

开发规划

开发人员

学号	姓名	职位
15331369	YYPyyp	技术经理，前端工程师
15331280	sysuts13	产品经理，产品设计
15331342	Exocoetidate-Xuyouri	JavaScript 工程师，后端工程师
14331133	李啟明	后端工程师
15331443	zhuwh9	后端工程师，数据库维护
15331049	chengwch	需求分析与设计

开发计划

序号	项目事件	项目细分	计划启动日期	计划完成日期
1	初始会议	可靠性研究	2018.3.20	2018.3.30
		范围估计	2018.3.20	2018.3.30
		候选架构	2018.3.20	2018.3.30
		估计	2018.3.20	2018.3.30
		商业案例	2018.3.20	2018.3.30
2	细化	功能需求	2018..4.5	2018.4.20
		用例	2018..4.5	2018..4.20
		设计模型	2018..4.5	2018..4.20
		计划	2018..4.5	2018..4.20
		接口	2018.4.15	2018.4.20
3	构造	点餐界面设计	2018.4.25	2018.4.30
		商家界面设计	2018.4.25	2018.4.30
		数据库搭建	2018.4.25	2018.4.30
		服务器搭建	2018.4.25	2018.4.30

		静态前端 点餐界面	2018.5.1	2018.5.15
		静态前端 商家界面	2018.5.1	2018.5.15
		动态前端 点餐界面	2018.5.20	2018.5.30
		动态前端 商家界面	2018.5.20	2018.5.30
		接口完善	2018.6.1	2018.6.5
4	测试	创建菜单	2018.6.10	2018.6.12
		删除菜单	2018.6.10	2018.6.12
		读取菜单	2018.6.12	2018.6.13
		下单	2018.6.13	2018.6.14
		接收订单	2018.6.14	2018.6.15
		读取订单	2018.6.15	2018.6.16
5	发布	打包	2018.6.20	2018.6.20
		部署	2018.6.21	2018.6.21

开发工具

工具	作用
Hbuilder	前端开发
夜神模拟器	真机模拟
Node.js、MongoDB	后端开发

总体设计

系统目标

使用本系统实现以下功能

商家管理系统

1. 创建菜单
2. 删除菜单
3. 读取订单
4. 接收订单

客户点餐系统

1. 读取菜单
2. 用户下单
3. 结算

接口规范

数据库

菜 表

菜名（主键） 价格 类别（用于在用户界面根据类别筛选） 图片路径 描述（字符串）

订单 表

订单号（用户提交订单时在后台生成） 下单时间:"2018-06-18-12:00" 桌号 json 格式的订单内容 商品总数 总价

json 格式的订单内容的示例：

```
{
  menu:[
    {
      menu_name: 螺蛳粉,
      menu_price: 13
      menu_num: 2
    },
    {
      menu_name: 过桥米线,
      menu_price: 11
      menu_num: 1
    }
  ]
}
```

商家创建菜单，存入数据库

菜单格式为 json 对象

```
{
  menu_name :菜名 ...,
  menu_price : 价格...,
```

```
menu_type : 类别...,
menu_description:描述...
menu_picture:图片 64base 编码

}
```

商家删除菜单，从数据库删除

```
商家上传 要删除的 菜名
{
    menu_name:螺蛳粉
}
```

用户提交订单

```
{
    table_num:桌号
    order-time:订单时间。如”2018-06-18-12:00”
    订单号由后台生成。
    menu : [
        {
            menu_name: 螺蛳粉,
            menu_price: 13
            menu_num: 2
        },
        {
            menu_name: 过桥米线,
            menu_price: 11
            menu_num: 1
        }
    ]
    total_num:商品总数
    total_price:商品总价
}
```

后台生成订单号，把订单发给商家

```
订单格式:
{
    order_num:订单号
}
```

```

order-time:订单时间。如"2018-06-18-12:00"
table_num:桌号
menu:[{
    menu_name: 螺蛳粉,
    menu_price: 13
    menu_num: 2
},
{
    menu_name: 过桥米线,
    menu_price: 11
    menu_num: 1
}
]
total_num:商品总数
total_price:商品总价
}
.

