

竞赛交流平台项目

业务需求报告

项目名称：竞赛交流平台建设

编制部门：第 9 平台工作组

编制日期：2024 年 9 月 25 日

目录

第一章 引言	5
1.1 项目背景	5
1.2 平台目标	5
1.3 文档约定	6
1.4 预期的读者和阅读建议	6
1.5 产品的范围	6
1.6 参考文献	7
第二章 系统概述	8
2.1 系统目标	8
2.2 用户特点	9
2.3 可行性分析	9
第三章 系统主要功能	10
3.1 系统前台主要功能	10
3.1.1 AI 竞赛助手	10
3.1.2 用户个人管理	11
3.1.3 竞赛项目查询	12
3.1.4 团队管理	13
3.1.5 热门竞赛展示	14
3.1.6 交流讨论区	15
3.1.7 反馈建议区	16
3.1.8 资源共享区	18
3.1.9 备赛专区	19
3.1.10 帮助中心	21
3.2 管理员主要功能	23
3.2.1 管理员管理	23
3.2.2 竞赛项目管理	23
3.2.3 用户管理	23
3.2.4 团队管理	24

3.2.5 消息管理	24
第四章 其它非功能需求	26
4.1 性能需求	26
4.1.1 系统可用性要求	26
4.1.2 时间特性要求	26
4.1.3 系统开放性要求	27
4.1.4 系统可扩充性要求	28
4.1.5 界面友好性要求	28
4.2 安全设施需求	29
4.2.1 硬件安全主要在于服务器硬件的安全	29
4.2.2 网络安全	30
4.3 安全性需求	30
4.3.1 系统安全	30
4.3.2 程序安全	30
4.3.3 数据安全	31
4.3.4 内容安全	31

文件更改记录

版本历史

版本	变更时间	变更章节	性质	变更内容	修改人	批准人
1.0.0	2024-09-25	封面	M	补充封页信息，调整格式	吴嘉楷	
1.0.1	2024-10-01	第一章	AM	修改各小章节的内容，添加“平台目标”小章节	吴嘉楷	
1.0.2	2024-10-01	第四章	M	修改第四章每一节的内容	吴嘉楷	
1.0.3	2024-10-02	第三章	A	构想出系统前台的前六个主要功能	吴嘉楷	
1.0.4	2024-10-02	第三章	AM	修改并补充管理员主要功能部分的文档内容	吴嘉楷	
1.0.5	2024-10-07	第二章	AM	修改并增加第二章每一节的内容	林浩晟	
1.0.6	2024-10-07	第三章	A	构想出系统前台的四个主要功能	林浩晟	
1.0.7	2024-10-08	第三章	A	添加三个主要功能的 UI 图	吴嘉楷	
1.0.8	2024-10-08	第三章	A	添加三个主要功能的 UI 图	林浩晟	
1.0.9	2024-10-09	第三章	A	绘制系统前台功能图	吴嘉楷	
1.1.0	2024-10-09	第三章	A	绘制系统后台功能图	林浩晟	
1.1.1	2024-10-09	全文	M	格式统一与修改	吴嘉楷	

注：性质可以为 N-新建、A-增加、M-更改、D-删除。

第一章 引言

1.1 项目背景

随着高等教育的广泛普及与持续深化，高年级本科生与低年级本科生在学术探索、科研活动及创新实践中的互动与合作愈发紧密。特别是在各类学科竞赛这一充满活力的舞台上，跨年级学生的协同参与已成为提升项目品质、促进知识与技能跨代传递的有效途径。然而，当前面临的挑战在于，缺乏一个高效、专门的交流平台来有效支撑这种跨层次、跨经验的学习合作模式。本项目立足于以下现状：

- 大赛参与度低：**近年来，计算机与软件学院在挑战杯等创新创业大赛中的参赛队伍数量严重不足，成功立项及通过校赛的队伍更是稀缺。亟需采取措施，增加参赛学生及队伍数量，尤其是提升高质量团队的参与度。
- 研究所与本科生沟通不畅：**当前，研究所与本科生，特别是新生之间缺乏有效连接，导致双方供需不匹配。学生难以找到加入研究所的途径，而研究所则急需本科生的参与，无论是新手还是已有研究基础的学生。信息差显著，亟需建立沟通桥梁，平衡供需关系。
- 学习资源分享不足：**随着学院发展，众多优秀毕业生和在校生拥有丰富学习资源。他们希望分享这些资源，帮助学弟学妹少走弯路，并希望得到认可。而新生若能尽早接触这些经验分享，将极大受益。因此，需搭建平台整合与分享这些学习资源。

1.2 平台目标

为了进一步强化高年级与低年级本科生在学术研究与实践探索领域的深度融合，激发他们的创新潜能，并培育卓越的团队协作精神，本项目聚焦于“知识传承、团队协作、竞赛实践”三大核心支柱：

- 知识传承：**鼓励高年级本科生作为知识的桥梁，将其在学术探索与项目实践中积累的宝贵知识和经验，无私地传授给低年级同学，加速他们的成长步伐，实现知识的有效接力与增值。
- 团队协作：**通过引入先进的团队协作模块，优化资源配置，促进团队成员间的有效沟通与协作，使每位学生都能在各自擅长的领域发挥最大效能，共同面对挑战，解决复杂问题。
- 竞赛实践：**依托竞赛实践模块，提供丰富的模拟竞赛环境与实战机会，让学生在接近真实场景的比赛中磨砺技能，积累宝贵的实战经验，从而全面提升其竞赛表现与

解决实际问题的能力。

鉴于此，我们计划构建一个集综合性、多层次性于一体的竞赛交流平台显得尤为重要。该平台旨在打破年级壁垒，促进知识、技能与经验的自由流动，为高年级与低年级本科生搭建起一座连接理论与实践、个人成长与团队协作的桥梁。

