

---

# WeChair 系统设计说明书



所属学院: 福州大学至诚学院

团队名称: WeChair

指导老师: 张栋

## 文档修订控制

序号	版本号	修订日期	修改概述	修订人	审批人	备注
1	1.0.0	2020.4.21	初步编写	组内所有成员	明锐	

---

# 目录

## 一、引言

### 1.1 编写目的

### 1.2 背景

### 1.3 参考资料

### 1.4 术语

## 二、总体设计

### 2.1 系统架构

#### 2.1.1 技术架构

#### 2.1.2 应用架构

#### 2.1.3 功能架构

### 2.2 基本设计概念和业务流程

### 2.3 外部接口

#### 2.3.1 用户接口

## 三、模块设计

### 3.1 功能

#### 3.1.1 简述各模块的功能。

#### 3.1.1 功能模块层次设计+功能模块层次图

#### 3.1.2 设计类图

### 3.2 性能

#### 3.2.1.个人信息精度（学生）

---

3.2.2 个人信息精度 (管理员)

3.2.3 座位信息精度

3.3 内部接口

3.3.1 登录验证接口

3.3.2 实名认证验证接口

3.3.3 座位状态判断接口

四、数据库设计

4.1 数据库模型

4.1.1 概念结构模型

4.1.2 逻辑结构模型

4.1.3 物理结构模型

4.2 ER 分析+表结构设计

4.3 安全保密设计

五、系统安全和权限设计

5.1 系统安全

5.1.1 系统安全控制

5.2 数据安全

5.2.1 数据用户身份鉴别

5.2.2 访问主体、访问对象的控制策略和实现方法

5.2.3 数据加密方法

5.3 备份与恢复

5.4 出错处理

---

5.4.1 列举出错情况

5.4.2 出错的形式、含义、处理方法

5.4.3 出错失效后的后备措施

5.5 权限设计

5.5.1 用户认证

5.5.2 用户授权

---

## 一、引言

### 1.1 编写目的

此系统设计说明书将全面介绍 WeChair 小程序系统的设计方案,旨在给读者全面地分析此小程序的功能、架构及安全性的问题,为预期成果打下理论基础。面向读者为用户、参与审核小程序的老师、专业投资者或投资公司、团队成员等专业性较强的个体或团体。

### 1.2 背景

近年来,随着高校招生规模的不断扩大、社会竞争的日益激烈,越来越多的学生为了应付各类考试而不断涌向图书馆看书备考,以提高自己的学习能力。图书馆为了更好地配合学校的教学工作,完善读者服务,面临着多方压力。因此实现座位资源计算机网络化管理、对图书馆的座位进行合理调配、提高座位资源的利用率、成为了“WeChair”提供图书馆优质服务给大家的动力。

“WeChair”是一款基于微信小程序的图书馆座位管理系统,主要原理是通过座位码绑定图书馆每一张座位。用户仅需使用微信扫码直接获取每一张座位的使用信息,就可以知道该座位的使用情况。与传统技术相比,操作流程简单,不需要下载 APP,不需要输入账号密码,不泄露个人隐私信息,交互双方座位信息同步,信息完全对等,直击占座问题要点。不仅有效的减少了学生的占座现象,同时也提高了图书馆资源利用率。以下将介绍“WeChair”的具体系统结构设计。

目前 WeChair 项目已经处于初级阶段,整体的设计、布局、架构已经准备就绪,前端、后端、数据库负责人都已经投入编写。此前在“软件工程”课堂上,团队成员有一起协同完成一个信息录入系统网站,这将给我们接下来的工作起到了不可或缺的作用,接下来我们将围绕系统设计说明书对我们的小程序进行制作与完善。由于我们是课程团队,可靠性和专业性可能达不到预期要求,但我们将努力做到最好,希望读者若发现不足能够及时提出宝贵意见。

### 1.3 参考资料

技术标准: (微信小程序设计指南)

<https://developers.weixin.qq.com/miniprogram/design/#%E5%8F%8B%E5%A5%BD%E7%A4%BC%E8%B2%8C>

参考资料:

[1]徐家骅, 白忠建, 吴磊, 软件工程——理论与实践 (第二版), [J]. 高等教育出版社, 2009.10.

[2]栾跃, 软件开发项目管理[M], 上海, 上海交通大学出版社, 2015, 20-34.

[3]段琳琳, 敏捷方法在需求工程中的研究与应用[D].湖南大学, 2018.1.

查阅资料:

[1] 微信小程序设计规范总结:

<https://www.jianshu.com/p/78c93889410f>

[2] 做一个微信小程序的完整流程:

<https://blog.csdn.net/bryant0917/article/details/82470433>

[3] 小程序文档整理之——开放接口:

<https://blog.csdn.net/bobobocai/article/details/77651566>

[4] 详解设计模式六大原则:

<https://www.cnblogs.com/toutou/p/4870926.html>

[5] 小程序开发之数据库备份:

<https://cloud.tencent.com/developer/article/1519863>

## 1.4 术语

列出本系统设计说明书中专门术语的定义和英文缩写词的原词组。

表 1-1 专门术语表

术语名称	术语含义	备注
应用程序接口	预先定义的函数或指软件系统不同组成部分衔接的约定	Application Programming Interface
单一职责原则	应该有且仅有一个原因引起类的变更	Single Responsibility Principle
里氏替换原则	任何基类可以出现的地方, 子类一定可以出现	Liskov Substitution Principle
依赖倒置原则	程序要依赖于抽象接口, 不要依赖于具体实现	Dependence Inversion Principle
接口隔离原则	不出现臃肿的接口	Interface Segregation Principle, ISP

迪米特法则	一个软件实体应当尽可能少的与其他实体发生相互作用	Law of Demeter, LOD
开闭原则	模块应尽量在不修改原代码的情况下进行扩展	
WXML	是框架设计的一套标签语言	WeiXin Markup Language
WXSS	一套样式语言，用于描述 WXML 的组件样式	WeiXin Style Sheets

## 二、总体设计

### 2.1 系统架构

#### 2.1.1 技术架构

参考 MVC 设计模式将系统分为三层，展示层、业务层、数据层，画出相关技术架构图。层与层之间又通过一定的模式联系，使数据实体、业务逻辑与呈现视图分离，同时降低耦合性、提高重用性和可维护性。

