疫情问答检索系统v1.0

系统设计文档

日期：2020年6月24日

目录

[1 运行环境 3](#_Toc43910629)

[1.1. 硬件环境 3](#_Toc43910630)

[1.2. 软件环境 3](#_Toc43910631)

[2 开发工具 3](#_Toc43910632)

[2.1 Pycharm 3](#_Toc43910633)

[2.2 MYSQL 4](#_Toc43910634)

[3 系统总体结构设计 4](#_Toc43910635)

[3.1 设计原则 4](#_Toc43910636)

[3.2 流程设计 5](#_Toc43910637)

[3.3 总体结构 7](#_Toc43910638)

[4 数据库设计 1](#_Toc43910639)

[4.1 数据库概念结构设计 1](#_Toc43910640)

[4.2 数据库逻辑结构设计 4](#_Toc43910641)

[5 系统功能设计 5](#_Toc43910642)

[5.1 问题展示模块设计 6](#_Toc43910643)

[5.2 回答展示模块设计 7](#_Toc43910644)

[5.3 相关回答展示模块设计 7](#_Toc43910645)

[6 系统界面设计 7](#_Toc43910646)

[6.1问题检索界面 7](#_Toc43910647)

[6.2回答检索界面 9](#_Toc43910648)

[6.3查看评论界面 9](#_Toc43910649)

## 运行环境

### 硬件环境

1. 处理器: 1.8GHz Intel 酷睿2或者同级别AMD处理器以上。
2. 内存容量:Windows 7 :2G Windows 10 4G
3. 硬盘容量: 15G以上

### 软件环境

在Windows 7/8/10操作系统之上，采用作为开发平台，以MYSQL作为后台数据服务支撑，采用Django后台开发框架，layui前端开发框架进行开发。

## 开发工具

### Pycharm

PyCharm是一种Python IDE，带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制。此外，该IDE提供了一些高级功能，以用于支持Django框架下的专业Web开发。PyCharm是使用最广泛的一种集成开发环境适用于Python编程语言。目前，Twitter，Pinterest，HP，Symantec和Groupon等大型企业正在使用Python IDE。可以用来构建简洁、简洁、易于使用的软件应用程序，可读的代码库。Pycharm中每个文件都有其输出窗口，可以终止进程（只要点下按钮就行），并且设置了各种提示便于编程，其超强的索引功能也是我们选择此工具开发的原因。

本项目需要采集知乎网站上新冠肺炎疫情相关问题及回答，构造新冠肺炎问答检索系统，在系统的前端设计上，我们选用了Django框架，Django是高水准的Python编程语言驱动的一个开源模型，视图、控制器风格的Web应用程序框架，它起源于开源社区。使用这种架构，程序员可以方便、快捷地创建高品质、易维护、数据库驱动的应用程序。综合以上因素，本项目采用pycharm进行开发。

### MYSQL

MySQL 是一个关系型数据库管理系统，其使用的 SQL 语言是用于访问数据库的最常用的标准化语言。由于 MySQL 数据库体积小、速度快、总体拥有成本低、开放源代码，其有着广泛的应用。MySQL性能卓越服务稳定，很少出现异常宕机，其开放源代码且无版权制约，自主性强、使用成本低，并且历史悠久、社区及用户非常活跃，遇到问题，可以很快获取到帮助。同时，MySQL软件体积小，安装使用简单，并且易于维护，安装及维护成本低，支持多种操作系统，提供多种api几口，支持多种开发语言。所以我们选择将爬虫爬取的知乎网站上新冠肺炎疫情相关问题及回答相关内容存入MySQL数据库中，便于后续抽取数据进行可视化展示。

## 系统总体结构设计

### 设计原则

系统设计的好坏在根本上决定了软件系统的优劣。为确保系统的建设成功与可持续发展，在系统的建设和技术方案设计时我们需要遵循如下设计原则：

（1）合理性。在系统设计时，充分考虑系统的容量及功能的扩充，方便系统扩容及平滑升级。系统对运行环境(硬件设备、软件操作系统等)具有较好的适应性，不依赖于某一特定型号计算机设备和固定版本的操作系统软件。

（2）实用性**。**系统应提供统一的界面风格，构造清晰、简洁、友好的中文人机交互界面，操作简便、灵活、易学易用，便于管理和维护。

（3）安全性。系统能保证数据安全一致，高度可靠，应提供多种检查和处理手段，保证系统的准确性。针对主机、数据库、网络、应用等各层次制定相应的安全策略和可靠性策略保障系统的安全性和可靠性。

