浙江工业大学

移动应用开发课程实验报告

题目 基于 Android 的智能会议室管理系统

学号 201806010108

班级 软工 1801

姓名 项建航

提交日期 2021.06.20

目录

1.设计目的和任务

1.1 目的

随着人脸识别技术越来越成熟，人脸识别不断的被用于各行各业中，如银行、居住区、城市、校园等场景。为了能够让这样的技术更快更好更容易的应用到生活当中，虹软于 2017 年将自己的人脸识别相关技术免费开放给了公众，大大降低了中小型企业在人脸识别上的门槛。而会议室的管理正是其中一个场景，人脸识别在会议室管理系统中的应用能够更好的解决资源冲突的问题，同时期望能够更高效的召开会议。

会议室的预定和管理问题，在企业、园区、学校等人员\组织多，会议多的地方非常的普遍。纯电脑端的会议预定系统，看的人少，经常发生会议冲突的问题；而在门口填写预定信息，则不知道哪个会议室有空。在企业里，就经常会碰到类似问题：临时开会，找了多个会议室都有人在开会；预定了会议室，有人会议延迟太久结束，影响后面的人开会；有人未预定会议室却在使用。如何准确的知道哪些会议室在什么时间段是空闲的，如何让会议室资源利用的更加高效、同时更加灵活则对会议管理提出了更高的要求。

此类会议室预约系统应用可行性较高，如我们在学校里面的教室预约系统也有着类似的设计思路，实用性较强。

1.2 任务

使用 web 开发和 Android 编写一个智能会议室管理系统，由于本次课程设计是移动应用开发课设，所以主要介绍移动端的软件应用情况。此外由于，该 app 是在 web 端基础上迁移过来的，所以对 web 端也进行一个简要地介绍。

- **总体目的：**
  1) 巩固和加深对 Android 开发基本知识的理解和掌握；
  2) 掌握程序调试的基本技能；
  3) 掌握设计应用软件的基本思路和方法；
  4) 提高运用 Android 应用程序解决实际问题的能力；
  5) 培养设计书写报告的能力。

- **移动端：**

（1）快速预约：

·输入联系电话与人数快速预约会议。

·预约成功后生成预约流水号与二维码。

·通过复制预约流水号与保存二维码分享给其他用户来加入会议。

（2）加入会议：

·输入会议的预约流水号即可加入会议。

·可以通过扫码获取预约流水号。

（3）退出会议

（4）结束会议

（5）会议通知：

·被加入会议会通过消息推送给与会人员

- **web 端：**

（1）普通预定：

·通过输入的会议信息普通预定会议。

·预约成功后生成预约流水号与二维码。

·通过复制预约流水号与保存二维码分享给其他用户来加入会议。

（2）紧急预定：

·临时需要会议室时简单输入手机号和人数紧急预定会议室。

（3）加入会议：

·输入会议的预约流水号即可加入会议。

（4）会议室预约情况查询：

·查看当时起所有被成功预约的会议室预约情况。

（5）用户预约的会议查询：

·查看本会议的所有与会人员并可以对与会人员进行编辑管理。

·可以通过 Excel 文件批量导入与会人员。

·可以取消或完结会议。

（6）用户参加的会议查询：

·可以查看参与的会议。

·可以退出会议。

2.开发环境

2.1 硬件环境

操作系统：

前期：

型号名称： MacBook Pro
处理器名称： 八核 Intel Core i9
处理器速度： 2.3 GHz
处理器数目： 1
核总数： 8
L2 缓存（每个核）： 256 KB
L3 缓存： 16 MB
超线程技术： 已启用
内存： 32 GB

后期：

型号名称： 小米 air13
处理器名称： i5-8250U CPU
处理器速度： 1.8 GHz
内存： 8 GB

解释：前期 web 开发在 mac 上开发，后来在 android 开发遇到较多问题，所以我就迁移到 windows 上进行开发了。

2.2 软件环境

开发平台：Android Studio

开发语言：Java

3.设计题目

3.1 题目名称

基于 Android 的智能会议室管理系统

3.2 题目详细描述

“MRMS 智能会议室管理系统”的创新不仅有技术上的创新，而且结合了许多现在已经成熟的技术，提高了会议室管理系统的效率。

现在的会议室管理系统大多只具备会议室预约、查询功能。但本项目实现的系统能实现人脸识别完成签到开门，考虑了紧急预约优化用户体验，考虑了可以提早 10 分钟签到进入会议室，考虑了可以提早结束会议完结该预约状态，来使会议室的使用效率得到最大化。并且实现了移动端适配，优化用户体验。实现了二维码的生成与扫码使移动端与 web 端巧妙结合并且大大优化用户体验。同时，本项目依然考虑了预约时输入联系电话，而非直接导入用户对于的电话。

本项目实现的系统与同类产品的比较情况如下表所示：

表 1-1 与同类产品的比较

系统名称	存在的问题	主要创新点
五米会议管理系统	无平板端的应用优化	实现了人脸识别
搜麦智能办公系统	无具体座位的预定	实现了人脸识别、紧急预约
DMRS 德睿会议预定系统	无具体座位的预定	实现了紧急预约
钉钉智能会议室	无大数据支持	实现了紧急预约
浙江工业大学图书馆研讨室预约系统	无可用房间的可视化	允许提前 10 分钟进入会议室，允许提前完成会议并结束会议，实现了人脸识别、紧急预约

