**UI设计思路**

本文档将介绍《电气设备维护app》的UI界面设计思路，将逐个逐个界面地进行讲叙。总体配色采用的是蓝绿色调，色调清新。控件摆放采用传统app的方式，使用方法简单明了，插图选取的是电力相关方面的插图，填充空白区域，丰富界面的同时没有违和感。

**1. 初始界面（MainActivity）**



图1.1

界面如图1.1所示，主要分为三个部分，从上至下分别为App的logo图，用户的身份选择以及版本信息。首先，App的logo图是一双手守护着一个圆形的电力标志，寓意着我们此App是从电力设备的维护出发的，符合我们的app用户。中间的用户身份选择，可以快速帮助用户确定自己是普通用户亦或是维修公司的员工，进入到不同的界面，便于用户快速熟悉app的使用（由于功能尚未开发完成，本此主要介绍工人的功能界面）。底部的版本信息可以清晰看到版本更新，是软件的常见信息。此外，如果删除掉底部的版本信息，整个界面就会聚集在上部分，底部显得十分空荡，不够美观。



图1.2

为了更方便用户清楚是否已经点击按钮，增加了按钮的点击效果（ripple效果），用户在点击按钮时，会生成白色波澜，如图1.2所示。

**2. 登录界面（WorkerLoginActivity）**



图2.1

此登录界面为维修工人登陆界面（普通用户登录界面与之类似，不做展示），本界面从上至下仍是分为三个部分，分别是logo图，登录信息表，以及附属功能。此界面的Logo图仍是选择了电力设备相关的图片，加深用户对于本app功能的印象。登录信息表顶部是一个textview，文本为ForMaintenance，意思为为维修工人，作为维修工人登录界面的标识之一，随后便是常间的edittext，使用户得以输入信息。用户名使用明文输入，密码输入栏使用密文。再往下的便是为了满足用户忘记密码以及注册新账号的需求。



图2.2

用户输入信息之后的界面如图2.2所示，但是我们不能保证每个用户的输入都是合法且正确的，因此我们需要设计信息提示来告诉用户他们的输入是否合法。

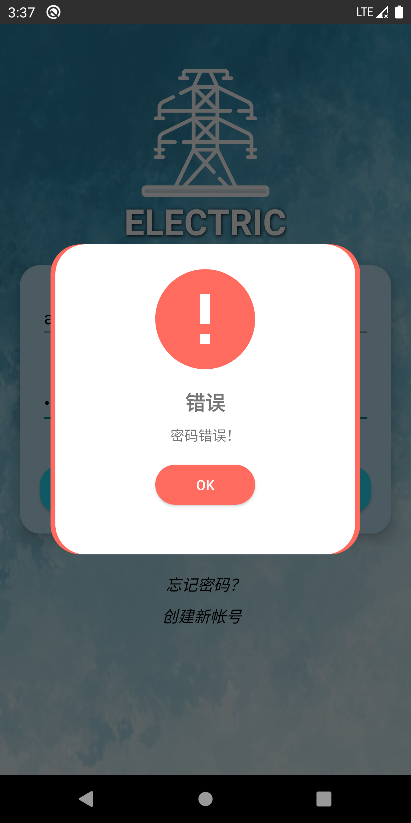


图2.3 密码错误提示

当用户输入的密码不正确时，我们需要跳出一个AlertDialog来提示用户，让其修改密码。错误提示使用惯例的红色，符合大众的习惯，配上错误信息的文字表达，方便用户快速定位错误进行修改。



图2.4 密码输入为空

对于空的输入，我们也是不能接受的，因此需要对此进行处理。为了更直观和更方便的处理这种情况，我们使用了Android Studio自带的错误提示信息。

if(userpwd.isNullOrEmpty()) userpwd.error = “密码不能为空”

接下来所有类似的判空操作都是采用以上操作，就不做展示了。



图2.5

当我们点击登录后，有可能由于网络延迟的原因，验证用户的输入是否合法并且已经存在于我们的数据库中，因此我们设计了一个登录中的加载页面。为三个蓝灰色的圆点，不断循环，直到验证完毕跳出下一步的信息提示，优化了用户的等待体验。