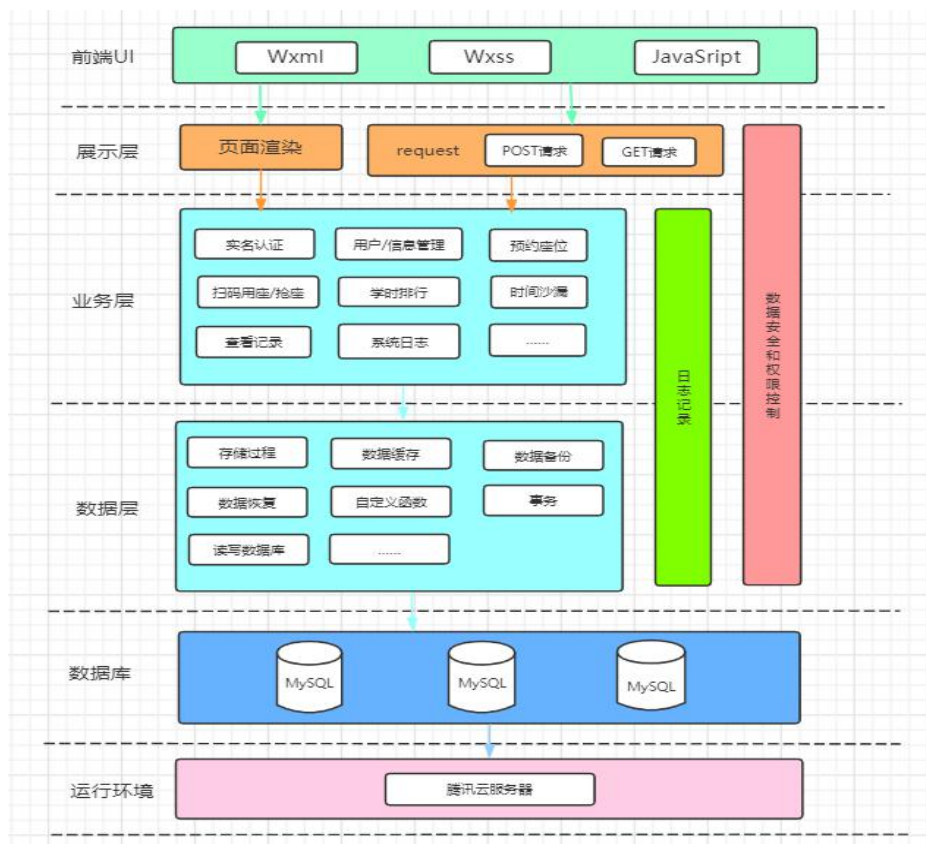


图 2-1 MVC 技术架构图

## 2.1.2 应用架构

根据前后端交互及应用部署的环境结构，画出系统应用架构图。

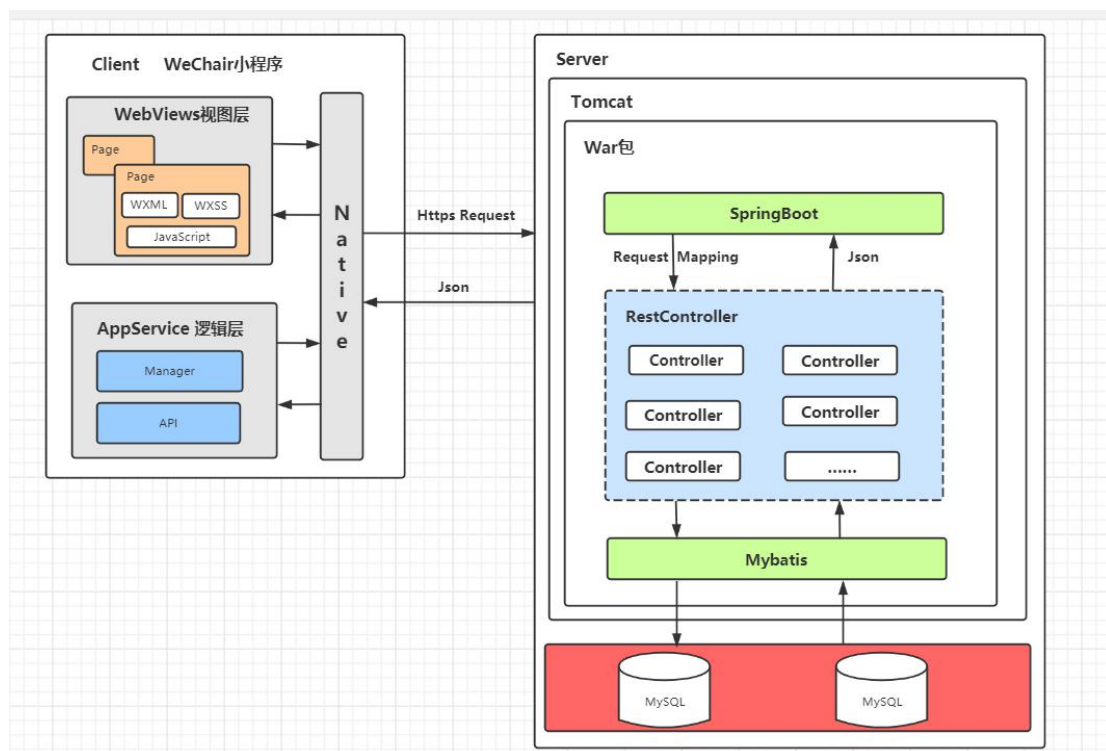


图 2-2 WeChair 系统架构图

## 2.1.3 功能架构

通过用例图展示系统的功能架构

## 2.3 基本设计概念和业务流程

小组设计的活动图更直观的展示出 WeChair 的基本设计和一整套业务流程。



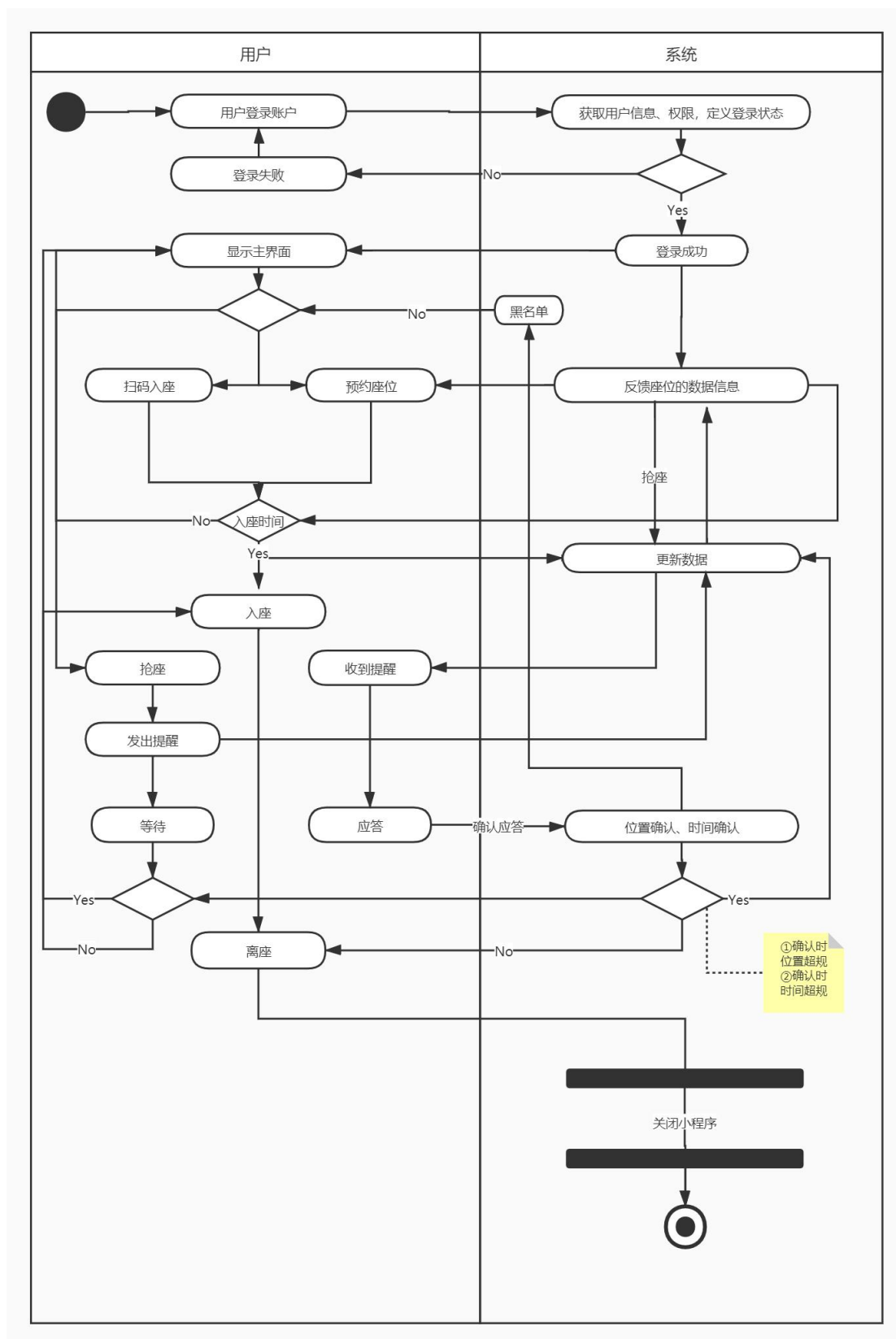


图 2-3 活动图

## 2.4 外部接口

### 2.3.1 用户接口

重新设计后的原型界面

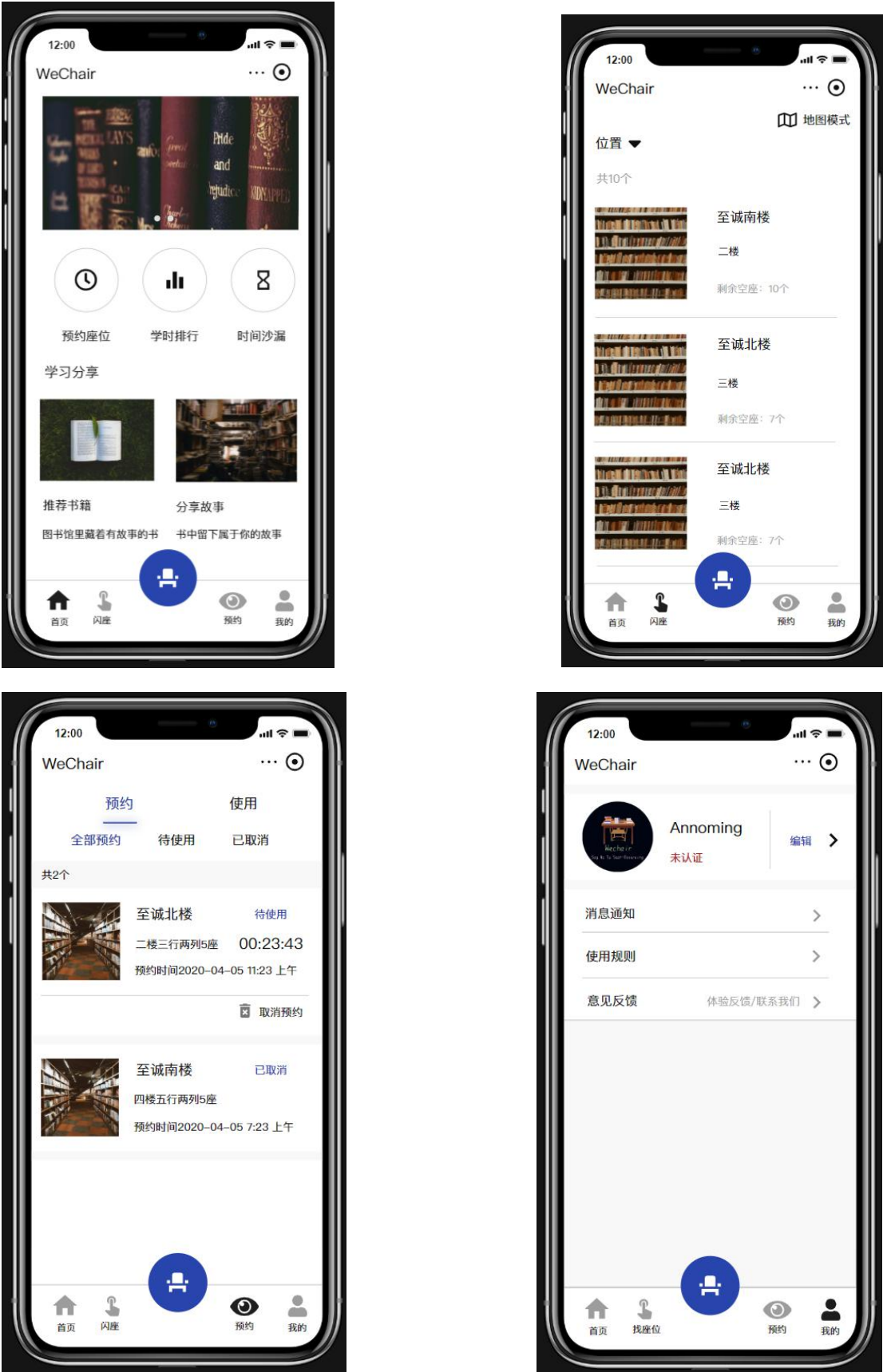


图 2-4 小程序原型设计

---

(1)小程序登录模块与微信后台的接口:

授权接口 `wx.getSetting`

获取登录码的接口 `wx.login`

获取用户信息的接口 `wx.getUserInfo`

(2)小程序扫码访问手机摄像头或相册接口:

```
wx.scanCode({  
  
  //调用成功则放回调用结果  
  
  success (res) {  
  
    //res.data 能够获取座位的号码  
  
  }  
  
})
```

(3)小程序获取用户地理位置接口:

```
wx.getSetting({  
  success(res) {  
    if (!res.authSetting['scope.userLocation']) {  
      wx.authorize({  
        scope: 'scope.userLocation',  
        success () {  
          // 用户已经同意小程序使用定位，后续调用 wx.getLocation 接口不会弹窗询问  
          wx.getLocation({  
            type: 'wgs84',  
            success (res) {  
              //获取用户的地理位置  
              const latitude = res.latitude  
              const longitude = res.longitude  
              const speed = res.speed  
              const accuracy = res.accuracy  
            }  
          }  
        }  
      })  
    }  
  }  
})