1.3 文档约定

1. 本需求分析说明书采用 PDF 格式，本说明书中的图片均采用 jpg 格式；
2. 本说明书有封面、页眉页脚、页码和目录；章节分级，最多不超过三级；
3. 大标题用二号；二级标题用四号；三级标题用小四。正文字体统一为五号。全文采用宋体。
4. 行间距为 1.25 倍行距，段前后间距各 0.5 行，多个列项算在一段中；
5. 图片构图均衡、美观大方，下方附有图片标题及序号；
6. 表格字体、左右边界、标题栏等保持一致风格。

1.4 预期的读者和阅读建议

本文旨在满足不同角色读者的需求，具体包括**详细设计人员、开发人员、测试人员及用户**，各角色阅读重点各异：

- **详细设计人员与开发人员**：需全面理解系统，以本文书为详细设计基础。因此，这两类人员需通读全文，确保充分掌握文档中规定的用户需求。
- **测试人员**：负责系统功能与非功能测试。需重点关注“系统特性”以规划功能测试，同时阅读“其它非功能需求”以制定非功能测试计划。
- **用户**：作为软件使用者，界面设计至关重要。用户应阅读“外部接口需求”中的“用户界面部分”，确保界面设计符合操作习惯。此外，还需浏览“其它非功能需求”的“业务规则”部分，核实业务规则与实际运营的一致性，并阅读专为用户准备的“用户文档”。

1.5 产品的范围

本竞赛交流平台，类似于学术与实践领域的 B2B 平台，旨在促进深圳大学本科生在竞赛和科研中的交流与合作。平台的显著特性包括高互动性、安全性、易于使用、良好的可扩展性以及直观的用户界面，这些特性共同提升了学生参与竞赛的效率和质量。竞

赛交流平台相较于传统交流方式的优势在于，它不受时间和空间的限制，能够广泛整合教师、行业专家、研究生和本科生的资源，极大地丰富了参与者的选择范围。同时，借助平台的实时通讯技术，学生能够更便捷地获取信息和支持，显著降低了沟通成本。

本产品具有如下的系统特性：

- **高互动性：**平台支持即时通讯和讨论组功能，使得学生能够实时交流想法和解决问题。
- **安全性：**平台采用先进的数据加密技术，确保用户信息和交流内容的安全。
- **易于使用：**用户界面友好，操作简单直观，便于学生快速上手。
- **良好的可扩展性：**平台架构设计灵活，能够根据用户需求和学校发展进行功能扩展。
- **直观的用户界面：**界面设计清晰，功能分区明确，便于用户快速找到所需功能。

1.6 参考文献

《软件需求(第2版)》【作者】(美) Karl E. Wiegiers 【译者】刘伟琴 刘洪涛

《需求分析与系统设计》【作者】 Leszek A. Maciaszek 【译者】金芝

《实用软件需求》【作者】(美) Benjamin L. Kovitz 【译者】胡辉良 张罡

《Software Requirements Specification》【作者】(美) Richard H. Thayer; Mark Christensen;

《心中有读者——论软件项目需求文档的撰写》【作者】(中) 薛蓓燕

《软件需求工程理论、方法与实践》【作者】(中) 刘佳

第二章 系统概述

2.1 系统目标

系统设计的基本原则是以系统总体目标为宗旨，为用户提供一个技术先进、成熟可靠、灵活可用、性能优秀的网站应用系统。以下几个方面将贯穿整个方案：

- 1) **技术先进性：**采用前沿的技术和框架，确保平台能够满足现代网络应用的需求，例如高效的前后端技术（如 React、Node.js、Django 等），使用主流数据库（如 MySQL、PostgreSQL）进行数据存储。
- 2) **用户友好性：**系统设计需简洁直观，操作便捷，提供完善的用户引导和帮助系统，降低用户使用门槛，确保不同年级的本科生都能够方便地使用和参与平台上的活动。
- 3) **灵活可用性：**系统功能模块化设计，支持扩展和升级，以适应不断变化的竞赛需求和用户规模，提供丰富的自定义选项，例如个性化的用户设置和竞赛模块。
- 4) **高效性能：**优化前后端代码，确保页面加载迅速，响应及时，提供流畅的用户体验，尤其是对实时协作与交流模块（如即时消息、视频会议等）的性能需求进行重点优化。
- 5) **安全性：**建立严格的安全机制，防止数据泄露、恶意攻击等问题。包括用户认证、权限管理、数据加密等多个安全措施，保障用户的个人信息和交流内容安全。
- 6) **数据存储与管理：**为用户的交流记录、竞赛信息、知识资料等数据提供高效的存储和管理，建立良好的数据结构，便于检索、管理和更新，支持多种类型的数据存储，包括文本、图片、视频等。
- 7) **交流与互动模块：**提供高效的互动工具，例如论坛、实时聊天、讨论组等模块，便于高年级学生分享经验、低年级学生提出问题，以及多方协作解决问题。
- 8) **竞赛资源管理：**提供竞赛题库、资料分享、课程资源等模块，方便学生进行自学和复习。管理员可以方便地上传和更新竞赛资源，保证内容的持续更新和丰富性。
- 9) **团队协作工具：**支持小组讨论、任务分配、进度跟踪等功能，帮助学生在团队协作中提升自身能力，更好地完成竞赛任务。
- 10) **反馈与改进：**提供反馈渠道，用户可以针对系统问题或提出建议，管理员定期收集用户反馈，以便对系统进行迭代改进，提升用户体验。

2.2 用户特点

本系统的最终用户将是前台用户（参与竞赛交流的同学们），后台管理人员以及系统维护人员。

- 1) **前台用户：**是使用该系统的主要人员。对同学们无要求，只要会基本电脑操作即可。
- 2) **后台管理用户：**需要经过一定时间的培训。
- 3) **系统维护人员：**需要熟悉系统日常维护相关知识。

2.3 可行性分析

近几年来计算机与软件学院参与挑战杯创新创业等大赛的队伍数量过少，成功立项与通过校赛的队伍数更是少之又少，因此迫切的需要一个解决方案来增加学院参加各种项目的学生以及队伍，尤其是高质量的队伍。

