# 第一章 项目简介

## 1.1项目背景

本系统主要为图书馆提供座位管理功能和为学习的人提供便捷的座位预约服务。随着国家对大学生教育标准和社会对大学生需求的逐渐严格，高校中学生的自习需求呈现直线增加趋势。但因为大学生时间自由，人群基数大，导致学习环境恶劣，很难进入学习状态。而图书馆的学习环境氛围良好、资料查询方便，逐渐成为自习圣地，但图书馆存在大量的座位管理问题，在没有有效的座位管理机制前，会发现日常很多座位只有书本或其它学习用具，且长期没有人。这种现象的出现是大家为了保证自己能够及时拥有座位，但因这种长期占座的不良现象，导致大量座位被浪费。因此我们设计和简单开发了一个可以预约座位和及时展示座位使用情况的微信小程序，以避免和减少座位浪费，尽可能使大家可以及时有效的拥有座位。

## 1.2国内研究现状

目前图书馆座位占座问题已经受到大家广泛关注，但因为每个学校占座情况不同、管理系统和方法不同，因此没有一个统一的方式方法来完成和实现座位管理问题。部分大学对此问题已经开始重视，并提出一些解决措施，例如通过小程序、APP等平台完成对座位的预约或定期通过人工对座位进行清理。

## 1.3主要工作

本系统主要分为供学生使用的微信小程序以及后端、供管理人员使用的管理前后端。该微信小程序的设计和实现中，我们先期的工作主要是根据自身需求以及借鉴和发现现有的图书馆选座系统中存在的问题。通过参考类似的解决方案，制定出需求分析；根据需求分析，做出系统的设计；再根据设计，选取适合的技术方案，完成系统的初步实现；之后，对系统进行功能测试，测试完成之后，做出论文报告。微信小程序的后端选用比较熟悉的Spring Boot框架及其扩展组件。管理人员使用的前后端在已有的基础管理框架上添加本系统所需要的功能。

在完成本项目时，我们能够学习和提高发现问题、分析问题、需求分析报告的编写等能力。在微信小程序的编写上学会原生组件的使用和利用Taro框架开发微信小程序。在后端服务的开发上深入了解Spring Boot的开发过程，学会MyBatis-Plus、Spring Security、JWT、Swagger等的使用。

## 1.4技术介绍

1、微信小程序是一种不需要下载安装即可使用的应用，它实现了应用“触手可及”的梦想，用户扫一扫或者搜一下即可打开应用。也体现了“用完即走”的理念，用户不用关心是否安装太多应用的问题。应用将无处不在，随时可用，但又无需安装卸载。对于开发者而言，微信小程序开发门槛相对较低，难度不及APP，能够满足简单的基础应用，适合生活服务类线下商铺以及非刚需低频应用的转换。微信小程序能够实现消息通知、线下扫码、公众号关联等七大功能。其中，通过公众号关联，用户可以实现公众号与微信小程序之间相互跳转。

2、Taro 是一个开放式跨端跨框架解决方案，支持使用 React/Vue/Nerv 等框架来开发微信/京东/百度/支付宝/字节跳动/ QQ 小程序/H5 等应用。现如今市面上端的形态多种多样，Web、React Native、微信小程序等各种端大行其道，当业务要求同时在不同的端都要求有所表现的时候，针对不同的端去编写多套代码的成本显然非常高，这时候只编写一套代码就能够适配到多端的能力就显得极为需要。

3、Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。

4、JWT是一种用于双方之间传递安全信息的简洁的、URL安全的表述性声明规范。JWT作为一个开放的标准（RFC 7519），定义了一种简洁的，自包含的方法用于通信双方之间以Json对象的形式安全的传递信息。因为数字签名的存在，这些信息是可信的，JWT可以使用HMAC算法或者是RSA的公私秘钥对进行签名。简洁(Compact): 可以通过URL，POST参数或者在HTTP header发送，因为数据量小，传输速度也很快；自包含(Self-contained)：负载中包含了所有用户所需要的信息，避免了多次查询数据库。

