

专利搜索引擎系统

项目计划书

组长：刘昕宇

组员：焦洋、梁文杰、姚张宸、赵佳乐（字典序）

日期：2024年10月20日

版本：Version 3.0

修改历史

修订日期	版本号	修改人	修改内容	审核人
2024-10-13	1.0	赵佳乐， 姚张宸， 焦洋， 刘昕宇， 梁文杰	终稿	刘昕宇

修改历史

1 引言

1.1 项目背景

1.2 编写目的

1.3 软件系统名称

1.4 相关人员及项目支持

1.4.1 任务提出者

1.4.2 项目开发者

1.4.3 用户

1.4.4 实现该软件的计算机网络

1.5 读者对象

2 项目概述

2.1 前言介绍

2.2 项目介绍

2.3 名词介绍

2.3.1 HTML

2.3.2 CSS

2.3.3 JAVASCRIPT

2.3.4 SPRING BOOT

2.3.5 ELASTIC SEARCH

2.3.6 Vue

2.3.7 MySQL

2.3.8 Nginx

2.3.9 Redis

2.4 产品成果

2.4.1 产品

2.4.2 手册

2.4.3 服务

3 成本估计

3.1 硬件成本

3.2 其他成本

4 开发技术分析

4.1 开发技术栈

4.2 技术分析

4.2.1 前端技术栈分析

4.2.2 后端技术栈分析

4.2.3 搜索引擎服务框架分析

4.2.4 数据库分析

4.2.5 代码管理和协作开发工具分析

4.2.6 文档管理分析

4.2.7 代码规范分析

4.2.8 测试工具分析

5 开发进度计划

6 角色分配

7 软硬件资源说明

7.1 服务器配置要求

7.2 客户端配置要求

7.3 软件依赖

8 专题计划要点

8.1 开发人员培训计划

8.2 质量保障计划

8.3 测试计划

8.3.1 静态测试

8.3.2 单元测试

8.3.3 系统测试

9 参考文献

1.1 项目背景

随着全球科技创新步伐的加快，专利数量迅猛增长，专利信息已经成为企业和科研机构进行技术开发、市场竞争与知识产权保护的重要资源。然而，海量的专利信息给检索、分析和使用带来了巨大挑战。现有的通用搜索引擎在面对专利信息的专业性、结构化和多语言需求时，往往无法提供精确、高效的检索体验。

专利搜索引擎系统专注于提供专利文献的搜索、过滤和分析服务，致力于解决专利信息检索中的痛点，基于用户提供的数据进行需求分析，帮助用户快速定位、获取相关专利信息。这一搜索引擎将结合自然语言处理（NLP）、机器学习（ML）等技术，构建一个专业的、高效的专利信息检索平台。

1.2 编写目的

本项目章程旨在说明整个专利搜索引擎项目实施过程中的开发技术栈、开发进度安排、角色分配、软硬件资源、测试约定、工作规范、工作要点等内容，从而对开发的内容和进度进行管控，更好地推动项目的进展。另外，通过项目管理策略与方法，说明如何计划、组织和实施项目开发工作。

1.3 软件系统名称

专利搜索引擎

1.4 相关人员及项目支持

1.4.1 任务提出者

浙江大学软件工程管理课程组：

- 任课教师邵健

1.4.2 项目开发者

浙江大学软件工程管理第二小组

组长：刘昕宇

组员：赵佳乐，姚张宸，焦洋，梁文杰

1.4.3 用户

所有对专利数据搜索有需求的潜在用户。

1.4.4 实现该软件的计算机网络

基于公网IP的服务器和数据库。

1.5 读者对象

- ①软件客户代表
- ②项目经理
- ③产品经理
- ④项目开发人员
- ⑤项目测试人员
- ⑥项目监督管理和质量保证人员

2 项目概述

2.1 前言介绍

垂直搜索引擎是针对某一个行业的专业搜索引擎，是搜索引擎的细分和延伸，是对网页库中的某类专门的信息进行一次整合，定向分字段抽取出需要的数据进行处理后再以某种形式返回给用户。垂直搜索是相对通用搜索引擎的信息量大、查询不准确、深度不够等提出来的新的搜索引擎服务模式，通过针对某一特定领域、某一特定人群或某一特定需求提供的有一定价值的信息和相关服务。其特点是“专、精、深”，且具有行业色彩，相比较通用搜索引擎的海量信息无序化，垂直搜索引擎则显得更加专注、具体和深入。

