|  |  |  |
| --- | --- | --- |
| 文档状态 | 保密级别 | 中 |
| [ ]草稿  [√ ]修订  [ ]发布 | 文档编号 | G03-CZX-1 |
| 管理部门 | G03组 |
| 修订年月 | 2024/3/16 |
| 版本号 | V1 |

|  |  |  |
| --- | --- | --- |
| **修订人签字** | **审核人签字** | **批准人签字** |
| 日期： | 日期： | 日期： |

**超算中心运营管理门户网站需求工程项目计划**

变更履历

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **序号** | **变更日期** | **版本** | **变更位置** | **变更原因** | **修订人** | **审核人** | **批准人** |
| 1 | 2024/3/16 | V1 | [Luojh1/Luojh.github.io: self-use](https://github.com/Luojh1/Luojh.github.io) | 建立初稿 |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

说明：“变更原因”主要是分为：

1. 建立初稿
2. 内容修订
3. 正式发布

目录

[1. 项目概述 4](#_Toc161583005)

[1.1. 项目说明 4](#_Toc161583006)

[1.2. 项目背景 4](#_Toc161583007)

[1.3. 项目目标 4](#_Toc161583008)

[1.4. 项目范围 4](#_Toc161583009)

[1.5. 项目的假设与限制 4](#_Toc161583010)

[2. 项目需求 4](#_Toc161583011)

[2.1. 项目需求的分析和整理 4](#_Toc161583012)

[2.2. 项目需求的优先级和范围管理 4](#_Toc161583013)

[2.3. 需求规格说明 4](#_Toc161583014)

[3. 项目任务分解 4](#_Toc161583015)

[3.1. WBS（Work Breakdown Structure） 4](#_Toc161583016)

[3.2. 关于WBS中每项任务的说明 4](#_Toc161583017)

[4. 项目进度安排 4](#_Toc161583018)

[4.1. 每项任务时间的预估 4](#_Toc161583019)

[4.2. 甘特图和分析说明 4](#_Toc161583020)

[4.3. 网络图和分析说明 4](#_Toc161583021)

[5. 团队分工 4](#_Toc161583022)

[5.1. OBS图（组织分解结构） 4](#_Toc161583023)

[5.2. 分工安排 4](#_Toc161583024)

[6. 项目干系人 4](#_Toc161583025)

[6.1. 项目干系人分析 4](#_Toc161583026)

[6.2. 利益相关者分析 4](#_Toc161583027)

[7. 项目的子计划 4](#_Toc161583028)

[7.1. 风险 4](#_Toc161583029)

[7.2. 人力资源 4](#_Toc161583030)

[7.3. 预算 4](#_Toc161583031)

[7.4. 开发计划 4](#_Toc161583032)

[7.5. 测试计划 4](#_Toc161583033)

[7.6. 部署与上线计划 4](#_Toc161583034)

[7.7. 维护计划 4](#_Toc161583035)

[8. 风险管理 5](#_Toc161583036)

[8.1. 可能存在的风险 5](#_Toc161583037)

[8.2. 风险评估 5](#_Toc161583038)

[8.3. 风险的应对计划 5](#_Toc161583039)

[8.4. 风险的管控 5](#_Toc161583040)

[9. 预算说明 5](#_Toc161583041)

[9.1. 预算组成 5](#_Toc161583042)

[9.2. 预算估计 5](#_Toc161583043)

[9.3. 预算控制与管理 5](#_Toc161583044)

[10. 需求变更及跟踪管理 5](#_Toc161583045)

[10.1. 线性责任表 5](#_Toc161583046)

[10.2. 设置通知和提醒 5](#_Toc161583047)

# 项目概述

## 项目说明

超算中心运营管理门户网站作为公共服务平台，为全校师生提供统一的在线服务平台。具体包括新闻动态以及通知公告的发布、超算平台账号在线申请、充值计费管理、超算平台用户手册查询、超算中心制度与政策查询、成果奖励申报等。

## 项目背景

随着科学技术的不断发展，超级计算成为推动科学研究、工程设计和创新发展的关键力量。2021年，国家发改委等四部门联合发布了《全国一体化大数据中心协同创新体系算力枢纽实施方案》，明确提出布局全国算力网络枢纽节点，启动实施“东数西算”工程。2023年2月，《数字中国建设整体布局规划》提出系统优化算力基础设施布局，促进东西部算力高效互补和协同联动，引导通用数据中心、超算中心、智能计算中心、边缘数据中心等合理梯次布局。并且随着云计算技术水平发展，AI、5G、物联网技术的成熟和推广，智慧政务、智慧能源、智慧金融、生物医药、工业仿真、无人驾驶、航空遥感等应用场景的高性能计算需求越来越多。为了给用户提供高效、便捷的高性能计算资源服务，构建面向超算中心的统一运营管理平台尤为重要。基于该平台，用户可以快速了解整个超算中心的算力资源配置情况、计费方式以及平台使用方法，实时获取超算中心最新的动态信息。一键式完成账号申请、充值结算以及成果奖励申报等流程。既便利了用户零门槛的使用超算中心的计算资源，也为超算中心管理人员提供了很好的平台管理方式。

我校已建成浙江省内公办高校首家校级超算中心，为我校师生提供支撑科学研究的信息化基础设施，打造支撑学校交叉创新研究和产业创新的基础平台，成为学校的金名片。作为公共服务平台，应建立运营门户网站为全校师生提供统一的在线服务平台。

## 项目目标

目前有很多的门户网站，但是面向超算中心的门户网站较少，同时由于每个学校的超算中心的情况都不一样，需要针对自身情况设计相应的运营门户网站。该网站建设完成后，主要为学校师生提供超算平台账号在线申请、账号充值以及成果奖励申报等功能。同时也为超算中心运营管理人员提供新闻动态以及通知公告的发布，账号收费管理以及成果管理等功能。

该网站的开发将在学期结束前完成，开发资源有：5个合作愉快的人员；dreamwaver、photoshop、project,officetools和其他必备的软件和硬件，如有余力可以完成对应移动版app的开发。

## 项目范围

最终成品：超算中心运营管理门户网站、相应的后台、使用手册、SRS（需求规格说明书）

## 项目的假设与限制

成功地开发该网站，我们首先得得到教师和超算中心的支持和认可；还需要得到教师，同学以及运营管理人员的高度配合；需要有的软件有：dreamwaver、rationalrose、officetools、photoshop,project和可以上网的电脑。

