



**信管综合实验报告**

**题 目：**个人知识管理系统

**学 院： 信息管理学院**

**专 业： 信息管理与信息系统**

**班 级： 信管2241**

**学 号： 22080078**

**姓 名： 骆鸿威**

**二○二五年六月**

**摘 要**

在当今这个信息爆炸的时代，每个人有许多数字化的知识与信息，随着知识的积累，信息的收集，传统的知识管理方式，如纸质文档管理或简单的电子文档存储，已经无法满足我们对知识进行高效检索、分类、关联和利用知识的需求。

本文通过研究互联网头部公司对开发文档、知识、代码的管理，揭示了个人在知识管理中存在的问题，如文档管理混乱、缺乏有效分类、搜索功能不足和知识关联弱等问题。分析了个人在知识管理方面可以进行借鉴的地方，如进行工程化管理、建立清晰的标签对文档实现分类、采用MySQL数据库实现全文搜索+模糊匹配、结合知识图谱对文档进行知识关联等。

**关键词：**知识管理、工程化管理、文档分类、全文搜索、知识图谱

# 概述

## 研究设计背景

知识管理系统是为了满足用户应对持续增长的、难以管理的知识与信息的管理工具。并在用户对自己的知识与信息进行上传、管理、关联的一系列活动中实现价值增长。

## 现有系统现状描述

本文从文档上传、文档查看、全文搜索、知识关联四个方面对现状进行描述。

1. 文档上传

文档上传主要给用户提供上传文件、选择标签（创建标签）的功能，服务端将根据用户创建的标签或选择的标签对文档进行分类。

1. 文档查看

文档查看主要给用户展示其创建或选择的标签列表，然后点击标签会展示该标签下的所有文档，再点击文档可以查看文档内容。

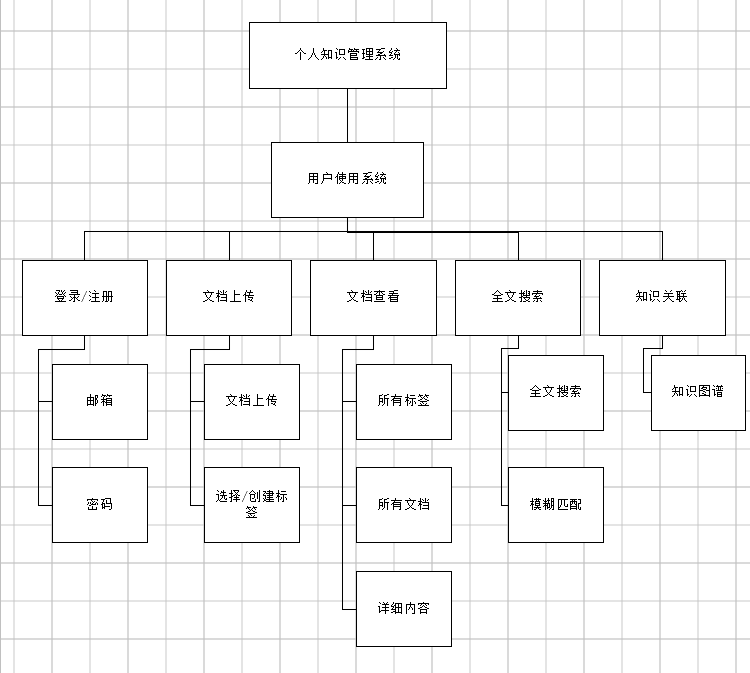
1. 全文搜索

全文搜索主要给用户提供在所有文档中进行搜索，包括标题的模糊匹配，文档内容的全文搜索等。

1. 知识关联

知识关联主要给用户提供将自己的不同文档通过某种方式进行联系，方便用户形成自己的知识网。

个人知识管理系统的组织结构图见图1。

**图1 个人知识管理组织结构图**

## 现有系统存在的问题

目前系统还存在以下问题：

1. 上传文件功能不全

目前只有单一文件上传功能，并没有文件夹上传功能，这对于在本地有许多电子文档的用户的体验感产生了极大的影响。

1. 支持文件类型较少

目前该系统只支持较少的几种文件类型，对于复杂的专业文件类型（如PS、CAD等专业文件）做不到精确解析。

1. 无法对上传的文件进行标签或内容修改

目前该系统不支持对已经上传的文档进行标签或内容修改，如果用户已经上传文档，但是需要修改文档的标签或内容，可能需要通过删除-重新上传等繁琐步骤，极大降低了用户的体验感。

