图片包含 文本

描述已自动生成

**基于 新闻APP  
项目系统开发说明文档**

小组 13组

小组成员 李虹、田野、刘晓艺、梁林英

院系＿计算机学院

班级＿网络工程班＿

目录

[<一> 产品设计方案 1.项目实施可行性报告 1.1行业市场分析 3](#_Toc137812303)

[1.2同类产品分析 4](#_Toc137812304)

[1.3自身条件分析 5](#_Toc137812305)

[2.产品的定位和目标 5](#_Toc137812306)

[2.1 产品目标 5](#_Toc137812307)

[2.2 产品定位 5](#_Toc137812308)

[2.3 用户群分析 6](#_Toc137812309)

[３．产品内容总策划 7](#_Toc137812310)

[３.１应用流程规划 7](#_Toc137812311)

[3.1.1系统功能模式图 7](#_Toc137812312)

[3.1.2数据流程图 8](#_Toc137812313)

[3.２设计与测试规范 8](#_Toc137812314)

[3.2.1业务逻辑测试 8](#_Toc137812315)

[3.2.2安全测试 10](#_Toc137812316)

[3.2.3专项测试 10](#_Toc137812317)

[3.３开发日程表 13](#_Toc137812318)

[4.技术解决方案 14](#_Toc137812319)

[4.1 UI设计与实现： 15](#_Toc137812320)

[4.2 数据获取和缓存： 15](#_Toc137812321)

[4.3 架构模式和组件化编程： 15](#_Toc137812322)

[4.4 推荐算法和个性化推荐: 16](#_Toc137812323)

[4.5第三方服务和API集成： 16](#_Toc137812324)

[4.6 安全性和可靠性保障： 16](#_Toc137812325)

[5.推广方案 工作需求：吸引用户、留住用户。 16](#_Toc137812326)

[5.1渠道推广 1）基础上线：各大下载市场、应用商店、大平台、下载站的覆盖Android版本发布渠道，推广的第一步是要上线，这是最基础的，无需砸钱，只需最大范围的覆盖。 17](#_Toc137812327)

[6.运营规划 19](#_Toc137812328)

[6.1市场分析 19](#_Toc137812329)

[<二> 产品实现方案 20](#_Toc137812330)

[1.UI设计和软件完成情况展示 1.1项目原型 20](#_Toc137812331)

[2.关键技术以及难点 关键技术：json数据解析+api接口调用 36](#_Toc137812332)

[<三> 用户体验记录和改进方案 38](#_Toc137812333)

[1.用户体验和建议 38](#_Toc137812334)

[2.已完成的改进和存在的问题 40](#_Toc137812335)

[问题一：新闻分栏:标题不显著 40](#_Toc137812336)

[问题二：登录方面:个人登录后功能太少，且登录方式单一 40](#_Toc137812337)

[问题三：APP背景选择：太过花哨影响阅读体验 41](#_Toc137812338)

[问题四：在进行标准兼容测试时，有45%的测试设备出现APP Crash问题 41](#_Toc137812339)

[<四> 测试大纲和测试报告 41](#_Toc137812340)

[1.UI测试 41](#_Toc137812341)

[1.1.1 用户界面布局 41](#_Toc137812342)

[1.1.2 用户友好性 42](#_Toc137812343)

[1.2 功能测试 43](#_Toc137812344)

[1.2.1 登录、注册 43](#_Toc137812345)

[1.2.2 新闻页面 44](#_Toc137812346)

[1.2.3 安装和卸载 46](#_Toc137812347)

[1.3 标准兼容测试 46](#_Toc137812348)

[1.3.1 测试机型 46](#_Toc137812349)

[1.3.2 测试结论 47](#_Toc137812350)

[1.3.3 问题分布 48](#_Toc137812351)

[1.3.4 问题机型 49](#_Toc137812352)

[1.3.5 性能分析 49](#_Toc137812353)

[<五> 团队分工 55](#_Toc137812354)

### **<一> 产品设计方案** 1.项目实施可行性报告 1.1行业市场分析

技术发展总是推动着传播方式的革新，互联网技术正在加速与社会各领域的深度融合，不仅改写着现代人的生活方式，也推动着传媒行业的转型。根据中国互联网络信息中心（CNNIC）发布的中国互联网络 展状况统计报告显示，截至2017年12月，我国手机网民规模达7.53亿，网民中使用手机上网人群的占比97.5%，网民继续向移动端渗透。2017年，各类手机应用的用户规模不断上升，其中，新闻 App 是最基础的互联网应用之一，手机网络新闻用户规模达到6.19亿，占手机网民的82.3%。新闻 App 之所以能够有广阔的市场，是由于人们对于新闻有兴趣有需求，新闻的种类在如今也是五花八门，军事新闻、政治新闻、娱乐新闻、体育新闻等等，人们关注着各种各样的 新闻，在新闻中获取自己生活工作所需信息，因此新闻具有广阔的市场。而另一方面，许多新闻媒体也希望通过这一新的阅读模式来吸引读者，扩大信息传播的途径，增加收益，所以在应用商店中新闻App 成为必不可少的应用软件。在现今，纸媒和互联网已经不是新闻最新潮流的媒体，手机体积小便于携带，已经成为了人们首选的阅读方式，再加上 App 新闻在手机上的准入门槛比较低，使得新闻行业内新闻 App 的发展成为必然。

## 1.2同类产品分析

当下常用的新闻类APP中，基本由互联网公司和传统媒体成立，从大方向上可以分为三类：门户网站的APP，传统媒体的独立APP，聚合类新闻APP。例如腾讯旗下的腾讯新闻、网易旗下的网易新闻、搜狐建立的搜狐新闻、新浪旗下的新浪新闻等，刚开始都是以新闻门户网站的形式存在，后来纷纷建立了自己的APP;传统媒体拥抱互联网和新媒体浪潮后也成立了自己的新闻APP，例如凤凰台的凤凰新闻、南方报业集团的南方周末、央视新闻和人民日报自身的同名APP平台;聚合类新闻APP中比较知名的则有今日头条、百度新闻、ZAKER等等。像腾讯新闻、网易新闻、新浪新闻这类由门户网站发展建立的APP，背靠互联网企业，可以借助企业旗下的其他平台进行推广，腾讯可以借助QQ和微信、网易可以借助网易云音乐和有道词典、新浪可以借助新浪微博，所以这些新闻APP平台的用户基数大，传播范围更加广泛，但和传统媒体成立的新闻APP相比，新闻的质量的深度都有所不足;传统媒体成立的APP也具有自身的优势，它们受传统媒体的影响，所发出的新闻更加严谨和客观，质量更高，新闻评论更有深度，所以用户黏性高，但在用户体量和传播度上有所欠缺;聚合类新闻APP由第三方技术团队开发，通过整合内容资源或利用其他网站的公开接口，调用这些网站的新闻内容进行编排提供给用户，自身缺乏生产新闻的能力，但这类APP非常重视用户体验，会针对用户的阅读喜好进行个性化推荐，娱乐性质浓厚。

## 1.3自身条件分析

1、APP可以使用的技术实现方案例如数据爬虫技术、数据库技术、人工智能小组成员已经接触过并掌握一些，移动应用开发技术将会进一步学习。

2、小组成员熟练掌握Python、java等编程语言。

3、互联网上现有许多成熟的APP，其功能和优势可以作为参考。

# 2.产品的定位和目标

新闻app是一个基于移动设备的新闻阅读应用程序，旨在为用户提供及时、准确、全面的新闻报道和分析。

## 2.1 产品目标

新闻App的产品定位是提供给用户及时、全面、准确的新闻报道和资讯，满足用户对于新闻及时获取和深度了解的需求。其目标是通过优质内容、个性化推荐和数据分析等手段，不断提升用户体验和满意度，稳定用户群体，并扩大用户规模，成为用户首选的新闻资讯平台。

