小型餐馆点餐系统概要设计规约

**修订历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编写日期** | **SEPG** | **版本** | **说明** | **作者** |
| 2015/1/15 | 文档 | 1.01 | 增加了文档描述 | 吴雨霏 |
| 2015/1/15 | 文档 | 1.02 | 增加界面设计和数据库设计 | 刘雅骏 |
| 2015/1/15 | 文档 | 1.03 | 增加了接口设计，部署图，架构设计，以及包图 | 赵明阳 |

目录

1. 引言 2

2. 概要设计 2

# 1. 引言

1.1 概要设计依据

* 主要依据：需求/需求分析规约
* 体系结构设计：类之间的方法调用（序列图/VOPC）
* 数据库设计：ERD
* 接口设计：数据字典

1.2. 参考资料

《面向对象分析与设计课件 》

《同济大学杜庆峰老师上课笔记》

同济大学软件学院服务器软件工程杜庆峰老师 [\\10.60.41.1](file:///\\10.60.41.1)

<Software Engineering—A Practitioner’s Approach> Seventh Edition Roger S. Pressman

同济大学软件学院服务器UML孙萍老师课件 [\\10.60.41.1](file:///\\10.60.41.1)

1.3. 假定和约束

开发期限：2~3周

经费限制：无经费

设备条件：设备均由开发者自己提供

用户资料准备：由于是一个课程设计项目，并没有实际商家参与，所以需要模拟商家的信息，包括商家名，商家餐品信息。

# 2. 概要设计

2.1. 系统总体架构设计

* 应用层为顾客界面，提供点菜等服务

