**客户名字和logo**

餐厅订餐系统-概要设计说明书

LUO ZHI

日期：2015-01-12

文档版本历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修订日期 | 修订人 | 审核人 | 变更内容 |
| 0.1 | 2015-01-08 | LUO ZHI |  | 初始化 |
|  | 2015-01-11 | LUO ZHI |  | 编写3.2内容 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 文档介绍 1](#_Toc408771879)

[1.1 文档目的 1](#_Toc408771880)

[1.2 文档范围 1](#_Toc408771881)

[1.3 缩写词列表 1](#_Toc408771882)

[1.4 参考内容 1](#_Toc408771883)

[2 系统架构 2](#_Toc408771884)

[3 系统模块 3](#_Toc408771885)

[3.1 权限管理模块 3](#_Toc408771886)

[3.1.1 模块概述 3](#_Toc408771887)

[3.1.2 功能划分 3](#_Toc408771888)

[3.1.3 使用技术 4](#_Toc408771889)

[3.2 移动点菜模块 4](#_Toc408771890)

[3.2.1 模块概述 4](#_Toc408771891)

[3.2.2 功能划分 4](#_Toc408771892)

[3.2.3 使用技术 4](#_Toc408771893)

[4 系统模块流程/关系 4](#_Toc408771894)

[5 系统接口实现技术 4](#_Toc408771895)

[5.1 REST风格的Web-Service 5](#_Toc408771896)

[5.2 JSON数据格式 5](#_Toc408771897)

[5.3 安全性 5](#_Toc408771898)

[5.4 接口协议 6](#_Toc408771899)

[5.4.1 获取用户信息 6](#_Toc408771900)

[5.4.2 获取目前所有分类列表 6](#_Toc408771901)

[5.4.3 获取某一个分类下面所有菜品列表 7](#_Toc408771902)

[5.4.4 查询某个会员的历史订餐记录 8](#_Toc408771903)

[5.4.5 查询某个订单详情 9](#_Toc408771904)

[6 数据库设计 10](#_Toc408771905)

[6.1 数据字典 10](#_Toc408771906)

# 文档介绍

## 文档目的

文档基于架构设计说明书与需求规格说明书，对系统模块及其子功能进行定义，对系统数据字典进行描述，对Web-Service接口协议进行规范，开发团队基于本文档对系统模块功能的具体实现进行详细设计文档编写。

## 文档范围

本文的仅对系统模块功能定义，使用技术，接口协议与数据字典进行描述，系统架构，需求以及具体实现均不在本文档范围内。

## 缩写词列表

|  |  |
| --- | --- |
| 缩写词 | 解释 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 参考内容

# 系统架构

系统机构和技术方案在*系统架构说明书*中有详细阐述，根据系统架构说明书总结系统开发技术方案如下表，

|  |  |
| --- | --- |
| **MVC** | Hibernate 3.x |
| Spring 3.x |
| Struts2 |
| **WEB** | Bootstrap 3 |
| JQuery |
| **移动应用方案** | Android 4.x |
| **Web-Service** |  |
| **数据库** |  |

# 系统模块

根据需求规格说明书中的需求定义，餐厅订餐系统可以分为移动点餐模块，后台收银模块，菜单管理模块，会员管理模块和用户权限管理模块共五个模块，系统模块图下图所示



# 权限管理模块

## 模块概述

权限管理模块用于维护系统用户，系统权限，系统角色的数据以及三者之间的关系。系统权限唯一映射了对系统某一功能的使用权力，例如添加用户这个功能；那么系统角色则定义了一类有相关性或者依赖性的功能集合，例如管理员角色能使用权限管理模块中的所有功能；而系统用户是系统角色的拥有者，通过角色获得系统功能使用的权利。

