

基于 C/S 架构的高校座位管理系统^①



马 可¹, 黄恒熠², 栗云鹏³

¹(河海大学 机电工程学院, 常州 213022)

²(南京航空航天大学 计算机科学与技术学院, 南京 211106)

³(浩鲸云计算科技股份有限公司, 南京 211153)

通讯作者: 马 可, E-mail: bravery1978@163.com

摘 要: 为了实现高校座位的统一智能化管理, 并满足不同区域座位的规则独立设定需求, 开发了一套基于薄膜压力传感器检测和 C/S 架构、符合 Android MVP 设计思维的高校座位管理系统. 分析了高校座位管理的需求, 基于用户友好和界面友善原则, 对系统的功能模块、数据流向和系统架构进行了详细分析和设计. 系统包括 C/S 三层架构, 基于 Android MVP 模式进行分层开发, 实现了数据的映射、访问和持久化, 并通过用户端 APP 以及检测终端的设计和调试, 实现了座位使用情况的可靠检测和数据的信息化综合管理, 提高了座位管理及使用效率, 对解决高校座位管理难题, 以及校园管理的信息化具有积极意义.

关键词: 座位管理; Android MVP; SQLite; C/S 模式; 薄膜压力传感器

引用格式: 马可, 黄恒熠, 栗云鹏. 基于 C/S 架构的高校座位管理系统. 计算机系统应用, 2021, 30(9): 77-84. <http://www.c-s-a.org.cn/1003-3254/8065.html>

Seat Management System of University Based on C/S Architecture

MA Ke¹, HUANG Heng-Yi², LI Yun-Peng³

¹(College of Mechanical and Electrical Engineering, Hohai University, Changzhou 213022, China)

²(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

³(Whale Cloud Technology Co. Ltd., Nanjing 211153, China)

Abstract: For the unified intelligent management of university seats and independent setting of seat rules in different regions, a set of university seat management system based on film pressure sensors and C/S architecture is developed, which is in line with the Android MVP design mode. The requirements for university seat management are analyzed, and the system's functional modules, data flow, and system architecture are designed and analyzed in detail according to the principles of user-friendliness and friendly interface. The system includes the C/S three-tier architecture and relies on the Android MVP model for layered development to achieve data mapping, access and persistence. Through the design and debugging of a user APP and a detection terminal, the reliable detection of seat usage and comprehensive information-based management of data are realized, and the seat management and use efficiency are improved, which is of positive significance to university seat management and information-based campus management.

Key words: seat management; Android MVP; SQLite; C/S mode; film pressure sensor

高校拥有大量的座位资源供学生自习, 包括图书馆、自习室和实验室等, 但高校的座位管理存在

很多难题, 如占座严重、寻座困难、难以统一管理等等^[1,2]. 已有不少高校采用了各种图书馆座位管理系

① 基金项目: 江苏省重点研发计划 (BE2018004-01); 国家级大学生创新创业训练计划 (201810294065); 常州市科技计划 (CE20205044)

Foundation item: Key Research and Development Program of Jiangsu Province (BE2018004-01); National College Student Innovation and Entrepreneurship Training Program (201810294065); Science and Technology Program of Changzhou City (CE20205044)

收稿时间: 2020-12-02; 修改时间: 2021-01-04; 采用时间: 2021-01-13; csa 在线出版时间: 2021-09-02

统进行图书馆座位分配^[3-5],一定程度上缓解了占座问题,但诸如寻座困难、难以统一管理等问题依旧存在。

基于FSR薄膜传感器^[6,7]检测终端的C/S架构的高校座位管理系统可以有效的解决上述问题,在座位中配置检测终端进行座位使用情况的判定,生成检测数据,由管理系统对座位情况和使用用户进行记录,由管理员进行座位使用规则的独立设定,学生可以通过移动APP进行座位的查询和预约,实现座位的独立设置规则并统一管理,方便学生预约,减少学生寻座时间,快速排查占座行为,提高座位的有效利用率和管理效率。

1 需求分析

高校座位信息管理系统是一个对高校所有座位以及与座位相关的使用规则进行统一管理,为管理者和用户提供统一规范的系统。该系统主要适用于高校管理者对高校座位资源的管理和高校师生对高校座位资源的查询和使用。管理系统必须满足以下需求:

(1) 数据资源的有效管理

① 实现系统数据资源的存储,通过本地以及云端,可靠地存储座位使用情况的相关信息和数据;② 实现数据及数据库的管理和维护,为系统各项数据进行存储、查询、修改和校验等,并提供文档、图片等不同数据的传输。

(2) 数据资源的共享与适用

该管理系统适用于高校管理,是高校管理的组成部分,系统的各类信息和数据应符合高校管理的规范,数据类型与结构与高校管理的其他系统相统一,确保管理人员能够获取正确的数据。

(3) 用户友好和界面友善

该系统需为用户提供便捷的登陆注册界面以及系统座位信息服务界面,用户登陆后系统按照用户的身份权限为用户规划出可以使用的座位和可以预约的座位,用户可以通过点击对应的设备来操作符合其权限的设备,进行业务的操作和实现,使得用户可以通过简单的操作完成系统的任务,提高工作效率。

(4) 功能的可扩展性

该系统为高校座位管理服务,需要提供座位查询、预约、管理等基本功能,同时还需要能够高效的扩展,以满足高校座位管理的新需求。

2 系统设计

整个系统可分为3个部分,包括用户端应用、云服务器端应用和检测终端。用户端应用面向使用者,提供交互服务;云服务器端应用负责逻辑处理和数据存储;检测终端作为硬件检测设施,与座位配套,负责检测座位的使用情况。为了实现系统的可扩展性和易维护性目标,系统各个不同部分直接进行交互时,根据云服务器端应用的数据结构类型进行通信时数据的格式化规范。

2.1 功能模块设计

高校座位包括了图书馆、自习室和实验室等不同区域,各个区域的管理规范各不相同,负责的管理人员繁多,系统对各个区域的座位进行统一管理能够提升管理的效率。针对系统需求的不同部分,将系统的功能模块主要划分为:用户管理、座位配置、座位管理和座位检测。系统的功能模块如图1所示。

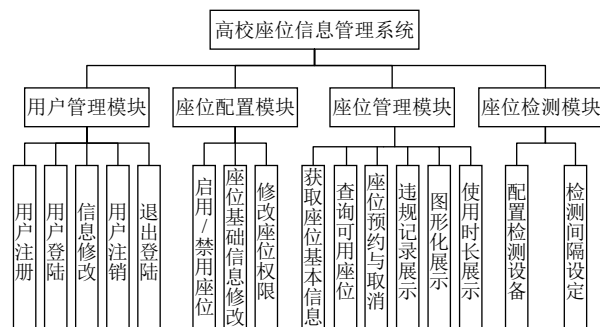


图1 系统的功能模块图

(1) 用户管理模块:即管理该系统的用户,负责管理用户的基本信息。包括用户的注册、用户的信息录入、用户登陆与用户退出、用户信息修改、用户注销。为了满足高校管理时信息的统一管理,系统除了开放用户账户的注册方式,也可以通过统一导入账户信息以创建用户账号,接入校园信息管理系统,实现信息的共享。

(2) 座位配置模块:即管理如何配置该系统中的座位使用权限,面向高权限座位管理员。由各个区域的管理员对座位进行设置。该模块主要包括了设置座位的启用情况、修改座位的基础信息和修改座位的使用权限等。通过对座位功能的统一设定,实现不同区域不同座位管理方式的自由设定,实现座位使用的自由、公开管理。

(3) 座位管理模块:即管理座位的使用,面向系统

中高校座位使用者. 包括为用户获取座位的基本信息, 如座位的位置、使用方式、管理公告等, 为用户提供可用座位的查询、座位的预约以及预约的取消, 同时将座位信息进行图形化显示, 为用户记录违规信息和使用时长, 以供统计和管理.

(4) 座位检测模块: 即检测座位的使用情况. 该模块主要包括了为座位配置对应的座位检测设备, 以及为座位检测设备设置检测间隔, 包括座位检测时间间隔和判定未使用的时长间隔等.

2.2 数据流设计

该高校座位信息管理系统包括了大量的数据通信和数据存储等操作, 采取 C/S 架构进行开发, 需要明确处理数据的在网络中的流动和管理. 由于采取了 C/S 架构, 系统中的数据流动可以按照需求进行设定, 系统的数据流向图如图 2 所示.

