CSDN 搜索引擎

概要设计说明书

作者：吴宇涛 李德宏

2021 年 11 月 3 日

1 引言

1.1 编写目的

该文档是关于“CSDN 搜索引擎”软件的功能和性能描述，详细阐述了针对用户需求定制的设计方案，对系统中的各项功能需求，技术需求、实现环境及所使用的实现技术进行了明确定义。同时，对软件应具有的功能和性能及其他有效性需求也进行了定义。此外，本说明书还明确了系统的数据结构和软件结构，还将给出内部软件和外部系统部件之间的接口定义，各个软件模块的功能说明，数据结构的 细节以及具体的装配要求，并作为软件设计阶段的主要输入。

1.2 背景

软件系统的名称：CSDN 搜索引擎
开发工具：python
开发者： 吴宇涛，李德宏

1.3 定义

用户：被授权使用 CSDN 搜索引擎系统的人员。

1.4 参考资料

[1]概要设计说明书标准[S].GB 8567-88.
[2]张海潘、牟永敏.软件工程导论.第六版：清华大学出版社

2 总体设计

2.1 简述

爬取 CSDN 博客,利用 Whoosh 实现倒排索引与排序,django 作为后端实现小型 CSDN 搜索引擎。并实现高亮、相关搜索等功能。

2.2 需求规定

（1）易用性
要求操作简单，快捷，功能分类清晰并能最大限度满足需要，避免复杂的选择，简化数据的输入。
（2）可靠性
系统运行稳定可靠，考虑系统在平时和峰值的情况下，安全可靠地运行并记录数据，确保不死机，确保不丢失数据。

（3）可扩展性
在设计中不仅考虑当前的业务需求，更应该满足未来业务量和需求的增长。
软件应具备逐步升级的能里，采用模块化设计，能添加新功能以满足需求，或对
各部分的功能灵活地进行升级，扩展。
（4）可管理性
软件应满足提供良好的应用操作维护界面，维护操作简单。管理员对数据库的运行进行监控以及维护。
（5） 灵活性
软件的设计和实现考虑到运行环境的变化，能够在运行环境变化的情况下正常使用。同时，软件兼容其他软件接口的变化，保证在不同运行环境，不同软件接口的情况下的正常使用。具体要求如下：
运行环境的变化：软件支持在 Windows Server 2003/Windows XP SP3 及
以上 Windows 系统部署运行。
同其他软件接口的变化：当其他软件的接口发生变化时，该软件应能够
适应接口的变化。
精度和有效时限的变化：软件能够方便的适应精度和有效时限的变化。
计划的变化或改进：软件具有足够的灵活性，可以支持将来有可能会出
现的需求更改或增加。