3.3 功能要求

（1）快速预约：

·输入联系电话与人数快速预约会议。

·预约成功后生成预约流水号与二维码。

·通过复制预约流水号与保存二维码分享给其他用户来加入会议。

（2）加入会议：

·输入会议的预约流水号即可加入会议。

·可以通过扫码获取预约流水号。

（3）退出会议

（4）结束会议

（5）会议通知：

·被加入会议会通过消息推送给与会人员

4.实验体设计与分析

4.1 功能设计

智能会议室管理 APP 的具体功能如下：

1）输入账号密码进行登录；

2）登录失败时会提示相关异常信息；

3）输入电话和人数进行会议室预约；

4）会议室预约成功则返回预约号和相关二维码，预约失败则进行提示；

5）快速加入会议室；

6）二维码扫描，辅助快速加入会议室；

7）消息推送功能，用户被加入会议室时，会向 APP 推送相关信息；

8）取消会议，结束会议；

9）退出会议室

10）退出-退出 APP。

4.2 数据库设计

4.2.1 关系模式设计

（1）管理员 admin（账号 aid，密码 apassword，联系电话 aphone）

（2）用户 user（账号 uid，姓名 uname，密码 upassword，联系电话 uphone，人脸照片路径 upicture）

（3）会议室 room（会议室号 rid，可容纳人数 rnum，维护状态 rstate，位置 raddress）

（4）预约 reserve（预约流水号 reid，用户账户 uid，会议室号 rid，联系电话 rephone，预约状态 state，会议主题 title，会议日期 date，开始时间 starttime，结束时间 endtime，是否开放加入 open）

（5）会议 confernce（用户账户 uid，预约流水号 reid，参会身份 cidentity，签到时间 checkintime

签退时间 signingouttime）

（6）消息推送 information（用户账号 uid，消息记录 itext，会议状态 istate）

4.2.2 数据类型定义

（1）用户数据类型定义表如表所示。

表 4-2-1 用户数据类型定义

用户	数据类型	长度	完整性约束
用户账号 uid	varchar	6	主键
用户姓名 uname	varchar	18	
用户密码 upassword	varchar	18	
联系电话 uphone	varchar	13	
人脸照片路径 upicture	varchar	50	

（2）管理员数据类型定义表如表所示。

表 4-2-2 管理员数据类型定义

管理员	数据类型	长度	完整性约束
管理员账号 aid	Varchar	4	主键
密码 apassword	Varchar	8	
联系电话 aphone	Varchar	13	

（3）会议室数据类型定义表如表所示。

表 4-2-3 会议室数据类型定义

会议室	数据类型	长度	完整性约束
会议室号 rid	varchar	3	主键(前缀 R)
可容纳人数 rnum	int		
维护状态 rstate	varchar	2	0/1
位置 raddress	varchar	100	

（4）预约记录数据类型定义表如表所示。

表 4-2-4 预约记录数据类型定义

预约记录	数据类型	长度	完整性约束
预约流水号 reid	Varchar	30	主键
用户账户 uid	Varchar	6	外键
会议室号 rid	Varchar	3	外键
联系电话 rephone	Varchar	13	
预约状态 state	Varchar	2	0/1/2
会议主题 title	Varchar	50	
会议日期 date	date		
开始时间 starttime	time		
结束时间 endtime	time		
是否开放加入 open	varchar	2	0/1

（5）消息推送数据类型定义表如表所示。

表 4-2-5 消息推送数据类型定义

消息推送	数据类型	长度	完整性约束
用户账号 uid	Varchar	6	主键
消息文本 itext	varchar	100	
会议状态 istate	varchar	6	

（6）会议记录数据类型定义表如表所示。

表 4-2-6 会议记录数据类型定义

会议记录	数据类型	长度	完整性约束
用户账户 uid	varchar	6	主键，外键
预约流水号 reid	varchar	30	主键，外键
参会身份 cidentity	varchar	6	“预约者”或“参会者”
签到时间 checkintime	time		
签退时间 signingouttime	time		

