**餐厅到店点餐系统概要设计**

**（版本v1.0.0）**

中软国际信息技术有限公司

2015年10月17日



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 成文信息 | | | | | | | |
| 主题词： | | 概要设计 | | | | | |
| 作 者： | | 中软国际ETC | | 文档类别： | | | 技术方案、设计文档 |
| 审 核： | |  | |
| 批 准： | |  | | 文档性质： | | | 正式稿 |
| 主 送： | |  | | 存档日期： | | |  |
| 抄 送： | |  | | 发布日期： | | |  |
| 签收信息 | | | | | | | |
| 发送方 | | | 接收方： | 接收方： | | 接收方： | |
| 代表人： | | | 代表人： | 代表人： | | 代表人： | |
| 日期： | | | 日期： | 日期： | | 日期： | |
| 变更信息 | | | | | | | |
| 版本 | 原因 | | | | 作者 | 日期 | |

目录

第一章 引言 1

1.1 编写目的 1

1.2 背景 2

1.3 定义 3

1.4 参考资料 3

第二章 总体设计 4

2.1 需求规定 4

2.2 运行环境 6

2.2.1 运行环境网络结构说明 7

2.2.2 运行环境说明细则 8

2.2.3 支持软件 8

2.3 基本设计概念和处理流程 8

2.3.1 整体流程 8

2.3.2 设计概念 9

2.4 结构设计 19

2.4.1 功能模块划分 19

2.5 人工处理过程 47

第三章 接口设计 48

3.1 数据请求与返回接口 48

第四章 系统出错处理 50

4.1 出错信息 50

4.2 补救措施 50

4.3 系统维护设计 51

# 引言

## 编写目的

通过对中小餐饮企业工作流程的深入研究，点餐系统项目组发现制约餐饮企业工作效率的各种因素中最主要是普遍使用手工登记菜谱的现象。点餐系统运用计算机软件系统来实现餐饮企业工作流的信息化管理，采用面向对象的开发方法以及Java语言，依靠Eclipse作为开发工具,使用MySQL数据库存储数据。实现了客户点餐、厨师配餐、收银管理、系统设置等功能，解决餐饮企业中客户点餐速度慢、服务员登记不清晰、厨师配餐漏配、菜谱更新不及时且不直观、会员管理混乱的现象。

本设计书是从总体上把握餐厅到店点餐系统设计框架，包括模块划分、处理流程和接口设计，设计书对上述内容作了总体描述，体现了餐厅到店点餐系统用户需求与应用系统实现之间的关系，在设计过程中起到了提纲挈领的作用，最终实现一下目的：

1、对系统概要设计的阶段任务成果形成文档，以便阶段验收、评审，最终的文档验收。

2、对需求阶段的文档再次确认过程，对前一阶段需求没有做充分或错误的提出修改。

3、明确整个系统的功能框架和数据库结构，为下一阶段的详细设计、编码、和测试提供参考依据。

4、明确编码规范和命名规范，统一程序界面。

本设计书预期读者：系统详细设计人员；系统开发人员；软件维护人员；技术管理人员；执行软件质量保证计划的专门人员；参与本项目开发进程各阶段验证、确认以及负责为最后项目验收、鉴定提供相应报告的有关人员。

## 背景

在我们这个“食文化”高度发达的国度里，餐饮业毋庸置疑地成为第一大传统产业。然而，正是这历史悠久、规模庞大的产业，却也是信息化程度最慢和最低的产业之一。至今为止，绝大多数餐饮企业的经营方式仍然停留在“一张纸、一支笔、经理喊破嗓、员工跑断腿”的境地，其本质仍然是以人脑计算为基础，不但工作量巨大，而且极易出错。市场上常见的餐厅场景是：服务员进进出出地飞奔于户外桌台、大堂吧台和厨房之间，而餐厅管理员则用计算器算得满头大汗的场景，蔚为壮观。且不说这种日复一日的重复、单调的劳动意义何在，单就消费计算方面的出错而引发的与客户之间的矛盾纠纷，就不可胜计。

其实在人类文明已经进入电子信息的时代，绝大多数重复、单调、可程式化的工作都已经可以交由计算机来完成。面对复杂和庞大的计算任务，计算机在效率和准确度方面是人脑所不能望其项背的。作为传统产业的餐饮业，同其他很多传统产业一样，同样可以引入信息化改造。

餐饮业的信息化改造内容，主要有：由传统的笔和纸点餐，改用电子点餐；由传统的人工计算、统计、管理，改用电脑进行计算、统计和管理；将现代的电子技术、无线技术、计算机技术和网络技术引入到餐饮行业中来，替代部分人工操作，加快餐饮企业的运转效率，提升餐饮企业的管理水平，使餐饮企业运作更规范、更科学、更高效。

在餐饮企业的信息化改造当中，点餐系统，成为其核心内容，也是餐饮企业进行信息化改造的必由之路。本系统实现的目的在于通过餐饮企业的信息化改造替代传统的纸和笔进行电子点餐（点菜）操作，实现网络化信息即时传递。系统也能通过标准的URL和XML数据接口，集成到企业综合管理系统中，使餐饮企业真正实现全面信息化。

待开发软件系统名称：中软国际-餐厅到店点餐系统

项目任务提出者：公共事业产品部

项目开发者：ETC

项目用户：各类型餐厅经营者

项目运营方式：单一客户独立部署运营

## 定义

MIS: 管理信息系统

系统: 管理信息系统

## 参考资料

《到店点餐系统需求规格说明书》

# 总体设计

## 需求规定

(1) 对功能的规定

在充分理解《餐厅到店点餐系统需求规格说明书》的基础上，根据各类型餐厅实现点餐、传餐、结算等通用流程的需要，设计此方案，来最大限度满足各类型餐厅运营的需要。

餐厅到店点餐系统包括三个部分：餐厅管理员、餐厅点餐服务员、餐厅后厨人员。系统的结构框图如下：

餐厅到店点餐系统

餐厅管理员

餐厅点餐服务员

餐厅后厨人员

员工管理

菜品管理

结单管理

历史数据

发布公告

结账

详情

免单

实时获取结算订单

获取实时公告

点餐

结餐

列表

详情

更新

接收实时传菜消息

推送后厨

保存订单

烹制状态确认

推送传菜消息

获取实时点菜列表

(2) 对性能的规定

精度：本软件数据传输、处理时数据要求保留2位小数点，超过两位以上小数采取四舍五入方式处理。

时间特性的要求：对于任何功能，数据的处理、响应、展现超时不应该超过1秒钟。

灵活性：系统在设计之初要考虑到系统以后的扩展，除了采用层次结构，保证功能实现与通讯接口最大限度的独立，在与通信平台的接口设计上尽量标准化，以实现和新系统的无缝连接。当客户端有更灵活的需求时，应该能够在不该服务端的情况下方便的扩展Android、IOS等原生APP客户端。

