**排队信息共享ＡＰＰ的设计与开发**

* 1. **引言**

随着计算机技术的不断发展和智能手机的逐渐普及，高科技产品已经慢慢地进入到了每个人的生活当中，特别是智能手机的发展，给人们的生活带来了极大的便利。人们可以通过类似酷狗音乐、网易云音乐等手机APP来听音乐，放松自己；还可以通过类似QQ、微信等APP来进行社交活动；当然了，还可以通过百度地图、铁路12306来方便出行；通过手机淘宝、WPS Office、钉钉来方便日常的办公及生活服务。但是，随着生活节奏的加快，人们对于时间的管理要求也越来越严格，因此，需要一款手机APP来帮助人们了解居住地附近的饭店、银行、医院、景点等地方的人流量及排队情况，这样能够大大节约人们的出行时间，方便人们的时间安排。

* 1. **系统任务概述**

这里我们将该排队APP的应用范围定义在任意地区，支持全国，将应用对象定位在携带有智能手机的全体居民。用户可以通过该APP管理自己的账号信息；发送营业场所的排队情况；查询自己关心的场所的拥挤情况；该系统还具有各营业场所的经营分析功能，如不同时段的人流量变化情况\忙闲预测\拥挤度排名等；系统可根据实时拥挤情况向用户推荐合理的出行时间安排。

* 1. **可行性分析**

（1）政策支持：随着“大众创业，万众创新”口号的号召，越来越多的人加入到创业创新的队伍当中。在新经济增长模式下，创业创新离不开银行服务。在当前互联网发达的快节奏时代，人们越来越多地期望“足不出户”就可以了解到自己关心的地点的拥挤度情况，从而节省自己的时间。而福州地区近几年来自主创业人数剧增，对于银行等服务业的需求量也急剧加大；

（2）当下社会环境的需求：当下社会属于互联网时代，人们的生活节奏快，压力大，需要合理的规划自己的时间。该APP可通过实时的拥挤度分析，帮助人们节省大量时间。对于平时人流量比较大的地方，像银行、饭店、医院等地方，通过该APP，也可有效缓解这些营业场所的拥挤情况，提高服务效率。在当下旅游业发达的时代，人们可通过该APP合理安排自己的旅游出行计划；

（3）从推广角度来看，由于该APP适用于于特定地区的所有Android智能手机用户，用户数量大；对于使用该APP的普通用户与营业场所的工作人员来说，双方都能够实现利益最大化，有利于软件的推广。

* 1. **目标群体**

所有持有Android 4.0.3以上系统安卓智能手机的的营业场所的工作人员或者需要查询营业场所排队情况的客户。

**2.1 APP功能需求分析及模块设计**

**1.个人信息管理模块**

* 注册与登录

每个用户有一个ＩＤ唯一的用户名和密码，用户凭此登录系统。用户可以分为两类：一类是普通用户，负责营业场所信息的查询与使用；另一类是商家用户，负责营业场所排队情况的上传及日常使用。用户注册时可用手机号进行注册，录入信息，同时手机号用来用户信息的增、删、改、查的验证；

* 申请认证成为商家

每个普通用户可以通过申请，填写必要的个人及营业场所信息，申请认证成为商家用户。

* 信誉度评价

给每个商家用户加入信誉分统计功能。类似淘宝的DSR功能，每位普通用户可以对商家服务进行满意度打分，然后根据分数，再计算得出商家的信誉分，为顾客的服务提供参考。

* 实时聊天功能

商家用户和普通用户均可通过实时聊天功能进行实时聊天，方便及时掌握营业场所的排队情况。

**2.查询及交易模块**

* 市场模块

当用户通过登陆界面进入主界面后，系统的首页界面显示的就是该市场模块，该模块获取所有的商家的信息并进行分类汇总后以列表的形式将商家的简要信息（如：ＩＤ，位置信息，信誉等级等）显示在主页界面，并实时更新。提供多种排序方式，如：按距离，拥挤度，信誉分等。用户可以根据自身需要选择排序方式。