这个网站的实现方法将和其他的网站一样，没有特殊的技术。网站的范围是：1.信息发布；2.资料下载；3.账号申请；4.账号充值；5.成果奖励申报等。不再另外开设可供教师和学生使用的邮箱，如有邮件都将使用个人自己在其他网站上的邮箱。

# 项目需求

## 项目需求的分析和整理

#### 2.1.1 用户需求

该项目的主要用户为学校师生、运营管理人员以及其他对超算中心感兴趣的人员，：

学校师生的需求：

A、网站上要有超算中心的简介，包括组织架构、发展历程以及联系方式等。

B、网站上要有超算中心软硬件资源的简介。

C、网站上要有超算中心的各项管理制度。

D、网站上要有超算中心日常的新闻动态以及通知公告。

E、网站上要提供用户账号申请、账号算力充值与计费、账单查询等功能。

F、网站上要提供基础的平台使用教程以及常见问题解答。

G、网站上要提供用于科研成果奖励的在线申请入口。

运营管理人员的需求：

A、网站要支持管理员实时发布新闻动态和通知公告。

B、网站要支持管理员对于用户账号、账号充值以及账单管理。

C、网站要支持管理员发布各项管理制度和流程政策。

D、网站要支持管理员对用户申报的科研成果进行管理。

该网站的目的主要是为用户提供一个便捷的使用超算中心算力服务的入口，方便老师和学生较低门槛的使用平台资源，也为超算中心运营管理人员提供一个对用户使用情况进行管理的平台。

师生能够快速了解整个超算中心的算力资源配置情况、计费方式以及平台使用方法。

师生能够实时获取超算中心最新的动态信息。

师生能够一键式完成账号申请、充值结算以及成果奖励申报等流程。

管理员能够对用户进行统一的管理，包括账号管理、计费管理以及成果管理等。

游客也可以通过该网站快速了解我校超算中心的整体情况。

#### 2.1.2 业务需求

服务目标主要是全校师生和校外的一些需要算力的企业。业务流程是用户通过登录账号（若没有账号要先通过手机号或者邮箱注册），确认账户里有金额后，提交实例和相关数据，若当前超算中心能承载完成这个实例的计算，那就开始计费，若不能，需要告知用户原因，是因为目前超算中心运载的实例过多，需要排队，还是这个实例太复杂，超算中心无法完成，或者是因为用户账号的余额不足。计费的方式选择按分钟收费，具体价格参照校方给出的意见。若这个实例在超算中心的运行下取得了一定的研究成果，可以向官方提交成果奖励申请，如果经过官方审核通过，会在对应的账户里充值算力，审核最终解释权归校方所有。

#### 2.1.3 功能需求

1.提供账号的申请

2.提供账号的算力充值服务

3.提供账号的账单查询

4.有能上传实例和数据的入口

5.有能获得实例和数据结果的出口

6.算力的计费功能

7.管理员对账号、计费功能、科研成果的管理功能。（增删改查的功能都要实现）

8.用户的成果申报功能

9.管理员的成果审核功能。

#### 2.1.4非功能需求

性能需求：涉及系统的响应时间、吞吐量、并发性能等。系统的响应时间不超过2s，至少保证能并行处理300个请求。

可靠性需求：包括系统的可靠性、可用性、容错性、可恢复性等。系统需要能连续，稳定的运行，不能老是报错。遇到出错和崩溃的情况，应该将出错的情况，详细记录到日志文件中，方便恢复。

安全需求：涉及系统的安全性、数据保护、身份验证、授权等。做好网络安全防护，前端和后端代码都要有验证功能。

可维护性需求：网站需要可以维护。

可用性需求：包括系统的易用性、用户界面设计、可访问性等。网站系统要做到UI简洁，功能突出，有操作过程的示例。

界面需求：涉及系统与外部系统的接口、数据交换格式、通信协议等。用httpserver，数据交换格式以json格式为主，通信协议用https。

法律和法规需求：确保系统的合规性、隐私保护、数据安全等。审核人员需要判断用户提交的请求是否是一个合法的行为。

环境需求：考虑系统的部署环境、硬件要求、软件依赖等。要能在Linux、Windows、MAC OS上稳定运行

成本需求：包括系统的开发成本、运营成本、维护成本等。依照校方给出的具体方案。

## 项目需求的优先级和范围管理

项目成员都需要使用PingCode，（PingCode上建立了需求跟踪矩阵）在进入到PingCode的对应组织中，你可以看到你负责的项目需求，有对应的优先级，截止时间，以及需求的详细说明。若后续有需求变动，你在PingCode上也会收到通知。

## 需求规格说明

审查需求规格说明书：项目团队和相关干系人（如客户）会仔细审查需求规格说明书，检查其内容的准确性、完整性和一致性。这包括检查需求是否清晰、无歧义，是否覆盖了所有必要的业务场景，以及需求之间是否存在冲突或不一致之处。

验证需求的可行性和可实现性：团队会评估需求的技术可行性、经济可行性和操作可行性。这涉及到确定是否存在技术难题、成本是否合理以及操作流程是否顺畅等。此外，还需要确认是否有足够的技术资源和时间来实现这些需求。

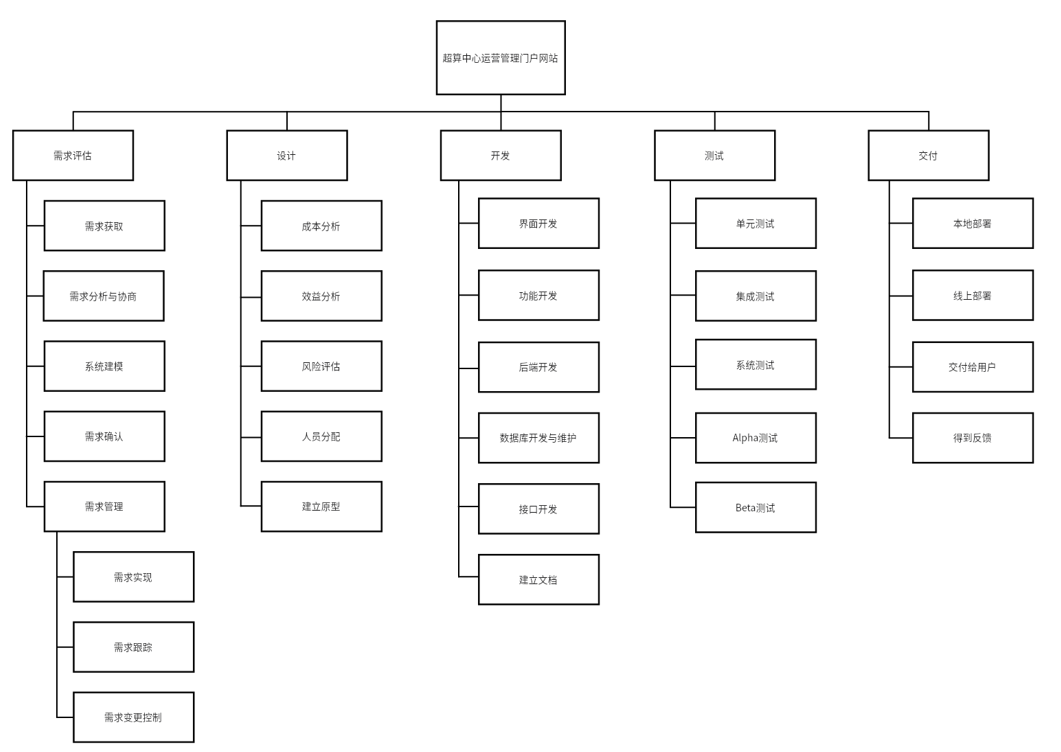
收集干系人反馈：干系人（特别是最终用户和关键利益相关者）会对需求规格说明书提供反馈。这有助于确保需求符合他们的期望和需求，同时也有助于发现和解决可能存在的问题或遗漏。

