# 需求文档：Web Order 系统

## 1. 项目概述

项目名称: Web Order 系统

项目背景: 本项目旨在为用户提供一个在线乐器订购平台，用户可以方便地浏览乐器、将乐器添加到购物篮、提交订单、查看历史订单和评价商品。

项目目标: 开发一个基于 Web 的乐器订单管理系统，用户能够完成从登录、浏览乐器到下单购买的所有操作。

## 2. 功能需求

### 2.1 用户管理

- 登录（Log in）

- 用户能够通过用户名和密码登录系统。

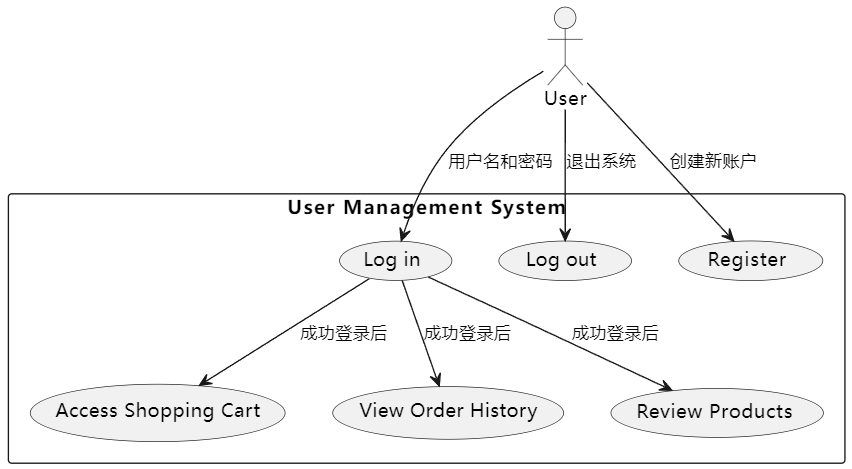
- 用户登录后可以访问购物车、订单历史和评价商品等功能。

- 登出（Log out）

- 用户可以随时退出系统。

- 注册用户

- 提供注册页面，允许用户创建新账户。



### 2.2 商品管理

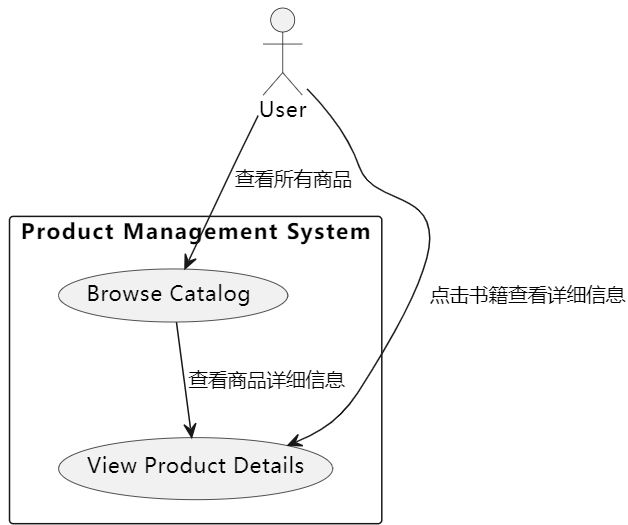
- 浏览商品目录（Browse Catalog）

- 用户可以查看所有可供购买的乐器。

- 每个商品的详细信息（如价格、描述、评价）将显示在列表中。

- 商品详情

