系统分析与设计

**科技专家资源共享平台**

**软件架构说明书**

|  |  |  |
| --- | --- | --- |
| **项目组成员信息** | | |
| **小组名称** | **洪洋大小姐不想爆肝** | |
| **小组联系人** | **杨洪洋、林未** | |
| **学号** | **姓名** | **本文档中主要承担的工作内容** |
| 16211028 | 杨洪洋 | 负责质量属性说明，审核完善文档，完成整个文档。 |
| 16211085 | 索宇澄 | 负责前端框架说明及安全性说明。 |
| 16211093 | 林未 | 设计软件整体架构，设计服务器部署方案。 |
| 16211079 | 赵双叶 | 负责横切关注点说明，负责数据库说明及服务器说明。 |
| 16211088 | 凌国辉 | 负责通信机制说明。 |
| 16211117 | 王嘉星 | 负责爬虫技术的说明以及确定数据源的选取。 |
| 16211077 | 王启鹏 | 负责后端框架说明及统计分析算法的确定 |
| 16211125 | 赵昀睿 | 协助前端架构说明，完成前端原型图 |

2019-04

版本变更历史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 提交日期 | 主要编制人 | 审核人 | 版本说明 |
| 01 | 2019.4.3 | 索宇澄、凌国辉、王启鹏，林未 | 杨洪洋 | 对前后端框架进行介绍，完成整个平台的包图 |
| 02 | 2019.4.8 | 赵双叶、杨洪洋 | 杨洪洋 | 对质量属性和横切关注点进行描述 |
| 03 | 2019.4.9 | 林未、杨洪洋 | 杨洪洋 | 完善部署方案、组织文档结构 |
| 04 | 2019.4.10 | 林未 | 杨洪洋 | 完善部署方案 |

目 录

[1. 概述 4](#_Toc5804622)

[1.1 系统背景 4](#_Toc5804623)

[1.2 系统需解决的问题 4](#_Toc5804624)

[1.3 文档概述 4](#_Toc5804625)

[2. 应用程序类型及系统部署方案 4](#_Toc5804626)

[2.1 应用程序类型 4](#_Toc5804627)

[2.2 系统架构设计 4](#_Toc5804628)

[2.3 系统部署模型 5](#_Toc5804629)

[3. 技术选型 7](#_Toc5804630)

[3.1 技术框架表格 7](#_Toc5804631)

[3.2 重要技术使用描述 8](#_Toc5804632)

[3.2.1 后端框架 8](#_Toc5804633)

[3.2.2 前端框架 8](#_Toc5804634)

[3.2.3 通信机制 8](#_Toc5804635)

[3.2.4 爬虫机制 10](#_Toc5804636)

[4. 待解决业务问题及解决方案 10](#_Toc5804637)

[4.1 待解决问题 10](#_Toc5804638)

[4.1.1 待解决问题列表 10](#_Toc5804639)

[4.1.2 高并发大数据访问情况下系统的正常运行 10](#_Toc5804640)

[4.1.3 审核效率及审核方式 11](#_Toc5804641)

[4.1.4 推荐算法的选择 11](#_Toc5804642)

[4.1.5 非结构化数据的存储形式 11](#_Toc5804643)

[4.1.6 科技资源成果的获取方式 11](#_Toc5804644)

[4.1.7 积分体系与收费模式 12](#_Toc5804645)

[4.1.8 对科技专家资源的统计和分析算法 13](#_Toc5804646)

[5. 质量属性说明 14](#_Toc5804647)

[6. 横切关注点说明 17](#_Toc5804648)

[6.1 安全 17](#_Toc5804649)

[6.1.1 数据库安全设置 17](#_Toc5804650)

[6.1.2 服务器安全设置 17](#_Toc5804651)

[6.1.3 Web安全 17](#_Toc5804652)

[6.1.4 传输安全 18](#_Toc5804653)

[6.1.5 源码安全 18](#_Toc5804654)

[6.1.6 前端数据安全 18](#_Toc5804655)

[6.1.7 监视、日志记录和报告 18](#_Toc5804656)

[6.1.8 主机安全（云镜） 18](#_Toc5804657)

[6.2 异常处理 19](#_Toc5804658)

[6.3 缓存 19](#_Toc5804659)

[6.3.1 软件缓存 19](#_Toc5804660)

[6.3.2 硬件缓存 19](#_Toc5804661)

# 概述

## 系统背景

科技专家资源共享平台主要用于有效地推动科研成果合作和科技资源共享，通过在统一管理的基础上建立科技专家门户系统以实现管理和维护科技专家资源。平台基本的运营和管理包括：获得专家科技资源成果、科技专家申请管理和维护自己的门户系统、普通用户检索和查看科技专家各类资源、对各类专家资源进行有效的统计和分析。通过进行需求调研、建立需求模型和分析模型，可以建立软件的架构设计。通过对平台前后端框架进行分析与设计，并对平台的质量属性和横切关注点进行刻画，为后来的平台开发奠定基础。

## 系统需解决的问题

本系统为面向全国乃至全世界的科技专家资源共享平台，其对系统规模、数据量和并发量具有较高的要求。为了满足广大用户的需求，在平台设计时必须确定可靠的数据源，在数据爬取时必须符合法律的要求。同时，为了保证平台的效率问题，科技专家的审核以及科技成果的转让必须使用安全快速的方式。除此之外，为了实现对各类专家资源的有效统计和分析，在平台设计时必须采用可靠高效的统计分析算法。

