

哈爾濱工業大學

## 《物联网工程实践》设计报告

学 生 董成相

学 号 181110305

同组人员 吉张雄

日 期 2021 年 5 月 1 日

系统基本情况

| 界面 |     |    | 数据库     | 驱动程序 | 硬件使用  |       |      |
|----|-----|----|---------|------|-------|-------|------|
| QT | Web | 其它 |         |      | 嵌入式平台 | 单片机平台 | 外接模块 |
| √  |     |    | SQLite3 |      | IMX6  |       | RFID |

组内成员工作量占比

| 学号        | 姓名  | 工作量占比 |
|-----------|-----|-------|
| 181110305 | 董成相 | 50%   |
| 181110308 | 吉张雄 | 50%   |

## 1. 绪论

我在使用我们学校的图书馆座位预约系统时，注意到我们学校的图书馆座位预约系统上虽然有续约的按钮，但是都不能使用。因此便想借着物联网工程实践的机会基于 RFID 做一个可用的座位预约系统出来。

## 2. 系统整体设计

### 2.1. 系统整体构架

如图 2-1 所示，系统主要由现场的触摸屏、服务器，以及安卓手机和数据库组成。实际上，触摸屏、服务器、数据库全都是运行在 IMX6 平台上。触摸屏给用户提供了现场操作的入口，有点类似图书馆的现场预约机的功能；服务器则用于处理各种请求，包括触摸屏发来的请求以及安卓手机客户端发来的请求；安卓手机通过连接 WiFi 与服务器进行通信，进而实现数据的保存。

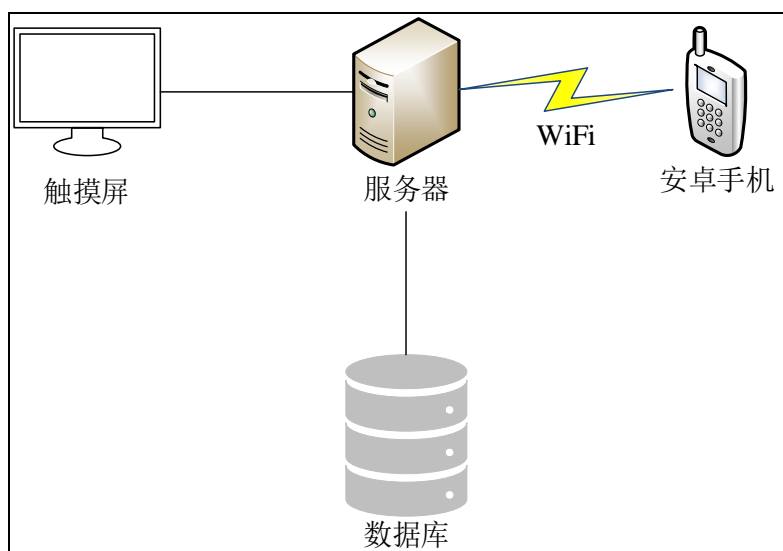


图 2-1 系统整体构架

### 2.2. 软件主要功能模块

#### 2.2.1. 主窗口

主窗口实际上并不实际存在，不同的操作状态下由不同的子界面充当主窗口。主窗口由四个大的部分组成：管理员界面、首页、用户操作界面和 TcpServer。后续将一一介绍这几个部分。主窗口同时也能访问数据库以便实现从首页切换到用户操作界面时用户的合法性检验；呼出数字键盘和输入对话框以便实现从首页切换到管理员界面时的管理员身份核验；获取 RFID 读取的信息，以便在从首页切换到用户操作界面时获取用户的身份识别信息。

