

浙江大学



Techhub 后端知识库搜索引擎系统

软件概要设计说明书

Group 08

黄亦非、傅诤哲、胡瀚丹、陈鑫

2019/07/10

目录

1. 引言	4
编写目的	4
背景	4
定义	5
参考资料	5
2. 需求规定	6
2.1 功能性需求	6
2.2 其他非功能性需求	8
2.2.1 性能需求	8
2.2.2 防护性需求	8
2.2.3 安全性需求	9
2.2.4 软件质量属性	10
对用户重要的属性	10
2.2.5 对开发者重要的属性	13
2.2.6 用户文档	13
3. 设计说明	14
3.1 用户接口	14
3.2 运行环境	16
3.3 设计和实现上的约束	18
3.4 系统体系结构设计	19
3.5 系统用例设计	21
4. 系统实现时间表与里程碑	24
5. 测试策略	25
5.1 测试方案	25

5.2 测试标准	25
6. 数据字典	26
6.1 数据元素定义表.....	26
6.2 数据流定义表	27
6.3 外部项定义表	28
6.4 数据集成、留存与销毁.....	29

1. 引言

编写目的

由前面的需求分析，得出了系统的基本需求和基本的数据流图，要实现整个系统，需要对用户的需求进行设计，概要设计主要是利用比较抽象的语言对整个需求进行概括，确定对系统的物理配置，确定整个系统的处理流程和系统的数据结构，接口设计，实现对系统的初步设计。预期读者为客户代表和程序设计人员。

背景

Techhub 后端知识库搜索引擎系统主要是面向在校大学生用户，致力于在互联网上学习后端的相关技术，能够帮助用户快速的搜索相关领域的技术文档，包括该技术的描述、使用手册、教学视频、使用当中的相关问题等方面。对于用户而言，可以根据输入的问题或者关键字，准确的定位用户的意图，并且返回最准确的结果给用户。同时用户也可以根据提供的几个过滤条件对返回的结果进行对应的过滤，实现更加精确的结果定位。

同时系统对于数据的来源和处理也非常的重视，致力于提供最准确、最完整、最精确的知识库架构。对于数据我们会进行相关的过滤、去重、结构化信息的提取等等数据处理的动作，保证系统数据的稳定和准确。

定义

- MongoDB: 系统服务器所使用的非关系型数据库 (DBMS)。
- React: 一种 JavaScript MVC 框架。
- Material-UI: 一个实现了 Google's Material Design 设计规范的 react 组件库。
- HTML (HyperText Markup Language): 超文本标记语言。
- UML (Unified Modeling Language): 统一建模语言、是一套用来设计软件蓝图的标准建模语言, 是一种从软件分析、设计到编写程序规范的标准建模语言。
- Spring-Boot: 基于 Java 的微服务框架
- Solr: 分布式搜索引擎框架
- scrapy: 基于 python 语言的分布式爬虫框架

参考资料

《软件设计文档国家标准》

《软件工程项目开发文档范例》

《Software Requirements edition2》Karl E. Wiegers

《软件需求》刘伟琴、刘洪涛 译

2. 需求规定

2.1 功能性需求

2.1.1 用户需求

- 用户在该网站上面可以输入自己想要搜索和查找的内容，然后网站进行查询相关的内容。
- 同时在一定的响应阈值时间之内，网站能够展示相对应的搜索结果。
- 同时网站能对搜索结果进行内部排序然后返回显示，同时用户能够重置搜索搜索结果。
- 点击具体的条目时候显示条目的具体内容，并且可以链接到原网站当中。
- 支持分页显示，使得页面的显示更加的美观。
- 同时还支持用户使用不同的搜索条件进行查询相对应的更加准确的信息。

2.1.2 系统管理员需求

- 支持数据的手动导入，在后续还有数据更新的时候能够手动导入更新后台的数据库支持。
- 同时也支持数据的定时 import，这样为了方便管理员不需要随时随刻的去手动导入搜索引擎当中，当数据库当中有变动的时候，可以被自动的导入到搜索引擎当中。
- 管理员进行信息爬取，这个是系统的最基本的需求，需要在各个网站上爬取相关的信息，作为后台数据库的支持。
- 信息自动滤重功能，网上爬取的各种信息可能会存在很多的重复，所以需要 对爬取的数据进行滤重的动作。
- 信息自动分类和摘要，这个对于信息是比较重要的一个后处理，通过对信息自动摘要和分类，方便后续用户进行条件搜索。
- 信息自动爬取，当用户搜索结果小于我们设定的阈值的时候，能够支持实时的去各大搜索引擎当中爬取新的信息并且导入我们的后台数据库当中。

