



项目计划书

垂直搜索引擎

组长：马梓睿

组员：程梦烨 毕予然 沈乐明 李楠

目录

| | | |
|-------|-------------------|---|
| 1 | 引言 | 4 |
| 1.1 | 编写目的 | 4 |
| 1.2 | 相关人员及项目支持 | 4 |
| 1.2.1 | 任务提出者 | 4 |
| 1.2.2 | 开发者 | 4 |
| 1.2.3 | 用户 | 4 |
| 1.2.4 | 实现该软件的计算机网络 | 4 |
| 1.3 | 读者对象 | 4 |
| 2 | 项目概述 | 4 |
| 2.1 | 项目背景 | 4 |
| 2.2 | 项目介绍 | 5 |
| 2.3 | 名词定义 | 5 |
| 2.4 | 参考资料 | 5 |
| 2.5 | 产品成果 | 5 |
| 2.5.1 | 产品 | 5 |
| 2.5.2 | 手册 | 5 |
| 2.5.3 | 服务 | 6 |
| 3 | 开发进度计划 | 6 |
| 3.1 | 小组成员特长分析 | 6 |
| 3.2 | 进度计划 | 6 |
| 4 | 角色分配 | 7 |
| 5 | 软硬件资源说明 | 8 |
| 6 | 专题计划要点 | 8 |
| 6.1 | 开发人员培训计划 | 8 |
| 6.1.1 | 培训目的 | 8 |
| 6.1.2 | 培训方法 | 8 |
| 6.1.3 | 培训目标 | 9 |
| 6.1.4 | 培训内容 | 9 |
| 6.1.5 | 时间计划 | 9 |

| | | |
|-------|--------------|----|
| 6.2 | 质量保证计划 | 9 |
| 6.3 | 测试计划 | 9 |
| 6.3.1 | 静态测试 | 9 |
| 6.3.2 | 单元测试 | 10 |
| 6.3.3 | 系统测试 | 11 |
| 6.4 | 维护计划 | 12 |
| 6.4.1 | 维护类别 | 12 |
| 6.4.2 | 维护工作安排 | 12 |
| 6.4.3 | 巡检计划 | 13 |

1 引言

1.1 编写目的

此项目计划书的编写主要是为了给开发《垂直搜索引擎》做主要的规划和整合，在开发过程中起到引导作用，保证项目团队按时保质地完成项目目标，便于项目团队成员更好地了解项目情况，使项目工作开展的各个过程合理有序，以文件化的形式，把对于在项目生存周期内的工作任务范围、各项工作的任务分解、项目团队组织结构、各团队成员的工作责任、团队内外沟通协作方式、开发进度、经费预算、项目内外环境条件、风险对策等内容做出的安排以书面的方式，作为项目团队成员以及项目干系人之间的共识与约定，它又是项目生命周期内的所有项目活动的行动基础、项目团队开展和检查项目工作的依据。

1.2 相关人员及项目支持

1.2.1 任务提出者

浙江大学《项目实训》任课老师：邵健

1.2.2 开发者

浙江大学《项目实训》课程小组：马梓睿 程梦烨 毕予然 沈乐明 李楠

1.2.3 用户

广大商家

1.2.4 实现该软件的计算机网络

基于公网 IP 的服务器与数据库

1.3 读者对象

- 软件客户代表
- 项目经理项目
- 开发人员
- 软件质量分析员
- 软件维护人员
- 其它需要了解本项目质量管理情况的人员都可参阅

2 项目概述

2.1 项目背景

垂直搜索引擎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。通过针对某一特定需求、某一特定人群或某一特定领域提供的有一定价值的信息和相关服务。

商品购物是消费者的日常需要。本项目旨在开发一个商品购物的搜索引擎，为消费者提供从多种在线零售网站中进行商品价格、网站信誉、购物方便性等方面的比较资料。其作用不仅表现在为在线消费者提供方便，也为在线销售上推广产品提供了机会，获得的信息更加集中更全面。该购物搜索引擎的检索结果来自于网上购物网站，如淘宝、京东、亚

