**微信公众号景区信息查询系统**

**面向对象分析与设计**

**姓名：**

**学号：**

**班级：**

**计算机与控制工程学院**

**年 月 日**

目录

[1. 业务说明 1](#_Toc44144895)

[2. 面向对象分析 2](#_Toc44144896)

[2.1 用例分析 2](#_Toc44144897)

[2.2 分析类模型 3](#_Toc44144898)

[2.3动态模型 4](#_Toc44144899)

[3. 面向对象设计 6](#_Toc44144900)

[3.1 体系结构图 6](#_Toc44144901)

[3.2 界面设计 6](#_Toc44144902)

[3.3 数据库设计 6](#_Toc44144903)

[4. 设计心得 7](#_Toc44144904)

# 业务说明

1. 选题背景：在信息时代的今天，大量信息的数字化极大地方便了我们的生活。而在一些外地游客前往景区时，往往会遇到这样那样的问题，例如找不到路，找不到停车场，不知道去哪吃饭等等。而借用现在发达的网络与信息处理系统，便可极大地方便我们出行，也可方便景区商户招揽顾客，创造更多的收入。本系统使用大多数人都在使用的微信公众号系统为基础框架。
2. 功能分析

在旅游景区，经常会遇到游客打听某某地方怎么去，哪里有停车场等的问题，以及景区当地的天气情况，道路情况等等。设计系统做信息收集与分析工作，并展示给用户。

**功能点1**：线路查询。查询从一个地方到另一个地方有多远，路线怎么走，交通工具等。

**功能点2**：道路情况查询。查询某条路某个路段的路况怎么样，堵不堵车，是不是在修路，有没有限高限宽。

**功能点3**：天气情况的实时显示，能够实时查询到景区当地的天气情况及近期的天气情况。

**功能点4**：游玩建议及游玩疑问的问答系统，通过微信公众号的文字交互系统进行疑问的匹配回答。

**功能点5：**查看游玩历史，可以通过登录后查看自己曾经去过哪些景点，哪些景点还没去。

**功能点6：**门票购买。在某些需要购买门票的地方可以在网上订票，到了景点后出示二维码来使用门票、车票。

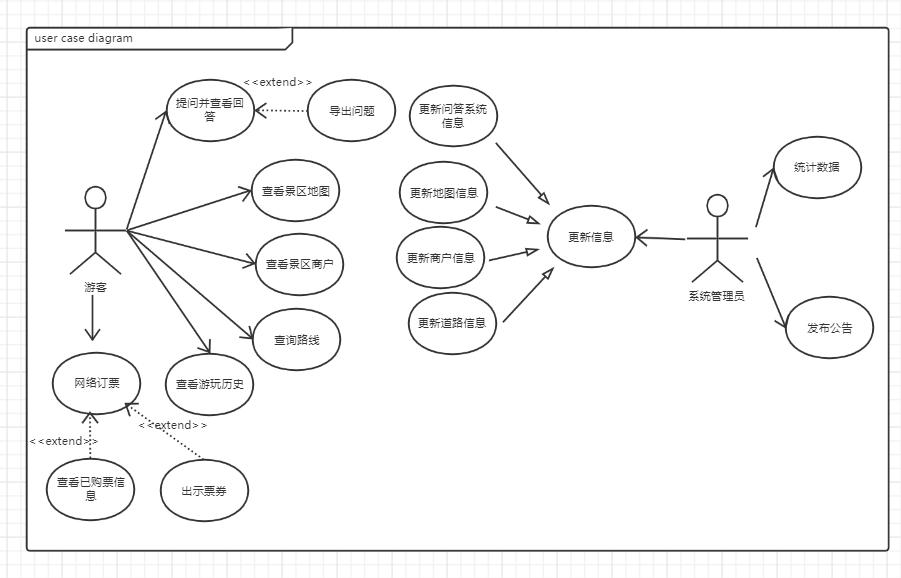
# 2. 面向对象分析

## 2.1 用例分析

根据前面的功能点，将系统分为总的两个用例对象：普通用户和系统管理员。

由于系统基于微信公众号系统开发并实现，故普通用户的使用前提为用户下载微信，注册并登入微信系统，并进入微信公众号系统搜索并关注该微信公众号，进入该公众号界面即可使用系统功能。由于公众号功能有限，地图与天气显示，以及网络订票，查询历史等的功能使用微信公众号界面实现过于复杂与不切实际，故使用web处理。用户需在公众号页面点击使用功能，随后自动跳转页面至web端。

由于微信自带账户系统，在使用订票/历史查询时只需在授权界面使用微信号登录即可直接登入系统。

用户提问并查看回答用例规约:

用户直接向微信公众号发送提问信息，服务器寻找回答后回答用户。

|  |  |  |
| --- | --- | --- |
| 元素 | 描述 | 备注 |
| 用例编号 | 001 |  |
| 用例名称 | 用户提问并查看回答 |  |
| 主要参与者 | 用户 | 默认参与者 |
| 前置条件 | 关注公众号并进入对话页面 |  |
| 基本事件流 | 1.用户登录微信进入公众号系统  2.用户关注公众号进入交互界面  3.用户提交问题  4.系统自动匹配回答  5.用户查看问题的回答 | 登录操作为微信系统操作，不在此系统使用范围。 |
| 后置条件 | 若系统中匹配到了问题，则给出解答；若没有匹配到问题，则询问用户是否去提问区提问。  用户可继续使用公众号的其它功能 |  |
| 异常事件流 | 服务器没开启、网络不通畅则系统不会做出反应。 |  |
| 其它事件流 | 多次提交无效问题时会触发提示，发送一条指导问题怎么问的消息。 |  |
| 注释 | 无 |  |

用户查看景区地图用例规约:

用户查看景区地图时，先点击查看景区地图按钮，就会进入查看地图的web页面，再web页面中，即可随便查看景区地图。

|  |  |  |
| --- | --- | --- |
| 元素 | 描述 | 备注 |
| 用例编号 | 002 |  |
| 用例名称 | 用户查看景区地图 |  |
| 主要参与者 | 用户 | 默认参与者 |
| 前置条件 | 关注公众号并进入对话页面 |  |
| 基本事件流 | 1.用户登录微信进入公众号系统  2.用户关注公众号进入交互界面  3.用户选择查看地图  4.系统返回景区地图网址  5.用户点击网址并查看景区地图  6.返回公众号继续使用其它功能 | 登录操作为微信系统操作，不在此系统使用范围。 |
| 后置条件 | 用户关闭微信内置浏览器后返回公众号界面，可继续使用公众号其它功能，若为外部浏览器则不会返回。 |  |
| 异常事件流 | 服务器没开启、网络不通畅则系统不会做出反应、微信内置浏览器唤起失败、账户自动验证失败。在浏览地图的过程中断网，将会导致点击地图没有反应。 |  |
| 其它事件流 | 在地图界面30分钟停留无操作自动断开连接 |  |
| 注释 | 无 |  |