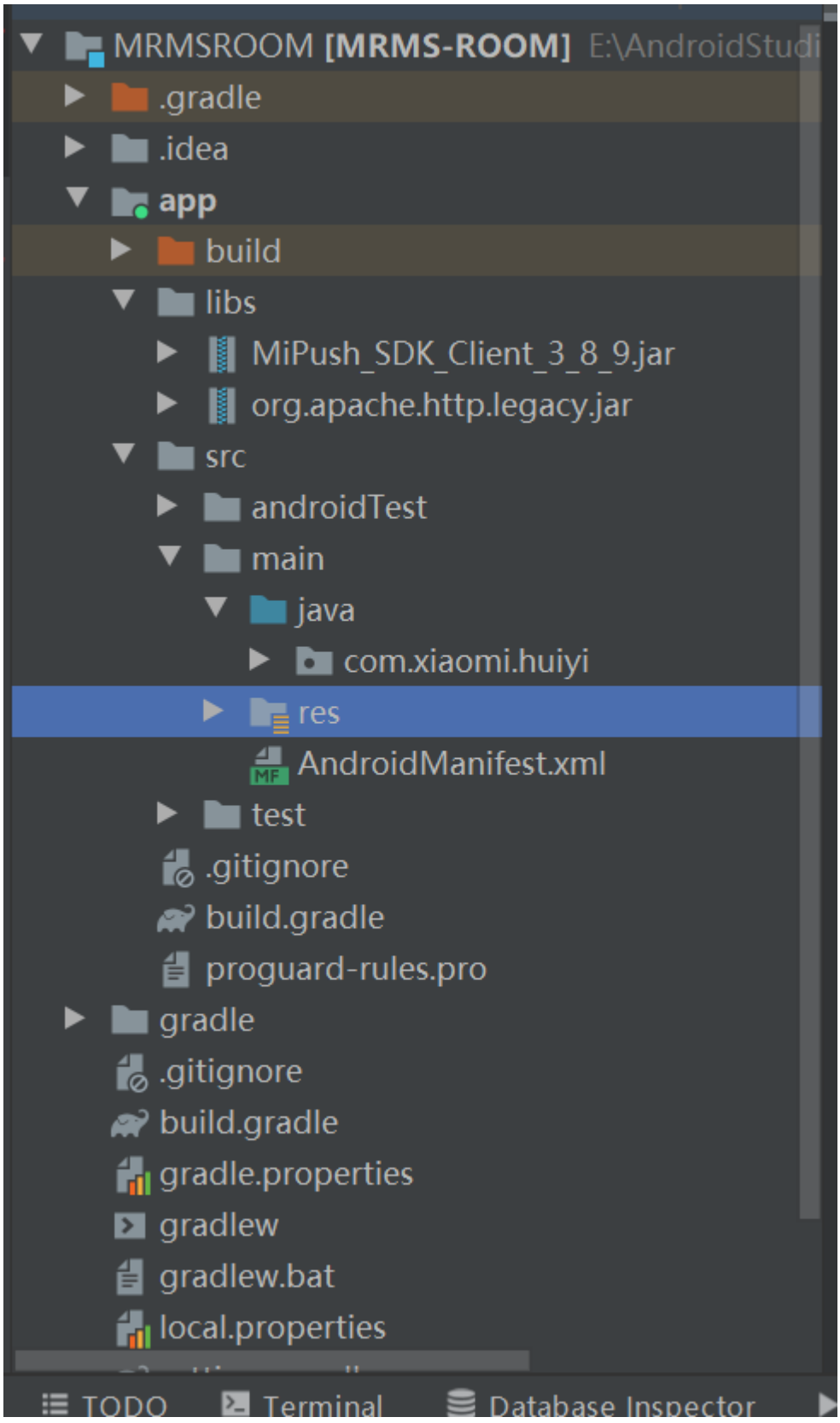
4.3 架构设计

运行界面模块：俄罗斯方块游戏的操作界面中，使用了 ImageView、TextView 等相关控件；通过添加相应的事件响应监听器进行逻辑处理并反馈。

二维码模块：预约二维码生成和扫描板块通过现成的二维码接口实现，从而实现预约成功生成二维码和通过扫描二维码进入会议室。

服务器模块：本项目的相关的后台数据布置在服务器上，通过实现相应的 HTTP 工具来连接远程服务器。

4.4 项目结构


MRMSROOM [MRMS-ROOM] E:\AndroidStudi
.gradle
.idea
app
build
libs
MiPush_SDK_Client_3_8_9.jar
org.apache.http.legacy.jar
src
androidTest
main
java
com.xiaomi.huiyi
res
AndroidManifest.xml
test
.gitignore
build.gradle
proguard-rules.pro
gradle
.gitignore
build.gradle
gradle.properties
gradlew
gradlew.bat
local.properties
TODO
Terminal
Database Inspector

