



测试报告

垂直搜索引擎

组长：马梓睿

组员：程梦烨 毕予然 沈乐明 李楠

Contents

1	测试介绍	3
1.1	测试对象	3
1.2	测试目的	3
1.3	测试范围	3
2	测试基础	4
2.1	测试环境	4
2.1.1	软件运行环境	4
2.1.2	硬件运行环境	4
2.2	版本信息	8
3	测试方法	8
3.1	静态测试和动态测试	8
3.2	黑盒测试、白盒测试和灰盒测试	8
3.3	手动测试和自动化测试	9
3.4	等价类划分和边界值分析法	9
4	测试用例	10
5	测试结果	12
5.1	测试用例执行情况	12
5.2	测试结果	12
5.2.1	执行情况	12
5.2.2	通过情况	13
6	缺陷指标	14
6.1	缺陷摘要	14
6.2	缺陷跟踪曲线	15
7	测试风险	16
7.1	需求风险	16
7.2	测试环境风险	16
8	后续处理措施	16

1 测试介绍

1.1 测试对象

本文档的测试对象为本小组开发的 Buy.Wiki 垂直搜索引擎，具体的测试点主要包括以下几个方面：

- 普通搜索：输入搜索关键词，搜索结果
- 高级搜索：输入多个搜索关键词，输入搜索方式、关键词关系（与/或），搜索结果
- 查看搜索信息列表：用户查看搜索结果信息列表，选择结果排序方式
- 查看搜索结果详情：用户点击某个商品信息，查看详情页
- 更新知识图谱：用户选择更新知识图谱，重新爬取
- 更新爬虫网站源：用户选择添加爬虫网站，更新图数据库内容

1.2 测试目的

在已经规定好的条件下对本项目进行测试，目的在于通过测试去发现软件中程序的错误或者是 BUG，衡量软件的质量，然后对软件是否满足最初的要求或者初衷做出一个正确的判断。

1.3 测试范围

本文档的测试主要是对功能进行黑盒测试，测试的功能为系统的基本功能。包含普通搜索模块、高级搜索模块、搜索结果显示模块、更新知识图谱模块、更新爬虫网站源模块。

2 测试基础

2.1 测试环境

2.1.1 软件运行环境

表 2-1 客户端软件配置

项目	说明
浏览器	Google Chrome, Microsoft Edge, Firefox, Safari, Opera
数据库	MySQL Ver 8.0.21-0ubuntu0.20.04.4 for Linux on x86_64 ((Ubuntu))
测试数据	由测试团队设计

2.1.2 硬件运行环境

2.1.2.1 服务器端

表 2-2 服务器硬件配置

项目	说明
服务器	学校服务器
处理器	Intel(R) Xeon(R) Silver 4108 CPU @ 1.80GHz
核数	32
内存	96GB RAM
缓存	360448 KB
硬盘	512GB SSD
带宽	1000Mbps
地址大小	46 bits physical, 48 bits virtual
操作系统	Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-31-generic x86_64)
数据库	OrientDB 3.2.0

2.1.2.2 客户端

表 2-3 成员沈乐明的客户端硬件配置

项目	信息
供应商	LENOVO
操作系统	Windows 10 Pro 64-bit (10.0, Build 18363)
系统模型	20LBA01KCD
BIOS 版本	N27ET32W (1.18) (type: UEFI)
处理器	Intel(R) Core(TM) i5-8350U CPU @ 1.70GHz (8 CPUs), ~1.9GHz
内存	32.00GBRAM
硬盘	1.0TB SSD 1.0TB HDD
显卡	Intel(R) UHD Graphics 620 NVIDIA Quadro P500

显示器	<ul style="list-style-type: none"> 名称: Wide viewing angle & High density FlexView Display 1920x1080 分辨率 (刷新频率): 1920 x 1080(p) (59.977Hz) 名称: Generic PnP MonitorAOC2701 分辨率 (刷新频率): 1920 x 1080(p) (60.000Hz) 名称: Generic PnP Monitor DELL U2518D 分辨率 (刷新频率): 2560 x 1440(p) (59.951Hz)
网络	Microsoft ATSC Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001Microsoft DVBC Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001Microsoft DVBS Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001Microsoft DVBT Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001Microsoft Network Provider,0x00200000,0,1,MSNP.ax,10.00.18362.0001
鼠标	Logitech M330
键盘	MOTOSPEED GK89

