|  |  |  |
| --- | --- | --- |
| 文档状态： | 文档编号： |  |
| [ ] Draft  [√] Released  [ ]Modifying | 编 撰： |  |
| 编撰日期： |  |
| 保密级别： |  |
| 文档版本： | 1.0.0 |

食堂管理系统

（Canteen management system）

**软件体系结构设计说明书**

**食堂管理系统项目小组**

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **时间** | **修订内容** | **修订人** |
| 1 | 2023年03月29日 | 创建文档 | **刘康炼**  **符丽梅**  **黄嘉敏** |
|  |  |  |  |
|  |  |  |  |

目录

[（Canteen management system） 1](#_Toc23065)

[软件体系结构设计说明书 1](#_Toc13582)

[修订记录 2](#_Toc16856)

[1.简介 6](#_Toc31919)

[1.1.背景和目的 6](#_Toc14841)

[1.2.范围 6](#_Toc17501)

[1.3.术语和缩略语 6](#_Toc31579)

[2.系统总体设计 7](#_Toc15639)

[2.1.任务概述 7](#_Toc18480)

[2.1.1. 目标 7](#_Toc26798)

[2.1.2. 需求概述 7](#_Toc27080)

[2.2.设计概述 8](#_Toc1773)

[2.2.1. 总体约束 8](#_Toc13716)

[2.2.2. 系统外部接口 8](#_Toc8460)

[2.2.3.设计方案概述 8](#_Toc6991)

[2.3.系统架构设计 9](#_Toc19889)

[2.3.1. 系统的逻辑架构设计 9](#_Toc3447)

[2.3.2. 系统的物理架构设计 11](#_Toc23056)

[2.4.子系统定义 11](#_Toc24183)

[2.4.1. 子系统列表 11](#_Toc10467)

[2.4.2. 子系统间关系 12](#_Toc16429)

[3.前台管理子系统设计 13](#_Toc24148)

[3.1.任务概述 13](#_Toc16573)

[3.1.1. 目标 13](#_Toc26645)

[3.1.2. 需求概述 13](#_Toc11051)

[3.2.设计概述 13](#_Toc11588)

[3.2.1. 总体约束 13](#_Toc21542)

[3.3.子系统架构设计 15](#_Toc32143)

[3.4.模块定义 16](#_Toc5654)

[3.4.1.模块列表 17](#_Toc360)

[3.4.2. 模块间关系 17](#_Toc1838)

[3.4.3. 模块描述 17](#_Toc11099)

[4.后台管理子系统设计 18](#_Toc14410)

[4.1.任务概述 18](#_Toc5962)

[4.1.1. 目标 18](#_Toc21078)

[4.1.2. 需求概述 19](#_Toc7355)

[4.2.设计概述 19](#_Toc12438)

[4.2.1. 总体约束 19](#_Toc27752)

[4.2.2. 子系统外部接口 20](#_Toc31200)

[4.2.3.设计方案概述 20](#_Toc22833)

[4.3.子系统架构设计 20](#_Toc24824)

[4.4.模块定义 21](#_Toc23870)

[4.4.1.模块列表 22](#_Toc27852)

[4.4.2. 模块间关系 23](#_Toc17261)

[4.4.3. 模块描述 23](#_Toc5386)

[5. 开发视图 24](#_Toc6842)

[6.物理视图 25](#_Toc3062)

[7.运行视图 25](#_Toc28940)

# 1.简介

## 1.1.背景和目的

本文档编制的目的是说明对食堂管理软件系统的设计考虑，包括软件系统的基本处理流程，软件系统的组织结构、模块划分、功能分配、接口设计、运行设计、数据结构设计和出错处理设计等，为软件的详细设计奠定基础。

本应用本着方便用户提前了解当天食堂菜品，可进行线上提前点餐等功能，方便商家对菜品的管理以及对订单的管理等。这样既能提高了工作效率，又节省了人力资源，为食堂巨大人流量的创造了更便捷的发展空间。

## 1.2.范围

本文档适用于食堂管理系统的服务端。

## 1.3.术语和缩略语

1、Browser /Server模式的体系结构:B/S结构是WEB兴起后的一种网络结构模式，WEB浏览器是客户端最主要的应用软件。这种模式统一了客户端，将系统功能实现的核心部分集中到服务器上，简化了系统的开发、维护和使用。

2、Dreamweaver MX 软件的工作界面， 该软件把它所有的功能都集中成在一个个功能面板上, 用户除了可以像其他软件一样通过菜单命令操作外,还可以通过 MX 界面上的各类面板直接进行操作。

# 2.系统总体设计

## 2.1.任务概述

### 2.1.1. 目标

实现目标：这个是食堂的食堂员工对顾客点餐的管理更方便，更高效；为了减少学生的点餐时间，方便更多的顾客能快速愉快的进行点餐；为了方便管理员对销售数据和学生们的意见反馈进行分析，根据具体情况进行菜单调整。

功能:①学生:登录/注册、模拟学生就餐(点餐，消费金额，消费记录)、意见反馈(评论店铺/查看评论);②员工:登录/注册、 订单管理(查看订单，配送订单等);③管理员:登录/注册、学生管理、员工管理、菜品管理（确定菜单，包括种类、数量、价格等）、对销售数据和学生们的意见反馈进行分析，管理员根据具体情况进行调整。

### 2.1.2. 需求概述

食堂管理系统是一套功能强大、操作简单、使用的自动化管理软件。顾客。包括用户管理、员工管理、订餐管理、菜单管理、后台管理等。系统应该能够准确记录每个食堂的销售数据和成本，以便管理人员进行决策和分析。此外，系统还应该提供用户友好的界面和易于使用的功能，以满足食堂管理人员和食堂用户的需求。

## 2.2.设计概述

### 2.2.1. 总体约束

本项目的约束包括:无项目经费;项目开发时间不超过一年;主要负责人1人，开发小组共3人;对信息安全和保密无约束。

### 2.2.2. 系统外部接口

软件接口：与数据库接口，传递信息：学生账号，管理员号，员工号。

### 2.2.3.设计方案概述

建立一套食堂管理系统顾客点餐员工配餐管理员管理的方案:

