**铁路售票系统**

**架构设计文档**

**组长：张付俊**

**组员：孙帅、黄东鹏**

目录

[一、项目背景 2](#_Toc497122444)

[1.概述 2](#_Toc497122445)

[2.目标 2](#_Toc497122446)

[二、需求分析 2](#_Toc497122447)

[1.功能性需求 2](#_Toc497122448)

[2.非功能性需求 3](#_Toc497122449)

[3.系统定义 3](#_Toc497122450)

[4.细化用例 3](#_Toc497122451)

[5.用例图 4](#_Toc497122452)

[三、系统架构设计 8](#_Toc497122453)

[1.体系结构 8](#_Toc497122454)

[2.质量场景与策略选择 9](#_Toc497122455)

[1）性能场景 9](#_Toc497122456)

[2）可用性场景 9](#_Toc497122457)

[3）安全性场景 9](#_Toc497122458)

[4）灵活性场景 9](#_Toc497122459)

[5）易用性场景 9](#_Toc497122460)

[3.类图设计 10](#_Toc497122461)

[4.数据表设计 10](#_Toc497122462)

# 一、项目背景

## 1.概述

选课是各大高校、每学期、每名学生都将参与的一项活动，网上选课，作为一种普遍接受的选课形式，已经成为教务管理中不可或缺的一部分。网上选课系统也随着使用频率、课程种类要求、选课规则的完善，而越来越复杂。早期的选课系统，由于未考虑到选课人数的庞大，选课规则的复杂，而难以胜任需求。

计算机网络的发展对一种基于B/S模式的课务系统提出了新的需求。由于选课管理任务的较高并发性与规则的特殊性，课务系统需要一定的反应速度，对规则的可扩展性，同时除了基本的选课、改选等功能外，还要满足响应多点请求等功能。

## 2.目标

在满足售票系统的基本登录、选课、改选、退退、查询等功能的同时，充分考虑其安全性和稳定性。

# 二、需求分析

## 1.功能性需求

|  |  |
| --- | --- |
| 功能 | 说明 |
| 用户信息管理 | 1.对注册的用户进行身份信息认证  2.对注册成功的用户的身份信息进行管理  3.对用户的登陆密码进行修改 |
| 课程信息管理 | 课务信息添加、删除等操作 |
| 课程查询 | 根据用户对课程类型、时间、地点、学分等要求，对符合条件的课程的最大容量、当前人数、授课老师进行查询和显示。 |
| 选课 | 根据用户的课程选择进行验证、生成相应请求、修改选课记录等操作。 |
| 退选 | 支持用户对已经支付的车票进行退订，并根据相关规定扣除手续费后进行退款操作 |
| 改选 | 支持用户对已经支付的车票进行改签，对自己的行程进行修改，并根据相关规定扣除手续费后进行收款或退款操作 |
| 课程查询 | 支持用户对自己的已选、已修课程的信息进行查询。 |

## 2.非功能性需求

|  |  |
| --- | --- |
| 质量属性 | 说明 |
| 高并发性 | 由于选课需求的特殊性和复杂性，在开放选课期间能够对用户的查询、选课操作进行尽可能快速处理，如果在短时间未能相应用户操作需求，则显示相应信息 |
| 安全性 | 1.对注册用户的身份信息进行验证  2.给管理用户信息的部分信息高权限管理  3. 在web数据库客户端，web服务器和数据库服务器之间应设有防火墙保护，防止网络上的非法数据请求  4.对用户购票进行验证码验证处理，防止脚本登陆抢课行为  5.用户修改密码时进行手机验证码验证 |
| 可用性 | 选课系统在开放选课期间全时段正常运作，减少系统故障率，存在故障时减少系统故障时间，尽可能减少维护时间，使用户购票更为便利 |
| 灵活性 | 售票系统应该具有可移植性，可以在不同平台上运行，方便用户以不同形式进行使用。并且对选课规则的 |
| 易用性 | 售票系统面向广大学生用户，因此要方便使用，并且做到界面简洁易操作，方便对互联网操作基础各异的各类人群进行使用。 |

## 3.系统定义

根据功能需求，可得本系统主要面向人群为普通用户和系统管理人员，系统管理人员分为用户信息管理人员和课务信息管理人员。

|  |  |
| --- | --- |
| 系统角色 | 要求 |
| 用户信息管理人员 | 根据信息登录获得系统用户信息管理权限，并可以使用系统对用户的信息进行查询和管理 |
| 课务信息管理人员 | 根据信息登录获得系统课务信息管理权限，并可以对课务信息进行更改，查询和管理 |
| 普通用户 | 根据身份验证进行登录成功后可对课程信息进行查询，进行正常选课，退选，改选操作。 |

