

CheeseBox 测试文档

课程: 软件工程

软件名称: CheeseBox 知识管理软件

组名: 盒子裡的猫

0. 测试环境

1. Linux(Manjaro 20.2, Kernel 5.8.18), Python3.9 & Python 3.7
2. Windows 10, Python 3.9 & Python 3.7

测试框架选用PyTest, (如果有时间的话)自动测试UI的框架参考[QTest](#)

1. 测试范围

考虑到开发进度, 采用单元测试、集成测试以及系统测试对于软件的功能进行验证和测试, 此外不进行特别的程序性能以及其他非功能的测试, 比如程序的静态测试。

2. 测试策略

首先根据类图的设计, 对于功能类进行单元测试, 测试覆盖软件util和pipe包下所有的类。单元测试完成后对于各个模块进行组合, 按照UI、逻辑两个思路进行集成测试。单元测试和集成测试都使用白盒方法。最后对于整体软件系统, 按照软件需求中的涉及到的用例, 使用黑盒测试覆盖所有用例。

3. 单元测试

对于软件中的最小可测试单元进行验证和检查, 包括但不限于数据结构、网络通信等非UI的部分, 此外, 将软件的独立单元与软件的其余部分独立开来进行测试。

对于程序的白盒测试, 在编写程序同时进行。使用PyTest进行自动化测试, 重点测试非UI功能的实现(比如数据结构)。

4. 集成测试

在单元测试的基础上按照设计架构封装为的子系统集成测试, 重点测试每个子系统是否功能工作正常, 此外系统之间通信是否正常(UI相关, 测试信号与槽)。

5. 系统测试

考虑到需求文档里面涉及到的用例, 使用黑盒测试的方式对于整个软件系统进行测试。如果时间充分, 部分用例可以使用自动化框架进行白盒测试。

TODO: 按照用例设计测试报告

5.1 选择文件系统来源

测试要点:

1. 左侧窗口文件系统加载正确
2. 窗口文件系统点击交互正确, 触发效果正确
3. 文件树显示
4. 文件系统根目录未指定时错误处理

5.2 更改显示目录

测试要点：

1. 双击子树操作交互
2. 文件树刷新正确
3. 顶部地址栏刷新地址

5.3 删除文件系统

测试要点：

1. 选中文件系统操作交互(弹出交互列表)
2. 文件树刷新操作(显示背景，无文件树)

5.4 刷新文件系统

测试要点：

1. 点击刷新按钮交互以及错误处理(未选中文件树//不操作)
2. 测试刷新文件树显示(更改文件系统/不更改文件系统分别测试)

5.5 导入新的文件系统

测试要点：

1. 点击导入按钮交互(弹出选项列表：本地/网上；本地：弹出文件夹选择窗口；网上：地址)
2. 导入文件系统操作验证(检查文件系统正确性，更新文件系统列表)

5.6 添加文件系统子节点

测试要点：

1. 选择文件树节点操作
2. 点击"+"按钮操作交互(弹出节点类型列表)
3. 刷新文件树缓存与显示UI

5.7 删除文件系统子节点

测试要点：

1. 选择文件树节点操作
2. 点击"-"按钮操作交互
3. 删除子树节点操作：数据缓存&UI刷新

5.8 用户搜索

测试要点：

1. 点击搜索框操作(支持历史搜索记录测试?)
2. 选择搜索模式(文件名/tag)
3. 搜索文件时间测试&准确性测试
4. 多tag/tag&文件名 联合搜索测试

5.9 用户添加tag

测试要点

1. 用户选中文件树节点操作(多选?单选)
2. 用户在右侧直接修改tag列表，可以增加/删除tag列表
3. 刷新tag列表与数据库记录

5.10 用户备份本地文件系统

测试要点

1. 选中文件系统测试
2. 清华云盘操作连接
3. 清华云盘上传
4. 后台挂起文件上传操作

5.11 用户分享本地文件系统

测试要点：

1. 用户选中文件树
2. 上传至清华云盘(重复上传检测)
3. 生成分享链接

5.12 在本地文件浏览器/自定义软件中打开文件节点

测试要点：

1. 选中文件树节点
2. 点击打开按钮交互(右键单机可以弹出打开方式设计，仿造windows)
3. 软件打开测试

5.13 用户操作UI节点

测试要点：

1. 用户选中某个文件节点
2. 在右侧窗口中更改属性操作测试
3. 展开节点、关闭展开测试

6. 测试结果

目前由于编写进度的原因，主要测试了逻辑相关的代码，修改了一些bug。UI相关的代码正在开发中，需要之后进行测试。