# 需求分析

## 需求分析概述

通过体验互联网头部公司的云服务产品、研究其对应的工程化管理办法、结合自身体验等方式收集目前有关知识管理系统的存在的问题及创新需求点。此外针对现有系统存在的问题，通过评估其技术难度、成本、可行性等进行优先级划分，筛选掉不必要或难以实现的需求，从而对系统进行升级修改。

## 需求分析的任务和目的

1. 需求分析的步骤

识别现有系统存在的问题，上文已经对此进行阐述。

与其他现有相似的产品进行对比，从中发现问题，并进行优化。

通过网上论坛、博客寻找是否有对知识管理的新需求。

通过知网查找现有的知识管理的创新、模型、框架来发掘潜在需求。

综上结合实际整理后，根据系统开发设计的可行性进行分析，综合得出需要优化或新增的功能模块，更好地满足用户需求。

## 需求分析具体内容

1. 在上传文件时，提供多选功能

其主要目的在于减少本地存储较多文档用户的上传时间成本，以此来提高用户体验，留住用户。

1. 新增对现有文件的标签进行修改

其主要目的在于，免去了用户需要修改文档标签时，还需重新删除并且上传的繁琐步骤。

1. 新增对现有文件进行覆盖的功能

其主要目的在于，用户需要修改或更新现有文档内容时，可以通过选中文档，或根据文档id，重新上传文档以对其进行覆盖达到更新文档。

# 总体设计

## 总体设计概述

结合需求分析，在此对系统进行总体设计。总体设计包括系统设计和结构设计两个阶段.

系统设计阶段，本文结合数据流图构想所需要的系统的功能的实现的物理方案。

结构设计阶段，本系统遵循工程化、模块化的思想，根据模块间高内聚低耦合，设计控制层（接收请求并返回数据）、服务层（对业务数据进行逻辑处理等操作）、持久层（操作数据库返回数据）、数据模型层（将数据库抽象成Python代码以更好的操作和防止SQL注入）、utils层（所有工具类函数）。

权限角度，由于本系统只提高个人的知识管理功能，并未涉及与其他用户进行交互的功能，故只设计了用户权限。

功能模块角度，主要划分为文件功能、搜索功能两大功能模块，详情见图2



**图2 系统功能模块设计**

## 总体设计的目的和任务

总体设计的基本目的是回答系统“系统如何实现”的问题，本系统通过Python+Flask+Vue3+MySQL实现。任务是设计软件的结构。文档上传、文档查看、知识图谱是由文档模块实现的；全文搜索、模糊匹配是由搜索模块实现的。

## 系统设计的原则

1. 实用性原则

紧密围绕用户在知识管理方面的实际需求进行设计。鉴于现有系统存在上传文件功能不全、支持文件类型少、无法便捷修改文档标签和内容等问题，新系统设计着重解决这些实用性问题。如提供文件多选上传功能，减少用户上传时间成本；新增文档标签修改和内容覆盖功能，优化用户操作流程，切实满足用户对知识管理系统的基本使用需求，提升用户体验。

1. 可扩展性原则

充分考虑系统未来的发展和功能拓展。采用工程化、模块化的设计思想，将系统划分为控制层、服务层、持久层、数据模型层和 utils 层，各模块间保持高内聚低耦合。这种架构使得系统在后续开发中，无论是添加新的功能模块，还是对现有模块进行升级优化，都能相对独立地进行，降低模块间的相互影响，便于系统功能的扩展和维护。例如，当需要增加新的知识管理功能时，可以在不影响其他模块正常运行的情况下，在相应的功能模块中进行开发。

1. 性能优化原则

在系统设计中注重性能的提升。对于涉及数据查询和处理的功能，如全文搜索和模糊匹配，利用 MySQL 的相关功能进行优化。通过合理设计数据库索引、优化查询语句等方式，提高系统对大量文档数据的处理速度和查询效率，确保用户在使用搜索功能时能够快速获取所需信息，减少等待时间，提升系统的响应性能。

1. 安全性原则

尽管系统主要是个人知识管理，用户权限设计相对简单，但仍需重视数据的安全性。在数据模型层将数据库抽象成 Python 代码，一方面便于操作，另一方面可以有效防止 SQL 注入攻击，保护用户数据的安全。同时，在系统开发过程中，对用户上传的数据进行合理的存储和保护，防止数据泄露或损坏，确保用户知识资产的安全性。