## 2.2 产品定位

1. 实时性：新闻App需要第一时间获取最新消息，及时更新新闻资讯，让用户获得最全面的信息；

2. 内容丰富：新闻App需要提供多元化的新闻内容，包括国际、国内、社会、财经、科技、娱乐、体育等方面的报道，以满足用户的不同需求；

3. 个性化推荐：新闻App需要根据用户的浏览历史和兴趣爱好，推荐符合用户口味的新闻内容，提高用户的阅读体验；

4. 数据分析：新闻App需要通过用户行为数据分析，了解用户的需求和偏好，优化产品功能和内容，进一步提升用户体验。

## 2.3 用户群分析

以下是针对新闻App用户群的分析报告：

1. 用户年龄和性别分布：

- 年龄分布：以18-35岁为主，其中25-30岁的用户数量最多；

- 性别分布：男女比例大致相等，略偏向男性用户。

2. 用户地域分布：

- 大城市人口集中，如北京、上海、广州、深圳等；

- 许多用户属于一线城市，但也有不少二线城市的用户。

3. 用户使用场景和行为特点：

- 大部分用户在早晨、晚上以及周末使用App，午间使用人数相对较少；

- 大多数用户的使用时间不超过30分钟，快速浏览新闻内容，少部分用户会针对感兴趣的内容进行详细阅读；

- 很多用户喜欢分享讨论热门新闻，社交互动特性明显。

4. 用户偏好和需求：

- 新闻类别方面，用户对时事、财经、科技、娱乐等领域的关注度较高；

- 用户更喜欢简短、有趣、易读、内容有价值的文章；

- 对于推送功能，用户更倾向于个性化推荐，推文量不宜过多。

基于以上分析结果，建议新闻App在以下方面做出改进：

- 加强对年轻用户的吸引力，提供适合该年龄段的新闻内容和服务；

- 改进推送算法，提高个性化推荐的准确度；

- 优化内容质量和阅读体验，以及社交互动体验；

- 考虑地域差异，丰富不同城市区域的新闻内容。

# ３．产品内容总策划

３.１应用流程规划

### 3.1.1系统功能模式图

根据系统功能要求，可以将系统分解成几个模块来分别设计应用程序界面。

图示

描述已自动生成

### 3.1.2数据流程图

图示

描述已自动生成

3.２设计与测试规范

### 3.2.1业务逻辑测试

1. UI交互体验测试
2. 页面按照UI设计原型图实现，布局符合用户使用习惯。
3. 恰当的利用窗体和控件的空白，以及分割线条。
4. 窗口切换、移动、改变大小时，界面显示正常。
5. 刷新后，页面显示正常。
6. 无错别字，或者提示性文字措词恰当准确。
7. 控件（如菜单、对话框、按钮，单选框、复选框）的布局、风格、是否正确，界面是否美观，操作是否友好。
8. 业务逻辑功能测试
9. 正向用例

针对被测功能点的特性列出相应类型的测试用例对其进行覆盖。

1. 反向用例

结合各自业务场景，通过八大用例设计方法来编写。

1. 正常用例调整

测试实现的各个阶段跟踪测试实现与需求输入的覆盖情况。

1. 特殊功能点
2. 推送通知（PUSH消息）

检查Push消息是否按照指定的业务规则发送；

检查在免打扰时间段内，用户是否接收不到Push；

点击推送消息后是否跳转至应用里；

检查Push消息是否只针对登录用户；

检查不接收推送消息时，用户是否不会再接收到Push消息。

1. 账号注册

通过唯一用户名注册；

1. 登录

静态密码登录；

### 3.2.2安全测试

* 1. 敏感信息是否加密。
  2. 必要参数是否后端也进行校验。
  3. 接口是否越权请求。
  4. 接口是否能防止拦截篡改。
  5. 接口参数不能是密码明文。
  6. 控制台和日志均不能打印出密码明文。

### 3.2.3专项测试

* + 1. 权限测试

1. 扣费风险：包括短信、拨打电话、连接网络等。
2. 隐私泄露风险：包括访问手机信息、访问联系人信息等。
3. 对App的输入有效性校验、认证、授权、数据加密等方面进行检测。
4. 限制/允许使用手机功能接入互联网。
5. 限制/允许使用手机发送接受信息功能。
6. 限制/允许使用本地连接。
7. 限制/允许使用手机读取用户数据。
8. 限制/允许使用手机写入用户数据。
9. 限制/允许使用应用程序来注册自动启动应用程序。
   * 1. 安装、卸载和升级
10. 安装

软件安装后是否能够正常运行，安装后的文件夹以及文件是否写到了指定的目录里；

软件安装各个选项的组合是否符合概要设计说明；

软件安装向导的UI测试；

安装后没有生成多余的目录结构和文件。

1. 卸载

测试系统直接卸载程序是否有提示信息；

测试卸载后文件是否全部删除所有的安装文件夹；

卸载是否支持取消功能，单击取消后软件卸载的情况。

1. 升级

后台配置升级，APP启动正常；

应用商店升级，APP启动正常。

* + 1. 网络测试

1. WIFI和5G间、5G和4G间、4G和3G间、WIFI和无网络间、5G和无网络间切换时，APP的数据提交和页面跳转加载，数据查询等场景下，APP响应正常。
2. 低丢包率（5%）时，低延迟（上行）和（下行）；

低丢包率（5%）时，高延迟（上行）和（下行）；

高丢包率（5%）时，低延迟（上行）和（下行）；

高丢包率（5%）时，高延迟（上行）和（下行）。

1. 使用真实的SIM卡在移动，电信，联通间，选择真实的5G、4G、3G、2G网络，做测试。
   * 1. 兼容性测试
2. 与本地及主流App是否兼容；
3. 不同分辨率下，APP的UI交互体验是否友好；
4. 相同品牌不同操作系统UI和功能表现是否一致。
   * 1. 中断测试
5. 锁屏中断：APP运行时，锁屏后再打解锁，app 仍处于该页面；
6. 插拔数据线：APP运行时，插拔数据线后，对APP无影响；
7. 普通SIM电话来电中断；
8. 网络电话中断：微信视频电话、QQ语音电话等；
9. 闹钟中断；
10. 蓝牙文件传输中断等；
11. 断电和自动关机等造成的中断。
    * 1. 特殊场景
12. 前后台切换（多任务切换和单任务切换）；
13. 横竖屏切换中，APP是否有异常；
14. 分享功能（微信和QQ，微博等），在第三方软件中打开无异常；
    * 1. 功耗测试
15. 响应时间

APP安装、卸载的响应时间；

APP各类功能性操作的响应时间。

1. 系统资源占用

页面跳转过程和提交过程查看CPU占用率；

页面跳转过程和提交过程查看内存使用率；

一段时间后，查看APP的耗电量；

页面流量的大小，结合实际业务场景是否有异常。

3.３开发日程表



# 4.技术解决方案

基于JAVA语言开发新闻App

## 4.1 UI设计与实现：

可以使用Sketch工具进行UI设计，并遵守Material Design规范，确保应用程序快速响应，同时提供优秀的用户体验。使用ConstraintLayout或LinearLayout等布局实现界面的布局。使用RecyclerView控件来处理列表和网格布局。

## 4.2 数据获取和缓存：

使用Retrofit或OkHttp等网络框架来访问后台数据。将返回的JSON数据反序列化到自定义的Data Model中，以方便操作。同时在本地使用SQLite、Realm或Room等本地数据库进行缓存，以便在离线状态下访问数据。使用Glide或Picasso等图片加载库来加载网络上的图片。

## 4.3 架构模式和组件化编程：

采用MVVM或MVC架构模式和组件化编程，以方便代码维护和测试。使用LiveData或RxJava等响应式编程库，将数据流动自动更新到UI控件上。采用Dependency Injection（DI）来轻松管理代码库的依赖关系，以使应用程序的各个部分之间彼此独立。