用户查看景区商户用例规约

|  |  |  |
| --- | --- | --- |
| 元素 | 描述 | 备注 |
| 用例编号 | 003 |  |
| 用例名称 | 用户查看景区商户 |  |
| 主要参与者 | 用户 | 默认参与者 |
| 前置条件 | 关注公众号并进入对话页面 |  |
| 基本事件流 | 1.用户登录微信进入公众号系统  2.用户关注公众号进入交互界面  3.用户选择查看景区商户信息  4.系统返回景区商户信息  5.用户选择访问商户详细信息  6.用户可选择继续使用其它功能 | 登录操作为微信系统操作，不在此系统使用范围。 |
| 后置条件 | 用户可继续使用公众号的其它功能 |  |
| 异常事件流 | 服务器没开启、网络不通畅则系统不会做出反应 |  |
| 其它事件流 | 用户输入没匹配到商户名的情况下放回未查到此商户。在商户界面停留30分钟无操作自动断开连接 |  |
| 注释 | 无 |  |

用户查询景区路线用例规约：

|  |  |  |
| --- | --- | --- |
| 元素 | 描述 | 备注 |
| 用例编号 | 004 |  |
| 用例名称 | 用户查询景区路线 |  |
| 主要参与者 | 用户 | 默认参与者 |
| 前置条件 | 关注公众号并进入对话页面 |  |
| 基本事件流 | 1.用户登录微信进入公众号系统  2.用户关注公众号进入交互界面  3.用户选择查询某景点到某景点怎么走  4.系统返回路线与路线信息  5.用户选择查询某路的路线情况  6.系统返回路线情况  7.用户可选择继续使用其它功能 | 登录操作为微信系统操作，不在此系统使用范围。 |
| 后置条件 | 用户可继续使用公众号的其它功能 |  |
| 异常事件流 | 服务器没开启、网络不通畅则系统不会做出反应 |  |
| 其它事件流 | 查询时发送了无效地点会返回无效地点，多次提交无效问题时会触发提示，发送一条指导路线怎么查的消息。 |  |
| 注释 | 无 |  |

用户查看景区天气用例规约：

|  |  |  |
| --- | --- | --- |
| 元素 | 描述 | 备注 |
| 用例编号 | 005 |  |
| 用例名称 | 用户查看景区天气 |  |
| 主要参与者 | 用户 | 默认参与者 |
| 前置条件 | 关注公众号并进入对话页面 |  |
| 基本事件流 | 1.用户登录微信进入公众号系统  2.用户关注公众号进入交互界面  3.用户选择查看景区天气  4.系统返回一张当前景区的天气状态图。  5.用户输入详细地点  6.返回详细地点的天气状态图 | 登录操作为微信系统操作，不在此系统使用范围。 |
| 后置条件 | 用户可继续使用公众号的其它功能 |  |
| 异常事件流 | 服务器没开启、网络不通畅则系统不会做出反应。若在观看地图时断网，则不会对查看地图造成影响，但无法继续使用后续功能。 |  |
| 其它事件流 | 若景区天气系统未更新则会显示“请联系管理员更新天气信息”；若查询详细地点天气时输入无效地名则会提示地址无效。 |  |
| 注释 | 无。 |  |

查看历史游玩信息用例规约：

|  |  |  |
| --- | --- | --- |
| 元素 | 描述 | 备注 |
| 用例编号 | 006 |  |
| 用例名称 | 查看历史游玩信息 |  |
| 主要参与者 | 用户 | 默认参与者 |
| 前置条件 | 关注公众号并进入对话页面 |  |
| 基本事件流 | 1.用户登录微信进入公众号系统  2.用户关注公众号进入交互界面  3.用户选择查看游玩历史  4.系统返回游玩历史查询的网络连接  5.用户点击连接打开微信内置浏览器  6.浏览器显示请求授权使用微信的账号信息  7.点击授权使用  8.查询并显示游玩的历史信息 | 登录操作为微信系统操作，不在此系统使用范围。授权使用账号信息时只需点击按钮即可。 |
| 后置条件 | 用户关闭微信内置浏览器后返回公众号界面，可继续使用公众号其它功能，若为外部浏览器则不会返回。 |  |
| 异常事件流 | 服务器没开启、网络不通畅则系统不会做出反应。若在浏览历史信息时断网则会导致浏览信息不能更新。 |  |
| 其它事件流 | 若该用户之前没有使用过公众号系统或之前没有来游玩的记录则显示“查无游玩记录”。 |  |
| 注释 |  |  |

网络订票用例规约

用户通过网络订票时，首先再微信公众号中点击景区购票，系统打开web界面，并自动跳转至账号授权界面，授权后即可进行订票。

|  |  |  |
| --- | --- | --- |
| 元素 | 描述 | 备注 |
| 用例编号 | 007 |  |
| 用例名称 | 用户网络订票 |  |
| 主要参与者 | 用户 |  |
| 前置条件 | 服务器正常运行、微信号注册并关注系统。 |  |
| 基本事件流 | 1. 用户登录微信进入公众号系统 2. 用户关注公众号进入交互界面 3. 用户选择网络订票 4. 系统返回网络订票网页 5. 用户点击进入网页 6. 用户同意授权登录 7. 用户选择购买的门票 8. 创建订单 9. 用户付款 10. 生成票务二维码 11. 用户查看已购票 12. 系统显示已的票 | 网络订票需要用户授权登录后使用。 |
| 后置条件 | 用户可继续订票或退回公众号继续使用 |  |
| 异常事件流 | 若使用中突然断网，则会跳转至404网页， |  |
| 其它事件流 | 若由于账户金额不足或网络不通畅导致支付失败，则不会取消订单。  若订单15分钟内没有被支付则取消订单。 |  |
| 注释 | 无 |  |

管理员管理系统信息用例规约：

|  |  |  |
| --- | --- | --- |
| 元素 | 描述 | 备注 |
| 用例编号 | 008 |  |
| 用例名称 | 管理员管理系统信息 |  |
| 主要参与者 | 系统管理员 |  |
| 前置条件 | 服务器运行且系统管理员登入服务器 |  |
| 基本事件流 | 1. 系统管理员使用ssh远程登录或直接使用操作系统界面登入服务器后台 2. 输入管理员用户名与密码 3. 验证成功后进入系统管理界面 4. 提交修改的信息 5. 系统修改后显示修改成功 6. 管理员可继续更新修改或直接退出系统。 | 同一时间只能由一个管理员操作系统 |
| 后置条件 | 若修改成功，则系统内容被更改；若修改失败，则系统内容不被更改。若点击退出则退出系统 |  |
| 异常事件流 | 服务器登入失败、管理员账号密码验证失败，会导致管理员不能继续管理系统。  数据库没开启或运行等出现问题会导致信息更新失败，提示重启数据库或重启系统。 |  |
| 其它事件流 | 管理员可以通过编程来增加新功能。 |  |
| 注释 | 管理员账号只有一个，同一时间只有一个设备能登录该账号。 |  |