2.2.2. 管理员界面

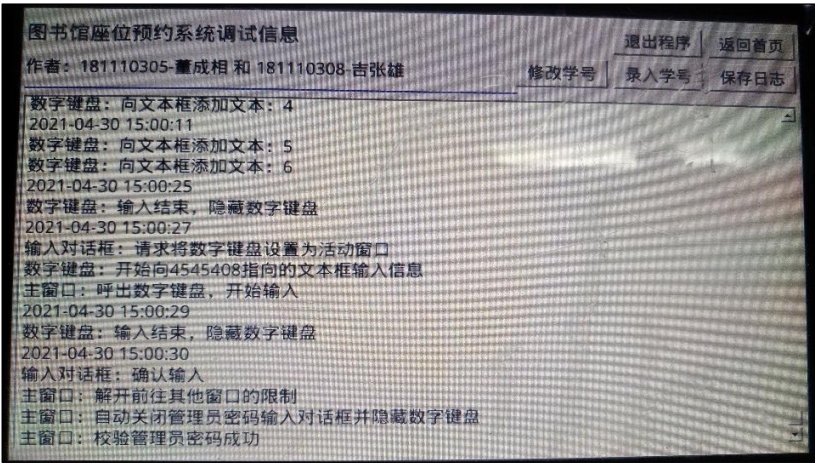


图 2-2 管理员界面

如图 2-2 所示，管理员界面包含了系统标题、作者信息、以及五个操作按钮和日志文本框。

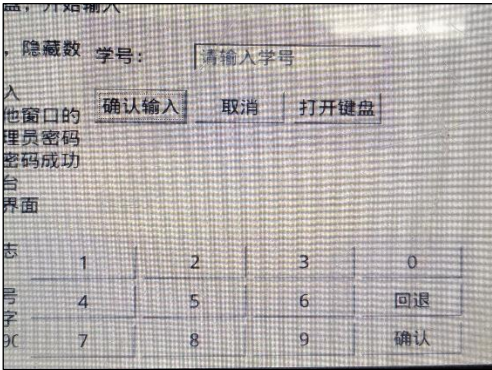


图 2-3 录入学号

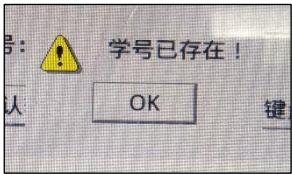


图 2-4 学号已存在错误提示

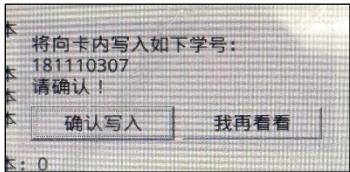


图 2-5 写入学号确认

如图 2-3 所示，管理员点击录入学号按钮后，会呼出输入对话框和数字键盘，实现修改目标卡片内存储的学号并将学号存入数据库中的功能。

若数据库中已存在待写入的学号，则不会执行任何操作，包括修改卡片内的学号，同时会提示用户学号已存在（如图 2-4 所示）。

若数据库中尚未保存待写入的学号，则会让用户确认写入（如图 2-5 所示）。

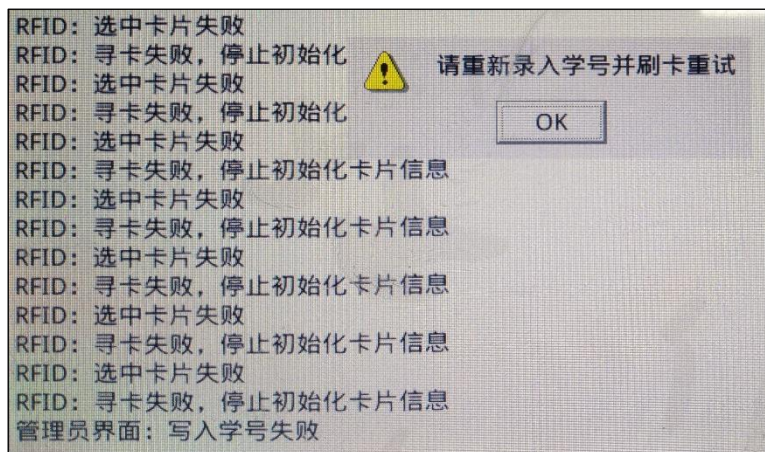


图 2-6 向 RFID 卡片内写入学号失败

在管理员确认要写入学号之后，将会尝试修改卡片内存储的学号，在写入卡片内成功之后才会将学号写入数据库。如图 2-6 所示，如果在多次尝试写入卡片都失败之后，则会提示用户刷卡重试。

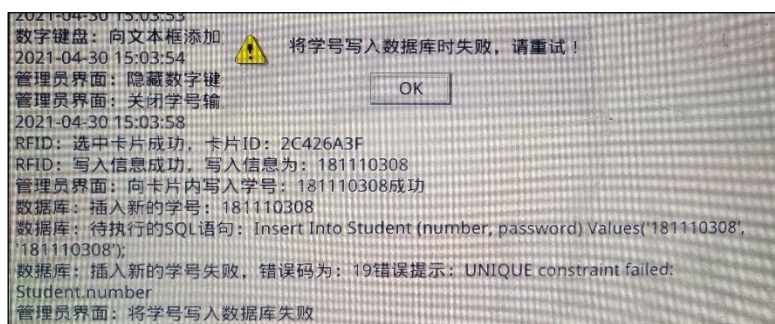


图 2-7 修改学号时尝试写入数据库

管理员点击修改学号按钮之后，也会呼出输入对话框和数字键盘，与图 2-3 展示的录入学号的效果一致，只不过修改学号不会检验学号是否已存储于数据库中，而只会修改目标卡片内存储的学号。逻辑上来讲，也不应该尝试写入数据库，但是由于现在的逻辑有点缺陷，所以还是会尝试写入数据库，不过得益于数据库的主键约束，如果时数据库中已存在的学号，则不能正常写入（如图 2-7 所示），但是数据库中不存在的还是能成功写入。

