同济大学学生宿舍管理系统

编码文档

小组成员：涂远鹏-1652262

刘铸煌-1652313

黎盛烜-1652310

指导老师：王继成

一.编程语言

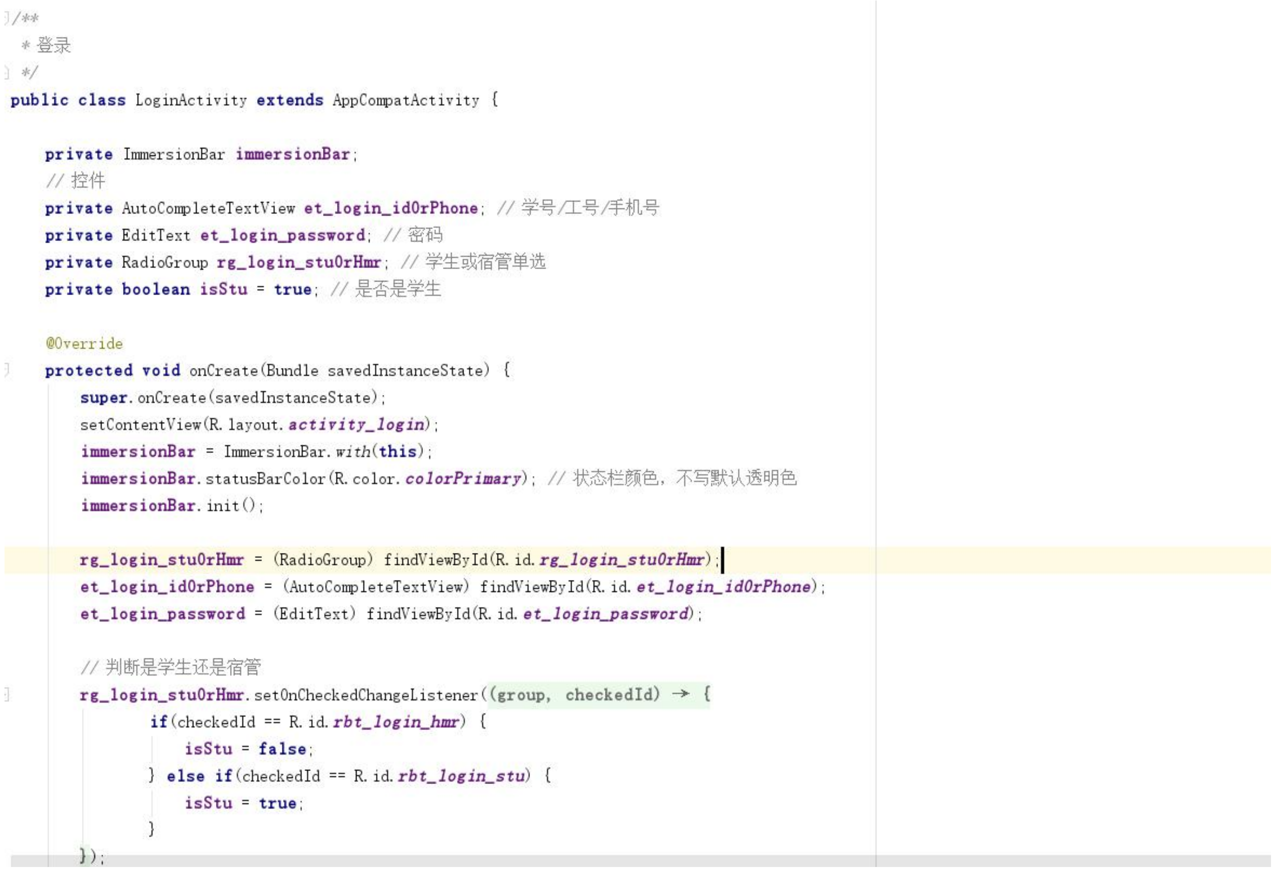
采用的编程语言为面向对象的高级编程语言 java，使用的编程软件为 android开发常用的version 2.2的android studio软件，编译环境为JDK10.0.1版本，JRE8.1.1 版本的 java 编译环境。

二. 关键算法

由于本项目中学生用户功能实现主要为在线数据库内容的查询，宿舍管理员功能实现主要为数据库的写入与查询而人事管理部门功能实现为在线数据库内容的增删改减，与算法实现关系甚微，此部分略去。

三. 关键代码（仅实现部分功能的代码，全部代码在项目代码压缩包中）

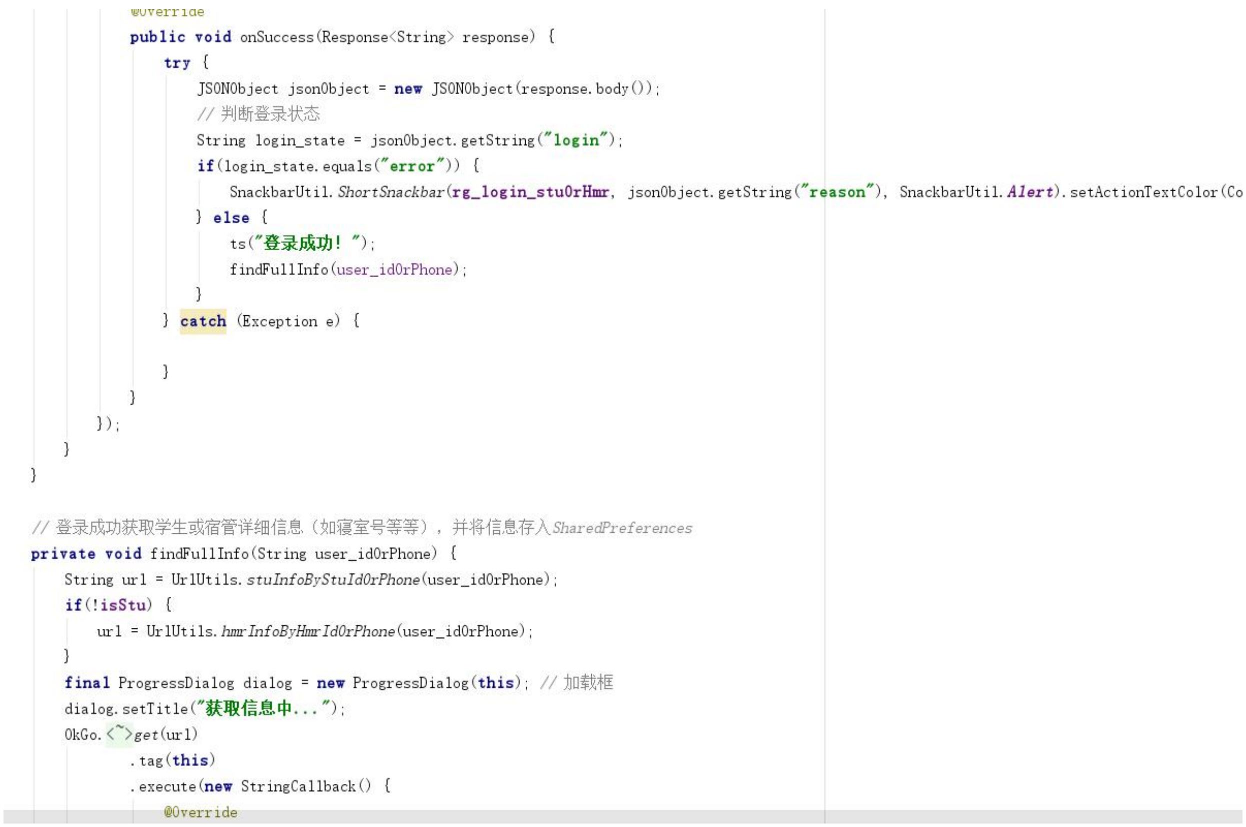
登录功能实现部分：通过登录用户的学工号判断是否为学生用户/人事管理人员用户/宿管用户，并设置 click_forgetPwd 设置忘记密码选项，登陆成功时，将访问云端数据库将登录数据及个人信息存入 sp



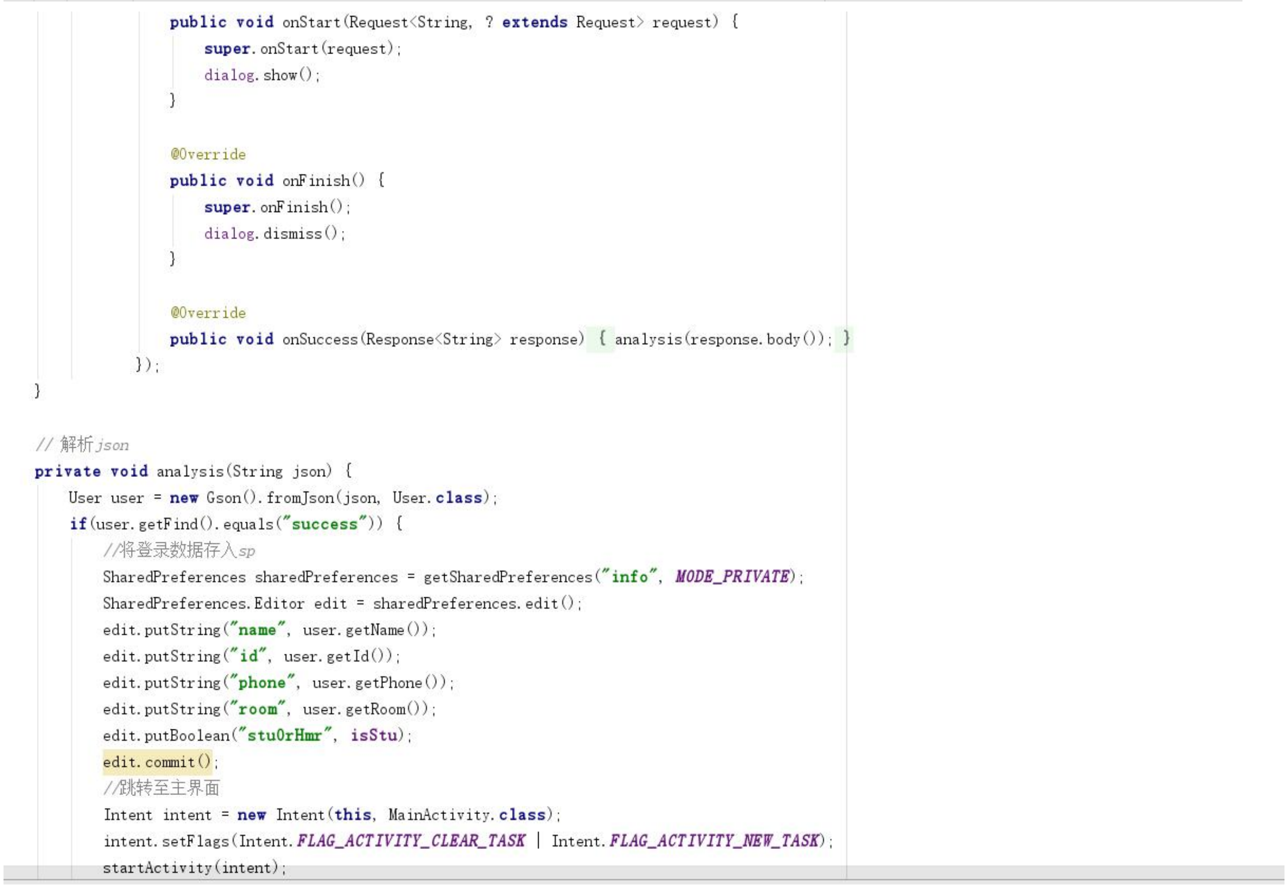
```java
/**
 * 登录
 */
public class LoginActivity extends AppCompatActivity {

    private ImmersionBar immersionBar;
    // 控件
    private AutoCompleteTextView et_login_idOrPhone; // 学号/工号/手机号
    private EditText et_login_password; // 密码
    private RadioGroup rg_login_stuOrHmr; // 学生或宿管单选
    private boolean isStu = true; // 是否是学生

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        immersionBar = ImmersionBar.with(this);
        immersionBar.statusBarColor(R.color.colorPrimary); // 状态栏颜色，不写默认透明色
        immersionBar.init();

        rg_login_stuOrHmr = (RadioGroup) findViewById(R.id.rg_login_stuOrHmr);
        et_login_idOrPhone = (AutoCompleteTextView) findViewById(R.id.et_login_idOrPhone);
        et_login_password = (EditText) findViewById(R.id.et_login_password);

        // 判断是学生还是宿管
        rg_login_stuOrHmr.setOnCheckedChangeListener((group, checkedId) → {
                if(checkedId == R.id.rbt_login_hmr) {
                    isStu = false;
                } else if(checkedId == R.id.rbt_login_stu) {
                    isStu = true;
                }
        });
```

```java
// 登录
public void click_login(View view) {
    // 重置错误
    et_login_idOrPhone.setError(null);
    et_login_password.setError(null);

    // 类型强转
    final String user_idOrPhone = et_login_idOrPhone.getText().toString().trim();
    String user_password = et_login_password.getText().toString().trim();

    // 验证输入是否有空
    if (TextUtils.isEmpty(user_idOrPhone)) {
        et_login_idOrPhone.setError("此字段不能为空！");
        et_login_idOrPhone.requestFocus();
    } else if (TextUtils.isEmpty(user_password)) {
        et_login_password.setError("此字段不能为空！");
        et_login_password.requestFocus();
    } else {
        // 登录验证，网络请求
        String url = UrlUtils.stuLogin;
        if(!isStu) {
            url = UrlUtils.hmrLogin;
        }
        PostRequest<String> tag = OkGo.<~>post(url).tag(this);
        // 添加请求参数
        if(user_idOrPhone.length() == 11) {
            if(isStu) {
                tag.params("stuPhone", user_idOrPhone);
            } else {
                tag.params("hmrPhone", user_idOrPhone);
```

