



桂林电子科技大学  
GUILIN UNIVERSITY OF ELECTRONIC TECHNOLOGY

# 课程设计报告

课题名称： 知网文献爬虫

课程名称： 软件工程课程设计

学 院： 计算机与信息安全学院

成 员： 1. 1701420218 黎业辉

2.

3.

4.

5.

指导教师： 王宇英

报告日期： 2020 年 12 月 7 日

表一 团队任务分工表

题目	知网文献爬虫		
负责人	黎业辉		指导教师 王宇英
序号	学号	姓名	个人任务描述
1	1701420218	黎业辉	绪论、需求分析、概要设计、数据库设计、详细设计的编写、软件开发、软件测试完成。
2			
3			
4			
5			

## 摘 要

知网文献爬虫是为了快速获取知网网页 HTML 中的各种有效信息，并对它们做分析处理，保存。而定制化开发的网络爬虫。

本项目的用户主要为在校的大学生或老师，供学术文献收集，或者其他学术用途。

本项目所使用的编程语言及开发库为 JavaScript 和 NodeJs，区别于写爬虫常用的 Python 语言。同时借助 JavaScript 在 Web 开发中的原生优势，可以写出更美观的 UI 界面，和响应式的组件。

本程序基于 MVC 架构模型来设计实现，其中的 View 组件单独提取出来，使用 ReactJs/Electron 框架进行开发，Model 和 Control 业务层则使用基础的 NodeJs 进行编程，UI 设计利用 CSS/HTML 库 Bootstrap 来实现。

**关键词：**知网；爬虫；NodeJs；Electron；MVC 架构模型

## 目录

1	绪论.....	1
1.1	背景和研究的意义.....	1
1.2	爬虫技术的现状.....	1
1.3	可行性分析.....	2
2	需求分析.....	3
2.1	模块分析.....	3
2.2	用例分析.....	4
2.3	E-R 图设计.....	7
3	概要设计.....	8
3.1	技术栈简介.....	8
3.2	软件结构层次.....	10
4	详细设计.....	14
4.1	核心模块设计.....	14
4.2	数据库设计.....	21
4.3	UI 设计.....	23
5	成本分析.....	28
5.1	总计成本.....	28
5.2	软硬件成本.....	29
5.3	开发成本.....	30
5.4	测试成本.....	31
6	测试计划.....	32
6.1	模拟请求模块.....	32
6.2	HTML 分析模块测试.....	33
6.3	XLSX 导入导出模块测试.....	34
7	总结.....	35
7.1	项目成果.....	35
7.2	不足.....	35
8	参考文献.....	36

# 1 绪论

## 1.1 背景和研究的意义

知网是由世界银行提出的，国家知识基础设施的概念。而中国知网是由清华大学、清华同方发起，始建于 1999 年 6 月。对于还在大学工作、学习的老师和学生们并不陌生。作为全球最大的中文文献数据库，涵盖资源丰富。各种用户可以找到文献数不胜数。多到用户一时间没办法通过手动搜索的方式将他们收集起来。

现代互联网的信息浩如烟海，不可能去每一个网页去点去看，然后再复制粘贴。所以我们需要一种能自动获取网页内容并可以按照指定规则提取相应内容的程序。网络爬虫是一种自动收集互联网信息的程序。使用广泛的搜索引擎就网络爬虫应用的一种。爬虫的主要作用就是将网页的关键信息进行一个大致的提取，汇总，再之后导入数据库供其他软件应用使用，或者导出为可读的文档格式。用于数据收集的网络爬虫，正好可以用于解决知网文献信息收集这一问题。

由上面的描述可知，需要一个定制化的数据收集爬虫，用于收集提炼各种知网的论文资料，本项目正是用于解决这一问题，帮助用户通过跨平台的爬虫技术快速获取和整理来自知网的文献资料。通过和数据库的交互，爬取记录可以存储在数据库中，在需要时也可以将数据导出，例如导出为 XLSX 文件。

## 1.2 爬虫技术的现状

理论上，只要编程语言有完整的对 HTTP 协议通信的支持（极少部分网站需要 Web Socket 协议支持），便可以用来编写网络爬虫。因此爬虫本身其实对语言的限制并不大。因此大多数爬虫是通过脚本语言编写的，其中 python 是编写爬虫的主流脚本语言，在 Python 库里也有许多与爬虫相关联的网络库 Scrapy、BeautifulSoup、Pyquery、Mechanize 等。而非脚本语言的编译型语言如 C++ 和 Java 和 Go 因为对高并发有更好的支持，因此在搜索引擎等需要高并发处理的爬虫的编写中具有更大的优势。值得关注的是另一门脚本语言 JavaScript，它是现在 WEB 页面主流的网页脚本语言，原生上就对网页信息处理具备优势。本项目使用的就是 JavaScript 作为编程语言。

### 1.3 可行性分析

#### 1.1.1 经济可行性分析

爬虫工具运行所需要的占用资源并不高，只需要支持对知网正常的网络访问能力，不需要非常高的带宽，知网爬取的并发要求也不是很大，因此对 CPU 的需求也不大，普通的 PC 机或笔记本即可提供最基础的支持。一般来说价格在 3500 到 7000 区间的电脑即可满足爬虫的运行需要。在经济的可承受范围之内。

综上所述，本课题在经济上是可行的。

#### 1.1.2 技术可行性分析

本项目基于 MVC 的设计架构进行开发，由 JavaScript 通过 Electron 同时提供前后端支持（本项目的爬虫使用类似 WEB 前后端分离的技术）。JavaScript 在引入 WEB 原生的前端技术支持之后，对 UI 的编写有得天独厚的优势。目前 JavaScript 的 WEB 页面开发有两大热门框架 Vue 和 ReactJs，这两门语言都可以写出响应式的 WEB，再结合 Bootstrap，Layui，ElementUI 等成熟 UI 框架的支持，可以写成具有良好交互性的美观的 UI。

随着近几年 Nodejs 技术的火热，Nodejs 的第三方开源库中亦不乏对网络访问有良好支持的网络库。

同时本项目使用 JavaScript 的超集 TypeScript 作为直接的开发语言，TypeScript 的存在类似于 C++ 之于 C，能给 JavaScript 项目的开发带来更好的规范和特性支持，在 JavaScript 的基础上提供了更多特性支持，例如镜像类型。TypeScript 最终也会被编译到 JavaScript 由 Nodejs 运行。

以基于对以上技术的掌握来开发爬虫工具的难度并不大，在一定时间内的学习中，大致掌握即可达到开发要求。

综上所述，本项目在技术上是可行的。

## 2 需求分析

### 2.1 模块分析

知网论文爬虫工具的用户而言，需求来自三点实际需要

- 1.自动化地进行重复的、大量的、信息获取。
- 2.结构或半结构化的信息提取和转换。
- 3.可自定义参数和拓展功能的良好工具。

从这些需要中便可以得到三个大的模块，自动化爬取模块、信息解析处理模块、参数调整模块。而这些大的模块便可划分为小的实际的模块。

对于自动化搜索（爬取）模块而言主要的功能在于模拟一般用户行为收集，所以又可分为：粗略搜索、定向信息获取。

而信息解析处理模块可分为网页内容解析、数据库交互模块、xlsx 文件导入导出模块。

参数调整模块对访问中需要用到的各种参数做一个汇总的管理，给用户提供一个可以调整操作的接口。如爬虫参数的调整，数据库连接参数的调整。

利用层次方框图可直观地展示系统的模块结构系统的层次方框图，如下图 2-1 所示。

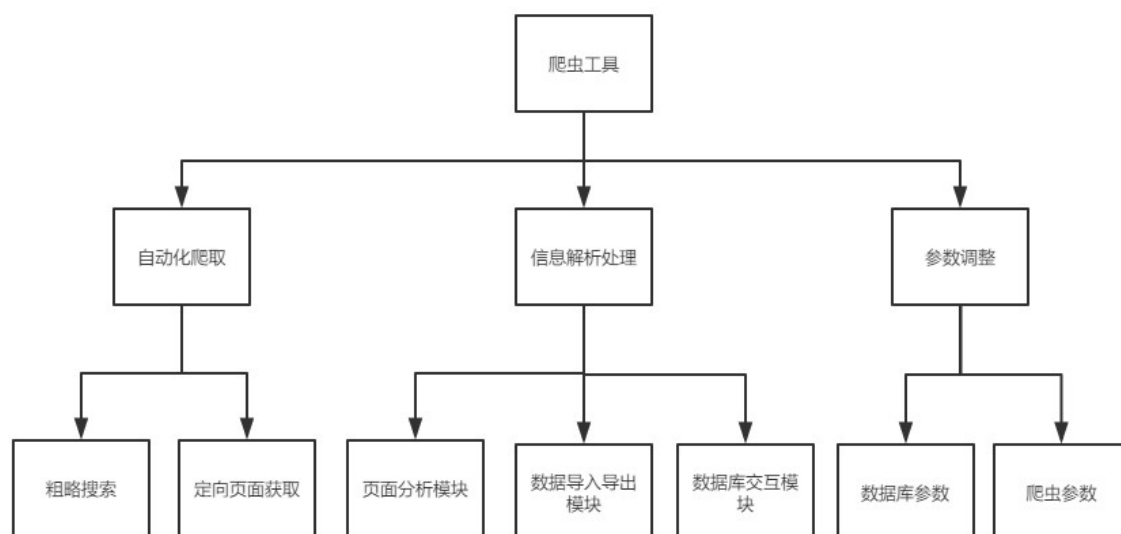


图 2-1 系统层次方框图

## 2.2 用例分析

对于爬虫的用户而言，用例主要分为两大方面【爬虫任务管理】，【历史任务查看】，【爬虫设置】。

【爬虫任务管理】包括【添加爬虫任务】，【取消爬虫任务】、【暂停爬虫任务】。用户可以在管理界面对爬虫正在进行的任务进行管理，并从实时的反馈中查看当前状态，并可以将爬虫结果导出。或是取消/添加爬虫。

