# 个人知识库系统

**可行性分析**

****

**委托单位：软件需求分析课程**

**承办单位：G03**

# 版本记录

| **版本** | **修订日期** | **修订人** | **修订说明** | **修订状态** | **审批日期** | **审核人** | **批准人** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0.1 | 2021-3-13 | 吕博图 | 首次创建 | S | 2021-3-13 | 吕博图 | 吕博图 |
| 1.0 | 2021-3-26 | 岑盛泽 | 完善内容 | M | 2021-3-26 | 岑盛泽 | 吕博图 |

**修订状态：S--首次编写，A--增加，M--修改，D--删除；**

**日期格式：YYYY-MM-DD。**

**目录**

第1章 引言 3

1.1 编写目的 3

1.2 背景 3

1.3 定义 4

1.4 参考资料 4

第2章 可行性研究的前提 5

2.1 要求 5

2.2 目标 6

2.3 条件、假定和限制 6

2.4 进行可行性研究的方法 6

2.5 评价的尺度 7

第3章 对现有系统的分析 7

3.1 数据流程和处理流程 8

3.2 工作负荷 8

3.3 费用开支 8

3.4 人员 9

3.5 设备 9

3.6 局限性 9

第4章 所建议的系统 9

4.1 对所建议系统的说明 10

4.2 数据流程和处理流程 10

4.3 改进之处 10

4.4 影响 11

4.4.1 对设备的影响 11

4.4.2 对软件的影响 11

4.4.3 对用户单位机构的影响 11

4.4.4 对系统运行的影响 11

4.4.5 对开发的影响 11

4.4.6 对地点和设施的影响 11

4.4.7 对经费开支的影响 11

4.5 局限性 12

4.6 技术条件方面的可行性 12

第5章 可选择的其他系统方案 17

5.1 可选择的系统方案1 17

5.2 可选择的系统方案2 17

5.3 …… 18

1. 引言
   1. 编写目的

可行性研究目的是通过运用科学的方法对拟议中的工程项目进行全面的、综合的技术经济分析，来回答：本项目在技术上是否可行，经济上是否有生命力，财务上是否有利可图，需要多少投资，资金来源能否保证，建设周期多长，需要多少物力、人力资源等，进而判断本项目“行”还是“不行”;建设，还是放弃，以探讨各种具有实际意义的可能方案中遴选出最佳方案。

* 1. 背景

待开发软件系统的名称：

个人知识库系统

本项目的任务提出者：

杨枨老师和课程企业助教（陈幼安学长、陈炜舜学长）

开发者：

浙江大学城市学院SRA-2021-G03小组

用户：

杨枨老师、陈幼安学长、陈炜舜学长以及浙大城市学院学生

实现该软件的计算中心或计算机网络：

阿里云ECS轻量级应用服务器

* 1. 定义
  2. 参考资料

ISO9001 软件工程术语

ISO9001 计算机软件开发规范

ISO9001 计算机软件产品开发文件编制指南

ISO9001 计算机软件质量保证计划规范

硕士论文及文档：

案例库系统介绍-学院本科教学会议-2010-3-6

“基于项目的案例学习系统”学习评价模型的设计与实现-王明成

项目化案例对象的研究与设计-徐鹏飞

项目化案例对象的研究与设计 - 修改版

pbcls用户手册-2013-5-26

案例教学系统

王朝成-基于项目的案例学习系统-最终版

书籍资料：

《软件工程原书第八版》 机械工业出版社 RogerS.Pressman Bruce R.Maxim著 2017年1月第1版 第294545号

《软件工程导论》 清华大学出版社 张海藩等 2013年8月第6版 第150343号

《软件需求》 清华大学出版社 Karl Wiegers, Joy Beatty著 李忠利 李淳 霍金健 孔晨辉 译 2016年3月第3版

《UML用户指南》 人民邮电出版社 Grady Booch, James Rumbaugh, Ivar Jacobson著 邵维忠 麻志毅 马浩海 刘辉 译 2013年1月第1版

