等。该模块的设计如图 3.10 所示，根据需求分析和概要设计，详细设计及实现的要点如下：

- a. 可以在一个界面内切换查看出售和收藏的车辆，在切换时无需跳转到新的 view controller，考虑采用 UISegmentedControl 实现，该控件是一个分段控件，类似于 UITabBar，在选择项比较少且无需放置在屏幕底端时较常用，该控件的 UI 如图 4.13 所示。



图 4.13 UISegmentedControl

- b. 查看出售车辆时，点击一个车辆后可以查看车源详情，类似于 4.2.2 中介绍的车源详情。但由于本模块负责的是信息管理，所以将车源详情的底端由联系卖家改为了一系列其他按钮，包括删除、刷新、修改等，如图 4.14 所示



图 4.14 信息管理模块的车源详情页

- c. 在该模块的车源详情下点击删除后，会跳回到该模块的首页，此时要对正在出售的车源进行刷新，以确保刚刚删除的车源不再存在。刷新工作在 `-(void)viewDidAppear` 中完成，

即视图完全加载后执行刷新操作。

- d. 该模块首页的导航栏右侧按钮为全局设置按钮，点击后跳转到全局设置界面，包括查看版本，用户反馈，关于等。用户在该界面可以退出账号，如图 4.15 所示。



图 4.15 全局设置界面

4.7 聊天模块的详细设计与实现

聊天模块是负责用户间相互通讯的模块，该模块负责的功能较为单一，即处理用户间即时通讯的需求。但由于单独实现通信功能需要额外搭建服务器，造成不必要的成本。参考市面上的各种 App 内的即时通讯实现方案后，选择融云第三方即时通讯框架作为本系统的实现方案。

融云即时通讯云可以在应用内支持单聊，群聊，语音通话等服务。在 iOS 应用内使用融云，首先要在融云官网注册开发者账号并申请 App Key，然后集成融云 SDK 到 iOS 应用中。集成步骤如下：

- (1) 在 AppDelegate 中初始化融云 SDK，获取一个 token
- (2) 使用该 token 连接到融云服务器
- (3) 在需要使用地方启动融云会话界面

其中，启动融云会话界面，需要使用融云的会话类 `RCConversationViewController`，并设置实

例变量的 conversationType，支持的类型如表 4.2 所示。

表 4.2 融云会话类的 conversationType 支持的类型

类型名称	类型说明
ConversationType_PRIVATE	私聊
ConversationType_DISCUSSION	讨论组
ConversationType_GROUP	群组聊天
ConversationType_CHATROOM	聊天室
ConversationType_CUSTOMERSERVICE,	客服消息
ConversationType_SystemMessage	系统消息

本系统采用了多种类型应用于不同的场景，在主界面提供了全国聊天室和本地聊天室的入口，可以方便所有的用户进行沟通。在车源详情界面，点击卖家头像可以进行私聊，私聊界面如图 4.16 所示：



图 4.16 私聊界面

聊天模块同时还提供语音通话功能，使用的是融云的 VoIP 点对点语音通话服务，限于篇幅本文将不再介绍。

4.8 工具模块的详细设计与实现

工具模块实际上并不是一个独立的模块，它是很多小工具的集合，为了便于开发和介绍，本系统将其看做一个模块。该模块提供了车辆估价，违章查询，限迁标准查询等多种第三方的工具，以 webView 的形式进行呈现，车辆估价的界面如图 4.17 所示。



图 4.17 车辆估价页面

iOS 中使用 UIWebView 加载网页，所以本模块的关键是在 UIWebView 中读取相应地 URL。由于网络环境会发生变化，所以需要为其设置代理方法，判断不同的读取进度，UIWebViewDelegate 的代理方法如表 4.3 所示：

表 4.3 UIWebViewDelegate 方法列表

方法名称	方法说明
- webView:shouldStartLoadWithRequest:navigationType:	即将开始加载页面
- webViewDidStartLoad:	已经开始加载页面
- webViewDidFinishLoad:	已经完成加载页面
- webView:didFailLoadWithError:	加载页面错误

4.9 本章小结

本章主要介绍了系统的详细设计与实现，根据第三章的需求分析，将系统分为了八个模块，分别是用户登录模块、二手车列表模块、搜索模块、车辆求购模块、车辆发布模块、信息管理模块、聊天模块和工具模块。本章依次介绍了每个模块的详细设计与实现，并在具体的细节部分配有图片和表格说明。

装
订
线