# 软件测试计划书

邓伟亮，傅信燊，吕韦辰，向航，欧阳家睿 *

中山大学（深圳校区） 智能工程学院 528406

* 所有作者为同等贡献，按姓名首字母排序

# 1 引言

## 1.1 测试背景

FocusToDo小程序项目的实施是为了满足现代生活中对时间管理和工作效率提升的需求。该项目由开发者共同合作完成，旨在为用户提供一款简单实用、功能完善的时间管理工具,其中番茄工作法作为一个有效的专注方法，是本项目的灵感来源。

本软件测试计划书旨在对小程序 "FocusToDo" 进行全面的功能和非功能测试，以验证其功能的完整性和稳定性，确保用户能够获得良好的使用体验。

## 1.2 目的

对小程序 "FocusToDo" 进行全面的功能和非功能测试确保小程序的功能符合需求文档中的描述，验证小程序的稳定性和安全性，确保在各种情况下均能正常运行，发现并解决可能存在的缺陷和问题，提高小程序的质量和可靠性。

## 1.3 测试目标

1.确认小程序各项功能的正确性和完整性。

2.验证小程序在不同操作系统和终端设备上的兼容性。

3.检查小程序的数据安全性和用户隐私保护措施。

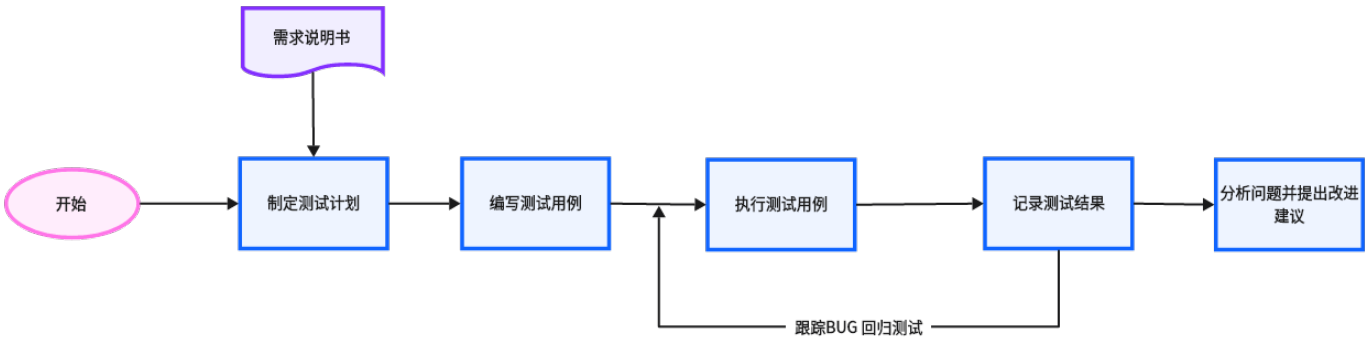
4.确保小程序界面友好，响应速度快，用户体验良好。

## 1.4 适用范围与读者对象

本测试计划书适用于小程序 "FocusToDo" 的开发和测试团队。读者对象包括但不限于开发人员、测试人员、产品经理和项目经理。

# 2 测试实施流程

## 2.1 测试工作总体流程




## 2.2 测试内容

### 2.2.1 代码测试

单元测试

集成测试

系统测试

### 2.2.2 功能测试

专注计时功能测试：自定义设置时分，正确倒计时；可选择对应代办事项

代办事项管理功能测试：正常添加、修改、删除代办事项

打卡功能测试：正常学习、工作、休息打卡

个人中心功能测试：个人信息，专注时长排行榜

数据统计功能测试：每次专注打卡记录数据

AI助手功能测试：正常对话问答

白噪声背景音功能测试：选择播放对应背景音

### 2.2.3 非功能测试

兼容性测试：验证在不同操作系统和终端设备上的兼容性

前端兼容性测试，侧重于客户对客户端的数据和功能的展示及操作的易用性来测试。此类测试针对后台功能的交互输出数据及前台展示的实际效果做出符合客户第一视角的感官来做出判断。用户核实用户与软件之间的交互。测试的目标是确保用户界面会通过测试对象的功能来为用户提供相应的功能。另外，测试还可确保对象按照预期的方式运行，并符合需求要求标准。

