

**计算机学院移动智能应用设计课程设计说明书（产品实现方案）**

**设计名称：**《码上运维》产品设计

**组别：**第十八组

**组长：**张煜彬

**组员：**蒋春林、李昊、赵崇彦、郑晶霏

**指导老师：**李慧

**日期：**2021年1月9日

目录

[**一、** **产品的主要功能** 2](#_Toc61191880)

[**二、** **UI界面设计** 3](#_Toc61191881)

[**（一）初始界面（MainActivity）** 3](#_Toc61191882)

[**（二）登录界面（WorkerLoginActivity）** 4](#_Toc61191883)

[**（三）修改密码界面（WorkerChangePwdActivity）** 8](#_Toc61191884)

[**（四）注册界面（WorkerResigisterActivity）** 11](#_Toc61191885)

[**（五）工人主界面（WorkMainAcitivity）** 12](#_Toc61191886)

[**（六）巡检功能界面（CheckActivity）** 13](#_Toc61191887)

[**（七）上报故障界面、内容查看界面（My\_Upload,InformationActivity）** 15](#_Toc61191888)

[**（八）使用设备界面** 16](#_Toc61191889)

[**（九）用户界面（UserActivity）** 16](#_Toc61191890)

[**（十）流程图** 20](#_Toc61191891)

[**（十一）总结** 20](#_Toc61191892)

[**三、** **关键技术和技术难点** 20](#_Toc61191893)

[**四、** **用户体验记录和分析** 22](#_Toc61191894)

[**五、** **已完成的改进和存在的问题** 27](#_Toc61191895)

[**（一）改进方案** 28](#_Toc61191896)

[**（二）存在的问题** 29](#_Toc61191897)

1. **产品的主要功能**

这是一个基于APP的通用电力设备运维管理系统，旨在利用这个系统进行设备的运行维护，实现设备档案无纸化、计划制定自动化、维护流程规范化、故障响应实时化、数据分析可视化。

|  |  |
| --- | --- |
| **主要功能** | 通过扫描设备所携带的二维码，在安卓终端上查看、修改设备信息，进行日常运维的签到、登记的常规操作，完成对故障设备的信息上传、更新、维修，接收修复任务等功能。另外还有基础的不同类型客户的注册、登录、忘记密码、修改密码、更改权限信息等基础功能。 |
| **主要用户** | 主要用户是电力公司使用电力设备的人员以及进行设备维保的人员。 |

1. **UI界面设计**

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。因为是电力运维系统，所以想营造的是一种偏工业风的氛围。在色调的选择上，选择了给人感觉较“轻”的冷色调，给人一种清爽的感觉，也会给使用者冷静、舒适的感受。

控件摆放采用传统app的方式，使用方法简单明了，控件的间距始终，在填充页面的同时不会让人感到拥挤；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。

**（一）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。

**（二）登录界面（WorkerLoginActivity）**



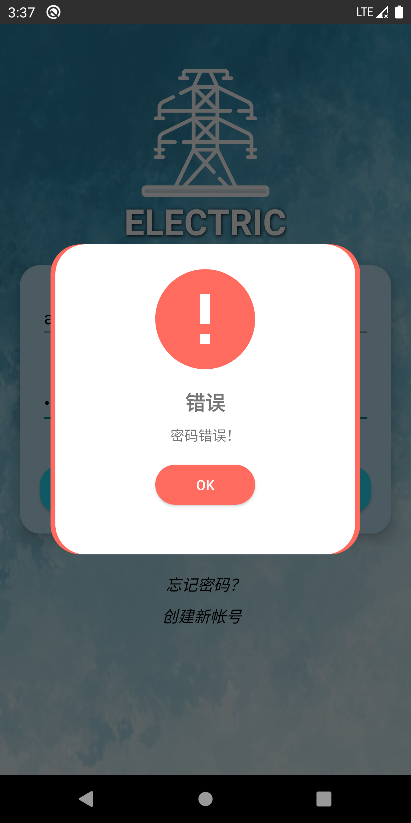
**图2.1**

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”修改咯，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。