进行变更管理和跟踪：如果在确认过程中发现了问题或需要变更，团队会记录这些变更，并更新需求规格说明书。同时，还需要确保所有变更都得到了适当的批准和跟踪，以保持需求的一致性和完整性。

编写确认报告：完成确认过程后，团队会编写一份确认报告，总结确认活动的结果、发现的问题以及建议的改进措施。这份报告将作为项目文档的一部分，供后续开发和维护工作参考。

# 项目任务分解

## WBS（Work Breakdown Structure）



## 关于WBS中每项任务的说明

超算中心：

概述：作为核心部分，超算中心负责整个项目的资源管理和技术支持。

需求评估：

需求获取：收集并整理各方对超算中心运营管理门户网站的具体需求。

成本分析：分析项目实施所需成本，包括人力、物力等方面的投入。

需求分析与协商：深入分析获取到的需求，并与相关方进行沟通和协商，确保需求的准确性和可行性。

效益分析：评估项目实施后可能带来的经济效益和社会效益。

设计开发：

界面开发：设计并开发用户友好的门户网站界面。

功能开发：根据需求实现网站所需的各种功能。

后端开发：负责网站的后端逻辑处理和数据交互。

数据库开发与维护：设计和维护网站的数据库系统，确保数据的安全和高效处理。

测试交付：

单元测试：对网站的各个功能模块进行单独的测试，确保功能的正确性。

集成测试：将各个功能模块集成后进行整体测试，确保系统各部分的协同工作。

Alpha测试：在小范围内进行实际环境测试，收集用户反馈，优化网站。

Beta测试：扩大测试范围，进行更大规模的测试，进一步完善网站。

系统测试：对整个系统进行全面的测试，确保网站的质量和稳定性。

交付给用户：将测试通过、符合要求的网站正式交付给用户使用。

此外，还有一些辅助性项目如：

需求管理：对整个项目需求进行管理和跟踪。

需求确认：确保项目需求得到准确理解和确认。

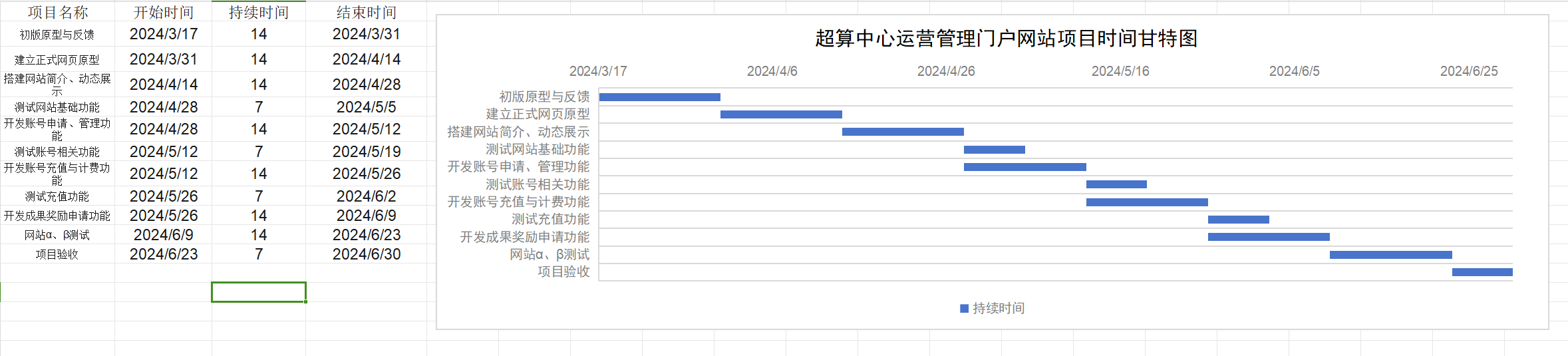
人员分配：根据项目需求进行人员配置和任务分配。

建立文档：对整个项目过程中的文档进行整理和归档。

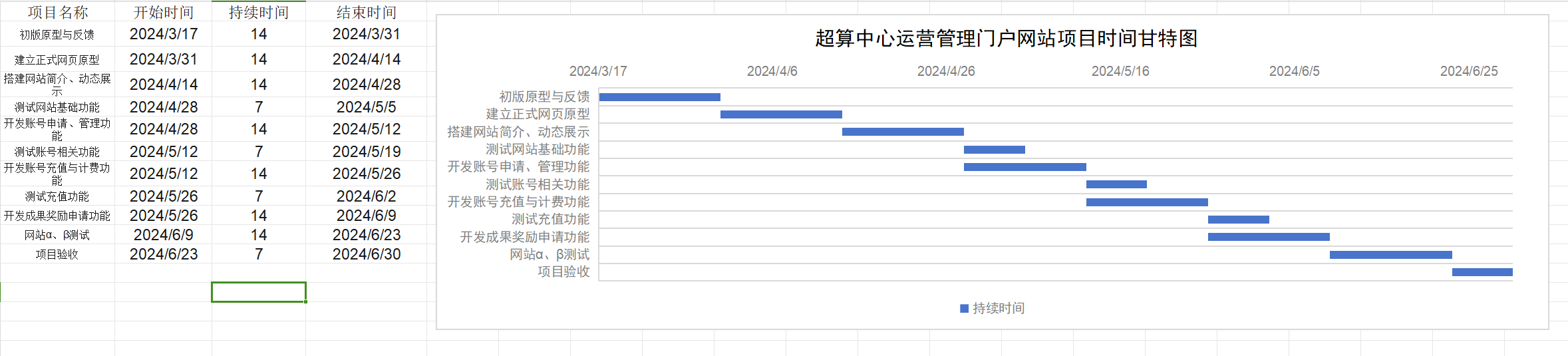
需求跟踪与变更控制：跟踪项目需求的变更，并进行相应的控制和管理。

# 项目进度安排

## 每项任务时间的预估

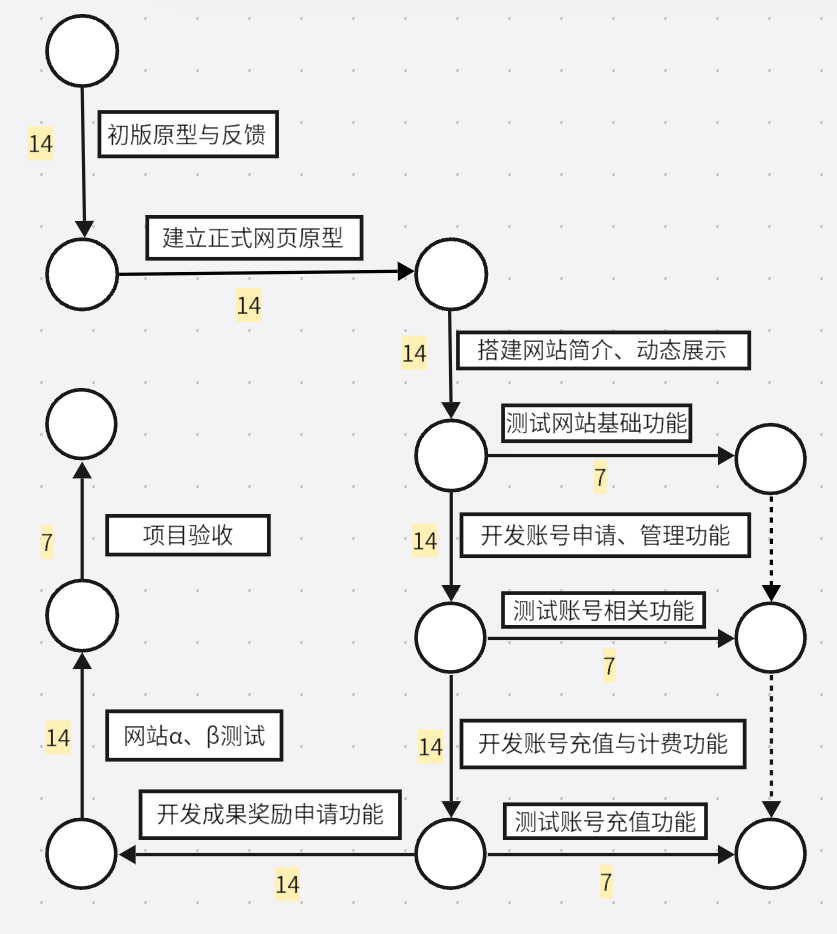


## 甘特图和分析说明



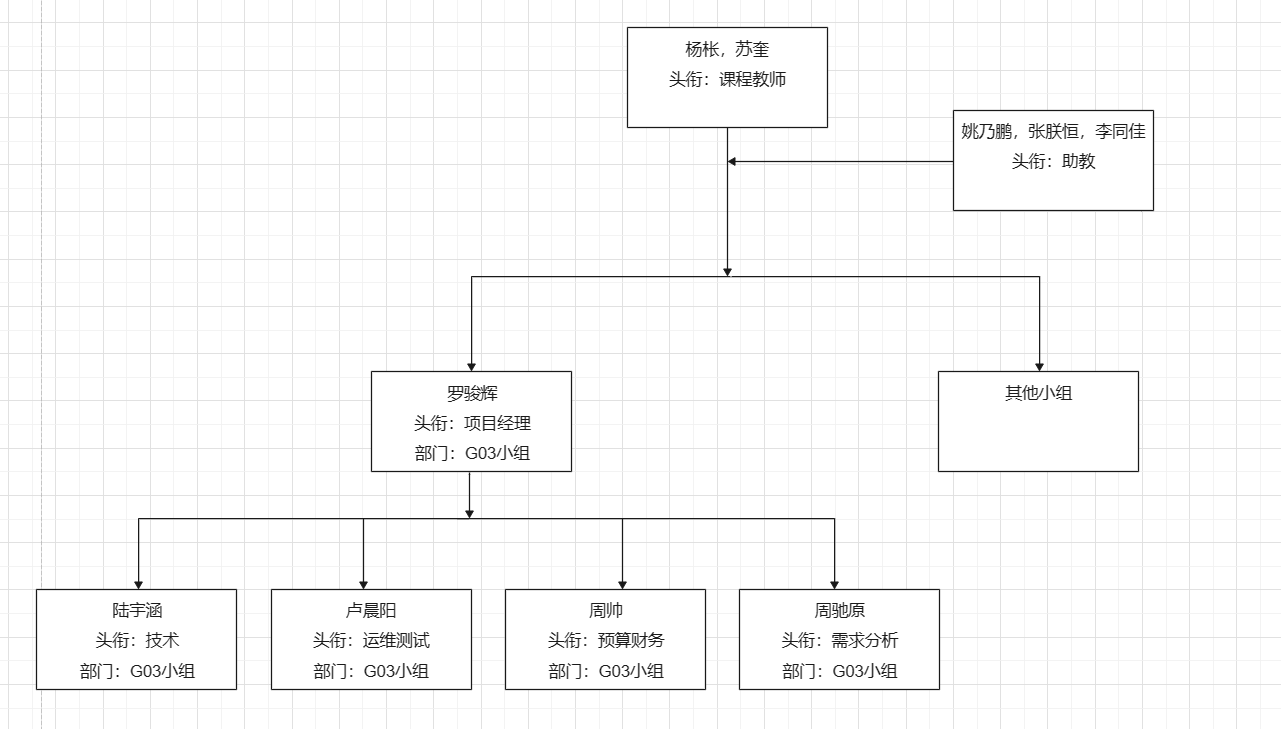
我们团队是按两个礼拜一个里程碑事件来做的。

## 网络图和分析说明



# 团队分工

## OBS图（组织分解结构）



## 分工安排

人员分工

罗骏辉

主要负责设计阶段，在其他阶段帮助小组成员完成

需要完成项目可行性分析，需求工程项目计划

周驰原

主要负责需求评估阶段，在其他阶段帮助小组成员完成，完成项目章程

