[项目LOGO]



[项目名称]

**[Project Name]**

项目测试文档

**Version [Number]**

**[基于wifi定位的商城导购系统]**

[YYYY.MM.DD]

**Written by [我们做的都对]**

[Team LOGO]



**All Rights Reserved**

目录

[1 测试计划 1](#_Toc45267059)

[1.1 测试策略与目标 1](#_Toc45267060)

[1.2 测试范围 1](#_Toc45267061)

[1.3 测试环境 1](#_Toc45267062)

[2 功能测试 1](#_Toc45267063)

[2.1 \*\*\*\*功能 1](#_Toc45267064)

[2.1.1 测试用例与执行分析 1](#_Toc45267065)

[测试用例： 1](#_Toc45267066)

[分析结果： 1](#_Toc45267067)

[2.1.2 测试结果综合分析及建议 2](#_Toc45267068)

[2.1.3 测试经验总结 2](#_Toc45267069)

[3 系统测试 2](#_Toc45267070)

[3.1 定位精度 2](#_Toc45267071)

[3.1.1 室内环境（形状、面积、设备、采样点、测试点） 2](#_Toc45267072)

[3.1.2 定位精度（CDF曲线） 2](#_Toc45267073)

[3.1.3 定位结果分析总结 2](#_Toc45267074)

[3.2 房间识别精度 2](#_Toc45267075)

[3.2.1 评价指标 2](#_Toc45267076)

[3.2.2 结果分析 3](#_Toc45267077)

[3.3 APP性能 3](#_Toc45267078)

[3.3.1 延时 3](#_Toc45267079)

[3.3.2 存储 3](#_Toc45267080)

[3.3.3 功耗 3](#_Toc45267081)

[4 测试结果总结及优缺点分析 3](#_Toc45267082)

# 测试计划

## 测试策略与目标

**测试重点识别：**

功能测试：验证教学楼三层内所有功能区域的使用功能是否按设计要求。

性能测试：评估系统在高负载情况下的响应时间和稳定性，如同时容纳大量学生上课或活动的场景，我们选取很多人的情况

兼容性测试：确保系统在不同设备（如不同品牌电脑、手机等）和浏览器上的兼容性。

**测试难点分析：**

复杂场景模拟：如何模拟教学楼内复杂的日常使用和应急场景进行测试。

系统集成测试：与教学楼其他楼层或校园管理系统的集成测试，确保数据一致性和流程顺畅。

性能瓶颈定位：在高并发情况下，快速定位并优化性能瓶颈。

**测试分层：**

单元测试：针对各个功能模块进行独立测试，确保每个单元功能正确。

集成测试：将多个模块组合在一起进行测试，验证模块间的接口和数据交互无误。

系统测试：对整个教学楼三层系统进行全面测试，包括功能、性能、安全等方面。

验收测试：邀请用户参与测试，根据用户反馈进行最后的调整和优化。

**目标**

我们的测试目标是定义YF教学楼三层项目在发布时的质量等级，确保系统：

功能完备：所有既定功能均已实现并稳定运行。

性能优越：在高负载情况下仍能保持良好的响应速度和稳定性。

安全可靠：能够有效抵御各类安全威胁，保护用户数据安全。

用户友好：界面设计直观易用，符合用户操作习惯。

易于维护：系统架构清晰，便于后续维护和升级。

## 测试范围

### 测试广度

测试范围覆盖YF教学楼三层的整体系统，包括但不限于以下几个方面：

物理环境：包括教室、走廊、楼梯间等所有区域的布局和设施。

硬件设备：教学设备（如投影仪、电脑）、个人设备（手机）等。

### 测试深度

在测试深度上，我们将对以下几个方面进行深入测试：

功能深度：不仅验证基本功能，还测试各种边界条件和异常情况的处理。

性能压力：通过模拟不同场景下的用户行为，测试系统的负载能力和稳定性。

用户体验：通过用户测试和问卷调查，收集用户反馈，优化界面设计和操作流程。

## 测试环境

硬件环境：联想电脑一部，华为弹性云服务器，Ubunto20.0

网络环境：逸夫楼校园网

测试工具：华为两部，真我手机一部，小米手机，联想电脑一台，测试相关的app

# 功能测试

## 登录注册功能

### 测试用例与执行分析

### 测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 001 | | | |
| 功能描述 | 用户可以进行登录注册功能 | | | |
| 用例目的 | 实现登录注册 | | | |
| 前提条件 | 用户点击应用 | | | |
| 特殊的规程说明 |  | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 动作 | 期望结果 | 实际结果 | 备注 |
| 1 | 点击注册 | 成功注册 | 成功注册 |  |
| 2 | 点击登录 | 成功登录 | 成功登录 |  |
| … | … | … | … | … |

### 分析结果：登录注册功能实现成功，我们的指纹识别运行有效，验证登录登跳转功能成功实现

### 测试结果综合分析及建议

在针对用户登录注册功能的测试过程中，我们重点关注了用户体验和数据安全性两大方面。通过一系列测试，我们得出了以下综合分析及建议：

****用户体验方面****：

****快速登录功能****：记住密码和指纹识别等功能的实现，极大地提升了用户的登录效率。用户无需每次登录都输入复杂的密码，特别是在移动设备或高频使用的场景下，这种便利性尤为显著。建议继续优化这些功能，确保其在不同设备和网络环境下的稳定性和兼容性。

****界面友好性****：登录注册页面的布局合理，提示信息清晰明了，有助于用户快速理解并完成操作。建议根据用户反馈，持续优化界面设计，比如增加动画效果、优化按钮布局等，以进一步提升用户体验。

****数据安全性方面****：

****1唯一性验证****：在测试过程中，我们验证了用户名关键信息的唯一性，有效防止了数据重复和潜在的安全风险。这一措施对于保护用户数据安全至关重要。建议继续加强这方面的验证机制，如增加手机号验证、验证码验证等，以进一步提高账户安全性。

****2.密码策略****：虽然测试中没有直接涉及密码策略的验证，但建议系统实施强密码策略，如要求密码包含大小写字母、数字和特殊字符，并设定最小长度。同时，定期提示用户更改密码，避免密码泄露带来的风险。

****建议****：

****多因素认证****：考虑引入多因素认证机制（如短信验证码、邮箱验证、二次密码等），以增加账户的安全性。

****用户教育****：通过用户指南、安全提示等方式，提高用户对账户安全的认识和重视程度，引导用户采取正确的安全措施。

****持续监控与响应****：建立账户安全监控体系，及时发现并处理异常登录、密码泄露等安全问题。同时，建立快速响应机制，确保在发生安全事件时能够迅速采取措施，减少损失。

### 测试经验总结

**需求理解与分析的重要性**：在测试前，深入理解用户需求是制定有效测试策略的前提。通过需求分析，我们可以明确测试的重点和难点，从而有针对性地设计测试用例。

**自动化测试的应用**：对于重复性的测试任务，如密码强度验证、唯一性检查等，可以考虑引入自动化测试工具，以提高测试效率和准确性。

**用户体验与数据安全的平衡**：在测试过程中，我们需要同时关注用户体验和数据安全两个方面。既要确保用户能够快速、便捷地完成操作，又要保证用户数据的安全性和隐私性。

**持续改进与优化**：测试不是一次性的工作，而是一个持续的过程。我们需要根据测试结果和用户反馈，不断优化产品功能和测试策略，以提升产品的竞争力和用户满意度。

## 主页面功能

### 测试用例与执行分析

### 测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 002 | | | |
| 功能描述 | 用户可以在主页面跳转页面导航功能 | | | |
| 用例目的 | 个性化服务 | | | |
| 前提条件 | 用户成功登录注册 | | | |
| 特殊的规程说明 |  | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 动作 | 期望结果 | 实际结果 | 备注 |
| 1 | 点击商家 | 成功跳转 | 成功跳转 |  |
| 2 | 点击导航 | 成功导航 | 成功导航 |  |
| 3 | 点击评论 | 成功评论 | 成功评论 | … |

### 分析结果：主界面相关功能实现成功

### 测试结果综合分析及建议

通过对用例002的执行和分析，我们可以看到主界面上的主要功能——页面跳转、导航和评论均运行正常，这表明系统在用户交互方面表现良好。同时，系统响应速度快，界面设计合理，为用户提供了良好的使用体验。