除去以上阐述的功能，管理员界面还有三个功能入口：退出程序、返回首页、保存日志。退出程序用于主动正常结束程序，以便程序在运行过程中产生的动态内存能被正常回收。返回首页则是在管理员操作完成之后返回到首页，让其他用户继续正常



使用座位预约系统。保存日志则是主动将收集到的日志信息保存到文件中，以便之后的查看复盘。

### 2.2.3. 首页

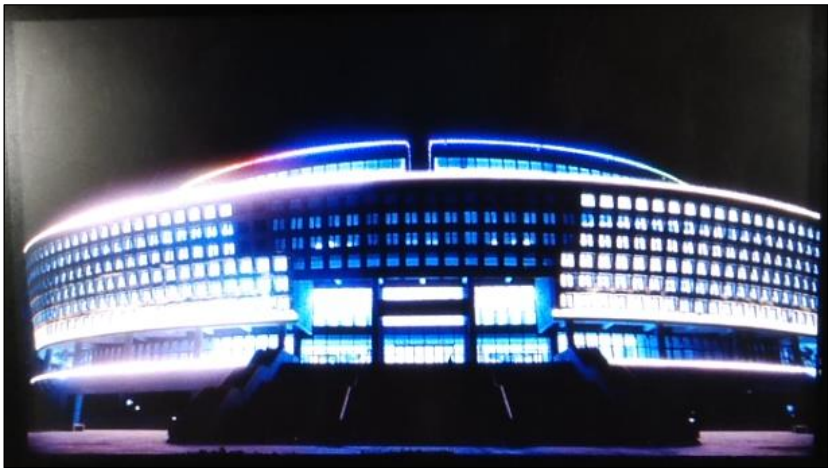


图 2-8 首页

如图 2-8 所示，首页的实现则比较简单，就是一个图片标签，能响应鼠标点击事件。首页分为两个区域，一个是左上角 100\*100 像素的一个小的区域，不妨称作区域 A，在区域 A 之外的不妨称作区域 B。区域 A 用于响应进入管理员界面的点击事件，而区域 B 则用于响应进入操作界面的点击事件。

如图 2-9 所示，在用户点击区域 A 之后将会弹出输入对话框和数字键盘，让用户输入管理员密码以便验证当前正在操作的用户是否具备进入管理员界面的权限。



图 2-9 管理员密码验证

如图 2-10 所示，如果用户点击了区域 B，但是并没有同时刷卡，则会提示用刷卡以便进入操作界面。若同时刷卡，则会核验用户的身份，如果是被识别的用户则会切入操作界面，否则则会提示用户找管理员注册（这里忘记拍照片，不过之前测试的这个核验用户的功能是好用的）。