## 文档概述

本文档基于系统需要解决的问题，从应用程序类型、技术方案、质量属性和横切关注点对平台进行刻画。通过对每一部分的详细介绍与描述进一步确定系统所使用的技术以及针对于关键问题的解决方案。

# 应用程序类型及系统部署方案

## 应用程序类型

在需求调研阶段，我们曾就应用程序类型做过专门的问卷调查。调查结果显示，有关于科技平台的形式，大多数管理员希望通过网页端进行科技平台的管理。通过网页端可以在一定程度上解决平台不一致所造成的困难，获得了管理员的普遍支持。对于用户而言，网页端省去了用户在使用前下载、安装应用程序的步骤，不仅降低了学习成本，而且对各种设备均具有比较好的兼容性，尤其在配置较低的机器上体验友好。

综上所述，本平台拟采用网页端进行开发、部署与运行。

## 系统架构设计

科技专家资源共享平台的包图如图2-1所示。

C:\Documents and Settings\Administrator\Desktop\Image2.EMF

**图2-1 科技专家资源共享平台包图**

“科技专家资源共享平台”的分包过程采用“B-C-E”策略，即“边界-控制-实体”。用户界面包存放平台的界面类，如科技成果界面类、个人空间界面类等；业务逻辑包中包含实现各种业务的控制类，如管理科技成果控制类、充值消费积分控制类等，负责与前端进行交互，并处理与平台相关的业务流程；基础业务包用于存放平台的基础业务数据，如各种记录数据实体类、科技成果实体类等；外部接口包存放着与本平台相关的各个外部接口；从可扩展性的角度考虑，将“第三方支付系统接口类”封装为子系统，用来实现与第三方支付系统的多个接口，所以它依赖于外部接口包。

