



**本科毕业设计（论文）**

**基于安卓平台的社区管理APP开发**

**学 院 机电工程学院**

**专 业 数字媒体技术**

**年级班别 2014级2班**

**学 号 3114000773**

**学生姓名 张福源**

**指导教师 韦宇炜**

**2018年 06 月**

摘要

智慧社区管理平台APP《慧集》采用了Ionic+Angular+Cordova混合开发方式，目前仅兼容安卓平台。此APP作为移动社区管理平台，充分考虑到当前社区管理的信息化、智慧化和数字化的升级状况，通过互联网+的方式来改造建设，从而给社区物业管理者提供一个具有发布信息、智能管理和业务增值的多功能平台，带给社区用户更便捷的渠道地去及时获取信息、智能管理生活和与业主双向沟通。

该平台的客户端作为信息与服务的前端呈现，通过简单的手机号码登录注册，一方面可以实名认证用户，另一方面方便物业管理者与用户双向沟通。另外主页分成信息展示、智慧物管、个人中心三大模块，以简单明了、快速便捷的方式提供服务入口。其服务端采用基于Windows的Node.js服务器，使用JavaScript语言进行服务端开发，而框架方案选择的是Express，同时还引入基于Express框架开发的相关插件模块共同实现服务端的功能。最后使用MongoDB数据库实现将客户端数据通过服务端判断、传入以及对数据库的数据集合进行增删改查的操作。

在APP中，每一位用户登录或注册的操作均将他们的数据以JSON的格式post到服务器，然后查询或新增数据至数据库中，最后返回带有操作状态、信息以及数据等信息返回至客户端。社区用户可以在首页查看所属社区发布的社区公告、社区新闻以及缴费通知，每一条信息都可以点击进去查看其详细的内容、发布时间以及发布者等信息，而社区管理者还可以新增或编辑对应所属社区的社区公告的标题、内容以及图片等信息，另外平台管理员还可以对广告轮播图进行额外的处理。同时智慧物管和个人中心也均通过区分用户的权限让对应的角色进行对应的操作。其中图片等文件资源将存储至服务器文件夹中，而其他数据则存储于数据库集合中。

**关键字：** 移动APP， 混合开发， 移动社区，智慧物管

**Abstract**

The wisdom community management platform APP "huiji" adopts the Ionic+ and Cordova hybrid development mode, currently only compatible with android platform.This APP as mobile community management platform, fully considering the current community management informationization, wisdom and digital upgrade status, through the Internet + way transformation construction, and to the community property managers to provide a release information, intelligent management and business value-added multi-functional platform, bring a community of users more convenient channel to timely access to information, intelligent management and two-way communication with the landlord.

The platform of the client as the front end of the information and service, through a simple mobile phone number registered, log in on the one hand can be real-name authentication of users, on the other hand convenient property managers and users are two-way communication.In addition, the homepage is divided into three modules: information display, intelligent material management and personal center, which provide service entrance in a simple and convenient way.Its server based on Windows Node. Js server, for server development, using the JavaScript language and framework scheme selection is Express, also introduce related plug-in modules based on Express framework development jointly implement server-side functionality.Finally, using the MongoDB database implementation, the client data can be used to judge, pass in and delete the data set of the database.

In the APP, the operation of each user login or register all their data in JSON format the post to the server, and then query or new data to the database, finally return with operating status, information and data and other information back to the client.Community of users can view their communities on the front page news and announcements of the community, community payment notice, each message can click go in to see the detailed information such as the content, the release time and publishers, and community manager can also add or edit the title of the corresponding to the community's community bulletin, content and information such as images, another platform administrator can also extra processing on advertising by graph.At the same time, both the intellectual property management and the personal center also allow corresponding roles to be operated according to the user's permission.The file resources such as images are stored in the server folder, while other data is stored in the database collection.

**Key words:** Mobile APP, Mixed development, Mobile community, Wisdom property

[摘要 1](#_Toc482923700)

[1 绪论 5](#_Toc482923701)

[1.1 课题简介 5](#_Toc482923702)

[1.2 课题研究背景及意义 6](#_Toc482923703)

[1.3 国内外社区管理平台状况 6](#_Toc482923704)

[1.4 课题基本内容 7](#_Toc482923705)

[2 APP《慧集》的介绍 8](#_Toc482923706)

[2.1 《慧集》的基本说明 8](#_Toc482923707)

[2.2 《慧集》的基本使用 8](#_Toc482923708)

[2.3 开发《慧集》所应用的技术 9](#_Toc482923709)

[3 相关开发技术及原理的介绍 9](#_Toc482923710)

[3.1 开发环境 9](#_Toc482923711)

[3.2 开发工具 10](#_Toc482923712)

[3.3 客户端技术及原理 10](#_Toc482923713)

[3.4 服务端技术及原理 10](#_Toc482923712)

[3.5 数据库技术及原理 10](#_Toc482923713)

[4 APP整体设计与开发方案 10](#_Toc482923714)

[4.1 客户端设计方案 11](#_Toc482923715)

[4.2 服务端设计方案 21](#_Toc482923716)

[4.3 数据库设计方案 23](#_Toc482923717)

[4.4 整体开发流程 26](#_Toc482923718)

[5 分析APP的功能需求 27](#_Toc482923720)

[5.1 客户端功能需求分析 27](#_Toc482923721)

[5.2 服务端功能需求分析 28](#_Toc482923722)

[5.3 数据库功能需求分析 28](#_Toc482923723)

[6 详细的开发实现步骤 28](#_Toc482923724)

[6.1 开发环境的配置 28](#_Toc482923725)

[6.2 客户端的开发 35](#_Toc482923726)

[6.3 服务端的开发 28](#_Toc482923725)

[6.4 数据库的开发 35](#_Toc482923726)

[7 程序运行与调试 38](#_Toc482923727)

[7.1 客户端运行与调试 38](#_Toc482923728)

[7.2 服务端运行与调试 40](#_Toc482923729)

[8 程序测试与优化 45](#_Toc482923730)

[8.1 测试结果与问题解决 38](#_Toc482923728)

[8.2 内容与性能优化 40](#_Toc482923729)

[9 APP上线及演示成果 46](#_Toc482923731)

[9.1 APK打包步骤与上线流程 46](#_Toc482923732)

[9.2 演示成果 47](#_Toc482923733)

[总结 49](#_Toc482923735)

[参考文献 50](#_Toc482923736)

[致谢 51](#_Toc482923737)

1. 绪论

##### 课题简介

随着信息技术全球化的不断发展，智能手机与其丰富多彩的APP已经成为人们日常生活的一部分，而其发展的方向朝着更智能化、更信息化、和更网络化。APP为手机用户提供越来越丰富的服务，而手机的升级改造也相应带动APP的体验越来越人性化、智能化。一方面是随着极速带宽无线网络的迅猛发展，另一方面是越来越多的企业以高度的热情投入此领域，每天都有成千上万的APP被开发出来，也因此，人们的工作和生活方式均受到这些APP或多或少的影响。【13】

另一方面，开发APP的技术也逐渐成熟，国内外涌现出一大批新兴的创新的开发技术，作为新型的互联网技术不断带动行业的发展与完善。目前主流的手机系统主要有苹果公司的IOS和谷歌的安卓系统，它们提供了可以与电脑媲美的运行环境，给人们的手机APP体验更丰富。当然在APP的各个设计环节，我们也需要融入更多可扩展性的用户体验和设计理念，使产品不仅能满足各种各层次的用户需求，也能打破一些应用软件见同质化的竞争，满足和符合APP长期发展的条件。目前APP应用大致分类为：工具类、休闲类、生活服务类、社交类和行业应用类，但不管如何，APP的发展理应回归日常生活的常态需求。【14】

本课题服务端以NodeJS来设计，客户端则用基于安卓平台的APP开发，设计并实现一个用于社区管理的智能化平台《慧集》，此平台可满足社区管理者与社区用户之间双向便捷沟通了解，让物业管理模式多样、业务增值，也能使住户们及时获取社区资讯、便捷申请利用社区资源，打造双向平等、互利互助的社区平台，满足当前人们对社区生活信息化、智能化和人性化的需求和愿景。

##### 课题研究背景及意义

本课题是研究基于安卓平台开发的关于社区管理平台的应用程序《慧集》。当前我省创新型现代化产业体系建设步伐加快，新兴产业蓬勃发展，数字经济、人工智能、共享经济等新技术、新产业、新业态加快成长。其中有人大代表提出：建议加速“智慧社区”建设，依托社区管理网络员队伍，通过信息采集平台对线下采集的纸质信息进行登记，并充分利用互联网、云计算、移动APP等新一代信息技术，打造基于信息化、智能化社区管理与服务的新型管理模式。实现对“人、地、事、物、组织、舆情”六要素信息常态化采集上报、流转跟踪、部门数据动态实时更新交换、社区矛盾及时化解、治安事件高效防控，做到“小事不出社区、大事不出街道”。