(3) 输入/输出要求

输入：普通文本数据使用操作人员键盘输入，涉及到的数字数据精度要求保留2位小数，用户头像及菜品图片部分为图片文件导入输入。

输出：系统所有交互结果均由屏幕输出。

(4) 数据管理能力要求

根据不同的需要，可以对数据库采取实时热备份或定时备份的策略。

(5) 故障处理要求

客户端硬件故障：一般性的计算机硬件故障,不能启动、硬件损坏，最好有备用机器。

通讯故障：客户端与服务器无法通讯，导致局部业务暂停，应检查通讯线路及设置，保证网络通讯畅通。

服务器硬件故障：web服务器站无法启动，导致办公业务暂停，应有备件或备机替代。

数据服务器硬件故障：数据无法访问，业务暂停，应有备件或备机替代。

客户端应用程序故障：部分业务处理不正常，应检查设置或浏览器等配套设置。

后台服务器故障：部分业务处理不正常，应检查设置或更换配套程序。

(6) 其它专门要求

系统的管理系统与维护系统设置了用户管理权限，采用帐号权限区分管理，系统用户密码使用MD5方式加密，对管理员采取有效的管理，最大限度降低内部管理、操作失误带来的不必要的损失。

## 运行环境

(1) 设备

数据库服务器:

Intel core i3

内存4G以上  
硬盘500GB

100M 网卡

应用服务器:

Intel core i3

内存4G以上  
硬盘500GB

100M 网卡

交换机：

100M交换机

无线路由器：

802.11BGN无线路由

客户端PC：

奔腾双核

内存2G以上

硬盘500G

100M网卡

移动客户端：

Android4.4+ PAD

IPAD2+

(2) 支持软件

服务器操作平台：Windows 10

应用服务器：Tomcat6

客户端：IE 8.0及以上

网络环境：Intranet

支持协议：TCP/IP、SMTP、POP3、HTTP、SSL

数据库： MySQL5

支撑环境：JDK1.6及以上

开发工具：Eclipse

设计工具: Visual Paradigm

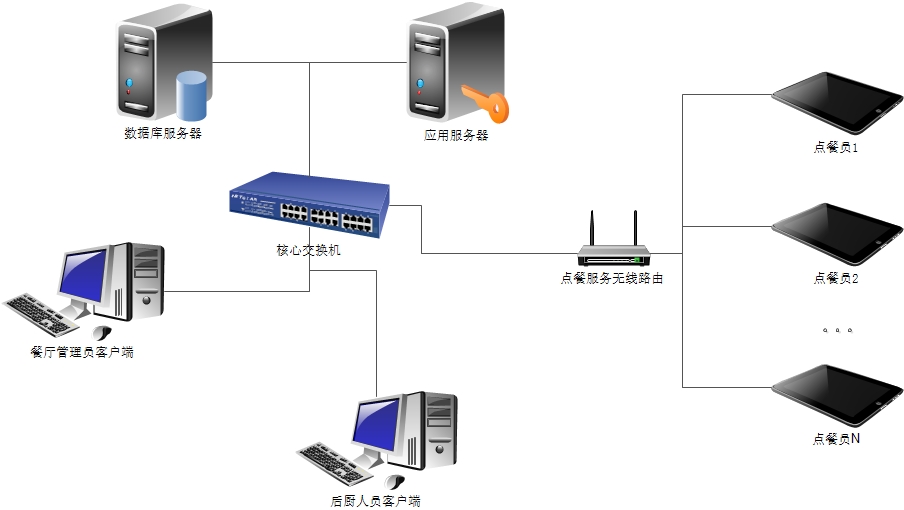
(3) 接口

提供明确的业务规范和业务流程，给第三方系统提供开放的接口，以便提供灵活的集成机会，最终将该系统建设成为一个餐厅综合业务平台，本软件所有的业务过程都有严格数据接口，具体数据传输采用HTTP协议，数据接口采用标准的XML格式集成异构系统。

(4) 控制

本系统采用B/S体系架构，服务器采用Tomcat6，运行只需在服务器端起动Tomcat服务，客户端运行主流浏览器，访问服务器端地址和端口。即可运行。所有的控制利用JAVA的多线程技术，实现对共享资源的一致性维护。

### 运行环境网络结构说明



### 运行环境说明细则

使用表格对运行环境的细节进行说明。

示例：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 设备名称 | 设备型号 | 数量 | 安放位置 | 支持软件 | 说明 |
| 1 | 数据库服务器 | PC | 1台 | 内网 | MySQL5 | 数据存储 |
| 2 | 应用服务器 | PC | 1台 | 内网 | Tomcat | 提供应用业务服务 |
| 3 | 客户端 | PC | 2台 | 内网 | IE10.0、FireFox、Chrome | 餐厅管理员及后厨人员业务 |
| 4 | 移动客户端 | IPad/Android Pad | 按需 | 内网 | 浏览器 | 点餐服务 |

