“一族一谱”管理系统

测试文档

小组：软件工程第六组

组员：王聪 曾德龙 蔡楷欣 曾欣

程钰涵 陈冠旭 郑湘萍

版次： v1.5

日期：2021年5月15日

文档修改历史记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本号 | 修改人 | 日期 | 修改内容 | 备注 |
| v1.0 | 郑湘萍 | 2021.5.20 | 活动部分的测试 | 初始文档内容 |
| V1.1 | 陈冠旭 | 2021.5.24 | 添加有关出错处理的细节 | 初始文档内容 |
| v1.2 | 曾欣 | 2021.5.26 | 进行首页部分的测试 | 初始文档内容 |
| V1.3 | 王聪 | 2021.5.30 | 进行族谱书的测试 | 初始文档内容 |
| V1.4 | 曾欣 | 2021.6.10 | 首页部分测试调试，对修复结果再进行更新 |  |
| V1.5 | 曾德龙 | 2021.6.15 | 将白盒测试并入报告 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1 小程序部分测试方案 4](#_Toc76571201)

[1.1 黑盒测试部分 4](#_Toc76571202)

[1.1.1 首页部分 4](#_Toc76571203)

[1.1.2 成书部分 22](#_Toc76571204)

[1.1.3 活动部分 26](#_Toc76571205)

[1.2 白盒测试部分 32](#_Toc76571206)

[1.2.1统计人数的函数测试 33](#_Toc76571207)

[1.2.2年份比较函数的测试 35](#_Toc76571208)

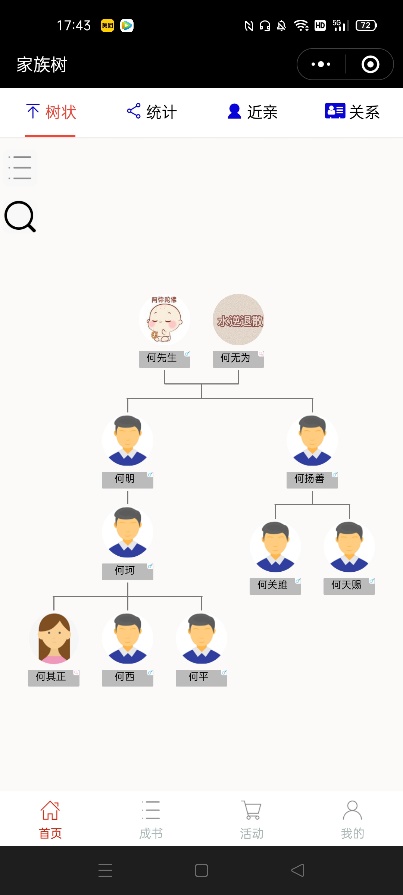
[1.2.3搜索函数的测试 36](#_Toc76571209)

1 小程序部分测试方案

1.1 黑盒测试部分

1.1.1 首页部分

首页部分划分为树状图显示部分，统计家族信息部分，近亲显示图部分以及关系查询部分。整体效果如图显示。



在首页部分应实现的功能如下：

表1 功能需求对照表

|  |  |  |
| --- | --- | --- |
| 功能需求 | 需求说明 | 实现效果 |
| 匹配族谱 | 用户通过个人姓名在族谱中进行搜索，再根据个人生日信息实现族谱中的定位。 | 注册个人信息后能够有效的在族谱树状图，近亲表中定位，同时可以进行个人姓名的搜索。 |
| 查看族谱信息 | 用户可读取并查看存储在微信小程序端数据库的部分族谱信息。 | 族谱树状图，统计家族信息以及近亲关系树状图等显示信息完整 |
| 申请修改 | 用户能提交对自己以及他人的修改申请 | 用户可以提交修改，提交信息在后台能够显示 |

基本功能均已实现，详细测试计划方案如下：

1. 功能显示部分

**测试方案：**

|  |  |  |
| --- | --- | --- |
| 测试编号 | 所属模块 | 测试方案 |
| 1 | 树状图 | 点击不同的用户信息测试是否显示完整 |
| 2 | 树状图 | 测试数据库中存在的根节点列表是否显示完整 |
| 3 | 统计 | 不同的用户登录，测试统计信息是否正确显示 |
| 4 | 近亲图 | 不同的用户登录，测试是否正确显示包含本人在内的三代信息(上三代和下三代) |