- 用户可以点击乐器，查看详细的乐器信息，包括价格、库存状态和用户评论。



### 2.3 购物篮管理

- 添加商品到购物篮（Add Item）

- 用户可以将商品添加到购物篮中。

- 系统会验证库存是否足够。

- 移除商品（Remove Item）

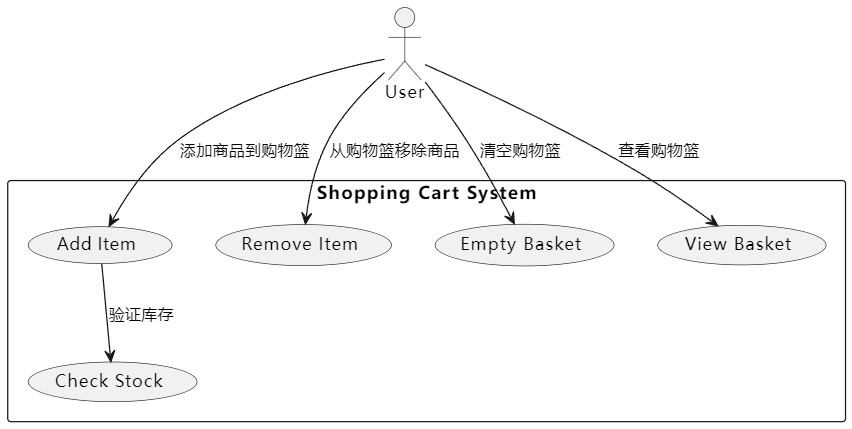
- 用户可以从购物篮中移除商品。

- 清空购物篮（Empty Basket）

- 用户可以清空购物篮中的所有商品。

- 查看购物篮

- 用户可以在购物篮页面查看所有已添加的商品，包括数量、单价和总价。



### 2.4 订单管理

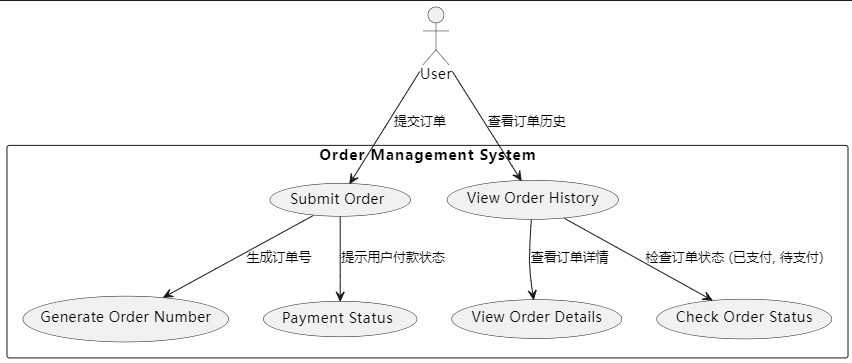
- 提交订单（Submit Order）

- 用户在购物篮中确认所选商品后可以提交订单。

- 系统会生成订单号，并提示用户付款状态。

- 查看订单历史（View Order History）

- 用户可以查看已提交的历史订单，包括订单详情和状态（已支付、待支付等）。



### 2.5 商品评价

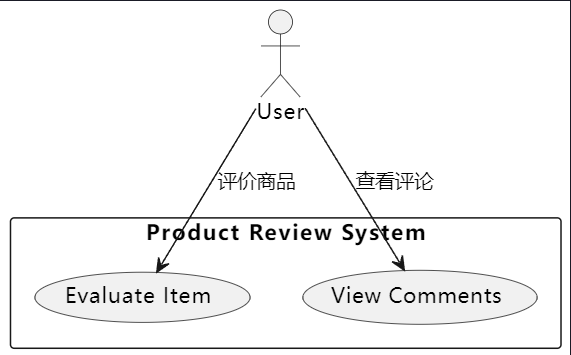