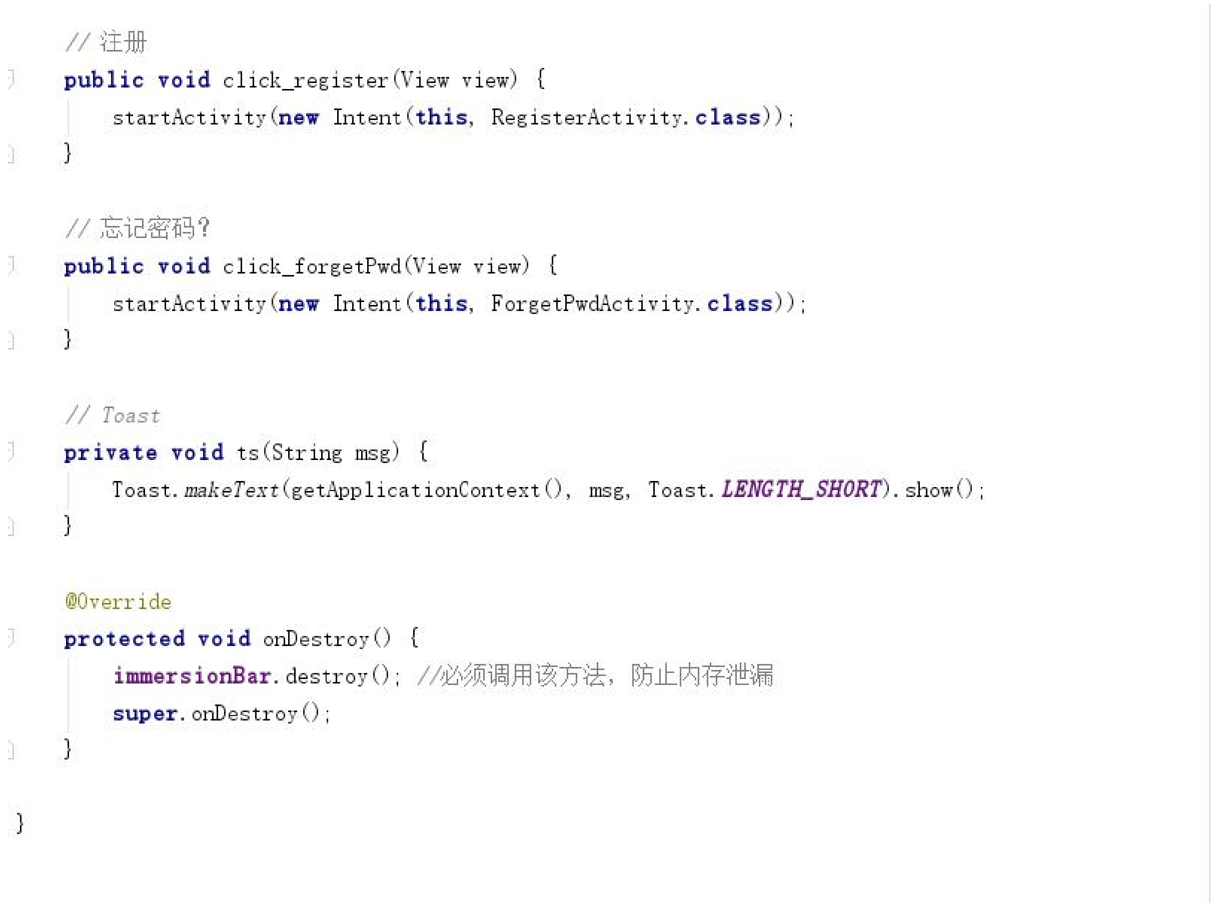
```java
        } else {
            if(isStu) {
                tag.params("stuId", user_idOrPhone);
            } else {
                tag.params("hmrId", user_idOrPhone);
            }
        }
        if(isStu) {
            tag.params("stuPwd", user_password);
        } else {
            tag.params("hmrPwd", user_password);
        }
        // 解析json
        final ProgressDialog dialog = new ProgressDialog(this); // 加载框
        dialog.setTitle("登录中...");
        tag.execute(new StringCallback() {
            @Override
            public void onStart(Request<String, ? extends Request> request) {
                super.onStart(request);
                dialog.show();
            }

            @Override
            public void onFinish() {
                super.onFinish();
                dialog.dismiss();
            }

            @Override
            public void onSuccess(Response<String> response) {
                try {
                    JSONObject jsonObject = new JSONObject(response.body());
                    // 判断登录状态
```



```
                @Override
                public void onSuccess(Response<String> response) {
                    try {
                        JSONObject jsonObject = new JSONObject(response.body());
                        // 判断登录状态
                        String login_state = jsonObject.getString("login");
                        if(login_state.equals("error")) {
                            SnackbarUtil.ShortSnackbar(rg_login_stuOrHmr, jsonObject.getString("reason"), SnackbarUtil.Alert).setActionTextColor(Co
                        } else {
                            ts("登录成功! ");
                            findFullInfo(user_idOrPhone);
                        }
                    } catch (Exception e) {

                    }
                }
            });
    }
}

// 登录成功获取学生或宿管详细信息（如寝室号等等），并将信息存入SharedPreferences
private void findFullInfo(String user_idOrPhone) {
    String url = UrlUtils.stuInfoByStuIdOrPhone(user_idOrPhone);
    if(!isStu) {
        url = UrlUtils.hmrInfoByHmrIdOrPhone(user_idOrPhone);
    }
    final ProgressDialog dialog = new ProgressDialog(this); // 加载框
    dialog.setTitle("获取信息中...");
    OkGo.<~>get(url)
            .tag(this)
            .execute(new StringCallback() {
                @Override
```

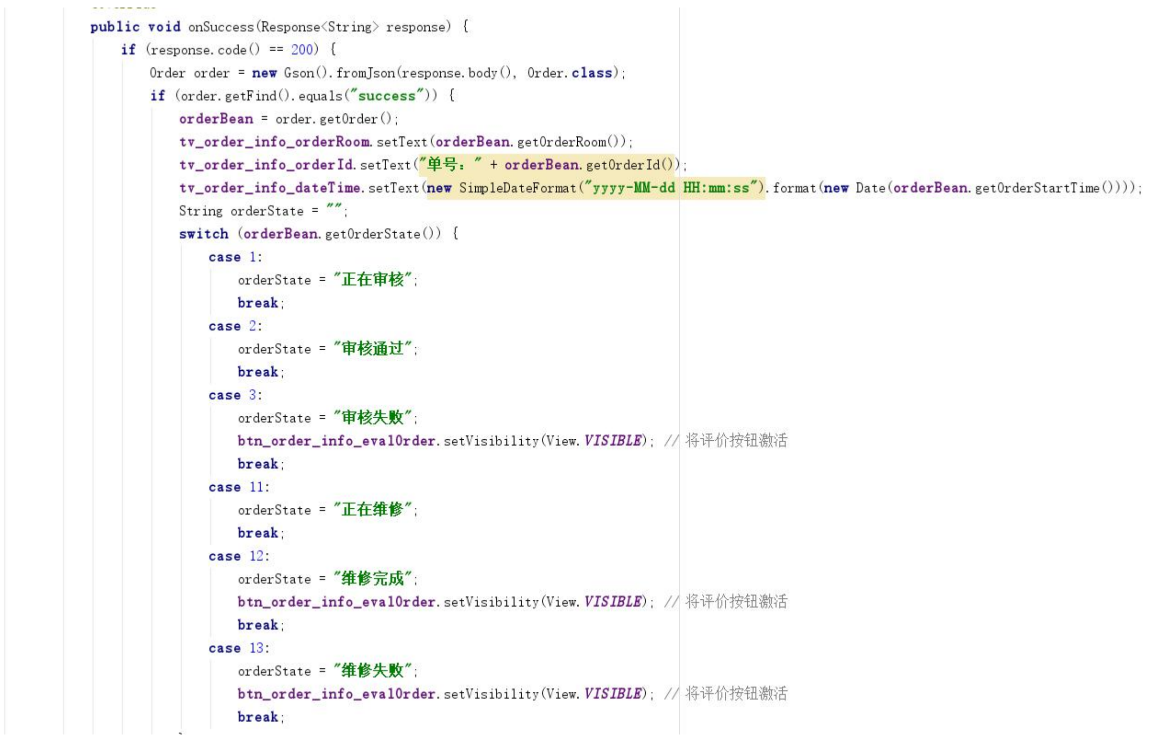


```
                public void onStart(Request<String, ? extends Request> request) {
                    super.onStart(request);
                    dialog.show();
                }

                @Override
                public void onFinish() {
                    super.onFinish();
                    dialog.dismiss();
                }

                @Override
                public void onSuccess(Response<String> response) { analysis(response.body()); }
            });
}

// 解析json
private void analysis(String json) {
    User user = new Gson().fromJson(json, User.class);
    if(user.getFind().equals("success")) {
        //将登录数据存入sp
        SharedPreferences sharedPreferences = getSharedPreferences("info", MODE_PRIVATE);
        SharedPreferences.Editor edit = sharedPreferences.edit();
        edit.putString("name", user.getName());
        edit.putString("id", user.getId());
        edit.putString("phone", user.getPhone());
        edit.putString("room", user.getRoom());
        edit.putBoolean("stuOrHmr", isStu);
        edit.commit();
        //跳转至主界面
        Intent intent = new Intent(this, MainActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
```

