动漫作品、角色信息爬取及管理系统

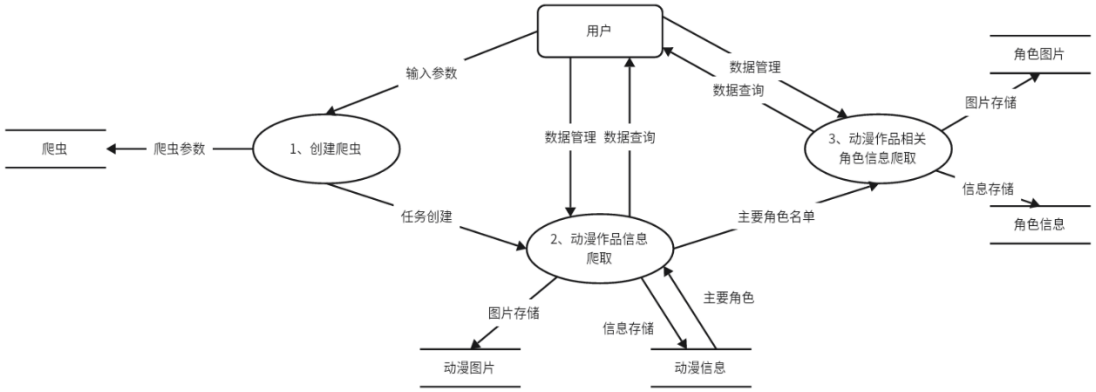
25 春 数据库及实现

主要功能及需求分析：

项目目标是用户能够创建爬虫项目，使爬虫根据用户给定参数从 bangumi（一个 ACG 作品收录网站）爬取符合要求的番剧，从该网站中获取该番剧信息及主要角色后再从萌娘百科（一个 ACG 角色百科网站）上爬取作品中每个主要角色的详细信息，并将所有爬取到的信息存入数据库。在数据库中可以对爬虫、作品和角色信息进行查询和管理。

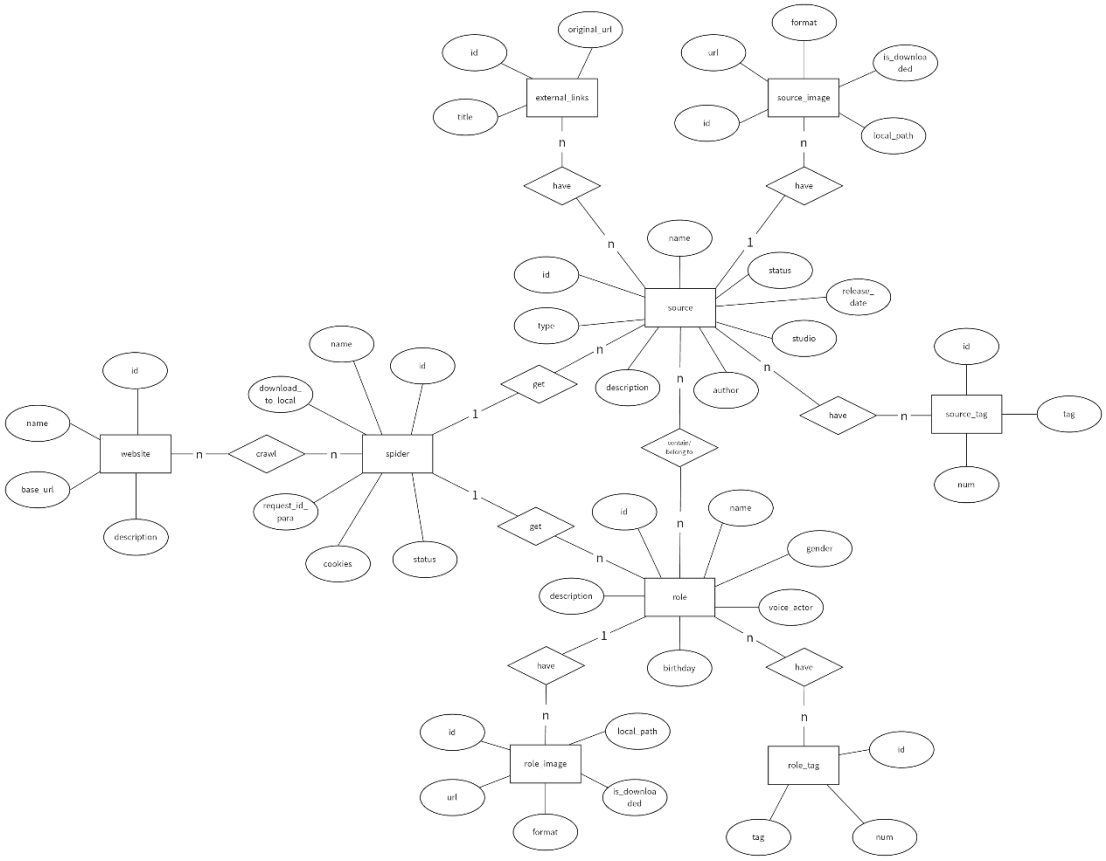
该项目需要实现爬虫根据指定参数爬取网络信息、给用户提供输入参数的菜单以及一个清晰的可视化数据库查询管理窗口。