****建议****:

****持续优化用户体验****: 尽管当前功能实现良好，但可以考虑进一步优化用户界面，提升操作的便捷性和美观性。例如，优化导航菜单的布局，使其更加直观易用。

****增加测试场景****: 随着系统的不断发展，建议增加更多复杂的测试场景，如多用户并发操作、弱网络环境下的性能测试等，以全面评估系统的稳定性和可靠性。

****关注用户反馈****: 密切关注用户的使用反馈，特别是关于界面设计和功能操作方面的意见，以便及时发现问题并进行改进。

### 测试经验总结

****细致规划测试场景****: 在进行测试前，需要细致规划测试场景，确保覆盖所有关键功能和用户路径。这有助于发现潜在的问题，并提前进行修复。

****注重用户体验测试****: 用户体验是评估系统质量的重要指标之一。在测试过程中，应特别关注用户界面的友好性、操作的便捷性以及系统的响应速度等方面。

****及时记录和分析测试结果****: 测试过程中应及时记录测试结果，并对异常情况进行深入分析。这有助于快速定位问题原因，并制定相应的解决方案。

## 商家界面功能

### 测试用例与执行分析

### 测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 003 | | | |
| 功能描述 | 用户可以进行评论打电话等功能 | | | |
| 用例目的 | 实现商家界面查看等 | | | |
| 前提条件 | 用户登录注册 | | | |
| 特殊的规程说明 |  | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 动作 | 期望结果 | 实际结果 | 备注 |
| 1 | 点击商家 | 展示商家商品 | 成功展示商家商品 |  |
| 2 | 点击评论 | 展示商家评论 | 成功展示商家评论 |  |
| 3 | 点击商家Logo | 切换不同商家 | 成功切换不同商家 | … |
| 4 | 点击评论和打电话 | 发起评论和打电话 | 发起评论和打电话功能 |  |

### 分析结果：商家界面功能实现成功

### 测试结果综合分析及建议

****综合分析****:  
通过对用例003的执行和分析，可以看出商家界面的主要功能——查看商品、评论、切换商家以及打电话（如果提供）均运行正常。这表明系统在商家信息展示和用户交互方面表现良好。然而，为了确保用户体验的进一步提升，我们仍需关注以下几个方面：

****建议****:

****增强界面一致性****: 确保不同商家的展示页面在风格和设计上保持一致性，以提高用户的使用舒适度。

****优化加载速度****: 虽然当前页面加载速度已经不错，但在未来版本中应继续优化，特别是在网络条件不佳的情况下。

****增加错误处理机制****: 在用户进行评论或打电话等操作时，应增加相应的错误处理机制，以应对可能出现的异常情况。

****提升用户体验****: 考虑增加一些用户友好的功能，如评论排序、筛选等，以进一步提升用户体验。

### 测试经验总结

****全面覆盖测试场景****: 在测试过程中，应尽可能覆盖所有可能的用户场景和操作流程，以确保系统的全面性和稳定性。

****注重细节测试****: 除了主要功能的测试外，还应关注一些细节方面的测试，如页面加载速度、错误提示信息等。

****记录并分析测试结果****: 测试过程中应及时记录测试结果，并对出现的问题进行深入分析。这有助于快速定位问题原因并制定相应的解决方案。

****持续改进****: 根据测试反馈和用户反馈，不断优化系统功能和用户体验。这是一个持续的过程，需要不断地进行迭代和改进。

## 导航界面功能

### 测试用例与执行分析

### 测试用例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 004 | | | |
| 功能描述 | 用户可以自主选择导航方式进行导航 | | | |
| 用例目的 | 根据目的地和当前位置绘制导航线 | | | |
| 前提条件 | 用户点击导航按钮 | | | |
| 特殊的规程说明 |  | | | |
| 用例间的依赖关系 | 无 | | | |
| 步骤 | 动作 | 期望结果 | 实际结果 | 备注 |
| 1 | 选择目的地 | 选择成功 | 成功选择 |  |
| 2 | 点击导航 | 成功绘制导航线 | 成功绘制 |  |
| 3 | 用户移动位置 | 导航线发生变化 | 成功变化 |  |
| 4 | 用户到的目的地 | 导航线消失，提示到达目的地 | 成功提示 |  |
| … | … | … | … | … |