## 4.细化用例

|  |  |
| --- | --- |
| 用例 | 说明 |
| 登录 | 用户使用身份信息进行登录 |
| 查询 | 用户根据课程时间，地点，课程类型对课程容量，当前人数，授课老师进行查询 |
| 选课 | 用户选择课程后系统向服务器发送选课请求。 |
| 改选 | 用户对已选的课程进行修改，系统自动生成新的选课请求，修改原有的选课信息。 |
| 退选 | 用户对选课成功的课程进行删除，系统自动删除相应选课记录，进行退选操作 |
| 课程信息修改 | 课务信息管理员在登录获得权限后，可对课务信息，车次信息，发车时间进行修改 |
| 课程信息查询 | 课务信息管理员可对课务信息进行查询 |
| 课程信息维护 | 课务信息管理员在登录获得权限后，可对课务信息进行维护和备份 |

## 5.用例图

。。。

|  |  |
| --- | --- |
| 要素 | 说明 |
| 用例名称 | 用户登录 |
| 简要描述 | 已经完成信息认证的用户登录系统 |
| 事件流 | 基本事件流：  1.用户填写用户名，密码  2.用户登录进入主界面 |
| 备选事件流 | 用户密码与用户名不匹配或用户名不存在，则  1.弹出错误提示界面  2.返回登录界面，用户重新进行登录  用户忘记密码，则  1.系统向注册时使用的邮箱发送验证邮件  2.用户使用验证邮件中的验证码进行验证  3.验证成功后，用户进入密码修改页面输入新密码并确认新密码 |
| 特殊需求 | 无 |
| 前置条件 | 用户已经成功注册 |
| 后置条件 | 进入用户界面 |

|  |  |
| --- | --- |
| 要素 | 说明 |
| 用例名称 | 用户查询课程 |
| 简要描述 | 用户根据选定的课程时间，地点，课程类型查询课务信息 |
| 事件流 | 基本事件流：  1.用户输入课程时间，地点，课程类型  2.系统将符合条件的课程信息，课程时间，容量，当前人数，授课老师，学分信息形成列表显示在主界面  3. 此用例结束 |
| 备选事件流 | 没有与用户输入的信息相匹配的课程，则界面显示相应提示信息 |
| 特殊需求 | 无 |
| 前置条件 | 用户已成功登陆系统 |
| 后置条件 | 系统调用数据库中数据进行显示 |

|  |  |
| --- | --- |
| 要素 | 说明 |
| 用例名称 | 用户提交选课 |
| 简要描述 | 用户在查询的列表中选择选课 |
| 事件流 | 基本事件流：  1.用户选定查询列表中选定课程  2.用户提交订单并弹出确认界面  3.用户选择确认完成选课  4.此用例结束 |
| 备选事件流 | 用户选择车票后没有确认，则不发送选课请求。 |
| 特殊需求 | 无 |
| 前置条件 | 用户输入相应筛选条件 |
| 后置条件 | 完成确认选课后，系统更新数据库 |

|  |  |
| --- | --- |
| 要素 | 说明 |
| 用例名称 | 用户查询已选课程 |
| 简要描述 | 用户对自己的已选课程进行查询 |
| 事件流 | 基本事件流：  1.用户选择查询已选课程  2.系统对已选记录中课程信息，时间、地点形成列表进行显示  3.此用例结束 |
| 备选事件流 | 没有相应的订单，则显示相应提示信息 |
| 特殊需求 | 无 |
| 前置条件 | 用户已成功登陆系统 |
| 后置条件 | 无 |

|  |  |
| --- | --- |
| 要素 | 说明 |
| 用例名称 | 用户退选 |
| 简要描述 | 用户对已经选的课程进行退选 |
| 事件流 | 基本事件流：  1.用户点击进入已选课程项  2.用户选择想要退选的订单，提交退选请求。  3.系统进行退选操作，修改数据记录。  4.此用例结束 |
| 特殊需求 | 无 |
| 前置条件 | 用户已成功支付订单 |
| 后置条件 | 系统对数据库中数据进行修改 |

|  |  |
| --- | --- |
| 要素 | 说明 |
| 用例名称 | 用户改选 |
| 简要描述 | 用户对已经选的课程进行修改 |
| 事件流 | 基本事件流：  1.用户点击进入已选课程项  2.用户选择想要改选的课程  3.用户进入查询界面重新选择课程  4.系统形成新的选课请求  5.此用例结束 |
| 特殊需求 | 无 |
| 前置条件 | 用户已成功完成选课 |
| 后置条件 | 系统更新数据库中信息 |