从后端获得商品的信息，在用户界面上进行显示。供用户选择。将用户选择的结果传递给商户点菜管理层，设置订单的表项。

* 商户点菜管理层

接受应用层顾客做出的请求以及做出响应（扩展功能：可以根据订单的信息调节商品的定价信息以及促销功能）

* 商品信息管理层为商家进行信息修改等操作

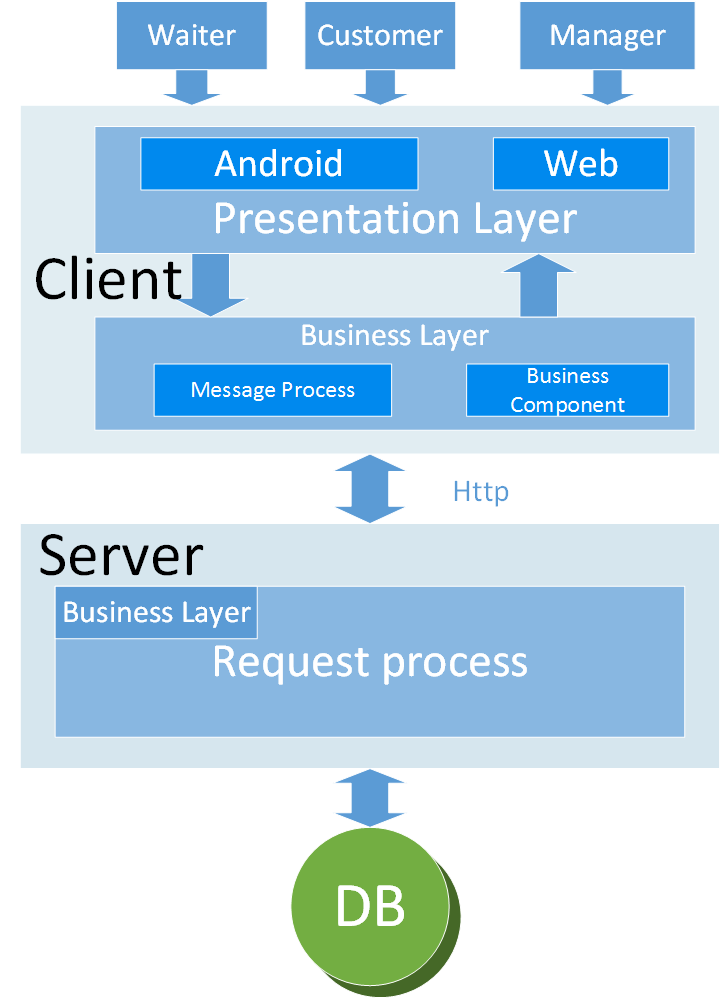
修改菜品的价格和优惠信息（折扣信息）

* 支持层

包括客户端操作系统Android，服务端的数据库和服务器，以及用户端的本地数据库

2.2. 系统软件结构设计

2.2.1 系统软件结构设计



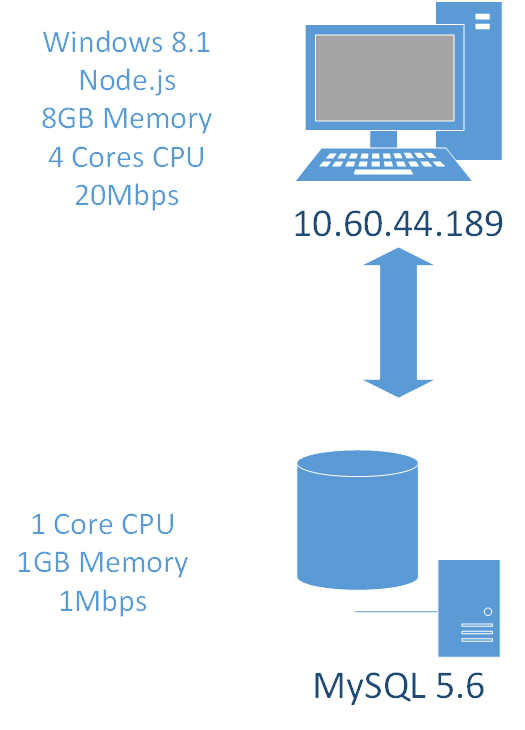
描述：

系统的使用者waiter , customer 通过app应用和系统交互，而餐馆的管理者，在web端，通过登陆确认身份，参与餐馆的管理以及查询。

所以，对于展示层分为Android 以及Web两大块应用。在客户端还有一层处理层，对于Android，手机端将会计算所点菜品的总价，并且处理从服务器端传来的json返回数据。Web处理层主要是对json数据进行处理并展示。

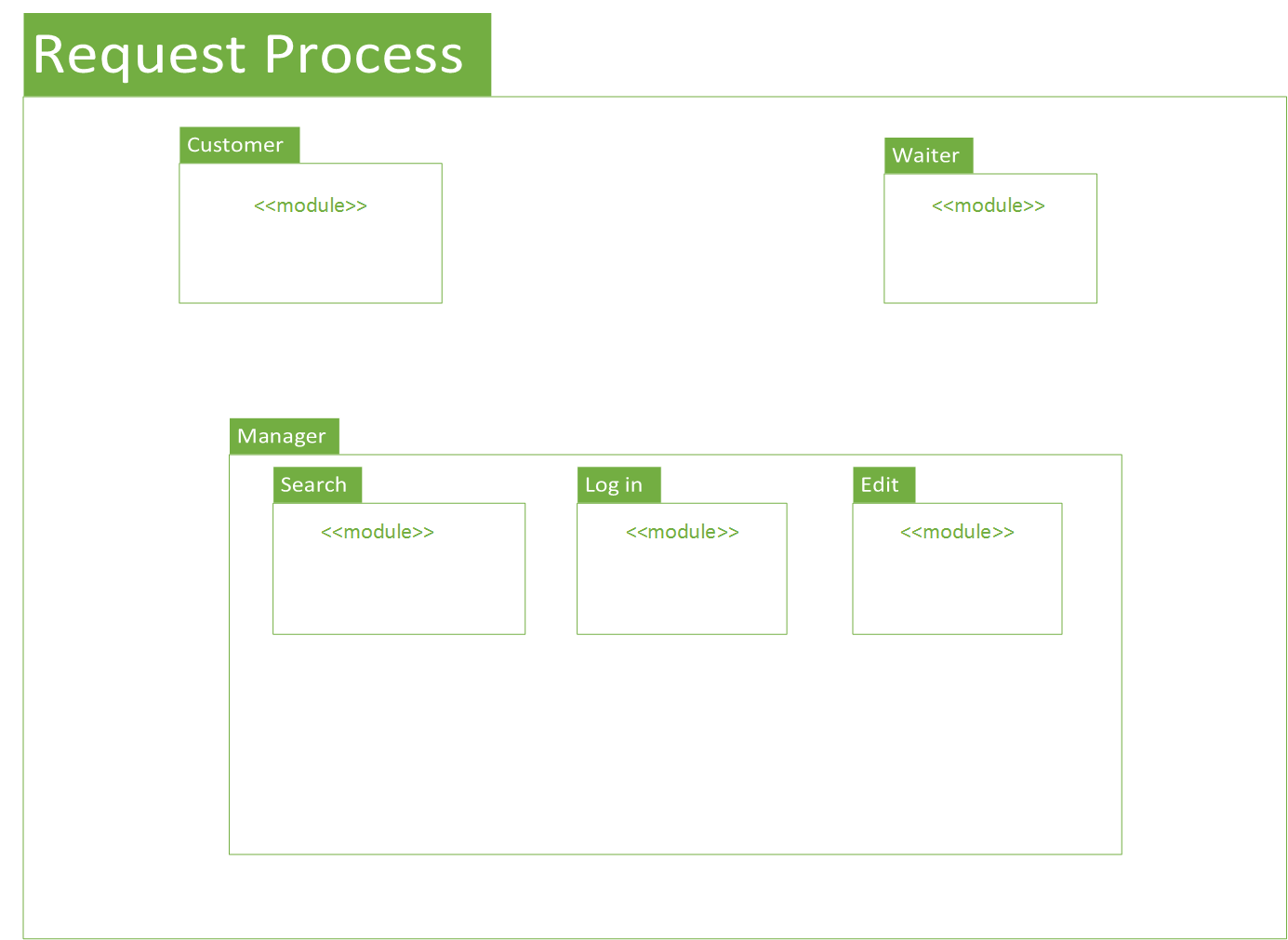
整个系统，客户端以及服务端以http协议进行通信，所以服务端主要是处理来自于客户端的http的请求，以及和数据库的交互。由于整个系统比较小，考虑项目的复杂性，本应在服务器处理层和数据库之间，有一层数据传输层，为求简便，在处理层直接与数据库进行交互。

2.2.2 项目部署图



对于系统部署，服务器采用一台实验室的电脑，这台电脑运行Windows8.1操作系统,8GB内存，4核心CPU，20M的带宽，服务器采用Node.js搭建。数据库架设在云端，运行MySQL 5.6版本，有1核CPU，1GB内存，以及1M的带宽。