## 4.4 推荐算法和个性化推荐:

使用机器学习等技术来优化新闻推荐算法，从而实现个性化推荐。这需要预先对用户进行分类，使用不同的算法来分析他们的阅读行为和兴趣，然后根据这些信息来推荐新闻。

## 4.5第三方服务和API集成：

与第三方服务如Weibo Sina等API集成，以实现新闻定位和社交分享等功能。同时，通过开放的API来增强应用程序的可扩展性和灵活性，进一步提高用户的互动性和粘性。

## 4.6 安全性和可靠性保障：

考虑安全性是应用开发过程中的重点之一。使用SSL或HTTPS等协议加密用户数据传输，从而保护用户数据的机密性和完整性。使用第三方代码库的更新机制来规避漏洞风险。在生产环境发布之前进行全面测试并确保稳定性，同时配备日志分析工具以便于监测应用程序错误和服务器不良行为。

# 5.推广方案 工作需求：吸引用户、留住用户。

基于市场流行趋势和通常推广手段，选出以下适合于本产品的推广手段，几种推广方式并行。

## 5.1渠道推广 1）基础上线：各大下载市场、应用商店、大平台、下载站的覆盖Android版本发布渠道，推广的第一步是要上线，这是最基础的，无需砸钱，只需最大范围的覆盖。

下载市场：安卓、安智、应用汇、91、木蚂蚁等。

应用商店：google商店、魅族商店、联想开发者社区、oppo应用商店等。

大平台：MM社区、沃商店、天翼空间、华为智汇云、腾讯应用中心等。

客户端：豌豆荚手机精灵、91手机助手、360软件管家等。

iOS版本发布渠道：AppStore、91助手、网易应用中心、同步推、快用苹果助手、itools、限时免费大全。

**5.2新媒体推广**

1）内容策划

根据受众定位，分析得出核心用户特征。

坚持原创内容的产出，在内容更新上保持一天三条左右有趣的内容。

抓住当天的热点跟进。

创意，让你的产品讲故事，拟人化。

2）品牌基础推广

百科类推广：在百度百科，360百科建立品牌词条。

问答类推广：在百度知道，搜搜问答，新浪爱问，知乎等网站建立问答。

3）论坛，贴吧推广

机锋、安卓、安智……，在手机相关网站的底端都可以看到很多的行业内论坛。

以官方贴、用户贴两种方式发帖推广，同时可联系论坛管理员做一些活动推广。

发完贴后，应当定期维护好自己的帖子，及时回答用户提出的问题，搜集用户反馈的信息，以便下个版本更新改进。

4）微博推广

内容：将产品拟人化，讲故事，定位微博特性，坚持原创内容的产出；在微博上抓住当周或当天的热点跟进，保持一定的持续创新力。参考同行业运营比较成功的微博大号，借鉴他们的经验。

互动：关注业内相关微博账号，保持互动，提高品牌曝光率。

活动：必要时候可以策划活动，微博转发等。

5）微信推广

微信公众号的运营推广需要一定时间沉淀，推广几步曲：

内容定位：结合产品做内容聚合推荐，内容不一定要多，但是一定要精并且符合微信号的定位。

小号积累：开通微信小号，每天导入目标客户群。

小号导大号：通过小号的粉丝积累推荐微信公众号，将粉丝导入到微信公众号。

**5.3线下推广**

宣传单：与合作商家商议，将宣传单曝光于商家跟用户接触的地方。

地推卡传单：制作精美传单，在办公区域相对集中的地方、商场发布传单。

# 6.运营规划

## 6.1市场分析

随着文化内容产业的发展进入稳定阶段，人们对精神文化层面的消费有了更加精细化的需求。不同特征的细分人群有了明显的媒体选择偏好。

新闻APP很大程度上满足了用户获取资讯这一内容消费刚需。新闻资讯作为文化内容产品大致经历了三个发展阶段：依赖于广播、电视、纸媒等渠道的传统媒体时代；以门户网站为代表的PC互联网时代；内容分发、即使互动、个性化推荐的移动互联网时代。

# <二> 产品实现方案

## 1.UI设计和软件完成情况展示 1.1项目原型

仿照腾讯新闻 实现侧边栏（登录注册关于） 新闻5快内容列表，（新闻内容的刷新、新闻详情、新闻搜索)

系统设计：

每条新闻类：

**public class** New {  
 **String title;  
 String time;  
 String src;  
 String pic;  
 String weburl**;  
  
}

新闻的标题title、新闻时间time、新闻出版社src、新闻图片pic、及新闻链接weburl

**1.2新闻部分总体情况：**

首先创建新项目并进行APP界面布局。项目的工程结构图及APP界面布局图如下。

项目工程结构图：图形用户界面, 应用程序

描述已自动生成

APP首页布局图：图标

描述已自动生成

真机显示图：新闻页面用listviewer实现，每个列表显示新闻的标题，出版社，新闻时间和新闻图片.

图形用户界面, 应用程序

描述已自动生成

**3.具体实现功能：**

**头顶标签功能的实现。**头标签主要使用的是fragent控件的使用，并且分为6块内容。如下图所示：

图形用户界面

低可信度描述已自动生成、

实现代码快：

**my\_tab**=(TabLayout) findViewById(R.id.***my\_tab***);  
 **my\_viewpager**=(ViewPager) findViewById(R.id.***my\_viewpager***);  
 **fragments**=**new** ArrayList<>(); *//碎片的集合* **fragments**.add(**new** Frgment1());  
 **fragments**.add(**new** Frgment2());  
 **fragments**.add(**new** Frgment3());  
 **fragments**.add(**new** Frgment4());  
 **fragments**.add(**new** Frgment5());  
 **fragments**.add(**new** Frgment6());  
  
 **titles**=**new** ArrayList<>(); *//标题的集合* **titles**.add(**"要闻"**);  
 **titles**.add(**"财经"**);  
 **titles**.add(**"NBA"**);  
 **titles**.add(**"科技"**);  
 **titles**.add(**"体育"**);  
 **titles**.add(**"军事"**);  
  
 MyAdapter adapter=**new** MyAdapter(getSupportFragmentManager());  
 **my\_viewpager**.setAdapter(adapter);  
 **my\_tab**.setupWithViewPager(**my\_viewpager**); *//关联TabLayout和ViewPager* toolbar.setNavigationOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 DrawerLayout drawerLayout = findViewById(R.id.***drawerlayout***);  
 drawerLayout.openDrawer(navigationView);  
 }  
  
 });  
 imageButton.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = **new** Intent(MainActivity.**this**,login.**class**);  
 startActivityForResult(intent,1);  
 }  
 });  
  
}

**上拉下拉功能的实现：**

上拉下拉刷新通过网上百度获取的动画效果编写而成

上拉和下拉效果如下图：

图形用户界面, 应用程序

描述已自动生成

实现代码：

**public** View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {  
 *//View view = inflater.inflate(R.layout.activity\_frgment1, null);* View view = inflater.inflate(R.layout.***activity\_frgment1***, container, **false**);  
 *//return view;* **if**(!EventBus.*getDefault*().isRegistered(**this**)){  
 EventBus.*getDefault*().register(**this**);  
 }  
 **recyclerview1**=view.findViewById(R.id.***rv***);  
 LinearLayoutManager layoutManager = **new** LinearLayoutManager(getActivity());  
 layoutManager.setOrientation(LinearLayoutManager.***VERTICAL***);  
 **recyclerview1**.setLayoutManager(layoutManager);  
 getHttpData(*k*);  
  
  
 **refreshLayout** =view.findViewById(R.id.***refreshLayout1***);  
 **refreshLayout**.setOnRefreshListener(**new** OnRefreshListener() {  
 @Override  
 **public void** onRefresh(RefreshLayout refreshlayout) {  
 **xinxi**.clear();  
 getHttpData(*k*);  
 *//adapters.notifyDataSetChanged();* refreshlayout.finishRefresh();  
 }  
 });  
 *//加载更多* **refreshLayout**.setOnLoadmoreListener(**new** OnLoadmoreListener() {  
 @Override  
 **public void** onLoadmore(RefreshLayout refreshlayout) {  
 *k*=*k*+10;  
 getHttpData1(*k*);  
 *//adapters.notifyDataSetChanged();  
 //adapters.notifyItemMoved(k,k+10);* refreshlayout.finishLoadmore();  
 *//recyclerview1.scrollToPosition(k+10);* }  
  
