**餐厅到店点餐系统详细**

**（版本v1.0.0）**

中软国际信息技术有限公司

2015年10月21日



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 成文信息 | | | | | | | |
| 主题词： | | 详细设计 | | | | | |
| 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](#_Toc433639044)

[1.1 编写目的 1](#_Toc433639045)

[1.2 背景 1](#_Toc433639046)

[1.3 定义 1](#_Toc433639047)

[1.4 列出有关的参考资料 1](#_Toc433639048)

[第二章 程序系统结构 2](#_Toc433639049)

[2.1 体系架构 2](#_Toc433639050)

[2.2 程序结构设计 3](#_Toc433639051)

[2.2.1 名称设计 3](#_Toc433639052)

[2.2.2 目录规范 3](#_Toc433639053)

[2.2.3 配置文件规划 5](#_Toc433639054)

[第三章 HTTP消息推送模块设计说明 6](#_Toc433639055)

[3.1.1 模块描述 6](#_Toc433639056)

[3.1.2 功能 6](#_Toc433639057)

[3.1.3 性能 7](#_Toc433639058)

[3.1.4 类设计 7](#_Toc433639059)

[3.1.5 算法 10](#_Toc433639060)

[第四章 用户管理模块设计说明 10](#_Toc433639061)

[4.1.1 模块描述 10](#_Toc433639062)

[4.1.2 功能 11](#_Toc433639063)

[4.1.3 人机交互 11](#_Toc433639064)

[4.1.4 类设计 14](#_Toc433639065)

[4.1.5 算法 15](#_Toc433639066)

[第五章 菜品管理模块设计说明 15](#_Toc433639067)

[5.1.1 模块描述 15](#_Toc433639068)

[5.1.2 功能 16](#_Toc433639069)

[5.1.3 人机交互 16](#_Toc433639070)

[5.1.4 类设计 19](#_Toc433639071)

[5.1.5 算法 20](#_Toc433639072)

[第六章 点餐模块设计说明 20](#_Toc433639073)

[6.1.1 模块描述 20](#_Toc433639074)

[6.1.2 功能 21](#_Toc433639075)

[6.1.3 人机交互 21](#_Toc433639076)

[6.1.4 类设计 24](#_Toc433639077)

[6.1.5 算法 24](#_Toc433639078)

[第七章 传菜模块设计说明 25](#_Toc433639079)

[7.1.1 模块描述 25](#_Toc433639080)

[7.1.2 功能 26](#_Toc433639081)

[7.1.3 人机交互 26](#_Toc433639082)

[7.1.4 类设计 28](#_Toc433639083)

[第八章 结算模块设计说明 28](#_Toc433639084)

[8.1.1 模块描述 28](#_Toc433639085)

[8.1.2 功能 29](#_Toc433639086)

[8.1.3 人机交互 29](#_Toc433639087)

[8.1.4 类设计 31](#_Toc433639088)

[8.1.5 算法 31](#_Toc433639089)

# 引言

## 编写目的

编写本详细设计的目的是：在餐厅到店点餐系统需求细化分析的基础上，完成系统的结构设计，划分系统模块并明确模块之间的关系，同时给出相关的用户界面设计。

本文的预期读者为： 餐厅到店点餐系统的设计、开发人员。 工程管理软件的验收、评审人员。

## 背景

待开发的软件为餐厅到店点餐系统。 此项目由中软国际ETC提出，ETC设计和开发。软件将在目标客户餐厅本地部署、运行，本软件属于餐厅运营管理服务系统，用于提高餐厅服务质量，提高餐厅客户满意度，提升餐厅整体形象。

## 定义

1、系统：若未特别指出，统指餐厅到店点餐系统。

2、平台：若未特别指出，特指本系统集成运行平台。

3、SQL：Structured Query Language（结构化查询语言）。

## 列出有关的参考资料

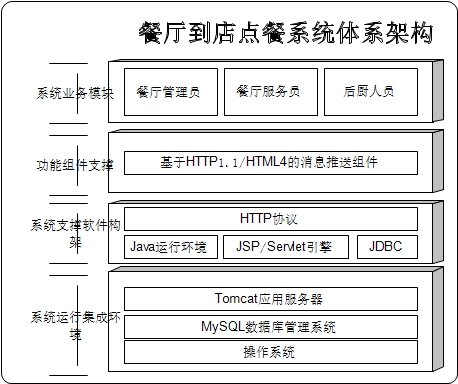
1、《餐厅到店点餐系统需求规格说明书》。

2、《餐厅到店点餐系统概要设计说明书》

# 程序系统结构

## 体系架构

从功能构成的角度出发划分系统的职能层次，按照从底层到顶层，服务到应用的划分方法，餐厅到店点餐系统基本上可以划分为四个部分：应用运行集成环境层、系统支撑软件架构、功能组件支撑和系统业务模块。其基本组成图如下：