需要完成项目的甘特图

陆宇涵

主要负责开发阶段，在其他阶段帮助小组成员完成

需要完成wbs

卢晨阳 主要负责测试阶段，在其他阶段帮助小组成员完成

需要完成obs

周帅 主要负责交付阶段，在其他阶段帮助小组成员完成

需要完成预算管理

# 项目干系人

## 项目干系人分析

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. 序号 | 名称 | 干系人角色 | 干系人在项目中的利益 | 影响 | 管理策略 |
| 1 | 指导老师 | 监督方与协助者 | 评价学生成绩 | H | 满足其要求，按时按规则提交项目进度 |
| 2 | 助教 | 监督方 | 提高成绩 | L | 提交项目进度时要抄送邮件 |
| 3 | 罗骏辉 | 项目组长 | 获得成绩 | H | 保持沟通，帮助完成项目任务 |
| 4 | 陆宇涵 | 项目成员 | 获得成绩 | H | 保持沟通，帮助完成项目任务 |
| 5 | 卢晨阳 | 项目成员 | 获得成绩 | H | 保持沟通，帮助完成项目任务 |
| 6 | 周驰原 | 项目成员 | 获得成绩 | H | 保持沟通，帮助完成项目任务 |
| 7 | 周帅 | 项目成员 | 获得成绩 | H | 保持沟通，帮助完成项目任务 |
| 8 | 其他小组 | 评审者 | 完成任务 | L | 完成展示的ppt |
| 9 | 浙大城市学院 | 出资方 | 发展学校 | H | 满足其要求 |
| 10 | 超算中心负责人（杨老师） | 主要需求提出者 | 获得管理门户网站 | H | 满足其需求 |
| 11 | 全体师生 | 使用者 | 使用 | L | 思考其可能的需求 |

注：H表示HIGH，L表示LOW

## 6.2. 利益相关者分析

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 序号 | 名称 | 影响 | 利益 | 参与程度 | 期望 | 策略 |
| 1 | 指导老师 | H | L | 决策 | 希望学生根据课程规则完成项目 | 满足其要求，按时按规则提交项目进度 |
| 2 | 助教 | L | L | 被通知 | 帮助指导老师管理 | 提交项目进度时要抄送邮件 |
| 3 | G03小组 | H | H | 参与 | 完成项目，获得成绩 | 保持沟通，帮助完成项目任务 |
| 4 | 其他小组 | L | L | 被通知 | 完成评审 | 完成展示的ppt |
| 5 | 浙大城市学院 | L | H | 提高参考意见 | 完成项目，发展学校 | 满足其要求 |
| 6 | 超算中心负责人（杨老师 | H | H | 提高参考意见 | 获得管理门户网站 | 满足其需求 |
| 7 | 全体师生 | L | L | 被通知 | 使用门户网站 | 思考其可能的需求 |

