**软件工程管理项目设计文档**

**1 引言**

**1.1 编写目的**

该软件概要设计说明书以软件需求规格说明书和系统设计与实现计划为基础，说明系统的总体架构，各个功能的实现方式及数据库设计，明确各个模块的外部接口、内部接口和用户接口，为软件系统的开发提供指导，为软件系统的维护提供参照。

**预期读者**：

- 项目经理

- 系统分析人员

- 系统设计人员

- 系统开发人员

- 系统测试人员

- 系统质量分析员

- 系统维护人员

**1.2 背景**

随着互联网技术的发展，“互联网+教育”“互联网+科研”之势已不可挡，学术搜索引擎重要性日渐提升。与一般搜索引擎相比，学术搜索引擎具有信息涵盖广、重复率低、相关性好、学术性强等特点，学习人员或相关从业者使用其应当能大幅提升学习、工作效率，对学术研究现状大有裨益。随着新一代搜索引擎的快速发展，学术搜索引擎应具备个性化、智能化、数据挖掘分析、学术圈等特色。

\***软件系统名称** ：中医药学术搜索引擎

​\***任务提出者** ：浙江大学软件工程管理课程任课老师邵健

​\***开发者** ：浙江大学软件工程管理课程14组

\***系统用户** ：中医药行业从业人员及学习者

**1.3 相关技术定义**

1. MySQL ：MySQL是一个关系型数据库管理系统，是最流行的关系型数据库管理系统之一

2. Flask ：Flask是一个轻量级的可定制框架，使用Python语言编写，较其他同类型框架更为灵活、轻便、安全

3. JavaScript ：JavaScript是一种具有函数优先的轻量级，解释型或即时编译型的编程语言

4. Inverted Index ：倒排索引源于实际应用中需要根据属性的值来查找记录。这种索引表中的每一项都包括一个属性值和具有该属性值的各记录的地址。由于不是由记录来确定属性值，而是由属性值来确定记录的位置，因而称为倒排索引。

**1.4 相关项目定义**

1. 用户场景：用户场景通常称为用例，它提供了系统将如何被使用的描述。

2. IPO 图：是输入/处理/输出图的简称，描述输入数据、对数据的处理和输出数据之间的关系。

3. 数据流图：是 SA 方法中用于表示系统逻辑模型的一种工具，它以图形的方式描述数据在系统中流动和处理的过程，它是一种功能模型。

4. 状态图：描绘一个系统或组件可能假设的状态，并且显示引起或导致一个状态切换到另一个 状态的事件或环境。

5. CRC 卡：模型实际上是表示类的标准索引卡片的集合。这些卡片被分为三部分，顶部写类名，下面左侧部分列出类的职责，右侧部分列出类的协作关系。

6. 数据词典：是对所有与系统相关的数据元素的一个有组织的列表、以及精确严格的定义，使得用户和系统分析员对于输入、输出、存储成分和中间计算有共同的理解。

7. 系统集成：将不同的系统，根据应用需要，有机地组合成一个一体化的、功能更加强大的新型系统的过程和方法。

**1.5 参考文献**

[1]《软件工程开发国家标准》

[2] [G06] “软件工程项目实训”项目计划书

[3] [G06] “软件工程项目实训”需求工程计划

[4] [G06]”软件工程项目实训”项目可行性报告

[5] [G06] “软件工程项目实训”前景与范围文档

[6] [G06] “软件工程项目实训”需求规格说明书

[7] [G06] “软件工程项目实训”系统设计及计划

[8] [G06] “软件工程项目实训”系统编码与实现计划

[9] [G06] “软件工程项目实训”用户手册

**2 总体设计**

**2.1 用户需求规定**

**信息查询者需求**

1. 搜索中医药

2. 按照不同属性搜索

3. 筛选搜索结果

4. 模糊搜索

5. 智能补全和热词推荐

6. 查看相关图片

7. 查看中医药搜索结果详情及其他信息

8. 保存搜索记录

9. 朗读界面信息。

**用户类及其特征**

|  |  |
| --- | --- |
| **用户类** | **描述** |
| 信息查询者 | 可以进行简单的关键词查询，查询包含对应名词的词条，另外也可以通过模糊查询根据某些条件来进行词条的筛选，另外可以根据网站提供的综合统计数据进行查看和分析对比，同时会对用户的信息查询行为进行搜集，构建出个性化推荐/推送的内容策略。 |

