## 2.3系统总体架构

**硬件架构**

1. Web服务器：使用一台或多台Web服务器来承载网站的前端页面和后端代码。选择常见的Web服务器软件Apache。

2. 应用服务器：使用一台或多台应用服务器来运行TP5框架和相关的业务逻辑。这些服务器可作为中间层，处理用户请求，查询数据库，生成动态页面，并返回给Web服务器。

3. 数据库服务器：使用一台或多台数据库服务器来存储和管理网站的数据。选择常见的关系型数据库MySQL。

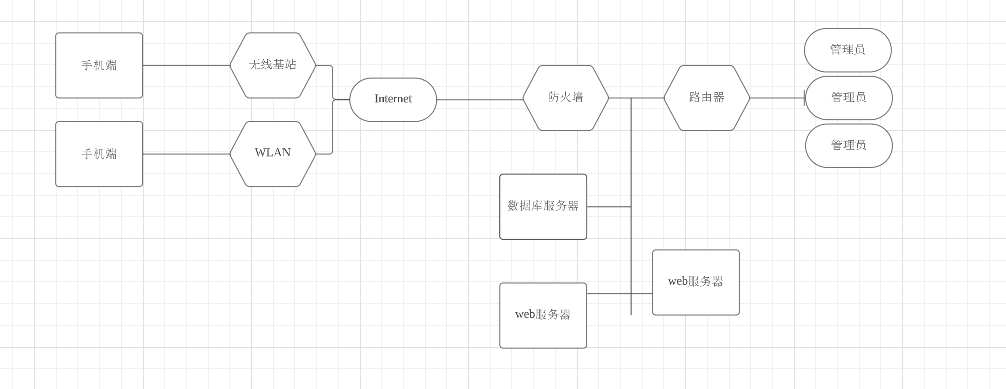
4. 负载均衡器：如果需要处理大量的并发请求，可以引入负载均衡器来平衡流量和请求分发到多台应用服务器上，提高网站的性能和可伸缩性。常见的负载均衡器有Nginx、HAProxy等。

5. 缓存服务器：使用缓存服务器来提高网站的性能和响应速度。可以选择常见的缓存服务器，如Redis、Memcached等，将频繁访问的数据缓存起来，减少对数据库的访问压力。

6. 文件存储：根据网站的需求，可以选择合适的文件存储解决方案。例如，使用对象存储服务，如Amazon S3、阿里云OSS等，来存储和管理大量的静态文件（如图片、视频等）。

7. 安全设备：为了保护网站和用户的数据安全，可以配置防火墙、入侵检测系统（IDS）、反DDoS系统等安全设备，确保网站的可用性和安全性。

8. 监控和日志系统：设置监控和日志系统来实时监控服务器、数据库和网络设备的运行状态，收集和分析关键指标和日志数据，发现潜在问题并进行及时调整和优化。



**软件架构**

1. 展示层（Presentation Layer）：

- 前端页面：使用HTML、CSS和JavaScript等前端技术来构建用户界面，通过模板引擎将动态数据嵌入到页面中，以呈现商品列表、购物车、用户登录等功能。

- 控制器（Controller）：处理用户请求并协调业务逻辑和数据交互，负责调用业务层的服务接口，并根据结果渲染合适的视图返回给用户。

2. 业务层（Business Layer）：

- 服务层（Service）：封装具体的业务逻辑，如商品管理、订单处理、用户注册等。负责处理数据验证、事务管理和协调多个数据操作，以及调用数据层的接口进行数据的增删改查操作。

- 领域模型（Domain Model）：定义业务实体和业务规则，表示电商领域中的核心概念，如商品、订单、用户等。

- 辅助类（Helper）：提供各种辅助函数和工具类，用于处理通用的任务，如文件上传、图片处理、缓存管理等。

3. 数据层（Data Layer）：

- 模型层（Model）：与数据库进行交互，封装数据访问和操作的方法，如查询、插入、更新、删除等。可以使用TP5的ORM（对象关系映射）功能，简化对数据库的操作。

- 数据库（Database）：存储和管理电商网站的数据，可以选择关系型数据库，如MySQL、PostgreSQL等，或者NoSQL数据库，如MongoDB、Redis等，根据需求和规模进行选择。