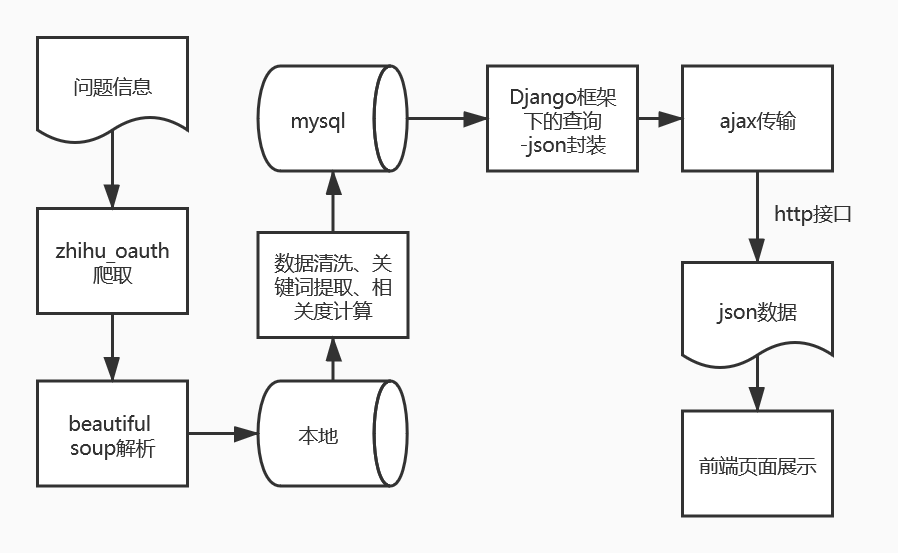
（4）可扩展性。信息系统设计要考虑到业务未来发展的需要，尽可能设计得简明，降低各功能模块耦合度，并充分考虑兼容性。系统具备良好的输入输出接口，可为各种增值业务提供接口。

（5）可维护性。系统操作简单，实用性高，具有易操作、易维护的特点，系统应具有专业的管理维护终端，方便系统维护。并且，系统具备自检、故障诊断及故障弱化功能，在出现故障时，能得到及时、快速地进行自维护。

（6）开放性。系统设计遵循开放性原则，能够支持多种硬件设备和网络系统，软硬件支持二次开发。各系统采用标准数据接口，具有与其他信息系统进行数据交换和数据共享的能力。

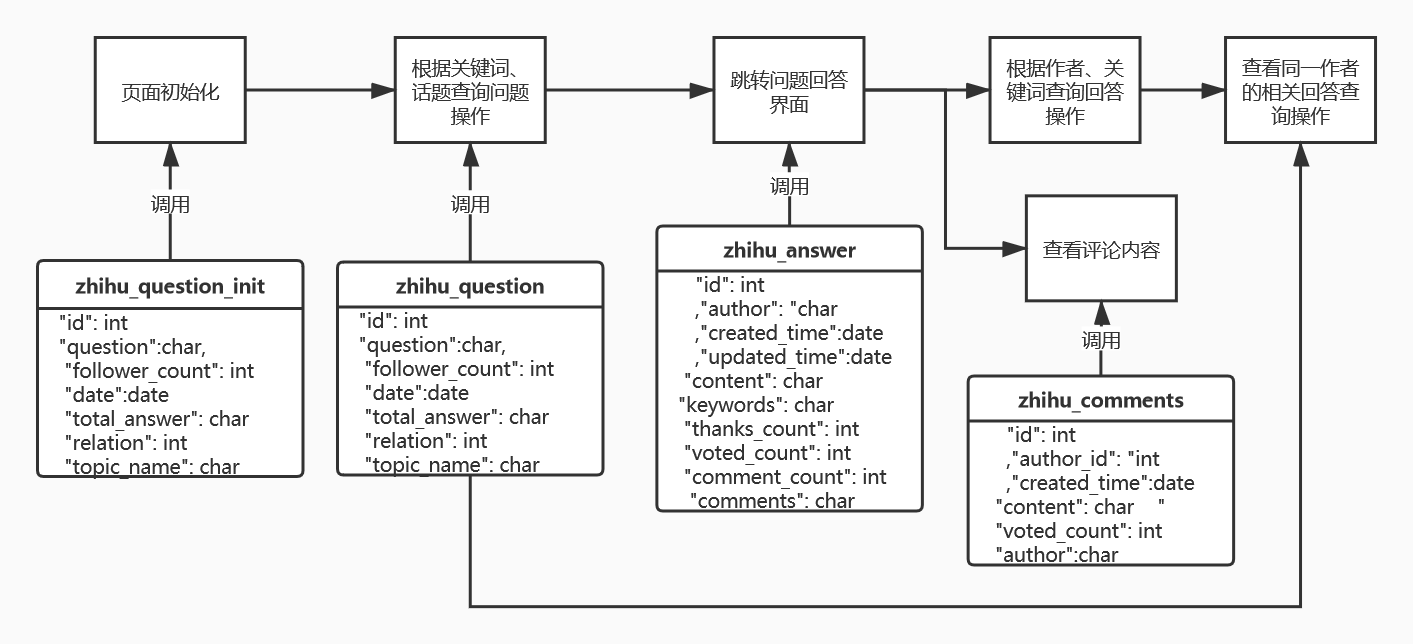
### 流程设计

1. 此系统的数据流程图设计如下：



使用zhihu\_oauth库进行知乎问答数据的爬取，将爬取的数据用数据清洗算法进行清洗后，提取关键词和文本相似度，将结果存入数据库中。利用Django框架进行后台数据管理和数据传输，前端用Layui框架进行界面展示。

