

分工文档

一、分工说明

二、总经理

三、项目经理

四、程序员 1（前端 - 前台）

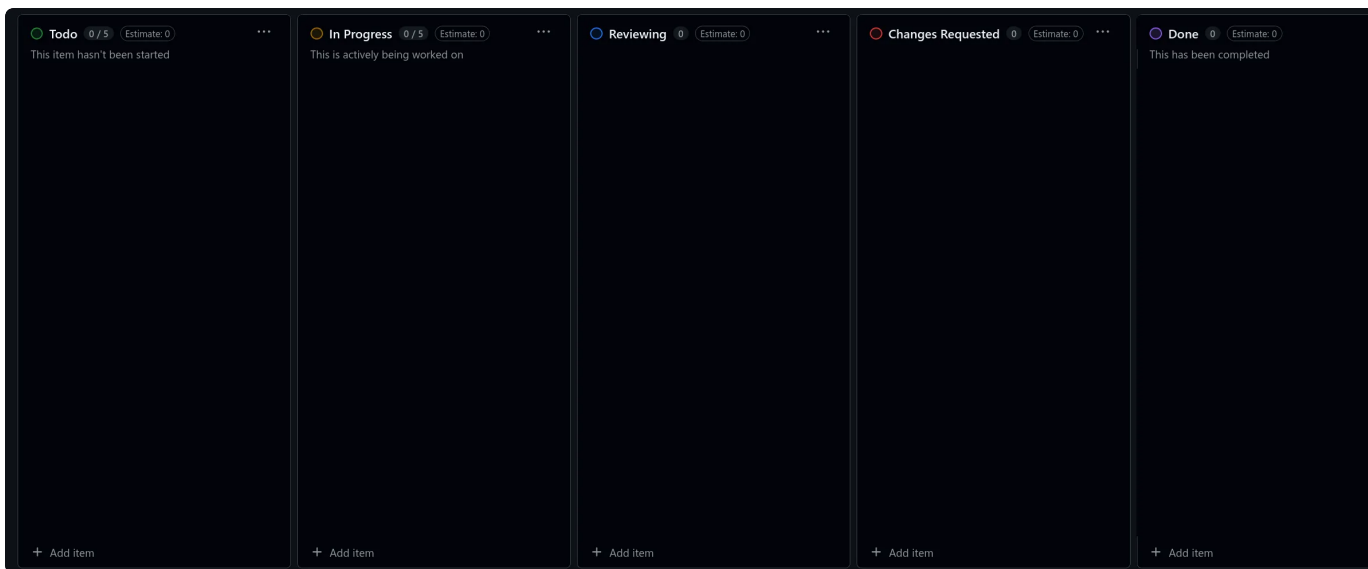
五、程序员 2（前端 - 后台）

六、程序员 3（后端 - 前台业务）

七、程序员 4（后端 - 后台业务）

一、分工说明

1. **依据**：仅基于《Web 订餐管理系统.docx》（以下简称“文档”）需求，覆盖“前台普通用户模块、后台管理员模块、MySQL 数据库设计、Vue3/Java/MySQL/Tomcat 技术栈”核心内容，不涉及文档外功能或额外技术细节；
2. **协作方式**：通过 GitHub 共同维护项目仓库，使用 GitHub Project 功能跟踪任务进度（标注“待开发 Todo - 开发中 In Progress - 待审核 Reviewing - 待修复 Changes Requested - 已完成 Done”状态），所有开发成果通过提交 PR（Pull Request）说明进度，PR 审核通过后合并至主分支；
3. **交付物存储**：文档类交付物统一存放于主分支docs目录，代码类交付物按“前端 / 后端”目录结构提交至仓库对应路径，不额外新增文档外的交付物要求；
4. **API 文档生成**：接口文档通过 Swagger/OpenAPI 自动生成，生成后同步至docs目录，无需手写接口文档。



