课程论文

学年学期 **2023** —**2024**学年第 一 学期

课程名称 软件工程综合实践2

题 目 仿12306铁路售票系统

姓 名 王邦城、郑鑫、田嘉怡、姜塬滨、张梓祺、衣红竹

学 院 计算机科学与信息工程学院

任课教师 房益国

论文评语（由任课教师填写）：

成 绩

**目录**

**1项目说明1**

1.1项目内容1

1.2项目背景1

1.1.1技术1

**2需求分析1**

2.1需求分析的方式1

2.1.1方式1

2.1.2步骤1

2.2产生需求1

**3项目设计2**

3.1项目框架设计2

3.2功能模块设计2

3.3系统设计2

3.4数据库设计2

3.5测试与维护3

**4项目实现3**

**5总结3**

**摘要**：12306是中国铁路部门推出的网上售票平台。是一个高度复杂、精密的系统，要参照该系统进行建模设计，尽可能将系统接近覆盖真实上线系统。需要支持高并发、低延时等特点，整体采用微服务的架构风格，其真正的视图层为通过Vue.js搭建的前端系统开发。

**关键词：**仿12306铁路售票系统

**第1章 项目说明**

**1.1项目内容**

12306是中国铁路部门推出的网上售票平台。作为一个拥有众多人口以及幅员辽阔地域面积的铁路大国，中国的12306购票系统也承担着每日最高千亿次访问量，一年售票超过千万张。在如此高并发的环境下，注定12306系统是一个高度复杂、精密的系统。作为工程实践，我们要参照该系统进行建模设计，尽可能将系统接近覆盖真实上线系统。**1.1.1技术**

**技术栈：**Spring Boot +Mybatis-Plus+ Redis +RocketMQ +ShardingSphere+SpringCloud Alibaba+ Canal + Sentinel +Hippo4j +XXL-Job+SkyWalking

**1.2 项目背景**

围绕哈尔滨新区开发的重点工作目标，以更加规范化地管理新区的交通往来为宗旨，以努力争取成为“双创”城市、为百姓的出行便利、政府的高效率管理为目标，以零差错的同步管理、分批次实施为原则，开发一个模拟12306铁路购票系统的项目，为群众购票、出行提供一系列的技术支持，更为政府的交通管理提供了一个可实验的平台，为未来新区的发展提供了强有力的保障。

**第2章 需求分析**

**2.1需求分析的方式**

**2.1.1方式**

HWM分析法（How Might We = 我们可以如何 = 有什么办法）：在头脑风暴前可以进行拆解分析，在分析用户反馈的时候可以用HWM进行分析，在和领导讨论的时候，可以用来进行分析。