1. 数据库调用关系如下图所示：



在页面初始化时，调用zhihu\_question数据表进行数据展示，在关键词查询功能中，将查询数据的关键词用Ajax传给后台，后台通过调用zhihu\_question表进行关键字查询。在回答展示页面用调用zhihu\_answer表进行回答查询。在评论展示界面调用zhihu\_comments表进行评论查询。

### 总体结构

#### 架构设计

我们的系统选择使用Django开发框架，Django是一款python的web开发框架，与MVC有所不同，属于MVT框架。其中，m表示model，负责与数据库交互；v 表示view，是核心，负责接收请求、获取数据、返回结果；t 表示template，负责呈现内容到浏览器。MVT框架结构如图3-1所示：

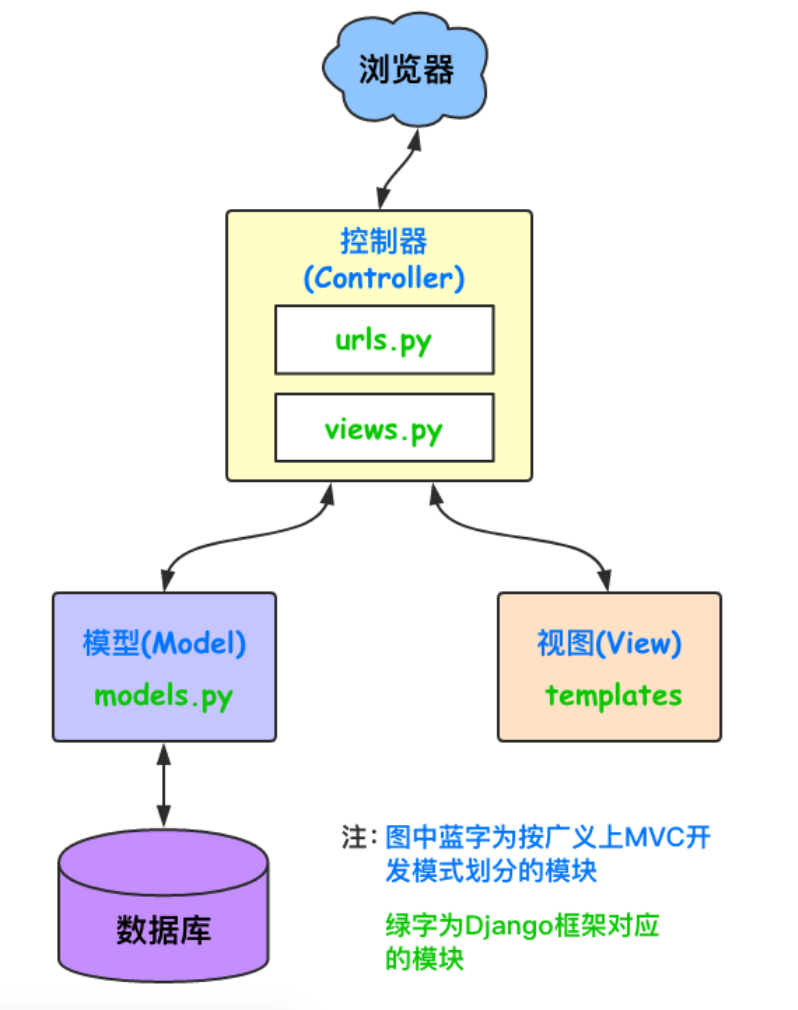


图 3-1 MVT框架结构图

在MVT框架中，客户端请求信息交给视图函数处理，当视图需要数据的时候调用模型Model，然后去数据库拿数据，接着返回给模型Model，再最后返回给视图。模型Model中不编写任何SQL语句。在关系数据库中的一行，相当于模型Model中的一个对象。在关系数据库中的一个表，相当于一个对象的集合。在python中用列表定义关系数据库中的一个表，我们直接在对象中写列表的操作就可以了，ORM会转换成SQL语句。之后我们得到一个列表对象返回给视图View, 视图View会进行一些处理，处理的结果需要用模板Templates呈现（渲染HTML和CSS执行JS），然后返回客户端。

#### 模块设计

##### （1）数据获取模块

我们的任务是自动采集知乎网站上新冠肺炎疫情相关问题及回答。在数据获取上，我们使用爬虫技术对知乎文旦进行爬取，首次登录知乎并保存token，后续就可以直接加载token来登录知乎。main-json.py模块实现了知乎数据爬取功能，通过设置知乎话题ID爬取想要话题下的问题、回答和评论数据。爬取的数据自动以模块中设计好的json格式放在对应话题名称的文件夹下。每一个json文件命名方式为“问题时间+问题名称”。

