2020-7-4

《系统设计报告》

——NotOnlyFilm 电影垂直搜索引擎

《项目实训》2 班 G23

组长：张佳瑶

组员：贺婷婷 应承峻 戴陈威 杨建伟

修改历史

修订日期	版本号	作者	修改内容	审核者
2020-07-04	1.0	贺婷婷、张佳瑶、应承峻、杨建伟、戴陈威	初稿	戴陈威
2020-07-06	2.0	贺婷婷、张佳瑶、应承峻、杨建伟、戴陈威	增加图例，增加项目架构设计、项目任务分解	张佳瑶
2020-07-07	3.0	贺婷婷、张佳瑶、应承峻、杨建伟、戴陈威	增加各模块设计	杨建伟
2020-07-07	4.0	贺婷婷、张佳瑶、应承峻、杨建伟、戴陈威	对图表进行了统一编号，根据项目的实际开发情况修改了需求以及模块设计部分的相关描述	戴陈威

目录

修改历史.....1

1 引言.....6

1.1 编写目的.....6

1.2 背景.....6

1.3 定义.....7

1.4 参考文献.....8

2 需求规定.....9

2.1 用户需求规定.....9

2.1.1 搜索词条.....9

2.1.2 查看搜索结果.....9

2.1.3 详情内容展示.....10

2.2 其它需求规定.....10

2.2.1 性能需求.....10

2.2.2 输入需求.....10

2.2.3 数据传输与并发需求.....11

2.2.4 数据管理需求.....11

2.2.5 权限与安全需求.....12

2.2.6 可视化需求.....13

2.2.7 防护性需求.....13

2.2.8 软件质量属性.....14

2.2.9 其它需求.....14

3 总体设计.....15

3.1 功能设计.....15

3.2 用户类型及用户特征.....16
3.3 运行环境.....16
3.4 基本概念和处理流程.....17
3.5 结构.....18
3.5.1 用户需求分析图.....18
3.5.2 系统模块架构图.....18
3.5.3 数据流图.....19
3.5.4 ER 图.....20
3.5.5 关键 IPO 图.....21
3.5.6 数据字典.....22
3.6 人工处理过程.....24
3.7 尚未解决的问题.....24
4 接口设计.....25
4.1 用户接口.....25
4.2 外部接口.....25
4.3 内部接口.....25
5 运行设计.....27
5.1 运行模块的组合.....27
5.2 运行控制.....27
5.3 运行时间.....27
6 总体数据设计.....28
6.1 数据存储.....28
6.2 数据安全.....28
6.3 逻辑结构设计要点.....28
6.3.1 ElasticSearch 索引设计.....28

6.3.2 MySQL 数据库设计 31
6.4 物理结构设计要点.....33
7 系统出错设计.....34
7.1 出错信息.....34
7.2 补救措施.....34
8 系统维护设计.....35
8.1 概述.....35
8.2 检测点设计.....35
8.2.1 搜索词条.....35
8.2.2 查看搜索结果.....36
8.2.3 详情内容展示.....36
8.3 相关维护设计.....36
9 模块设计计划.....37
9.1 项目架构设计.....37
9.2 项目任务分解.....37
9.3 前端.....37
9.3.1 搜索首页模块.....37
9.3.2 搜索结果模块.....38
9.3.3 内容展示模块.....40
9.4 搜索服务器.....41
9.5 后端服务器.....42
9.6 数据模块设计.....42
9.6.1 信息爬取.....42
9.6.2 定期爬取.....43
9.6.3 唯一标志.....43

9.6.4 数据分析.....43

1 引言

1.1 编写目的

该系统设计报告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

本项目开发的软件为一个电影垂直搜索引擎。自信息革命以来，全球经济高速发展，新兴工业化进程加快，后工业时代即将到来，娱乐产品逐渐形成产业链。如今，文娱已经成了人们生活中不可或缺的一部分。当今正值新冠肺炎肆虐之时，能够便捷地享受到足不出户的文娱活动成了人们迫切的需求。“电影是生活的隐喻”，电影作为文娱的代表元素，能够让我们转变看世界的角度，开阔视野，足不出户就能了解外界，实乃上佳的家庭娱乐方式。

在目前，电影的信息散落在互联网世界的四处，想要获得一部电影的资源往往需要通过搜索引擎的帮助。但传统的搜索引擎涉及面广，搜索结果包含很多与电影无关的杂乱信息，这些杂乱的信息使得用户检索成本变高。同时传统的搜索引擎也无法针对电影做分类筛选、分类内容展示等，用户想查看某种类型的电影

或关心电影的某种属性，往往需要在搜索中加入相应的关键词，但传统搜索引擎基于文本匹配往往难以理解用户的专业化要求。

针对电影领域的垂直搜索引擎能有效地解决上述问题，NotOnlyFilm 针对电影领域提供专门的信息检索服务，同时对电影信息数据进行分类、整理和聚合分析，能让用户进行高效地检索和信息查询，使用户能便捷地获得电影资源、资讯、种子、评分等信息。

网站系统名称：NotOnlyFilm 电影垂直搜索引擎

项目提出方：浙江大学项目实训课程任课老师——邢卫、邵健

项目开发者：浙江大学项目实训 G23 小组

系统用户：对电影搜索有需求的潜在用户

1.3 定义

HTML

超文本标记语言（Hyper Text Markup Language），是标准通用标记语言下的一个应用，用于描述因特网上的网页文档。

CSS

层叠样式表（Cascading Style Sheets），是一种用来表现 HTML 等文件样式的计算机语言，在网络中能够对网页中元素位置的排版进行像素级精确控制。

UML

统一建模语言（Unified Modeling Language），是一套用来设计软件蓝图的标准建模语言，是一种从软件分析、设计到编写程序规范的标准化建模语言。

B/S 系统

浏览器/服务器系统。只安装维护一个服务器(Server)，而客户端采用浏览器(Browser)运行软件。

Vue

Vue 是一个用于创建用户界面的开源框架，也是一个创建单页应用的 Web 应

用框架，一套用于构建用户界面的渐进式框架。

Spring Boot

后端服务框架。

Elasticsearch

Elasticsearch 是一个基于 Lucene 的搜索服务器。它提供了一个分布式多用户能力的全文搜索引擎，基于 RESTful web 接口。Elasticsearch 是用 Java 语言开发的，并作为 Apache 许可条款下的开放源码发布，是一种流行的企业级搜索引擎。

MySQL

一个小型关系型数据库管理系统。

Nginx

Nginx 是一款负载均衡服务器软件，它可以运行在几乎所有广泛使用的计算机平台上，由于其跨平台和安全性被广泛使用，是最流行的 Web 服务器端软件之一。

1.4 参考文献

[1]《软件工程 实践者的研究方法》，Roger S.Pressman，机械工业出版社

[2]《软件需求（第三版）》，Karl Wiegers Joy Beatty，清华大学出版社

[3]《软件工程开发国家标准》

[4]《G23-项目计划书-20200701》

[5]《G23-项目章程-20200701》

[6]《G23-质量保证计划-20200702》

[7]《G23-需求规格说明书-20200703》

2 需求规定

2.1 用户需求规定

2.1.1 搜索词条

用户可以输入关键字进行搜索，关键字支持与或非等逻辑运算。

系统支持用户进行高级搜索，用户可以根据搜索符号规则组织搜索输入语句，进行搜索范围限定，获得更加精准的搜索结果。例如：关键词过滤、关键词并集、关键词交集等。

用户输入时系统可以进行智能补全并识别用户输入的拼音搜索。

用户输入时，搜索框默认显示用户最近的两次搜索内容，在自动补全时，优先显示用户历史搜索内容。

2.1.2 查看搜索结果

用户在输入关键词或者查询语句，点击搜索按钮后，系统可以将搜索结果呈现给用户。搜索结果可以以卡片列表的形式展示出电影的基本信息，例如电影名、电影图片、导演、演员、评分、剧情简介等。