**2.2 其他需求规定**

**性能需求**

服务器性能要求能满足日常的访问需求，保持并发承受能力，在界面简洁美观的基础上，使网络延时尽可能短。

**输入输出需求**

客户端通过网页形式向用户展现一个简洁友好的操作界面，用户可以点击搜索框进行输入或点击热词推荐、滚动横幅跳转。

服务器后台处理数据后显示在网页界面上。

**数据管理需求**

1.安全 ：数据库对参与数据库操作的数据都会预处理后再加入查询语句，避免通过网页SQL注入的方式获取数据库信息或破坏数据库。

2.性能 ：对于需要频繁访问数据库的操作，需要建立持久的数据库连接。在预处理阶段建立倒排索引提高搜索效率。

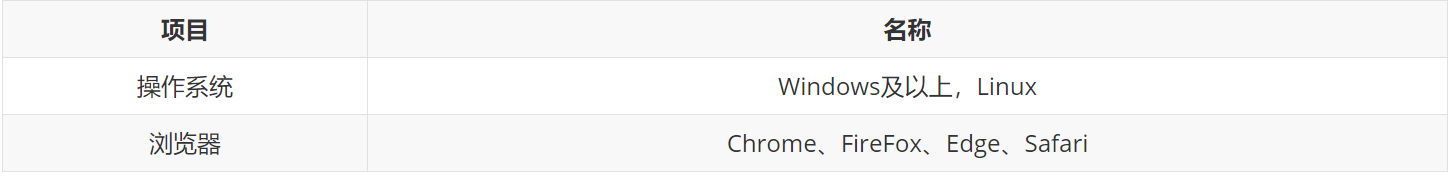
**故障处理需求**

1. 软件开发团队提供系统维护支持，有完备的内部文档和外部文档；

2. 系统报错信息可读性较高。

**2.3 运行环境**

**软件运行环境**



**硬件运行环境**

****

**2.4 基本设计概念和处理流程**

**2.4.1服务器**

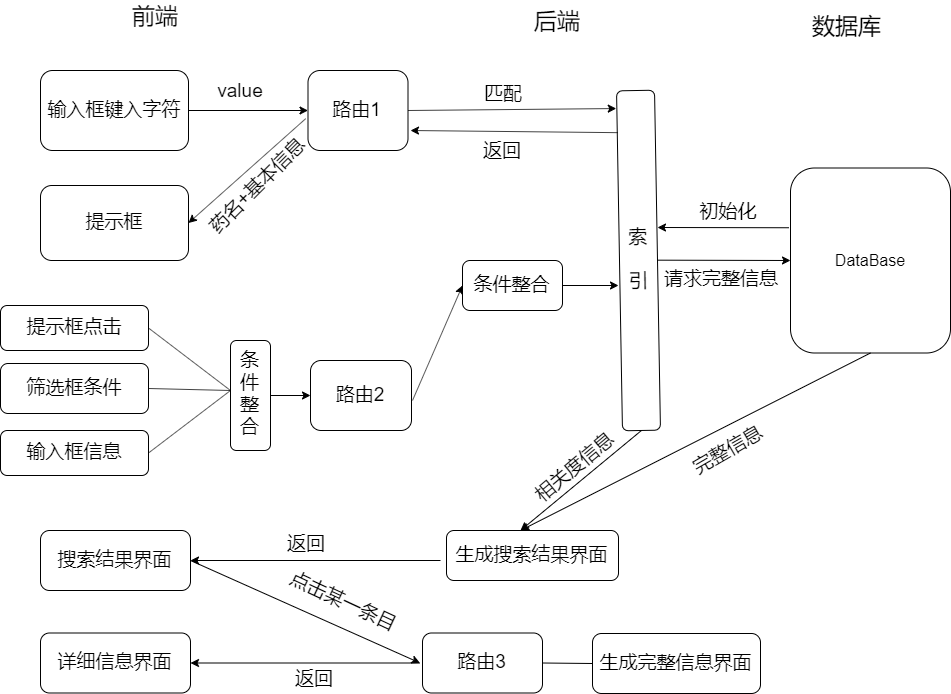
​使用Flask框架编写代码。数据库使用mysql。当用户输入网址以访问网站时，首先后端返回打包好的前端HTML与CSS和JAVASCRIPT文件，用户通过前端向服务器发出各类请求，服务器对请求进行解析，得到结果后返回后端。需要获取详细数据时，服务器必要时会向服务器发出查询指令，拿到数据后再返回前端。前端根据从服务器的到的数据，将内容渲染在网页上并呈现给用户。

**2.4.2客户端**

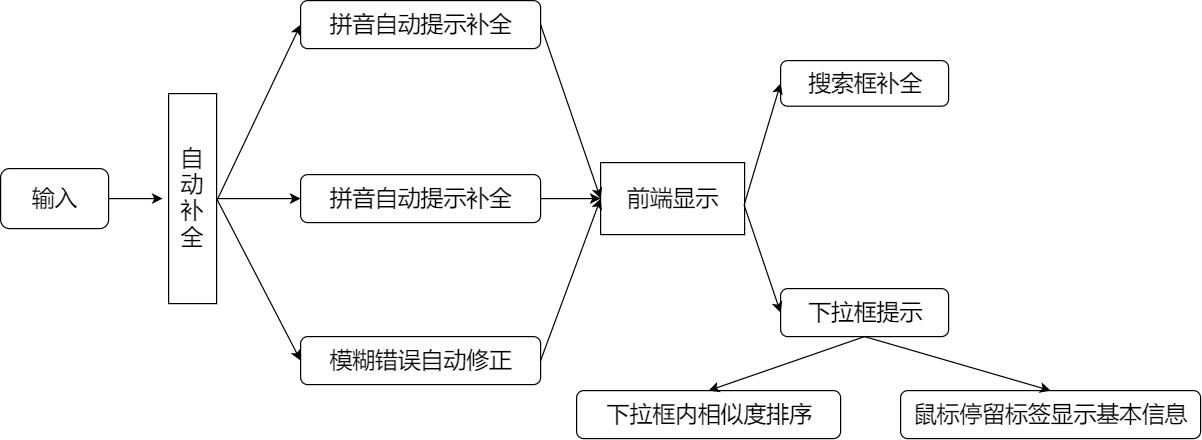
在react框架的加持下，程序可以运行在IE，Chrome, FireFox等多种浏览器上面。前端设计采用单界面应用的设计理念，从后端获取数据，前端负责内容呈现与界面跳转。当用户在前端执行过某个操作时，前端将数据以json的格式发送给服务器，等待服务器的响应数据，并将数据渲染在相应的位置上。