### 分析结果：功能实现成功

****选择目的地****：用户能够顺利选择目的地，系统准确记录并反馈了用户的选择，无错误或延迟现象。

****点击导航****：用户点击导航按钮后，系统迅速根据当前位置和所选目的地绘制了导航线，显示清晰且准确。

****用户移动位置****：在模拟用户移动位置的过程中，导航线能够实时更新，反映出用户与目的地的最新路径关系，体现了导航系统的实时性和准确性。

****用户到达目的地****：当用户按照导航线到达预设的目的地时，系统及时给出了“到达目的地”的提示，并正确地取消了导航线的显示，用户体验良好。

****建议****：

****增强用户体验****：考虑在导航过程中加入更多人性化的提示信息，如预计到达时间、剩余距离、交通状况等，以进一步提升用户体验。

****优化导航算法****：虽然当前导航算法表现良好，但可以考虑进一步优化算法，减少计算时间，提高导航的实时性和准确性。

****增加离线地图支持****：在无法连接网络的环境下，提供离线地图和导航功能，以扩大导航系统的适用范围。

****提升兼容性****：确保导航系统在不同型号的手机、平板等设备上都能稳定运行，避免因设备差异导致的兼容性问题。

### 测试经验总结

****详细规划测试步骤****：在测试开始前，需要详细规划测试步骤，确保覆盖所有可能的使用场景，以全面评估系统的功能和性能。

****注重实际场景模拟****：在测试中，应尽可能模拟用户的实际使用场景，如模拟用户移动位置、网络延迟等情况，以更真实地反映系统的表现。

****记录详细的测试数据****：在测试过程中，应记录详细的测试数据，包括动作、期望结果、实际结果和备注等，以便于后续分析和问题追踪。

****及时反馈与沟通****：在测试过程中发现的问题应及时反馈给开发团队，并与开发团队保持密切沟通，确保问题得到及时解决。

****持续迭代与优化****：测试是一个持续的过程，随着系统的不断迭代和优化，需要不断调整和完善测试用例，以确保测试的有效性和准确性。

# 系统性能测试

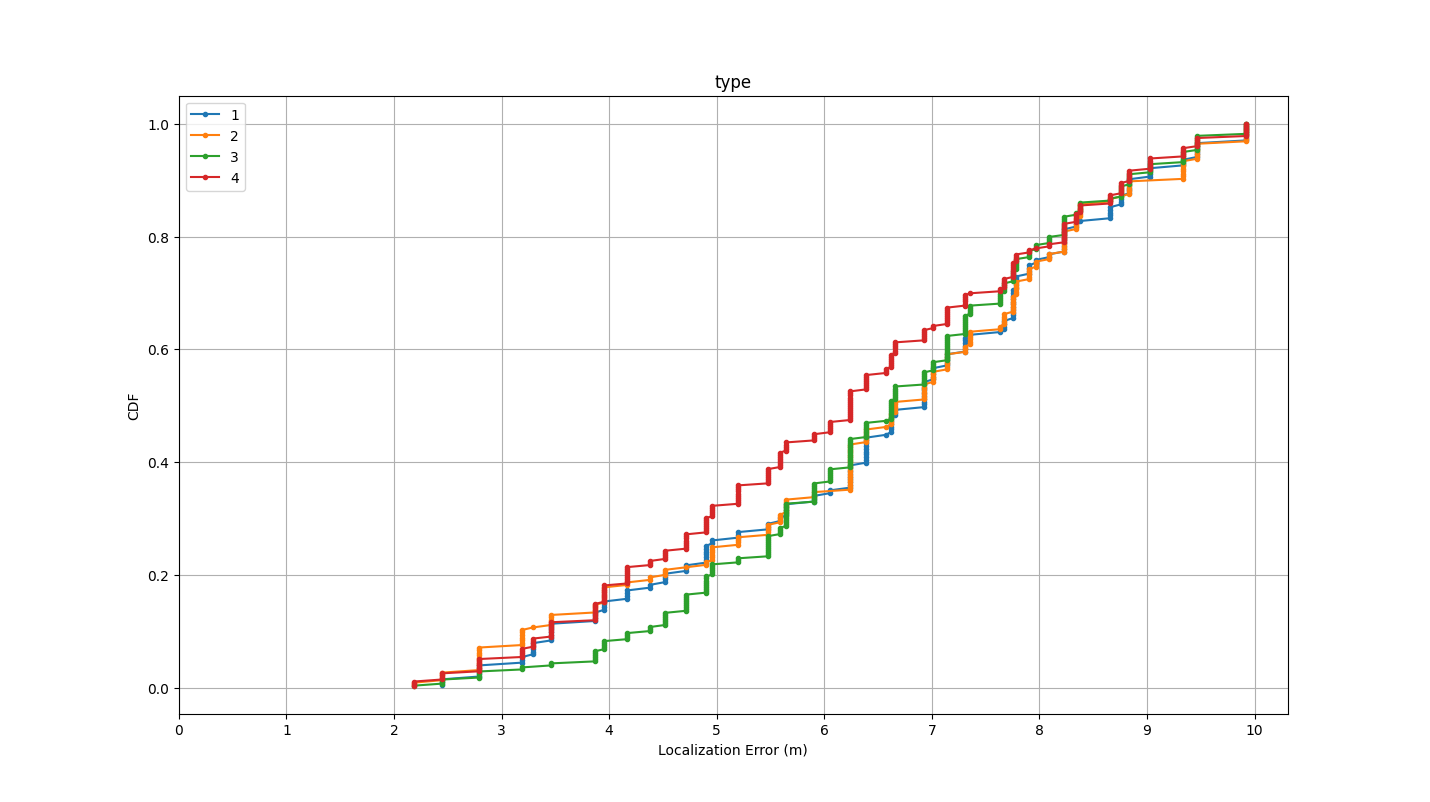
## 定位精度

### 室内环境（形状、面积、设备、采样点、测试点）

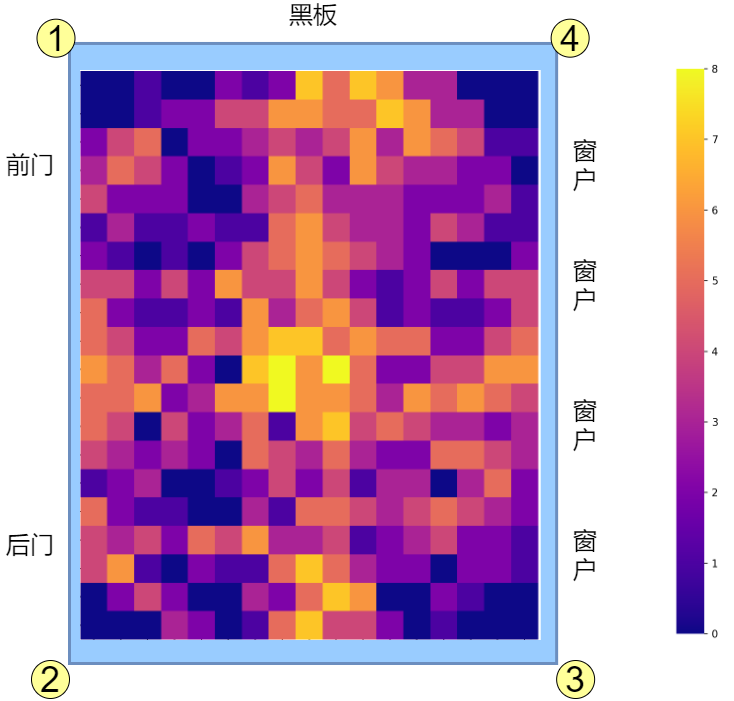
YF515教室，形状近似为矩形，长约20块地砖（0.6\*0.6m),宽约16块地砖，在教师四个角设置采样点，将区域分为四块，以每块地砖作为测试点，每个测试点测试10次，如果该测试点的某次测试中识别位置不符合对应划分的位置，则计入一次error,并且计算该点位距离对应采样点的距离。

设备：小米手机，联想电脑一台

### 定位精度（CDF曲线）



### 热力图



### 定位结果分析总结

最大误差8，平均误差3.1

观察热力图可以发现，在距离采样点最远处，即教室两条中轴线上，测试误差较大，而约靠近采样点误差越小，同时，在讲台处中轴线向靠窗方向也产生较大误差，猜测是收到那里电脑空调等设备的影响，而靠走廊一侧，前后门处的误差也较明显，可能是木质门对WiFi信号的阻挡作用较小，易受到外部环境影响。

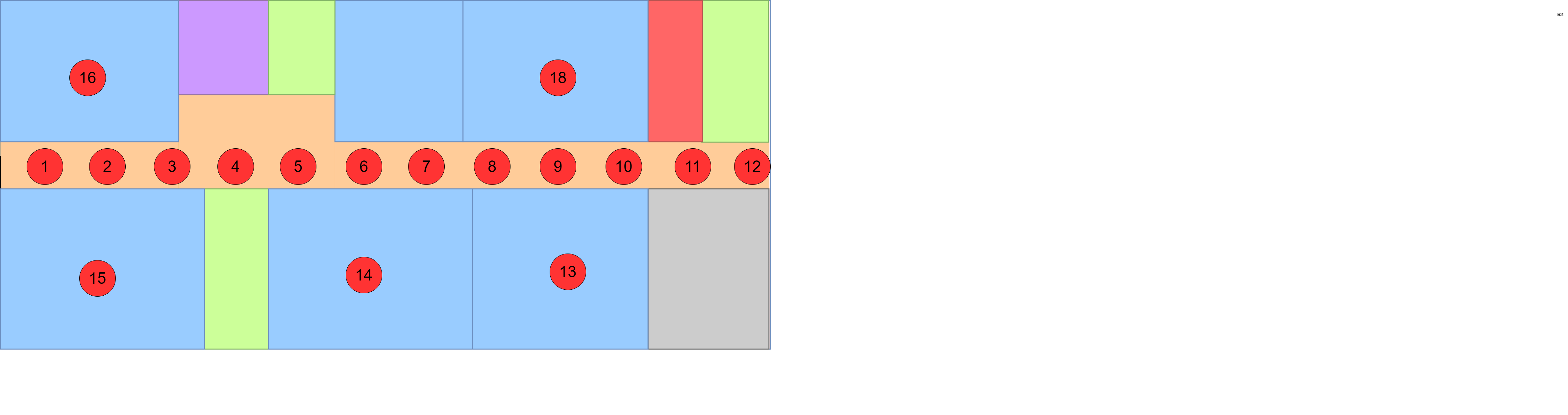
## 房间识别精度

### 实验设置

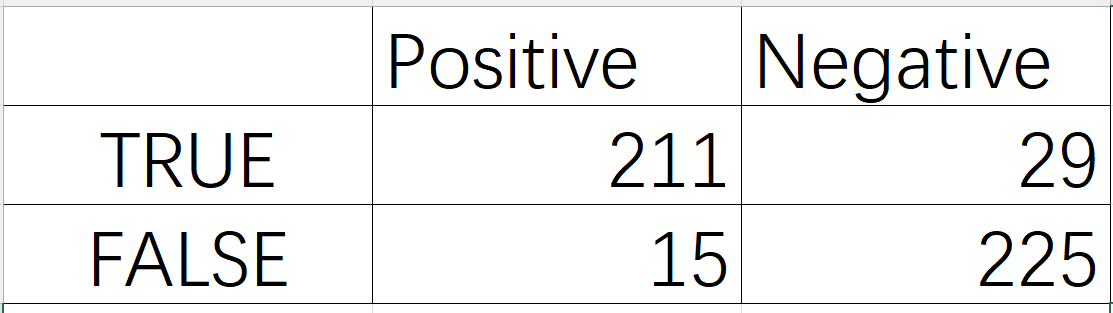
本次测试中，一共设置了16个采样点，其中走廊上有12个，五个教室中各有一个（有一个考研教室不方便进行测试），走廊上的采样点位于走廊中间距离左右墙壁位置相等处，每隔七块地砖，约4.2米设置下一个采样点，教室中的采样点位于教室正中间。

每个教室的测试点有16个，4\*4均匀分布在教室各处，每个测试点测试三次，每个走廊的采样点附近设置四个测试点，测试5次，如果定位信息与实际不符则计入相应的错误。

设备：小米手机，联想电脑一台



### 评价指标



Recall=49.5%

precision=87.9%

accuracy=90.8%

F-score=0.63

### 结果分析

本次测试中，发现在教室房间内的精准度相对较高，但靠近门窗的点位容易出现定位到走廊的现象，推测原因是门窗附近与走廊环境相连通，易受到外界环境干扰，采样时该点位信息与走廊信息差异较小，容易被判断为走廊点位。