5、MyBatis-Plus（MP）是一个 MyBatis 的增强工具，在 MyBatis 的基础上只做增强不做改变，为简化开发、提高效率而生。例如在MyBatis中，每一个业务逻辑都要在DAO层写一个方法，即使是最简单的条件查询都要在改变条件时在DAO层新增一个方法。而MP使用了非常易用的CRUD接口简化上述操作，但这种接口会将数据逻辑分散到不同的层面中，如果在使用时不注意代码层次的逻辑清晰，就会导致代码难以理解和管理，故在使用时需要巧妙的平衡，既能简化代码又能保持逻辑清晰。

# 第二章 项目功能

## 2.1项目分析

在对该系统有了初步的认识之后，调研了市场上常见的相似的系统，调研了同类型产品的角色分类，权限控制，以及数据的处理，分析得出了系统分为学生和管理员两种角色，两种角色存在不同的权限。

## 2.2系统角色分析

在该系统中，学生为中心，管理员监管。学生根据座位情况对座位按需预约，并按规定签到或取消预约，对于占座、超时签到、离开座位超时不退座等违规操作，系统自动或管理员手动对其账号进行一定的管理。

### 2.2.1学生用例图

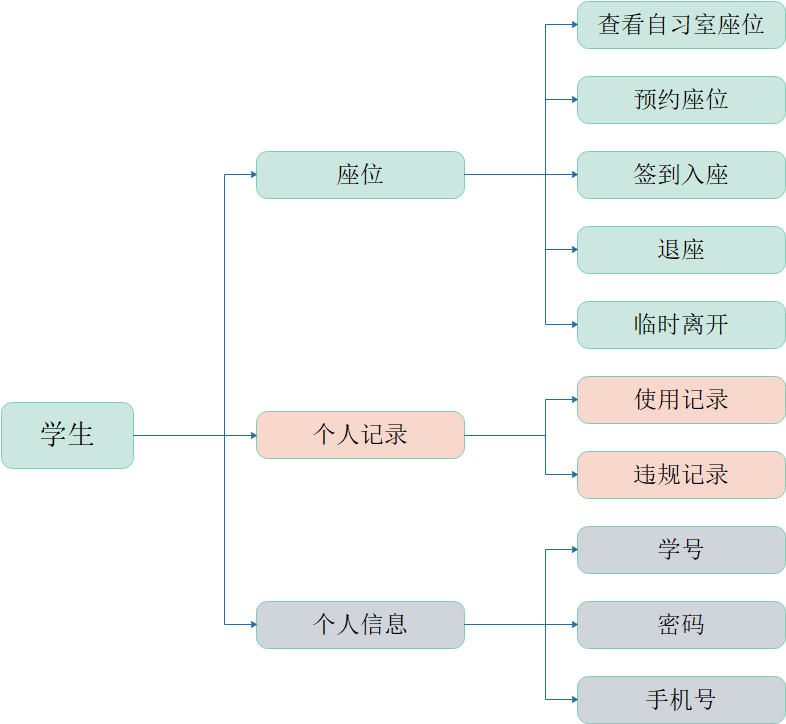


图1 学生用例图

本系统中学生用户不能主动在平台注册账号，需要由本校的图书馆管理员统一注册后，方可利用学号和初始密码正常登录账号。登录微信小程序后学生具有三个主要功能如上图所示；学生可根据当前座位使用、预约情况对空闲座位进行预约，违规预约行为会记录违规次数；预约后可进行座位的签到和退座操作，签到后若需暂时离开座位需要使用暂离功能并在规定时间内回到座位重新签到；违规次数超过规定的次数则会导致账号进入黑名单，通过相应惩罚才可继续使用平台各项功能；学生可以查看自己的违规记录和使用记录，在使用记录中可以看到座位的使用时间、位置和学习时间；违规记录中可以看到违规原因和时间。