## 2.2 分析类模型 根据用例分析，系统中的边界大致分为微信号的边界与服务器系统的边界。

微信的边界为与客户操作和与服务器交互，服务器的边界为与微信交互。其中与用户交互的界面由微信公众号系统自带，只需向公众号系统发送信息即可显示给用户。web端需要编写一定的前端程序来实现与用户的信息交互。

## 系统边界类为

## 

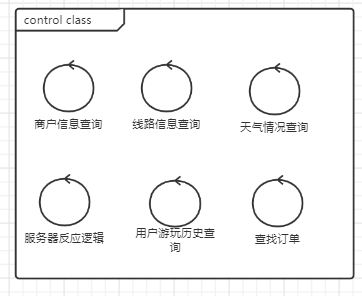
**系统的边界类**

|  |  |  |
| --- | --- | --- |
| **编号** | **类名称** | **类说明** |
| 1 | 微信公众号界面 | 微信公众号界面 |
| 2 | 地图界面 | 显示景区地图的界面 |
| 3 | 商户信息界面 | 显示景区商户的信息界面 |
| 4 | 用户信息拉取 | 系统从微信拉取用户的信息 |
| 5 | 历史信息界面 | 显示用户的历史游览信息 |
| 6 | 购票信息界面 | 显示用户的购票信息 |
| 7 | 登录信息授权界面 | 授权用户的登录信息 |

**系统的控制类**

系统中的控制类主要划分为控制微信公众号的类与控制服务器反应的类，信息查询的类。

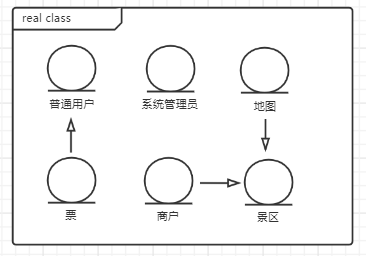
其中，控制微信公众号的类主要包含微信公众号类，此类主要控制与微信公众号，信息查询包括商户信息查询、线路信息查询等，并抽象出基类信息查询类来利用继承封装的特性方便对类的管理。



|  |  |  |
| --- | --- | --- |
| **编号** | **类名称** | **类说明** |
| 1 | 商户信息查询 | 用户查询景区商户信息 |
| 2 | 线路信息查询 | 用户查询景区线路 |
| 3 | 天气情况查询 | 用户查询景区天气情况 |
| 4 | 服务器反应逻辑 | 服务器的反应控制类 |
| 5 | 用户游玩历史查询 | 服务器查询某用户在景区的游玩历史 |
| 6 | 查找订单 | 订单查找 |

**系统的实体类**

系统中的实体类主要包括各个使用者的实体类与景区各个实体的实体类。包括系统管理员与普通用户等的类，景区实体类有景区、地图、商户等。

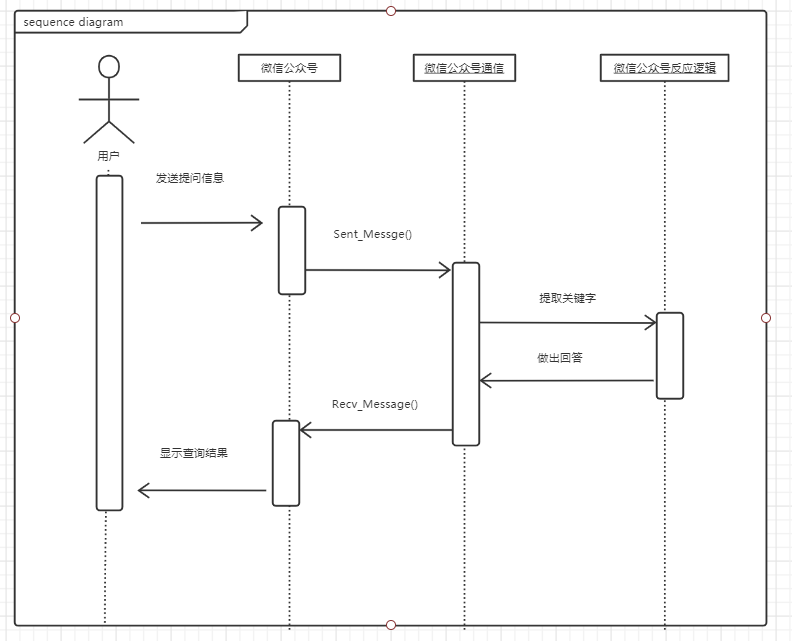


|  |  |  |
| --- | --- | --- |
| **编号** | **类名称** | **类说明** |
| 1 | 普通用户(common user) | 使用公众号的用户 |
| 2 | 系统管理员(manager) | 管理系统的管理员 |
| 3 | 地图(map) | 地图信息 |
| 4 | 商户(shop) | 商户信息 |
| 5 | 景区(view) | 景区信息 |
| 6 | 游玩历史(history) | 用户的游玩历史 |
| 7 | 票(ticket) | 用户购买的票（门票，车票） |