1. 应用运行集成环境层

包括本系统部署环境平台操作系统（本系统对操作系统无特殊要求，系统可以无缝迁移到Windows/Linux/Unix等常见平台之上）、系统使用的数据库管理系统（默认使用MySQL，系统内部数据库访问操作采用灵活的数据库方言机制，能够灵活的支持系统数据持久层迁移）以及本项目运行的应用服务器(Tomcat)

1. 系统支撑软件架构

包括操作系统中的TCP/IP协议栈、JRE运行环境，J2EE引擎以及访问数据持久层的JDBC API。

1. 功能组件支撑

本系统需要不同客户端界面之间的实时性响应，需要在基础资源管理层和业务模块之间实现业务间实时通讯的功能组件，该组件必须能够兼容目前最成熟，运用范围最广的HTML4/HTTP1.1标准。

1. 功能组件支撑

包括餐厅到店点餐系统的运营业务功能模块，如菜品管理、员工管理、点餐、备菜传菜、结餐结算等。

## 程序结构设计

### 名称设计

### 目录规范

为了管理餐厅到店点餐系统的部署程序和工作数据，避免因系统各组件目录重复导致的错误，故需要统一规划应用软件相关目录结构。

本系统各组件采用以下目录结构存储应用程序及相关数据。

JSP页面目录：

**OrderSys(ContextRoot) 存放所有的页面数据**

|--**pages** 存放所有不需要权限认定即可访问的页面

|--**admin** 存放餐厅管理员才能访问的页面

|--**users** 存放所有登录用户均能访问的页面

|--**kichen** 存放后厨人员才能访问的页面

|--**waiters** 存放点餐服务员才能访问的页面

Java程序包目录：

com.chinasofti.ordersys.filters ：

存放系统中跟业务权限相关的过滤器

com.chinasofti.ordersys.listeners：

存放系统中跟会话跟踪相关的监听器

com.chinasofti.ordersys.service：

存放外站非法请求判定的服务组件

com.chinasofti.ordersys.service.admin：

存放与管理员业务相关的服务组件

com.chinasofti.ordersys.service.login：

存放与用户登录功能相关的服务组件

com.chinasofti.ordersys.service.waiters：

存放与点餐服务员业务相关的服务组件

com.chinasofti.ordersys.servlets：

存放与外站非法请相关的跳转Servlet

com.chinasofti.ordersys.servlets.admin：

存放与管理员业务相关的Servlet

com.chinasofti.ordersys.servlets.common：

存放与通用功能组建相关的Servlet

com.chinasofti.ordersys.servlets.kitchen：

存放与后厨人员业务相关的Servlet

com.chinasofti.ordersys.servlets.login：

存放与用户登录功能相关的Servlet

com.chinasofti.ordersys.servlets.waiters：

存放与餐厅点餐服务员业务相关的Servlet

com.chinasofti.ordersys.vo：

存放系统中使用的所有实体类

com.chinasofti.util.bean：

存放用于利用反射自动填充JavaBean的功能组件

com.chinasofti.util.bean.convertor：

存放用于填充Bean时的自动类型转换器

com.chinasofti.util.jdbc.datasource：

存放一个建议的低侵入性的数据库连接池组件

com.chinasofti.util.jdbc.template：

存放JDBC的模版模式封装访问工具

com.chinasofti.util.jdbc.template.specialsqloperation：

存放JDBC访问工具中的数据库方言抽象和内置的数据库方言

com.chinasofti.util.sec：

存放对字符串的可逆和不可逆的加密算法组件

com.chinasofti.util.web.serverpush：

存放HTTP消息推送组件

com.chinasofti.util.web.upload：

存放带有利用Request自动填充JavaBean功能的文件上传组件

com.chinasofti.web.common.filters：

存放通用功能性质的过滤器组件

com.chinasofti.web.common.httpequest：

存放Servlet API共享组件

com.chinasofti.web.common.service：

存放验证码相关的服务组件

com.chinasofti.web.common.taglib：

存放防止重复提交的自定义标签库

com.chinasofti.web.servlets.common：

存放返回验证码的Servlet

### 配置文件规划

系统中存在的配置文件规定如下：

**JDBC数据库参数配置文件：**

配置文件路径：

**CLASSPATH**

|--**jdbc.xml**

配置文件类型：XML文件

配置文件格式：

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

<!-- 根节点 -->

<jdbc>

<!-- 驱动名 -->

<drivername>com.mysql.jdbc.Driver</drivername>

<!-- 连接字符串 -->

<connectionstring>jdbc:mysql://localhost:3306/ordersys

</connectionstring>

<!-- 数据库用户名 -->

<dbmsusername>root</dbmsusername>

<!-- 数据库密码 -->

<dbmspassword>root</dbmspassword>

</jdbc>

# HTTP消息推送模块设计说明

### 模块描述

本模块是系统功能组件支撑组件，用于保证系统中个功能组件之间的实时性消息数据交换，该模块利用了HTTP的长连接请求实现了在现阶段HTTP1.1/HTML4“主动请求，被动响应”环境中的数据推送功能。