图表 1：业务流程图



图表 2：数据流图

数据库概念模式设计：



图表 3：ER 图，若不清晰，可以参考 resources/ER.png

数据库设计：

1.website(

website_id: INT, PK, AUTO_INCREMENT，网站编号;

name: CHAR(20), NOT NULL, UNIQUE，网站名字;

base_url: VARCHAR(255), NOT NULL，网址;

description: TEXT，网站简介;

)

2.spider(

spider_id: INT, PK, AUTO_INCREMENT，爬虫编号;

website_id: INT, FK, NOT NULL，网站编号;

name: CHAR(20), NOT NULL, UNIQUE，名字;

download_to_local: BOOLEAN, DEFAULT FALSE，是否将爬取内容下载到本地;

request_id_para TEXT，目标爬取番剧名称列表;

cookies: VARCHAR(255);

status: ENUM('active', 'inactive', 'expired'), DEFAULT 'active'，爬虫的状态：激活、未激活、删除;

FOREIGN KEY (website_id) REFERENCES Website(website_id) ON DELETE CASCADE

)

3.Source(
source_id INT, PK, AUTO_INCREMENT, 作品编号;
source_type ENUM('animation', 'book', 'game'), DEFAULT 'animation', NOT NULL, 作品类型;
name CHAR(30) NOT NULL, 作品名;
description TEXT, 简介;
author VARCHAR(100), 作者;
studio VARCHAR(100), 制作厂商;
release_date DATE, 发布日期;
status ENUM('not_released', 'ongoing', 'ended'), 状态：未发布、连载、完结;
UNIQUE(type,name)
)

4.ExternalLinks (
link_id INT, PK, AUTO_INCREMENT, 外部链接编号;
title VARCHAR(64) NOT NULL, 名称;
original_url VARCHAR(255), NOT NULL, UNIQUE, 可以找到此内容的原始 URL;
)

5.LinksOnPage (
source_id INT NOT NULL,
link_id INT NOT NULL,
PK(source_id, link_id),
FOREIGN KEY (source_id) REFERENCES Source(source_id) ON DELETE CASCADE,
FOREIGN KEY (link_id) REFERENCES ExternalLink(link_id) ON DELETE CASCADE
)用于连接作品和外部链接的关系表

6.SourceImage (
image_id INT, PK, AUTO_INCREMENT, 作品图片编号;
url VARCHAR(255), NOT NULL, UNIQUE, 图片的 url;
format VARCHAR(20), 图片格式 (e.g., JPEG, PNG, GIF);
is_downloaded BOOLEAN DEFAULT FALSE, 是否将图片下载到本地;
local_path VARCHAR(255), 下载到本地的路径;
source_id INT, NOT NULL
FOREIGN KEY (source_id) REFERENCES Source(source_id) ON DELETE CASCADE
)

7.SourceTag(
tag_id INT, PRIMARY KEY, AUTO_INCREMENT, 作品标签编号;
tag CHAR(20) NOT NULL, UNIQUE, 标签名;
num INT NOT NULL, 拥有该标签的作品的数量;
)

8.SourceTagRelation(
    tag_id INT NOT NULL,
    source_id INT NOT NULL,
    PK(tag_id, source_id),
    FOREIGN KEY (tag_id) REFERENCES Tag(tag_id) ON DELETE CASCADE,
    FOREIGN KEY (source_id) REFERENCES Source(source_id) ON DELETE CASCADE
)用于连接作品和标签的关系表