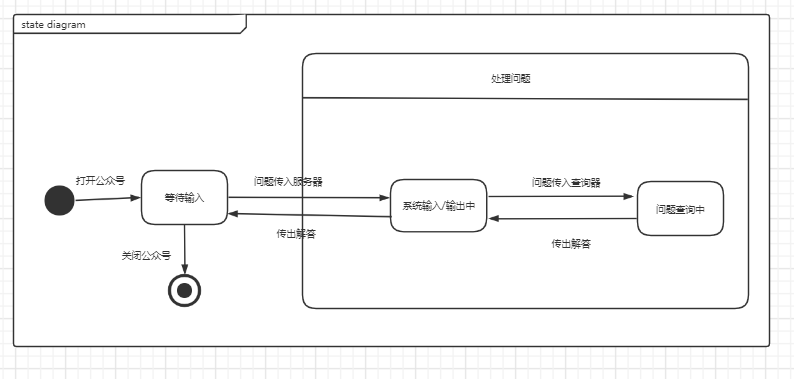
## 2.3动态模型

**问题查询、天气查询、线路查询等的时序图**

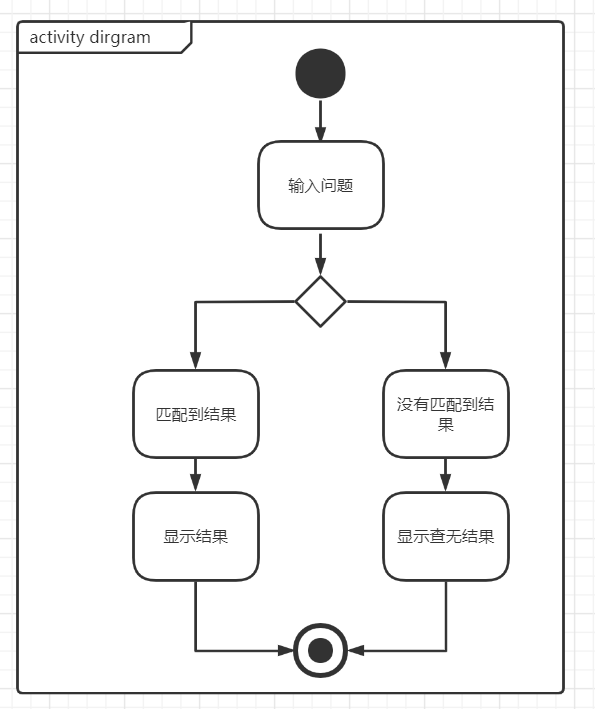
由于问题查询、天气查询、线路查询等使用微信公众号平台进行操作，操作逻辑、操作方式一致，故使用同一张时序图。在使用时，微信公众号接收到用户的使用请求，随后发送信息至服务器，服务器收到消息后根据事先写好的逻辑进行应答，随后应答传回微信用户。



**问题查询、天气查询、线路查询等的状态图**



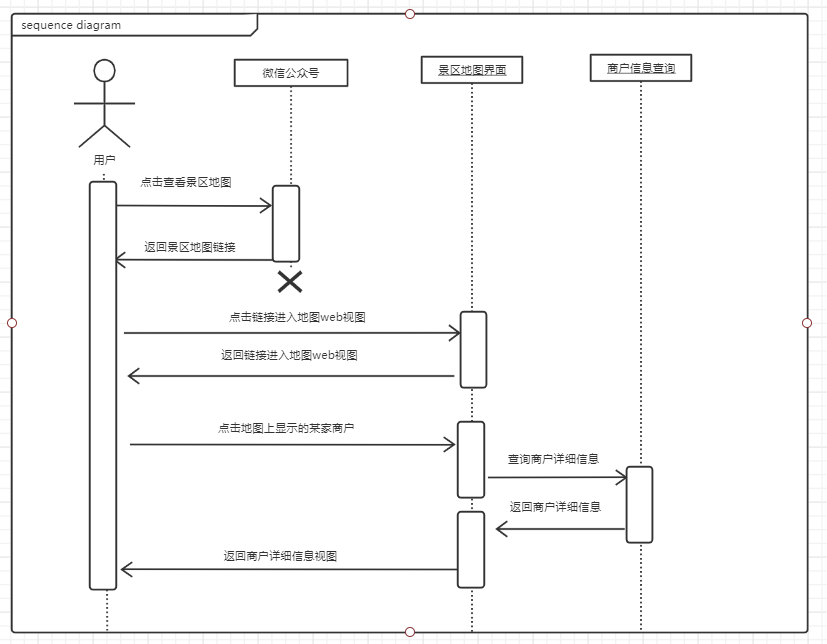
**问题查询、天气查询、线路查询等的活动图**

****

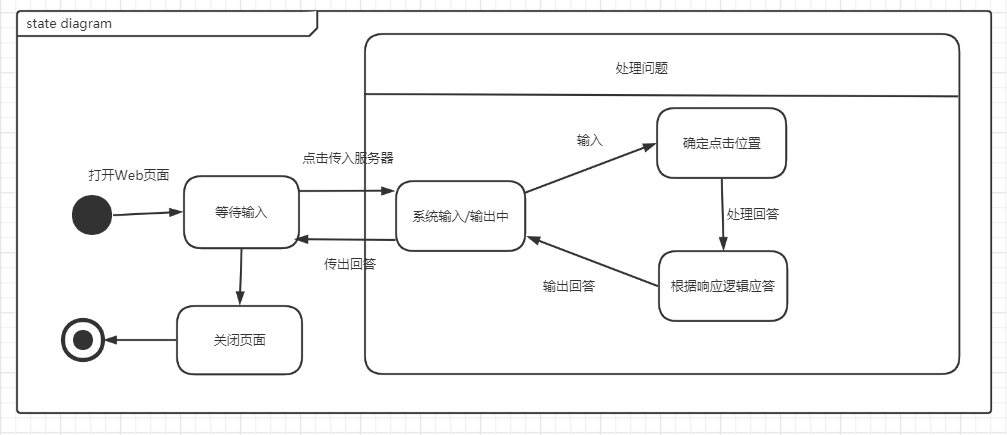
**地图查看、天气查看时序图**

由于微信公众号功能有限，故该功能将使用公众号与web浏览器共同实现。用户点击查看地图，公众号系统随即返回地图连接，用户点击连接进入浏览器后即可浏览景区地图。

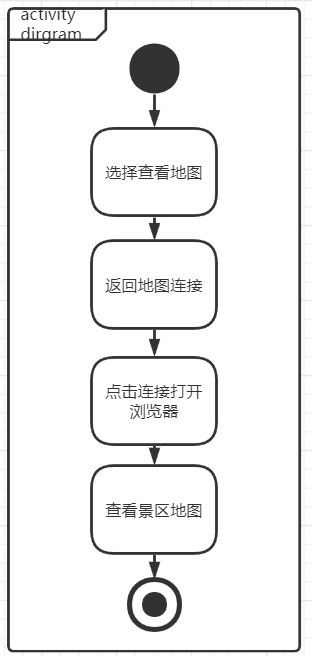
用户既可以使用微信公众号来输入地点查询目的地天气，也可以直接打开天气地图来直观地查看各个地方的天气情况。由于两者的功能、逻辑相同，故使用同一张时序图、状态图。