二、总经理

- 核心定位：技术架构管控 + GitHub 仓库统筹 + 进度跟踪
- 核心开发任务（对应文档模块）：

1. 后端核心架构搭建（文档技术栈要求）：配置 Java/J2EE 项目（IDEA 环境），集成 SpringMVC/Servlet，搭建 B/S 结构后端框架；配置 Tomcat 服务器，确保前后端可联调；搭建 MySQL 数据库连接池，保障文档提及的 5 张核心表稳定访问。
2. 全局权限框架开发（文档 2.4 节“权限区分”）：开发权限拦截器，实现“管理员可访问后台、会员仅访问前台、接单员仅操作订单受理”的权限控制，确保符合文档中不同角色的操作范围要求。
3. Swagger/OpenAPI 配置（接口文档生成）：在后端项目中集成 Swagger/OpenAPI，确保自动生成的 API 文档覆盖前台、后台所有业务接口，且符合文档功能描述。

- 协作与管控职责：

1. 初始化 GitHub 项目仓库，划分“前端 / 后端 / 文档”基础目录，配置 PR 审核规则；
2. 通过 GitHub Project 更新任务进度，每日审核成员 PR，重点检查代码是否符合架构规范、权限是否适配文档要求；
3. 在 GitHub Issues 中响应技术问题（如环境配置、联调阻塞），及时提供解决方案；
4. 输出基础技术规范（含前后端开发统一标准），提交至 docs 目录。

- 关键交付物：

1. 后端架构代码（含权限拦截器、数据库连接池配置）；
2. Swagger/OpenAPI 配置代码；
3. 技术规范文档（存docs目录）；
4. Tomcat 配置说明（存docs目录）。

三、项目经理

- 核心定位：需求落地 + 功能验收标准定义 + 数据库设计
- 核心开发任务（对应文档模块）：

1. 数据库设计与脚本开发（文档 5.4 节“5 张数据表”）：按文档要求编写 5 张表的 SQL 创建脚本，编写测试数据初始化脚本（如管理员账号、基础菜品分类），确保符合文档中表的核心用途。
2. 功能验收标准梳理（文档全章节核心功能）：按“前台 / 后台”分类编写功能验收标准，明确每个功能的正确表现（需关联文档对应章节或效果图），确保开发成果符合文档描述。
3. Swagger 接口标注审核（接口文档准确性）：审核后端成员添加的 Swagger 注解，确保自动生成的 API 文档与文档需求一致，无遗漏核心功能接口。

- 协作与管控职责：

1. 在 GitHubdocs目录提交需求拆解文档，将文档需求拆分为可执行小任务，并关联至 GitHub Project；
2. 作为 PR 审核人，对照验收标准验证成员提交的功能是否符合文档需求，在 PR 评论区提出修改意见；
3. 同步 API 文档更新至 GitHub Project，提醒前端成员基于最新接口文档开发。

- 关键交付物：

1. 数据库 SQL 脚本（含测试数据，存docs目录）；
2. 功能验收标准文档（存docs目录）；
3. 需求拆解文档（存docs目录）。

四、程序员 1（前端 – 前台）

- 核心定位：文档 6.3 节“普通用户前台模块”全功能开发
- 核心开发任务（对应文档模块）：

1. 前台主页面开发（文档 6.3.1 节 + 对应效果图）：用 Vue3 开发前台主页面，实现“菜单分类展示、菜品卡片、联系我们”功能，符合文档效果图布局要求。
2. 会员注册 / 登录页开发（文档 6.3.2 节 + 对应效果图）：开发注册、登录表单，实现表单校验（符合文档隐含的字段规则），调用后端接口完成注册、登录逻辑，处理接口返回结果。
3. 菜品详情 + 点餐单管理开发（文档 6.3.4 节 + 6.3.5 节 + 对应效果图）：开发菜品详情页（展示菜品信息、点餐按钮），开发点餐单管理页（实现菜品添加 / 删除 / 数量修改、总金额计算），符合文档功能描述。

- 协作与管控职责：

1. 在 GitHub Project 认领“前台模块”任务，更新任务状态；
2. 完成开发后提交 PR 至前端目录，在 PR 描述中说明“已完功能、测试情况、依赖的后端接口”；
3. 若接口调用出现问题，在 GitHub Issues 中 @对应后端成员，协作解决联调问题；
4. 根据 PR 审核意见优化代码，直至合并。