### 功能

#### 模块图：

接收消息模块

推送消息模块

**消息消费者子模块**

**消息生产者子模块**

请求消息

队列等待

发送消息

MSG

检索队列

MSG

### 性能

#### 精度

消息消费者子模块必须确保原样推送生产者子模块发布的文本消息，不能对消息进行任何形式的修改变更。

#### 灵活性

推送模块应该可以推送任何以文本为承载形式的消息内容，包括普通的文本字符串、XML格式化数据、JSON格式化数据，且不能应为推送作业过程的存在而改变这些数据的可解析性。

#### 时间特性

模块中对等待队列的检索处理和数据的发布功能单一执行时间不应该超过10ms，因此不管平台部署环境为广域网环境还是局域网环境，消息推送的过程不应该超过网络数据延迟+10ms。

#### 输入项

消息推送业务模块向对应需要就收消息的模块发送的实时消息，消息必须基于计算机可识别的UTF-8文本，文本格式化方式不限，长度受HTTP协议限制，建议控制在11M以内，在异构系统中可以发送文本字符串、XML格式数据或JSON数据。

#### 输出项

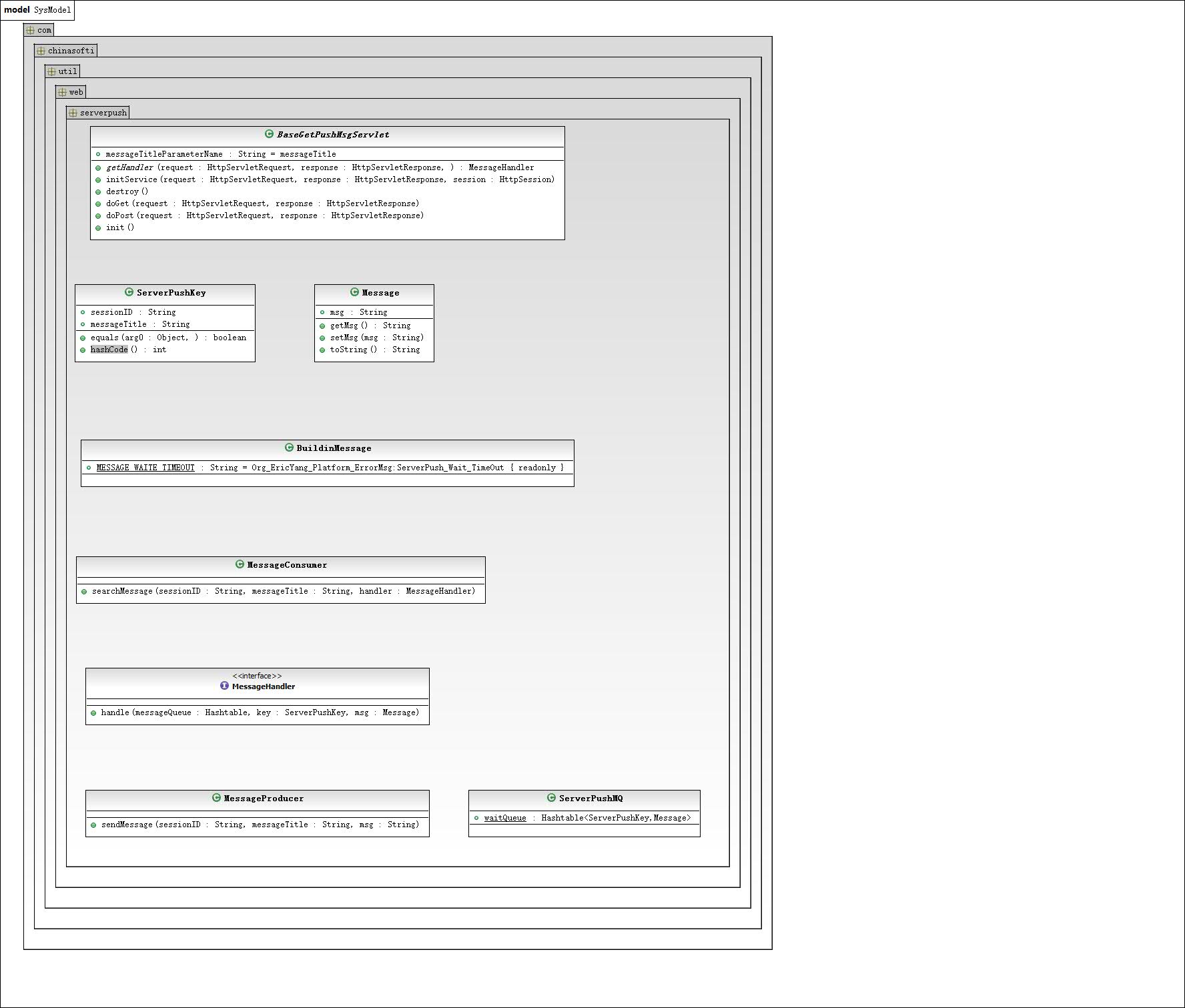
　　消息生产者子模块原样输出生产者推送的消息文本。

### 类设计

给出本模块的类设计，**包括类图和类说明**

对于J2EE应用，可以分控制类（例如用到的Servlet）、实体类（例如DAO）、业务类（例如处理业务的Handler）、视图类（例如JSP）、接口类（例如供别的模块调用的API）、工具类（例如对字符串进行处理的StringUtil）进行描述。JSP可以放在视图类中进行描述，描述包括使用到的重要的JavaScript。