2.2.3 模块设计

包图：

服务器的请求处理层，先以请求对象进行划分，划分为customer,waiter以及manager三个模块，customer,waiter模块已经很小，不再进行划分，对于manager，以主要的三个功能划分为三个模块，分为登陆认证功能，获取信息功能以及编辑功能。

2.3. 接口设计

2.3.1 内部接口

服务器端与客户端的所有接口设计，尽力遵循REST API标准。以GET方法以请求数据，以POST方法增加或者编辑数据，以DELETE方法删除数据库里的数据。

实例：  
 1 餐馆管理者的登陆：

API: http://10.60.44.189:3000/manager/login/ post 方法

Post 数据: username,password

2 Android客户端请求菜单信息

API: http://10.60.44.189:3000/menulist/manager\_id/?dish\_type=xxxx Get方法

通过在query域不同的参数，实现对特定餐馆、特定菜品类型的查找。

3 餐馆管理者对菜品进行删除

API: http://10.60.44.189:3000/console/dish/manager\_id DELETE 方法

Request body 数据 : 要删除的菜品的ID.

2.3.2 外部接口

API : Alipay / UnionPay 商户支付接口。

2.4. 界面设计

* **客户端界面设计**

首先完成核心功能的用户界面设计



* + 为完成此功能，需要设计菜品展示视图(菜品名，价格)
  + 提供对该菜品的选择按钮
  + 提供切换菜品类型的TAB
  + 提供选择结束进入结账界面的按钮