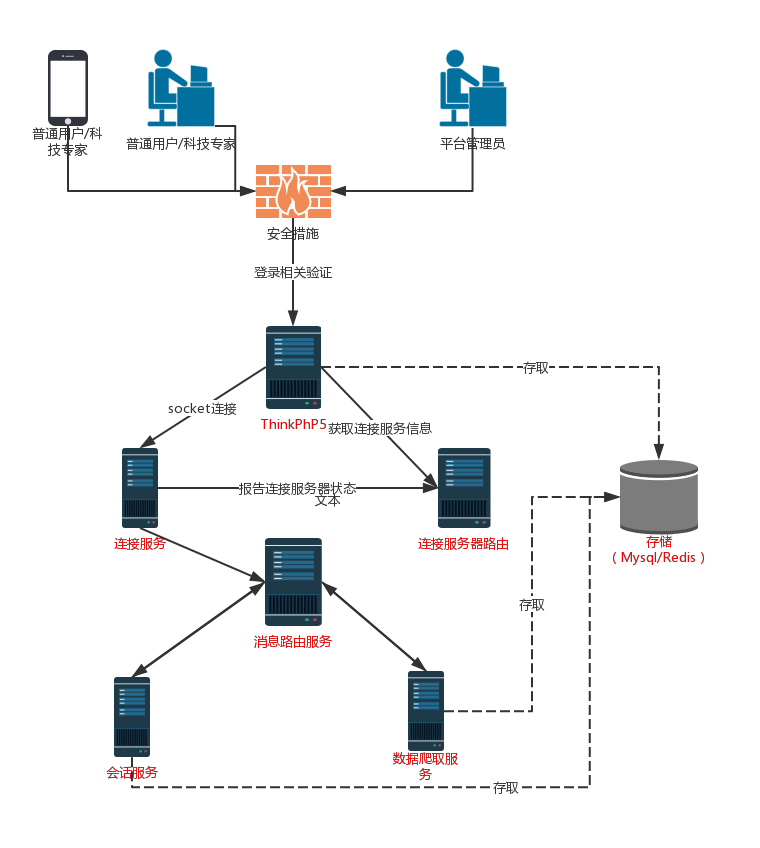
## 系统部署模型

科技专家资源共享平台的相关部署图如图2-2~图2-4所示。

C:\Documents and Settings\Administrator\Desktop\Image2.EMF

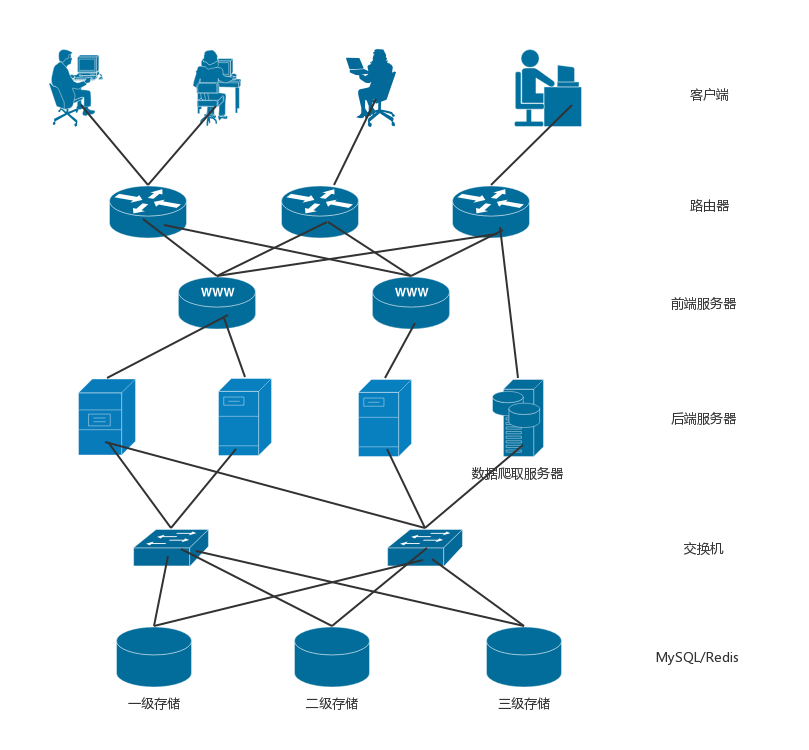
**图2-2 科技专家资源共享平台部署模型**

该模型包括一类客户端节点：Web客户端，而服务器端的节点均描述为设备节点；此外，在数据库服务器节点内部定义了一个执行环境节点说明所采用的数据库管理系统为MySQL。节点之间的关联关系代表节点间的通信路径，如Web客户端通过HTTP的方式访问Web服务器，而服务器之间则部署在同一个局域网内。



**图2-3 科技专家资源共享平台服务器部署方案（功能）**

上图从功能角度描述了平台服务器的部署方案。用户或管理员在登录平台时需要进行完备的安全验证，防止权益受损。后端核心框架为ThinkPhP5，与连接服务器采用socket连接。消息路由服务器为连接服务与会话服务及数据爬取服务之间建立起桥梁，实现数据的可靠传输，在保证效率的同时不会因突增的访问流量而宕机。这些服务器均通过与数据库（存储）服务器进行交互来存取数据。



**图2-4 科技专家资源共享平台服务器部署方案（拓扑）**

上图从拓扑角度描述了平台服务器的部署方案。可以看到，用户或管理员通过HTTP的方式访问平台前端服务器，前端服务器与后端服务器产生交互传递信息，当平台管理员有更新科技成果等需求时，可以通过互联网访问数据爬取服务器及时更新科技成果信息。后端服务器与数据库服务器在同一局域网内，因此通过交换机进行连接。数据库管理系统为MySQL，同时采用Redis三级缓存机制。

# 技术选型

## 技术框架表格

表3-1 技术框架表

|  |  |
| --- | --- |
| 应用框架 | Thinkphp，bootstrap，JQuery |
| 应用层 | Php |
| 工作流 | Tpflow |
| 代码整合管理 | Git |
| 数据框架 | Json |
| 分布式对象-关系型数据库 | Mysql |
| Web前端开发工具 | VSCode |
| Dreamweaver |
| Web服务器端开发工具 | Phpstrom |
| Pycharm |
| VSCode |
| 移动端 | HTML5 |
| 富客户端应用服务 | HTML5 |
| JavaScript |
| 故障检测及容灾 | 腾讯云云盾 |
| 缓存控制 | 本地缓存 |
| 资源打包 | Webpack |

## 重要技术使用描述

### 后端框架

后端计划采用ThinkPhP5框架。ThinkPHP是一个快速、简单的基于MVC和面向对象的轻量级PHP开发框架。具有以下一些优点：

1. 异常严谨的错误检测和安全机制，详细的日志信息
2. 减少核心依赖，扩展更灵活、方便，支持命令行指令扩展
3. 出色的性能和REST支持、远程调试，更好的支持API开发
4. 重构的数据库、模型及关联，MongoDB支持

在本次设计中，采用前后端分离的方式，前端向后端请求需要的数据或者服务，后端处理完成之后返回，前后端通过AJAX实现连接完成消息互通和页面更新等功能。

### 前端框架

前端计划采用Bootstrap框架。Bootstrap是基于HTML、CSS、JavaScript 开发的简洁、直观、强悍的前端开发框架，使得 Web 开发更加快捷。Bootstrap提供了优雅的HTML和CSS规范，简单灵活，可用于架构流行的用户界面和交互接口的html、css、javascript工具集。基于html5、css3的Bootstrap具有大量的诱人特性：友好的学习曲线，卓越的兼容性，响应式设计，12列格网，样式向导文档。并且具有自定义JQuery插件，完整的类库，基于Less等优势。Bootstrap框架为用户提供了一套响应式移动设备优先的流式栅格系统，拥有完备的框架结构，整体效果和谐，对谷歌，火狐，IE等浏览器均可支持，项目开发方便快捷。

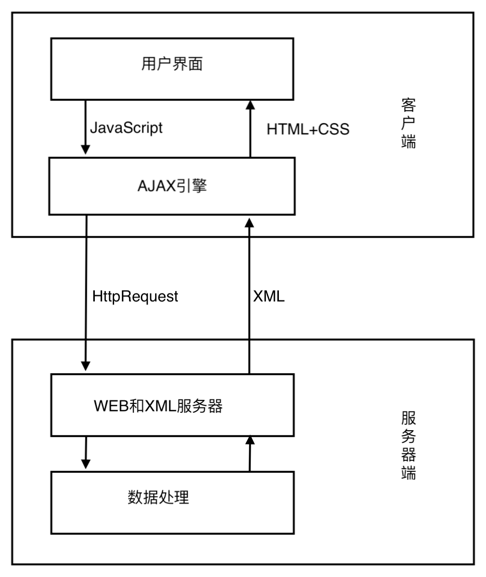
### 通信机制

* + - 1. 前后端通信机制

对于本平台，前后端之间的通信将采用AJAX技术。

AJAX全称为“Asynchronous JavaScript and XML”（异步JavaScript和XML），是一种创建交互式网页应用的网页开发技术，通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。除此之外，AJAX技术可以在不刷新整个页面的情况下与服务器通信保持原有页面状态；使用异步方式与服务器通信不需要终端操作；将服务器负担转嫁给客户端，减轻服务器和带宽，减少冗余请求。