## APP性能

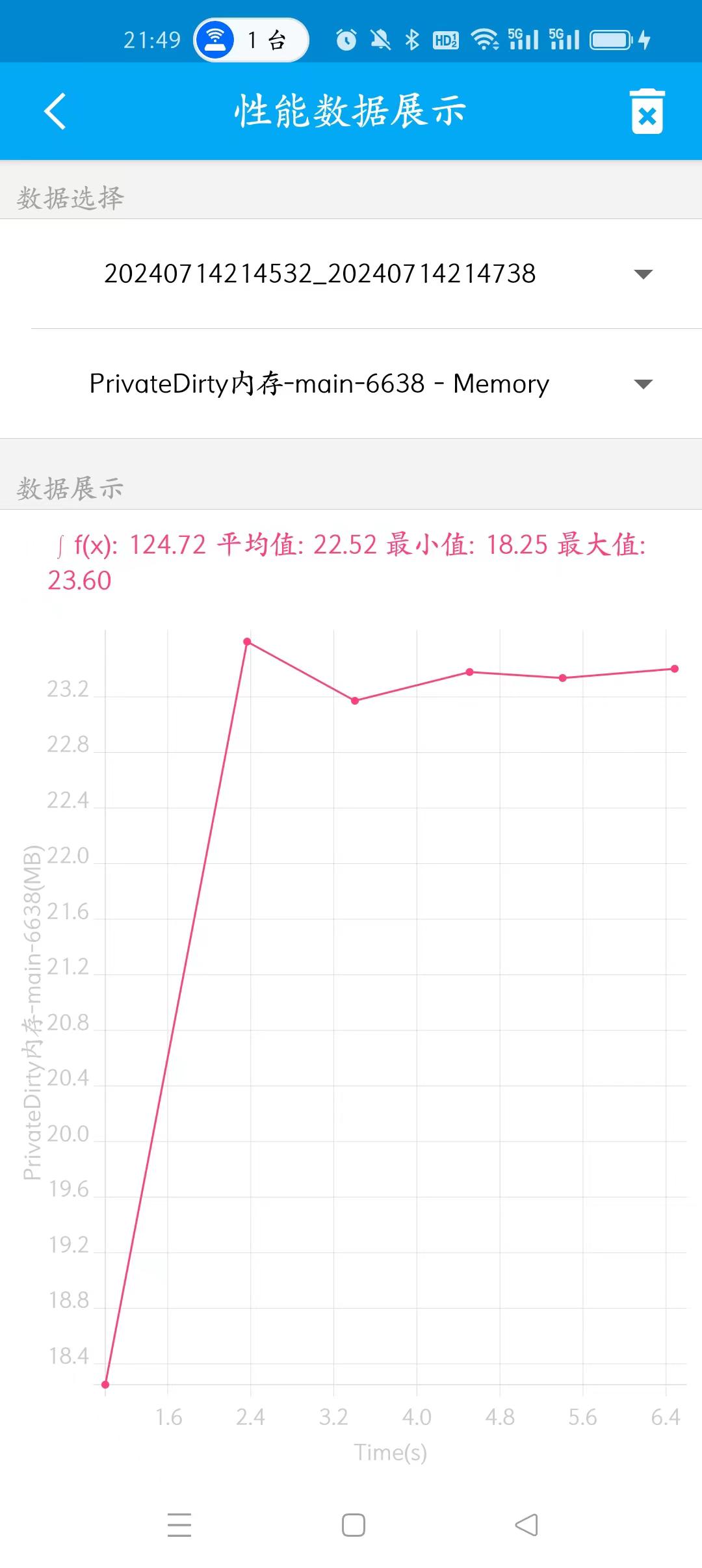
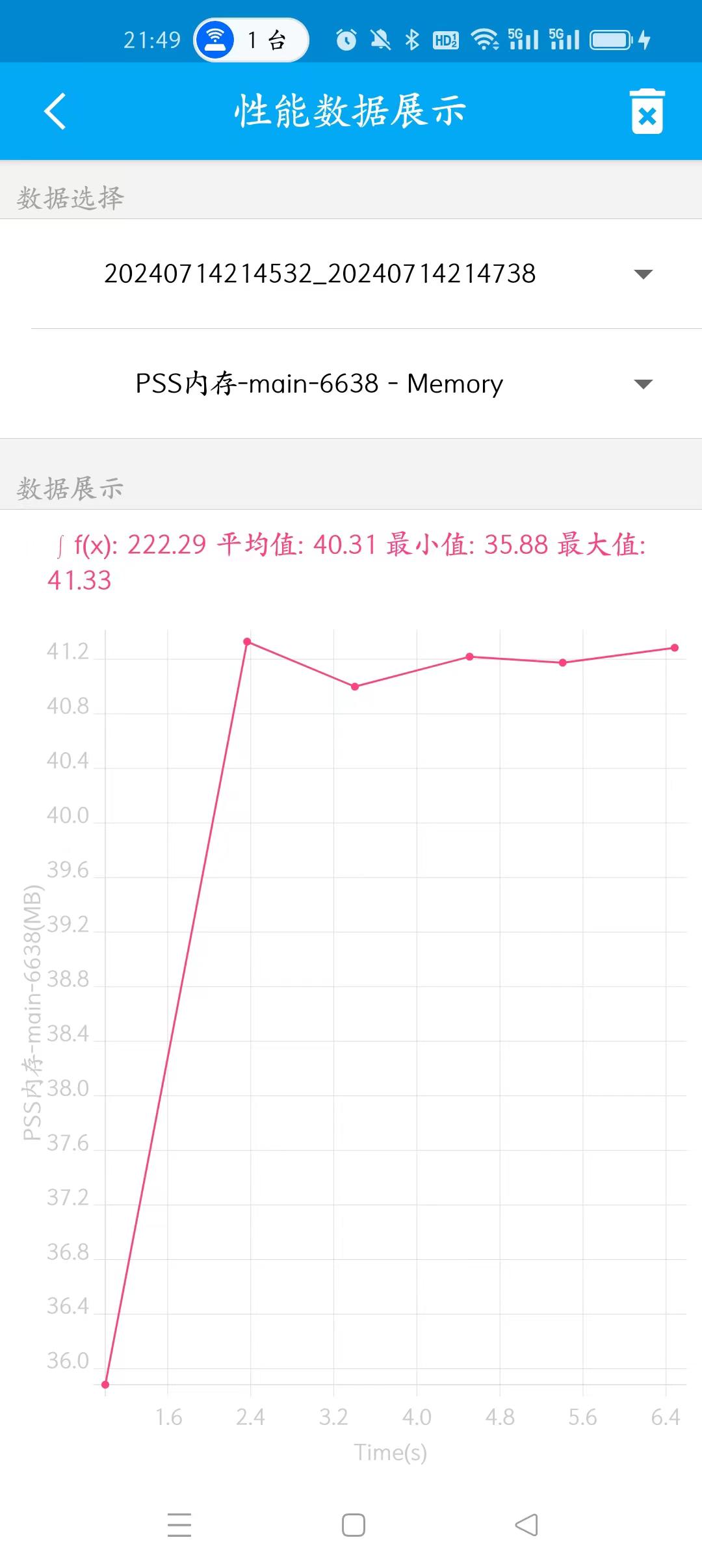
### 延时