目前缺乏一个研究所与本科生，尤其是新生之间的桥梁，导致有很多想要或者是有加入研究所意愿的学生不知道参与途径，另一方面，有不少研究所本身也是亟需本科生，无论是愿意从零努力学习的新生还是已经有一定研究基础的本科生的参加研究成果。这导致研究所与新生之间存在严重的信息差，缺乏一个能够沟通两边的桥梁，将本科生和研究所的信息整合起来，实现两边的供需关系平衡。

随着近几年计算机与软件学院的发展，已经有了越来越多的优秀毕业生和优秀在校生，这些优秀的学长学姐们手中有着丰富的学习资源与笔记，他们或在个人博客上撰写文章，或通过代代相传的方式，将这些笔记流传着，他们希望能有更多的人看到这些笔记，少走一些他们走过的弯路，也希望得到后续学弟学妹们对他们分享的笔记的认可。而另一方面有不少刚入学的学弟学妹们，刚刚步入大学生活，在与高中学习方式大相径庭的大学，如果能够更早的了解并运用这些学长学姐们的经验和笔记分享，对于他们的大学学习生活大有裨益，因此希望能够搭建一个平台来对这些学习资源进行整合与分享。

因此，我认为竞赛交流平台的开发是完全可行的。

第三章 系统主要功能

3.1 系统前台主要功能

前台包括以下主要功能模块：AI 竞赛助手、用户个人管理、竞赛项目查询、团队管理、热门竞赛展示、交流讨论区、反馈建议区、资源共享、备赛专区、帮助中心。

3.1.1 AI 竞赛助手

AI 竞赛助手为用户提供智能化的竞赛信息咨询与竞赛知识问答服务。用户可以通过与 AI 竞赛助手进行自然语言的互动，快速获取竞赛相关信息，如竞赛介绍、报名入口、参赛要求、赛事日程等。此外，AI 竞赛助手还能解答用户提出的竞赛知识题目，提供学习资源，帮助用户更好地备赛。

a) 说明及优先级

AI 竞赛助手为参赛学生提供了一种高效、便捷的竞赛信息和知识获取方式，简化了复杂的信息查询过程。借助 AI 的实时响应能力，用户可以在对话中直接得到精准的竞赛解答。由于该功能可以显著提升用户体验和平台的服务效率，优先级应设为高。

b) 响应/激励

在平台首页的右下角，用户可以看到“AI 竞赛助手”的入口图标（悬浮按钮）。点击图标后，用户将进入与 AI 竞赛助手的实时聊天界面。AI 助手依托强大的 AI 大模型与云端竞赛信息数据库，结合用户的具体问题，生成个性化、准确的回答，确保用户获得高质量的竞赛信息和解决方案。激励机制可以通过引导用户使用 AI 助手获得专属竞赛资讯，参与平台互动，提升用户粘性。

c) 功能需求

1. 入口设计:

在页面上方或右下角显示悬浮按钮，用户可随时点击进入 AI 助手界面。

智能对话：基于自然语言处理的 AI 模型支持与用户的文本互动。该模型能够理解用户意图，并从云端数据库中检索相关竞赛信息、答题内容或资源推荐。

2. 个性化推荐:

根据用户的搜索记录和行为，AI 助手可以推荐个性化的竞赛和学习资源，增强用户体验。

3. 持续学习:

AI 助手通过用户的反馈和互动，不断优化其知识库和回答准确度，提升智能化水平。

d) UI 设计图



3. 1. 2 用户个人管理

支持用户注册、登录，以及个人信息查看与修改。用户可查阅参与的竞赛记录、获得的积分及收藏的竞赛资源。

a) 说明及优先级

为参赛学生和导师提供注册登录功能，并支持用户信息的更新，同时为网站使用消费者提供会员注册功能，以及对会员信息和会员相关行为活动的功能。优先级设为高。

b) 响应/激励

用户查看个人信息、以及购买过的竞赛资料都需要进行登录操作。系统如果检测用户没有登录直接跳转到登录页面。登录后，用户可方便地管理个人竞赛信息。

c) 功能需求

实现登录、注册界面，个人中心页面展示个人信息、竞赛历史、积分详情等。

3.1.3 竞赛项目查询

用户按竞赛项目名称等关键字进行模糊检索需要的信息。提供关键词或类别搜索功能，支持模糊匹配，快速检索竞赛项目信息。

a) 说明及优先级

在网站复杂的列表中进行人工查看和检索是一件耗时且令人不快的事，对一些数据提供自动查询功能，直接得到用户想要的信息，可以提高业务效率和用户满意度，也可以获得更多的业务量。快速定位感兴趣的项目，对用户至关重要，优先级设为高。

b) 响应/激励

用户进入网站后，无论是否登录，网页上方都有一个文本框输入区域，用户在其中输入要搜索的任意字符串，按下搜索按钮，系统接受请求后即在数据库中进行字符串匹配工作，将匹配的结果返回；前台则转发到相应页面，页面根据返回结果向用户显示。

c) 功能需求

1. 支持按项目名称、类别等条件进行搜索：

用户可以输入项目名称的关键词，系统会根据输入的关键词进行匹配，并展示相关的竞赛项目。系统提供竞赛项目的分类选项，用户可以选择特定的类别来筛选竞赛项目。

2. 支持模糊搜索，提供相似匹配结果：

当用户输入的关键词不够精确时，系统能够识别并展示与关键词相关的竞赛项目，提供模糊匹配的结果。系统还会根据关键词的相似度对结果进行排序，以便用户更快地找到最符合需求的竞赛项目。