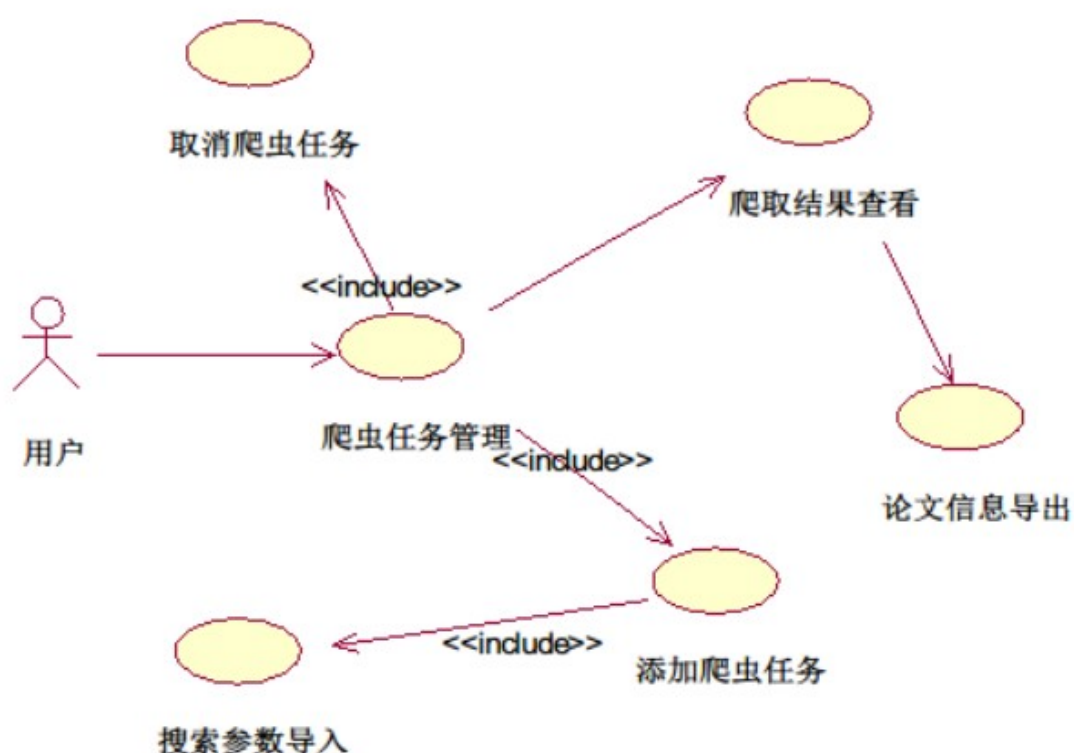


图 2-2 爬虫任务管理用例图



【历史爬虫任务查看】则有【论文信息导出】、【参数导出】两大功能。历史任务任务界面，用户可以通过查看数据库中的记录，查看进行过的爬虫任务，对任务选中后，可以查看任务的爬取记录，并进行导出。

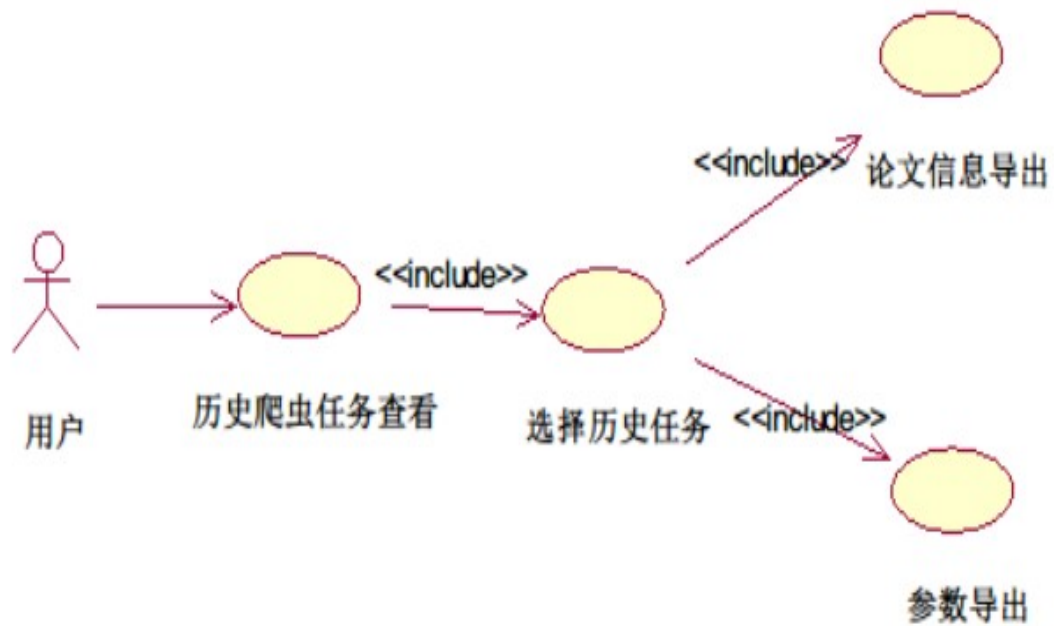


图 2-3 历史爬虫任务查看用例图

【爬虫设置】包括【延迟间隔调整】、【Cookie 设置】、【数据库参数调整】。爬虫设置界面中用户可以配置数据库的连接参数。也可以通过调整延迟间隔调整，调整爬虫的爬取速度。Cookie 设置用于设置爬虫模拟一般用户所用到的 Cookie。