**测试过程：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 1 | 点击“何先生”查看信息 |  | 通过 |  |
| 点击“何其正”查看信息 |  | 通过 |  |
| 2 | 点击“根节点”查看列表显示 |  | 通过 |  |
| 3 | 对于“何先生”为根节点树，查看统计信息 |  | 通过 |  |
| 3 | 对于“何冲之”为根节点树，查看统计信息 |  | 不通过 | 1 |
| 4 | 用户作为“0”代用户，查看近亲关系 |  | 不通过 | 2 |

|  |  |  |
| --- | --- | --- |
| Bug编号 | 问题描述 | 原因分析 |
| 1 | 何冲之作为根节点时，统计关系图没有进行相应的更新 | 统计图是对后台所有数据进行统计的，并没单独针对某一支系 |
| 2 | 用户作为初代登录时，近亲关系图显示不完全 | 未对相应情况进行判断（已修复） |

**修复后的测试：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试Bug编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 2 | 用户作为“0”代用户，查看近亲关系 |  | 通过 |  |

1. 搜索部分：

**测试方案：**

|  |  |  |
| --- | --- | --- |
| 测试编号 | 所属模块 | 测试方案 |
| 1 | 树状图 | 输入不同的搜索对象，查看搜索功能是否正常 |

**测试过程：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 1 | 搜索“何无为” |  | 通过 |  |
| 1 | 搜索“小小” |  | 不通过 | 1 |
| 1 | 搜索“12345  6789012580%” |  | 不通过 | 2 |

|  |  |  |
| --- | --- | --- |
| Bug编号 | 问题描述 | 原因分析 |
| 1 | 输入不存在数据库中的姓名时，没有搜索失败等提示 | 提示框没有进行分类  （已修复） |
| 2 | 输入不符合规范，没有相应的提示说明 | 没有对输入的姓名进行有效行判别（已修复） |

**修复后的测试：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试Bug编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 1 | 搜索“宝贝” |  | 通过 |  |
| 2 | 搜索“888986” |  | 通过 |  |

1. 申请修改请求部分

**测试方案：**

|  |  |  |
| --- | --- | --- |
| 测试编号 | 所属模块 | 测试方案 |
| 1 | 树状图 | 输入不同的添加信息请求，看是否能在Bomb后台apply表中显示 |
| 2 | 树状图 | 输入不同的修改信息请求，查看是否能在Bomb后台apply表中显示 |
| 3 | 树状图 | 删除信息请求是否能在Bomb后台apply表中显示 |

**测试过程：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 1 | 在“何无为”上添加配偶 |  | 不通过 | 1 |
| 1 | 在“何无为”上添加子女 |  | 通过 |  |
| 1 | 在”何明”上添加配偶 |  | 通过 |  |
| 1 | 添加配偶名字为“何” |  | 通过 |  |
| 1 | 配偶名字23456789… | 无法提交 | 通过但希望有更多的提示 |  |
| 1 | 添加不符合逻辑的子女年龄 |  | 通过 |  |
| 1 | 添加同性配偶 |  | 通过 |  |
| 2 | 修改“何无为”姓名为“何太” | 姓名有改动，但是此时性别为男 | 不通过 | 2 |
| 2 | 修改“何无为”姓名为“何\_太” | 姓名成功修改为”何\_太” | 不通过 | 3 |
| 2 | 修改“何无为”性别为男 | 性别修改为男 | 不通过 | 4 |
| 2 | 修改“何平“生日为1900年 | 申请提交年份为1900年 | 不通过 | 5 |
| 3 | 销毁某人信息 | 数据库相应信息栏消息为空 | 通过 |  |

|  |  |  |
| --- | --- | --- |
| Bug编号 | 问题描述 | 原因分析 |
| 1 | 错误添加配偶应提示“不能一夫多妻” |  |
| 2 | 编辑时性别默认是男，但对于女性用户，仅修改名字时也会将性别设置为男。 | 不输入性别时未进行相应的判断。（已修复） |
| 3 | 输入非法名时没有相应提示 | 目前名字满足2-6个就算合法 |
| 4 | 将有配偶的一方的性别修改之后，应该有相应的提示 | 未对相关信息进行判断（已修复） |
| 5 | 将子女的生日修改的比父辈还大，应该有相应提示 | 未对相关信息进行判断（已修复） |