2.2 其他非功能性需求

2.2.1 性能需求

- PR-1：系统将容纳 500 以上的静态用户（注册用户）以及 200 以上的动态用户（在线用户），并发数将有 100 以上。
- PR-2：当单个用户在线时，系统响应用户动作的时间将小于 0.2s。
- PR-3：对相关搜索信息或者关键字进行搜索时，系统响应时间不超过 5s。
- PR-4：系统将能够连续运行一周。

2.2.2 防护性需求

- SA-1：当网站崩溃时，系统将把用户尚未保存的资料自动进行保存。
- SA-2：允许用户进行数据的备份和恢复，以弥补数据的破坏和丢失。
- SA-3：该系统将记录系统运行时所发生的所有错误，包括本机错误和网络错误。这些错误记录便于查找错误的原因，日志同时记录用户的关键性操作信息。

2.2.3 安全性需求

- SE-1: 每个用户的查询数据尽可能的进行保密
- SE-2: 除管理员之外的用户不能进行后台知识库的手动添加
- SE-3: 当系统发现用户对进行交换的数据进行无意或恶意的修改时,系统将防止此操作
- SE-4: 系统对一些重要的数据按一定算法进行加密,如一些重要参数等

2.2.4 软件质量属性

对用户重要的属性

2.2.4.1 可用性

- AV-1: 工作日期间, 当地时间早上 6 点至午夜, 系统的可用性至少达到 99.5%; 晚上 6 点到 12 点, 系统的可用性至少达到 99.95%。

2.2.4.2 有效性

- EF-1: 在预计的峰值负载条件下, 至少 25%的处理器能力和应用程序可用内存必须留出备用。
- EF-2: 在过多用户同时在线时, 性能降低程度不会超过百分之二十。

2.2.4.3 灵活性

- FL-1: 一个至少具有 6 个月产品支持经验的程序维护人员, 可以在大于一个小时的时间内为系统添加一个新的模块, 包括代码修改和测试。

2.2.4.4 完整性

- IN-1：只有管理员权限的用户能对后台的知识数据库进行手动导入和相关修改，系统将组织其他任何用户对后台知识数据库的修改。
- IN-2：系统能撤回管理员权限用户的有误操作。

2.2.4.5 可靠性

- RE-1：由于软件故障引起搜索失败或数据更新失败的概率不超过 1%。

2.2.4.6 健壮性

- R0-1：如果在用户搜索关键字时候发生故障，那么下次同一用户启动程序时，搜索引擎能恢复在故障发生时的相关搜索结果。
- R0-2：系统具有一定的容错和抗干扰能力，在非硬件故障或非通讯故障时，系统能够保证正常运行，并有足够的提示信息帮助用户有效正确地完成任务。

2.2.4.7 易用性

- US-1: 网站用户应该能在 5 秒之内, 得到相关的搜索引擎返回的结果。
- US-2: 系统管理员应该可以在 30s 内, 完成对后台知识库的导入和修改。
- US-3: 以前从未使用过该系统的网站用户, 能在不超过 5 分钟的适应后, 正确的使用该系统的搜索引擎和相关的过滤条件得到自己想要的搜索结果。
- US-4: 该系统能够对数据的来源进行相关的过滤, 聚焦在数据的精、深、和准确度等方面, 完成垂直搜索引擎的相关特性和功能特点。

2.2.5 对开发者重要的属性

2.2.5.1 可维护性

- MA-1. 程序维护人员应该在 20 小时或更短时间内，对现有报告进行更改。

- MA-2. 函数调用的嵌套层次不能超过两层。

- MA-3. 每个软件模块中，注释与源代码语句的比例至少为 1:2.