《UML2基础、建模与设计教程》 清华大学出版社 杨弘平等 2015年10月第1版

《IT项目管理》 机械工业出版社 Kathy Schwalbe著 孙新波 朱珠 贾建锋 译 2017年10月第1版

《PMBOK（项目管理知识体系指南）》第六版 Project Management Institute 著 电子工业出版社 2018年5月第1版

1. 可行性研究的前提
   1. 要求

系统在实现方式上是一种基于网页以及 APP的个人知识库管理系统。因此，对于用户来说，需要通过客户端进行个人知识库的管理，包括创建便签、正式文档，浏览文档，有合理、方便的整理文档的工具。同时，需要一个可以在各大主流平台上兼容的分享方式。对于网站管理员来说，应该能够批量管理用户，对于特定的的用户和文档应该也能单独管理。

作为一个个人知识库管理系统，这个项目的大致需求是，在客户端轻量的同时，能够方便地对文档管理、储存、共享。下面是对需求的描述：

管理员需求：

1. 对用户的CURD，包括批量和单独操作。
2. 对于文档的CURD，包括批量和单独操作。
3. 修改自身账户信息。
4. 能有渠道接收用户反馈。

用户需求：

1. 随时能够快捷地创建一个新的便签类文档。
2. 管理自己的文件(CURD以及建立索引)。
3. 搜索文档，反馈关键信息(大小，路径，修改时间等)。
4. 共享文档，支持批量打包。
5. 修改账户信息。
6. 文档应支持纯文本、doc、markdown等多种格式
   1. 目标