当用户忘记的密码的时候，我们就需要点击“忘记密码”修改噜，具体界面请看图2.6，如下所示：



图2.6 点击文本

但是我们如果对“忘记密码”不做任何操作的话，我们的点击操作就会十分枯燥，如果处理器卡顿的时候，用户甚至会怀疑是否已经点击成功。因此我们添加了点击操作，当用户点击对应的功能文本时，我们会把文本颜色进行修改，然后再进入另外一个界面，十分人性化。

“创建新帐号”也是类似的变色操作，都是十分的人性化，提升用户的使用体验，也是一个良好的UI界面不可或缺的因素。

上一步，我们已经点了“忘记密码”的功能文本，所以接下来就让我们进入到修改密码的界面，看看有什么新的元素吧。

**3. 修改密码界面（WorkerChangePwdActivity）**



图3.1

在重置密码界面，我们还是从上至下三个部分，但是元素已经变换了。最顶上的部分，我们在这里加了一个状态栏，方便用户因为误触进入而返回到登录界面。中间部分的文字说明也提示用户此界面的功能是什么，如果不是用户想要的，就可以点击状态栏的箭头返回到登录界面。底下部分的cardview，就是让用户填入各种信息，方便我们在数据库中进行查询与修改操作。

这里我们设计了一个新的功能按钮，那就是获取验证码的按钮。众所周知，如果用户可以快速且无限地对我们的服务器进行请求，我们的服务器压力会十分大，所以我们的“获取验证码”按钮需要在点击后进行变化，如图3.2所示。



图3.2

在倒计时结束后，我们也应该标记用户已经获取过一次验证码，因此文本不能与初始时一致，令文本变为“重新获取”，如图3.3所示。



图3.3

此外，由于验证码的功能我们还没做完，这次演示只能跳过这个获取验证码的环节了。下面来看看对于用户输入的提醒处理。

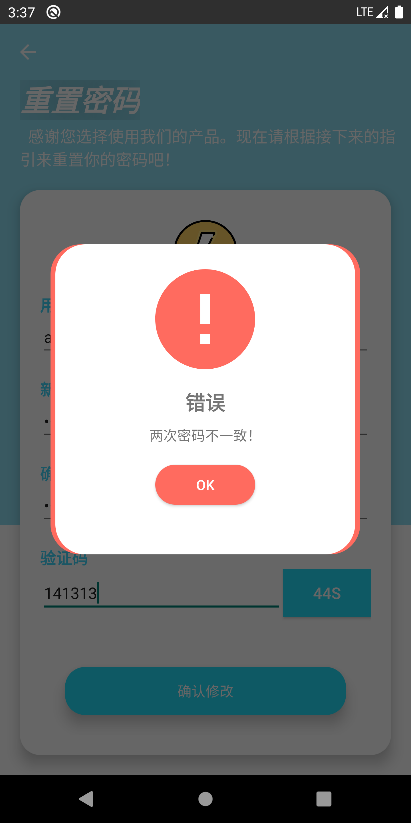
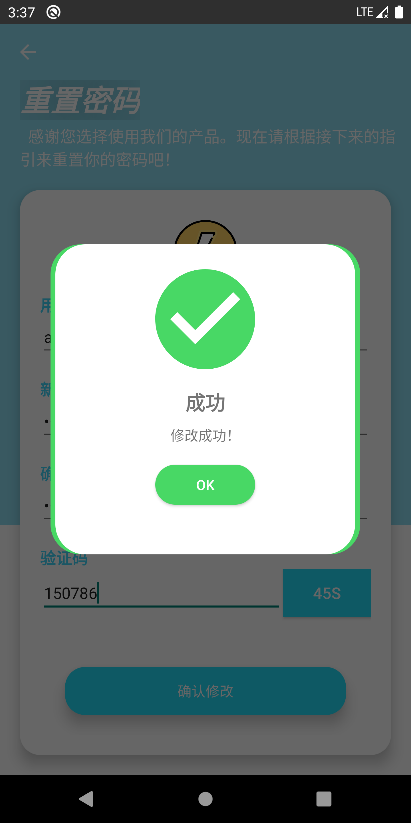