**地图查询、天气查询的状态图**

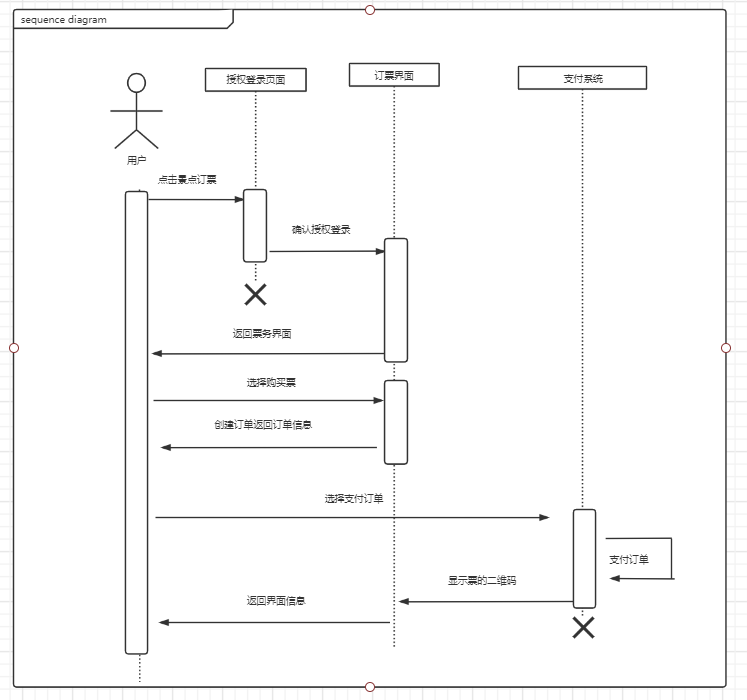


**地图查询、天气查询的活动图**

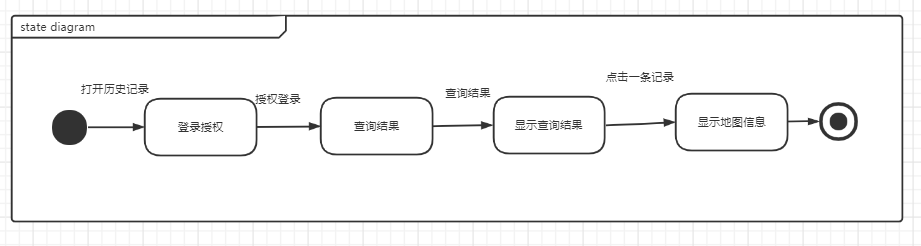


**历史游玩信息查询**

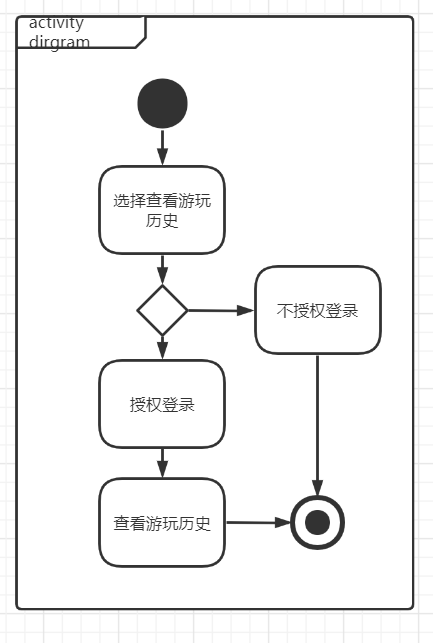
由于每个用户的历史信息不同，故在查询历史游玩信息时，游客应首先授权登录系统，待系统查询到游客的游览历史信息后，再对信息进行显示。