9.Role(
    role_id INT, PRIMARY KEY, AUTO_INCREMENT，角色编号;
    name CHAR(30) NOT NULL，角色姓名;
    gender ENUM('male', 'female', 'unknown') NOT NULL，性别;
    description TEXT，简介;
    birthday DATE，生日;
    voice_actor VARCHAR(20)，声优;
)

10.RoleSourceRelation(
    role_id INT NOT NULL,
    source_id INT NOT NULL,
    PK(role_id, source_id),
    FOREIGN KEY (role_id) REFERENCES Role(role_id) ON DELETE CASCADE,
    FOREIGN KEY (source_id) REFERENCES Source(source_id) ON DELETE CASCADE
)用于连接作品和角色的关系表

11.RoleImage (
    image_id INT, PK, AUTO_INCREMENT，角色图片编号;
    image_url VARCHAR(255), NOT NULL, UNIQUE，角色图片 url;
    format VARCHAR(20)，图片格式 (e.g., JPEG, PNG, GIF);
    is_downloaded BOOLEAN DEFAULT FALSE，是否下载到本地;
    local_path VARCHAR(255)，下载到本地路径;
    role_id INT, NOT NULL
    FOREIGN KEY (role_id) REFERENCES Role(role_id) ON DELETE CASCADE
)

12.RoleTag(
    tag_id INT, PRIMARY KEY, AUTO_INCREMENT，角色标签编号;
    tag CHAR(20) NOT NULL, UNIQUE，标签名;
    num INT NOT NULL，拥有该标签的角色数量;
)

13.RoleTagRelation(
    tag_id INT NOT NULL,
    role_id INT NOT NULL,

PK(tag_id, role_id),
FOREIGN KEY (tag_id) REFERENCES Tag(tag_id) ON DELETE CASCADE,
FOREIGN KEY (role_id) REFERENCES Role(role_id) ON DELETE CASCADE
)用于连接角色和标签的关系表

以下为代码文档及操作说明

1. 代码文档

代码主要由以下部分组成：

```
主文件夹/
├─── app/
│    ├─── 页面组件
│    └─── 数据库接口
├─── resources/
│    ├─── default_image.jpg
│    ├─── anime.sql
│    ├─── 项目文档.pdf
│    └─── ER.png
├─── kirakiradokidoki/
│    └─── 爬虫组件
├─── init.py
├─── main_window.py
├─── config.ini
└─── requirements.txt
```

具体介绍如下：

(1) app

页面组件：包含四个主要页面 searchpage.py，detailpage.py，tagpage.py，settingpage.py 以及组成页面的小部件

数据库接口：databaseapi.py，给前端提供数据库接口并操作

(2) resources

包含 default_image.jpg，用作详情页默认图片；anime.sql 为数据库数据；ER.png 为 ER 图，不清晰时可以参考

(3) kirakiradokidoki

主程序调用 add_single_source.py，包含用于读取数据库爬虫数据并爬取清洗相应数据的代码与接口，输入一个字符串列表，字符串为 Bangumi 相应条目的 6 位 id