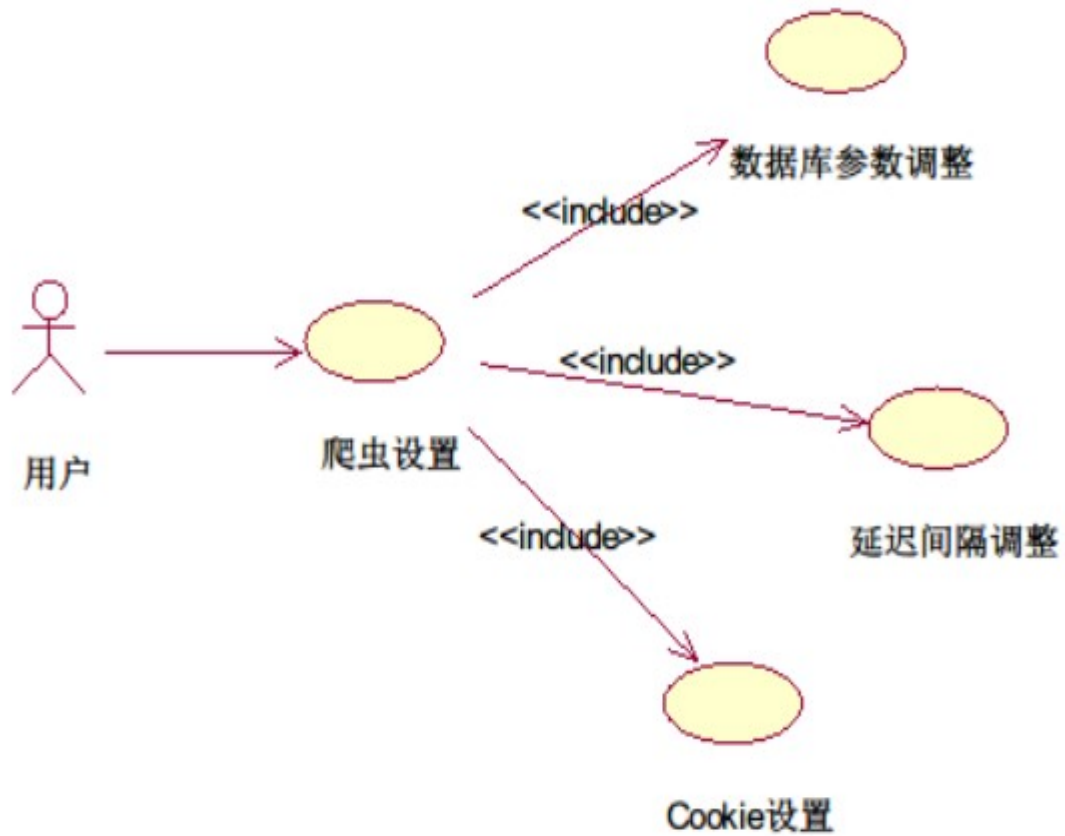


图 2-4 爬虫设置用例图

### 2.3 E-R 图设计

数据库主要记录的是爬虫的任务信息，包括爬取条件、爬取时间。和爬虫爬取的论文记录。

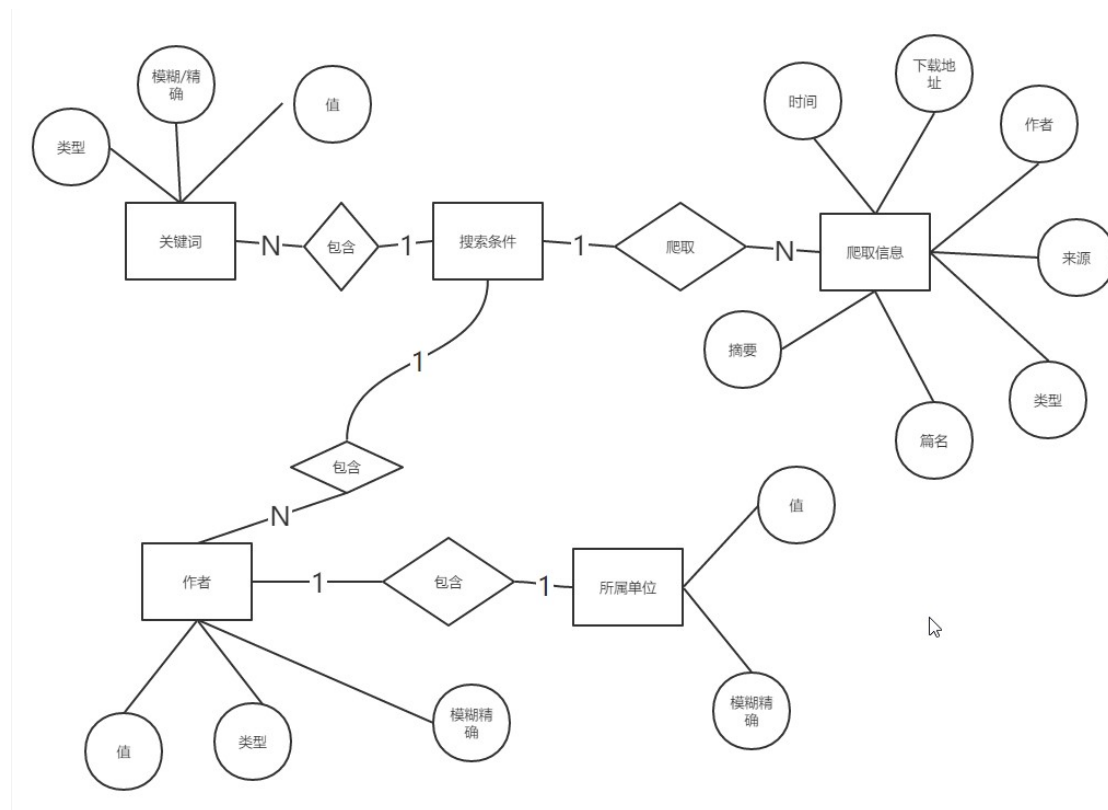


图 2-5 爬取记录 E-R 图

## 3 概要设计

### 3.1 技术栈简介

项目的爬虫软件是通过 JavaScript/NodeJs/Electron 实现的跨平台桌面应用程序。可以在 Windows, Linux 等系统环境下运行。前端通过 ReactJs 和 Bootstrap 进行 UI 绘制和组件响应处理。后端使用 got/cheerio 进行爬取和分析。项目使用 Mysql 数据库存储记录。

#### 3.1.1 JavaScript/NodeJs

在本项目中，使用 JavaScript 作为开发语言，Nodejs 作为开发环境。

JavaScript 是主流的 web 网页脚本语言，相对于写爬虫常用的 python 语言来说也不乏自己的优势。

Nodejs 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。其作用类似于 JavaSE Development Kit (JDK) 对于 JAVA 的作用，可以让 JavaScript 脱离浏览器 WBE 环境单独运行，使得 JavaScript 可以用于编写后端服务或桌面应用。

#### 3.1.2 Electron

Electron 是基于 NodeJs，使用 JavaScript，HTML 和 CSS 构建跨平台的桌面应用程序的开发框架。可以将成熟的 WEB UI 开发引入桌面应用的开发。

Electron 的工作方式是将 JavaScript 程序分作两部分运行，渲染进程运行于 Web 环境下，主进程运行于 NodeJs 环境下，用类似 WEB 应用程序前端和后端分离的方式进行开发，如下图 3-1。微软的轻量级 IED Visual Studio Code 便是使用这一技术开发的。

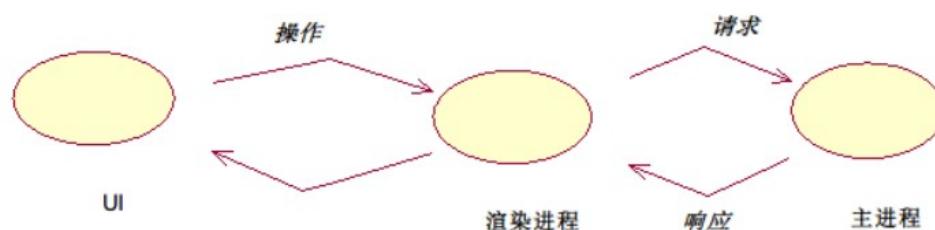


图 3-1 Electron 的工作方式

### 3.1.3 ReactJs/Bootstrap

ReactJs 是 web 前端开发框架，用于用于构建用户界面。ReactJs 是具有单向数据流响应的函数式设计的 JavaScript 框架。

而 Bootstrap 是 Twitter 推出的一个用于前端开发的开源工具包。它由 Twitter 的设计师 Mark Otto 和 Jacob Thornton 合作开发,是一个 CSS/HTML 框架。

### 3.1.4 Got/Cheerio

Got, 人性化且功能强大的 HTTP 请求库，用于网页请求模拟。并且对请求处理做了一定程度的简化，适用于快速构建客户端请求。

Cheerio 是 jquery 核心功能的一个快速灵活而又简洁的实现，用于分析网页请求结果。

### 3.1.5 Mysql

Mysql 是应用广泛的开源关系型数据库，被广泛地应用于各种项目开发之中。可以给本项目有兼容性的数据库支持。

### 3.2 软件结构层次

软件采用三层架构设计，即表示层（UI）、业务逻辑层（BLL）和数据访问层（DAL）。

表示层负责绘制用户 UI，通过 ReactJs 框架编写。业务逻辑层是爬虫软件的核心部件。主要负责爬虫的管理和数据分析。数据访问层负责与数据库进行交互，提供增删查改功能。

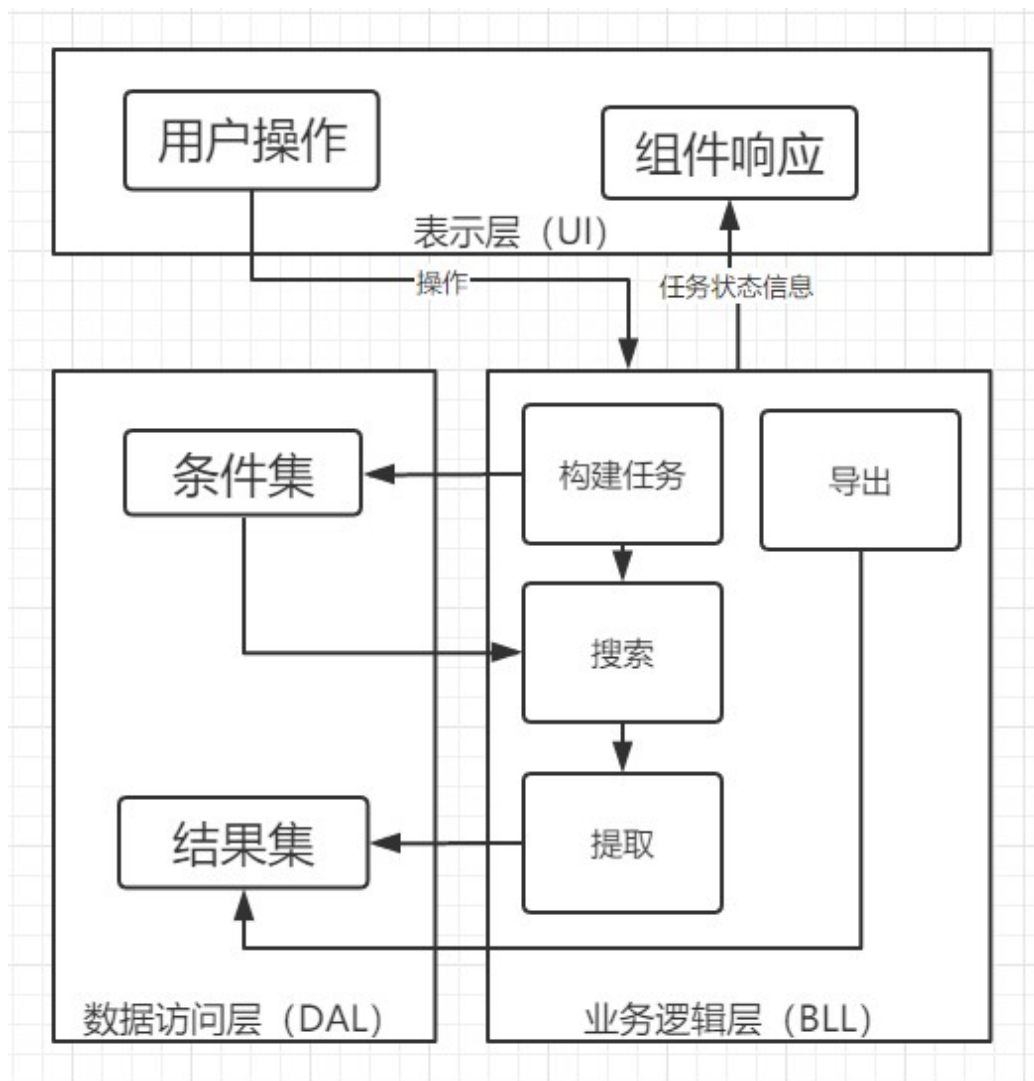


图 3-2 软件架构设计图

### 3.2.1 表示层结构

表示层分为三个部分，组件渲染、用户操作处理、前后端交互。组件渲染负责 UI 的绘制和组件响应的处理。用户操作，对用户的输入进行处理。前后端交互负责程序的渲染进程和主进程进行交互