当前，在智慧社区的投资方中，房地产与物业管理公司的属性基本相同，与业主的日常生活联系紧密，投资运营公司和互联网企业则希望通过提供基于智能硬件入口的生活服务现实围绕社区周边的商业生态模式运营。目前，不同的智慧社区在建设过程中都会建立起自己的智能体系，现有的智慧社区APP五花八门，且质量相对较差。面对潜在的巨大市场，目前用户与社区设备之间并未建立起强关系，移动端好坏参差不齐，业主虽然对资讯、娱乐、教育等有需求，但是却往往得不到满足，社区APP在内容增值服务上存在着巨大的市场。

城市社区网格化管理作为创新社会管理、做好群众工作、加快城市建设的有效途径，具有重大的社会意义和现实意义，比如：

1.实现社区网格化管理是加强社会管理的迫切需要；

2.实行社区网格化管理是提升城市管理水平的迫切需要；

3.实行社区网格化管理是加快社区管理体制机制转变的迫切需要。【6】

本课题旨在提供社区管理者与社区用户间的便捷交流平台，在此基础上增加智慧物业、便民服务和增值业务等功能，以O2O模式让客户在线互动、线上线下一体化的体验，相信《慧集》这类智慧社区管理平台可以真正解决物业管理难、用户获取信息不及时等各种难题，促进城市社区的发展，提高城市社区的管理水平，从而实现中华民族的伟大复兴。

##### 国内外社区管理平台状况

随着云计算、人工智能、物联网及移动互联网等新一代信息技术在社区的应用不断发展，社区将变得更加信息化、智慧化、数字化和人性化。

目前我国的社区管理水平仍处于初级阶段，存在着许多问题，就社区管理平台而言，主要体现在如下几个方面：

物联网技术在社区应用还较少，智慧应用处于初级阶段。随着物联网技术的发展，自动抄表、智能家居等物联网应用走进社区，进入家庭，智慧应用逐渐增多，智能家居成为最有潜力的物联网应用领域之一。

智慧社区应用主要集中在大城市的主要社区。智慧城市建设如火如荼，智慧社区成为智慧城市重要建设内容，但由于受建设的成本和消费的水平影响比较大，也因此，智慧社区的发展还不是很平衡。广州、北京、上海、深圳等沿海城市、直辖市和各省级中心城市发展较快，智慧社区还主要集中在这些大城市的主要社区。

智慧社区产品与技术方案尚不成熟。技术方案选择时存在考虑不全面的情况。

智慧社区建设标准与规划缺乏。对于建成后的系统缺乏相应的验收、测试标准，也没有相关部门组织验收，所以目前计较急需针对这些技术的规范出台推广。

缺乏适应智慧化社区管理与服务的人才。在某些重要岗位不是一般人员通过简单的学习和短时间的培训就能够胜任的，所以社区管理与服务机构特别是社区服务中心与物业管理中心需要配备高素质的技术管理及业务办理人才。【11】

而我国技术与国外相比还存在整体架构不完善，社区信息建设的人才匮乏，社区信息管理的组织机构设置不统一，资金投入捉襟见肘，信息资源的采集与共享和人们对信息及电脑应用技术的程度不高等问题。国外的社区管理平台主要研究物业管理客户满意程度方面，社区都建立了为居民服务的计算机网络，通过将社区分成许许多多网格再进行管理。这样既有综合性，也有专业性。

在未来，随着专家日后研究的方向，也许社区管理平台的形式会发生翻天覆地的变化，从一般客户端、网站表单转变为BBS、网络论坛等形式，这样才能增加居民与管理人员之间的交流，实现双向互动，双向监督，使得我们的社区管理服务工作效率得到提高，从而促进和谐社会发展。【12】

##### 课题基本内容

在本课题中，利用PHP提供的网络编程，在Linux上创建一个服务器用于数据的传输和发送，对客户端的请求指令进行处理，将相关的数据进行对应处理以做到不对服务器及数据库过大负载；通过爬虫技术，对精准获取其他网站相应数据，并转码并打包成JSON数据POST回客户端这项技术要点进行深入的研究。

在客户端方面，利用腾讯地图对用户进行定位，并从服务器获取用户相关的足迹在地图上mark显示出来；同时利用微信语音接口，使用户可以通过语音接口录制自己所处位置时的想要的分享；利用小程序的罗盘功能，为用户提供指南针效果，方便用户知道各个方向,为不熟悉方位的人群提供方便；由于微信小程序开发不支持html代码，需对爬虫所得数据如何转成微信标签代码做一个研究。

微信小程序包的大小不能超过2M，需要对资源进行整理，以做到精简且完整，最后整合服务端和添加一些功能需求，完善小程序。

1. APP《慧集》的介绍

##### 《慧集》的基本说明

程序的类型是作为出行社交类微信小程序，用户可以通过该小程序记录足迹并发表感受，同时其发布的内容可以被用户所在城市的其他用户看到并评论；整个微信小程序除了社交方面内容，还有出行方面的内容，用户可以通过该小程序查看所在城市的攻略，从而减少对该城市的认知感疲乏等感受，同时为用户更好地了解该城市提供了方便。

##### 《慧集》的基本使用

该小程序通过map地图上的controls进行地图定位和打开信息发布的导航栏，再通过导航栏链接到发布界面，在地图mark足迹；通过定位导航查看用户所在位置上其他用户发布的内容。在“我的”导肮栏中由三个栏目——相册、周边攻略、指南针，其中“相册”用于查看用户发布过的信息，“周边攻略”通过定位用户所在城市，获取该城市的吃喝游览等攻略，提供指南针功能，为没有方向感的用户提供帮助。

##### 开发《慧集》所应用的技术

本小程序中，通过为服务端http颁发ssl签名证书，将其转为https，以保证用户访问、用户支付及表单提交的安全；同时利用正则表达将爬虫下来的网页进行归类从而获取有用的信息，且由于微信小程序并不支持html标签，所以仍需要使用正则表达式将爬虫下来的数据信息，转换成微信小程序支持使用的标签；通过 AES-128-CBC算法对微信userInfo数据包中的用户数据进行加解密，从而在保证用户信息安全的同时获取到用户的表层信息，而核心信息将通过sha1方式存储于数据库中以保证数据不被窃取。

1. 相关开发技术及原理的介绍

##### 开发环境

客户端设计主要分成三个大模块，分别是“足迹”模块、“定位”模块、“我的”模块。

1.其中“足迹”模块通过记录更新地图数据来获取用户在每个地方做的记录，同时二级模块用来通过语音、图文、文字发布更新位置；

2.“定位”模块，通过GPS获取用户所在位置的经纬度，再通过腾讯地图的坐标逆运算API获取该位置所在的城市，从而获取后台服务器中该城市用此微信小程序的用户发布的信息。

3.“我的”模块分为三个子模块，其中“相册”模块用于用户查看个人发布的信息，不设地址限制；“周边攻略”通过获取马蜂窝旅游网中的部分攻略，通过GPS定位获取用户经纬度再通过腾讯地图的坐标逆运算API获取该位置所在城市，再从马蜂窝处实时获取该城市相关的旅游攻略，通过正则转码显示在微信小程序中；“指南针”针对北方人群以方位定方向的问题，为不了解方位的用户提供一个准确方位指向的便捷操作。

##### 开发工具

服务端大体功能分为两个，分别时收发数据和处理数据，由于服务器需要处理的数据量相对较多，而且处理的数据总类也各不一样，所以将服务器的数据处理划分为两个功能模块，分别是数据处理模块，以及指令请求模块。

1.数据处理模块：通过curl向相关网站或API获取数据，进行批量处理或正则转换处理，再将其重新发向客户端或者存储到数据库中；

2.指令请求模块：客户端通过特定的请求数据进行封装打包成JSON数据以GET或POST等方法传向服务器，服务器通过对特定的数据格式进行判定，从而接下操作，或与数据库进行交互来获取或插入相关数据，或将客户端传过来的文件存储到特定文件夹中。

同时数据库的结构设计，通过概念结构设计和逻辑结构设计，将客户端传来需要存储的数据进行归类。

##### 客户端技术及原理

1. 首先确定微信小程序的大体功能模块，确定微信小程序的系统需求和功能需求。
2. 对数据库进行结构设计——包括概念结构及逻辑结构。
3. 绘制客户端与服务端相应的功能模块间的关系图。
4. 根据服务器模块之间的关系图，创建服务器的大体框架，并添加相应的功能模块。
5. 根据客户端模块之间的关系图，创建客户端基本框架，并添加相应的功能模块。
6. 为服务端添加SSL证书，调节服务端的可访问性。
7. 调试客户端与服务端之间的数据交互功能。
8. 获取马蜂窝网站数据，归类截取有效数据。
9. 调试程序，重构代码。
10. 整理资源，上线。

##### 服务端技术及原理

1. 首先确定微信小程序的大体功能模块，确定微信小程序的系统需求和功能需求。
2. 对数据库进行结构设计——包括概念结构及逻辑结构。
3. 绘制客户端与服务端相应的功能模块间的关系图。
4. 根据服务器模块之间的关系图，创建服务器的大体框架，并添加相应的功能模块。
5. 根据客户端模块之间的关系图，创建客户端基本框架，并添加相应的功能模块。
6. 为服务端添加SSL证书，调节服务端的可访问性。
7. 调试客户端与服务端之间的数据交互功能。
8. 获取马蜂窝网站数据，归类截取有效数据。
9. 调试程序，重构代码。