表 2-4 成员毕予然的客户端硬件配置

项目	信息
供应商	LENOVO
操作系统	Windows 10 家庭中文版 64 位 (10.0, 版本 18363)
系统模型	81BV
BIOS 版本	6MCN14WW
处理器	Intel(R) Core™ i7-8550U CPU @ 1.80GHz (8 CPUs), ~2.0GHz
内存	8192MB RAM
硬盘	256GB SSD 1.0TB HDD
显卡	Intel(R) UHD Graphics 620
显示器	名称: Intel(R) UHD Graphics 620 桌面分辨率: 1920*1080 有源信号分辨率: 1920*1080 刷新频率: 60Hz 位深度: 8 位 颜色格式: RGB 颜色空间: 标准动态范围 (SDR)
网络	Microsoft ATSC Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft DVBC Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft DVBS Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001

	Microsoft DVBT Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft Network Provider,0x00200000,0,1,MSNP.ax,10.00.18362.0001
鼠标	HID-compliant mouse
键盘	HID Keyboard Device PS/2 标准键盘

表 2-5 成员程梦烨的客户端硬件配置

项目	信息
供应商	LENOVO
操作系统	Windows 10 家庭中文版 64 位 (10.0, 版本 18363)
系统模型	81BS
BIOS 版本	5ZCN31WW
处理器	Inter® Core™ i7-850U CPU @ 1.80GHz (8 CPUs), ~2.0GHz
内存	8192MB RAM
硬盘	
显卡	Inter® UHD Graphics 620 NVIDIA GeForce MX150
显示器	显示器 1: 已连接到 Intel® UHD Graphics 620 桌面分辨率: 1920×1080 有源信号分辨率: 1920×1080 刷新频率(Hz): 59Hz 位深度: 8 位 颜色格式: RGB 颜色空间: 标准动态范围(SDR)
网络	Microsoft ATSC Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft DVBC Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft DVBS Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft DVBT Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft Network Provider,0x00200000,0,1,MSNP.ax,10.00.18362.0001
鼠标	ThinkLife WLC200-M
键盘	

表 2-6 队员马梓睿的客户端硬件配置

项目	信息
供应商	LENOVO
操作系统	Windows 10 Pro 64-bit (10.0, Build 18363)
系统模型	20LBA01KCD
BIOS 版本	N27ET32W (1.18) (type: UEFI)
处理器	Intel(R) Core(TM) i7-8750HQ CPU @ 2.20GHz (12 CPUs), ~4.10GHz
内存	16.00GB RAM
硬盘	512GB SSD
显卡	NVIDIA GTX 1050Ti
显示器	<ul style="list-style-type: none"> 名称: Generic PNP 分辨率 (刷新频率) : 1920 x 1080(p) (59.977Hz)
网络	Intel AX200 802.11 ax 160 MHz
鼠标	Logitech G102
键盘	HID Keyboard Device

表 2-6 队员李楠的客户端硬件配置

项目	信息
供应商	HP
操作系统	Windows 10 家庭中文版 64 位 (10.0, 版本 18363)
系统模型	81BS
BIOS 版本	5ZCN31WW
处理器	Inter® Core™ i5-750U CPU @ 1.80GHz (8 CPUs), ~2.0GHz
内存	8192MB RAM
硬盘	
显卡	Intel(R) UHD Graphics 620
显示器	显示器 1: 已连接到 Intel® UHD Graphics 620 桌面分辨率: 2560×1920 有源信号分辨率: 2560×1920 刷新频率(Hz): 59Hz 位深度: 8 位 颜色格式: RGB 颜色空间: 标准动态范围(SDR)
网络	Microsoft ATSC Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft DVBC Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft DVBS Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001 Microsoft DVBT Network Provider,0x00200000,0,1,MSDvbNP.ax,10.00.18362.0001

	Microsoft Network Provider,0x00200000,0,1,MSNP.ax,10.00.18362.0001
鼠标	ThinkLife WLC200-M
键盘	

2.2 版本信息

Buy.Wiki 垂直搜索引擎版本 v1.0.0_beta

报告版本 v1.0

3 测试方法

3.1 静态测试和动态测试

静态方法是指不运行被测程序本身，仅通过分析或检查源程序的语法、结构、过程、接口等来检查程序的正确性。静态方法通过程序静态特性的分析，找出欠缺和可疑之处，例如不匹配的参数、不适当的循环嵌套和分支嵌套、不允许的递归、未使用过的变量、空指针的引用和可疑的计算等。静态测试结果可用于进一步的查错，并为测试用例选取提供指导。

动态方法是指动态测试是通过运行软件来检验软件的动态行为和运行结果的正确性。通过运行被测程序，检查运行结果与预期结果的差异，并分析运行效率和健壮性等性能，这种方法由三部分组成：构造测试实例、执行程序、分析程序的输出结果。