**（三）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。



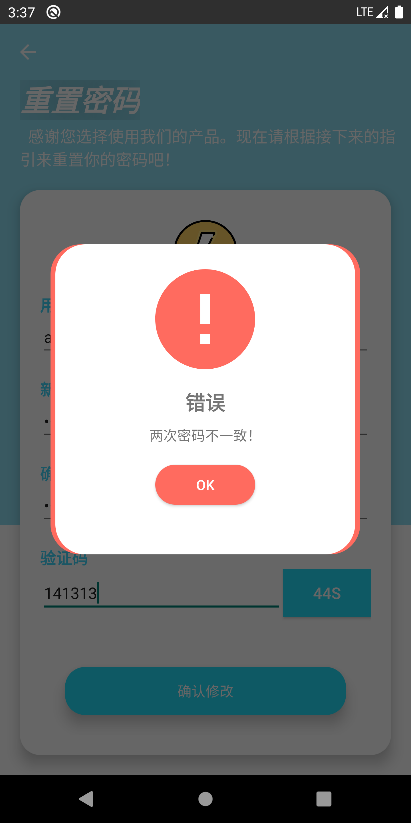
**图3.2**

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。



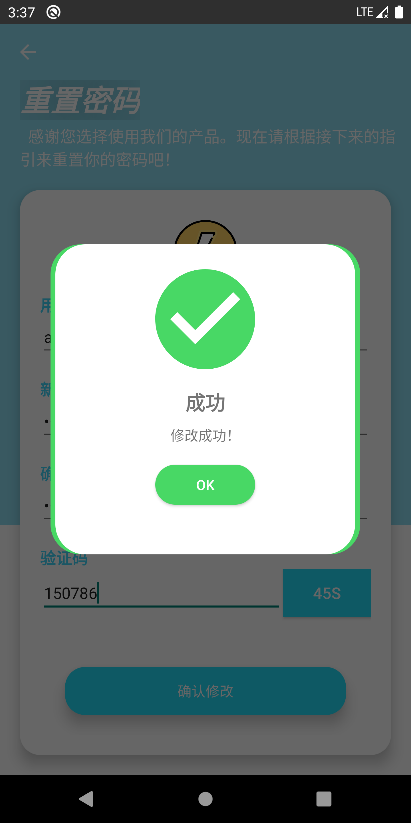
**图3.3**

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。

**（四）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。

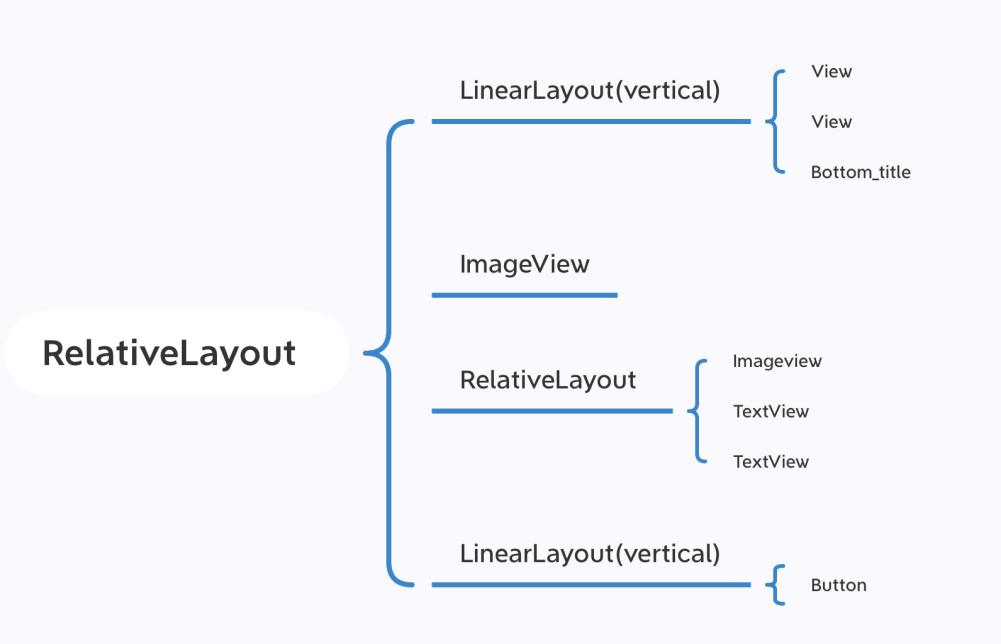
接下来就让我们进入到维修工人的界面吧！