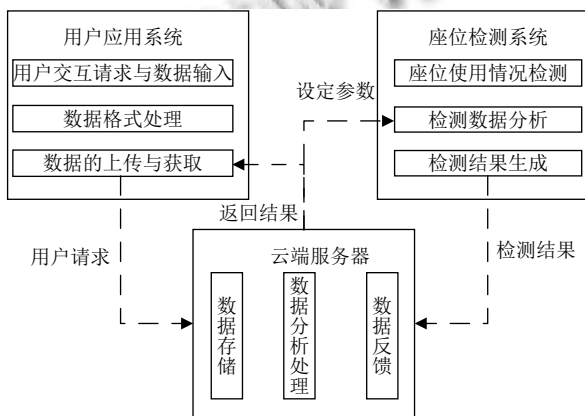


图2 系统的数据流向图

系统中主要包含了用户的请求数据和座位检测数据, 数据作为信息的载体, 在各个不同模块间流动和传输, 将系统的模块连接起来, 实现了系统各个模块的解耦, 利于系统的层次化设计.

2.3 系统总体架构设计

按照系统需求和数据流设计, 系统的总体架构设计如图 3, 主要包括了移动手机 APP^[8]、云端应用服务器、座位检测终端、座位检测服务器、MySQL 数据库服务器.

系统架构将系统划分为云端服务、用户端交互、终端座位检测 3 个部分, 各个部分功能可以独立开发, 提高系统的开发效率和稳定性. 该系统架构适合采用广泛的 C/S 开发模式, 并适合于将系统划分为“用户界面/数据采集—功能逻辑—数据库”的 3 个层次. 对系统

进行该架构的划分使得系统各个部分独立开发, 使得各个层次的功能透明, 便于各个层次的独立维护和扩展.

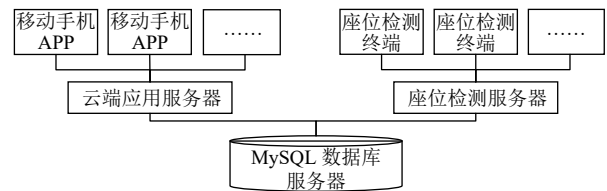


图3 系统总体架构图

2.4 系统软件架构设计

根据功能的可扩展性的需求, 该系统采用了 Android MVP 设计模式进行系统的开发, 使得系统划分为了 4 层: 用户应用层、逻辑服务层、数据模型层、数据存储层. 系统的软件架构如图 4.

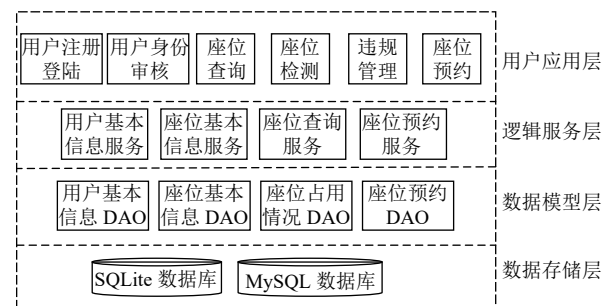


图4 系统软件架构图

采用 Android MVP 设计模式的优点在于: 系统的各个部分相对独立, 开发者可以将一个功能拆分为多个部分进行开发, 使得代码有极高的可读性, 功能实现十分明晰. 通过面向接口, 隐藏和忽略其他层的实现方式, 使得开发者可以实现各层的独立设计, 也使得系统的功能可以高效的进行扩展和维护.

其中用户应用层主要负责交互、显示以及座位使用情况的检测, 主要对应了用户应用的 APP 界面和座位检测终端; 其中逻辑服务层主要负责功能的逻辑实现, 对应了用户应用中的控制层和服务器应用的逻辑层; 其中数据模型层主要负责数据的面向对象的实体类的管理以及数据库的操作; 其中数据存储层主要包括了 MySQL 数据库服务器和本地数据库 SQLite, 分别负责系统的数据存储管理和用户应用的本地数据存储.