1. 兼容性原则

考虑到用户可能会使用不同类型的文件进行知识管理，系统设计应尽量提高对多种文件类型的兼容性。虽然目前系统支持的文件类型有限，但在后续开发中，应逐步增加对复杂专业文件类型（如 PS、CAD 等）的解析和支持能力，确保用户能够顺利上传和管理各类文件，满足不同用户在知识管理方面的多样化需求。

## 系统架构的选择

1. B/S架构

B/S架构的技术壁垒较小，且有较成熟的社区论坛和框架论坛，开发难度小，功能扩展简单，故本系统选择B/S架构。

1. 后端框架：Python+Flask

Python：作为一种高级编程语言，Python 拥有简洁易读的语法，丰富的第三方库资源。在本系统中，其能便捷地实现各类复杂功能。同时，Python 对数据库的操作支持良好，为与 MySQL 数据库交互提供便利，能轻松实现数据的增删改查操作。

Flask：轻量级 Web 框架 Flask 灵活且易于扩展。它可以快速搭建后端服务，开发 RESTful API，处理前端传来的请求。比如，在文档上传功能中，Flask 接收前端的文件数据，调用相关处理函数后存储到指定位置，再将数据存入数据库。而且，Flask 支持使用蓝图进行项目管理，将不同功能模块的代码进行分离，如将用户认证、文档管理、搜索功能等分别放在不同的蓝图中，使项目结构清晰，易于维护和开发。

1. 前端框架：Vue3

Vue3 是流行的前端框架，具有高效的响应式系统和组件化开发能力。它可以构建交互性强、用户体验好的界面。在本系统中，Vue3 负责展示文档上传、查看、搜索等功能的界面。其组件化特性允许将界面拆分成多个可复用组件，像文档列表展示组件、搜索框组件等，提高开发效率和代码的可维护性。同时，Vue3 结合 Axios 库，方便与后端 Flask API 进行数据交互，实现前后端的数据传输和功能联动。

1. 数据库：MySQL

MySQL 是开源的关系型数据库，性能稳定、可靠性高。在本系统中，用于存储用户信息、文档元数据、标签数据以及知识图谱相关数据等。其支持事务处理，能确保数据操作的一致性和完整性，比如在更新文档标签时，保证数据的准确更新。并且，MySQL 提供的全文搜索和模糊匹配功能，能高效地实现系统的搜索需求，满足用户对文档快速检索的要求。

## 开发平台的选择

本实验采用Pycharm+VSCode+Navicat数据库可视化工具。Pycharm主要负责编写Python代码，VSCode主要负责前端Vue代码，Navicat主要负责数据设计及测试。

