

测试计划



课程名称 软件工程

题目名称 测试计划报告

任课教师 杨 枨

组 号： G18

组 长： 谢子文

组 员： 梁泽生

组 员： 黄 馨

2020 年12月 9 日

**目录**

[1引言 3](#_Toc58673715)

[1.1编写目的 3](#_Toc58673716)

[1.2背景 3](#_Toc58673717)

[1.3定义 4](#_Toc58673718)

[1.4参考资料 4](#_Toc58673719)

[1.5版本历史 4](#_Toc58673720)

[2计划 5](#_Toc58673721)

[2.1软件说明 5](#_Toc58673722)

[2.2测试内容 5](#_Toc58673723)

[2.3测试1（标识符） 5](#_Toc58673724)

[2.3.1进度安排 5](#_Toc58673725)

[2.3.2条件 5](#_Toc58673726)

[2.3.3测试资料 6](#_Toc58673727)

[2.3.4测试培训 6](#_Toc58673728)

[2.4测试2（标识符） 6](#_Toc58673729)

[3测试设计说明 6](#_Toc58673730)

[3.1测试1（标识符） 6](#_Toc58673731)

[3.1.1控制 6](#_Toc58673732)

[3.1.2输入 7](#_Toc58673733)

[3.1.3输出 7](#_Toc58673734)

[3.1.4过程 7](#_Toc58673735)

[3.2测试2（标识符） 7](#_Toc58673736)

[4评价准则 7](#_Toc58673737)

[4.1范围 7](#_Toc58673738)

[4.2数据整理 7](#_Toc58673739)

[4.3尺度 7](#_Toc58673740)

**测试计划（GB8567——88）**

# 1引言

## 1.1编写目的

本测试计划的具体编写目的，指出预期的读者范围。

测试计划是在软件测试中最重要的步骤之一，它在软件开发的前期对软件测试做出清晰，完整的计划，不光对整个测试起到关键性的作用，而且对开发人员的开发工作，整个项目的规划，项目经理的审查都有辅助性作用。

编写测试计划用来定义测试的范围、测试的方法、所需的资源、进度等，明确需要测试的产品项，需要覆盖的功能特性，需要执行的测试任务，每项任务的负责人，识别相关的风险。测试计划编写好后，领导可以根据测试计划做宏观调控，进行相应的资源配置，测试人员能够了解整个项目测试情况以及项目测试不同阶段所要进行的工作，便于其他人员了解测试人员的工作内容，进行有关配合工作，尤其是系统开发人员，要根据测试计划来安排自己的工作，以便测试人员找到Bug后，开发人员可以及时去掉Bug，很好的安排自己的工作。测试计划可以有效地预防计划的风险，保证计划的顺利进行。

## 1.2背景

说明：

1. 测试计划所从属的软件系统的名称；
2. 该开发项目的历史，列出用户和执行此项目测试的计算中心，说明在开始执行本测试计划之前必须完成的各项工作。
3. 待开发软件系统的名称：