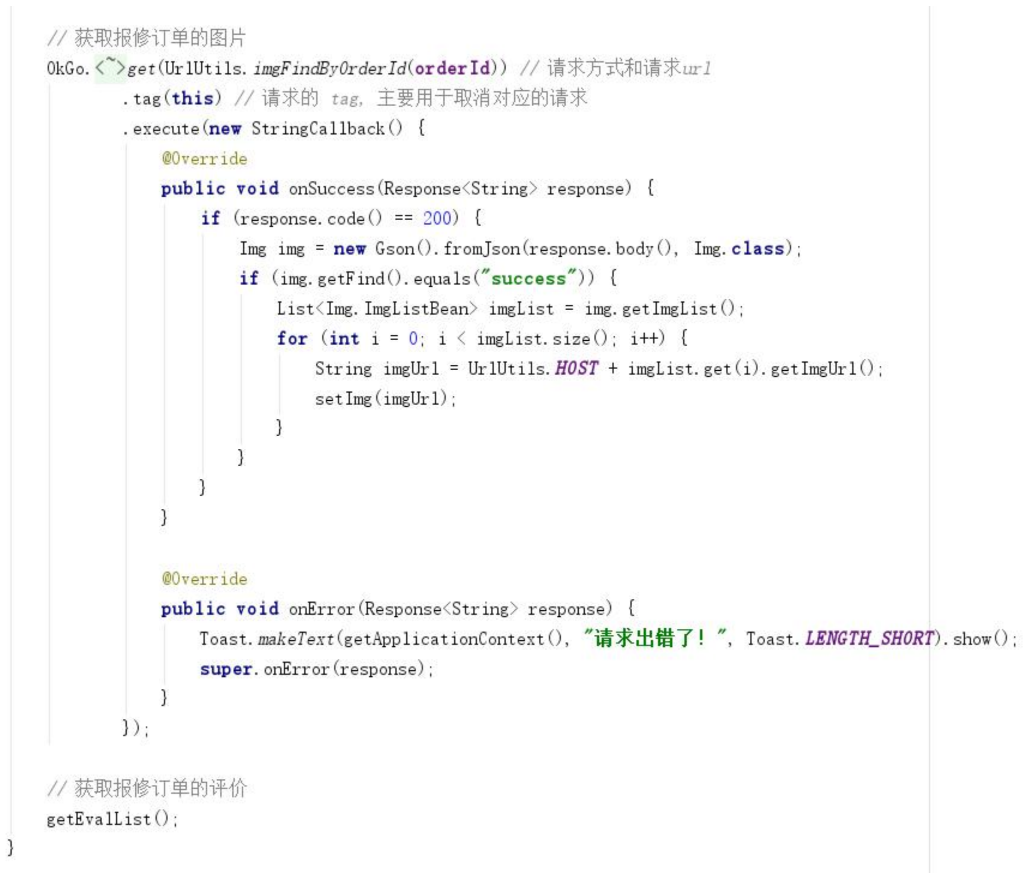


```java
    // 注册
    public void click_register(View view) {
        startActivity(new Intent(this, RegisterActivity.class));
    }

    // 忘记密码?
    public void click_forgetPwd(View view) {
        startActivity(new Intent(this, ForgetPwdActivity.class));
    }

    // Toast
    private void ts(String msg) {
        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onDestroy() {
        immersionBar.destroy(); //必须调用该方法，防止内存泄漏
        super.onDestroy();
    }

}
```

获取报修处理详情以及进行报修评价功能实现代码：



```java
    public void onSuccess(Response<String> response) {
        if (response.code() == 200) {
            Order order = new Gson().fromJson(response.body(), Order.class);
            if (order.getFind().equals("success")) {
                orderBean = order.getOrder();
                tv_order_info_orderRoom.setText(orderBean.getOrderRoom());
                tv_order_info_orderId.setText("单号：" + orderBean.getOrderId());
                tv_order_info_dateTime.setText(new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new Date(orderBean.getOrderStartTime())));
                String orderState = "";
                switch (orderBean.getOrderState()) {
                    case 1:
                        orderState = "正在审核";
                        break;
                    case 2:
                        orderState = "审核通过";
                        break;
                    case 3:
                        orderState = "审核失败";
                        btn_order_info_evalOrder.setVisibility(View.VISIBLE); // 将评价按钮激活
                        break;
                    case 11:
                        orderState = "正在维修";
                        break;
                    case 12:
                        orderState = "维修完成";
                        btn_order_info_evalOrder.setVisibility(View.VISIBLE); // 将评价按钮激活
                        break;
                    case 13:
                        orderState = "维修失败";
                        btn_order_info_evalOrder.setVisibility(View.VISIBLE); // 将评价按钮激活
                        break;
```



```java
    // 获取报修订单的图片
    OkGo.<~>get(UrlUtils.imgFindByOrderId(orderId)) // 请求方式和请求url
            .tag(this) // 请求的 tag, 主要用于取消对应的请求
            .execute(new StringCallback() {
                @Override
                public void onSuccess(Response<String> response) {
                    if (response.code() == 200) {
                        Img img = new Gson().fromJson(response.body(), Img.class);
                        if (img.getFind().equals("success")) {
                            List<Img.ImgListBean> imgList = img.getImgList();
                            for (int i = 0; i < imgList.size(); i++) {
                                String imgUrl = UrlUtils.HOST + imgList.get(i).getImgUrl();
                                setImg(imgUrl);
                            }
                        }
                    }
                }

                @Override
                public void onError(Response<String> response) {
                    Toast.makeText(getApplicationContext(), "请求出错了！", Toast.LENGTH_SHORT).show();
                    super.onError(response);
                }
            });

    // 获取报修订单的评价
    getEvalList();
}
```

动画效果实现：
实现多个 View 间的顺序执行动画
1.AnimManager 实现 AnimationListener 接口，并且提供接口输入 Views 和 Anims。

2.使用时，需在代码中初始化，然后传入 Views 与 Anims，最后执行 startAnimation()。

3.startAnimation()中检测 Views 和 Anims 的大小并抛出异常，最后执行 excute();

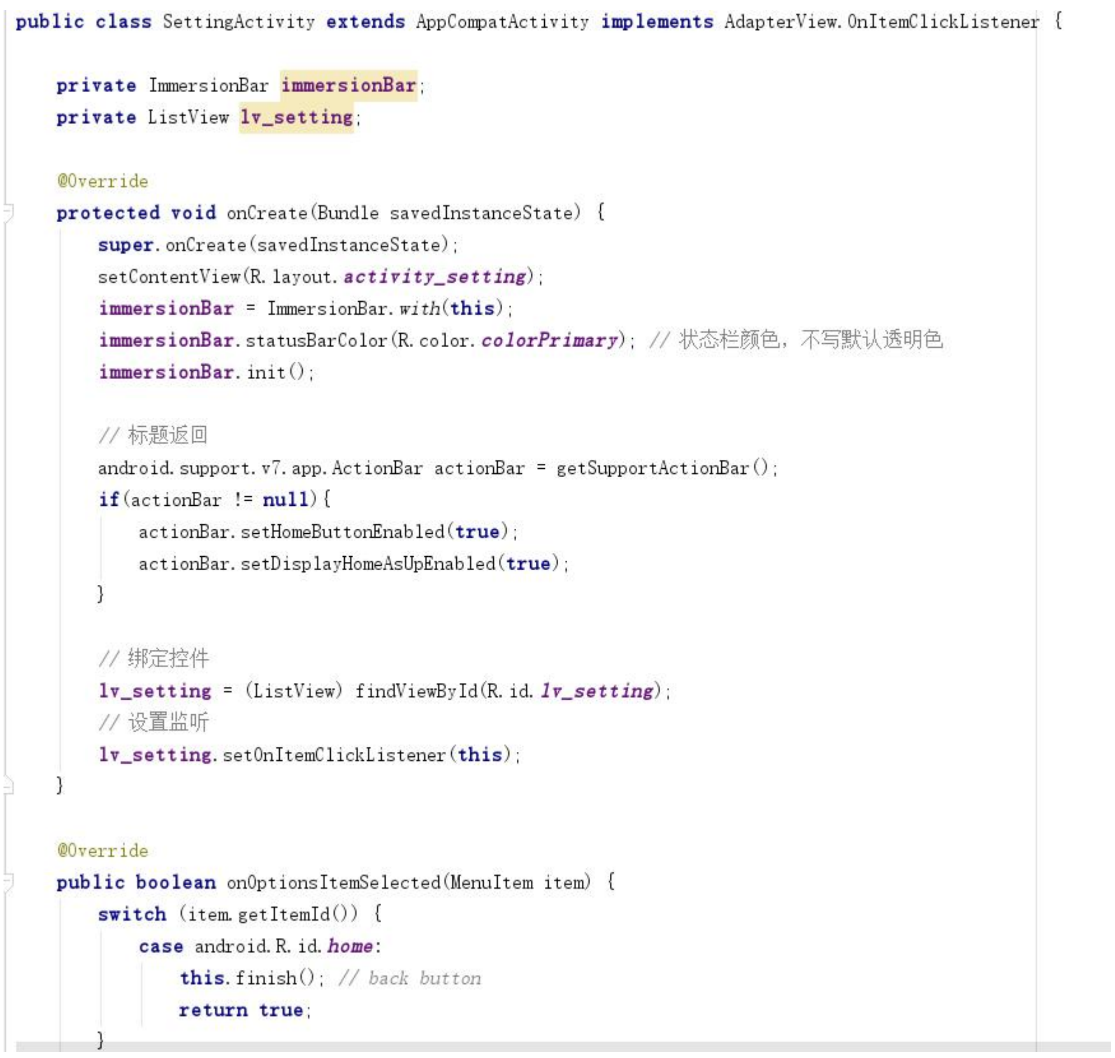
4.excute()中获取当前 position 的 View 和 Anim 并设置 AnimationListener，最后执行当前 position 的动画