2.5 系统硬件架构设计

此系统为了实现作为的统一管理, 需要设计终端进行座位信息的绑定以及使用情况的检测, 通过硬件

终端检测座位使用情况,可以生成座位管理需要的大量数据,便于后续的分析,本系统的座位检测终端原理如图5所示。

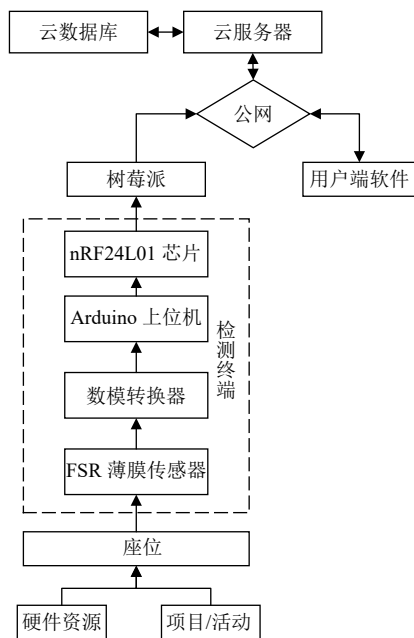


图5 检测终端原理图

检测终端将实验室、自习室和图书馆等不同的区域中的座位录入系统,并在每个录入系统的座位上安装检测终端,使用 FSR 薄膜传感器,能够检测出座位是否有人使用,检测结果通过 A/D 转换器转换为数字信号,经 Arduino 平台上的 nRF24L01 无线模块传输数据到上位机。系统使用基于 ARM 的微型电脑 Raspberry Pi 作为检测终端上位机,接收检测终端数据,并上传到网络服务器中,并由服务器系统软件对检测数据进行处理,结合服务器系统软件提供的其他功能子模块,向用户端软件提供系统功能,实现对高校实验室、自习室和图书馆的不同资源进行统一管理。

3 主要方法和技术

3.1 Android MVP 技术

(1) 技术思路

MVP 的核心在于将程序中的视图和数据分离,减轻 Activity^[9,10] 的负担,也使得程序的逻辑更加简洁。其中, M 层负责数据的获取, V 层负责视图的处理, P 层负责逻辑的实现。其中 V 层不直接调用 M 层也不持有 V 层,而是由 P 层持有 V 层对象和 M 层对象,作为

中介来获取数据^[11,12]。虽然在一定程度上加大了代码量,但是在逻辑上的简洁性和明确性使得这个缺点可以承受。

(2) 具体的分层结构

该系统采用 MVP 的思想进行构建,分别构建了 M 层、V 层和 P 层,项目结构图见图 6。

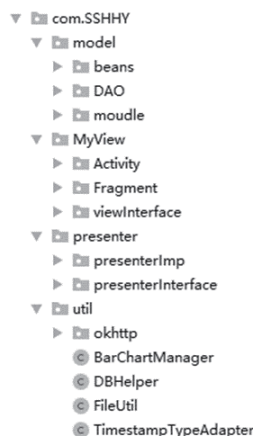


图6 项目结构图

图 6 中 MyView 对应的是 V 层, model 对应的是 M 层, presenter 对应的是 P 层, util 是常用工具类。MyView 下包含了 Activity 类、fragment 类和自定义 view 类以及对应的视图接口类; model 下包含了实体的抽象类 Bean、网络请求的自定义 bean 类、用于本地数据库查询的 DAO 类和用于网络 HTTP 请求的 Moudle 类; presenter 包含了层现器类和层现器的接口类。

(3) 以单个功能模块的数据流为例,系统实现 MVP 设计思想的流程如下:

- ① Activity 继承对应的 View 接口,实现其中方法;
- ② Activity 在 onCreate() 中绑定用户信息实体的父容器 LinearLayout,并实例化对应的 Presenter 层现器,调用层现器中检查用户身份表版本的方法 checkUserRoleVersion() 和请求加载用户数据的请求 getSeatPartUser(Seat seat);
- ③ 层现器通过在实例化时持有 View 对象,并创建 Moudle 对象和 DAO 对象;
- ④ 层现器调用 Moudle 中网络请求方法获取用户数据,调用 DAO 中数据库请求方法获取本地数据库数据;
- ⑤ 返回结果数据后, Moudle 回调层现器方法,层现器回调 View 方法,将数据返回到页面 Activity;
- ⑥ Activity 调用动态创建实体方法,利用得到的数