* 搜索功能

提供两种搜索功能，分为按关键字搜索和按标签进行搜索。

* 定位功能

该APP加入定位功能，实时获取用户的位置信息。方便对于用户附近的营业场所进行筛选与推荐。

* 关注功能

增设关注功能，用于收藏用户关心的营业场所，能够及时将这些场所的排队情况推送给用户以及方便用户进行快速查询。

* 营业场所经营分析

该模块能够根据系统中已经获取的当前排队人数信息和即将到达的人数信息来预测特定时间段的该场所的排队状况。并能够展示该不同时间段的人流量情况，拥挤度排名情况等。

* 评论功能

用户可以在该营业场所的下面发表自己的评论或者回复他人的评论，点击“写评论”即可以图文的形式进行发表。

**3.排队&预约功能模块**

* 营业场所预约排队

用户可以点击预约排队按钮，将自己即将要出行的营业场所的预约信息，如：场所名称，到达时间等加入到数据库中。

**４.系统管理模块**

* 用户管理

主要负责用户（包括普通用户和商家用户）的管理，含有用户个人信息、关注地点、用户信誉度、评论等的管理。

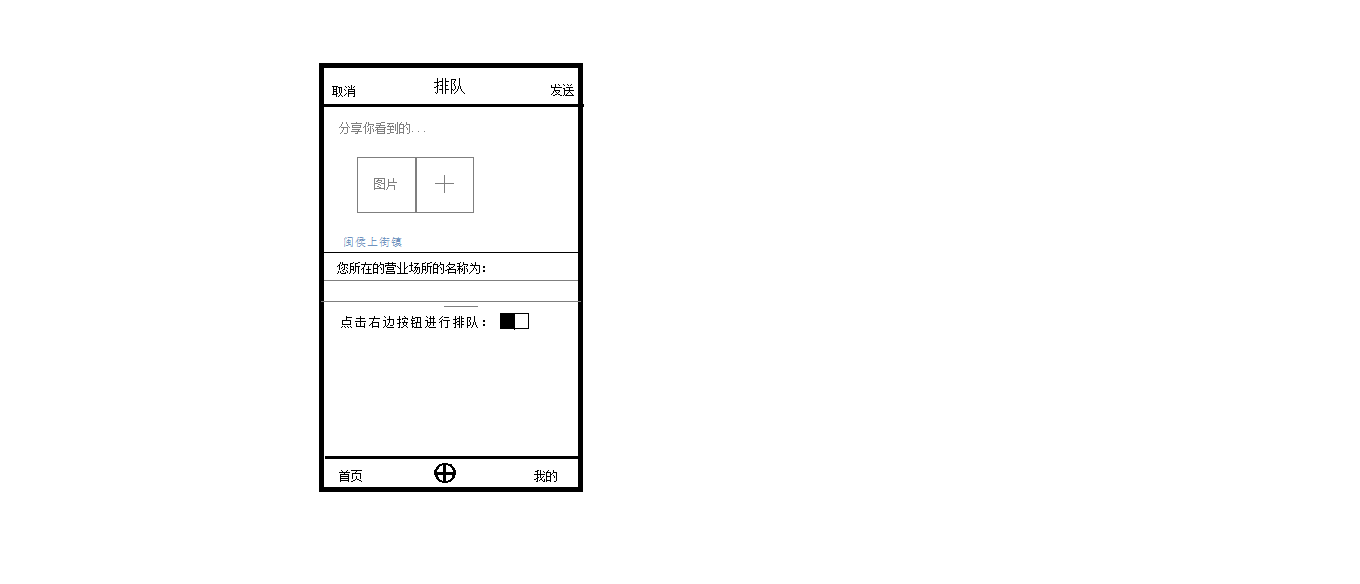
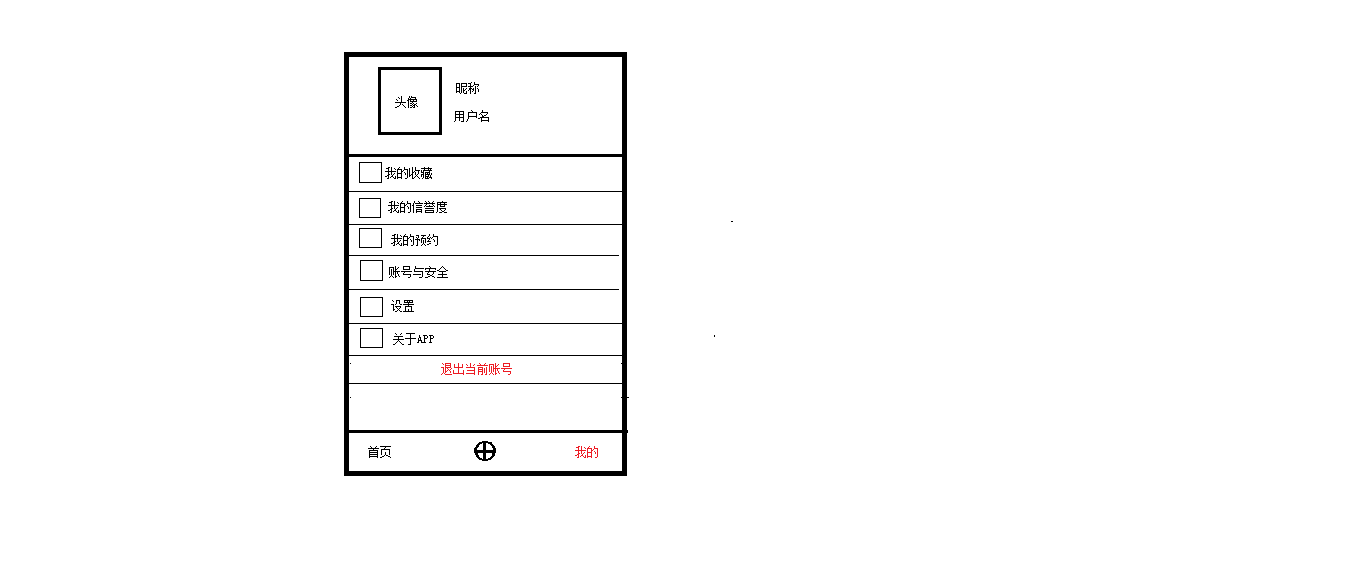
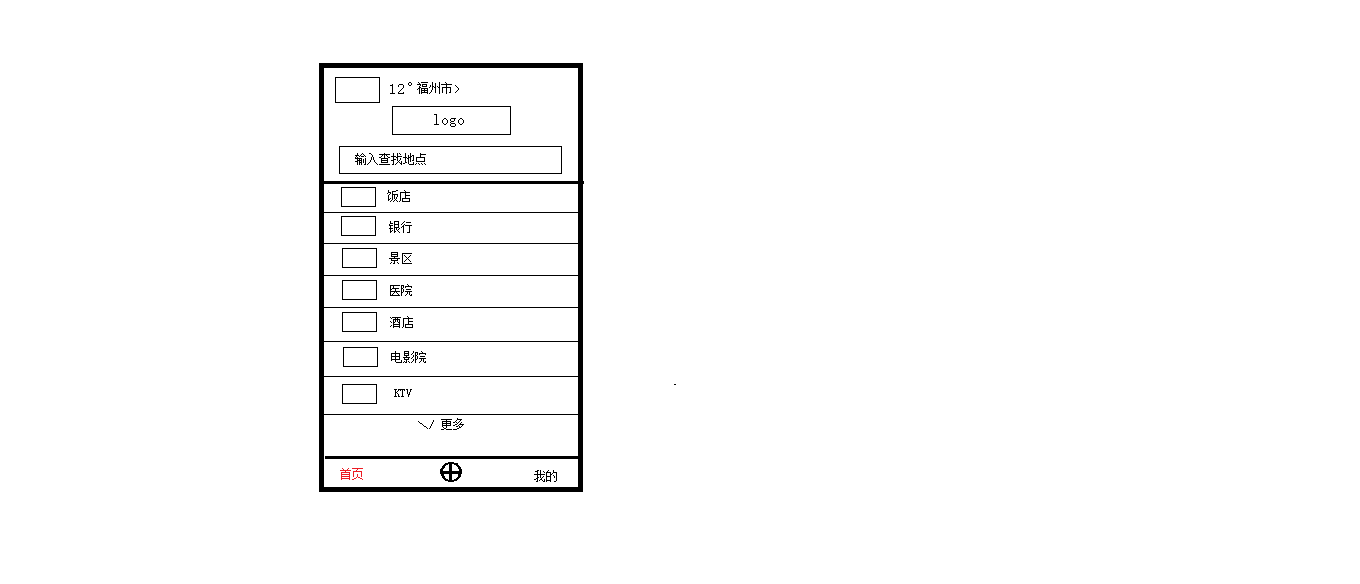
* 信息管理