**（五）工人主界面（WorkMainAcitivity）**



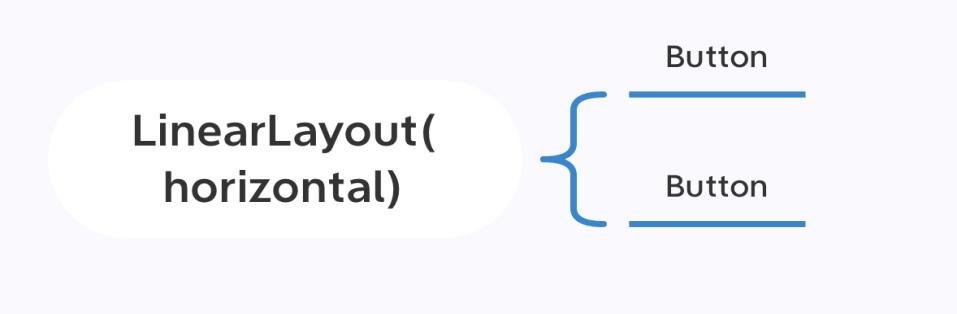
**图5.1**

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。

**（六）巡检功能界面（CheckActivity）**

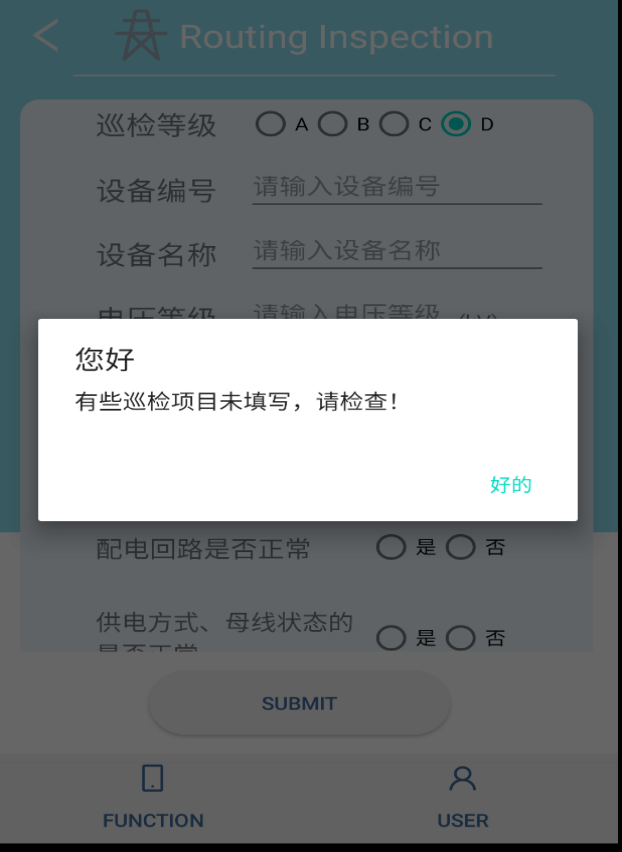


**图6.1**

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。

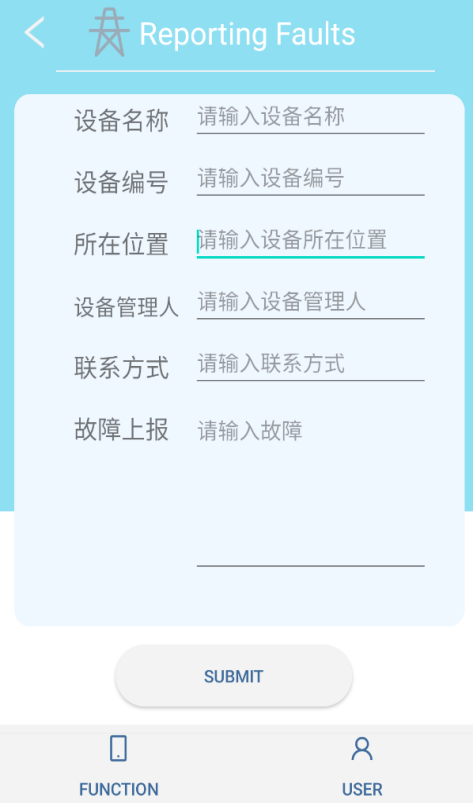
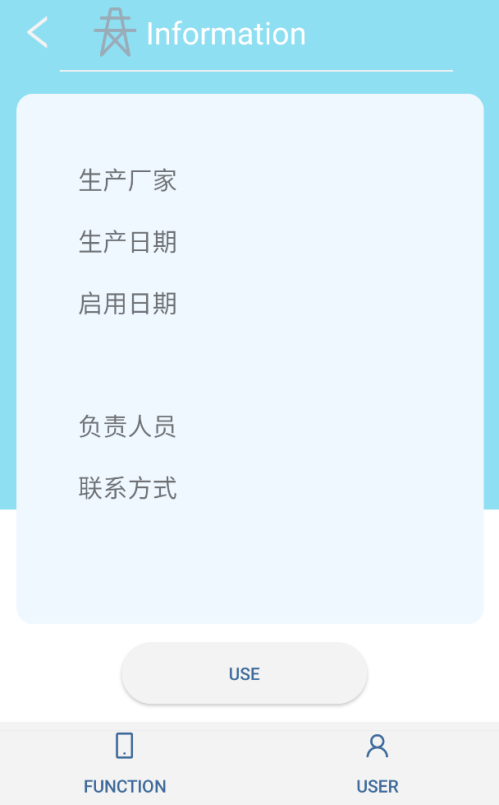


**图6.3**



**图6.4**

**（七）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。按钮是一个带有进度条的按钮，显示提交的进度。