#### 类图：



#### 类说明

1. Message类说明：
   1. 功能：用于保存并在生产者和消费者子模块间传递消息的类
   2. 主要方法：public void setMsg(String msg)：设置消息内容

public String toString()：获取消息内容

1. MessageHandler接口说明：
   1. 功能：用于定义自定义消息处理器的接口
   2. 主要方法：handle(Hashtable<ServerPushKey , Message> messageQueue , ServerPushKey key , Message msg)：自定义消息处理方法
2. MessageConsumer类说明：
   1. 功能：消息消费者子模块
   2. 主要方法：public void searchMessage(String sessionID , String messageTitle , MessageHandler handler)：消息消费方法

public String toString()：获取消息内容

1. MessageProducer类说明：
   1. 功能：消息生产者子模块
   2. 主要方法：public void sendMessage(String sessionID, String messageTitle, String msg)：生产消息方法
2. BaseGetPushMsgServlet类说明：
   1. 功能：消息消费者基础Servlet
   2. 主要方法：public abstract MessageHandler getHandler(final HttpServletRequest request , final HttpServletResponse response)：获取自定义消息处理器的方法

public void initService(final HttpServletRequest request , final HttpServletResponse response , final HttpSession session)：初始化Servlet方法

### 算法

**消息生产者：**

for (i = 0; i < 等待带列; i++) {

查询等待队列中是否存在等待的消费者

if (存在) {

消费者.消息=用户提供的消息字符串

消费者。消息.notify

｝

**消息消费者：**

new 消息对象

将新的消息对象设置给消费者

消息对象.wait

将消息对象加入等待队列

# 用户管理模块设计说明

### 模块描述

餐厅管理员能够利用本模块进行用户的管理，包括浏览与更新。

### 功能

#### 模块图：

系统数据库

浏览模块

ID

**更新子模块**

删除

修改

添加

用户详情

### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 用户点击添加用户菜单项 | 无 | 显示添加用户界面 |
| 用户点击编辑用户菜单项 | 用户ID | 显示编辑人员界面 |
| 用户点击用户详情菜单项 | 用户ID | 显示用户详情界面 |
| 用户点击删除用户菜单项 | 用户ID | 显示删除用户操作提示 |

#### 输入项

　　用户ID：用户在数据库中的标识项，为一个整数型数据。

#### 输出项

　　所有输出均显示在用户浏览器中。

#### 界面设计

1. 界面一：浏览用户列表



1. 界面二：浏览用户详情

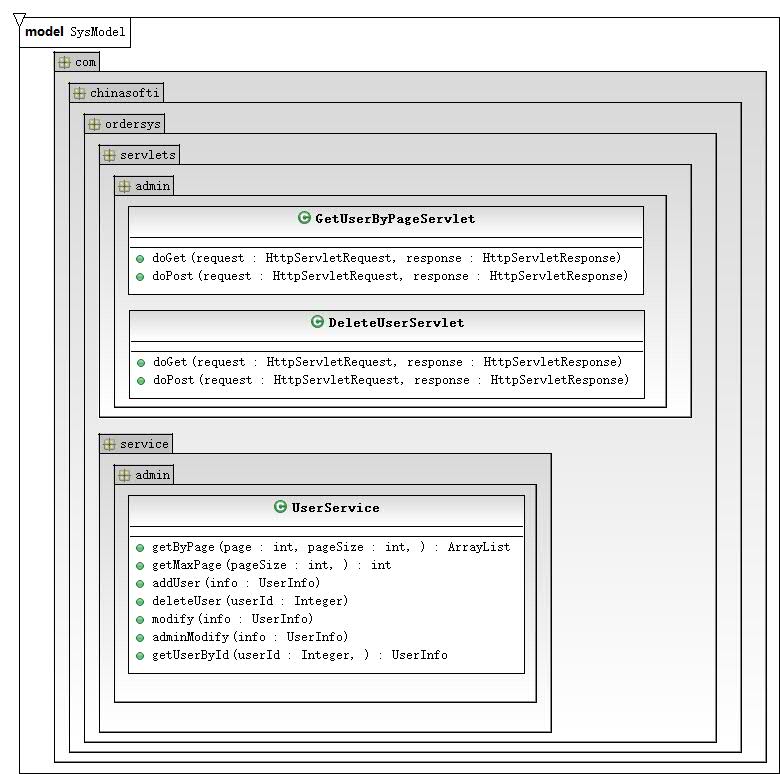


1. 界面三：添加用户



### 类设计

#### 类图：



#### 类说明

1. UserService类说明：
   1. 功能：对用户信息进行操作的服务组件
   2. 主要方法：public ArrayList<UserInfo> getByPage(int page, int pageSize)：以分页方式查询用户列表的方法。

public int getMaxPage(int pageSize)：获得最大页数的方法

public void addUser(UserInfo info)：添加用户的方法

public void deleteUser(Integer userId)：删除用户的方法  
 public void modify(UserInfo info)：修改用户的方法

public UserInfo getUserById(Integer userId)：查询用户详情的方法

### 算法

**查询用户列表：**

if(页码<1)