### 支持软件

## 基本设计概念和处理流程

### 整体流程

1.用户在登录界面，输入登录名，密码后，提交。

2.用户有效性检查：用户密码有效性检查，用户状态检查。

3. 根据用户端提供的客户名称、口令，取出对应的加密后的口令，检查用户口令的正确性。

4. 若用户登录不合法，则返回用户登录不成功界面。

5. 若用户登录合法，进入用户登录后界面生成模块 。

6. 进入用户登录后界面视图，组织并生成用户登录后页面。

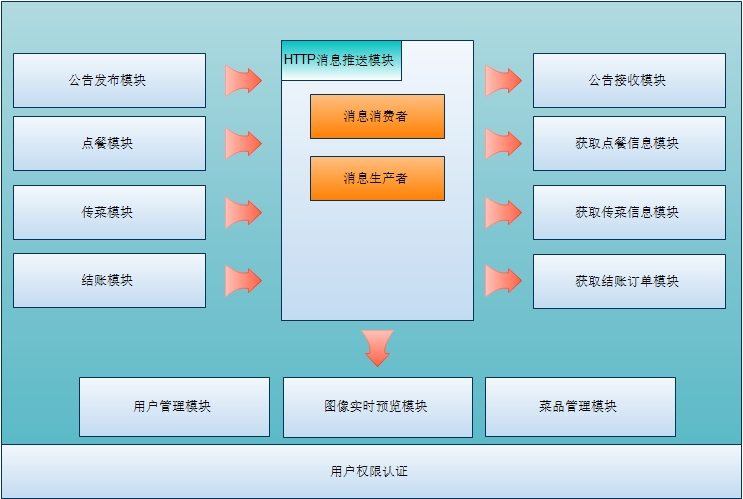
7. 根据客户的身份分别进入不同处理流程。

8. 用户点击导航栏的菜单条目，则再进入下级业务界面。

9. 客户点击通知信息条目，则在弹出对话框显示通知信息内容 。

10. 用户点击指向业务处理的菜单条目，则立即进入相应的业务处理流程。

模块结构：



### 设计概念

#### 消息推送模块

##### 设计模型

模型描述：

1. 输入数据：推送消息字符串
2. 内部处理逻辑：
   * 1. 用户请求获取实时推送消息—>发送HTTP请求—>创建空白消息对象—>使用空白消息对象作为当前请求线程锁阻塞线程—>将空白消息对象加入消息等待队列。
     2. 需要发送推送消息的用户填写推送消息（餐厅管理员填写发送的实时公告、点餐员确定点餐后系统生成实时点餐信息列表、点餐员确定结账后系统生成实时买单订单信息、后厨管理员确定菜品烹制完成后系统生成实时传菜信息）—>发送HTTP请求将消息传递给服务器—>从消息等待队列中检索目标消费者—>将消息保存到目标消费者的消息对象中—>唤醒消息消费者线程—>将消息返回给消息消费者请求用户。
3. 输出数据： 获取到的推送消息字符串

##### 处理流程

本模块不对数据进行任何处理，消息生产者发送的消息内容将原样传递给消息消费者。

##### 数据分析

模块需要处理的数据：实时消息，本模块中传递的消息是一个标准的unicode字符串。

#### 用户验证模块

##### 设计模型

模型描述：

1. 输入数据： 用户登录名

用户登录口令

验证码字符串

1. 内部处理逻辑：
2. 用户输入用户名、口令、验证码—>匹配验证码是否正确—>验证用户名密码是否正确—>跳转界面
3. 输出数据： 登录验证错误提示信息

不同角色用户主界面

##### 处理流程

用户密码口令不能使用明文存储，因此在验证用户口令之前，需要先获取口令的MD5码，用于和数据库中的原始数据进行匹配。

##### 数据分析

模块需要处理的数据：用户名是一个长度不超过20个字符的字符串，密码是一个长度不超过18个字符的字符串，验证码是一个长度为6位的字符串，进行验证码匹配时，不区分字符大小写。

#### 图像实时预览模块

##### 设计模型

模型描述：

1. 输入数据：用户头像或菜品图片
2. 内部处理逻辑：
3. 用户选择图片—>上传图片到指定文件夹—>图片上传成功后向用户推送图片上传成功消息—>自动在图片区域预览上传成功的图片
4. 输出数据： 实时预览图片

##### 处理流程

需要将用户头像图片和菜品图片上传并正确拷贝到规定的文件夹中。

##### 数据分析

模块需要处理的数据：用户头像、菜品图片：图像文件（jpg、bmp、png）。

#### 用户管理模块

##### 设计模型

模型描述：

1. 输入数据： 用户详细信息（用户登录名、用户密码、用户角色信息、用户头像）。

用户ID。

列表页码数

1. 内部处理逻辑：
2. 餐厅管理员输入新的用户登录名、用户密码、用户角色信息—>判定两次输入的用户口令密码是否匹配—>将新的用户信息保存到数据库
3. 管理员提供用户ID—>从数据库中删除该ID对应的用户—>给出输出用户成功提示。
4. 管理员提供用户ID—>从数据库中查询该ID对应的用户信息—>显示在管理界面中—>管理员修改用户信息—>将更新后的用户信息保存到数据库中
5. 管理员提供要显示的用户列表页码数—>从数据库中查询对应页码的数据—>生成XML数据—>解析XML并展现在界面上
6. 输出数据： 用户列表

用户详细信息

##### 处理流程

用户密码口令不能使用明文存储，因此在新增或更新用户时需要保存用户密码的MD5值。

##### 数据分析

模块需要处理的数据：用户列表及用户详细信息功能需要返回对应的XML文档数据。

#### 菜品管理模块

##### 设计模型

模型描述：

1. 输入数据： 菜品详细信息（菜品名称、菜品简介、菜品详细信息、是否推荐餐品、菜品单价、菜品图片）。

菜品ID。

列表页码数

1. 内部处理逻辑：
2. 餐厅管理员输入新的菜品信息—>将新的菜品信息保存到数据库—>给出菜品删除成功提示
3. 管理员提供菜品ID—>从数据库中删除该ID对应的菜品—>给出添加菜品成功提示。
4. 管理员提供菜品ID—>从数据库中查询该ID对应的菜品信息—>显示在管理界面中—>管理员修改菜品信息—>将更新后的菜品信息保存到数据库中
5. 管理员提供要显示的菜品列表页码数—>从数据库中查询对应页码的数据—>生成XML数据—>解析XML并展现在界面上
6. 输出数据： 菜品列表

菜品详细信息

##### 处理流程

由于需要在页面中显示菜品详细信息中可能本身包含控制字符如双引号等，因此详细信息字符串需要预处理，将双引号等字符替换成自定义的占位符。

##### 数据分析

模块需要处理的数据：菜品列表及菜品详细信息功能需要返回对应的XML文档数据。

#### 公告模块

##### 设计模型

模型描述：

1. 输入数据： 实时公告字符串。
2. 内部处理逻辑：
3. 餐厅管理员输入实时公告—>获取公告字符串—>向服务员及后厨人员推送公告
4. 输出数据： 实时公告字符串

##### 处理流程

数据无特殊处理。

##### 数据分析

模块需要处理的数据：实时公告字符串：标准的unicode字符串。

#### 通知买单模块

##### 设计模型

模型描述：

1. 输入数据： 待买单订单号。
2. 内部处理逻辑：
3. 点餐服务员提供待买单订单号—>从数据库中查询该订单详细信息—>将该订单详细信息格式化为XML数据—>将通知买单信息推送到餐厅管理员界面
4. 输出数据：结餐并准备买单的订单详情

##### 处理流程

数据无特殊处理。

##### 数据分析

模块需要处理的数据：订单数据，包括订单的详细列表及点餐时间、桌号

、点餐服务员信息。

#### 点餐模块

##### 设计模型

模型描述：

1. 输入数据： 点餐数据列表（菜品ID，对应的菜品数量）。
2. 内部处理逻辑：
3. 点餐服务员输入点餐数据—>将点餐数据存入数据库—>将点餐数据实时推送到后厨界面
4. 输出数据：实时点餐数据