注：H表示HIGH，L表示LOW

# 7. 项目的子计划

## 风险

1.技术风险：技术实现难度大，特别是算力充值与计费、成果奖励申请等功能的实现可能存在技术难点。

应对措施：采用原型开发模型，通过多轮原型反馈进行需求确认，确保技术的可行性和稳定性。

2.项目管理风险：项目进度可能受到各种因素的影响，导致延期或超出预算。

应对措施：制定详细的项目计划，并进行定期审查和调整。确保项目团队成员对项目进度和预算有清晰的认识，并及时沟通和协调。

3.用户需求变更风险：在开发过程中，用户的需求可能会发生变化，导致已完成的工作需要重新调整。

应对措施：在项目开始前进行充分的需求调研，并与用户进行深入的沟通和讨论。同时，采用敏捷开发方法，以适应用户需求的变更。

4.数据安全与隐私风险：超算中心涉及大量敏感数据，如账号信息和计算数据，可能面临数据泄露或损坏的风险。

应对措施：加强数据安全管理，采取加密存储、访问控制等手段保护用户数据。同时，建立数据备份和恢复机制，确保数据的完整性和可用性。

5.法规合规性风险：超算中心的运营可能涉及相关的法律和法规要求，如知识产权保护等。

应对措施：了解并遵守相关法律法规的要求，确保超算中心的合法合规运营。

6.人员素质风险：项目团队成员的技能和经验可能无法满足项目的需求，导致工作效率低下或质量不佳。

应对措施：对团队成员进行必要的培训和技能提升，确保他们具备完成任务的能力。同时，进行合理的任务分配和团队建设，提高团队的整体素质和协作能力。

## 人力资源

1.项目管理团队：

职责：负责整体的项目规划、进度控制和资源协调。监控项目进展，并处理可能出现的问题。

人员配置：1名项目经理，具备良好的组织和沟通能力，有丰富的项目管理经验。

2.需求分析与设计团队：

职责：与超算中心负责人（杨老师）沟通确认需求，完成网站的用户界面设计和系统架构设计。

人员配置：1-2名需求分析师，能够准确理解用户需求；1-2名UI/UX设计师，具备良好的设计思维和创意能力。

3.软件开发团队：

职责：根据需求分析结果进行软件编码，实现网站的各项功能。

人员配置：2-4名软件工程师，具备前端和后端开发能力，熟悉至少一种编程语言（如Java、Python等）。

4.测试与质量保证团队：

职责：负责网站的测试工作，包括功能测试、性能测试等，确保软件质量。

人员配置：1-2名软件测试工程师，具备严谨的逻辑思维和细致的工作态度。

5.运维支持团队：

职责：负责网站的日常运行维护和支持，包括服务器管理、网络安全防护等。

人员配置：1-2名系统管理员/网络管理员，具备相关的技术背景和实践经验。

6.培训与教育团队：

职责：负责编制培训材料，为超算中心的用户提供使用指南和培训课程。

人员配置：1名教育专员，具备良好的沟通能力和教学技巧。

7.市场推广与宣传团队：

职责：负责网站的推广工作，提升超算中心的品牌知名度。

人员配置：1-2名市场营销人员，具备相关的市场分析和推广能力。

## 预算

1.人力资源成本：

包括项目经理、需求分析师、UI/UX设计师、软件工程师、测试工程师、系统管理员等人员的工资和福利。这是项目的主要开支，需要根据团队规模和项目期限来预算。

2.硬件设备成本：

购买或租赁服务器、存储设备、网络设备等必要的硬件资源，以及相应的维护和升级费用。

3.软件许可成本：

如果使用商业软件或第三方服务，需支付相应的许可费用。

4.开发工具和环境成本：

包括开发工具、测试工具、版本控制系统等软件的购买或订阅费用。

5.培训与教育成本：

编制培训材料、组织培训课程以及进行用户教育的相关费用。

6.市场推广与宣传成本：

网站推广、品牌宣传等相关的市场活动费用。

7.运营成本：

包括服务器托管费用、网络带宽费用、数据备份和恢复服务费用等。

8.风险预留金：

为应对可能出现的风险和不确定性，应设立风险预留金，一般占总预算的一定比例。

9.项目管理费用：

包括项目进度跟踪、质量控制、风险管理等项目管理活动的相关费用。

10.法律和合规费用：

如果涉及到相关法律和合规事项，可能需要支付律师费、合规审查费等。

## 开发计划

1.需求分析与设计阶段（1-2个月）：

主要任务：与超算中心负责人（杨老师）沟通，明确网站的主要功能和用户群体，编制需求分析报告。

交付物：需求规格说明书、功能列表、简易草图等。

2.技术选型与原型开发阶段（2-3个月）：

主要任务：根据需求分析结果，选择合适的技术框架和开发工具。采用原型开发模型，进行多轮原型反馈，逐步完善网站的功能和界面。

交付物：原型设计图、技术选型报告等。

3.编码与单元测试阶段（3-4个月）：

主要任务：软件开发团队根据原型和需求规格说明书进行编码，完成网站的各项功能。同时进行单元测试，确保代码质量。

交付物：源代码、单元测试报告等。

4.集成测试与系统测试阶段（1-2个月）：

主要任务：对网站进行集成测试和系统测试，检查是否存在功能缺陷或性能问题。测试团队负责编写测试用例，记录测试结果。

交付物：测试报告、问题追踪表等。

5.用户验收测试与反馈阶段（1个月）：

主要任务：邀请部分最终用户参与用户验收测试，收集用户的反馈意见，对网站进行相应的调整和优化。

交付物：用户验收测试报告、用户反馈汇总等。

6.部署与上线阶段（半个月）：

主要任务：在通过用户验收测试后，进行网站的部署和上线工作，包括服务器配置、数据库迁移、安全设置等。

交付物：部署手册、上线计划等。

## 测试计划

1.测试策略制定（时间：1-2周）