选择本项目的理由是希望能够通过实现专利搜索引擎项目，展现专门针对专利领域的精准和专业的搜索结果，依据课堂上的AI+产业合作，设计了AI专利匹配工具。

2.2 项目介绍

本项目旨在提供一个专门针对专利的垂直搜索引擎平台，为用户提供准确的专利搜索服务，允许用户在特定领域内进行专利检索，并利用AI工具实现专利匹配。项目的功能将包括专利搜索和领域匹配。用户可以通过输入关键词、专利号或其他相关信息来进行专利检索。此外，平台将利用AI工具实现专利匹配，允许用户在特定领域内进行专利研究方向的匹配。

用户将从项目中获得准确的专利搜索结果，有助于他们在特定领域内获取所需的专利信息。通过AI工具实现的专利匹配将帮助用户快速找到与其研究方向相关的专利，促进跨领域合作和创新。

2.3 名词介绍

2.3.1 HTML

HTML的英文全称是 Hyper Text Markup Language，即超文本标记语言。HTML是由Web的发明者 Tim Berners-Lee和同事 Daniel W. Connolly于 1990年创立的一种标记语言，它是标准通用化标记语言 SGML的应用。用HTML编写的超文本文档称为HTML文档，它能独立于各种操作系统平台(如UNIX，Windows等)。使用HTML，将所需要表达的信息按某种规则写成HTML文件，通过专用的浏览器来识别，并将这些HTML文件“翻译”成可以识别的信息，即现在所见到的网页。

2.3.2 CSS

层叠样式表(英文全称：Cascading Style Sheets)是一种用来表现HTML（标准通用标记语言的一个应用）或XML（标准通用标记语言的一个子集）等文件样式的计算机语言。CSS不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。

2.3.3 JAVASCRIPT

JavaScript（简称“JS”）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。虽然它是作为开发Web页面的脚本语言而出名，但是它也被用到了很多非浏览器环境中，JavaScript 基于原型编程、多范式的动态脚本语言，并且支持面向对象、命令式、声明式、函数式编程范式。

2.3.4 SPRING BOOT

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Spring Boot致力于在蓬勃发展的快速应用开发领域(rapid application development)成为领导者。

2.3.5 ELASTIC SEARCH

Elastic Search 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。Elastic Search 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。

2.3.6 Vue

Vue.js（简称Vue）是一个用于构建用户界面的渐进式JavaScript框架，通常用于开发单页应用（SPA，Single Page Application）。它的设计灵活、易上手，核心库只关注视图层，但可以通过生态系统或集成其它库来开发复杂的应用程序。

2.3.7 MySQL

MySQL是一个基于关系模型的数据库管理系统，使用C和C++编写。旨在为各种应用提供可靠的数据存储和管理解决方案。MySQL是一个典型的关系型数据库管理系统，它采用结构化查询语言来管理数据。MySQL使用严格的表结构和关联来组织数据，使得它更适合于需要严格数据一致性和复杂查询的应用场景。MySQL支持的数据类型丰富，查询语言强大且灵活，采用类似面向对象的结构化查询语言，能

够实现复杂的数据检索和操作，同时支持广泛的查询优化和索引功能。

2.3.8 Nginx

Nginx是一个高性能的开源Web服务器，最初由Igor Sysoev创建于2004年。Nginx以其卓越的性能和稳定性而闻名，常用作反向代理服务器、负载均衡器和HTTP缓存，适用于处理高流量网站。Nginx还可以作为一个通用的反向代理服务器，帮助提高网站的安全性和性能。Nginx支持模块化的架构，允许开发者根据需求扩展其功能，使其成为一个灵活且强大的工具。