##### 处理流程

数据无特殊处理。

##### 数据分析

模块需要处理的数据：点餐数据，包括点餐的桌号，点餐服务员信息，点餐时间，食品列表（菜品ID及对应的数量）。

#### 传菜模块

##### 设计模型

模型描述：

1. 输入数据： 传菜信息（桌号、菜品名）。
2. 内部处理逻辑：
3. 后厨确定菜品烹制完毕—>将传菜数据实时推送到后厨界面
4. 输出数据：实时传菜数据

##### 处理流程

数据无特殊处理。

##### 数据分析

模块需要处理的数据：传菜数据，烹制完成的菜品名称和目标桌号。

#### 结算模块

##### 设计模型

模型描述：

1. 输入数据： 订单号。
2. 内部处理逻辑：
3. 餐厅管理员提供订单号—>查询该订单的详细信息—>将该订单详细信息格式化为XML数据—>显示订单详细信息
4. 餐厅管理员提供订单号—>将数据库中的订单状态改变为免单—>提示操作成功
5. 餐厅管理员提供订单号—>将数据库中的订单状态改变为订单结算成功—>提示操作成功
6. 输出数据：订单详细信息

##### 处理流程

数据无特殊处理。

##### 数据分析

模块需要处理的数据：订单详细信息，包括订单开餐时间、结单时间、点单服务员、桌号、订单菜品详细列表。

#### 历史数据查询模块

##### 设计模型

模型描述：

1. 输入数据： 结餐时间
2. 内部处理逻辑：
3. 餐厅管理员提供开始时间和结束时间作为查询条件—>查询结单时间在提供的开始时间和结束时间之间的订单数据—>将订单数据格式化为XML文档—>显示查询结果
4. 输出数据：历史订单列表

##### 处理流程

数据无特殊处理。

##### 数据分析

模块需要处理的数据：订单详细信息，包括订单开餐时间、结单时间、点单服务员、桌号、订单菜品详细列表。

## 结构设计

### 功能模块划分

#### 实时消息推送功能

##### 功能概述

在现行的HTTP协议版本下，实现基于WEB的实时数据交换，希望推送消息的业务组件可以简单、方便的通过推送消息生产者向特定的目标发送推送消息，希望获取实时消息的业务组件可以通过实时消息消费者直接接受到该推送消息。

##### 使用角色和权限划分

所有登录的用户均可以使用该功能模块中的消息生产者与消息消费者。

##### 数据库设计

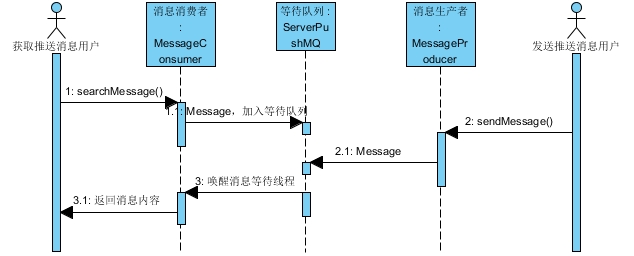
该功能不对应任何数据库表。

##### 基本设计概念和流程

###### 基本设计概念

实时推送消息功能实现利用了现行HTTP协议版本特点、请求处理线程特性及Java中对多线程的调度操作完成。任然符合当前HTTP协议中主动请求-被动响应的通讯方式，本质上是利用了HTTP长连接请求获取实时数据，当服务器捕获消息请求时并不直接返回响应而是等待获取到消息数据时再将消息返回给客户端，客户端获取消息并处理完成后发起再次请求。

###### 时序图



##### 数据流向

数据传递方式：HTTP

数据传递格式：普通字符串

数据传递流向：消息生产者🡪消息消费者

#### 判定非法请求功能

##### 功能描述

当用户从非本网站内部发送登录请求时，系统能够做出非法请求判定并对用户做出提示。

##### 使用角色和权限

所有用户均可使用本模块

##### 数据库设计

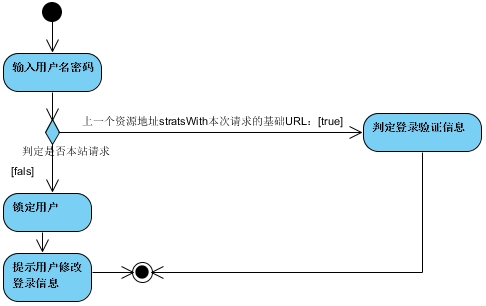
该功能不对应任何数据库表。

##### 基本设计概念和流程

###### 基本设计概念

外站非法请求识别的基本实现方式是判定请求数据来源URL，根据HTTP协议的协定，浏览器请求资源时，如果请求由另一个资源内部发起，那么请求头将会包含名为Referer的头属性，属性值为发起请求的上一个资源路径（包括详细的QueryString参数信息），而如果本资源请求是由用户直接在地址栏输入URL并确认请求，那么请求头中不包含该参数信息，在Java语言中尝试获取该属性将获取到null，根据这个约束可以利用上一个资源的URL和本次请求的资源路径进行比对，用以判定是否是同一网站发送的请求。

###### 活动图



##### 流程描述

1. 用户输入URL打开用户登录界面。
2. 用户输入用户名、密码信息并点击登录
3. 数据提交到服务器后，服务器在执行登录操作之前先判定提交用户登录信息的界面是否是钓鱼网站：
4. 获取当前登录请求资源的基础URL信息（精确到项目的ContextRoot）
5. 通过请求头的Referer属性获取前一资源的URL信息
6. 如果前一资源的URL信息startsWith当前登录请求资源的基础URL信息，则说明请求合法，否则说明用户的登录请求来源于非法的钓鱼网站。
7. 如果请求合法，则进行用户登录验证判定
8. 如果请求不合法，则锁定对应用户，并向用户提出警告，要求用户修改变更登录信息。

#### 菜品列表功能

##### 功能描述

餐厅管理员能够以分页的形式获取本餐厅提供的所有菜品列表，列表中优先显示确定的推荐菜品。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 菜品信息表(dishesinfo )

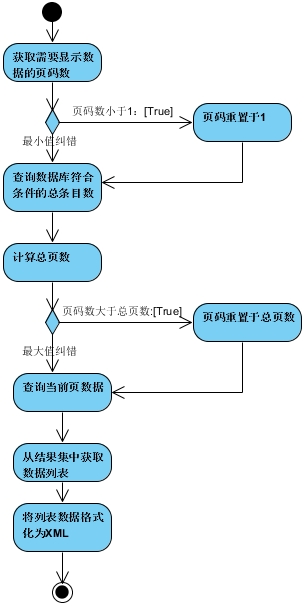
记录本餐厅所有的菜品信息，包括菜品的编号，菜品名称、菜品简介、菜品详细信息、菜品单价、是否推荐菜品、菜品图片路径信息。