4.5 界面设计

图 4-5-1 登录界面设计

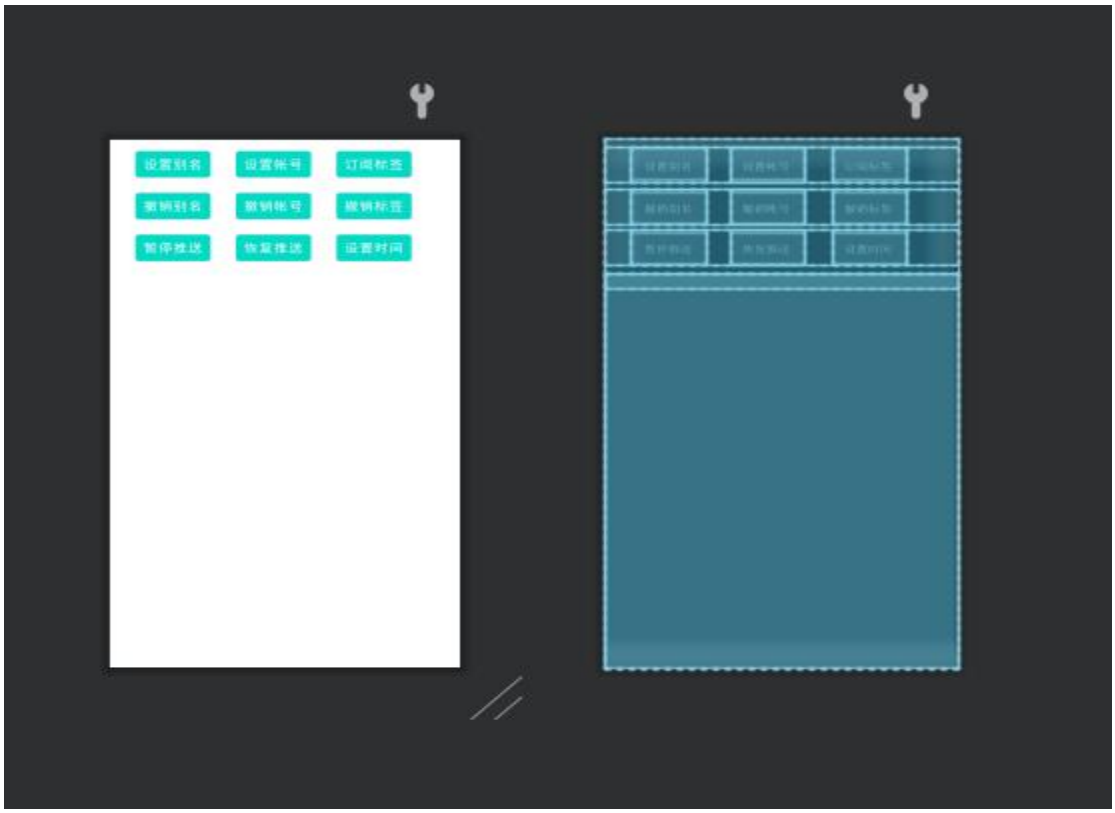


图 4-5-2 main_activity 界面

解释：由于本项目其它界面 UI 类似，此处不展开说明，效果可见下文。

4.6 布局结构设计

例如：activity_main.xml

- 外层：RelativeLayout（相对布局）;
- 内层：

（1） LinearLayout（线性布局）; orientation:vertical(垂直属性)

（2） RelativeLayout（相对布局）

（3） TextView

（4） ImageView：邮箱图标

（5） LinearLayout（线性布局）; orientation:horizontal (水平属性)

（6） RelativeLayout（相对布局）

5.详细设计与实现

5.1 会议室管理 app 的实现

5.1.1APP 图标

图 5-1 图标

5.1.2 初始界面

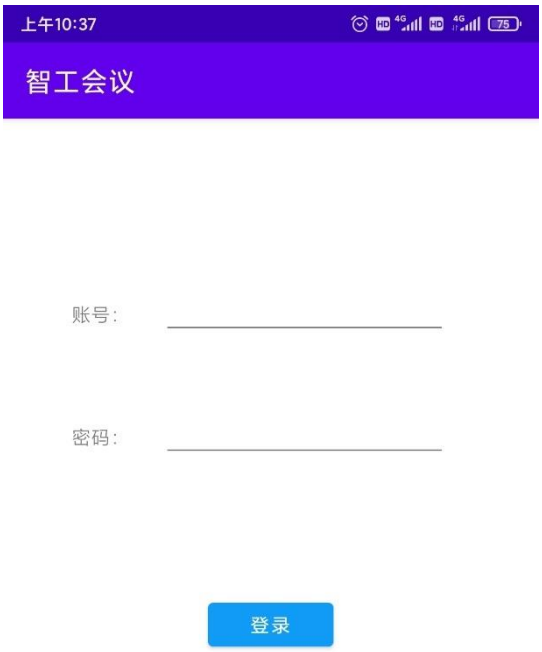


图 5-2 登录界面

解释：安卓端管理主要是实现快速预约会议室，以及通过扫码加入会议室，最大限度的简化用户的操作。进入 APP 需要登录，输入用户名和密码.