随后进行结账界面的设计，这是第二个核心功能

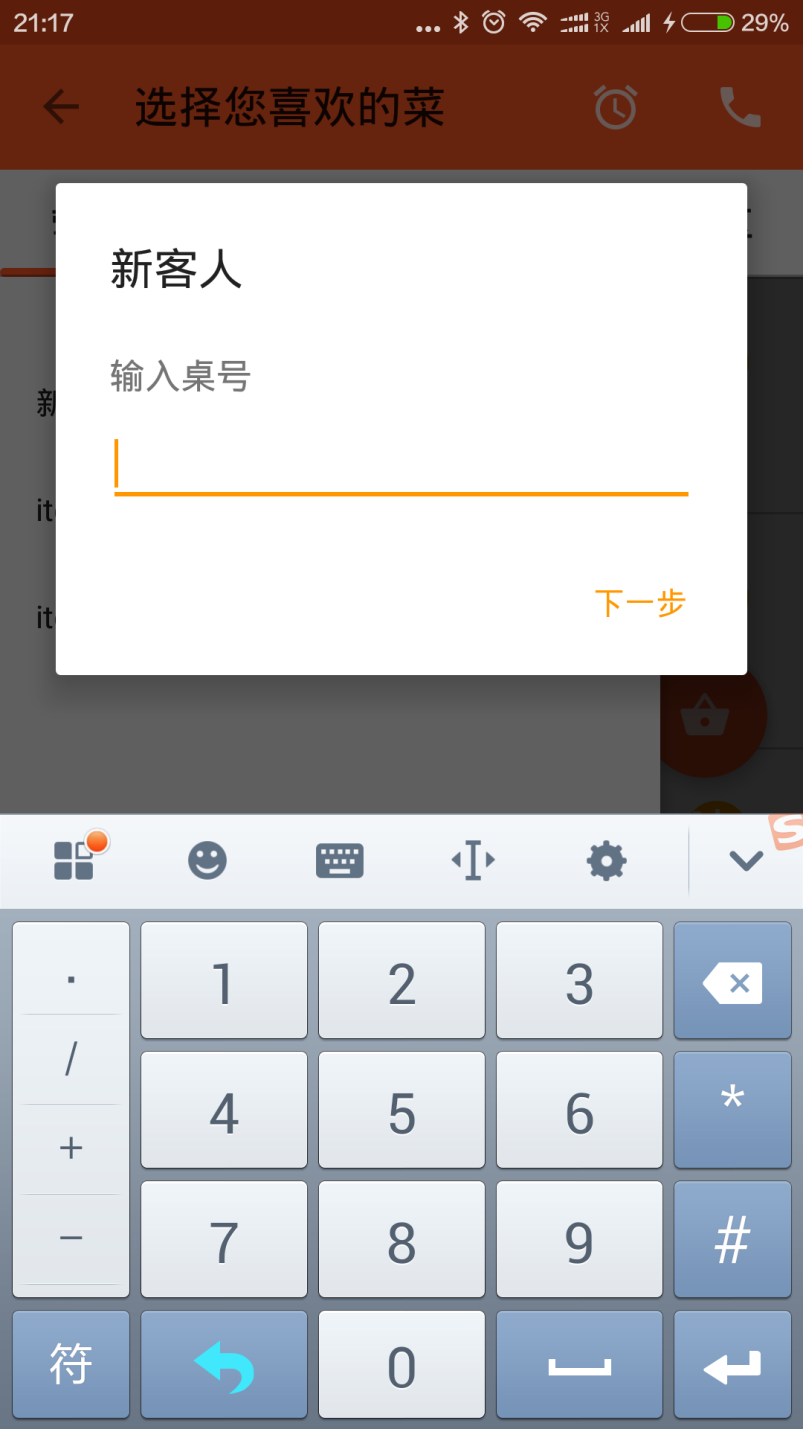


* + 选择对订单数量的修改
  + 显示总价
  + 清空购物车功能
  + 确认提交按钮

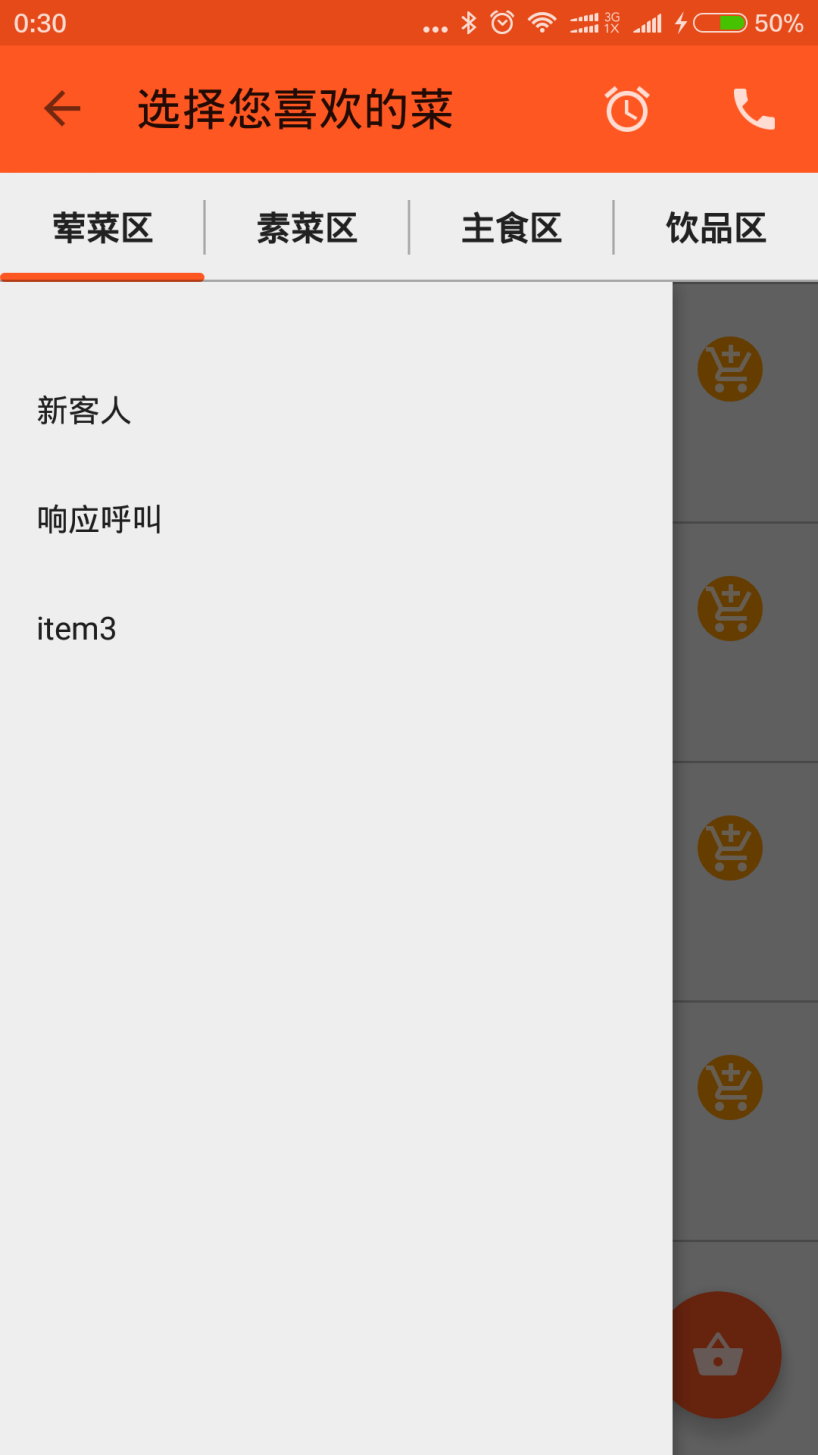
以上完成了系统的核心功能的界面设计，接下来的界面设计主要在完善界面元素和增加与用户的交互性，如：