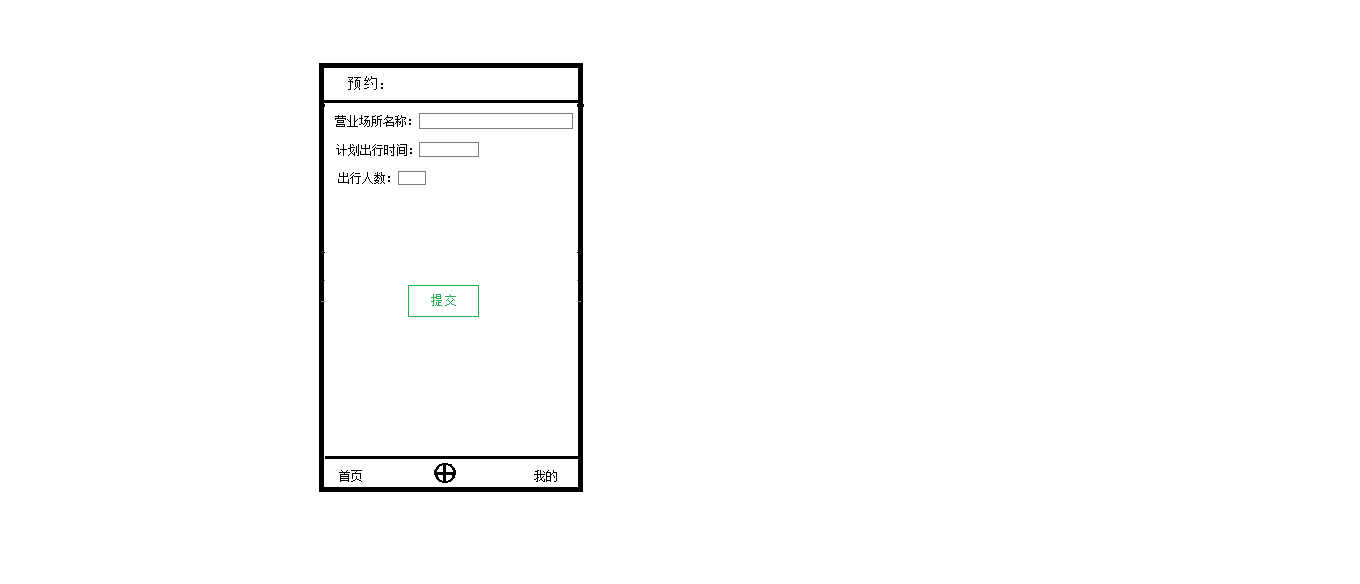
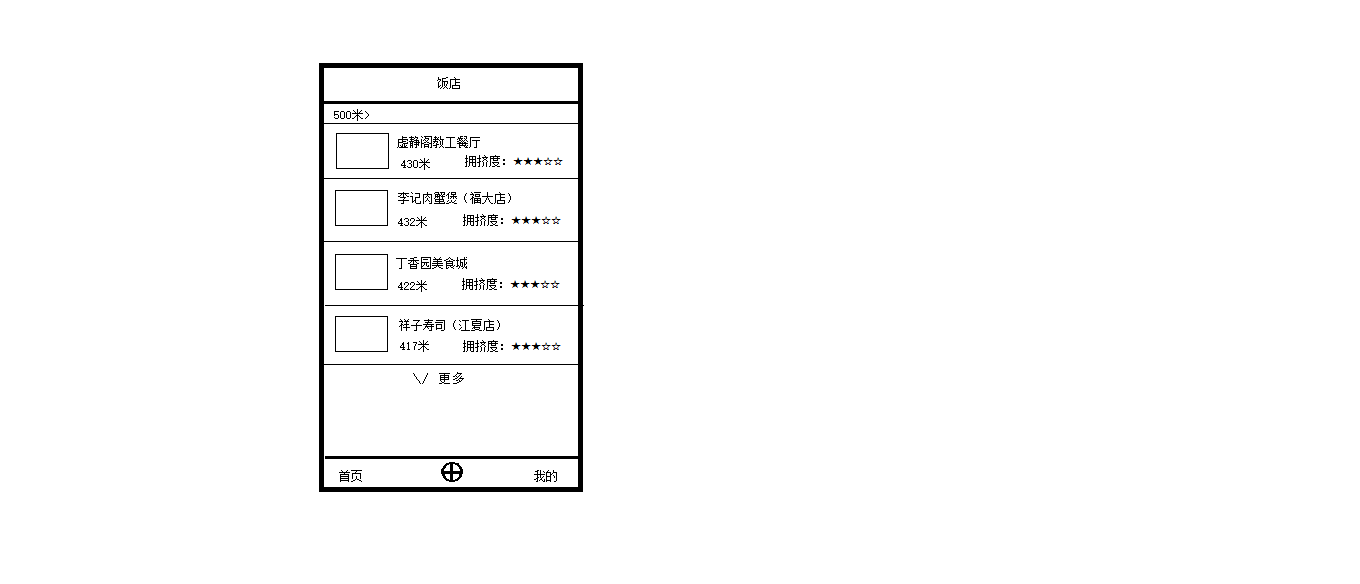
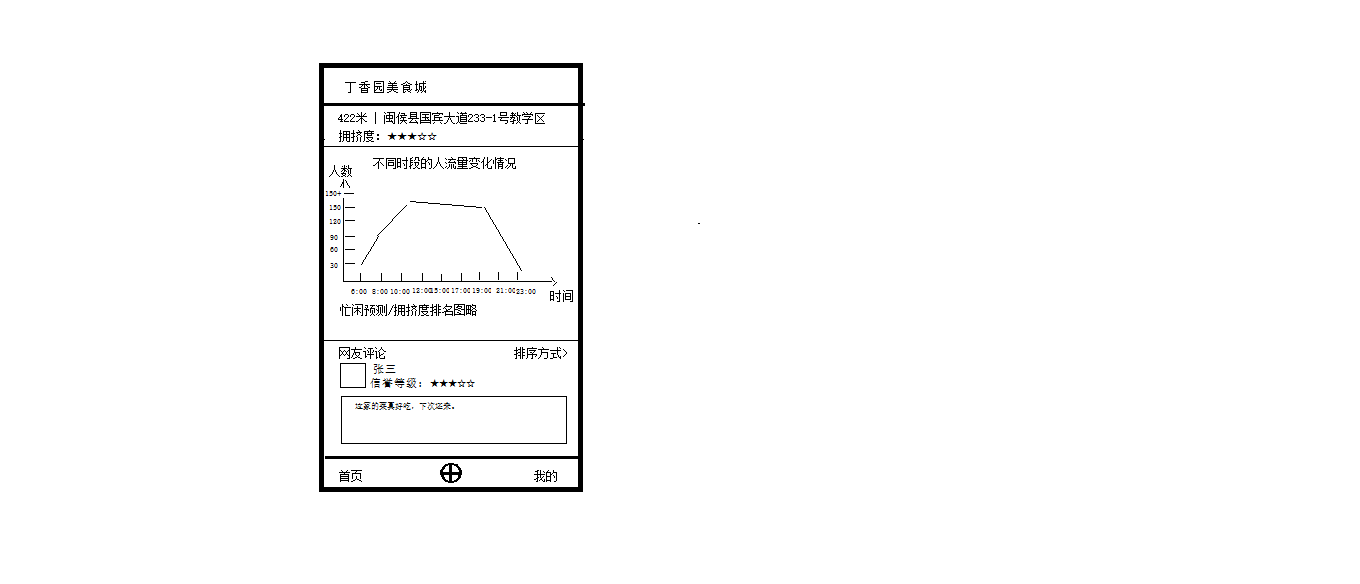
这部分主要包括两部分：一部分是对于板块列表、营业场所列表、新闻列表等的维护与管理；另一部分是对交易信息的管理，如商家信息、场所信息、评论信息等的审核。