- 评价商品（Evaluate Item）

- 用户可以对已购买的商品进行评分和评价。

- 用户可以在订单完成后留下评论。

- 查看评论（View Comments）

- 用户可以查看其他用户对某本乐器的评价和评分。



### 2.6 Web 订单管理

- Web Order Shopper

- 系统允许用户通过Web界面下订单并进行支付。

### 2.7 数据库服务器管理

- 数据库设计

- 系统应具备高效的数据库架构，支持用户、商品、订单和评价等数据的管理。

- 需要确保数据一致性和安全性。

## 3. 非功能需求

### 3.1 性能需求

- 系统应能支持至少1000个并发用户访问。

- 商品页面的加载时间应控制在2秒以内。

### 3.2 安全需求

- 用户密码应加密存储，保证用户数据安全。

- 系统应使用SSL加密，保障数据传输的安全性。

### 3.3 可用性需求

- 系统应具有友好的用户界面，易于导航。

- 应支持在PC端和移动端浏览器中无缝使用。

### 3.4 扩展性需求

- 系统架构应支持未来功能的扩展，如支持更多的支付方式和促销功能。

## 4. 业务流程

### 4.1 用户购买流程

1. 用户登录或注册。

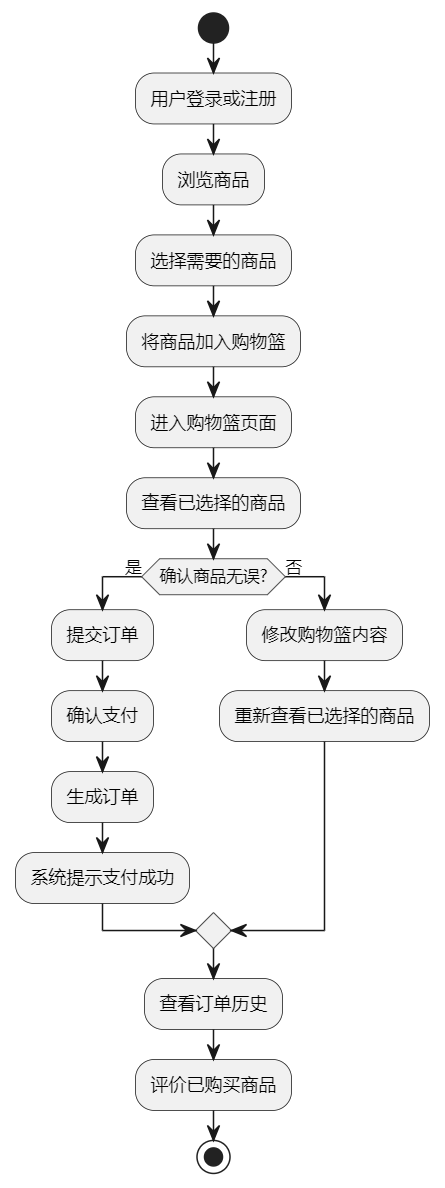
2. 用户浏览商品并选择需要的乐器。

3. 用户将乐器加入购物篮。

4. 用户进入购物篮页面，查看已选择的商品。

5. 用户提交订单并确认支付。

6. 用户可以查看订单历史并评价已购买的商品。



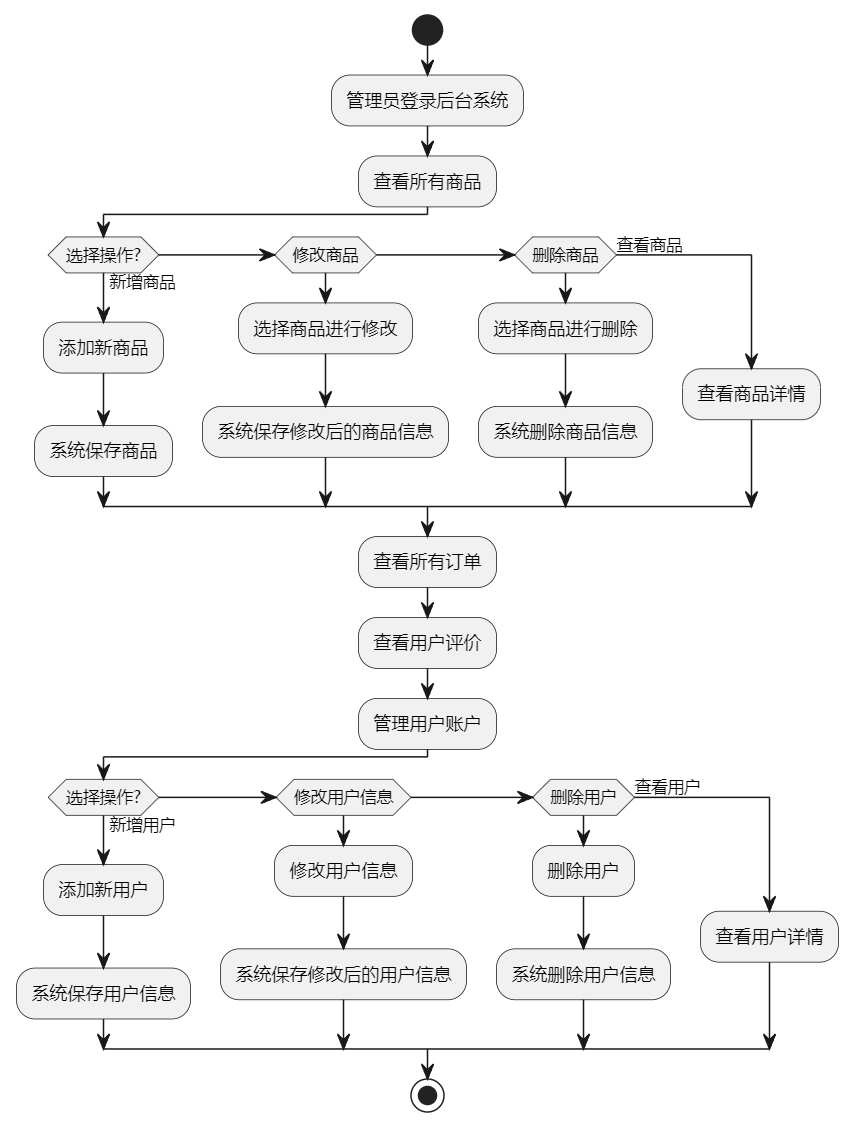
### 4.2 管理员流程

1. 管理员登录后台。

2. 管理员可以查看所有商品、订单和用户的评价。

3. 管理员可以对商品进行增删改查操作。

4. 管理员可以管理用户账户。



## 5. 数据需求

- 用户表

- 包含用户ID、用户名、密码、邮箱、角色（普通用户/管理员）等信息。

- 商品表

- 包含商品ID、乐器名、价格、库存、描述、评价平均分等信息。

- 订单表

- 包含订单ID、用户ID、订单状态、商品列表、总金额等信息。

- 评价表

- 包含评价ID、用户ID、商品ID、评分、评论内容、评价时间等信息。