用户可以通过点击具体的搜索条目进入查看电影详情，电影详情页展示电影更多元的信息，例如电影的演职员信息、各大视频网站的跳转链接、影评、电影资讯等，同时还会智能化地猜测你喜欢的电影并给出推荐。

搜索结果默认按照搜索的相关度、时间排序。同时，用户可以通过自主选择排序规则和筛选条件，例如点击量、点赞数以及相关电影结构化信息的不同类别等，对搜索结果进行二次排序和筛选。

用户可以对每一条电影搜索结果进行点赞或者取消赞。

用户可以在页面右侧查看到系统针对用户输入进行的个性化推荐内容，即一些同类词条，用户可以通过点击跳转进入相应页面。

用户还可以在搜索结果页面与智能机器人进行对话，询问一些基本的问题，例如：周星驰和林子聪合作过哪些电影、冯小刚演过哪些类型的电影等，机器人会给出相应的解答。

2.1.3 详情内容展示

用户可以通过点击搜索结果，进入搜索引擎对该搜索结果进行聚合分析后的页面。该页面包含有电影的基本信息、最新资讯、影评、评分和各大视频网站的观看链接。同时该页面还会展示一些分析得出的数据，以图形化和列表等多样化的形式展现，例如电影的实时热点排行、电影的搜索指数、同类电影的纵横向对比、相关电影智能化推荐、电影的关键词词云等。

2.2 其它需求规定

2.2.1 性能需求

①系统应保证运行稳定，避免出现崩溃；

②主流浏览器均能正常访问本系统；

③系统应能保证至少 1000 人的并发访问；

④系统应允许 20 人同时下载电影种子链接的平均速度达到 5Mbps；

⑤当用户进行任何操作时，系统应该能及时进行反应，反应的时间 1s 以内；

⑥系统应该能及时检测出各种非正常情况，如与设备的通信中断断开，无法连接数据库服务器等情况，避免用户长时间等待；

⑦用户提交查询、跳转页面等操作后，响应时间不超过 3s；

⑧每个页面一般情况下应在 1s 内加载完毕，高峰期应在 3s 内加载完毕；

⑨系统保证在一周内不超过一次维护与重启。

2.2.2 输入需求

①用户输入数据时，应对数据输入进行数据有效性和安全性检查；

②用户搜索电影时，应对文本的长度和安全性进行检查；

③用户搜索电影时，应对数据的有效性和合法性进行检查；

④此外，系统应通过程序控制出错几率，减少系统因用户人为的错误引起的破坏，开发者应当尽量周全地考虑到各种可能发生的问题，使出错的可能降至最小。

2.2.3 数据传输与并发需求

①系统能支持 20 名用户同时搜索电影，并且人均速度能达到 5Mbps；

②系统应支持 1000 名用户并发使用，并保证性能不受影响；

③在网页中，系统生成的所有 Web 页面，通过速率为 5Mbps 的调制解调器在不超过 3 秒的时间内可以全部下载下来。

2.2.4 数据管理需求

系统既要与其他系统有接口，又必须保证本系统的独立性与完整性。即应防止未经授权的各类人员对本系统进行设置和修改或进行有关统计。

系统服务器软件必须提供可靠的数据备份和恢复手段，在服务器软件或硬件出现严重故障时，能够根据备份的数据和账户信息等必要的配套信息，迅速彻底地恢复正常运行环境。

系统的用户信息管理相关模块，决定了其他众多系统的账户安全性，必须保证统计数据准确、安全，用户信息应当提供完整的备份及恢复措施。

无论访问者账户信息还是管理者账户信息，都必须提供完备手段由用户自定义和备份保存，软件开发者不得在系统中预留任何特殊账户和密码。

除此之外，系统应具备加密登录、数据加密传输等安全方面的保障，保证数据在不用系统间传输过程中的保密性与安全性。

以下为具体细则：

①系统服务器应具备至少 100GB 的存储空间，以应对数据增长；

②本系统用于日志等记录的数据增长约为 0.5MB/天，用于存储搜索引擎数据

更新及增加的数据增长约为 10MB/天，具体增长速度由实际爬取的数据量决定；

③本系统会在刚上线时为提供一定的搜索来源数据量，爬取现有的相关信息，希望预留 20-30GB 的存储空间，具体增长量由实际抓取的数据量决定，后续定期爬取数据更新搜索引擎存储资料，故系统服务器应具备足够的存储空间；

④系统管理员应做好备份策略设计，每周备份增量数据，每月备份全量数据，当出现重大事故造成数据丢失后，系统应能在 48 小时内恢复数据；

⑤当系统崩溃后，系统应能在 24 小时内恢复运行；

⑥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。数据库中密码通过不可逆加密存储，限制过短的密码，减低密码泄露的风险；

⑦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。一般操作，为减轻数据库负荷，在操作完成后断开连接。

2.2.5 权限与安全需求

对于任何一个系统来说，安全是保证其正常运行的关键因素之一。因此在我们的系统中，对于安全与权限进行了如下设计：

①所有涉及功能信息或个人信息的网络事务，都应进行加密操作；

②只有系统管理员有权查看及修改底层数据库与搜索引擎的数据，且行为应被系统日志记录，用户无法非法修改数据库；

③系统应该能够记录系统运行时所发生的所有错误，包括本机错误和网络错误，以便于查找错误的原因。系统日志应同时记录下用户的关键性操作信息；

④当流量过大时，优先限制游客流量防止恶意访问。

除此之外，系统应当保证系统自身的安全：

①系统应当提供一定的限制功能，即只允许在局域网内特定机器上运行用户管理功能；

②系统应具备数据加密传输、数据存储等安全方面的保障，以确保系统的安全性；

③系统是基于开放的操作系统平台和数据库上的，因此，要求建立操作系统和数据库的安全保障体系，保证操作系统和数据库的安全；

④对可能发生严重后果的操作要有补救措施，通过补救措施用户可以回到原来的正确状态。对可能造成等待时间较长的操作应该提供取消功能；

⑤对一些特殊符号和计算机代码的输入，与系统使用的符号相冲突的字符等进行判断并阻止用户输入该字符；

⑥对错误操作支持可逆性处理，如取消系列操作。在输入有效性字符之前应该阻止用户进行只有输入之后才可进行的操作。

2.2.6 可视化需求

用户在完成操作后，总是会想知道自己的操作是否出错，为了提高本系统的友好性，我们将对操作结果进行可视化。

①用户搜索电影信息时，相关电影信息会以相关图表的形式可视化呈现；

②用户搜索电影信息时，相关搜索记录会记录在平台中；

③用户对于电影搜索结果提出反馈后，能够在页面内跟踪到处理的进度。

2.2.7 防护性需求

①文件格式错误时，系统提出警告，保持数据库数据不变；

②数据库误删除时，可以使用撤销删除修复；

③重复操作导致卡死时，系统提出警告；

④访问无权限时，系统发出提示并禁止用户访问；

⑤上行文件出错时，系统应提供自主覆盖功能；

⑥系统应该及时信息备份防止病毒攻击；

⑦系统应该能检测到恶意操作；

⑧当检测到恶意重复操作时，系统应提出警告并在一段时间内不允许操作。

2.2.8 软件质量属性

①可用性：系统保证早上 6 点到晚上 12 点之间可用，但在发生紧急情况时允许停止运行一段时间；

②可维护性：系统运行时要保存运行日志，用来维护分析。每周一的凌晨 1 点到 5 点为维护时间，在此期间用户不能使用系统。此外，维护人员需要在系统正常运行时能保持联系；