**修复后的测试：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试Bug编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 2 | 对女性用户仅修改姓名 |  | 通过（修复后所有选项均需填写，没有默认信息） |  |
| 4 | 修改有配偶一方的性别 |  | 通过 |  |
| 5 | 修改子辈年龄使其比父辈大 |  | 通过 |  |

1. 关系查询部分

**测试方案：**

|  |  |  |
| --- | --- | --- |
| 测试编号 | 所属模块 | 测试方案 |
| 1 | 关系查询 | 输入的对象组合查看是否符合关系 |

**测试过程：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 1. | 何天赐与何西的关系 |  | 通过 |  |
| 1 | 何扬善与何西的关系 |  |  |  |
| 1 | 何西和何先生的关系 |  | 通过 |  |
| 1 | 何珂与何西的关系 |  | 通过 |  |
| 1 | 何平与何西的关系 |  | 通过 |  |
| 1 | 何先生与何冲之的关系 |  | 不通过 | 1 |
| 1 | 何无为与何冲之的关系 |  | 不通过 | 2 |

|  |  |  |
| --- | --- | --- |
| Bug编号 | 问题描述 | 原因分析 |
| 1 | 何冲之没有出现在树状图上，是另一个根节点 | 未对不在同一颗树上的两个结点进行分类（已修复） |
| 2 | 何先生和何无为是夫妻关系，而又分别与何冲之是弟弟妹妹关系，属于近亲结婚 | 同上 |

**修复后的测试：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试Bug编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 1、2 | 关系查询不同分支的近亲关系 |  | 通过 |  |

1.1.2 成书部分

成书部分的展示效果如下所示，该模块实现了目录查找、翻页、增大减小字体等功能。



在成书部分应实现的功能如下：

表2 成书功能需求对照表

|  |  |  |
| --- | --- | --- |
| 功能需求 | 需求说明 | 实现效果 |
| 显示族人 | 将所有的家族成员的信息汇编成书，该书就称为族谱书 | 显示族人的基本信息，姓名、性别、年龄、婚配和出生日期 |
| 翻页 | 可以快捷的查看族谱书的内容，并且不保证出错 | 对族谱书实现了翻页，并且可以通过目录进行跳转 |
| 修改字体 | 为了方便查看族谱书，最好能够方便的调整族谱书中字体的大小 | 实现增加、减小字体的大小 |
| 目录 | 可以查看整个族谱书的内容，最好有目录功能 | 实现目录功能，可以快捷的查看族谱书的所有的族人的名字 |

从上表中可以看出，基本功能均已实现，详细测试计划方案如下：

**测试方案：**

|  |  |  |
| --- | --- | --- |
| 测试编号 | 所属模块 | 测试方案 |
| 1 | 族谱书 | 检查第一页、最后一页和随机一页是否显示正常 |
| 2 | 族谱书 | 检查目录的正确性 |
| 3 | 族谱书 | 检查字体缩放功能是否正常 |

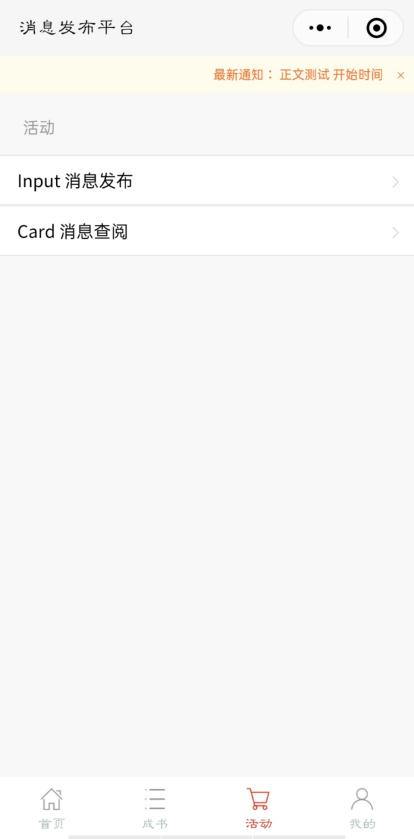
**测试过程：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 1 | 显示第一页族谱书 |  | 通过 |  |
| 显示最后一页族谱书内容 |  | 通过 |  |
| 随机显示一页族谱书 |  | 通过 |  |
| 2 | 检查目录的正确性 |  | 通过 |  |
| 3 | 将字体变大 |  |  |  |
| 将字体变小 |  |  |  |