**（八）使用设备界面**



**图8.1**

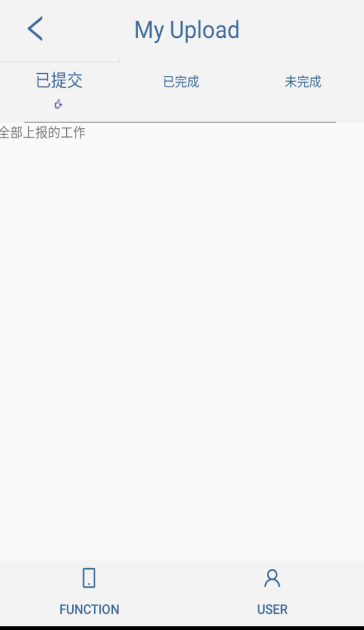
该界面提供三个按钮：拍照、选择和提交，拍照直接调用系统相机，选择可以进入相册选择图片，选择图片后图片会显示在中间外星人图片的部分。设备ID及使用人是自动填充。

**（九）用户界面（UserActivity）**



**图9.1**

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、系统设置、设备使用记录等等。此界面主要展示故障上报功能，如图9.2，9.3，9.4所示。

**图9.2 图9.3 图9.4**

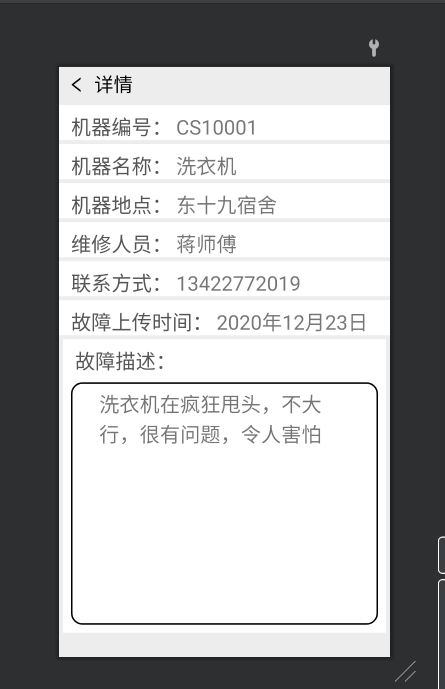
图9.2，9.3，9.4分别是“我的上报”中的三种状态记录，分别为已提交，已完成和未完成三部分，点击按钮队友动态的选中效果，且相应位置的布局会被切换，显示内容。