```

---

## 三、模块设计

### 3.1 功能

#### 3.1.1 简述各模块的功能。

经过重新设计和修改后 WeChair 小程序包括六大模块，将详细介绍抢座功能：

##### (1)预约座位模块：

通过学生认证的用户能够实时地对座位进行预约，点击首页的预约座位会跳转到闪座的页面，能够查看图书馆的座位分布情况和空座情况，有地图模式和列表模式两种查看模式，用户选择好一个座位后便可以进行预约，预约成功的座位用户需三十分钟内进入图书馆扫码上座，同时预约状态的座位不会被抢座抢走。

##### (2)学时排行模块：

通过学生认证的用户，可以选择是否参加学时排行，参加学时排行的用户能在首页点击学时排行进入查看自己的日学习时长和周学习时长排行。

##### (3)时间沙漏模块：

通过学生认证的用户，系统会记录用户进入图书馆学习的时长，同时对用户每周和每月的学习时长绘制时间曲线，帮助用户清晰地了解自己某一阶段进入图书馆学习的情况。

##### (4)用座/抢座模块：

通过点击小程序导航栏中间部分的按钮进入扫码界面。

1) 选择用座模式进行扫码，扫码座位未被预约和使用即可使用该座位，反之，将会跳转至闪座页面，帮助用户查看图书馆中的座位分布和空座分布，方便用户找到空闲状态的座位进行扫码上座。

2) 选择抢座模块，必须是用户没有正在使用的座位的情况下才能抢座，所抢座位必须为无人使用但是小程序上显示正在使用，一个座位同时只会被一个人发起抢座。

a. 如果发起抢座的座位为其他用户预约的座位但是无人的座位，预约发生时间(从开始预约到抢座触发的时长)二十分钟之内扫码抢座会给抢座用户显示预约发生时间但是无法进行抢座，预约发生时间超过二十分钟其他用户可以扫码发起抢座。被抢座位的用户会收到跟小程序绑定公众号发送的信息，并提醒用户进入小程序查看并获取用户定位判断用户是否会在不超过有效预约时间内来到图书馆扫码用座。如果可以到达图书馆会给抢座用户反馈抢座失败，如果不能及时到达将抢座用户抢座成功，并反馈信息

给预约的用户，并记录一次无效预约。抢座成功的用户需要在十分钟内扫码上座。如果预约座位的用户未注意到消息或者故意不理睬消息，五分钟后将抢座用户抢座成功，反馈信息给预约用户，并记录一次无效预约。

b. 如果抢座座位为正在使用但是无人的座位，被抢座位的用户会收到跟小程序绑定公众号发送的信息，并进行提醒用户进入小程序查看，进入小程序的用户会被提醒十五分钟内返回座位扫码继续使用座位或者用户离开图书馆超过一定范围而被直接释放。如果用户未注意到消息或者故意不理睬消息，十五分钟后该座位将被释放给抢座用户，抢座成功的用户需在十分钟内扫码上座。

### (3) 查看预约/使用模块:

通过学生认证的用户才使用预约座位功能, 才能进入预约页面使用查看预约和取消预约等功能，预约页面方便地提供给用户查看所有预约、当前预约和取消预约等功能。预约页面的查看历史使用座位、正在使用、已经使用等功能所有用户都可以使用。