* 发布公告

对于系统的通知，可以以公告的形式及时发送给用户，用户可以及时掌握系统的情况。

**2.2 APP主要界面设计草图**

首页界面 排队界面 个人信息界面 



预约界面 饭店类界面 具体饭店经营分析界面

图2-1 APP主要界面设计草图

**3.1 程序执行流程概述**

**1.用户注册与登录界面**

用户注册与登录界面即是系统的入口，该界面主要用于用户的注册与登录，当用户输入的用户名与密码等正确时，进入下一个页面。

**2.系统主界面**

当用户登录成功后，则进入该界面。该界面主要包括两个部分：界面上面部分用于系统首页的展示，而下面的底栏部分则显示三个用于切换界面的按钮，分别是：首页，排队，我的按钮。

**3.首页界面**

该界面以分类的方式向用户展示各种不同种类的营业场所。用户可根据分类点击进入，也可以直接在上方搜索栏中输入营业场所的名称进行查询。

**-子界面**

当用户点击具体的某个营业场所进入后，可显示该场所的名称、位置、拥挤度、信誉度等信息，还可以展示经营分析的结果，以及网友的评论信息等。

**4.我的界面**

当用户点击我的按钮后，进入我的界面。该界面主要以ListView的方式向用户展示其个人信息的管理界面。

**5.排队界面**

当用户点击排队按钮时，进入该界面。该界面可用于用户当前排队信息的上传，当点击排队后，系统会获取其位置并且添加他的排队信息。

**3.2 APP原型设计**

登录界面 首页界面 排队界面 我的界面

饭店页面 我的关注页面 我的预约页面 经营分析页面