**历史信息查询状态图**

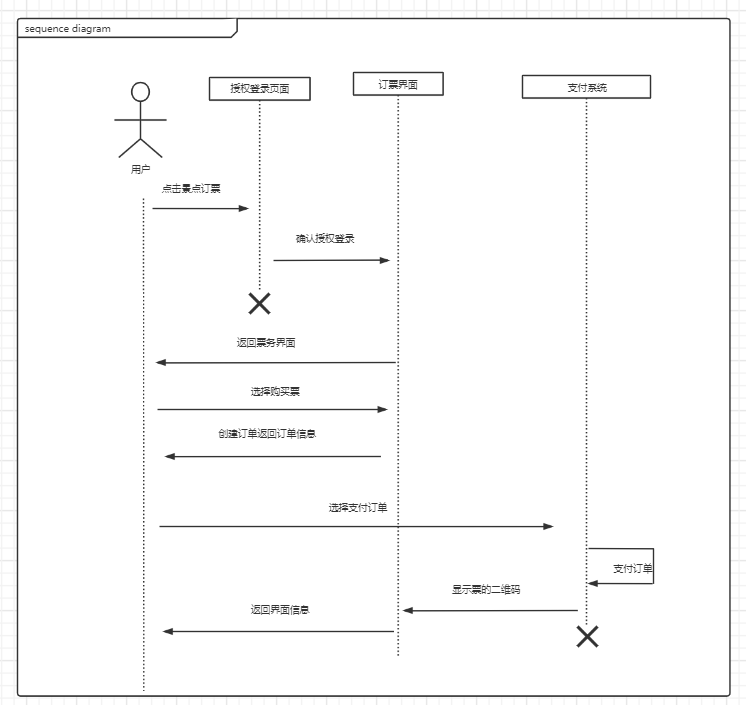


**历史信息查询活动图**

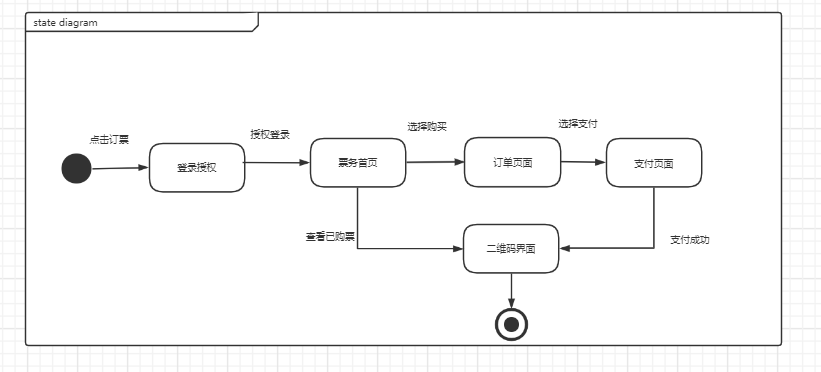


**游客订票时序图：**

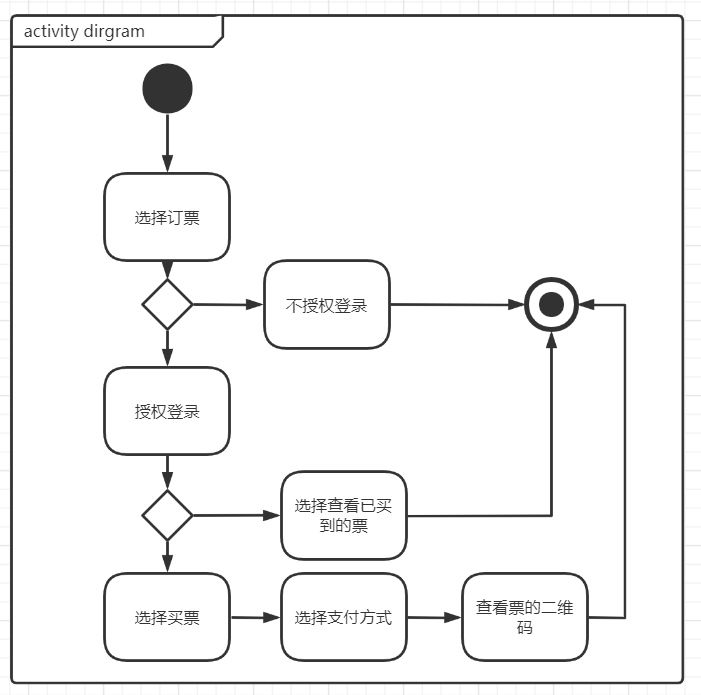
游客订票时，首先从公众号点击票务系统后自动跳转到授权登录系统的web网页，再游客同意授权登录后进入售票系统。用户在订票首页点击自己想要购买的票，系统就生成订单，用户支付订单后，系统生成二维码。用户使用二维码即可使用自己买的票。游客可在票务系统首页查看已购买的票。



**游客订票状态图：**



**游客订票活动图**



# 面向对象设计

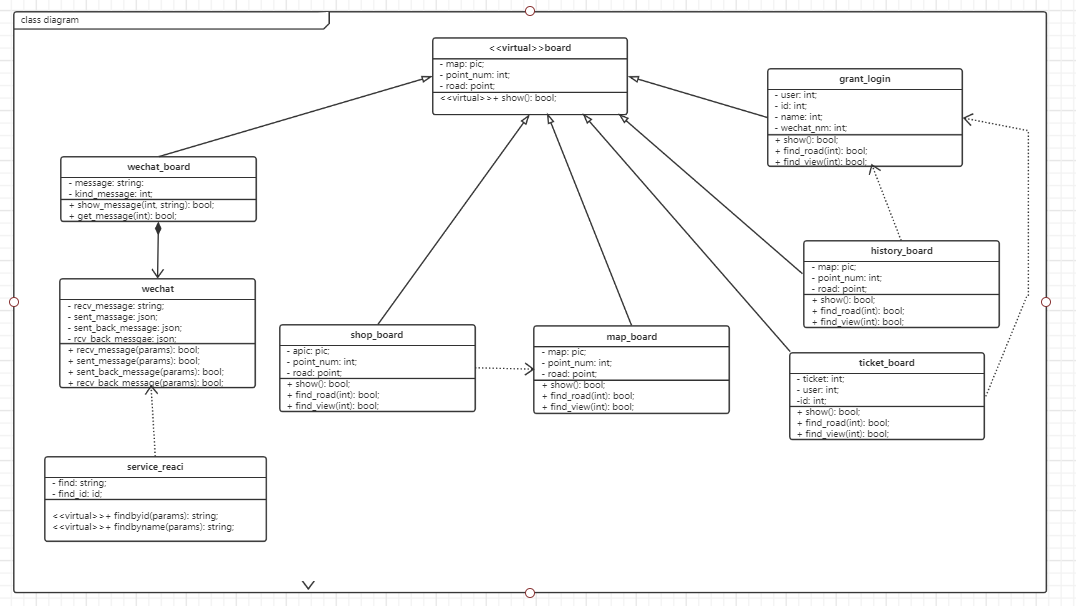
根据前面有关用例的分析与分析了模型的分析，将系统中有关的类划分在了一个类图中。

其中，将系统中的控制类的查找类提取其中的方法，形成了一个虚基类，类似的还有将显示类提取出基类，来缩短开发时间，并增强系统的健壮性和可修改性。

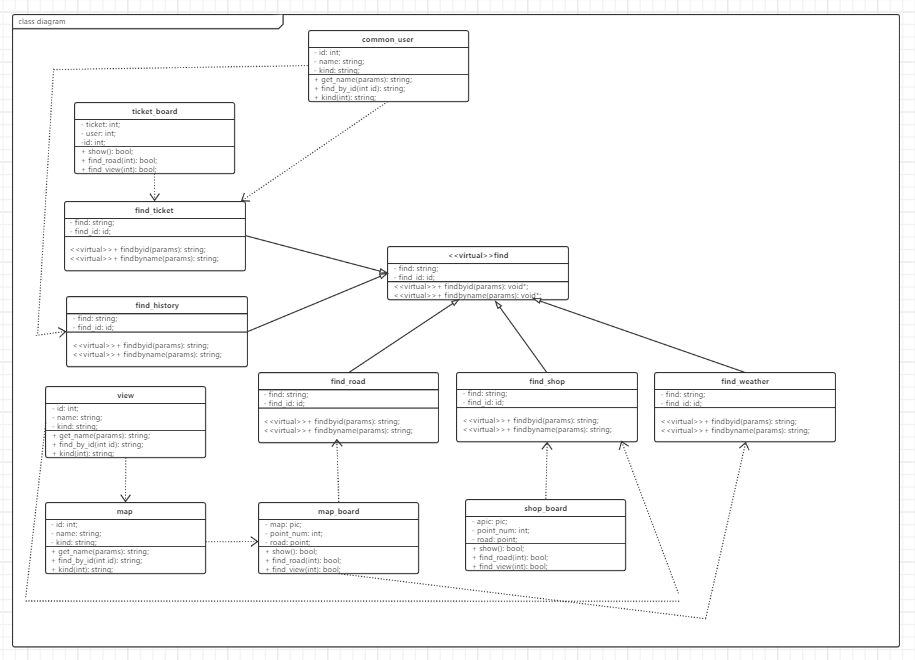
由于将所有类放在一个类图中太过臃肿，故按照系统的边界类，控制类，实体类大致划分为三个类图。但当存在跨类图之间的联系时，一个类可能会同时出现在两个类图。