目的：确定测试的范围、方法、资源和时间表。

活动：审阅需求文档，与开发团队沟通，明确测试的重点和难点。

输出：测试策略文档。

2.测试环境搭建（时间：1-3周）

目的：建立稳定可靠的测试环境，模拟真实的运行环境。

活动：配置测试服务器、数据库、网络等。

输出：测试环境搭建报告。

3.测试用例设计（时间：2-4周）

目的：设计详细的测试用例，覆盖所有功能点和业务场景。

活动：编写测试用例，包括正常流程测试、异常流程测试、性能测试、安全测试等。

输出：测试用例文档。

4.测试执行（时间：4-6周）

目的：按照测试用例执行测试，发现并记录软件的缺陷。

活动：执行测试用例，记录测试结果，提交缺陷报告。

输出：测试执行报告、缺陷报告。

5.缺陷修复与回归测试（时间：3-5周）

目的：修复发现的缺陷，并进行回归测试以确保缺陷已经被正确修复。

活动：开发团队修复缺陷，测试团队进行回归测试。

输出：缺陷修复报告、回归测试报告。

6.性能测试（时间：1-2周）

目的：验证网站的性能指标，如响应时间、并发处理能力等。

活动：使用性能测试工具进行压力测试、负载测试等。

输出：性能测试报告。

7.安全测试（时间：1-2周）

目的：确保网站的安全性，防止数据泄露和其他安全威胁。

活动：进行渗透测试、漏洞扫描等。

输出：安全测试报告。

8.用户验收测试（时间：1-2周）

目的：让最终用户参与测试，确保网站满足用户需求。

活动：邀请部分师生进行测试，收集反馈意见。

输出：用户验收测试报告、用户反馈汇总。

## 部署与上线计划

1.部署准备阶段（时间：1-2周）

主要任务：准备部署环境，包括服务器、网络、安全设施等。确保硬件和软件环境符合部署要求。

交付物：部署环境准备报告、部署手册。

2.数据迁移与备份阶段（时间：1周）

主要任务：将测试环境中的数据迁移到生产环境，同时制定数据备份策略，确保数据的安全性和完整性。

交付物：数据迁移报告、数据备份策略。

3.部署执行阶段（时间：1周）

主要任务：按照部署手册进行网站的部署工作，包括应用程序的安装、配置、测试等。确保网站在生产环境中正常运行。

交付物：部署执行报告。

4.性能优化与监控阶段（时间：1-2周）

主要任务：对网站进行性能优化，确保网站能够承载实际的用户访问量。同时建立监控机制，实时监控系统的运行状态。

交付物：性能优化报告、监控报告。

5.上线前的内部测试阶段（时间：1周）

主要任务：在生产环境中进行内部测试，模拟用户操作，确保网站的稳定性和可靠性。

交付物：内部测试报告。

6.培训与宣传阶段（时间：1-2周）

主要任务：为超算中心的师生提供网站使用培训，介绍网站的功能和操作方法。同时开展宣传活动，提高网站的知名度和使用率。

交付物：培训资料、宣传材料。

7.正式上线阶段（时间：1周）

主要任务：在完成所有准备工作后，将网站正式对外开放，供全体师生使用。

交付物：上线公告。

## 维护计划

1.维护策略制定阶段（时间：1周）

主要任务：制定网站的维护策略，包括维护的频率、责任人、维护内容等。

交付物：维护策略文档。

2.周期性维护阶段（长期进行）

主要任务：按照维护策略定期对网站进行检查、更新和优化，包括软件升级、硬件检查、安全漏洞修复等。

交付物：维护记录报告。

3.故障应急响应阶段（长期进行）

主要任务：建立故障应急响应机制，对突发的系统故障进行快速定位和处理，确保网站的正常运行。

交付物：故障处理报告。

4.用户反馈收集与处理阶段（长期进行）

主要任务：建立用户反馈渠道，收集用户对网站的意见和建议，及时处理用户反馈的问题。

交付物：用户反馈处理报告。

5.性能监控与优化阶段（长期进行）

主要任务：对网站的性能进行持续监控，发现性能瓶颈并进行优化，提升网站的访问速度和响应能力。

交付物：性能监控报告、性能优化报告。

6.数据备份与恢复阶段（长期进行）

主要任务：定期对网站的数据进行备份，以防数据丢失。同时建立数据恢复机制，确保在发生数据丢失时能够快速恢复。

交付物：数据备份报告、数据恢复报告。

7.安全监控与升级阶段（长期进行）

主要任务：对网站的安全状况进行持续监控，及时发现并处理安全威胁。同时定期进行安全升级，提升网站的安全防护能力。

交付物：安全监控报告、安全升级报告。

8.功能更新与迭代阶段（根据需求进行）

主要任务：根据用户的需求和市场的变化，对网站的功能进行更新和迭代，保持网站的竞争力。

交付物：功能更新报告、迭代计划。

# 风险管理

## 可能存在的风险

1.技术风险

新技术适应性：如果项目采用新的技术或工具，可能会遇到技术适应问题，导致开发延迟或增加成本。

系统集成复杂性：多个系统或服务（如算力计费系统、用户账户管理系统）需要无缝集成，可能存在兼容性或数据同步问题。

2.需求风险

需求变更频繁：在项目执行过程中，杨老师或其他干系人的需求可能发生变化，导致额外的开发工作。

用户需求理解不足：前期需求调研可能未能完全捕捉到所有师生的需求，导致后期需求调整。

3.人力资源风险

关键人员缺乏：特定技能的人才短缺可能导致项目延期或质量问题。

团队稳定性：项目团队成员变动可能影响项目进度和知识传承。

4.时间管理风险

进度安排过紧：过于紧张的时间表可能导致无法按时完成所有功能。

测试和部署延时：软件测试可能发现更多问题，导致修复和重新测试，增加项目时间。

5.预算风险

成本估算不准确：预算可能因需求变更、市场价格波动等因素超出初步估计。

