<交大点评>

软件架构文档

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 5.20.2023 | 1.0 | 制定软件架构 | 彭奕涛，陈一鸣，朱亦腾，雷宇昊 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

目录

[1. 简介 4](#_Toc135556220)

[1.1 目的 4](#_Toc135556221)

[2. 用例视图 4](#_Toc135556222)

[2.1 主要用例 4](#_Toc135556223)

[2.1.1 Login 登录 4](#_Toc135556224)

[2.1.2 Order food 点菜 4](#_Toc135556225)

[2.1.3 Comment on merchants 对商家进行评论 4](#_Toc135556226)

[2.1.4 View the catering list 查看餐饮清单 5](#_Toc135556227)

[2.1.5 Menu recommend 菜单推荐 5](#_Toc135556228)

[2.1.6 Provide information 提供信息 5](#_Toc135556229)

[2.1.7 Manage comments 管理评论 5](#_Toc135556230)

[2.1.8 Manage merchants info 管理商家信息 5](#_Toc135556231)

[3. 逻辑视图 5](#_Toc135556232)

[3.1 概述 5](#_Toc135556233)

[3.1.1 Application层（应用层） 5](#_Toc135556234)

[3.1.2 Business Service层（领域中间件） 6](#_Toc135556235)

[3.1.3 Middleware层（分布式中间件） 7](#_Toc135556236)

[3.2 在构架方面具有重要意义的设计包 7](#_Toc135556237)

[3.2.1 评论相关功能 7](#_Toc135556238)

[3.2.2 菜品上传功能 7](#_Toc135556239)

[3.2.3 菜品查询功能 8](#_Toc135556240)

[3.2.4 社交相关功能 8](#_Toc135556241)

[3.2.5 菜品推荐功能 8](#_Toc135556242)

[4. 部署视图 8](#_Toc135556243)

[4.1 概述 8](#_Toc135556244)

[4.1.1 web服务器 9](#_Toc135556245)

[4.1.2 application server 应用服务器 9](#_Toc135556246)

[4.1.3 database数据库 9](#_Toc135556247)

[4.1.4 user\_client用户端 9](#_Toc135556248)

[5. 实现视图 9](#_Toc135556249)

[5.1 展示层 9](#_Toc135556250)

[5.2 业务逻辑层 10](#_Toc135556251)

[5.3 数据访问层 10](#_Toc135556252)

[6. 技术视图 10](#_Toc135556253)

[6.1 编程语言 10](#_Toc135556254)

[6.2 开发工具 10](#_Toc135556255)

[6.3 框架 10](#_Toc135556256)

[6.4 数据库 10](#_Toc135556257)

[7. 核心算法设计 10](#_Toc135556258)

[7.1 概述 10](#_Toc135556259)

[7.1.1 当前热门推荐 10](#_Toc135556260)

[7.1.2 个性化推荐 10](#_Toc135556261)

[7.1.3 时下热促推荐 10](#_Toc135556262)

[7.2 核心算法实现 11](#_Toc135556263)

[7.2.1 数据收集和准备 11](#_Toc135556264)

[7.2.2 特征提取 11](#_Toc135556265)

[7.2.3 用户建模 11](#_Toc135556266)

[7.2.4 相似度计算 11](#_Toc135556267)

[7.2.5 推荐算法 11](#_Toc135556268)

[7.2.6 反馈和迭代 11](#_Toc135556269)

[8. 质量属性的设计 12](#_Toc135556270)

[8.1 易用性 12](#_Toc135556271)

[8.2 可靠性 12](#_Toc135556272)

[8.3 性能 12](#_Toc135556273)

[8.4 可支持性 12](#_Toc135556274)