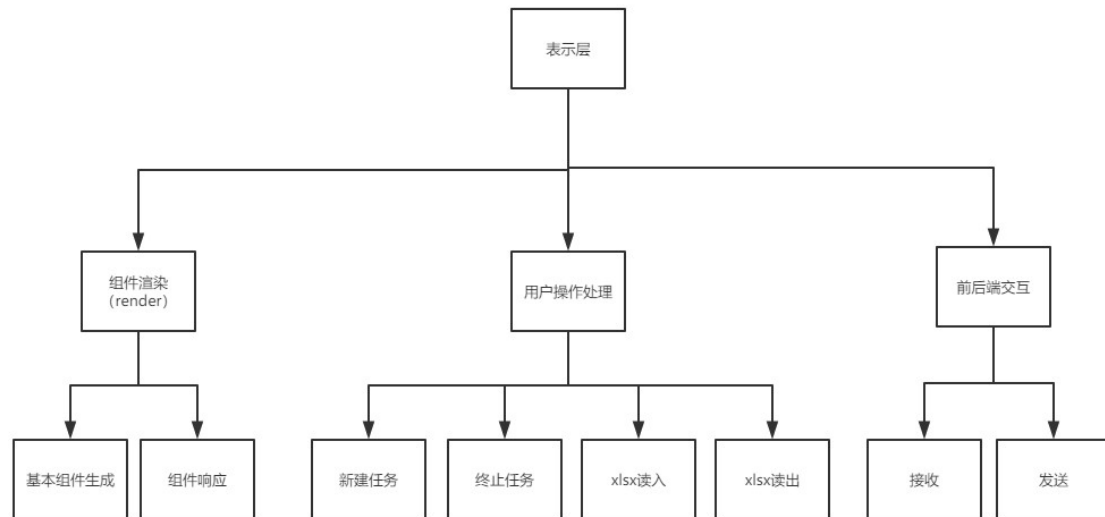


图 3-3 表示层结构图

### 3.2.2 业务逻辑层核心爬取流程

业务逻辑层的核心功能就是对知网的信息爬取。爬取过程中由前后端交互提供参数。然后传入模拟请求模块模拟用户请求，得到搜索结果，然后搜索结果进行分析，得到各个论文的详情页，然后模拟访问详情页，再对详情页进行分析，得到最终结果。

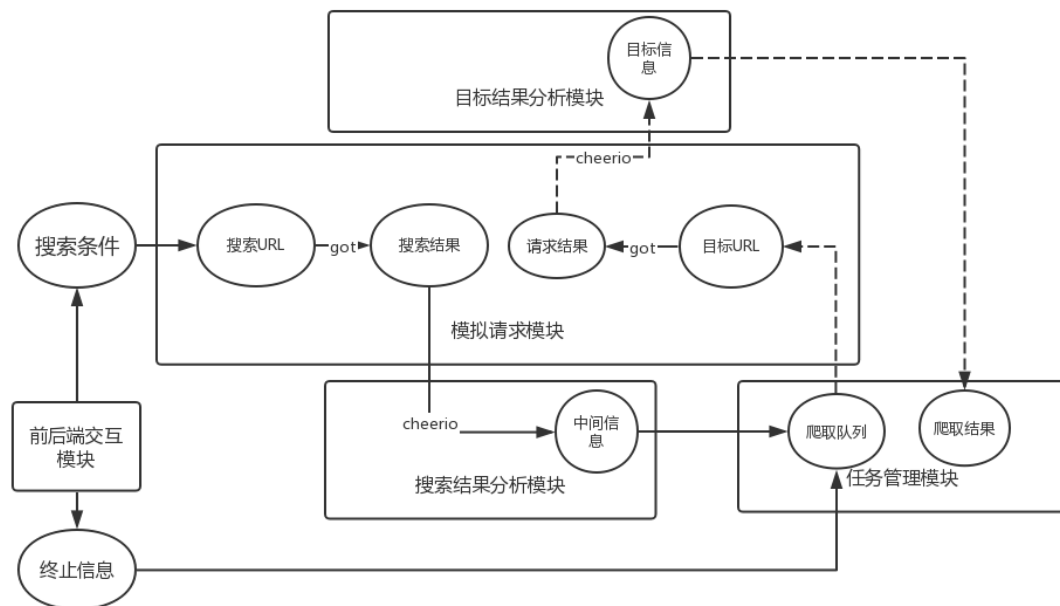


图 3-4 业务逻辑层爬取流程图



### 3.2.3 数据访问层结构

数据访问层主要对爬取记录和爬取结构进行管理，并且对爬虫进程的多线程操作进行处理。

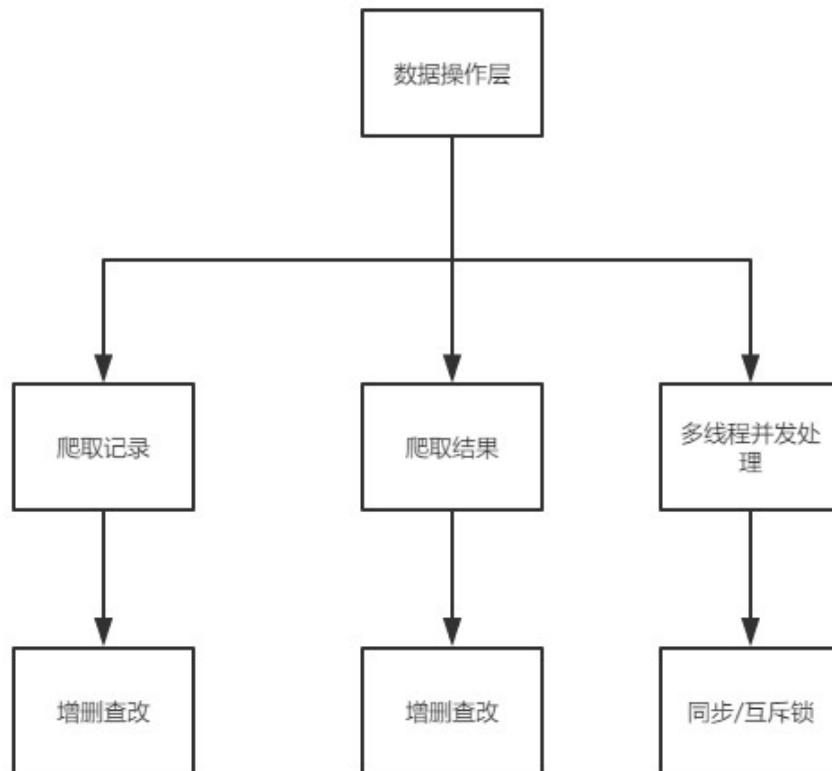


图 3-5 数据访问层结构图

## 4 详细设计

### 4.1 核心模块设计

#### 4.1.1 相关类图

通过搜索任务这个对象将用户的操作进行记录,并将爬取记录和搜索条件关联起来。

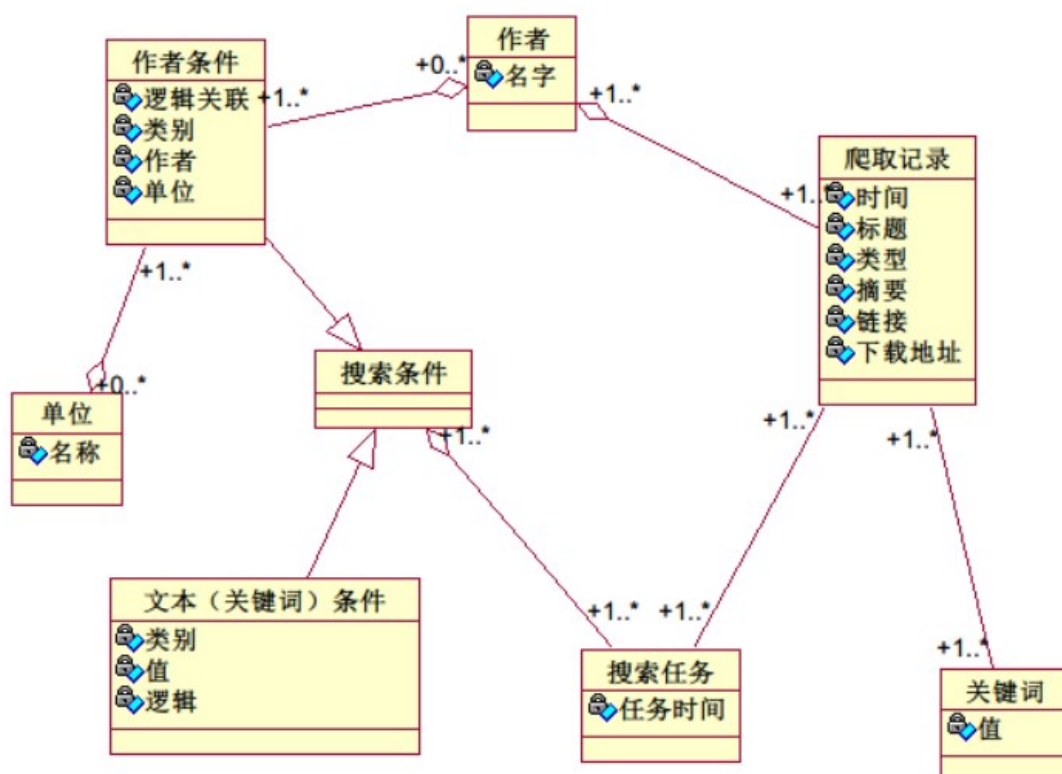


图 4-1 搜索任务关联类图

#### 4.1.2 用户创建爬取任务

用户使用爬虫爬取信息时，先导入搜索条件，然后爬虫将会构建搜索任务，每次获取搜索结果之后，对文献详情页再进行一次深入的爬取。然后将数据记录，一直循环至搜索结果为空。

