## **一、总体设计**

### 1. 系统目标

“外卖管理系统”旨在为用户提供方便快捷的在线外卖订餐服务，实现从餐厅选择、下单支付到订单配送的全流程数字化管理。同时，系统支持多角色（用户、商家、配送员、管理员）的高效协作。

### 2. 系统架构设计

采用 **B/S架构（浏览器/服务器架构）**：

* **前端**：使用 HTML、CSS、JavaScript，结合 Vue.js 框架实现交互式用户界面。
* **后端**：采用 PHP 或 Python（Django/Flask）开发服务器端逻辑。
* **数据库**：使用 MySQL 或 PostgreSQL 存储和管理数据。
* **API**：通过 RESTful API 实现前后端数据交互。
* **部署**：部署于云服务器，使用 Nginx 或 Apache 配合负载均衡。

### 3. 系统模块划分

按照功能划分为以下模块：

1. 用户管理模块：
   * 用户注册、登录、注销功能。
   * 用户资料修改和权限管理。
2. 餐厅管理模块：
   * 餐厅信息的展示和搜索。
   * 菜品的管理（添加、删除、修改）。
3. 订单管理模块：
   * 下单、支付和配送流程管理。
   * 订单状态的实时更新。
4. 支付模块：
   * 在线支付功能（集成支付宝、微信等支付方式）。
   * 支持退款处理。
5. 评价管理模块：
   * 用户对订单和菜品进行评价。
   * 展示评价和评分统计。
6. 后台管理模块：
   * 管理员对系统中用户、商家和配送员的管理。
7. 配送管理模块：
   * 配送员接收并处理订单任务。
   * 配送状态的实时跟踪。

### 4. 技术框架

* **前端**：Vue 3、Axios、Bootstrap
* **后端**：Laravel（PHP）/ Django（Python）
* **数据库**：MySQL、Redis（缓存机制）
* **其他工具**：Git（版本控制）、Postman（API测试）

## **二、概要设计**

### 1. 系统数据流图（DFD）

**层次0：**

* 用户与系统交互：用户输入需求，系统返回餐厅、菜品、订单等信息。

**层次1：**

* 用户数据流：用户注册、登录信息流转到用户管理模块。
* 餐厅数据流：餐厅和菜品信息流转到餐厅管理模块。
* 订单数据流：订单生成后流转到订单管理模块，再经支付模块完成支付。
* 配送数据流：订单分配给配送员，由配送管理模块更新配送状态。

### 2. 数据库设计

| **表名** |
| --- |
| User |
| Restaurant |
| MenuItem |
| Order |
| Payment |
| Delivery |
| Review |

### 3. 功能流程设计

**示例：下单流程**

1. 用户从餐厅列表中选择餐厅。
2. 查看菜品，加入购物车。
3. 确认订单，填写配送地址。
4. 选择支付方式完成支付。
5. 系统通知商家接单，生成配送任务。
6. 配送员接单并配送，更新订单状态。

### 4. 界面原型设计（概要）

* 用户界面：
  + 首页：显示推荐餐厅和菜品。
  + 菜品页面：餐厅信息和菜品列表。
  + 订单页面：订单详情、评价入口。
* 商家界面：
  + 菜品管理：添加、编辑、删除菜品。
  + 订单管理：查看用户订单并确认接单。
* 配送员界面：
  + 待配送订单：显示当前需要配送的订单。
  + 配送完成按钮：完成配送后更新状态。
* 管理员界面：
  + 用户和商家账号管理。
  + 系统日志查看。