- 关键交付物：

1. 前台 Vue3 组件代码（存前端目录）；
2. 前台表单校验、接口调用逻辑代码（存前端目录）。

五、程序员 2（前端 – 后台）

- 核心定位：文档 6.2 节“管理员后台模块”全功能开发
- 核心开发任务（对应文档模块）：

1. 管理员登录页开发（文档 6.2 节 + 对应效果图）：用 Vue3 开发后台登录页，实现 AJAX 异步验证、登录结果处理，符合文档效果图布局。
2. 菜单管理页面开发（文档 6.2.1 节 + 6.2.2 节 + 对应效果图）：开发菜单类别管理页（实现类别增删、列表展示）、菜单信息管理页（实现菜品录入、列表展示、删除），符合文档功能描述。
3. 会员 + 订单 + 密码修改页面开发（文档 6.2.3 节 + 6.2.4 节 + 6.2.5 节 + 对应效果图）：开发会员管理页（会员列表、删除）、订单管理页（订单列表、明细查看、受理）、密码修改页（表单校验、密码更新），符合文档功能要求。

- 协作与管控职责：

1. 在 GitHub Project 认领“后台模块”任务，更新任务状态；
2. 完成开发后提交 PR 至前端后台目录，在 PR 描述中说明“已完成功能、测试情况”；
3. 若后端接口未就绪，在 GitHub Issues 中 @对应后端成员，说明接口依赖需求；
4. 根据 PR 审核意见优化代码，直至合并。

- 关键交付物：

1. 后台 Vue3 组件代码（存前端后台目录）；
2. 后台表单校验、接口调用逻辑代码（存前端后台目录）。

六、程序员 3（后端 – 前台业务）

- 核心定位：文档 6.3 节“普通用户前台模块”后端业务接口开发
- 核心开发任务（对应文档模块）：

1. 会员业务接口开发（文档 6.3.2 节会员注册 / 登录）：开发会员注册、登录接口，实现参数校验、数据存储 / 查询，添加 Swagger 注解确保接口文档准确。
2. 菜品业务接口开发（文档 6.3.1 节菜品展示、6.3.4 节菜品详情）：开发菜品分类查询、菜品列表查询、菜品详情接口，实现数据关联查询，满足前台页面数据需求。
3. 购物车 + 订单业务接口开发（文档 6.3.5 节点餐单、订单提交）：开发购物车（添加 / 修改 / 删除）、订单提交接口，实现数据缓存 / 存储、订单生成逻辑，符合文档业务流程。

- 协作与管控职责：

1. 在 GitHub Project 认领“前台后端接口”任务，更新任务状态；
2. 完成接口开发后提交 PR 至后端目录，在 PR 描述中说明“接口功能、测试方法”；
3. 配合前台成员联调，在 GitHub Issues 中响应接口问题，及时排查解决；
4. 根据 PR 审核意见优化接口逻辑，直至合并。

- **关键交付物：**

1. 前台业务接口代码（存后端目录）；
2. 接口测试报告（存docs目录）。

七、程序员 4（后端 – 后台业务）

- **核心定位：**文档 6.2 节“管理员后台模块”后端业务接口开发
- **核心开发任务（对应文档模块）：**

1. **管理员登录 + 权限接口开发（文档 6.2 节后台登录、2.4 节权限区分）：**开发管理员登录、权限校验接口，实现管理员身份识别、权限控制，添加 Swagger 注解。
2. **菜单业务接口开发（文档 6.2.1 节菜单类别管理、6.2.2 节菜单信息管理）：**开发菜单类别（增删查）、菜品（增删查）接口，仅允许管理员调用，满足后台菜单管理需求。
3. **会员 + 订单 + 密码修改接口开发（文档 6.2.3 节会员管理、6.2.4 节订单管理、6.2.5 节修改密码）：**开发会员（查询 / 删除）、订单（查询 / 明细 / 受理）、密码修改接口，实现数据操作、状态更新，符合文档后台业务流程。

- **协作与管控职责：**

1. 在 GitHub Project 认领“后台后端接口”任务，更新任务状态；
2. 完成接口开发后提交 PR 至后端目录，在 PR 描述中说明“接口功能、权限控制逻辑”；
3. 配合后台前台成员联调，在 GitHub Issues 中响应接口问题，及时解决；
4. 接口开发完成后，触发 Swagger 重新生成 API 文档并同步至docs目录。

- **关键交付物：**

1. 后台业务接口代码（存后端目录）；
2. 接口权限校验逻辑代码（存后端目录）；
3. 后台接口测试报告（存docs目录）。