  
 });  
  
 **refreshLayout**.setEnableRefresh(**true**);*//启用刷新* **refreshLayout**.setEnableLoadmore(**true**);*//启用加载* **refreshLayout**.finishRefresh();  
 **refreshLayout**.finishLoadmore();  
 **return** view;  
}  
List<New> **xinxi**=**new** ArrayList<>();  
**public void** getHttpData(**int** key){  
  
 OkHttpClient client=**new** OkHttpClient();  
 Request request = **new** Request.Builder()  
 .url(**"https://api.jisuapi.com/news/get?channel=头条&start=0&num=10&appkey=527bd20c4cc928e3"**).build();  
  
  
 client.newCall(request).enqueue(**new** Callback(){  
  
 @Override  
 **public void** onFailure(Call call, IOException e) {  
 System.***out***.print(**"rrr"**);  
 }  
  
 @Override  
 **public void** onResponse(Call call, Response response) **throws** IOException {  
 **if**(!response.isSuccessful())**throw new** IOException(**"Unexpected code"**+response);  
  
 **try** {  
 JSONObject responseJson=**new** JSONObject(response.body().string());  
 System.***out***.print(responseJson);  
 JSONObject result=responseJson.getJSONObject(**"result"**);  
 JSONArray array=result.getJSONArray(**"list"**);  
 **for**(**int** i=0;i<array.length();i++)  
 {  
 JSONObject json=array.getJSONObject(i);  
 New xw = **new** New();  
 xw.**title**=json.getString(**"title"**);  
 xw.**time**=json.getString(**"time"**);  
 xw.**src**=json.getString(**"src"**);  
 xw.**pic**=json.getString(**"pic"**);  
 xw.**weburl**=json.getString(**"weburl"**);  
 **xinxi**.add(xw);  
 }  
 EventBus.*getDefault*().post(**new** MessageEvent(0));  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
}

**新闻内容的实现：**

新闻内容来源自极速数据，新闻内容每天免费100条，刷新一次10条

调用方式：

Request request = **new** Request.Builder()  
 .url(**"https://api.jisuapi.com/news/get?channel=头条&start=0&num=10&appkey=527bd20c4cc928e3"**).build();

实现代码块：

图形用户界面, 文本, 应用程序

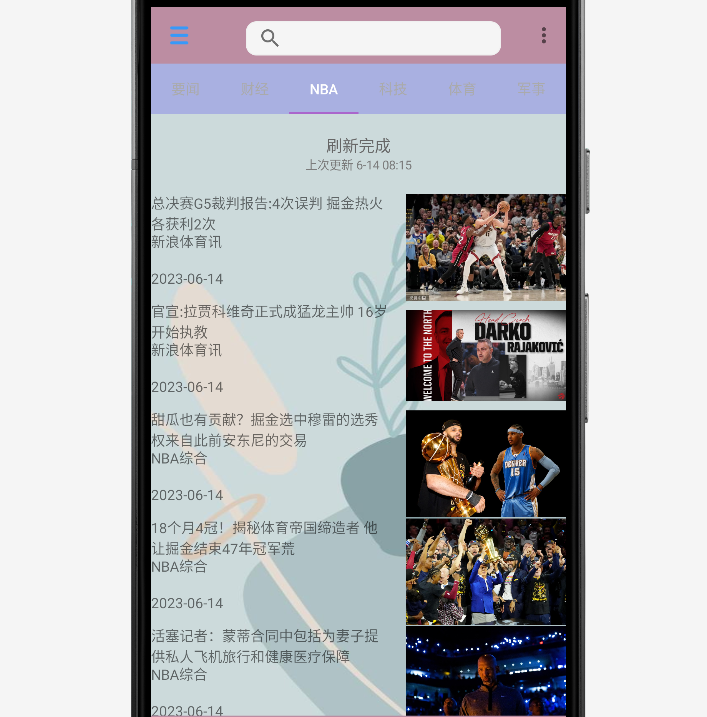
描述已自动生成

**public void** onFailure(Call call, IOException e) {  
 System.***out***.print(**"rrr"**);  
 }  
  
 @Override  
 **public void** onResponse(Call call, Response response) **throws** IOException {  
 **if**(!response.isSuccessful())**throw new** IOException(**"Unexpected code"**+response);  
  
 **try** {  
 JSONObject responseJson=**new** JSONObject(response.body().string());  
 System.***out***.print(responseJson);  
 JSONObject result=responseJson.getJSONObject(**"result"**);  
 JSONArray array=result.getJSONArray(**"list"**);  
 **for**(**int** i=0;i<array.length();i++)  
 {  
 JSONObject json=array.getJSONObject(i);  
 New xw = **new** New();  
 xw.**title**=json.getString(**"title"**);  
 xw.**time**=json.getString(**"time"**);  
 xw.**src**=json.getString(**"src"**);  
 xw.**pic**=json.getString(**"pic"**);  
 xw.**weburl**=json.getString(**"weburl"**);  
 **xinxi**.add(xw);  
 }  
 EventBus.*getDefault*().post(**new** MessageEvent(0));  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
}

**新闻刷新功能实现：**

刷新获取代码块，刷新代码块主要是再次调用新闻显示的函数，当刷新时候重新执行函数，从而显示出新的新闻内容， 实现新闻的刷新。

实现图:



实现代码：从写函数，实现刷新

**private void** getHttpData1(**int** k) {  
 OkHttpClient client=**new** OkHttpClient();  
 Request request = **new** Request.Builder()  
 .url(**"https://api.jisuapi.com/news/get?channel=头条&start=0&num=10&appkey=527bd20c4cc928e3"**).build();  
  
 client.newCall(request).enqueue(**new** Callback(){  
  
 @Override  
 **public void** onFailure(Call call, IOException e) {  
 System.***out***.print(**"rrr"**);  
 }  
  
 @Override  
 **public void** onResponse(Call call, Response response) **throws** IOException {  
 **if**(!response.isSuccessful())**throw new** IOException(**"Unexpected code"**+response);  
  
 **try** {  
 JSONObject responseJson=**new** JSONObject(response.body().string());  
 System.***out***.print(responseJson);  
 JSONObject result=responseJson.getJSONObject(**"result"**);  
 JSONArray array=result.getJSONArray(**"list"**);  
 **for**(**int** i=0;i<array.length();i++)  
 {  
 JSONObject json=array.getJSONObject(i);  
 New xw = **new** New();  
 xw.**title**=json.getString(**"title"**);  
 xw.**time**=json.getString(**"time"**);  
 xw.**src**=json.getString(**"src"**);  
 xw.**pic**=json.getString(**"pic"**);  
 xw.**weburl**=json.getString(**"weburl"**);  
 **xinxi**.add(xw);  
 }  
 EventBus.*getDefault*().post(**new** MessageEvent(1));  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });

**新闻详情实现：**

文本

描述已自动生成

**实现代码块**

总体数据代码**：**

**protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_webnewxqq***);  
 **webView** = (WebView)findViewById(R.id.***webView***);  
 **button\_right2**=(Button)findViewById(R.id.***button\_right2***);  
  
  
 Intent intent=getIntent();   
 String name=intent.getStringExtra(**"test"**);

init();  
 **webView**.loadUrl(name);  
}  
  
**void** init(){  
 WebSettings webSettings = **webView**.getSettings();  
 webSettings.setJavaScriptEnabled(**true**);  
 **webView**.setWebViewClient(**new** MyWebViewClient());  
  
 webSettings.setUseWideViewPort(**true**); *//将图片调整到适合webview的大小* webSettings.setLoadWithOverviewMode(**true**); *// 缩放至屏幕的大小* webSettings.setSupportZoom(**true**); *//支持缩放，默认为true。是下面那个的前提。* webSettings.setBuiltInZoomControls(**true**); *//设置内置的缩放控件。若为false，则该WebView不可缩放* webSettings.setDisplayZoomControls(**false**); *//隐藏原生的缩放控件  
  