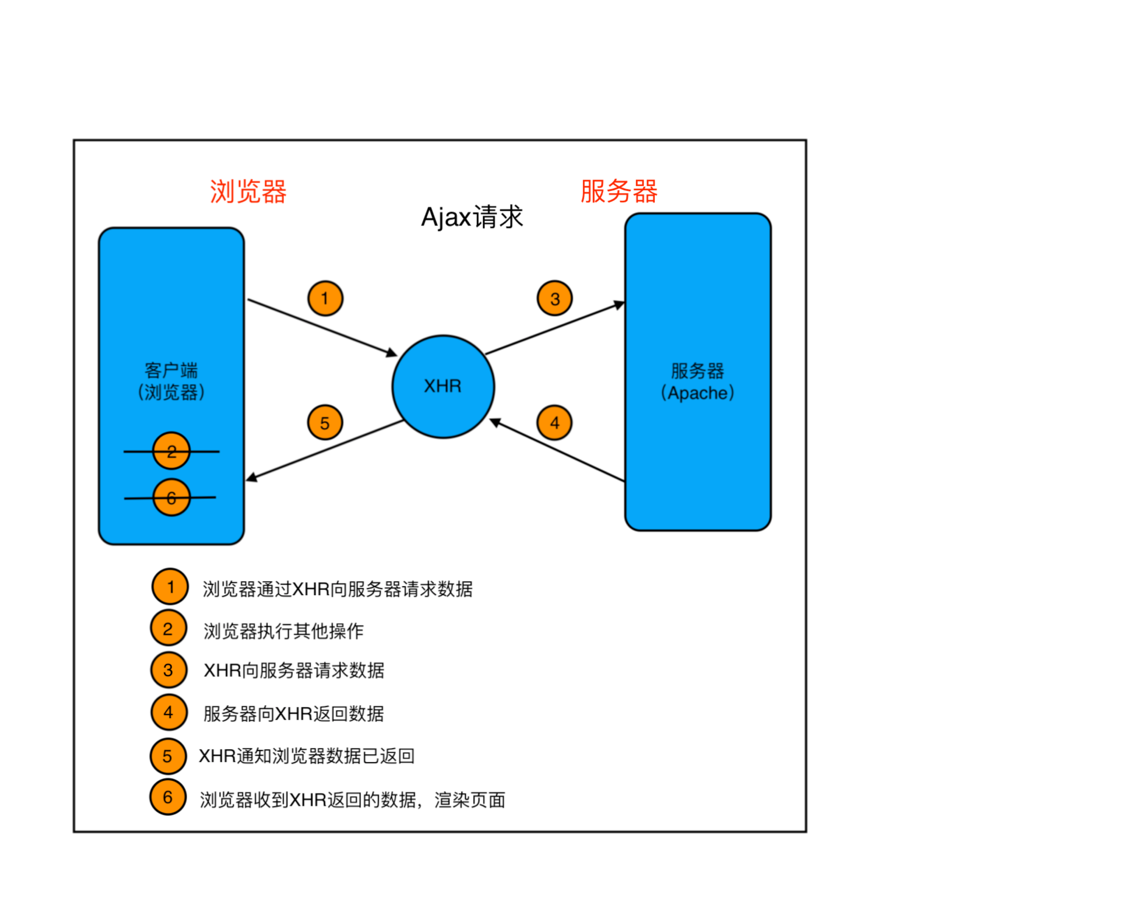
AJAX的工作原理如下图：



**图3-1 AJAX通信机制**

AJAX使用XMLHttpRequest对象发送请求并得到服务器响应，在不需要重新载入整个页面的情况下，就可以通过DOM及时将更新的内容显示在页面上。

AJAX请求数据的过程如下：



**图3-2 AJAX请求数据机制**

* + - 1. 组件间通信机制

对于组件间的通信，本平台拟采用ActiveMQ。ActiveMQ是Apache下的一个子项目，使用Java完全支持JMS1.1和J2EE 1.4规范的 JMS Provider实现，仅仅少量代码就可以高效地实现高级应用场景。它有可插拔的传输协议支持，支持常用的多种语言客户端，支持通过JDBC和journal提供高速的消息持久化从设计上保证了高性能的集群，除此之外它对于Ajax也提供支持。

### 爬虫机制

爬虫的主要思路为先确定需要爬取的网页的url地址，通过HTTP/HTTP协议来获取对应的HTML页面，提取HTML页面中有用的数据，在本项目中如论文信息、专家信息等。

数据爬取采用Python语言，利用scrapy框架进行开发。Scrapy运行流程大概如下：

1.引擎从调度器中取出一个链接(URL)用于接下来的抓取

2.引擎把URL封装成一个请求(Request)传给下载器

3.下载器把资源下载下来，并封装成应答包(Response)

4.爬虫解析Response

5.解析出实体（Item）,则交给实体管道进行进一步的处理

6.解析出的是链接（URL）,则把URL交给调度器等待抓取

# 待解决业务问题及解决方案

## 待解决问题

### 待解决问题列表

1. 高并发大数据访问的情况下，如何保证系统的正常运行？
2. 专家申请管理门户系统时的审核效率问题，在平台设计时需要考虑审核的方式。
3. 推荐算法的选择和实现，搜索相关资源的技术。
4. 系统中的非结构化数据例如图片、论文、专利以及项目文件等，该如何进行存储？
5. 结构化数据和非结构化数据的数据规模大约是多少？如何解决数据规模大的问题？
6. 科技资源成果的获取方式，主要为数据源的选取以及如何抓取平台需要的数据。
7. 如何构建平台的积分体系与平台的收费模式？
8. 对于各类专家资源，如何进行有效的统计和分析？