### 2.2.2管理员

管理员不仅同学生账号一样对自身账号具有管理权利，由后端管理系统直接增添，同样对学生账号也拥有管理权利，增删查改学生账户。当学生违规行为严重时，可永久冻结其账号，在完成相应的惩罚后，在重新解冻账号恢复正常使用。

### 2.2.3账户登录

该系统中，学生进行选座时需要登录。目前的登录机制是在登录页面选座对应的学校，然后使用学号和密码进行登录。登录成功后会将后端返回的JWT保存在微信小程序的本地存储文件中，等下一次打开小程序时直接读取JWT实现自动登录。

### 2.2.4学生账号添加

因为本系统是针对不同大学图书馆而设计的通用系统，在登录时用户需要选择对应的大学。故在后端管理系统添加学生账号时，各学校的管理员只拥有管理到本校的学生账的权限，在添加上可以通过教务系统导出Excel文件然后导入到本系统中。

### 2.2.5座位管理

用户可以通过座位图表来确认空闲座位进行预约；预约成功后可以在主页的我的座位卡片中看到已完成预约的座位信息，在规定时间内可以对无法签到的座位预约进行取消；同样也可以按时达到图书馆，对预约进行签到学习，学习完离开座位时点击退座释放座位供其它人使用。

### 2.2.6通知管理

该平台的主要通知分为系统通知、学校通知和我的通知，其中我的通知主要涉及违规时系统自动发送通知。设计时管理员可根据个别用户使用情况发布“我的通知”，根据学校对图书馆相关安排发布“学校通知”，根据系统规则调整和更新发布“系统通知”。

### 2.2.9用户问题反馈

平台管理机制总会出现纰漏，因此用户可以通过问题反馈机制来反馈自己的问题或疑惑，管理员与之进行交流，解决疑惑和问题。

## 2.3功能流程图

### 2.3.1学生主要功能流程图

## 

图2 功能流程图

图书馆管理员在后端管理平台中将本校学生的账户信息批量保存在数据库中；学生在微信小程序中进行登录，验证账户信息与数据库的信息一致，即可完成登录。登录成功后进入主界面，根据学生的需求可进行预约座位、归还座位、查看所有区域座位的分布情况、查看自身预约信息和退出系统。



图1 登录界面



图2 首页

预约座位：通过学生点击列表中展示的自习室，进入作为选择界面后可以看到当前自习室中的座位分布情况和空闲状态；若存在是否有空位，则学生可以点击该作为并进行预约。当预约成功后，该区域的空位数量减少一个。



图3 选座页面

签到座位：学生在预约成功作为或者暂离座位后，需要在规定的时间内到达座位处进行签到入座。如果学生超过规定时间为进行签到则会记录违规次数，违规次数超过限制后会自动拉入黑名单并在一段时间后解除。

暂离座位：学生在入座后因某些原因需要暂时离开座位一会，可以使用该功能保留座位不被释放且不能被举报。如果超时未回来则会主动释放座位。

归还座位：学生可以在我的座位卡片中对不需要的座位进行退座操作。

查看我的座位：学生通过此功能可以看到当前的座位状态以及可以进行的操作，例如签到、归还等。



图4 座位信息

查看使用记录：学生通过该功能可以看到之前的座位信息，比如使用时间、使用时长等信息；



图5 使用记录

查看违规记录：如果学生在使用系统中出现违规行为，在此功能模块中可以看到记录信息。



图6 违规记录

# 第三章 后台管理系统

# 3.1系统简介

这是一套简单通用的后台管理系统，主要功能有：权限管理、菜单管理、用户管理，系统设置、实时日志，实时监控，API加密，以及登录用户修改密码、配置个性菜单等。我们通过简单的修改，主要实现了对于我们图书馆座位预约系统的人员管理。