2.2.5.2 可测试性

- TE-1. 一个模块的最大循环复杂度不超过 20。

2.2.6 用户文档

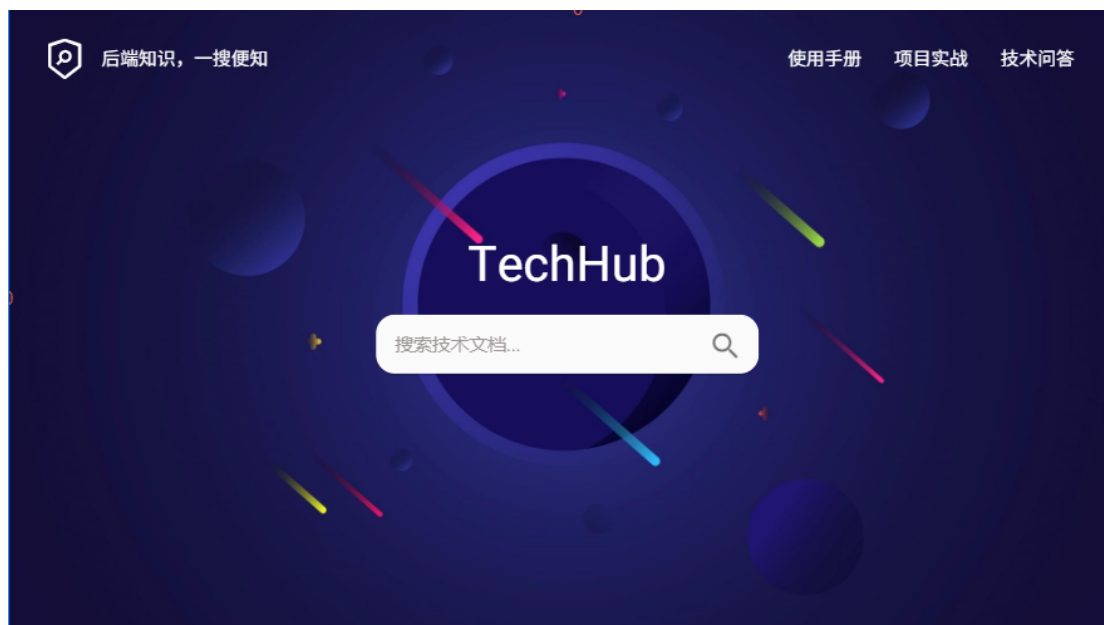
- 安装手册：Word 格式文件、PDF 格式文件
- 用户手册：Word 格式文件、PDF 格式文件
- 在线帮助