### 高并发大数据访问情况下系统的正常运行

首先在逻辑设计时要保证数据库的一致性和完整性，在涉及到过多的表间关联，尽可能的降低数据的冗余；其次保证在实现功能的基础上，尽量减少对数据库的访问次数，通过搜索参数，尽量减少对表的访问行数，最小化结果集，从而减轻网络负担；可以使用试图进行加速查询，或者简历高校的索引；尽量避免大事务操作、尽量避免向客户端返回大数据量、尽量避免使用不兼容的数据类型。

### 审核效率及审核方式

对于本平台的科技专家申请认证模块，采用人工+自动审核的方式。对于人工审核规定期限，如果在1天内未处理科技专家的认证申请，则转为平台自动审核。

平台通过前端传递的数据对科技专家的身份进行审核。科技专家的认证数据主要分为身份证数据以及认证邮箱数据。

针对身份证数据，后端可以通过使用腾讯云的ocr.tencentcloudapi.com接口请求域名实现对身份证的识别，接着将身份证识别的关键信息（如姓名、出生日期、籍贯等信息）与科技专家的信息进行核对和审查，最后向前端返回比对结果。

针对认证邮箱数据，后端PHP通过基于PHP的邮箱地址验证API检查邮箱存在与否，接口调用地址为http://v.juhe.cn/verifyemail/query。并且可以通过PHPMailer这个现成的类文件实现验证码的发送。用户登录认证邮箱获取验证码，在平台输入验证码。若验证码正确，则通过审核。

### 推荐算法的选择

考虑选择协同过滤算法、user-based-cf或item-based-cf算法，利用现有的搜索引擎，如RiSearch、Sphinx，完成推荐和搜索的核心功能。

### 非结构化数据的存储形式

通过将资源文件存储在服务器的文件系统之中，在数据库的表中存储资源在服务器文件系统之中的绝对路径。这样，当需要取用某个资源时，根据它在磁盘上的路径便可很快找到它。

根据系统用户估计，结构化的信息总量大约在100GB，结构化数据存储所需空间的总量约为100GB，非结构化数据存储所需的空间为200GB，总计大约为300GB。为了防止数据丢失，需要在远程服务端对于数据库内容进行备份，需要相同的额外存储空间。

### 科技资源成果的获取方式

* + - 1. 数据源的选取

数据源选择百度学术、中国知网的学术资源，在数据源选择过程遵循专指性原则、专业性原则、全面性原则、整体化原则、通用性原则。

百度学术收录了包括知网、维普、万方、Elsevier、Springer、Wiley、NCBI等的120多万个国内外学术站点，索引了超过12亿学术资源页面，建设了包括学术期刊、会议论文、学位论文、专利、图书等类型在内的4亿多篇学术文献，是全球文献覆盖量最大的学术平台。

“中国知网”系列数据库(简称CNKI)为一系列大规模集成整合传播我国期刊、博硕士学位论文、工具书、会议论文、报纸、年鉴、专利、标准、科技成果、古籍等各类文献资源的大型全文数据库和二次文献数据库，以及由文献内容挖掘产生的知识元数据库。至今为止收录了10000余种学术期刊、37万余篇博士论文、357万余篇优秀硕士论文、国内公开发行的500多种重要报纸等，拥有大量有价值的文献资源。

综上，数据源确定为百度学术与中国知网的学术资源。

* + - 1. 数据的抓取方式

爬虫的主要思路为先确定需要爬取的网页的url地址，通过HTTP/HTTP协议来获取对应的HTML页面，提取HTML页面中有用的数据，在本项目中如论文信息、专家信息等。

数据爬取采用Python语言，利用scrapy框架进行开发。Scrapy运行流程大概如下：

1.引擎从调度器中取出一个链接(URL)用于接下来的抓取

2.引擎把URL封装成一个请求(Request)传给下载器

3.下载器把资源下载下来，并封装成应答包(Response)

4.爬虫解析Response

5.解析出实体（Item）,则交给实体管道进行进一步的处理

6.解析出的是链接（URL）,则把URL交给调度器等待抓取

对于可能遇到反爬机制，平台采取相应解决方案：

1.所爬取网站以检查User-Agent作为反爬虫机制，通过设定Request Headers中的User-Agent，可以突破这种机制。

2.网站限制访问频率，或通过检测访问时间间隔判断是否为爬虫。可以选用Selenium访问网站内容,由于Selenium打开页面本身就需要一段时间，从而避免了频率检查的反爬虫机制，这种做法缺点为效率较低。本项目对爬虫速度有较高要求，故应选择代理IP或分布式爬虫的方式，代理IP为通过设置维护一个可用代理IP池进而规避网站对访问频率的限制。分布式爬虫则会部署在多台服务器上，每个服务器上的爬虫统一从一个地方拿网址。这样平均下来每个服务器访问网站的频率也就降低了。

### 积分体系与收费模式

* + - 1. 积分体系

积分体系用于维护平台正常运营，是用户正常进行学术资源下载的必需条件，促进学术平台运营工作的可持续发展。普通用户积分获取方式为充值获取，10积分/元。积分可用于下载科技成果，下载科技成果所需积分由拥有该科技成果的科技专家设置。