2.3.9 Redis

Redis是一个开源的内存数据库管理系统，于2009年由Salvatore Sanfilippo创建。它以其快速、灵活和可扩展性而闻名，常用作缓存、消息代理和数据存储。Redis支持多种数据结构，包括字符串、哈希、列表、集合和有序集合，使其适用于各种不同的用例。由于其高性能和灵活性，Redis已成为许多Web应用程序和分布式系统中常用的关键组件之一。

2.4 产品成果

2.4.1 产品

最终成果是AI辅助专利搜索匹配引擎，所有产品包括：

- 前后端代码
- 数据库建立代码
- 系统开发、维护文档
- 网站所需资源文件
- AI专利匹配工具

2.4.2 手册

1. 验收报告以及交接手册
2. 用户操作手册

提供给使用此搜索引擎网站的用户。对于系统的界面、功能以及操作说明进行描述，使用户对于网站的操作可以快速上手。

3. 应用配置手册

提供给系统管理员及系统维护人员进行阅读，对于网站的部署方法进行说明，对于相关参数的用途进行说明，使用有一定技术的用户得以自己解决多数问题。

4. 软件维护手册

主要包括搜索引擎网站系统说明、程序模块说明、操作环境、支持软件的说明、维护过程的说明，便于维护人员对于系统的维护。

2.4.3 服务

提供使用教程以及咨询服务，使解决用户在使用网站过程中的困难。提供后续一定年限的维护服务，当软件出现问题时提供技术支持。

3 成本估计

3.1 硬件成本

资源说明	级别	详细配置	获取方式	使用说明
开发应用服务器	普通	CPU&内存：2核4G 带宽：5Mbps SSD系统盘：80G	阿里云购买	部署项目时使用
数据库存储	普通	CPU&内存：1核2G SSD系统盘：20G 存储引擎：InnoDB	阿里云购买	项目开发及部署中的数据库
域名服务	普通	无	NameSilo购买	部署项目时使用

3.2 其他成本

资源说明	级别	详细配置	获取方式	使用说明
Visual Studio Code, IntelliJ IDEA	关键	符合主流版本要求	官网下载	开发阶段使用的IDE
Git	关键	版本：2.41.0及以上	官网下载	代码托管&版本管理
Edge、Google等浏览器	普通	无	官网下载	前端效果测试以及最终展示

4 开发技术分析

4.1 开发技术栈

前端技术栈：Vue3 + Element-Plus + Apache ECharts

后端技术栈：Spring Boot + Ngnix + Redis + Mybatis-Plus

搜索引擎服务框架：Elastic Search

数据库：MySQL

代码管理和协作开发工具：GitHub

文档管理：钉钉 + 语雀

代码规范：Eslint + Prettier

测试工具：ApiFox

4.2 技术分析

4.2.1 前端技术栈分析

- **Vue3:** Vue3 作为前端框架，提供了更高效的响应式系统、更好的 TypeScript 支持以及更小的包体积。其组合式 API 使得代码组织更加灵活，便于维护和扩展。
- **Element-Plus:** Element-Plus 是基于 Vue3 的 UI 组件库，提供了丰富的组件和样式，能够快速构建美观且功能强大的用户界面。其组件库的成熟度和社区支持使得开发效率大大提升。
- **Apache ECharts:** ECharts 是一个强大的数据可视化库，支持丰富的图表类型和交互功能。结合 Vue3，可以轻松实现数据的可视化展示，提升用户体验。

4.2.2 后端技术栈分析

- **Spring Boot:** Spring Boot 是一个快速开发框架，简化了 Spring 应用的配置和部署。其自动配置和约定优于配置的理念，使得开发者可以专注于业务逻辑的实现，而不必过多关注底层配置。
- **Nginx:** Nginx 作为反向代理服务器，提供了高性能的负载均衡和静态资源服务。通过 Nginx，可以有效提升系统的并发处理能力和稳定性。
- **Redis:** Redis 是一个高性能的内存数据库，适用于缓存、会话管理等场景。通过 Redis，可以显著提升系统的响应速度和并发处理能力。
- **Mybatis-Plus:** Mybatis-Plus 是 MyBatis 的增强工具，提供了代码生成、分页查询、自动填充等功能。其简化了对数据库的操作，减少了重复代码的编写。