##### 基本设计概念和流程

###### 基本设计概念

直接利用常见数据库的分页数据查询语法完成当前页面数据的查询操作，利用数据库函数查询当前数据记录条目数，并计算总页数，在进行分页控制时利用数值比较完成简单的纠错功能（页码不能小于1，也不能大于总页数），当页码值小于1时，显示第一页的内容，当页码值大于总页数时显示最后一页的数据，查询到的数据需要格式化为XML文档。

###### 活动图



##### 流程描述

1. 用户通过URL QueryString参数提供需要展现的菜品列表页码。
2. 系统进行参数最小值纠错，如果页码参数值小于1，则直接重置为1.
3. 系统查询符合显示条件的数据库总条目数
4. 利用查询到的总条目数信息结合每页显示的条目数计算当前数据库中数据分页后的总页数
5. 系统进行参数值最大纠错，如果页码参数值大于总页数，则直接重置为总页数。
6. 查询当前页数据
7. 将查询到的数据格式化为XML文档

#### 用户列表功能

##### 功能描述

餐厅管理员能够以分页的形式获取本系统所有用户列表。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 用户信息表(userinfo )

记录本餐厅所有用户信息，包括用户编号，用户登录帐号，用户加密后的密码，用户角色id，用户头像信息。

1. 角色信息表

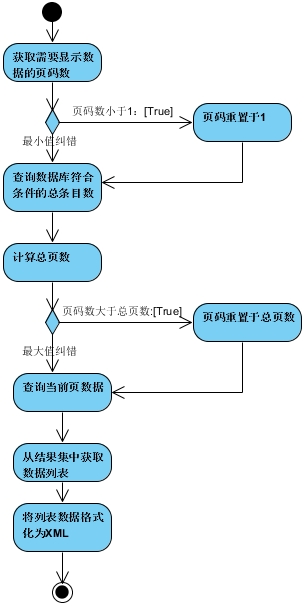
记录系统中存在的不同角色信息，包括角色编号，角色名称信息

##### 基本设计概念和流程

###### 基本设计概念

直接利用常见数据库的分页数据查询语法完成当前页面数据的查询操作，利用数据库函数查询当前数据记录条目数，并计算总页数，在进行分页控制时利用数值比较完成简单的纠错功能（页码不能小于1，也不能大于总页数），当页码值小于1时，显示第一页的内容，当页码值大于总页数时显示最后一页的数据，查询到的数据需要格式化为XML文档。

###### 活动图



##### 流程描述

1. 用户通过URL QueryString参数提供需要展现的菜品列表页码。
2. 系统进行参数最小值纠错，如果页码参数值小于1，则直接重置为1.
3. 系统查询符合显示条件的数据库总条目数
4. 利用查询到的总条目数信息结合每页显示的条目数计算当前数据库中数据分页后的总页数
5. 系统进行参数值最大纠错，如果页码参数值大于总页数，则直接重置为总页数。
6. 查询当前页数据
7. 将查询到的数据格式化为XML文档

#### 添加用户功能

##### 功能描述

管理员可以添加新的系统用户。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 用户信息表(userinfo )

记录本餐厅所有用户信息，包括用户编号，用户登录帐号，用户加密后的密码，用户角色id，用户头像信息。

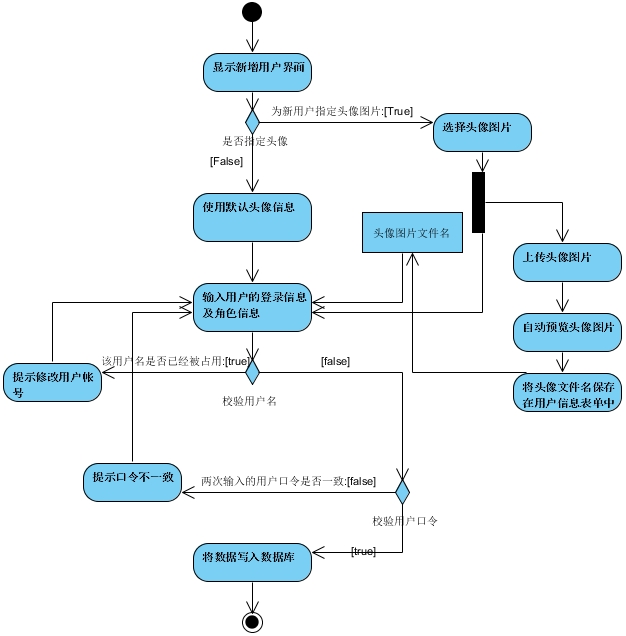
##### 基本设计概念和流程

###### 基本设计概念

管理员可以为新用户提供头像图片，如果不提供指定的头像，系统会提供默认的头像图片。

添加用户时，需要查验当前用户是否被占用，需要检验两次输入的用户口令密码是否匹配。

###### 活动图



##### 流程描述

1. 管理员选择是否要为新用户上传指定的头像图片。

若管理员选择上传头像，确定后不刷新页面，头像上传成功后在头像区域自动预览，并将文件名传递到用户信息表单对应表单域。

若管理员不上传头像，则系统提供默认的头像信息。

1. 管理员可以重复进行头像信息上传操作。
2. 管理员填写用户的登录信息，包括用户的登录名、密码及身份信息，填写时系统自动验证用户名是否可用，两次输入的密码信息是否一致，当用户名可用、两次密码一致时确定按钮才可用，填写完毕点击确定按钮后将用户的所有信息存放到数据库中。

#### 用户详细信息功能

##### 功能描述

管理员可以查看特定用户的详细信息。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 用户信息表(userinfo )

记录本餐厅所有用户信息，包括用户编号，用户登录帐号，用户加密后的密码，用户角色id，用户头像信息。

1. 角色信息表

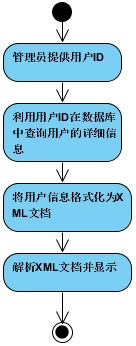
记录系统中存在的不同角色信息，包括角色编号，角色名称信息

##### 基本设计概念和流程

###### 基本设计概念

管理员可以通过给定用户ID查看该用户的详细信息。

###### 活动图



##### 流程描述

1. 餐厅管理员提供要查看详情的用户ID。
2. 系统从数据库中查询该用户的详细信息。
3. 将用户信息格式化为XML文档。
4. 解析XML文档并在界面中显示结果。

#### 更新用户信息功能

##### 功能描述

管理员可以更新特定ID用户的信息。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 用户信息表(userinfo )