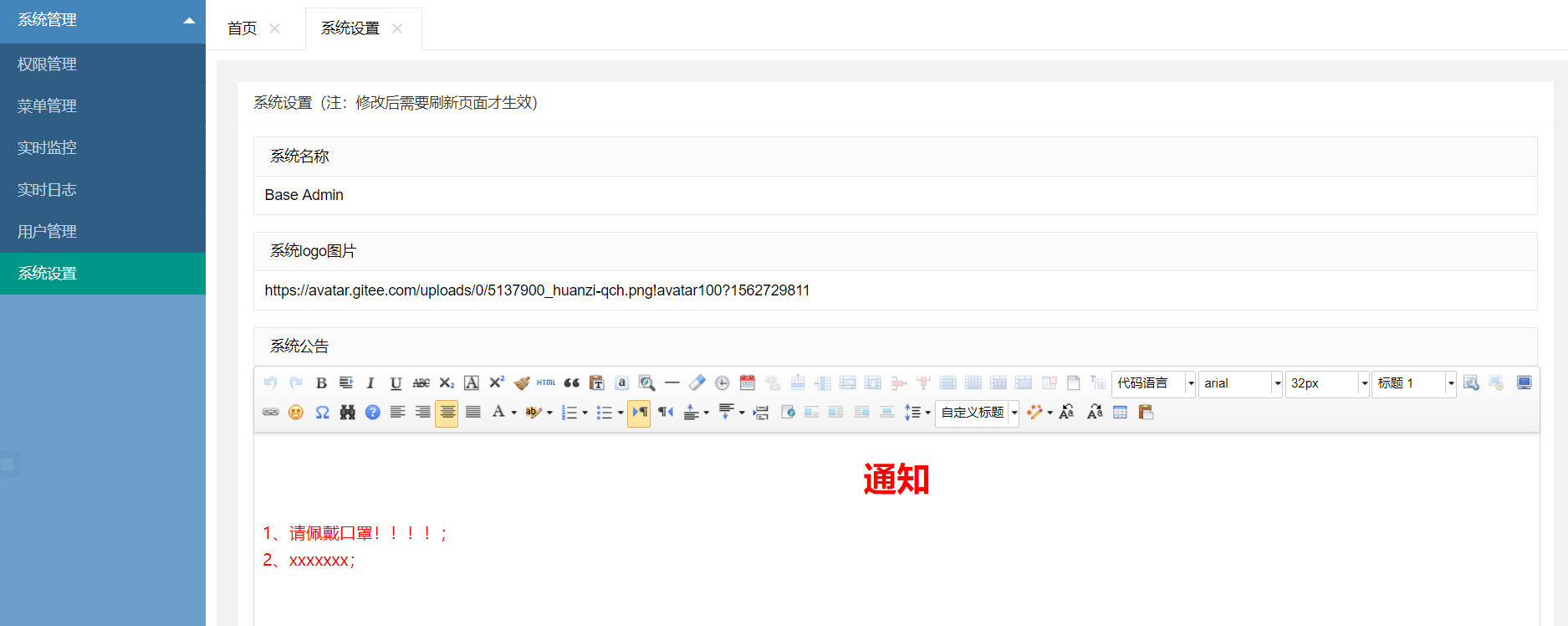
前端主要采用LayUI实现了前端界面的展示，后端主要采用了SpringBoot + Thymeleaf + Spring Security + SpringData-Jpa + MySql。