|  |  |  |
| --- | --- | --- |
| Bug编号 | 问题描述 | 原因分析 |
| 0 | 无 |  |

1.1.3 活动部分

整体效果如图显示。活动部分划分为消息发布模块和消息查阅模块，整体效果如图所示。



在活动部分应实现的功能如下：

表3 功能需求对照表

|  |  |  |
| --- | --- | --- |
| 功能需求 | 需求说明 | 实现效果 |
| 消息发布 | 管理员能对消息每一项具体内容（标题、正文、起始和结束时间，举办方，举办地点）进行规范编辑并实现发布 | 规范输入消息内容并点击发布之后，在小程序顶端会有弹窗显示，并将消息同步到各个用户的列表中去 |
| 消息查阅 | 管理员能够查看简略消息列表并点击查看具体消息内容 | 对历史消息能够进行简略的查询 |

基本功能均已实现，详细测试计划方案如下：

**测试方案：**

|  |  |  |
| --- | --- | --- |
| 测试编号 | 所属模块 | 测试方案 |
| 1 | 消息发布 | 顺序输入消息中的每一项内容，从输入为空至输入完整的每一种编辑情况均进行测试 |
| 2 | 消息发布 | 保证其他输入完整规范的情况下，对正文内容字数的不同情况进行测试 |
| 3 | 消息发布 | 保证在其他输入完整规范的情况下，分别使消息中的标题、正文、时间、地方、举办方输入为空，测试输出是否正确 |
| 4 | 消息发布 | 对输入模块中的起始和结束时间进行逻辑错误测试 |
| 5 | 消息发布 | 测试历史消息是否能正确显示 |
| 6 | 消息查询 | 测试消息列表是否能正确显示 |
| 7 | 消息查询 | 测试具体消息内容是否正确显示 |

**测试过程：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 1 | 输入为空 | 输出提示“输入信息不完整，请检查输入” | 不通过 | 1 |
| 仅输入标题“这是一个标题” | 输出提示“输入信息不完整，请检查输入” | 不通过 |
| 仅输入标题“这是一个标题”，正文“这是正文内容” | 输出提示“输入信息不完整，请检查输入” | 不通过 |
| 仅输入标题“这是一个标题”，输入正文“这是正文”，输入起始时间“2021-6-22-09：20” | 输出提示“输入信息不完整，请检查输入” | 不通过 |
| 仅输入标题“这是一个标题”，输入正文“这是正文”，输入起始时间“2021-6-22-09：20”，输入结束时间“2021-6-22-10：20” | 输出提示“输入信息不完整，请检查输入” | 不通过 |
| 仅输入标题“这是一个标题”，输入正文“这是正文”，输入起始时间“2021-6-22-09：20”，输入结束时间“2021-6-22-10：20”，输入举办地点“某某村庄” | 输出提示“输入信息不完整，请检查输入” | 不通过 |
| 输入标题“这是一个标题”，输入正文“这是正文”，输入起始时间“2021-6-22-09：20”，输入结束时间“2021-6-22-10：20”，输入举办地点“某某村庄”，输入举办方“某某村委会” | 消息成功发布  321417d83cf8b386e9dee2e6ddfec15 | 通过 |  |
| 2 | 输入正文：“这是正文，小于200字”，其他输入按如上规范输入 | 消息成功发布  e24105dbb193b7b6e18407397f5a76e | 通过 |  |
| 输入正文：“这是正文，文文文文文...刚好200字”（使正文内容达到200字），其他输入按如上规范输入 | 消息成功发布  c745f179f4171566273c480ffa959b1 | 通过 |  |
| 输入正文“这是正文，文文文文文...超过200字”（使正文内容超过200字），其他输入按如上规范输入 | 消息发布，但超过200字的内容被删除  fd8cc2a56186cb3e817edc7c79fba25 | 通过 |  |
| 3 | 其他输入按如上规范输入，输入标题为空 | 输出提示“输入信息不完整，请检查输入” | 不通过 | 1 |
| 其他输入按如上规范输入，输入正文为空 | 输出提示“输入信息不完整，请检查输入” | 不通过 | 1 |
| 其他输入按如上规范输入，输入起始时间为空 | 消息发布，但是缺乏起始时间显示  8c4ec1930bf288ea92a93cf64b03ee9 | 不通过 | 2 |
| 其他输入按如上规范输入，输入结束时间为空 | 输出提示“输入信息不完整，请检查输入” | 不通过 | 1 |
| 其他输入按如上规范输入，输入举办地点为空 | 输出提示“输入信息不完整，请检查输入” | 不通过 | 1 |
| 其他输入按如上规范输入，输入举办方为空 | 输出提示“输入信息不完整，请检查输入” | 不通过 | 1 |
| 4 | 输入起始时间“2021-6-22-09：20”，输入结束时间“2021-6-22-08：00”，其他输入按如上规范输入 | 002ffcbf234c36159a44a4601385e30 | 通过 |  |
| 输入起始时间“2021-6-22-09：20”，输入结束时间“2021-6-22-09：20”，其他输入按如上规范输入 | 002ffcbf234c36159a44a4601385e30 | 通过 |  |
| 5 | 点击历史消息 | 成功跳转并显示列表  0360b0e097ded583429db10c85021fb | 通过 |  |
| 6 | 点击消息列表 | 成功跳转并显示列表  0360b0e097ded583429db10c85021fb | 通过 |  |
| 7 | 点击其中一条简略消息“测试” | 成功跳转并显示消息具体内容  00b48c6183fdda621974b8338430c88 | 通过 |  |