1)前端运用HTML、CSS、JavaScript、jQuery（js框架）构建可视化界面后端运用jsp、servlet建立后台业务逻辑和数据交互。

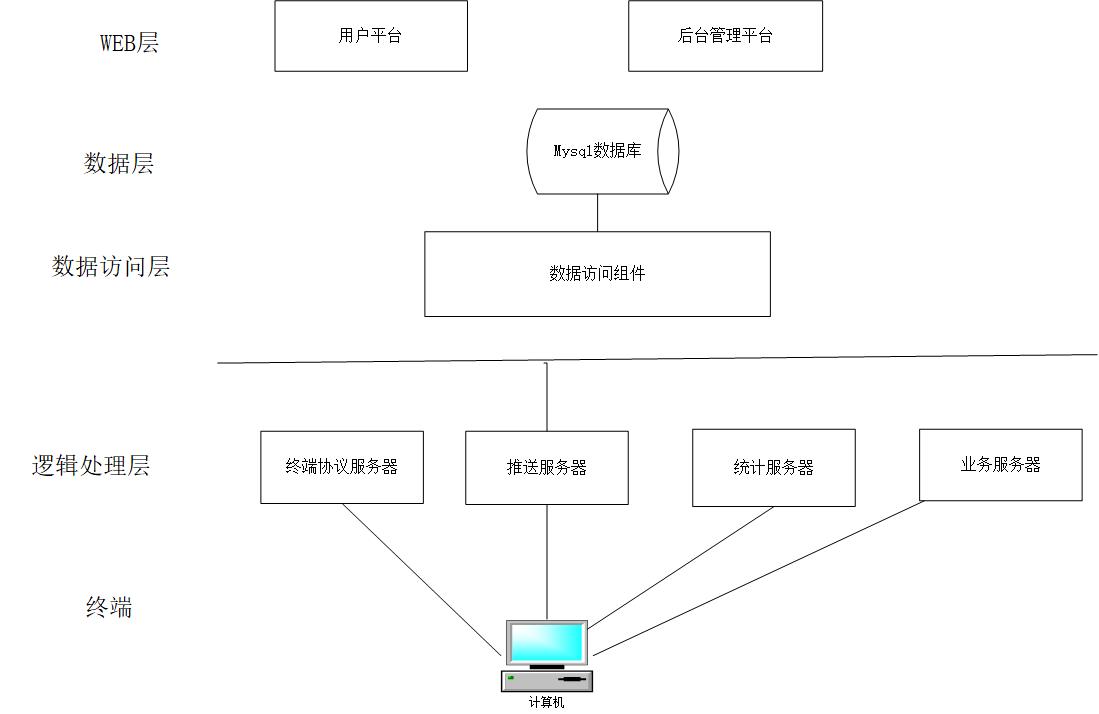
2) 分别建立后台管理平台：管理员和员工登录注册模块，用户管理模块，订单管理模块，菜品管理模块，意见反馈模块。

用户平台：用户登录注册模块，订单模块，菜品模块，意见反馈模块。

## 2.3.系统架构设计

### 2.3.1. 系统的逻辑架构设计

#### 2.3.1.1.单IDC服务器架构图



分五层结构：逻辑处理层、数据访问层、数据层、WEB层。

1)逻辑处理层：终端协议服务器、推送服务器（RS）、统计服务器。

a)终端协议服务器：终端往接入服务器发送协议包，接入服务器转发给终端协议服务器处理；

b)推送服务器：负责从MQ取出订单消息、解析消息（条件体和消息体）、终端在线和离线查询、按各接入服务器有哪些终端登录把消息包分发给接入服务器；

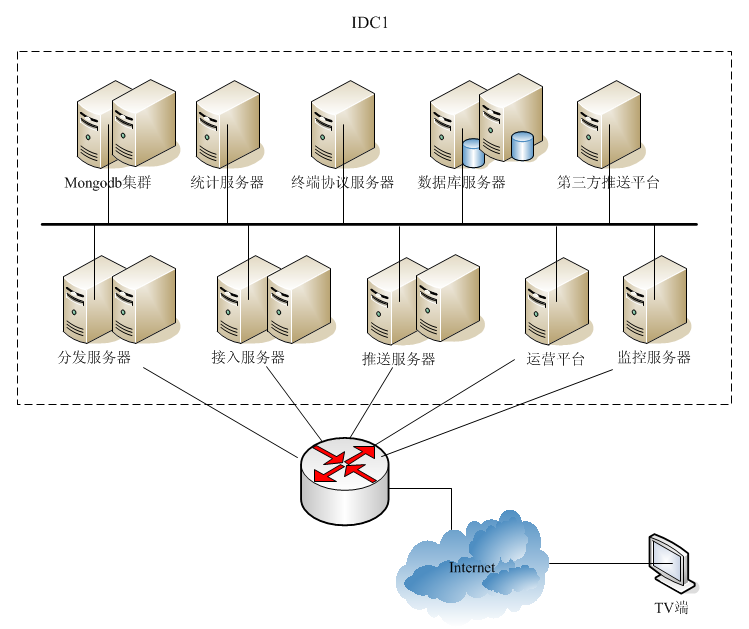
c)统计服务器：运用quartz执行定时任务，实现终端每天在线数、终端每小时在线数、终端每天激活数、终端登录次数、消息推送结果等统计功能。

4)数据访问层：提供数据访问组件，如Mysql JDBC接口。

5)数据层： Mysql数据库（主从+读写分离）。

6)WEB层：用户平台，后台管理平台。

### 2.3.2. 系统的物理架构设计



## 2.4.子系统定义

### 2.4.1. 子系统列表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统编号 | 子系统名称（标识） | 功能描述 | 开发方式 |
| 1 | 运营平台子系统 | 由管理员和运营人员使用，提供第三方管理、基础数据、消息管理、运维、统计、系统设置、日志查询等功能。 | 自行开发 |
| 2 | 前台管理子系统 | 用户端的是用户登录注册模块，订单模块，菜品模块，意见反馈模块 | 自行开发 |
| 3 | 后台管理子系统 | 管理员和员工登录注册模块，用户管理模块，订单管理模块，菜品管理模块，意见反馈模块 | 自行开发 |

# 3.前台管理子系统设计

## 3.1.任务概述

### 3.1.1. 目标

### 实现客户登录/注册、模拟学生就餐(点餐，消费金额，消费记录)、意见反馈(评论店铺/查看评论)等。

### 3.1.2. 需求概述

### 参考需求规格说明书

## 3.2.设计概述

### 3.2.1. 总体约束

#### 3.2.1.1.开发规范

1.遵循MVC设计模式：MVC是Model-View-Controller的缩写，是一种常用的设计模式，用于将应用程序的输入、处理和输出分离。在JavaWeb应用程序中，Model代表数据模型，View代表用户界面，Controller代表控制器。通过MVC设计模式，可以实现应用程序的高内聚、低耦合，提高代码的可维护性和可扩展性。

2. 遵循分层架构设计：分层架构是将应用程序按照不同的层次进行划分，每个层次都有自己的职责和功能。常用的分层架构包括三层架构和四层架构。在JavaWeb应用程序中，常用的分层架构是MVC架构和三层架构。通过分层架构设计，可以实现应用程序的模块化、可重用性和可测试性。