3. 搜索结果展示：

系统以列表或卡片的形式展示搜索结果，包括竞赛项目的名称、类别、简介、截止日期等关键信息。用户可以点击搜索结果中的项目名称，进入详细的竞赛项目页面，了解更多信息。

4. 搜索历史与推荐：

系统记录用户的搜索历史，并根据用户的搜索习惯和偏好，智能推荐相关的竞赛项目。用户可以在推荐列表中快速找到可能感兴趣的竞赛项目，提高查询效率。

d) UI 设计图



3.1.4 团队管理

团队管理模块是竞赛平台中至关重要的一个部分，它允许用户轻松地创建、加入或管理自己的竞赛团队。通过这一模块，用户可以查看团队成员的详细信息，包括他们的专业背景、技能特长等，从而更好地了解团队成员的优势和不足，为团队任务的分配提供有力支持。此外，用户还可以直接在团队内部进行任务分配，确保每个成员都明确自己的职责和任务，从而提升团队的协作效率。

a) 说明及优先级

通过这一模块，用户可以更轻松地组建和管理自己的团队，从而提升团队的凝聚力和竞争力。这对于提高竞赛项目的质量和成功率具有重要意义。优先级设为高。

b) 响应/激励

团队管理模块不仅提供了便捷的团队组建和管理功能，还通过平台为用户提供了寻找志同道合的队友的机会。用户可以在平台上浏览其他用户的资料和竞赛经历，找到与自己兴趣和技能相匹配的队友，从而组建一个强大的竞赛团队。同时，通过任务分配和进度跟踪等功能，用户可以更好地协调团队成员的工作，提升团队协作能力，为竞赛的成功打下坚实的基础。

c) 功能需求

1. 创建/加入团队：

用户可以创建自己的竞赛团队，并设置团队的名称、简介、目标等基本信息。用户可

以浏览平台上的团队列表，并根据自己的兴趣和需求选择加入合适的团队。团队创建者和管理员可以设置团队的加入条件，如是否需要审核、是否需要邀请码等。

2. 管理团队成员：

团队管理员可以查看团队成员的详细信息，包括他们的姓名、专业背景、技能特长等。团队管理员可以邀请新用户加入团队，并对团队成员进行角色分配，如队长、副队长、普通成员等。团队管理员可以处理团队成员的退出申请，确保团队的稳定性和凝聚力。

3. 分配任务：

团队管理员可以在团队内部进行任务分配，明确每个成员的职责和工作内容。团队成员可以查看自己的任务列表，了解任务的进度和完成情况。系统可以支持任务进度的实时更新和跟踪，确保团队成员能够按时完成任务。

3.1.5 热门竞赛展示

热门竞赛展示模块旨在向用户展示当前平台上最受关注、参与度最高的竞赛项目。通过这一模块，用户可以迅速了解到当前竞赛领域的热门话题和趋势，从而选择自己感兴趣的竞赛项目进行参与。这不仅有助于提升用户的参与度，还能促进平台上的竞赛氛围，增强用户之间的交流和互动。

a) 说明及优先级

通过这一模块，我们可以有效地引导用户关注并参与当前最热门的竞赛项目，从而提升平台的整体竞争力和影响力。鉴于热门竞赛对于吸引用户参与和提高平台活跃度的重要作用，我们将热门竞赛展示模块设定为高优先级。

b) 响应/激励

热门竞赛展示模块通过动态更新最热门的竞赛项目，激励用户积极参与其中。参与热门竞赛不仅可以提高用户的曝光度和知名度，还有机会获得更多的奖励和认可。同时，这也为用户提供了一个展示自己才华和能力的平台，有助于提升他们的自信心和竞争力。

c) 功能需求

1. 动态展示热门竞赛：

系统能够实时或定期更新热门竞赛列表，确保用户看到的是当前最受欢迎的竞赛项目。热门竞赛的展示形式可以包括竞赛名称、封面图片、简介、参与人数、截止日期等关键信息。系统可以根据竞赛的参与度、关注度等多个维度综合评估，确定热门竞赛的排名和展示顺序。

2. 用户互动与引导:

在热门竞赛展示模块中，用户可以点击竞赛名称或封面图片进入详细的竞赛页面，了解更多信息并参与竞赛。系统可以提供参与竞赛的快捷方式，如一键报名、快速加入团队等，降低用户参与门槛。系统还可以展示竞赛的进展和成果，如已提交的作品、获奖情况等，进一步激发用户的参与热情。

3. 个性化推荐:

系统可以根据用户的兴趣和历史参与记录，智能推荐与用户相关的热门竞赛项目。推荐算法可以基于用户的浏览记录、搜索关键词、参与的竞赛类别等多个维度进行优化，提高推荐的准确性和有效性。

3.1.6 交流讨论区

交流讨论区是一个为竞赛参与者提供互动交流的社区平台。在这里，用户可以自由地发布关于竞赛项目的讨论帖子，分享自己的经验、心得和见解。同时，其他用户也可以对这些帖子进行评论和回复，形成积极的互动氛围。这一功能旨在促进学生间的知识交流，增强竞赛社区的凝聚力，为用户提供更多学习和成长的机会。

a) 说明及优先级

通过这一功能，用户可以更加便捷地获取竞赛相关的信息和资源，同时也可以在与其它用户的互动中不断提升自己的能力和水平。考虑到交流讨论区在促进学生间知识交流和提升竞赛社区活跃度方面的重要作用，我们将其优先级设定为中。

b) 响应/激励

在交流讨论区中，用户可以自由地提问、回答和分享经验。系统会对用户的帖子、评论和回复进行实时展示和更新，确保信息的及时性和有效性。同时，为了激励用户积极参与讨论，系统还可以设置一些奖励机制，如积分、勋章等，以表彰用户的贡献和活跃度。

c) 功能需求

1. 帖子发布功能:

用户可以自由地发布关于竞赛项目的讨论帖子，包括标题、内容和相关图片或视频等。系统会对用户发布的帖子进行审核，确保内容的合规性和质量。帖子发布后，系统会在讨论区进行展示，并通知相关用户进行查看和互动。

2. 评论和回复功能:

用户可以对其他用户的帖子进行评论和回复，表达自己的观点和看法。评论和回复内容需要经过系统审核，确保内容的合规性和友好性。系统会对评论和回复进行实时展示和

更新，确保信息的及时性和有效性。用户可以对评论和回复进行点赞、踩或举报等操作，以表达对其他用户内容的认可或不满。

3. 帖子管理和优化:

用户可以对自己的帖子进行编辑、删除或置顶等操作，以优化帖子内容和展示效果。系统可以对帖子进行排序和分类，如按时间、热度或重要性等维度进行展示。系统还可以提供搜索功能，方便用户快速找到感兴趣的帖子和讨论内容。

d) UI 设计图



3.1.7 反馈建议区

反馈建议区是专为提升平台用户体验而精心设计的，其核心目的在于快速识别并解决各类问题，从而驱动平台实现持续不断的改进与升级。这一区域不仅强化了用户与平台之间的沟通与互动，显著提升了用户的忠诚度，还通过引入公开透明的反馈机制，为构建积极向上的社群氛围提供了有力支持。更重要的是，对用户反馈问题的及时响应与解决，不仅有效缓解了客服的工作压力，降低了平台的维护成本，更为平台的稳健长远发展奠定了坚实的基础。

a) 说明及优先级

通过这一模块，竞赛参与者和用户可以积极表达意见、提出建议或报告问题，从而

提高竞赛参与者和用户的体验，这样可以进一步激发竞赛参与者的积极性。反馈建议区应是平台建设中的高优先级模块之一，因其与用户体验直接相关。通过及时的反馈收集和处理，可以帮助平台更快地迭代，保持良好的用户满意度和互动性。对于一个长期运营的竞赛平台，用户反馈能够为系统提供持续的改进方向。

b) 响应/激励

在反馈建议区，用户可以自由表达使用过程中遇到的不适体验，反馈范围包括功能问题、界面设计、操作流畅度等各个方面。后台人员会积极跟进每个反馈，快速响应并优化平台，确保用户的意见能够切实影响平台的改进方向。这不仅能显著提升竞赛参与者的整体满意度，还能优化用户在竞赛中的操作体验，激发他们的参与热情，进而提高竞赛的活跃度。

c) 功能需求

1. 反馈提交功能：

用户可以通过反馈表单轻松提交意见和建议，表单包含详细描述、联系方式以及截图或附件上传等选项。这种设计不仅使用户能够更准确地表达他们遇到的问题，还为后台管理人员提供了更全面的信息支持，使其能够迅速定位问题所在，分析问题根源。通过这一高效的反馈渠道，平台管理者可以快速响应并积极处理反馈内容，确保用户的意见得到及时关注和解决，从而进一步优化用户体验和提升平台的服务质量。

2. 反馈跟踪功能：

在提交了反馈后，反馈跟踪功能让用户能够随时查看自己提交的反馈处理进度，提供清晰的状态更新，如“已读”、“处理中”、“已解决”等。通过这一功能，用户可以实时了解反馈的处理情况，避免因信息不对称而产生的焦虑或不满。此外，平台还可以向用户推送重要的状态变化提醒，如问题解决后的通知或处理过程中需要更多信息的请求，确保用户始终参与反馈过程。

3. 反馈评价功能：

在反馈处理流程圆满结束后，用户能够便捷地对平台的处理流程及结果进行综合评价。这一用户评价机制至关重要，它不仅为用户提供了一个表达意见和反馈的便捷渠道，而且在很大程度上促进了平台处理流程和结果质量的提升。用户有权从响应速度、处理成效、沟通质量等多个核心维度进行打分与留言，这样的多维度评价机制为平台收集到了大量宝贵且极具价值的反馈信息与数据，为平台的持续优化与进步奠定了坚实基础。

d) UI 设计图

12:00

请详细描述您遇到的问题:

请上传截图:

其他信息

邮箱(必填):

电话(选填):

请写出您使用该平台的真实感受(选填):

12:00

反馈处理进度查询

反馈1

反馈内容.....

处理完成

感谢您的宝贵意见!

请给这次处理打分

3.1.8 资源共享区

a) 说明及优先级

该模块旨在为用户提供一个便捷的途径，用于分享和获取竞赛相关的学习资源、代码库、经验文章、解决方案等。用户可以通过类似于 GitHub 的形式发布文章、上传资源，并与其他用户进行互动交流。这一功能鼓励知识共享和共同进步。由于许多低年级同学获取资源的渠道较少，而竞赛参与者普遍需要学习资料、代码模板、工具及竞赛经验，因此我们较该模块优先级设为高。

b) 响应/激励

在该模块中，资源发布者能够快速上传资源并为其添加相关描述、标签、分类，以便其他用户能够轻松查找和使用。模块也提供完善的搜索和筛选功能，用户可以基于资源类型、竞赛类别、发布日期等条件快速找到所需的资源。同时通过积分系统，奖励那些分享优质资源的用户，积分可以用于获取平台的特权功能或实物奖励，比如深大周边等。

c) 功能需求

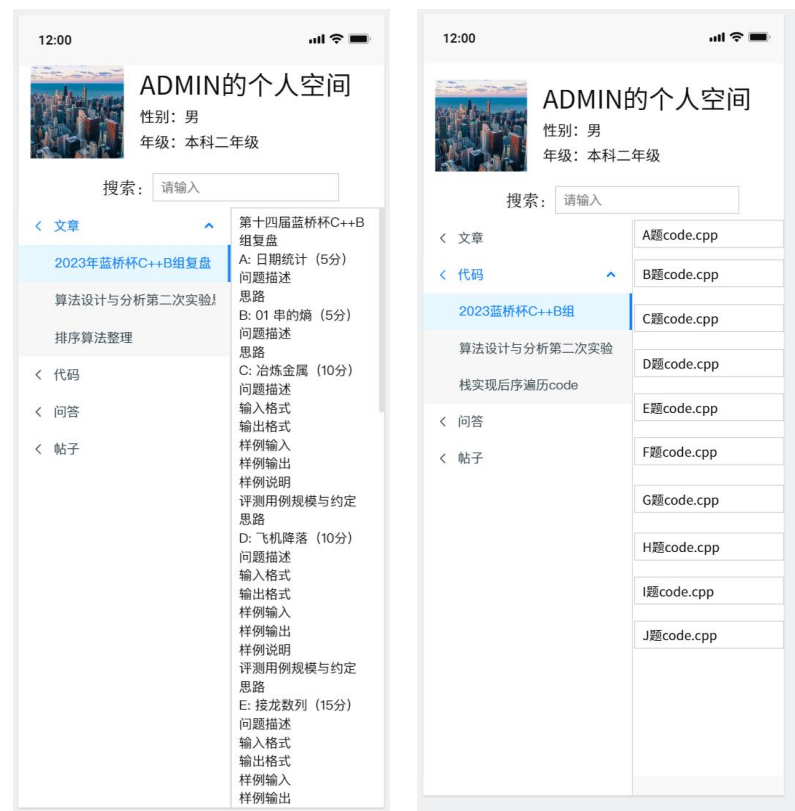
1. 资源上传与管理:

