PPT Copilot

PPT Copilot: 软件系统测试报告

2051857 曾诗容——系统测试人员,
2052636 陈骁——覆盖测试人员,
2050250 李其桐——覆盖测试人员,
2054080 林奕如——系统测试人员,
2053865 刘昱彤——系统测试人员,
1751118 吴达鹏——系统测试人员
2053868 于采篱——接口压力测试人员

Computer Science and Technology Department, College of Electronic and Information Engineering(CEIE), Tongji University.
同济大学 电子与信息工程学院 计算机科学与技术系

2023 年 6 月 12 日

项目内容简介

项目开发背景

- 当前，ChatGPT 已经发布并且对公众开放了服务接口，这无疑标志着一个人工智能的新纪元已然到来。通过 ChatGPT 的强势赋能，使得许多传统工作流都得到了极大的颠覆与创新，在达到更高效率的同时也能够确保质量。本项目正是在这一背景下，基于 ChatGPT 的公开 API 接口，搭建一个能够根据用户的自然语言描述需求，自动生成 Markdown 格式文档，然后通过 Markdown 解析器处理文件文本内容从而生成 ppt 的软件系统，从而能够在人们的实际文档设计与编写工作中，以一个可靠软件助手的姿态提供辅助，有效提升人们的工作效率。

项目核心功能需求

1. 在线 PPT 文档编辑，提供与本地编辑器类似的编辑能力
2. 通过 ChatGPT 生成 PPT 文档内容
3. 文档项目管理，包括项目创建删除、项目文件管理、项目分享克隆等
4. 用户账号管理

软件定义时期

甲方参与流程

本小组在 PPT Copilot 软件系统项目开发流程中，作为甲方参与项目，与乙方积极交流、沟通，合作，主要参与了软件生命周期中的以下的项目开展环节：

- 问题定义：本小组作为甲方，在项目开展初期对项目需求进行了详细的定义和规划，对软件的功能、性能需求以及目标用户进行了明确的设定。
- 可行性分析：本小组作为甲方，在项目开展初期对项目的可行性进行了分析，对软件的技术可行性、经济可行性、法律可行性以及进度可行性进行了详细的分析。
- 需求分析：本小组作为甲方，同样在项目开展初期明确了软件应用的场景，用户需求，核心产品痛点和主要抓手等需求内容，随后与乙方进行交流沟通，进一步调整并且完善软件系统需求内容，并且将这些需求清晰地向乙方表达。
- SRS 规格说明文档：本小组作为甲方，将上述的三阶段分析内容整理为了详细的 SRS 文档，从而使得乙方能够更好地理解软件的需求，为软件的开发提供了明确的目标。

软件开发时期

甲方参与流程

- 系统部署：本小组作为甲方，根据乙方提供的部署操作说明，正确在 ECS 上部署并且运行了软件系统，以便后续测试工作的开展。
- 单元测试：本小组作为甲方，在乙方将系统源码发给我们后，简单分析了一下项目模块单元内部的实现代码架构与逻辑，随后主要采用白盒测试的相关覆盖方法进行了单元测试，从而保证了软件系统的基本单元功能的正确性。
- 综合测试：本小组作为甲方，采用等价类划分、场景法等黑盒测试方法，通过 Apifox 和乙方实现的 Web 端界面，合理设计测试用例，对于系统功能进行了比较全面且详细的测试，并且将测试结果向乙方反馈，进一步进行沟通交流，从而保证了软件系统的功能性、可用性、易用性、稳定性等质量属性。
- 测试报告：本小组作为甲方，对于软件系统的测试结果进行了详细的整理和总结，撰写了详细的测试报告，从而使得乙方能够更好地理解软件系统的测试情况，为软件的上线提供了明确的目标。

软件维护时期

甲方参与流程

- 系统维护：TODO
- 系统重构：TODO

主功能模块

修改内容

本小组在需求分析阶段，与乙方积极沟通，对于如下 SRS 需求文档中的主功能模块进行了修改：

- 账号管理：简化了账号管理模块功能，移除了用户充值、用户权限管理等部分用例
- 用户充值：完全移除了用户充值功能模块需求，将用户充值功能作为后续的扩展功能进行考虑
- 内容自动化生成：简化了 ChatGPT 对接逻辑，删除了关于 Markdown 中间件生成的用例，由乙方在详细设计中自行决定如何通过 ChatGPT 驱动 PPT 文档内容生成
- 项目管理：删除了部分项目管理用例，使得项目管理模块更加简洁