3. 遵循设计模式：设计模式是一种经过实践证明的、可重用的软件设计经验。在JavaWeb应用程序中，常用的设计模式包括工厂模式、单例模式、观察者模式、策略模式等。通过设计模式的应用，可以提高应用程序的可维护性、可扩展性和可重用性。

4. 遵循Java编码规范：Java编码规范是一种约定俗成的编码规范，包括命名规范、代码格式规范、注释规范等。在JavaWeb应用程序中，遵循Java编码规范可以提高代码的可读性和可维护性。

（1）编码约定

1. 命名规范：变量、方法、类、接口、包等的命名应该清晰、简洁、有意义，符合驼峰命名法。类名应该以大写字母开头，方法名和变量名应该以小写字母开头。  
  
2. 代码格式规范：代码应该按照一定的格式进行排版，以提高代码的可读性和可维护性。应该使用缩进、空格、换行等符号进行排版。  
  
3. 注释规范：注释应该清晰、简洁、有意义，对代码的功能、参数、返回值等进行说明。应该在方法、类、变量等的上方进行注释。  
  
4. 异常处理规范：在代码中应该对可能出现的异常进行处理，避免程序出现崩溃等问题。应该使用try-catch语句对可能出现的异常进行捕获和处理。  
  
5. 安全规范：在JavaWeb应用程序中，应该注意对用户输入的数据进行校验和过滤，以避免安全漏洞。应该使用安全框架和加密算法等技术保证应用程序的安全性。  
  
6. 性能规范：在JavaWeb应用程序中，应该注意代码的性能和效率。应该使用缓存、连接池等技术提高应用程序的性能。

（2）文件约定

1. 目录结构：JavaWeb应用程序的目录结构应该按照一定的规范进行组织，以便于管理和维护。通常情况下，应该将Java源代码、配置文件、JSP文件、静态资源文件等分别放置在不同的目录下。  
  
2. Java源代码：Java源代码应该存放在src目录下，按照包名进行组织。每个包应该对应一个功能模块或者一个业务领域，包名应该使用小写字母，多个单词之间使用点号分隔。  
  
3. 配置文件：JavaWeb应用程序通常需要使用一些配置文件，如web.xml、log4j.properties等。这些配置文件应该存放在WEB-INF目录下，以便于保护应用程序的安全性。  
  
4. JSP文件：JSP文件应该存放在WebContent目录下，按照功能模块进行组织。每个JSP文件应该对应一个功能模块或者一个业务领域。  
  
5. 静态资源文件：静态资源文件包括图片、CSS、JavaScript等文件。这些文件应该存放在WebContent目录下的相应子目录中，按照功能模块进行组织。  
  
6. 数据库脚本：数据库脚本应该存放在一个独立的目录下，以便于管理和维护。通常情况下，应该将数据库脚本放置在src目录下的一个子目录中。

（3）目录约定

1. src目录：该目录用于存放Java源代码，按照包名进行组织。每个包应该对应一个功能模块或者一个业务领域，包名应该使用小写字母，多个单词之间使用点号分隔。  
  
2. WebContent目录：该目录用于存放Web应用程序的资源文件，包括JSP文件、静态资源文件、配置文件等。通常情况下，该目录下包含以下几个子目录：  
  
- WEB-INF目录：该目录用于存放Web应用程序的配置文件和受保护的资源文件，如web.xml、classes目录、lib目录等。  
  
- META-INF目录：该目录用于存放Web应用程序的元数据文件，如MANIFEST.MF等。  
  
- lib目录：该目录用于存放Web应用程序所需的第三方库文件。  
  
- classes目录：该目录用于存放编译后的Java类文件。  
  
- css目录：该目录用于存放CSS文件。  
  
- js目录：该目录用于存放JavaScript文件。  
  
- images目录：该目录用于存放图片文件。  
  
3. db目录：该目录用于存放数据库脚本文件，如sql、DML等。

#### 3.2.1.2.内外部约束

1. 分层约束：将系统划分为不同的层次，每一层都有其特定的职责和功能。常见的分层约束包括MVC（Model-View-Controller）模式、三层架构模式等。  
  
2. 模块化约束：将系统划分为多个独立的模块，每个模块都有其特定的职责和功能。模块之间通过接口进行交互，实现松耦合的设计。常见的模块化约束包括OSGI（Open Service Gateway Initiative）模式、插件化模式等。  
  