专注包子 微信小程序

1. 此项目的任务提出者：

杨枨教授

1. 此项目的开发者：

谢子文、黄馨、梁泽生

1. 此项目的用户：

自制力不强，想要专注学习和工作的人们。

1. 以下列出目标用户：

杨枨教授、软工1802陈骁、法学1802王新、建筑1801王宇键

1. 将运行该软件的计算站（中心）：

腾讯云服务器1核2G

50G云数据库MySQL

GitHub云端版本配置管理

## 1.3定义

列出本文件中用到的专门术语的定义和外文首字母组词的原词组。

1. 单元测试：单元测试（unit testing），是指对软件中的最小可测试单元进行检查和验证。对于单元测试中单元的含义，一般来说，要根据实际情况去判定其具体含义，如C语言中单元指一个函数，Java里单元指一个类，图形化的软件中可以指一个窗口或一个菜单等。总的来说，单元就是人为规定的最小的被测功能模块。单元测试是在软件开发过程中要进行的最低级别的测试活动，软件的独立单元将在与程序的其他部分相隔离的情况下进行测试。
2. 集成测试：集成测试，也叫组装测试或联合测试。在单元测试的基础上，将所有模块按照设计要求（如根据结构图）组装成为子系统或系统，进行集成测试。
3. 系统测试：系统测试，英文是System Testing。是对整个系统的测试，将硬件、软件、操作人员看作一个整体，检验它是否有不符合系统说明书的地方。这种测试可以发现系统分析和设计中的错误。如安全测试是测试安全措施是否完善，能不能保证系统不受非法侵入。再例如，压力测试是测试系统在正常数据量以及超负荷量(如多个用户同时存取) 等情况下是否还能正常地工作。

## 1.4参考资料

列出要用到的参考资料，如：

1. 本项目的经核准的计划任务书或合同、上级机关的批文；
2. 属于本项目的其他已发表的文件；
3. 本文件中各处引用的文件、资料，包括所要用到的软件开发标准。列出这些文件的标题、文件编号、发表日期和出版单位，说明能够得到这些文件资料的来源。