5.1.3 登录异常

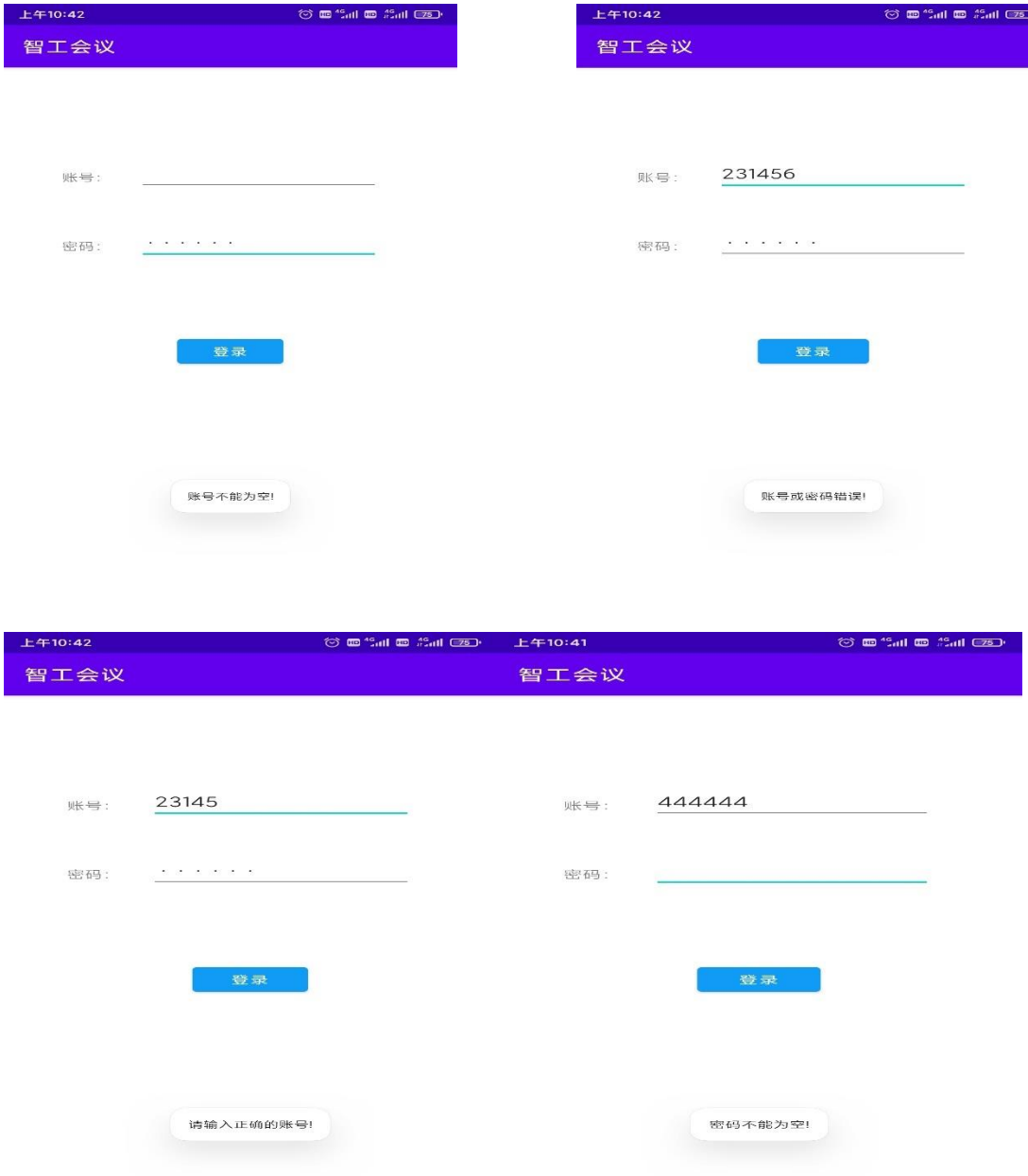


图 5-3 登录异常

解释：如果输入的账号或者密码不符合规范或者不正确会有提示

5.1.4 游戏运行

图 5-4 快速预约界面

解释：登陆完成进入系统后，在这可以进行紧急预约，只要输入手机号码和与会人数，系统会自动匹配会议室

5.1.5 消除行

图 5-5 预约成功界面

解释：如果预约成功，会返回预约流水号和对应的二维码。

5.1.6 预约失败

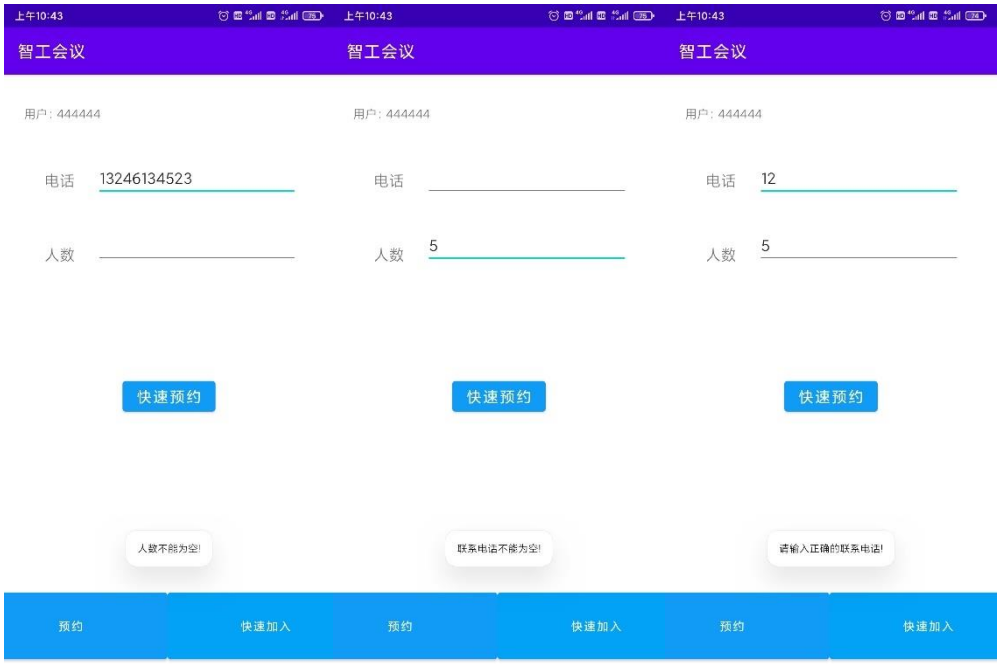


图 5-6 预约失败界面

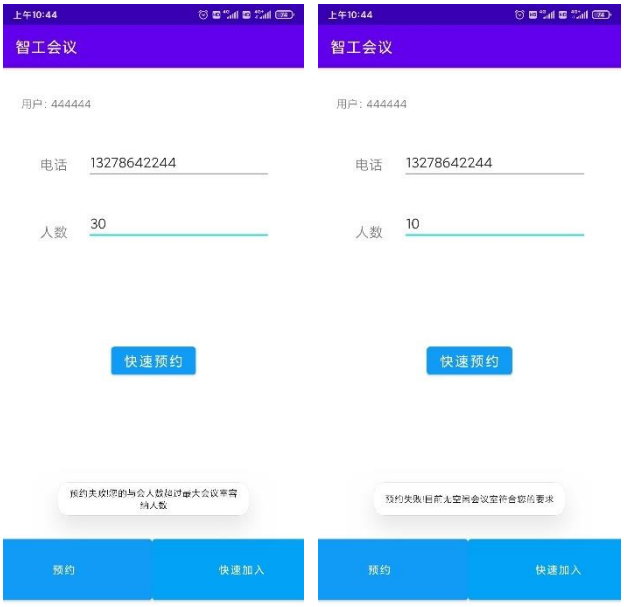


图 5-7 预约失败界面

解释：当前无会议室可用时，以及输入不规范会提示对应的信息

5.1.7 快速加入会议室

图 5-8 快速加入会议室界面

解释：点击下面菜单里的快速加入进入快速加入页面，在这个页面用户可以输入预约流水号，也可以通过扫描二维码获取预约号

5.1.8 扫描二维码页面

图 5-9 扫描二维码页面

解释：点击扫码获取预约号进入扫描二维码界面

5.1.9 加入会议室

图 5-10 加入会议室界面

解释：如果已经在会议里，则会提示异常原因，已在会议里界面

5.1.10 消息推送

图 5-11 消息推送界面

解释：如果已经在会议里，则会提示异常原因，已在会议里界面如图 6-32 所示。

当一位用户被加入到某个会议得时候，服务器就会向其得 app 发送推送消息。

5.1.11 消息推送相关界面

图 5-12 消息通知页面

解释：点击推送即可进入消息页面查看详细消息

5.1.12 我的会议界面

图 5-12 我预约的会议页面

解释：点击我预约的会议可以查看我所有预约的会议，并且可以取消或者结束会议

5.1.13 提出会议相关界面

图 5-13 推出会议页面

解释：点击我参与的会议可以查看我参与的会议，可以选择退出会议

5.2 关键技术

5.2.1 二维码生成

```
public static Bitmap createQRImage(String url, final int width, final int height) {
        try {
            // 判断 URL 合法性
            if (url == null || "".equals(url) || url.length() < 1) {
                return null;
            }
            Hashtable

                    hints = new Hashtable

                    ();
            hints.put(EncodeHintType.CHARACTER_SET, "utf-8");
            // 图像数据转换，使用了矩阵转换
            BitMatrix bitMatrix = new QRCodeWriter().encode(url,
                    BarcodeFormat.QR_CODE, width, height, hints);
            int[] pixels = new int[width * height];
            // 下面这里按照二维码的算法，逐个生成二维码的图片，
            // 两个 for 循环是图片横列扫描的结果
            for (int y = 0; y < height; y++) {
                for (int x = 0; x < width; x++) {
                    if (bitMatrix.get(x, y)) {
                        pixels[y * width + x] = 0xff000000;
                    } else {
                        pixels[y * width + x] = 0xffffffff;
                    }
                }
            }