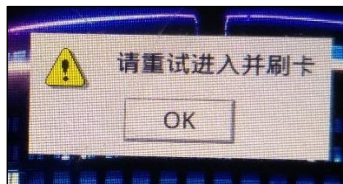


图 2-10 提示用户刷卡

## 2.2.4. 用户操作界面

用户操作界面包括现场预约、我的预约以及顶部部件三个部分。

### 2.2.4.1. 现场预约

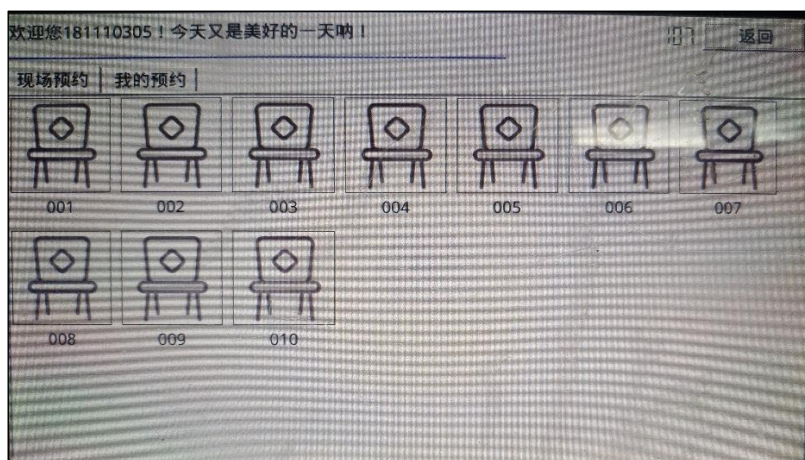


图 2-11 现场预约

如图 2-11 所示，现场预约界面中罗列了所有的座位，如果座位多了起来，右侧将会出现一个滚动条用于向下滑动。每一个座位都包含一个座位图片和座位的编号。



图 2-12 选择预约时间

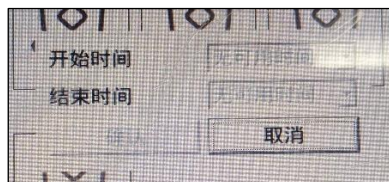


图 2-13 座位无可用时间

如图 2-12 所示，点击座位图片之后将会呼出选择时间的对话框。可以在这里选择开始时间和结束时间，结束时间会随着被选中的开始时间而变化，从而保证用户选择

的时间是合法时间段。当然在没有可用时间段时，会直接限制用户的操作，如图 2-13 所示。

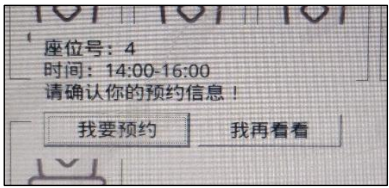


图 2-14 确认预约对话框

如图 2-14 所示，在用户选好预约时间段之后，会弹出一个确认预约信息的对话框让用户核对自己的预约信息。

2.2.4.2.我的预约

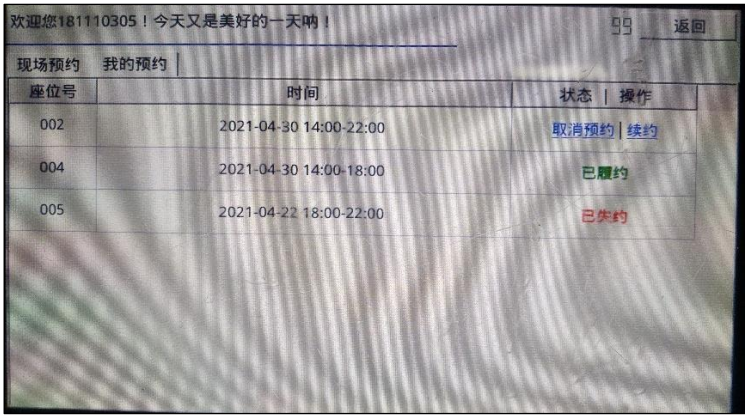


图 2-15 我的预约

我的预约界面则有一个表格，用于展示用户所有的预约记录，针对不同的预约状态有不同的颜色显示，如图 2-15 所示，如果用户正常履约，状态则会显示为“已履约”，颜色是绿色，而如果失约，状态则会显示为“已失约”，颜色为红色，如果用户未签到（尚未到达预约时间段）或者正在履约，则显示取消预约和续约的按钮，以便用户执行适当的操作。

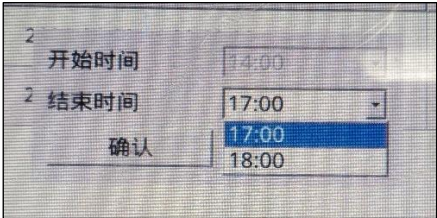


图 2-16 选择续约结束时间



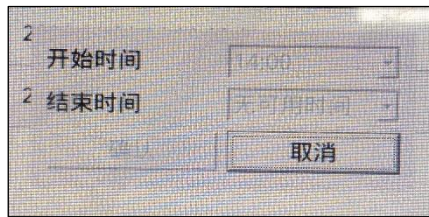


图 2-17 续约时无可用结束时间

如图 2-16 所示，如果用户点击了续约，那么会弹出一个选择续约结束时间的对话框让用户选择且仅让用户选择新的结束时间，并且新的结束时间从原本的结束时间之后的第一个可用时间开始，如果不存在可用的续约时间，则会提示用户无可用时间，并限制用户的操作，如图 2-17 所示。

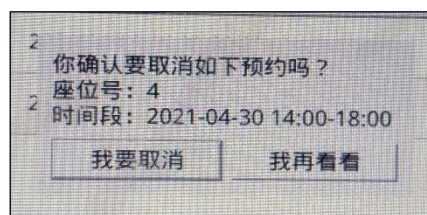


图 2-18 取消预约确认对话框

| 现场预约 | 我的预约                   |           |
|------|------------------------|-----------|
| 座位号  | 时间                     | 状态   操作   |
| 002  | 2021-04-30 14:00-22:00 | 取消预约   续约 |
| 004  | 2021-04-30 14:00-18:00 | 已履约       |

图 2-19 取消预约成功

如图 2-18 所示，如果用户点击了取消预约，则会弹出预约信息让用户确认是否取消预约，若用户确认取消预约，则会认为用户完成了预约，将用户该条预约的状态修改为“已履约”（如图 2-19 所示）。