[1] 《GB856T——88》国标文档

[2] 《软件工程导论》清华大学出版社，张海藩、牟永敏编著

[3] 微信小程序平台

<https://mp.weixin.qq.com/cgi-bin/wx>

1. 测试计划的设计与编写

<https://www.cnblogs.com/muzipaopao/p/10730165.html>

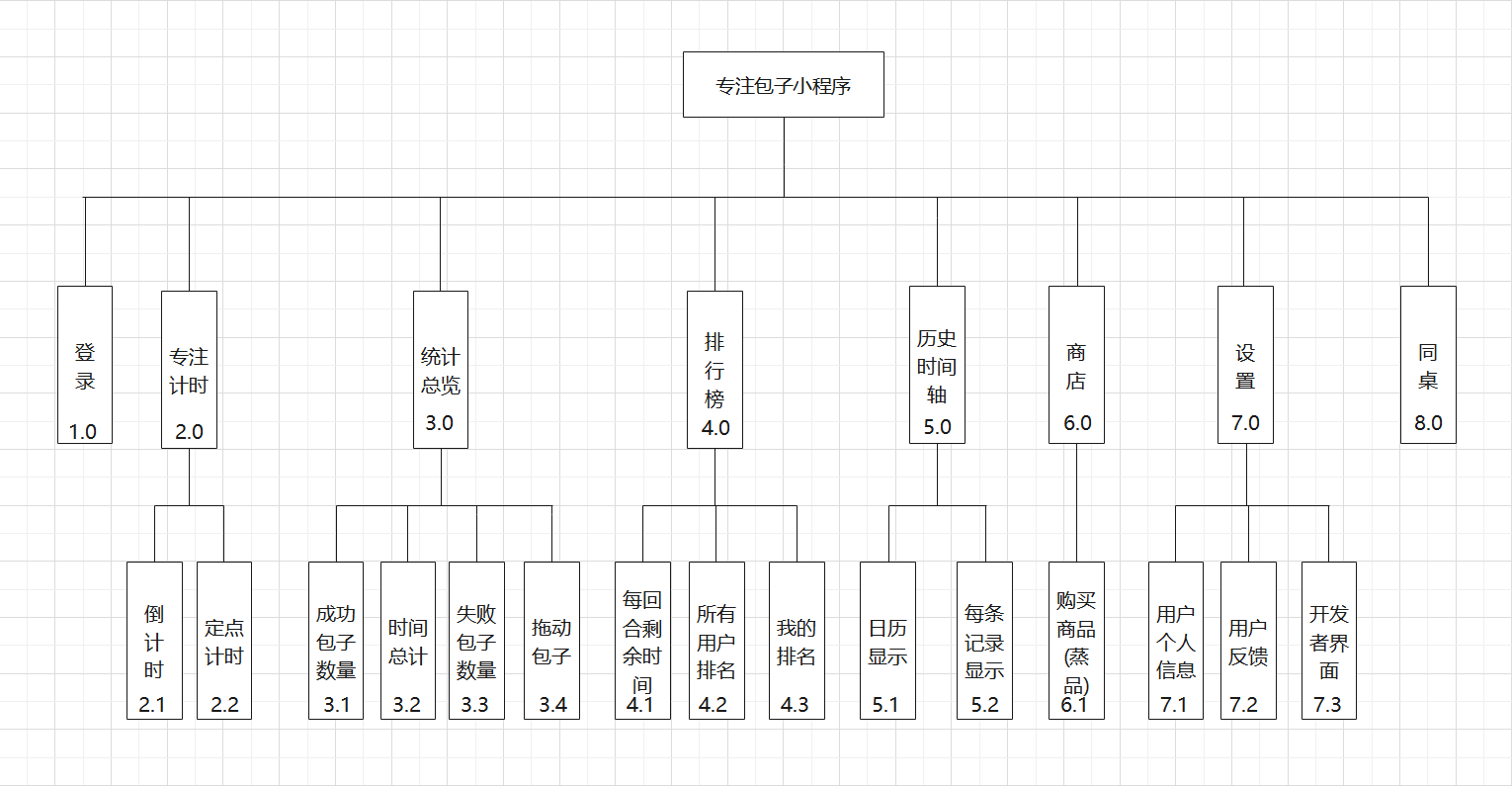
## 1.5版本历史

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 版本修订日期 | 版本修改人 | 是否通过审核 |
| 测试计划v0.1.0 | 2020/12/08 | 谢子文、梁泽生、黄馨 | 是 |
| 测试计划v0.1.1 | 2020/12/08 | 谢子文 | 是 |
| 测试计划v0.1.2 | 2020/12/09 | 黄馨 | 是 |
| 测试计划v0.1.3 | 2020/12/10 | 黄馨 | 是 |
| 测试计划v0.1.4 | 2002/12/27 | 黄馨 | 是 |

# 2计划

## 2.1软件说明

提供一份图表，并逐项说明被测软件的功能、输入和输出等质量指标，作为叙述测试计划的提纲。



## 2.2测试内容

列出组装测试和确认测试中的每一项测试内容的名称标识符、这些测试的进度安排以及这些测试的内容和目的，例如模块功能测试、接口正确性测试、数据文卷存取的测试、运行时间的测试、设计约束和极限的测试等。

### 2.2.1单元测试：

1. 倒计时单元测试
2. 定点计时单元测试
3. 统计总览单元测试
4. 排行榜单元测试
5. 历史时间轴单元测试
6. 用户反馈单元测试
7. 用户个人信息单元测试
8. 开发者页面单元测试
9. 购买蒸品单元测试

### 2.2.2集成测试：

1. 专注计时模块集成测试
2. 统计总览模块集成测试
3. 排行榜模块集成测试
4. 历史时间轴模块集成测试
5. 商店模块集成测试
6. 设置模块集成测试

### 2.2.3系统测试

暂无

### 2.2.4确认测试

参照文件：详细设计文档

参与人员：测试人员、体验用户

测试方法：黑盒测试方法

|  |  |  |
| --- | --- | --- |
| 编号 | 测试名称 | 测试内容 |
| 1 | 功能检验 | 确认小程序的功能是否满足用户的需求 |
| 2 | 性能检验 | 确认小程序的性能是否达到预期要求 |
| 3 | UI检验 | 确认小程序的UI界面用户是否满意 |
| 4 | 安全性检验 | 确认小程序的安全性是否符合要求 |
| 5 | 兼容性检验 | 确认小程序的兼容性是否符合要求 |
| 6 | 可维护性检验 | 确认小程序是否具有可维护性 |

## 2.3单元测试1（标识符）

给出这项测试内容的参与单位及被测试的部位。

### 2.3.1进度安排

给出对这项测试的进度安排，包括进行测试的日期和工作内容（如熟悉环境。培训、准备输入数据等）。

### 2.3.2条件

陈述本项测试工作对资源的要求，包括：

1. 设备所用到的设备类型、数量和预定使用时间；
2. 软件列出将被用来支持本项测试过程而本身又并不是被测软件的组成部分的软件，如测试驱动程序、测试监控程序、仿真程序、桩模块等等；
3. 人员列出在测试工作期间预期可由用户和开发任务组提供的工作人员的人数。技术水平及有关的预备知识，包括一些特殊要求，如倒班操作和数据键入人员。