图3-1 APP的原型设计界面截图

**3.3 数据库设计**

UML设计采用StarUML；

E-R图设计采用Visio；

物理模型设计采用PowerDesigner。

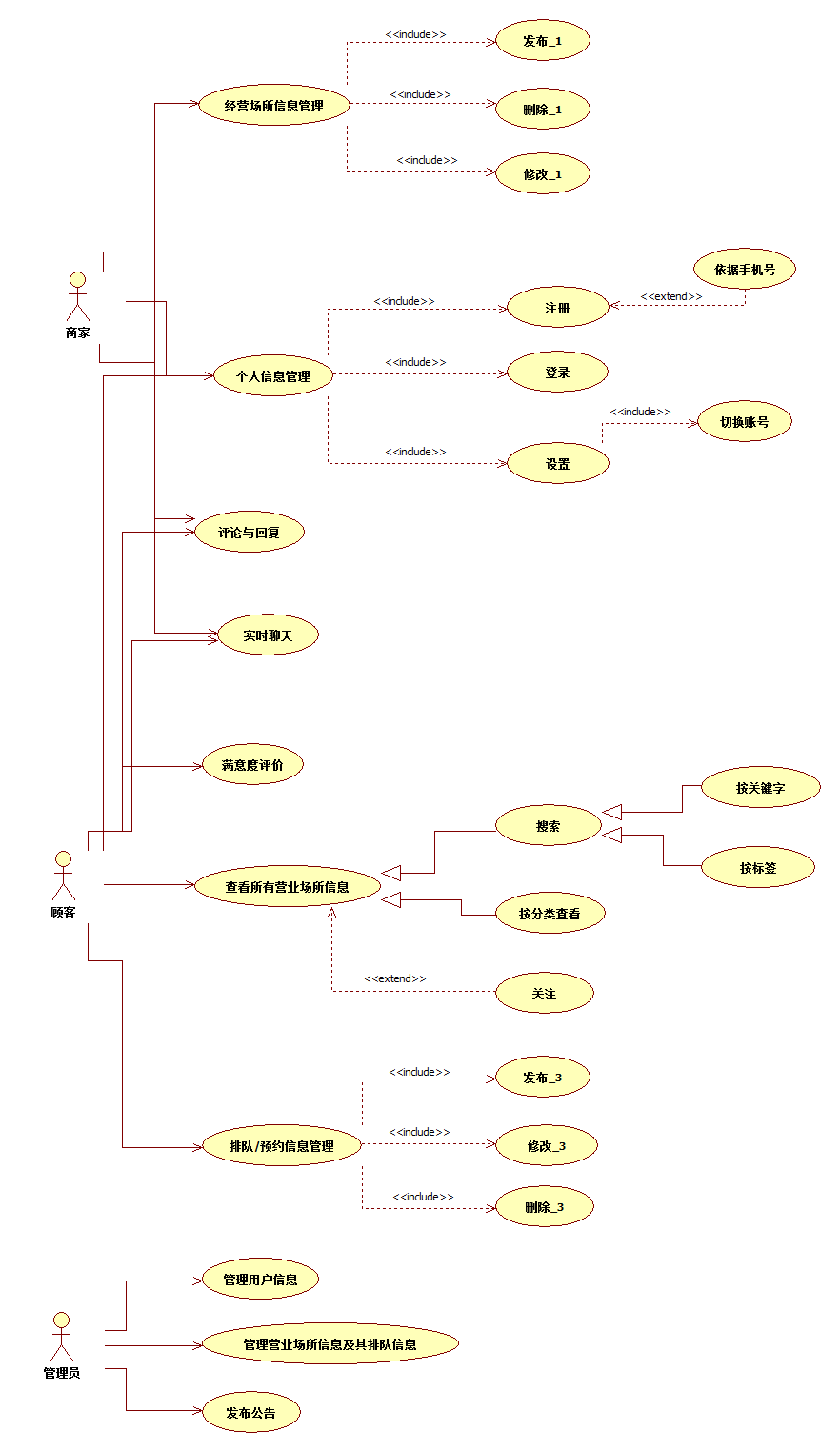
**3.3.1 用例图**

图3-2 系统用例图

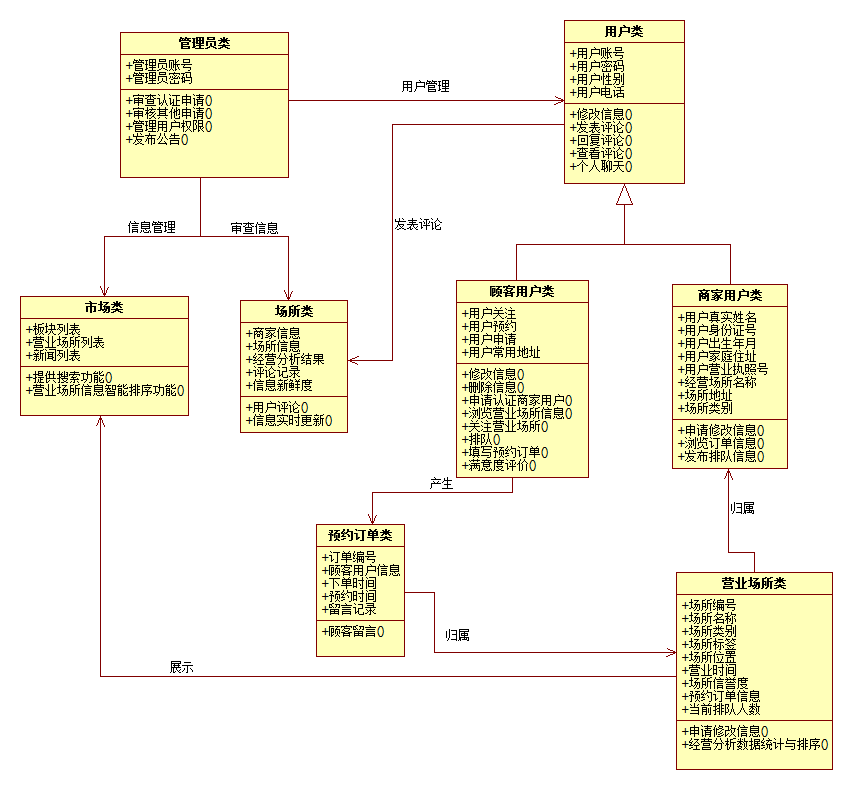
**3.3.2 类图**

图3-3 系统类图

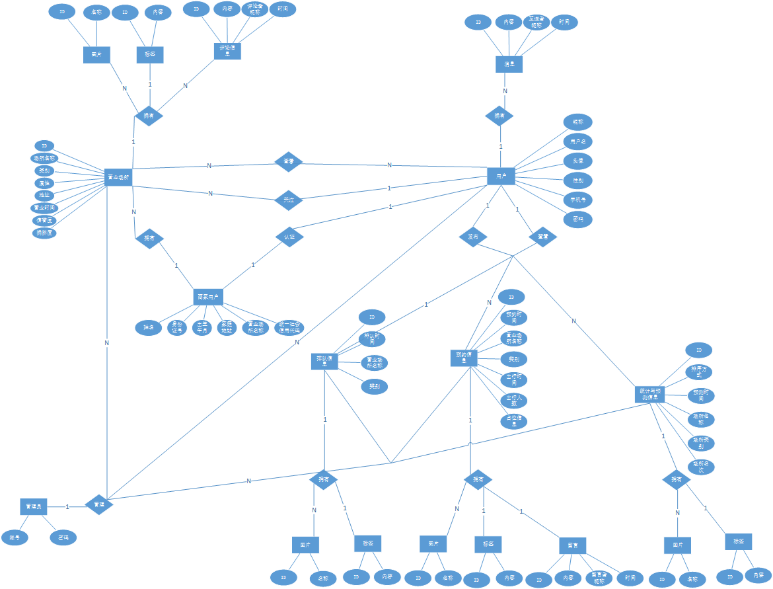
**3.3.3 E-R图**

图3-4 系统E-R图

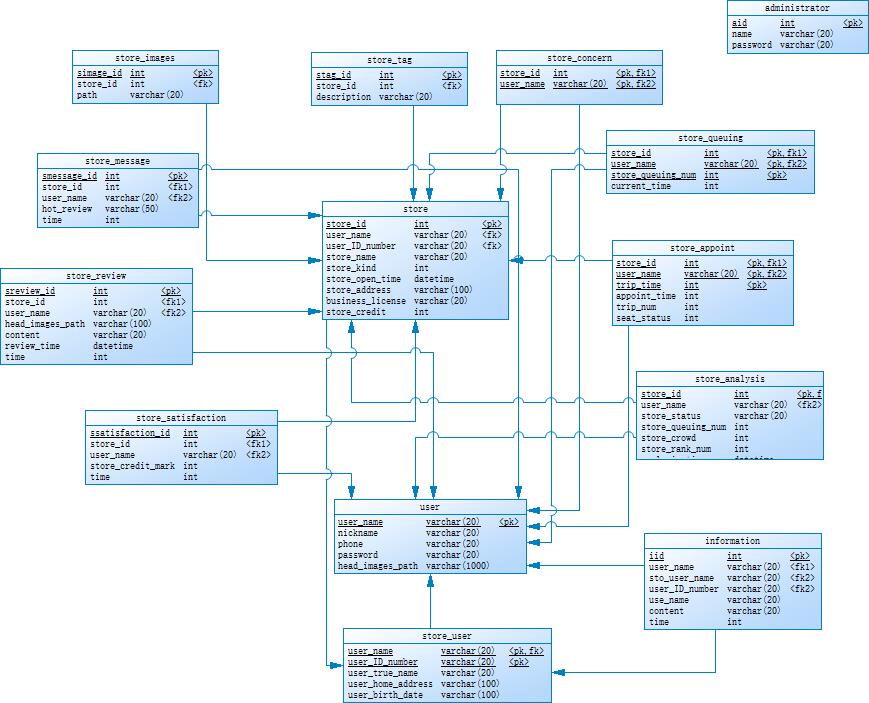
**3.3.4 物理模型**

图3-5 数据库的物理模型

**4.1 结构设计**

**4.1.1 Android端**

APP开发采用MVP模式，如图所示：

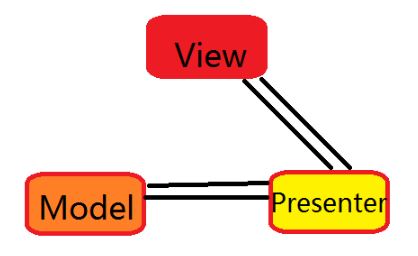


图4-1 MVP模式图

注：M（Model）：数据相关层；

V（View）：视图层，如Activity上的布局；

P（Presenter）纽带层，用来连接Model层与View层。

MVP开发在Android中的基本流程：

（1）View层定义View.interface，用来定义View的行为。一般由Activity或者是Fragment来实现这个接口，它定义了View视图的各种变化，如设置Textview,加载对话框，更新进度条等。