马逊等，这样当用户检索某个商品时，所有销售该商品的网站上的产品记录都会被检索出来，用户可以根据产品价格、对网站的信任和偏好等因素进入所选择的网上购物网站购买产品，而一般来说，购物搜索引擎本身并不出售这些商品。

2.2 项目介绍

本项目提供了一个基于购物场景下的通用垂直搜索引擎，通过对前期数据的爬取、对数据的结构化提取设计以及知识图谱的构建，系统建立了一个较为完整的知识图谱。用户可以利用本系统，对某一领域的商品环境建立起较为完整的知识图谱数据库，为其了解当前形势、协助商业决策提供了一个有力的工具。

2.3 名词定义

- 垂直搜索是指针对特定行业互联网网页中特定信息内容的搜索。这种搜索是针对某一个行业、领域、人群或特定需求的专业搜索引擎。它对网页库中的某类专门信息进行整合，定向、分字段抽取出需要的数据，处理后再以某种形式返回给用户。
- 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 是指系统浏览器/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.js 是一个用于创建用户界面的开源 JavaScript 框架，也是一个创建单页应用的 Web 应用框架。2016 年一项针对 JavaScript 的调查表明，Vue 有着 89% 的开发者满意度。

2.4 参考资料

- 《软件设计文档国家标准》
- 《软件工程项目开发文档范例》
- 《软件需求（第三版）》

2.5 产品成果

2.5.1 产品

最终提交的是一个经过测试符合客户需求，具有较强可用性、鲁棒性、安全性的垂直搜索引擎平台。包括：

- 前后端代码
- 数据库构建代码
- 系统开发、维护文档
- 所需资源文件

2.5.2 手册

- 验收报告及交接手册
- 用户使用手册，包含面向普通用户的使用说明部分和面向专业用户的网站添加部分

- 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。
- 软件维护手册，主要包括系统说明、程序模块说明、操作环境、支持软件的说明、维护过程的说明，便于维护人员对系统进行维护。

2.5.3 服务

提供使用教程、培训以及咨询服务，协助用户解决使用网站过程中的问题。提供一定年限的维护服务以及一定范围的升级服务。

3 开发进度计划

3.1 小组成员特长分析

- 马梓睿：对软件整体设计和架构规划较为了解，擅长自动化部署和 DDP 等编程模型。
- 沈乐明：对于软件整体的架构、开发流程等较为熟悉，擅长后端代码的编写。
- 毕予然：对于数据整合分析、selenium 库、测试过程比较熟悉，擅长爬虫、测试开发等。
- 程梦烨：对于软件的功能和原型总体设计等较为熟悉，擅长前端代码的编写。
- 李楠：对软件开发的工程化流程与生命周期有一定认知，擅长用户界面设计。

3.2 进度计划

表 3-1 进度计划表

| 项目阶段 | 持续时间 | 负责人 | 主要工作 | 输出内容 |
|------|-----------------------|-----|-----------------------------|-------------------|
| 项目启动 | 2021.07.09 | 马梓睿 | 进行项目可行性分析，制定项目计划 | 完成《项目计划》《项目章程》等 |
| 需求分析 | 2021.07.09-2021.07.12 | 沈乐明 | 确定系统运行环境，确定系统功能及性能，建立系统逻辑模型 | 完成《软件需求规格说明书》 |
| 系统设计 | 2021.07.13-2021.07.15 | 马梓睿 | 进行系统设计 | 完成《系统设计报告》 |
| 编程实现 | 2021.07.16-2021.07.18 | 程梦烨 | 进行系统编码 | 实现并部署网站，完成《用户手册》 |
| 需求维护 | 2021.07.19-2021.07.20 | 毕予然 | 进行需求变更控制 | 更新《软件需求规格说明书》 |
| 系统测试 | 2021.07.21-2021.07.22 | 李楠 | 进行系统测试，项目总结 | 完成《测试报告》与《项目总结报告》 |