## 6. 用户界面设计

### 6.1 登录页面

- 输入用户名和密码。

- “登录”按钮和“注册”按钮。

### 6.2 商品目录页面

- 显示商品列表，包括商品缩略图、乐器名、价格。

- “加入购物篮”按钮。

### 6.3 购物篮页面

- 显示已加入购物篮的商品，包括乐器名、数量、价格和“移除”按钮。

- “提交订单”按钮。

### 6.4 订单历史页面

- 显示用户的历史订单，包括订单号、下单时间、状态和订单总额。

### 6.5 商品详情页面

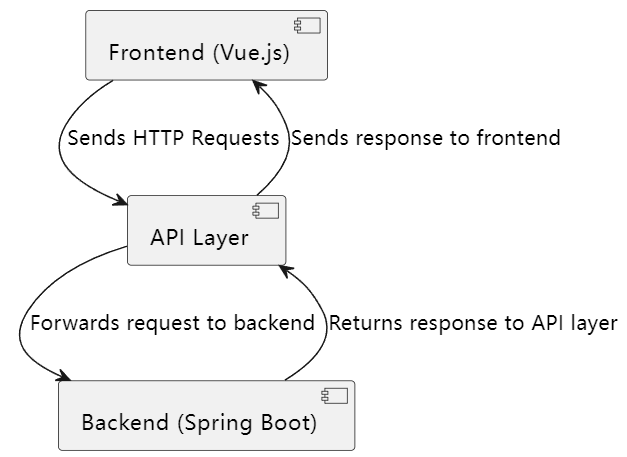
- 显示乐器详细信息、库存状态、用户评论。

## 7. 系统架构设计

### 7.1 客户端

- 前端框架： Vue.js

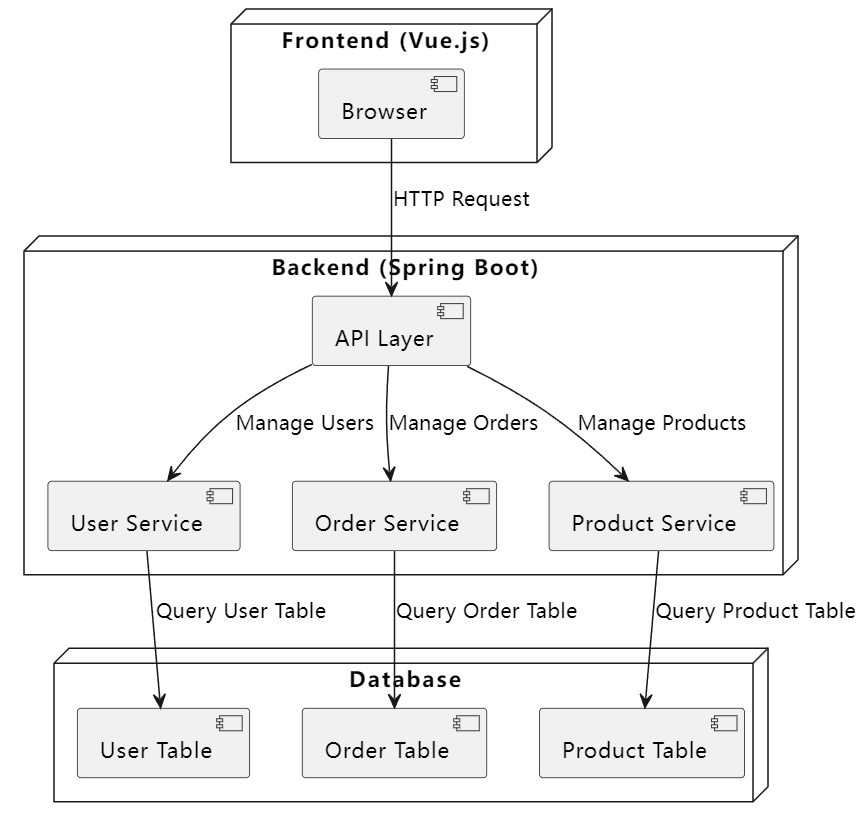
用户通过浏览器与前端进行交互，前端负责呈现UI和与后端API进行通信。



### 7.2 服务器端

- 后端框架：Spring boot

处理业务逻辑，包括用户管理、订单处理和商品管理。