(4) init.py：用于建立并初始化数据库

(5) main_window.py：前端的主窗口，用于启动前端并操作

(6) config.ini：用于填写数据库相关信息

(7) requirements.txt：程序所需要的库

2. 操作说明

本系统基于 windows10+，python3.13+和 MySQL8.0+实现，在使用前请确保已安装

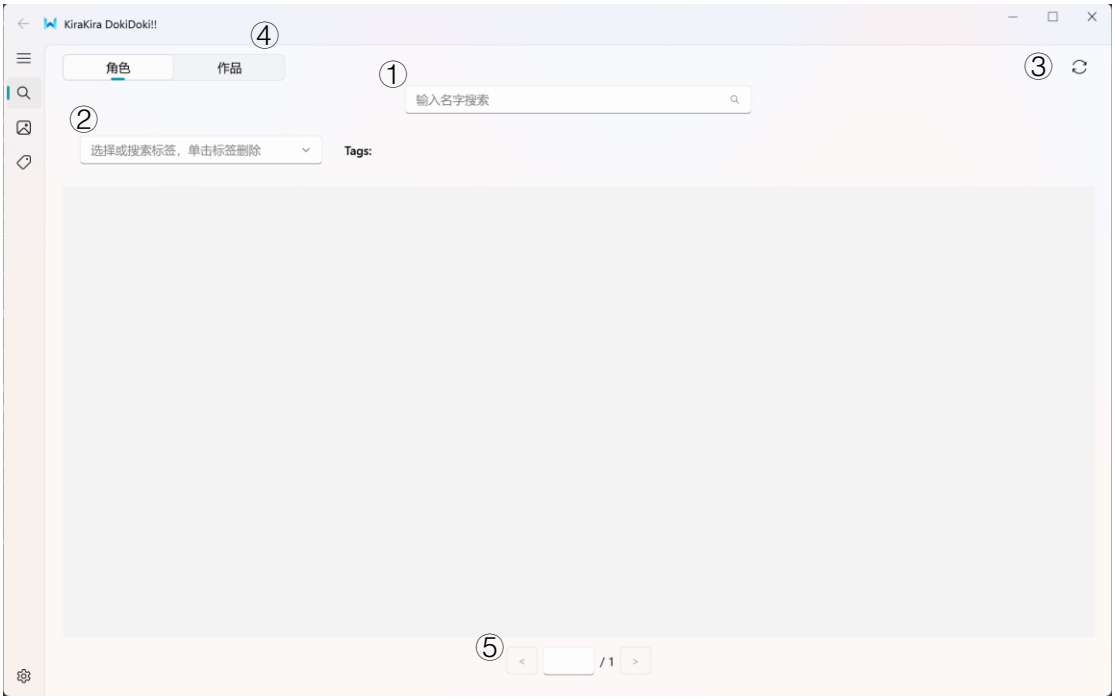
(1) 初始化

首次运行时在在命令行中切换到你需要的 python 虚拟环境，并输入以下指令：

```
pip install -r requirements.txt
```

然后更改 config.ini 文件，将其中的用户与密码改为你自己数据库的用户与密码并保存，然后在主文件夹下运行 init.py 文件创建数据库，该数据库名为 anime，请确保没有同名数据库存在/也可以通过创建名为 anime 的数据库并导入 resources/anime.sql 文件来初始化数据库

(2) 搜索页



功能介绍：

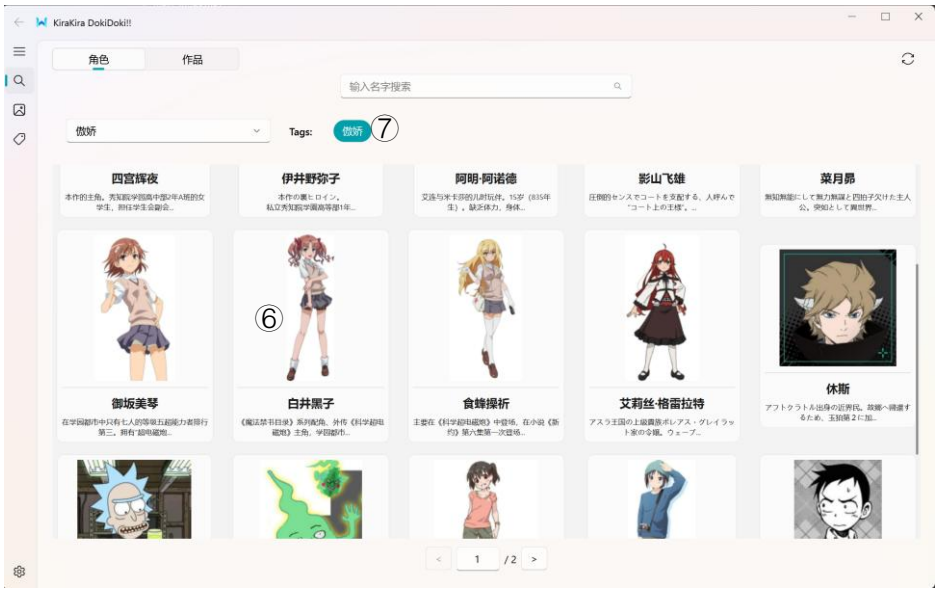
① 名字搜索框，支持名字的模糊匹配，输入名字后按 enter 键或点击搜索按钮即可搜索，在未输入任何文字与标签的情况下为搜索全部