据完成页面更新。

3.2 数据持久化技术

数据的持久化^[13]是该系统的核心技术之一,在服务器端通过使用数据库进行数据的持久化,通过数据类型映射进行数据的访问和存储,并通过 JavaBean 技术进行服务器与移动端应用的数据交互。在 Android 移动端应用中,对于数据的本地存储也有一定的需求,为了能够应对不同大小、不同类型的数据,本系统实现了 SharedPreference、SQLite 存储方式。

(1) 数据类型映射

本系统中包括了大量不同类型的数据,在数据库中建立数据了对应的条目,为了实现数据库中的数据在系统应用中的流动和交互,系统设计了对应的数据映射类型,实现数据格式的统一化,有利于数据的网络交互的数据库存储读取规范化。其中最核心的数据映射类型包括了用户类 User、座位类 Seat、用户身分类 UserRoleBean、座位预约类 Appointment。以用户类为例,其映射结构如下:

```
@Setter
@Getter
@ToString
public class User {
    @NotBlank
    private String userId;
    private String userName;
    ...
}
```

(2) SharedPreference

对于系统中常用的一些数据,如简单参数和系统状态等,若其数据量较小,本系统将其存储在 XML 格式的文件中。

以本系统进行暂存当前用户登陆状态和基本信息的 SharedPreference 为例:存储路径如下: /data/data/com.SSHHY/shared_prefs/userinfo.xml

流程:实例化 SharedPreferences 对象,存储 SharedPreferences 对象,读取 SharedPreferences 对象。

其代码实现如下:

```
# 实例化 SharedPreferences 对象
private SharedPreferences sharedPreferences;
# 读取 SharedPreferences 对象
sharedPreferences=getSharedPreferences("userinfo",
```

```
Context.MODE_PRIVATE);
```

```
Boolean loginFlag= sharedPreferences.getBoolean("loginFlag",false);
```

```
# 修改和存储 SharedPreferences 对象
```

```
SharedPreferences.Editor editor = sharedPreferences.edit();
```

```
SharedPreferences.Editor editor = sharedPreferences.edit();
```

```
editor.apply();
```

(3) SQLite

对于系统中的基础数据,当其数据量相对较大时,本系统使用 SQLite 轻量级数据库进行存储,存储在 db 格式的文件中。

以本系统中的 SQLite 数据库为例,存储路径如下: /data/data/com.SSHHY/databases/myDemo.db

SQLite 数据库支持 SQL 语法,同时 Android 提供了 SQLiteOpenHelper 类进行操作。为了使得数据库操作的逻辑符合 MVP 的思想,本系统开发时将 SQLite 数据库的操作划分入 M 层,由专门的 DAO 类进行操作。所有对数据库的操作均从 Activity 的 UI 主线程分离出来,由 P 层作为中介进行连接。

为了进行数据库的查询等操作时,建立了对应数据库的 DAO 类,通过持有 DBHelper 对象实现数据库的增删改查。结合数据的存储和访问技术,通过数据类型的映射,可便捷实现用户端两数据库的交互。

4 系统实现

4.1 移动用户端界面

(1) 登陆注册界面

用户通过该界面进行新账号的申请,也可以通过已有账号的用户账户和密码进行登陆,如图 7。

(2) 个人信息及查询界面

用户完成登陆后进行个人信息界面,查看、完善和修改个人的信息;可以进行查询,按照顶部导航栏中的下拉框进行选择即可查询得出该教室中所有座位的使用情况如图 8。

(3) 预约提示

用户查询到座位后,对座位进行点击即可对该座位进行预约,系统按照座位的使用情况、管理员设置的使用期限和用户个人权限进行预约,并弹出对应的提示框进行提示预约情况,如图 9。



图7 登陆注册界面



图8 个人信息及查询界面

(4) 预约界面

用户成功完成预约后自动进去预约界面, 期间无法进行其他座位的查询和预约, 用户在预约时间未按时使用座位或离开座位时间过久会被自动取消座位预约资格, 防止用户占座行为, 如图10.