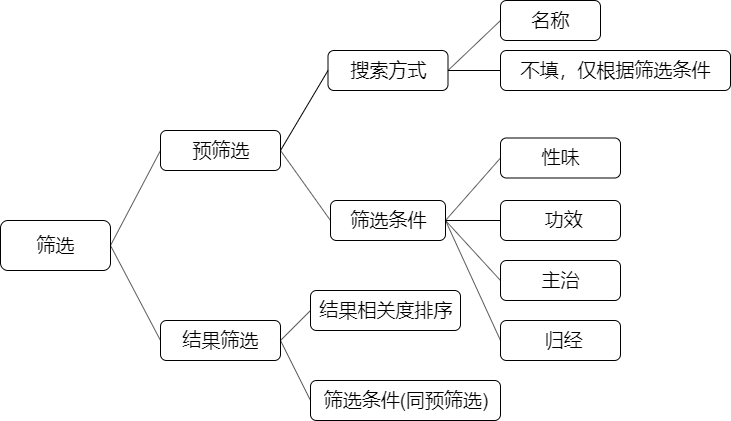
**2.5 结构**

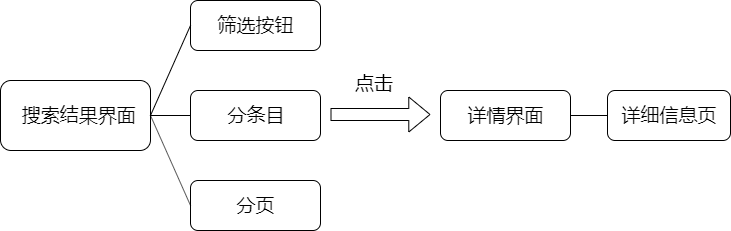
**2.5.1 系统模块设计图**



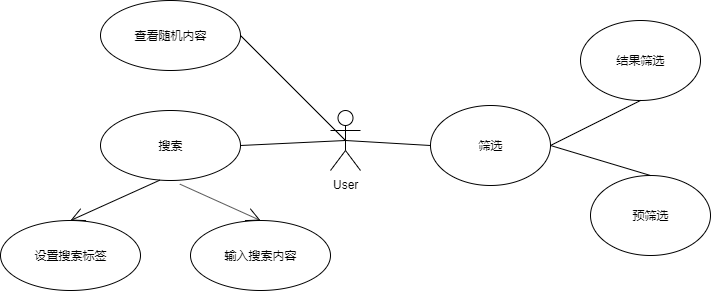
**2.5.2 系统功能设计图**



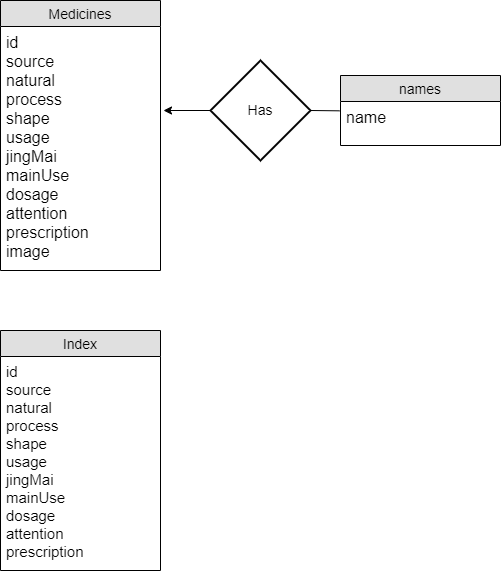




**2.5.3用例图**



**2.5.4 E-R图**



**2.6 用例**

**2.6.1 选择搜索属性**

|  |  |
| --- | --- |
| **用例** | **选择搜索属性** |
| **主要参与者** | 用户 |
| **目标** | 选择搜索的目标属性（可多选） |
| **前提条件** | 无 |
| **触发器** | 用户点击属性列表中的对号 |
| **工作流程** | 1. 点击属性列表中的勾选方框可以添加搜索属性 2. 点击搜索栏上方的已勾选属性可以删除搜索属性 |
| **异常** | 1. 页面渲染问题 2. 数据库连接问题 |
| **优先级** | 必须实现 |
| **何时可用** | 首次增量 |
| **使用频率** | 一般 |
| **使用方式** | 通过浏览器 |
| **次要参与者** | 无 |