4.2.3 搜索引擎服务框架分析

- **Elastic Search:** Elastic Search 是一个分布式搜索引擎，提供了强大的全文搜索和数据分析功能。通过 Elastic Search，可以实现高效的数据检索和分析，满足复杂查询需求。

4.2.4 数据库分析

- **MySQL:** MySQL 是一个广泛使用的关系型数据库，具有良好的性能和稳定性。通过 MySQL，可以存储和管理结构化数据，满足业务需求。

4.2.5 代码管理和协作开发工具分析

- **GitHub:** GitHub 是一个代码托管平台，提供了版本控制、协作开发、代码审查等功能。通过 GitHub，团队成员可以高效地进行代码管理和协作开发。

4.2.6 文档管理分析

- **钉钉 + 语雀:** 钉钉作为即时通讯工具，提供了高效的沟通和协作功能。语雀作为文档管理工具，提供了结构化的文档编辑和分享功能。通过钉钉和语雀的结合，可以实现高效的文档管理和团队协作。

4.2.7 代码规范分析

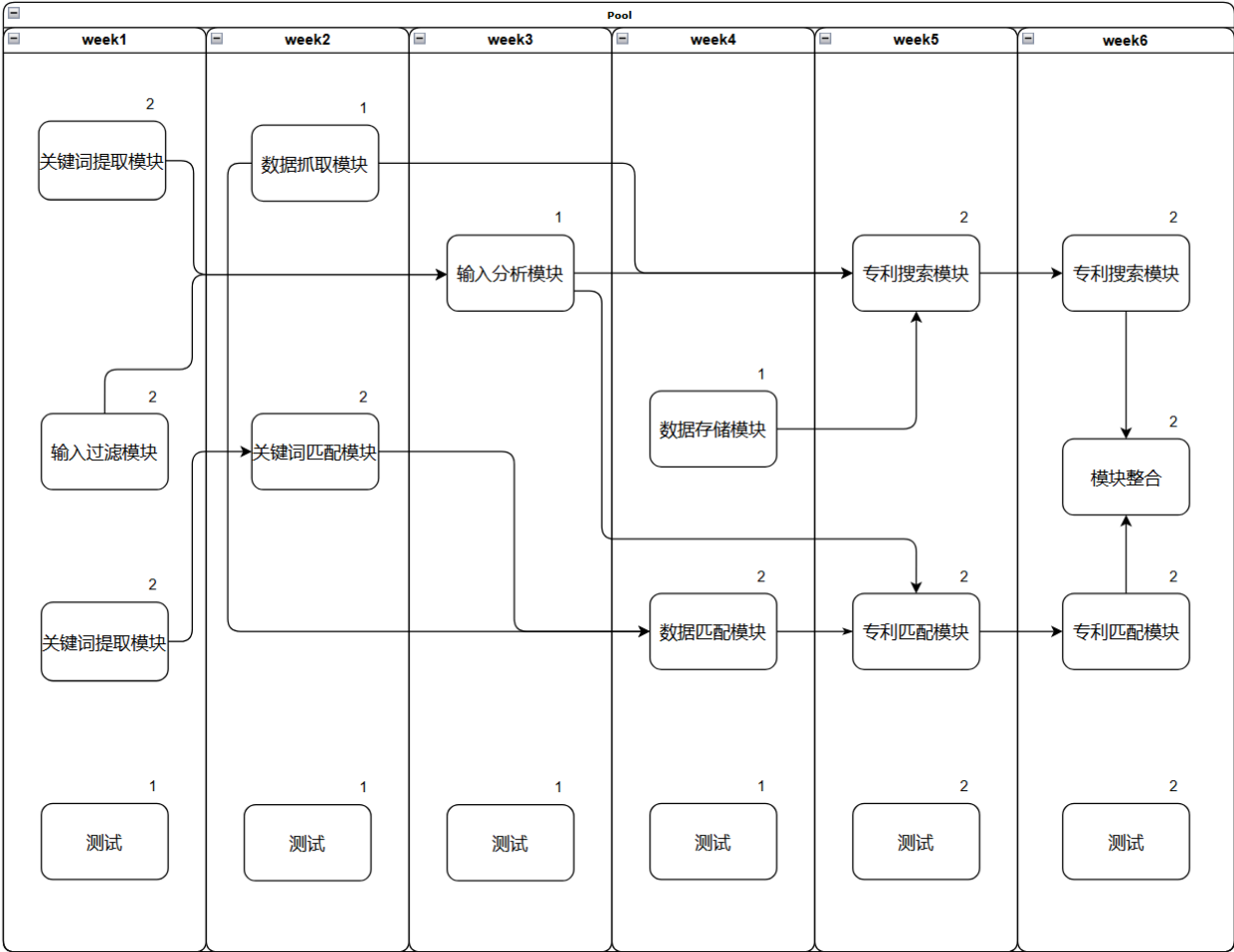
- **Eslint + Prettier:** Eslint 是一个代码检查工具，用于发现和修复代码中的潜在问题。Prettier 是一个代码格式化工具，用于统一代码风格。通过 Eslint 和 Prettier，可以提升代码质量和可维护性。