* + - 1. 收费模式

平台收费方式应方便用户支付，防止用户因复杂的付费过程放弃充值，故采用当前用户使用较多的支付宝或微信作为第三方收费接口。PC端可采用扫描二维码方式付款，手机端可直接调用支付应用进行付款。在收到用户的付费后应保证在一分钟内将相应数额的积分添加到该用户的账户中。为用户提供客服联系渠道，解决充值过程中发生的问题（积分未到账、积分数额充值错误等）。

积分结算应完成充值的校核验证，将每一笔充值进行结算生成账单，将数据结算结果返回到数据库里，更改用户积分余额。

以微信支付为例，微信支付功能可采用php语言进行开发，用户支付流程为：

1.用户在使用模块功能需要支付时，模块调用父类的$this->pay($params);方法来调用系统支付功能。

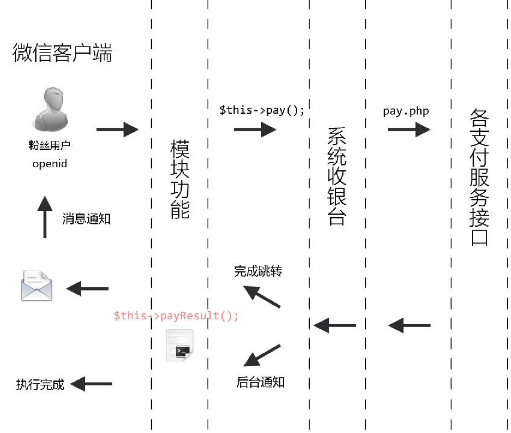
2.系统收银台接到模块调用请求，生成支付界面（金额，商品信息等）。

3.用户在收银台中选择支付手段，如微信支付，支付宝支持等（需要在支付参数中设置开启）。

4.收银台根据用户选择的支付手段，请求到相应的支付接口上，用户完成支付。

5.完成支付后，系统会调用模块中的 $this->payResult($params); 方法来通知支付状态（成功或是失败）。

如图所示：



**图4-1 收费模式的实现方式（以微信支付为例）**

### 对科技专家资源的统计和分析算法

* + - 1. 科技资源统计

针对每个科技资源，统计其搜索量、引用量、下载量等相关信息，之后对所有的科技资源按照不同的指标进行排序，统计分析，筛选热门资源。

对于不同的科研机构或团体，统计其拥有的科技成果信息，之后对其科技成果进行分类，按照不同的指标选择热门资源。

对于所有的科技资源排序时，数据量大，采用分治的方法，按照选择的指标将数据划分为不同部分，利用外排序的算法，对当前部分排序之后进行合并，最终完成所有资源的统计分析。

* + - 1. 热点分析

每个科技成果拥有多个分类标签，统计各类科技资源的数量，每类资源的搜索量、点击量、下载量、浏览量，之后分别按照不同的指标进行排序，按照不同的指标统计当下的科研领域热点，最后可以给出每类资源的综合排名，向用户推荐。

对于热点领域需要进行排序，由于科技资源所属的类别有限，可以直接使用已有的排序算法（如快速排序、归并排序等）实现。

* + - 1. 专家关系网络分析

专家关系网络的挖掘的本质即为社区检测，社区检测算法目前比较成熟，有基于图分割算法、层次聚类算法、谱方法和基于PCA的方法等，在专家关系网络中，我们采用基于层次的聚类算法。

对于所有专家，按照其论文相互引用量建立网络图，社区具有聚类的特点，即统一社区中的专家相互引用次数高，不同社区专家之间相互引用次数低，利用层次聚类的挖掘算法构建社区网络，对应的专家关系网络即为同一社区内与之相关的专家同其构建的网络。在得到专家关系网络之后，通过前后端的通信机制，在前端向用户推荐相关专家或者机构。