### (4) 个人信息模块:

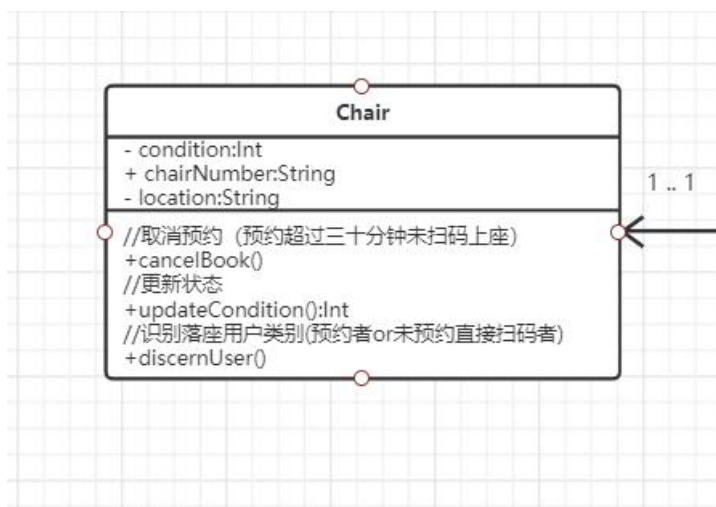
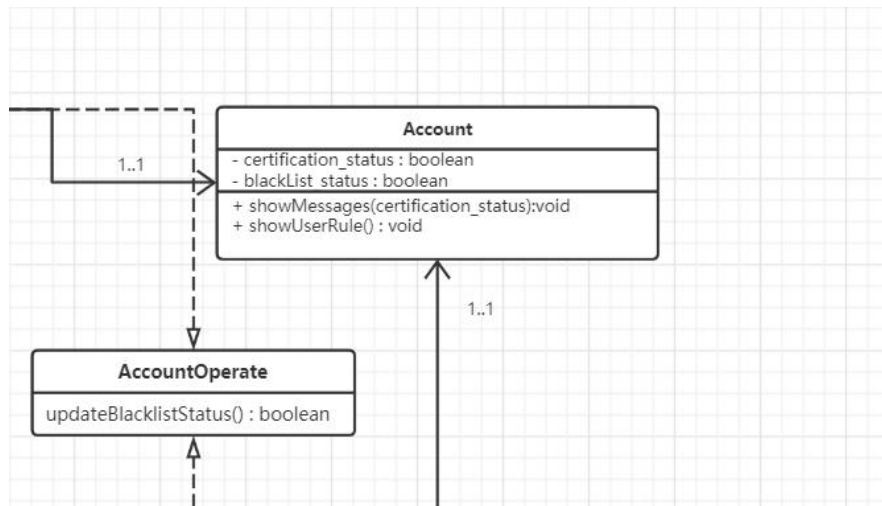
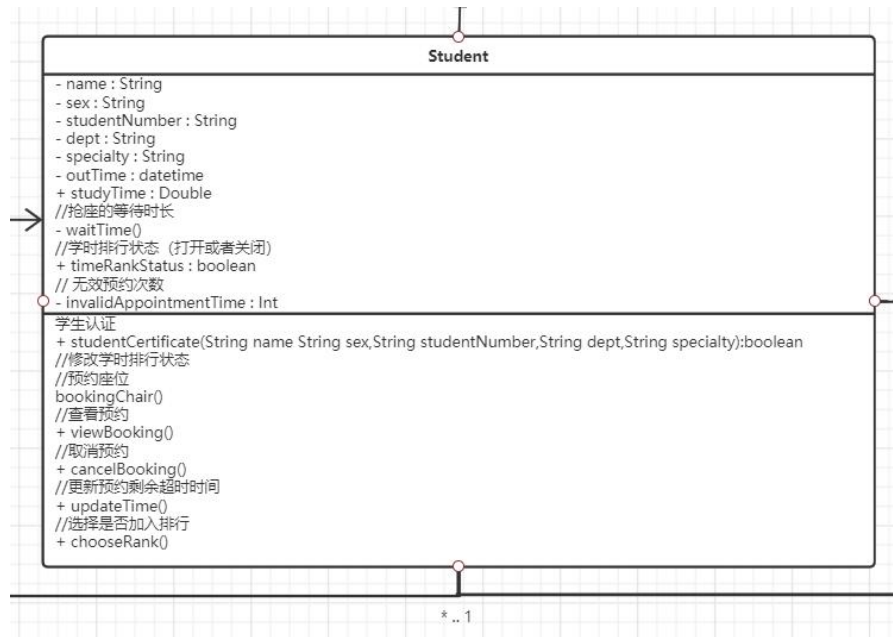
个人信息模块能查看 WeChair 小程序的使用规则、能提供反馈，个人资料页面能修改用户名、修改头像以及最重要的学生认证。

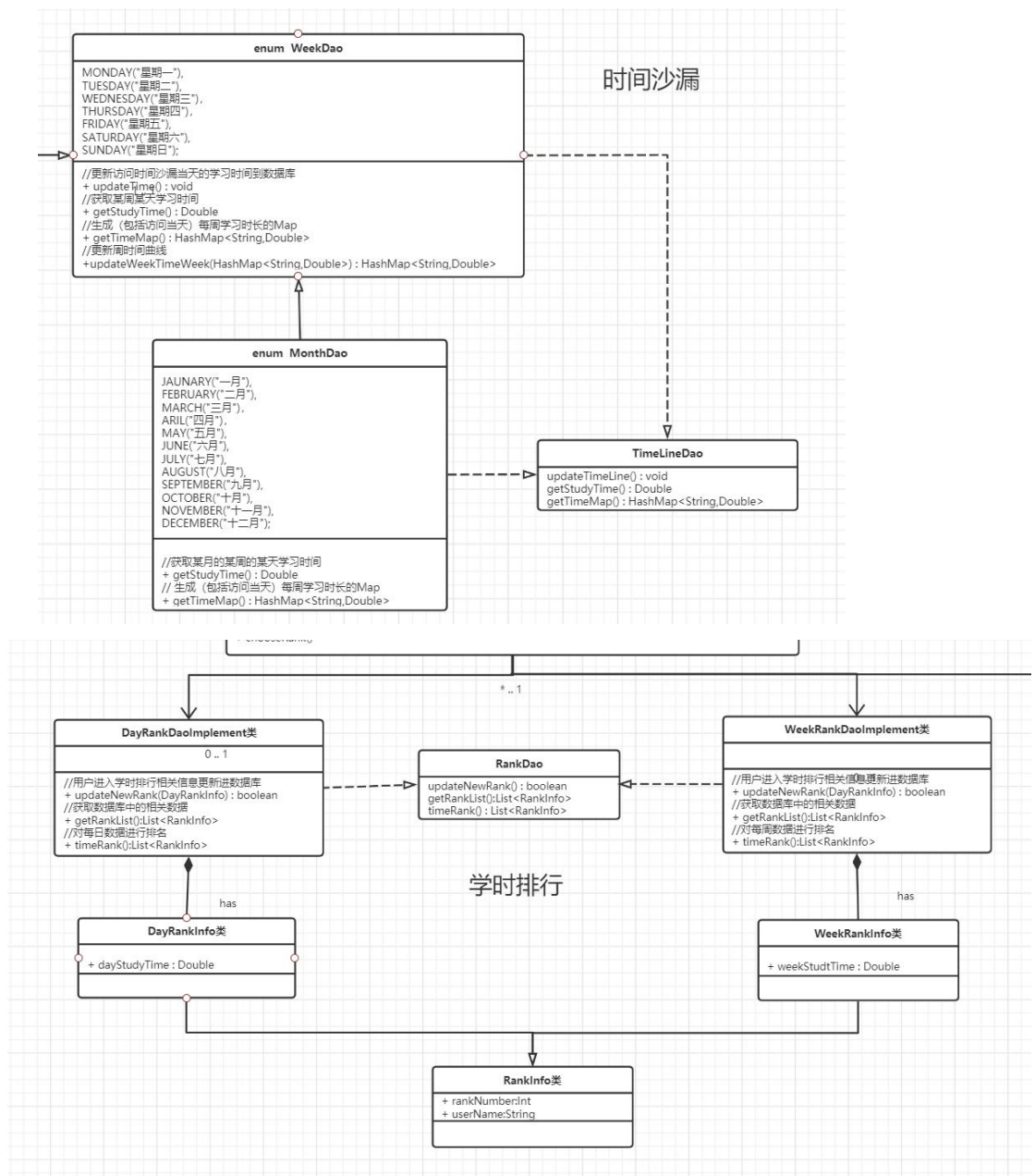
## 3.1.1 功能模块层次图

## 3.1.2 类图设计

由于类图设计的大小不方便整张图展示，对设计的类图进行拆分展示。







## 3.2 性能

### 3.2.1.个人信息精度（学生）：

- 用户名：9 位数字（学号），不可为空。INT
- 用户密码：密码由 6-12 个字符组成。密码只能由数字、英文字符、下划线组成。不可为空。
- 用户昵称：长度不超过 20 个字符，不包含中文、大写英文、小写英文、数字、下划线及其组合以外的特殊字符。不可为空。
- 用户性别：单选框，只有“男”、“女”两种选择。不可为空。



- 
- 用户手机号码：只允许 11 位的数字组成。可为空。
  - 用户邮箱：符合邮箱的正则表达式。包含@字符。可为空。
  - 经纬度：decimal(10, 7)，不可为空。
  - 座位：参考座位信息精度。
  - 预约时间：DATE 类型。可为空。
  - 学习时长及排名：参考学习时长及排行榜精度。

### 3.2.2 个人信息精度（管理员）：

- 用户名：x 位数字（工号），不可为空。
- 用户密码：密码由 6-12 个字符组成。密码只能由数字、英文字符、下划线组成。不可为空。
- 用户昵称：长度不超过 20 个字符，不包含中文、大写英文、小写英文、数字、下划线及其组合以外的特殊字符。不可为空。
- 用户手机号码：只允许 11 位的数字组成。可为空。
- 用户邮箱：符合邮箱的正则表达式。包含@字符。可为空。

### 3.2.3 座位信息精度：

- 区域：单选框，只有“南楼”、“北楼”两种选择。不可为空。
- 楼层：单选框，只有“1 楼”、“2 楼”、“3 楼”、“4 楼”、“5 楼”五种选择。不可为空。
- 服务项目：单选框，只有“图书馆阅览一室”、“图书馆阅览二室”、“图书馆阅览三室”、“报纸、期刊阅览室”、“自修室”五种选择。不可为空。
- 座位号：1 号到 n 号（n：视服务项目而定），INT，不可为空。
- 座位标识：“0”代表未使用，“1”代表正在使用，“2”代表已预约。不可为空。
- 经纬度：decimal(10, 7)，不可为空。