记录本餐厅所有用户信息，包括用户编号，用户登录帐号，用户加密后的密码，用户角色id，用户头像信息。

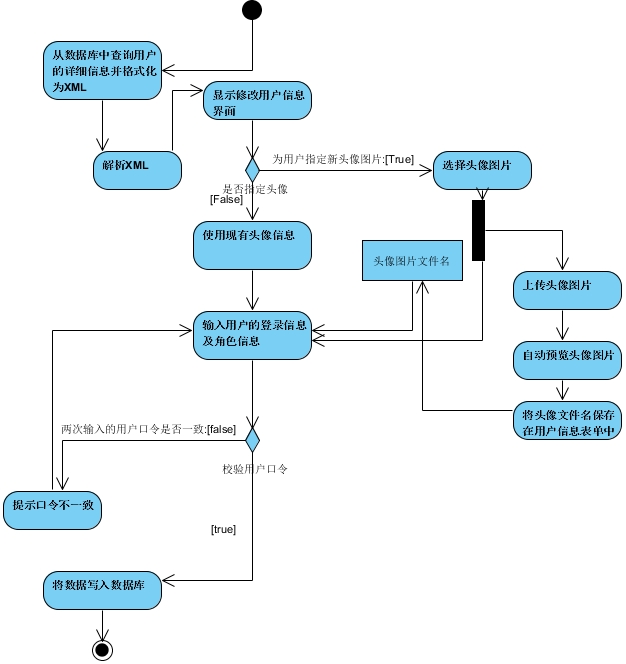
##### 基本设计概念和流程

###### 基本设计概念

和新增用户功能一样，管理员可以为新用户提供头像图片，如果不提供指定的头像，系统会提供默认的头像图片。

用户的登录帐号信息不能修改。

###### 流程图



##### 流程描述

1. 餐厅管理员提供需要修改的用户ID
2. 系统查询该用户的详细信息并格式化为XML文档
3. 系统显示修改用户信息界面并解析用户详细信息XML数据
4. 管理员选择是否要为用户上传新的头像图片。

若管理员选择上传头像，确定后不刷新页面，头像上传成功后在头像区域自动预览，并将文件名传递到用户信息表单对应表单域。

若管理员不上传头像，则系统默认使用用户现有的头像信息。

1. 管理员可以重复进行头像信息上传操作。
2. 管理员填写用户新的登录信息，包括用户的密码及身份信息，填写完毕点击确定按钮后将用户的所有信息存放到数据库中。

#### 删除用户功能

##### 功能描述

管理员可以删除特定ID的用户。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 用户信息表(userinfo )

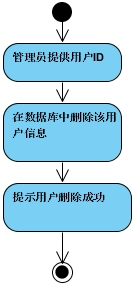
记录本餐厅所有用户信息，包括用户编号，用户登录帐号，用户加密后的密码，用户角色id，用户头像信息。

##### 基本设计概念和流程

###### 基本设计概念

管理员可以通过给定用户ID删除该用户。

###### 活动图



##### 流程描述

1. 餐厅管理员提供要查看删除的用户ID。
2. 系统从数据库中删除该用户的信息。
3. 系统提示用户删除操作成功信息。

#### 添加菜品功能

##### 功能描述

管理员可以添加新的菜品。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 菜品信息表(dishesinfo )

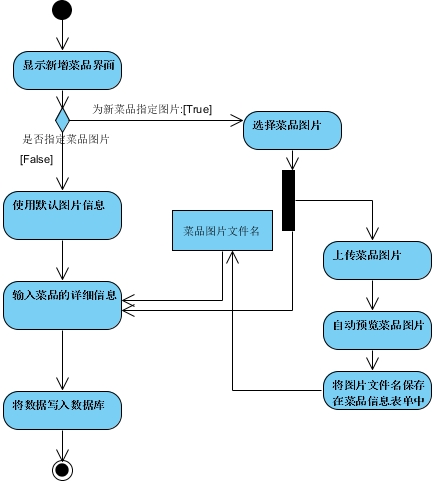
记录本餐厅所有的菜品信息，包括菜品的编号，菜品名称、菜品简介、菜品详细信息、菜品单价、是否推荐菜品、菜品图片路径信息。

##### 基本设计概念和流程

###### 基本设计概念

管理员可以新增一个菜品，菜品需要保存的信息包括：菜品编号（由系统自行生成），菜品名称、菜品简介、菜品详细信息、菜品的单价（可使用两位以内的小数点，单位：元），该菜品是否为推荐菜品，菜品的图片路劲信息。

###### 流程图



##### 流程描述

1. 新增菜品界面分为两个区域，一个区域设置菜品的图片信息，另一个区域设置菜品的文本相关属性信息。
2. 餐厅管理员可以选择是否为菜品上传特有的图片描述。

若管理员选择上传图片，确定后不刷新页面，头像上传成功后在图片区域自动预览，并将文件名传递到菜品信息表单对应表单域。

若管理员不上传图片，则系统提供默认的菜品图片信息。

1. 管理员可以重复进行图片信息上传操作。
2. 管理员填写菜品的文本描述信息，包括菜品的名称、简介、详细介绍、是否推荐菜品、菜品价格，填写完毕点击确定按钮后将用户的所有信息存放到数据库中。

#### 更新菜品信息功能

##### 功能描述

管理员可以更新特定ID菜品的信息。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 菜品信息表(dishesinfo )

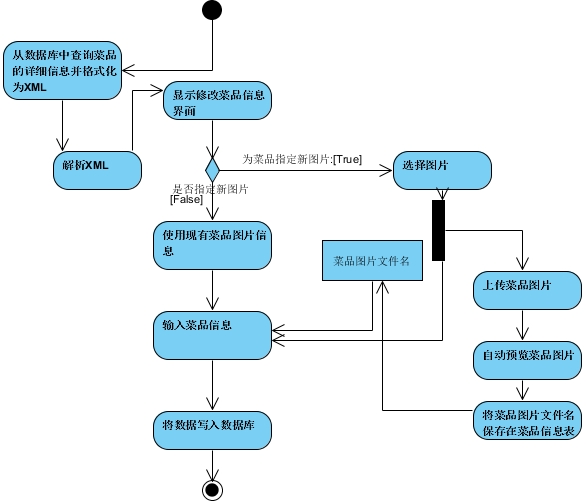
记录本餐厅所有的菜品信息，包括菜品的编号，菜品名称、菜品简介、菜品详细信息、菜品单价、是否推荐菜品、菜品图片路径信息。

##### 基本设计概念和流程

###### 基本设计概念

和新增菜品功能一样，管理员可以为菜品提供图片，如果不提供指定的图片，系统会提继续使用现有的菜品图片。

###### 流程图



##### 流程描述

1. 餐厅管理员提供需要修改的菜品ID
2. 系统查询该菜品的详细信息并格式化为XML文档
3. 系统显示修改菜品信息界面并解析菜品详细信息XML数据
4. 管理员选择是否要为菜品上传新的描述图片。