③兼容性：系统需要保证在主流浏览器（Chrome、IE）上可以正常浏览和使用，对于其他市场占有率超过 5%的浏览器，保证实现系统的主要功能；

④易用性：系统界面应该简洁明了、操作简单，功能按钮的位置符合用户的日常习惯。此外，系统应该要有导航和清晰简短的用户使用手册；

⑤可扩充性：系统在设计上考虑到了网站可能的后续发展，在后端设计和前端设计上尽可能地在满足所有需求的同时，增强了网站的可扩充性。一旦有扩展需要，客户可以联系系统维护人员，维护人员需要在 1-4 个工作日内完成客户的内容扩充需求，主要包括增加新的功能、增加新的模块、界面优化、系统性能提升等。

2.2.9 其它需求

①软件必须提供对系统中各种码表的维护、补充操作；

②软件对用户的所有误操作或不合法操作进行检查，并给出提示信息。

3 总体设计

3.1 功能设计

表 3-1-1 功能设计

功能模块	功能
前端子系统	查询搜索内容
	点赞搜索词条
	搜索结果排序
	搜索结果筛选
	个性化推荐
	搜索结果与详情展示
搜索服务器子系统	Data Import
	Search 接口
后端 Web 子系统	前端搜索接口
爬虫子系统	信息爬取
	自动滤重
	自动分类
	自动爬取

3.2 用户类型及用户特征

表 3-2-1 用户类型及用户特征

用户类	特征与说明
网站用户	1.主要用户； 2.同时使用该网站的用户数目可能较多，主要服务的对象为想要找到相关电影的信息以及关于电影的分析数据或者相关问题解答的电影爱好者； 3.要求能够返回足够精确和足够深度的结果； 4.能够根据用户的喜好进行返回数据的过滤； 5.支持根据用户的输入进入搜索引擎搜索，并且返回相关的内容到网站搜索结果列表页面。
系统管理员	1.次要用户； 2.但是权限比较高，具有数据库和搜索引擎数据库的更新和审核等管理权限； 3.面向的用户数目和电影数据比较多，要求能够提供方便并且快速的管理员接口进行管理； 4.操作频率相对较低，但是每一次的操作对于系统造成的影响比较大。

3.3 运行环境

本网站主要服务于热爱电影的网友，保证至少 1000 名网友同时取得服务的需求，包括数据存储能力和网络吞吐能力，保证账户一定的安全性。

表 3-3-1 软件运行环境

项目	名称	版本
操作系统	Windows 7 及以上，Linux	
网站服务器	Nginx	1.15.8
数据库服务器	Linux socket	
数据库服务器类型	MySQL、Fuseki、Elasticsearch	8.0

浏览器	Chrome	

表 3-3-2 硬件运行环境

项目	名称
操作系统	CPU：CORE i5 及以上
	内存：2G 及以上
	硬盘：500G 及以上
应用服务器 数据库服务器	内存：512M 及以上 硬盘：50G 及以上
通讯设备	网线：具有良好的数据传输能力

3.4 基本概念和处理流程

服务器

以 Nginx 为服务器，Java 语言编写后端代码，数据库采用 MySQL，搜索引擎使用 Elasticsearch。当用户通过浏览器使用网站系统时，浏览器接收用户的请求，并传送到服务器，执行相应的程序，通过搜索引擎 Elasticsearch 查询相关内容，并且从数据库接口函数向数据库发送 SQL 查询语句，数据库接收 SQL 查询语句后执行，返回查询结果，处理查询结果后返回给前端，并显示在网站页面上。

客户端

浏览器采用常用的 IE、Chrome、Firefox 等。客户端在不频繁的操作页面时完成操作后断开与数据库的连接以减轻服务器负荷，在操作频繁时保持连接以增加访问速度。