### 3.2.4：学习时长及排行榜精度：

- 日期：DATE 类型，自动生成。不可为空。
- 时长：TIME 类型，自动生成。不可为空。
- 排行榜名次：整型，自动生成，正整数。不可为空。
- 用户昵称：根据个人信息（学生）生成。不可为空。



### 3.3 内部接口

#### 3.3.1 登录验证接口:

www\login\_check.jsp

在验证登录时，调用该接口，传入的参数为用户唯一标识 **userid** 和 **session** 会话维持

表 3-1

参数名	类型	长度	备注
Userid	String	255 字节	必填
Session	String	255 字节	必填

如果用户存在且会话维持码正确返回 111，如果用户存在但还是会话维持码不正确返回 112，如果用户不存在则返回 113。

#### 3.3.2 实名认证验证接口:

在进行实名认证验证时，调用该接口，传入的参数为用户唯一标识 **userid**

www\identify\_judeg.jsp

表 3-2

参数名	类型	长度	备注
Userid	String	255 字节	必填

如果用户未验证且会话维持码正确返回 131，如果用户验证后但还是会话维持码不正确返回 132。

#### 3.3.3 座位状态判断接口:

www\seat\_judeg.jsp

该接口为座位状态判断接口，传入的参数为座位号、用户唯一标识 **userid**

表 3-3

参数名	类型	长度	备注
Seat_num	Int	11 字节	必填
Userid	String	255 字节	必填

再扫二维码的情况下，如果座位是自己预约的或自己进行暂离的，则返回 120，如果座

位被预约了，但是不是自己预约的返回 121，如果座位的状态是暂离的，则返回 122。在不是扫码的情况下，已经被预约返回 123。

## 四、数据库设计

### 4.1 数据库模型

#### 4.1.1 概念结构模型

学生信息(用户 ID、学号、是否删除、姓名、性别、昵称、密钥、手机号码、邮箱、经纬度)