5.重写了 onAnimationStart()和 onAnimationEnd() 其中 onAnimationStart()将当前 View 的可见性设置为 VISIBLE,需要在代码中将 Views 中所有 View 事先设置为不可见 其中 onAnimationEnd()将 position 加 1 后再次执行 excute()，实现顺序执行

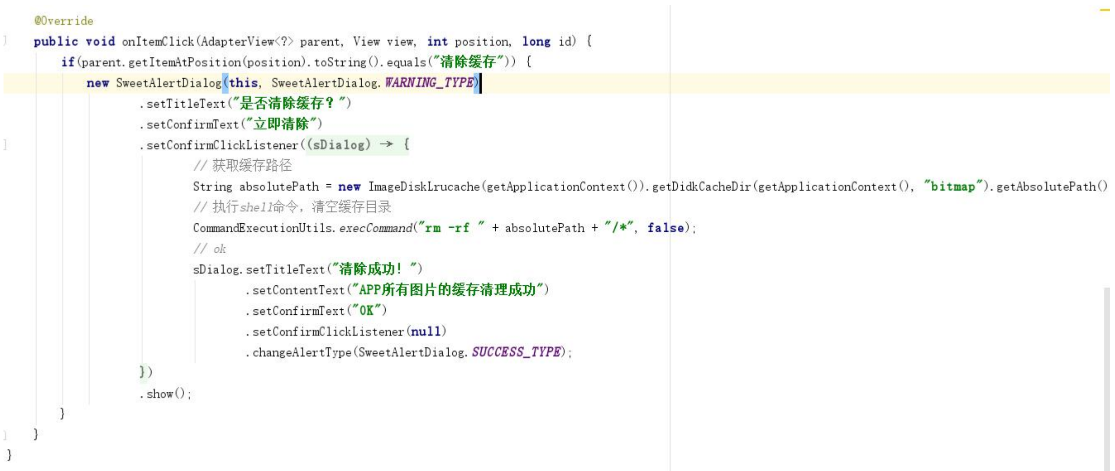
```java
public class AnimManager implements Animation.AnimationListener {
    private final List<View> views;
    private final List<Animation> anims;
    private boolean isLast;
    private int position;
    private View view;
    private Animation animation;
    //构造函数
    public AnimManager(List<View> views,List<Animation> anims){
        this.views = views;
        this.anims = anims;
    }
    @Override
    public void onAnimationStart(Animation animation) {
        if(view == null){
            throw new RuntimeException("第" + position + "个 View 是空的");
        }
        view.setVisibility(View.VISIBLE);
    }

    @Override
    public void onAnimationEnd(Animation animation) {
        if(position == views.size() - 1){
            //动画执行结束
            return;
        }
```

```
            position++;
            excute();
        }
        @Override
        public void onAnimationRepeat(Animation animation) {
        }
        public void startAnimation(){
            if(views == null || views.size() < 0){
                throw new RuntimeException("views 集合为空！无法开始动画！");
            }
            if(anims ==null || anims.size() < 0){
                throw new RuntimeException("anims 集合为空！无法开始动画！");
            }
            excute();
        }
        //当前 View 执行动画
        private void excute() {
            view = views.get(position);
            animation = anims.get(position);
            animation.setAnimationListener(this);
            view.startAnimation(animation);
            //view.startAnimation(animation);
        }
        public int getPosition(){
            return position;
        }
    }
```

软件缓存清除功能及界面控件监听实现：

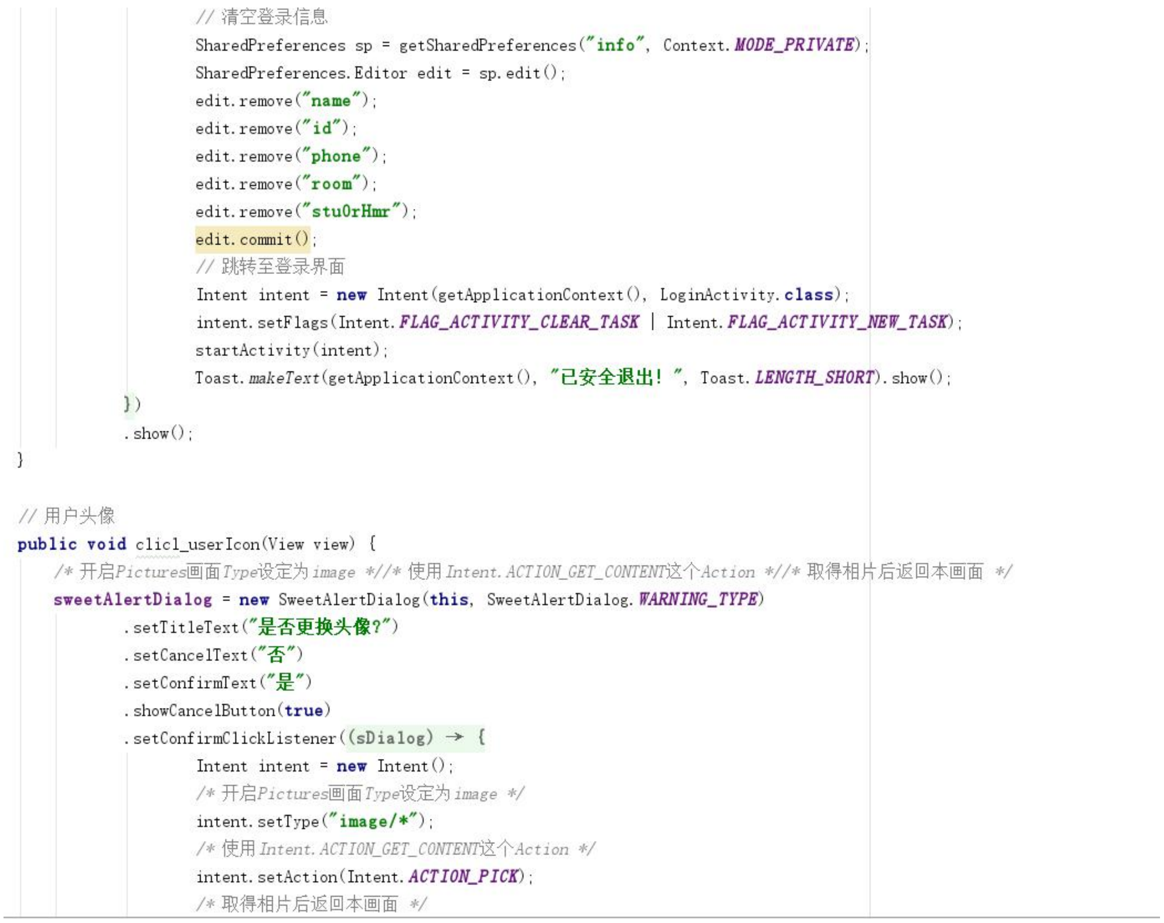


```java
public class SettingActivity extends AppCompatActivity implements AdapterView.OnItemClickListener {

    private ImmersionBar immersionBar;
    private ListView lv_setting;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_setting);
        immersionBar = ImmersionBar.with(this);
        immersionBar.statusBarColor(R.color.colorPrimary); // 状态栏颜色，不写默认透明色
        immersionBar.init();

        // 标题返回
        android.support.v7.app.ActionBar actionBar = getSupportActionBar();
        if(actionBar != null){
            actionBar.setHomeButtonEnabled(true);
            actionBar.setDisplayHomeAsUpEnabled(true);
        }

        // 绑定控件
        lv_setting = (ListView) findViewById(R.id.lv_setting);
        // 设置监听
        lv_setting.setOnItemClickListener(this);
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                this.finish(); // back button
                return true;
        }
```

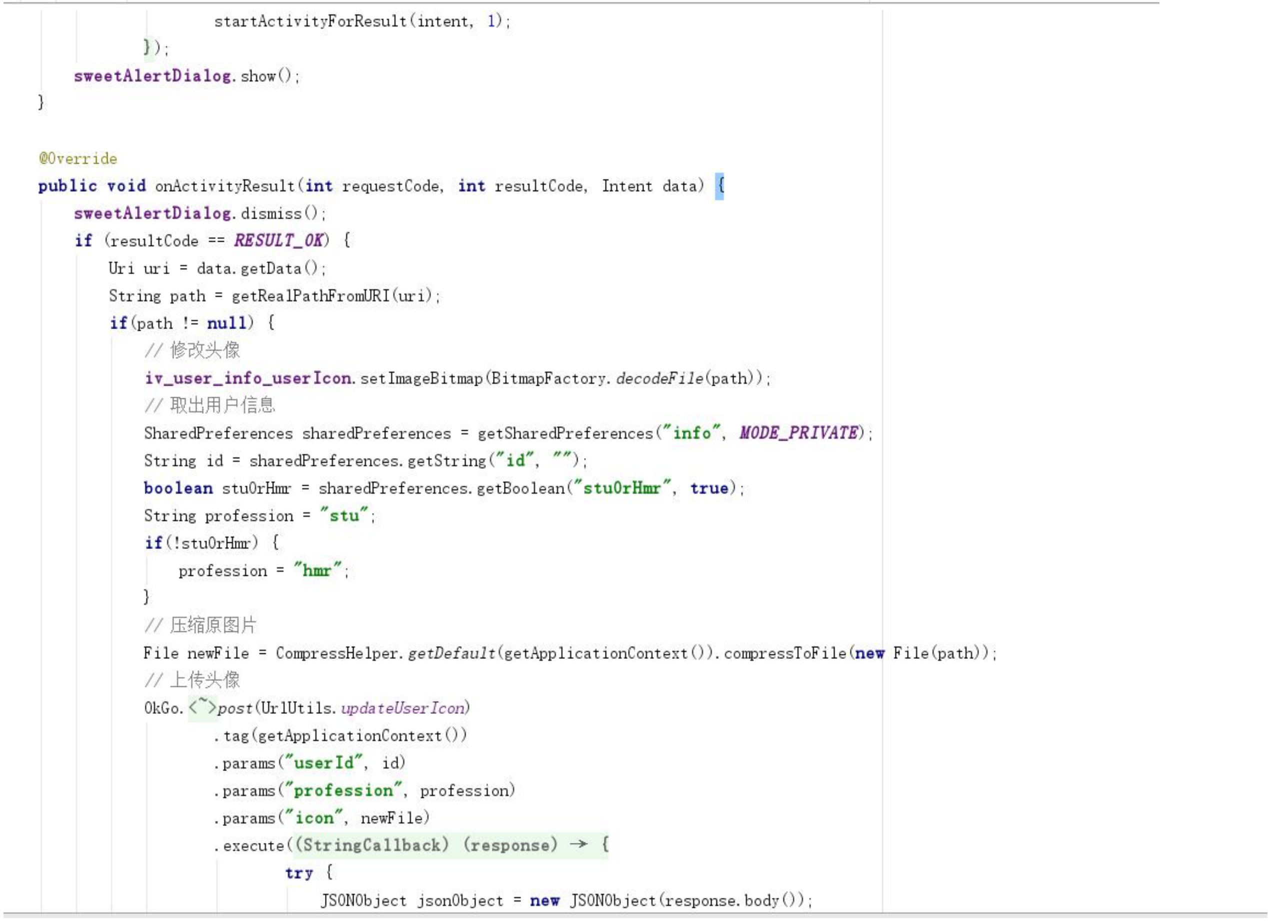


```java
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        if(parent.getItemAtPosition(position).toString().equals("清除缓存")) {
            new SweetAlertDialog(this, SweetAlertDialog.WARNING_TYPE)
                    .setTitleText("是否清除缓存？")
                    .setConfirmText("立即清除")
                    .setConfirmClickListener((sDialog) → {
                        // 获取缓存路径
                        String absolutePath = new ImageDiskLrucache(getApplicationContext()).getDidkCacheDir(getApplicationContext(), "bitmap").getAbsolutePath()
                        // 执行shell命令，清空缓存目录
                        CommandExecutionUtils.execCommand("rm -rf " + absolutePath + "/*", false);
                        // ok
                        sDialog.setTitleText("清除成功！")
                                .setContentText("APP所有图片的缓存清理成功")
                                .setConfirmText("OK")
                                .setConfirmClickListener(null)
                                .changeAlertType(SweetAlertDialog.SUCCESS_TYPE);
                    })
                    .show();
        }
    }
}
```