            // 生成二维码图片的格式，使用 ARGB_8888
            Bitmap bitmap = Bitmap.createBitmap(width, height,
                    Config.ARGB_8888);
            bitmap.setPixels(pixels, 0, width, 0, 0, width, height);
            return bitmap;
        } catch (WriterException e) {
            e.printStackTrace();
        }
        return null;}
```

5.2.2HTTP 远程连接服务器并进行登录验证

```
public class HttpLogin {
    public static String LoginByPost(String uid, String password) {
        String path = "http://h364f35375.qicp.vip/MRoom_MSystem/LoginServlet";
        String result;
        String status=null;
        Map<String, String> map = new HashMap<>();
                map.put("uid", uid);
                map.put("password", password);
                map.put("status", status);
        try{
            HttpClient httpClient = new DefaultHttpClient();
            HttpPost postRequest = new HttpPost(path);
            List<NameValuePair> nameValuePairs = new ArrayList<>();
            for (Map.Entry<String, String> mapItem : map.entrySet()) {
                String key = mapItem.getKey();
                String value = mapItem.getValue();
                NameValuePair nameValuePair = new BasicNameValuePair(key, value);
                nameValuePairs.add(nameValuePair);
            }
            HttpEntity entity = new UrlEncodedFormEntity(nameValuePairs, "UTF-8");
            postRequest.setEntity(entity);
            HttpResponse response = httpClient.execute(postRequest);
            if (response.getStatusLine().getStatusCode() == 200) {
                result = EntityUtils.toString(response.getEntity());
                try {
                    JSONObject jsonObject = new JSONObject(result);
                    status = jsonObject.optString("status");
                } catch (JSONException e) {
                    e.printStackTrace();
                }
                return status;
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
```