### 2.3.3测试资料

列出本项测试所需的资料，如：

1. 有关本项任务的文件；
2. 被测试程序及其所在的媒体；
3. 测试的输入和输出举例；
4. 有关控制此项测试的方法、过程的图表。

### 2.3.4测试培训

说明或引用资料说明为被测软件的使用提供培训的计划。规定培训的内容、受训的人员及从事培训的工作人员。

## 2.4测试2（标识符）

用与本测试计划2.3条相类似的方式说明用于另一项及其后各项测试内容的测试工作计划。

# 3集成测试设计说明

## 3.1集成测试1（专注计时）

采用路径覆盖法。

### 3.1.1控制

控制方式：人工

并对可能出现的结果进行记录。

### 3.1.2输入

测试用例1：（设定专注倒计时时间）

输入：选择时间为10分钟

测试用例2：（设定定向计时时间）

输入：选择时间为23:07（现在是23:04）

测试用例3：（根据设定的时间开始专注计时）

输入：点击开始按钮

### 3.1.3输出

测试用例1：

输出：显示10分钟

测试用例2：

输出：显示23:07

测试用例3：

输出：专注计时开始

## 3.1.4过程

倒计时模块：

1.设定专注时间，例如10分钟，然后开始专注计时。

2.关闭手机或者退出微信app。

3.打开手机回到微信发现计时仍在进行。

4.继续计时下去知道时间结束或者提前结束这个番茄钟。

定向计时模块：

1.设定定点时间，例如现在的时间是23:04，我设为23:07结束，然后开始专注计时。

2.关闭手机或者退出微信app。

3.打开手机回到微信发现计时仍在进行。

4.继续计时下去知道时间结束或者提前结束这个番茄钟。

## 3.2集成测试2（统计总览/蒸笼）

采用路径覆盖法。

### 3.2.1控制

控制方式：人工

并对可能出现的结果进行记录。

### 3.2.2输入

测试用例1：（进入蒸笼界面）

输入：点击菜单栏蒸笼按钮

测试用例2：（拖动包子）

输入：按住包子（小球）可以进行拖动

### 3.2.3输出

输出用例1：

输出：显示蒸笼界面

输出用例2：

输出：包子被拖动

## 3.2.4过程

1.点击菜单栏蒸笼按钮。

2.进入页面后，切换月份，显示不同时间的成功失败包子数量和专注时长。

3.按住包子可以进行拖动。

## 3.3集成测试3（排行榜）

采用路径覆盖法。

### 3.3.1控制

控制方式：人工

并对可能出现的结果进行记录。

### 3.3.2输入

测试用例1：（进入菜单栏排行榜）

输入：点击菜单栏排行榜按钮

### 3.3.3输出

测试用例1：

输出：显示用户排行榜，显示距离本回合结束的剩余时间。

## 3.3.4过程

1.点击菜单栏排行榜按钮。

2.进入页面后，显示各个用户的排行榜，还有用户自己的排行榜名次等信息。

## 3.4集成测试4（历史时间轴）

采用路径覆盖法。

### 3.4.1控制

控制方式：人工

并对可能出现的结果进行记录。

### 3.4.2输入

测试用例1：（进入历史时间轴页面）

输入：点击菜单栏历史时间轴按钮

测试用例2：（显示不同日期的专注记录）

输入：点击日历上的其他日期

### 3.4.3输出

测试用例1：

输出：显示历史时间轴页面

测试用例2：

输出：显示不同日期的专注记录。

## 3.4.4过程

1.点击菜单栏历史时间轴按钮。

2.进入页面后，显示出一个很大的日历表，当天可以查看当天的专注记录。

3.用户点击日历表上其他天，不同的日期下方会显示该日期当天的专注记录。

## 3.5集成测试5（商店）

采用路径覆盖法。

### 3.5.1控制

控制方式：人工

并对可能出现的结果进行记录。

### 3.5.2输入

测试用例1：