模块支持用户上传多种格式的资源，如 PDF、图片、代码压缩包等，满足用户分享竞赛相关资料的需求，同时用户也可以在模块中自己编写文章。用户在上传资源时，可以上传文档文件（如 PDF、Word）、图片文件（如 JPG、PNG）、代码和压缩文件（如 ZIP、RAR），从而分享教程、题解等多种类型的内容。此外，用户可为资源添加标签，帮助其他用户快速搜索和查找相关资源，平台也会提供常用标签建议，支持用户自定义标签，以提高资源的可见度和检索效率。

2. 搜索与筛选：

平台提供强大的资源筛选和搜索功能，允许用户根据资源的关键词、标签、类别等进行精准筛选，帮助用户快速找到所需的竞赛相关资源。每个资源上传时都会被要求设置标题、描述和标签，平台将充分利用这些信息来提升资源的可搜索性。

d) UI 设计图



3.1.9 备赛专区

在竞赛交流平台上，备赛专区是一个关键模块，它将算法按类型进行分类，并为每种算法类型设立相应的题库，帮助用户针对性地进行练习，掌握不同类型的算法，有效提高用户的编码能力。

a) 说明及优先级

算法分类题库区的旨在帮助用户系统化地掌握各类算法，提升解决竞赛题目的能力。通过将算法分类（如动态规划、贪心算法、图论等），用户可以根据自己的弱项或目标有针对性地选择某一类算法进行深入练习。每个分类下设立多个难度递进的题库，帮助用户从基础到高阶逐步掌握该类算法。鉴于有其他的编码网站，因此将该模块的优先级设为中。

b) 响应/激励

用户可以根据不同算法类别（如贪心算法、动态规划等）选择题库，每个题库包含从简单到复杂的多种难度的题目，覆盖经典题型及竞赛常见题型。每道题目都提供解题思路、官方题解以及用户讨论区，帮助用户通过学习他人的解法提高理解和解题能力。同时用户在完成特定数量的题目后可以获得成就和积分，例如“完成 20 道贪心算法题”、“完成 50 道动态规划题”等，积分可以用于兑换平台奖励或在社区中展示荣誉。

c) 功能需求

1. 算法分类体系：

平台需要提供清晰的算法分类，涵盖基础、进阶和竞赛常见算法（如贪心算法、二分搜索、动态规划、图算法等），并为每个分类下提供大量练习题目。

2. 难度分级题库：

每个算法分类下的题目应按照难度（如简单、中等、困难）进行分类，帮助用户根据自身水平选择合适的题目进行练习。

3. 进度跟踪系统：

平台会记录用户在每个算法分类中的练习进度，包括做题数量、正确率、解题时间等，并生成可视化的学习进度图表，便于用户自我评估。

4. 题解与讨论区：

每道题目都有详细的题解，用户也可以提交自己的解题思路，同时可以在讨论区与其他用户交流解题方法或提出问题，形成良好的互动学习氛围。

5. 积分与成就系统：

用户通过完成题目、参与挑战等活动积累积分，平台可通过积分兑换奖励或解锁高级功能，并提供成就系统鼓励用户完成更多练习。

6. 智能推荐与复习机制：

平台可以根据用户的解题记录，智能推荐尚未掌握的算法题目或历史错题，帮助用

户反复练习，巩固弱项。

3.1.10 帮助中心

本模块旨在助力用户全面了解算法竞赛体系的总体知识框架，协助用户迅速掌握竞赛相关内容，顺利入门数据结构与算法。模块主要提供竞赛、算法、数据结构等内容的官方参考资料，用以解答用户相关疑问，提升用户知识水平。

a) 说明及优先级

帮助中心模块由平台官方编写，包括竞赛指南、常用算法详解、数据结构基础等内容，旨在为用户提供权威、全面的参考资料。该模块帮助用户在遇到问题时快速找到答案，同时补充他们在竞赛中的知识空缺。由于该模块主要面对入门竞赛的用户，仅保证全面，但是不保证知识深度。因此将该模块优先级设为中。

b) 响应/激励

帮助中心会提供涵盖竞赛基本规则、常用算法分类及其应用、各类数据结构的操作和实践的内容。用户可以根据需要浏览不同主题的文档，帮助他们掌握解决实际问题所需的知识。通过这些简明易懂的内容，帮助中心有效降低了初学者的学习门槛，帮助他们快速上手竞赛，并通过实践获得初步的成功体验，从而激发他们进一步参与竞赛的热情和动力。

c) 功能需求

1. 知识库分类：

模块将提供系统化、详尽的内容分类，涵盖多个方面的知识资源，包括竞赛规则、竞赛技巧、常见算法的理论与具体实例分析，以及各种数据结构的详细讲解和实际操作指南。每个知识点都经过精心组织 and 分类，确保用户能够根据自己的学习需求有条不紊地获取相关信息。