* + 增加对话框

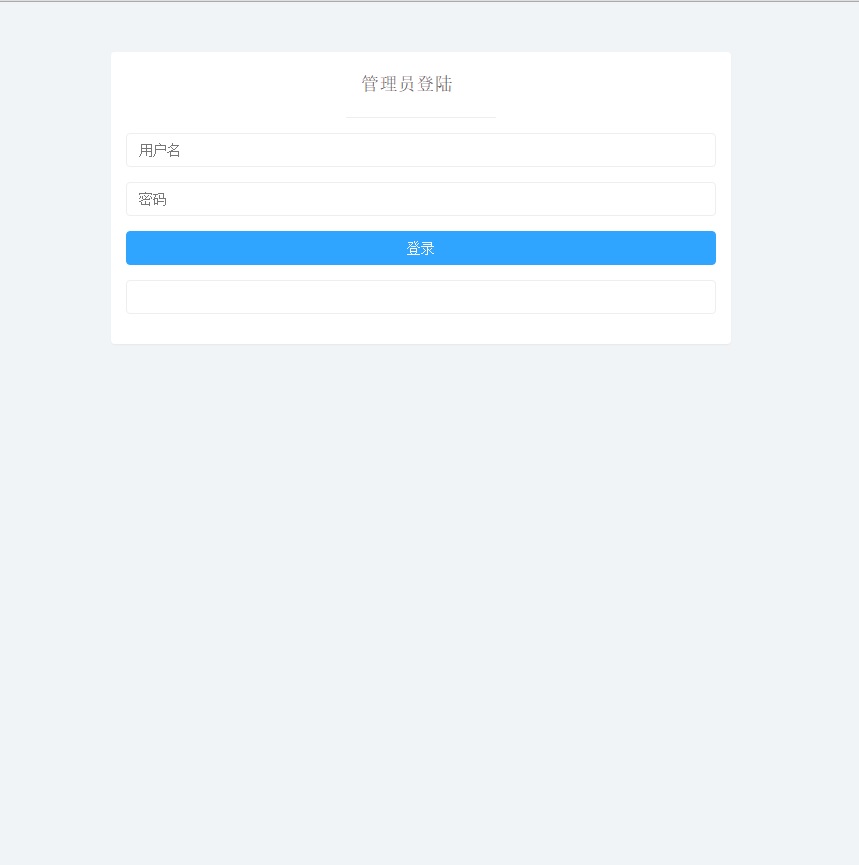


* + 增加输入界面



* + 增加其他非核心功能
* **管理端界面设计**

管理控制台基于网页端实现，经理可在这里直接监控餐馆的运营状况，也可以直接操纵数据库，修改菜单。

****

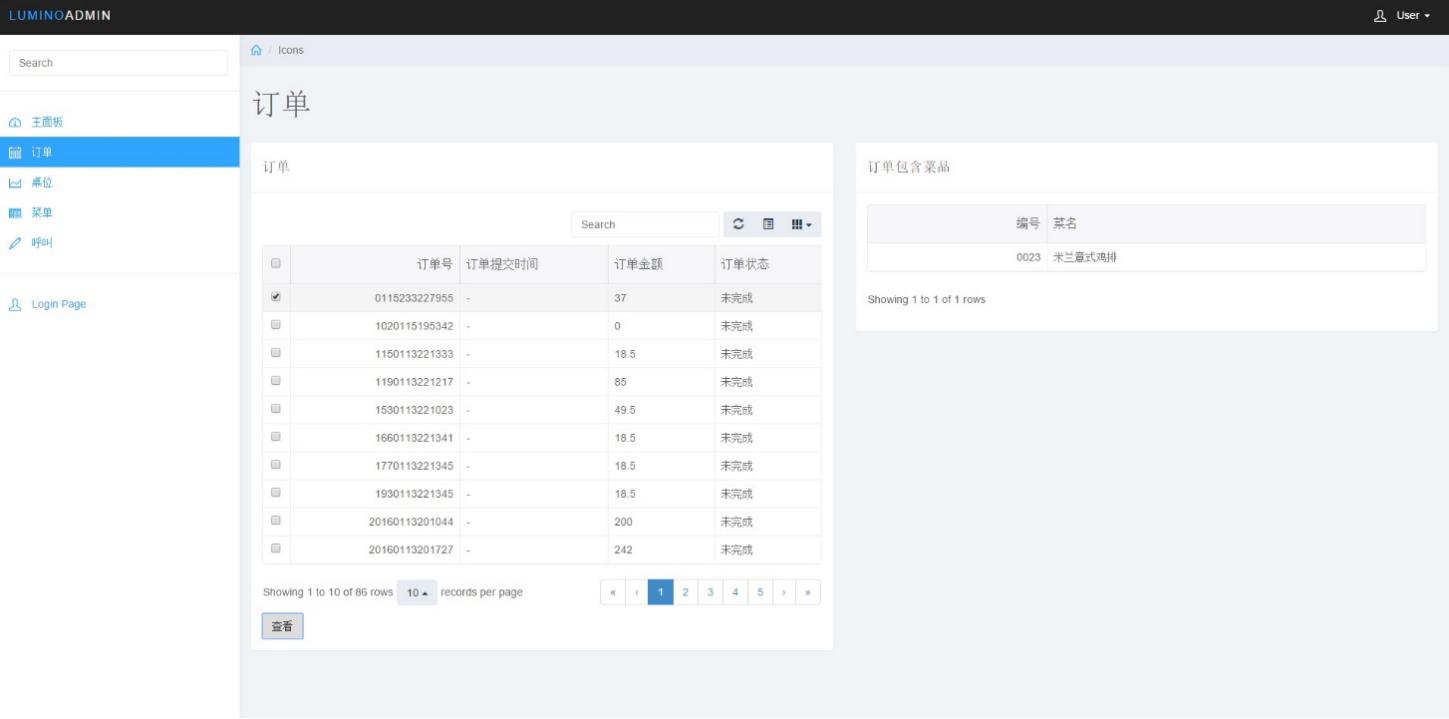
* + **登陆界面**

**管理员在此输入用户名和密码，进入控制台。**

****

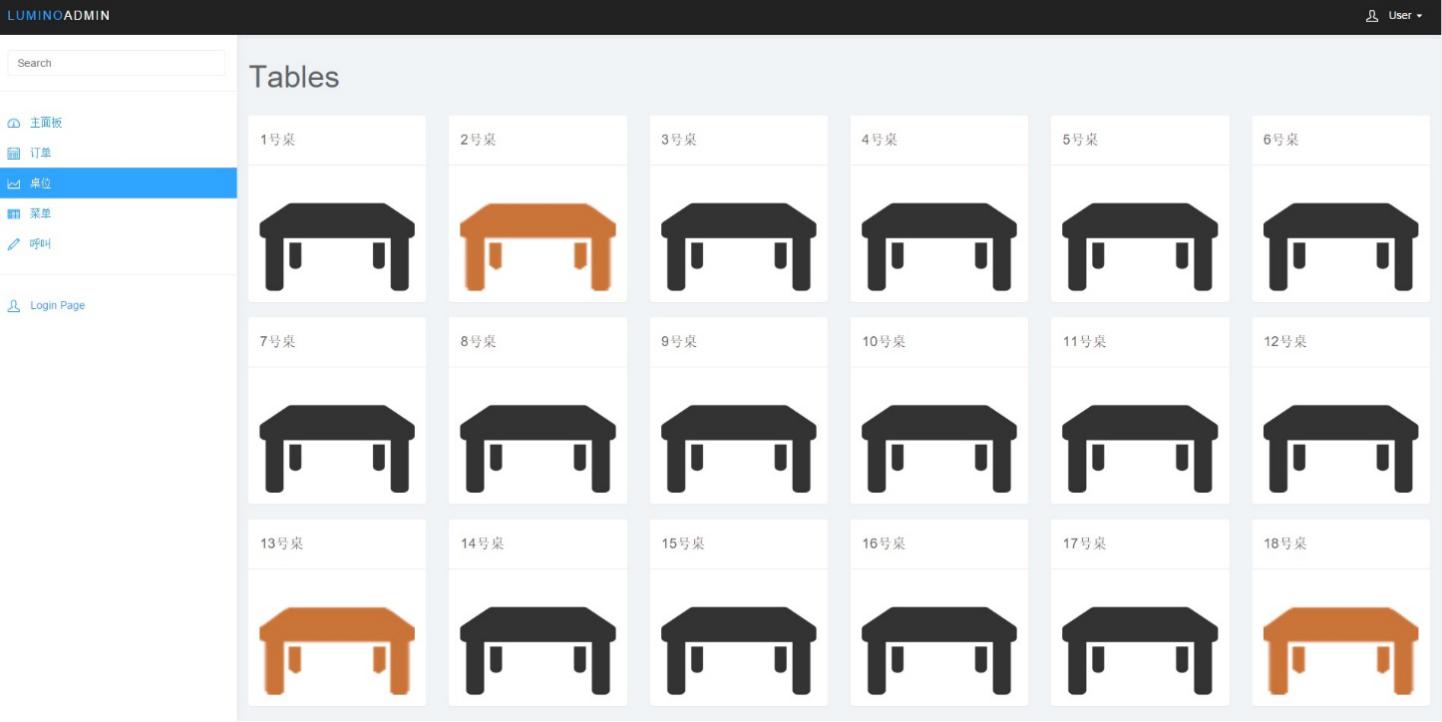
* + **主面板界面**

**主面板包含了餐馆经营信息的概览，包括订单数，顾客呼叫数，空余桌位以及日流水金额。曲线图表示一段时间内的餐馆客流量图。**



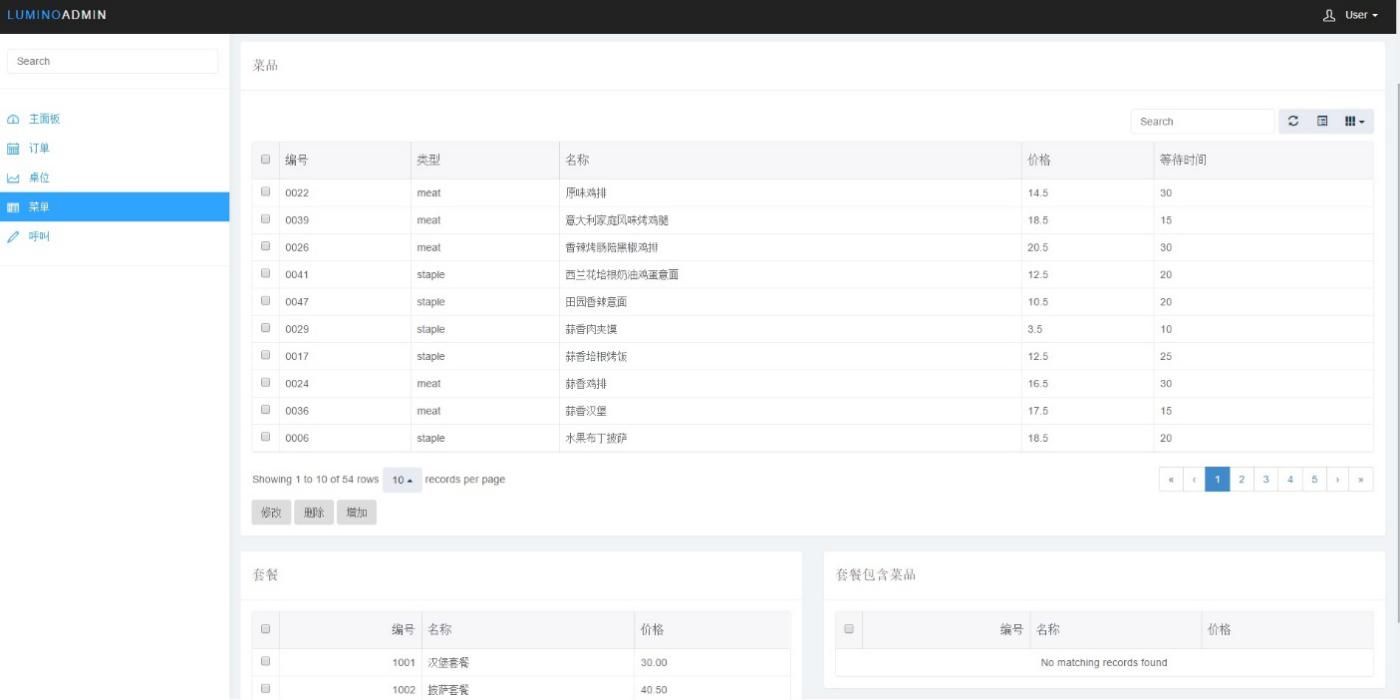
* + **订单界面**

**在此可以看到用户提交的所有订单，并查看订单内包含的菜品。**

****

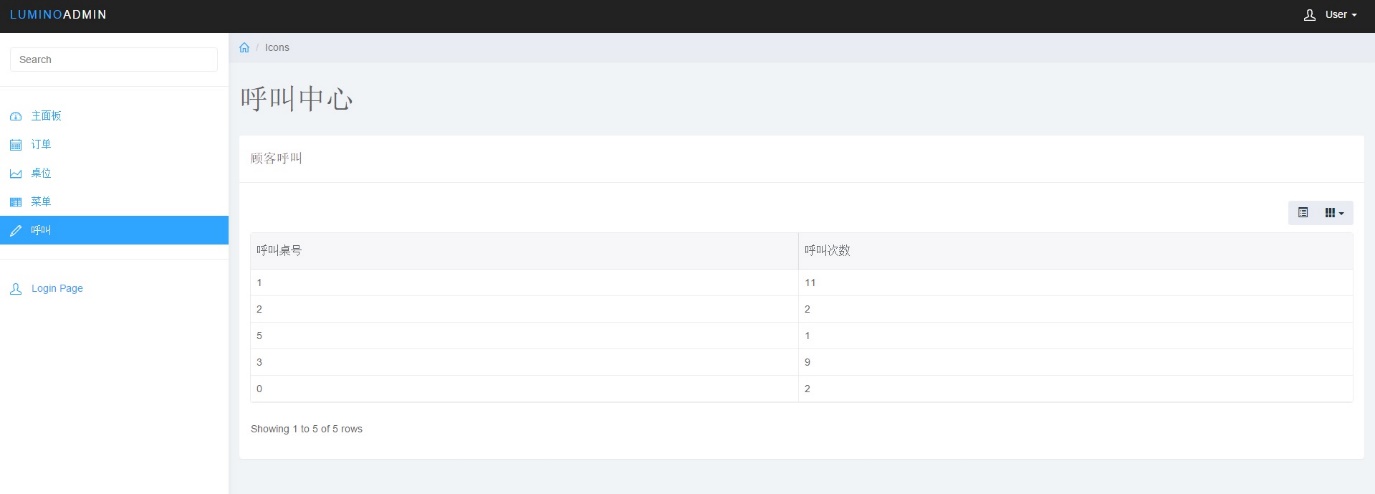
* + **桌位界面**

**桌位界面以图片的方式展示出当前餐馆内的就坐情况。黑色表示空闲，橙色表示当前被占用。**

****

* + **菜单界面**

**菜单界面中，可以查看当前餐馆的菜单，并对菜品和套餐进行增删改查操作。**

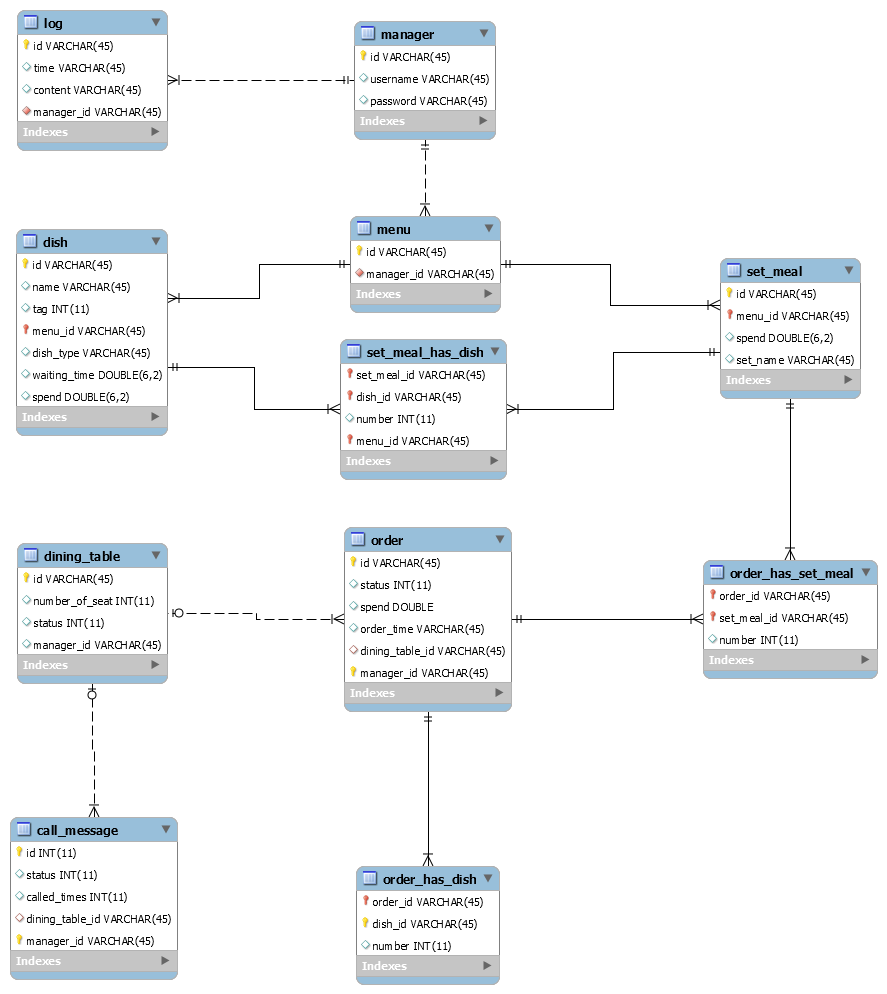
****

* + **呼叫中心**

**当顾客发出请求呼叫服务员时，管理控制台可以看到呼叫信息，并派遣服务员响应顾客的呼叫。**

2.5 数据库设计

根据系统模块的划分，数据库主要围绕顾客点餐模块和经理管理模块涉及到的实体类来建立。最终包含11张表，数据库设计图如下



数据库表设计：

1. log表