**2.1.2步骤**

1）明确用户和问题

2）拆解问题的方向

否定（逆向思维）：如何想办法让用户放弃这个想法

积极（发挥积极影响）：如何让用户提升自己来解决这个问题

转移（移除消极影响/换其他方案）：如何让其他人解决问题，继而解决这个用户的问题

脑洞大开：发散思维，从需求或环境中创造类似的体验，或改变现状……

分解：将大问题拆分成很多小问题

3）针对每个HWM做方案

穷举：头脑风暴，列举能想到的所有可能性

打开思路：先想，不可思议的也可以写出来，后面再做限制

**2.2产生需求**

1、会员服务

（1）乘车人管理：根据用户名查询乘车人列表、根据乘车人 ID 集合查询乘车人列表、新增乘车人、修改乘车人、移除乘车人；

（2）会员管理：根据用户名查询用户信息、检查用户名是否已存在、注册用户、用户登录、通过 Token 检查用户是否登录、用户退出登录。

2、网关服务：验证用户是否登录、系统访问白名单、系统访问黑名单。

3、购票服务

（1）车站管理：查询车站&城市站点集合信息、根据列车ID查询站点信息；

（2）车票管理

a、根据条件查询车票；

b、购买车票：高铁在线选座、多人邻座选择、创建车票订单；

c、初始化任务：地区站点查询定时任务、站点详细信息定时任务、列车路线信息定时任务、列车站点余票定时任务。

4、订单服务——订单管理

（1）创建车票订单：自定义订单号生成器、延迟关闭订单、防止订单重复提交；

（2）取消订单：订单状态反转；

（3）订单信息补充：支付回调；

（4）订单查询：敏感数据查询脱敏、敏感数据入库加密、乘车人查询本人订单。

5、支付服务

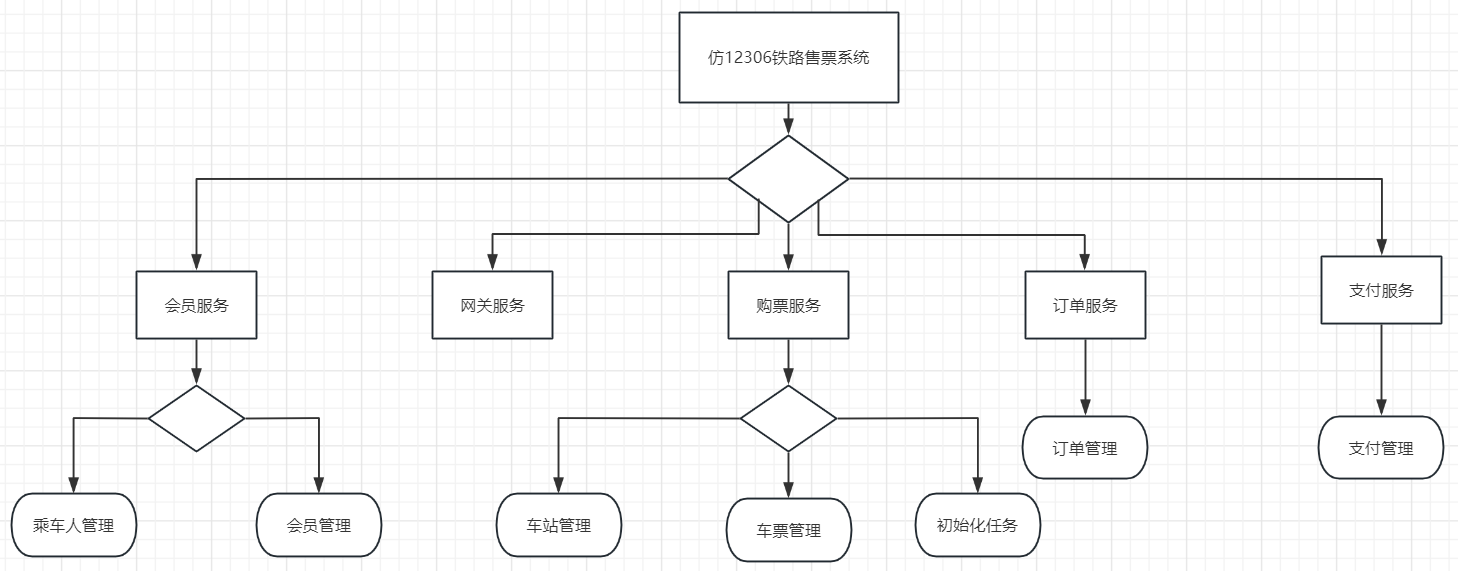
（1）支付管理：支付宝支付、微信支付、银行卡支付、支付状态查询。

**第3章 项目设计**

**3.1技术框架设计**

本系统需要支持高并发、低延时等特点，整体采用微服务的架构风格，而对于后端系统而言，采用SpringCloud框架搭建基本微服务，本系统前端采用Vue.js搭建，对于微服务的每个组件而言，其后端系统架构类似于MVC架构，Mode1模型层用来表示数据模型以及其对应的持久化操作。在MVC架构下，模型封装了数据。View视图层用来展示UI界面，和用户做交互。它将从模型层获取的数据展示给用户看。Controller控制器层作用在模型层和视图层上，使得视图层和模型层可以分解耦合，视图层只关心页面表示，而模型层只关注数据表示。在本系统中，MVC架构的View被基于RESTful风格的api代替，而其真正的视图层为通过Vue.js搭建的前端系统开发。

**3.2功能模块设计**



**3.3系统设计**

根据需求分析的结果，我们进行了系统的整体设计，如下

(1)架构设计:基于springboot，vue3

(2)界面设计：根据用户需求和操作习惯，我们设计了友好的用户界面，包括输入框、下拉框、按钮等元素。使用CSS制作了网页背景图、鼠标经过及选中导航变色效果、下划线等相关技术来美化相关界面，部分采用了javascript来做校验。 使用html5，以及css3等相关技术完成技术的布局，在本作品中，会使用常见的布局，常见的浮动布局，flex布局都会有使用到哦。同时在操作方面上运用了html5和css3，采用了div+css结构、表单、超链接、浮动、绝对定位、相对定位、字体样式、引用视频等基础知识，同时使用了一些js的相关知识。

(3)接口设计：用户登入示例，通过登录，用户方可使用本软件，且我们可以根据用户存储相对于的资料等信息，接口

**3.4数据库设计**

1）、数据库类型选择：根据系统需求和性能要求，选用关系型数据库MySQL

2）、数据库表设计：根据系统功能需求，设计相应的数据库表，包括用户表、管理员表、订单表等。

3）、数据库关联设计：通过主键和外键的设计，实现各表之间的关联关系。

**3.5测试与维护**

1)、测试

在完成详细设计后，我们将进行系统测试，包括单元测试、集成测试和系统测试。通过测试，我们将发现并修复可能存在的缺陷，确保系统的稳定性和可靠性。

2)、维护

为了保持系统的正常运行，我们将制定维护计划，包括定期检查、升级和维护。在发现缺陷或需要进行功能扩展时，我们将及时进行修复或修改，以确保系统的持续可用性和安全性能。

**注：系统设计原则**

1、用户友好性：确保系统的用户界面简洁、易于理解和操作，降低用户的学习成本。

2、可扩展性：考虑到业务发展的需要，系统应具备可扩展性，以适应未来的需求变化。

3、安全性：系统的设计必须严格遵循信息安全标准，确保数据隐私和保密性。

4、灵活性：能够灵活应对不同的业务需求，方便进行功能调整和扩展

**第4章 项目实现**

**第5章 总结**

在不使用工业工程的思想优化时、我们很容易发现、原系统造成了无谓的浪费，在时间上会直接导致服务时间延长，在服务上也会导致质量的下降，使大量的人处于不停的等待状态。同样的道理，如果能很好地将软件工程的思想应用到服务业的流程设置和改造上，必然会使其服务质量有所提高，并且有效地约束服务行为。这些方法对其他行业也同样适用。