活动图：

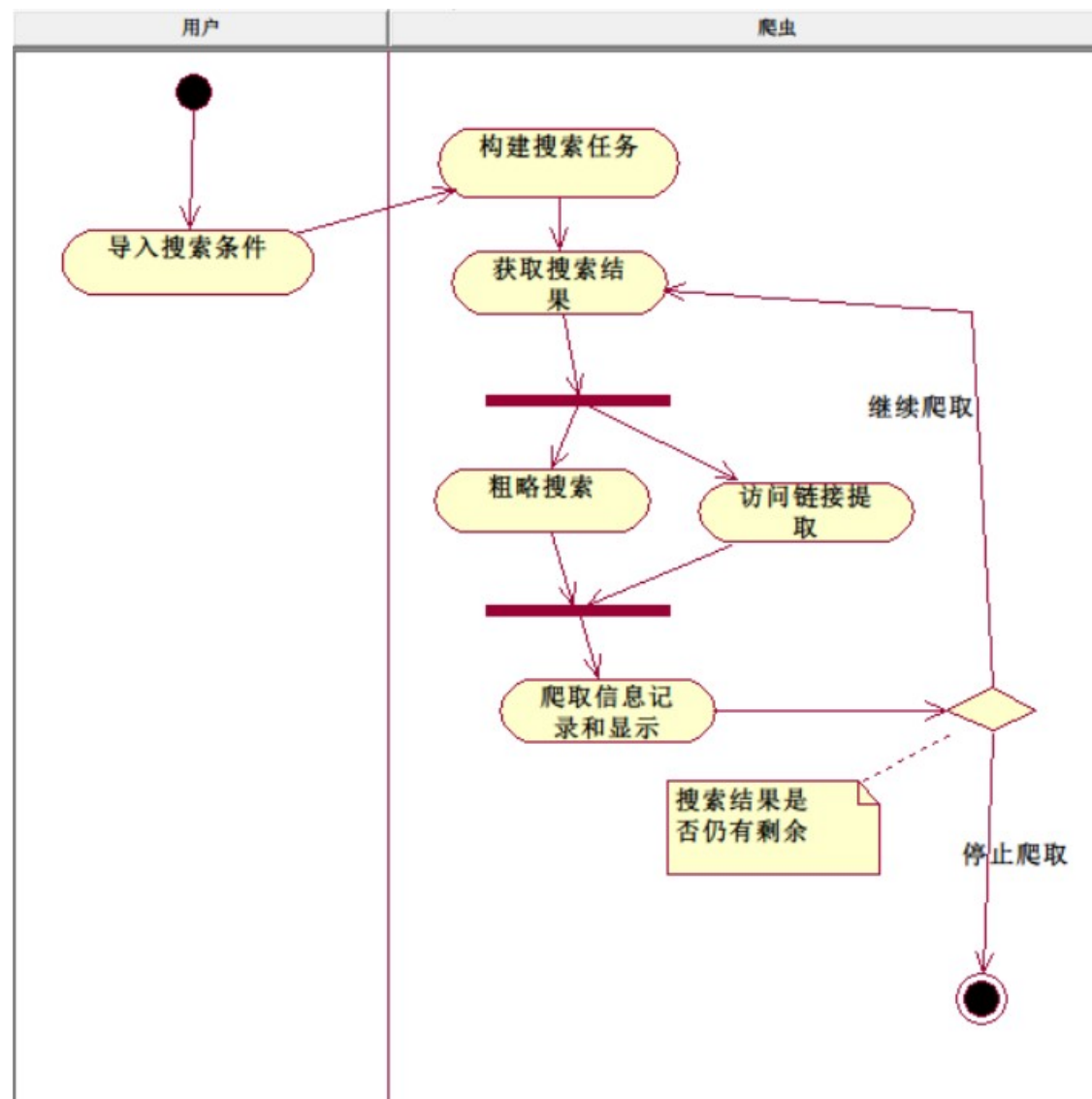


图 4-2 创建爬取任务活动图

时序图：

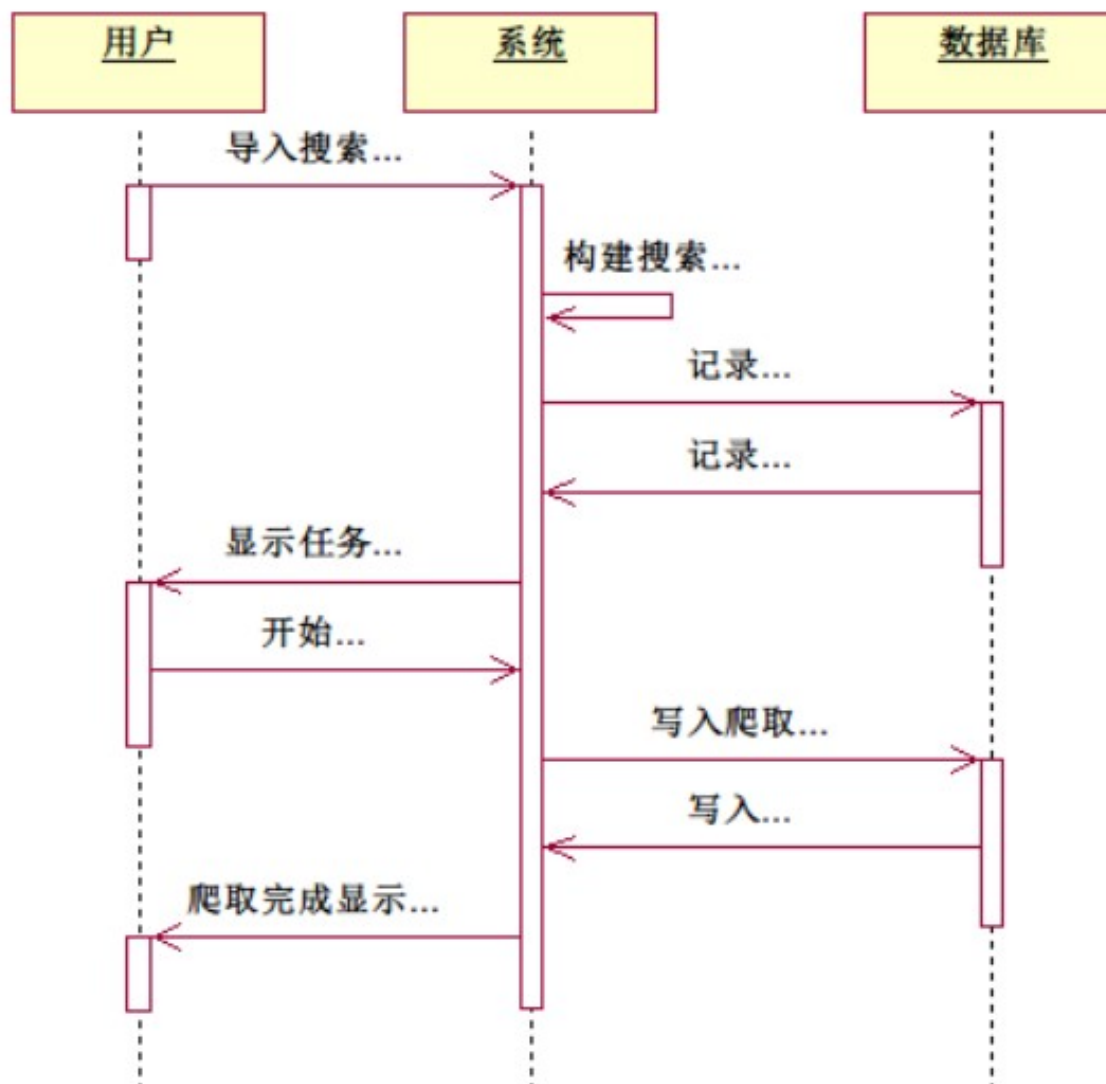


图 4-3 创建爬取任务时序图

状态图：

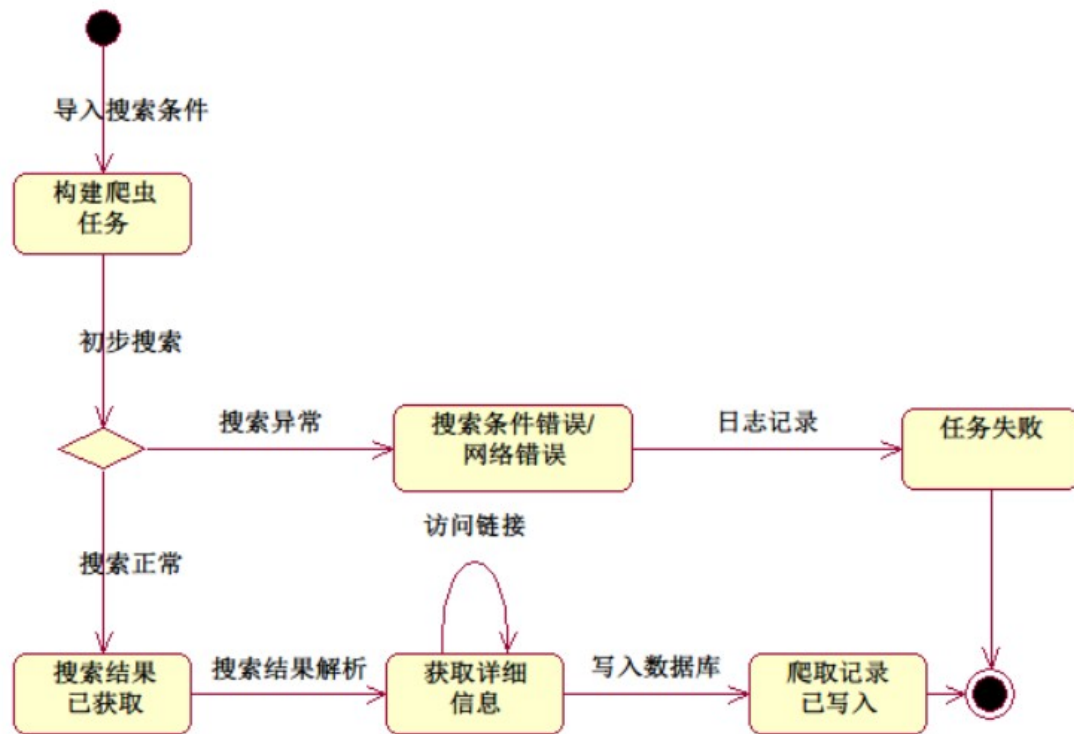


图 4-4 创建爬取任务状态图

### 4.1.3 用户导出历史任务

用户导出历史任务时，会先通过 UI 界面查询历史任务，然后再选择历史任务，软件会构建 XLSX 文件并向其写入爬取的信息记录。

活动图：

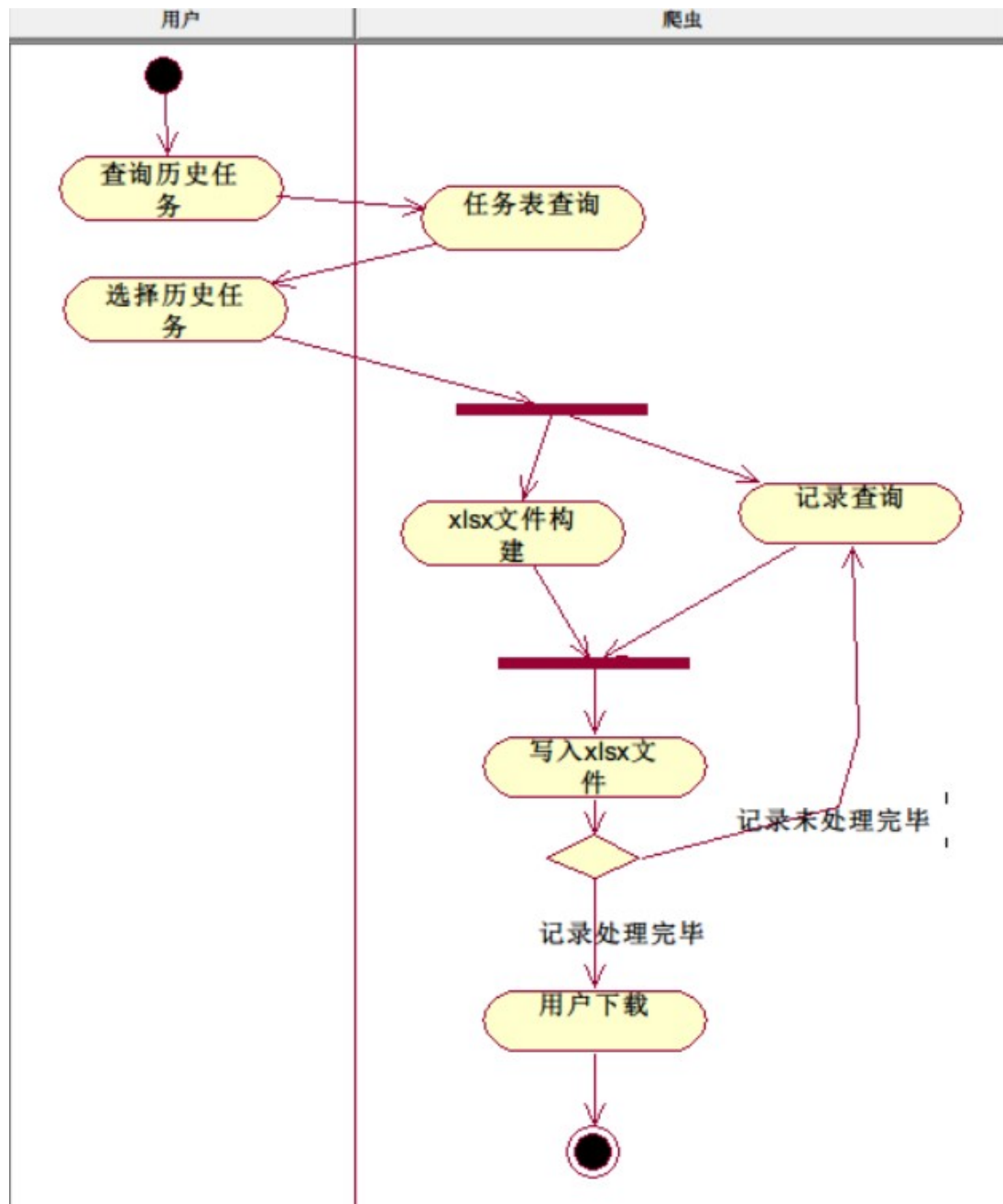


图 4-5 导出历史任务活动图

时序图：

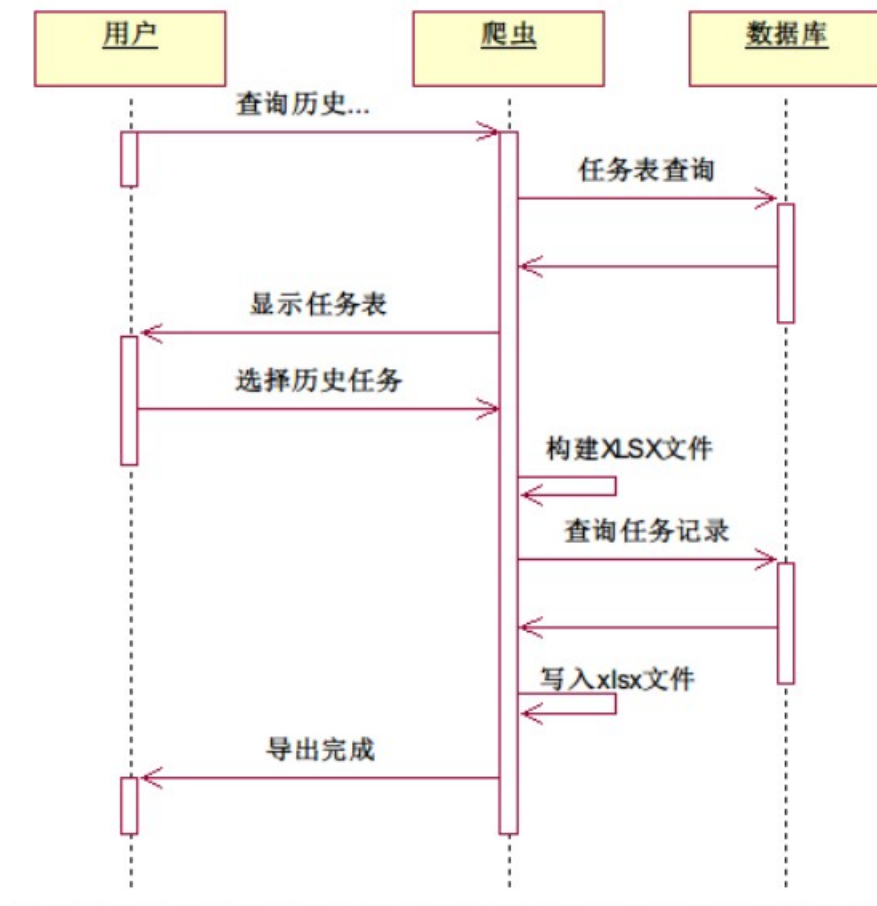


图 4-6 导出历史任务时序图

#### 4.1.4 用户修改爬虫设置

用户修改爬虫设置时，UI 界面会加载当前的设置，用户在 UI 界面修改爬虫的参数值，然后点击保存，程序会将设置保存到本地文件。

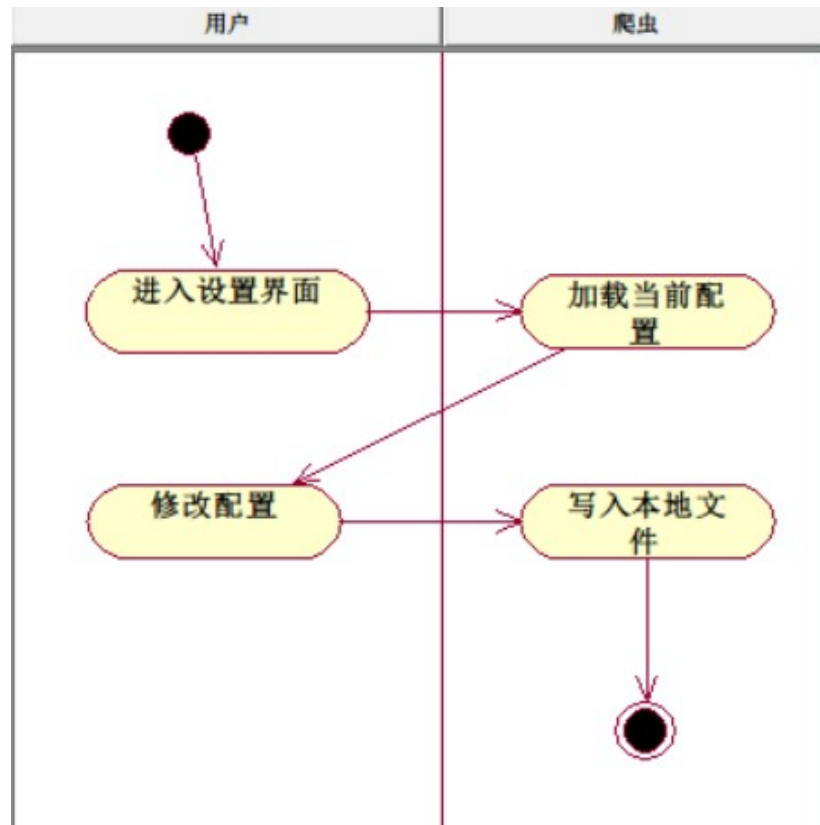


图 4-7 修改爬虫设置活动图



## 4.2 数据库设计

### 4.2.1 爬取记录表【TaskRecords】

爬虫爬取的文献信息将会记录到创建爬取记录表【TaskRecords】中。其中 authors 字段和 keywords 字段直接存储在表中，冗余存储，不再另建表。

字段名	数据类型	字段描述	约束性	可空	默认值
task	VARCHAR	任务名	主键、外键	否	无
index	INT	记录序号	主键	否	无
title	TEXT	标题	无	否	无
authors	TEXT	作者	无	否	无
origin	TEXT	来源	无	否	无
date	TEXT	日期	无	否	无
type	TEXT	文献类型	无	否	无
download	TEXT	下载地址 URL	无	否	无