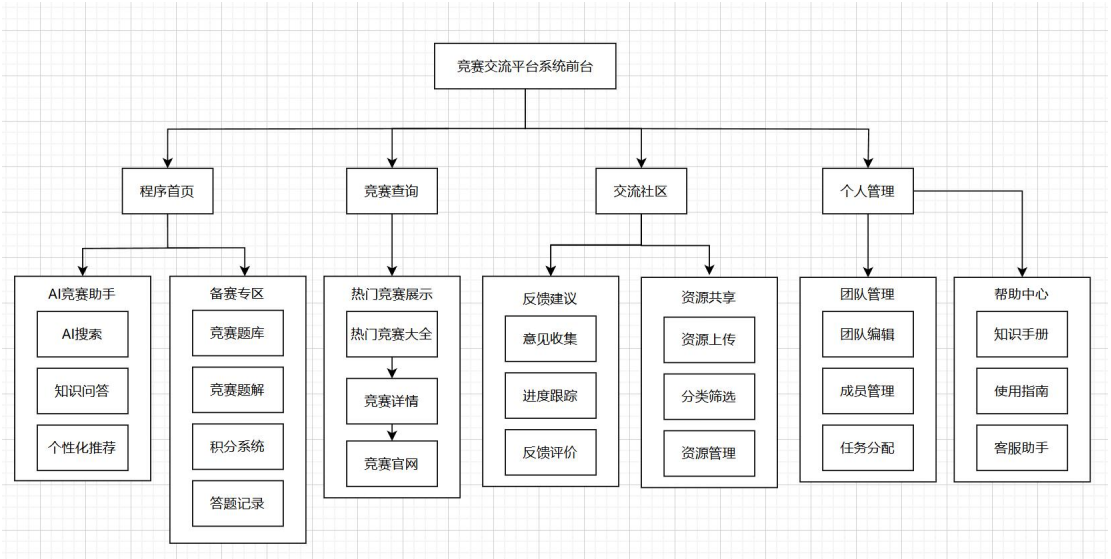
2. 搜索与导航：

模块将提供一个强大的搜索功能，用户可以通过输入关键词、选择标签或浏览分类，快速查找与自己具体问题相关的解决方案。这个搜索功能将支持模糊匹配和自动补全，使用户能够轻松找到相关的帮助文档，而无需记住所有的细节。此外，帮助中心还将配备一个清晰且直观的导航栏和目录，用户可以通过简单的点击，快速定位到所需的主容。

d) UI 设计图



系统前台功能图：



3.2 管理员主要功能

管理员后台为竞赛交流平台的管理人员提供竞赛项目、用户及系统整体的管理功能。通过后台，管理员可以高效地进行竞赛项目的发布、用户信息管理、团队管理等操作，从而确保平台的正常运行与维护。

3.2.1 管理员管理

系统超级管理员可以添加和管理其他管理员账户。管理员通过登录后台系统，可以查看、修改个人信息，并进行竞赛项目及用户的管理。系统支持管理员信息的增删改查操作，确保后台管理的灵活性与安全性。

功能需求：

- 1. 管理员登录：**管理员通过后台登录界面，使用账户和密码进入管理系统。
- 2. 添加管理员：**超级管理员可以创建新的管理员账户，设置其权限范围。
- 3. 修改管理员信息：**管理员可以修改个人信息，如密码和联系方式。
- 4. 删除管理员：**超级管理员可以删除不再需要的管理员账户，保持系统的安全和简洁。

3.2.2 竞赛项目管理

管理员负责平台所有竞赛项目的创建、编辑、删除和审核操作。通过竞赛项目管理模块，管理员可以发布新的竞赛项目，维护现有项目的更新，审核用户提交的项目申请，以及确保项目按时上线并更新。

功能需求：

- 1. 创建竞赛项目：**管理员填写竞赛项目的详细信息（如名称、时间、规则、报名方式等），并将其发布至平台前台供用户查看和报名。
- 2. 编辑竞赛项目：**管理员可以对已发布的竞赛项目进行修改，如更改竞赛时间、调整项目内容等。
- 3. 删除竞赛项目：**当竞赛项目完成或不再需要时，管理员可以将其删除。
- 4. 审核竞赛申请：**管理员可以审核用户提交的竞赛项目申请，决定是否批准其参与竞赛。

3.2.3 用户管理

管理员可以查看所有注册用户的信息，包括用户的注册时间、竞赛参与记录、团队情

况等。平台提供禁用用户的功能，管理员可以根据需要对违规用户进行处理，维护平台的秩序和安全。

功能需求：

- 1. 查看用户信息：**管理员可以查看系统中所有用户的详细信息。
- 2. 禁用用户：**对于违反平台规定或存在恶意行为的用户，管理员可以禁用其账户，限制其使用平台功能。
- 3. 恢复用户：**管理员可以对之前禁用的用户进行账户恢复操作。

3.2.4 团队管理

管理员可以查看、编辑和管理平台用户创建的竞赛团队。管理员能够在团队管理模块中进行团队成员管理、团队任务分配，以及解决团队内部的纠纷和问题。

功能需求：

- 1. 查看团队信息：**管理员可以查看所有团队的名称、成员信息及团队状态。
- 2. 编辑团队：**管理员可以对团队的详细信息进行修改，如调整团队成员、更新团队任务等。
- 3. 删除团队：**对于违反规定的团队或不再活跃的团队，管理员可以将其删除。