3. 安全约束：保护系统的安全性，防止未经授权的访问和数据泄露等安全问题。常见的安全约束包括身份认证、权限控制、数据加密等。  
  
4. 性能约束：保证系统的性能，提高系统的响应速度和吞吐量。常见的性能约束包括缓存、负载均衡、数据库优化等。  
  
5. 可扩展性约束：保证系统的可扩展性，方便系统的扩展和升级。常见的可扩展性约束包括面向接口编程、依赖注入、插件化设计等。

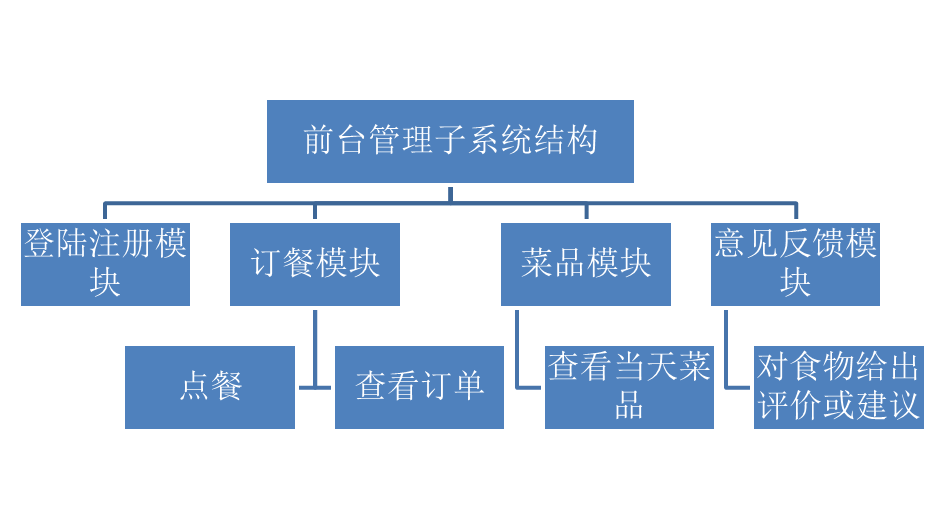
3.2.2. 子系统外部接口

硬件接口：与出单机接口，即用户提交订单后，员工那边会及时出单号，传递订单信息，包括菜品、就餐方式、配送时间等。

软件接口：与数据库接口，传递用户账号、选择的菜品等信息。3.2.3.设计方案概述

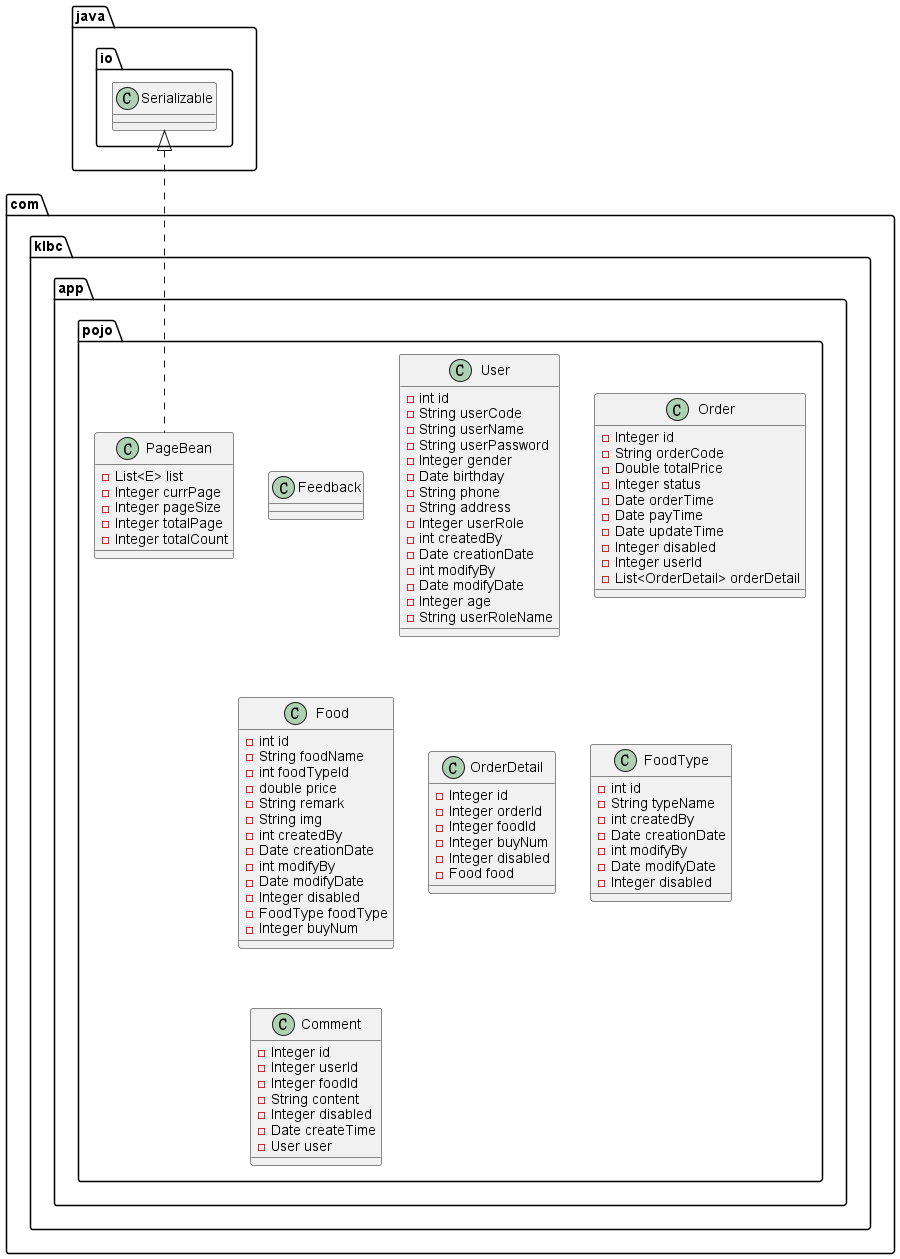
建立一套食堂管理系统用户顾客点餐的方案:前端设计用户端APP主要运用了HTML、CSS、JavaScript、jQuery（js框架）构建用户端，即是用户登录注册的模块，订单模块，菜品模块，意见反馈模块