本系统在本质上是一个云端个人知识库，类似的应用有语雀、石墨文档等。

对于用户主体而言，使用本系统的目的在于通过该系统，方便的管理个人文档并且可以实现共享。

* 1. 条件、假定和限制
  2. 进行可行性研究的方法

经济可行性：

* 需要阿里云服务器一台，目前是阿里云学生服务器，一年价格为114元，五个开发人员的工资为51570元（按照成员在项目中平均工作150小时，平均工资69元）。

技术可行性：

本项目软件主程序编写的语言：Python，Java等。Flask框架实现微服务。

操作可行性：

通过墨刀进行界面设计，将部分界面外包给设计专业的学生，设计出符合现代人审美的界面。

* 1. 评价的尺度

优秀：

1.正确的获取需求

2.系统可正常运行

3.目软件的各项功能需求已得到实现

4.系统界面友好，易于交互。

5.系统易于更新优化，添加新的功能

合格：

1.系统可正常运行

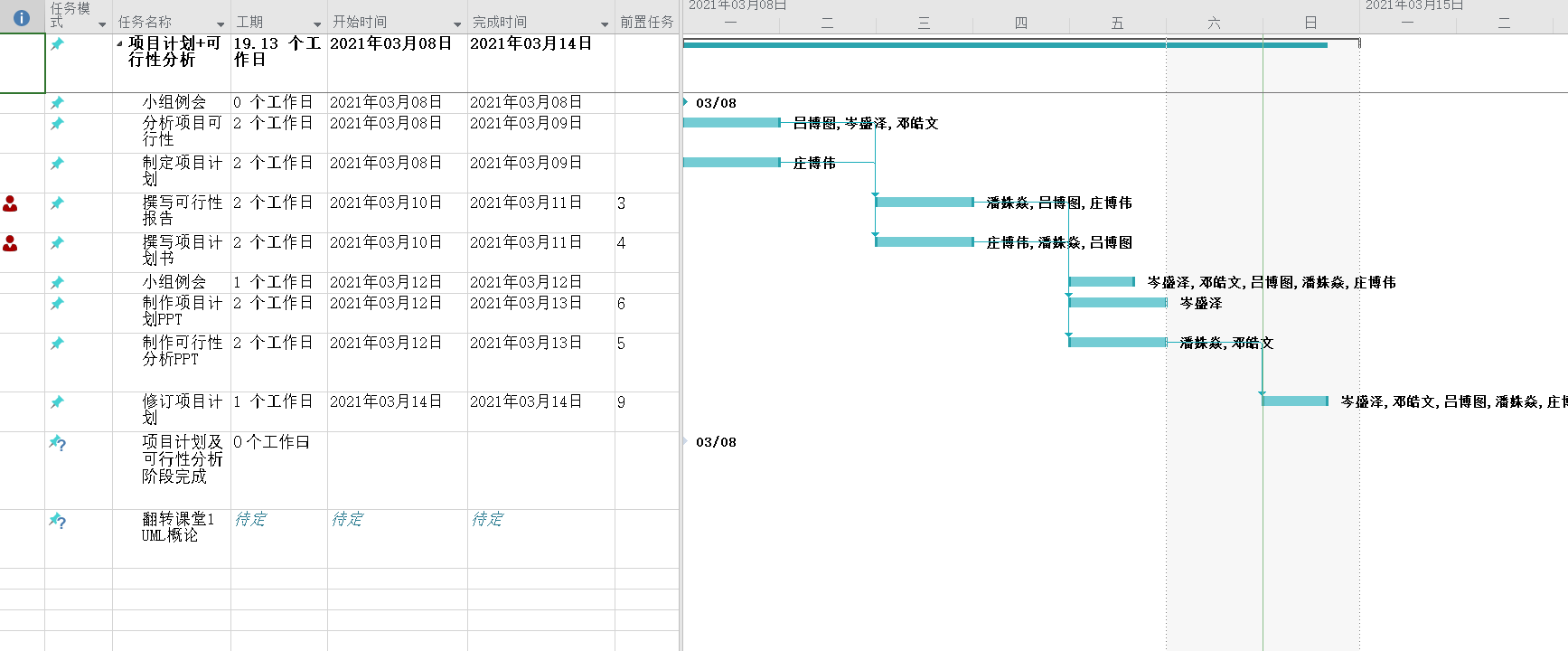
2.系统的各项功能需求的60%以上

不合格：

1.系统不能正常运行

2.不能实现60%以上的系统的各项功能需求。

1. 对现有系统的分析
   1. 数据流程和处理流程
   2. 工作负荷



* 1. 费用开支

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 项目名称 | 规格 | 单价（元） | 数量 | 单位 | 小计（元） | 备注 |
| 阿里云服务器 | 1核cpu  2G内存  1M带宽  40GB系统盘 | 114 | 1 | 个 | 114 | 作为系统使用的服务器 |
| 团建费用 |  | 400 | 5 | 次 | 2000 | 用于团队建设的物资购买 |
| 话费 |  | 0 | 1 | 个 | 0 | 用于组员之间互相联系 |
| 人工费 |  | 0 | 5 | 人 | 0 | 工作工资 |
| 资料购买 | 《IT项目管理》---[美]凯西·施瓦尔贝 | 78 | 5 | 本 | 390 | 作为自主学习资料 |
| 印刷 | 0元 | | | | | |
| 总计 | 2504元 | | | | | |

* 1. 人员

吕博图、岑盛泽、邓皓文、潘姝焱、庄博伟

* 1. 设备

Windows10系统电脑5台。

* 1. 局限性

成员项目经验不足可能会影响开发进度。

若团建未到位，人员配合问题以及积极性不足可能影响开发进度。

第4章 对所建议系统的说明

### 4.1.1优势（strength）

1. 针对用户明确，主要面向学校学生，需求会更加照顾特定用户，若系统功能完善，用户愿意使用该系统的可能性较高。可以“在红海中杀出一条血路”。
2. 已有语雀、石墨等优秀实例在先，项目完成的可能性较高。

### 4.1.2劣势（weekness）

1. 小组成员项目经验差距较大，在交流上需要多加注意。
2. 小组中的部分成员文档能力较弱，可能需要多加配合完成工作。

### 4.1.3机会（opportunity）

1. 本项目符合互联网+的时代潮流，在未来存在巨大的发展潜力。
2. 本项目在国内竞品较少，其核心功能尚未完善或不符合学生需求，而本系统针对用户明确，更容易拿下学生市场。