##### 数据库技术及原理

1. 首先确定微信小程序的大体功能模块，确定微信小程序的系统需求和功能需求。
2. 对数据库进行结构设计——包括概念结构及逻辑结构。
3. 绘制客户端与服务端相应的功能模块间的关系图。
4. 根据服务器模块之间的关系图，创建服务器的大体框架，并添加相应的功能模块。
5. 根据客户端模块之间的关系图，创建客户端基本框架，并添加相应的功能模块。
6. 为服务端添加SSL证书，调节服务端的可访问性。
7. 调试客户端与服务端之间的数据交互功能。
8. 获取马蜂窝网站数据，归类截取有效数据。
9. 调试程序，重构代码。
10. 整理资源，上线。
11. APP整体设计与开发方案

##### 客户端设计方案

AES算法属于对称加解密算法，其加密和解密过程中所使用的密钥相同，并且解密过程为加密过程的 逆运算。AES算法可支持128比特、192比特和256比特三种不同的密钥长度，不同密钥长度下所需的加解密 轮数也有所区别，但其所加解密的明文块长度被规定为128比特。图4-1所示为AES输入输出数据图，P为AES的明文块，K为输入AES的密钥，C为加密后的密文。

AES加密

AE解密

K

P

C

K

P

C

图4-1 AES输入输出数据图

AES的数据块处理方式分为四个步骤，加密时为字节替换、行移位、列混淆和轮密钥加，解密时为逆 字节替换、逆行移位、逆列混淆和轮密钥加。AES通过轮数定义每一个处理的轮换方式。对于128比特密钥 的AES算法，明文块P在第一轮加密处理前首先进行一次轮密钥加操作。第一轮至第九轮的操作相同，包 括字节替换、行移位、列混淆和轮密钥加四个操作。在加密处理的第十轮，AES算法只执行字节替换、行 移位和轮密钥加三个操作。图4-2和图4-3分别详细描述了加密以及解密的数据处理流程。

第10轮加密轮密钥

第9轮加密轮密钥

第n轮加密轮密钥

第0轮加密轮密钥

明文

初始密钥

字节代换

行移位

列混合

密钥扩展

第1轮加密轮密钥

字节代换

行移位

列混合

字节代换

行移位

密文

图4-2 AES加密的数据处理流程

第10轮加密轮密钥

第9轮加密轮密钥

第1轮加密轮密钥

第n轮加密轮密钥

第0轮加密轮密钥

密文

初始密钥

逆行移位

逆字节变换

逆列混合

密钥扩展

逆行移位

逆字节变换

逆列混合

逆行移位

逆字节代换

明文

图4-2 AES解密的数据处理流程

* + 1. 数据单位

AES算法的数据处理单位为字节，其基本数据单位包括五类：比特、字节、字、块和状态。其中，比特用0和1表示。字节是八个比特的集合，一个行矩阵或列矩阵包含八个字节，作为行矩阵时，数据从左到右按比特排列，作为列矩阵时，数据从上之下按比特排列。字是32bits数据的集合。AES解密数据都是以块为单位进行的，一块文本包含128bits的数据。每一轮中包含多个state状态信息，数据块通过状态的变换 完成加解密，128位的比特信息被分成16个字节，按顺序复制到一个4×4的矩阵中，成为state状态，AES 中所有变换都是基于状态矩阵的变换。图4-3详细描述了数据单位的排列方式。

Byte

Word

Block

State

Byte

b0 b1 b2 b3 b4 b5 b6 b7

b0 b1 b2 b3 b4 b5 b6 b7

Word

B0 B1 B2 B3

B0 B1 B2 B3

b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15

S

S0,0 S0,1 S0,2 S0,3

S1,0 S1,1 S1,2 S1,3

S2,0 S2,1 S2,2 S2,3

S3,0 S3,1 S3,2 S3,3

w0 w1 w2 w3

图4-3 数据单位排列方式

如图4-4所示，为数据单位转换示例图。假设需要进行加密的数据块为“AESUSEAMATRIXZZ”。将 数据块中的每个字母进行十六进制转换，将转换后的数据排列为状态矩阵。

A E S U S E S A M A T R I X Z Z

00 04 12 14 12 04 12 00 0C 00 12 11 08 23 19 19

Text

Hexadecimal

1. 12 0C 08

04 04 00 23

12 12 14 19

14 00 11 19

State

图4-4 数据单位转换示例图

* + 1. 轮密钥加

整个算法处理过程中，该步骤是唯一一个将密钥加入明文的处理步骤。轮密钥加为状态(State)矩阵 与密钥矩阵之间的异或计算，其输入的128比特数据与128比特扩展密钥通过异或计算，输出128比特的轮 密钥加结果。其加密过程与解密过程的轮密钥加过程实现方式完全相同。如式(4．1)和式(4．2)所示， 该运算方式用到了异或运算的可逆特性：若X Y = Z，则Z Y=X

2A 10 AB 2B

AA 3C 4D 0C

5D 34 CA 2E

46 00 5F 6D

07 3C 07 17

ED 3C 12 6C

4D 34 62 6E

ED 4F 12 62

2D 2C AC 3C

47 00 5F 60

10 00 AB 40

AA 4F 4D 2F

07 3C 07 17

ED 3C 12 6C

4D 34 62 6E

ED 4F 12 62

2A 10 AB 2B

AA 3C 4D 0C

5D 34 CA 2E

46 00 5F 6D

2D 2C AC 3C

47 00 5F 60

10 00 AB 40

AA 4F 4D 2F

=

=

(4.1)

(4.2)

* + 1. 字节替换

AES算法如同DES算法一样，都需要执行字节替换操作。不同的是，在AES算法中的字节替换是以字 节为单位执行的。整个字节替换的过程只需要一个s盒来实现，即如果两个进行替换的字节数据是一样的， 则其字节替换的结果也是一样的。除此之外，字节替换也可以通过有限域GF(28)实现。对于S盒的字节替换 实现是一个简单的查找表操作，S盒是由16x16个字节组成的矩阵，包含了'256个数。如图2—13所示，输入 的明文块被分为16个state数据，state中每个字节通过S盒按如下方式映射为另一个字节，将该字节的高四位 作为行值X，低四位作为列值Y，以行列值作为地址索引映射出新的元素作为输出。

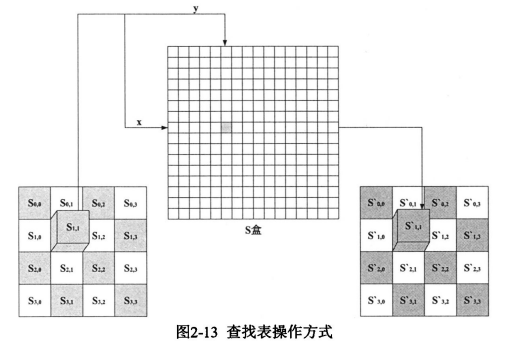


图4-5 查找表操作方式

S盒是可逆的，但并不是自逆的，由正向字节替换S盒无法从替换后的字节替换回原字节，解密时的逆字节代替变换需要用到逆S盒。

* + 1. 行移位变换

行移位变换根据加解密的工作方式的不同可以分为正向行移位变换和逆向行移位变换。经过行移位变 换后，矩阵中的每一个竖列都是由输入矩阵中的每个不同列中的元素组成。在正向行移位变换中，将输入 数据组成的矩阵的第一行保持不变，第二行循环左移一个字节，第三行循环左移两个字节，第四行循环左 移三个字节。如图4-6所示，为正向行移位变换的变换方式。

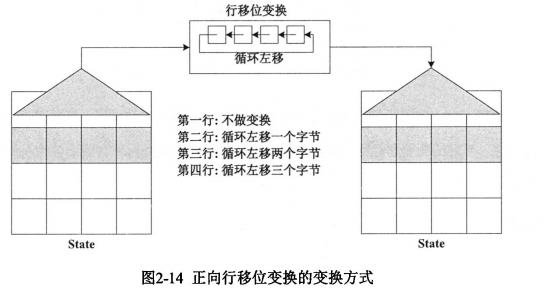


图4-6 正向行移位变换的变换方式

逆向行移位变换与正向行移位变换互为逆变换，逆向行移位变换是将后三行的移位变换依照正向行移 位变换按相反方向执行。

* + 1. 列混淆变换

列混淆指的是通过函数变换，将矩阵中的每一列的值映射为一个新值。在列混淆变换中，每一列的四个字节通过线性变换互相结合。这个变换可通过式(4．3)中的矩阵乘法表示。

02 03 01 01

1. 02 03 01
2. 01 02 03

03 01 01 02

S0,0 S0,1  S0,2 S0,3

S1,0 S1,1  S1,2 S1,3

S2,0 S2,1  S2,2 S2,3

S3,0 S3,1  S3,2 S3,3

S10,0 S10,1  S10,2 S10,3

S11,0 S11,1  S11,2 S11,3

S12,0 S12,1  S12,2 S12,3

S13,0 S13,1  S13,2 S13,3

=

（4.3）

列混淆变换中每一列的元素分别作为1，X，X2，X3的系数，合并为有限域GF(28)的多项式，通过有限 域的矩阵乘法与c(x)=3x3+x2+x+2在modx4+1下相乘，得出变换后的结果，如图4-7所示。