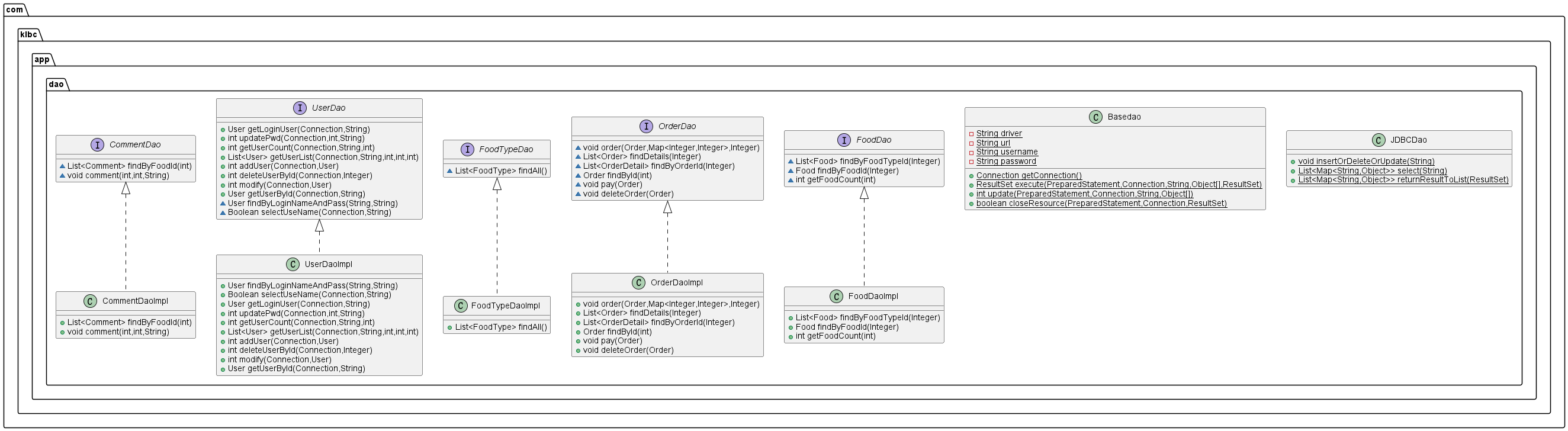
## 3.3.子系统架构设计

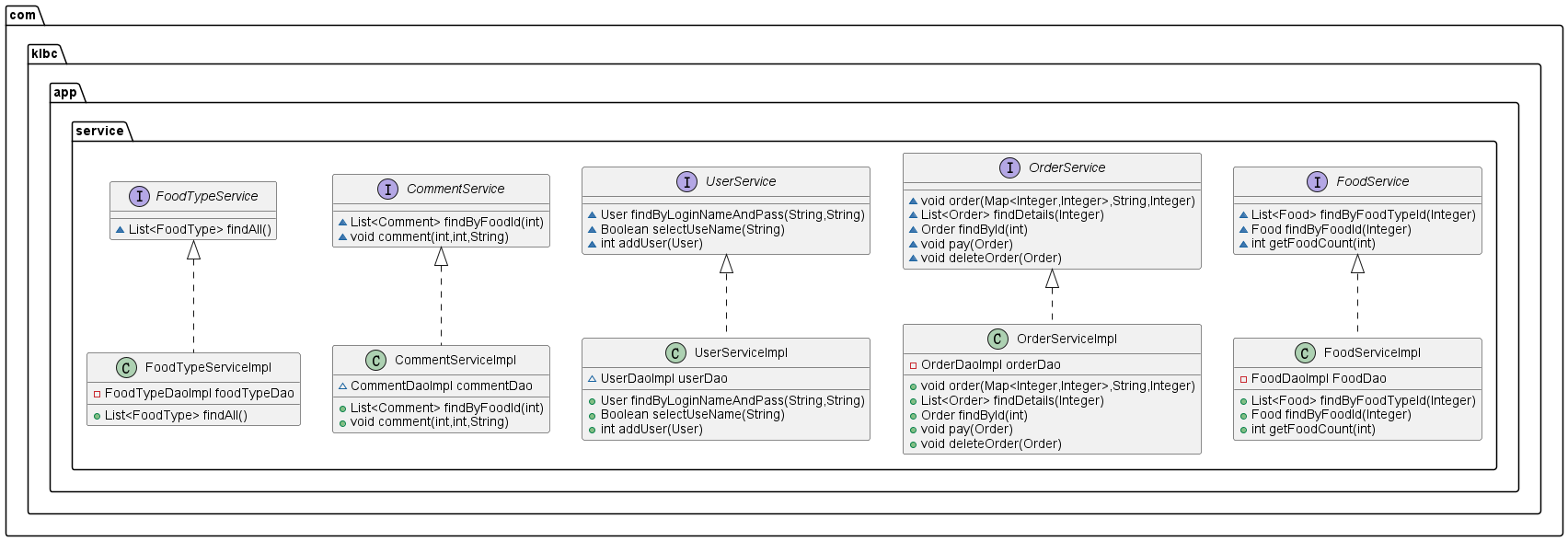


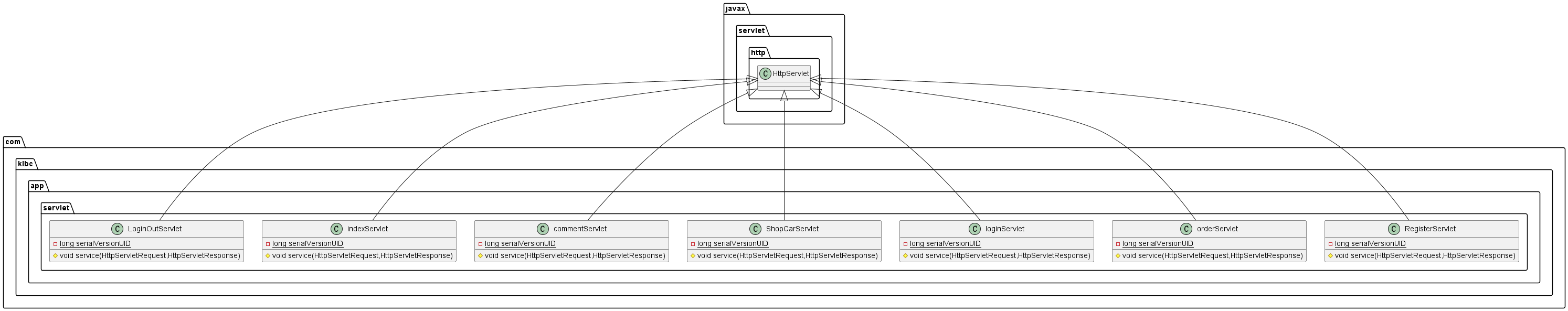
系统UML类图如下

用户端类图









## 3.4.模块定义

3.1. 登陆注册子模块：实现用户登录和注册

3.2. 订单子模块：用户可以进行点餐以及查看订单等

3.3. 菜品子模块：用户可查看当天菜品

3.4. 意见反馈子模块：当用户购买菜品后，可以给予一些意见和建议

### 3.4.1.模块列表

|  |  |  |
| --- | --- | --- |
| 模块名称（标识） | 功能简述 | 开发方式 |
| 登陆注册子模块 | 实现用户登录和注册 | 自行开发 |
| 订单子模块 | 用户可以进行点餐以及查看订单等 | 自行开发 |
| 菜品子模块 | 用户可查看当天菜品 | 自行开发 |
| 意见反馈子模块 | 当用户购买菜品后，可以给予一些意见和建议 | 自行开发 |

注：开发方式可选：采购、外包/自行开发/复用

### 3.4.2. 模块间关系

### *328a17d794421528b17869eb91b3b2b*3.4.3. 模块描述

#### 3.4.3.1.模块1

1）功能描述

实现用户登录和注册

2）性能描述

用户命令的响应时间小于1.5秒。

3）接口描述

软件接口：MYSQL数据库，传递用户账户信息。

#### 3.4.3.2.模块2

1）功能描述

用户可以进行点餐以及查看订单等

2）性能描述

用户命令的响应时间小于1.5秒，用户下单时间与员工收到订单时间差不超过3秒。

1. 接口描述

软件接口：MYSQL数据库，及时推送顾客订单信息及用户点餐信息。

# 4.后台管理子系统设计

## 4.1.任务概述

### 4.1.1. 目标

建立管理员和员工登录注册模块，用户管理模块，订单管理模块，菜品管理模块，意见反馈模块

### 4.1.2. 需求概述

### 参考需求规格说明书

## 4.2.设计概述

### 4.2.1. 总体约束

#### 4.2.1.1.开发规范

（1）编码约定

1. 命名规范：变量、方法、类、接口、包等的命名应该清晰、简洁、有意义，符合驼峰命名法。类名应该以大写字母开头，方法名和变量名应该以小写字母开头。  
  
2. 代码格式规范：代码应该按照一定的格式进行排版，以提高代码的可读性和可维护性。应该使用缩进、空格、换行等符号进行排版。  
  
3. 注释规范：注释应该清晰、简洁、有意义，对代码的功能、参数、返回值等进行说明。应该在方法、类、变量等的上方进行注释。  
  
4. 异常处理规范：在代码中应该对可能出现的异常进行处理，避免程序出现崩溃等问题。应该使用try-catch语句对可能出现的异常进行捕获和处理。  
  
5. 安全规范：在JavaWeb应用程序中，应该注意对用户输入的数据进行校验和过滤，以避免安全漏洞。应该使用安全框架和加密算法等技术保证应用程序的安全性。  
  
6. 性能规范：在JavaWeb应用程序中，应该注意代码的性能和效率。应该使用缓存、连接池等技术提高应用程序的性能。

（2）文件约定

1. 目录结构：JavaWeb应用程序的目录结构应该按照一定的规范进行组织，以便于管理和维护。通常情况下，应该将Java源代码、配置文件、JSP文件、静态资源文件等分别放置在不同的目录下。  
  