# 详细设计

## 一、代码设计

由于在本系统中，是由数据库的表来对文档、标签进行管理，故不特意对代码进行设计。

## 二、功能结构图设计

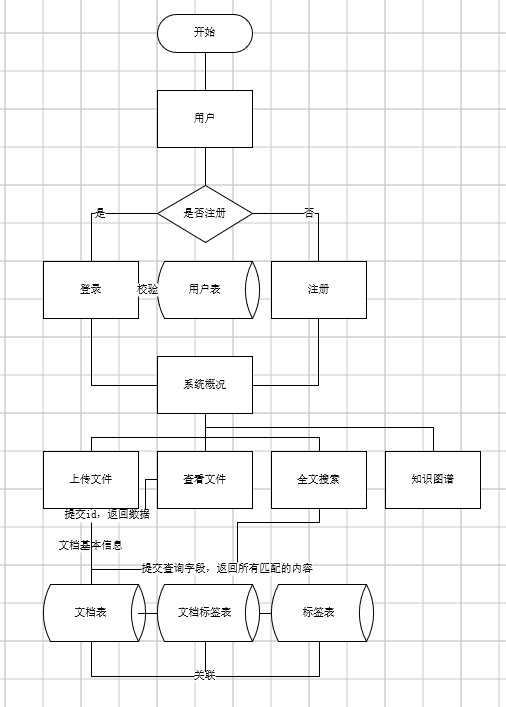
结合新增需求，功能结构设计图详见图3。



**图3 功能结构设计图**

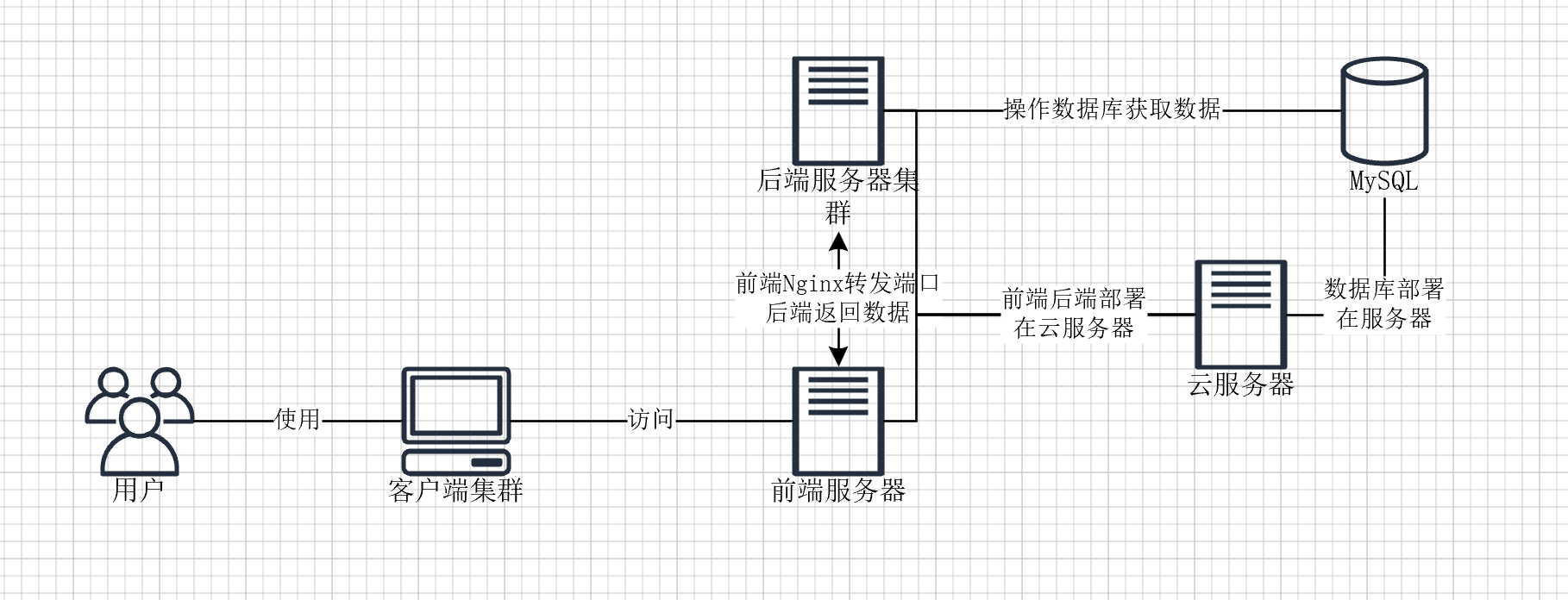
## 三、信息系统流程图设计

结合系统新增需求及用户使用流程，设计信息系统流程图如下，详见图4。



**图4 个人知识管理系统业务流程图**

## 系统物理配置方案设计

 由于自配服务器的购买成本与维护成本较高，目前本系统仅仅在开发阶段，仅在本机上进行测试，根据设想，本系统后期采用第三方云服务厂商的产品，将前后端服务器，数据库服务器部署在云服务器上，详情见图5。

**图5 个人知识管理系统网络拓扑图**

## 数据存储设计

结合本系统的需求，设计表：用户表、文档表、标签表、文档标签表四个表结构，具体内容如下：

1. 用户表：users

**表1 用户表**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 是否允许null |
| Id | 主键id | Int | N |
| Email | 邮箱 | Varchar | N |
| Password | 密码 | Varchar | N |
| Created\_at | 创建时间 | Datetime | Y |
| Loing\_ts | 登录时间 | Datetime | Y |

1. 文档表：documents

**表2 文档表**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 是否允许null |
| Id | 主键id | Int | N |
| Title | 文档标题 | Varchar | N |
| Content | 文档内容 | Varchar | Y |
| File\_path | 文档路径 | Varchar | N |
| File\_tag | 文档标签 | Varchar | N |
| User\_id | 该文档用户的id | Int | N |
| Upload\_time | 上传时间 | Datetime | N |
| Update\_time | 更新时间 | Datetime | N |