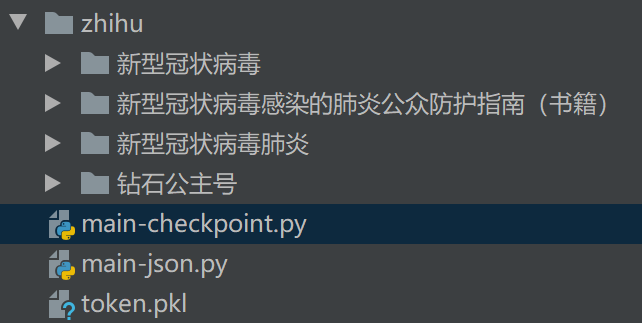


图 4-2 爬虫模块结构图

##### （2）数据清洗模块

1. 清洗步骤如下：

* 读取爬虫得到的json数据
* 清洗数据中的文本，去除网址链接，空白字符串，特殊字符等
* 设计并使用相关度算法，计算问题与主题“疫情”的相关度
* 连接数据库，将数据保存到数据库表中

1. 影响相关性的主要因素包括如下几方面
2. **关键词常用程度**

经过分词后的多个关键词，对整个搜索字符串的意义贡献并不相同。越常用的词对搜索词的意义贡献越小，越不常用的词对搜索词的意义贡献越大。

1. **词频及密度**

一般情况下，认为在没有关键词堆积时，搜索词在页面中出现的次数多，密度越高，说明页面与搜索词越相关。不过这只是一个大致规律，实际情况未必如此，所以相关性计算还有其他因素。出现频率及密度只是因素的一部分，而且重要程度越来越低。

1. **关键词位置及形式**

就像在索引部分中提到的，页面关键词出现的格式和位置都被记录在索引库中。关键词出现在比较重要的位置，如标题标签、黑体、H1 等，说明页面与关键词越相关。这一部分就是页面 SEO所要解决的。

1. **关键词距离**

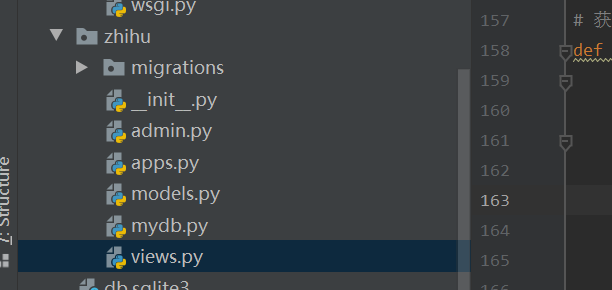
切分后的关键词完整匹配地出现，说明与搜索词最相关。比如搜索“优化方法”时，页面上连续完整出现“优化方法”四个字是最相关的。如果“优化”和“方法”两个词没有连续匹配出现，出现的距离近一些，也被搜索引擎认为相关性稍微大一些。

1. **链接分析及页面权重**

页面之间的链接和权重关系也影响关键词的相关性，其中最重要的是锚文字。页面有越多以搜索词为锚文字的导入链接，说明页面的相关性越强。链接分析还包括了链接源页面本身的主题、锚文字周围的文字等。

##### （3）数据管理展示模块

后台管理框架采用Django框架，该框架有强大的数据库功能、后台功能和集成了网址功能。Django框架主要包含以下几个模块：



1. Views: Django中View是一个可调用对象，接受一个request，并且返回一个response。
2. Models:在这个类中存在与数据库的关系对象映射，将其中的类映射到数据库的数据表中，根据代码中定义的类来自动生成数据表。
3. Mydb：在其中定义数据库的增删改查操作，操作对象是Models中定义的类。我们不需要关注SQL语句和数据库的类型，Django自动生成相应数据库类型的SQL语句，来完成对数据库数据的操作。
4. urls:在这个文件中定义数据接口，数据接口与Views文件中的函数相对应。

在前端使用Layui框架，该框架中封装了排序功能，可以实现对数据的排序与重载。Layui的组成目录如下：



#### 关键技术

##### 爬虫技术

爬虫任务获取基本数据对整个系统提供数据支持，我们使用基础的requests和BeautifulSoup对网页内容进行爬取，我们将网页看成是一个我们获取信息的接口，可以通过python爬虫从中获取我们所需要的信息。首先通过requests库爬取html页面的内容，然后使用BeautifulSoup库对爬取到的html页面进行解析，使用BeautifulSoup以及正则表达式来进一步提取我们想要的关键信息，最后将信息格式化并输出。