4.2 检测终端的实现

(1) 检测终端的连接

系统使用检测终端检测座位使用情况, 使用 Arduino 作为检测终端上位机, 由 Arduino 连接树莓派服务器. 如图11所示. 其中检测终端包含了 FSR 薄膜传感器、数模转换器、LED 显示灯、离座按键. FSR 薄膜传感器检测时不会对 10 kg 以下的书包、书籍等物品产生检测值, 只会对超过 40 kg 的人坐到座位上时才会产生

检测值. FSR 薄膜传感器通过数模转换器连接到 Arduino, 将产生的数字量转变成模拟量, 传输到 Arduino 可以识别的数据. LED 显示灯由两个不同的 LED 组成, 一个灯在按下离座按键时亮起, 指示暂离, 另一个灯亮起指示违规. Arduino 通过 nRF24L01 芯片射频通信与上位机树莓派进行连接, 将数据传输至树莓派.



图9 预约提示

(2) 检测终端的通信配置

系统在投入使用前, 需要配置网络设备保证检测终端与互联网联通. 检测终端中, 使用 Arduino 通过 nRF24L01 芯片射频通信与上位机树莓派进行连接, 将数据传输至树莓派. 树莓派连接至互联网公网, 树莓派中安装 Tomcat 网络服务器, 利用 Tomcat 访问服务器的公网 IP 地址, 将数据发送至服务器的规定端口. 服务器开放对应端口, 以接收数据. 服务器开放对应端口

以接收数据,如图 12 所示,其中 80 端口为 Tomcat 服务器默认端口,8081 为 APP 的服务器网络服务端口,6000 和 7000 为 FRP 内网穿透端口,20 和 21 是 FTP 服务端口.通过访问服务器公网 IP 地址下的对应端口号,以访问对应的服务或上传对应的数据.

```
Serial.println(val_AD);
delay(20);
}
```

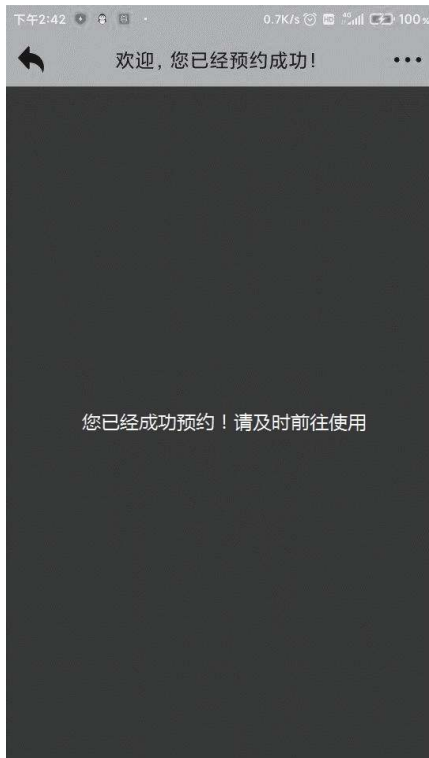


图 10 预约界面

(3) 检测终端读取的代码示例

FSR 薄膜传感器连接到 Arduino 的模拟输入引脚读取数值引脚后,可以由 Arduino 读取其模拟输入数值,使用 analogRead(pin) 函数可以读取 0 到 1023 之间的值,代表的是 0-5 V 的电压信号,这个电压值可以用于后续估计薄膜传感器上受到的压力范围,并可以通过串口监视器显示.其代码实现如下:

```
int val_AD = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  val_AD = analogRead(A0);
```