link	TEXT	知网详情页 URL	无	否	无
keywords	TEXT	关键词	无	是	无
abstract	TEXT	摘要	无	是	无

表 4-1 爬取记录表【TaskRecords】

#### 4.2.2 历史任务表【HistoryTask】

爬虫爬取的历史任务会记录到创建爬取记录表【TaskRecords】中。其中爬取条件具有半结构化的性质，冗余存储在 query 字段中。JSON 是 MYSQL 特有字段，和 TEXT 有等效性。

字段名	数据类型	字段描述	约束性	可空	默认值
task	VARCHAR	任务名	主键	否	无
time	BIGINT	任务时间戳	无	否	无
query	JSON	查询条件	无	否	无

表 4-2 历史任务表【HistoryTask】

## 4.3 UI 设计

### 4.3.1 主界面

主界面下共有三个功能菜单分别是：爬虫模块，历史记录模块设置模块

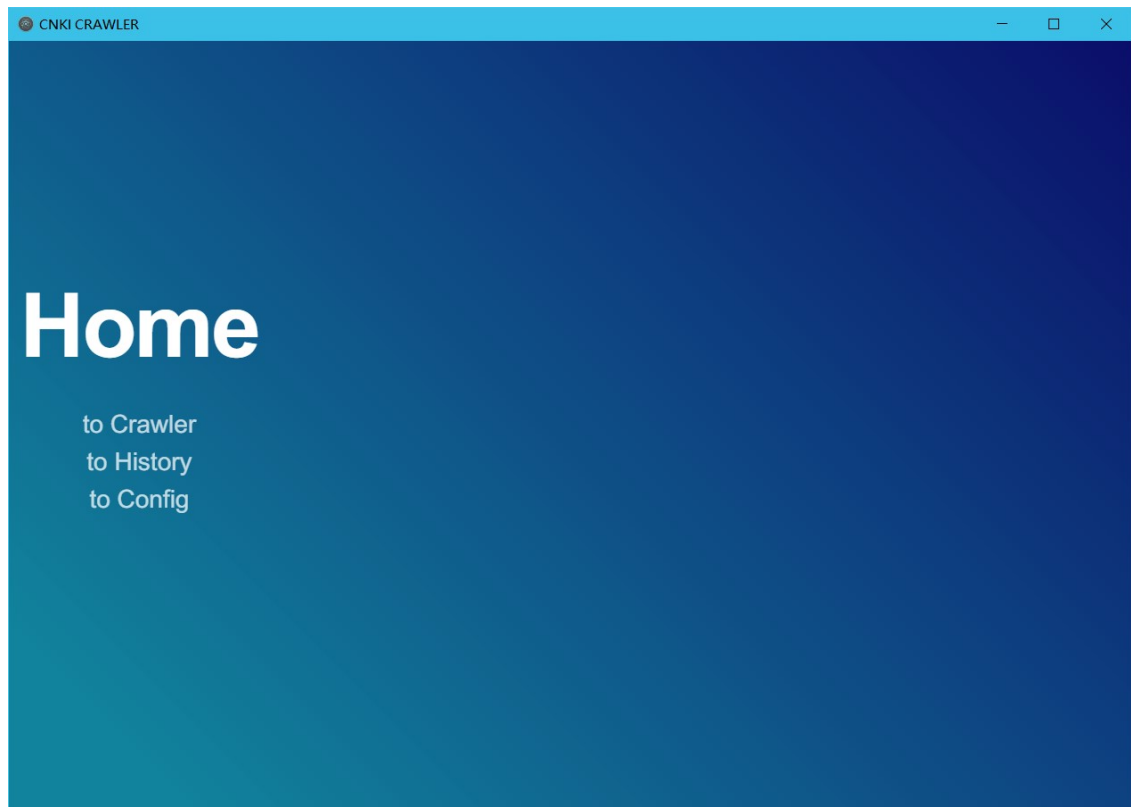


图 4-8 主界面

### 4.3.2 爬虫界面

在爬虫模块中，可以导入 xlsx 文件将爬取条件导入，点击开始后，在右上角会响应当前任务状态。



图 4-9 爬虫界面任务进行中

当爬取完成后，在下方的表格可以查看爬取结果，同时可以将爬取结果保存为 xlsx 文件。

查询条件XLSX: D:\query.xlsx

任务[TASK-1607316601422]已停止

打开

开始

保存

index	title	authors	origin	date	type
11	基层首席公共卫生医师制度的实践与探索研究	程清忠; 李战胜; 王爱莲; 肖爱华; 张伟; 王宪祥; 王胜; 于洋	中国农村卫生	2020-11-15	期刊