客户端动态页面：嵌入 Vue，动态网页以数据库技术为基础，能降低网站维护的工作量。

Vue：页面的各种搜索框、筛选框与按键操作能够完成，同时能实现无刷新页面的一些动画效果，包括下拉菜单等。

3.5 结构

3.5.1 用户需求分析图

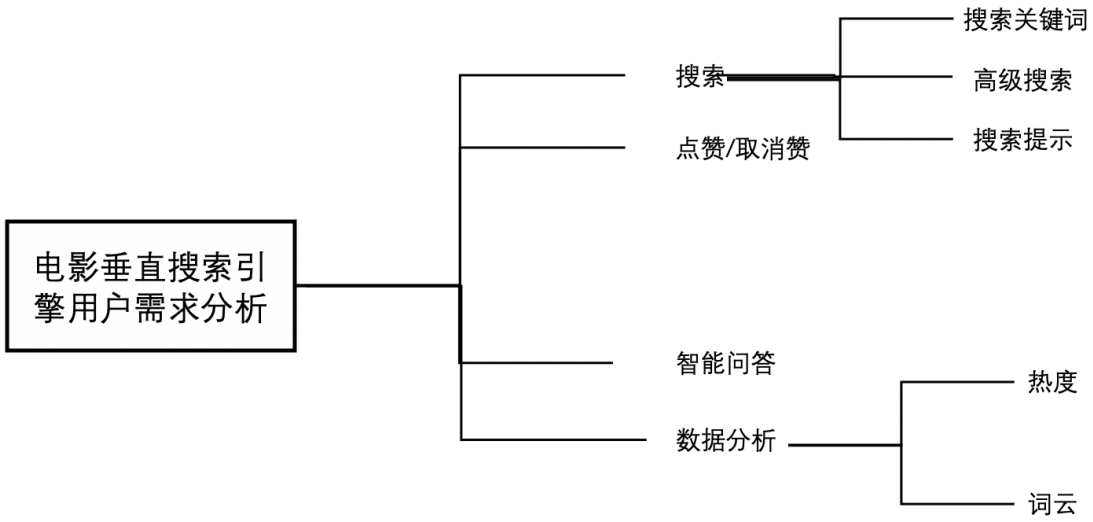

图 3-5-1 用户需求分析图

3.5.2 系统模块架构图

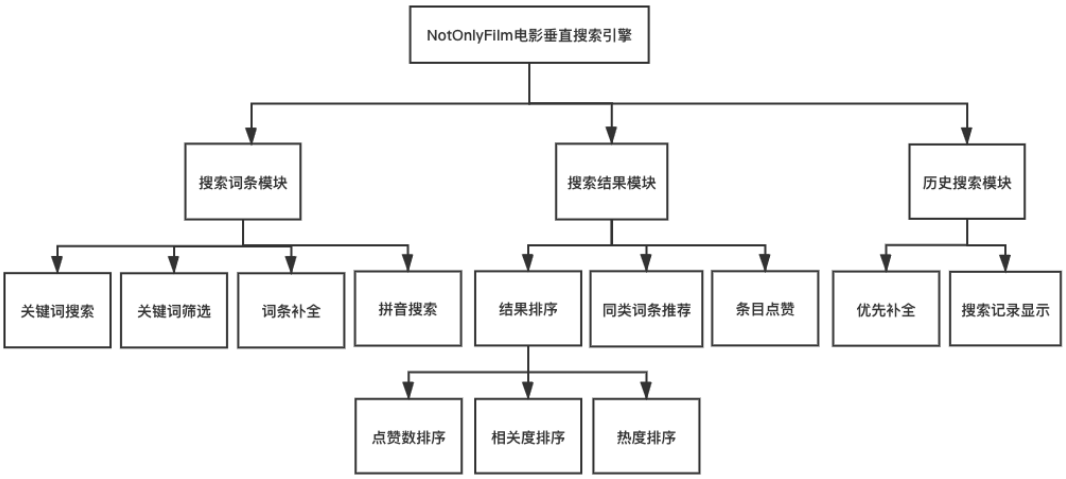

图 3-5-2 系统模块架构图

3.5.3 数据流图

3.5.3.1 搜索引擎子系统数据流图

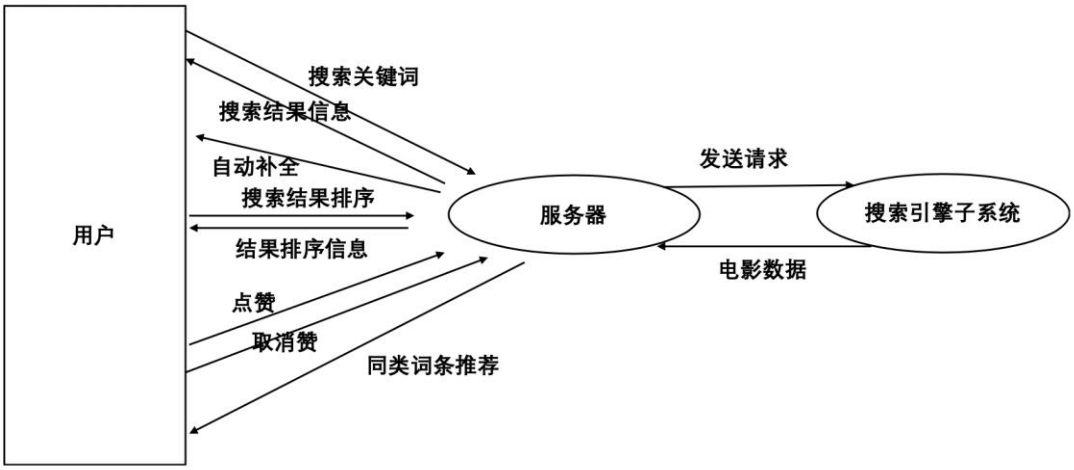

图 3-5-3 搜索引擎子系统数据流图

3.5.3.2 网站维护子系统数据流图

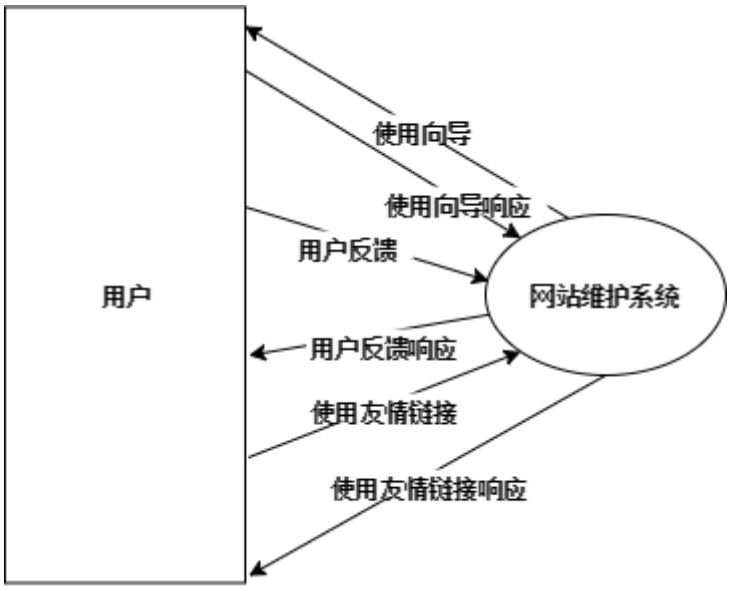

图 3-5-4 网站维护子系统数据流图

3.5.4 ER 图

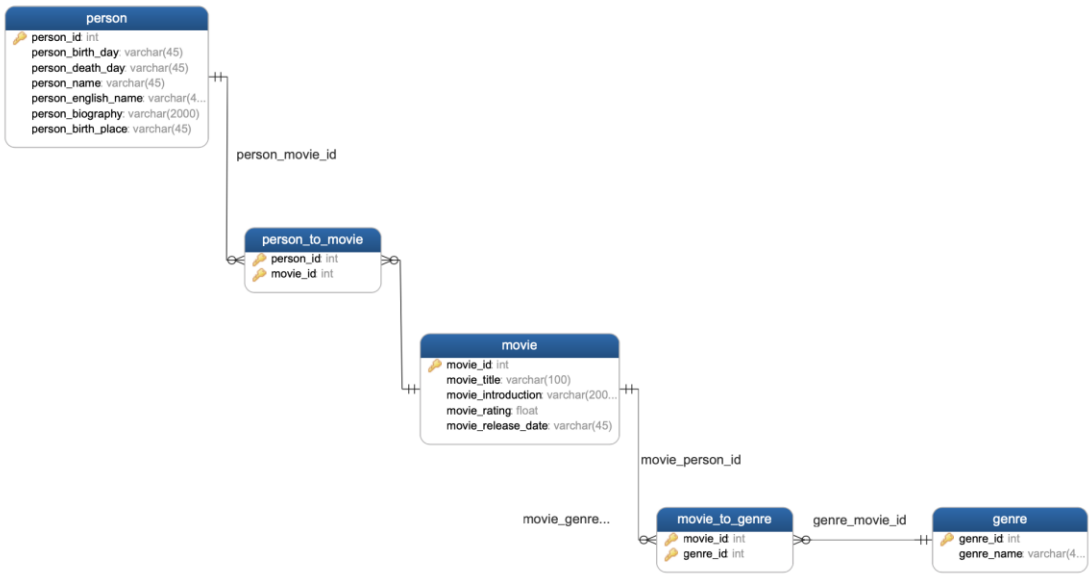

图 3-5-5 ER 图

3.5.5 关键 IPO 图

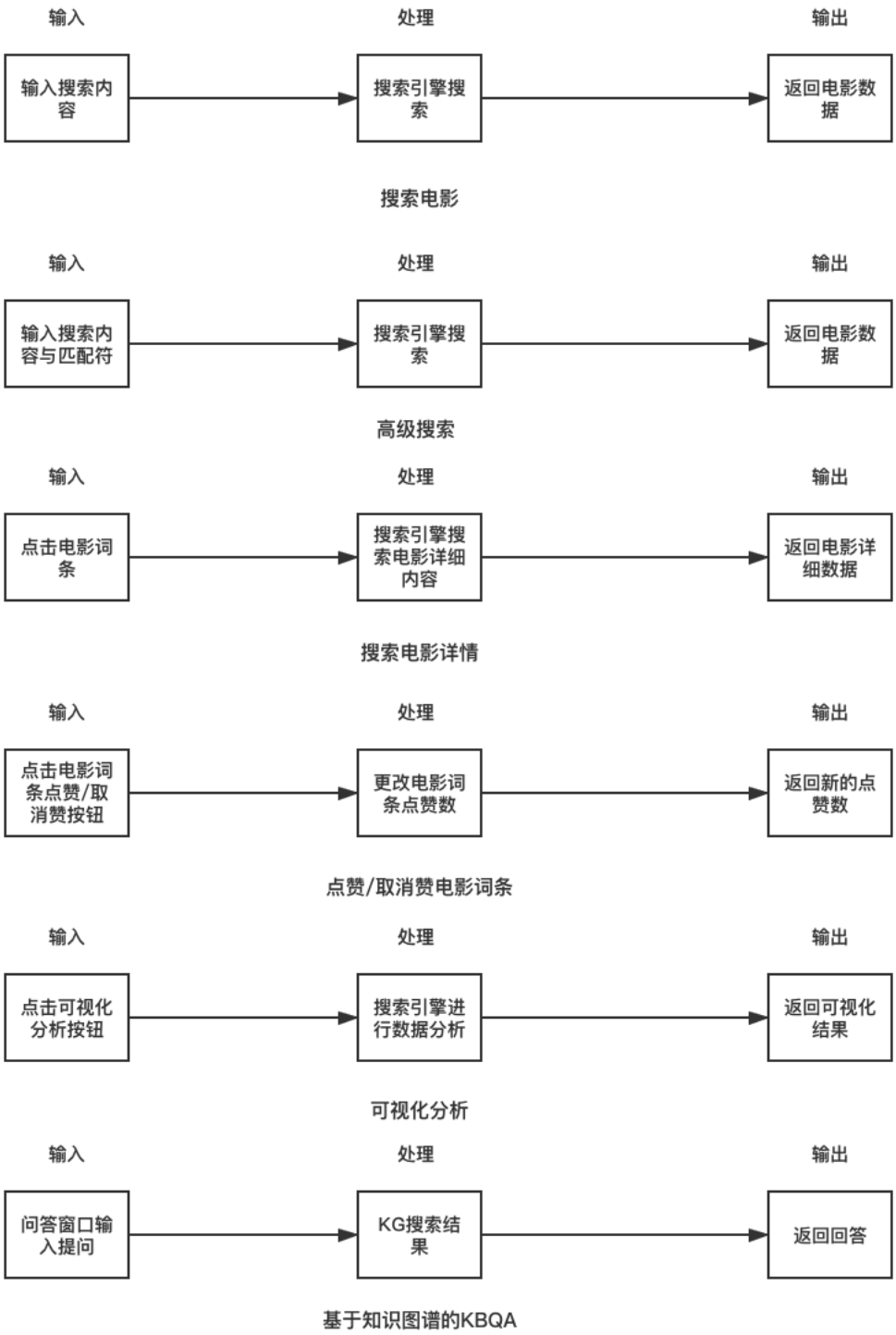

图 3-5-6 关键 IPO 图

3.5.6 数据字典

表 3-5-1 数据元素定义表

编号	数据元素名	类型	值域	说明
E1	查询字符串	字符串	/	用户在搜索引擎输入框中输入的查询字符串
E2	电影名称	字符串	/	/
E3	电影导演	字符串	/	/
E4	电影主演	字符串	/	/
E5	电影类型	字符串	/	/
E6	电影国家/地区	字符串	/	/
E7	电影语言	字符串	/	/
E8	电影上映日期	长整型	当日 0 点的时间戳格式	/
E9	电影片长	整型	0~1000	/
E10	电影别名	字符串	/	/
E11	电影剧情简介	字符串	/	/
E12	电影标签	字符串	/	/
E13	电影评分	浮点型	0.0~10.0	/

E14	电影链接	字符串	/	/
E15	电影热度	长整型	0~2^63-1	/
E16	影评标题	字符串	/	/
E17	影评内容	字符串	/	/
E18	影评链接	字符串	/	/
E19	词条类别	字符串	{“MOVIE”， “COMMENTARY”， “REVIEW”， “INFO”}	电影、解说、影评、咨询

表 3-5-2 数据精度表

数据元素名	类型	精度要求	说明	示例
查询字符串	字符串	128 个字符以内	用户在搜索引擎输入框中输入的查询字符串	速度 激情
电影名称	字符串	64 个字符以内	/	速度与激情
电影导演	字符串	64 个字符以内	/	YingChengJun
电影主演	字符串	64 个字符以内	/	ZhangJiaYao
电影类型	字符串	32 个字符以内	/	悬疑
电影国家/地区	字符串	32 个字符以内	/	中国大陆
电影语言	字符串	32 个字符以内	/	中文

电影上映日期	长整型	/	当日 0 点的时间戳格式	1577808000000
电影片长	整型	/	/	120
电影别名	字符串	128 个字符以内	/	速度激情
电影剧情简介	字符串	128 个字符以内	/	这是简单的剧情简介
电影标签	字符串	128 个字符以内	/	刺激
电影评分	浮点型	精确到 0.1	/	8.5
电影链接	字符串	128 个字符以内	/	www.baidu.com
电影热度	长整型	/	/	120
影评标题	字符串	128 个字符以内	/	速度与激情的影评
影评内容	字符串	128 个字符以内	/	这是影评的内容
影评链接	字符串	128 个字符以内	/	www.baidu.com

3.6 人工处理过程

在本系统的运行过程中，可能会出现一些系统无法自动解决的问题，需要人工处理介入来解决，包括电影记录的删改，相关电影信息的爬取等。

3.7 尚未解决的问题