软件架构文档

# 简介

## 目的

本文档将从构架方面对系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

# 用例视图

## 主要用例

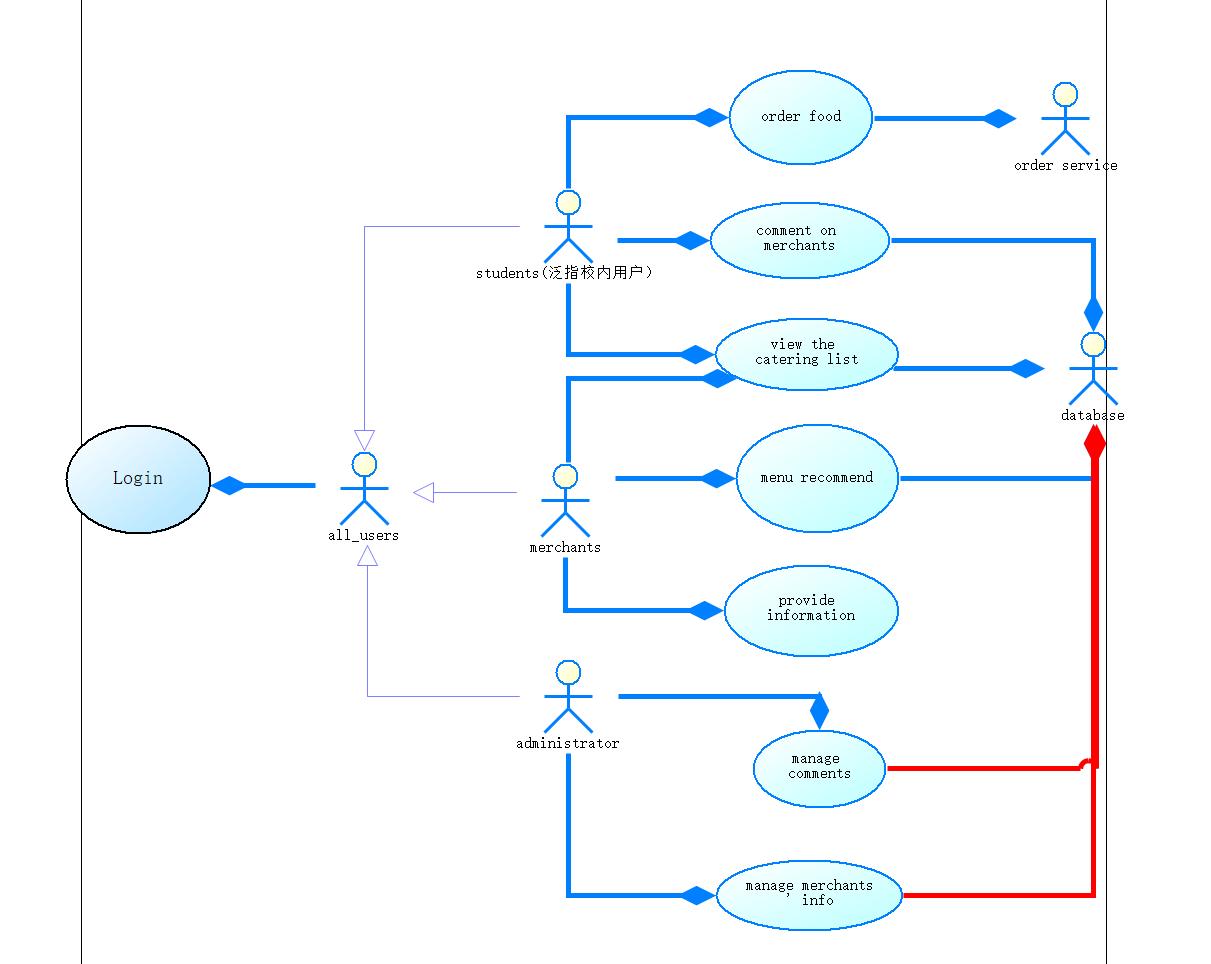


图2-1 交大点评用例图

### Login 登录

简要说明：该用例允许all\_users（一切用户）登录进入交大点评系统，根据账号种类登录为students（泛指校内用户），merchants（商家）和administrator（管理员）账号。该用例的主要执行者为一切用户。

### Order food 点菜

简要说明：该用例允许students（泛指校内用户）通过交大点评系统进行线上点餐，由order service提供服务。该用例的主要执行者为校内用户，次要执行者为order service。

### Comment on merchants 对商家进行评论

简要说明：该用例允许students（泛指校内用户）通过交大点评系统对商家进行评论，由data base提供服务。该用例的主要执行者为校内用户，次要执行者为data base。