# 3.2系统演示

登录，包含了验证码并支持多种登陆限制（允许/禁止多人在线）。





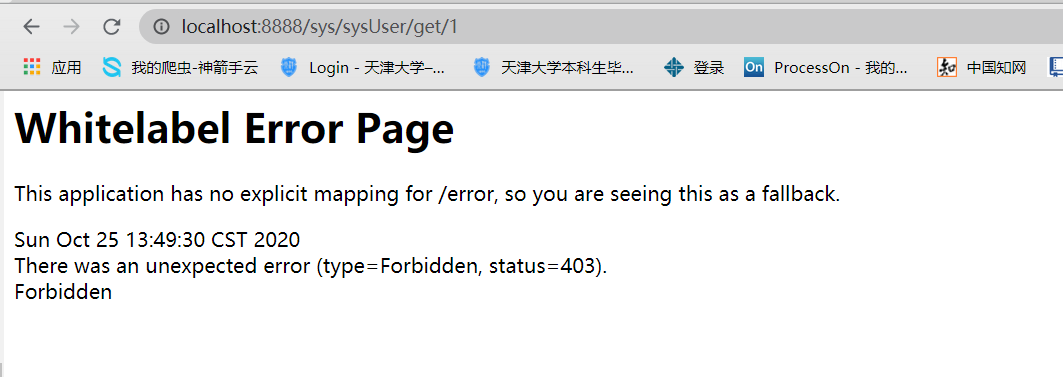
系统设置主要是一些简单的属性设置

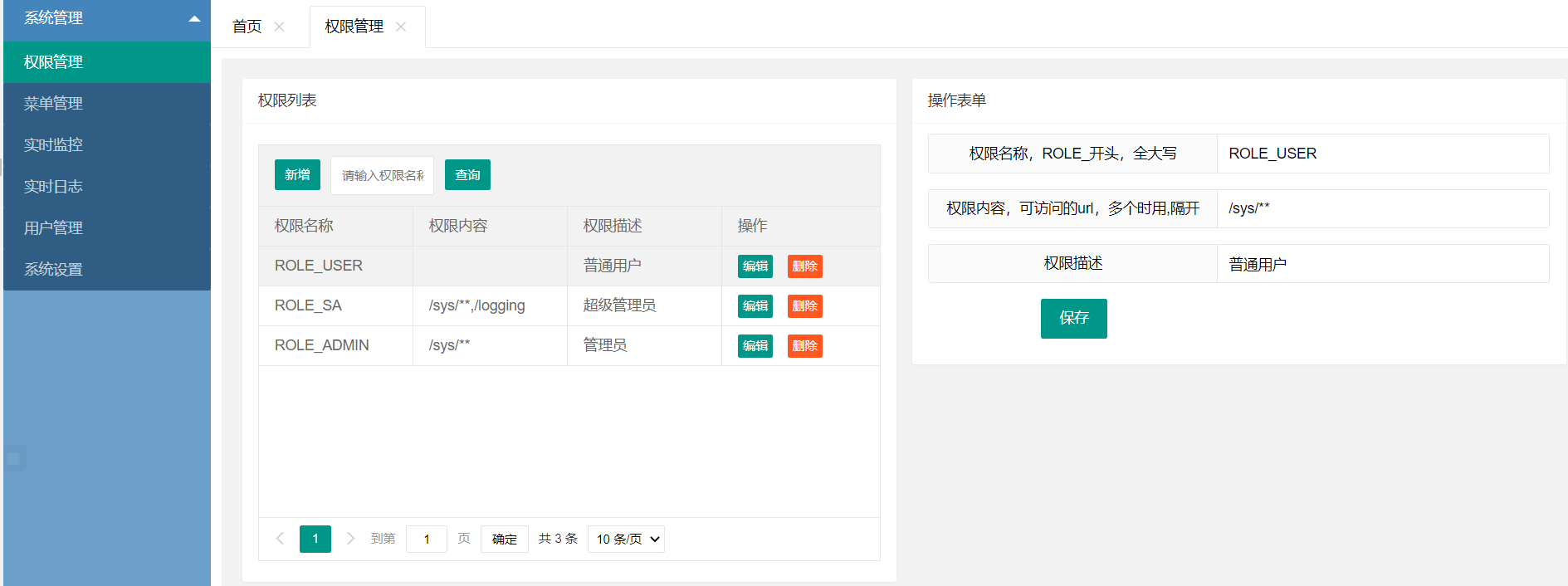
权限管理

权限的加载并不是写死在代码，而是动态从数据库读取，每次调用save方法时更新权限集合

　　1、如小王是ROLE\_USER权限，权限内容为空，无权访问/sys/下面的路径（http://localhost:8888/sys/sysUser/get/1）

　　2、使用sa超级管理员进行权限管理编辑，给ROLE\_USER的权限内容添加 /sys/\*\*，小王立即有权限访问（）







用户管理

主要包括用户信息、登录限制的维护，菜单、权限的分配等