3. 设计说明

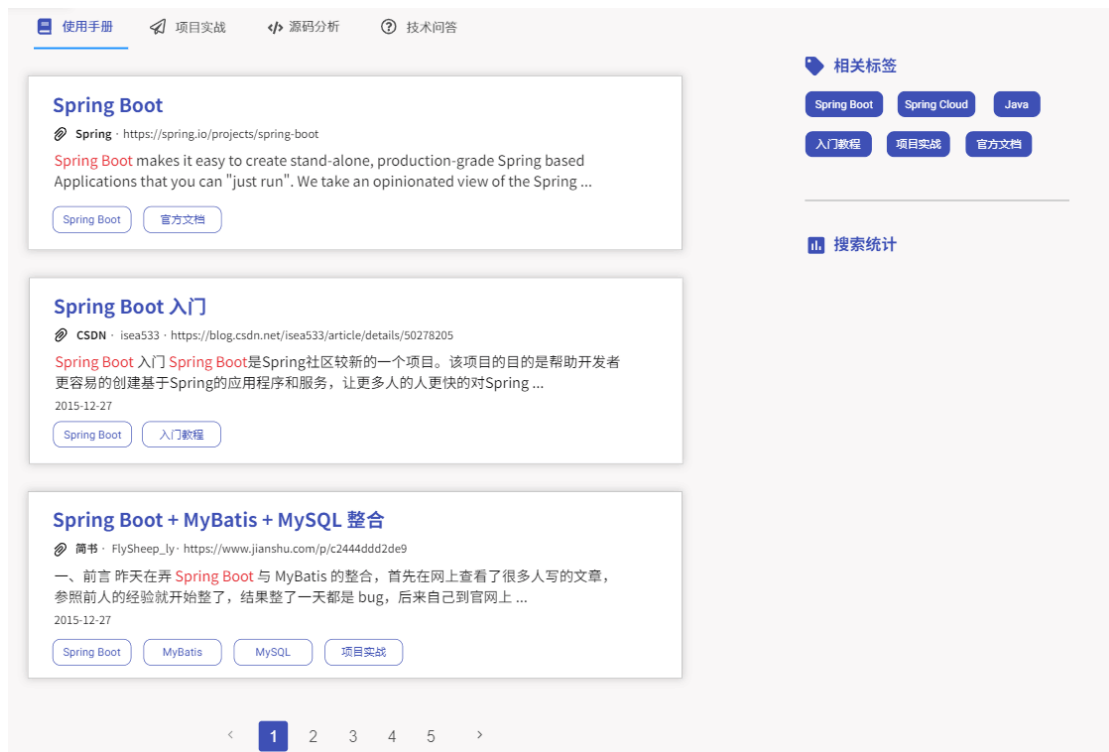
3.1 用户接口

- 所有界面采用 WEB 页面形式并适配移动端
- 简洁易用，方便用户的使用习惯
- 用户界面的具体细节将在软件概要设计说明书中描述

用户搜索界面原型：



搜索结果展示界面原型：



3.2 运行环境

软件层面

Techhub 后端知识库搜索引擎系统主要包括前端子系统、后端子系统（搜索引擎、后端 Web 系统）以及比较重要的爬虫子系统。其中前端子系统以基于 PWA 技术对 Web 端实现因此本系统需要现代网页浏览器进行访问及操作，浏览器需 Internet Explorer 11、Google Chrome 62、Mozilla Firefox 57、Safari 10 以及更高的版本以获得更好的体验。

服务端需运行在一个单独的服务器中，该服务器网卡接口需配置浙江大学校园网的静态 IP 地址供用户在浙江大学校内或在使用 RVPN 进行网站的访问，Techhub 后端知识库搜索引擎网站的网页服务由 Tomcat 或 Nginx 以及 Java Spring Boot 提供，同时服务器使用 MongoDB 数据库系统软件提供搜索引擎网站后台数据存储。

硬件层面

硬件层面对于所使用的服务器有以下要求:

- CPU:主频大于 2.0GHz
- 内存:大于等于 2GB
- 硬盘:硬盘容量大于 200GB、硬盘转速大于等于 5400 转/分钟
- 网卡:百兆网卡
- 网线:具有良好的数据传输能力
- 键盘:可以满足正常使用即可
- 鼠标:可以满足正常使用即可
- 显示器:可以满足正常使用即可

3.3 设计和实现上的约束

系统的设计、编码、以及维护将遵照所提交《项目可行性分析报告》、《项目总体计划》、《项目章程》、《软件质量保证计划》、《项目愿景和范围》、《需求工程计划》共计五个文档进行。在具体设计和实现上，按照以下约束进行：

3.3.1 数据存储

项目产品使用标准 MongoDB 数据库系统作为引擎，按照数据产生、转换和存储的策略，通过将数据导入数据库的方式进行数据的存储操作。

3.3.2 网络服务吞吐

根据项目要求，本项目要求提供对外服务的能力，以确保同时为至少 300 名用户同时进行相关内容搜索的要求。

3.3.3 数据安全

保证以下完整性、保密性以及可用性三个特性来保护用户的数据安全：

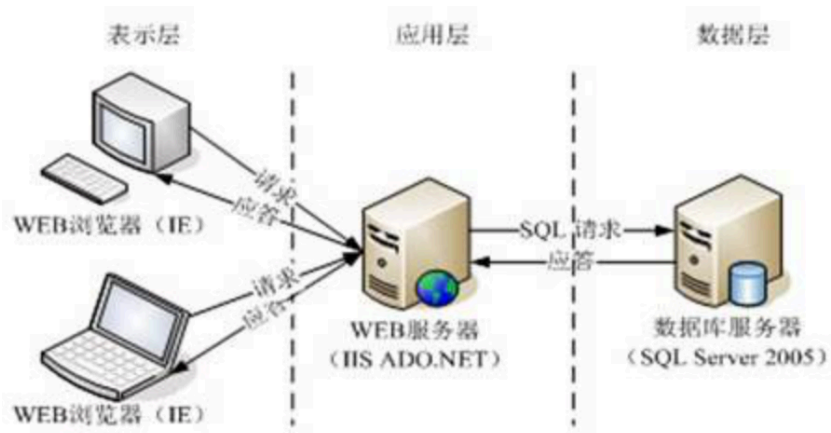
- 完整性要求数据未经授权不得进行修改，确保数据在传输和存储过程中不被篡改，盗用和丢失。通过利用安全的框架，在加密的基础上，运用多种方案和技术实现。
- 保密性要求对数据进行加密，只有授权者才能使用。这一特性要求加密技术必须自动、实时、精确、可靠。
- 可用性要求做到避免因为系统数据泄露而使合法使用者无法接触可用数据，通过对使用者身份的验证，为合法使用者提供更安全便捷的使用。