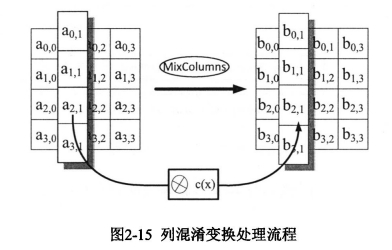


图4-7 列混淆变换处理流程

逆向列混淆变换为正向列混淆变换的逆，可由式(4．4)的矩阵乘法定义。

=

0E 0B 0D 09

09 0E 0B 0D

0D 09 0E 0B

0B 0D 09 0E

S0,0 S0,1  S0,2 S0,3

S1,0 S1,1  S1,2 S1,3

S2,0 S2,1  S2,2 S2,3

S3,0 S3,1  S3,2 S3,3

S10,0 S10,1  S10,2 S10,3

S11,0 S11,1  S11,2 S11,3

S12,0 S12,1  S12,2 S12,3

S13,0 S13,1  S13,2 S13,3

（4.4）

式(4．3)和式(4．4)中的矩阵的系数是基于码字间最大距离的线性编码，这使得state中所有字节之间 有良好的混淆性。列混淆变换和行移位变换使得数据在经过几轮变换后所有输出位均与所有输入位相关， 增加了破解明文的难度。

* + 1. 密钥扩展

对于密钥长度为128bits的AES算法，密钥扩展即将输入的密钥扩展为1l组128为密钥组，其中第0组为 输入密钥自身。每一组扩展后的密钥用于每一轮的实现过程。在扩展密钥中，每一个新增的字w[i]的值都 依赖于,w[i-1]和,w[i-4]。其密钥扩展的具体实现步骤如图4-8所示：

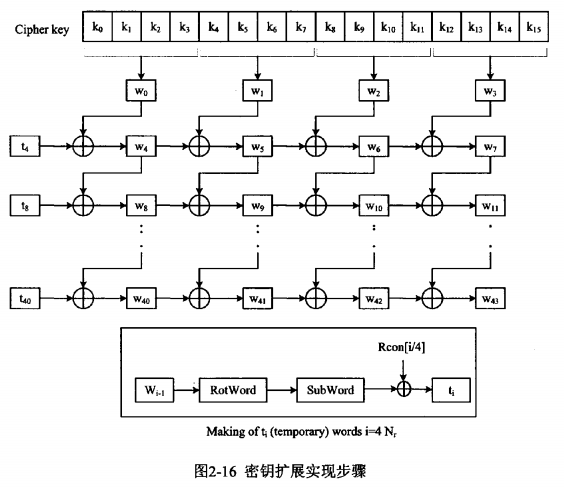


图4-8 密钥扩展实现步骤

将输入的密钥分为16个byte的k0到k15组成w0，以此类推组成w0，w1，w2，w3字组：

在接下来的计算中，如果w[i]中imod 4 0，则w[i] = w[i-1]w[i-4],如果w[i]中i mod 4 = 0,则w[i]=tw[i-4]，其中t由公式（4.5）得

(4.5)

RotWord是将一个字中的四个字节循环左移一个字节，然后将循环左移后的字利用S盒进行字节代换， 将结果与轮常量RCon相异或。如图2．17所示为轮密钥的生成方式。

w

w

B0 B1 B2 B3

B1 B2 B3 B0

S

S

S

S

B1 B2 B3 B4

RCi 000

图4-9 轮密钥生成方式图

轮常量为一个字，其最右边三个字节总为0，表4-1列出了处于不同轮时RCon的数值。

表4-1 RCon数值表

|  |  |  |  |
| --- | --- | --- | --- |
| 轮 | RCon数值 | 轮 | RCon数值 |
| 1 | (01 00 00 00)16 | 6 | (20 00 00 00)16 |
| 2 | (02 00 00 00)16 | 7 | (40 00 00 00)16 |
| 3 | (04 00 00 00)16 | 8 | (80 00 00 00)16 |
| 4 | (08 00 00 00)16 | 9 | (1B 00 00 00)16 |
| 5 | (10 00 00 00)16 | 10 | (36 00 00 00)16 |

* + 1. 有限域GF(28)运算

本质上，一个域即为一个集合，在该集合内，所有算术运算的结果都离不开这个集合。AES算法对于 有限域的性质有很大依赖性，有限域理论知识是理解AES算法设计的基础。AES算法中的所有运算都是在8 位的字节上进行的，所有算术运算也都是名EGF(2^8)上进行的，有限域处理的数据按字节表示为8比特，可以分为b7b6b5b4b3b2b1b0八个元素。有限域GF(28)中两个元素相加，相当于系数之间的模2。因此，其有限域GF(28)中两个元素相加与 两个字节的按位异或结果相同。两个元素相乘为模二元域GF(2)中一个8次不可约多项式相乘。如果乘法运 算的结果为次数大于7的多项式，那么必须将其除以某个次数为8的既约多项式并取余式。其中既约多项式 为：

(4.6)

由8次既约多项式m(x)为模的所有多项式组成的集合满足有限域公理，形成GF(28)有限域。假设GF(2^8) 上的多项式为

(4.7)

将其乘以x得出以下多项式

(4.8)

由公式g(x)可知，若b下标7=O，则结果为小于8的多项式，无需进行进一步计算，若b下标7=l，则需进行用既约多项式m(x)进行约化。一般在GF(2n)上对于n次多项式p(x)，有

(4.9)

由此可知公式(2．8)可根据公式(2．9)推导为如下公式

(4.10)

这表明乘以（0000010）的运算可以通过左移一位后根据条件按位异或（00011011）实现，故当b7=0时，乘以{02}的结果为（b6b5b4b3b2b10），当b7=0，乘以{02}的结果为（b6b5b4b3b2b10）（00011011）。

##### 服务端设计方案

HTTPS是提供互联网上Web服务通信隐私性的安全协议。该协议允许客户端与服务器应用之间进行窃听、防消息篡改及消息伪造的安全Web服务通信。该协议的构造如图4-10所示：

HTTP

SSL/TLS

TCP

IP ICMP IGMP

ARP RARP

HTTPS

应用层

传输层

网络层

链路层

图4-10 HTTPS协议结构图

其中，TCP/IP是整个Internet数据传输和通信所使用的最基本的控制协议。HTTP作为应用层传输协议位于TCP之上，用于从服务器传输超文本到本地浏览器的传送协议。而SSL层位于TCP/IP与HTTP协议之间，作为一种数据安全协议存在，以保证超文本文件的认证性，机密性和完整性。HTTPS协议就是这两个协议的有机结合，通过作为子层的SSL协议来加强安全性，实现HTTP数据安全传输，有效的避免HTTP数据的偷听、篡改及信息的伪造。在整个传输过程中，HTTPS的两个子层HTTP与SSL各司其职。其中，上层的HTTP协议只负责提供或接收HTTP数据包，数据的加密解密工作对其是透明的，并由SSL协议负责。

HTTPS 的工作原理：当客户端浏览器向服务器发出HTTPS请求后，服务器会响应该请求并提供相应内容给浏览器。其中，服务器响应该命令的文字和HTML 文档会在 HTTP 提供的通道中，由会话层套接字（Socket）的连接来传送，Socket 使两台远程的计算机能通过Internet虚拟互连，并进行通话。服务器端的内容在传送出去前，通过本地的SSL协议自动加密，而在客户端接收后，在客户端浏览器上被解密。整个过程，对于没有解密钥的攻击者来说，其中的资料是无法阅读的。

由上述分析可见，HTTPS 协议的安全基础是SSL 协议。

SSL (Secure Socket Layer 安全套接层)协议是由 Netscape 公司开发的网络安全传输协目前 Internet 网上点到点之间尤其是 Web 浏览器与服务器之间进行安全数据通讯所采用的主要协议之一。SSL 采用 TCP 作为传输协议提供数据的可靠传送和接收。SSL 工作在 Socket 层上，因此可以为应用层协议，特别是HTTP 提供透明安全的服务。

SSL 提供的安全服务和TCP 层一样，融合了非对称加密机制和对称加密机制对 Web 服务器和客户端的通信提供保密性，完整性和认证。其中，在 SSL 建立连接的过程中采用非对称密钥机制，协商会话所需的对称密钥，在会话过程中使用对称密钥机制，降低会话过程中服务器的负载。SSL 协议提供的服务包括以下三个方面：

1. 用户和服务器的合法性认证

在 SSL 通信过程中，客户端和用户之间需要使用标准的非对称加密技术和可靠的认证中心（CA）的证书，来确认对方身份的合法性。同时，在连接交换数据的过程中做数字认证，用以保证用户的合法性。