无

4 接口设计

4.1 用户接口

本系统作为垂直搜索引擎系统，用户所有行为均在网页页面上实现，用户通过鼠标点击或键盘输入完成与系统的交互。用户主要通过输入框、按钮、筛选框、下拉选择框等可视化元素与服务器后端进行交互。用户的主要接口有：

1. 搜索电影信息
2. 查看搜索结果列表
3. 点赞搜索结果条目
4. 查看详细电影信息、详细资讯信息
5. 查看电影数据聚合分析结果
6. 智能问答

4.2 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存储。资源文件及不适宜数据库表项存储的超长文本存储在文件中。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 Java、Vue 与 MySQL 和搜索引擎 ElasticSearch 之间的接口完成网站外部接口设计。

本系统的初始数据依靠人工导入存储。

4.3 内部接口

本系统总体分为前端、后端、数据爬取三个模块。各模块之间耦合度较低，各模块之间使用 JSON 进行数据传输。

前后端接口：

1. 关键词搜索
2. 电影内容筛选

3. 搜索提示

4. 电影详情搜索

数据爬取与后端接口：

1. 数据库存储

5 运行设计

5.1 运行模块的组合

本系统按照交互逻辑划分模块，每个模块不共享界面，相对独立。每个模块按照流程划分为客户端界面，客户端脚本和后台服务器程序。

各个模块之间不会共享界面，但共享数据库数据和搜索引擎，后台程序只共享数据库连接和搜索引擎。

5.2 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。

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、服务器资源等方面的欠缺，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

前台主要展示搜索结果信息、电影详情内容等显示信息，后台主要负责业务流程，控制前台显示信息，负责与搜索引擎和数据库交互。

5.3 运行时间

用户在做搜索时候，前端不断地向 ES 搜索引擎请求数据，会频繁与数据库交互以获取信息，这会占用较多的数据库资源。

6 总体数据设计

6.1 数据存储

项目使用标准 MySQL、Fuseki 数据库以及 ElasticSearch 搜索引擎，按照数据产生、转换和存储的策略，通过将数据导入数据库和搜索引擎的方式进行数据的存储操作。

6.2 数据安全

保证以下完整性、保密性以及可用性三个特性来保护用户的数据安全。

完整性

要求数据未经授权不得进行修改，确保数据在传输和存储过程中不被篡改、盗用和丢失。通过利用安全的框架，在加密的基础上，运用多种方案和技术实现。

保密性

要求对数据进行加密，只有授权者才能使用。这一特性要求加密技术必须自动、实时、精确、可靠。

可用性

要求做到避免因为系统数据泄露而使得合法使用者无法接触可用数据，通过对使用者身份的验证，为合法使用者提供更加安全便捷的使用。

6.3 逻辑结构设计要点

6.3.1 ElasticSearch 索引设计

6.3.1.1 标识

在 Elastic Search 的搜索引擎中，一个索引的定义由字段名和类型组成。如果该字段的类型是一个对象（Object）或者嵌套对象（Nested），则对象中的各个属性的定义也同样是由字段名和类型组成。以下是定义一个索引基本模板：

```
{
    "mappings": {
        "properties": {
            "字段": {
                "type": "<类型>",
                "analyzer": "<指定的分词器>"
            }
        }
    }
}
```