 //其他细节操作* webSettings.setAllowFileAccess(**true**); *//设置可以访问文件* webSettings.setJavaScriptCanOpenWindowsAutomatically(**true**); *//支持通过JS打开新窗口* webSettings.setLoadsImagesAutomatically(**true**); *//支持自动加载图片* webSettings.setDefaultTextEncodingName(**"utf-8"**);*//设置编码格式*}  
  
**class** MyWebViewClient **extends** WebViewClient {  
 @TargetApi(21)  
 @Override  
 **public boolean** shouldOverrideUrlLoading(WebView view, WebResourceRequest request) {  
 String url = request.getUrl().toString();  
 **try**{  
 **if**(url.startsWith(**"baiduboxapp://"**)||url.startsWith(**"baiduboxlite://"**)){  
 Intent intent = **new** Intent(Intent.***ACTION\_VIEW***, Uri.*parse*(url));  
 startActivity(intent);  
 **return true**;  
 }  
 }**catch**(Exception e){  
 **return true**;  
 }  
 **return super**.shouldOverrideUrlLoading(view, request);  
 }   
}

**关于功能：**就当个页面，版本信息

图形用户界面, 应用程序

描述已自动生成

**登录注册：**

登录注册通过点击侧边栏头像，实现登录和注册，登录和注册采用数据库的形式实现。

数据库为=调用类SQLiteOpenHelper

登录实现代码：

**EtStuNumber** = findViewById(R.id.***et\_username***);  
**EtStuPwd** = findViewById(R.id.***et\_password***);  
Button btnLogin = findViewById(R.id.***btn\_login***);  
btnLogin.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **boolean** flag = **false**;  
 **if**(CheckInput()) {  
 UserDbHelper dbHelper = **new** UserDbHelper(getApplicationContext(),UserDbHelper.***DB\_NAME***,**null**,1);  
 **users** = dbHelper.readUsers();  
 **for**(User user : **users**) {  
 *//如果可以找到,则输出登录成功,并跳转到主界面* **if**(user.getUsername().equals(**EtStuNumber**.getText().toString()) && user.getPassword().equals(**EtStuPwd**.getText().toString()) ) {  
 flag = **true**;  
 Toast.*makeText*(login.**this**,**"恭喜你登录成功!"**,Toast.***LENGTH\_SHORT***).show();  
 Intent intent = **new** Intent(login.**this**,MainActivity.**class**);  
 Bundle bundle = **new** Bundle();  
 **username** = **EtStuNumber**.getText().toString();  
 bundle.putString(**"username"**,**username**);  
 intent.putExtras(bundle);  
 startActivity(intent);  
 }  
 }  
 *//否则提示登录失败,需要重新输入* **if** (!flag) {  
 Toast.*makeText*(login.**this**,**"学号或密码输入错误!"**,Toast.***LENGTH\_SHORT***).show();

注册实现代码：

btnCancel.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 finish();  
 }  
 });  
 **tvStuNumber** = findViewById(R.id.***et\_username***);  
 **tvStuPwd** = findViewById(R.id.***et\_password***);  
 **tvStuConfirmPwd** = findViewById(R.id.***et\_confirm\_password***);  
 *//注册点击事件* Button btnRegister = findViewById(R.id.***btn\_register***);  
 btnRegister.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//首先确保不为空* **if**(CheckInput()) {  
 User user = **new** User();  
 user.setUsername(**tvStuNumber**.getText().toString());  
 user.setPassword(**tvStuPwd**.getText().toString());  
 UserDbHelper dbHelper = **new** UserDbHelper(getApplicationContext(),UserDbHelper.***DB\_NAME***,**null**,1);  
 dbHelper.addUser(user);  
 Toast.*makeText*(register.**this**,**"恭喜你注册成功!"**,Toast.***LENGTH\_SHORT***).show();  
 *//销毁当前界面* finish();  
 }  
 }  
 });  
}

界面图：

登录：

图形用户界面, 应用程序

描述已自动生成

注册：



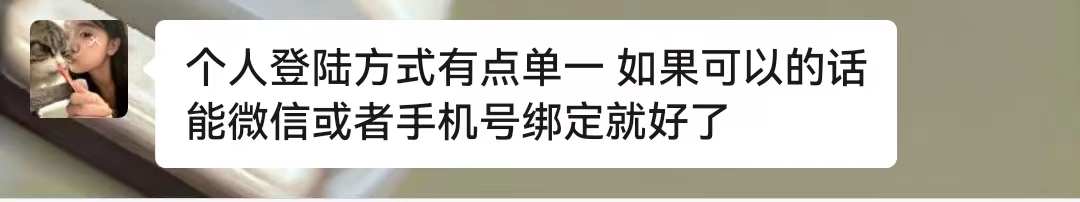
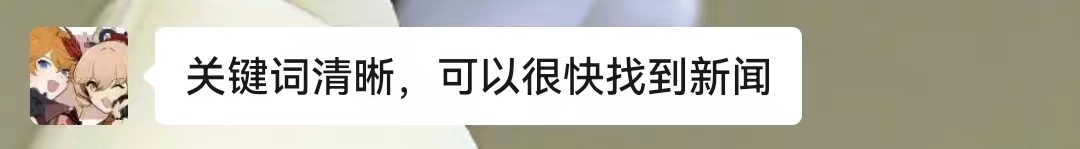
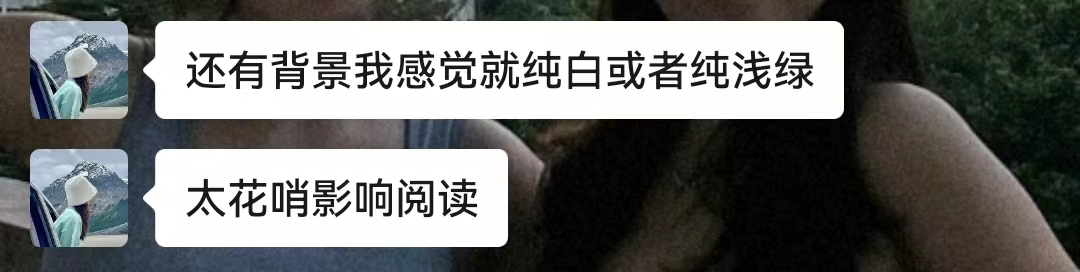
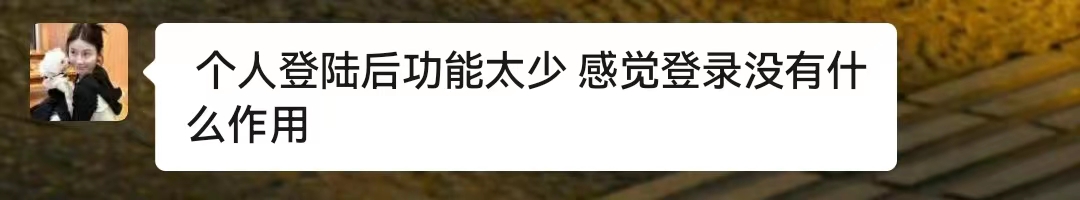
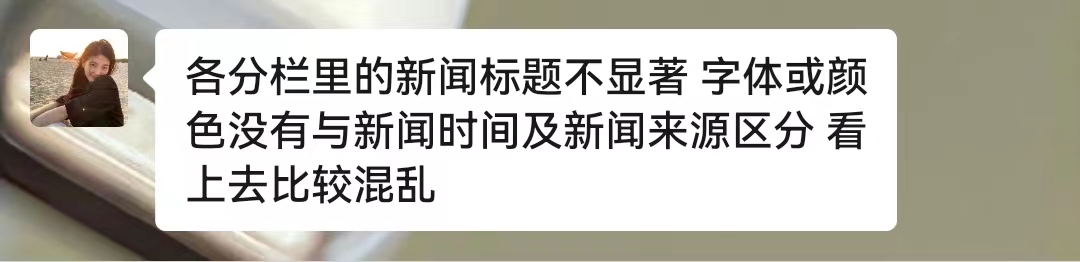
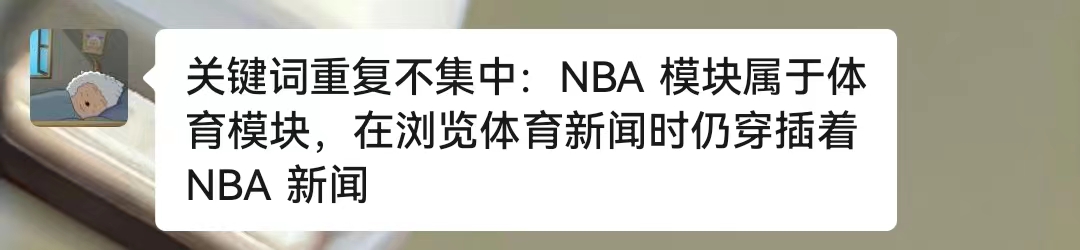
## 2.关键技术以及难点 关键技术：json数据解析+api接口调用