5.2.3 二维码扫描

```
@Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        //receive result after your activity finished scanning
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode != RESULT_OK || data == null) {
            return;
        }
        // Obtain the return value of HmsScan from the value returned by the onActivityResult method by using ScanUtil.RESULT as the key value.
        if (requestCode == REQUEST_CODE_SCAN) {
            Object obj = data.getParcelableExtra(ScanUtil.RESULT);
            if (obj instanceof HmsScan) {
                if (!TextUtils.isEmpty(((HmsScan) obj).getOriginalValue())) {
                    String resultData=((HmsScan) obj).getOriginalValue();
                    EditText e1=(EditText) findViewById(R.id.EditText3);
                    Toast.makeText(this, resultData, Toast.LENGTH_SHORT).show();
                    e1.setText(resultData);
                }
                return;
            }
        }
    }
    public void newViewBtnClick(View view) {
        Log.d("11","扫码");
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            this.requestPermissions(
                    new String[]{Manifest.permission.CAMERA, Manifest.permission.READ_EXTERNAL_STORAGE},
                    DEFAULT_VIEW);
        }
    }
```

5.2.4 消息推送机制

```
@Override
    public void onReceivePassThroughMessage(Context context, MiPushMessage
message) {
        Log.v(DemoApplication.TAG,
                "onReceivePassThroughMessage is called. " +
message.toString());
        String log = context.getString(R.string.recv_passthrough_message,
message.getContent());
        MainActivity.logList.add(0, getSimpleDate() + " " + log);

        if (!TextUtils.isEmpty(message.getTopic())) {
            mTopic = message.getTopic();
        } else if (!TextUtils.isEmpty(message.getAlias())) {
            mAlias = message.getAlias();
        }

        Message msg = Message.obtain();
        msg.obj = log;
        DemoApplication.getHandler().sendMessage(msg);
    }

    @Override
    public void onNotificationMessageClicked(Context context, MiPushMessage
message) {
        Log.v(DemoApplication.TAG,
                "onNotificationMessageClicked is called. " +
message.toString());
        String log = context.getString(R.string.click_notification_message,
message.getContent());
       //  MainActivity.logList.add(0, getSimpleDate() + " " + log);

        if (!TextUtils.isEmpty(message.getTopic())) {
            mTopic = message.getTopic();
        } else if (!TextUtils.isEmpty(message.getAlias())) {
            mAlias = message.getAlias();
        }
        List list= MiPushClient.getAllUserAccount(context);
        String uid=list.get(list.size()-1).toString();
       /*  Intent intent= new Intent();
       intent.setClass(context, com.xiaomi.huiyi.ReserveActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        context.startActivity(intent);*/
        Intent intent = new Intent();
```

```
            intent.setClass(context, com.xiaomi.huiyi.InformationActivity.class);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            intent.putExtra("uid",uid);
            context.startActivity(intent);
            Message msg = Message.obtain();
            if (message.isNotified()) {
                msg.obj = log;
            }
            DemoApplication.getHandler().sendMessage(msg);
        }

        @Override
        public void onNotificationMessageArrived(Context context, MiPushMessage
message) {
            Log.v(DemoApplication.TAG,
                    "onNotificationMessageArrived is called. " +
message.toString());
           // String log = context.getString(R.string.arrive_notification_message,
message.getContent());
            String log = message.getContent();
            MainActivity.logList.add(0, getSimpleDate() + " " + log);

            if (!TextUtils.isEmpty(message.getTopic())) {
                mTopic = message.getTopic();
            } else if (!TextUtils.isEmpty(message.getAlias())) {
                mAlias = message.getAlias();
            }
            Message msg = Message.obtain();
            msg.obj = log;
            DemoApplication.getHandler().sendMessage(msg);
            Uri uri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
            Ringtone ringtone = RingtoneManager.getRingtone(context, uri);
            ringtone.play();
            Vibrator vibrator = (Vibrator)
context.getSystemService(Service.VIBRATOR_SERVICE);
            // 暂停 500 毫秒，振动 0.3 秒，暂停 150 毫秒，振动 300 秒(可自行调
节震幅)
            long[] vibrationPattern = new long[]{500, 300, 150, 300};
            // 第一个参数为开关开关的时间，第二个参数是重复次数，振动需要
添加权限
            vibrator.vibrate(vibrationPattern, -1);//-1 表示震动一次，0 表示重复震
动
        }
```

5.2.5 时间间隔记录

```
private Button.OnClickListener clickListener = new Button.OnClickListener() {

        @Override
        public void onClick(View v) {
            switch (v.getId()) {
                case R.id.apply:
                    dismiss();
                    mTimeIntervalInterface.apply(mStartHour,  mStartMinute,
mEndHour, mEndMinute);
                    break;
                case R.id.cancel:
                    dismiss();
                    mTimeIntervalInterface.cancel();
                    break;
                default:
                    break;
            }
        }
    };
```