其中属性约束如下：是否删除：0=否，1=是

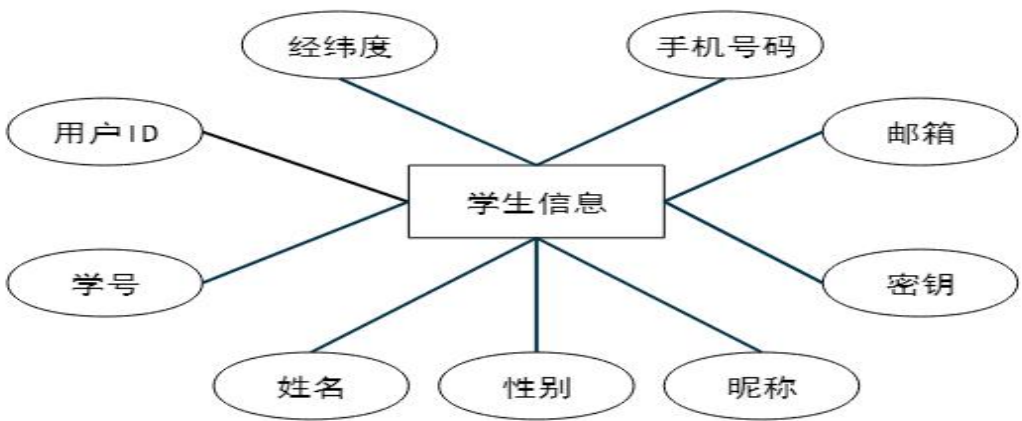


图 4-1 学生信息组成

管理员信息（用户 ID、工号、是否删除、姓名、性别、昵称、密钥、手机号码、邮箱）

其中属性约束如下：是否删除：0=否，1=是

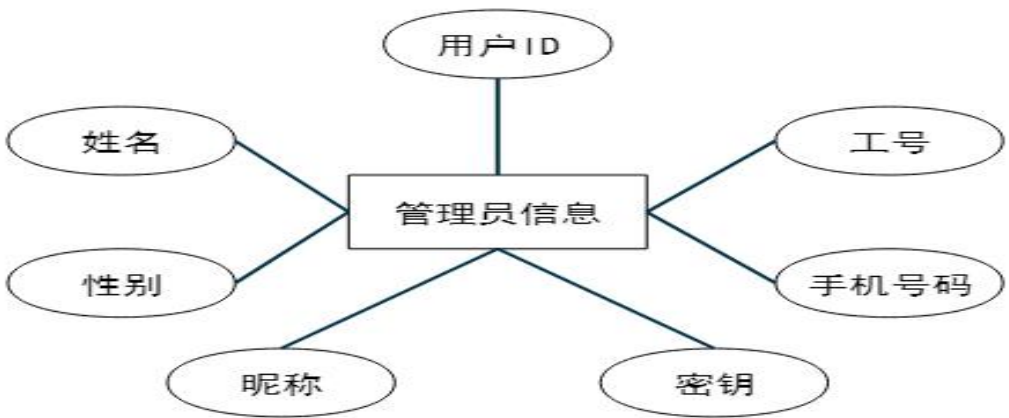


图 4-2 管理员信息组成

---

座位（区域、楼层、服务项目、座位号、状态标识、使用人）

其中属性约束如下：

状态标识：0=未使用，1=正在使用，2=已预约，3=暂离

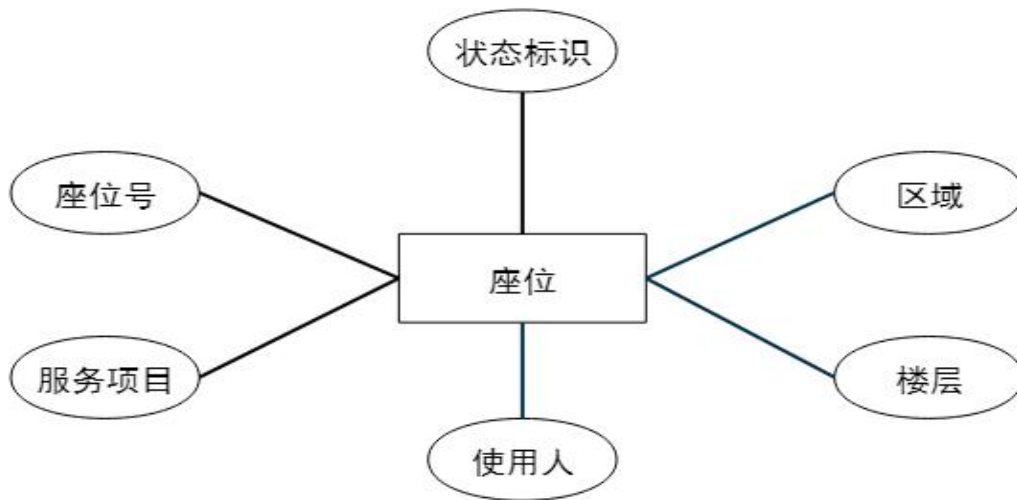


图 4-3 座位信息组成

使用记录（用户 ID、日期、上座时间、下座时间、使用时间、区域、楼层、服务项目、座位号）

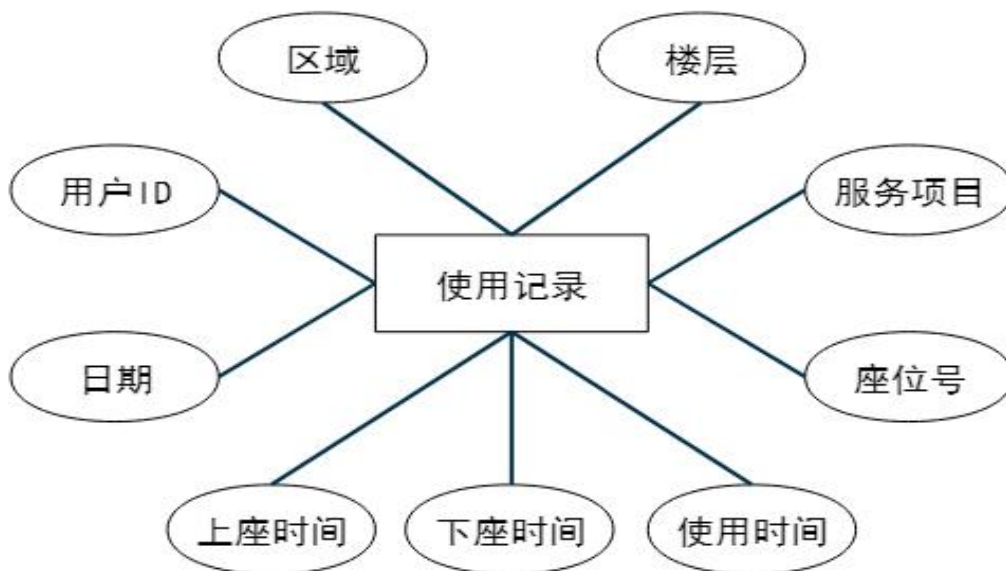


图 4-4 使用记录组成

预约记录表（用户 ID、申请时的日期、申请时的时间、预约的日期、预约的时间、预约状态、区域、楼层、服务项目、座位号）

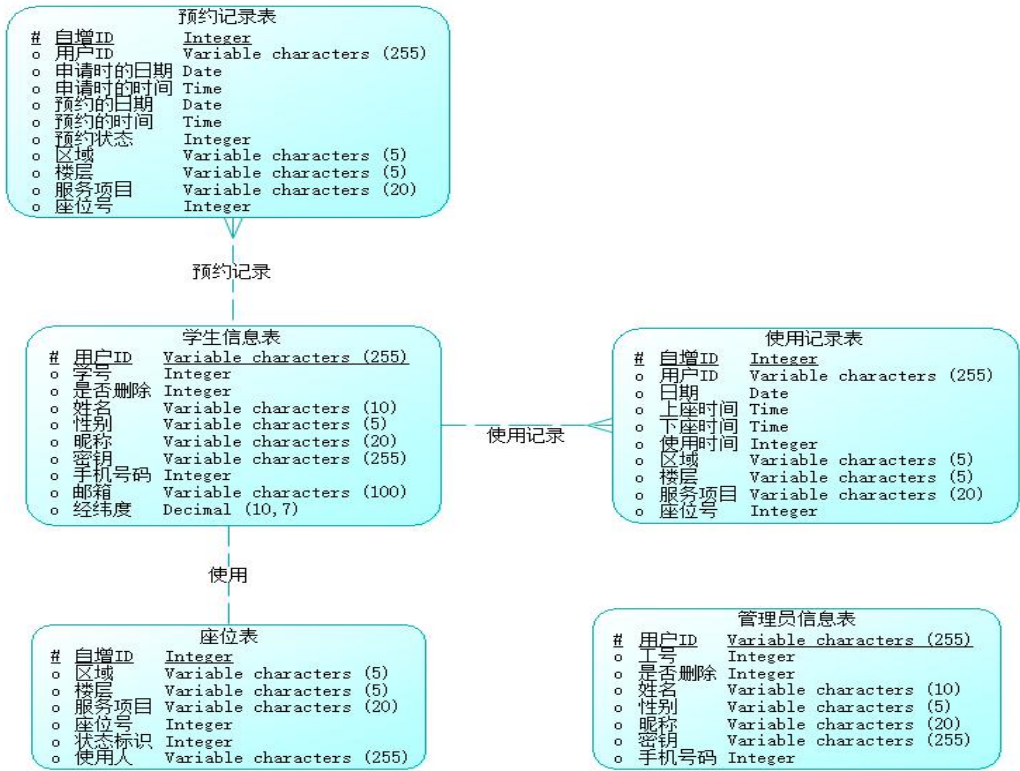
其中属性约束如下：

预约状态：0=预约成功，1=已上座，2=已取消，3=违约



图 4-5 预约记录组成

### 4.1.2 逻辑结构模型



4.1.3 物理结构模型

表 4-6 学生信息表

	Name	Code	Data Type	Lengt	Preci	P	F	M
1	用户ID	UserID	varchar(255)	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	学号	Sno	int(10)	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	是否删除	IsDelete	int(8)	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	姓名	Name	varchar(10)	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	性别	Gender	varchar(5)	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	昵称	Nickname	varchar(20)	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	密钥	Key	varchar(255)	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	手机号码	Mobile	int(12)	12		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	邮箱	Email	varchar(100)	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	经纬度	Location	decimal(10,7)	10	7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

表 4-7 管理员信息表

	Name	Code	Data Type	Lengt	Preci	P	F	M
1	用户ID	UserID	varchar(255)	255		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	工号	AdminID	int(10)	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	是否删除	IsDelete	int(8)	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	姓名	Name	varchar(10)	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	性别	Gender	varchar(5)	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	昵称	Nickname	varchar(20)	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	密钥	Key	varchar(255)	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	手机号码	Mobile	int(12)	12		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	邮箱	Email	varchar(100)	100		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

表 4-8 座位表

	Name	Code	Data Type	Lengt	Preci	P	F	M
1	自增ID	ID	int(15)	15		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	区域	Region	varchar(5)	5		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	楼层	Floor	varchar(5)	5		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	服务项目	Service	varchar(20)	20		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5	座位号	SeatID	int(200)	200		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	状态标识	Mark	int(8)	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	使用人	UserID	varchar(255)	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

表 4-9 使用记录表

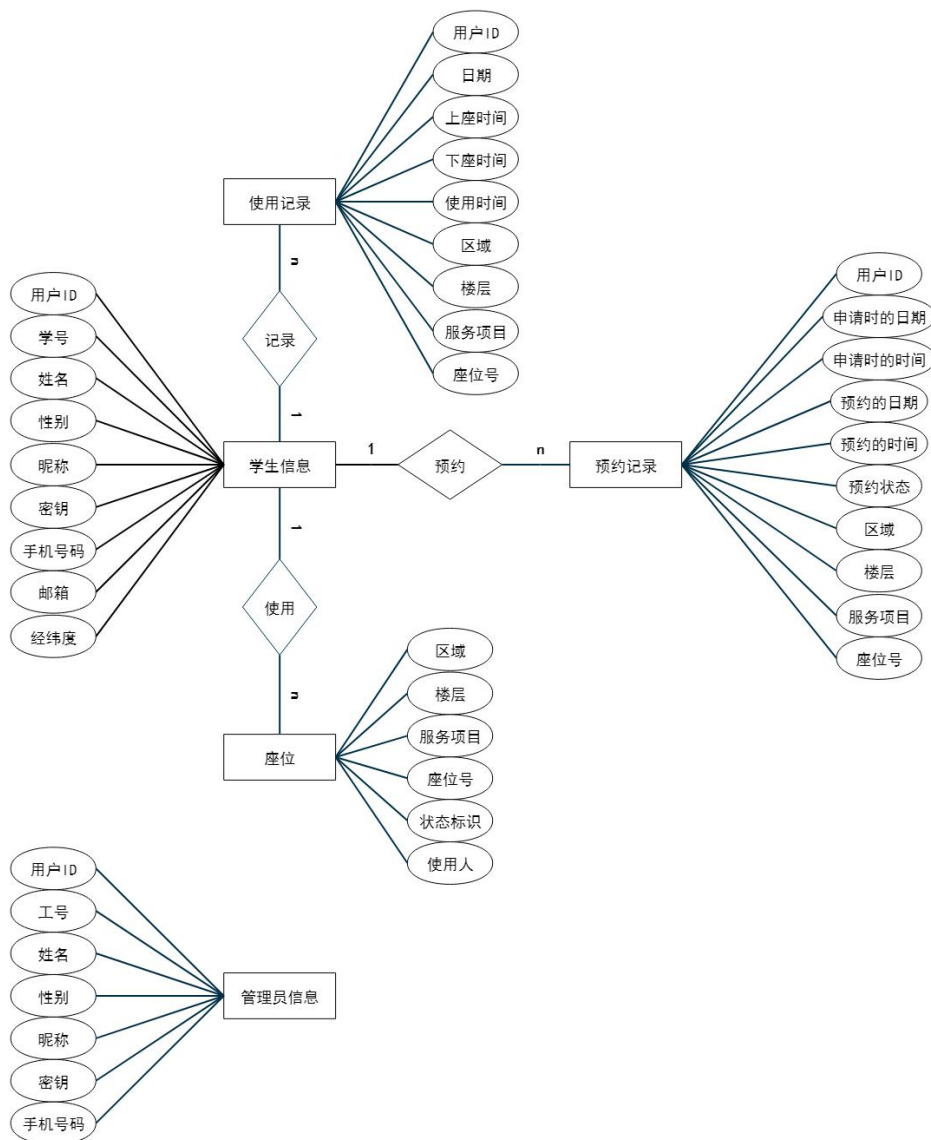
	Name	Code	Data Type	Lengt	Preci	P	F	M
1	自增ID	ID	int(15)	15		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	用户ID	UserID	varchar(255)	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	日期	Day	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	上座时间	Start	time			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	下座时间	End	time			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	使用时间	UseTime	int(1000)	1,000		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	区域	Region	varchar(5)	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	楼层	Floor	varchar(5)	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	服务项目	Service	varchar(20)	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	座位号	SeatID	int(200)	200		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>