2. Java源代码：Java源代码应该存放在src目录下，按照包名进行组织。每个包应该对应一个功能模块或者一个业务领域，包名应该使用小写字母，多个单词之间使用点号分隔。  
  
3. 配置文件：JavaWeb应用程序通常需要使用一些配置文件，如web.xml、log4j.properties等。这些配置文件应该存放在WEB-INF目录下，以便于保护应用程序的安全性。  
  
4. JSP文件：JSP文件应该存放在WebContent目录下，按照功能模块进行组织。每个JSP文件应该对应一个功能模块或者一个业务领域。  
  
5. 静态资源文件：静态资源文件包括图片、CSS、JavaScript等文件。这些文件应该存放在WebContent目录下的相应子目录中，按照功能模块进行组织。  
  
6. 数据库脚本：数据库脚本应该存放在一个独立的目录下，以便于管理和维护。通常情况下，应该将数据库脚本放置在src目录下的一个子目录中。

（3）目录约定

1. src目录：该目录用于存放Java源代码，按照包名进行组织。每个包应该对应一个功能模块或者一个业务领域，包名应该使用小写字母，多个单词之间使用点号分隔。  
  
2. WebContent目录：该目录用于存放Web应用程序的资源文件，包括JSP文件、静态资源文件、配置文件等。通常情况下，该目录下包含以下几个子目录：  
  
- WEB-INF目录：该目录用于存放Web应用程序的配置文件和受保护的资源文件，如web.xml、classes目录、lib目录等。  
  
- META-INF目录：该目录用于存放Web应用程序的元数据文件，如MANIFEST.MF等。  
  
- lib目录：该目录用于存放Web应用程序所需的第三方库文件。  
  
- classes目录：该目录用于存放编译后的Java类文件。  
  
- css目录：该目录用于存放CSS文件。  
  
- js目录：该目录用于存放JavaScript文件。  
  
- images目录：该目录用于存放图片文件。  
  
3. db目录：该目录用于存放数据库脚本文件，如sql、DML等。

#### 4.2.1.2.内外部约束

1. 分层约束：将系统划分为不同的层次，每一层都有其特定的职责和功能。常见的分层约束包括MVC（Model-View-Controller）模式、三层架构模式等。  
  
2. 模块化约束：将系统划分为多个独立的模块，每个模块都有其特定的职责和功能。模块之间通过接口进行交互，实现松耦合的设计。常见的模块化约束包括OSGI（Open Service Gateway Initiative）模式、插件化模式等。  
  