# 质量属性说明

表4-1 质量属性表

|  |  |  |
| --- | --- | --- |
| 分类 | 质量属性 | 描述 |
| 设计质量 | 概念完整性 | 统一的模块设计：   1. 后端根据不同服务划分模块，例如普通用户的服务与科技专家的服务可以分开。   统一的命名规范：   1. 后端开发变量名、参数名与方法名称采用驼峰命名法，前端则采用下划线的命名法。 2. 类的命名采用帕斯卡命名法。 3. 命名尽量完整，可以清楚表达含义与作用。   统一的代码风格：   1. 尽量避免一行过长的代码。 2. 适当空行，避免过多空行。 3. 花括号换行。 |
| 可维护性 | 1. 开发团队统一制定编码规范，每个人都必需按照编码规范进行编程。 2. 适当的代码注释。每个类前应有代码注释段，说明class的主要用途，基本用法，注意事项；每次修改都需要在此处添加修改日志，修改日志尽量详细；每个方法前有//注释来说明方法的用法，要说明方法功能，参数及返回值含义。 3. 系统使用“高内聚、低耦合”的模块化设计。 |
| 可重用性 | 系统采用模块化设计，保证软件的可重用性。前端一些组件可以被几个网页重用，不同类与方法可以被重用，一些模块可以被重用。 |
| 运行质量 | 可用性 | 综合考虑服务器等设备冗余和负载均衡来保证服务器能够在97%的7\*24时间内正常运行：   1. 应用服务器和存储服务器通过负载均衡设备组成集群，在系统繁忙时均摊负载，不繁忙时有一台可以宕机维护，有一台服务器故障系统可以正常使用。 2. 应用部署在多台服务器上同时提供访问，数据存储在多台服务器上互相备份。存储服务器应对数据进行实时备份。 3. 交换机与路由器有一台故障，系统可以正常使用。 4. 使用腾讯云安全服务，在虚拟实例之间采用了安全隔离，当实例故障时，可以漂移到其他实例保障业务的高可用。   在遇到不可抗拒因素、例如运营商网络故障等状况时，系统将无法正常运行，需系统管理员维护解决。 |
| 互操作性 | 1. 系统可以通过邮件系统发送邮件交互。 2. 系统可以使用第三方支付。 3. 系统可以从其他网站系统中爬取数据。 |
| 运行质量 | 性能 | 1. 在浏览器端，通过浏览器缓存、合理布局页面、减少Cookie传输等手段对性能进行改善，通过优化页面HTML式样与使用反向代理机制，使得浏览器尽快显示用户感兴趣的内容。 2. 在应用服务器端，使用服务器本地缓存和分布式缓存，通过缓存在内存中的热点数据处理用户请求，加快请求处理过程，减轻数据库负载压力。 3. 在代码层，使用多线程、改善内存管理、代码优化等手段优化性能。 4. 充足的数据存储空间。 5. 对网站的响应时间、TPS、系统性能计数器等指标进行监控，以进一步分析系统瓶颈、预测网站容量等。 |
| 可靠性 | 1. 前端和采集器都部署了两台服务器，在一台服务器故障时业务不会停止，只是性能会受到影响。后端部署了四台服务器，始终有一台冗余，在服务器出现故障时可以立刻补上。 2. 在三级存储结构中，一级存储采用 RAID6，保存了两种校验信息，能够很好的对故障区域进行恢复。二级存储采用 RAID5，在磁盘利用率较高的情况下也提供了较好的故障恢复。 3. 按功能模块划分用户权限，遵循最小权限原则。 4. 如果用户提交了错误的信息，系统应该有相应的容错机制，比如提醒用户输入错误或者后台可以对相应的错误数据进行处理，保证系统的健壮性。 5. 使用腾讯云安全服务，后端硬件设备和虚拟实例通过集群化实现高可靠 |
| 可伸缩性 | 1. 在nginx负载均衡和 NIC的基础上，可以简单地通过增加服务器 CPU、增加服务器数量的方式来扩展网站的性能。 2. 拥有足够的磁盘空间来存储和扩展数据库。 3. 路由器与交换机冗余的接口数量，便于网络结构的扩展。 4. 因为采用了 RAID，增加存储磁盘需成组增加。 5. 系统可以快速支持新兴平台，比如移动端访问。 |
| 安全 | 1. 系统设计时采取权限设置和数据库加密等措施，只有管理员可以更新修改数据库。 2. 防sql注入攻击。采取一些措施防止非法数据侵入系统。如查询条件使用数组方式、使用字符串时使用预处理机制、使用PDO方式来绑定参数、输入验证、错误消息处理、加密处理、安全评审等。 3. 敏感信息加密。对网页的敏感信息进行加密。比如，用户的密码等。 4. 数据备份和恢复，定期备份数据库，当网站出现故障导致信息丢失的时候，对数据库进行恢复。 5. 使用腾讯云安全服务，采用国家密码管理局批准的硬件芯片实现各类密码算法。 |
| 系统质量 | 可支持性 | 系统架构中有日志子系统，对系统运行建立日志记录。日志对于不同状态比如系统级错误，一般错误与正常状态的访问等有不同记录，容易区分，容易获取系统运行情况，这样在系统出现问题后更容易解决。 |
| 可测试性 | 1. 系统遵循“高内聚、低耦合”原则划分模块，前后端分离，后端采用微服务架构， 2. 代码简洁、清晰，尽量做到代码可以复用，系统开发方法与类都小型化设计。 3. 用户可以把科技专家资源共享平台的不同功能分支和相应流程从头到尾测试一遍，测试的过程中要考虑是不是有的过程可以去掉或者有的部分可以改进等。 |
| 用户质量 | 可用性 | 1. 系统界面简洁清晰，功能分区明了，主要使用人群面向未登录用户和科技专家。 2. 系统的搜索功能可以精确匹配用户的搜索内容；用户可以通过资源评价来对资源进行选择，方便用户找到最好的资源和专家。 3. 用户可以在电脑也可以在手机上网页登录我们的系统,可以采用响应式网站设计，采用优秀的前端框架JQuery+ Bootstrap, 更好地适配不同的设备。 |

# 横切关注点说明

## 安全

### 6.1.1 数据库安全设置

* 使用RAID技术对数据库的提供数据保护。
* 使用磁带进行数据备份。
* 使用用户身份鉴别机制为数据库提供最外层的安全保护措施。每一位管理员在系统中都有唯一的标识，每次在登录数据库管理系统的时候提供相应的口令进行验证。
* 使用存取控制以确保只授权给有资格的用户访问数据库的权限。比如，对于一些运营人员，只需要有权限能够更改首页的一些展示内容就可以了；而对于系统管理员，需要有权限访问数据库中所有的表的数据。存取控制主要分为自助存取控制和强制存取控制，考虑到安全性级别的问题，我们打算采用强制存取控制。。
* 使用视图将数据对象限制在一定范围内。具体来说，就是利用视图机制将需要保密的数据对无权存取的用户隐藏起来，只提供给它有权访问的数据。
* 添加身份检查，基于用户类型为用户的行为添加约束，通过区分不同的访问者，不同的访问类型和不同的数据对象，进行分别对待。对于数据库设计中涉及到的项目，如数据项、记录、系、文卷、模式、子模式等建立数据保护措施。数据库有专门的管理人员进行数据库操作和维护。