| 测试目标 | 测试前台功能展示的样式及格式 |
| --- | --- |
| **测试范围** | 不仅限于分辨率、系统版本及浏览器渲染等情况的测试 |
| **技术** | 通过第三方浏览器访问的方法 |
| **测试重点及优先级** | 前台展示及功能是否正常 |
| **需考虑的特殊事项** | 不同版本，不同终端（手机和PC），不同浏览器 |

性能测试：评估小程序的响应速度和性能表现

| 测试目标 | **涉及XXX接口，发现性能问题并配合研发优化解决问题，支持业务能在高并发的情况良好的运行。** |
| --- | --- |
| **测试范围** | 压力测试，测试峰值 |
| **技术** | LoadRunner 是一种预测系统行为和性能的负载测试工具。通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，LoadRunner 能够对整个企业架构进行测试。通过使用LoadRunner ，企业能最大限度地缩短测试时间，优化性能和加速应用系统的发布周期 |
| **测试重点及优先级** | 当前测试环境下增加到临界压力，监控当前情况下服务及server的资源使用情况，对后续风险项起到预警作用，同时针对不同压力情况下的负载测试和稳定性测试 |
| **需考虑的特殊事项** | 考虑当前环境是否符合测试标准 |

## 2.3 测试计划

### 2.3.1 测试资源与环境

测试人员：具备相关测试经验和技能的人员

测试环境：包括iOS和Android操作系统的手机和平板设备

测试工具：测试用例管理工具、安全漏洞扫描工具等

### 2.3.2 软件配置

小程序版本：最新发布版本

操作系统：iOS、Android

终端设备：手机、平板电脑

# 3 测试执行

## 3.1 测试策略

单元测试 ：将代码划分为独立的、可测试的部分，并测试其在各种情况下的行为和输出。通过这种方式，开发人员可以在代码被整合到更大的系统之前快速、准确地确定问题所在，并确保单个代码单元的正确性。

集成测试：测试不同的软件模块之间的交互和协作是否正常。主要目的是确保不同的软件模块能够无缝协作，形成一个完整的软件系统，并且能够满足系统的需求和规格。

## 3.2 测试技术与方法

黑盒测试：验证功能是否符合需求。不考虑程序内部结构和处理过程,仅针对程序是否能适当地接收输入数据、是否能产生正确的输出信息等进行测试。

白盒测试：检查代码逻辑和结构。通过对程序内部结构的分析与检测来寻找软件问题的方法称为白盒测试，又称结构测试。
白盒测试可以把程序看成是一个装在透明的白盒子里的代码,测试人员清楚地了解程序的内部结构和处理过程,通过检查程序的内部结构及逻辑路径是否正确、检查软件内部动作是否符合软件设计说明书的规定来发现程序中的缺陷

手动测试：在真实环境中进行测试

自动化测试：使用自动化测试工具进行重复性测试

## 3.3 测试用例

### 3.3.1 单元测试

### 3.3.2 Load function

- 测试用例编号： Code-Test-1
- 预置条件：todo_list不为空
- 测试用例标题：增加任务列表
  - 步骤描述：在todo_list的静态数据中增加“数学作业”

  - 预期输出：输出“数学作业”
  - 实际输出：输出“数学作业”
  - 执行结果：Y
- 测试用例标题：删除任务列表
  - 步骤描述：在todo_list的静态数据中删除“数学作业”
  - 预期输出：输出“null”
  - 实际输出：输出“null”
  - 执行结果：Y

#### 3.3.3 saveLog function

- 测试用例编号： Code-Test-2
- 测试用例标题：保存用户日志
  - 步骤描述：data.log里面增加log结构体，包含数学作业的完成信息
  - 预期输出：输出“数学作业完成信息”
  - 实际输出：输出“数学作业完成信息”
  - 执行结果：Y

