# 软件工程项目开发计划：Web Order 系统（敏捷开发）

## 1. 项目背景

开发一个在线乐器订单管理系统（Web Order系统），旨在为用户提供便捷的乐器浏览、订购和管理服务。系统主要功能包括：

- 添加商品（Add Item）

- 查看订单历史（View Order History）

- 移除商品（Remove Item）

- 评价商品（Evaluate Item）

- 清空购物篮（Empty Basket）

- Web 订单购物（WebOrder Shopper）

- 查看评论（View Comments）

- 提交订单（Submit Order）

- 登录（Log in）

- 登出（Log out）

- 浏览商品目录（Browse Catalog）

- 数据库服务器（Database Server）

## 2. 人员安排

团队总人数：5人，其中一人负责较多任务。

|  |  |  |
| --- | --- | --- |
| 角色 | 职责 | 人员安排 |
| 项目经理/敏捷教练 | 负责项目管理、敏捷流程协调、进度跟踪 | 冯哲熙 |
| 产品负责人 | 收集和定义需求，优先级排序，确保开发满足业务需求 | 冯哲熙 |
| 前端开发工程师 | 负责用户界面设计与实现，确保前后端交互 | 宋羽豪、李纪群、冯哲熙 |
| 后端开发工程师 | 负责服务器端逻辑、API 开发与数据库交互 | 宋羽豪、李纪群、冯哲熙 |
| 全栈开发工程师 | 负责协助前后端开发，处理多个模块的集成与优化 | 宋羽豪、张翼天 |
| 测试工程师 | 负责功能测试、性能测试、安全性测试和用户体验测试 | 马嘉笠 |

## 3. 技术选型

- 前端技术：Vue.js

- 后端技术：Node.js

- 数据库：MySQL

- 服务器：AWS 或 Azure

- 版本控制：Git（GitHub）

- 项目管理工具：Git（GitHub）

- 持续集成/部署：IntelliJ IDEA Ultimate Edition 2024.1.4

## 4. 敏捷迭代计划

总共4周，采用2个两周的迭代。

### 迭代1（第1-2周）：基础功能开发

目标：实现系统的基本架构和核心功能，为后续开发打下基础。

功能：

- 用户登录/登出（Log in / Log out）

- 浏览商品目录（Browse Catalog）

- 添加商品到购物篮（Add Item）

- 移除商品（Remove Item）

- 清空购物篮（Empty Basket）

- 数据库基础设计与实现

任务分配：

- 项目经理/敏捷教练：协调团队，组织每日站会，管理任务进度

- 产品负责人：定义和整理需求，编写用户故事，优先级排序

- 前端开发工程师：开发登录页面、商品浏览页面、购物篮功能

- 后端开发工程师：开发用户认证、商品管理API，设计数据库

- 全栈开发工程师：协助前后端开发，处理集成问题，优化系统性能

验收标准：

- 用户可以注册、登录、登出

- 用户可以浏览商品目录

- 用户可以添加、移除、清空购物篮

- 数据库设计完成并与后端成功交互

### 迭代2（第3-4周）：高级功能开发与系统优化

目标：实现系统的高级功能，完善用户体验，确保系统稳定性。

功能：

- 提交订单（Submit Order）

- 查看订单历史（View Order History）

- 评价商品（Evaluate Item）

- 查看评论（View Comments）

- Web 订单购物（WebOrder Shopper）

- 系统集成与测试

- 部署与上线

任务分配：

- 项目经理/敏捷教练：继续协调，确保迭代目标达成

- 产品负责人：进一步完善需求，收集反馈

- 前端开发工程师：开发订单提交、订单历史、商品评价与评论页面

- 后端开发工程师：开发订单管理、评价与评论API，优化数据库

- 全栈开发工程师：处理系统集成，优化前后端交互，协助测试与部署

验收标准：

- 用户可以提交订单并查看订单历史

- 用户可以评价商品并查看评论

- Web 订单购物功能正常运行

- 系统经过全面测试，无重大缺陷

- 系统成功部署上线，并通过用户验收

## 5. 项目时间表

|  |  |  |
| --- | --- | --- |
| 时间 | 任务 | 负责人 |
| 第1周 | 需求分析，用户故事编写，系统架构设计，数据库设计 | 全体团队 |
| 第2周 | 迭代1开发与测试，完成基础功能 | 前后端开发工程师，全栈开发工程师 |
| 第3周 | 迭代2开发与测试，开发高级功能 | 前后端开发工程师，全栈开发工程师 |
| 第4周 | 系统集成、优化、全面测试，部署上线，用户验收 | 测试工程师 |

## 6. 迭代细化

### 迭代1详细任务

- 用户登录/登出：

- 前端：设计并实现登录页面

- 后端：开发用户认证API（注册、登录、登出）

- 全栈开发工程师：整合前后端，确保认证流程顺畅

- 浏览商品目录：

- 前端：设计并实现商品列表页面

- 后端：开发商品管理API（获取商品列表、商品详情）

- 购物篮功能：

- 前端：设计并实现购物篮页面，添加/移除商品功能

- 后端：开发购物篮管理API（添加、移除、清空购物篮）

- 数据库设计：

- 设计用户、商品、订单、购物篮等表结构

- 实现基本的数据库操作（CRUD）

### 迭代2详细任务

- 提交订单：

- 前端：设计并实现订单确认页面

- 后端：开发订单处理API（创建订单、支付接口）

- 查看订单历史：

- 前端：设计并实现订单历史页面

- 后端：开发订单查询API（按用户ID获取订单列表）

- 评价商品与查看评论：

- 前端：设计并实现商品评价与评论页面

- 后端：开发评价与评论API（提交评价、获取评论）

- Web 订单购物：

- 优化购物流程，确保用户体验流畅

- 系统测试与优化：

- 编写并执行测试用例（功能测试、集成测试）

- 修复缺陷，优化系统性能

- 部署与上线：

- 配置服务器环境，部署应用

- 进行上线前的最终检查，确保系统稳定

## 7. 风险管理

- 技术风险：

- 技术栈选择不当导致开发困难或进度延误。

- 应对措施：在项目初期进行技术评估，确保技术选型适合项目需求。

- 进度风险：

- 开发任务超出预期时间，影响整体进度。

- 应对措施：每日站会跟踪进度，及时调整任务优先级，确保关键功能优先完成。

- 人员风险：

- 团队成员因故离职或无法按时完成任务。

- 应对措施：跨职能培训，确保团队成员具备多项技能，减少依赖单一成员。

- 需求变更风险：

- 用户需求频繁变更，影响开发进度。

- 应对措施：采用敏捷方法，灵活调整迭代计划，及时响应需求变化。

## 8. 质量保证

- 持续集成：使用CI工具（如 Jenkins 或 GitHub Actions）自动化构建和测试，确保代码质量。

- 代码审查：定期进行代码审查，确保代码符合规范，减少缺陷。

- 自动化测试：编写自动化测试用例，覆盖主要功能，确保系统稳定性。

- 用户反馈：通过敏捷迭代收集用户反馈，及时改进产品。

## 9. 交付与验收

- 迭代交付：每个迭代结束后交付可运行的功能模块，确保持续提供价值。

- 最终验收：系统部署上线后，进行全面的用户验收测试，确保满足业务需求和用户期望。

## 10. 项目总结

在4周内通过敏捷开发方法，合理分配团队资源，集中开发核心功能和高级功能，确保Web Order系统按时高质量交付。通过迭代的方式不断优化系统，快速响应需求变化，提高团队协作效率，最终实现用户满意的在线乐器订单管理系统。