提交了内容后如9.5图：



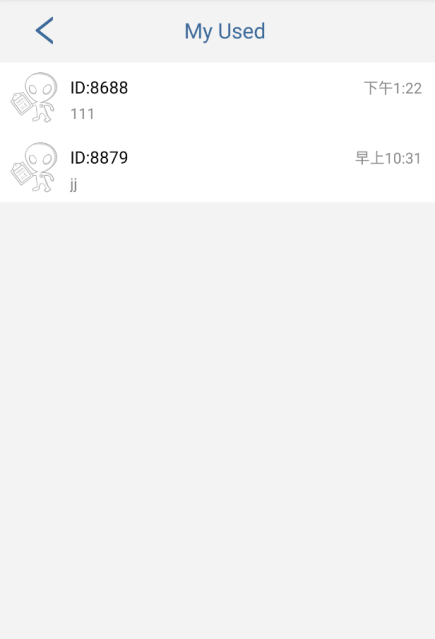
**图9.5**

点击相应的部分可以查看具体的内容情况，如图9.6



**图9.6**

我的使用功能：



**图9.7**

图9.7展示的是“我的使用”中设备的记录情况，包括了被使用的设备ID、使用时间、还有一些备注等。如果使用设备时上传了照片，也会在照片栏中显示里面的内容可以侧滑删除。

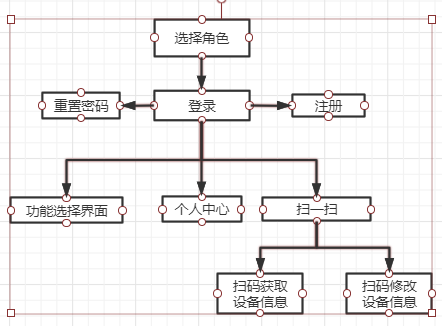
设置功能：



**图9.8**

9.8图内容如下：包括了修改昵称、修改手机号、修改密码、开启通知、退出当前账号的功能，另外在开启通知的部分用了一个TriStateToggleButton，具有两种状态（开启/关闭）。

**（十）流程图**



**（十一）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，还剩余一些细节的功能，比如开启通知、更改用户名等，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。并且希望能加入更多的动效元素，增强与用户的交互感与使用感。

1. **关键技术和技术难点**

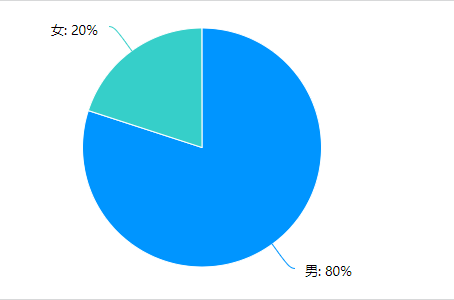
|  |  |
| --- | --- |
| **关键技术** | **技术在软件中的使用** |
| **二维码技术** | 二维码可用于存储各种且二维码是唯一的。因此我们使用二维码用于存储设备ID。这样实现了快速查找设备并在线上管理设备。 |
| **HTTP** | 使用http与后台进行各种数据交流，实现了普通工人端和维保工人端的数据同步。这个过程对于用户来说是透明的。 |
| **SpringBoot框架** | 使用Springboot快速搭建后台，并且方便了数据库资源和servlet资源的分配。 |

|  |  |
| --- | --- |
| **技术难点** | **解决方案** |
| **二维码扫码、解码技术** | 在本项目中，没有实现二维码扫码技术，而是通过拍照后上传二维码。二维码解码通过GitHub上封装过的zxing包实现。 |
| **打开系统相机并获取拍照结果** | 首先必须先在程序全局文件中声明获取打开相机权限。在使用相机检验是否已经获取权限，如果为获取则调用系统函数进行权限申请。如果已经申请则调用打开系统相机并在onActivityResult中拿到拍照的结果、做进一步的处理。 |
| **Hhtp请求、接收与解析** | 使用http3包进行实现http请求和接收，使用Gson进行请求内容的解析。 |
| **全局权限的传递** | 使用Android框架中的全局类（application）类实现。 |
| **异步刷新页面** | 使用handler实现。为activity定义一个handler并在其中书写更新页面的逻辑。 |
| **Activity的管理** | 在Android中可以通过返回键回到上一个activity中。所以在每次跳转时需要把上一次的activity销毁。使用finish()实现。 |
| **下拉刷新** | 先销毁原有的activity，然后显示下拉效果，下拉完毕后显示新的activity。此时activity内的数据已经更新。 |