2.3 运行环境

硬件环境
处理器（CPU）：Pentium 133M 或更高
内存容量（RAM）：64M 或更高

软件环境
操作系统：Windows XP 及以上 简体中文

数据库服务器端
操作系统：Microsoft Windows 10
数据库管理系统：MySQL，配置 TCP/IP 协议

2.4 接口设计

2.4.1 界面框架设计

2.4.2 外部接口设计

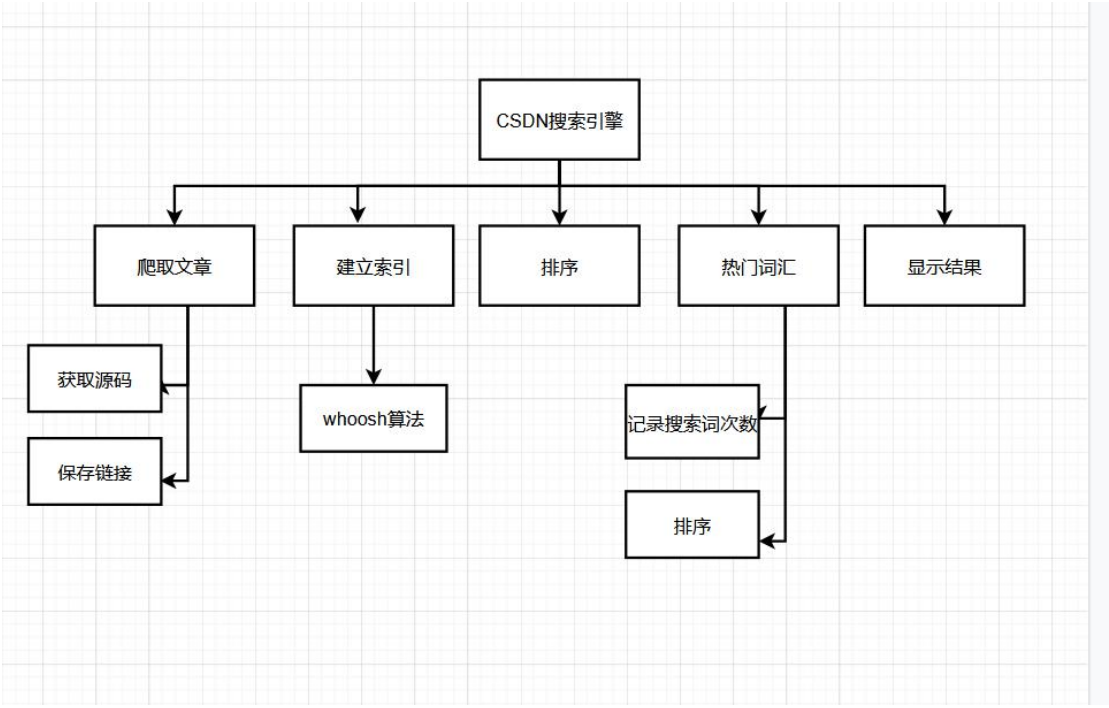
说明本系统同外界的所有接口的安排包括软件与硬件之间的接口、本系统与各支持软件之间的接口关系。

2.5 基本设计概念和流程处理

2.5.1 基本概念设计

我们实现的项目是“CSDN 搜索引擎”，此搜索引擎主要面对有一定计算机专业知识，经常使用 CSDN 平台交流学习的用户使用，主要为他们提供 CSDN 文章搜索，软件的界面简洁，操作简单，无任何广告，可以提高效率。此搜索引擎还提供热门搜索和排序功能，能帮助用户更快速找到想要的资源。

2.6 系统层次图



2.7 功能其与程序的关系

	用户模块	管理员模块
爬虫功能		√
搜索功能	√	√
排序功能	√	√
模拟登录功能		√

2.8 人工处理过程

1. 前端网页的优化与维护
2. 数据库备份与维护

3 接口设计

3.1 用户接口

显示搜索结果：
输入操作：输入关键字或文章标签

输出效果：显示按需排好序的搜索文章结果。

输入操作通过界面输入框进行。

4 运行设计

4.1 运行模块组合

4.1.1 模拟登录功能

输入：用户打开登录注册请求

处理：通过 SQL 查询数据库确认用户的账号密码信息是否正确，或者将用户注册的账号密码信息存入数据库

输出：登录/注册的信息

4.1.2 爬取数据功能

输入：文章链接 url

处理：请求获取文章源码，进行解析，保存到数据库

输出：文章解析后的文本文件，存储到数据库

4.1.3 搜索功能

输入：文章的关键字，标签

处理：点击搜索按钮，服务器查询并返回对应文章的信息到网页进行显示。

输出：相应文章的链接，标题，阅读量等。

4.1.4 排序功能

输入：点击排序按钮

处理：将搜索结果进行相应的排序，再显示出来

输出：按相应方式排序好的文章信息

4.1.5 热门搜索功能

输入：点击热门搜索的关键词

处理：搜索引擎按点击的关键词查询返回文章信息

输出：含有相关关键词的文章信息

INPUT 输入	PROCESS 处理	OUTPUT 输出
用户打开登录注册请求	通过 SQL 查询数据库确认用户的账号密码信息是否正确，或者将用户注册的账号密码信息存入数据库	登录/注册的信息
文章链接 url	请求获取文章源码，进行解析，保存到数据库	文章解析后的文本文件，存储到数据库
文章的关键字，标签	点击搜索按钮，服务器查询并返回对应文章的信息到网页进行显示。	相应文章的链接，标题，阅读量等。
点击排序按钮	将搜索结果进行相应的排序，再显示出来	按相应方式排序好的文章信息
点击热门搜索的关键词	搜索引擎按点击的关键词查询返回文章信息	含有相关关键词的文章信息