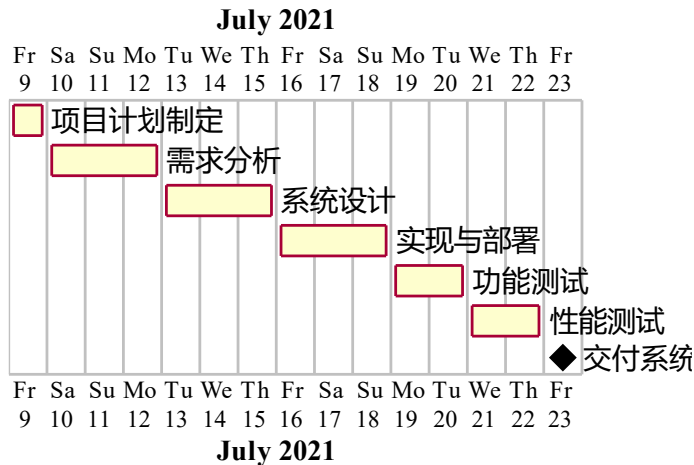


图 3-1 进度计划甘特图

4 角色分配

表 4-1 角色分配表

| 角色 | 职责 | 人员 |
|--------|--|------|
| 项目经理 | 在预算范围内按时优质地领导项目小组完成全部项目工作内容，并使客户满意。 | 马梓睿 |
| 产品经理 | 负责市场调查并根据用户的需求，确定开发何种产品，选择何种技术、商业模式等，根据产品的生命周期，协调研发、营销、运营等，确定和组织实施相应的产品策略。 | 马梓睿 |
| 设计总监 | 建立系统框架；数据库设计；概要设计；参加技术评审； | 程梦烨 |
| 测试经理 | 组织编写测试计划和测试方案，组织系统测试；参加技术评审； | 毕予然 |
| 美工 | 设计网站原型 | 程梦烨 |
| 质量经理 | 带领软件质量监督组成员制定质量保证计划，对监督组反映的质量问题进行汇总与产品经理、项目经理进行交流，当新的问题出现时最终由质量经理决定处理方式。 | 沈乐明 |
| 开发人员 | 负责进行编码工作与单元测试，进行系统集成，及时解决测试时出现的问题 | 全体成员 |
| 测试人员 | 编写测试方案和测试用例，进行系统测试，向开发组反馈 BUG。 | 全体成员 |
| 软件质量监督 | 实时对质量经理以及项目经理提供项目进度与项目实际开发时的差异提出报告，指出差异原因和改进方法。 | 李楠 |

5 软硬件资源说明

表 5-1 项目软硬件资源说明

| 软硬件资源名称 | 级别 | 详细配置 | 获取时间或方式 | 使用说明 |
|---|----|---|---------|---|
| 服务器 | 关键 | CPU: 4 核 内存: 8G OS: Ubuntu 20.04 带宽: 5Mbps | 购买阿里云服务 | 开发中期和最终部署阶段使用 |
| Visual Studio Code, PyCharm | 关键 | 配置相关语言编译器或解释器插件 | 官网下载 | 开发使用的代码编辑器 |
| Vue.js, Django, SQL, CSS, HTML | 关键 | 根据语言语法使用规范 | 无 | 建立前后端分离开发机制, 前端使用 Vue.js 框架, 后端使用 Python 下的 Django 框架, 后端与数据库的交互通过 SQL 语句实现 |
| Git | 关键 | 版本: 2.28.0 | 已经存在 | 配置管理以及代码托管 |
| Microsoft Edge Google Chrome Safari | 关键 | 版本: 44.18362.449.0, 83.0.4103.97, 13.0 | 已经存在 | 通过浏览器测试网页端显示效果与功能实现 |