log表用来记录餐馆经理对数据库所进行的操作。必要时可作为系统复原或移植的参考。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Id | 主键 |
| Varchar(45) | Time | 日志写入的时间 |
| Varchar(45) | Content | 日志内容 |
| Varchar(45) | Manager\_id | 外键，经理的id |

1. manager表

manager表记录餐馆经理的用户名和密码。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Id | 主键 |
| Varchar(45) | Username | 登录时的用户名 |
| Varchar(45) | Password | 登录密码 |

1. menu表

记录菜单的id号，以及菜单与经理的关联。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Id | 主键 |
| Varchar(45) | Manager\_id | 外键，经理的id |

1. Dish表

记录菜品的详细信息。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Id | 主键 |
| Varchar(45) | Name | 菜品名称 |
| Varchar(45) | Dish\_type | 菜品类型 |
| Double(6,2) | Waiting\_time | 等待上菜所需时间 |
| Double(6,2) | Spend | 菜品单价 |
| Varchar(45) | Menu\_id | 外键，菜单的id |

1. Set\_meal表

记录套餐的详细信息。套餐包含多个菜品，与Dish表通过关联表set\_meal\_has\_dish关联。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Id | 主键 |
| Varchar(45) | Set\_meal\_name | 套餐名称 |
| Double(6,2) | Spend | 套餐的价格 |
| Varchar(45) | Menu\_id | 外键，菜单的id |

1. Set\_meal\_has\_dish表

Set\_meal与dish的关联表。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Set\_meal\_id | 联合主键，套餐id |
| Varchar(45) | Dish\_id | 联合主键，菜品id |
| Int | number | 套餐中包含的特定菜品个数 |
| Varchar(45) | menu\_id | 联合主键，菜单的id |

1. Order表

订单表用来记录顾客提交的订单信息，包括金额与点菜。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Id | 主键 |
| Int | Status | 状态，数字含义参考用力规约（生成订单） |
| Double | Spend | 订单金额 |
| Varchar(45) | Order\_time | 订单生成时间 |
| Varchar(45) | Dining\_table\_id | 外键，订单所在的桌号 |
| Varchar(45) | Menu\_id | 外键，菜单id |

1. Order\_has\_set\_meal表

订单所包含的套餐，是两者之间的关联表。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Order\_id | 联合主键，订单的id |
| Varchar(45) | Set\_meal\_id | 联合主键，套餐的id |
| Int | Number | 订单包含的套餐数量 |