② 标签搜索选择栏，可以搜索当前所有已经存在的标签，点击可以下拉查看所有标签

③ 刷新键，可以刷新页面

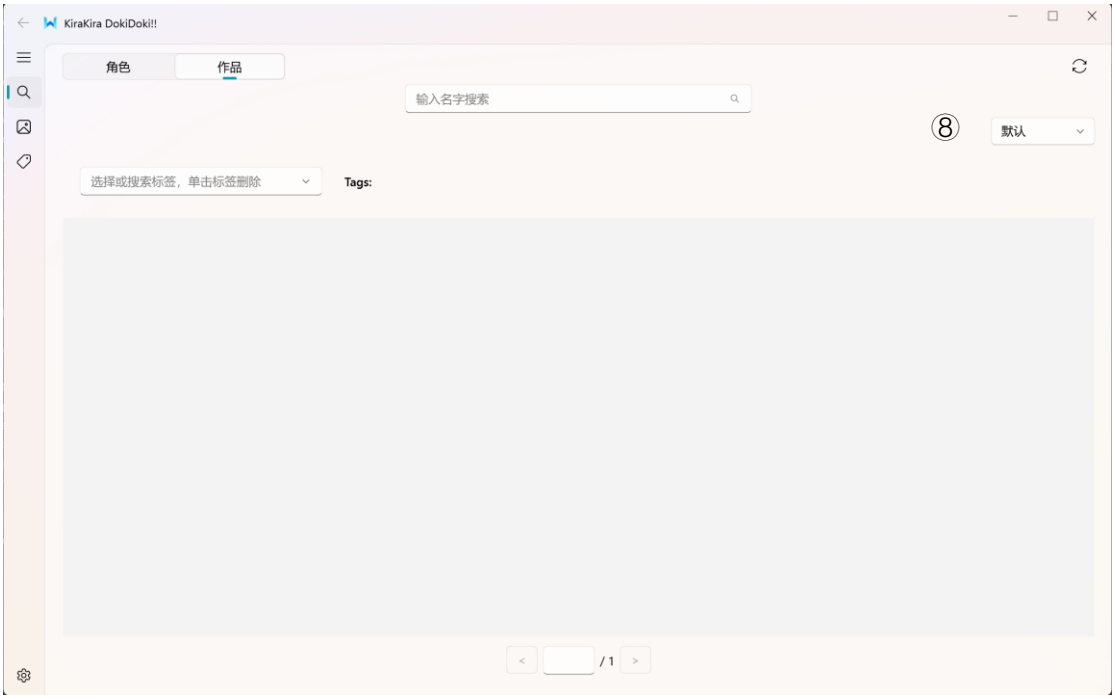
④ 切换作品与角色搜索栏，搜索内容会缓存

⑤ 切换搜索页数



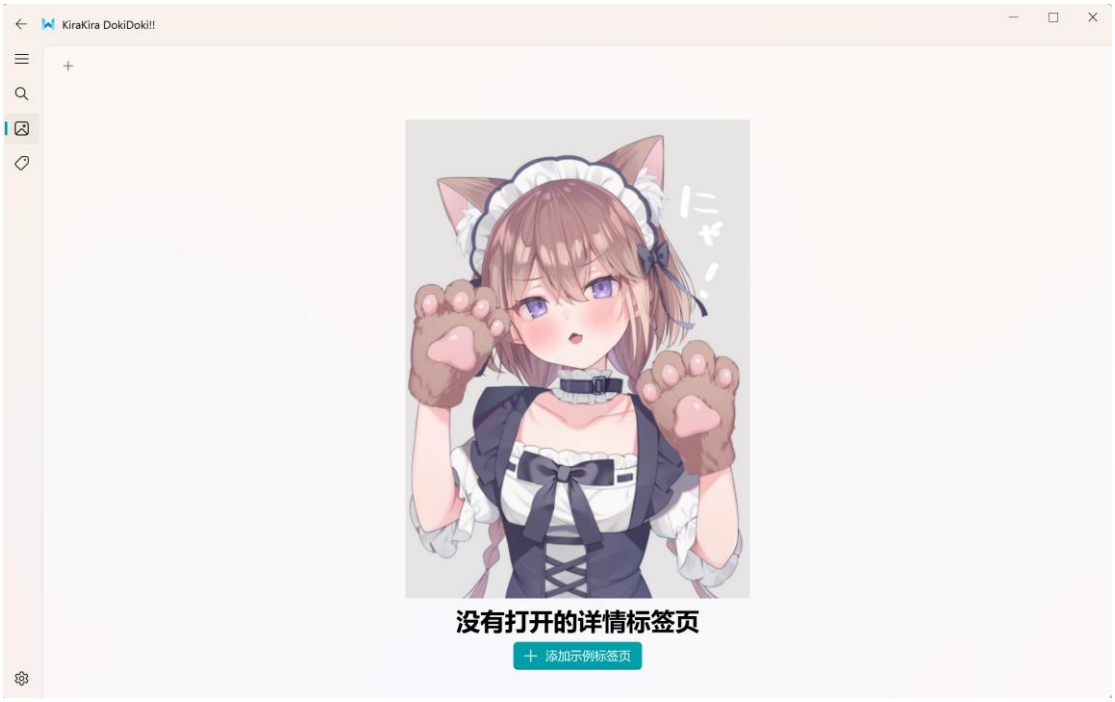
⑥ 角色/作品卡片，左键点击后可以跳转到详情页，右键可以复制名字，图片或者详情；可能出现加载时卡片错位的情况，刷新可以解决