### View the catering list 查看餐饮清单

简要说明：该用例允许students（泛指校内用户）和merchants（商家）通过交大点评系统查看餐饮清单，由data base提供服务。该用例的主要执行者为校内用户和商家，次要执行者为data base。

### Menu recommend 菜单推荐

简要说明：该用例允许merchants（商家）通过交大点评系统将菜单推荐到data base，由data base提供服务。该用例的主要执行者为商家，次要执行者为data base。

### Provide information 提供信息

简要说明：该用例允许merchants（商家）通过交大点评系统将为菜单提供详细的信息，由data base提供服务。该用例的主要执行者为商家，次要执行者为data base。

### Manage comments 管理评论

简要说明：该用例允许administrator（管理员）通过交大点评系统在社区对用户评论进行管理，由data base提供服务。该用例的主要执行者为管理员，次要执行者为data base。

### Manage merchants info 管理商家信息

简要说明：该用例允许administrator（管理员）通过交大点评系统在社区对商家信息进行管理，由data base提供服务。该用例的主要执行者为管理员，次要执行者为data base。

# 逻辑视图

## 概述

逻辑视图采用分层架构风格，主要由三层组成，分别是Application层（应用层），Business Service层（领域中间件）和Middleware层（分布式中间件）

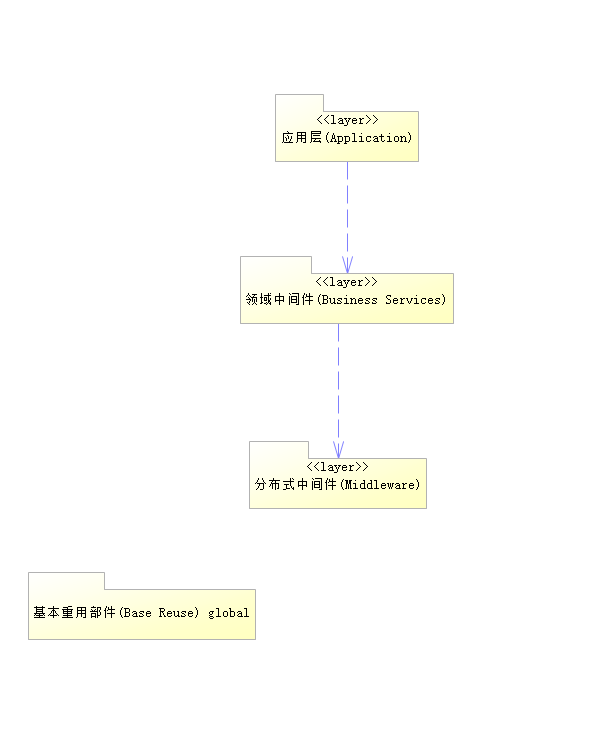


图3-1 系统总体架构图

### Application层（应用层）

应用层主要由5个子包组成，这5个子包分别是发布或回复评论，上传菜品信息，查询菜品信息，社交相关功能，菜品推荐。

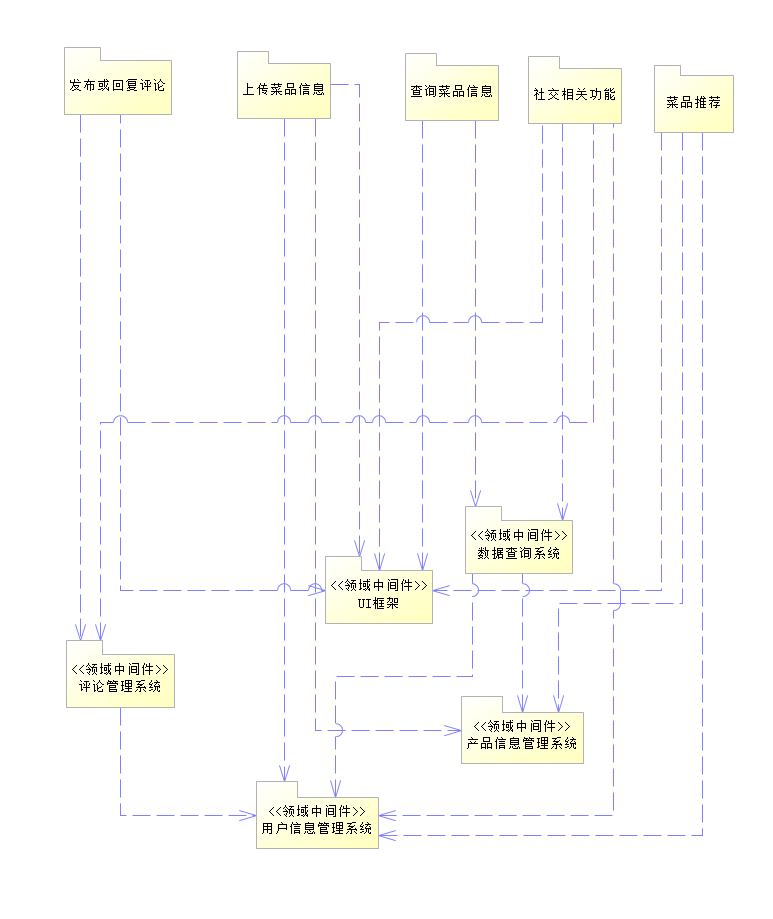


图4-2 Application层和Business Server层之间的详细架构

### Business Service层（领域中间件）

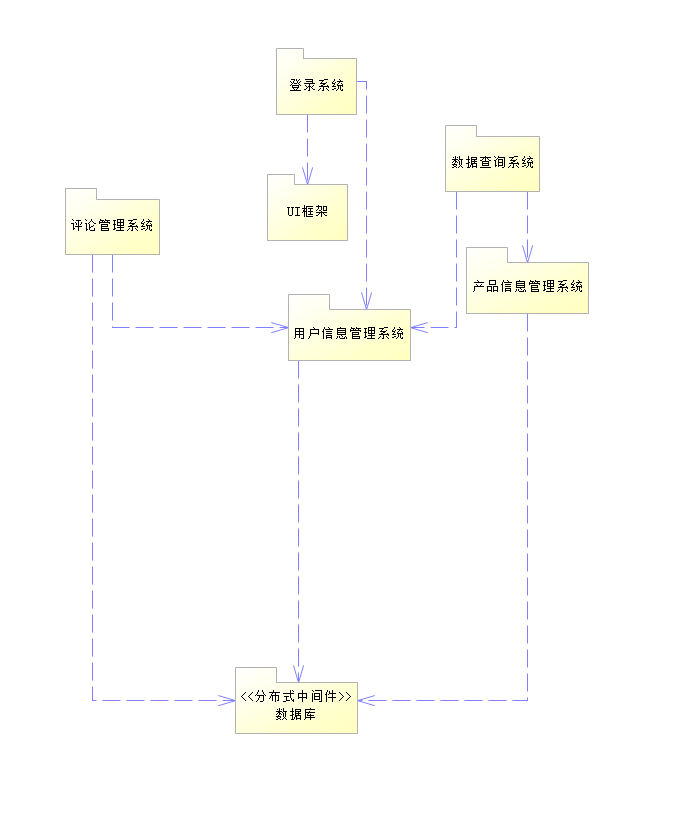
Business Service层主要由5个子包组成，这5个子包分别是UI框架，数据查询系统，评论管理系统，产品信息管理系统和用户信息管理系统。

图4-3 Business Server层和Middleware层之间的详细架构

### Middleware层（分布式中间件）

Middleware层由数据库1个子包组成。

## 在构架方面具有重要意义的设计包

### 评论相关功能

用户可以在UI框架中发布或回复评论，并将相关信息添加到评论管理系统中。评论管理系统负责存储所有评论，并为其他系统提供API接口，以便获取评论数据。当用户在UI框架上点击按钮时，UI框架会向评论管理系统发送请求，在评论管理系统中创建新的评论记录，并将评论内容、用户信息和时间戳等信息存储到数据库中。同时，评论管理系统会返回成功或失败的状态码给UI框架，用于更新界面上的操作提示。