1. Dining\_table表

餐桌表。表示正在就餐的一桌人，记录了餐桌人数与状态等信息。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Id | 主键 |
| Int | Status | 状态，数字含义参考用例规约（设置桌号） |
| Varchar(45) | Number\_of\_seat | 餐桌的座位数量 |
| Varchar(45) | Manager\_id | 外键，经理的id |

1. Call\_message表

表示顾客呼叫服务员的信息。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Id | 主键 |
| Int | status | 状态，数字含义参考用例规约（呼叫服务员） |
| Int | Called\_times | 呼叫次数 |
| Varchar(45) | Manager\_id | 外键，经理的id |

1. Order\_has\_dish表

订单与菜品之间的关联表。

|  |  |  |
| --- | --- | --- |
| 类型 | 字段名 | 备注 |
| Varchar(45) | Order\_id | 订单的id |
| Varchar(45) | Dish\_id | 菜品的id |
| Int | Number | 订单包含的菜品数量 |

2.6 系统出错处理设计

2.6.1 出错信息

2.6.2 补救措施

2.6.3 系统维护设计

* 预留字段，为了以后维护的方便和功能的扩展。在客户端预留了套餐的字段，已经封装了bean文件以及建立与已存在单品的联系
* 输出调试信息：代码中在容易出错和误解的地方输出了调试信息（如封装json语句时增加json字段的结果）
* 丰富的注释。对每一个代码字段进行了功能的描述以及调用关系，为后续维护提供了遍历。