图3.4

对于两次密码不一致，采用错误消息提醒框，还是惯例的红色，但是错误消息文本进行了修改，方便用户更改。



**图3.5**

当用户的输入都合理并且可以被程序接受处理后，提醒成功修改，然后自动跳转到登录界面。成功提示AlertDialog采用惯例的绿色，符合大众的审美心理。

**4. 注册界面（WorkerResigisterActivity）**



图4.1

注册界面在重用修改密码界面的xml代码上做出了简单的修改，主要是功能介绍文本的修改，以及需要信息的修改，故所有的用户输入处理样式均是重用的，此处就不再做展示了。

接下来就让我们进入到维修工人的界面吧！

**5. 工人主界面（WorkMainAcitivity）**



图5.1

我们一眼看过去就是主要的三个功能，二维码扫描，FUNCTION（工人可操作功能），USER（个人信息界面）。总体配色采用的是蓝绿色调，色调清新。控件摆放采用传统app的方式，使用方法简单明了，插图选取的是电力相关方面的插图，填充空白区域，丰富界面的同时没有违和感。总体布局采用的是相对布局，方便插画的布置，布局层次的结构如图5.2：

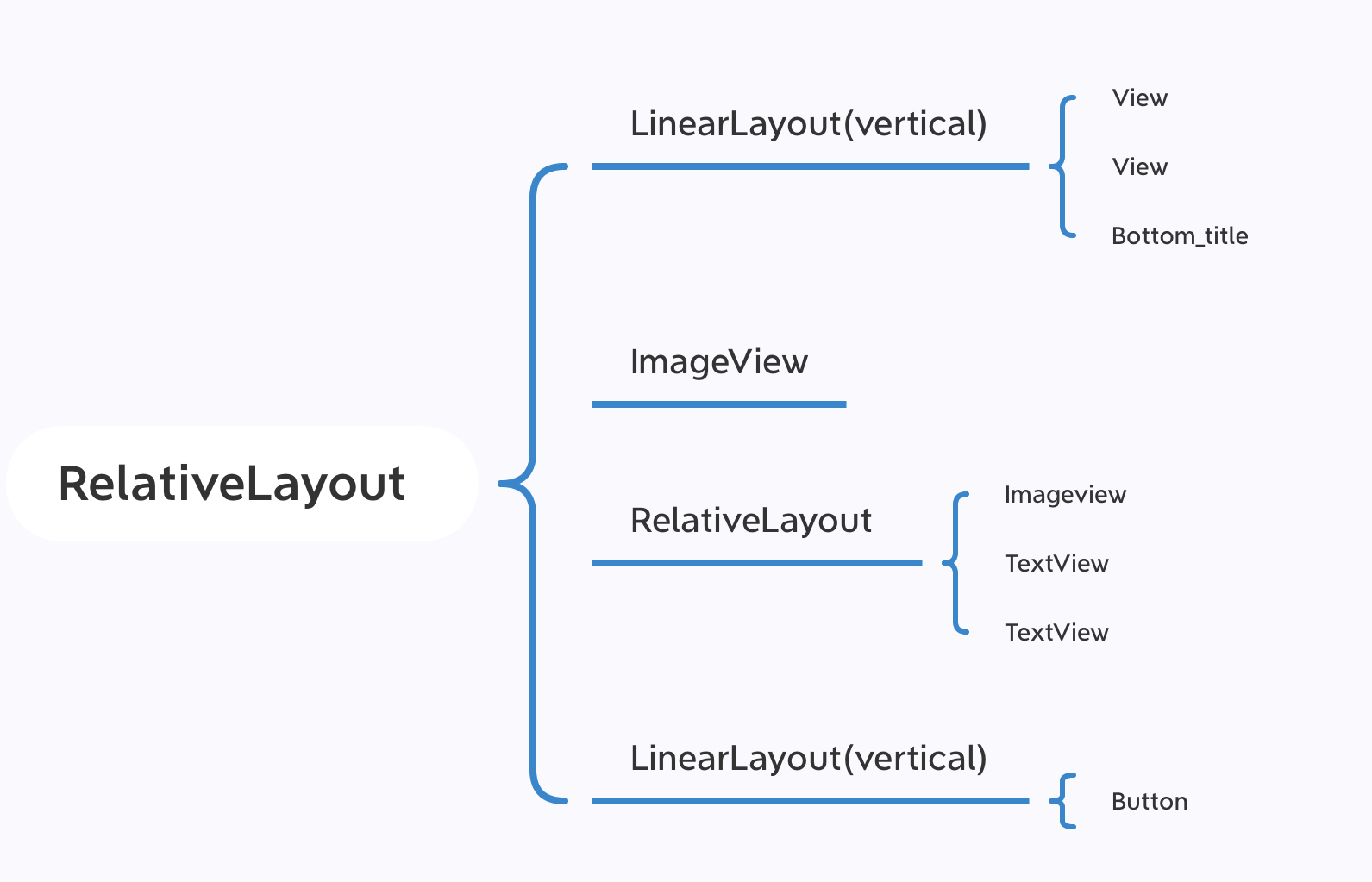


图5.2

第一部分的linealayout是背景部分，包括了蓝白拼接背景以及底部菜单，其中底部菜单为自定义封装的组件。结构为图5.3所示：

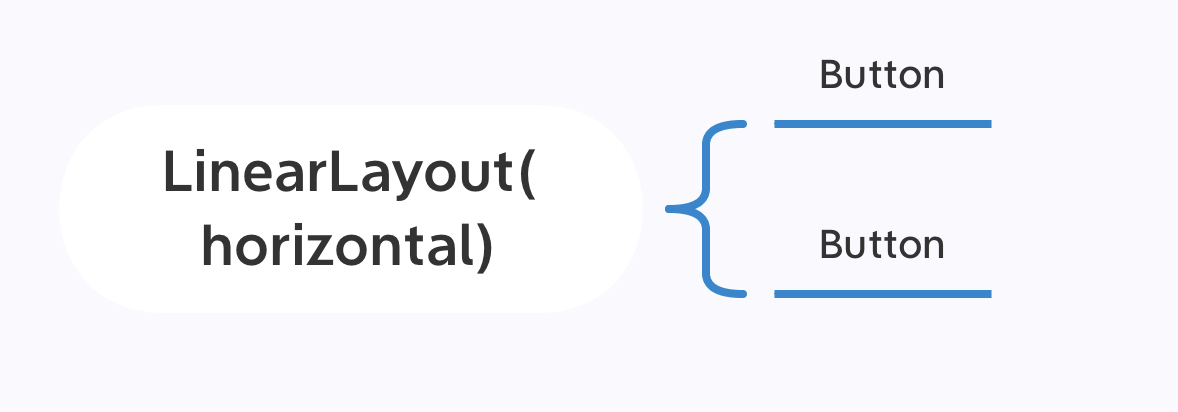


图5.3

第二部分是插画部分，第三部分的RalativeLayout是顶部标题部分，包括了一个图标，以及两个标题。

**6. 巡检功能界面（CheckActivity）**



图6.1

总体配色与主页面相匹配，用于填报检查项目，总体布局采用相对布局。总体分为三部分。第一部分包括了工具栏的返回按钮，图标，标题文字；第二部分采用了scroView，包括了若干个水平线性布局，便于填写且添加新内容，当内容过多时也不会超出屏幕范围，反而是可以滚动显示所需要的信息；第三部分为提交按钮，点击会出现AlertDialog，如果信息填写不完整会弹出AlertDialog作相对应的提醒，若填写完整供用户确认提交行为，确认无误后提交。如图6.2，6.3所示。