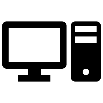
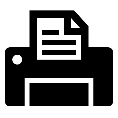
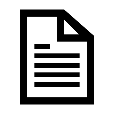
|  |  |
| --- | --- |
| 要素 | 说明 |
| 用例名称 | 用户信息管理员查询 |
| 简要描述 | 用户信息管理员对用户信息进行查询 |
| 事件流 | 基本事件流：  1.用户信息管理员输入身份信息登录系统  2.用户信息管理员输入权限码进入用户信息查询界面  3.用户信息管理员输入用户名及身份证号对用户信息进行查询  4..此用例结束 |
| 备选事件流 | 用户信息管理员输入权限码3次错误，则  暂时封锁该管理员账号 |
| 特殊需求 | 无 |
| 前置条件 | 用户信息管理员正确输入权限码 |
| 后置条件 | 无 |

|  |  |
| --- | --- |
| 要素 | 说明 |
| 用例名称 | 课务信息管理员修改课务信息 |
| 简要描述 | 课务信息管理员对课程信息进行修改 |
| 事件流 | 基本事件流：  1.课务信息管理员输入身份信息登录系统  2.课务信息管理员输入权限码进入修改课务信息界面  3.课务信息管理员对课务信息进行修改  4.课务信息管理员对课务信息进行保存  5.此用例结束 |
| 备选事件流 | 课务信息管理员输入权限码3次错误，则  暂时封锁该管理员账号 |
| 特殊需求 | 无 |
| 前置条件 | 课务信息管理员正确输入权限码 |
| 后置条件 | 系统自动更新数据库，并备份数据 |

# 三、系统架构设计

## 1.体系结构

本系统采用基于B/S的分层结构。这种结构有如下特点：节省投资、跨地域广；维护和升级方式简单，如果想对功能修改，可以方便的进行更改，大大减少维护成本。

表示层 业务逻辑 数据访问层

系统采用的设计模式：三层设计模式

视图层：提供显示界面，利用MFC实现

业务逻辑层：处理基本的业务逻辑以及实体类设计，利用Javascript ，Node.js实现

数据访问层：处理底层与数据库的交互，利用Python实现。

## 2.质量场景与策略选择

1）性能场景：在系统处于高峰时期，保证登陆的每个用户所作的选择和查询的响应时间能在2s以内，如果需要等待则给出有友好的提示。系统可以保证以最快速度同时响应500个用户的操作。

策略选择：限制队列大小和**LRU**缓冲战术。

2）可用性场景：选课系统尽可能在开放选课期间全时段正常运作，减少系统故障率，存在故障时减少系统故障时间，尽可能减少维护时间，出现故障时系统有相应的处理机制，而且处理时间必须较短。

策略选择：异常检查，故障恢复机制、资源调度机制。

3）安全性场景：杜绝非法用户试图绕过应用服务器直接连接到数据库服务器的端口上，防止非法窃取注册用户个人息；屏蔽某IP短时间内的大量无意义的访问，以防被挤爆，使正常用户无法使用，保证系统数据的机密性和完整性；防止脚本抢课。

策略选择：身份验证，验证码，服务器防火墙，用户信息的高权限管理，用户信息加密存储，用户密码MD5加密验证

4）灵活性场景：选课系统可能会在之后修改选课规则，因此需要系统具有良好的可延展性与可重用性，减少工作量。

策略选择：三层设计模式，使视图层，业务逻辑层，数据访问层分来

5）易用性场景：在该系统中，用户希望在运行时能尽快取消某操作使错误的影响降到最低，取消在1秒内发生；要求具有基本电脑操作常识的人，可以根据界面迅速学会使用方法，让熟手用户使用快捷键。

策略选择：简化界面，快捷键双键设定，用户独立接口

## 3.类图设计

。。。

## 4.数据表设计

用户类：

|  |  |  |
| --- | --- | --- |
| 属性名 | 数据类型 | 限制 |
| 姓名 | String | Not null |
| 用户名 | String | Not null |
| 密码 | String | Not null |
| 电话 | String | Not null |

课程类：

|  |  |  |
| --- | --- | --- |
| 属性 | 数据类型 | 限制 |
| 课程名称 | String | Not null |
| 容量 | Int | Not null |
| 当前人数 | Float | Not null |

//TODO