我们的任务是自动采集知乎网站上新冠肺炎疫情相关问题及回答。经过调研，我们发现了Zhihu-OAuth库，Zhihu-OAuth库使用方法简单，支持 Python 2 和 Python 3，其中能爬取的数据非常全面丰富，包括：用户动态、答案、文章、收藏夹、专栏、评论、首页信息流、知乎live、特定用户的信息、分享、问题、话题、私信会话和私信信息等等全部囊括在内。可以直接用pip命令装载。

对于我们的任务——爬取知乎特定话题下的相关回答，首次登录知乎，并保存token的代码，后续就可以直接加载token来登录知乎。本次爬取的知乎话题为“新型冠状病毒肺炎”，话题ID为21239580。最终爬取的数据放在“zhihu”下的“新型冠状病毒肺炎”文件夹内，以JSON形式存储，便于后续的数据清洗。每个问题数据包含问题名称、问题详情、时间以及若干条答案数据，每个答案包括回答内容、回答者、点赞数、回答时间、更新时间以及若干评论。每条评论包括评论内容、评论者、评论时间、点赞数等数据。

##### （2）关键字提取技术

TextRank算法可以用来从文本中提取关键词和摘要（重要的句子）。TextRank4ZH是针对中文文本的TextRank算法的python算法实现。

TextRank4ZH算法将原文本拆分为句子，在每个句子中过滤掉停用词（可选），并只保留指定词性的单词（可选）。由此可以得到句子的集合和单词的集合。

每个单词作为pagerank中的一个节点。设定窗口大小为k，假设一个句子依次由下面的单词组成：

w1, w2, w3, w4, w5, ..., wn

w1, w2, ..., wk、w2, w3, ...,wk+1、w3, w4, ...,wk+2等都是一个窗口。在一个窗口中的任两个单词对应的节点之间存在一个无向无权的边。

基于上面构成图，可以计算出每个单词节点的重要性。最重要的若干单词可以作为关键词。

##### （3）相似度分析技术

在调查研究的过程中发现，某些用户的回答内容与问题本身相关度很低，存在蹭热度的现象。为了计算回答内容与问题详情的相关程度，我们采用文本相似度计算方法来对回答内容与问题详情进行相关度计算。

我们分析了目前最流行的五种相似度计算方式：

1. SimHash + 海明距离：该算法对单词数量低于500的文章误差较大
2. 编辑距离 Levenshtein：计算复杂度偏高， 但是在实际场景中表现最为优秀。相似度计算最为准确。
3. 余弦相似度：计算复杂度偏高。然而，在许多实际应用中,数据稀疏度过高，通过余弦相似度计算会产生误导性结果。
4. 杰尔德相似度 Jaccard： 在问题描述中，很多回答人员为了偷懒，喜欢复制粘贴稍作修改，造成问题回答重复度高。通过提取回答的关键词，再计算两组关键词的交集并集。
5. MinHash：SimHash与Minhash都具有一般哈希方法所没有的局部敏感的性质（局部敏感哈希LSH是一类哈希函数），MinHash和SimHash会使两个相差不大的文档的哈希结果很相似。

经过分析比较以后，我们决定采取在实际文本中表现效果最优的编辑距离Levenshtein算法来进行相似度计算。

## 数据库设计

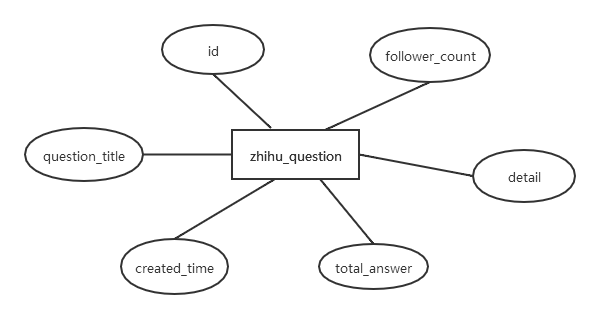
爬虫之后得到的各种问题、答案和评论数据，需要将其存储到数据库中，方便后续系统进行数据的查询和前端数据的展示。数据库设计是整个信息系统开发与设计的重要环节，系统能否高效率地运行取决于数据库设计的优劣，所以在制作某个系统之前，首先要根据系统的具体要求以及实现的功能去设计数据库，来满足各种用户的应用需求。