### 2.2.4.3.顶部部件



图 2-20 顶部部件

顶部部件在用户切换现场预约和我的预约时不会发生变化，所以这里单独将顶部部件设计为了一个部件。

如图 2-20 所示，顶部部件包含了一个返回首页的按钮，用于让用户在操作完成之后主动返回首页，同时还有一个倒计时的部件，用于自动返回首页。倒计时部件的设计目标是为了解决这样一个场景：用户 A 在操作完成之后并没有点击“返回按钮”主动返回首页，而这时用户 B 来到了机器这里，那么用户 B 就可以直接操作，从而给用

户 A 带来不确定的影响。最后剩下的就是一个欢迎提示语了。

2.2.5. TcpServer

本部分由吉张雄完成，这里不做进一步的介绍。

2.2.6. 数字键盘和输入对话框

2.2.6.1.数字键盘

如图 2-3 和图 2-9 所示，数字键盘仅包含 0-9 这 10 个数字。然后还有两个辅助性按钮，回退可以删除目标文本框中的一个字符，确认将会结束输入并隐藏数字键盘，之后仍然可以通过输入对话框的打开键盘来呼出键盘继续进行输入。

数字键盘的实现使用了单例模式，以便模拟保证程序只会从一个地方接收到输入，并且数字键盘使用多个实例是没有意义的，因为数字键盘没有明显的状态，所以使用单例模式可以从一定程度上减少程序内存的占用。

2.2.6.2.输入对话框

如图 2-3 和图 2-9 所示，输入对话框在不同的场合下有不同的表现，这是因为使用了多个输入对话框的实例。经过分析不难发现，输入对话框的表现几乎是一模一样的，出了显示效果，提示信息可能略有不同，所以完全可以对输入对话框进行一定程度的抽象，并对外提供设置不同之处的方法。

2.2.7. 安卓客户端

此部分由吉张雄完成，这里不做进一步的介绍。

2.2.8. 数据库

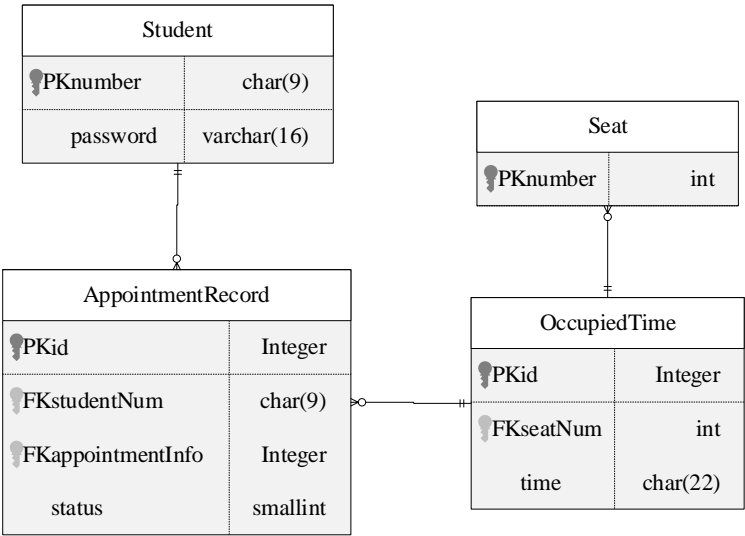


图 2-21 数据库 ER 图——Crow's Foot 方法

数据库使用了 SQLite3。数据库的设计也采用了单例模式，目的是为了尽量避免数据库操作的冲突。

数据库一共设计了四张表，如随后的表格所示。

表 2-1 数据库学生表 Student

| 字段 | 名称       | 数据类型        | 备注 |
|----|----------|-------------|----|
| 学号 | number   | char(9)     | 主键 |
| 密码 | password | varchar(16) | 无  |

表 2-2 数据库座位表 Seat

| 字段  | 名称     | 数据类型 | 备注 |
|-----|--------|------|----|
| 座位号 | number | int  | 主键 |

表 2-3 数据库占用时间表 OccupiedTime

| 字段    | 名称      | 数据类型     | 备注      |
|-------|---------|----------|---------|
| 编号    | id      | Integer  | 主键，自动增长 |
| 座位号   | seatNum | int      | 外键      |
| 占用时间段 | time    | char(22) |         |

表 2-4 数据库预约记录表 AppointmentRecord

| 字段   | 名称              | 数据类型     | 备注      |
|------|-----------------|----------|---------|
| 预约号  | id              | Integer  | 主键，自动增长 |
| 学号   | studentNum      | char(9)  | 外键      |
| 预约信息 | appointmentInfo | Integer  | 外键      |
| 状态   | status          | smallint |         |