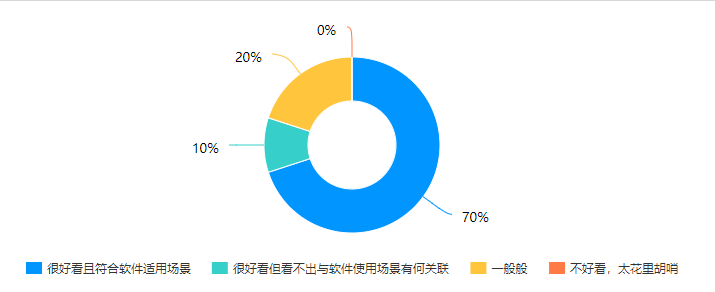
1. **用户体验记录和分析**

本小组根据本项目软件实际情况从软件界面、软件运行状态、软件容错机制、软件功能、对软件的建议和总体感受等方面设置了调查问卷（见附件），用于用户产品的体验记录。本次用户体验调查中，本小组共找10位同学进行了app的使用体验，并根据他们的使用感受按照实际情况填写了使用调查，最终得到10份有效体验记录（见附件），现根据用户体验记录数据对本小组app分析如下：

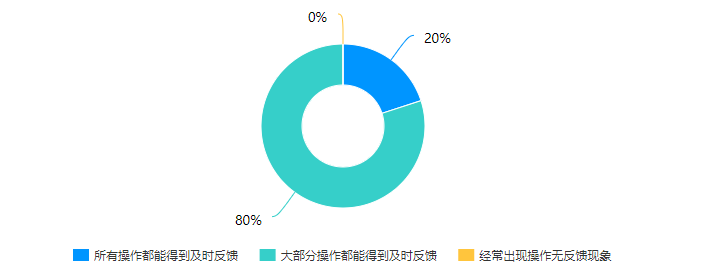
**1、**本次邀请进行产品使用体验的同学中男同学占比80%，女同学占比20%，同时覆盖了计算机学院、数学科学学院、美术学院等非计算机专业和计算机专业的同学，进行了猴子测试法的测试方法，具有较高的代表性和说服性。



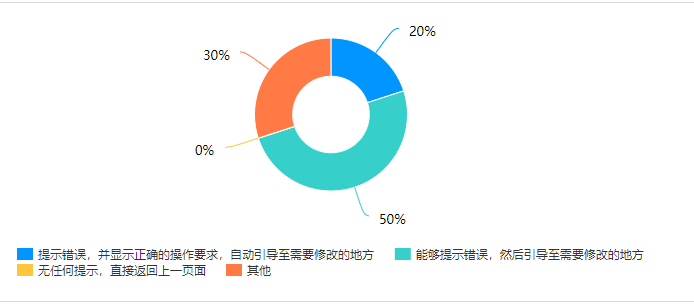
1. 在对软件界面的体验调查中，有**超过70%**的同学认为本小组的app界面**风格好看且符合适用场景，**这说明本项目的界面设计已经达到预期目的，可以给予用户较好的体验。总体上能够有吸引用户使用该产品。但是还有**20%**的体验用户认为本小组app的**界面一般般**，显然以目前的界面风格还是没有达到部分用户群体的期待要求，这对本小组的界面设计以及完善提出了新的要求和目标。



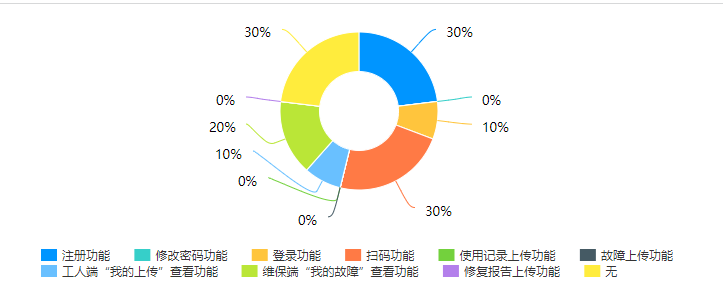
1. 在对系统运行状态的记录中，**超过80%**的用户认为本组产品的**大部分操作都可以得到及时反馈**，这说明本小组的app已经**大致实现了预期设计的所有按键响应和操作反馈**，这部分的用户体验反馈对我们产品打出了较高的分数，是对我们产品的认可。但是这部分数据说明我们产品还存在部分操作无法正常进行响应反馈，这对我们进一步优化产品提供了新的思路和方向。