我们各相关功能的耗时都在1s以内

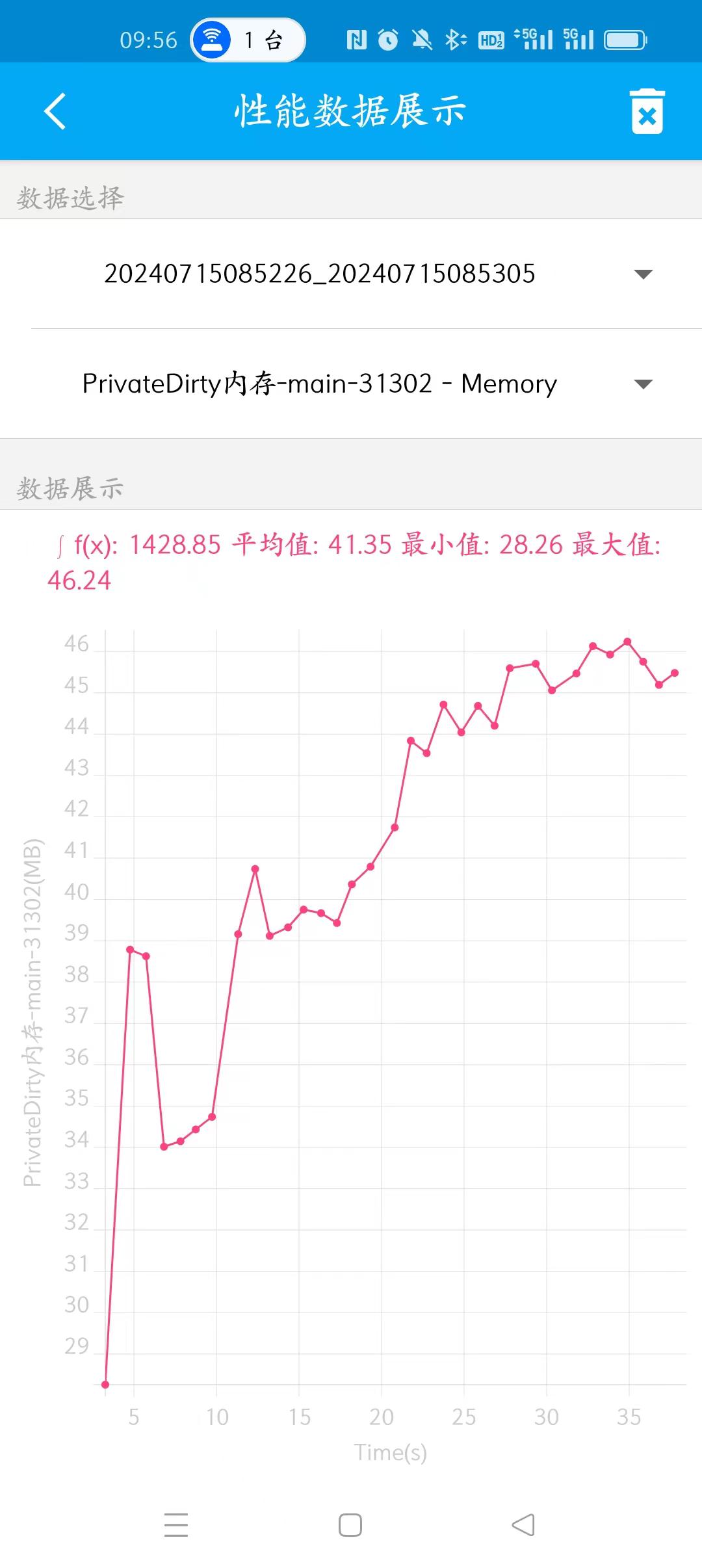
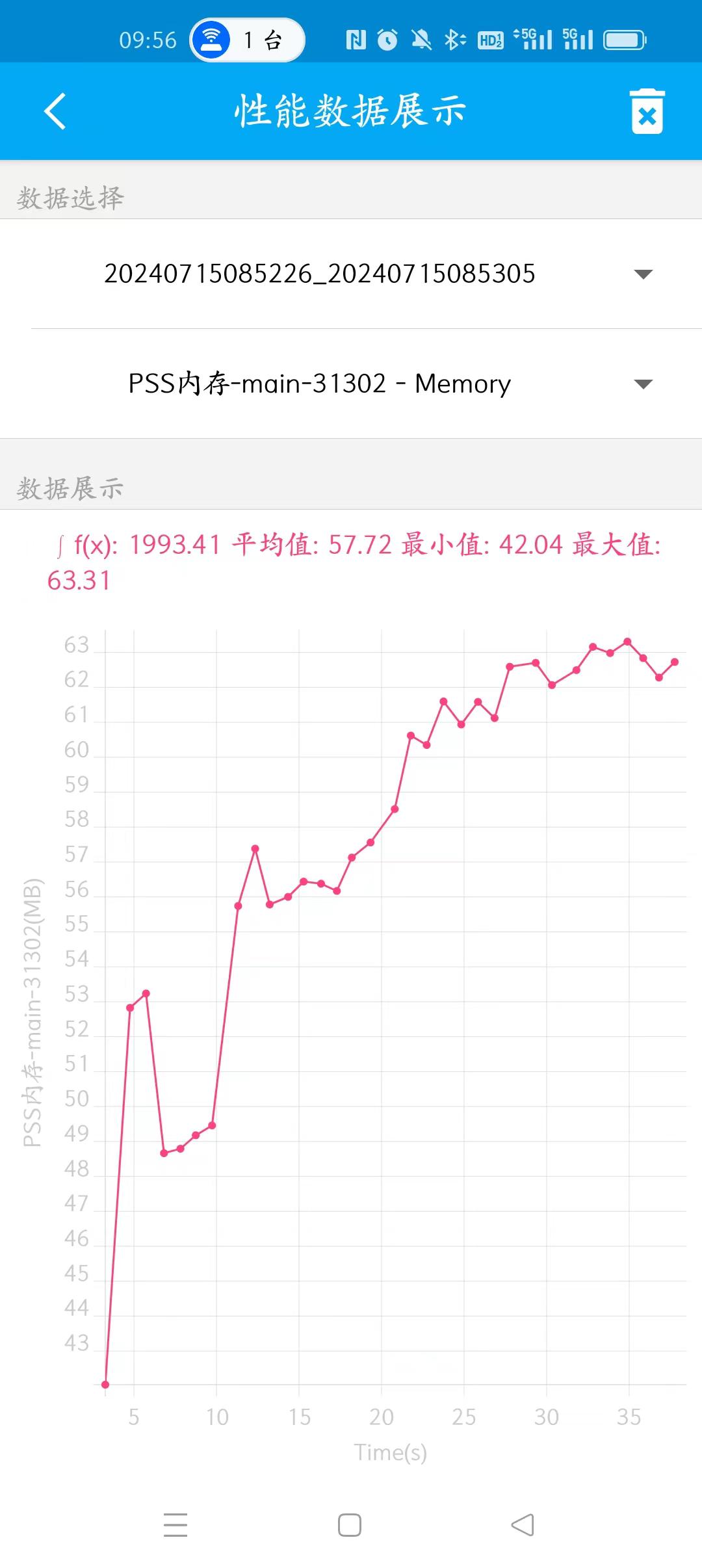
### 存储