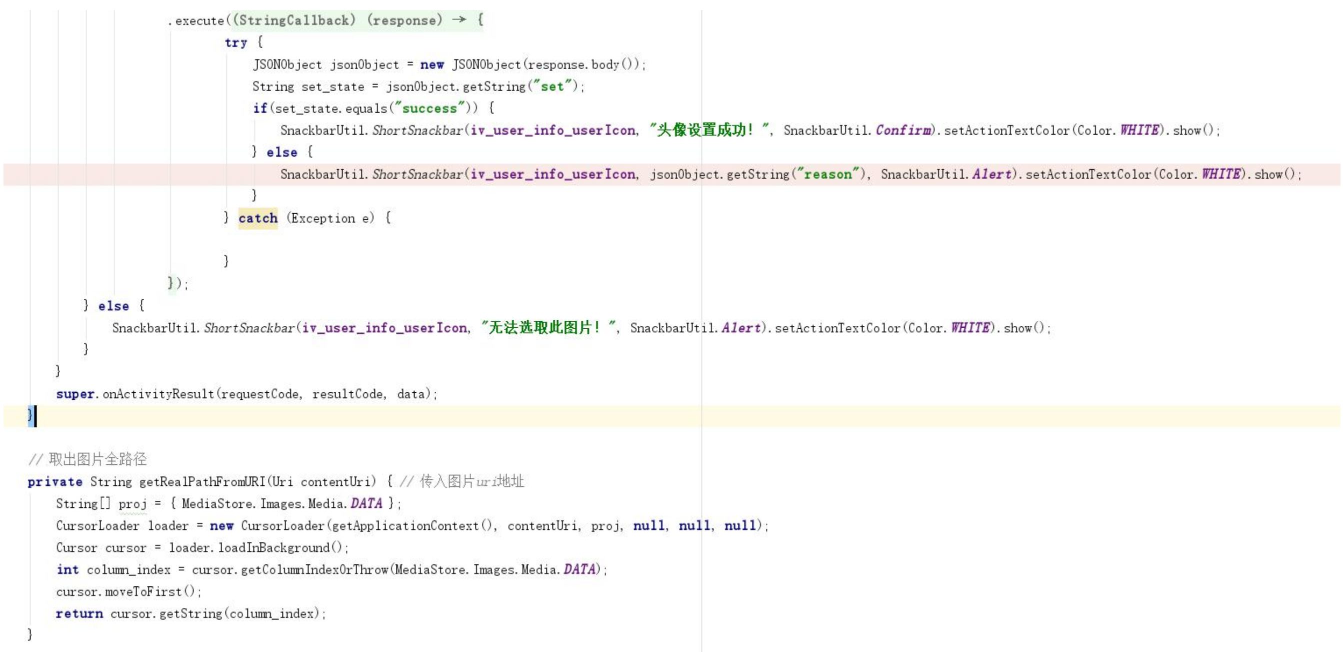
学生个人信息查询界面实现代码（包含个人信息查询以及个人账户头像上传修改）：



```java
                    // 清空登录信息
                    SharedPreferences sp = getSharedPreferences("info", Context.MODE_PRIVATE);
                    SharedPreferences.Editor edit = sp.edit();
                    edit.remove("name");
                    edit.remove("id");
                    edit.remove("phone");
                    edit.remove("room");
                    edit.remove("stuOrHmr");
                    edit.commit();
                    // 跳转至登录界面
                    Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
                    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);
                    startActivity(intent);
                    Toast.makeText(getApplicationContext(), "已安全退出！", Toast.LENGTH_SHORT).show();
                })
                .show();
}

// 用户头像
public void clicl_userIcon(View view) {
    /* 开启Pictures画面Type设定为image *//* 使用Intent.ACTION_GET_CONTENT这个Action *//* 取得相片后返回本画面 */
    sweetAlertDialog = new SweetAlertDialog(this, SweetAlertDialog.WARNING_TYPE)
            .setTitleText("是否更换头像?")
            .setCancelText("否")
            .setConfirmText("是")
            .showCancelButton(true)
            .setConfirmClickListener((sDialog) -> {
                    Intent intent = new Intent();
                    /* 开启Pictures画面Type设定为image */
                    intent.setType("image/*");
                    /* 使用Intent.ACTION_GET_CONTENT这个Action */
                    intent.setAction(Intent.ACTION_PICK);
                    /* 取得相片后返回本画面 */
```



```java
                    startActivityForResult(intent, 1);
            });
    sweetAlertDialog.show();
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    sweetAlertDialog.dismiss();
    if (resultCode == RESULT_OK) {
        Uri uri = data.getData();
        String path = getRealPathFromURI(uri);
        if(path != null) {
            // 修改头像
            iv_user_info_userIcon.setImageBitmap(BitmapFactory.decodeFile(path));
            // 取出用户信息
            SharedPreferences sharedPreferences = getSharedPreferences("info", MODE_PRIVATE);
            String id = sharedPreferences.getString("id", "");
            boolean stuOrHmr = sharedPreferences.getBoolean("stuOrHmr", true);
            String profession = "stu";
            if(!stuOrHmr) {
                profession = "hmr";
            }
            // 压缩原图片
            File newFile = CompressHelper.getDefault(getApplicationContext()).compressToFile(new File(path));
            // 上传头像
            OkGo.<~>post(UrlUtils.updateUserIcon)
                    .tag(getApplicationContext())
                    .params("userId", id)
                    .params("profession", profession)
                    .params("icon", newFile)
                    .execute((StringCallback) (response) -> {
                            try {
                                JSONObject jsonObject = new JSONObject(response.body());
```



```
                .execute((StringCallback) (response) → {
                    try {
                        JSONObject jsonObject = new JSONObject(response.body());
                        String set_state = jsonObject.getString("set");
                        if(set_state.equals("success")) {
                            SnackbarUtil.ShortSnackbar(iv_user_info_userIcon, "头像设置成功! ", SnackbarUtil.Confirm).setActionTextColor(Color.WHITE).show();
                        } else {
                            SnackbarUtil.ShortSnackbar(iv_user_info_userIcon, jsonObject.getString("reason"), SnackbarUtil.Alert).setActionTextColor(Color.WHITE).show();
                        }
                    } catch (Exception e) {

                    }
                });
        } else {
            SnackbarUtil.ShortSnackbar(iv_user_info_userIcon, "无法选取此图片! ", SnackbarUtil.Alert).setActionTextColor(Color.WHITE).show();
        }
    }
    super.onActivityResult(requestCode, resultCode, data);
}

// 取出图片全路径
private String getRealPathFromURI(Uri contentUri) { // 传入图片uri地址
    String[] proj = { MediaStore.Images.Media.DATA };
    CursorLoader loader = new CursorLoader(getApplicationContext(), contentUri, proj, null, null, null);
    Cursor cursor = loader.loadInBackground();
    int column_index = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
    cursor.moveToFirst();
    return cursor.getString(column_index);
}
```