额外资金需求：未预见的技术难题或需求变化可能需要额外的资金支持。

6.安全风险

数据安全问题：网站可能面临数据泄露、黑客攻击等安全威胁。

系统漏洞：新开发的系统可能存在未知漏洞，导致服务中断或数据丢失。

7.法律和合规风险

数据处理合规性：必须确保网站的数据处理遵守相关的数据保护法规。

知识产权问题：使用第三方软件或服务时，需要确保遵守相应的许可协议，避免侵权。

8.运营风险

用户接受度：新系统可能面临用户接受度低的风险，特别是如果操作复杂或与现有习惯不符。

维护和支持：长期维护和支持可能因为资源分配不足而变得困难。

## 风险评估

1. 技术风险

新技术适应性

可能性：中等

影响：高

原因：采用新技术可能需要额外的学习和适应时间，且可能存在未知的技术问题。

系统集成复杂性

可能性：高

影响：高

原因：多个系统需要集成，增加了项目的复杂性，可能导致兼容性问题和数据同步问题。

2.需求风险

需求变更频繁

可能性：高

影响：中到高

原因：项目周期长，干系人需求可能变化，导致需求频繁变更。

用户需求理解不足

可能性：中等

影响：中

原因：需求调研可能未能完全理解用户的实际需求，导致后期需求调整。

3.人力资源风险

关键人员缺乏

可能性：低

影响：高

原因：特定技能的人才可能难以招聘或保留，但这种情况不常见。

团队稳定性

可能性：中等

影响：中

原因：团队成员可能会变动，但通过适当的管理和培训可以降低影响。

4.时间管理风险

进度安排过紧

可能性：中等

影响：高

原因：过于紧张的时间表可能导致无法按时完成所有功能。

测试和部署延时

可能性：中等

影响：中

原因：测试可能发现更多问题，需要额外的修复和测试时间。

5.预算风险

成本估算不准确

可能性：中等

影响：中

原因：预算可能因为需求变更或市场价格波动而超出初步估计。

额外资金需求

可能性：低

影响：高

原因：未预见的技术难题或需求变化可能需要额外的资金支持。

6.安全风险

数据安全问题

可能性：中等

影响：高

原因：网站可能面临数据泄露、黑客攻击等安全威胁。

系统漏洞

可能性：中等

影响：高

原因：新开发的系统可能存在未知漏洞，导致服务中断或数据丢失。

7.法律和合规风险

数据处理合规性

可能性：低

影响：高

原因：必须遵守相关法规，否则可能面临法律风险。

知识产权问题

可能性：低

影响：中

原因：使用第三方软件或服务时，需要注意遵守许可协议。

8.运营风险

用户接受度

可能性：中等

影响：中

原因：新系统可能面临用户接受度低的风险。

维护和支持

可能性：低

影响：高

原因：长期维护和支持可能因为资源分配不足而变得困难。

## 风险的应对计划

一、技术风险应对

1.新技术适应性：

提前进行技术培训和预研。

选择稳定性好、社区支持强的技术和框架。

安排时间进行原型开发和测试。

2.系统集成复杂性：

在项目初期进行系统架构设计，确保各组件间的兼容性。

采用模块化设计，降低系统间耦合度。

引入专业的集成开发环境（IDE）和持续集成/持续部署（CI/CD）流程。

二、需求风险应对

1.需求变更频繁：

与杨老师和干系人建立良好的沟通机制，确保需求的及时反馈和确认。

采用敏捷开发方法，以适应需求的变化。

2.用户需求理解不足：

开展用户调研，收集反馈，确保需求的准确性。

定期进行用户测试，验证功能是否符合用户期望。

三、人力资源风险应对

1.关键人员缺乏：

建立人才储备计划，对关键职位进行双备份。

鼓励跨学科培训，提高团队的灵活性和适应性。

2.团队稳定性：

提供职业发展路径和员工激励计划。

建立明确的项目管理流程和知识共享机制。

三、时间管理风险应对

1.进度安排过紧：

制定详细的项目计划，包括里程碑和缓冲时间。

定期跟踪进度，适时调整计划。

2.测试和部署延时：

提前规划测试资源和时间，包括负载测试和安全测试。

确保有充足的时间进行用户验收测试（UAT）。

四、预算风险应对

1.成本估算不准确：

进行详细的成本分析，包括硬件、软件、人力和其他资源。

建立预算储备，以应对不可预见的支出。

2.额外资金需求：

与财务部门合作，确保有额外的资金来源或信贷安排。

优先处理关键功能，以最小化必要功能的实施成本。

五、安全风险应对

1.数据安全问题：

实施最佳实践的安全措施，如SSL加密、防火墙和入侵检测系统。

定期进行安全审计和漏洞扫描。

2.系统漏洞：

引入代码审查和自动化测试。

建立快速响应机制，以便及时修复发现的漏洞。

六、法律和合规风险应对

1.数据处理合规性：

了解并遵守相关的数据保护法规。

聘请法律顾问，确保所有操作都在法律框架内。

2.知识产权问题：

使用开源软件时，确保遵守许可协议。

对自有代码进行版权登记，保护知识产权。

七、运营风险应对

1.用户接受度：

提供用户培训和文档，帮助用户理解和使用新系统。

通过试点项目和逐步推广来增加用户接受度。

2.维护和支持：

建立用户支持中心，提供技术支持和帮助。

确保有足够的资源用于系统的长期维护和升级。

## 风险的管控

1.技术风险管控

定期技术审查：定期进行技术审查会议，评估技术进展和解决方案的有效性。

技术预警系统：建立技术预警机制，及时发现潜在的技术问题和不匹配。

2.需求风险管控

变更控制流程：实施严格的变更控制流程，确保所有需求变更都经过适当的评审和批准。