⑦ 已选择的角色标签，单击后取消选择



⑧ 可以选择默认排序或者按照时间排序，仅作品有这一选项

(3) 详情页



默认页：点击标签页处的加号或者添加示例标签页均可跳转至搜索页

功能介绍：

① 角色图片，点击可以查看原图

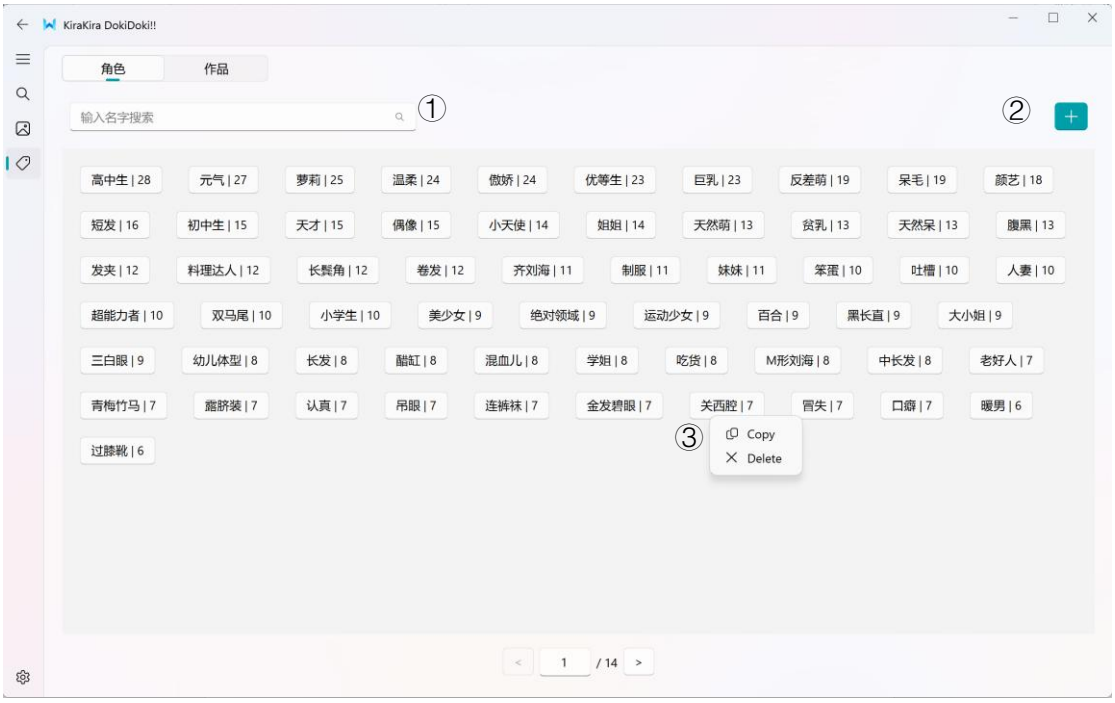
② 简介与标签修改，可以修改简介以及标签，生日应该为 MM-DD 的形式，发布日期应该为 YYYY-MM-DD 的形式，否则无法保存；标签可以通过加号进行添加，只能添加已有标签，单击标签删除；点击撤销键回退到未保存前的状态