### 6.1.2 服务器安全设置

* 本平台使用腾讯云服务器，所以采用腾讯云云盾盾进行服务器安全的控制。其主要优点有木马检测、针对大流量攻击提供DDoS防护，高可用性，自定义网段划分、IP地址和路由策略等。

### 6.1.3 Web安全

* 防止SQL注入攻击。采取如下措施：过滤非法字符——保证传来的字符串作为一个参数,而不是语句拼接的一部分；使用正则表达式来确保用户输入合法；sql语句查询时使用预处理语句而不是单纯进行字符串拼接。
* 使用腾讯云的安全服务。针对于Web 的安全进行Web安全漏洞扫描。漏洞扫描是用于监测网站漏洞的安全服务，为企业提供 7\*24 小时准确、全面的漏洞检测服务，并为企业提供专业的修复建 议，从而避免漏洞被黑客利用，影响网站安全。具体功能如下：
  + 无损扫描。在网站运维过程中网站的业务健康性是至关重要的，因此 Web 漏洞扫描的扫描服务采用了无损的漏洞扫描技术，以避免对网站业务的健康性造成影响。
  + 修复闭环管理。通过提供精准、全面的漏洞检测，并给出专业的修复建议，帮助进行有效验证和加固资产漏洞；与此同时还会对漏洞的修复情况进行跟踪，实现漏洞生命周期的全程闭环管理。
  + 免安装不限次。通过SaaS 模式的扫描服务，添加完站点验证即可一键开启扫描，无需部署，无需安装任何软硬件产品；强大的并发扫描能力，不限扫描次数，极大降低您的安全运维成本。
  + 报告展示。详尽的漏洞扫描报告，包括：系统存在的安全漏洞、安全配置问题、应用系统安全漏洞，系统存在的弱口令，不必要开放的账号、服务、端口等，及修复建议，引导并帮助用户修补漏洞。

### 传输安全

* 为防止在数据传输过程中，数据被恶意用户截获或者篡改，我们对用户敏感信息，比如密码和用户真实姓名等，进行加密后再进行传输，可选的方案有DES加密，AES加密等等。
* 使用JSON数据格式进行前后端数据的传输，而不是采用路径拼接的方式。

### 6.1.5 源码安全

* 源码安全主要为JavaScript的压缩和混淆，将整体进行压缩，去除不必要的缩进；将变量名进行替换，使用下划线、字母随机替换变量名，使得攻击者无法通过变量名对其功能进行猜测，加大了其攻击难度。

### 6.1.6 前端数据安全

* 将用户的密码输入进行隐藏，保证用户在输入时其他人无法从界面中查看得到用户的密码。对于用户的敏感数据，不对其进行储存和缓存。除了必要的sessionID和token以外，cookie中不储存其他内容。
* 与后端之间的请求，首先需要经过类型检查和数值检查，之后通过https连接进行传输，依靠TSL协议的非对称加密，可以保证攻击者无法破解用户向服务器发送的数据。

### 6.1.7 监视、日志记录和报告

* 采用监视、日志等措施监察、记录服务器中的异常事件，并及时报告给系统管理员，做到防微杜渐。

### 主机安全（云镜）

主机安全（云镜）基于腾讯安全积累的海量威胁数据，利用机器学习提供黑客入侵检测和漏洞风险预警等安全防护服务，主要包括密码破解拦截、异常登录提醒、木马文件查杀、高危漏洞检测等安全功能，解决当前服务器面临的主要网络安全风险，帮助企业构建服务器安全防护体系，防止数据泄露。具体功能如下：

* 木马查杀。基于机器学习对各类恶意文件进行检测，包括各类 WebShell 后门和二进制木马 。
* 密码破解提醒。对密码恶意破解类进行检测并提醒，告诉用户当前正在遭受的密码破解事件和破解结果。
* 登录行为审计。采集服务器上的登录日志进行分析，对非法登录进行实时告警。
* 漏洞管理。对主机上存在的高危漏洞风险进行实时预警和提供修复方案，包括系统漏洞、WEB 类漏洞，帮助企业快速应对漏洞风险。
* 资产管理。支持对机器进行分组标签管理，基于组件识别技术，快速掌握服务器中软件、进程、端口的分布情况，帮助企业构建资产管理全景图。

## 6.2 异常处理

* 使用日志子系统记录与ERM运行相关的信息，方便系统管理员对系统运行的追踪，修正错误信息等。
* 当遇到系统无法处理的错误时，抛出异常并及时通知系统管理员。
* 使用JavsScript的异常处理机制，对特定的异常进行try-catch处理，在测试环境中将错误信息和其调用栈输出到控制台中；在生产环境中，保证用户不能受到异常的影响，其次不能让用户看到异常信息。
* 当网络发生异常时，或用户操作存在异常时，在醒目的位置提示异常原因。

## 6.3 缓存

### 6.3.1 软件缓存

* 针对浏览器端，对于一些非敏感数据或者难以伪造的数据（例如token），可以使用cookie保存在浏览器端。如果需要存储的数据比较大的话可以缓存在localStorage中，对于一些只在会话期间生效的则可以使用sessionStorage。而对于图片、CSS等资源文件，可以通过Cache-Control制定浏览器缓存这些资源的策略（过期时间等）。

### 6.3.2 硬件缓存

* 配置Redis缓存服务器。