|  |  |  |
| --- | --- | --- |
| Bug编号 | 问题描述 | 原因分析 |
| 1 | 不管消息中的哪一项为空，输出均只提示输入不完整，没有进行具体问题描述 | 没有对各个输入设置优先级检查 |
| 2 | 当消息中起始时间未进行设置时，缺乏时间信息的消息能成功发布 |  |

对测试编号2、3根据所出现的bug进行对应的修改，再行测试结果如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试编号 | 测试用例内容 | 测试结果 | 是否通过 | Bug编号 |
| 1 | 输入为空 | b978d501ae45d90a06105b15afe2646 | 通过 |  |
| 仅输入标题“这是一个标题” | f57b95dc77dbcb4298d61f4b578afbf | 通过 |  |
| 仅输入标题“这是一个标题”，正文“这是正文内容” | 1a80f529f59323ae929151342de960f | 通过 |  |
| 仅输入标题“这是一个标题”，输入正文“这是正文”，输入起始时间“2021-6-22-09：20” | 78f7ea8d7e2e4b25b1d8d3ec2ec1ea9 | 通过 |  |
| 仅输入标题“这是一个标题”，输入正文“这是正文”，输入起始时间“2021-6-22-09：20”，输入结束时间“2021-6-22-10：20” | b20509596f5d8e5a3c97d0423e77cb4 | 通过 |  |
| 仅输入标题“这是一个标题”，输入正文“这是正文”，输入起始时间“2021-6-22-09：20”，输入结束时间“2021-6-22-10：20”，输入举办地点“某某村庄” | 953669f6a7a330cb6ae7d8bfc50616e | 通过 |  |
| 仅输入标题“这是一个标题”，输入正文“这是正文”，输入起始时间“2021-6-22-09：20”，输入结束时间“2021-6-22-10：20”，输入举办地点“某某村庄”，输入举办方“某某村委会” | 消息成功发布  321417d83cf8b386e9dee2e6ddfec15 | 通过 |  |
| 3 | 其他输入按如上规范输入，输入标题为空 | b978d501ae45d90a06105b15afe2646 | 通过 |  |
| 其他输入按如上规范输入，输入正文为空 | f57b95dc77dbcb4298d61f4b578afbf | 通过 |  |
| 其他输入按如上规范输入，输入起始时间为空 | 1a80f529f59323ae929151342de960f | 通过 |  |
| 其他输入按如上规范输入，输入结束时间为空 | 78f7ea8d7e2e4b25b1d8d3ec2ec1ea9 | 通过 |  |
| 其他输入按如上规范输入，输入举办地点为空 | b20509596f5d8e5a3c97d0423e77cb4 | 通过 |  |
| 其他输入按如上规范输入，输入举办方为空 | 953669f6a7a330cb6ae7d8bfc50616e | 通过 |  |