定期干系人会议：定期与杨老师和关键干系人沟通，确保需求的连续性和一致性。

3.人力资源风险管控

人力资源监控：监控团队的人力资源状况，确保关键职位的稳定性和工作的连续性。

培训和发展计划：实施培训和职业发展计划，提升团队的整体能力和适应性。

4.时间管理风险管控

进度跟踪：使用项目管理工具跟踪进度，确保项目按时完成。

缓冲时间管理：合理安排缓冲时间，以应对不可预见的延误。

5.预算风险管控

成本监控系统：实施成本监控系统，定期检查和报告成本状况。

预算审查：定期进行预算审查，确保资金使用的合理性和效率。

6.安全风险管控

安全策略执行：确保所有团队成员遵守安全策略和程序。

定期安全审计：定期进行安全审计，及时发现和修复安全漏洞。

7.法律和合规风险管控

合规性检查：定期进行合规性检查，确保项目符合所有相关法律和规定。

法律顾问协助：在需要时寻求法律顾问的协助，确保合法合规操作。

8.运营风险管控

用户反馈机制：建立有效的用户反馈机制，及时收集和响应用户的意见和建议。

维护和支持计划：制定详细的维护和支持计划，确保系统长期稳定运行。

9.具体措施：

定期更新风险登记册：记录所有风险的状态，包括识别、分析、应对和管控的结果。

风险管理会议：定期召开风险管理会议，讨论风险管控的进展和效果。

风险通报机制：建立风险通报机制，确保关键风险信息及时传达给所有相关方。

风险管理培训：对项目团队进行风险管理培训，提高团队的风险意识和管理能力。

# 预算说明

## 预算组成

1.域名注册费用：通常这是一次性的年度费用，价格因注册商和域名后缀不同而异，一般在几百元以内。

2.服务器托管费用：这取决于网站的预期流量和数据存储需求。如果预计流量较大，可能需要更高性能的服务器，费用可能在数千至数万元/年。

3.网站设计和开发费用：这部分费用变动较大，根据网站复杂性和功能需求，从几千到数十万不等。

4.内容管理系统（CMS）费用：如果采用现有的开源CMS系统，如WordPress，可以节省开发成本，但可能需要购买主题和插件，费用从几百到几千元不等。

5.SSL证书费用：为了保证网站数据传输的安全，SSL证书通常是必要的，价格一般在几百元左右。

6.备份和安全费用：确保网站数据安全和及时备份，这通常是云服务提供商包含的额外服务，或者可以购买专门的备份和安全服务。

7.维护费用：包括日常的内容更新、系统升级、安全监控等，这些费用取决于网站维护的频率和复杂性。

8.其他费用：可能包括备案、广告推广、人员培训等，具体费用根据实际情况而定

## 预算估计

1.域名注册：100元/年

2.服务器托管：

虚拟主机：1500元/年

云服务器：8500元/年

3.网站设计和开发：

模板建站：3000元

定制开发：12000元

全功能定制开发：35000元

4.CMS系统：1000元

5.SSL证书：500元/年

6.备份和安全：1000元

7.维护费用：2000元/年。

8.其他费用（如备案、推广等）：2000元。

总费用：65700左右

## 预算控制与管理

1.预算审批流程：确保每一笔支出都有明确的责任人和审批流程

2.资金使用透明化：所有资金的使用情况都应记录在案，以供审计和监督。

3.定期财务报告：定期生成财务报告，分析预算执行情况，及时调整预算策略。

4.风险管理：识别可能的财务风险并制定相应的应对措施。

5.优先级排序：对需求进行优先级排序，确保关键需求得到优先满足，而非必要或次要需求可以在预算允许的情况下再考虑。

6.迭代开发：采用迭代开发方法，将项目分为多个小阶段，每个阶段都有明确的预算和目标。这样可以更好地控制成本，并在每个阶段结束时评估项目的进展和财务状况。

7.多渠道筹集资金：除了政府拨款，还应积极寻求科研项目资助、产业合作等多元化资金来源。

# 需求变更及跟踪管理

首先，需求变更在软件项目中是不可避免的。由于市场环境、客户需求、技术进步等多种因素的影响，项目需求可能会在项目执行过程中发生变化。因此，建立有效的需求变更管理机制是确保项目成功的关键。需求变更管理应遵循“变更申请、审批、实施、重新确认”的流程，确保变更的合理性、必要性和可控性。

其次，需求跟踪管理对于确保项目成果符合用户需求具有重要意义。需求跟踪的目的是建立与维护“需求、设计、编程、测试”过程的一致性。通过正向跟踪和逆向跟踪相结合的方式，可以确保每个需求点都能在后继工作成果中找到对应点，同时确保设计文档、代码、测试用例等工作成果都能在需求规格说明书中找到出处。为了实现这一目标，需要建立并维护需求跟踪矩阵，以便随时查看和更新需求状态。

## 线性责任表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Who is responsible  WBS | 罗骏辉  (项目负责人) | 陆宇涵  （技术） | 卢晨阳  （运维测试） | 周帅  （预算、风险管理） | 周驰原  （需求） |
| 确定标准和需求 | 1 | 3 | 3 | 3 | 1 |
| 可靠性和功能性测试 | 1 | 1 | 1 | 3 | 3 |
| 制作原型 | 1 | 1 | 3 | 3 | 3 |
| 风险管理 | 1 | 3 | 3 | 1 | 3 |
| 预算成本控制 | 2 | 3 | 2 | 1 | 3 |
| 配合领导审查 | 1 | 2 | 2 | 2 | 2 |

## 设置通知和提醒

当出现需求变动的时候，通过PingCode设置通知和提醒团队成员，同时也要通知到利益相关者。