5.2.6HTTP 连接服务器并验证预约情况

```
public static String ReserveByPost(String uid, String number, String phone){
        String path = "http://h364f35375.qicp.vip/MRoom_MSystem/ReserveServlet";
        String result;
        String reid=null;
        Map<String,String> map = new HashMap<>();
        map.put("uid", uid);
        map.put("number", number);
        map.put("phone", phone);
        map.put("reid", reid);
        try{
            HttpClient httpClient = new DefaultHttpClient();
            HttpPost postRequest = new HttpPost(path);
            List<NameValuePair> nameValuePairs = new ArrayList<>();
            for (Map.Entry<String, String> mapItem : map.entrySet()) {
                String key = mapItem.getKey();
                String value = mapItem.getValue();
                NameValuePair nameValuePair = new BasicNameValuePair(key, value);
                nameValuePairs.add(nameValuePair);
            }
            HttpEntity entity = new UrlEncodedFormEntity(nameValuePairs, "UTF-8");
            postRequest.setEntity(entity);
            HttpResponse response = httpClient.execute(postRequest);
            if (response.getStatusLine().getStatusCode() == 200) {
                result = EntityUtils.toString(response.getEntity());
                try {
                    JSONObject jsonObject = new JSONObject(result);
                    reid = jsonObject.optString("reid");
                } catch (JSONException e) {
                    e.printStackTrace();
                }
                return reid;
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return null;
        }
```

5.2.7HTTP 连接服务器并进入会议室

```
public static String AttendByPost(String uid, String reid){
        String path = "http://h364f35375.qicp.vip/MRoom_MSystem/AttendServlet";
        String result;
        String status=null;
        Map<String,String> map = new HashMap<>();
        map.put("uid", uid);
        map.put("reid", reid);
        map.put("status",status);
        try{
            HttpClient httpClient = new DefaultHttpClient();
            HttpPost postRequest = new HttpPost(path);
            List<NameValuePair> nameValuePairs = new ArrayList<>();
            for (Map.Entry<String, String> mapItem : map.entrySet()) {
                String key = mapItem.getKey();
                String value = mapItem.getValue();
                NameValuePair nameValuePair = new BasicNameValuePair(key, value);
                nameValuePairs.add(nameValuePair);
            }
            HttpEntity entity = new UrlEncodedFormEntity(nameValuePairs, "UTF-8");
            postRequest.setEntity(entity);
            HttpResponse response = httpClient.execute(postRequest);
            if (response.getStatusLine().getStatusCode() == 200) {
                result = EntityUtils.toString(response.getEntity());
                try {
                    JSONObject jsonObject = new JSONObject(result);
                    status = jsonObject.optString("status");
                } catch (JSONException e) {
                    e.printStackTrace();
                }
                return status;
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return null;
        }
```

5.2.8HTTP 连接服务器并消息推送

```
public static List<Information> InformationByPost(String uid){
        String path = "http://h364f35375.qicp.vip/MRoom_MSystem/InforServlet";
        String result; List<Information> list = new ArrayList<Information>();
        Map<String,String> map = new HashMap<>();map.put("uid", uid);
        try{ HttpClient httpClient = new DefaultHttpClient();
            HttpPost postRequest = new HttpPost(path);
            List<NameValuePair> nameValuePairs = new ArrayList<>();
            for (Map.Entry<String, String> mapItem : map.entrySet()) {
                String key = mapItem.getKey();
                String value = mapItem.getValue();
                NameValuePair nameValuePair = new BasicNameValuePair(key, value);
                nameValuePairs.add(nameValuePair);
            }
      HttpEntity entity = new UrlEncodedFormEntity(nameValuePairs, "UTF-8");
            postRequest.setEntity(entity);
            HttpResponse response = httpClient.execute(postRequest);
            if (response.getStatusLine().getStatusCode() == 200) {
                result = EntityUtils.toString(response.getEntity());
                try {
                    JSONArray jsonArray=new JSONArray(result);
                    JSONObject jsonObject;
                    for(int i=0;i<jsonArray.length();i++){
                        jsonObject = jsonArray.getJSONObject(i);
                        Information information=new Information();
                        User user=new User();
                        user.setUid(uid);
                        information.setUser(user);
                        information.setUid(user.getUid());
                        information.setItext(jsonObject.getString("itext"));
                        information.setIstate(jsonObject.getString("istate"));
                        list.add(information);
                    }
                } catch (JSONException e) {e.printStackTrace(); }
                return list;
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        return null;
        }
```

6.实验总结

在本次课程设计中利用 Android 开发了一个智能会议室管理系统，熟悉了 Android 下各种控件的使用。本次实验中最重要的是各个方块类的编写，其定义的好坏和封装性的良好是整个程序运行的基础，属于程序的业务逻辑功能块，主框架中通过调用这些类，实现程序的表示层。

本项目基本实现了我们所需求的功能、模块，在此过程中，由于本次 APP 开发基于 web 端的管理网站，并在相关基础上进行迁移开发，因此我也更加熟悉了 SSH 框架，实现了人脸识别接口的实现，二维码的生成与扫码，消息推送。在本项目中，我未使用 android 自带的 SQLite 接口，而是使用了 mysql 数据库存储相关数据，当然由于项目时从 web 端迁移过来的，所以本项目的数据来自于服务器端的数据库。在 Android 部分的开发，需要我查询文献、询问学长，最后我们不仅完成了 Android 部分的功能，还使用了内网穿透实现各个设备都能连接到一个服务器上，而非几个设备都开启服务器，我认为这也是本项目的一个特点。

此外，在实验中，我体会到开发一个工程时，应该先制定好程序的框架，规划好相应的功能模块，使程序模块化，易于日后的扩展和完善。在项目初期开发时，应该先有一个大体的开发思路并可以通过 UML 图等形式将设计构建出来，其次是程序的数据结构，良好的数据结构能使程序高效化，功能强大。