密码重置及更新云端数据库登录信息功能实现代码，利用 SDK submitVerificationCode 服务设置 dialog 加载框监听：

```
public class ForgetPwdActivity extends AppCompatActivity {

    private ImmersionBar immersionBar;
    private EditText et_forget_pwd_phone, et_forget_pwd_phoneCode, et_forget_pwd_password,
et_forget_pwd_password1;
    private boolean isStu = true; // 是否为学生
    private EventHandler eventHandler;
    private Button btn_forget_pwd_sendPhoneCode;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_forget_pwd);
        immersionBar = ImmersionBar.with(this);
        immersionBar.statusBarColor(R.color.colorPrimary); // 状态栏颜色, 不写默认透明色
        immersionBar.init();
        // 标题返回
        android.support.v7.app.ActionBar actionBar = getSupportActionBar();
        if(actionBar != null){
            actionBar.setHomeButtonEnabled(true);
            actionBar.setDisplayHomeAsUpEnabled(true);
        }
        // 绑定控件
        et_forget_pwd_phone = (EditText) findViewById(R.id.et_forget_pwd_phone);
        et_forget_pwd_phoneCode = (EditText) findViewById(R.id.et_forget_pwd_phoneCode);
        et_forget_pwd_password = (EditText) findViewById(R.id.et_forget_pwd_password);
        et_forget_pwd_password1 = (EditText) findViewById(R.id.et_forget_pwd_password1);
        btn_forget_pwd_sendPhoneCode = (Button)
findViewById(R.id.btn_forget_pwd_sendPhoneCode);

        // 学生或宿管单选监听
        ((RadioGroup)
findViewById(R.id.rg_forget_pwd_stuOrHmr)).setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup group, int checkedId) {
                if(checkedId == R.id.rbt_forget_pwd_stu) {
                    isStu = true;
                } else if(checkedId == R.id.rbt_forget_pwd_hmr) {
                    isStu = false;
                }
            }
        });
        // SMS 监听
        // 创建 EventHandler 对象
        eventHandler = new EventHandler() {
            public void afterEvent(int event, int result, Object data) {
                System.out.println("结束！ ---" + event + "---" + result + "---" + data.toString());
                if (data instanceof Throwable) {
                    final Throwable throwable = (Throwable)data;
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            try {
                                JSONObject jsonObject = new
JSONObject(throwable.getMessage());
                                Toast.makeText(getApplicationContext(),
jsonObject.getString("detail"), Toast.LENGTH_SHORT).show();
                            } catch (JSONException e) {
                                e.printStackTrace();
                            }
                        }
                    });
                } else {
                    if (event == SMSSDK.EVENT_GET_VERIFICATION_CODE) {
                        // 获取验证码成功
                        sendBtnWait(); // 设置按钮状态
                    } else if(event == SMSSDK.EVENT_SUBMIT_VERIFICATION_CODE) {
                        // 发送验证码成功
```

```
                                runOnUiThread(new Runnable() {
                                    @Override
                                    public void run() {
                                        updatePwdCheck();
                                    }
                                });
                            }
                        }
                    }
                };
                // 注册监听器
                SMSSDK.registerEventHandler(eventHandler);
            }

            // 发送手机验证码
            public void click_forget_pwd_sendPhoneCode(View view) {
                // 验证手机号码
                String phone = et_forget_pwd_phone.getText().toString().trim();
                if(TextUtils.isEmpty(phone) || phone.length() != 11) {
                    et_forget_pwd_phone.setError("手机号码格式不对！ ");
                    et_forget_pwd_phone.requestFocus();
                    return;
                }
                SMSSDK.getVerificationCode("86", phone); // 发送验证码
            }
            // 重置密码
            public void click_forget_pwd(View view) {
                // 验证输入的参数
                String phone = et_forget_pwd_phone.getText().toString().trim();
                if(TextUtils.isEmpty(phone) || phone.length() != 11) {
                    et_forget_pwd_phone.setError("手机号码格式不对！ ");
                    et_forget_pwd_phone.requestFocus();
                    return;
                }

                String phoneCode = et_forget_pwd_phoneCode.getText().toString().trim();
                if(TextUtils.isEmpty(phoneCode)) {
                    et_forget_pwd_phoneCode.setError("验证码不能为空！ ");
                    et_forget_pwd_phoneCode.requestFocus();
                    return;
                }

                String password = et_forget_pwd_password.getText().toString().trim();
                if(TextUtils.isEmpty(password)) {
```

```
            et_forget_pwd_password.setError("此字段不能为空！");
            et_forget_pwd_password.requestFocus();
            return;
        }

        String password1 = et_forget_pwd_password1.getText().toString().trim();
        if(!password.equals(password1)) {
            et_forget_pwd_password1.setError("两次密码不一致！");
            et_forget_pwd_password1.requestFocus();
            return;
        }

        // 验证验证码是否正确
        SMSSDK.submitVerificationCode("86", phone, phoneCode); // 验证输入的验证码
    }

    // 重置密码验证
    public void updatePwdCheck() {
        String url = UrlUtils.stuUpdatePwd;
        if(!isStu) {
            url = UrlUtils.hmrUpdatePwd;
        }
        String phone = et_forget_pwd_phone.getText().toString().trim();
        String newPwd = et_forget_pwd_password.getText().toString().trim();
        final ProgressDialog dialog = new ProgressDialog(this); // 加载框
        dialog.setTitle("重置中...");
        OkGo.<String>post(url)
                .tag(this)
                .params("phone", phone)
                .params("newPwd", newPwd)
                .execute(new StringCallback() {
                    @Override
                    public void onStart(Request<String, ? extends Request> request) {
                        super.onStart(request);
                        dialog.show();
                    }

                    @Override
                    public void onFinish() {
                        super.onFinish();
                        dialog.dismiss();
                    }

                    @Override
```

```
                            public void onSuccess(Response<String> response) {
                                try {
                                    JSONObject jsonObject = new JSONObject(response.body());
                                    String update_state = jsonObject.getString("update");
                                    if(update_state.equals("error")) {
                                        SnackbarUtil.ShortSnackbar(btn_forget_pwd_sendPhoneCode,
jsonObject.getString("reason"), SnackbarUtil.Alert).setActionTextColor(Color.WHITE).show();
                                    } else {
                                        ts("重置密码成功！");
                                        finish();
                                    }
                                } catch (Exception e) {

                                }
                            }
                        });
    }

    // 按钮倒计时
    private void sendBtnWait() {
        btn_forget_pwd_sendPhoneCode.setClickable(false); // 发送短信并屏蔽按钮
        new Thread(new Runnable() {
            @Override
            public void run() {
                int i = 60; // 设置 60 秒计时
                Message msg = null; // 设置消息
                while (i > 0) {
                    msg = Message.obtain();
                    msg.obj = i + "秒后再次发送";
                    handler.sendMessage(msg);
                    SystemClock.sleep(1000);
                    i--;
                }
                Message message = Message.obtain();
                message.obj = "发送验证码";
                handler.sendMessage(message);
            }
        }).start();
        btn_forget_pwd_sendPhoneCode.setClickable(true); // 发送短信按钮恢复
    }
    private Handler handler = new Handler() {
        @Override
        public void handleMessage(Message msg) {
            btn_forget_pwd_sendPhoneCode.setText((String) msg.obj);
```

```java
        }
    };

    // Toast
    private void ts(String msg) {
        Toast.makeText(getApplicationContext(), msg, Toast.LENGTH_SHORT).show();
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:
                this.finish(); // back button
                return true;
        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        immersionBar.destroy(); //必须调用该方法，防止内存泄漏
        SMSSDK.unregisterEventHandler(eventHandler);
    }
}
```

学生用户新建报修服务更新 NewOrderFragment ：

```java
public class NewOrderFragment extends Fragment {
    private ZzImageBox imageBox;
    private Button btn_neworder_submit;
    private EditText et_neworder_orderContent;
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
        final View view = inflater.inflate(R.layout.fragment_neworder, null);
        // 添加图片
        imageBox = (ZzImageBox) view.findViewById(R.id.zz_image_box);
        imageBox.setOnImageClickListener(new ZzImageBox.OnImageClickListener() {
            @Override
            public void onImageClick(int position, String filePath, ImageView iv) {
                //Log.d("ZzImageBox", "image clicked:" + position + "," + filePath);
                Intent intent = new Intent(getContext(), LookImageActivity.class);
                intent.putExtra("imgPath", filePath);
                startActivity(intent);
```

```
                    }
                    @Override
                    public void onDeleteClick(int position, String filePath) {
                        imageBox.removeImage(position);
                        //Log.d("ZzImageBox", "delete clicked:" + position + "," + filePath);
                        //Log.d("ZzImageBox", "all images\n"+ imageBox.getAllImages().toString());
                    }

                    @Override
                    public void onAddClick() {
                        Intent intent = new Intent();
                    /* 开启 Pictures 画面 Type 设定为 image */
                        intent.setType("image/*");
                    /* 使用 Intent.ACTION_GET_CONTENT 这个 Action */
                        intent.setAction(Intent.ACTION_PICK);
                    /* 取得相片后返回本画面 */
                        startActivityForResult(intent, 1);
                        //imageBox.addImage(null);
                        //Log.d("ZzImageBox", "add clicked");
                        //Log.d("ZzImageBox", "all images\n"+ imageBox.getAllImages().toString());
                    }
                });
                // 提交报修单
                btn_neworder_submit = (Button) view.findViewById(R.id.btn_neworder_submit);
                btn_neworder_submit.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        // 取到报修内容，并验证
                        et_neworder_orderContent = (EditText)
view.findViewById(R.id.et_neworder_orderContent);
                        String orderContent = et_neworder_orderContent.getText().toString().trim();
                        if(TextUtils.isEmpty(orderContent)) {
                            et_neworder_orderContent.setError("此字段不能为空！");
                            et_neworder_orderContent.requestFocus();
                            return;
                        }
                        // 取到提交的用户 ID 和所在的楼号寝室
                        SharedPreferences sp = getContext().getSharedPreferences("info",
getContext().MODE_PRIVATE);
                        boolean stuOrHmr = sp.getBoolean("stuOrHmr", false);
                        final String id = sp.getString("id", null);
                        final String orderRoom = sp.getString("room", null);
                        // OkGo
                        PostRequest<String> tag =
OkGo.<String>post(UrlUtils.addOrder).tag(getContext());
                        // 设置请求参数
                        if(stuOrHmr) {
                            tag.params("stuId", id);
                        } else {
                            tag.params("hmrId", id);
                        }
                        tag.params("orderInfo", orderContent);
                        tag.params("orderRoom", orderRoom);
                        // 网络请求
                        final ProgressDialog dialog = new ProgressDialog(getContext()); // 加载框
                        dialog.setTitle("提交中...");
                        tag.execute(new StringCallback() {
                            @Override
                            public void onStart(Request<String, ? extends Request> request) {
                                super.onStart(request);
                                dialog.show();
                            }
                            @Override
                            public void onFinish() {
                                super.onFinish();
                                dialog.dismiss();
                            }
                            @Override
                            public void onSuccess(Response<String> response) {
                                try {
                                    JSONObject jsonObject = new JSONObject(response.body());
                                    String add_state = jsonObject.getString("add");
                                    if (add_state.equals("error")) {
                                        SnackbarUtil.ShortSnackbar(btn_neworder_submit,
jsonObject.getString("reason"), SnackbarUtil.Alert).setActionTextColor(Color.WHITE).show();
                                    } else {
                                        SnackbarUtil.ShortSnackbar(btn_neworder_submit, "订单已提
交！ ", SnackbarUtil.Confirm).setActionTextColor(Color.WHITE).show();
                                        // 清空历史报修内容
                                        et_neworder_orderContent.setText(null);
                                        // 上传图片
                                        uploadImg(jsonObject.getInt("orderId"));
                                    }
                                } catch (Exception e) {

                                }
                            }
                        });
```

```
            }
        });

        return view;
    }
```
管理员功能选择界面布局代码：
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app=""
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/timg">

    <TextView
        android:id="@+id/date"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#666666"
        android:textSize="12sp"
        android:layout_gravity="center"
        android:gravity="bottom|right"
        android:text="2015-12-10 10:23"
        />

    <Button
        android:text="请选择功能，对应跳转"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button4"
        style="@style/Widget.AppCompat.Button"
        android:background="@color/transparent"
        android:visibility="visible"
        android:textColor="@drawable/background_standard_black"
        android:elevation="0dp"
        />

    <Button
        android:text="学生信息查询"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button10"
        android:background="@drawable/background_kitkat_blue" />
```

```
<Button
    android:text=" "
    android:layout_width="match_parent"
    android:layout_height="31dp"
    android:id="@+id/button11"
    android:background="@color/transparent"/>

<Button
    android:text="住宿管理"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/button9"
    android:background="@drawable/background_kitkat_blue" />

<Button
    android:text=" "
    android:layout_width="match_parent"
    android:layout_height="35dp"
    android:id="@+id/button13"
    android:background="@color/transparent" />

<Button
    android:text="离返校情况"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/button8"
    android:background="@drawable/background_kitkat_blue" />

