

第一章 系统概要设计

1 系统名称与课题名称

本系统名称为“Chard 真货电商平台”，课题名称为“基于 Flask 的多角色电商网站设计与实现”。系统面向普通用户、商家和管理员三类角色构建，围绕商品展示、在线交易、订单履约、店铺经营和平台治理等业务需求展开设计。系统采用 Flask 作为后端框架，使用 SQL Server 作为业务数据库，并结合 Firebase 完成邮箱认证链路，以形成较完整的 Web 电商应用体系。

从课题定位上看，该系统并非单一页面展示程序，也不是仅实现局部功能的演示性网站，而是一个包含前台商城、商家后台、管理员后台以及后台状态任务在内的综合性 Web 应用。通过该系统的设计与实现，可以较系统地展示 Web 应用开发中常见的页面组织方式、角色权限划分方式、业务状态流转方式、数据存储方式以及部署运行方式。

2 系统开发背景

随着互联网应用场景的不断扩展，电子商务系统已经成为 Web 开发领域中最具代表性的业务类型之一。与传统静态网站相比，电商网站不仅需要承担商品信息展示功能，还需要处理用户注册登录、角色权限控制、购物车暂存、订单提交、支付结算、店铺经营管理和后台平台治理等复杂业务流程。对于课程设计和综合性项目而言，电商系统具有角色清晰、流程完整、功能层次丰富等特点，能够较好地体现系统分析、数据库设计、后端逻辑组织和页面交互设计等多个方面的综合能力。

在实际应用环境中，一个可用的电商平台通常并非只服务于单一用户群体，而是同时面向消费者、商家和平台管理者运行。普通用户需要浏览商品并完成购买，商家需要管理商品和处理订单，管理员需要进行审核、配置和治理。因此，系统在设计时不仅要实现前台交易链路，还必须考虑多角色协同、数据一致性、页面权限边界以及运行安全等问题。

基于上述背景，本课题选择以多角色电商平台为对象进行系统设计与实现。通过构建商品展示、购物车、订单支付、商家入驻、后台审核、店铺主页管理、站点品牌配置以及后台状态任务等模块，形成一个具备基本业务闭环的电商网站系统。同时，在系统建设过程中，还将认证、权限控制、请求限流、文件上传校验、富文本清洗和统一错误响应等基础安全策略纳入实现范围，以提高系统的工程完整性和实际可用性。

3 系统主要目标

本系统的建设目标主要体现在以下几个方面。

3.1 实现完整的电商交易主流程

系统需要支持普通用户完成从商品浏览到订单完成的完整交易流程，包括商品查看、商品搜索、加入购物车、提交订单、订单支付、订单查看和确认收货等操作。通过建立完整的交易主流程，使系统具备电子商务网站最基本的业务能力，并为后续订单状态推进和资金流转提供基础。

3.2 建立多角色协同运行机制

系统需要在统一平台下支持普通用户、商家和管理员三类角色运行，不同角色拥有不同的页面入口、功能权限和业务职责。通过角色划分和权限隔离，形成前台交易、店铺经营和后台治理相互配合的运行模式，从而提高系统的业务完整性和结构清晰度。

3.3 提供商家经营管理能力

系统需要支持商家入驻申请、商品管理、订单处理和店铺主页维护等功能，使平台不仅能够支撑用户购买行为，也能够支持商家在系统内开展基本经营活动。通过构建商家后台模块，使系统具备平台型网站的基本特征。

3.4 提供管理员平台治理能力

系统需要支持管理员对平台进行统一管理，包括商家申请审核、站点信息配置、用户钱包调账以及后台数据概览等功能。管理员模块的设计使平台具备审核、维护和治理能力，是多角色平台结构中不可缺少的组成部分。

3.5 构建基础安全与运行保障机制

系统除实现业务功能外，还需要具备基础安全能力和运行保障能力。主要包括登录认证、角色权限控制、CSRF 防护、安全响应头、请求限流、上传校验、富文本清洗、统一错误响应以及后台状态任务等内容。通过这些机制，确保系统在实现业务功能的同时具备较好的运行边界和维护基础。

4 系统角色划分

系统按照业务职责将使用者划分为普通用户、商家和管理员三类角色。不同角色通过统一认证体系进入系统，并在后端根据角色信息分配相应的访问权限和功能范围。

4.1 普通用户

普通用户是系统前台交易流程的主要参与者，其核心职责是完成商品浏览和购买行为。普通用户进入系统后，可以浏览首页、商品列表和商品详情页，使用搜索功能查找商品，将商品加入购物车，选择收货地址并提交订单。在订单生成后，普通用户可以执行订单支付、查看订单详情、确认收货、查看钱包余额与交易流水，并对个人资料和地址信息进行维护。