2. 加密数据已隐藏被传送的数据

SSL 协议在握手和会话过程中，采用多种加密机制，以保证数据的机密性，并且经数字证书鉴别，从而防止攻击者在劫持数据后暴力破解。

3. 维护数据的完整性

SSL 协议采用密码散列函数和机密共享的方法，提供完整信息性服务，来建立客户机和服务器之间的安全通道，使所有经过 SSL 协议处理的数据，在传输过程中都能完整和准确的到达客户端浏览器。

##### 数据库设计方案

微信小程序使用了前端技术栈 JavaScript/WXML/WXSS。但和常规的前端开发又有一些区别：

1. JavaScript: 微信小程序的 JavaScript 运行环境即不是 Browser 也不是 Node.js。它运行在微信 App 的上下文中，不能操作 Browser context 下的 DOM，也不能通过 Node.js 相关接口访问操作系统 API。所以，严格意义来讲，微信小程序并不是 Html5，虽然开发过程和用到的技术栈和 Html5 是相通的。
2. WXML: 作为微信小程序的展示层，并不是使用 Html，而是自己发明的基于 XML 语法的描述。
3. WXSS: 用来修饰展示层的样式。官方的描述是 “ WXSS (WeiXin Style Sheets) 是一套样式语言，用于描述 WXML 的组件样式。WXSS 用来决定 WXML 的组件应该怎么显示。” “我们的 WXSS 具有 CSS 大部分特性...我们对 CSS 进行了扩充以及修改。”基于 CSS2 还是 CSS3？大部分是哪些部分？是否支持 CSS3 里的动画？不得而知。
   * 1. 微信运行环境

微信小程序的运行环境应该更类似 ReactNative 之类，而不是纯 Html5。两者最大的不同在于，ReactNative 的界面是由原生控件渲染出来的，而 Html5 的界面是由浏览器内核渲染出来的。

原理上，小程序是如何在微信 App 里运行的呢？

1. 微信 App 里包含 javascript 运行引擎。
2. 微信 App 里包含了 WXML/WXSS 处理引擎，最终会把界面翻译成系统原生的控件，并展示出来。这样做的目的是为了提供和原生 App 性能相当的用户体验。

其运行过程可以简单解释成以下几点：

1. 用户点击打开一个小程序
2. 微信 App 从微信服务器下载这个小程序
3. 分析 app.json 得到应用程序的配置信息（导航栏，窗口样式，包含的页面列表等）
4. 加载并运行 app.js
5. 加载并显示在 app.json 里配置的第一个页面

其完成的运行如图4-11：



图4-11 微信小程序运行原理

##### 整体开发流程

目前爬虫最适合的语言是Python，相对来说PHP的爬虫技术相对比较弱，此刻进行简单介绍。通过curl获取到网页信息，并利用正则匹配提取相关DOM结构，再通过递归方式用关键字符将DOM结构中含有该关键字符的标签类或标签所包含的信息获取出来，进而再通过关键字符获取需要获取的信息，如：图片链接、标签内容、html格式等等。如图4-12所示：

网页

DOM结构

CURL

特定HTML内容

获取完毕

页面内容

正则匹配

正则匹配标签或类名

N

特定内容

正则匹配

获取完毕

Y

N

指定数据

Y

图4-12 正则获取网页特定数据的方式

1. 分析APP的功能需求

##### 客户端功能需求分析

1. 用户指令请求与操作
2. 爬虫获取其他网站的信息
3. 用户数据的管理

##### 服务端功能需求分析

1. 用户数据管理
2. 地图marker内容
3. 用户发布内容
4. 语音录制播放的功能实现
5. html转wml的功能实现
6. 通过微信小程序罗盘API开发指南针功能

##### 数据库功能需求分析

1. 用户列表的逻辑结构
2. 用户发布内容列表的逻辑结构
3. 发布内容评论列表的逻辑结构
4. 详细的开发实现步骤

##### 开发环境的配置

* + 1. 设计思路

服务端主要负责的是客户端的数据传输以及客户端中的一些指令请求给予答复，比如客户端请问获取地图marker信息，客户端发布消息等。

服务器主要分为三个功能模块，分别是数据存储模块、数据处理模块、数据获取模块。

具体的服务端结构图如图6-1所示：

服务端

客户端

数据传输

数据存储模块

数据处理模块

指令请求模块

数据获取模块

其他网站

图6-1 服务端结构图

* + 1. 服务器功能模块分析

指令请求模块

客户端通过设计相应的数据请求类型及指令请求格式，通过报头数据的插入进而对服务端进行请求及数据传输，服务端对该请求进行相应的应答，从而达到相应的数据交互。该模块的功能流程图如下：

指令请求模块

客户端

发送请求指令

数据接收处理

处理线程

判定解析指令

数据打包发送

图6-2 指令请求模块流程图

数据存储模块

数据存储模块是服务端最重要的模块，用于负责对小程序用户的信息及其他数据进行存储处理，其处理方式是：将客户端发送过来的数据包进行判定归类，将需要存储的数据获取出来，将其进行整理存储入数据库，若存储成功则发送处理结果，若存储不成功则返回错误。

数据存储模块

客户端

发送数据包

发送处理结果

处理线程

判定数据包格式

存储特定数据

图6-3 数据存储模块流程图

数据处理模块

服务端对客户端发送过来的数据进行判定整理，若数据存在问题则返回错误代码，若没有问题将对数据进行整理，并按请求数据类型及格式打包发送相应的数据内容。该模块的功能流程图如下：

图6-4 数据处理模块流程图

数据处理模块

客户端

发送请求数据

数据接收处理

处理线程

判定数据

数据打包发送

数据获取模块

数据获取模块是按照客户端请求的指令，爬取其他网站相应网页内容，在将该网页中客户端需要的内容通过正则匹配递归获取出来并打包传输回客户端，整个流程需要耗费比较大的服务器资源，同时由于比较繁琐，所以需要写一个关于数据爬虫的类。

数据获取模块

客户端

发送请求

数据接收处理

处理线程

获取其他网站内容

数据打包发送

解析获取相应内容

图6-5 数据获取模块流程图

* + 1. 相关功能类设计

数据库连接类是服务器连接数据库最主要的类，该类通过继承MySQLi，将可对数据库进行所有可操作的内容。该类的构造如下：

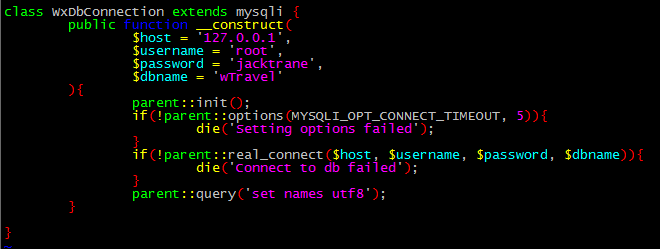


图6-6 WxDbConnection类构造

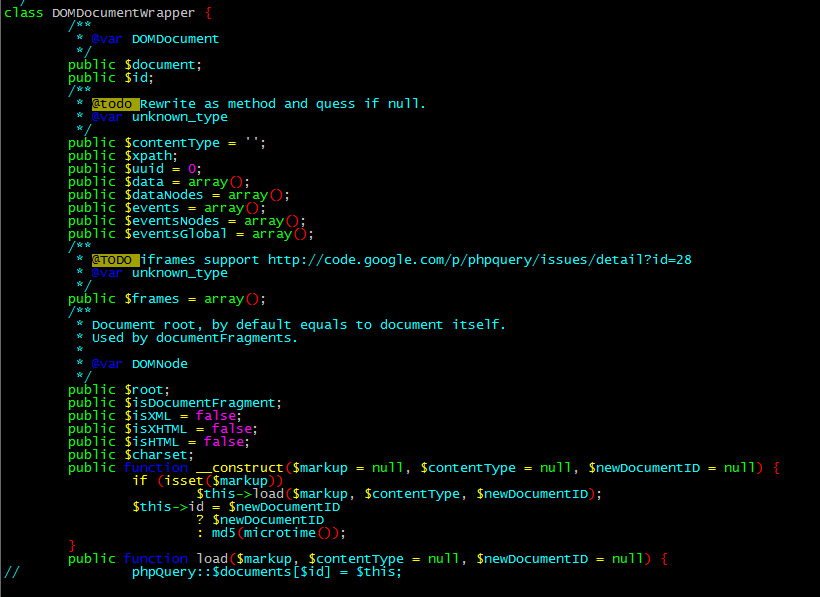
DOM结构类用于正则爬虫时获取特定内容的类，通过将DOM的标签结构、类及其他属性内容进行存储，然后通过调用特定的结构来爬取特定的内容，其大致构造如下：

图6-7 DOMDocumentwrapper类构造

* + 1. 数据库设计
       1. 需求分析

需求分析

用户角色管理

发布信息管理

信息评论管理

用户所在城市

发布的内容

图6-8 需求分析图

用户角色管理：用户的注册

发布信息管理：发布的信息主要要有内容、发布的城市、发布人

信息评论管理：用户对于发布内容进行评论

* + - 1. 概念结构设计

用户

信息发布

内容回复

自增ID

自增ID

自增ID

用户openid

昵称

图片

城市

省份

性别

发布人

评论人

内容

点赞

回复人

内容

图片链接

语音链接

经度

纬度

城市

用户ID

图6-9 概念结构图

* + - 1. 逻辑结构设计

用户关系模型 wx\_user(id, openid, user\_name, user\_img, city, province, gender);