（2）Model层定义Model.interface,这个是用来定义数据层发生变化时的通知接口，因为Model不能直接与View交互，所以它与Presenter交互，然后再通过Presenter间接达到与View的交互。

（3）Presenter翻译的意思是主持人，也就是主持场合，控制节奏的意思。Presenter负责具体的业务逻辑，请求数据，把数据送到Model，或者监听Model的数据变化，接受View层的动作，负责通过通知View层的视图变化。

**4.1.2 服务器端**

后端采用基于CodeIgniter框架的LNMP环境开发。选用CodeIgniter框架的原因：

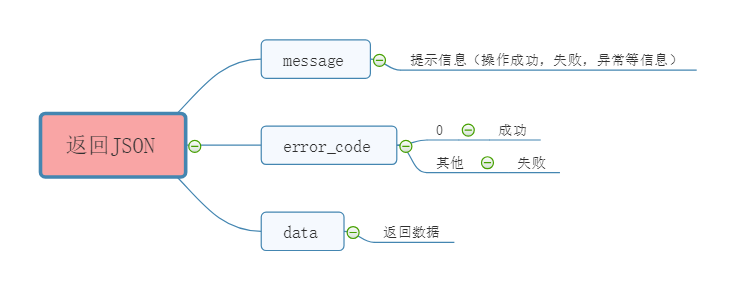
* 它一个小巧的框架，但有出色的性能；
* 它一个几乎零配置的框架；；
* 不需要被迫学习一种新的模板语言（当然如果你喜欢，你可以选择一个模板解析器）；
* 有着清晰、完整的文档。
* 容易上手

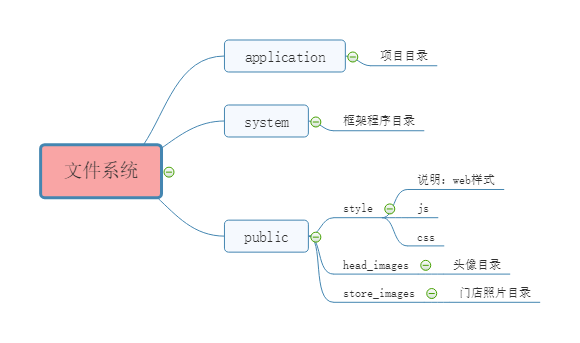
CodeIgniter框架 的开发基于MVC（模型-视图-控制器）设计模式。MVC是一种用于将应用程序的逻辑层和表现层分离出来的软件方法。

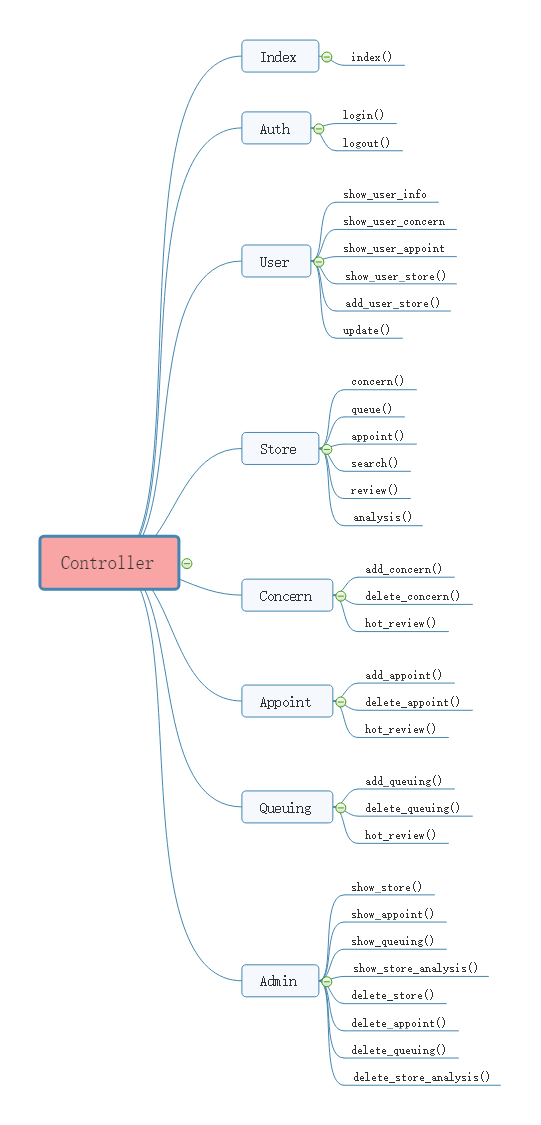
模型：代表存储的数据结构。通常来说，模型类包含对数据库进行增删改查的方法。这些方法是对数据处理的重要部分。