此外，还可以考虑引入其他组件来增强电商网站的功能和性能，例如：

- 缓存服务器（如Redis）：用于缓存热门商品、用户数据等，减轻数据库的负载。

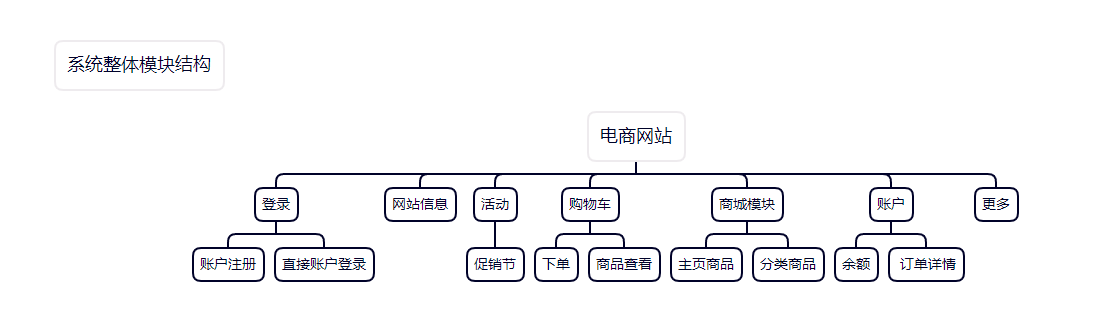
- 消息队列（如RabbitMQ）：用于异步处理订单、库存等耗时操作，提高响应速度和系统的可扩展性。