用户发布模型 wx\_user\_publish(id, content, img\_url, sound\_url, latitude, longitude, city, user\_id);

用户评论模型 wx\_publish\_comment(id, pub\_id, content, great, comment\_id);

如图所示：

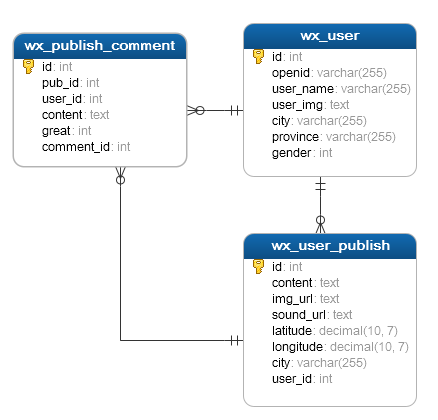


图6-10 逻辑结构图

* + - 1. 主要SQL语句

|  |
| --- |
| INSERT INTO `wx\_user\_publish`(`content`, `img\_url`, `latitude`, `longitude`, `city`, `user\_id`) VALUES('内容', '链接', '经度', '纬度', '城市', '用户id');  SELECT \* FROM `wx\_user\_publish` WHERE `city`='城市';  INSERT INTO `wx\_publish\_comment`(`content`, `pub\_id`, `user\_id`) VALUES('内容', '发布id', '用户id');  SELECT \* FROM `wx\_user\_publish` WHERE `user\_id`='用户id'; |

##### 客户端的开发

* + 1. 设计思路

客户端主要负责将数据前端呈现化出来，客户端设计主要分为六个模块， 其中地图接入模块通过接入腾讯地图接口，获取服务端中数据库内的mark数据，将其打包传回前端进行数据解析显示；罗盘模块则是利用微信小程序提供的罗盘API进行调整设置为指南针功能；文件上传下载模块将客户端内的文件上传的服务端在通过文件处理存储在相对应的文件夹位置，而服务端文件下载则是运用微信小程序downloadFile功能将其下载到本地进行展示；语音录制模块则是通过微信小程序提供语音录制接口录制相应的语音，同时利用播放接口播放语音信息，在确认无误后将其传入服务端，通过文件处理存储在相应的位置；数据处理发送模块主要用于用户发布内容处，用户通过输入相应内容，再通过JSON打包数据，进而POST向服务端，有服务端进行数据解析，从而做出相应应答；数据处理解析模块则是从服务端处获取相应的数据包，再将其解析以前端呈现。其功能模块流程图如下：

客户端

服务端

数据发送

数据解析

地图接入模块

罗盘模块

文件上传下载模块

语音录制模块

数据处理发送模块

数据处理解析模块

文件处理

图6-11 客户端模块流程图

* + 1. 客户端功能模块设计

1. 数据处理解析模块

该模块与数据发送模块相辅相成，将服务端的数据传回客户端，再由客户端解析显示在页面上

1. 地图接入模块

通过接入腾讯地图API，将地图在客户端进行呈现，同时利用腾讯地图的marker功能，将用户发布的信息直接显示在页面上，同时通过该API的坐标逆解析可以将经纬度直接转换成城市，让接下来的功能更加容易做。

1. 罗盘模块

罗盘模块通过调用微信小程序的罗盘接口，进而直接实现指南针功能；

1. 文件上传下载模块

通过uploadFile与downloadFile的功能，向服务端上传与下载相应文件指令，再由服务端判定，进而达到功能完成的目的

1. 语音录制模块

该模块可让用户直接使用录制功能，将其声音录制成文件发送向服务端。具体实现如下图所示：

语音录制模块

录制

暂停

录制完毕

播放

暂停

播放完毕

服务端

文件处理

数据返回

数据发送

客户端

图6-12 语音录制模块流程图

1. 数据处理发送模块

数据处理发送模块与数据解析模块相辅相成，通过将客户端的数据进行打包传输向服务器，再由服务端进行判定与反馈。

* + 1. 相关功能模块类设计

语音录制模块中，需要对录制的语音进行计时处理，通过以下函数进行计时：

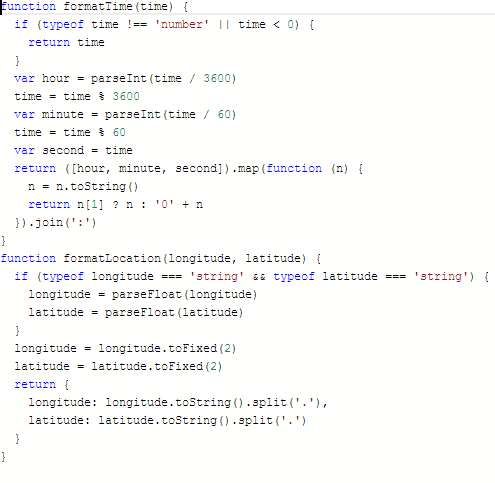


图6-13 计时类

数据解析模块中，服务端通过爬虫技术获取到的数据内容展示在客户端微信小程序中是有难度的，因为微信小程序有自己独立的标签wml，所以普通的html标签无法在其中进行渲染显示，这时候要通过正则将关于div、h1、p、img等标签内容全部转成view、iamge等标签，类如下图：

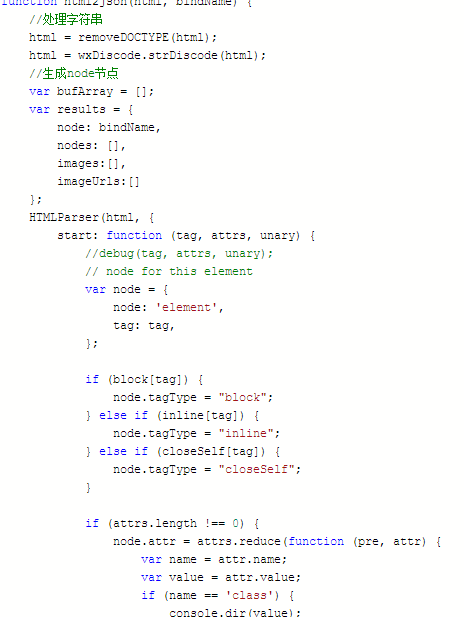


图6-13 解析类

##### 服务端的开发

* + 1. 设计思路

客户端主要负责将数据前端呈现化出来，客户端设计主要分为六个模块， 其中地图接入模块通过接入腾讯地图接口，获取服务端中数据库内的mark数据，将其打包传回前端进行数据解析显示；罗盘模块则是利用微信小程序提供的罗盘API进行调整设置为指南针功能；文件上传下载模块将客户端内的文件上传的服务端在通过文件处理存储在相对应的文件夹位置，而服务端文件下载则是运用微信小程序downloadFile功能将其下载到本地进行展示；语音录制模块则是通过微信小程序提供语音录制接口录制相应的语音，同时利用播放接口播放语音信息，在确认无误后将其传入服务端，通过文件处理存储在相应的位置；数据处理发送模块主要用于用户发布内容处，用户通过输入相应内容，再通过JSON打包数据，进而POST向服务端，有服务端进行数据解析，从而做出相应应答；数据处理解析模块则是从服务端处获取相应的数据包，再将其解析以前端呈现。其功能模块流程图如下：

##### 数据库的开发

* + 1. 设计思路

客户端主要负责将数据前端呈现化出来，客户端设计主要分为六个模块， 其中地图接入模块通过接入腾讯地图接口，获取服务端中数据库内的mark数据，将其打包传回前端进行数据解析显示；罗盘模块则是利用微信小程序提供的罗盘API进行调整设置为指南针功能；文件上传下载模块将客户端内的文件上传的服务端在通过文件处理存储在相对应的文件夹位置，而服务端文件下载则是运用微信小程序downloadFile功能将其下载到本地进行展示；语音录制模块则是通过微信小程序提供语音录制接口录制相应的语音，同时利用播放接口播放语音信息，在确认无误后将其传入服务端，通过文件处理存储在相应的位置；数据处理发送模块主要用于用户发布内容处，用户通过输入相应内容，再通过JSON打包数据，进而POST向服务端，有服务端进行数据解析，从而做出相应应答；数据处理解析模块则是从服务端处获取相应的数据包，再将其解析以前端呈现。其功能模块流程图如下：