1.2 白盒测试部分

小程序端的白盒测试主要在小程序的体验版发布之前，在函数部署到小程序前对函数进行白盒测试，因为函数尚未部署到小程序，所以只用于对一些与组件交互少的函数进行测试，与组件交互多的我们在黑盒测试完成，白盒测试我们主要使用到的工具使node.js的jest模块，具体如下：

1.2.1统计人数的函数测试

函数实现原理是利用两个数组存储男女的各个年龄段的人数，比如person[0]是男性各个年龄段的人数，其中数组长度为7，分别代表出生年份在['1960以前', '1960-69', '1970-79', '1980-89', '1990-1999', '2000-2009', '2010至今']的人数。根据路径覆盖给出如下用例：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 测试用例 | | 判定 | | 条件取值 | | 覆盖标准 |
| 年份 | 性别 | 数组位置 | 子数组位置 | person[0] | person[1] |
| 1 | 1959 | 男 | 0 | 0 | [1,0,0,0,0,0,0] | [0,0,0,0,0,0,0] | 路径覆盖 |
| 2 | 1959 | 女 | 1 | 0 | [0,0,0,0,0,0,0] | [1,0,0,0,0,0,0] |
| 3 | 1960 | 男 | 0 | 1 | [0,1,0,0,0,0,0] | [0,0,0,0,0,0,0] |
| 4 | 1960 | 女 | 1 | 1 | [0,0,0,0,0,0,0] | [0,1,0,0,0,0,0] |
| 5 | 1970 | 男 | 0 | 2 | [0,0,1,0,0,0,0] | [0,0,0,0,0,0,0] |
| 6 | 1970 | 女 | 1 | 2 | [0,0,0,0,0,0,0] | [0,0,1,0,0,0,0] |
| 7 | 1980 | 男 | 0 | 3 | [0,0,0,1,0,0,0] | [0,0,0,0,0,0,0] |
| 8 | 1980 | 女 | 1 | 3 | [0,0,0,0,0,0,0] | [0,0,0,1,0,0,0] |
| 9 | 1990 | 男 | 0 | 4 | [0,0,0,0,1,0,0] | [0,0,0,0,0,0,0] |
| 10 | 1990 | 女 | 1 | 4 | [0,0,0,0,0,0,0] | [0,0,0,0,1,0,0] |
| 11 | 2000 | 男 | 0 | 5 | [0,0,0,0,0,1,0] | [0,0,0,0,0,0,0] |
| 12 | 2000 | 女 | 1 | 5 | [0,0,0,0,0,0,0] | [0,0,0,0,0,1,0] |
| 13 | 2010 | 男 | 0 | 6 | [0,0,0,0,0,0,1] | [0,0,0,0,0,0,0] |
| 14 | 2010 | 女 | 1 | 6 | [0,0,0,0,0,0,0] | [0,0,0,0,0,0,1] |

测试结果为：



图2.1 统计函数第一次测试结果图

用例一和二出错，出错结果为

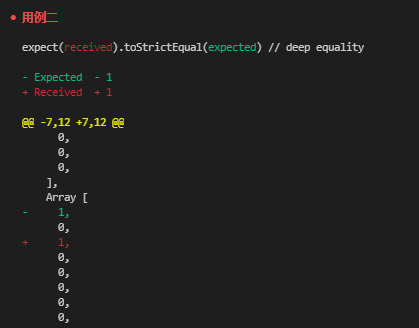
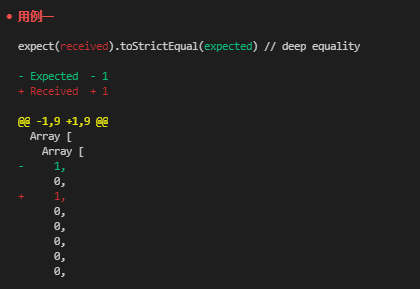


图2.2 统计函数测试出错图

出错形式：用例一二中1959年出生的人被归到了1960-1969出生了

出错原因：统计时通过计算年份与1960差几个10年来判断在数组中的位置。即

然后如果idx小于0时令idx为0，idx大于6时使idx为6，涉及取整操作，这里应该使用往下取整，但是这里采用的时js提供的Math.trunc函数，它是通过舍去小数位取整的，这样当year = 1959时，idx 本应该为 -1 + 0 = -1，这里变成0 + 1 = 1。