3.2 黑盒测试、白盒测试和灰盒测试

黑盒测试是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。其以用户的角度，从输入数据与输出数据的对应关系出发进行测试，如果外部特性本身设计有问题或规格说明的规定有误，用黑盒测试方法是发现不了的。

白盒测试又称结构测试、透明盒测试、逻辑驱动测试或基于代码的测试。白盒测试是一种测试用例设计方法，盒子指的是被测试的软件，白盒指的是盒子是可视的，即清楚盒子内部的东西以及里面是如何运作的。"白盒"法全面了解程序内部逻辑结构、对所有逻辑路径进行测试。"白盒"法是穷举路径测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。

灰盒测试，是介于白盒测试与黑盒测试之间的一种测试，多用于集成测试阶段，不仅关注输出、输入的正确性，同时也关注程序内部的情况。灰盒测试不像白盒那样详细、完整，但又比黑盒测试更关注程序的内部逻辑，常常是通过一些表征性的现象、事件、标志来判断内部的运行状态。

3.3 手动测试和自动化测试

手动测试是由人去一个一个的输入用例，然后观察结果，和机器测试相对应，属于比较原始但是必须的一个步骤。

自动化测试是把以人为驱动的行为转化为机器执行的一种过程。通常，在设计了测试用例并通过评审之后，由测试人员根据测试用例中描述的规程一步步执行测试，得到实际结果与期望结果的比较。在此过程中节省了人力、时间或硬件资源，提高了测试效率。

3.4 等价类划分和边界值分析法

等价类划分法将程序所有可能的输入数据（有效的和无效的）划分成若干个等价类。然后从每个部分中选取具有代表性的数据当做测试用例进行合理的分类，测试用例由有效等价类和无效等价类的代表组成，从而保证测试用例具有完整性和代表性。利用这一方法设计测试用例可以不考虑程序的内部结构，以需求规格说明书为依据，选择适当的典型子集，认真分析和推敲说明书的各项需求，特别是功能需求，尽可能多地发现错误。等价类划分法是一种系统性的确定要输入的测试条件的方法。

由于等价类是在需求规格说明书的基础上进行划分的，并且等价类划分不仅可以用来确定测试用例中的数据的输入输出的精确取值范围，也可以用来准备中间值、状态和与时间相关的数据以及接口参数等，所以等价类可以用在系统测试、集成测试和组件测试中，在有明确的条件和限制的情况下，利用等价类划分技术可以设计出完备的测试用例。这种方法可以减少设计一些不必要的测试用例，因为这种测试用例一般使用相同的等价类数据，从而使测试对象得到同样的反映行为。等价类划分的方法分为两个主要的步骤，划分等价类型和设计测试用例。

边界值分析法就是对输入或输出的边界值进行测试的一种黑盒测试方法。通常边界值分析法是作为对等价类划分法的补充，这种情况下，其测试用例来自等价类的边界。

大量的错误是发生在输入或输出范围的边界上，而不是发生在输入输出范围的内部。因此针对各种边界情况设计测试用例，可以查出更多的错误。

边界值测试与等价划分的区别如下：

- 边界值分析不是从某等价类中随便挑一个作为代表，而是使这个等价类的每个边界都要作为测试条件。
- 边界值分析不仅考虑输入条件，还要考虑输出空间产生的测试情况。

4 测试用例

表 4-1 高校教学平台测试用例设计

用例名称	测试编号	测试数据	预期结果	实际结果 PASS/FAIL
关键词搜索	GS-01-01	“口罩”	100 个与“口罩”相关的商品	PASS
	GS-01-02	“洗手液”	100 个与“洗手液”相关的商品	PASS
关键词过滤	GS-02-01	[透气的口罩]	100 个与“透气的口罩”相关的商品	PASS
	GS-02-02	^透气的口罩	100 个与“透气的口罩”相关的商品	PASS
拼音搜索	GS-03-01	“Kouzhaio”	100 个与“口罩”相关的商品	PASS
	GS-03-02	“xishouye”	100 个与“洗手液”相关的商品	PASS
结果显示	GS-04-01	完成搜索	结果页面按照默认相关度排序，显示搜索结果卡片列表	PASS
图片、视频展示 选中商品的图片、视频	GS-05-01		完成搜索，点击	PASS
	GS-06-01		显示图片、视频	FAIL
结果相关度排序	GS-06-01	点击“按相关度排序”	结果页面显示按相关度排序后的搜索结果卡片列表	PASS
关键词交集	AS-01-01	关键词关系：or	显示出搜索结果列表	PASS
	AS-01-02	用户不选择关键词关系	提示用户选择关键词关系	PASS
关键词并集	AS-02-01	关键词关系：and	显示出搜索结果列表	PASS
	AS-02-02	用户不选择关键词关系	提示用户选择关键词关系	PASS
优先补全	AS-03-01	用户输入任意关键词	自动补全关键词	PASS
搜索提示	AS-04-01	用户输入任意关键词	给出相关搜索提示	PASS
关键词条件筛选	AS-05-01	单条件：name,include	显示出搜索结果列表	PASS