**2.6.2 选择搜索属性逻辑**

|  |  |
| --- | --- |
| **用例** | **选择搜索属性逻辑** |
| **主要参与者** | 用户 |
| **目标** | 选择搜索属性之间的与或逻辑 |
| **前提条件** | 勾选多个搜索属性 |
| **触发器** | 点击逻辑选择下拉菜单 |
| **工作流程** | 点击下拉菜单中的与或逻辑 |
| **异常** | 1. 页面渲染问题 2. 数据库连接问题 |
| **优先级** | 必须实现 |
| **何时可用** | 首次增量 |
| **使用频率** | 一般 |
| **使用方式** | 通过浏览器 |
| **次要参与者** | 无 |

**2.6.3 搜索自动补全**

|  |  |
| --- | --- |
| **用例** | **搜索自动补全** |
| **主要参与者** | 用户 |
| **目标** | 根据搜索框内容进行自动补全提示 |
| **前提条件** | 搜索框中有输入内容 |
| **触发器** | 在搜索框中输入内容 |
| **工作流程** | 1. 用户改变输入框内容 2. 前端回调函数请求数据库相似数据 3. 在搜索框下显示自动补全内容 |
| **异常** | 1. 页面渲染问题 2. 数据库连接问题 |
| **优先级** | 必须实现 |
| **何时可用** | 首次增量 |
| **使用频率** | 一般 |
| **使用方式** | 通过浏览器 |
| **次要参与者** | 无 |

**2.6.4 自动补全显示图片卡片**

|  |  |
| --- | --- |
| **用例** | **自动补全显示图片卡片** |
| **主要参与者** | 用户 |
| **目标** | 用户把鼠标放在自动补全上时显示出条目小卡片 |
| **前提条件** | 有自动补全 |
| **触发器** | 把鼠标放在自动补全上 |
| **工作流程** | 1. 把鼠标放在自动补全上 2. 前端请求后端条目数据 3. 前端渲染该条目的带图片小卡片 |
| **异常** | 1. 页面渲染问题 2. 数据库连接问题 |
| **优先级** | 必须实现 |
| **何时可用** | 首次增量 |
| **使用频率** | 一般 |
| **使用方式** | 通过浏览器 |
| **次要参与者** | 无 |

**2.6.5 搜索**

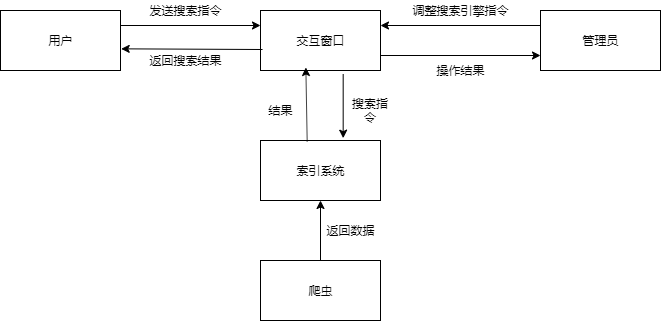
|  |  |
| --- | --- |
| **用例** | **搜索** |
| **主要参与者** | 用户 |
| **目标** | 根据所有条件进行搜索功能 |
| **前提条件** | 搜索的各种条件都充分 |
| **触发器** | 点击搜索按钮 |
| **工作流程** | 1. 用户选择搜索条件并点击搜索按钮 2. 前端请求后端的索引数据 3. 跳转到搜索结果页面 |
| **异常** | 1. 页面渲染问题 2. 数据库连接问题 |
| **优先级** | 必须实现 |
| **何时可用** | 首次增量 |
| **使用频率** | 一般 |
| **使用方式** | 通过浏览器 |
| **次要参与者** | 无 |

**2.6.6 随机推荐**

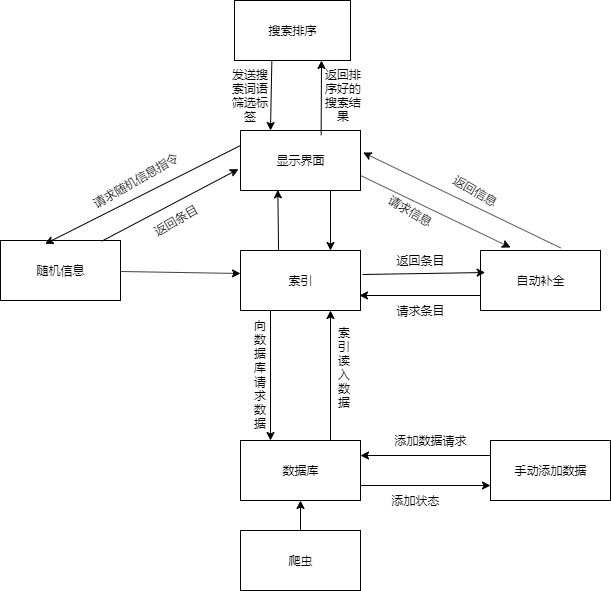
|  |  |
| --- | --- |
| **用例** | **随机推荐** |
| **主要参与者** | 用户 |
| **目标** | 随机推荐中医药条目 |
| **前提条件** | 无 |
| **触发器** | 渲染页面时，之后每隔一段时间也会触发 |
| **工作流程** | 1. 页面生成时，前端请求后全数据 2. 根据随机返回的中药条目渲染随机推荐列表 3. 之后每隔一段时间重复上述步骤 |
| **异常** | 1. 页面渲染问题 2. 数据库连接问题 |
| **优先级** | 必须实现 |
| **何时可用** | 首次增量 |
| **使用频率** | 一般 |
| **使用方式** | 通过浏览器 |
| **次要参与者** | 无 |