图6.3

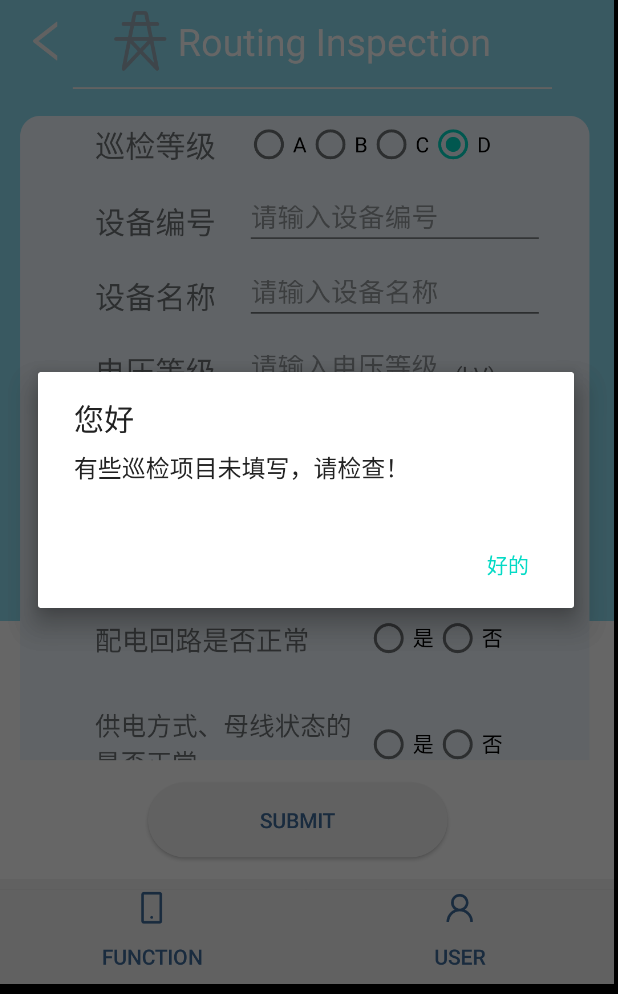


图6.4

**7. 上报故障界面、内容查看界面（My\_Upload,InformationActivity）**

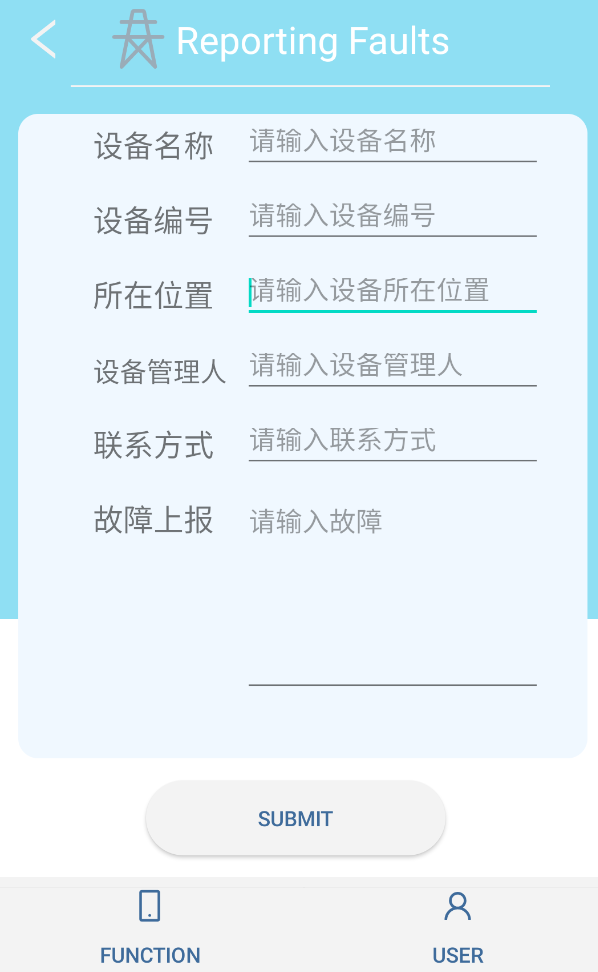
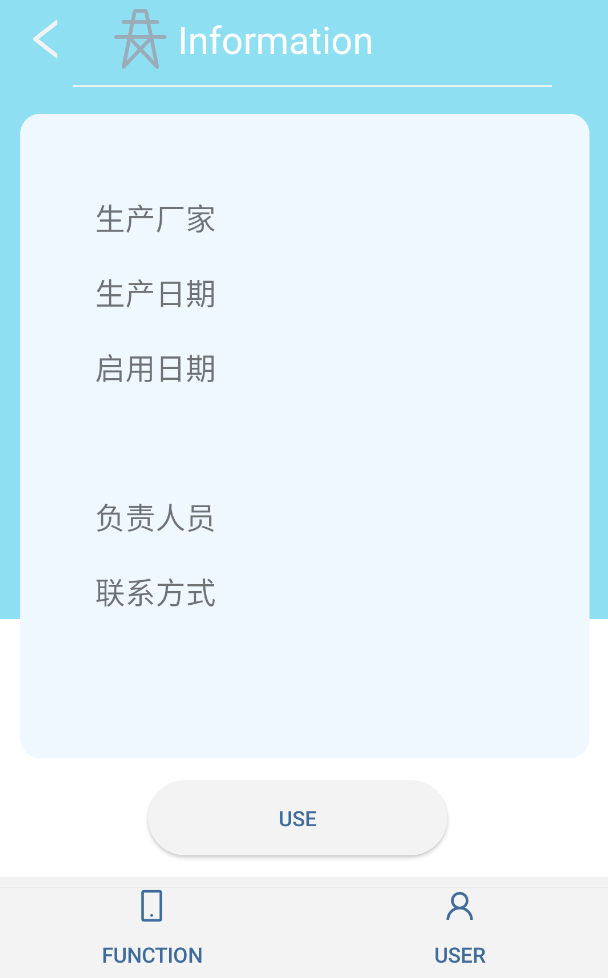
 