**public void** getHttpData(**int** key){  
  
 OkHttpClient client=**new** OkHttpClient();  
 Request request = **new** Request.Builder()  
 .url(**"https://api.jisuapi.com/news/get?channel=头条&start=0&num=10&appkey=3be800fc1a8c39d9"**).build();  
 client.newCall(request).enqueue(**new** Callback(){  
 @Override  
 **public void** onFailure(Call call, IOException e) {  
 System.***out***.print(**"rrr"**);  
 }  
  
 @Override  
 **public void** onResponse(Call call, Response response) **throws** IOException {  
 **if**(!response.isSuccessful())**throw new** IOException(**"Unexpected code"**+response);  
  
 **try** {  
 JSONObject responseJson=**new** JSONObject(response.body().string());  
 System.***out***.print(responseJson);  
 JSONObject result=responseJson.getJSONObject(**"result"**);  
 JSONArray array=result.getJSONArray(**"list"**);  
 **for**(**int** i=0;i<array.length();i++)  
 {  
 JSONObject json=array.getJSONObject(i);  
 New xw = **new** New();  
 xw.**title**=json.getString(**"title"**);  
 xw.**time**=json.getString(**"time"**);  
 xw.**src**=json.getString(**"src"**);  
 xw.**pic**=json.getString(**"pic"**);  
 xw.**weburl**=json.getString(**"weburl"**);  
 **xinxi**.add(xw);  
 }  
 EventBus.*getDefault*().post(**new** MessageEvent(0));  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
}  
**private void** getHttpData1(**int** k) {  
 OkHttpClient client=**new** OkHttpClient();  
 Request request = **new** Request.Builder()  
 .url(**"https://api.jisuapi.com/news/get?channel=头条&start=0&num=10&appkey=3be800fc1a8c39d9"**).build();  
  
 client.newCall(request).enqueue(**new** Callback(){  
  
 @Override  
 **public void** onFailure(Call call, IOException e) {  
 System.***out***.print(**"rrr"**);  
 }  
  
 @Override  
 **public void** onResponse(Call call, Response response) **throws** IOException {  
 **if**(!response.isSuccessful())**throw new** IOException(**"Unexpected code"**+response);  
  
 **try** {  
 JSONObject responseJson=**new** JSONObject(response.body().string());  
 System.***out***.print(responseJson);  
 JSONObject result=responseJson.getJSONObject(**"result"**);  
 JSONArray array=result.getJSONArray(**"list"**);  
 **for**(**int** i=0;i<array.length();i++)  
 {  
 JSONObject json=array.getJSONObject(i);  
 New xw = **new** New();  
 xw.**title**=json.getString(**"title"**);  
 xw.**time**=json.getString(**"time"**);  
 xw.**src**=json.getString(**"src"**);  
 xw.**pic**=json.getString(**"pic"**);  
 xw.**weburl**=json.getString(**"weburl"**);  
 **xinxi**.add(xw);  
 }  
 EventBus.*getDefault*().post(**new** MessageEvent(1));  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
  
}

# <三> 用户体验记录和改进方案

## 1.用户体验和建议

本环节进行在项目基本完成的阶段，分别请身边同学使用该APP，并请他们分享使用体验和修改建议，交流部分通过微信完成，此文档中列出了一些较为实用的建议，如下图所示。

****

总结：

1.新闻分栏：标题不显著，与新闻时间等字体没有区分；新闻缩略图格式不一，导致排版看上去很乱

2.登录方面：个人登录后功能太少，且登录方式单一

3.app背景选择：建议纯色，太过花哨会影响阅读体

## 2.已完成的改进和存在的问题

**问题一：新闻分栏:标题不显著**

改进方案：

* 增加标题字体大小和粗细，突出标题重要性；
* 根据新闻发布时间远近进行区分，时间字体可以更小一些，避免干扰标题的视觉效果；
* 统一缩略图大小和相对位置，避免过多且不必要的空白；
* 确定一套统一的排版规范，并采用响应式布局适应不同屏幕的大小和分辨率；
* 利用视觉层次感设计区分新闻栏目之间的结构，如不同的颜色和分隔线；
* 定期进行用户体验测试，根据用户反馈进行调整和优化。

**问题二：登录方面:个人登录后功能太少，且登录方式单一**

改进方案：

* 增加第三方登录方式，例如使用微信、QQ、Apple账号等登录，提高用户登录的便利性和多样性；
* 建议增加“记住密码”功能，方便用户下次访问时自动登录；
* 根据用户登录后的需求，提供更多的个性化功能，例如用户偏好设置、上传头像等；
* 对于需要更高安全性的用户，可以增加双重验证（例如手机验证码、密保问题等）来保障安全性；
* 加入社交化登录功能，用户可以通过和朋友分享来获得更多的功能（例如优惠券、积分等）

**问题三：APP背景选择：太过花哨影响阅读体验**

改进方案：

建议选择比较简洁的纯色背景，可以让用户更专注于内容的阅读，减少干扰和视觉疲劳。同时，选择合适的字体颜色和大小，以确保文字清晰易读。

**问题四：在进行标准兼容测试时，有45%的测试设备出现APP** **Crash问题**

改进方案：

* 编写高质量的代码和健壮的程序逻辑，提高应用程序的稳定性；
* 对网络请求进行错误处理，合理优化应用程序与服务器之间的通信；
* 合理设置异常捕获并且在崩溃时记录错误日志，以便于后续的调试和分析。