```

后台保存着新的订单，商家界面每隔 10 秒向后台请求，若没有新订单，则返回一个空 json 对象；若有新订单，返回订单。

用户打开界面，从后台读取菜单

```

{
    menu:[
        {
            menu_name:
            menu_price:
            menu_type:
            menu_description:
            menu_picture:图片 64base 编码
        }
        {
            menu_name:
            menu_price:
            menu_type:
            menu_description:
        }
        .....
    ]
}

```

商家打开界面，从后台读取菜单和历史订单

菜单的数据格式如上。

订单格式：

```
{
  orders: [
    {
      order_num: 订单号
      order-time: 订单时间。如"2018-06-18-12:00"
      table_num: 桌号
      menu : [{
        menu_name: 螺蛳粉,
        menu_price: 13
        menu_num: 2
      },
      {
        menu_name: 过桥米线,
        menu_price: 11
        menu_num: 1
      }
    ]
    total_num: 商品总数
    total_price: 商品总价
  },
  {
    order_num: 订单号
    order-time: 订单时间。如"2018-06-18-12:00"
    table_num: 桌号
    menu : [{
      menu_name: 螺蛳粉,
      menu_price: 13
      menu_num: 2
    },
    {
      menu_name: 过桥米线,
      menu_price: 11
      menu_num: 1
    }
  ]
    total_num: 商品总数
    total_price: 商品总价
  }
}
```

```
.....  
    ]  
}
```

技术选型及理由：

前端：

使用 jQuery 框架；

理由：jQuery 是一个快速、简洁的 JavaScript 框架，它封装 JavaScript 常用的功能代码，提供一种简便的 JavaScript 设计模式，优化 HTML 文档操作、事件处理、动画设计和 Ajax 交互。jQuery 具有独特的链式语法和短小清晰的多功能接口；具有高效灵活的 css 选择器进行扩展；拥有便捷的插件扩展机制和丰富的插件。并且 jQuery 兼容各种主流浏览器，如 IE 6.0+、FF 1.5+、Safari 2.0+、Opera 9.0+等。

后端：

使用 Express+Node.js+MongoDB；

理由：Express 是一个基于 Node.js 平台的极简、灵活的 web 应用开发框架，丰富的 HTTP 快捷方法和任意排列组合的 Connect 中间件，让后端开发时创建健壮、友好的 API 变得既快速又简单。Express 不对 Node.js 已有的特性进行二次抽象，只是在它之上扩展了 Web 应用所需的基本功能。