**2.7 数据流图**

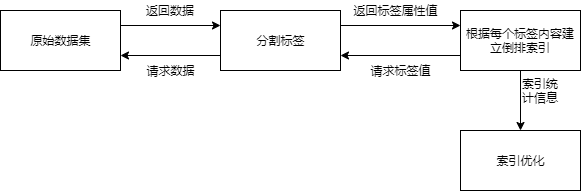
**2.7.1 顶层数据流图**

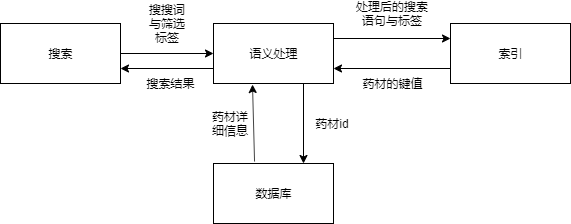


**2.7.2 中层数据流图**



**2.7.3 底层数据流图**





**2.8 数据字典**

**2.8.1 用户搜索请求**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **数据元素** | **描述** | **数据类型** | **数据长度** | **数据取值** |
| words | 用户在搜索框之中输入的内容 | 字符串 | （0-50） | 任意中英文字符 |
| tags | 用户所要选择的搜索的药材性质 | | 列表 | 根据药材性质 | 只能选取药材具有的性质 |
| logic | 用户搜索时使用的逻辑 | 字符串 | 只有两个值可以选择 | 'or' 'and' |

**2.8.2 药材详细信息**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **数据元素** | **描述** | **数据类型** | **数据长度** | **数据取值** |
| id | 药材编号 | int | 无 | 正整数递增 |
| source | 来源 | varchar | 1023 | 无 |
| Image | 药材图片 | blob | 无 | 无 |
| process | 炮制 | varchar | 1023 | 无 |
| shape | 性状 | varchar | 1023 | 无 |
| usage | 功效 | varchar | 1023 | 无 |
| jingMai | 经脉 | varchar | 1023 | 无 |
| dosage | 用法用量 | varchar | 1023 | 无 |
| attention | 注意禁忌 | varchar | 1023 | 无 |
| prescription | 药方 | varchar | 2047 | 无 |

**3 接口设计**

**3.1 用户接口**

本系统为中草药搜索引擎系统，用户所有的行为均在网页页面实现，用户通过在搜索框输入关键词，选择标签，选择查询逻辑并点击查询按钮完成搜索功能，用户主要通过按钮，下拉联想词条框等可视化元素雨服务器后端进行交互，主要接口如下.

|  |  |  |  |
| --- | --- | --- | --- |
| **一级链接** | **二级链接** | **三级链接** | **四级链接** |
| 搜索引擎主页 | 搜索内容 |  |  |
| 搜索页面 | 搜索结果列表 | 详细信息 | 关键词出现情况 |
| 选择标签 | 选择标签列表 |  |  |
| 详情界面 | 药材详细信息 | 图片 |  |

**使用者接口**

1. 搜索关键字

2. 选择搜索标签

3. 指定搜索逻辑

4. 查看下拉联想词条

5. 查看药材详细信息

6. 查看并点击药材词条

**3.2 外部接口**

本系统的部分数据存储在服务器及数据库中，搜索引擎所需数据以文本形式以及中间文件形式存储。索引建立之后统一以字典的形式存储在数据库之中。网页前端获取用户输入后，由网页后端完成与服务器及数据库的交互。利用Python，Flask以及索引之间的接口完成网站外部接口数据。初始数据由人工爬取并存储在数据库之中。

**3.3 内部接口**

本系统总体分为搜索界面，搜索结果显示界面以及详情显示等模块，各模块之间耦合度较低，分别有不同的用户界面，但各模块共享数据库表项，各模块之间相对可以独立开发，但必须先统一数据库数据。实现功能的互通与链接。

**4 运行设计**

**4.1 运行模块的组合**

本搜索引擎按照用户操作来划分模块，各个模块不共享界面，相对独立，每个模块按照流程可以划分为客户端界面，客户端脚本以及后台服务程序（与数据库链接）。

​ 各个模块之间不共享界面，但是共享数据库数据以及搜索引擎，后台程序只共享数据库连接和搜索引擎。

**4.2 运行控制**

1. 界面是用户直接与系统交互的部分，界面力求简洁而不简陋，能引导用户进行无碍操作。设计时，以在提供用户便捷操作的基础上增加美观度为基准。同时，需要在有限的空间内尽量丰富功能，达到方便而又不冗余的结构。

2. 运行控制的条件与限制

本项目的开发要求小组成员足够的参与度，能及时保质保量完成任务。且项目开发过程中可能会有技术上的难点和设备方面的欠缺，需要开发小组合理利用现有设备和资源，积极查找资料解决问题，在完成项目开发的基础上，同时保证项目的可用性、安全性、可维护性等。