### 7.3 数据存储

- 数据库：MySQL

数据库用于存储用户、订单、商品和评论等信息。

## 8. 质量要求

### 8.1 测试需求

- 功能测试：确保每个功能模块正常运行。

- 性能测试：保证系统能够在高并发下平稳运行。

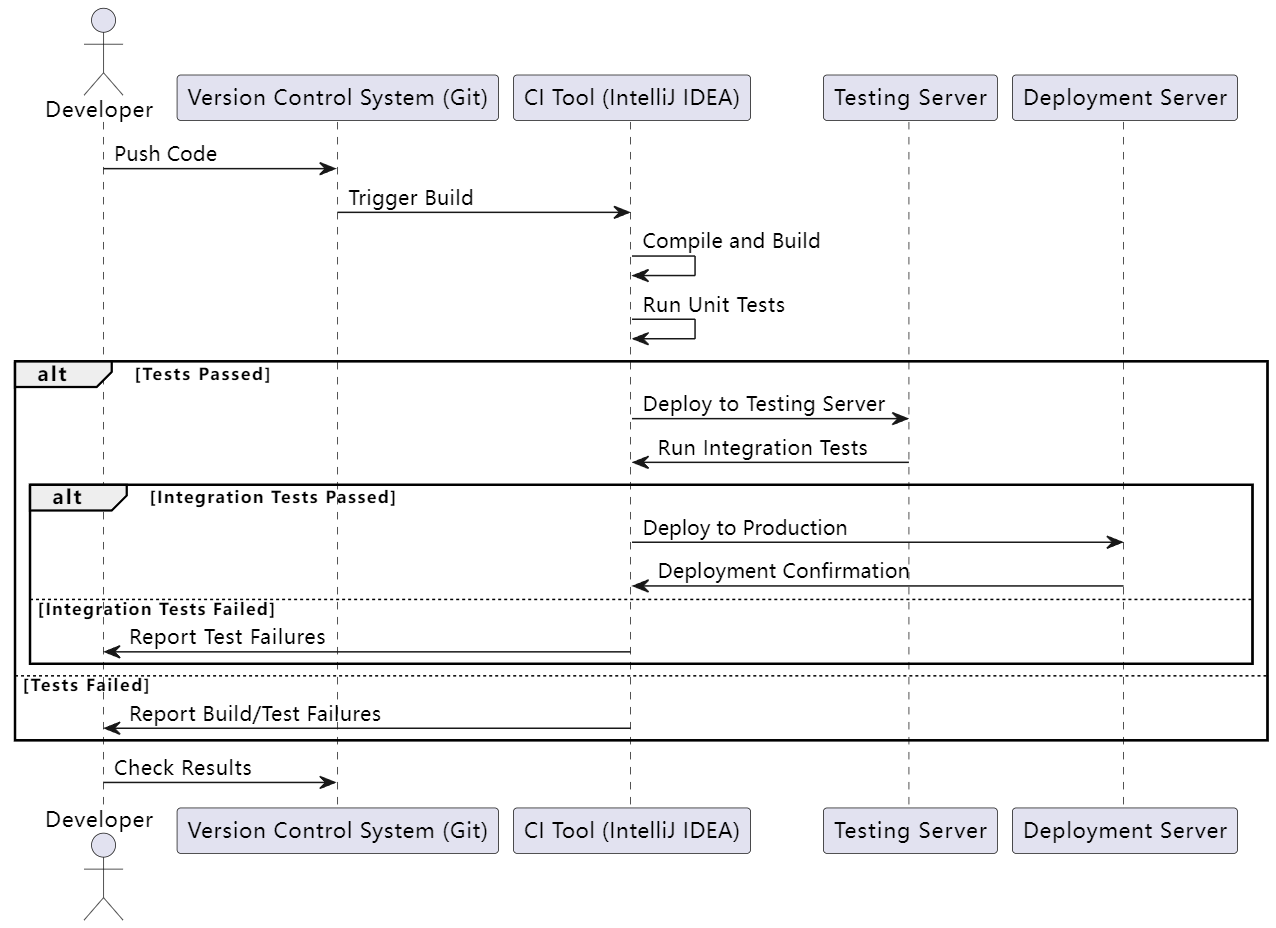
- 安全测试：确保系统安全性，防止SQL注入等安全问题。

### 8.2 代码审查

- 定期进行代码审查，确保代码符合最佳实践。

### 8.3 持续集成

- 使用持续集成工具（IntelliJ IDEA Ultimate Edition 2024.1.4）进行自动化构建和测试。



## 9. 项目管理

### 9.1 项目进度

- 采用敏捷开发模式，分为多个迭代，每个迭代为期两周。

### 9.2 风险管理

- 技术风险：技术选型不当或第三方库不兼容。

- 进度风险：由于任务分配不合理导致项目延误。

- 需求变更风险：用户需求频繁变化，影响开发进度。

## 10. 项目验收

- 系统功能实现并通过测试。

- 系统性能满足需求，无重大缺陷。

- 用户验收通过，系统正式上线。