应用私有的全部内存保持在50mb以下。

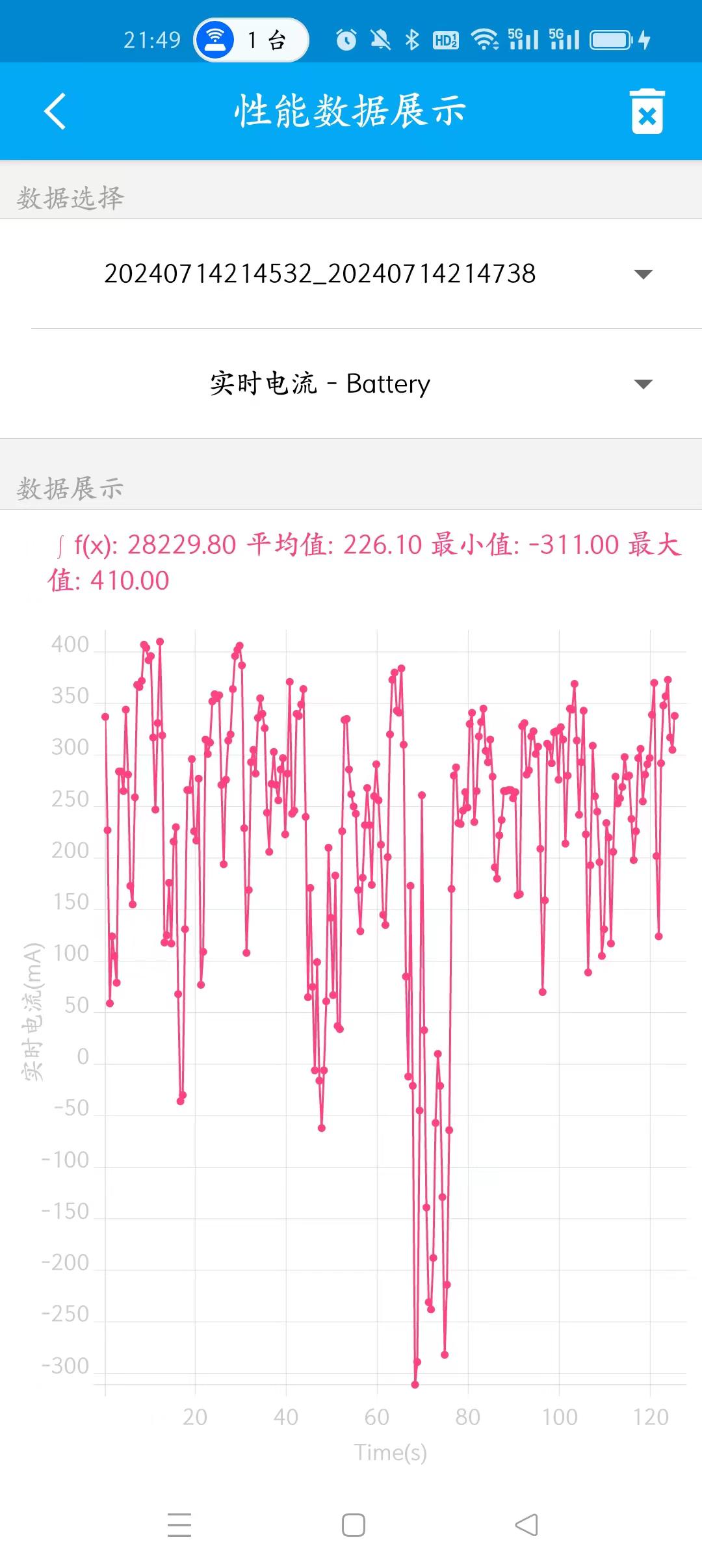
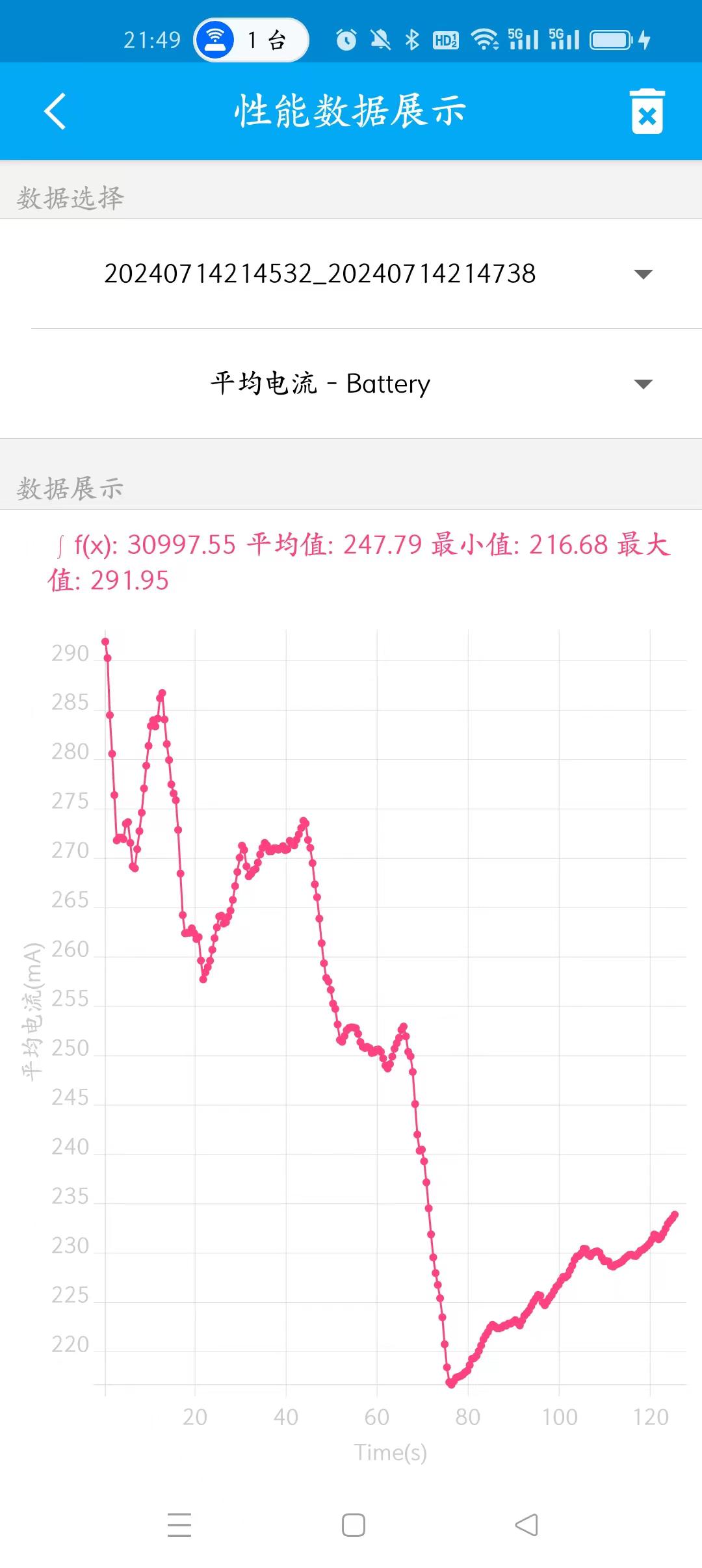
应用私有已修改内存占用保持在30mb以下