<Button
    android:text=" "
    android:layout_width="match_parent"
    android:layout_height="33dp"
    android:id="@+id/button12"
    android:background="@color/transparent" />

<Button
    android:text="卫生评比"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/button7"
    android:background="@drawable/background_kitkat_blue" />

<Button
```

```
        android:text=" "
        android:layout_width="match_parent"
        android:layout_height="33dp"
        android:id="@+id/button14"
        android:background="@color/transparent" />

    <Button
        android:text="报修管理"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button6"
        android:background="@drawable/background_kitkat_blue" />

    <Button
        android:text=" "
        android:layout_width="match_parent"
        android:layout_height="33dp"
        android:id="@+id/button15"
        android:background="@color/transparent" />

    <Button
        android:text="楼层公告"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/button5"
        android:background="@drawable/background_kitkat_blue" />

</LinearLayout>
```

学生用户登录成功后跳转界面即公告查询界面界面布局文件：

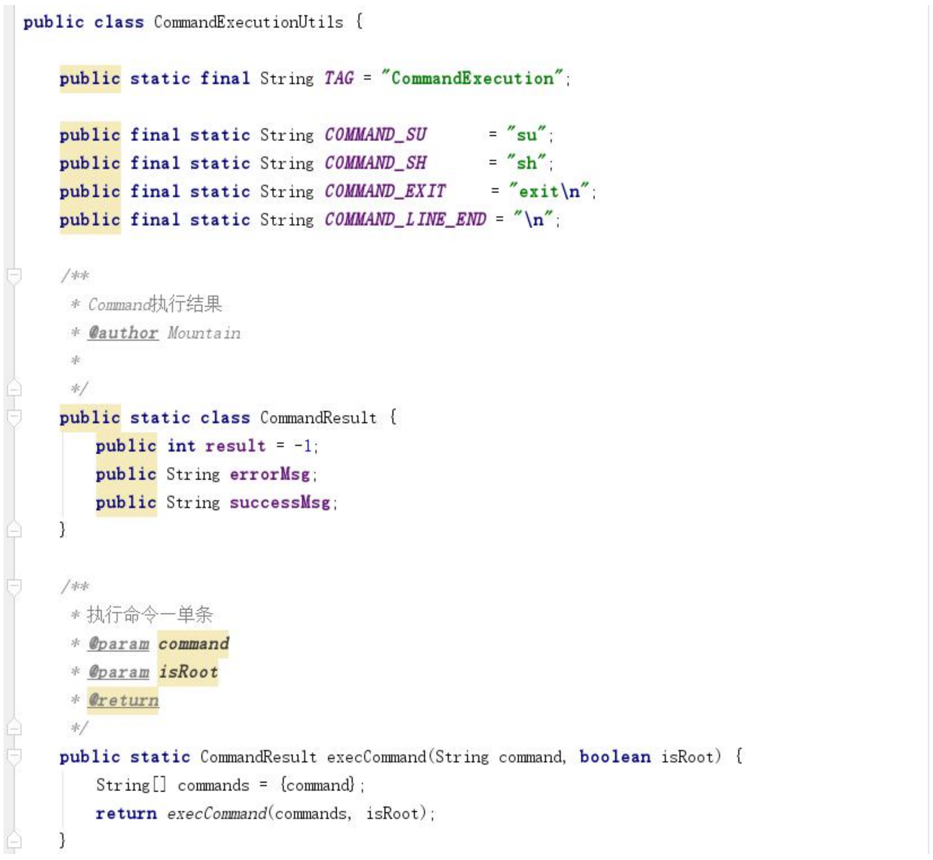
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/name">

    <ScrollView
        android:id="@+id/scroll_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        >

        <com.nightonke.saver.ui.MyGridView
        android:id="@+id/grid_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:numColumns="4"
        android:verticalSpacing="10dp"
        android:horizontalSpacing="5dp"
        android:columnWidth="60dp"
        android:paddingTop="20dp"
        android:paddingBottom="20dp"
        android:stretchMode="columnWidth"
        />

</ScrollView>

<Button
    android:text="楼层公告"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/button16"
    android:background="@color/transparent"
    android:textColor="@color/bg_swipe_item_pinned"
    android:layout_below="@+id/scroll_view"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<Button
    android:text="发布时间"
    android:layout_width="198dp"
    android:layout_height="wrap_content"
    android:id="@+id/button17"
    android:layout_below="@+id/button16"
    android:background="@color/transparent"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:textColor="@color/baby_header" />

<Button
    android:text="内容"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/button16"
    android:layout_toRightOf="@+id/button17"
    android:background="@color/transparent"
    android:id="@+id/button19"
    android:layout_alignBottom="@+id/button17"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:textColor="@color/baby_header" />

<Button
    android:text="三楼由于管道施工从 7 月 15 日早晨 8:00 开始停水........"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button24"
    android:background="@color/transparent"
    android:layout_below="@+id/button19"
    android:layout_toEndOf="@+id/button17"
    android:textColor="@color/baby_header"
    android:layout_toRightOf="@+id/button17" />

<Button
    android:text="详情"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button22"
    android:textColor="@color/baby_header"
    android:background="@drawable/bg_item_dragging_state"
    android:layout_below="@+id/button23"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

<Button
    android:text="2018 年 7 月 16 日"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/transparent"
    android:id="@+id/button21"
    android:textColor="@color/baby_header"
    android:layout_below="@+id/button22"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_toLeftOf="@+id/button20"
    android:layout_toStartOf="@+id/button20" />

<Button
```

```
    android:text="2018 年 7 月 13 日"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/transparent"
    android:id="@+id/button23"
    android:textColor="@color/baby_header"
    android:layout_below="@+id/button17"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignBottom="@+id/button24"
    android:layout_toLeftOf="@+id/button19"
    android:layout_toStartOf="@+id/button19" />

<Button
    android:text="发布时间"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/transparent"
    android:id="@+id/button18"
    android:layout_alignBottom="@+id/button21"
    android:layout_toRightOf="@+id/button21"
    android:layout_toEndOf="@+id/button21" />

<Button
    android:text="近日台风来袭请同学们尽量减少外出....."
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button20"
    android:background="@color/transparent"
    android:textColor="@color/baby_header"
    android:layout_below="@+id/button22"
    android:layout_toRightOf="@+id/button23"
    android:layout_alignBottom="@+id/button21"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />

<Button
    android:text="详情"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button25"
    android:textColor="@color/baby_header"
    android:background="@drawable/bg_item_dragging_state"
    android:layout_below="@+id/button21"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

<Button
    android:text="2018 年 8 月 3 日"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button28"
    android:background="@color/transparent"
    android:layout_below="@+id/button25"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_toStartOf="@+id/button27"
    android:layout_alignBottom="@+id/button27"
    android:layout_toLeftOf="@+id/button27"
    android:textColor="@color/baby_header" />

<Button
    android:text="近期有同学遗失贵重财物，再....."
    android:background="@color/transparent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button27"
    android:layout_below="@+id/button25"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_toRightOf="@+id/button21"
    android:layout_toEndOf="@+id/button21"
    android:textColor="@color/baby_header" />

<Button
    android:text="详情"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button30"
    android:layout_below="@+id/button28"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:background="@drawable/bg_item_dragging_state"
    android:textColor="@color/baby_header" />
```