1. 标签表：tags

**表3 标签表**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 是否允许null |
| Id | 主键id | Int | N |
| Name | 标签名 | Varchar | N |
| User\_id | 创建用户的id | Int | N |

1. 文档标签表：document\_tags

**表4 文档标签表**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 说明 | 数据类型 | 是否允许null |
| Document\_id | 文档id | Int | N |
| Tag\_id | 标签id | Int | N |

## 输入设计

1. 用户注册/登录

输入内容：注册时，用户需输入邮箱、密码，确认密码（用于验证密码输入的准确性）以及验证码；登录时，输入邮箱和密码。

输入格式：邮箱采用标准的邮箱格式，如 “example@domain.com”，使用正则表达式进行格式验证；密码要求长度在 8 - 20 位之间，包含数字、字母（大小写）和特殊字符中的至少两种，同样通过正则表达式验证。

输入方式：在前端注册 / 登录页面，通过文本输入框让用户手动输入信息。输入框提供清晰的提示文本，如 “请输入您的邮箱”“请输入密码” 等。

1. 文档上传

输入内容：文件（支持多选以满足批量上传需求）、文档标题（读取上传文件的信息）、文档标签（可选择已有标签或创建新标签）。

输入格式：文件格式根据系统支持的类型进行限制；文档标题长度限制在 20 字以内，不允许包含特殊字符（如 “/”“\”“:” 等）；标签名长度限制在 10 字以内，同样不允许包含特殊字符。

输入方式：通过文件选择框选择本地文件，支持拖拽文件到指定区域进行上传；文档标题和标签通过文本输入框输入。对于标签输入框，提供自动完成功能，当用户输入时，自动匹配已有的标签并提示，方便用户选择。

1. 文档修改

输入内容：若修改文档标签，输入新的标签名或选择新的标签；若修改文档内容，需要重新上传文件以覆盖原来的内容。

输入格式：文件格式根据系统支持的类型进行限制；文档标题长度限制在 20 字以内，不允许包含特殊字符（如 “/”“\”“:” 等）；标签名长度限制在 10 字以内，同样不允许包含特殊字符。

输入方式：对于标签修改，在文档详情页面的标签管理区域进行操作；对于内容修改，选择需要覆盖的文档或输入其id，并重新上传文件进行修改。

1. 全文搜索

输入内容：搜索关键词。

输入格式：无特殊格式限制，但对关键词长度进行限制，如不超过 20 字，避免过长的关键词影响搜索性能。

输入方式：在搜索框中输入关键词，支持模糊搜索。

## 输出设计

1. 登录/注册反馈

输出内容：注册成功时，自动登录跳转到系统首页（系统概况）；若注册或登录失败，输出具体的错误原因，如 “邮箱格式错误”“密码错误”“该邮箱已被注册” 等。

输出格式：以弹窗形式在前端页面展示提示信息，对于错误信息，使用红色字体突出显示；对于成功信息，使用绿色字体显示。用户 ID 等信息可在用户个人中心页面展示，以表格或卡片形式呈现用户详细信息。

1. 文档列表展示

输出内容：文档标题、文档标签、上传时间、文档id（用于直观区分文件类型），对于用户有权限操作的文档，还应展示操作按钮（如查看详情、编辑、删除）。

输出格式：在前端页面以列表形式展示，每一行代表一个文档，每个文档的信息按照固定的列布局展示。使用表格进行布局时，设置合适的列宽以确保信息完整显示且排版美观；也可采用卡片式布局，每个卡片展示一个文档的信息，使页面更具可读性。

1. 文档详情内容查看

输出内容：文档详细的内容。

输出格式：使用GitHub开源的markdown展示库和其对应的CSS样式文件，将文件内容按照markdown格式美观的展示在前端。

1. 搜索结果展示

输出内容：匹配的文档标题、文档标签、文档上传时间，同样展示操作按钮（查看详情、编辑、删除）。

输出格式：以列表形式展示搜索结果，与文档列表展示类似，但对于匹配的内容片段，使用特殊颜色（如黄色）高亮显示关键词，方便用户快速定位相关信息。同时，在页面顶部显示搜索结果的数量和搜索所花费的时间。