基于iOS的二手车信息平台的设计与实现

摘要

关键词：

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

Abstract

Key words:

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 本文的主要工作内容及意义 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

# 1 综述

## 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机关登记注册，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。

## 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。

## 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：

1. 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。
2. 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。
3. 完成需求分析，并根据需求分析完成概要设计和详细设计。
4. 根据详细设计划分模块，并完成代码编写。
5. 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。

## 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。主要从接口设计，系统架构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。

# iOS开发理论基础

## 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公式的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。

## 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。