页码=1

select 最大页码数

if(页码>最大页码数)

页码=最大页码数

select 当前页码的数据内容

构建用户DOM对象

将DOM对象转换为XML

# 菜品管理模块设计说明

### 模块描述

餐厅管理员能够利用本模块进行菜品的管理，包括浏览与更新。

### 功能

#### 模块图：

系统数据库

浏览模块

ID

**更新子模块**

删除

修改

添加

菜品详情

### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 用户点击添加菜品菜单项 | 无 | 显示添加菜品界面 |
| 用户点击编辑菜品菜单项 | 菜品ID | 显示编辑菜品界面 |
| 用户点击菜品详情菜单项 | 菜品ID | 显示菜品详情界面 |
| 用户点击删除菜品菜单项 | 菜品ID | 显示删除菜品操作提示 |

#### 输入项

　　菜品ID：菜品在数据库中的标识项，为一个整数型数据。

#### 输出项

　　所有输出均显示在用户浏览器中。

#### 界面设计

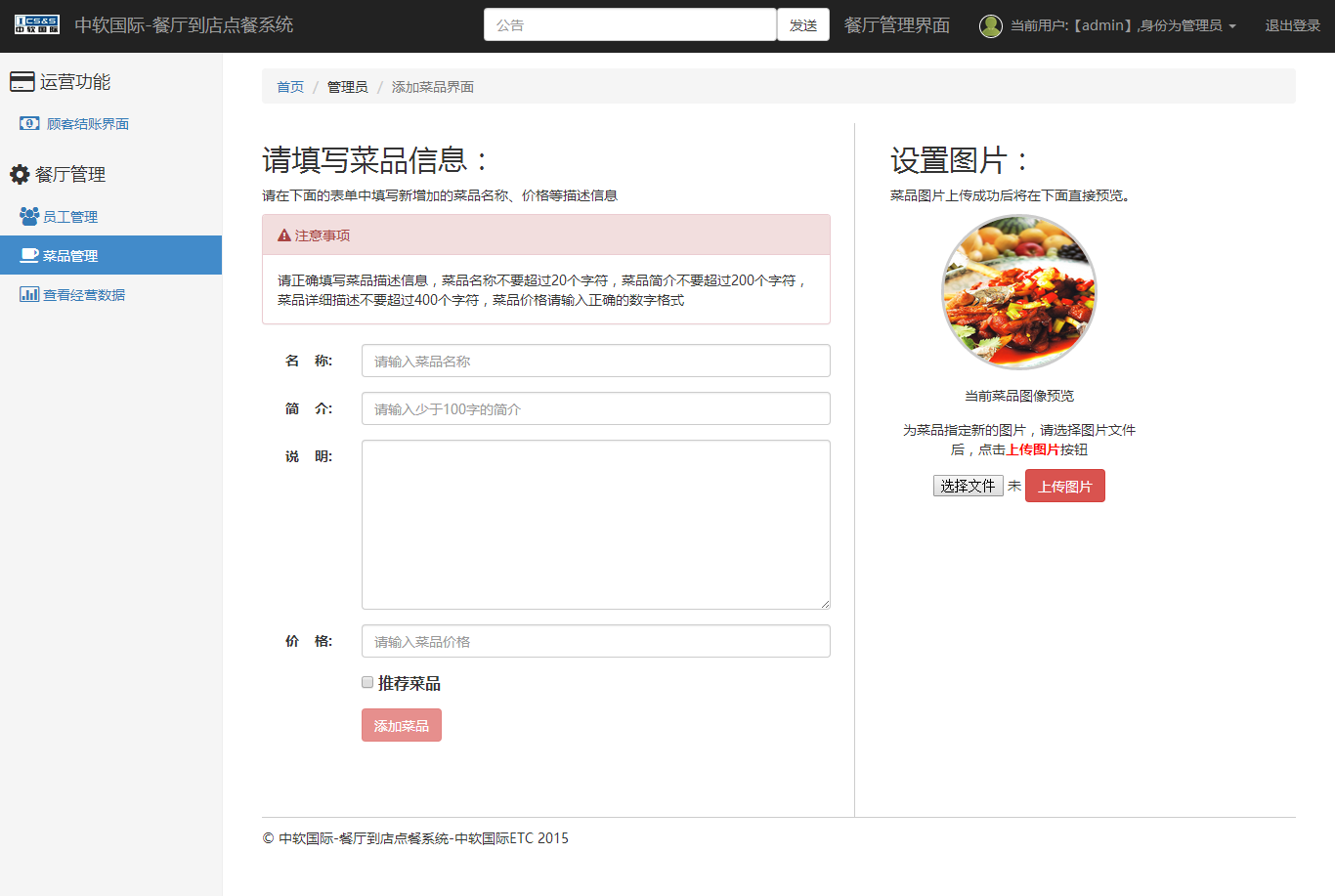
1. 界面一：浏览菜品列表



1. 界面二：浏览菜品详情

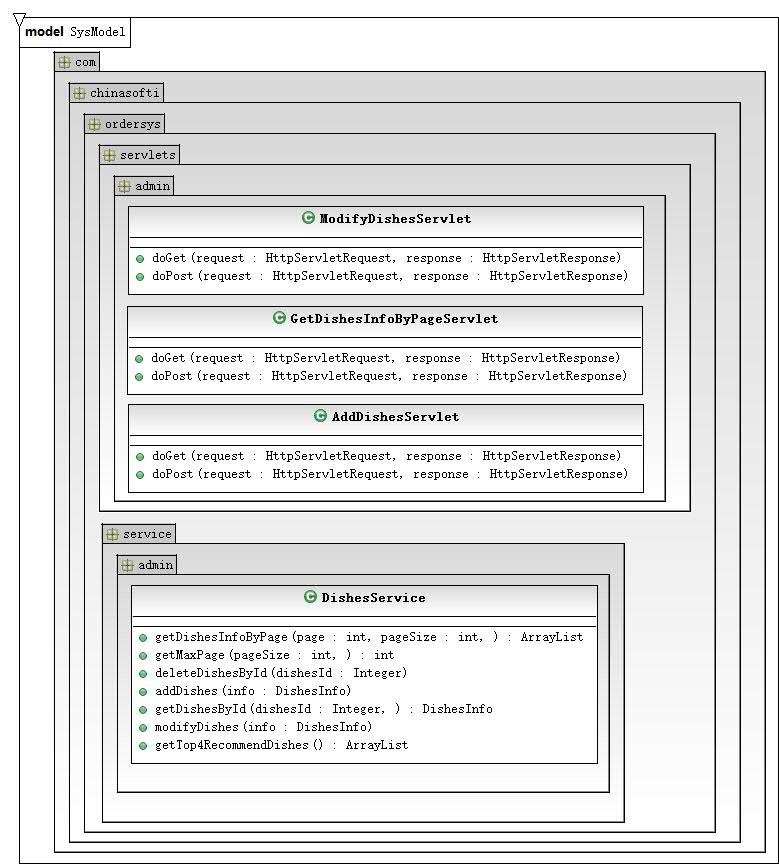


1. 界面三：添加菜品



### 类设计

#### 类图：



#### 类说明

1. DishesService类说明：
   1. 功能：对菜品信息进行操作的服务组件
   2. 主要方法：public ArrayList<DishesInfo> getDishesInfoByPage(int page, int pageSize)：以分页方式查询菜品列表的方法。

public int getMaxPage(int pageSize)：获得最大页数的方法

public void addDishes(DishesInfo info)：添加菜品的方法

public void deleteDishesById(Integer dishesId)：删除菜品的方法  
 public void modifyDishes(DishesInfo info)：修改菜品的方法

public DishesInfo getDishesById(Integer dishesId)：查询菜品详情的方法

### 算法

**查询用户列表：**

if(页码<1)

页码=1

select 最大页码数

if(页码>最大页码数)

页码=最大页码数

select 当前页码的数据内容

构建菜品DOM对象

将DOM对象转换为XML

# 点餐模块设计说明

### 模块描述

餐厅服务员利用点餐模块为顾客点餐，并实时将点取的菜品信息和数量推送到后厨界面。

### 功能

#### 模块图：

浏览模块

ID

**推送模块**

推送

数量

订单模块

后厨

### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 用户点击添加菜品数量按钮 | 现有数量 | 显示添加菜品界面 |
| 用户点击减少菜品数量按钮 | 现有数量 | 显示编辑菜品界面 |
| 用户点击菜品详情菜单项 | 菜品ID | 显示菜品详情界面 |
| 用户点击确认点单按钮 | 无 | 后厨显示点单信息 |