图 6-3-1 ES 搜索引擎索引基本模板

其中 mapping 指明了这是索引的映射配置，properties 指明里面的内容配置的是索引的字段，type 指明索引的类型，analyzer 指明索引的分词器。

本文中涉及的索引类型 type 类型如下：

- text：普通文本，分词器会将这一类文本进行分词，进行倒排索引
- keyword：关键词，分词器不会将这类文本进行分词
- date：日期，格式为”yyyy-MM-dd”或毫秒数
- integer：普通整数
- long：长整数
- double：双精度浮点数
- object：普通对象，在 Elastic Search 的底层存储中，object 对象会被扁平化成数组存储。
- nested：嵌套对象，在 Elastic Search 的底层存储中，nested 对象不会被扁平化，而是将嵌套子对象子存储在单独的文档（doc）中。

6.3.1.2 电影（Movie 类）的索引设计

表 6-3-1 电影（Movie 类）的索引设计

Properties	Type	Analyzer
movieId	keyword	
title	text	ik_pinyin_analyzer
alias	text	ik_pinyin_analyzer

poster	keyword	
staff	object	
staff.staffId	keyword	
staff.name	text	ik_pinyin_analyzer
staff.role	text	ik_pinyin_analyzer
staff.photo	keyword	
directors	text	ik_pinyin_analyzer
scriptwriters	text	ik_pinyin_analyzer
releaseTime	nested	
releaseTime.time	date	
releaseTime.region	keyword	
year	integer	
type	keyword	
country	keyword	
duration	long	
onlineVideo	nested	
onlineVideo.source	keyword	
onlineVideo.url	keyword	
onlineVideo.tag	keyword	
tags	keyword	
introduction	text	
rate	nested	
rate.source	keyword	
rate.rating	double	
rate.ratingNum	long	
rate.url	keyword	

6.3.1.3 电影种子（Magnet 类）的索引设计

表 6-3-2 电影种子（Magnet 类）的索引设计

Properties	Type	Analyzer
title	text	ik_pinyin_analyzer
magnetId	keyword	
magnetTitle	keyword	
magnet	keyword	
size	keyword	
tags	keyword	

6.3.1.4 资讯（News 类）的索引设计

表 6-3-3 资讯（News 类）的索引设计

Properties	Type	Analyzer
title	text	ik_pinyin_analyzer
source	keyword	
author	keyword	
releaseTime	date	
content	text	

6.3.1.5 影评（Reviews 类）的索引设计

表 6-3-4 影评（Reviews 类）的索引设计

Properties	Type	Analyzer
title	text	ik_pinyin_analyzer
source	keyword	
author	keyword	
releaseTime	date	
content	text	

6.3.2 MySQL 数据库设计

6.3.2.1 电影类型

表 6-3-5 电影类型 MySQL 数据库设计

Field	Type	Allow Null
genre_id	int	No
genre_name	varchar(45)	Yes

6.3.2.2 电影

表 6-3-6 电影 MySQL 数据库设计

Field	Type	Allow Null
movie_id	int	No
movie_title	varchar(100)	Yes
movie_introduction	varchar(2000)	Yes
movie_rating	float	Yes
movie_release_date	varchar(45)	Yes

6.3.2.3 电影和电影类型的对应

表 6-3-7 电影和电影类型的对应 MySQL 数据库设计

Field	Type	Allow Null
genre_id	int	No
movie_id	int	No

6.3.2.4 电影人

表 6-3-8 电影人 MySQL 数据库设计

Field	Type	Allow Null
person_id	int	No
person_birth_day	varchar(45)	Yes
person_death_day	varchar(45)	Yes
person_name	varchar(45)	Yes
person_english_name	varchar(45)	Yes
person_biography	varchar(2000)	Yes
person_birth_place	varchar(45)	Yes

6.3.2.5 电影人和电影的对应

表 6-3-9 电影人和电影的对应 MySQL 数据库设计

Field	Type	Allow Null
person_id	int	No

movie_id	int	No

6.4 物理结构设计要点

1. 数据表项存储在数据库中，通过 SQL 语句访问数据库获取。
2. 文件资源存储在磁盘中，通过搜索引擎访问存储位置获取文件。

7 系统出错设计

7.1 出错信息

表 7-1-1 出错信息

输出信息形式	含义	处理方法
数据库连接失败	由于并发操作的用户数量很大，导致 ES 访问读写率降低；或者 ES 的节点配置不对，导致 ES 连接失败	修改 ES 节点配置，尝试重连
数据库账户信息泄露	后台服务器中的数据库被入侵，用户信息泄露	使用 sha1 和 md5 对密码进行双层加密
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、搜索引擎读取过程编码不一致	统一各处的编码方式
搜索结果列表为空	搜索引擎无法获得电影内容	手工检索

7.2 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。

8 系统维护设计

8.1 概述

1. 连接数据库时，需要在创建数据库连接、销毁数据库连接时使用 try catch 语句捕获异常，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。

8.2 检测点设计

8.2.1 搜索词条

- 关键字普通搜索
- 关键字与运算搜索
- 关键字或运算搜索
- 关键字非运算搜索
- 关键字混合逻辑运算搜索
- 关键词过滤高级搜索
- 关键词并集高级搜索
- 关键词交集高级搜索
- 搜索过程输入框智能补全
- 拼音搜索
- 搜索过程历史搜索记录显示

8.2.2 查看搜索结果

- 搜索结果页面显示
- 搜索结果类型筛选
- 电影详情页面跳转
- 搜索结果排序
- 电影搜索结果筛选
- 搜索条目点赞
- 搜索条目取消赞
- 个性化推荐内容显示
- 智能机器人基本问答

8.2.3 详情内容展示

- 电影结构化信息显示
- 数据聚合分析可视化显示
- 电影猜你喜欢推荐显示

8.3 相关维护设计

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。

数据库维护：定期备份数据库文件。

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。

9 模块设计计划

9.1 项目架构设计

NotOnlyFilm 电影垂直搜索引擎采用 B/S 结构，一共分为前端、搜索服务器、后端服务器、爬虫四个大模块。模块之间采用 JSON 进行数据传输。其中，前端使用 Vue 全家桶技术，后端使用 Spring Boot，数据持久化使用 Elastic Search、MySQL 以及 Fuseki，项目管理工具使用 cornerstone，代码托管平台使用 GitLab。

前端提供搜索引擎的图形化交互界面，合理、美观地呈现搜索结果以及内容展示；后端接收前端的请求，配置资源，查询底层数据库与搜索引擎信息，返回结构化数据；数据爬取端收集多维度的信息，持久化到数据库中，同时进行数据分析工作。