**类图：**

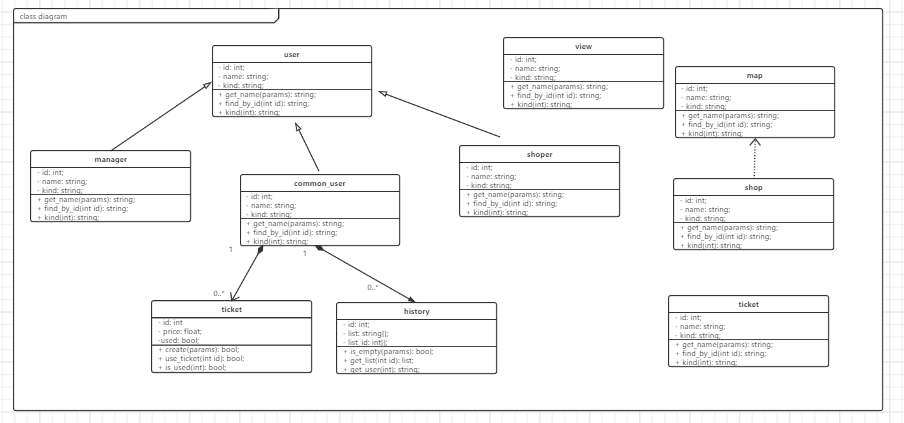
此类图中主要包含系统中的边界类

****

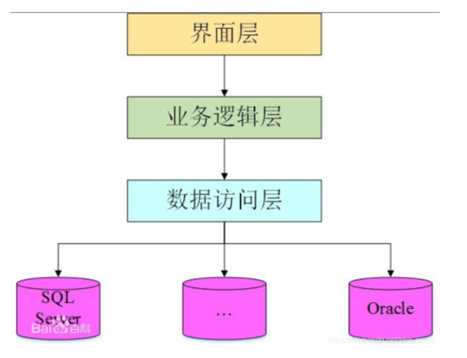
此类图中主要包含系统中的控制类



此类图主要包含系统中的实体类，系统中的管理员、普通用户、商户被抽象出基类用户，其中，普通用户的查询历史记录被单独分为一个类。

****

## 3.1 体系结构图



由于该系统基于微信与微信公众号系统，可适配大多的移动端设备。而微信业务的处理需要服务器-终端来实现，用户只需要安装微信注册并关注公众号即可使用该系统的功能。且方便软件的开发、推广与维护。