架构设计：

用例设计：

消费者用例：

3. 点餐

主成功场景

点击商品列表浏览菜品，可直接点击搜索输入菜名进行浏览，添加需要的菜品，可取消多点的菜品，点餐完成后提交订单，支付后完成订单。

交替场景

订单提交失败：选择重新提交或取消订单。

余额不足支付失败：30 分钟内完成支付或取消订单。

4.取消订单

主成功场景：

消费者点击我的订单进入订单列表，进入需要取消的订单，填写取消理由后点击取消订单后等待系统处理，若符合取消条件（菜品未制作）则系统取消订单，退回货款。

交替场景：

菜品制作中或已制作完成：拒绝取消订单或与商家联系请商家在后台取消订单。

商家用例：

1. 增加菜品

商家点击添加菜品，填写菜名，价格，菜品描述，参考图片等信息后提交，菜品添加成功。

2. 删除菜品

点击需要删除的菜品，进入菜品详情界面，点击删除菜品，确认后菜品删除。

3. 修改菜品

点击要修改的菜品，进入菜品详情界面，点击编辑菜品信息，对需要修改的部分重新编辑后提交，确认后修改成功。

4. 查询菜品

浏览菜品列表，点击需要查询的菜品进入菜品详情界面。或直接搜索菜名进入菜品详情界面。

5. 处理订单

对消费者提交的订单，选择接受或拒绝订单，接受后订单进入待制作订单列表。

拒绝订单后由系统退款并删除订单，订单进入已拒绝订单列表。

菜品制作完成后点击菜品制作完成，菜品进入待配送列表。

菜品配送至消费者手中，消费者确认后订单完成，进入已完成订单列表，否则 7 天后订单自动完成。

6.取消订单

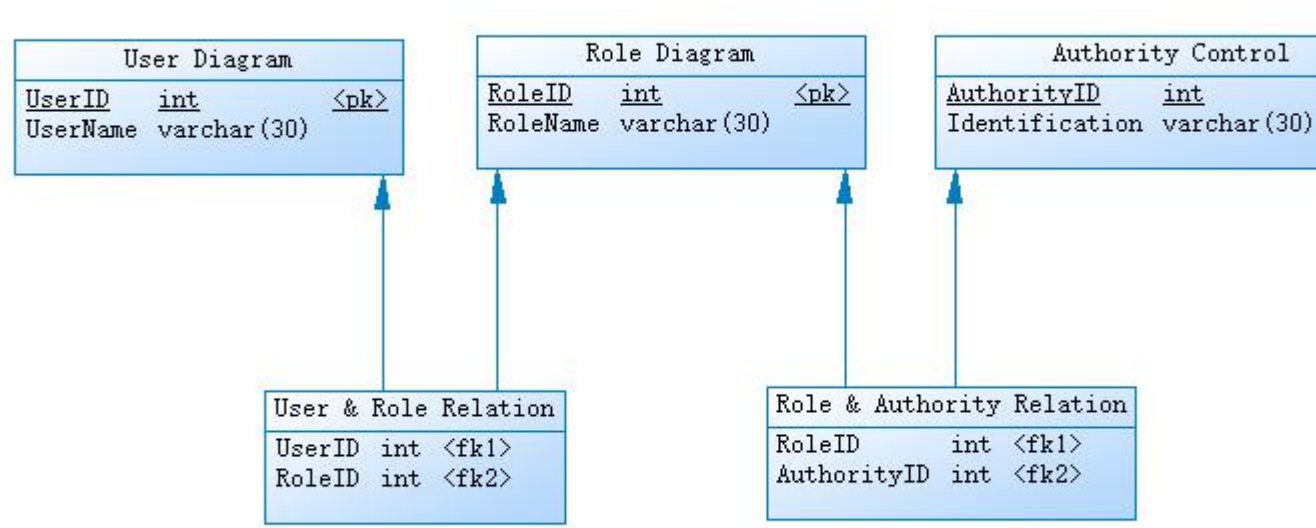
对已接受的订单进行取消，进入退款界面，退款完成后等待系统处理，成功后订单进入已取消订单列表。

7.订单查询

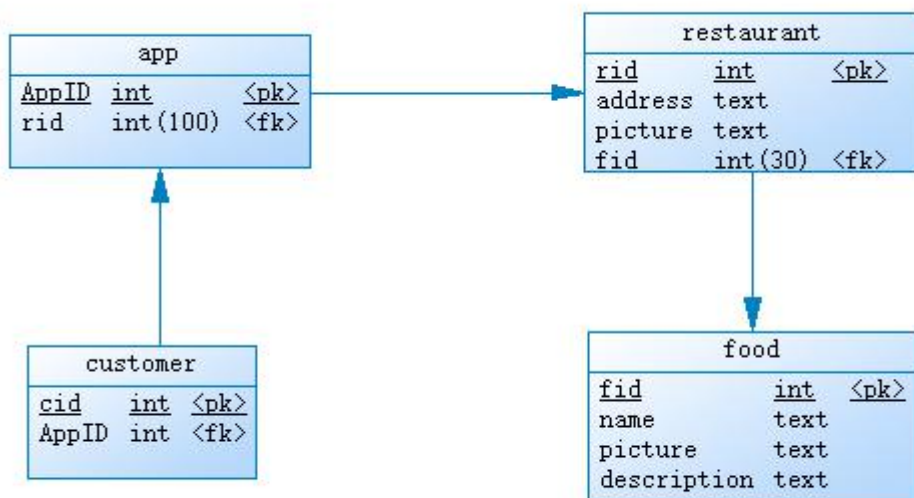
点击订单进入订单列表浏览订单，并分为已拒绝订单，已取消订单，待制作订单，待配送订单，已完成订单 5 个子列表，点击后进入相应的子列表进行查询，