若管理员选择上传图片，确定后不刷新页面，图片上传成功后在图片区域自动预览，并将文件名传递到菜品信息表单对应表单域。

若管理员不上传图片，则系统默认使用菜品现有的图片信息。

1. 管理员可以重复进行菜品图片信息上传操作。
2. 管理员填写菜品新的描述信息，包括菜品名、简介、详细介绍、是否推荐、菜品单价信息，填写完毕点击确定按钮后将菜品的所有信息存放到数据库中。

#### 删除菜品功能

##### 功能描述

管理员可以删除特定ID的菜品。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 菜品信息表(dishesinfo )

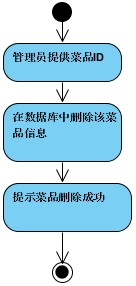
记录本餐厅所有的菜品信息，包括菜品的编号，菜品名称、菜品简介、菜品详细信息、菜品单价、是否推荐菜品、菜品图片路径信息。

##### 基本设计概念和流程

###### 基本设计概念

管理员可以通过给定菜品ID删除该菜品。

###### 活动图



##### 流程描述

1. 餐厅管理员提供要查看删除的菜品ID。
2. 系统从数据库中删除该菜品的信息。
3. 系统提示菜品删除操作成功信息。

#### 发布公告功能

##### 功能描述

餐厅管理员可以发布实时公告，公告一经发布就会即时显示在服务员和后厨人员界面的公告区域内。

##### 使用角色和权限

发布公告：餐厅管理员

接受公告信息：餐厅点餐服务员、餐厅后厨人员

##### 数据库设计

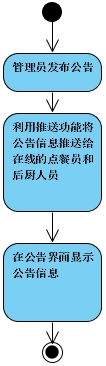
该功能不对应任何数据库表。

##### 基本设计概念和流程

###### 基本设计概念

利用消息推送组件可以直接实现公告消息的推送。

###### 活动图



##### 数据流向

数据传递方式：HTTP

数据传递格式：XML文件

数据传递流向：餐厅管理员🡪餐厅后厨人员、餐厅点餐服务员

##### 流程描述

1. 餐厅管理员在发布公告界面填写公告信息并点击确定。
2. 系统构建推送消息生产者，将公告信息推送到符合条件的所有消息消费者（餐厅服务员和后厨人员创建的公告消息消费者）
3. 消息消费者将消息传递给客户端界面

#### 点餐功能

##### 功能描述

餐厅点餐服务员对顾客提供点餐服务。

##### 使用角色和权限

点餐：餐厅服务员

显示点餐菜品及数量：后厨人员

##### 数据库设计

1. 订单信息表(orderinfo )

记录订单信息，包括订单的编号，订单点单时间，订单结单时间，订单桌号，点餐服务员编号，订单的结算状态信息。

1. 订单菜品映射表（orderdishes）

记录订单的详细菜品列表，包括映射编号，订单编号，菜品编号，菜品数量信息

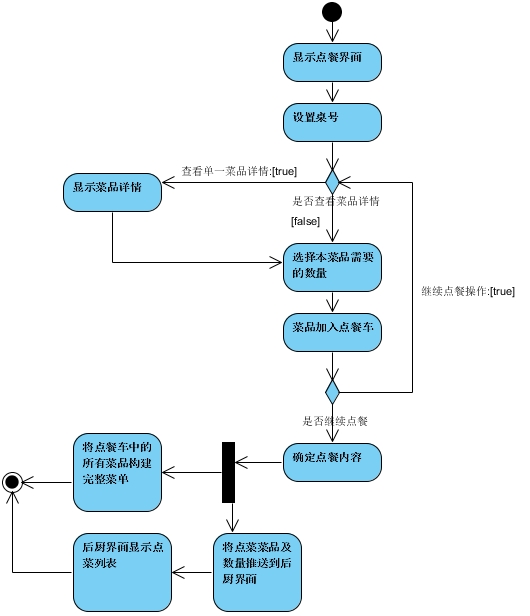
##### 基本设计概念和流程

###### 基本设计概念

服务员提供点单服务之前需要先设置当前的桌号，订单中的菜品可以查看详细信息，订单中的菜品可以选择不同的数量。

订单确定之后需要实施将菜品及数量发送给后厨用于备菜。

###### 活动图



##### 数据流向

数据传递方式：HTTP

数据传递格式：XML文件

数据传递流向：点餐服务延🡪餐厅后厨人员

##### 流程描述

1. 系统展现点餐界面，在点餐界面中服务员需要预先确定当前点餐的餐台桌号。
2. 服务员可以利用点餐系统为顾客韩式顾客需要了解的特定菜品详细信息。
3. 服务员可以利用界面中提供的+/-按钮选择某个菜品下单的数量
4. 完成点餐后点击确定订单按钮。

点击按钮后执行两个并行的操作：

* 1. 利用选择的菜品和数量构建订单信息并存入数据库。
  2. 将点菜的结果（菜品、数量、桌号）推送带后厨以便于后厨人员备菜。

#### 备菜功能

##### 功能描述

后厨人员接收到点菜列表后根据厨师的操作在系统中改变菜品烹制状态，完成烹制后发送传菜信息告知服务员传菜。

##### 使用角色和权限

备菜：后厨人员

接受传菜信息：餐厅点餐服务员

##### 数据库设计

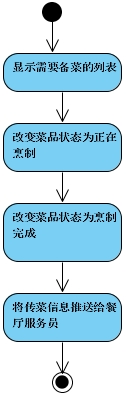
该功能不对应任何数据库表。

##### 基本设计概念和流程

###### 基本设计概念

利用消息推送组件可以直接实现传菜消息的推送。

###### 活动图



##### 流程描述

1. 获取点餐服务员的菜品订单列表后显示在备菜界面中。
2. 开始烹饪时点击菜品状态按钮，将菜品状态变为：正在烹饪
3. 菜品烹制结束后点击菜品状态按钮，将菜品状态变为：烹制完成（菜品从列表中删除，将本菜品对应的传菜信息推送到餐厅点餐服务员界面）

#### 准备结账功能

##### 功能描述

餐厅服务员利用系统消息推送告知餐厅管理员有餐台用餐结束准备结账。

##### 使用角色和权限

准备结账：餐厅服务员

显示准备结账订单：餐厅管理员

##### 数据库设计

1. 订单信息表(orderinfo )