#### 3.3.4 handleAddToDo function

- 测试用例编号： Code-Test-3
- 预置条件：任务列表为空
- 测试用例标题：用户增加任务列表
  - 步骤描述：调用该函数增加“数学作业”的任务
  - 预期输出：静态存储数据中的"todo_list"会增加数学作业
  - 实际输出：静态存储数据中的"todo_list"会增加数学作业
  - 执行结果：Y

#### 3.3.5 clearCompletedHandle function

- 测试用例编号： Code-Test-4
- 预置条件：任务列表不为空
- 测试用例标题：清理所有已经完成的任务
  - 步骤描述：在todo_list的静态数据中增加“数学作业”，并且标记为已经完成，调用该函数
  - 预期输出：输出“null”

  - 实际输出：输出“null”
  - 执行结果：Y

#### 3.3.6 getLogs function

- 测试用例编号： Code-Test-5
- 预置条件：logs列表不为空
- 测试用例标题：读取日志列表
  - 步骤描述：在日志中存储5个log结构体
  - 预期输出：输出“5个结构体的具体信息”
  - 实际输出：输出“5个结构体的具体信息”
  - 执行结果：Y

#### 3.3.7 sendSocketMessage function

- 测试用例编号： Code-Test-6
- 测试用例标题：GPT请求测试
  - 步骤描述：自定义若干条测试请求
  - 预期输出：针对每个请求都得到了对应的回复
  - 实际输出：针对每个请求都得到了对应的回复
  - 执行结果：Y

#### 3.3.8 handleGetup function

- 测试用例编号： Code-Test-7
- 测试用例标题：起床打卡函数测试
  - 步骤描述：直接运行该函数
  - 预期输出：静态数据logs会更新，并且跳转到share页面
  - 实际输出：静态数据logs会更新，并且跳转到share页面
  - 执行结果：Y

#### 3.3.9 handleSleep function

- 测试用例编号： Code-Test-8
- 测试用例标题：睡觉打卡函数测试
  - 步骤描述：直接运行该函数
  - 预期输出：静态数据logs会更新，并且跳转到share页面

  - 实际输出：静态数据logs会更新，并且跳转到share页面
  - 执行结果：Y

#### 3.3.10 集成测试

#### 3.3.11 计时器和设置集成

- 测试用例编号： Code-Test-9
- 测试用例标题：计时器和设置集成
  - 步骤描述：调整setting中的changeWorkTime函数，分别设置参数为20分钟，50分钟和80分钟
  - 预期输出：设置成功，设置成功，设置失败
  - 实际输出：设置成功，设置成功，设置失败
  - 执行结果：Y

#### 3.3.12 计时器和任务列表集成

- 测试用例编号： Code-Test-10
- 预置条件：todo_list数据不为空
- 测试用例标题：计时器和任务列表集成
  - 步骤描述：多次调用handleAddToDo函数，然后再调用index里面的handlePickerChange函数
  - 预期输出：有多个任务的输出
  - 实际输出：有多个任务的输出
  - 执行结果：Y

#### 3.3.13 打卡和分享集成

- 测试用例编号： Code-Test-11
- 预置条件：logs不为空
- 测试用例标题：打卡和分享集成
  - 步骤描述：调用getLogs函数之后，再调用onShareAppMessage函数
  - 预期输出：打开分享页面，并且页面上的打卡次数和打卡页面的次数相同
  - 实际输出：打开分享页面，并且页面上的打卡次数和打卡页面的次数相同
  - 执行结果：Y

#### 3.3.14 计时器和打卡集成

- 测试用例编号： Code-Test-12
- 测试用例标题：计时器和打卡

  - 步骤描述：调用两次index里面的startTimer，然后查看打卡功能里面的getLogs的输出是否同步
  - 预期输出：getLogs的输出与startTimer的时间和任务同步
  - 实际输出：getLogs的输出与startTimer的时间和任务同步
  - 执行结果：Y