在对软件容错性的调查中，我们发现在有操作错误的70%用户中，有**100%的错误操作可以提示错误，不存在出现错误操作而系统无所作为**的情况，表明我们的系统已经满足基本的容错性要求，但是在这些错误提示中**仅有不到30%的情况显示了正确的操作要求**，这说明本软件对待错误操作时对用户的提示指引还不足，同时还没有办法自动改正错误操作，这将是本产品在容错性方面继续优化的目标。



1. 在对系统功能是否正常运行的调查中，出现了较多功能用户体验不好的情况，这些功能主要集中在注册功能（用户反映不足程度30%），登录功能（用户反映不足程度10%），扫码功能（用户反映不足程度30%），工人端“我的上传”查看功能（用户反映不足程度10%），维保端“我的故障”查看功能（用户反映不足程度20%）。



**（1）其中注册功能反映的问题有：**



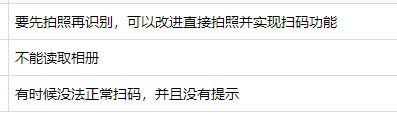
注册功能出现以上不正常运行情况的原因一方面是注册功能后端代码的逻辑出现了问题，具体表现在界面逻辑跳转和提示框显示，一方面是注册界面前端代码的问题，具体表现在数据文本框显示与不显示属性混淆，最后一方面是数据库唯一性未保障。

**（2）登录功能反映的问题有：**



该问题的存在给予了用户很不好的使用体验，当需要切换账号的时候都要退出整个app，无法实现软件操作的连续性。

**（3）扫码功能反映的问题有：**



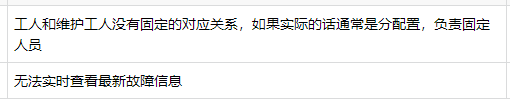
扫码功能出现以上不正常运行情况的原因是因为无法实现类似微信扫码的模式，而本产品的扫码模式为先拍照二维码，然后上传后台进行解析。基于这种扫码模式，就容易出现拍了照片而无法正常解析的情况，同时也没有给予用户扫码失败操作，使用户体验感下降。此外本产品的设备扫码功能也不支持上传本地图片进行解析，实用性进一步下降。

**（4）工人端“我的上传”查看功能反映的问题有：**



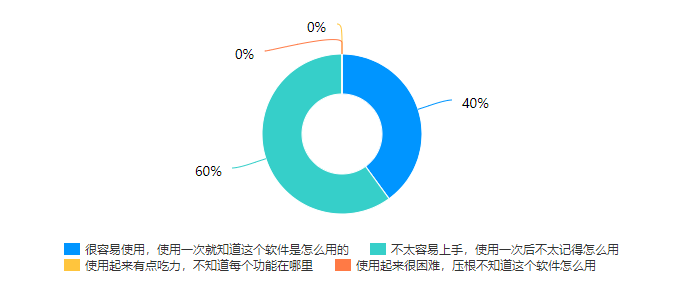
该问题的存在无法使用户实时获取最新的故障或者维修信息，需要退出页面后才可以查看最新的消息，用户体验感很不好，所以在这个界面要增加刷新功能，使用户不用退出该界面就可以获取最新的信息。

**（5）维保端“我的故障”查看功能反映的问题有：**

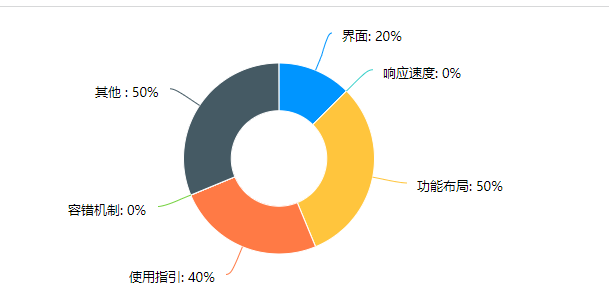


维保端“我的故障”查看功能的问题与工人端“我的上传”查看功能的问题类似，但对于用户反映的第一个工人和维护工人无固定对应关系的问题，由于本产品是实际应用于电力设备公司的设备维保方面，所以从公司维保的角度看不需要工人和维护工人有固定对应关系，而是根据工作任务量随机将故障分配给维保工人。