运行模块 IPO 表

4.2 运行控制

4.2.1 模拟登录功能

使用 IDE 打开 django 文件代码后，用户点击运行，浏览器弹出并显示“模拟登录”页面。

4.2.2 爬虫功能

使用 IDE 打开 django 文件代码后，用户点击运行，控制台显示爬取的链接，并且解析成为文本文件存入数据库。

4.2.3 登录注册功能

使用 IDE 打开 django 文件代码后，用户点击本地链接(127.0.0.1)，进入搜索引擎首页。再搜索框中输入文章关键字，返回相关关键字或标签的文章列表。

4.2.4 排序功能

使用 IDE 打开 django 文件代码后,用户点击本地链接(127.0.0.1),搜索文本并且按下回车，触发下一个 html 显示，文章有三种排序：按相关度排序、按时间排序、按阅读量排序。用户点击不同的排序方式可以把文章按想要的方式排序显示。

4.2.5 热门搜索功能

使用 IDE 打开 django 文件代码后,用户点击本地链接(127.0.0.1),搜索文本并且按下回车，触发下一个 html 显示，为用户显示搜索次数最高的五个关键词。

4.3 运行时间

软件使用过程中，用户在各个功能模块的触摸等操作事件的响应时间小于 1 秒。

对软件不同模块间的数据交互，要求数据的转换和传送时间不得超过 3 秒。

5 系统数据结构设计

5.1 逻辑结构设计要点

本系统根据功能需求分析结果建立 1 个数据库，包含 3 个数据库表，每个数据表完成相应的数据处理功能，存储相应的数据。

数据库名称	数据表名称	功能描述
csdn_crawler	Crawler_csdnblog	记录文章信息
	Search_engine_query	热门词汇信息
	Url_queue	待爬取文章链接
	visited	已爬取文章信息

5.1.1 文章信息表

数据表名称	字段	数据类型	描述

文章信息表 crawler_csdnblog	Url	varchar(100)	文章链接
	title	varchar(100)	文章标题
	writer	varchar(100)	作者昵称
	Writer_id	varchar(100)	作者 id
	Read_count	varchar(100)	阅读量
	date	varchar(100)	发布时间
	Content	varchar(100)	文章评论

5.1.2 热门词汇信息

数据表名称	字段	数据类型	描述
热门词汇信息 Search_engine_query	id	varchar(100)	唯一主键,订单唯一标识
	Query	varchar(100)	搜索的关键词
	Date	Date	文章发布的日期

5.1.3 待爬取文章链接

数据表名称	字段	数据类型	描述
待爬取文章链接 missions	Url	varchar(100)	唯一主键，等待爬取的文章的链接

5.1.4 已爬取文章信息

数据表名称	字段	数据类型	描述
已爬取文章信息	Url	varchar(100)	唯一主键，已爬取

missions			文章信息

5.2 物理结构设计要点

系统建立了 4 个数据表，分别完成不同的功能 crawler_csdnblog：记录文章信息，search_engine_query：记录热门词汇信息，url_queue：待爬取文章链接，visited：已爬取文章信息 所有数据在程序中使用 sql 插入查询等语句进行访问，最终由 MySQL 数据库管理系统持久化到磁盘 I/O 设备上加密保存。

5.3 数据结构与程序的关系

序号	数据表/模块	用户模块	任务模块	用户管理模块	任务管理模块
1	文章信息表	√		√	√
2	热门词汇表		√		√
3	待爬取文章链接		√		√
4	已爬取文章信息			√	√

6 系统出错处理设计

6.1 出错信息

可能出错或故障情况	处理方法
用户没有输入信息时提交搜索	弹窗提醒用户应输入信息
用户接受与计算机无关的搜索词汇	提示警告
用户同时发大量的数据请求操作	使用异步操作数据库
数据库被远程入侵或中病毒	对于恶意错误应马上停止数据库对外开发权限，并及时做好备份
系统由于严重错误奔溃无法运行	程序自动重启

6.2 补救措施

a. 后备技术：系统发布的信息数据都来源于数据库，界面只是把它显示出来，而且用户的发布任务，接受任务等所有操作都记录到数据库里，所以只需确保数据库以较高的频率备份数据，这样当系统因为突发情况崩溃时，系统可以根据日志文件和备份数据恢复。

b. 降效技术：可通过人工操作数据库等方法在系统出现问题时修正系统数据。

6.3 系统维护设计

系统维护工作主要在及时备份数据库数据，以防数据错误丢失时及时恢复，以及系统发生不可预知的严重错误时，人工重启服务器程序。