# **<四> 测试大纲和测试报告**

**1.UI测试**

* + 1. **用户界面布局**

1. 一致性

用户界面的各个部分保持一致性，包括颜色、字体、按钮样式、布局等。确保了整个应用在不同页面、不同功能模块中的一致性。



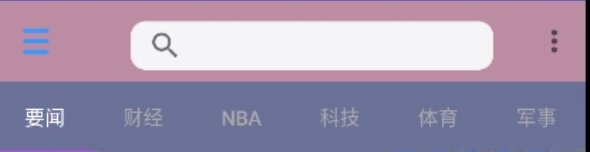
1. 可视化和布局

应用的可视化效果和布局清晰、易于理解。各个元素是否对齐、层次结构是否清晰。用户可以清楚地了解应用界面的布局和视觉层次。

* + 1. **用户友好性**

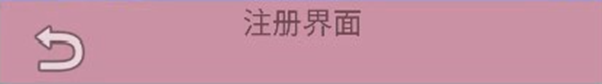
1. 导航和结构

应用的导航和结构，包括菜单、标签、按钮的位置和功能，确保用户可以轻松地找到所需的功能，并且导航流畅。



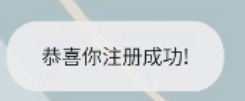
1. 文字和标签

应用中的文字和标签的准确性、清晰度和一致性较好，文字信息易于理解，没有语法错误或拼写错误。按钮、标签和其他交互元素的文本与其功能相符。



1. 反馈与响应

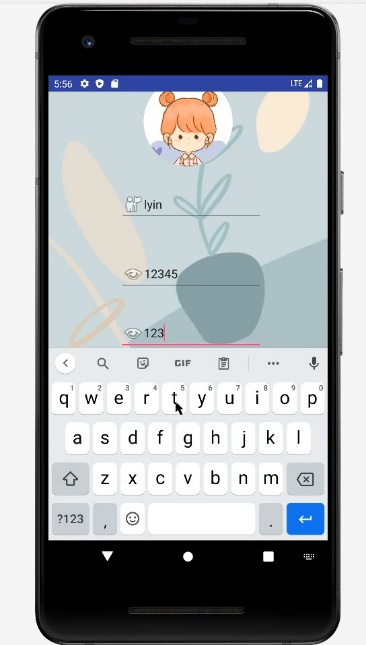
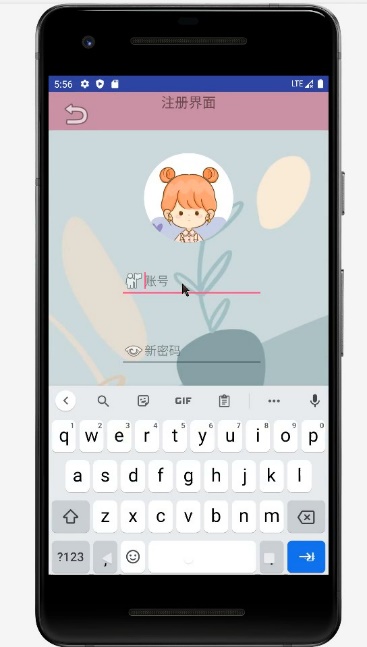
当用户执行某个操作时，应用能够提供明确的反馈，应用对用户操作做出及时响应，避免让用户感到迷惑或不知道他们的操作是否生效。