3.4 系统体系结构设计

B/S 体系结构

在系统中使用浏览器-服务器(Browser/Server, B/S)体系结构,采用 B/S 的三层体系结构,将系统的整个业务应用划分为表示层、业务逻辑层和数据访问层,这样有利于系统的开发、维护、部署和扩展。B/S 结构的基本原则是将计算机应用任务分解成多个子任务,由多台计算机分工完成,即采用“功能分布”原则。客户端完成数据处理,数据表示以及用户接口功能;服务器端完成 DBMS 的核心功能。这种客户请求服务、服务器提供服务的处理方式是一种新型的计算机应用模式。

B/S 体系结构如下图所示:



3.5 体系结构实现

对于这一种系统，采用如下方式进行实现：

(1)采用如下的服务端、客户端要求：

Techhub 后端知识库搜索引擎系统客户端使用现代网页浏览器进行访问及操作，网页适配适配移动端查看及访问，浏览器对象为 UC 浏览器，QQ 浏览器，欧朋浏览器，百度手机浏览器，360 安全浏览器，谷歌浏览器，搜狗手机浏览器，猎豹浏览器，以及一些手机的自带浏览器。网页组件由 Material UI 库提供。

Techhub 后端知识库搜索系统需运行在一个单独的服务器中，该服务器网卡接口需配置公网静态 IP 地址供用户在全球各地通过互联网进行网站的访问，Techhub 后端知识库搜索引擎系统网站的网页服务由 Node.js 以及 Java Spring Cloud 提供，同时服务器使用 MongoDB 数据库系统软件提供管理系统网站后台数据存储。

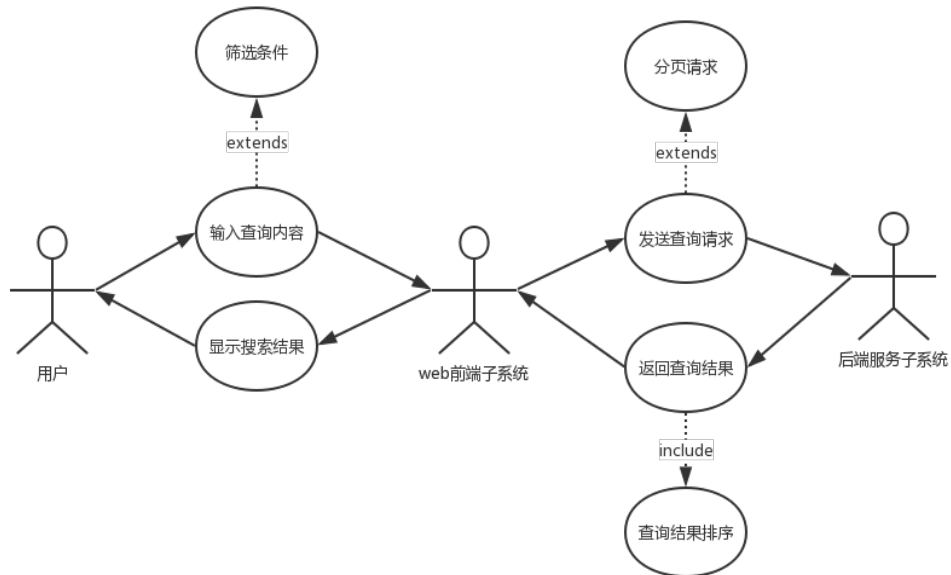
(2)采用如下的服务端、客户端配置：

服务器：以 Node.js 以及 Java Spring Cloud 为 Web 服务器，服务器端采用 Java 语言编写，数据库采用 MongoDB。

客户端：浏览网页主要采用谷歌浏览器，通过 React 技术进行网页的动态交互，以声明式编写 UI，编写更加可靠且方便调试的代码。并使用 Material UI 对客户端进行美化。