12	Purification and dissociation of raw palygorskite through wet ball milling as a carrier to enhance the microwave absorption performance of Fe <sub>3</sub> O <sub>4</sub>	Wang Sheng; Ren Hengdong; Lian Wei; Wang Jiangzhe; Zhao Yan; Liu Yin; Zhang Tianshu; Kong Ling Bing	Applied Clay Science	2020-11-13	外文期刊
13	基于互感和霍尔效应的地下位移测量系统设计	王胜; 申屠南瑛; 李青; 王丰	科技通报	2020-10-31	期刊
14	A field trials-based authentication study of conventionally and organically grown Chinese yams using light stable isotopes and multi-elemental analysis combined with machine learning algorithms	Lyu Chaogeng; Yang Jian; Wang Tielin; Kang Chuanzhi; Wang Sheng; Wang Hongyang; Wan Xiufu; Zhou Li; Zhang Wenjin; Huang Luqi; Guo Lanping	Food Chemistry	2020-10-31	外文期刊
15	Oxygen - Deficient Bimetallic Oxide FeWO <sub>x</sub> Nanosheets as Peroxidase - Like Nanozyme for Sensing Cancer via Photoacoustic Imaging	Gong Fei; Yang Nailin; Wang Yong; Zhuo Mingpeng; Zhao Qi; Wang Sheng; Li Yonggang; Liu Zhuang; Chen Qian; Cheng Liang	Small	2020-10-26	外文期刊