4.2.8 测试工具分析

- **ApiFox:** ApiFox 是一个 API 测试工具，提供了接口测试、性能测试、自动化测试等功能。通过 ApiFox，可以确保 API 的正确性和稳定性，提升系统的可靠性。

5 开发进度计划

项目阶段	持续时间	负责人	主要工作	输出内容
项目启动	10.16-10.21	刘昕宇	制定项目计划和项目章程，进行可行性分析	完成《项目计划书》和《项目章程》等文档
需求分析	10.22-10.29	赵佳乐	进行项目的需求分析，确定系统的功能和基本的业务逻辑模型	完成《需求规格说明书》等需求分析性文档
系统设计	10.30-11.6	梁文杰	设计系统的总体架构和具体细节，如数据库表等	完成《系统设计计划》等设计性文档
编程实现	11.7-11.14	全体成员	完成系统的源代码开发等工作	实现并部署搜索引擎
系统测试	11.15-11.22	全体成员	完成系统测试等工作	完成《测试报告》等报告
项目总结	11.23	全体成员	项目总结汇报	完成个人部分和最终答辩



W1	W2	W3	W4	W5	W6
关键词提取模块（搜索）					
输入过滤模块					
关键词提取模块（匹配）					
	数据抓取模块				
	关键词匹配模块				
		输入分析模块			
			数据存储模块		
			数据匹配模块		
				专利搜索模块	
				专利匹配模块	
测试					

6 角色分配

编号	角色	角色职责	具体负责人
1	项目经理	在预算范围内领导小组成员完成所有的工作	刘昕宇
2	产品经理	负责进行市场调研，了解和设计具体的用户需求，并领导制定本项目的需求规格说明书	赵佳乐

编号	角色	角色职责	具体负责人
3	设计总监	建立系统框架，负责系统的总体架构和具体模块的设计，参加技术评审	梁文杰
4	测试团队经理	负责领导测试团队完成整个项目的测试	焦洋
5	美术监督	设计网站原型，并负责维持网站的美术质量	姚张宸
6	质量经理	带领软件质量监督巡视组成员制定质量保证计划等文档，并对实际的文档和代码的质量进行监督评审	刘昕宇
7	开发团队	负责完成搜索引擎的具体开发和部署工作	全体成员
8	测试团队	负责完成整个系统的测试和维护	全体成员
9	软件质量监督师	负责实时对质量经理以及项目经理提供的项目进度和项目实际开发时的差异提出报告并指出改进的办法	全体成员