其它需求

修改内容

- 性能需求：降低了对于系统并发性能、吞吐量的要求
- 质量属性需求：降低了对于系统的可用性、易用性、可靠性、可维护性、可移植性的要求，使得系统更加注重功能性

测试用例设计

黑盒测试，主要采用场景法设计用例，大约设计了 60 个左右的 Testcases：

功能点/issue	测试用例 id	测试用例	优先级
账号管理-用户注册	TC001	正常注册流程，在注册界面中输入相关信息，点击注册，显示注册成功	最高
账号管理-用户注册	TC002	在注册时用户账号已被注册，提示失败并且结束用例	高
账号管理-用户注册	TC003	在注册时邮箱已经被绑定，提示失败并且结束用例	高
账号管理-用户注册	TC004	在注册时未将表单填写完整，提示无法提交注册表单并显示未填写的字段	高
账号管理-用户登录	TC005	正常登录流程，输入账号密码后成功登录	最高
账号管理-用户登录	TC006	登录时账号不存在，提示登录失败	高
账号管理-用户登录	TC007	登录时账号密码错误，提示登录失败	高
项目管理-项目搜索	TC008	正常搜索流程，输入搜索内容输出匹配项目	最高
项目管理-项目新建	TC009	正常创建流程，点击新建项目输入项目信息后创建	最高
项目管理-查看项目上传的文件	TC010	用户查看不存在的项目文件	中
项目管理-查看所有项目	TC011	正常查看所有项目	最高
项目管理-重命名	TC012	正常重命名项目	最高
项目管理-重命名	TC013	用户输入空项目名字后点击确认，应当提示失败	中
项目管理-上传封面	TC014	正常上传项目封面	最高
项目管理-上传封面	TC015	上传非图片格式的封面文件	中
项目管理-删除项目	TC016	正常删除项目	最高
项目管理-项目新建	TC017	新建具有相同名字的项目	中
项目管理-上传文件	TC018	正常上传文件	最高
项目管理-上传文件	TC019	上传大小大于 100MB 的大文件	中
项目管理-上传文件	TC020	编辑上传文件的文件名	最高

表：部分测试用例展示

需求覆盖分析

测试用例 id	是否通过	备注
TC001	通过	无
TC002	通过	无
TC002	通过	无
TC003	不通过	只实现了登录时填写邮箱信息，未实现验证功能
TC004	通过	无
TC005	通过	无
TC006	通过	无
TC007	通过	无
TC008	通过	但搜索不存在的项目时，界面不太友好，未出现相应提示信息，界面不变；界面的时间显示紊乱
TC009	通过	无
TC010	通过	无
TC011	通过	无
TC012	通过	无
TC013	不通过	能重命名项目为空；能创建多个名字相同的项目；重命名后后缀丢失，PPT 文件重命名（未加后缀）后无法打开
TC014	通过	无
TC015	不通过	无
TC016	通过	无
TC017	不通过	无
TC018	通过	无
TC019	不通过	文件过大时无法上传
TC020	不通过	修改上传文件名时网页总是提示错误信息

表: 部分需求覆盖展示

缺陷统计

序号	Bug 描述	重要性等级	所属模块	是否修复	尚未修复	备注
1	项目中的非 PPT 的 json 文件后出现打开按钮（正常应该只出现在 PPT 文件后），点击后卡在加载页面	一般	项目管理	是		
2	用户可以操作非自己创建的项目，如删除、重命名文件	一般	项目管理	是		
3	注册时有要求填写邮箱信息，但在注册界面并没有实现验证功能	一般	账号管理	是		
4	项目可以重命名为空	较严重	项目管理	是		
5	项目可以重名	一般	项目管理		是	不影响使用，可以准确区分项目的创建者
6	PPT 重命名不加后缀，之后不能打开	较严重	项目管理	是		
7	搜索结果界面的时间显示紊乱	一般	项目管理	是		
8	上传头像/PPT 封面时未对文件进行限制	较严重	项目管理/个人信息	是		
9	个人信息修改界面的邮箱验证无效	一般	个人信息	是		
10	搜索界面时间显示紊乱	一般	项目管理	是		

表: 部分缺陷统计展示

性能测试

- 性能测试中，我们主要使用 Apifox 测试了软件系统同时并发支持用户注册登录的性能表现，发现随着并发用户数量的增加，服务器的响应时间和吞吐量也相应增加，但在达到某个阈值后，这两个指标开始稳定。这意味着软件系统能够在一定的并发访问量下保持高效的处理能力。
- 当并发用户数量进一步增加，服务器的错误率也开始增加。这表明在极端的并发访问情况下，系统可能会遇到性能瓶颈，这是可能需要进一步优化的地方。

性能测试

Apifox 报告

测试场景	用户管理接口压力测试	
运行时间	2023-05-22 11:26:43	
运行工具	Apifox v2.2.39	
	总数	**失败数**
循环数	50	0
HTTP 接口请求数	300	0
断言数	0	0
总耗时	1m 52.80s	
总返回数据	27.45KB	
平均接口请求耗时	70ms	
通过率	100.00%	
失败率	0.00%	
未测率	0.00%	

测试工作总结

- 软件系统整体功能 Bug 较少，运行性能达到 SRS 文档中的相关性能需求
- 与乙方同学的沟通非常顺畅，在反馈 Bug 后，乙方修复 Bug 和重新部署系统的速度也非常快
- 最后修复后的软件系统通过了我们的测试验证，达到了预期的测试目标