1. 界面层

界面层主要包括web端的地图界面与天气界面、商户界面和商户信息详细界面。

该层包括类为

1. 微信公众号
2. 地图界面
3. 天气界面
4. 商户信息界面
5. 用户信息拉取
6. 历史信息界面
7. 业务逻辑层

业务逻辑层主要包括微信公众号信息处理逻辑，web端信息处理逻辑，用户、商户的信息查询与显示等等。

该层包括的类有：

1. 用户类
2. 商户类
3. 微信公众号通信类
4. 地图类
5. 道路信息类
6. 天气信息类
7. 数据访问层

为了访问系统中存储在数据库中的有关地图信息，商户信息、用户信息等的数据，设置数据访问层。

该层主要的类有：

1. 商户信息查询
2. 用户信息查询
3. 地图信息查询
4. 天气信息查询
5. 数据库连接与运行状态

## 3.2 界面设计

本程序界面主要分为公众号界面与web地图界面、web天气界面、购票界面、游玩历史界面。

其中，公众号界面样式为：

用户将使用打字或者点击按钮的形式使用公众号功能

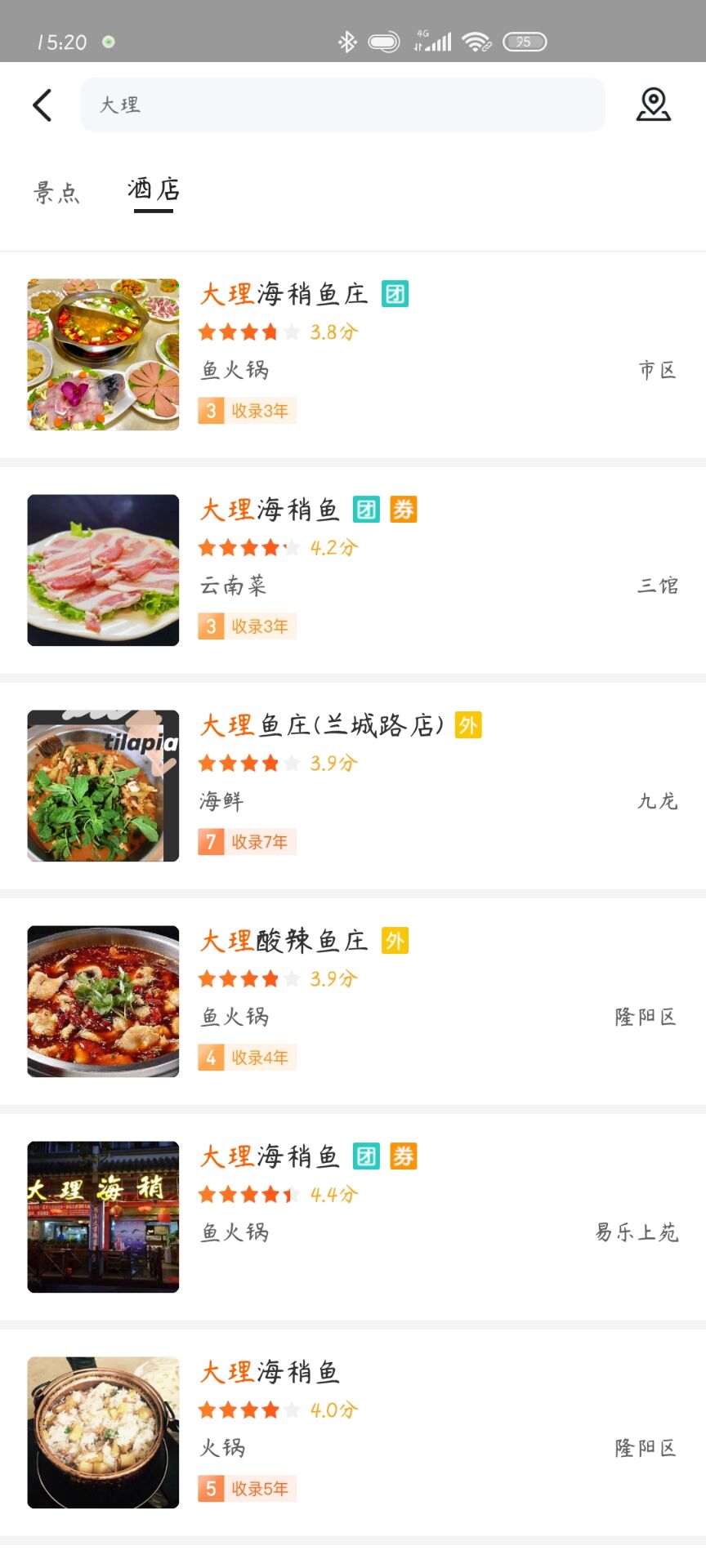


web地图界面样式为：

用户点击某家商户即可在地图页面浏览商户信息



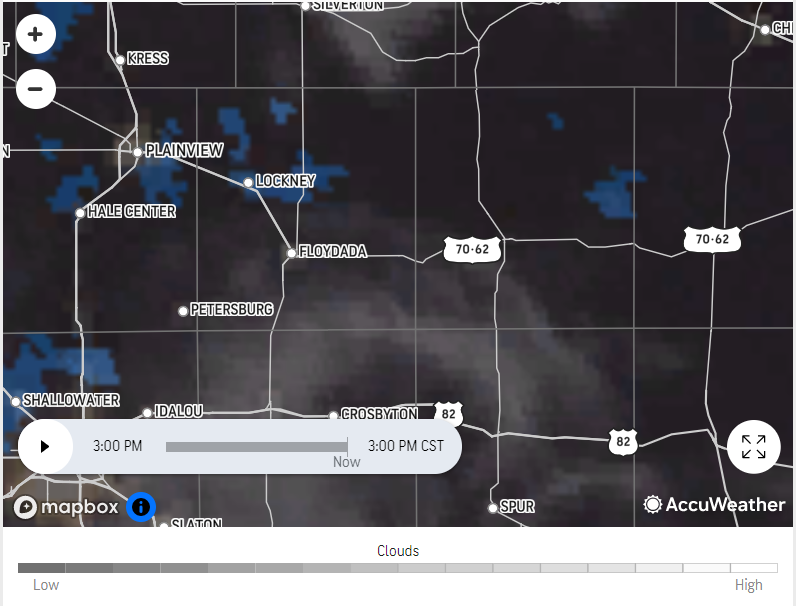
商店列表样式为：



商店详细信息界面样式为：



天气界面样式为：



## 3.3 数据库设计

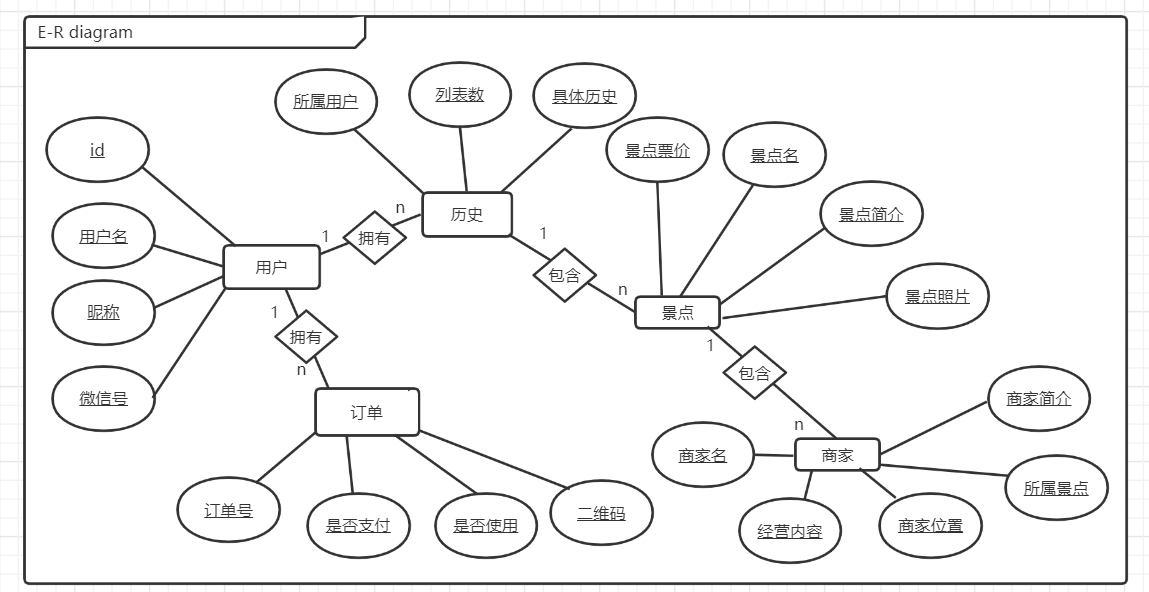
1. **数据库功能分析**

此系统使用Oracle数据库，数据库将记录系统中的景点数据、用户数据、票务订单数据等，数据量大，数据种类多的数据。

其中一个用户应可以有零或多条浏览历史，也可以有零或多个订单。一条浏览历史可以包含多个景点。一个景点可以包含多个商家。为了唯一标识用户，用户不仅有昵称，也有一条自动生成的id号。

1. **数据库概念模型设计**

Oracle数据库

****

数据字典：

用户表：

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| id | integer | 主键 |
| 用户名 | string |  |
| 昵称 | string |  |
| 微信号 | integer |  |

游览历史表：

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| 所属用户 | integer | 外键（用户表） |
| 列表数 | integer | 有几次浏览历史 |
| 具体历史 | string |  |

景点表：

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| 景点名 | string | 主键 |
| 景点简介 | string |  |
| 景点票价 | double |  |
| 景点照片 | photo |  |
| 景点位置 | point | 景点所在位置 |

商家表：

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| 商家名 | string | 主键 |
| 商家简介 | sting |  |
| 所属景点 | integer |  |
| 商家位置 | point | 商家所在位置 |
| 经营内容 | string |  |

票务订单具有时效性，在订单创建后15分钟内未支付，则订单会被删除。

票务订单表：

|  |  |  |
| --- | --- | --- |
| 字段名 | 数据类型 | 备注 |
| 订单号 | integer | 主键 |
| 是否支付 | bool |  |
| 门票种类 | string | 票的种类，车票/门票 |
| 是否使用 | bool |  |
| 二维码 | photo |  |
| 所属景点 | integer | 外键（景点） |
| 价格 | float | 门票/车票的价格 |
| 过期日期 | date |  |
| 所属用户 | integer | 外键（用户） |

# 设计心得

通过此次的软件工程课程设计，我实际通过从选题到需求分析，再到程序模型分析设计等一步步操作完成了此次的程序设计。在这次的设计操作中，我进一步认识到了软件设计的步骤和要点，也总结出了许多经验和教训。

在此次的课程设计中，我认为我做的比较好的是我完整地从选题到分析，一步一步脚踏实地地分析、然后得出分析报告，着重分析了许多系统重点，对重点需求进行详细分析与建模，最后得出清晰明确的软件模型与正确的可行性分析报告。

当然，在此次的课程设计实验中，我也有许多做的不足的地方。例如：在分析并绘制软件的用例图时，由于我对用例图绘制的理解不够全面和彻底，导致第一次绘制用例图的时候出现了许多的错误，经过了老师耐心的指正后，才最终分析绘制出了正确并符合软件需求要求的用例图。

经过此次的软件工程课程设计，我进一步了解了如何对一个软件项目进行详细的功能需求分析和用例分析，并设计软件系统并最终设计出活动图、类图，达到开发一个软件的前提与要求。通过精细的设计与分析，最终做出一款合格的软件系统。