表 4-10 预约记录表

	Name	Code	Data Type	Lengt	Preci	P	F	M
1	自增ID	ID	int(15)	15		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	用户ID	UserID	varchar(255)	255		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	申请时的日期	ApplyDay	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	申请时的时间	ApplyTime	time			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	预约的日期	AimDay	date			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	预约的时间	AimTime	time			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	预约状态	Mark	int(8)	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8	区域	Region	varchar(5)	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9	楼层	Floor	varchar(5)	5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10	服务项目	Service	varchar(20)	20		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
11	座位号	SeatID	int(200)	200		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## 4.2 ER 分析+表结构设计



学生信息表记录学生登陆账号和联系方式等基本信息。

表 4-11 Student 的结构

Code	Data Type	Length	Precision	Primary	Mandatory	Description
UserID	varchar(255)	255		TRUE	TRUE	用户ID
Sno	int(10)	10				学号
IsDelete	int(8)	8				是否删除 (0=否, 1=是)
Name	varchar(10)	10				姓名
Gender	varchar(5)	5				性别
Nickname	varchar(20)	20				昵称
Key	varchar(255)	255				密钥
Mobile	int(12)	12				手机号码
Email	varchar(100)	100				邮箱
Location	decimal(10,7)	10	7			经纬度

管理员信息表记录管理员登陆账号和联系方式等基本信息。

表 4-12 Admin 的结构

Code	Data Type	Length	Precision	Primary	Mandatory	Description
UserID	varchar(255)	255		TRUE	TRUE	用户ID
AdminID	int(10)	10				工号
IsDelete	int(8)	8				是否删除 (0=否, 1=是)
Name	varchar(10)	10				姓名
Gender	varchar(5)	5				性别
Nickname	varchar(20)	20				昵称
Key	varchar(255)	255				密钥
Mobile	int(12)	12				手机号码
Email	varchar(100)	100				邮箱

座位表存储座位位置及使用情况等基本信息。

表 4-13 Seat 的结构

Code	Data Type	Length	Precision	Primary	Mandatory	Description
ID	int(15)	15		TRUE	TRUE	自增ID
Region	varchar(5)	5				区域
Floor	varchar(5)	5				楼层
Service	varchar(20)	20				服务项目
SeatID	int(200)	200				座位号
Mark	int(8)	8				状态标识(0=未使用, 1=正在使用, 2=已预约, 3=暂离)
UserID	varchar(255)	255				使用人

使用记录表存储使用座位等基本信息。

表 4-14 Record 的结构

Code	Data Type	Length	Precision	Primary	Mandatory	Description
ID	int(15)	15		TRUE	TRUE	自增ID
UserID	varchar(255)	255				用户ID
Day	date					日期
Start	time					上座时间
End	time					下座时间
UseTime	int(1000)	1,000				使用时间
Region	varchar(5)	5				区域
Floor	varchar(5)	5				楼层
Service	varchar(20)	20				服务项目
SeatID	int(200)	200				座位号

预约记录表存储预约等基本信息。

表 4-15 Reservation 的结构

Code	Data Type	Length	Precision	Primary	Mandatory	Description
ID	int(15)	15		TRUE	TRUE	自增ID
UserID	varchar(255)	255				用户ID
ApplyDay	date					申请时的日期
ApplyTime	time					申请时的时间
AimDay	date					预约的日期
AimTime	time					预约的时间
						预约状态(0=预约成功, 1=已上座, 2=已取消, 3=违约)
Mark	int(8)	8				
Region	varchar(5)	5				区域
Floor	varchar(5)	5				楼层
Service	varchar(20)	20				服务项目
SeatID	int(200)	200				座位号

## 4.3 安全保密设计

通过区分不同的访问者、不同的访问类型和不同的数据对象，进行分别对待而获得的数据库安全保密设计考虑。数据库由专门数据库管理员对数据库操作。管理员权限最大，可以控制所有数据。

# 五、系统安全和权限设计

## 5.1 系统安全

### 5.1.1 系统安全控制

(1) ip 段的限制:

在程序中设置限制，当出现大批量某一字段 ip 访问部署在服务器上的域名时，为了防止被攻击，对这种类似 ip 进行禁止访问。

(2) 登录时间段限制:

由于图书馆有开馆时间限制，为了和图书馆不存在矛盾，登录时间段限制在早上八点到晚上九点。

(3) 程序资源访问控制安全:

对程序资源的访问进行安全控制，在客户端上，为用户提供和其权限相关的用户界面，仅出现和其权限相符的菜单，操作按钮；在服务端则对 URL 程序资源和业务类方法的调用进行访问控制。



---

### 5.1.2 用户身份鉴别

使用 WeChair 小程序的用户本身是微信的用户，微信的用户具有 openid 这个身份的唯一标识，而小程序的用户需要进行学生认证，学号也会作为区别不同用户的标识。

### 5.1.3 用户对系统的访问权限和范围

用户分为普通用户和进行学生认证成功的学生用户。

普通用户只能使用系统的扫码上座、抢座、以及查看座位使用记录和使用个人信息页等功能。

通过学生认证的学生用户在不违背系统规则情况下可以使用系统的任何功能。

对系统的使用范围包括一个时间范围：及早上八点到晚上九点期间才能使用系统，其他时间系统都处于停止服务状态。

## 5.2 数据安全

### 5.2.1 数据用户身份鉴别

使用数据库中数据的用户指管理员，管理员的身份通过学工号来进行鉴别，会提前在数据库中建立管理员表，使用小程序的管理员进行身份认证时会同数据库中信息比对。

### 5.2.2 访问主体、访问对象的控制策略和实现方法

采用基于角色的访问控制模型，即 RBAC 模型，对用户进行划分角色：管理员、学生用户和普通用户，不同角色拥有不同的访问许可权。为了更好地管理用户，对用户进行分组归类，简称为用户分组。用户可以自己拥有权限信息，可以归属于  $0 \sim n$  个角色，可属于  $0 \sim n$  个组。权限集是自身具有的权限、所属的各角色具有的权限、所属的各组具有的权限的合集。它与权限、角色、组之间的关系都是  $n$  对  $n$  的关系。

### 5.2.3 数据加密方法

小程序需要进行保护的数据主要是进行学生认证所填写的学生信息。我们对数据进行加密的采用的方法是，加密用 aesUtil.js 和 md5.js 源码，在 pulic.js 中进行加解密操作，其中密钥和密钥偏移量要与后台的一致。

## 5.3 备份与恢复

### 5.3.1 数据备份

通过查阅微信的文档，可以发现云开发提供了数据导出接口 databaseMigrateExport

---

POST

[https://api.weixin.qq.com/tcb/databasemigrateexport?access\\_token=ACCESS\\_TOKEN](https://api.weixin.qq.com/tcb/databasemigrateexport?access_token=ACCESS_TOKEN)

通过这个接口，结合云函数的定时触发功能，我们就可以做数据库定时自动备份了。大致的流程：

创建一个定时触发的云函数

方法调用接口，导出数据库备份文件

将备份文件上传到数据库中另外的数据库或者表以供使用

### (1) 获取 access\_token

调用微信的接口需要 access\_token，所以我们首先要获取 access\_token。通过微信开发文档了解到使用 auth.getAccessToken 接口可以用小程序的 appid 和 secret 获取 access\_token。

```
// 获取 access_token

request.get(
  `https://api.weixin.qq.com/cgi-bin/token?grant_type=client_credential&appid=${appid}&secret=${secret}`,
  (err, res, body) => {
    if (err) {
      // 处理错误
    }
    return;
  }
);
```

### (2) 创建数据库导出任务

获取 access\_token 后，就可以使用 databaseMigrateExport 接口导出数据进行备份。

databaseMigrateExport 接口会创建一个数据库导出任务，并返回一个 job\_id。调用 databaseMigrateExport 接口需要传入环境 Id，存储文件路径，导出文件类型（1 为 JSON，2 为 CSV），以及一个 query 查询语句。JSON 类型的数据，兼容性更好。需要备份的数据可以用 query 来约束，使用 db.collection('data').get() 备份 data 集合的全部数据。

```
request.post(
  `https://api.weixin.qq.com/tcb/databasemigrateexport?access_token=${accessToken}`,
  {
    body: JSON.stringify({
      env,
      file_path: `${date}.json`,
      file_type: '1',
      query: 'db.collection("data").get()'
    })
  },
  (err, res, body) => {
    if (err) {
      // 处理错误
      return;
    }
    const data = JSON.parse(body);
    // data.job_id
  });
```