### 4.1.4威胁（threat）

1. 没有有效的方法获取大量经典有效的案例。

## 4.2 处理流程和数据流程。

建立案例

建立项目-加入项目流程

项目开发过程

## 4.3 改进之处

　　暂无

## 4.4 影响

暂无

### 4.4.1对设备的影响

暂无

### 4.4.2对软件的影响

需要搭建开发环境并配置服务器

### 4.4.3对用户单位机构的影响

暂无

### 4.4.4对系统运行过程的影响

暂无

### 4.4.5对开发的影响

暂无

### 4.4.6对地点和设施的影响

暂无

### 4.4.7对经费开支的影响

需要租用阿里云服务器，开通墨刀会员。

## 4.5 技术条件方面的可能性

### 4.5.1建站方式

#### 4.5.1.1 Python+Flask

因为小组主力开发成员熟悉Flask框架，所以对建站方式为Python+Flask

#### 4.5.1.2 最终选择的方案

#### 最终选择Python+Flask的方案

### 4.5.2系统的的运行环境：

**Windows server 2008 R2**

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * 是我们日常熟悉的操作系统，没有使用门槛 * 有原生图形界面 * 支持iis | * 稳定性不好 * Apache支持性不好 * 不是主流的PHP环境，教程较少 |
| 机会(Opportunities) | SO | WO |
| * 使用简单，无上手门槛 | -更快的推进项目进程 | -腾出时间对系统进行优化 |
| 风险(Threats) | ST | WT |
| * 与原开发环境不符 * 可能的搭建失败 * 系统性能不佳 | -预留充足的时间进行系统的适应修改 | -提高技术，对原有系统进行适应性修改  -正确面对问题，注意规避风险 |

**Ubuntu 18.04 LTS**

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * Apache支持好 * 主流PHP环境，教程容易找 * 系统稳定 | * 使用命令行来操作系统，对能力要求较高 |
| 机会(Opportunities) | SO | WO |
| * 系统小，适合作为服务端系统 * 系统性能很好 * 与原开发环境类似 | -充分利用Linux的支持性  -充分利用系统性能  -模拟原开发环境 | -改变原有的思维模式  -多使用命令行来操作系统  -少用图形界面 |
| 风险(Threats) | ST | WT |
| * 不熟悉的领域 * 可能的开发失败 | -对于不熟悉的领域积极的进行学习  -利用教程多的特点，积极学习 | -进行风险评估分析  -查找资料、教程，多学习 |

通过比较，我们可以发现Windows server对于Python的环境，性能不如Linux，而Linux系统虽然使用命令行，但作为主流服务器操作系统，网上教程丰富，而且我们也有一定的Linux学习基础，上手难度不大

综上所述，我们最后选择了Ubuntu 18.04 LTS

### 4.5.3数据库：

**MySQL**

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * 体积小、速度快 * 开源 * 与修习过的数据库课程使用的软件一样，比较熟悉 * 支持多种运行环环境 | * 不支持热备份 * 安全系统复杂而非标准 |
| 机会(Opportunities) | SO | WO |
| * 软件小，安装较为方便 * 开源，开发成本低 * 与原开发环境类似 * 熟悉，上手快 * 易移植 | -充分利用软件的熟悉  -节省成本  -模拟原开发环境  -利用易移植，加快环境搭建 | -以快速上手及低成本来降低风险可能造成的损失 |
| 风险(Threats) | ST | WT |
| * 备份不方便 * 安全性不佳 | -针对开源暴露的问题有效的规避 | -进行风险评估分析并制定针对措施 |

**MS SqlServer**

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * 有无可比拟的索引 * 集成性较好 * 对数据库文件加密，安全性好 * 对web技术支持好 * 自带用户图形界面 | * 开放性低 * 偏重于桌面应用NT server * 付费软件 * 与原系统运行环境不一致 |
| 机会(Opportunities) | SO | WO |
| * 对web技术支持好 * 图形界面使用简便 * 搜索速度快 | -充分利用图形界面便于操作的特性  -充分利用对web技术的支持性 | -利用对web技术支持好的特性  -利用索引提升系统性能  -使用旧版本的软件 |
| 风险(Threats) | ST | WT |
| * 只能用于windows系统 * 开发成本高 * 转换运行环境可能导致问题 | -利用图形界面便于操作的特性，快速解决运行环境变化带来的问题 | -进行风险评估分析  -查找资料、教程，多学习 |