图7.1 图7.2

此部分与巡检部分为代码重用部分，不作过多展示。

8. 用户界面（UserActivity）

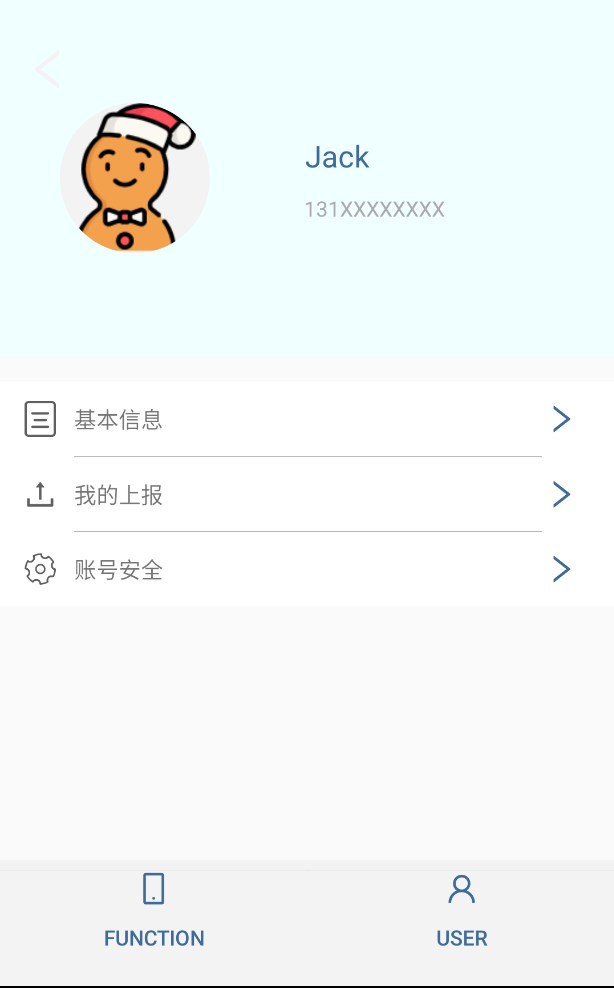


图8.1

此部分包含了用户的头像（头像上传功能尚未实现），用户名，手机号码，以及用户可以需要的功能，故障上报记录等等。此界面主要展示故障上报功能，如图8.2，8.3，8.4所示。



图8.2



图8.3



图8.4

图8.2，8.3，8.4分别是“我的上报”中的三种状态记录，分别为已提交，已完成和未完成三部分，点击按钮队友动态的选中效果，且相应位置的布局会被切换，显示内容。

**总结**

至此，目前所完成的UI界面基本展示完成，总体UI界面已经设计完成，还剩余二维码扫码界面以及用户功能的界面，将会在后续的工作中完成。

总体来说，目前的UI界面设计视觉效果还行，但是仍存在许多优化的地方，后续可能会研究一下颜色搭配基础，更好的AlertDialog样式以及Acitivity之间的切换流畅度等等。希望能够做得越来越好。