#### 输入项

菜品ID：菜品在数据库中的标识项，为一个整数型数据。

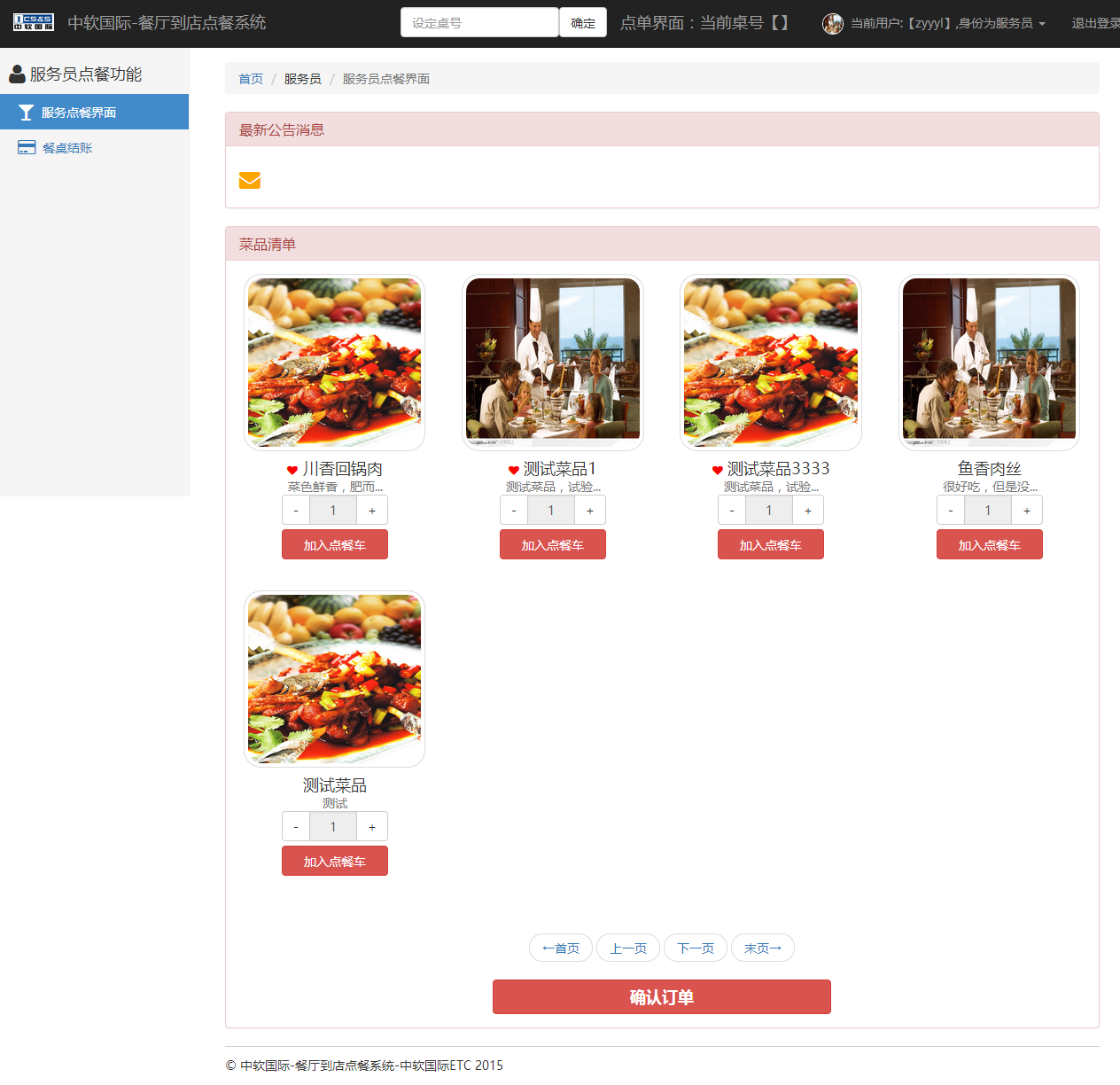
菜品数量：用户选择某一菜品的数量，不能小于1。

#### 输出项

　　所有输出均显示在用户浏览器中。

#### 界面设计

1. 界面一：浏览菜品列表



1. 界面二：浏览菜品详情

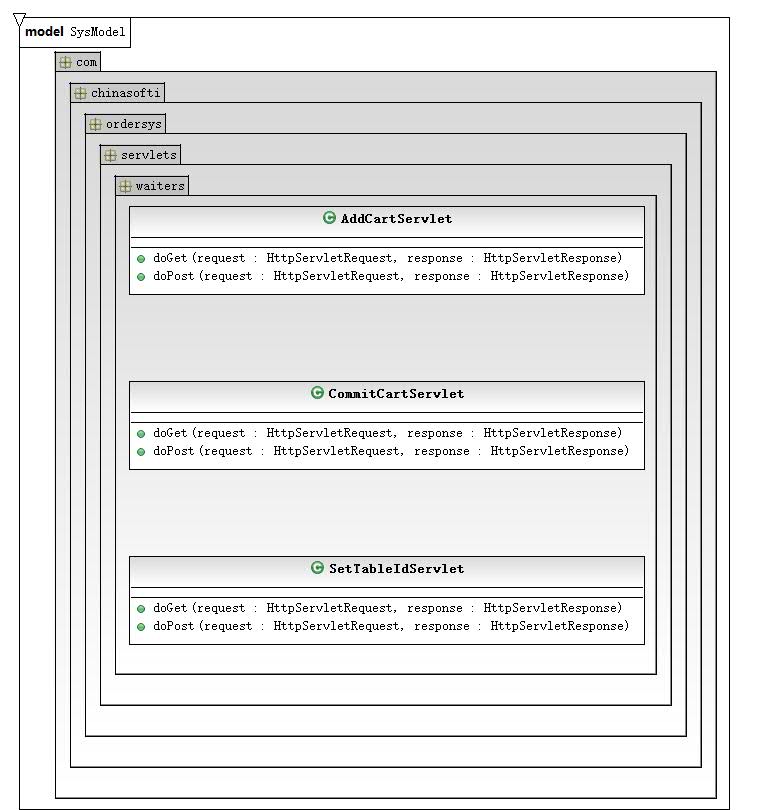


1. 界面三：后厨显示菜品推送消息



### 类设计

#### 类图：



### 算法

**查询用户列表：**

提交桌号

session.设置桌号

while(继续点菜){

if（查看菜品详情）

显示菜品详情

if(add)