9.2 项目任务分解

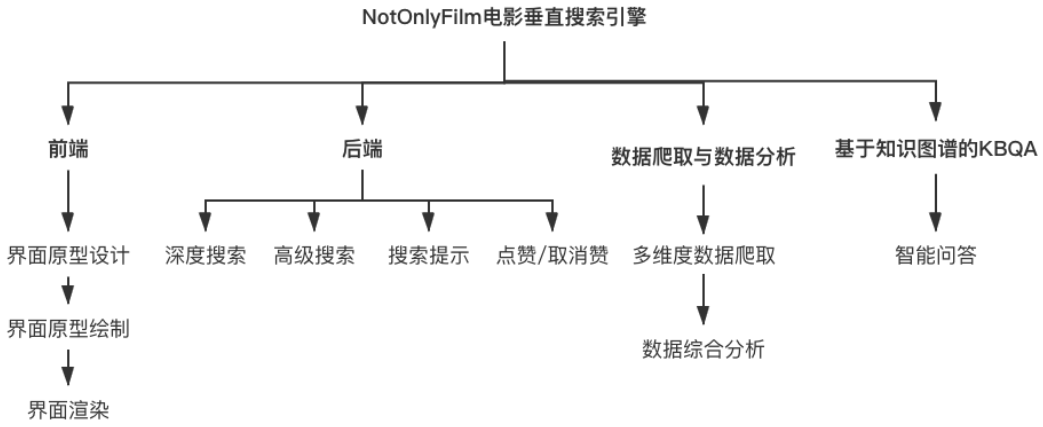

图 9-2-1 项目 WBS 划分图

9.3 前端

9.3.1 搜索首页模块

搜索页是网站的首页，为了提高用户体验，NOF（Not Only Film）采用了极

简主义设计，仅保留搜索框和必要的页头信息，让用户一目了然。另外，为了体现搜索引擎的主题以及给予用户沉浸式体验，背景图采用了深色的电影院座位图片，既美观大气，又契合主题。

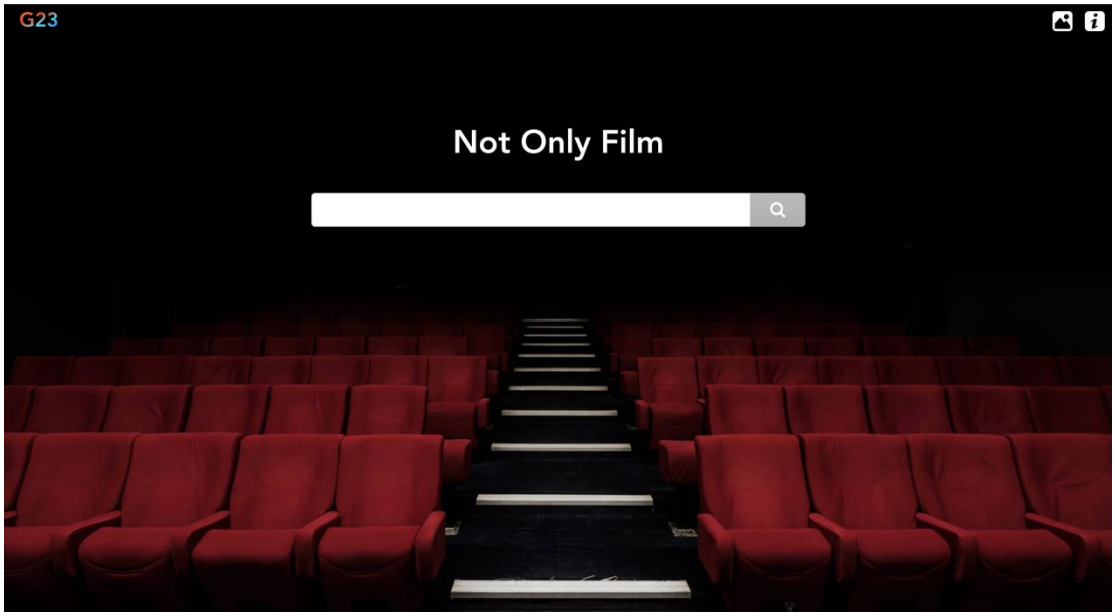

图 9-3-1 搜索引擎首页

如上图 9-3-1 所示，首页功能区设计有两个，页头、搜索框。页头包括左侧动态渐变色的 Logo 和右侧功能 Button。搜索框可以输入关键词和布尔操作符并带有智能提示自动补全功能。

后续考虑在功能区添加分类筛选搜索相关按钮，搜索框下部采用仿百度 UI 设计的个性化推荐电影以及电影资讯等。

9.3.2 搜索结果模块

搜索结果页的主要功能是展示搜索结果。除此之外，还要考虑高级的特性，比如支持筛选、支持排序、支持搜索。

据此，搜索结果页共设置四个功能区，分别是页头、搜索框、搜索结果展示侧、拓展功能区，如下图 9-3-2 所示。

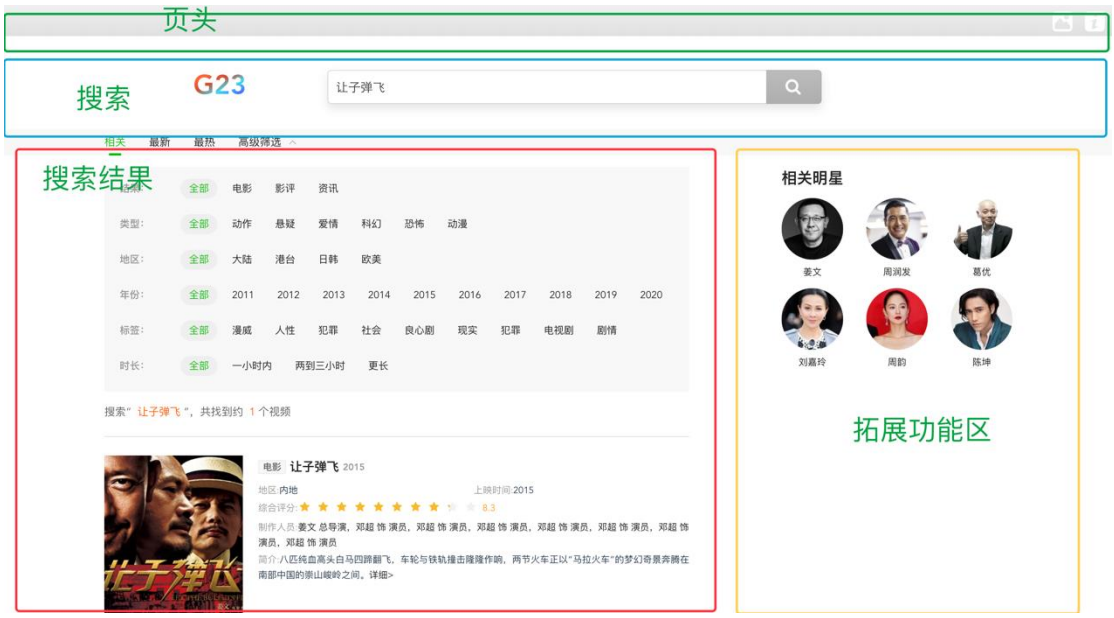

图 9-3-2 搜索结果页面

页头的左侧包括后续可拓展的功能性按钮，目前有一个帮助按钮和一个主题控制按钮。

搜索区包括渐变 Logo 和搜索框，搜索框支持历史记录与自动补全。

搜索结果区顶部是排序选项和一个可折叠的高级筛选面板。下面是分类展示的搜索结果，如下图 9-3-3 所示。

图 9-3-3 搜索结果分类展示

拓展功能区为拓展功能留白，后续可加入的拓展功能有关联的电影明星展示，

智能问答系统接入，搜索热榜展示等。

9.3.3 内容展示模块

内容展示的主要功能是展示电影的详细信息、相关资料和由 NOF 后台算法聚合分析的数据信息。

据此，界面分为三个功能区，页头、电影信息展示区、相关资料与数据内容展示区。

页头与信息展示区如下图 9-3-4 所示。页头含有渐变展示的主题 Logo 图标以及支持历史记录与智能补全的搜索框。信息展示区包括电影的海报、名称、别名、制片国家/地区、片长、上映时间、类型、导演、编剧、标签以及豆瓣和 IMDB 网站对于该电影的综合评分以及评分人数。