1. 程序运行与调试

##### 客户端运行与调试

服务端调试主要调试各个功能模块的运行情况，并根据反馈来获取程序运行情况。调试情况如下

地图marker列表调试如图7-1所示：

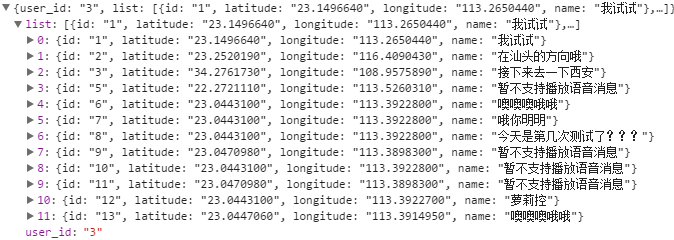


图7-1 地图marker列表调试图

通过获取marker数据，之后在微信小程序中，以腾讯地图marker的方式显示在地图上，因腾讯地图marker数据不支持语音和图片，所以无法在地图上显示语音播放及图片显示。

通过经纬度获取用户所在城市调试如图7-2所示：

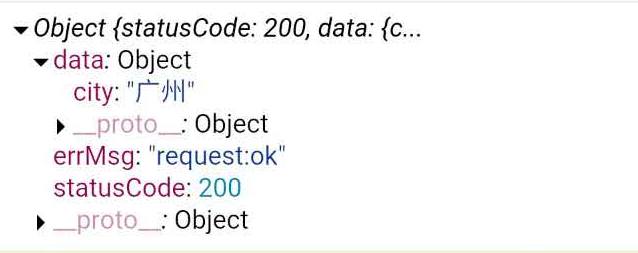


图7-2 获取用户所在城市

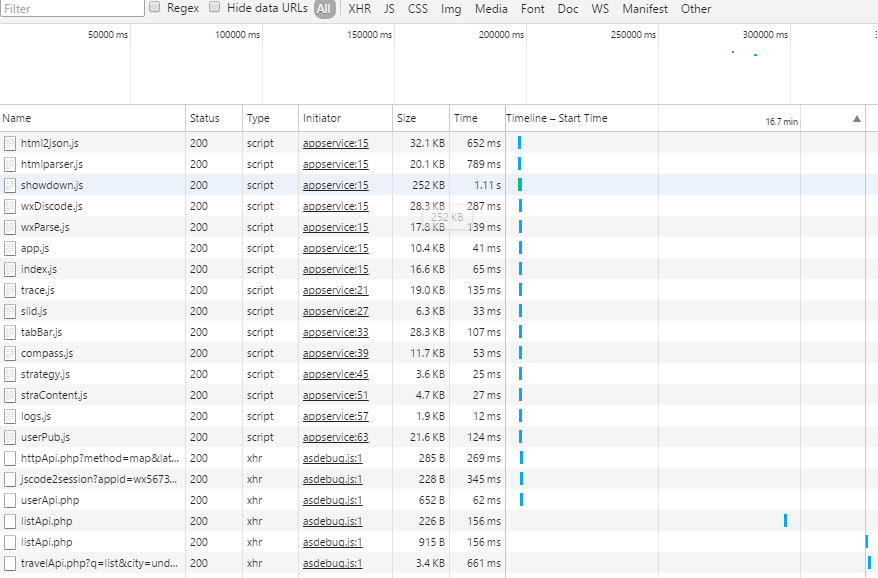
通过经纬度获取用户所在城市，用于接下来的用户发布，周边攻略，“定位”模块的消息展示。