记录订单信息，包括订单的编号，订单点单时间，订单结单时间，订单桌号，点餐服务员编号，订单的结算状态信息。

1. 订单菜品映射表（orderdishes）

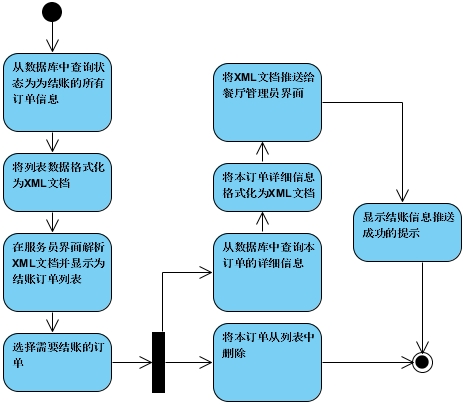
记录订单的详细菜品列表，包括映射编号，订单编号，菜品编号，菜品数量信息

##### 基本设计概念和流程

###### 基本设计概念

服务员从为结账订单列表中选择即将买单的订单号后在后台查询订单详细数据信息后生成XML文档推送给餐厅管理员用于结算确认。

###### 活动图



##### 数据流向

数据传递方式：HTTP

数据传递格式：XML文件

数据传递流向：餐厅服务员🡪餐厅管理员

##### 流程描述

1. 系统查询数据库中所有为结账订单信息并生产列表对应的XML文档并显示在服务员界面中。
2. 服务员选择需要结账的订单。
3. 服务员确定结账订单后执行：
4. 将订单信息在列表中删除
5. 根据服务员选择的订单号查询该订单对应的详细信息，包括订单编号、订单桌号、订单开餐时间、订单点餐服务员、订单详细菜品及数量列表、订单总价信息并格式化为XML文档。
6. 将XML文档推送到餐厅管理员界面，在餐厅管理员界面上的待结算订单列表中追加一条

#### 订单结算功能

##### 功能描述

餐厅管理员完成订单的最终收款操作。

##### 使用角色和权限

餐厅管理员

##### 数据库设计

1. 订单信息表(orderinfo )

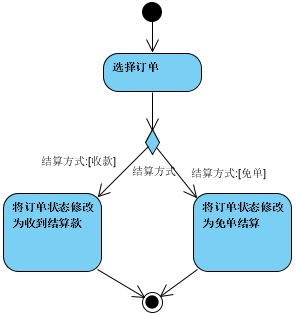
记录订单信息，包括订单的编号，订单点单时间，订单结单时间，订单桌号，点餐服务员编号，订单的结算状态信息。

##### 基本设计概念和流程

###### 基本设计概念

餐厅管理员能够进行的两种结算操作：收款和免单。

###### 活动图



##### 流程描述

1. 餐厅管理员界面显示待结算订单后显示收款和免单按钮。
2. 管理员点击收款按钮：
   1. 将订单从列表中删除
   2. 将订单状态修改为确认收款
3. 管理员点击免单按钮
   1. 将订单从列表中删除
   2. 将订单状态修改为直接免单
4. 显示操作成功提示

## 人工处理过程

本系统中的第一个管理员账户需要人工添加到数据中，该管理员帐户可以创建其他所有用户的登录信息。

# 接口设计

## 数据请求与返回接口

以分页形式获取用户列表信息接口：

请求URL：http://域名:端口号/OrderSys/getuserbypage.order?page=页码数

数据返回格式：

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!-- 根据页码获取用户信息返回XML的根节点 -->

<users>

<!-- 单个用户信息 -->

<user>

<!-- 用户编号 -->

<userId>1</userId>

<!-- 用户登录名 -->

<userAccount>admin</userAccount>

<!-- 用户角色ID -->

<roleId>1</roleId>

<!-- 用户角色名称呢个 -->

<roleName>餐厅管理员</roleName>

<!-- 用户是否被锁定 -->

<locked>0</locked>

<!-- 用户头像图片名称呢个 -->

<faceimg>default.jpg</faceimg>

</user>

…

<user>…</user>

<!-- 当前的页码数 -->

<page>1</page>

<!-- 最大页码数 -->

<maxPage>1</maxPage>

</users>

以分页形式获取菜品信息列表接口：

请求URL：http://域名:端口号/OrderSys/getdishesbypage.order?page=页码数

数据返回格式：

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<!-- 根据页码获取菜品信息返回XML的根节点 -->

<disheses>

<!-- 单个菜品节点 -->

<dishes>

<!-- 菜品编号 -->

<dishesId>1</dishesId>

<!-- 菜品名称呢个 -->

<dishesName>川香回锅肉</dishesName>

<!-- 菜品简介 -->

<dishesDiscript>菜色鲜香，肥而不腻</dishesDiscript>

<!-- 菜品图片名称呢个 -->

<dishesImg>1.jpg</dishesImg>

<!-- 菜品详细介绍 -->

<dishesTxt>采用上等五花肉制作，配以酸姜、\"酸辣椒\"及\'秘制\'豆豉酱，味道鲜美，营养丰富</dishesTxt>

<!-- 是否推荐菜品：1-推荐，0-非推荐 -->

<recommend>1</recommend>

<!-- 菜品单价 -->

<dishesPrice>26.5</dishesPrice>

</dishes>

…

<dishes >…</dishes >

<!-- 当前页码 -->

<page>1</page>

<!-- 总页数 -->

<maxPage>1</maxPage>

</disheses>

# 系统出错处理

## 出错信息

所有出错信息均以字符串的方式，在系统运行界面中显示。所有出错信息分为三种：

一种是由于输入错误信息超出或不符合预定各式的错误，属于处理错误

二是由于系统的预设不能执行的错误，属于设定错误

三是由于网路传输超时、服务器响应超时等属于系统错误

对于处理错误需在操作成功判断及输入数据验证模块由数据进行数据分析，判断错误类型，再生成相应的错误提示语句，送到输出模块中，对于设定错误，应在开始提交信息类别中，依据权限等判定错误类别，再生成相应出错信息语句，输出到输出模块中。对于系统错误，根据Tomcat服务器的响应内容，判断错误类别输出。 出错信息必须给出相应的出错原因，如：

“用户名不存在”

“您的登录数据请求来源与钓鱼网站”

“用户身份信息由管理员给定，不能修改”

“两次输入的密码不一致”

## 补救措施

所有的客户机及服务器都必须安装不间断电源以防止停电或电压不稳造成的数据丢失的损失。若真断电时，客户机上将不会有太大的影响，主要是服务器上：在断电后恢复过程可采用数据库的日志文件，对其进行ROLLBACK 处理，对数据进行恢复。在网络传输方面，可考虑建立一条成本较低的后备网络，以保证当主网络断路时数据的通信。在硬件方面要选择较可靠、稳定的服务器机种，保证系统运行时的可靠性。

## 系统维护设计

维护方面主要为对服务器上的数据库数据进行维护。可使用 数据库管理系统的数据库的维护功能机制。例如，定期为数据库进行Backup，维护管理数据库死锁问题和维护数据库内数据的一致性等。