

三饭订餐系统架构设计文档

1. 引言

该文档描述了三饭订餐系统的整体架构和关键设计决策。以下是该系统的架构设计。



2. 系统架构

我们的三饭订餐系统采用分层体系结构。系统包括以下主要层次：

(1) 表现层 (Presentation Layer)

表现层主要包括用户界面和商家界面，用户和商家通过Web界面和系统交互。

- 用户能够通过界面注册、登录、浏览饭堂窗口和菜单、下订单以及查看自己历史订单
- 商家能够通过界面注册、登录、查看自己窗口的未完成和历史订单、完成订单

(2) 应用层 (Application Layer)

该层处理用户请求并协调系统的各个部分。它包含业务逻辑，负责接收来自表现层的请求，进行验证和处理，然后将结果传递给下一层。

- 用户管理模块：处理用户注册、登录、验证和用户信息管理的功能。
- 商家管理模块：负责商家注册、登录、验证和商家信息管理功能。
- 订单管理模块：处理订单的创建、支付、取消以及订单状态的更新。

- 菜单管理模块：负责菜单的展示和菜品详情的查询

(3) 领域层 (Domain Layer)

领域层包含系统的核心业务逻辑和实体。它独立于具体的技术实现，定义了系统的业务规则和操作。

- 用户：包含用户的属性和行为，提供用户相关的业务逻辑，主要是验证密码是否正确
- 商家：包含具体窗口的各种属性，包括名称、位置、菜单等；提供相关业务逻辑，比如修改菜单
- 订单：包含订单号，订单的商家和用户等等；提供相关业务逻辑，如修改订单状态
- 饭菜：包含具体饭菜的名称、价格等属性

(4) 数据访问层 (Data Access Layer)

该层负责与数据存储进行交互，包括数据库或其他持久化机制。它提供对数据的读取和写入操作，并将数据转换为领域层可以理解的格式。

3. 关键设计决策

在三饭订餐系统的设计中，我们做出了以下关键设计决策：

- 前端web界面采用vue3框架编写。Vue 3提供了强大的响应式数据绑定能力，使得界面与数据之间的同步更新变得简单而高效。
- 后端使用java语言编写。
- 持久化方面，选择将领域层的多个实体序列化为json文件保存到磁盘，而不是使用数据库。原因有两方面：
 - 数据库需要进行表设计、模式定义和查询语言学习等工作，相对而言较为复杂。而使用序列化保存数据，可以更加直接地将数据对象序列化为文件或其他格式，并进行读写操作，简化了数据的处理过程，减少了开发的复杂性和学习成本
 - 使用数据库如mysql作为持久化的方法，容易引起各种安全问题，如sql注入攻击等