加入了导航功能，可以发现占用的内存有了上升

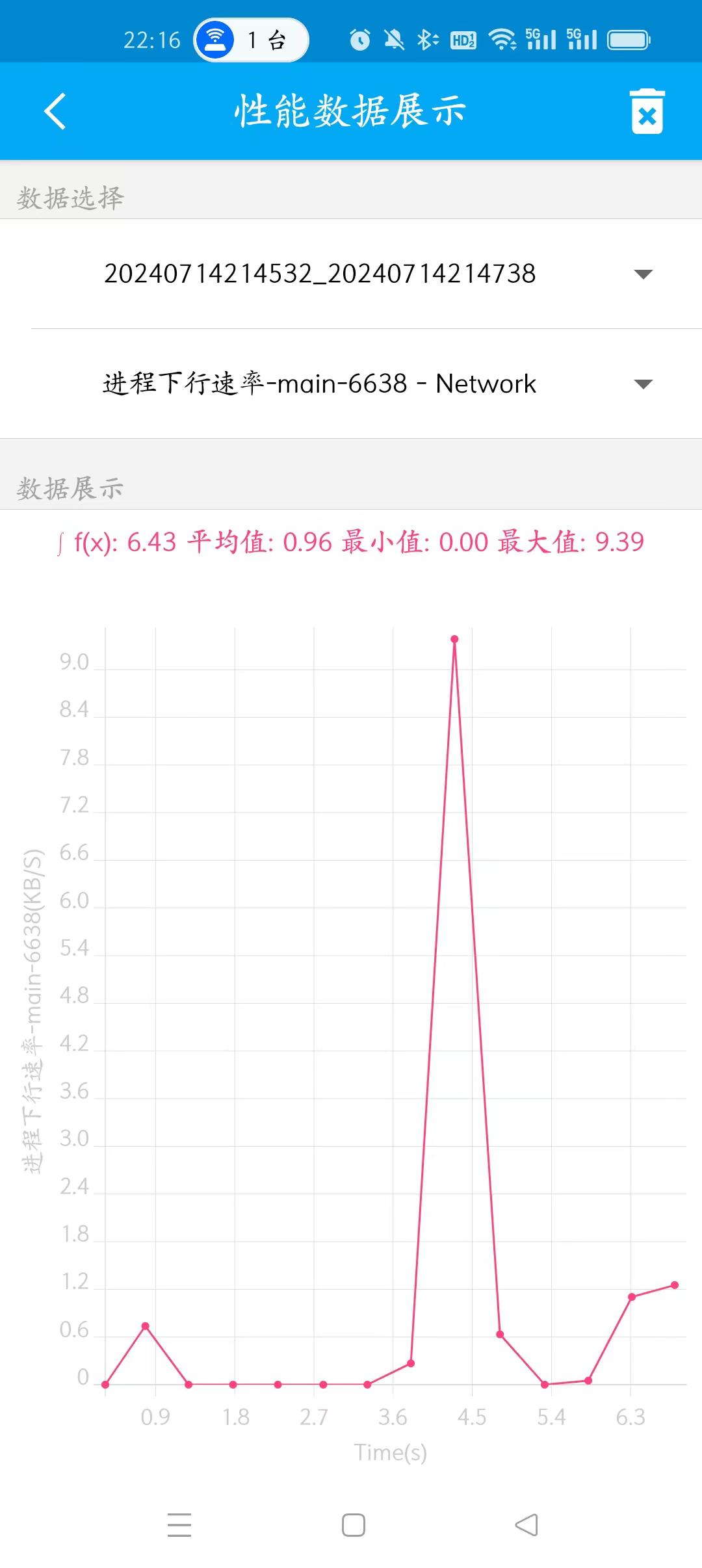
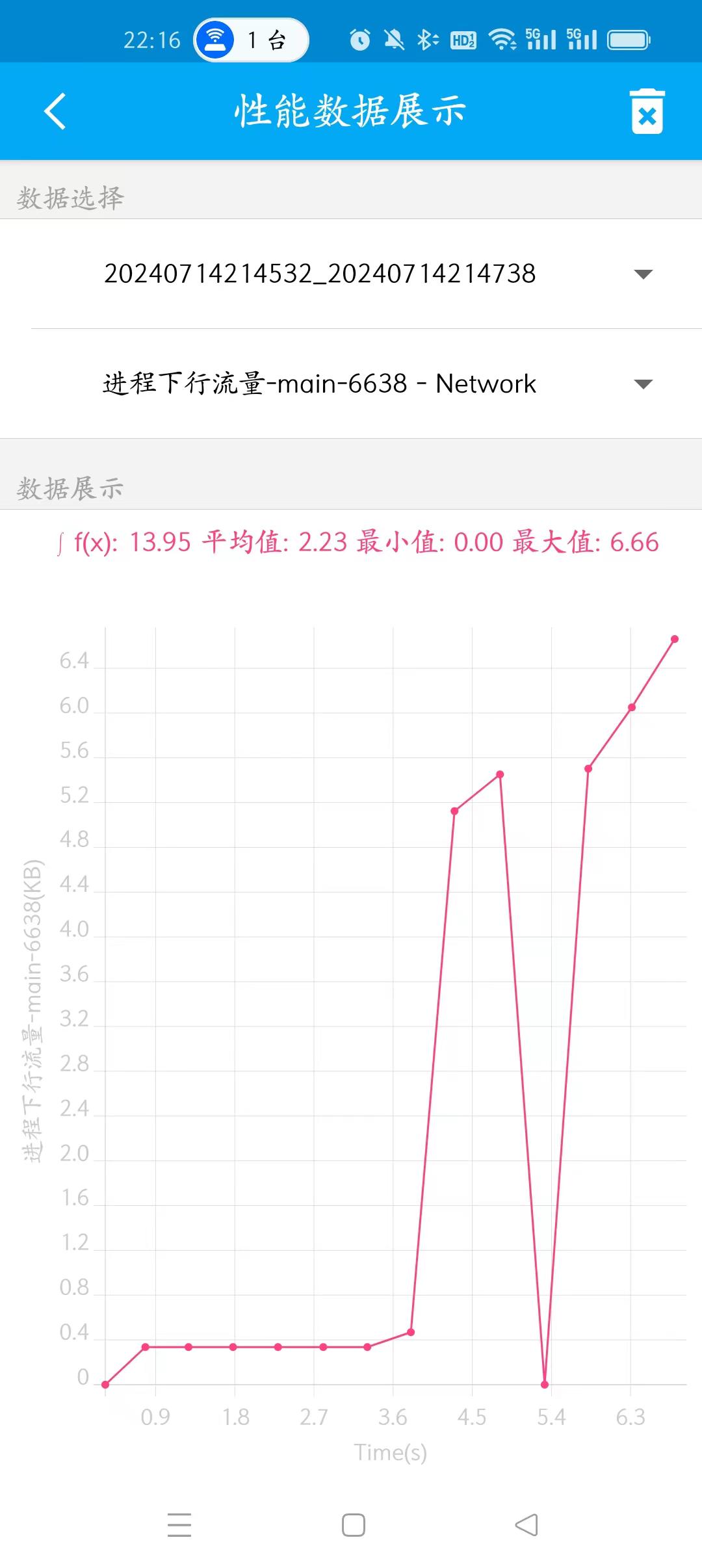
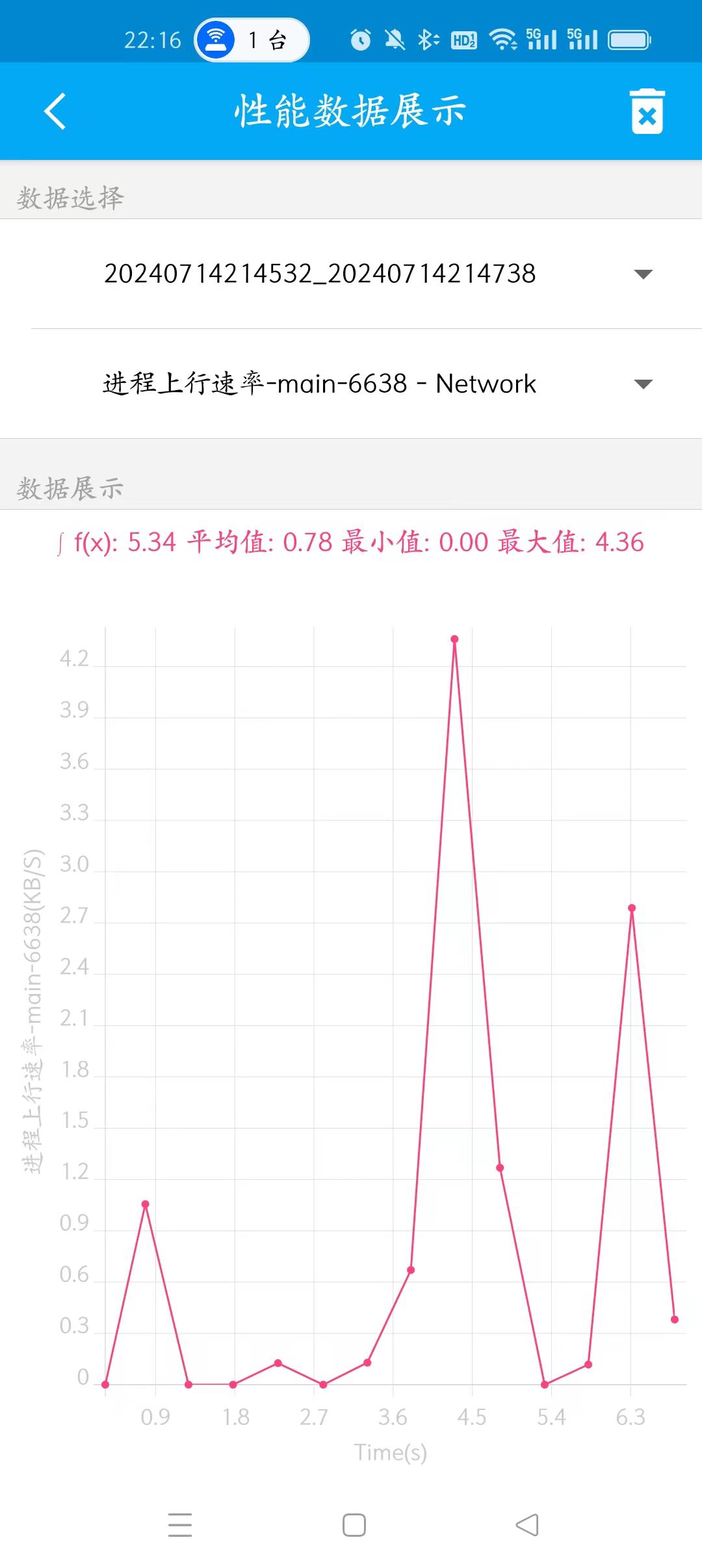
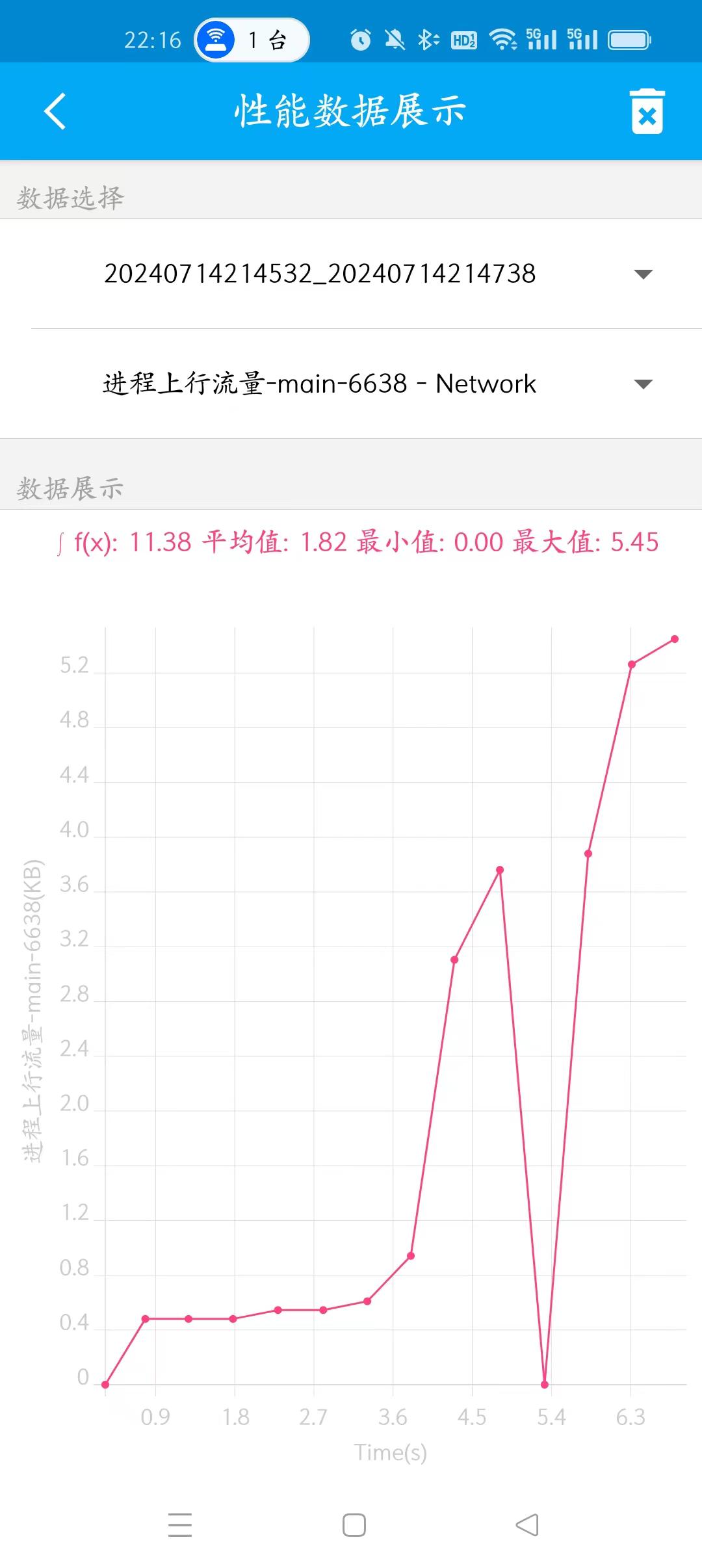
、