#### 3.3.15 系统测试

#### 3.3.16 系统测试概述

- **测试用例编号**：Sys-Test-1
- **测试用例标题**：整体系统测试
- **预置条件**：所有模块已集成，系统处于可操作状态。

#### 3.3.17 用户注册和登录系统

- **测试用例编号**：Sys-Test-2
- **预置条件**：无用户注册信息
- **测试用例标题**：用户登录功能测试
  - **步骤描述**：用户通过点击登录输入用户名和头像
  - **预期输出**：用户的头像和用户名显示在设置页中
  - **实际输出**：用户的头像和用户名显示在设置页中
  - **执行结果**：Y
- **测试用例编号**：Sys-Test-3

#### 3.3.18 任务管理系统

- **测试用例编号**：Sys-Test-4
- **预置条件**：用户已登录并处于任务管理页面
- **测试用例标题**：创建新任务
  - **步骤描述**：用户在任务管理页面输入任务名称并点击创建任务按钮
  - **预期输出**：任务列表中显示新创建的任务，任务数据存储在静态数据中
  - **实际输出**：任务列表中显示新创建的任务，任务数据存储在静态数据中
  - **执行结果**：Y
- **测试用例编号**：Sys-Test-5
- **预置条件**：用户已登录并存在已创建的任务
- **测试用例标题**：删除任务

- **步骤描述**：用户选择一个已创建的任务，点击删除按钮，并确认删除操作
- **预期输出**：任务列表中不再显示被删除的任务，任务数据从数据库中移除
- **实际输出**：任务列表中不再显示被删除的任务，任务数据从数据库中移除
- **执行结果**：Y

#### 3.3.19 日志系统

- **测试用例编号**：Sys-Test-6
- **预置条件**：用户已登录并进入日志界面保存一些日志
- **测试用例标题**：查看日志
  - **步骤描述**：用户在日志页面查看所有自己写过的日志
  - **预期输出**：日志页面显示所有日志
  - **实际输出**：日志页面显示所有日志
  - **执行结果**：Y

## 3.4 测试结束标准

1. 单元测试用例设计已经通过评审
2. 核心代码100％ 经过Code Review
3. 单元测试功能覆盖率达到100％
4. 单元测试代码行覆盖率不低于80％
5. 所有发现缺陷至少60％都纳入缺陷追踪系统且各级缺陷修复率达到标准
6. 按照单元测试用例完成了所有规定单元的测试
7. 软件单元功能与设计一致

## 3.5 风险管理

| 项目风险 | 风险分析 | 规避方法 | 风险概率 |
| --- | --- | --- | --- |
| **测试未通过** | 如果开发提交的测试版本，执行冒烟测试未通过，测版本必须打回重新提交，这样可能造成计划进度延误，影响后续的测试工作安排。 | 在提交测试版本前，开发应该抽出时间进行自测，如果没有进行单元测试和集成测试，则需要安排进行。如果到期提交的版本被打回，为了不影响整体计划的进度，需要开发人员适当安排增加人手或加班。 | |
| **开发进度延误** | 如果开发到计划规定时间不能按时完成，则造成后续测试工作安排延误，从而造成测试计划执行的风险。 | 请项目经理在项目进行过程中严格把控进度，如有推迟风险请立即通知测试人员，协商解决。如果到期仍然不能按时完成计划，则测试人员需申请修改测试计划，或者申请加班，并通知相关领导。 | |
| **难以修复的权限造成测试用例阻碍** | 如果测试执行过程中，被测试版本发现难以修复的bug，造成被测试模块的功能阻碍无法执行测试，测试进度安排收到影响。 | 出现这样的问题，需要开发人员全力配合测试，及时修改出现的问题。如果不能完全修复，也要给测试提供可以测试被阻碍模块的接口或着可以绕过的方式。 | |

## 3.6 附录

- **参考文档**: 《需求分析报告》