## 功能划分

### 用户登录

用户使用用户名和与用户名匹配的密码登录到系统，根据用户拥有的角色获得不同的系统菜单；使用无效用户名或者错误的密码，会得到系统的错误信息提示

### 创建/编辑用户

拥有权限管理模块功能使用权限的系统用户，使用这个功能创建新的用户和更新现有用户信息

### 用户角色分配

拥有权限管理模块功能使用权限的系统用户，使用这个功能为系统用户分配相应的角色。一个用户可以拥有多个角色，默认新建用户获得会员这一角色

### 创建/编辑角色

拥有权限管理模块功能使用权限的系统用户，使用这个功能创建新的角色和更新现有角色信息。一个角色代表了一组相关的权限集合

### 角色权限分配

拥有权限管理模块功能使用权限的系统用户，使用这个功能为系统中的角色分配相应的权限。一个角色可以拥有多个权限，但是最好是将有相关性的权限分配给一个角色

### 创建/编辑权限

拥有权限管理模块功能使用权限的系统用户，使用这个功能创建权限和对系统鲜有权限信息进行更新

## 使用技术

* Hibernate
* Struts2
* Spring
* JSP with Tiles
* Bootstrap
* JQuery

# 移动点菜模块

## 模块概述

移动点菜模块用于在手持设备上完成菜单浏览，点菜，下单和查看会员历史菜单功能。

## 功能划分

### 浏览菜单

用户通过移动客户端浏览菜单分类，以及分类下菜品详情。移动客户端通过Web-Service接口获取需要的菜单分类和菜品详情信息

### 点菜

用户通过移动客户端将客户需要的菜品加入当前订单

### 编辑订单明细

用户通过移动客户端，可以修改菜品数量，添加/删除菜品

### 订单下单

用户通过移动客户端对当前订单进行确认，输入服务员员工号，桌号和客户会员打折信息，将订单提交后台系统。客户端通过Web-Service接口提交订单数据

### 浏览会员订单历史

用户通过移动客户端输入会员ID（手机号）则可以获得该会员近三个月的消费历史订单和订单详情。客户端通过Web-Service接口获取历史订单信息

## 使用技术

* Hibernate
* Struts2
* Spring
* JSP with Tiles
* Bootstrap
* JQuery
* Android
* Web-Service

# 菜单管理模块

## 模块概述

## 功能划分

### 创建/编辑菜单分类

### 创建/编辑菜品

### 浏览菜单分类

### 浏览菜单明细

### 浏览菜品明细

## 使用技术

# 后台收银模块

## 模块概述

## 功能划分

### 订单结帐

### 打印账单

### 浏览流水

## 使用技术

# 会员管理模块

## 模块概述

## 功能划分

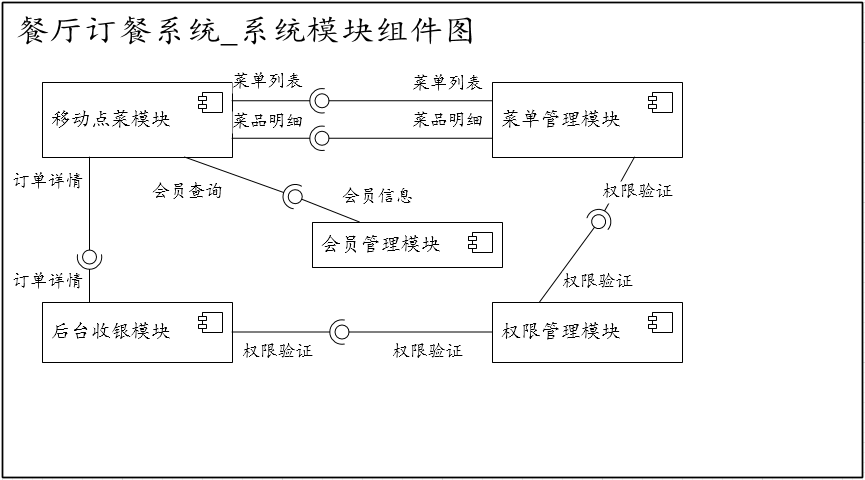
### 创建/编辑会员等级

### 查询会员等级信息

### 编辑用户等级信息

## 使用技术

# 系统模块流程/关系



# 系统接口实现技术

餐厅订餐系统五大模块之间没有明显的互相调用，所以模块与模块之间没有明显的接口定义。由其他模块输入的数据，均有使用模块直接访问数据库获取。

餐厅订餐系统的移动点菜模块物理上分为移动客户端和后台Web-Service服务，移动客户端通过消费后台Web-Service服务完成点菜流程。后台Web-Service使用REST风格定义，并使用JSON作为数据格式。

## REST风格的Web-Service

REST风格中文直译是含状态传输（Representational State Transfer），与传统的SOAP和XML-RPC等方式相比更加简洁与容易理解。REST风格扩充使用了HTTP协议中不常被使用的PUT和DELETE这2个动词（相对于GET和POST更为大家熟悉而言）。

REST风格使用URI来定义资源，每一个资源都有唯一的标识符；使用HTTP协议提供的GET/POST/PUT/DELETE四种方法来定义对资源的动作。恰好这四种方法又可以映射到数据库操作的CRUD四种基本操作。下表是一个简单的REST风格举例，

|  |  |  |  |
| --- | --- | --- | --- |
| 资源 | | User | |
| 动作 | GET | <http://host/ws/users> | 获取所有用户集合 |
| GET | <http://host/ws/users/1> | 获取ID为1的用户 |
| POST | <http://host/ws/users> | 创建一个新用户 |
| PUT | <http://host/ws/users/1> | 更新ID为1的用户数据记录 |
| DELETE | <http://host/ws/users/1> | 删除ID为1的用户数据记录 |

## JSON数据格式

JSON是一种类似于XML的数据表示格式，但是比XML更为简洁和轻量，它没有XML众多的Element节点，也不需要进行Validation，尤其JSON与JavaScript的天然集成，有效提高了开发效率。

在JSON的世界里，只有对象。JSON使用花括弧{}和键值对来定义对象，下面是一个使用JSON来表示用户对象的一个简单例子，

*{*

*“username”: “Jerry Gao”,*

*“password” : “12345”,*

*“mobile\_phone”: “13312345678”*

*“roles”: [*

*{*

*“role\_name”: “admin”*

*},*

*{*

*“role\_name”: “cashier”*

*}*

*]*

*}*

## 安全性

使用HTTPS加密通道以及BASE64加密算法对数据进行保护。