3. 前台与后台的关系

前台主要展示随机信息，搜索框，以及在搜索之后具体展示搜索信息，内容信息。后台主要负责业务流程逻辑以及检索功能，控制前台显示信息，负责与搜索引擎和数据库的交互。

**4.3 运行时间**

用户在做搜索时候前端不不断的向后端搜索引擎请求数据，会频繁与数据库交互以获取信息，会占用较多的数据库资源。

**5系统数据结构设计**

**5.1 逻辑结构设计要点**

**倒排索引设计**

1. 对文档集合进行分词，将每篇文档中的所有词语提取出来。
2. 为每个词语建立一个词条，词条包含该词语在文档集合中出现的次数和位置信息。
3. 将所有词条按照词语的字典序进行排序。
4. 建立一个指向词条位置的索引，使用哈希表存储。
5. 将倒排索引的二进制字节流储存在数据库中，当程序运行时从数据库读取倒排索引的二进制字节流并将其序列化。

**MySQL数据库设计**

Table: medicines:

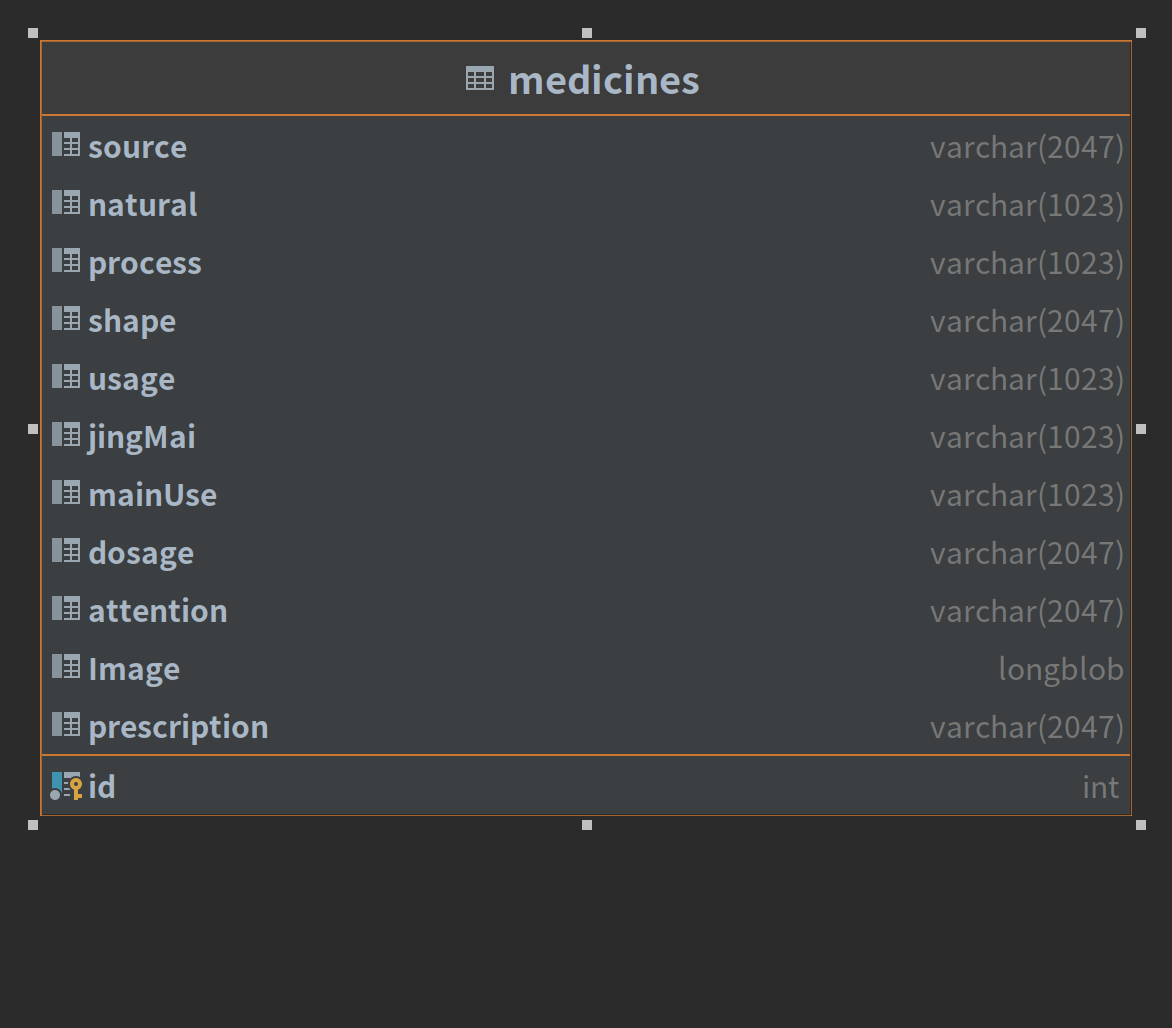


Table: names:

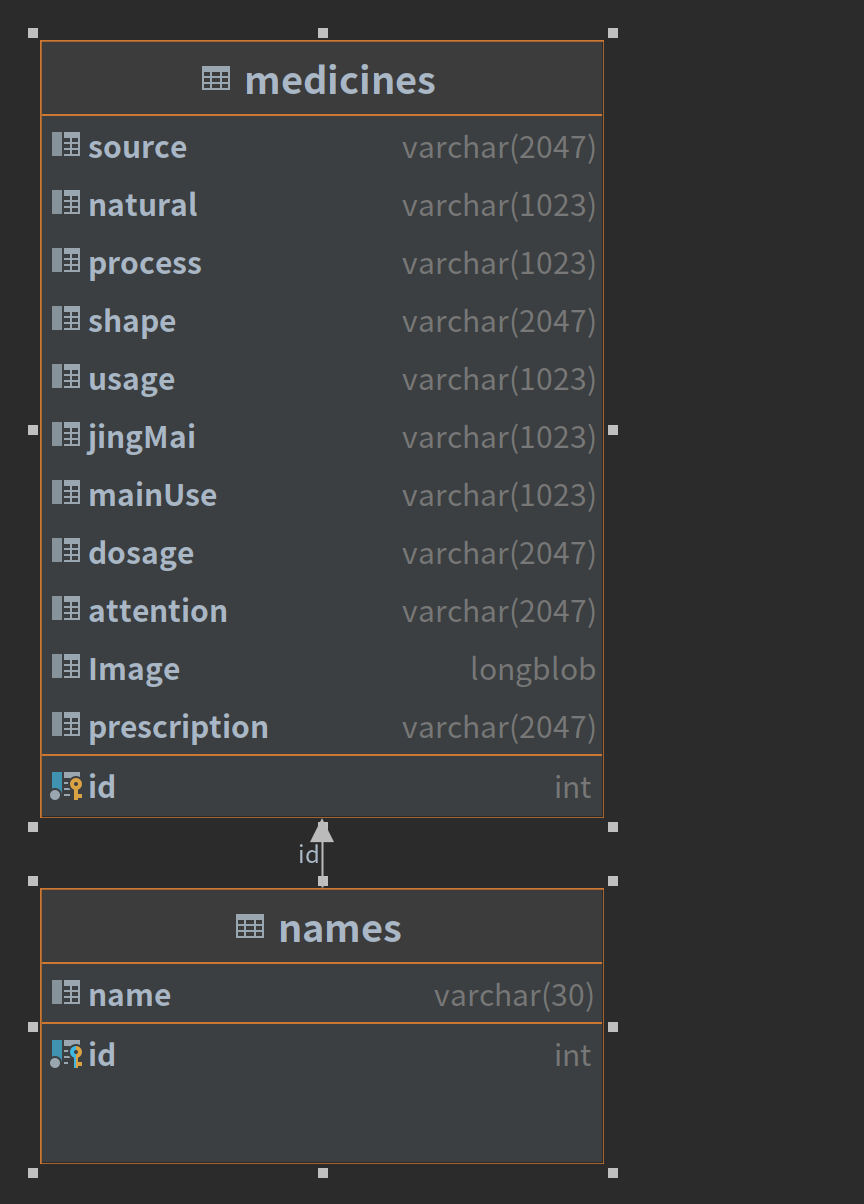


Table: Alias:

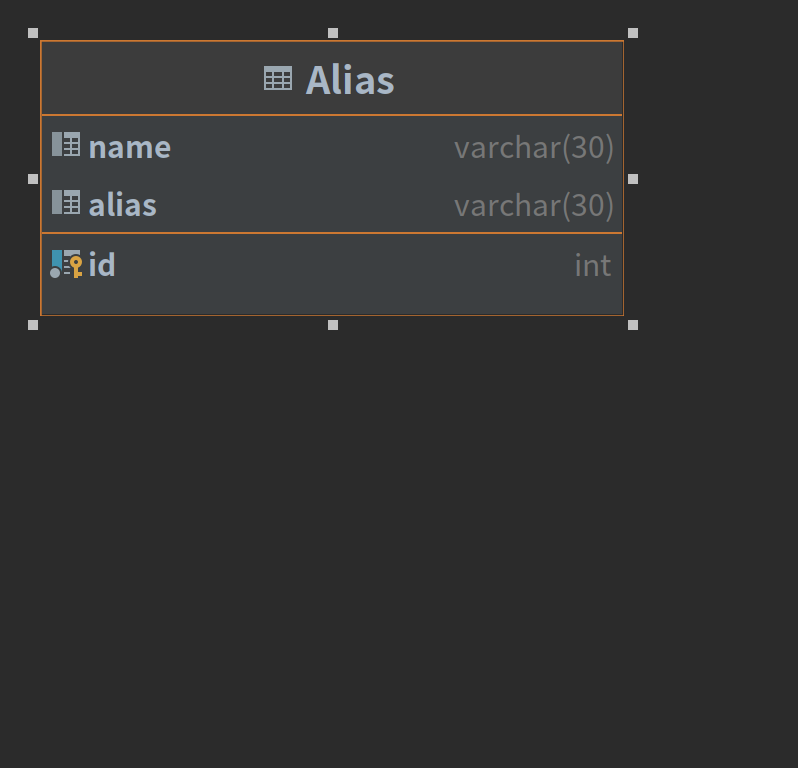
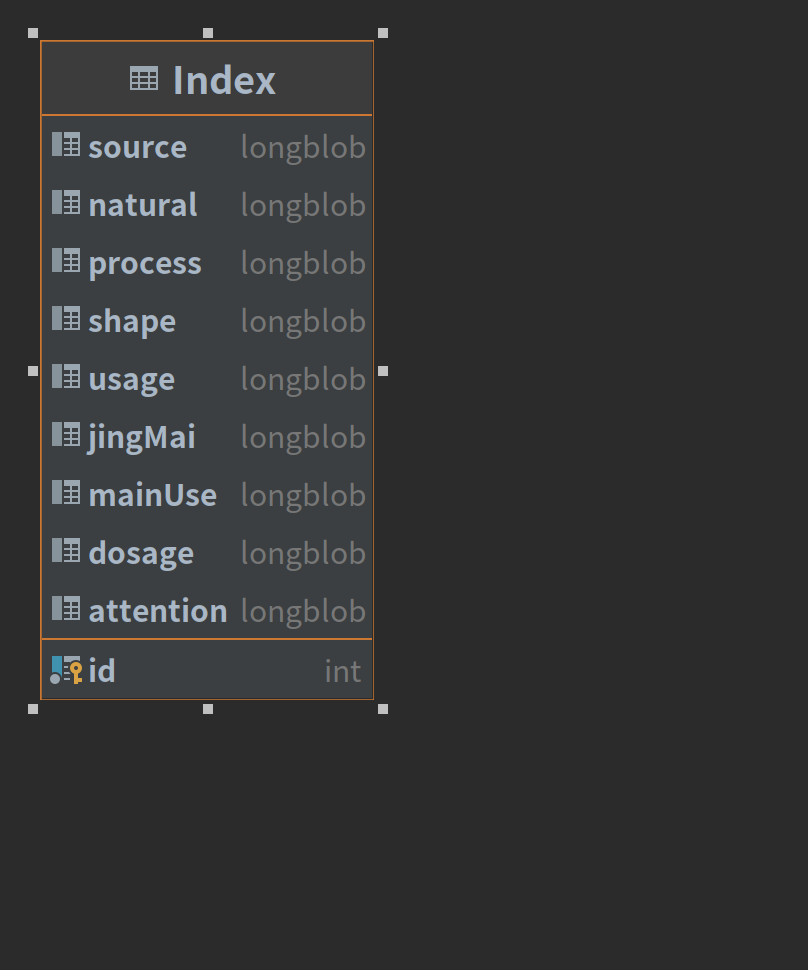


Table: Index:



**5.2 物理结构设计要点**

1. 药材信息存储在数据中，通过SQL语句访问数据库获取。
2. 倒排索引储存在数据库中，当“中医药百科”启动时，自动从数据库中获取，并且常驻内存。

**6. 系统出错设计**

**6.1 出错信息**

|  |  |  |
| --- | --- | --- |
| **输出信息形式** | **含义** | **处理方法** |
| 数据库连接失败 | 由于并发访问数据库的操作数量很大，导致某些对数据库的访问出现异常进而导致网站崩溃 | 对数据库访问的语句使用“try…… catch……”语句，当访问出现异常时，回滚该操作 |
| 磁盘损坏 | 由于物理因素等，导致数据库中的数据丢失 | 定期对数据库中的数据进行备份 |
| 数据库读取乱码或汉字输出为'?' | 客户端页面，数据库，数据库读取过程编码不一致 | 客户端页面，数据库，数据库读取过程编码不一致 |

**6.2补救措施**

系统备份 ：定期备份系统数据，当系统数据因不可抗拒力丢失时，可以启用备份数据。

分布式部署 ：将系统部署到不同计算机上，减小硬件损坏造成的数据丢失的影响。

**7 系统维护设计**

**7.1 概述**

1. 操作数据库时，try catch语句捕获异常，并对操作进行回滚。对不同的错误信息尽量区分输出。
2. 管理员有权对整个网站的状况进行控以防系统出现不可预计的错误防止系统显示不合法信息。
3. 系统维护人员每次维护后需要留下完备可读的系统维护日志便于管理员和其他维护人员查看。

**7.2 检测点设计**

**7.2.1 搜索功能**

测试流程：

1. 进入首页
2. 输入搜索内容
3. 选择搜索标签
4. 查看搜索结果
5. 查看详情页面

**7.2.2 随机返回药材**

测试流程：

1. 将每次随机返回的药材数量提高至200种，观察系统是否崩溃
2. 鼠标移动到随机返回的药材图片
3. 查看药材的简要信息
4. 点击药材图片
5. 查看药材详细信息

**7.2.3 管理员修改数据库资料**

测试流程：

1. 输入要导入药材的具体信息
2. 提交到数据库
3. 打开数据库，查看对应表是否已经修改
4. 打开搜索页面，查询新导入药材的信息
5. 查看新药材信息的详情页面

**7.3 系统维护设计**

1. 硬件资源维护：定期清理服务器硬盘垃圾，可根据网站实际需求选择升级服务器性能
2. 数据库维护：定期备份数据库文件
3. 系统功能升级：根据用户实际访问平台的需求，对于系统功能进行合理的更新