### 数据库概念结构设计

1. 名称：zhihu\_quesstion

描述：爬虫爬取下来的所有问题相关数据。

定义：包含问题id、标题、问题详情、创建时间、答案数量、粉丝数量。

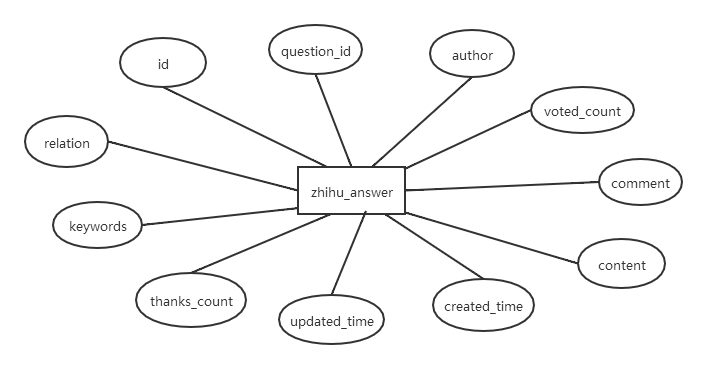
****

**图4.1 问题实体属性图**

1. 名称：zhihu\_comment

描述：每个答案下的回答相关数据。

定义：包含answer id、对应问题id、作者、点赞数、评论数量、回答内容、创建时间、更新时间、感谢数量、答案关键词、答案与问题详情相关度。

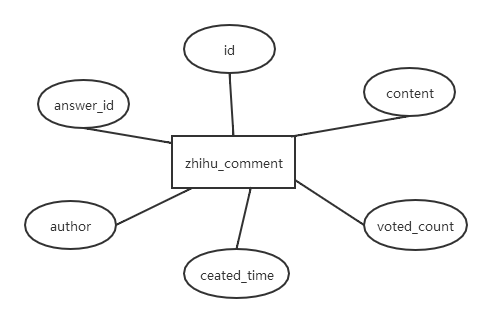
****

**图4.2 回答实体属性图**

1. 名称：zhihu\_comment

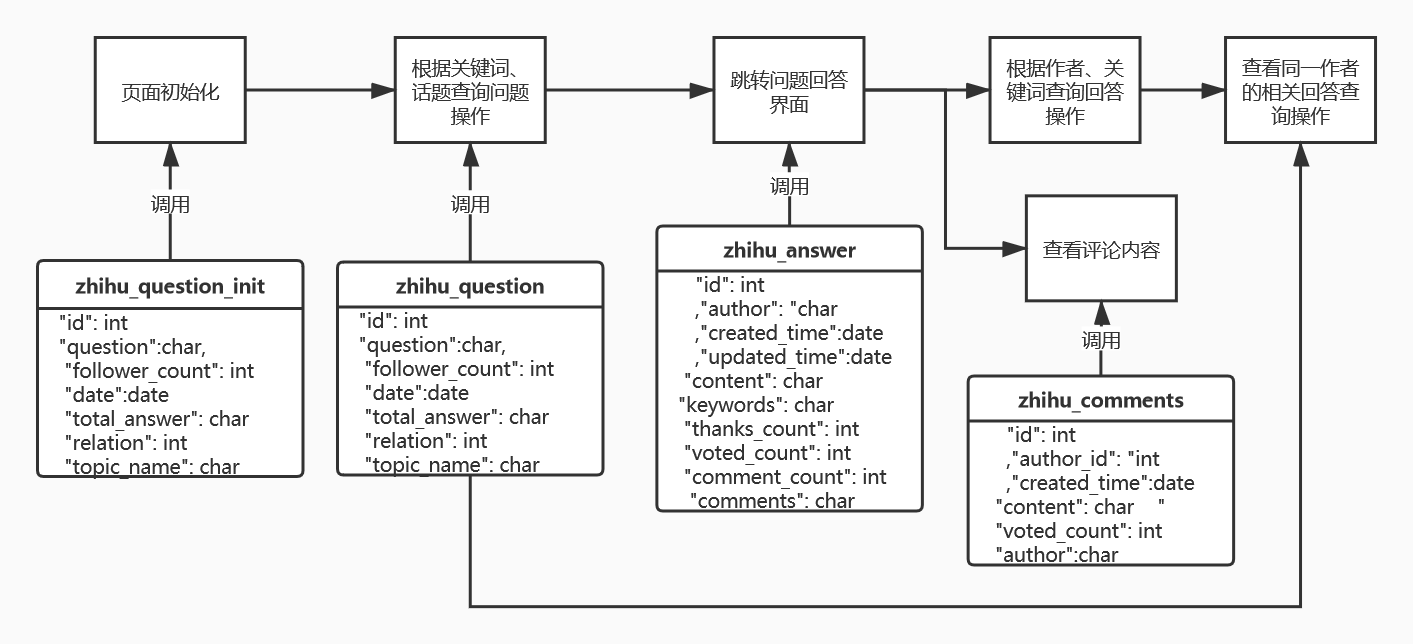
描述：每个回答下的评论数据.