图 9-3-4 页头与信息展示区

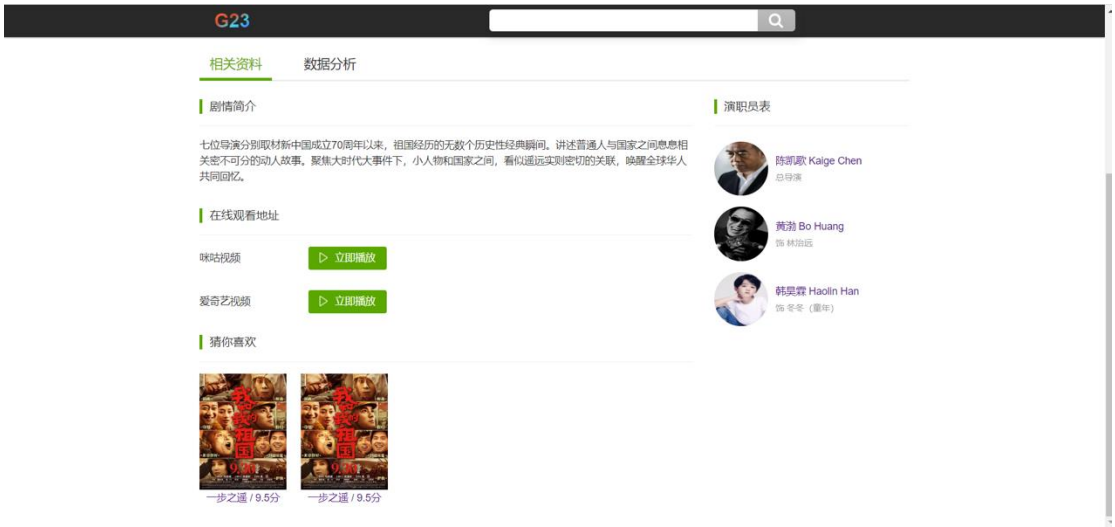

图 9-3-5 相关资料与数据展示区

相关资料与数据展示区如上图 9-3-5 所示，包括剧情简介，各大视频网站（咪

咕视频、爱奇艺视频、腾讯视频、哔哩哔哩、优酷视频、西瓜视频等）观看链接、演职员表以及智能化推荐的“猜你喜欢”电影。

如下图 9-3-6 所示，数据分析应以可视化图表的形式展现出来，包括但不限于同类电影的横纵向对比、影评的热度分析、相关的搜索指数以及关键词词云等。

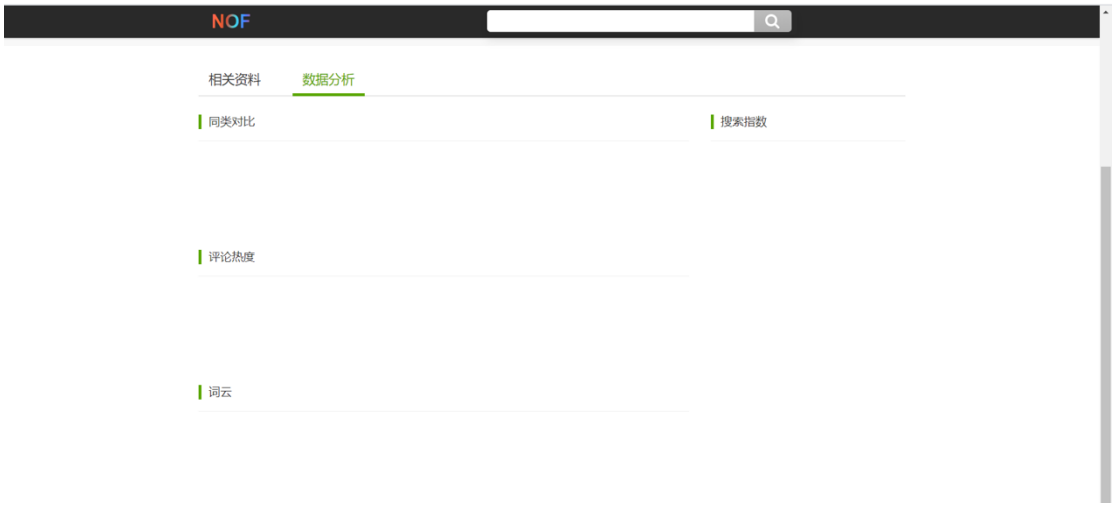

图 9-3-6 数据聚合分析展示区

9.4 搜索服务器

表 9-4-1 搜索服务器设计

功能点 ID	功能点名称	功能点描述	负责人	截止时间
ES-01	数据导入	向后端服务器提供导入数据的 HTTP API 向管理员提供直接可执行的脚本和 JSON 模板配置文件	应承峻	7.8
ES-02	数据搜索	向后端服务器提供搜索的 HTTP API	应承峻	7.8
ES-03	索引建立	向后端服务器提供索引建立的 HTTP API 向管理员提供直接可执行的脚本和 JSON 模板配置文件	应承峻	7.8
ES-04	数据更新	向后端服务器提供搜索的 HTTP API	应承峻	7.8

9.5 后端服务器

表 9-5-1 后端服务器设计

功能点 ID	功能点名称	功能点描述	负责人	截止时间
BE-01	信息爬取	系统在服务用户的过程中一般情况下不会实时爬取来自网络的数据，而是通过读取数据库中经过提炼和整合的数据来更好得服务用户。为了保证系统的信息具有一定的实时性，管理员必须隔一段时间爬取一次来自网络的各种信息。	应承峻	7.10
BE-02	信息过滤	在管理人员爬取大量网络文本资源后，需要对一些相似性很高的文本进行过滤重复或者是合并，由于网络上的资源有很多重复的内容，给用户提供多个重复内容的情况在各类搜索引擎中并不少见，因此需要对文本数据进行过滤。	应承峻	7.10
BE-03	数据搜索	向用户提供数据搜索接口，包括关键词查询、搜索智能提示、命中高亮等功能。	应承峻	7.10
BE-04	数据管理	向管理员提供一系列的接口供管理员创建索引、导入数据。	应承峻	7.10

9.6 数据模块设计

9.6.1 信息爬取

为了满足用户对于信息资料的需求，需要在网络上爬取满足该垂直领域的所有资源。为了保证系统的高响应性，采用提前爬取而非实时爬取的方式，将相关数据经过提炼和整合，根据其应用范围的不同存储入搜索引擎、数据库中。

9.6.2 定期爬取

为了保证搜索引擎的时效性，需要定期对数据资料进行更新、追加。对于时效性极高的搜索指数等数据，系统通过定时脚本以天为单位启动相应爬虫实现定时爬取的目的。对于时效性要求较低且信息更新速率低的电影基本信息等数据，可由系统管理员以周或月为单位手动操作，以减少对于网络的负担。

9.6.3 唯一标志

数据资料需要在系统中有唯一的标志来进行区分，系统采用数据来源网站简写+资源在该网站中的 ID 予以标志，并将电影 ID、演员 ID 等特定的数据存入数据库中，以指导下一步的资源爬取以及在定期爬取中去重。

9.6.4 数据分析

基于垂直搜索引擎中的深度分析需求，系统提供了电影维度的猜你喜欢、影评词云、近期热度、历史评论热度、同类对比等功能。在爬取完数据分析所需的资料后将其存入数据库中，启动脚本对于采集到的数据进行再加工，将统计分析后的数据、生成的词云等存入数据库中，当用户访问时返回深度分析后的数据。