通过SWOT分析，我们决定使用MySQL数据库。一是它与原系统运行环境一致；二是它开源免费；三是它跨平台支持好，小巧，易于安装搭建环境

而MS sqlserver决定了它只能运行与Windows操作系统，极大的限制了环境，也与原本环境不符，可能导致不可预料的问题

最终我们选择了MySQL

**个人设备操作系统：**

Windows10

优点：软件兼容全平台，安全性提升

缺点：容易出现未知错误

**虚拟机：**

Windows7

优点：BUG出现的几率小，软件的兼容性较好

缺点：Win7已经停止服务支持，安全性降低

Linux（Ubuntu 18.04 LTS）

优点：apache支持好，作为服务器性能优异

缺点：使用命令行操作系统，使用不够方便

**运行环境：**

**对于运行环境，我们尽量还原原本的运行环境，个别版本过旧无法找到的，选择最为相近的版本**

Pycharm

优点：熟悉的IDE，成熟的开发环境

缺点：非轻量级IDE，打开速度非常慢

费用：学生免费

**办公软件：**

Microsoft Office

优点：操作方便、稳定、跨平台

缺点：兼容性问题、付费使用

费用:499（包年）（包涵了Microsoft project）

Microsoft Visio 2016

优点：成熟，功能完善的画图工具

缺点：风格过于扁平化，不适合大型系统开发，付费使用

费用：95/月

Rational Rose

优点：UML建模工具，适合大型系统开发

缺点：从未接触过的软件，需要学习，已经停止支持，难以找到安装包

Rational RequisitePro

优点：成熟的需求管理工具

缺点：从未接触过的软件，需要学习，已经停止支持，难以找到安装包

DrawIO

优点：轻量级画图工具，有各类文档中常用图模板（UML,ER,流程图等）

缺点：在线版国内网络访问较慢。

**界面设计工具：**

Axure RP

优点：使用广泛，沟通、传输、修改方便，比较成熟的专业原型设计工具。功能一般比其他的软件较为完善，使用起来操作复杂度不会很高，上手容易

缺点：需要收费，不熟悉

费用：专业版 4000团队版 7000企业版 9000

墨刀

优点：使用方便，容易上手

缺点：需要收费

费用：专业版30一个月，199一年

# 5. 可选择的其他系统方案

## 5.1 可选择的系统方案1

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * 使用和推广比较方便 * 开发成本低，支持iOS和Android双平台，无需重复开发 * 消息容易推送 | * 原生小程序不支持文件上传功能 * 只能在手机设备上使用 |
| 机会(Opportunities) | SO | WO |
| * 微信普及度高，可以带动产品快速宣传 * 微信小程序开放的功能逐渐增加 | -充分利用微信的普及度在高校中快速宣传 | -积极适配小程序新功能，增加用户粘性。 |
| 风险(Threats) | ST | WT |
| * 手机端受限于屏幕大小，对文档的处理较为不方便 * 微信小程序限制较多，受限于微信已开放的功能接口。 | -增加文档操作系统中的快捷方式，方便用户修改文档。 | -开发云端文档修改系统，避免本地文件操作的权限问题。  -通过接入外部框架来实现小程序需要的功能。 |

## 5.2 可选择的系统方案2

|  |  |  |
| --- | --- | --- |
| 内部能力  外部因素 | 优势(Strength) | 劣势(Weakness) |
| * 使用和推广比较方便 * 消息容易推送 | * 只能在手机上使用 * 需要开发Android和iOS双平台 * IOS对于文件系统管控较为严格，本地文档相关操作开发困难 * Android版本众多，适配困难。 |
| 机会(Opportunities) | SO | WO |
| * 智能手机普及率高，且用户对手机的依赖性较强。 | -充分例用手机app的便携性在高校进行宣传推广 | -花费一定的时间对主流设备进行适配 |
| 风险(Threats) | ST | WT |
| * 手机端受限于屏幕大小，对文档的处理较为不方便 * Android系统操作权限逐渐收缩，未来本地文档操作相关权限可能会收到限制 | -增加文档操作系统中的快捷方式，方便用户修改文档。 | -开发云端文档修改系统，避免本地文件操作的权限问题。 |

## 5.3 最终选定的方案

最后选择网站以及安卓app方式，通过Flask框架搭建后端服务，并用墨刀设计原型界面。