### 菜品上传功能

用户可以在UI框架中上传自己的菜品信息，并将相关信息添加到用户信息管理系统和产品信息管理系统中。在UI框架中，用户需要填写菜品名、图片、描述、所需食材等信息，并在提交之前选择相应的分类。当用户点击上传按钮时，UI框架会向产品信息管理系统发送POST请求，添加新的产品信息到数据库中。同时，UI框架会将上传的菜品信息与用户信息匹配，并将用户ID和产品ID存储到用户信息管理系统中。这样，系统就可以识别用户上传的菜品，并将其与其他相关的数据进行关联。

### 菜品查询功能

用户可以通过UI框架查询自己或他人上传的菜品信息，并获得相关的详细信息。UI框架会向数据查询系统发送查询请求，数据查询系统会调用用户信息管理系统和产品信息管理系统的API接口，从数据库中获取相应的信息，并将结果返回给UI框架进行显示。在查询结果中，用户可以看到菜品的图片、描述、所需食材、上传时间以及上传者的信息等详细信息。

### 社交相关功能

这个功能主要是为了增强用户之间的互动性。在UI框架上，用户可以查看自己或他人的菜品评论，并回复或点赞。当用户点击评论按钮时，UI框架会向评论管理系统发送请求，获取该菜品的所有评论。同时，UI框架也会调用用户信息管理系统和产品信息管理系统的API接口，获取相关的用户和产品信息。当用户提交评论或点赞时，UI框架会将相关信息发送给评论管理系统，更新评论记录，并返回成功或失败的状态码给UI框架。

### 菜品推荐功能

这个功能是为了提供更好的个性化服务，为用户推荐可能感兴趣的菜品。在UI框架中，菜品推荐模块会根据用户的浏览历史和上传历史，从产品信息管理系统中获取相关的菜品信息，并通过算法对其进行推荐排名。最终，排名前几位的菜品会被展示给用户，以供选择。这个功能需要涉及到大量的数据处理和算法优化，因此需要专业的团队进行研发和实现。