6 专题计划要点

6.1 开发人员培训计划

6.1.1 培训目的

让用户了解系统使用方式和高级功能。

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面和视野。

6.1.2 培训方法

现场项目培训采用 PPT 讲解和实际操作同时进行的方式, 包括网站各功能的正常使用流程和遇到某些非常情况时的应对方式。

除现场培训外, 考虑到一部分用户不能到达现场, 因此应提供现场培训的录屏或进行直播供这部分用户进行培训。

6.1.3 培训目标

- 普通用户：学会使用普通搜索和高级搜索
- 专业用户：学会添加网站的高级功能

6.1.4 培训内容

- 爬虫基础，包括常见爬虫工具如 Selenium 的使用
- 图数据库基础，包括图数据库 OrientDB 的使用及相关 SQL 语句的编写
- Vue 前端开发基础，包含 HTML, CSS, JS 三件套的基本语法和 Vue 框架的基本架构、常用库如 Escharts 的使用
- Flask 后端开发基础，包含环境的配置和 python 语法基础，以及 flask 框架的基本概述

6.1.5 时间计划

| 第一次培训 | | | |
|-------|---------------------------|------|--------|
| 培训时间 | 2021/7/25 10： 00 – 11： 00 | 培训地点 | 曹西 503 |
| 培训主题 | 网站功能培训 | 主讲 | 马梓睿 |
| 参与者 | 普通用户代表， 专业用户 | | |
| 培训内容 | 网站基本功能 | | |
| | | | |
| 第二次培训 | | | |
| 培训时间 | 2021/7/25 9： 30 – 11： 00 | 培训地点 | 曹西 503 |
| 培训主题 | 网站功能培训 | 主讲 | 毕予然 |
| 参与者 | 专业用户 | | |
| 培训内容 | 网站高级功能 | | |

6.2 质量保证计划

- 项目质量保证工作项目的质量，质量保证工作由项目经理，质量经理和保证组人员来负责，质量保证人员在启动项目之后适用于单个项目的《质量保证计划》，并检查和督促计划的试实
- 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

6.3 测试计划

6.3.1 静态测试

静态测试是指无需执行被测代码，而是借助专用的软件测试工具评审软件文档或阅读程序代码，特别是不运行的部分，如规范测试、软件模型测试、文档测试等，度量程序静

态复杂度，检查软件是否符合编程标准，借以发现编写的程序的不足之处，减少错误出现的概率。主要方法有代码走查、技术评审、代码审查等。

在静态测试中，一大重点是实现软件编码规范检查。编码规范是程序编写过程中必须遵循的规则，一般会详细规定代码的语法规则、语法格式等，是根据项目本身及所处阶段适合编程人员的规范。测试工程师或程序员自身可借助一些专业的工具来实施编码规范检查。例如，在 C/C++ 语言的编程规则检查方面，比较专业的工具有 C++Test、LINT 工具、KlockWork 等。

静态测试还可实现静态质量度量。静态质量度量所依据的标准是 ISO9126，在该标准中，软件的质量以功能性、可靠性、可用性、有效性、可维护性和可移植性来衡量，其中最重要的是可维护性。要衡量软件的可维护性，可以从可分析性、可改变性、稳定性和可测试性去考虑。具体到软件的可测试性，又可以分为三个度量元，即圈复杂度、输入/输出个数。圈复杂度越大或输入/输出个数越多，说明代码中路径越多，测试用例越多。在具体的实践中，这一阶段专业的工具有 Testbed、Logiscope 等。

由修正成本与错误检测时间的正相关关系，错误越晚发现，修正的成本就越高。传统的错误检测是动态系统测试的范围，但为了减少成本，在静态分析代码时应加入错误检测。此类静态工具有 PolySpacee、PREFix 等。