### 功耗

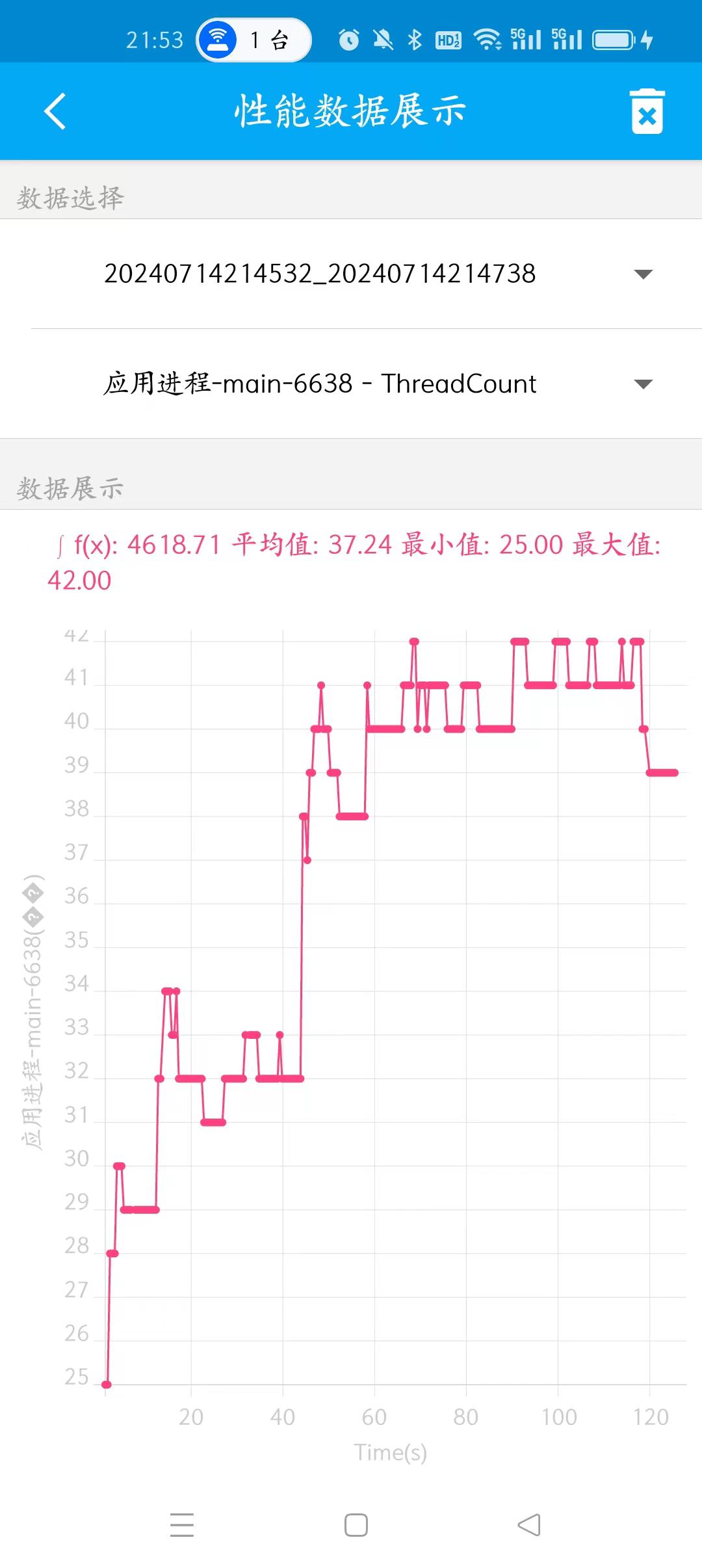


### network

我们对应用上传以及下载的内容进行简化各种，对服务端处理的各种结果进行编码，确保了消息发送的精简性。除此之外，对图片个中资源做了压缩处理，降低其储存、网络要求的同时保证了资源质量。



### 应用进程



# 测试结果总结及优缺点分析

软件测试是软件开发过程中至关重要的一环，它旨在发现软件中的错误、缺陷或不符合需求的地方，以确保软件的质量和可靠性。

### **优点**

我们的软件极大的考虑了用户体验，在各项功能使用时从实际出发，确保app的运行有效，从测试的各项指标来看我们的app体现出了不错的性能。

****提高软件质量****：通过测试我们发现并修复软件中的错误和缺陷，在我们的测试中，发现评价功能和评价展示在多用户的情况下有些问题，我们进行了修复，从而显著提升软件的质量，减少用户在使用过程中遇到的问题。

****增强用户满意度****：从用户的角度，体验整个app，能够让我们的app提供更好的用户体验，减少用户投诉和退货，从而增强用户满意度和忠诚度。

****降低维护成本****：早期发现并修复缺陷，可以避免在软件发布后需要花费大量时间和资源去修复这些问题，从而有效降低长期维护成本。

****提供文档和反馈****：测试过程和结果可以作为软件开发的文档之一，帮助开发者理解软件的行为和性能，同时也为未来的开发和改进提供有价值的反馈。

****增强风险管理****：通过测试，可以提前识别并评估软件中存在的潜在风险，有助于项目团队制定有效的风险缓解策略。

### **缺点**

****性能问题****：

****并发处理能力差****：在多用户或者频繁使用的情况下，后台会出现丢包现象，导致了评价功能有所问题，这个问题在测试的时候出现过，使用的时候并未出现。

****安全性问题****：

****漏洞和弱点****：软件中存在可被恶意利用的漏洞或弱点，这是我们未考虑的

****数据加密不足****：敏感数据未进行充分加密或保护，容易被窃取或篡改。

### **总结**

****加强测试团队的培训和技能提升****：提高我们自身关于测试专业能力和经验，确保测试结果的准确性和有效性。

****采用自动化测试****：利用自动化测试工具减少人工测试的工作量，提高测试效率和覆盖率，在此次测试中我们成功使用

****制定合理的测试策略****：根据项目的实际情况和需求，制定合理的测试计划和策略，确保测试的全面性和有效性。

总结来说，我们的软件做到了运行有效，支持多用户，用户体验不错，这是一次成功的安卓开发。