2.2.9. RFID

RFID 的设计则较为简单，直接封装了示例程序 RFID 相关的操作代码，最终设计了两个对外提供的接口：写入学号和修改学号。RFID 类的设计也采用了单例模式。

2.3. 通信协议

安卓客户端与服务器的通信采用了采用 TCP/IP 协议。

3. 系统具体设计与实现

部分具体设计已在软件主要功能模块中进行了介绍，这里不再赘述，仅对一些关键的业务流程进行介绍。

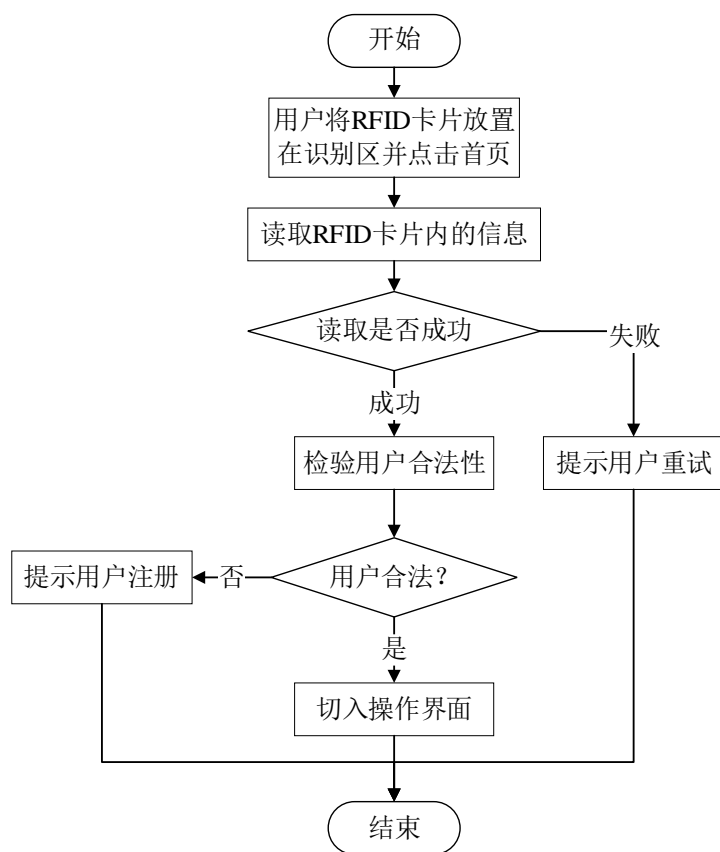


图 2-22 用户进入操作界面流程图

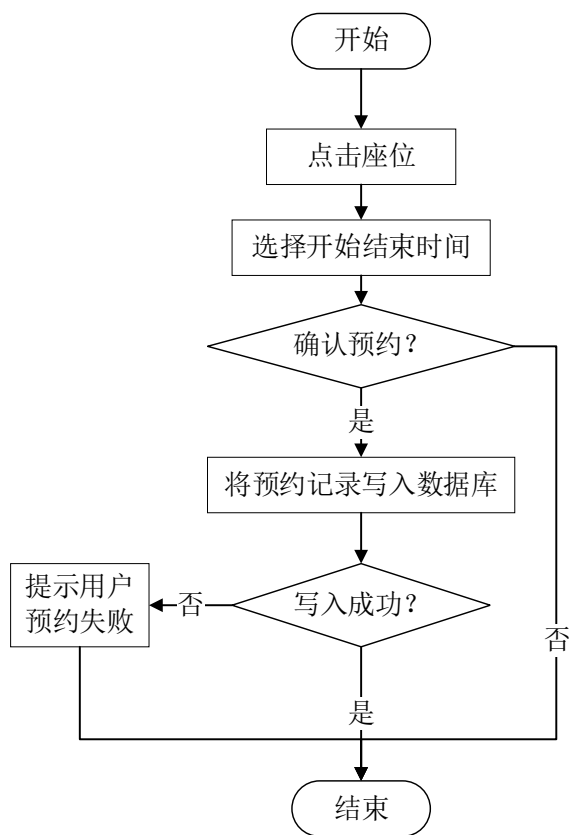


图 2-23 用户预约流程图