用户查看 关键词	KW-01-01	用户点击了查看 关键词的按钮	显示当前正在等待 队列中的所有关键词	PASS
用户增加 关键词	KW-02-01	用户输入了合法 的新关键词	系统提示新关键词 正在爬虫中，一天 后能够搜索到相关 内容	PASS
	KW-02-02	用户输入了非法 格式的新关键词	系统提示搜索格式 错误	PASS
用户更新知识图 谱	MAI-39	访问网页并截图	显示网页界面	PASS
	MAI-40	操控键鼠输入数 据	跳转到新页面并携 带数据	FAIL
	MAI-41	前进后退和标签 页切换	均切换到正确页面	PASS
	MAI-42	两种方式搜索元 素	获得搜索框元素	PASS
	MAI-43	发送 POST 请求	服务器开始爬取	PASS
用户查看 爬虫网站源	UD-WB-01	查看爬虫网站源	显示爬虫网站源	PASS
用户更新 爬虫网站源	UD-WB-02	添加未添加的爬 虫网站源	显示添加成功	PASS

5 测试结果

5.1 测试用例执行情况



图 5-1 测试用例执行情况

5.2 测试结果

5.2.1 执行情况

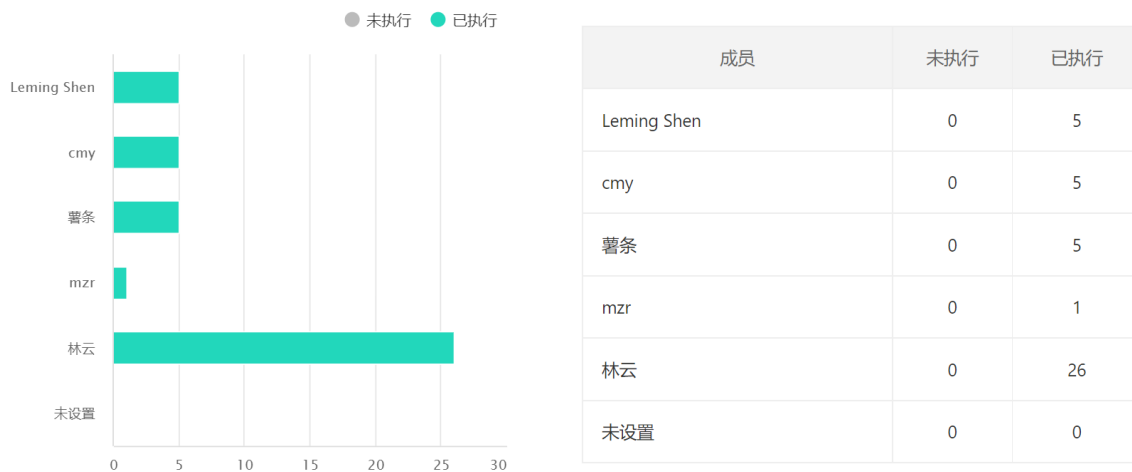


图 5-1 测试用例执行情况

5.2.2 通过情况



图 5-2 测试用例通过情况

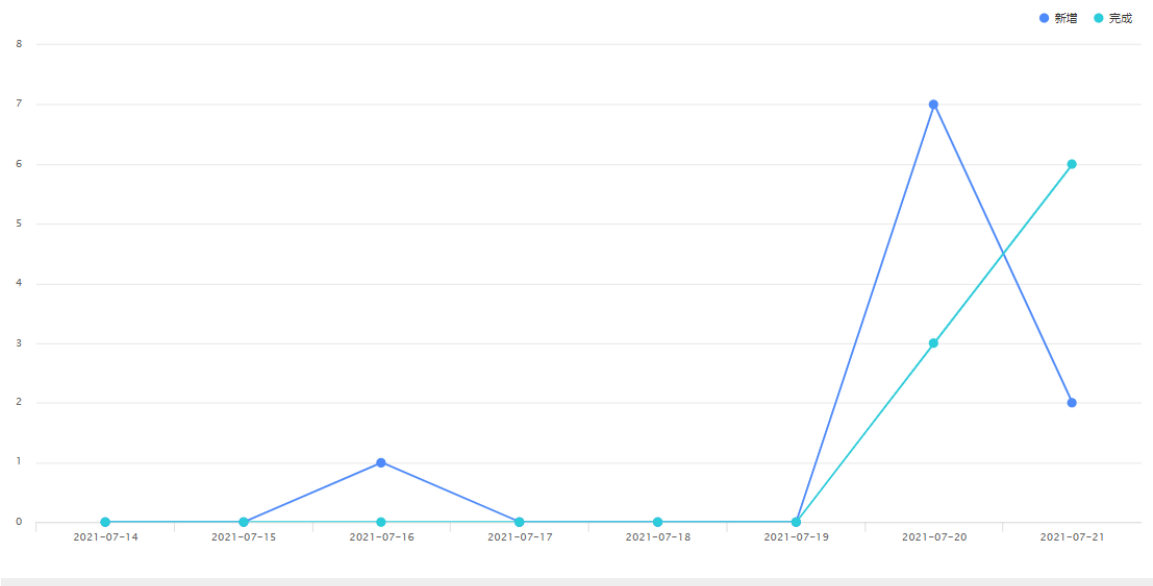
6 缺陷指标

6.1 缺陷摘要

表 6-1 缺陷摘要

缺陷编号	摘要	分类	处理状态	优先级	严重性	复现性	创建时间
DF-01	OrientDB 无法连接	知识图谱构建端	已发布	最高	致命	必现	7.16
DF-02	Java 搭建的 HTTP 通信难以解析 Json	知识图谱构建端	已发布	较高	严重	必现	7.20
DF-03	OrientDB 难以在 Docker 中运行	知识图谱构建端	已发布	普通	一般	必现	7.20
DF-04	OrientDB 连接久了会断开	搜索端	已发布	普通	严重	小概率复现	7.20
DF-05	返回结果为 Unicode 编码	搜索端	已发布	最高	致命	必现	7.20
DF-06	淘宝对访问 ip 限制	爬虫端	已发布	较高	严重	小概率复现	7.20
DF-07	相关商品不在返回结果中	爬虫端	已发布	较高	严重	必现	7.20