定义：包括评论id、对应回答id、评论内容、作者、点赞数量、创建时间。



**图4.3 评论实体属性图**

以上数据库的概念结构设计是后续逻辑结构设计的基础，通过对数据库概念结构的分析与设计，对系统对数据库的相关操作流程示意图如下所示。



**图4.4 数据库流程示意图**

### 数据库逻辑结构设计

数据库的概念设计是与任意一种数据模型相独立的信息结构。数据库的逻辑结构设计就是把上面概念结构设计得到的各个实体和实体—实体间的联系转换为与系统选用的数据库产品所支持的数据模型相符合的逻辑结构。

**表4.1 问题信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **可否为空** | **说明** |
| **id** | **int** | **not null** | **问题id（主键）** |
| **question\_title** | **text** | **not null** | **标题** |
| **follower\_count** | **char** | **null** | **粉丝数** |
| **detail** | **text** | **null** | **详情** |
| **created\_time** | **int** | **null** | **创建时间** |
| **total\_answer** | **date** | **null** | **答案数量** |

**表4.2 回答信息表**

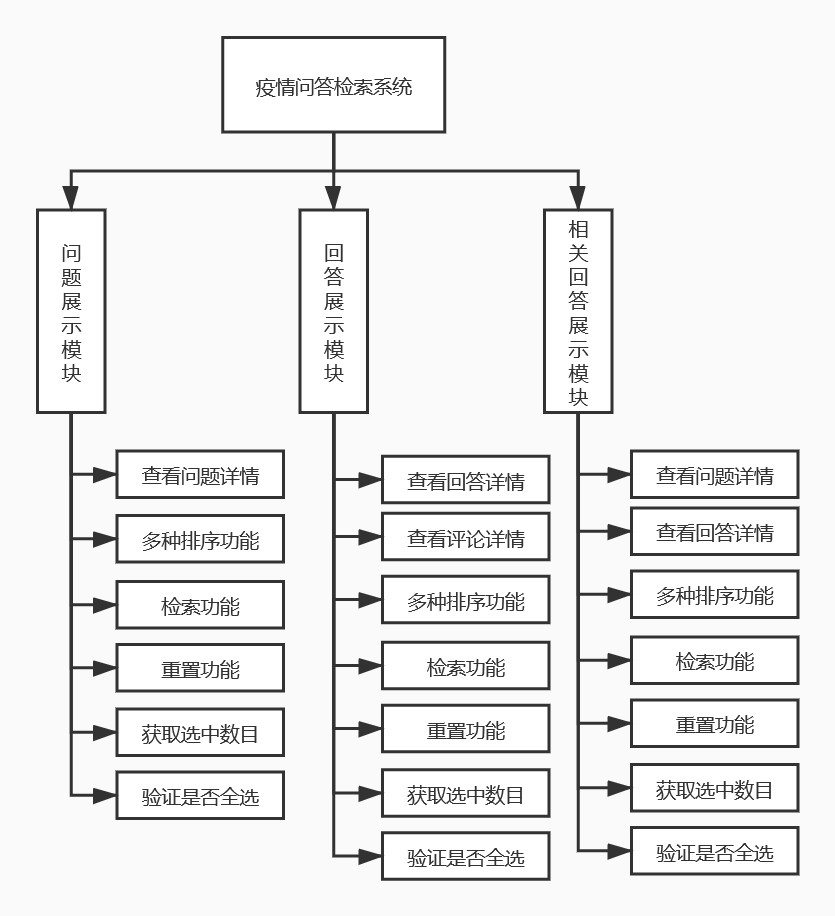
|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **可否为空** | **说明** |
| **id** | **int** | **not null** | **回答id（主键）** |
| **question\_id** | **int** | **not null** | **问题id（zhihu\_question表的外键）** |
| **anthor** | **char** | **null** | **作者** |
| **voted\_count** | **int** | **null** | **点赞数** |
| **comment\_count** | **int** | **null** | **评论数** |
| **content** | **text** | **null** | **回答内容** |
| **created\_time** | **date** | **null** | **创建时间** |
| **update\_time** | **date** | **null** | **更新时间** |
| **thanks\_count** | **int** | **null** | **感谢数** |
| **keywords** | **varchar** | **null** | **回答关键词** |
| **relation** |  | **null** | **回答内容与问题详情的相似度** |

**表4.3 评论信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名称** | **数据类型** | **可否为空** | **说明** |
| **id** | **int** | **not null** | **评论id（主键）** |
| **answer\_id** | **int** | **not null** | **回答id（zhihu\_answer表的外键）** |
| **content** | **text** | **null** | **评论内容** |
| **author** | **varchar** | **null** | **作者** |
| **created\_time** | **date** | **null** | **创建时间** |
| **voted\_count** | **int** | **null** | **点赞数** |

## 系统功能设计

疫情问答系统可以划分为三个模块：问题展示模块、回答展示模块、相关回答展示模块。如下图4.5所示：



**图4.5 系统功能结构图**

### 问题展示模块设计

问题展示模块包含查看问题详情、多种排序功能、检索功能、重置功能、获取选中数目、验证是否全选等功能。查看问题详情：系统可以从数据库中取出问题详情数据并展示在界面上。多种排序功能：系统可按照多种排序方式显示表格数据，例如按粉丝数排序、按回答数量排序、按时间排序等。检索功能：用户在输入框里输入关键词，系统可以在界面上展示相关搜索内容。重置功能：用户在搜索时需要重新键入关键词时，可点击【重置】按钮进行重置。获取选中数目：用户选中多个多选框时，统计选中数目。验证是否全选：验证用户是否全选了所有问题，并将结果返还给用户。

