



第十六届全国大学生软件创新大赛

文档编号: SWC2023-332-淀粉队



轻运食刻

Light Time

项目测试文档

Version:1.4



淀粉队

2023-2-25

All Rights Reserved

目录

1	测试计划	1
1.1	测试策略与目标	1
1.2	测试范围	2
1.3	测试环境	3
2	单元测试	5
2.1	****模块	5
2.1.1	测试用例与结果分析	5
2.1.2	测试结果综合分析及建议	5
2.1.3	测试经验总结	5
2.2	****模块	5
3	功能测试	6
3.1	****功能	6
3.1.1	测试用例与结果分析	6
3.1.2	测试结果综合分析及建议	6
3.1.3	测试经验总结	6
3.2	****功能	6
4	系统测试	7
4.1	模型性能测试	7
4.1.1	测试用例与结果分析	7
4.1.2	测试结果综合分析及建议	7
4.1.3	测试经验总结	7

文档修订历史

[illegible]

1 测试计划

1.1 测试策略与目标

【测试策略：大概分为

功能测试

UI 测试

兼容性测试

性能测试

1.功能测试：确保应用程序的主要功能正常工作，包括：

注册和登录功能是否正常

添加、编辑和删除饮食记录的功能是否正常

添加、编辑和删除运动记录的功能是否正常

查看饮食和运动历史记录的功能是否正常

2.UI 测试：测试应用程序的用户界面，包括：

界面元素是否与设计一致

用户可以正确地通过界面与应用程序进行交互

应用程序是否对不同的设备尺寸和方向进行了适当的优化

3.兼容性测试：测试应用程序在不同版本的 iOS 上的表现，包括：

测试应用程序在最新版本的 iOS 上是否兼容

测试应用程序在较旧版本的 iOS 上是否兼容

4.性能测试：测试应用程序的性能，包括：

应用程序的响应速度是否足够快

应用程序在各种网络环境下的表现

应用程序的稳定性和可靠性

5.安全测试：测试应用程序的安全性，包括：

用户的数据是否安全地存储和传输

应用程序是否具有足够的身份验证和授权机制

应用程序是否具有足够的防护措施来防止黑客攻击和数据泄漏

6.自动化测试：使用自动化测试工具来自动化测试应用程序的各个方面，包括：

通过执行以上测试策略，可以项目的应用程序的各个方面的质量和可靠性，提高用户的满意度和应用程序的成功率。】

【目标：根据项目的进度总结出了如下测试目标

1.功能测试目标:

应用程序的所有功能均能正常使用

应用程序的所有功能都遵循预期的逻辑和流程

应用程序的功能满足用户需求和期望

2.UI 测试目标:

应用程序的用户界面易于使用, 与设计一致, 具有一致性和美观性

应用程序的用户界面与不同的设备尺寸和方向兼容

应用程序的用户界面符合用户习惯和期望

3.兼容性测试目标:

应用程序能够在所有支持的 iOS 版本上正常运行

应用程序能够在所有支持的 iOS 设备上正常运行

应用程序的功能和性能不受不同的硬件和软件配置影响

4.性能测试目标:

应用程序的响应速度足够快, 不会出现长时间等待的情况

应用程序在各种网络环境下的表现稳定和可靠

应用程序的资源利用率合理, 不会过度消耗设备的电量和存储空间

5.安全测试目标:

应用程序能够安全地存储和传输用户的数据

应用程序能够正确地进行身份验证和授权, 保护用户的隐私和安全

应用程序能够有效地防止黑客攻击和数据泄漏

6.自动化测试目标:

自动化测试能够有效地检测应用程序中的缺陷和问题

自动化测试能够快速地进行回归测试, 减少测试的时间和成本

自动化测试能够提高测试的覆盖率和准确性

通过实现以上测试目标, 可以保证应用程序的质量和可靠性, 提高用户的满意度和应用程序的成功率。】

1.2 测试范围

根据项目的目前进度从测试广度和测试深度两个方面来了解整个测试项目的测试规模:

1.测试广度:

测试广度指的是测试项目需要覆盖的功能和特性的数量和范围。在一个 iOS 健康饮食身体记录 app 的测试项目中, 需要覆盖的功能和特性包括但不限于:

(1)用户注册和登录功能

(2)记录饮食、身体活动、睡眠等数据的功能

(3)统计和分析用户数据的功能

(4)提供健康建议和指导的功能

(5)提供社交和分享功能

(6)提供数据备份和恢复功能

(7)提供设置和个性化功能

(8)测试广度还需要考虑测试不同的设备、不同的操作系统版本和不同的网络环境下的应用程序的兼容性和稳定性, 以确保应用程序的可靠性和用户体验。

2.测试深度:

测试深度指的是测试项目需要覆盖的每个功能和特性的详细程度和测试方法的选择。本项目是基于 iOS 的 app 测试项目中, 需要进行的测试深度包括但不限于:

(1)对用户注册和登录功能的验证

(2)对记录饮食、身体活动、睡眠等数据的准确性和完整性进行测试

(3)对统计和分析用户数据的算法和方法进行验证和测试

(4)对提供健康建议和指导的内容和准确性进行测试

(5)对提供社交和分享功能的兼容性和稳定性进行测试

(6)对数据备份和恢复功能的可靠性和恢复性进行测试

(7)对设置和个性化功能的兼容性和稳定性进行测试

(8)测试深度还需要考虑不同的测试方法, 包括但不限于手动测试、自动化测试、性能测试、安全测试、压力测试等, 以覆盖不同的测试场景和模拟真实的使用情况, 以确保应用程序的质量和可靠性。

1.3 测试环境

1.硬件环境:

本项目是基于 iOS 健康饮食身体记录 app。为了有效地测试需要在 iOS 设备上运行, 因此需要考虑测试设备的型号、屏幕尺寸和操作系统版本。测试设备的型号和屏幕尺寸需要考虑到应用程序在不同设备上的布局 and 适应性, 而测试设备的操作系统版本需要考虑到应用程序在不同操作系统版本上的兼容性。

兼容 iOS10 以上的各种 iPhone 机型。

2.网络环境:

本项目是基于 iOS 健康饮食身体记录 app。为了有效地测试需要使用网络连接来保存和同步用户数据, 因此需要考虑在不同的网络环境下的应用程序的性能和稳定性。需要测试应用程序在 2G、3G、4G 和 Wi-Fi 等不同的网络环境下的响应时间、加载时间、数据传输速度和容错能力等, 以确保应用程序的网络连接性

和用户体验。

3.测试工具:

本项目是基于 iOS 健康饮食身体记录 app。为了有效地测试, 需要选择适当的测试工具。:

Xcode: Xcode 是一款集成开发环境, 提供了应用程序的构建、调试和测试功能。Xcode 可以用于手动测试和自动化测试。

Appium: Appium 是一款开源的自动化测试工具, 支持 iOS 和 Android 平台。Appium 可以用于模拟用户行为、生成测试报告和集成测试等。

Charles: Charles 是一款网络代理工具, 可以用于模拟不同的网络环境, 监控应用程序的网络连接, 捕获和分析网络请求和响应。

XCTest: XCTest 是 Xcode 提供的测试框架, 可以用于单元测试、性能测试和 UI 测试等。XCTest 可以用于验证应用程序的正确性和性能。

除了上述工具, 还可以使用其他测试工具, 如 Jenkins、Selenium、LoadRunner 等, 以覆盖不同的测试场景和测试方法, 以提高测试效率和测试覆盖率。

2 单元测试

2.1 ****模块

2.1.1 测试用例与结果分析

单元测试用例:

用例编号	001			
测试单元描述				
用例目的				
前提条件				
特殊的规程说明				
用例间的依赖关系				
具体步骤	输入	期望输出	实际输出	备注
1				
2				
...

测试结果分析:

2.1.2 测试结果综合分析及建议

2.1.3 测试经验总结

2.2 ****模块

3 功能测试

3.1 ****功能

3.1.1 测试用例与结果分析

测试用例:

用例编号	001			
功能描述				
用例目的				
前提条件				
特殊的规程说明				
用例间的依赖关系				
具体步骤	输入	期望结果	实际结果	备注
1				
2				
...

测试结果分析:

3.1.2 测试结果综合分析及建议

3.1.3 测试经验总结

3.2 ****功能

4 系统测试

4.1 模型性能测试

4.1.1 测试用例与结果分析

预期性能指标测试用例:

用例编号	001			
性能描述				
用例目的				
前提条件				
特殊的规程说明				
用例间的依赖关系				
具体步骤	输入/动作	期望的性能 (平均值)	实际的性能 (平均值)	备注
1				
2				
...

测试结果分析:

4.1.2 测试结果综合分析及建议

4.1.3 测试经验总结