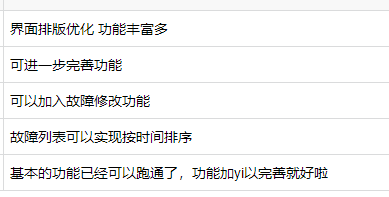
**5、**在对软件的总体评价中，有40%的用户认为该产品容易使用，使用一次就知道怎么用，但是还存在60%的用户认为本产品不太容易上手。由于**本产品的定位使用环境是在电力设备公司的维保需求下**，所以本次进行软件体验的用户在**无相关经验**下对本产品的使用会有一定困难，同时本产品**暂未提供使用说明加剧了这种使用难度，这属于正常现象。**但是用户对本产品的总体评价达到该数据水平说明本产品的用户交融还是比较好的。



**6、体验用户在对于本产品的建议数据如下：**



其中界面的用户建议改进程度是20%，功能布局的用户建议改进程度是50%，使用指引的用户建议改进程度是40%，同时还有其他的改进建议，如下：



这些体验用户所提建议是我们产品进行改进和优化途中的宝贵资料。

1. **已完成的改进和存在的问题**

**（一）改进方案**

改进方案中部分方案未执行。

根据用户体验记录的分析可以总结出本小组app的改进方案如下：

1. **界面**
2. 进一步优化界面设计和UI动画效果，加入更多产品使用环境的图片元素。
3. 对界面的功能布局进行重新编排，使界面的功能排版更加简洁易懂，让软件重点或者常用功能放在界面显眼的重点位置。
4. 对界面的细节部分进行优化，比如文本框、按钮、标题的对齐和清晰显示，避免各种组件相互遮挡，同时设置好各个组件的属性如隐藏和显示等。
5. 对界面跳转的逻辑和界面提示框的显示进一步优化，根据实际跳转情况消除已有界面，避免不符合要求的返回。
6. **功能**
7. 增加退出登录选项，使app可以在不退出的情况下随意切换登录用户。
8. 改进扫码功能，实现微信扫码模式，同时增加从本地上传照片进行解析的支持项。
9. 在查看功能部分增加刷新功能，实现表单的下拉刷新获取最新数据。同时对表单数据进行排序，以按照时间的顺序来排序。
10. 对各个按钮或者表单项的响应事件和反馈进行进一步优化，避免出现操作无响应情况。
11. 对未实现功能进行具体实现，如个人中心的个人基本信息查看，使用记录上传使用照片等。
12. 改进app的容错机制，对操作错误或者系统本身出现的错误进行精确提示，避免出现系统崩溃的情况。此外对出错情况进行正确操作的提示，尽力实现对错误操作或者系统错误的自动修复和修改。
13. **后台**
14. 对后台代码进一步优化，同时增加后台的健壮性。
15. 对后台数据库进行唯一性保障，同时对个人用户基本信息数据进行扩充和丰富，请求专业人员对设备基本信息进行扩充和修改。
16. 涉及到两个表的查询语句使用联表查询，提高后台性能。
17. **其他**
18. 制作产品app使用指引和手册，保障普通用户初次使用也可以正确操作，支持用户在短时间内掌握app的功能使用。
19. 进行app的进一步宣传，获得用户群体的支持和青睐。

**（二）存在的问题**

1、 后台程序中涉及到两个表的查询语句没有使用联表语句，对性能有一定的影响。

2、目前扫码功能通过拍照上传二维码实现，没有实现类似微信扫码的实时扫码，同时不支持从本地文件上传照片解析。

3、界面风格和UI效果还不足，同时功能布局不够简洁明了，文本框、按钮、标题等组件排版不够整齐，界面跳转和提示框逻辑有待改善。

4、某些功能还未实现，查看功能的表单数据未实现有序排列，系统容错机制还不足，系统容易出现崩溃现象。

5、系统目前还没有使用手册和使用指引，这对初始使用者不够友好。