# 部署视图

## 概述

本部分描述了架构的物理视图，说明了最典型平台配置的各种物理节点。同时还描述了任务从流程视图分配到物理节点的情况。

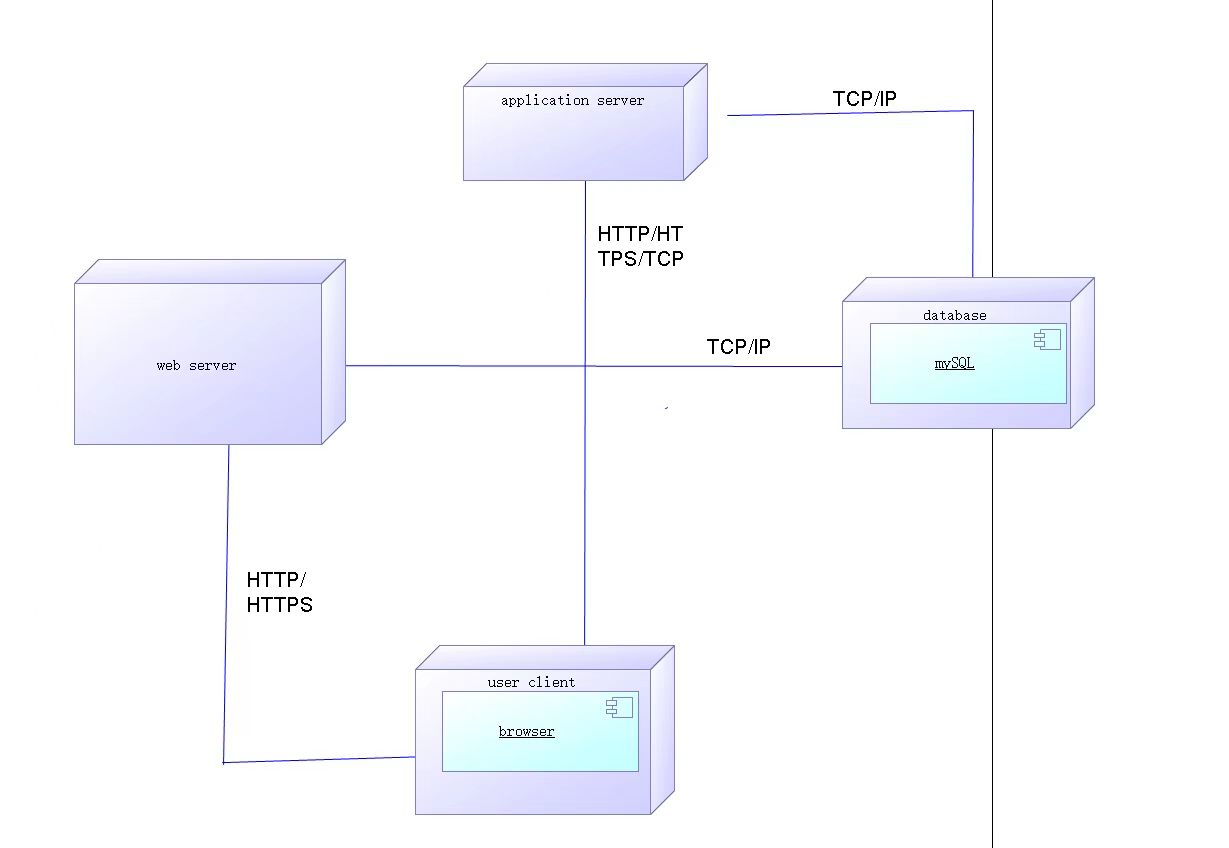


图4-1 部署视图

### web服务器

Web服务器主要用来处理静态页面，静态页面指的是服务器上已经存在的、不需要额外处理的HTML、CSS、JavaScript等文件，这些文件的内容在用户访问时是不会发生变化的，所以只需要简单地读取服务器上的文件并将其返回给浏览器即可。

### application server 应用服务器

应用服务器通常用于处理各种服务请求，包括但不限于处理动态页面、执行业务逻辑、连接数据库、处理事务等。

与Web服务器只处理静态页面不同，应用服务器能够动态地生成内容并在服务器端执行代码逻辑。当客户端发起请求时，应用服务器会根据请求的内容进行相关的处理，并生成相应的动态页面或其他响应。应用服务器的处理能力和扩展性通常比Web服务器更强，因此它们更适合处理需要进行大量计算和处理的动态页面。

除了处理动态页面外，应用服务器还可以用来执行业务逻辑和连接数据库。例如，在交大点评中，应用服务器可以执行上传、查询菜品信息，发布评论和社区管理等业务逻辑。同时，应用服务器还可以连接到数据库，从而实现对数据的读取和写入，以及处理复杂的事务操作。

### database数据库

数据库被用来储存各类菜品信息、用户数据及评论等重要数据。

对于菜品信息，数据库中通常会存储菜品的名称、描述、价格、图片等相关信息。这些信息可以被用来在网站上进行展示。

对于用户数据，数据库中通常会存储用户的基本信息，如用户名、密码等。此外，一些网站还会存储用户的历史信息和个人偏好等数据，以便提供更个性化的服务和推荐。

对于评论数据，数据库中通常会存储用户对菜品的评价、评分和评论时间等相关信息。这些数据可以用来帮助其他用户做出更好的点餐决策，同时也可以帮助餐厅进行品牌营销和服务改进。

### user\_client用户端

用户端指的是通过浏览器访问网站的用户界面。用户可以通过浏览器进行评论、搜索等操作并与网站进行交互。同时，用户端还可以与服务器进行其他类型的交互，例如上传图片等文件、使用地理位置服务等。

# 实现视图

## 展示层

展示层主要负责与用户交互和呈现数据，主要为项目的前端部分，主要使用的框架为React。展示层需要向用户展示首页、菜品详情页等页面及相关的图片与文字信息。展示层还负责与用户的交互，包括通过页面表单、按钮等组件收集和返回用户输入的菜品信息或评论等数据，并根据用户的操作实时更新页面显示内容。

## 业务逻辑层

业务逻辑层负责处理业务逻辑，包括对菜品信息的管理和处理，以及对用户评价的统计和分析等操作。该层中主要使用Java语言和Spring Boot框架来开发相关的后端服务。业务逻辑层负责对菜品信息进行管理和处理，包括菜品的添加、删除、修改等操作。业务逻辑层还需要进行数据校验和验证，确保数据的安全性和有效性。业务逻辑层也需要对用户评价进行统计和分析，例如计算菜品的平均分数、收集用户的建议和意见等。在此之外，业务逻辑层还需要进行安全控制和权限管理，确保系统的稳定性和可靠性。例如，对管理员和普通用户进行身份验证和授权，限制不合法的操作。