### 回答展示模块设计

回答展示模块设计包括查看回答详情、查看评论详情、多种排序功能、检索功能、重置功能、获取选中数目、验证是否全选等。查看回答详情：系统查询数据库并将回答内容显示在界面上。查看评论详情：用户可根据问题查看该问题的相关评论。检索功能、重置功能、获取选中数目、验证是否全选的功能与问题展示模块一致。

### 相关回答展示模块设计

相关回答展示模块设计包括查看问题详情、查看回答详情、查看评论详情、多种排序功能、检索功能、重置功能、获取选中数目、验证是否全选等。

## 系统界面设计

### 6.1问题检索界面

1. **问题显示界面**



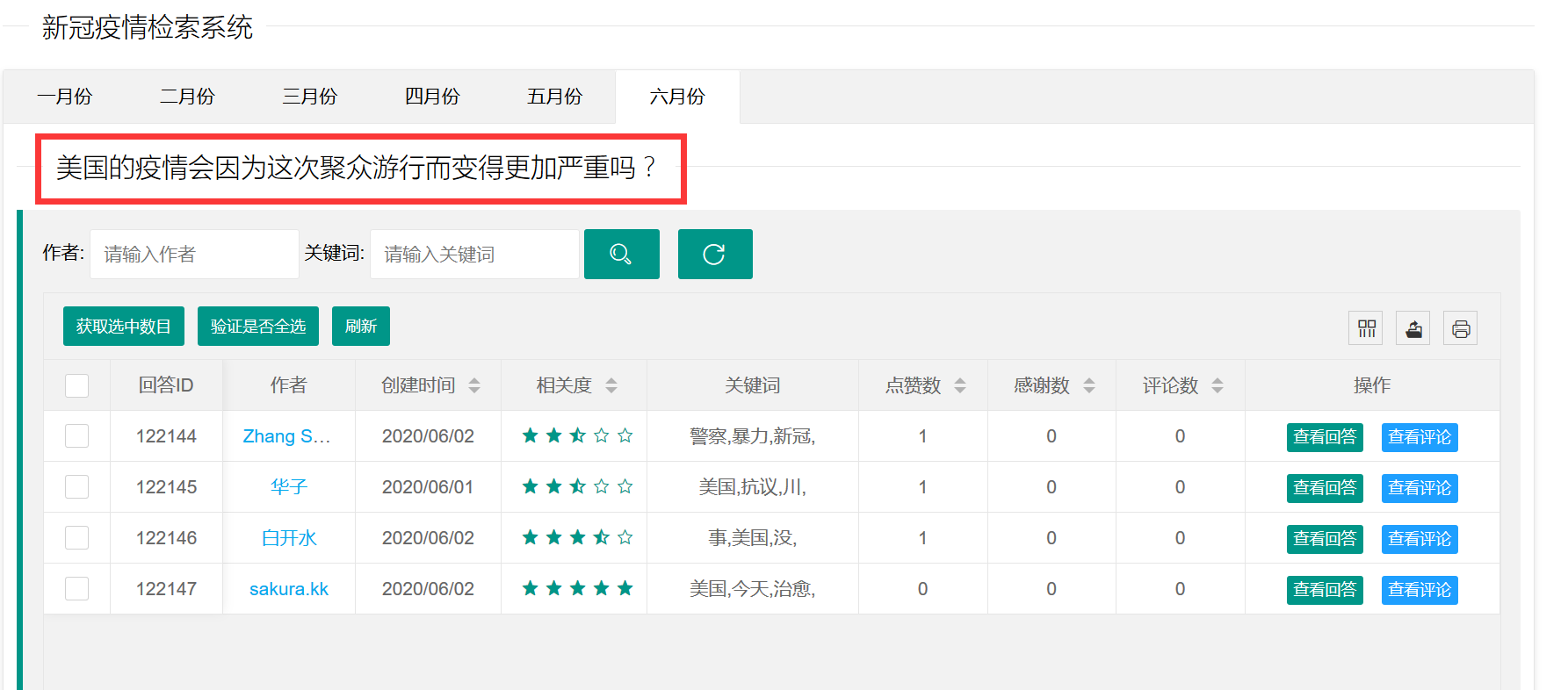
点击首页中的【一月份】展示一月份的问题，点击【二月份】显示二月份的问题。同理类推。

1. **详情展示界面**



点击表格右侧的【问题详情】按钮，显示该问题的详情。

### 6.2回答检索界面



点击问题的标题可以跳转到回答详情页面，且在界面上方显示该问题的标题。

### 6.3查看评论界面



点击每个记录后面的【查看评论】按钮，显示该问题的评论信息。