3. 安全约束：保护系统的安全性，防止未经授权的访问和数据泄露等安全问题。常见的安全约束包括身份认证、权限控制、数据加密等。  
  
4. 性能约束：保证系统的性能，提高系统的响应速度和吞吐量。常见的性能约束包括缓存、负载均衡、数据库优化等。  
  
5. 可扩展性约束：保证系统的可扩展性，方便系统的扩展和升级。常见的可扩展性约束包括面向接口编程、依赖注入、插件化设计等。

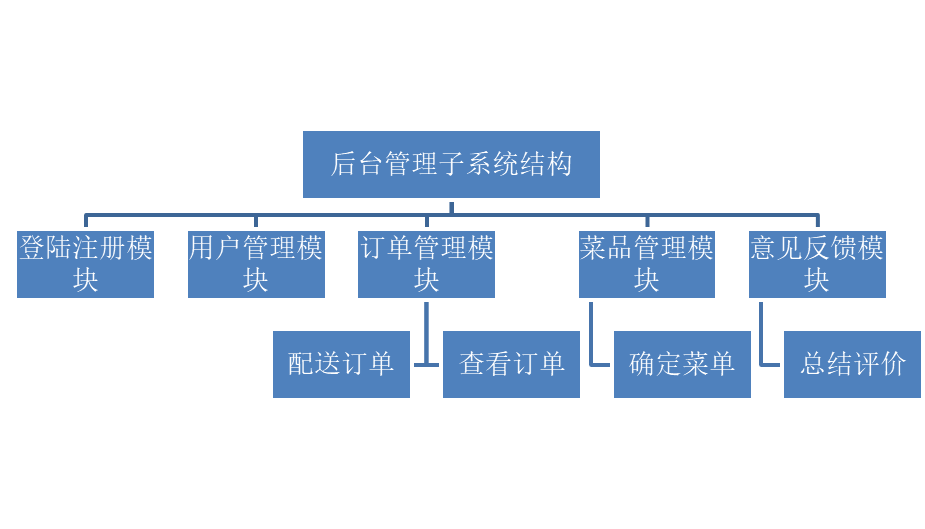
### 4.2.2. 子系统外部接口

### 软件接口：与数据库接口，传递管理员号、员工账号账号、订单信息等。

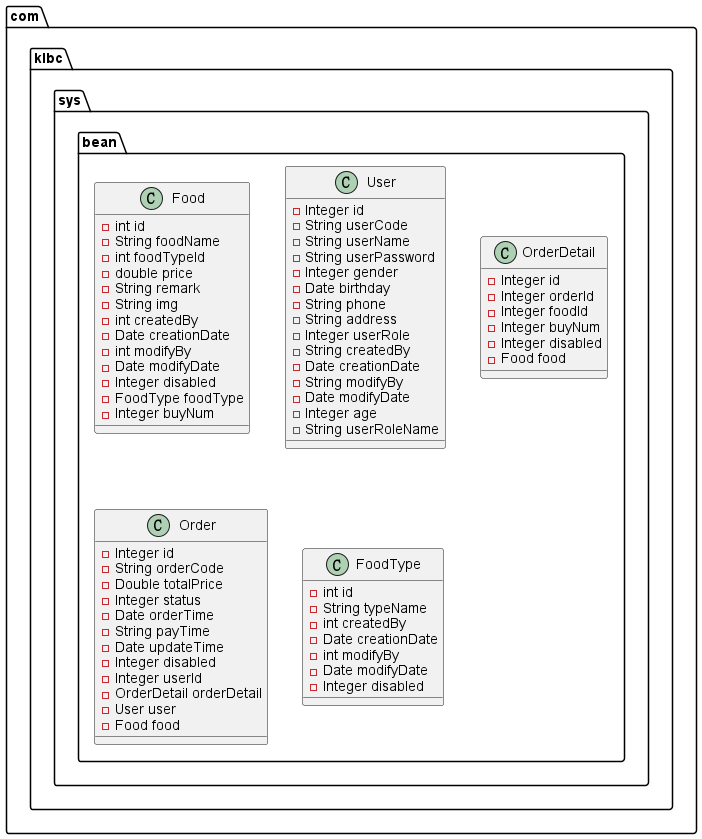
### 4.2.3.设计方案概述

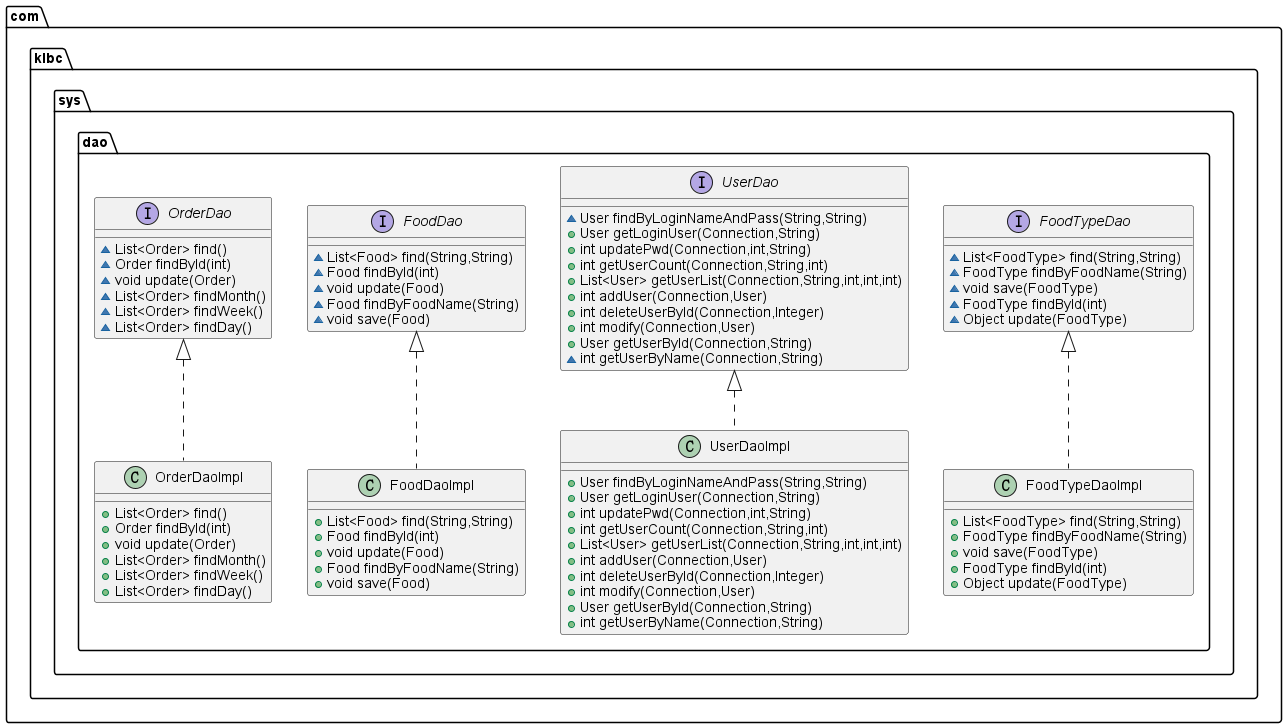
### 建立一套食堂管理系统员工配餐管理员管理的方案:后端运用jsp、servlet建立管理员和员工登录注册模块，用户管理模块，订单管理模块，菜品管理模块，意见反馈模块