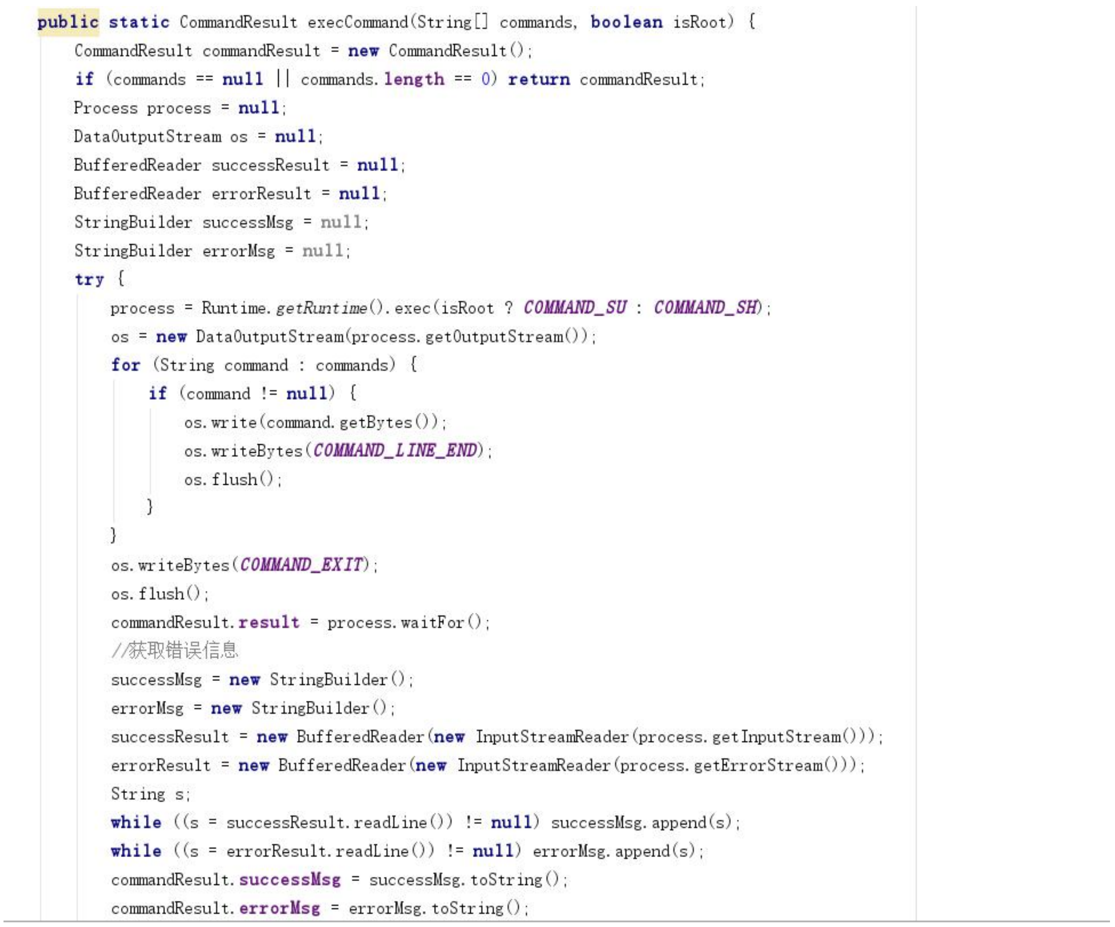
```
</RelativeLayout>
```
Snackbar 控件监听函数（在创建了一个 Snackbar 对象后，通过调用 set**方法进行设置，其中 setAction()方法用于设置右侧的文字显示以及点击事件，setCallback()方法用于设置一个状态回调，在 Snackbar 显示和消失的时候会触发方法，使用静态的 make 方法，并且其中的 view 参数是一个查找父布局的起点）：

```
public class SnackbarUtil {

    public static final    int Info = 1;
    public static final   int Confirm = 2;
    public static final   int Warning = 3;
    public static final   int Alert = 4;
    public static   int red = 0xfff44336;
    public static   int green = 0xff4caf50;
    public static   int blue = 0xff2195f3;
    public static   int orange = 0xffffc107;
    /**
     * 短显示 Snackbar，自定义颜色
     * @param view
     * @param message
     * @param messageColor
     * @param backgroundColor
     * @return
     */
    public static Snackbar ShortSnackbar(View view, String message, int messageColor, int
backgroundColor){
        Snackbar snackbar = Snackbar.make(view,message, Snackbar.LENGTH_SHORT);
        setSnackbarColor(snackbar,messageColor,backgroundColor);
        return snackbar;
    }
    /**
     * 长显示 Snackbar，自定义颜色
     * @param view
     * @param message
     * @param messageColor
     * @param backgroundColor
     * @return
     */
    public static Snackbar LongSnackbar(View view, String message, int messageColor, int
backgroundColor){
        Snackbar snackbar = Snackbar.make(view,message, Snackbar.LENGTH_LONG);
        setSnackbarColor(snackbar,messageColor,backgroundColor);
        return snackbar;
    }
    /**
     * 自定义时常显示 Snackbar， 自定义颜色
     * @param view
     * @param message
     * @param messageColor
     * @param backgroundColor
     * @return
     */
    public static Snackbar IndefiniteSnackbar(View view, String message,int duration,int 
messageColor, int backgroundColor){
        Snackbar snackbar = Snackbar.make(view,message, 
Snackbar.LENGTH_INDEFINITE).setDuration(duration);
        setSnackbarColor(snackbar,messageColor,backgroundColor);
        return snackbar;
    }

    /**
     * 短显示 Snackbar， 可选预设类型
     * @param view
     * @param message
     * @param type
     * @return
     */
    public static Snackbar ShortSnackbar(View view, String message, int type){
        Snackbar snackbar = Snackbar.make(view,message, Snackbar.LENGTH_SHORT);
        switchType(snackbar,type);
        return snackbar;
    }

    /**
     * 长显示 Snackbar， 可选预设类型
     * @param view
     * @param message
     * @param type
     * @return
     */
    public static Snackbar LongSnackbar(View view, String message,int type){
        Snackbar snackbar = Snackbar.make(view,message, Snackbar.LENGTH_LONG);
        switchType(snackbar,type);
        return snackbar;
    }

    /**
     * 自定义时常显示 Snackbar， 可选预设类型
     * @param view
     * @param message
     * @param type
     * @return
     */
    public static Snackbar IndefiniteSnackbar(View view, String message,int duration,int type){
        Snackbar snackbar = Snackbar.make(view,message,
Snackbar.LENGTH_INDEFINITE).setDuration(duration);
        switchType(snackbar,type);
        return snackbar;
    }

    //选择预设类型
    private static void switchType(Snackbar snackbar,int type){
        switch (type){
            case Info:
                setSnackbarColor(snackbar,blue);
                break;
            case Confirm:
                setSnackbarColor(snackbar,green);
                break;
            case Warning:
                setSnackbarColor(snackbar,orange);
                break;
            case Alert:
                setSnackbarColor(snackbar,red);
                break;
        }
    }

    /**
     * 设置 Snackbar 背景颜色
     * @param snackbar
     * @param backgroundColor
     */
    public static void setSnackbarColor(Snackbar snackbar, int backgroundColor) {
        View view = snackbar.getView();
        if(view!=null){
            view.setBackgroundColor(backgroundColor);
        }
    }

    /**
     * 设置 Snackbar 文字和背景颜色
     * @param snackbar
     * @param messageColor
     * @param backgroundColor
     */
    public static void setSnackbarColor(Snackbar snackbar, int messageColor, int backgroundColor)
{
        View view = snackbar.getView();
        if(view!=null){
            view.setBackgroundColor(backgroundColor);
            ((TextView) view.findViewById(R.id.snackbar_text)).setTextColor(messageColor);
        }
    }

    /**
     * 向 Snackbar 中添加 view
     * @param snackbar
     * @param layoutId
     * @param index 新加布局在 Snackbar 中的位置
     */
    public static void SnackbarAddView( Snackbar snackbar,int layoutId,int index) {
        View snackbarview = snackbar.getView();
        Snackbar.SnackbarLayout snackbarLayout=(Snackbar.SnackbarLayout)snackbarview;

        View add_view = LayoutInflater.from(snackbarview.getContext()).inflate(layoutId,null);

        LinearLayout.LayoutParams p = new
LinearLayout.LayoutParams( LinearLayout.LayoutParams.WRAP_CONTENT,LinearLayout.LayoutParam
s.WRAP_CONTENT);
        p.gravity= Gravity.CENTER_VERTICAL;
        snackbarLayout.addView(add_view,index,p);
    }
}
```

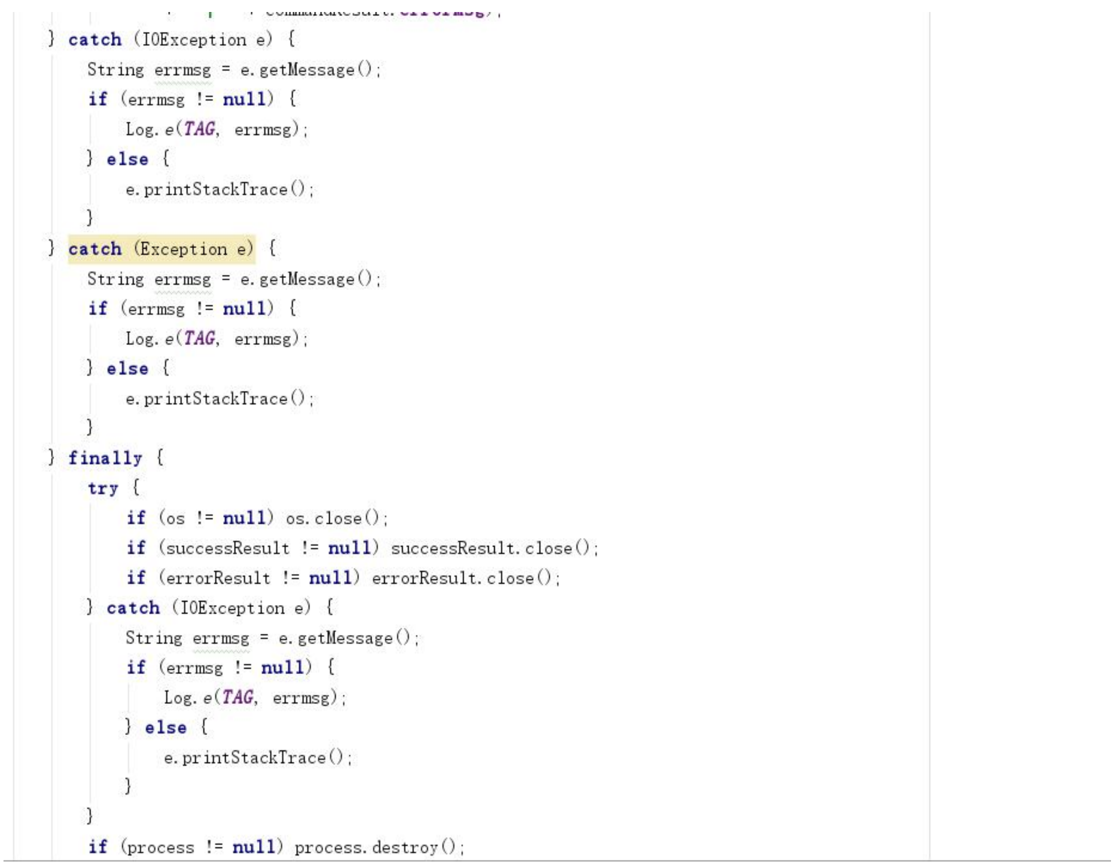
执行 shell 脚本工具类（Runtime.getRuntime().exec() 方法执行 shell 的命令或 脚本）



```java
public class CommandExecutionUtils {

    public static final String TAG = "CommandExecution";

    public final static String COMMAND_SU       = "su";
    public final static String COMMAND_SH       = "sh";
    public final static String COMMAND_EXIT     = "exit\n";
    public final static String COMMAND_LINE_END = "\n";

    /**
     * Command执行结果
     * @author Mountain
     *
     */
    public static class CommandResult {
        public int result = -1;
        public String errorMsg;
        public String successMsg;
    }

    /**
     * 执行命令—单条
     * @param command
     * @param isRoot
     * @return
     */
    public static CommandResult execCommand(String command, boolean isRoot) {
        String[] commands = {command};
        return execCommand(commands, isRoot);
    }
```



```java
public static CommandResult execCommand(String[] commands, boolean isRoot) {
    CommandResult commandResult = new CommandResult();
    if (commands == null || commands.length == 0) return commandResult;
    Process process = null;
    DataOutputStream os = null;
    BufferedReader successResult = null;
    BufferedReader errorResult = null;
    StringBuilder successMsg = null;
    StringBuilder errorMsg = null;
    try {
        process = Runtime.getRuntime().exec(isRoot ? COMMAND_SU : COMMAND_SH);
        os = new DataOutputStream(process.getOutputStream());
        for (String command : commands) {
            if (command != null) {
                os.write(command.getBytes());
                os.writeBytes(COMMAND_LINE_END);
                os.flush();
            }
        }
        os.writeBytes(COMMAND_EXIT);
        os.flush();
        commandResult.result = process.waitFor();
        //获取错误信息
        successMsg = new StringBuilder();
        errorMsg = new StringBuilder();
        successResult = new BufferedReader(new InputStreamReader(process.getInputStream()));
        errorResult = new BufferedReader(new InputStreamReader(process.getErrorStream()));
        String s;
        while ((s = successResult.readLine()) != null) successMsg.append(s);
        while ((s = errorResult.readLine()) != null) errorMsg.append(s);
        commandResult.successMsg = successMsg.toString();
        commandResult.errorMsg = errorMsg.toString();
```



```java
                + commandResult.errorMsg);
    } catch (IOException e) {
        String errmsg = e.getMessage();
        if (errmsg != null) {
            Log.e(TAG, errmsg);
        } else {
            e.printStackTrace();
        }
    } catch (Exception e) {
        String errmsg = e.getMessage();
        if (errmsg != null) {
            Log.e(TAG, errmsg);
        } else {
            e.printStackTrace();
        }
    } finally {
        try {
            if (os != null) os.close();
            if (successResult != null) successResult.close();
            if (errorResult != null) errorResult.close();
        } catch (IOException e) {
            String errmsg = e.getMessage();
            if (errmsg != null) {
                Log.e(TAG, errmsg);
            } else {
                e.printStackTrace();
            }
        }
        if (process != null) process.destroy();
```