图 4-10 爬虫界面任务完成

### 4.3.3 历史任务界面

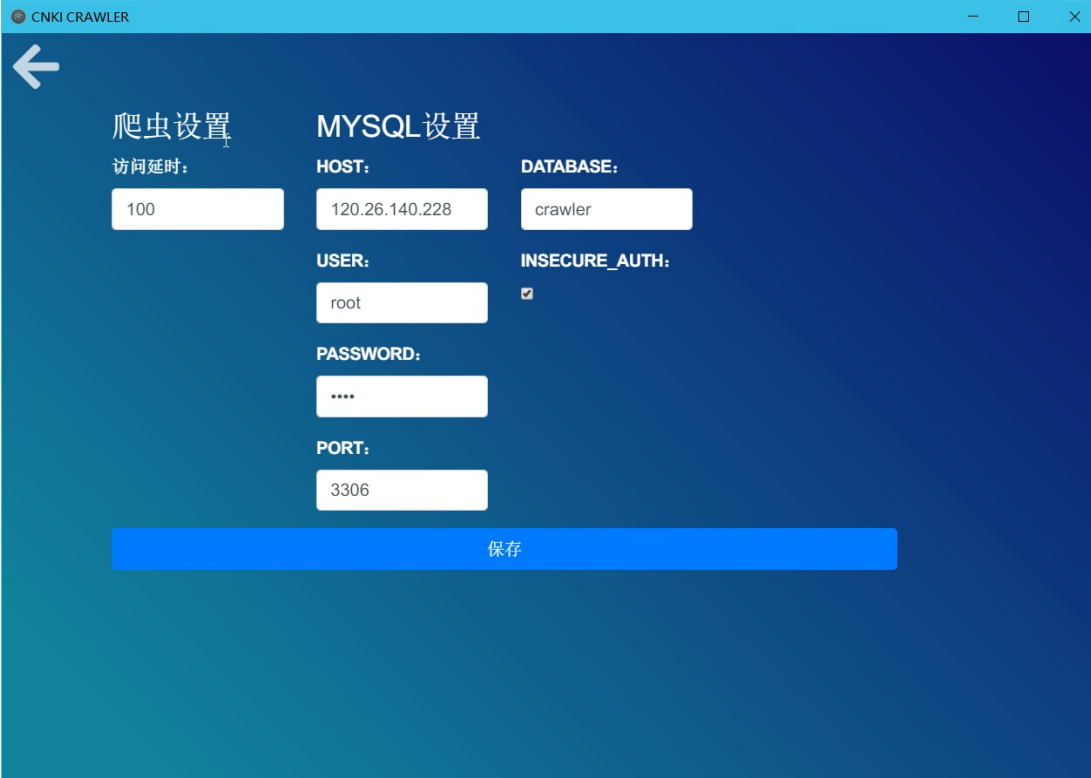
在历史模块中，在上方的表格中可以查看历史任务，选中读取后，可以在下方的表格查看任务爬取的记录。同时也可以保存到 xlsx 文件。



图 4-11 历史任务界面

#### 4.3.4 设置界面

在设置模块中：可以对爬虫的延时访问参数进行设置。也可以对 mysql 数据库进行设置。



The screenshot shows the 'CNKI CRAWLER' application window. It features a dark blue background with white text and input fields. On the left, a back arrow icon is visible. The settings are organized into two columns. The first column, titled '爬虫设置', contains a label '访问延时:' followed by a text input field containing '100'. The second column, titled 'MYSQL设置', contains several labels and input fields: 'HOST:' with '120.26.140.228', 'DATABASE:' with 'crawler', 'USER:' with 'root', 'PASSWORD:' with a masked field '\*\*\*\*', 'PORT:' with '3306', and 'INSECURE\_AUTH:' with a checked checkbox. At the bottom center, there is a prominent blue button labeled '保存'.

图 4-12 设置界面

## 5 成本分析

### 5.1 总计成本

详细人力成本见开发成本和测试成本。

类型	细则	人*天	合计	工资/天*人*元	总计/元
人工成本	需求分析	3	25	300	7500
	系统设计	2			
	程序开发	14			
	软件测试	6			

表 5-1 人力成本总计

软件整体开发成本包括软硬件成本人工成本

类型	合计/元	总计/元
软硬件成本	500	8000
人工成本	7500	

表 5-2 成本总计表



## 5.2 软硬件成本

类型	备注	名称	单价/元	件数	合计/元	总计/元
开发 ide	开源免费	vscode/nodejs	0	~	0	500
测试工具	开源免费	jest	0	~	0	
数据库	开源免费	mysql	0	~	0	
电脑	折旧费用		500	1	500	

表 5-3 软硬件成本表

## 5.3 开发成本

类型	细则	备注	人*天	总计/(人*天)
需求分析	需求调研	进行需求调研	1	3
	需求分析	需求分析的主要内容是系统主要输入内容（搜索条件）及输出（输出文件格式）	2	
系统设计	架构设计	系统架构设计及评审	2	2
程序开发/ 前端	组件渲染	react 开发	3	7
	用户操作处理	electron 开发	2	
	导入导出		1	
	前后端交互		1	
程序开发/ 后端	分析模块	Cheerio 开发	2	7
	模拟请求模块	Got 开发	2	
	任务管理模块	Nodejs 开发	2	
	数据库交互	Nodejs 开发	1	

表 5-4 开发成本表

## 5.4 测试成本

类型	细则	备注	人*天	总计/(人*天)
软件测试	前期	详见测试计划	3	6
	中期		2	
	后期		2	

表 5-5 测试成本表

## 6 测试计划

### 6.1 模拟请求模块

类型：单元测试、白盒测试

阶段：前期

具体时期：模拟请求模块完成基本功能实现。

测试过程：检查 http 请求状态码和 response 报文是否与正常的 web 浏览器请求相似或一致。

测试目的：判断模拟 web 浏览器浏览知网的功能是否符合要求。

测试用例：

错误是否已修正：无错误

编号	模块输入	备注	实际结果	预期结果	评价
1	https://www.cnki.net/	知网主页	Status 200	Status 200	通过
2	https://kns.cnki.net/kns/request/SearchHandler.ashx	知网搜索 api	Status 200 内容：结果 API	Status 200 内容：结果 API	通过
3	https://kns.cnki.net/kns/request/GetWebGroupHandler.ashx 没有 Cookie	知网结果 api	返回空白 HTML	返回空白 HTML	通过
4	https://kns.cnki.net/kns/request/GetWebGroupHandler.ashx 有相应 Cookie	知网结果 api	返回有内容 HTML	返回有内容 HTML	通过
5	127.0.0.1	无效访问	服务器无响应	服务器无响应	通过

表 6-1 模拟请求模块测试用例

## 6.2 HTML 分析模块测试

类型：单元测试、白盒测试

阶段：前期

具体时期：模拟请求模块完成基本功能实现。

测试过程：检查 http 请求状态码和 response 报文是否与正常的 web 浏览器请求相似或一致。

测试目的：判断模拟 web 浏览器浏览知网的功能是否符合要求。

测试用例：




编号	模块输入	模块操作	预期结果	实际结果	评价
6	 <p>输入实际上为知网搜索结果的 HTML，表格中为截图</p>	分析结果 数量	页大小 50 结果 6174 页数 120	页大小 50 结果 6174 页数 120	通过
7	 <p>输入实际上为知网搜索结果的 HTML，表格中为浏览器截图</p>	分析结果 数量	页大小 50 结果 0 页数 0	页大小 50 结果 0 页数 0	通过
8	空文本	分析结果 数量	返回错误 数量	返回错误	通过
9	 <p>输入实际上为知网搜索结果的 HTML，表格中为截图</p>	分析结果 中的作者	崔大林; 庄红山; 王晓飞; 于冰; 姜健琳 共 5 个作者	错误地将 “崔大林; 庄红山; 王 晓飞; 于 冰; 姜健琳” 识别为 1 个作者	未通过

表 6-2 HTML 分析模块测试测试用例

### 6.3 XLSX 导入导出模块测试

类型：单元测试、黑盒测试

阶段：中期

具体时期：数据库操作模块基本功能实现，数据表构建完成。

测试过程：使用数据库操作模块对数据库进行增删改查。

测试目的：判断数据库操作模块能否和数据库正常的交互。

编号	模块输入	模块操作	备注	实际结果	预期结果	评价
10	XLSX 表格 表格空	导入搜索参数		返回错误	返回错误	通过
11	XLSX 表格 表格不空	导入搜索参数		返回搜索条件	返回搜索条件	通过
12	爬取记录 数组 数组空	导出爬取记录		导出表格内 容为空,但是 包含表头	导出表格为 内容空, 但 是包含表头	通过
13	爬取记录 数组 数组不空	导出爬取记录		导出表格包 含所有结果	导出表格为 空, 但是包 含表头	通过

## 7 总结

### 7.1 项目成果

本爬虫软件基于 nodejs 技术开发，考虑到爬取的结果除了导出 Execl 文档以外，还可能被使用于其他用途，同时为以后的拓展做考虑，选用了 mysql 数据库存储爬取的信息数据。

对爬虫工具的基本功能进行分析和评估之后，结合自己掌握的技术，使用了 Electron 框架进行了软件 UI 的开发。尽量将 UI 美观，易用的方向修缮。

同时，Electron 框架将主进程和渲染进程分开，从形式上分成了前后端开发，降低了代码的耦合度。

### 7.2 不足

由于前期的需求分析做的不够细致导致，导致进行软件测试时的对功能测试的安排不能按原计划进行，只得根据开发进度做修正。

UI 的开发不够完全，在输入条件方法只有导入 XLSX 文件一种，没有在界面直接输入的方法。

## 8 参考文献

- [1] 朴灵.深入浅出 Node.js[M].人民邮电出版社:北京,2013:1-.
- [2] (美)Mike Cantelon.[等].Node.js 实战[M].人民邮电出版社:北京,2014:1-.
- [3] (美)Ethan Brown.Node 与 Express 开发[M].人民邮电出版社:北京,2015:1-.
- [4] 刘兵.Web 数据挖掘[M].清华大学出版社:北京,2013:1-.
- [5] 张俊林.这就是搜索引擎[M].电子工业出版社:2012:1-.
- [6] 崔庆才.Python 3 网络爬虫开发实战[M].人民邮电出版社:北京,2018:1-.