6.3.2 单元测试

单元测试是代码正确性验证的最重要的工具，也是唯一需要编写代码才能进行测试的一种测试方法。单元测试的代码与实际程序的代码具有同等的重要性。每一个单元测试，都是用来定向测试其所对应的一个单元的数据是否正确。单元测试是由程序员自己来完成，证明某段代码的行为和我们期望的一致。单元测试还能够协助程序员尽快找到 BUG 的具体位置、让程序员对自己的程序更有自信、让程序员在提交项目之前就将代码变的更加健壮、能够协助程序员更好的进行开发、能够向其他程序员展现你的程序该如何调用、能够让项目主管更了解系统的当前状况。

单元测试又称为模块测试，主要侧重于模块中的逻辑和数据结构，一般使用白盒测试法，单元测试可以对多个模块同时进行。主要测试五个方面：

- 模块接口，保障数据的流出流入
- 数据结构，局部数据结构错误是常见的错误
- 执行路径，路径测试是基本任务，确保覆盖核心路径
- 出错处理，预测出错并提供合理的处理方法
- 边界条件，其是编程中常见错误

单元测试的对象可以是任何类、接口、结构等实体中的字段、属性、构造函数、方法等。创建单元测试大致可以分为两类：

- 整体测试，即为整个类创建单元测试，为整个类可以被测试的内容全部添加测试方法，开发人员直接在这些自动生成的测试方法中添加单元测试代码就可以了
- 单独测试，是单独对某个方法、属性、字段进行测试

编写单元测试代码时需要考虑哪些代码需要添加单元测试与单元测试代码如何写。单元测试代码需要逐步推进，第一步应该对所有程序集中的公开类以及公开类里面的公开方法添加单元测试；第二步对于构造函数和公共属性进行单元测试；第三步添加全面单元测试。在产品全面提交之前可以先完成第一步的工作，二三步可以待其他所有功能完成之后再添加。而在第一步单元测试添加的时候，也需要有选择性的进行，我们要抓住重点进行测试。重点有两方面，一是可复用代码测试，可以大大加强底层操作的正确性和健壮性；二是为业务逻辑层对界面公开的方法添加单元测试。这样可以让业务逻辑保持正确。

在编写单元测试代码的时候还需要考虑以下几个方面：

- 所测试的方法的代码覆盖率必须达到 100%，注意特殊情况包括边界条件数据、空数据与格式不正确数据
- 所测试的代码内部的状态，例如执行了某个方法之后，该方法所在的类中某个属性或者返回值是否与预期相同
- 被测试代码所使用的外部设备的状态，如数据库是否可读、网络是否可用、打印机是否可用、WebService 是否可用等等。每一段单元测试代码，必须考虑到以上的三个问题，并且对于这些问题都要有相应的测试

6.3.3 系统测试

这里的系统测试是相对于单元测试的整体测试，包括了集成测试、确认测试和系统测试。

集成测试就是将模块按照系统设计说明要求，组合成一个整体进行测试，是构造软件体系结构的系统化技术，主要是为了发现与接口相关的错误。集成测试有两种方式：非增量集成与增量集成，前者就是分别测试各模块，再组合起来进行整体测试；而后者则是不断向整体添加模块来运行测试，有自顶向下、自底向上、回归测试、冒烟测试等方法。

当集成测试完成后进行确认测试。确认测试主要集中于用户可见和可识别的系统输出，即是软件系统的整体行为，是通过针对软件需求或规格说明来制定测试用例，确保软件系统的行为与需求或规格说明吻合。确认测试也要关注软件系统的配置评审工作，即检查软件、文档、数据是否齐全有序。常用的方法有 α 测试与 β 测试。 α 测试是由有代表性的最终用户在开发者的场所进行，软件在一般的环境下使用，开发者全程观看，并记录错误和使用问题； β 测试是在一个或多个最终用户的场所中进行，这种测试开发者通常不在现场，由最终用户记录测试过程中的所有问题并定期向开发者报告。