## 数据访问层

数据访问层负责与数据库进行交互，包括对菜品信息进行增删改查操作，并为业务逻辑层提供数据访问接口。该层所需的数据库在团队成员实际开发过程中主要是MySQL数据库，并使用Spring JPA。数据访问层需要进行数据库连接和配置，确保与数据库的正常交互。数据访问层还需要对数据库中的菜品信息进行增删改查等操作，并根据业务逻辑层的需要提供相应的数据访问接口。

# 技术视图

## 编程语言

编程语言采用Java（后端）和JavaScript（前端）。

## 开发工具

开发工具采用IntelliJ。

## 框架

框架采用Spring Boot（后端）和React（前端）。

## 数据库

数据库采用MySQL。

# 核心算法设计

## 概述

推荐算法为交大点评系统中的核心算法，通过推荐算法，能够做到当前热门推荐、个性化推荐、时下热促推荐等功能。

### 当前热门推荐

后台获取当下好评率/浏览量最高的菜品降序排列进行推荐

### 个性化推荐

通过计算各菜品特征向量与用户特征向量的余弦值选择最优值

### 时下热促推荐

通过后台获取优惠度最大的菜品，降序排列进行推荐

## 核心算法实现

### 数据收集和准备

收集菜品相关的数据，包括价格、菜品评价、用户喜好等。（可利用文本挖掘技术辅助收集）

### 特征提取

从数据中提取特征，暂定为：菜品分类，如面食，炒饭，麻辣烫等；口味，如辣，甜，清淡等；价格；油腻度；位置；营养价值；饱腹程度；

对每个数据进行标准化处理，整理成0-100的double变量。

-为每个菜品构建特征向量表示。Double[ ] feature

### 用户建模

收集用户的历史行为数据，例如浏览菜品、点赞、收藏等。

根据用户行为构建用户兴趣模型。(通过记录，以浏览、点赞、收藏分别为0.2，0.4，0.4的权重计算用户的个人爱好特征向量)

-譬如，可遍历将用户的所有浏览数据（即上次处理模型之后新增的数据），对其中菜品的的各类特征向量取平均，以0.2的权重与其他两项结合后，再以某一权重与原模型整合。

记录用户的特别诉求，如价格，不得吃辣，清真需求等，在相似度计算中加以修正

### 相似度计算

计算菜品之间的相似度，可以使用余弦相似度（计算方法：

记x，y分别为用户，当前菜品的特征向量，则

、欧氏距离（可选，可利用kd树加速查找）。

可考虑使用加权方式对不同特征进行相似度计算，根据不同用户偏好进行加权，如某用户对价格因素特别重视，则可对价格向量进行两次计算等

### 推荐算法

基于用户的兴趣模型和菜品的相似度，为用户生成个性化推荐。

（由于数据量小，可对所有菜品进行相似度计算，选取最大的几个作为推荐菜品）

当前采用推荐相似度最高的菜品为推荐算法，之后可加入基于用户的协同过滤推荐算法（通过计算不同用户对菜品的兴趣，得到相似度最高的一个或多个用户，若相似用户点赞或收藏某菜品，则将该菜品推荐给该用户）等

### 反馈和迭代

收集用户的反馈数据，例如点击、购买、评价等，将其评价修正至用户的兴趣模型中

-根据用户反馈不断优化算法，提高推荐的准确性和个性化程度。

# 质量属性的设计

本系统采用的软件架构可以很好支持系统质量方面的需求：

## 易用性

用户无需教程即可自行使用网页或软件上传评论内容或进行其他操作

## 可靠性

网站需使用jAccount登录，用户需对自己的言论负责，保证信息准确性，网站整体结构较为简单，易于维护，保证整体稳定性。

## 性能

支持8GB业务数据，1500用户在线，300个并发用户运行的负载下，连续运行72小时，带宽5M，吞吐量50000TPS，网页精确度90%以上，响应时间在5秒以内。

## 可支持性

支持Windows,MacOS网页版、微信小程序以及手机APP使用