普通用户角色反映了系统前台交易链路是否完整，是验证系统基本业务能力的重要角色。

4.2 商家

商家角色主要承担店铺经营和订单履约职责。普通用户在提交商家申请并经管理员审核通过后，可获得商家角色。商家进入后台后，可以进行商品新增、商品编辑、商品上下架、商品图片上传、富文本内容维护、订单查看与发货、店铺主页编辑、店铺简介维护和店铺 Logo 管理等操作。

商家角色的存在使系统具备经营侧能力，使平台不再只是商品展示和用户下单的前台网站，而成为同时支持交易与经营的电商平台。

4.3 管理员

管理员角色负责平台级别的维护与治理。管理员可以进入后台查看平台总体情况，审核商家入驻申请，修改网站名称、首页头部标题、浏览器标题和网站图标等品牌信

息，并根据需要对用户钱包执行调账操作。管理员并不直接参与商品购买和店铺经营，而是对平台的整体运行状态和规则执行情况进行维护。

5 系统总体功能模块划分

根据系统业务结构，可以将系统整体划分为以下功能模块。

1. **用户认证与账号模块：**负责用户注册、登录、邮箱验证、登录态恢复、退出登录和本地兼容认证。
2. **前台首页与商品展示模块：**负责首页展示、热门商品展示、推荐店铺展示、商品列表、商品详情和搜索功能。
3. **购物车与结算模块：**负责商品加入购物车、购物车数量调整、删除购物车项、选择收货地址和提交订单。
4. **订单与支付模块：**负责订单列表、订单详情、单订单支付、批量支付、订单状态推进和确认收货。
5. **钱包与地址管理模块：**负责用户钱包余额与流水展示，以及收货地址的新增、设置默认地址和删除操作。
6. **商家入驻与审核模块：**负责普通用户提交商家申请，以及管理员对申请进行审核处理。
7. **商家后台经营模块：**负责商品管理、订单发货、富文本编辑、店铺主页维护和店铺资料管理。
8. **管理员后台治理模块：**负责平台概况展示、站点品牌设置、商家申请审核和钱包调账等治理功能。
9. **店铺展示模块：**负责公开店铺主页以及全部店铺页面的展示，为用户提供以店铺为单位的浏览入口。
10. **安全防护与异常处理模块：**负责权限控制、CSRF 防护、请求限流、上传校验、富文本安全清洗和统一错误响应。
11. **后台任务与状态轮询模块：**负责订单状态自动检查、超时处理与后台 worker 调度。

上述模块共同构成了系统的总体功能结构。其中，用户认证、商品展示、购物车、订单和钱包模块构成前台交易链路；商家入驻、商家后台和店铺展示模块构成经营链路；管理员后台治理模块承担平台管理功能；安全防护与后台任务模块则作为系统运行保障层，为整体业务提供支撑。

6 系统总体架构设计

从系统运行方式上看，本系统采用分层结构设计。前端页面层主要由 Flask 路由与 Jinja2 模板组成，用于向用户、商家和管理员提供对应页面；业务逻辑层由 `app.py` 中的路由、辅助函数和状态处理逻辑构成，用于完成订单、支付、角色控制、店铺管理和后台治理等核心业务；数据存储层采用 SQL Server，用于存放用户、角色、商家、店铺、商品、购物车、订单、地址、钱包和上传资源关联信息。

在线上部署环境中，系统采用 Nginx、Gunicorn 与 Flask 的组合运行方式。Nginx 负责处理外部 HTTP/HTTPS 请求并执行反向代理，Gunicorn 负责托管 Flask 应用进程，Flask 完成业务逻辑处理和模板渲染。用户上传的头像、商品图片、店铺 Logo 和站点图标统一保存在上传目录中，并通过统一路径对外提供访问。除此之外，系统还通过后台 worker 与 systemd 服务支持订单状态自动轮询和超时处理，从而使订单流转不完全依赖人工操作。

7 项目目录结构说明

为了保证系统的可维护性和可部署性，项目在目录组织上采用了相对明确的结构划分。主程序目录中，`app.py` 作为 Flask 应用入口，负责核心业务逻辑；`templates/` 目录负责保存前台页面、商家后台页面、管理员后台页面及错误页面模板；`.env.runtime` 负责保存运行环境配置；`site_branding.json` 负责保存站点品牌相关配置；后台状态任务由独立 worker 脚本负责执行。

除线上部署结构外，项目还提供用于本地演示和验收的辅助脚本与运行文件，以支持在不同操作系统环境中完成系统启动与测试。通过将程序入口、页面模板、运行配置、后台任务和部署辅助脚本分离，系统在交付、迁移和维护时具有较好的结构清晰度。