DF-08	服务器对跨域请求的限制	前端	已发布	较高	严重	必现	7.20
DF-09	爬虫生成格式错误	添加网站	已发布	较高	严重	必现	7.21
DF-10	前端自动化部署失败	添加网站	已拒绝	普通	一般	必现	7.21

6.2 缺陷跟踪曲线



表格

日期	新增数	完成数
2021-07-14	0	0
2021-07-15	0	0
2021-07-16	1	0
2021-07-17	0	0
2021-07-18	0	0
2021-07-19	0	0
2021-07-20	7	3
2021-07-21	2	6

图 6-1 缺陷跟踪曲线

7 测试风险

7.1 需求风险

对软件需求本身认识的不清晰，或者甲方对产品的需求特性的理解总结有偏差，或者测试人员对软件需求认知不准确，都会导致最终开发出的产品不是用户真正需要的，不符合用户实际对功能的需求。此外，需求变更后，测试用例的不及时更新也会导致测试风险。

7.2 测试环境风险

测试人员在测试的过程中所使用的测试环境，不一定符合用户在实际使用时的真实环境，即使原则上要求两者完全一致，但实际上这样的模拟与用户不同场景下的使用存在一定的偏差，这样就存在了一定的测试环境风险，由于软件的某些功能只有在特定的条件下（包括硬件、网络、操作系统、flash 插件和软件的不同版本）才会发生缺陷，因此测试环境的不完全性会导致测试风险的产生。

8 后续处理措施

表 8-1 缺陷后续处理措施

缺陷编号	摘要	后续解决措施
DF-01	OrientDB 无法连接	修改连接配置
DF-02	Java 搭建的 HTTP 通信难以解析 Json	按字符串读入后使用库解析
DF-03	OrientDB 难以在 Docker 中运行	使用现成的 docker image
DF-04	OrientDB 连接久了会断开	使用数据库池化技术
DF-05	返回结果为 Unicode 编码	在 JSON 化时进行手动解码
DF-06	淘宝对访问 ip 限制	限制访问量；使用分布式策略
DF-07	相关商品不在返回结果中	爬取其他信息源
DF-08	服务器对跨域请求的限制	服务端设置 Access-Control-Allow-Origin
DF-09	爬虫生成格式错误	重新爬取数据
DF-10	前端自动化部署失败	搁置