* 1. **功能测试**
     1. **登录、注册**

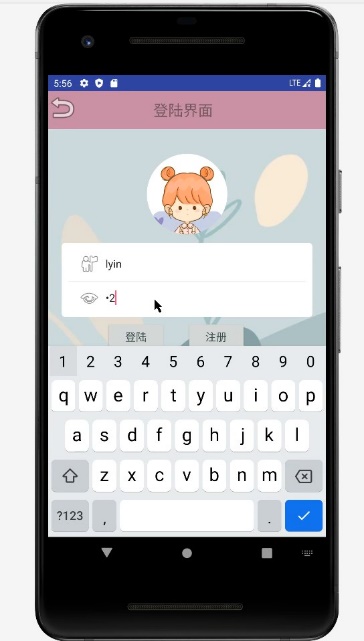
1. 注册

用户注册需要用户填写一些必要的信息：账号、密码、确认密码。



1. 登录

用户登录则需要用户输入正确的用户名和密码，系统根据输入验证用户身份，并将其与用户数据进行匹配。



* + 1. **新闻页面**

1. 主页

主页有要闻、财经、NBA、科技、体育和军事6个频道。



1. 刷新

下拉可以刷新新闻。



1. 新闻页详情

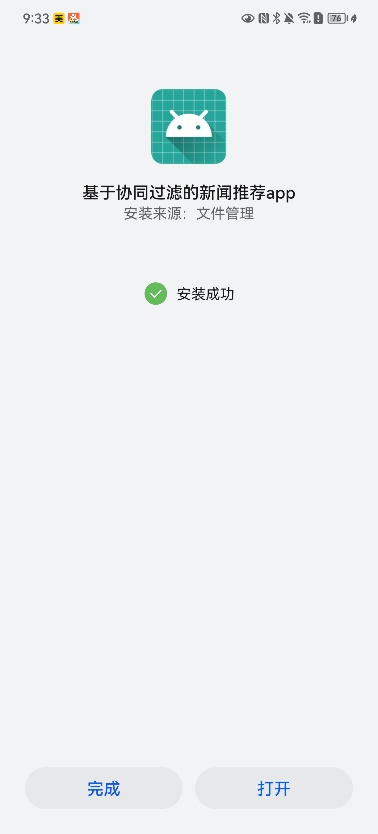
点击一条新闻可以进入详情页面。



* + 1. **安装和卸载**

可以正常进行卸载和安装。

1. 安装



1. 卸载



* 1. **标准兼容测试**

使用腾讯Wetest平台自动化测试中的标准兼容测试。

* + 1. **测试机型**

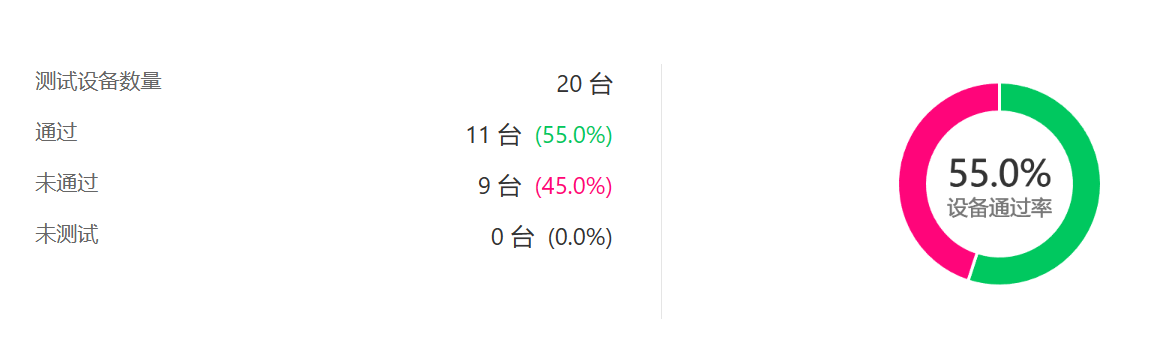
测试机型数：随机20台

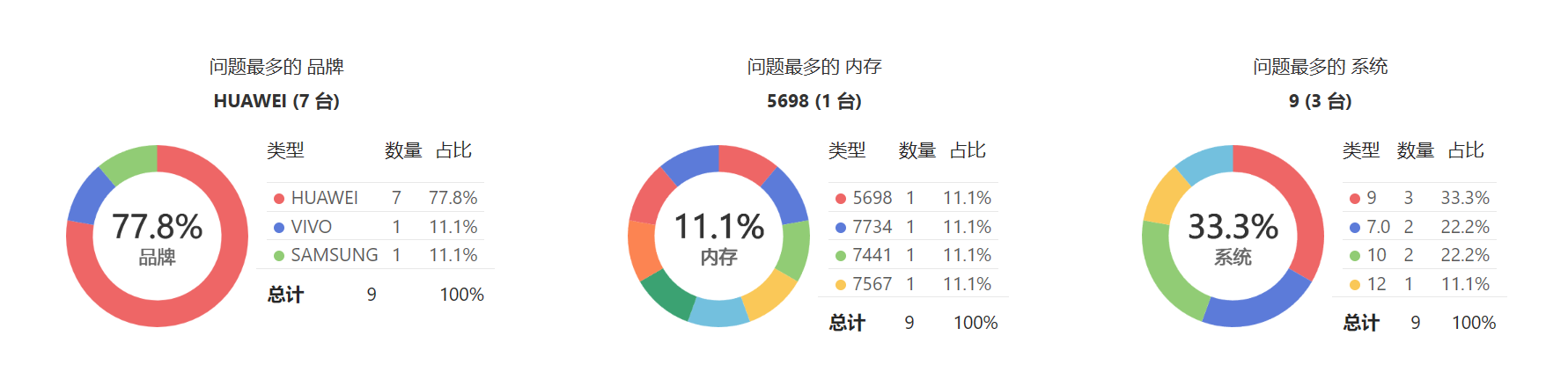
测试机型分布：

|  |  |  |
| --- | --- | --- |
| 设备品牌 | 设备别名 | 设备型号 |
| GOOGLE | Pixel 2XL | Pixel 2 XL |
| HUAWEI | 荣耀30s | CDY-AN90 |
| HUAWEI | 华为P40 | ANA-AN00 |
| HUAWEI | 荣耀X10 | TEL-AN00a |
| HUAWEI | 为畅享9 Plus | JKM-AL00a |
| HUAWEI | 畅玩6X | BLN-AL30 |
| HUAWEI | 华为 nova 3 | PAR-AL00 |
| HUAWEI | 华为 Mate 20 Pro | LYA-AL00 |
| HUAWEI | 荣耀V9 | DUK-AL20 |
| HUAWEI | 荣耀Play | COR-AL10 |
| HUAWEI | 荣耀20S | YAL-AL50 |
| HUAWEI | 畅享9 | DUB-AL00a |
| HUAWEI | 华为畅享9 | DUB-AL00 |
| OPPO | OPPO R15 | PACM00 |
| samsung | 三星Galaxy S20+ 5G | SM-G9860 |
| samsung | 三星 Galaxy S9+ | SM-G9650 |
| SAMSUNG | 三星GALAXY S6 Edge+ | SM-G9280 |
| VIVO | S15 Pro | V2207A |
| Xiaomi | Redmi 9 | M2004J19C |
| XIAOMI | 红米Pro | Redmi Pro |

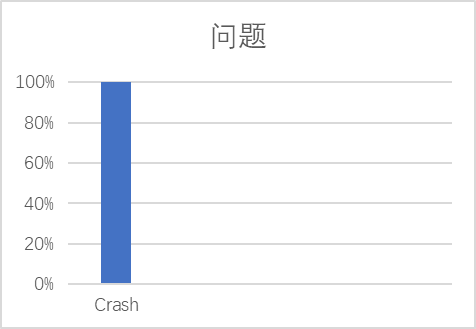
* + 1. **测试结论**

测试通过率为55%，影响用户的问题有APP Crash，建议关注APP Crash。





* + 1. **问题分布**



|  |  |  |  |
| --- | --- | --- | --- |
| 问题列表 | | | |
| 问题类别 | 问题描述 | 涵盖设备 | 操作 |
| Crash | android.content.ActivityNotFoundException(Instrumentation.java:1850) | 1台 | BLN-AL30/畅玩6X | https://wetest.qq.com/n/console/default-ci5do59u1957001f4etg/testlab-function/standard-compatible/report/2023061500251817/problem/113?errorid=367 |
| Crash | android.content.ActivityNotFoundException(Instrumentation.java:1977) | 1台 | DUK-AL20/荣耀V9 | https://wetest.qq.com/n/console/default-ci5do59u1957001f4etg/testlab-function/standard-compatible/report/2023061500251817/problem/113?errorid=368 |
| Crash | android.content.ActivityNotFoundException(Instrumentation.java:2086) | 1台 | YAL-AL50/荣耀20S | https://wetest.qq.com/n/console/default-ci5do59u1957001f4etg/testlab-function/standard-compatible/report/2023061500251817/problem/113?errorid=365 |
| Crash | android.content.ActivityNotFoundException(Instrumentation.java:2112) | 2台 | JKM-AL00a/为畅享9 Plus等 | https://wetest.qq.com/n/console/default-ci5do59u1957001f4etg/testlab-function/standard-compatible/report/2023061500251817/problem/113?errorid=370 |
| Crash | android.content.ActivityNotFoundException(Instrumentation.java:2118) | 2台 | CDY-AN90/荣耀30s等 | https://wetest.qq.com/n/console/default-ci5do59u1957001f4etg/testlab-function/standard-compatible/report/2023061500251817/problem/113?errorid=369 |
| Crash | android.content.ActivityNotFoundException(Instrumentation.java:2135) | 1台 | V2207A/S15 Pro | https://wetest.qq.com/n/console/default-ci5do59u1957001f4etg/testlab-function/standard-compatible/report/2023061500251817/problem/113?errorid=371 |
| Crash | java.lang.RuntimeException(DisplayListCanvas.java:260) | 1台 | SM-G9280/三星GALAXY S6 Edge+ | https://wetest.qq.com/n/console/default-ci5do59u1957001f4etg/testlab-function/standard-compatible/report/2023061500251817/problem/113?errorid=372 |

* + 1. **问题机型**

问题最多的品牌：华为(77.8%)

问题最多的系统版本：9(33.33%)

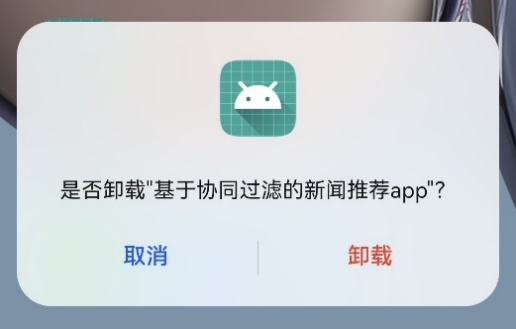
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 设备品牌 | 设备型号 | 设备别名 | 系统版本 | 内存(MB) | 分辨率 | CPU核数 | CPU名称 | CPU频率 | CPU架构 |
| VIVO | V2207A | S15 Pro | 12 | 7680 | 1080x2376 | 8 |  | 2.85 | arm64-v8a |
| SAMSUNG | SM-G9280 | 三星GALAXY S6 Edge+ | 7.0 | 4096 | 1440x2560 | 8 | 三星 Exynos 7420 | 2.1 | arm64-v8a |
| HUAWEI | CDY-AN90 | 荣耀30s | 10 | 7680 | 1080x2400 | 8 | 麒麟 海思820 | 1.9 | AArch64 Processor rev 0 (aarch64) |
| HUAWEI | ANA-AN00 | 华为P40 | 10 | 5632 | 1080x2340 | 8 | 海思 麒麟990 | 2 | AArch64 Processor rev 1 (aarch64) |
| HUAWEI | JKM-AL00a | 华为畅享9 Plus | 9 | 4096 | 1080x2340 | 8 | 海思 麒麟710 | 1.8 | arm64-v8a |
| HUAWEI | BLN-AL30 | 畅玩6X | 7.0 | 3072 | 1080x1920 | 8 | 海思 麒麟655 | 2.112 | arm64-v8a |
| HUAWEI | DUK-AL20 | 荣耀V9 | 8.0.0 | 6144 | 1440x2560 | 8 | 海思 麒麟960 | 1.9 | AArch64 Processor rev 1 (aarch64) |
| HUAWEI | COR-AL10 | 荣耀Play | 9 | 6144 | 1080x2340 | 8 | 海思 麒麟970 | 1.9 | AArch64 Processor rev 2 (aarch64) |
| HUAWEI | YAL-AL50 | 荣耀20S | 9 | 8192 | 1080x2340 | 8 | 海思 麒麟810 | 1.9 | AArch64 Processor rev 0 (aarch64) |

* + 1. **性能分析**

1. 安装耗时
2. 启动耗时
3. CPU占用百分比
4. 内存占用
5. FPS
6. 产品安装和使用说明







# **<五> 团队分工**

组长李虹：主要负责各个阶段的工作部署和进程推进，完成APP的设置部分相关功能，完成UI设计的部分功能，产品定位及目标和部分技术解决方案的撰写。

组员田野：完成APP的登录注册板块的代码设计，以及相关界面呈现设计，项目实施可行性报告的撰写。

组员梁林英：完成APP的个人—我的模块个人功能的相关代码设计及界面设计，推广方案和运营计划书的撰写。

组员刘晓艺：负责首页界面呈现的代码设计及分栏新闻的实现，产品内容总策划的撰写，以及测试报告的完成。