修改方式：先不取整，判断idx小于0的话令idx 为 0，否则对idx取整判断是否大于6。

改正后测试结果通过：

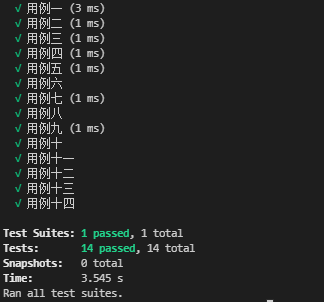


图2.3 统计函数第二次测试

1.2.2年份比较函数的测试

该函数的功能是实现输入两个出生日期，比较谁年纪更大。其流程如下：

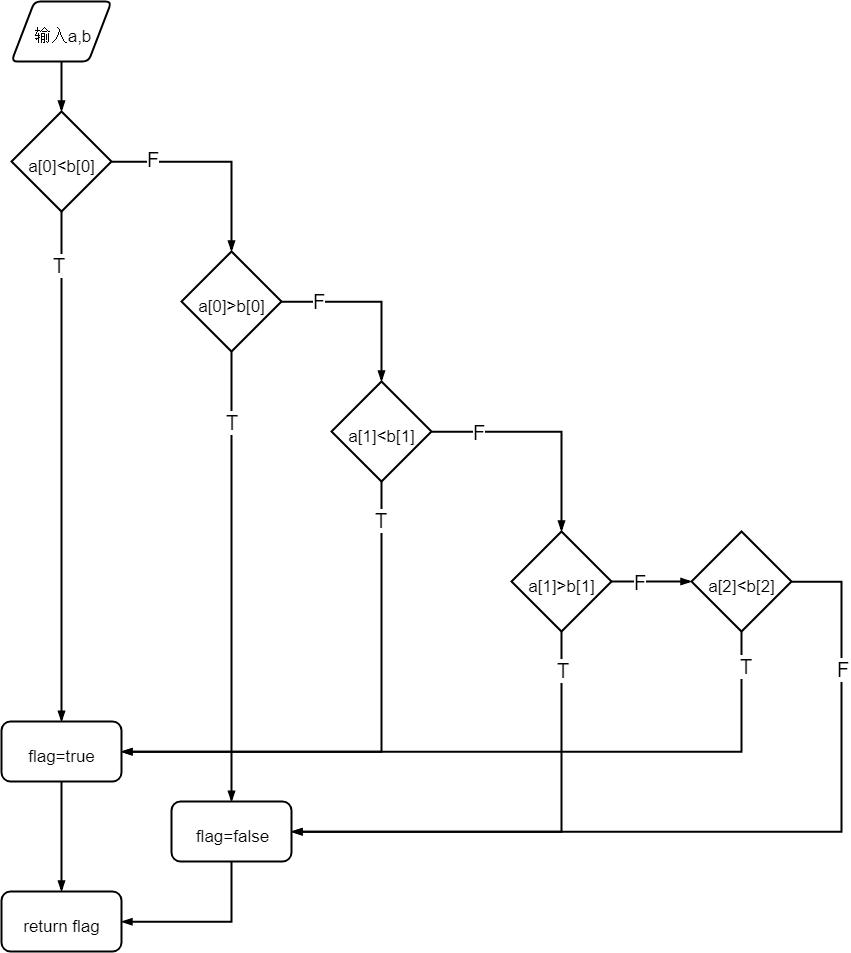


图2.4 年龄比较流程图

为此设计如下测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 输入数据 | | 期望输出 | 覆盖标准 |
| 序号 | A（日期1） | B（日期2） | Flag（A是否比B早） |
| 1 | 2021-08-10 | 2021-08-09 | false | 路径覆盖 |
| 2 | 2021-10-10 | 2029-10-31 | True |
| 3 | 2010-12-01 | 2000-03-23 | false |
| 4 | 2019-11-11 | 2019-11-11 | false |
| 5 | 2016-10-05 | 2016-10-14 | False |
| 6 | 2015-05-01 | 2015-05-09 | True |
| 7 | 2015-05-10 | 2015-04-20 | False |
| 8 | 2010-05-10 | 2010-08-05 | True |
| 9 | 2008-08-08 | 2002-02-20 | False |
| 10 | 2009-09-09 | 2010-12-12 | True |