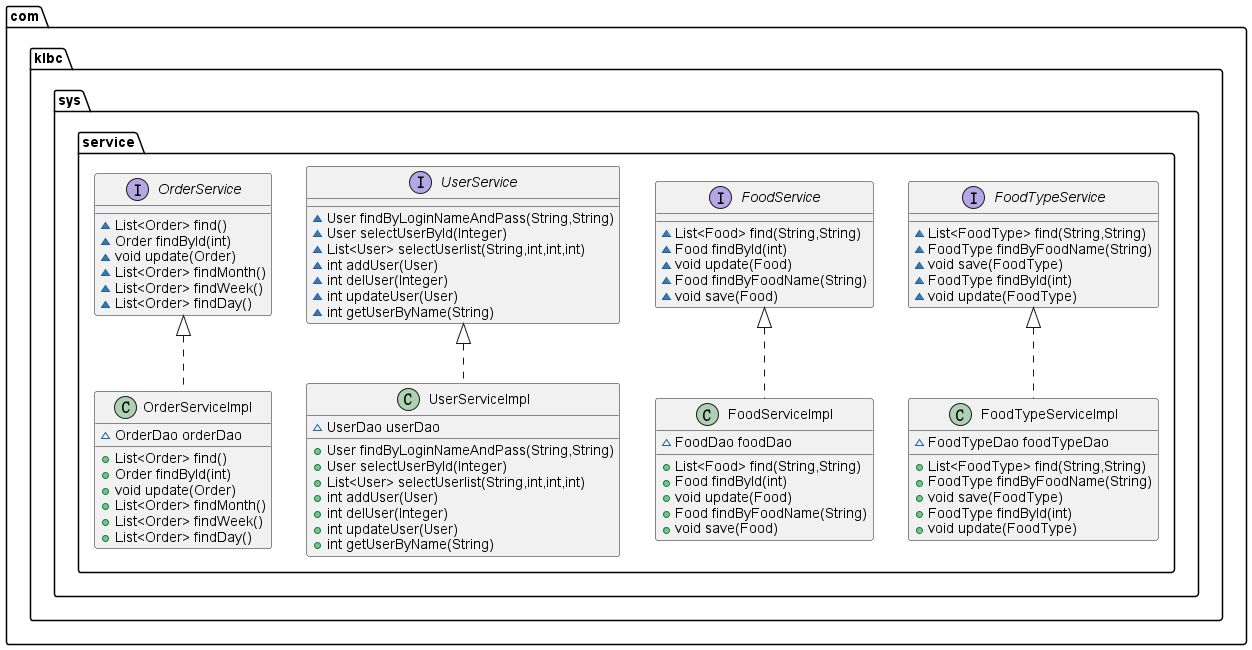
## 4.3.子系统架构设计

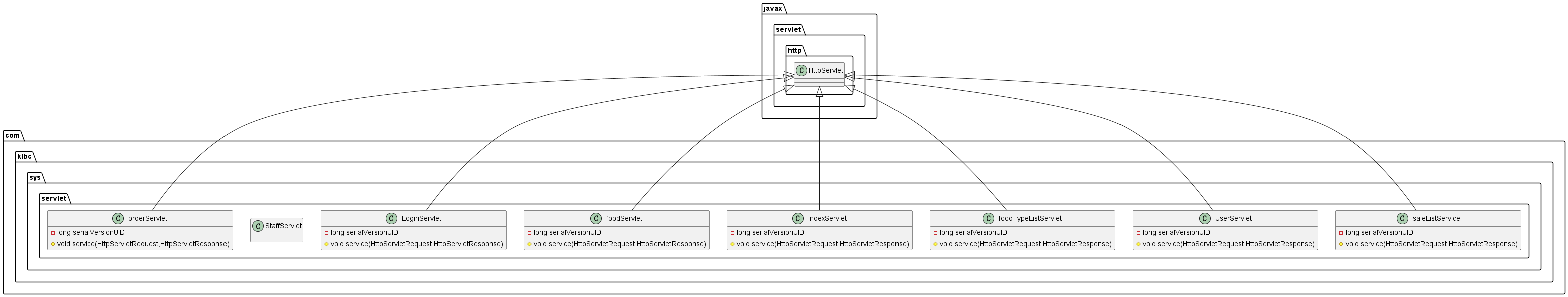


后台端类图









## 4.4.模块定义

4.1. 登陆注册子模块：实现员工和管理员登录和注册

4.2. 用户管理子模块：管理员可对员工和用户进行管理（增删改查等）

4.3. 订单管理子模块：员工可以查看订单并且提前备餐，根据需要看是否进行配送

4.4. 意见反馈子模块：员工可根据运营情况提供相关建议；管理员查看用户和员工建议，适当做出改进

4.5．菜品管理子系统：管理员提前确定当天菜品信息，包括菜品种类、价格、剩余数量等

### 4.4.1.模块列表

|  |  |  |
| --- | --- | --- |
| 模块名称（标识） | 功能简述 | 开发方式 |
| 登陆注册子模块 | 实现员工和管理员登录和注册 | 自行开发 |
| 用户管理子模块 | 管理员可对员工和用户进行管理（增删改查等） | 自行开发 |
| 订单管理子模块 | 员工可以查看订单并且提前备餐，根据需要看是否进行配送 | 自行开发 |
| 意见反馈子模块 | 员工可根据运营情况提供相关建议；管理员查看用户和员工建议，适当做出改进 | 自行开发 |
| 菜品管理子模块 | 管理员提前确定当天菜品信息，包括菜品种类、价格、剩余数量等 | 自行开发 |

注：开发方式可选：采购、外包/自行开发/复用

### 4.4.2. 模块间关系

### *e16ba222ce474eeab57a8b2dd67c84f*

### 4.4.3. 模块描述

#### 4.4.3.1.订单管理模块

1）功能描述

员工可以查看订单并且提前备餐，根据需要看是否进行配送

2）性能描述

员工命令的响应时间小于1.5秒，用户下单时间与员工收到订单时间差不超过3秒。

3）接口描述

软件接口：MYSQL数据库，及时传递用户订餐情况。

#### 4.4.3.2.菜品管理模块

1）功能描述

管理员提前确定当天菜品信息，包括菜品种类、价格、剩余数量，对菜品的增删改查等

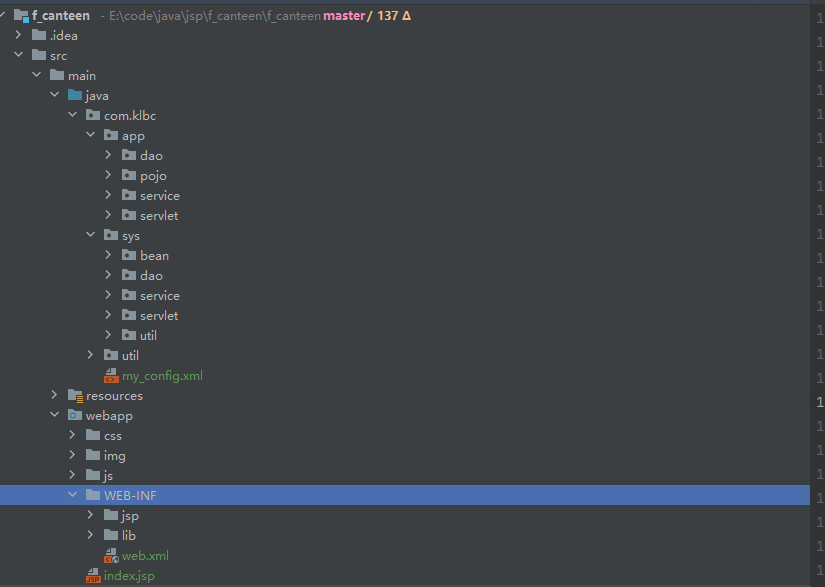
2）性能描述

命令的响应时间小于1.5秒。

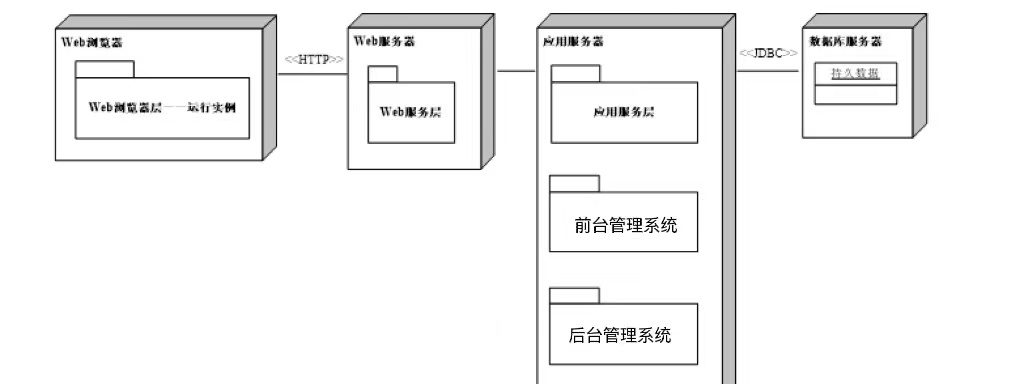
1. 接口描述

软件接口：MYSQL数据库，及时传递当天菜品情况。

# 5. 开发视图



# 6.物理视图



# 7.运行视图