数据库设计：

用户及权限系统数据库设计

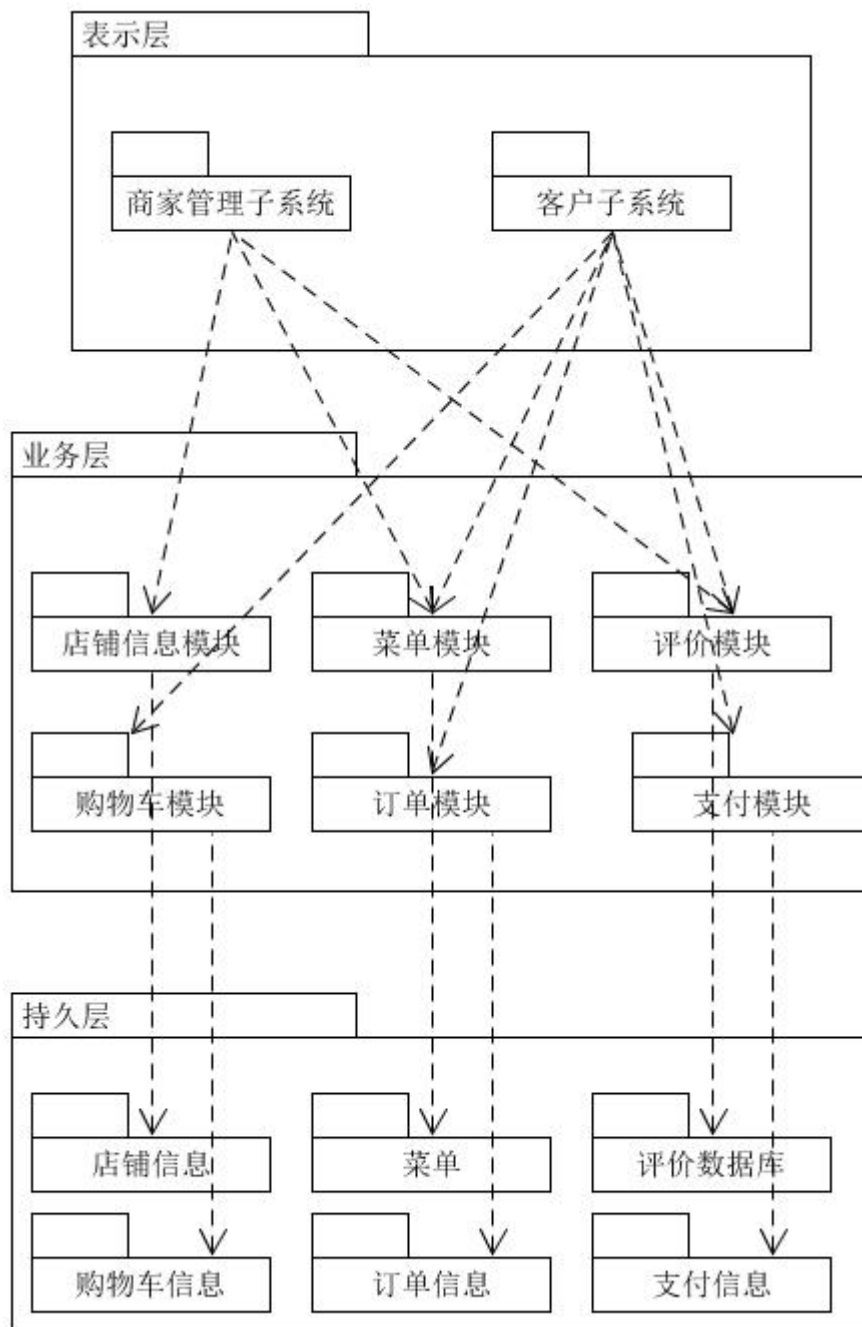


订单系统数据库设计

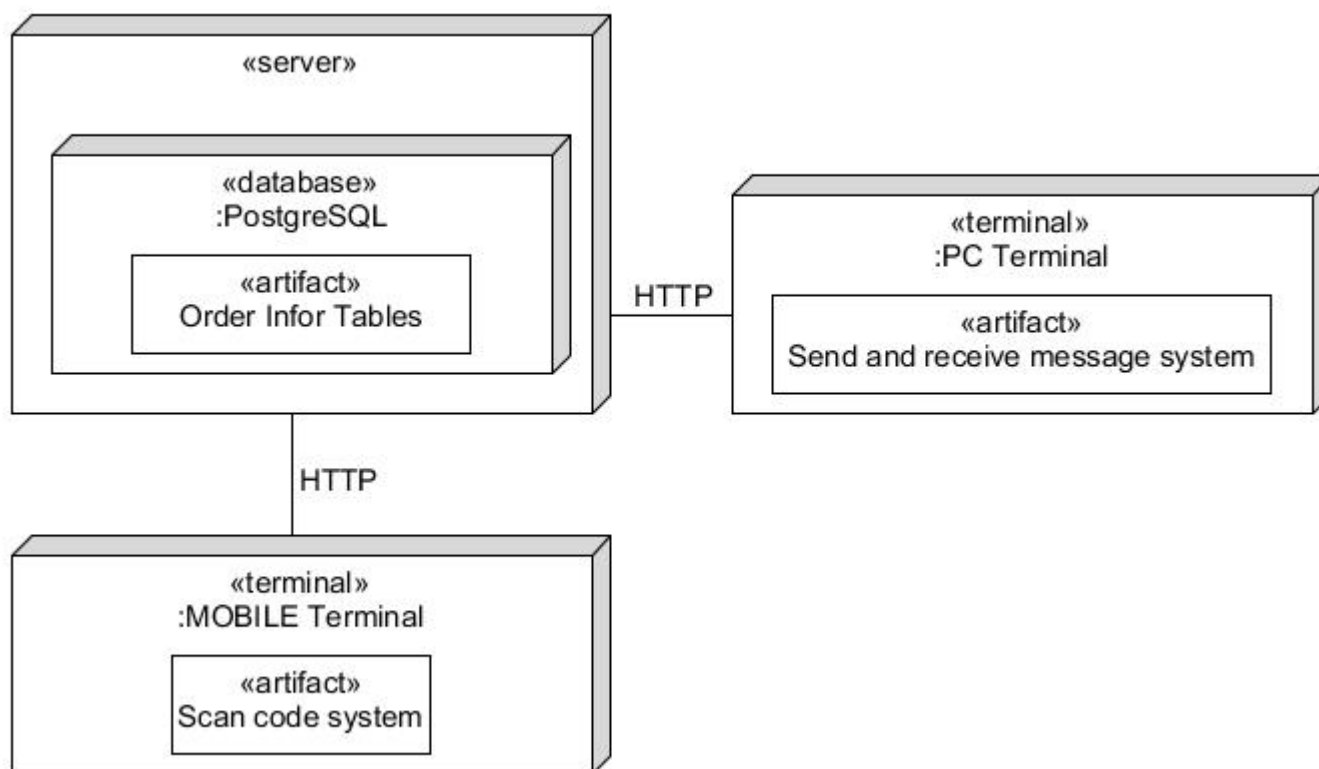


系统设计:

逻辑视图:



物理视图：理解为部署视图。完整的系统工作需要 3 个部分：系统服务器、商家计算机、顾客手机。系统被部署在系统服务器上，商家计算机用来修改菜单并接受来自顾客的订单信息，顾客手机用来查看菜单并提交订单信息。



模块划分：

一级模块	二级模块	功能简述	优先级
系统管理	商家		高
	桌子号码		高
	系统参数		高
基础数据管理	菜单信息		高
	订单信息		高
	菜单类别		中

通讯服务器系统

一级模块	二级模块	功能简述	优先级
通信服务	接收订单	接收来自客户的订单	高
	发送订单	将订单发给商家	高
	创建菜单	商家创建菜单	高
	读取菜单	客户读取菜单	高

商家系统

一级模块	二级模块	功能简述	优先级
------	------	------	-----

创建菜单			高
删除菜单			高
读取菜单		读取所有菜单	高
接收订单			高
读取订单		读取历史订单	高

客户系统

一级模块	二级模块	功能简述	优先级
读取菜单		从数据库读取菜单	高
用户下单			高
读取订单		从数据库读取订单	高

软件设计技术：

Design pattern:

Design pattern 是一套被反复使用、经过分类编目的、代码设计经验的总结。使用 Design pattern 是为了可重用代码、让代码更易让他人理解、保证代码可靠性。

Design pattern 原则： 开闭原则，里氏代换原则，聚合复用原则，接口隔离原则等。

Design pattern 在项目中的具体体现：

1.使用 mui 框架来追求最接近原生 APP 体验。

例如：

(1)选项卡模块

```
<div class="mui-segmented-control">
  <a class="mui-control-item mui-active" href="#item1">点菜</a>
  <a class="mui-control-item" href="#item2">商家</a>
</div>
```



(2)滚动模块

```
<div class="mui-scroll-wrapper mui-slider-indicator mui-segmented-control mui-segmented-control-inve:
  <div class="mui-scroll" style="height: 130px;">
```

横向滚动：



竖向滚动：