### (3) 查询任务状态，获取文件地址

在创建好数据库导出任务后，会得到一个 `job_id`，当导出完成后，会返回一个 `file_url`，即下载数据库导出文件的临时链接。

```
request.post(
  `https://api.weixin.qq.com/tcb/databasemigratequeryinfo?access_token=${accessToken}`,
  {
    body: JSON.stringify({
      env,
      job_id: jobId
    })
  },
  (err, res, body) => {
    if (err) {
      reject(err);
    }
    const data = JSON.parse(body);
    // data.file_url
  });
```

获取到文件下载链接之后，可以将文件下载下来，存入到腾讯云服务器其他地方或者其他数据库或者数据表，做备份使用。当需要恢复备份的时候，可通过 `job_id` 查询到链接，下载数据进行恢复。为了方便数据库实时恢复就放在数据库中。

```
await db.collection('db_back_info').add({
  data: {
    date: new Date(),
    jobId: jobId
  }
});
```

#### (4) 函数定时触发器

配置函数每天凌晨二点触发，这样就可以每天都对数据库进行备份。在云函数目录下新建 config.json 文件，写入如下内容：

```
{
  "triggers": [
    {
      "name": "dbTrigger",
      "type": "timer",
      "config": "0 0 2 * * * *"
    }
  ]
}
```

### 5.4 出错处理

表 5-1

错误类型	范围	处理方式
操作错误	用户进行交互时不满足输入规则、输入范围等发生的错误	校验用户输入 提示正确规则 强制其改正
运行错误	与外部资源交互发生错误, 例如: 网络错误、数据库错误等	记录并抛出异常 界面提示 记录错误日志信息
.....		

### 5.5 权限设计

#### 5.5.1 用户认证

使用小程序的主体在进行身份认证时，需要提供学生信息和身份信息。

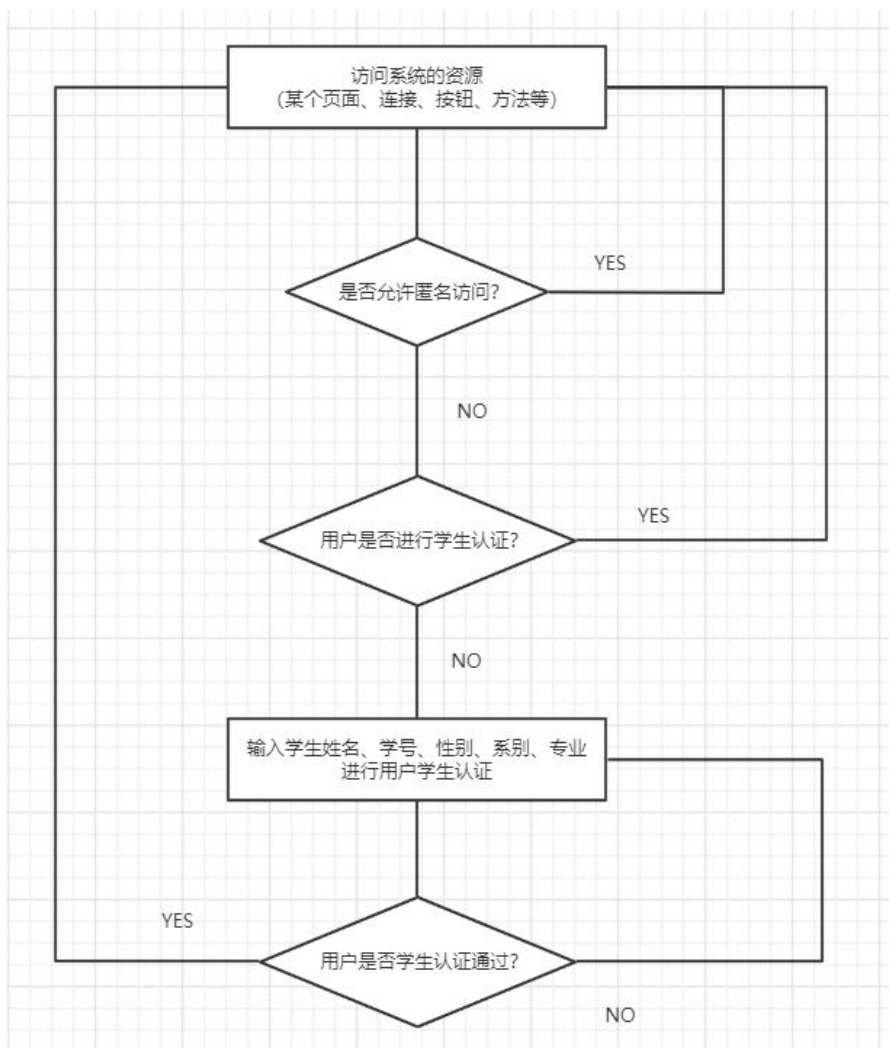


图 5-2 用户认证环节

## 5.5.2 用户授权

给不同用户分配不同的角色, 不同的角色授权不同的资源, 不同的用户就有不同的权限。  
学生用户和普通用户有不同的权限。

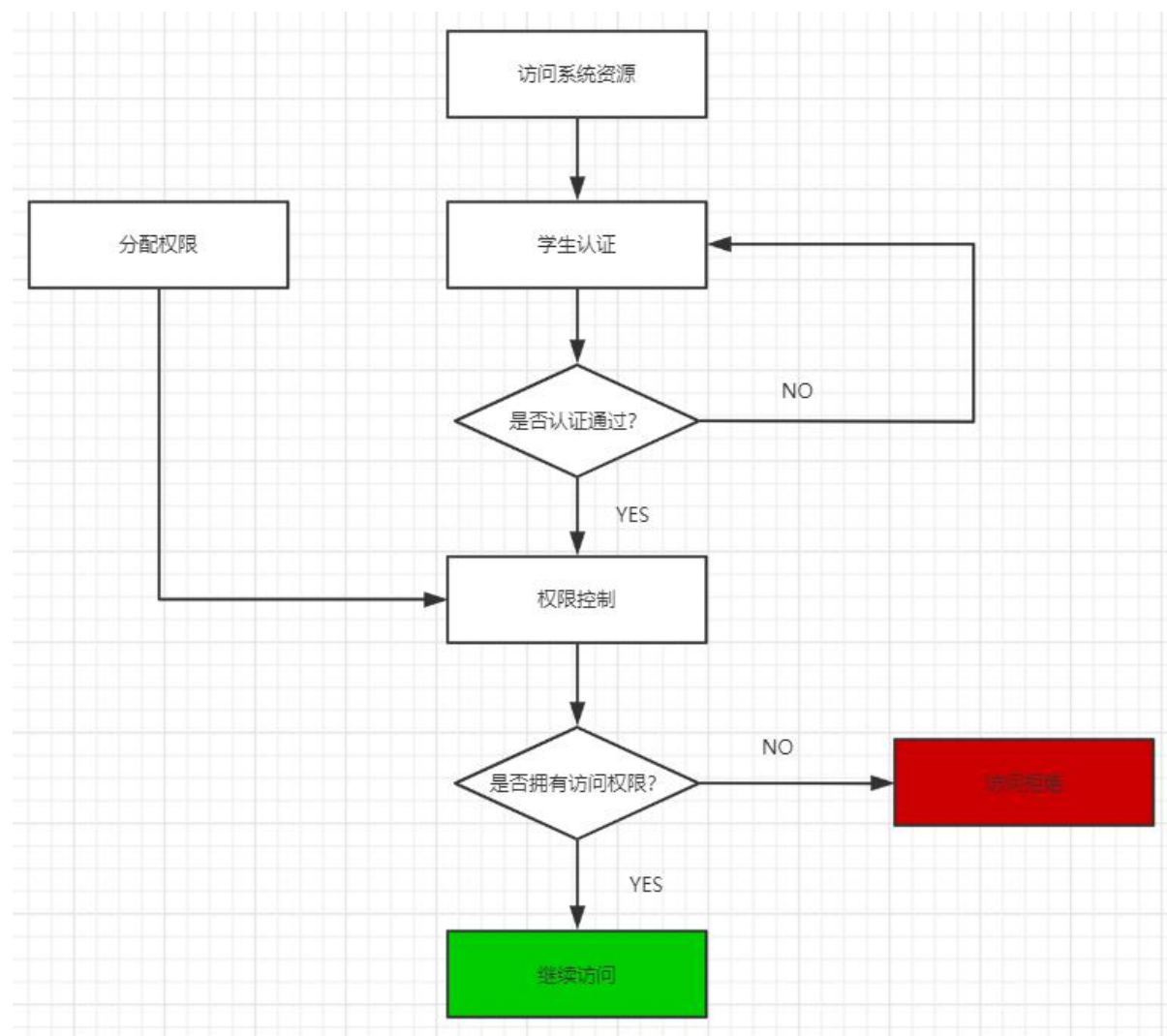


图 5-3 权限分配