测试结果为：

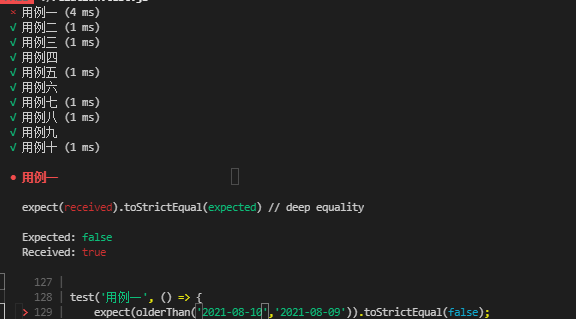


图2.5 年份比较第一次测试结果图

对代码进行审查，发现当同年同月时用了第一个人的月份去比另一个人的年份（因为年月日存储为一个三元数组，数组下标打错了）：

return (date1[1] < date2[2]);

修正后重新测试，结果如下：

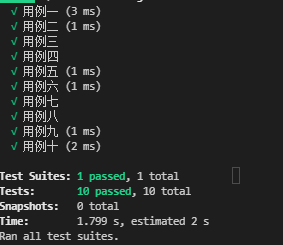


图2.6 年份比较第二次测试结果图

1.2.3搜索函数的测试

该函数的功能是实现获取用户在搜索框中的输入，然后从数据库中搜索相关人名（在函数中提出了用户人名的数组），其流程如下：

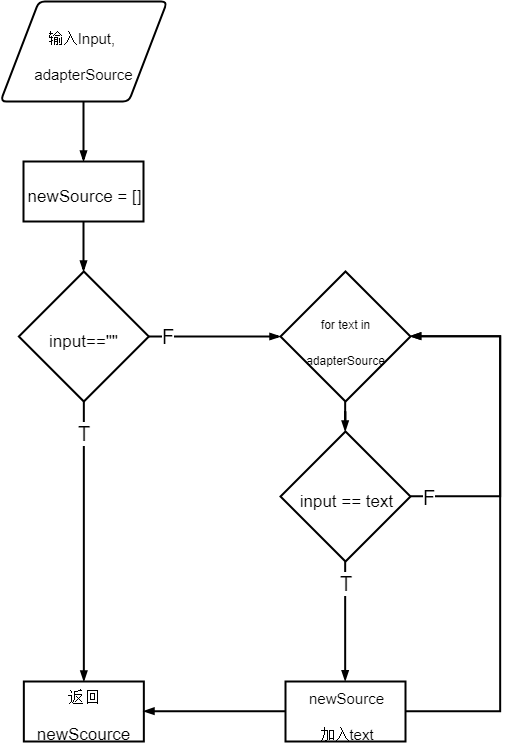


图2.7 搜索流程图

为此设计如下测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 输入数据 | | 预期输出 | 覆盖标准 |
| 序号 | input（字符串） | adapterSource)（字符串数组） | newSource（输出匹配的字符串数组） |
| 1 | “a” | [“a”, “b”, “c”, “abc”] | [“a”, “abc”] | 路径覆盖 |
| 2 | “abc” | [“abc”, “cafbdsaf”, “fesaffdsf”, “fweaf”] | [“abc”, “cafbdsaf”, “fesaffdsf”, “fweaf”] |
| 3 | “name” | [“fsdafe”, “ewrew”] | [“fsdafe”, “ewrew”] |
| 4 | “” | [“fsafsaf”, “fdfe”] | [“fsafsaf”, “fdfe”] |
| 5 | “” | [“”] | [“”] |
| 6 | “@@@” | [“@”] | [“@”] |
| 7 | “#@$%” | [“#@@@Q”, “DSD”] | [“#@@@Q”] |
| 8 | “LOVE” | [“KIDDING”, “love”] | [“KIDDING”] |
| 9 | “1234” | [“12”, “33333”, “123321”, “4444”] | [“12”, “33333”, “123321”, “4444”] |
| 10 | “...” | [“????”, “[]”, “.”, “&&”] | [“.”] |

测试结果为：

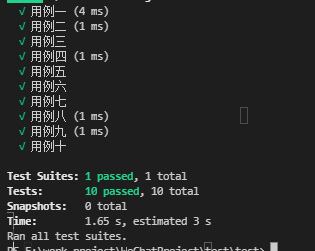


图2.8 搜索函数测试结果图

测试通过。