③ 来源链接，内部链接到其作品的详情页

④ 删除键，将该角色/作品彻底删除

⑤ 外部链接，点击后可以打开浏览器，跳转到作品的官方网站

(4) 标签页



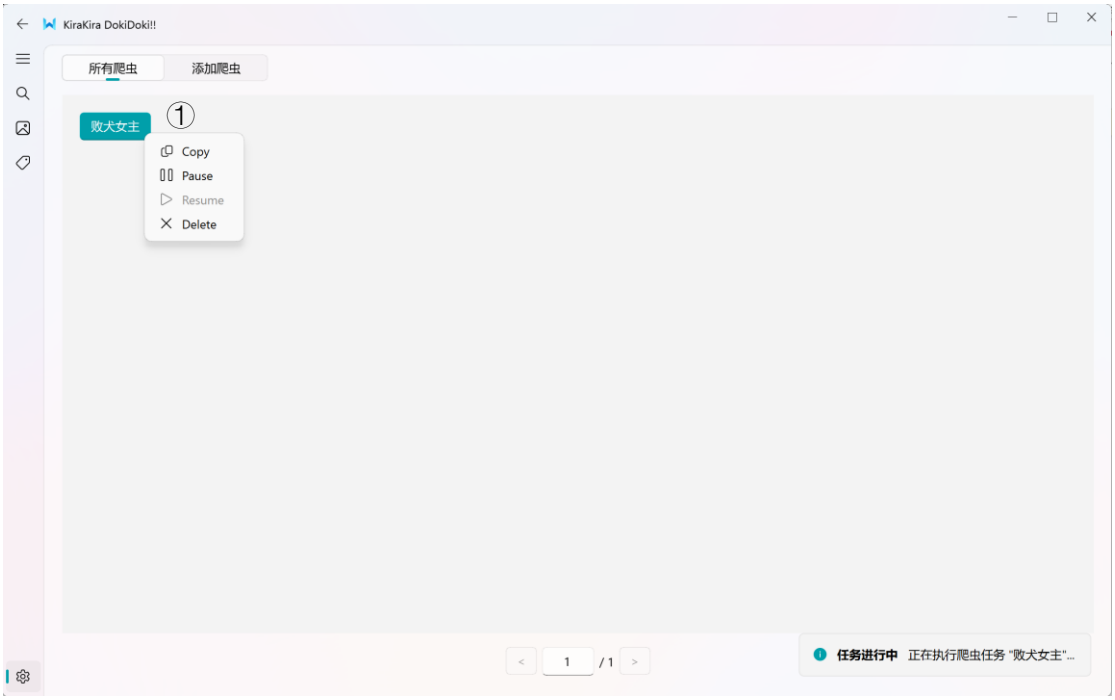
作品/角色切换栏，页数切换此处不再赘述

① 搜索栏，可以搜索已有的标签

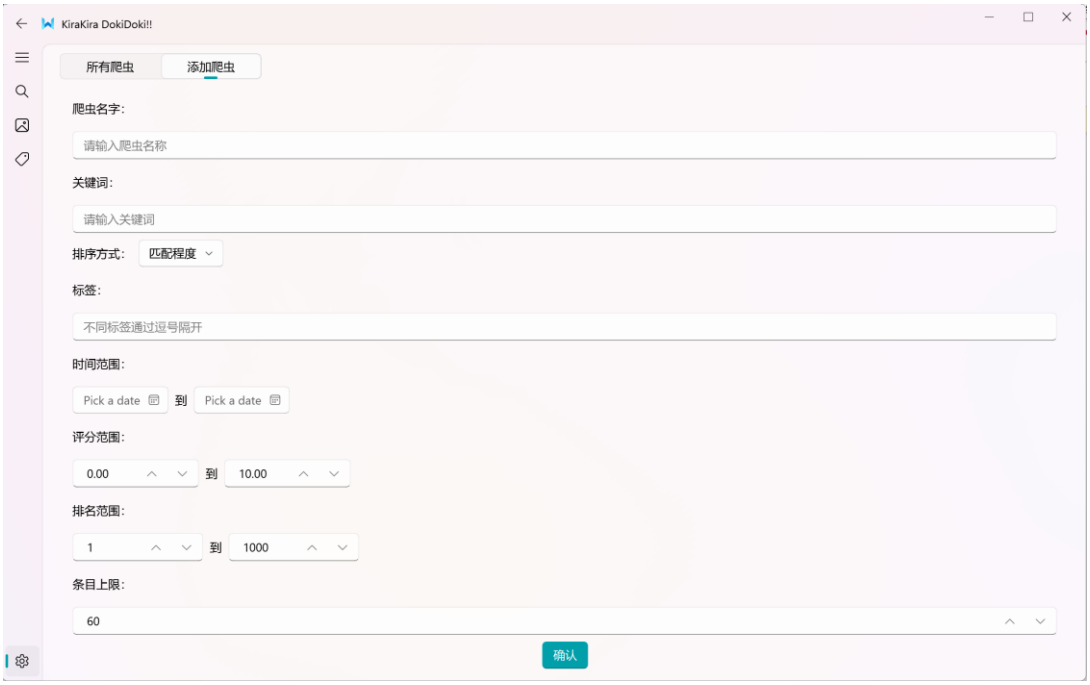
② 添加栏，可以添加标签，该标签必须不存在于原有标签，否则无法添加

③ 标签，左侧为标签名字，右侧为标签数量，左/右键点击后会跳出菜单栏，菜单中可以选择复制标签或者删除标签

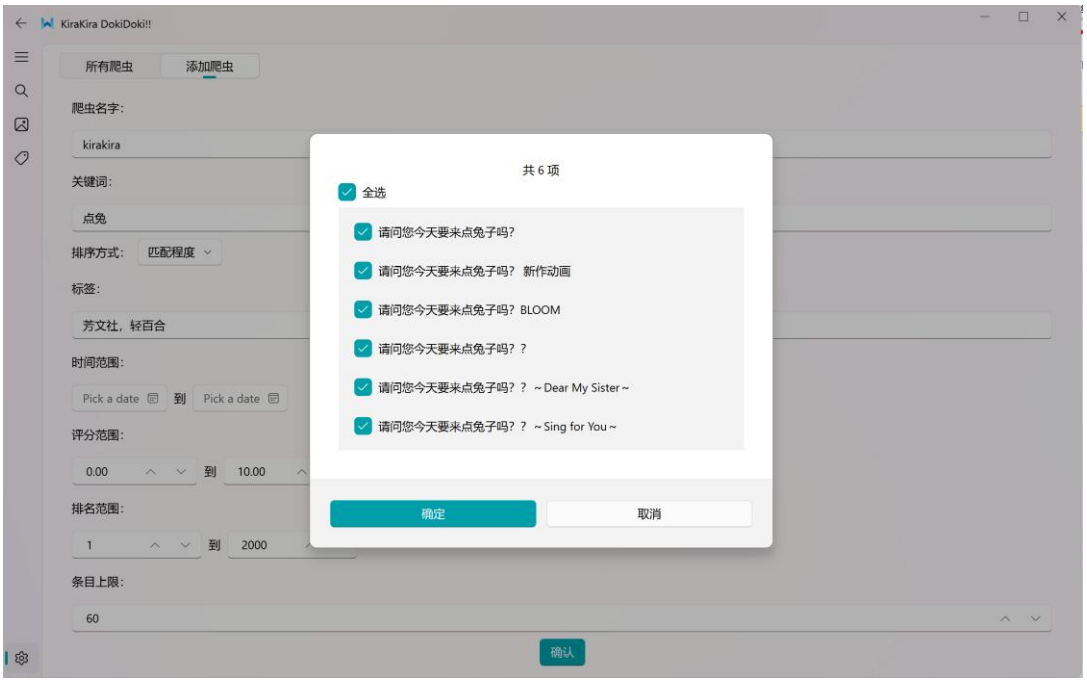
(5) 设置页（爬虫）



① 爬虫详情按钮，显示为爬虫名字，高亮（如图）表示正在运行，未高亮（白色）表示停止运行，灰色表示已抛弃，左/右键点击后会弹出菜单，可以选择复制爬虫名字，暂停爬虫，使爬虫重新开始运行以及删除爬虫；删除爬虫时为防止错误，不会立即删除，但是抛弃后的爬虫无法再次被点击或运作，会在每次打开程序时统一删除



添加爬虫界面，按照提示可以进行搜索需要的内容并爬取，注意以下事项：爬虫名字不能与已有的重复，必须提供；搜索内容是通过 Bangumi 进行的，设置了限速，所以搜索时可能会需要花费一定时间，并且搜索条目有上限（200 条），搜索后界面如下：



可以选择你想要爬取的进行爬取，爬取时信息会输出在 shell 终端

温馨提示：建议不要同时开启多个爬虫，由于爬虫是通过多线程进行的，主程序在线程增多时容易卡住

3. 代码模块

代码主要由三个模块构成，分别为：前端，数据库操作以及爬虫。

前端由 pyside6 和 qfluentwidgets 构成，给用户提供一个简洁易操作的页面。

数据库操作主要由 mysql 和 sqlalchemy 构成，用于将前端操作转换为数据库操作，并在数据库中执行。

爬虫主要由 beautifulsoup4 进行，通过获取 html 具体信息并进行清洗获得结构清晰的数据。