- 搜索引擎（如Elasticsearch）：提供全文搜索功能，支持商品搜索和筛选。

- 支付网关（如支付宝、微信支付）：集成第三方支付服务，实现在线支付功能。

## 2.4总体设计之功能模块设计

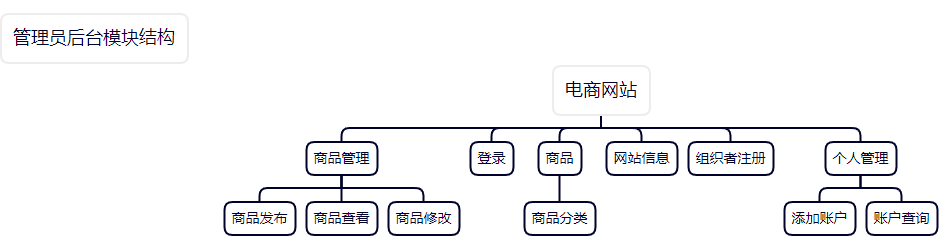
**系统总体模块设计**



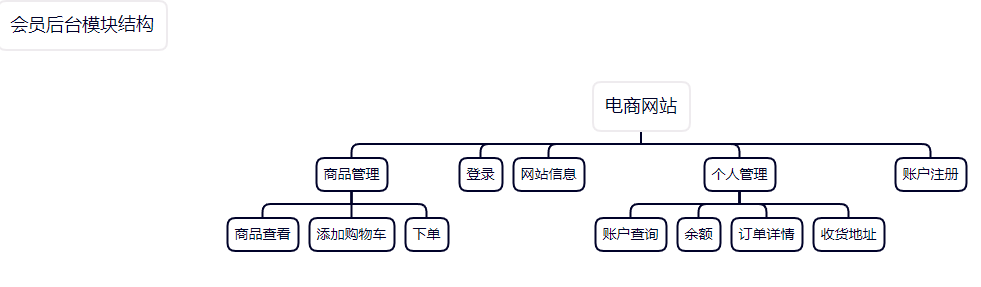
前台总体模块设计



管理员后台整体模块设计

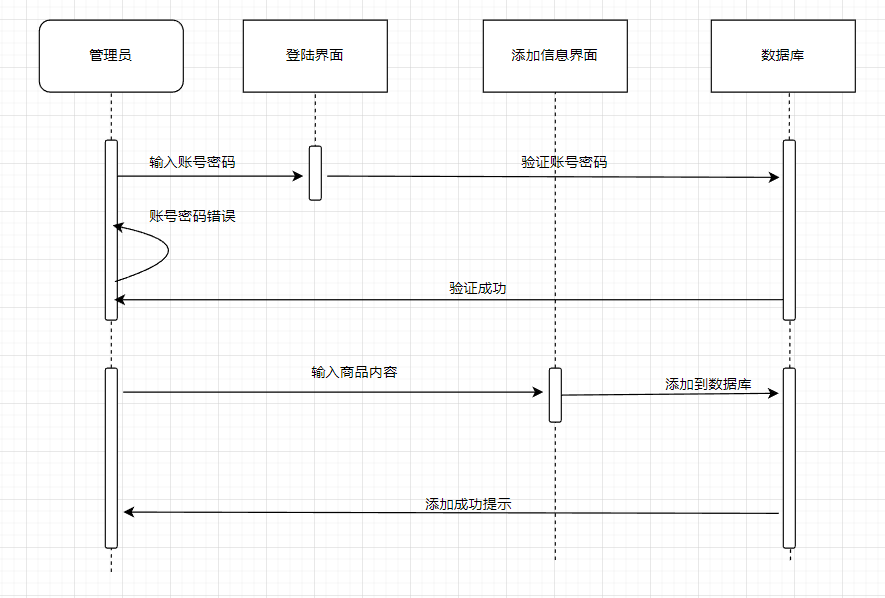


会员后台整体模块设计

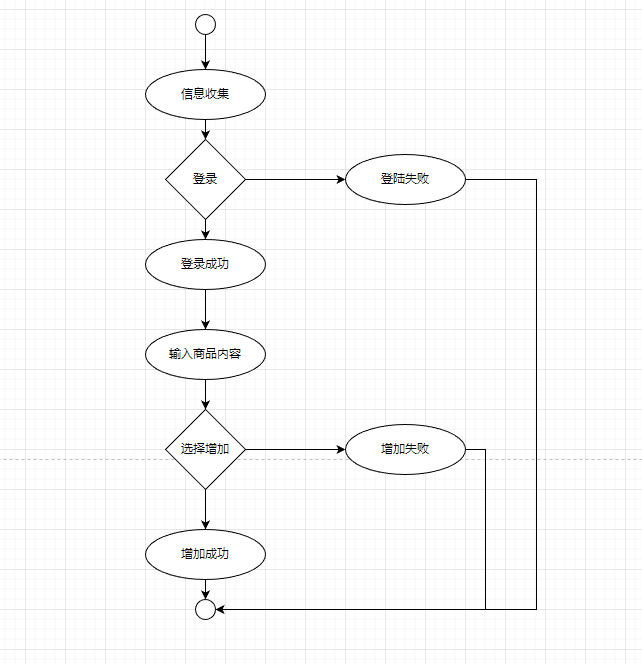


## 功能模块设计

添加商品顺序图

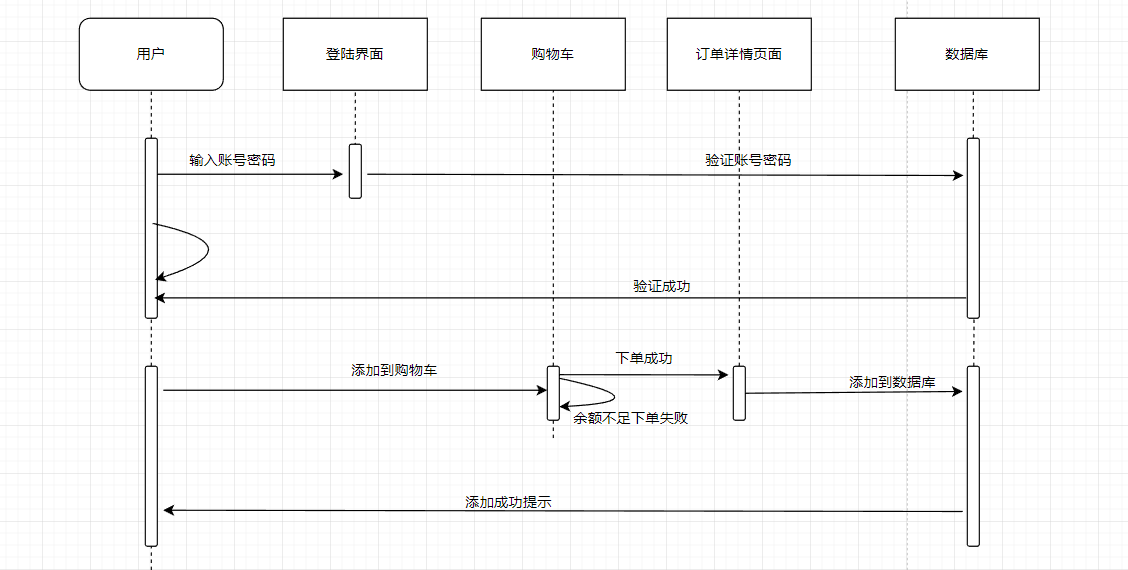


添加商品活动图



用户下单模块

活动图



活动图