系统测试是将已经确认的软件与计算机硬件、网络、外部设备等其他因素结合在一起进行的测试，这是一种贴近实际使用环境的测试。目的是发现系统整体是否与用户需求不符。主要分为恢复测试（通过各种方式强制让系统出现故障，并验证是否能从故障中恢复，并在约定时间内恢复运行，并且不造成任何损害。如果系统具备自我恢复的能力，则需要重新验证数据是否正确。如果是由人工干预的恢复，则要对恢复平均耗时进行评估）、安全性测试（测试人员模拟非法入侵者，采取不同的方法尝试突破系统，安全性测试的准则是确认进行非法入侵所要花费的代价是否大于系统被攻破后泄露信息的价值）、压力测试（以非正常的数量、频率或容量等方式执行，使系统在大幅超出设计指标的情况

运行以观察结果)、性能测试(注重于测试运行的性能,常与压力测试一同进行,而且需要硬件和软件作为工具,以严格的方式监控处理器、内存等各种资源的利用量)和部署测试(将软件在将要运行的每一种环境中执行的测试,并检查面向客户的所有安装程序和文档)。

6.4 维护计划

6.4.1 维护类别

- **改正性维护**: 改正性维护是指改正在系统开发阶段已发生而系统测试阶段尚未发现的错误。这方面的维护工作量要占整个维护工作量的 17%~21%。所发现的错误有的不太重要,不影响系统的正常运行,其维护工作可随时进行;而有的错误非常重要,甚至影响整个系统的正常运行,其维护工作必须制定计划,进行修改,并且要进行复查和控制。
- **适应性维护**: 适应性维护是指使用软件适应信息技术变化和管理需求变化而进行的修改。这方面的维护工作量占整个维护工作量的 18%~25%。由于计算机硬件价格的不断下降,各类系统软件屡出不穷,人们常常为改善系统硬件环境和运行环境而产生系统更新换代的需求;企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。这些因素都将导致适应性维护工作的产生。进行这方面的维护工作也要像系统开发一样,有计划、有步骤地进行。
- **完善性维护**: 完善性维护是为扩充功能和改善性能而进行的修改,主要是指对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。这些功能对完善系统功能是非常必要的。另外,还包括对处理效率和编写程序的改进,这方面的维护占整个维护工作的 50%~60%,比重较大,也是关系到系统开发质量的重要方面。这方面的维护除了要有计划、有步骤地完成外,还要注意将相关的文档资料加入到前面相应的文档中去。
- **预防性维护**: 预防性维护为了改进应用程序的可靠性和可维护性,为了适应未来的软硬件环境的变化,应主动增加预防性的新的功能,以使应用系统适应各类变化而不会被淘汰。例如将专用报表功能改成通用报表生成功能,以适应将来报表格式的变化。这方面的维护工作量占整个维护工作量的 4%左右。

6.4.2 维护工作安排

表 6-1 项目维护工作安排

| 工作内容 | 维护周期 | 需要用户配合的事宜 | 其他说明 |
|-------|------|---------------------------|---|
| 服务器维护 | 一个月 | 暂停对网站的访问 | 主要是维护保证服务器的性能 |
| 数据库维护 | 一周 | 无 | 主要是维护保证数据库数据的正确性和完整性,必要时可修改数据表结构及数据库迁移等 |
| 后端维护 | 半周 | 提醒用户在使用过程中可能会出现卡顿或无法响应的情况 | 主要是对后端与数据库的数据交互和数据处理代码进行优化 |

| | | | |
|------|----|---------------------------|---------------------------------|
| 前端维护 | 半周 | 提醒用户在使用过程中可能会出现页面显示不正常的情况 | 主要是对前端框架的整改、界面设计的美化以及前后端数据交互的规范 |
|------|----|---------------------------|---------------------------------|

6.4.3 巡检计划

- 巡检周期：1 周
- 巡检方式：主要以现场为主，适当结合远程巡检
- 巡检工程师：沈乐明
- 巡检产出物：产品巡检记录表、《巡检报告》