输入：点击菜单栏商店按钮

测试用例2：

输入：选择某个商品，点击购买

### 3.5.3输出

测试用例1：

输出：显示商店界面

测试用例2：

输出：购买成功

## 3.5.4过程

1.点击菜单栏商店按钮。

2.进入页面后，显示各个商品（蒸品）的信息和价格。

3.点击某个商品，可进行购买。

## 3.6集成测试6（设置）

采用路径覆盖法。

### 3.6.1控制

控制方式：人工

并对可能出现的结果进行记录。

### 3.6.2输入

测试用例1：

输入：点击菜单栏设置按钮

测试用例2：

输入：点击头像那一栏（进入个人中心）

测试用例3：

输入：点击用户反馈

测试用例4：

输入：点击开发者界面

### 3.6.3输出

测试用例1：

输出：显示设置界面

测试用例2：

输出：显示个人中心界面

测试用例3：

输出：可进行用户反馈

测试用例4：

输出：显示开发者管理界面

## 3.6.4过程

1.点击菜单栏设置按钮。

2.进入页面后，会显示个人信息、用户反馈等事情。

3.点击自己头像那一栏进入个人中心，出现个人中心页面。

4.点击用户反馈按钮，可进行用户反馈。

5.实现开发者界面。

# 4系统测试计划

# 5用户测试及反馈

体验用户：法学1802 王新

体验时间：2021.01.02

用户反馈：界面不够精美，可以尽量完善，人机交互设计不够完美需要改善，功能较为齐全，只可惜有些还未实现，总体还行。

体验用户：建筑1801 王宇键

体验时间：2021.01.02

用户反馈：页面设计可以再具有层次性一些，界面的功能还可以更丰富一下，小程序已经初具雏形，可以继续优化细节贴近用户使用。

体验用户：软工1802 陈骁

体验时间：2021.01.02

用户反馈：页面设计可以再具有层次性一些，界面的功能还可以更丰富一下，小程序已经初具雏形，可以继续优化细节贴近用户使用。

# 6确认测试

|  |  |  |
| --- | --- | --- |
| **编号** | **测试名称** | **测试内容** |
| 1 | 功能检验 | 确认小程序的功能是否满足用户的需求 |
| 2 | 性能检验 | 确认小程序的性能是否达到预期要求 |
| 3 | UI检验 | 确认小程序的UI界面用户是否满意 |
| 4 | 安全性检验 | 确认小程序的安全性是否符合要求 |
| 5 | 兼容性检验 | 确认小程序的兼容性是否符合要求 |
| 6 | 可维护性检验 | 确认小程序是否具有可维护性 |

# 7评价准则

## 7.1范围

说明所选择的测试用例能够接查的范围及其局限性。

本计划主要针对单元测试。暂无不接受测试的特性和功能。

风险或意外事件：

1.代码在测试时不小心丢失了

2.测试完后数据没有及时保存

3.测试人员因为粗心大意而漏测错测

（描述测试的各个阶段，例如：单元测试、集成测试或系统测试，并说明本计划所针对的测试类型（如功能测试或性能测试）。简要地列出测试对象中将接受测试或将不接受测试的那些特性和功能。

列出可能会影响测试设计、开发或实施的所有风险或意外事件。 列出可能会影响测试设计、开发或实施的所有约束。）

## 7.2数据整理

陈述为了把测试数据加工成便于评价的适当形式，使得测试结果可以同，已知结果进行比较而要用到的转换处理技术，如手工方式或自动方式；如果是用自动方式整理数据，还要说明为进行处理而要用到的硬件、软件资源。

手动记录数据并且整合到一个文档中。

## 7.3尺度

说明用来判断测试工作是否能通过的评价尺度，如合理的输出结果的类型、测试输出结果与预期输出之间的容许偏离范围、允许中断或停机的最大次数。

（1）输出结果的类型合理；

（2）输出结果与预期偏差允许有微量偏移；