发布信息的数据提交如图7-3所示：



图7-3 用户发布信息的数据提交

程序需要调用到的网络数据调试，如图7-4所示：

图7-4 网络数据调试

微信小程序有特定的代码编辑工具，通过该工具可以直接查看开发的微信小程序的数据链接、console等方面的东西。

##### 服务端运行与调试

“足迹”模块的UI调试如图7-5：



图7-5 “足迹”模块的UI调试

“我的”模块UI调试如图7-6：



图7-6 “我的”模块 UI调试

“我的相册”模块的调试如图7-7：

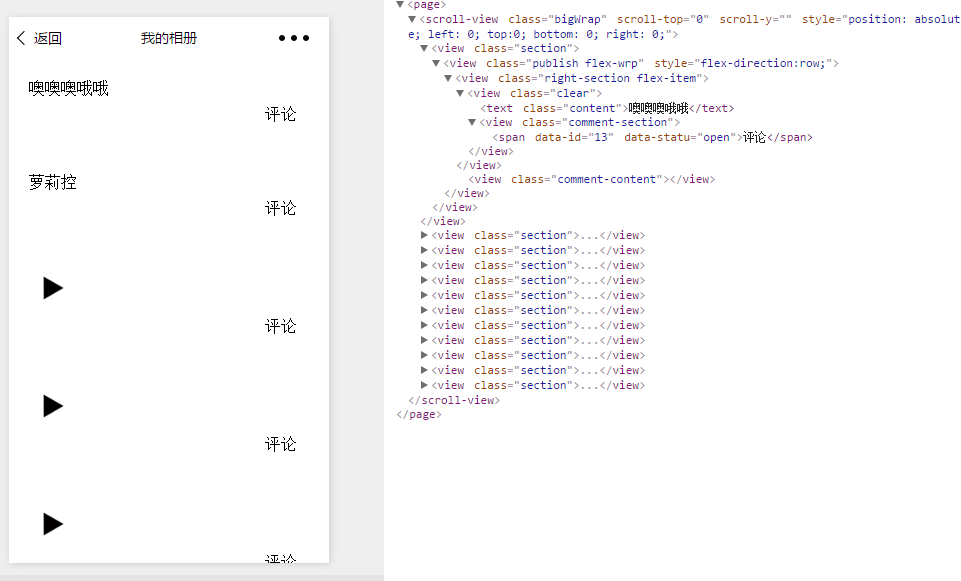


图7-7 “我的相册”模块调试

“周边攻略”模块的调试如图7-8：

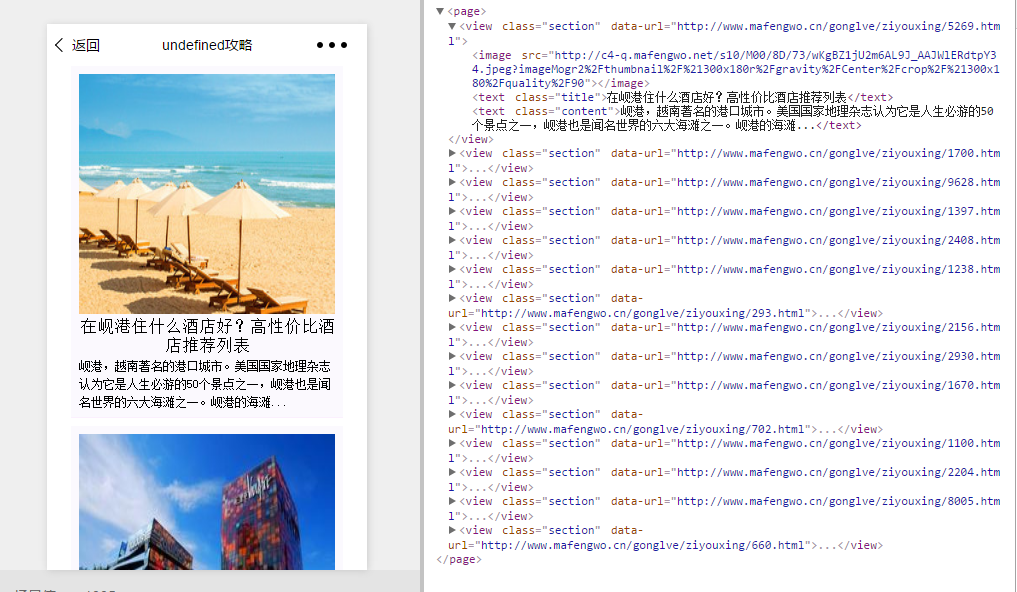


图7-8 “周边攻略”模块调试

“攻略内容”模块的调试如图7-9：



图7-9 “攻略内容”模块调试

“指南针”模块的调试如图7-10：

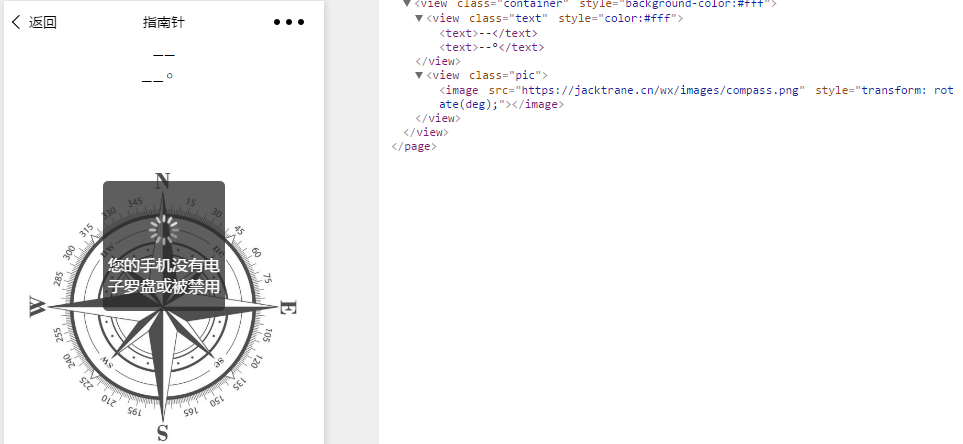


图7-10 “指南针”模块调试

“语音发布”模块的调试如下：

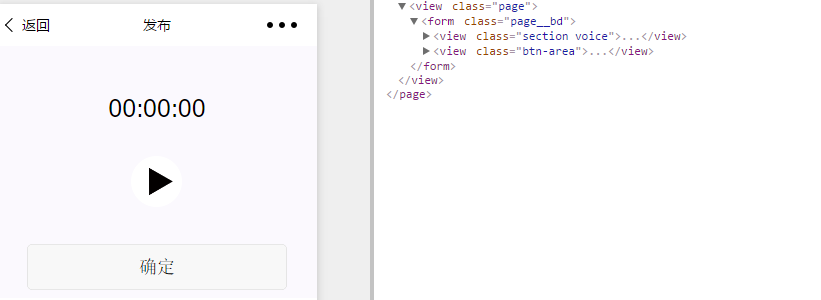


图7-11 “语音发布”模块调试（1）

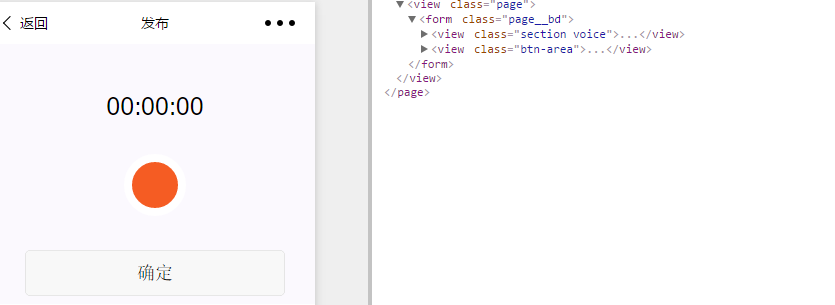


图7-12 “语音发布”模块调试（2）

图7-13 “语音发布”模块调试（3）

“图文发布”模块的调试如下：

图7-14 “图文发布”模块调试（1）

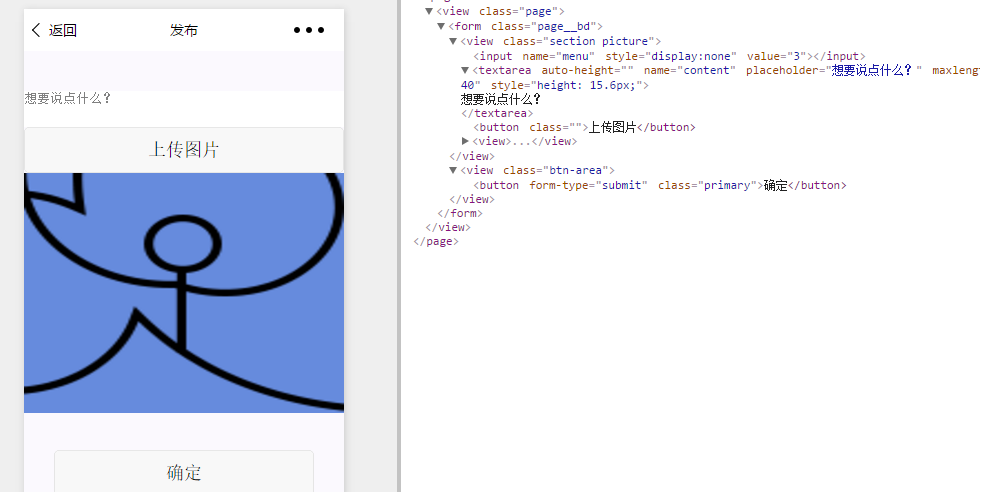


图7-15 “图文发布”模块调试（2）

由上述的调试监视图可知，在模块完全运行时并得到相应的数据后，并没有占用过高的网络资源，说明的数据资源方面优化得可以接受。但数据监视器需要消耗较大部分的CPU性能，同时容易被不法分子窃取相应数据进行攻击，所以在不需要的时候，必须要关闭数据跟踪功能，防止不必要的损失。

1. 程序测试与优化

##### 测试结果与问题解决

“足迹”模块的UI调试如图7-5：

1. 优化代码，将可封装的代码进行封装，减少代码冗余；
2. 优化资源，将资源最大化放上服务器或托管给七牛云，减少打包内容过多而导致的打包不成功，注：微信小程序大小最大不能超过2M；
3. 优化界面，减少不必要的操作，使界面更加爽朗，更适用；
4. 优化数据库，减少冗余字段
5. 提交代码，上传审核
6. 审核成功，《顽途》小程序上线

##### 内容与性能优化

“足迹”模块的UI调试如图7-5：

1. APP上线及演示成果

##### APK打包步骤与上线流程

图9-1 首页 图9-2 菜单 图9-3 录音



图9-4 播放并发布

##### 演示成果

图9-6 定位 图9-7 评论

总结

大四上学期的学期末，在与导师的讨论中确定了我的论文题目《基于微信小程序的社交应用开发》，因为这样的话，开发完之后上线还可以让同学朋友体验一把，但当时微信小程序还没有开放个人开发，所以在确定题目之后，也开始寻找代替该题目的另外一种开发方式，当然是备选方案，因为可以的话还是要继续使用微信小程序进行开发的。

那时候一直在寻找需求，因为我要做的小程序是出行社交类的，所以也开始去找这一类的app，还有微信小程序，比如：去哪儿、足迹等，后来发现大部分的出行社交类app要么只是专注于出行，比如：去哪儿、携程、同程旅游等，要么比较专注于社交方面的内容，比如：旅咖、蚂蜂窝等，要么是专注于记录足迹的，如：我的路、世界迷雾等。突然在想，如果可以把这几样给进行一个精简集中，然后放在微信小程序里，或许可以方便很多户外出行爱好者；在与同学朋友讨论之后觉得这挺适合的，所以开始来做，但是后来发现如果想要将吃住这一块的东西统一进行实际上是不科学，不说处理难度的问题，关于利益这一块的东西，还是暂时不去接触，免得引起纠纷；所以决定将后面那两个集合一起，做一个有攻略、有社交、有足迹的小程序，不大够用就行。

选择永远存在，不管是论文议题还是开发形式，都要好好选。由于确定的毕设是做安卓软件，而本人对于安卓开发只是一知半解，那么问题就来了，如何让一个刚入门的人在短短的几个月的时间内开发出一个安卓软件，这个太冒险了，但好在方法总是比难题多，接下来我选了两个备选方案：ionic开发、微信小程序开发，也得到导师的认可，但由于微信小程序还不能个人开发，所以退而求其次，我使用ionic这个h5打包工具进行开发，但这个开发的话，直接导致的结果是，没办法简单地让用户去使用，一个很工具的工具软件，还要费心用户去寻找下载安装，太繁琐了，但没办法，只能这么做了。

经过一个多月的ionic开发，然而微信小程序终于开放个人开发，于是在学了半个月ionic开发之后又转向学习微信小程序的开发，确实很吃力，这样导致的结果是前面的开发基本废掉了，除了思想已经组织好了只要，开发回到原点，重新出发，这样也就意味这开发周期缩短了一半，但没办法，熬几个夜，硬着头皮做下去吧，终于在2017年5月10号左右完成了小程序的开发与测试，为了尽快赶工，以至于缩减了许多当时构思的功能，比如地图上的轨迹；同时发现，微信小程序的开发实际上还没有成熟，太多本身存在的bug了，比如语音播放要点两次才能够播放、无法进行DOM操作将新的内容插入wml中，目前还没有找到解决的方法。

剩下的时间就是论文的编写，15000字的论文还有5000字的英翻中文献，着实比较吃力，所以在收集资料的过程中，我利用有道云笔记，将该记录的东西全部保存下来，好查找，遇到问题便跟舍友讨论，互相解决问题，论文也慢慢成型了。

我相信，在这几个月的毕业设计中，可以学到很多，耐心、信心等，不要害怕困难，因为世上只要有问题就一定有解决问题的方法，所以一定要胆大心细，不怕问题，同时要听取多方意见，综合取最优，相信每个人的意见都有一定的作用。

参考文献

[1]百度百科. 马斯洛需求层次理论

<http://baike.baidu.com/link?url=BPU5h6Hraf0_rP5IoFwkMUhpi2lTd0tzVMLq8j9jqqj6DCE-D5VUa0216J85ABWZtcE58WEzSCjQfX3hjAQXwqL6FTSJ8AgyfbCuj7pPIVPVmcGBPaB2abTgYWlOkZERIjZ6SKsH6Zj2MfQ5PBPek8JMhg3lST6wTxqz6mkWQAdMDwH4VZqWXW4zALTwi_df>

[2] 光谱.Instant Apps 是什么？为什么它让 Android 开发者欣喜若狂？

<http://www.pingwest.com/what-is-instant-apps-why-is-it-so-good/>

[3]卢刚. 用于IPSec协议的AES-128-CBC算法高速硬件设计[D]. 东南大学, 2015.

[4]张宝玉. 浅析HTTPS协议的原理及应用[J]. 网络安全技术与应用, 2016(7):36-37.

致谢