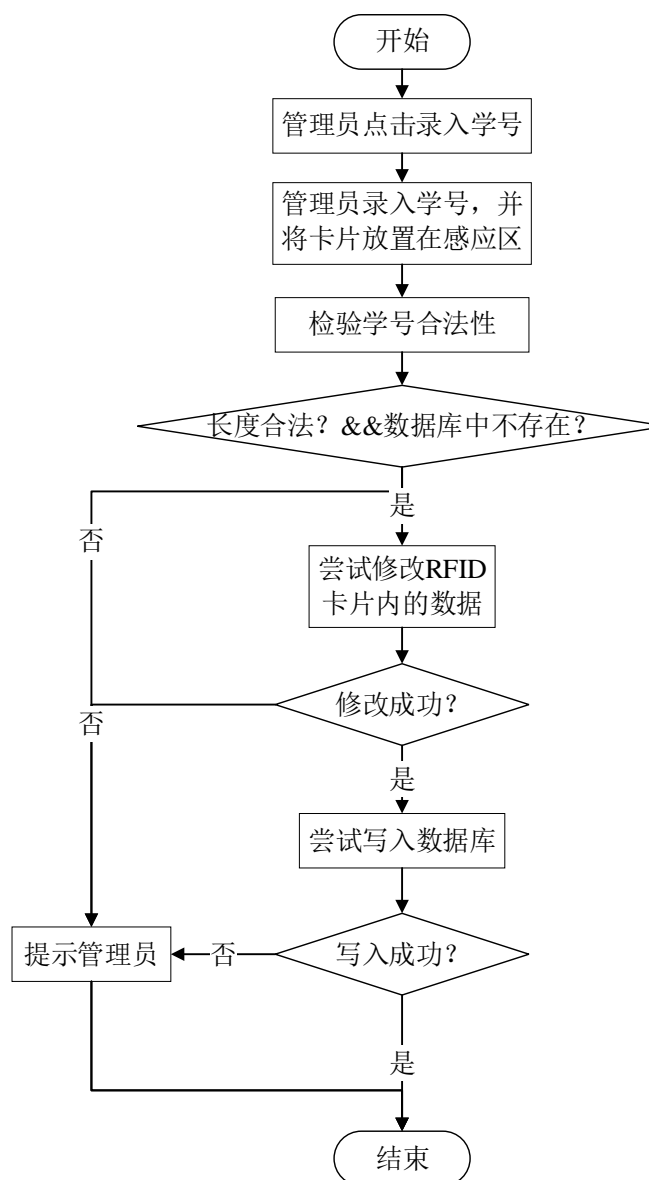


图 2-24 管理员录入学号

## 4. 系统运行与测试

### 4.1. 运行环境搭建

将 RFID 模块与 IMX6 试验平台连接好, 开启 RFID 模块的电源, 运行程序即可。

### 4.2. 运行界面与运行结果展示

部分运行界面已在软件主要功能模块中进行了介绍, 这里将介绍一些未在前面体现出来的界面。

如图 2-25 所示, 可以发现切入到管理员界面时所执行的操作分别有: 键盘输入密码, 在输入结束时自动隐藏数字键盘, 并将前往其他界面的锁打开 (这个锁是为了解决在进行页面切换操作的过程中重复操作的问题), 并校验管理员密码, 最后将会切