7 软硬件资源说明

7.1 服务器配置要求

资源说明	级别	详细配置	获取方式	使用说明
开发应用服务器	关键	CPU&内存：2核4G 带宽：5Mbps SSD系统盘：80G	阿里云购买	部署项目时使用
数据库存储	普通	CPU&内存：1核2G SSD系统盘：20G 存储引擎：InnoDB	阿里云购买	项目开发及部署中的数据库

7.2 客户端配置要求

可以连接互联网的具有浏览器的个人PC

资源说明	级别	详细配置	获取方式	使用说明
客户端设备	普通	CPU&内存：2核4G 带宽：5Mbps	自行配置	用户访问和使用系统

7.3 软件依赖

- **Visual Studio Code**: 前端代码开发工具，主要用于实现 **Vue3** 框架下的前端代码的开发。
- **IntelliJ IDEA**: 后端开发工具，主要用于实现 **springboot** 框架下的后端代码开发。
- **Git**: 代码托管以及版本管理的工具，主要用于实现组内代码进度的统一以及不同版本代码的管理。
- **不同浏览器**: 前端页面渲染工具，主要用于查看开发过程中的前端展示页面，并且提供后续的用户交互页面。

8 专题计划要点

8.1 开发人员培训计划

培训内容

1. **Vue3**前端框架及**Element-Plus**组件库使用培训，以及部分辅助框架(如**Apache ECharts**)的使用培训。
2. 数据库基础，主要为**MySQL**编程语言培训，编程工具的培训。
3. 数据索引与搜索引擎基础，主要为 **Elasticsearch** 使用的培训。以及**SpringBoot**前后端链接框架搭建，**Mybatis-Plus**使用培训。针对需求，进一步加强软件开发规范和软件开发过程质量，了解客户的真正需求，提高开发人员开发素质，使开发人员成为高质量的软件工程师。结合项目进行案例讨论，根据讨论过程进行评估，通过上述培训，进一步提高开发人员技术水平，扩展开发人员的技术知识和视野。

8.2 质量保障计划

保证工作项目的质量，保证工作由质量经理、质量保证人员及项目负责人负责，质量保证人员在项目启动后制定适用于单个项目的《质量保证计划》，并检查和督促计划的实施。应该按照质量保证计划的规定进行项目的生存周期的各项活动，加强各阶段的早期评审和检查工作，质量保证团队的人员应该参加所有的评审和检查活动。

在项目的生存周期过程中，各阶段的负责人应该按照项目的进展情况完成对应的技术文档和项目日志，还需要按规范以及其他相关文件的要求填写相应记录，质量保证组通过检查各阶段输出的文档和日志记录检查发现质量问题。

8.3 测试计划

8.3.1 静态测试

静态测试是指不执行程序而找出程序存在的错误，这种方法以人工的、非形式化的方法对程序进行分析和测试，不以来程序运行的测试。在测试中，找出程序中的语法错误修改源程序的语法和逻辑错误，进一步完善程序功能，保证代码实现目标和最初的功能需求一致。这一阶段，我们还要规范程序的格式，添加必要的注释，使程序结构变的清晰、层次分明，以便其他开发人员的理解、开发、维护和修改。

8.3.2 单元测试

单元测试是指对单个子程序、过程的逻辑测试，我们的任务是检测整体功能和接口说明是否有不符合规定的情况，以及代码是否出错，边界条件正确性，错误处理等测试。经过静态测试之后，要注意测试每一个单元，在输入输出的测试案例上进行调整。

8.3.3 系统测试

经过单元测试之后，需要在单元测试的基础上把每个单元按照设计要求逐步建立起来，进行整个专利搜索系统的集成测试，并且进行接口的数据类型的匹配，验证系统是否满足用户的功能需求和非功能需求。对整个系统进行性能测试和安全性测试，保证整个系统的安全可靠。

9 参考文献

- [1] P. DuBois, MySQL Cookbook: Solutions for Database Developers and Administrators, O'Reilly Media, 2014.
- [2] Nginx HTTP Server by Clément Nedelcu, Packt Publishing, 2016
- [3] Mastering Redis by Jeremy Nelson, Packt Publishing, 2016