3.5 系统用例设计

3.5.1 Web 前端子系统用例图



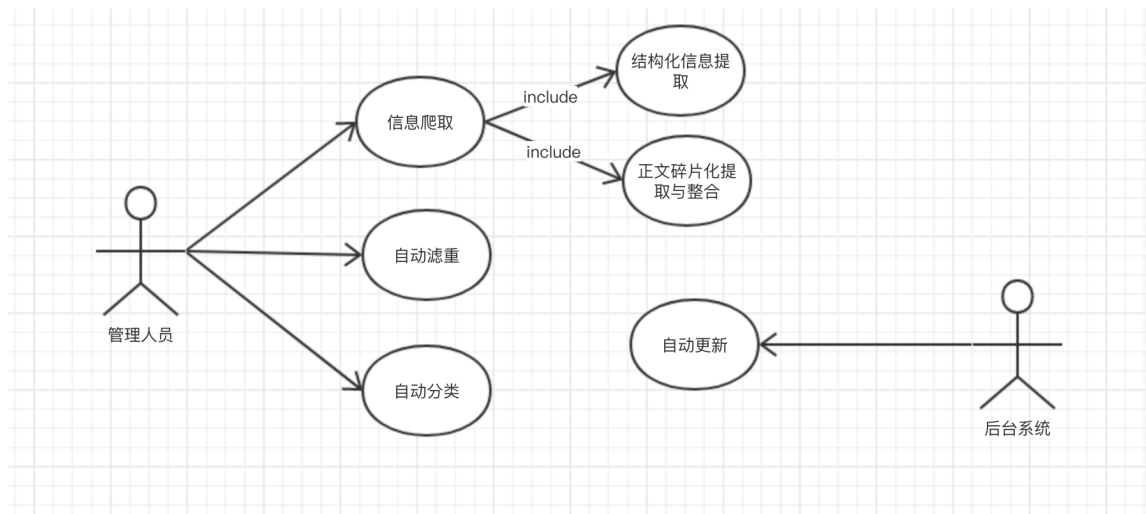
3.5.2 搜索引擎用例图



3.5.3 后端 Web 服务器用例图



3.5.4 爬虫子系统用例图



4. 系统实现时间表与里程碑

项目阶段	里程碑名称	里程碑事件	重要交付物
系统实现	编码	7.13	
	单元测试	7.14	
	BUG 修改	7.14	
	继承计划与方案	7.14	编码与系统实现计划
	进行集成	7.15	集成代码
	进行测试	7.15	测试代码
	BUG 修改	7.16	
	继承测试报告	7.17	测试报告
系统验收	编写验收测试计划		项目总结计划
	完成验收		

5. 测试策略

5.1 测试方案

整个测试方案基本上以黑盒测试或灰盒测试为主，测试将整个系统分为几个项目模块进行单元测试，测试依照被测系统的评价标准种类进行划分。

在不同的测试模块中根据模块的特性使用合适的测试方法，测试用例的选择基本上遵循多而全的准则，以确保测试的覆盖率，但在不同的项目中也有不同的侧重点，如数据及数据库集成测试侧重于实现的完整性和可用性;在功能测试中，基于用例的描述，将测试数据分为有效以及无效，在两种情况下对一个功能进行测试，侧重于功能的完备性。在用户界面测试中，考虑用户可进行的所有操作进行测试用例的设计，侧重于用户界面的可用性;在安全测试中，更改测试方法，使用边界测试等方法对系统在各种情况下的表现进行测试，设计测试用例时一方面考虑常见的攻击手段进行用例设计，另一方面则是针对系统中可能潜在的编码疏忽进行检测;在负载测试中则是根据用户数量的分级设计测试用例，性能测试中则根据系统数据库中数据量的分级设计测试用例。

5.2 测试标准

本次测试中测试文档的编写、测试用例的编写、具体的执行测试以及测试中各项资源的分配和估算，都是以理解软件质量保证与测试课程的知识和方法为前提，软件的执行以系统逻辑设计构架为依据。

6. 数据字典

6.1 数据元素定义表

编号	数据元素名	类型	值域	说明
E1	搜索内容	Varchar(32)	4{[1..9 a..z A..Z _]}32	
E2	文章标题	Varchar(32)	8{[1..9 a..z A..Z _]}32	
E3	文章链接	Varchar(64)	8{[1..9 a..z A..Z _]}32	
E4	文章标签	列表		
E5	筛选条件	表单		

6.2 数据流定义表

编号	数据	来源	去向	组成	说明
D1	用户搜索内容	Web前端	后端服务	E1	
D2	搜索结果	搜索引擎	后端服务、Web前端	E1+E2+E3+E4	
D3	条目详情	Web前端	后端服务	E2+E3+E4	
D4	筛选搜索条件	Web前端	搜索引擎、后端服务	E1+E5	

6.3 外部项定义表

编号	外部项名称	输入数据流	输出数据流	说明
W1	用户	搜索内容	搜索结果	
		筛选条件		
		排序方式		

6.4 数据集成、留存与销毁

DIRD-1: 系统将持续保留用户基本信息与房屋信息，直至用户注销账户或被管理员删除。

DIRD-2: 系统将保留所有订单 3 年，包括对用户和对服务商

DIRD-3: 系统将保留最新的设备维护信息，最多 10 条/设备