到管理员界面，隐藏首页（这里日志中没有体现出来）。

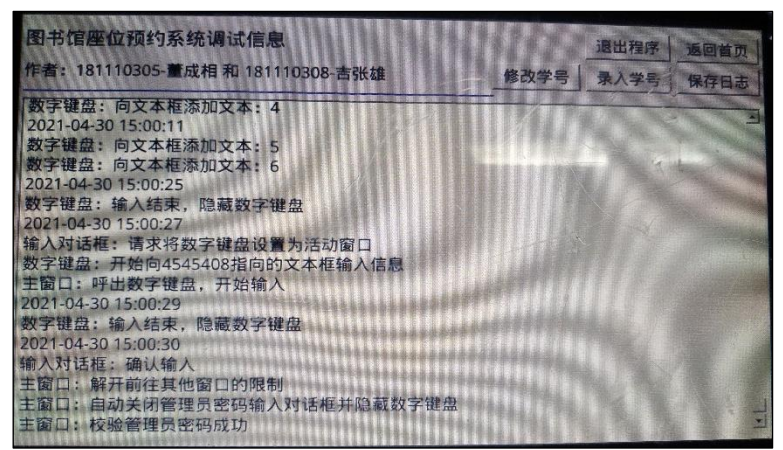


图 2-25 日志信息

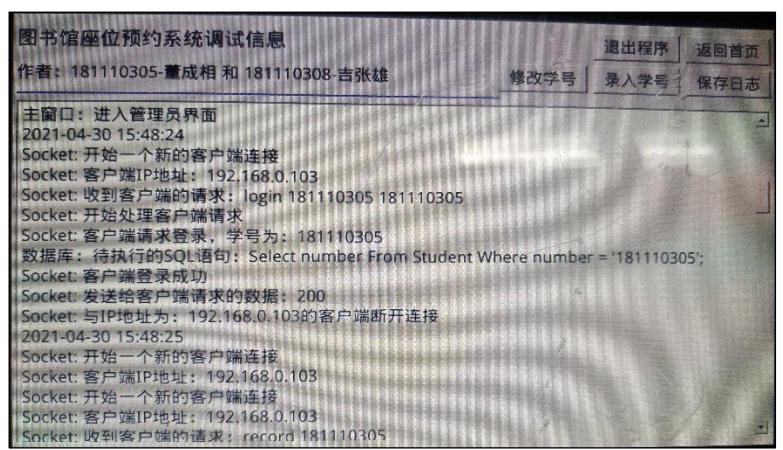


图 2-26 客户端登录日志信息 1

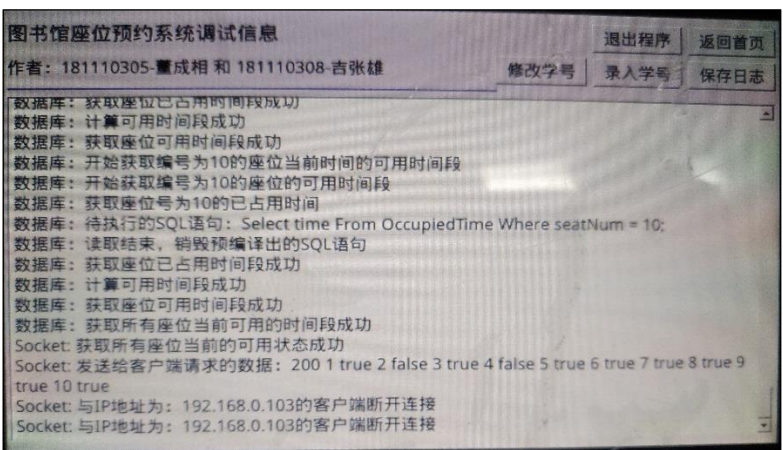


图 2-27 客户端登录日志信息 2

如图 2-26 和 2-27 所示，在客户端登陆时，会先验证登录用户的合法性，随后客户端会继续获取用户的预约记录以及获取所有座位当前的可用状态，这个过程中会进行大量的数据库操作。

## 5. 总结

### 5.1. 亮点与特色

单例模式：使用单例模式实现了数据库、RFID、数字键盘的唯一实例，进而减少了冗余资源的使用，限制了单一资源的使用。

自动切回首页：充分利用信号和槽机制实现了自动从操作界面切回首页的功能。考虑到有些用户可能会在使用完毕之后不主动返回首页，故设计了一个倒计时部件用于自动返回首页。

自动隐藏的消息提示对话框：在自动返回首页之后，可能会存在尚未关闭的对话框，这可能会对后续使用系统的人造成困扰，所以程序需要对所有打开的对话框进行管理，在程序中我使用了一个槽函数来完成这项功能，在操作界面的 `gotoBack()` 方法被调用的时候，会在内部调用现场预约和我的预约两个子部件的 `hideDialog()` 方法，隐藏可能打开的所有对话框，至此问题应当已经解决了（管理员界面也设计了类似的 `gotoBack()` 方法）。后来在使用了错误提示对话框之后，之前的方案似乎不能起作用了，因为是直接调用的静态方法，我拿不到实例变量，所以我索性设计了一个可以在一定时间之后自动关闭的对话框，这个问题就迎刃而解了。

日志系统：利用 QT 的信号与槽机制实现了日志系统的构建，程序运行过程中产生的日志信息全都可以管理员界面查看，为管理员排除问题提供了依据。每次运行程序都会产生新的日志文件，不必担心日志文件过于庞大。

常量组织文件：利用 C++ 的命名空间特性将程序可能会使用到的所有常量保存在一个命名空间中，进而实现类似配置文件的功能。

### 5.2. 不足与可提升之处

输入对话框和数字键盘的信息交互的方式，是通过将输入对话框中的文本框的指针传递给数字键盘，动态更新数字键盘的输入目标实现的，这种方式隐患比较大，把一个类内私有成员交给了外部，很不安全。

系统尚不能够实现自动更新违约信息，也就是用户进行了预约，却没有及时签到，逻辑上是需要将用户的预约信息设置为“已失约”的，但是目前程序尚不能做到。

用户当前可以利用系统连续预约多个座位，即使当前有未完成预约，这是不符合逻辑的，应当限制用户每时每刻只能有一条有效未完成预约。

服务器与安卓客户端的交互尚存在不足之处，这里主要是由于数据的交换格式没有约定好，导致对接的时候出现了一些问题。

数据库的设计没有必要把预约记录表和占用时间表分开。

如需要源代码可附在报告的最后或是上交电子版。

指导教师意见：

成绩： \_\_\_\_\_

日期： \_\_\_\_\_