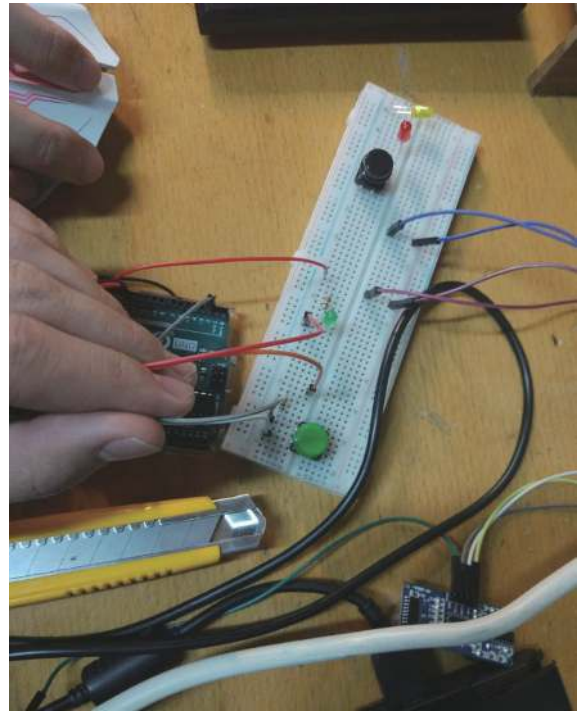


图 11 检测终端连接示意图

<input type="checkbox"/>	允许	自定义 TCP	80/80
<input type="checkbox"/>	允许	自定义 TCP	8081/8081
<input type="checkbox"/>	允许	自定义 TCP	6000/6000
<input type="checkbox"/>	允许	自定义 TCP	8080/8080
<input type="checkbox"/>	允许	自定义 TCP	7000/7000
<input type="checkbox"/>	允许	自定义 TCP	20/21

图 12 服务器开放端口

5 结论与展望

本文对高校座位的信息化管理进行探索,研究了对应的软件管理系统,有利于数据的统一管理、信息

的安全存储,为相关管理人员提高了统一的人员导入、规则设定、信息统计等功能,为高校座位的统一管理提供了更便捷、更可靠的管理方式.满足了管理人员对信息化管理的需求,方便高校用户对高校资源的利用率,提高高校管理的信息化水平,提高管理人员的管理效率,在实际测试中,满足了使用者的基本使用需求.

参考文献

- 1 谢发徽. 高校图书馆座位管理系统的读者行为分析. 图书馆论坛, 2018, 38(3): 108–116.
- 2 尹相权. 基于图书馆座位管理系统的用户行为分析. 数字图书馆论坛, 2019, (10): 37–44. [doi: 10.3772/j.issn.1673-2286.2019.10.006]
- 3 温红超. 高校图书馆读者使用座位预约管理系统的经济学分析——以吉林大学鼎新图书馆为例. 图书情报工作, 2019, 63(17): 110–117.
- 4 罗志鹏, 张伟, 陈琪玮, 等. 基于 LoRa 的智能图书馆座位管理系统设计与实现. 计算机测量与控制, 2019, 27(3): 259–263.
- 5 周耀, 蒋天民. 基于微信公众平台构建图书馆座位管理系统. 现代情报, 2016, 36(12): 114–119. [doi: 10.3969/j.issn.1008-0821.2016.12.022]
- 6 Li QY, Luo SJ, Wang QM. Piezoresistive thin film pressure sensor based on carbon nanotube-polyimide nanocomposites. *Sensors and Actuators A: Physical*, 2019, 295: 336–342. [doi: 10.1016/j.sna.2019.06.017]
- 7 金忠, 谢锋, 何迎辉, 等. 高精度薄膜压力传感器的研制. 自动化与仪表, 2020, 35(7): 79–82, 88.
- 8 洗远清, 匡珍春, 初庆柱. 基于 Android 的智慧教学 APP 设计. 实验技术与管理, 2019, 36(9): 128–131.
- 9 刘奥, 过辰楷, 王伟静, 等. Android 应用 Activity 启动环研究. 计算机学报, 2020, 43(3): 537–554. [doi: 10.11897/SP.J.1016.2020.00537]
- 10 Wu TY, Liu JR, Xu ZB, *et al.* Light-weight, inter-procedural and callback-aware resource leak detection for Android APPs. *IEEE Transactions on Software Engineering*, 2016, 42(11): 1054–1076. [doi: 10.1109/TSE.2016.2547385]
- 11 陈星, 杨春花. Android 异步技术的对比实验及应用. 计算机应用与软件, 2016, 33(2): 188–191, 279. [doi: 10.3969/j.issn.1000-386x.2016.02.045]
- 12 薛海龙, 陈渝, 雷蕾, 等. Android 应用异步编程模型性能分析. 计算机科学与探索, 2018, 12(10): 1604–1613. [doi: 10.3778/j.issn.1673-9418.1710024]
- 13 魏巍, 蒋德钧, 熊劲, 等. 基于非易失存储器件的内存键值存储系统的性能研究. 高技术通讯, 2017, 27(6): 519–529. [doi: 10.3772/j.issn.1002-0470.2017.06.005]