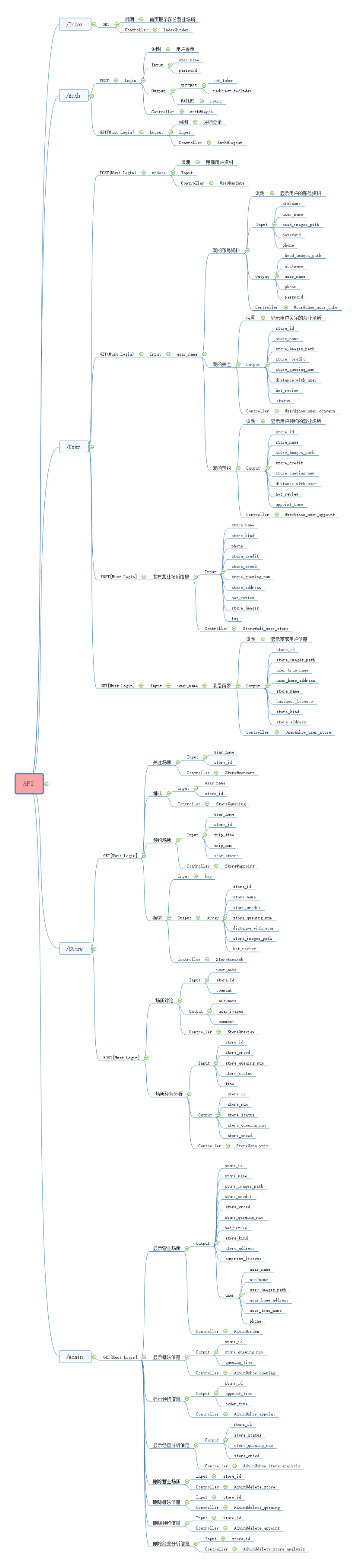
视图：是要展现给用户的信息。一个视图通常就是一个网页，但是我们的APP中，使用到的视图主要是在web端部分。

控制器：是模型、视图以及其他任何处理 HTTP 请求所必须的资源之间的中介。安卓端我们将会使用到CI的M/C。

前台使用okhttp获取后台数据，后台使用php编写API接口。







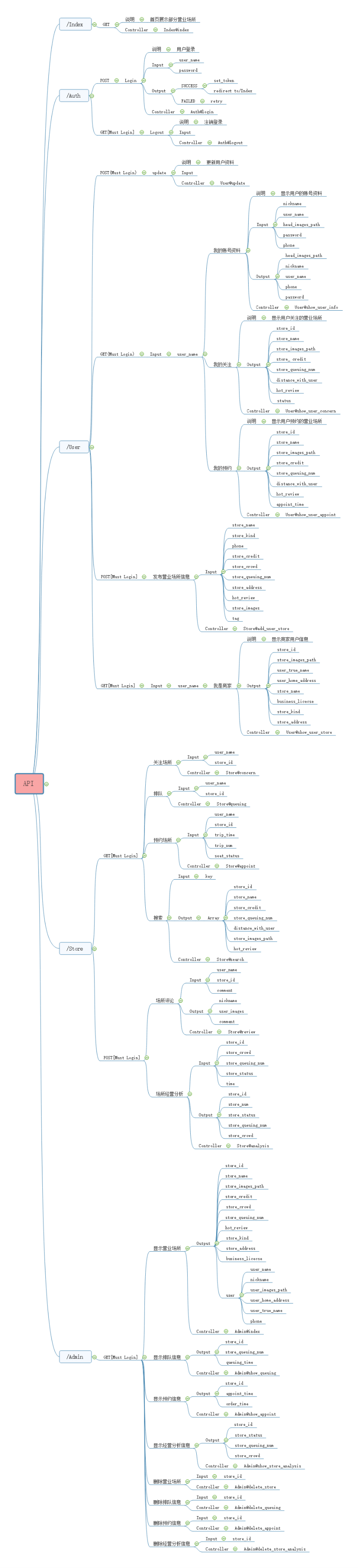


图4-2 APP后端设计图

**4.2 编码规范**

参考已制定好的编码规范：

（1）Android：

<https://github.com/13110538870/QueuingInfoSystem/document/Android项目编码规范.md>

（2）PHP

[https://github.com/13110538870/QueuingInfoSystem/document/PHP代码规范.md](https://github.com/Maple27/FzuYiBao/blob/master/document/PHP代码规范.md)