增加菜品数量

If(sub&&数量>1)

减少菜品数量

}

提交点菜信息

# 传菜模块设计说明

### 模块描述

后厨人员可以记录菜品的烹饪状态，菜品烹制完成后推送传菜信息到餐厅点餐服务员界面。

### 功能

#### 模块图：

备菜模块

菜名

**推送模块**

推送

桌号

传菜模块

点餐员

### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 用户点击开始烹制按钮 | 无 | 显示烹制结束按钮 |
| 用户点击烹制结束按钮 | 无 | 服务员显示传菜信息 |

#### 输出项

　　所有输出均显示在用户浏览器中。

#### 界面设计

1. 界面一：烹制状态：



1. 界面二：显示传菜



### 类设计

#### 类图：



# 结算模块设计说明

### 模块描述

餐厅服务员将结算订单信息推送到餐厅管理员，餐厅管理员可以选择结算或直接免单。

### 功能

#### 模块图：

请求模块

ID

**推送模块**

推送

**结算模块**

详情

结算

免单

### 人机交互

#### 人机交互总表

|  |  |  |
| --- | --- | --- |
| **操作** | **输入** | **输出** |
| 用户点击订单详情按钮 | 订单ID | 显示订单详情界面 |
| 用户免单按钮 | 订单ID | 删除订单显示条目 |
| 用户结算按钮 | 订单ID | 删除订单显示条目 |

#### 输入项

订单ID：订单在数据库中的标识项，为一个整数型数据。

#### 输出项

　　所有输出均显示在用户浏览器中。

#### 界面设计

1. 界面一：选择结算订单



1. 界面二：待结算订单推送列表：

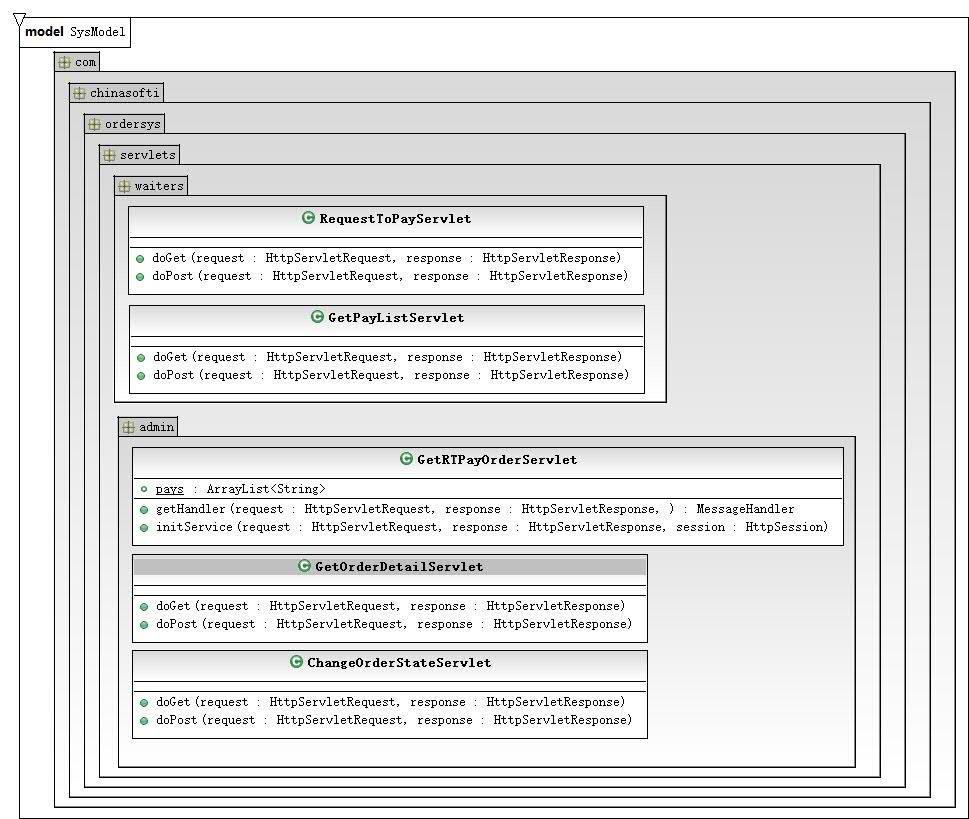


1. 界面三：订单详情



### 类设计

#### 类图：



### 算法

**查询用户列表：**

if(结账){

订单状态修改为已经结账

if(免单){

订单状态修改为已经免单