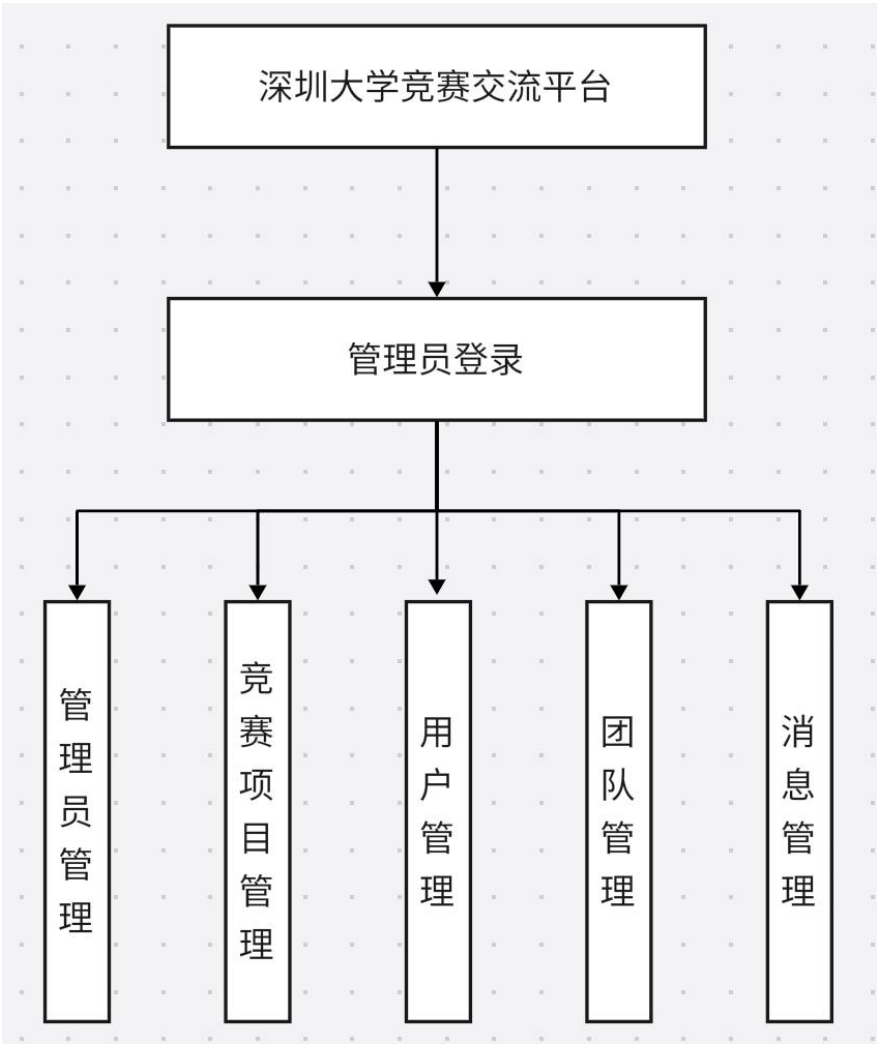
3.2.5 消息管理

管理员可以通过消息管理模块发送平台公告、竞赛通知和系统更新提示，确保所有用户及时接收到平台的重要信息。

功能需求：

- 1. 发布平台公告：**管理员可以在平台上发布公告，向用户通知平台的规则、更新等重要信息。
- 2. 竞赛通知推送：**管理员可以推送竞赛报名时间、比赛开始提醒等重要时间节点的通知。
- 3. 系统消息推送：**管理员可以发布系统升级或维护的通知，确保用户了解平台的最新动态。

系统后台功能图:



第四章 其它非功能需求

4.1 性能需求

4.1.1 系统可用性要求

确保系统操作便捷且内容全面，是满足用户使用的基石。为此，需精准把握各用户群体的特性、任务及使用环境，在以下方面全面满足用户需求：

- **有效性：**确保系统在完成特定任务和达成目标时，展现出高度的正确性和完整性。
- **效率：**提升任务完成的正确性和完整性，同时优化资源（如时间）的使用率。
- **满意度：**增强用户在使用产品过程中的主观满意度和接受度。

在可用性方面，系统需满足以下要求：

- **操作简便：**业务操作应直观明了，用户界面设计需符合用户操作习惯，便于业务人员通过简单点击按钮和超链接即可完成业务操作。
- **数据完整准确：**系统需具备强大的数据备份和恢复能力，实时监控数据可用性，确保数据的完整性和正确性。
- **容错性强：**尽量减少系统出错时间，提高系统容错性。为此，需对关键模块进行镜像处理，并对相互关联的模块进行解耦合，以延长系统的有效运行时间。

4.1.2 时间特性要求

- **系统处理能力：**系统需应对复杂业务与众多参与人员，因此必须具备出色的大用户量并发响应能力和高吞吐量。为实现这一目标，磁盘结构与数据库系统的选择需精心考虑。
- **并发用户处理能力：**系统需支持大量用户同时访问，且在处理多个用户请求时，必须保持响应速率稳定，避免降低或出现错误。
- **持续服务时间：**理想状态下，系统应提供7×24小时不间断服务。
- **响应速度要求：**每笔业务需迅速响应，响应速度应控制在用户可接受范围内，并提供明确提示，避免用户盲目等待。一般操作响应时间建议控制在1-2秒内，更新处理时间不超过1秒，数据转换和传送时间不超过5秒。具体数据以实际测试结果为准。
- **用户端性能：**用户端采用浏览器进行业务处理，页面下载时间应在30-60秒内完成。

网络实际吞吐量应至少达到1Mbps以上，以确保流畅的用户体验。

- **数据空间管理：**系统初期运行可能效率较高，但随着业务频繁处理，数据库中的插入、删除、修改等操作将日益增多，导致数据分散。这不仅影响存储空间利用，还可能降低数据检索和传输速率。因此，必须重视数据空间管理，确保系统长期稳定运行。

4.1.3 系统开放性要求

为了确保深圳大学本科生参与竞赛交流平台的持久可用性与灵活扩展性，本平台在系统开放性方面设定了以下核心要求：

- **模块化架构设计：**平台遵循模块化设计原则，确保各功能模块独立开发、测试及部署，为未来功能迭代与系统维护奠定坚实基础。
- **标准化API接口：**提供一套标准化、易用的API接口，便于第三方开发者无缝集成或扩展平台功能，促进生态繁荣。
- **数据交换兼容性：**全面支持JSON、XML等主流数据交换格式，确保平台数据与其他系统或服务间的高效互操作性。
- **广泛兼容性：**平台需与主流操作系统及浏览器全面兼容，保障用户在不同设备上均能享受一致且流畅的使用体验。
- **安全防护体系：**在保持开放性的同时，构建完善的数据传输与存储安全防护体系，严格遵守数据保护法规，确保用户数据安全无虞。
- **可扩展架构：**采用可扩展架构设计，确保平台能够轻松应对用户增长与功能扩充带来的挑战。
- **详尽文档与技术支持：**提供详尽的开发文档、API指南及持续的技术支持服务，助力第三方开发者快速上手并充分利用平台资源。
- **多语言支持：**平台内置多语言支持功能，满足不同语言背景用户的需求，提升国际化水平。
- **跨环境可移植性：**确保平台能在多种硬件与软件环境中稳定运行，便于在不同部署场景下灵活应用。
- **用户自定义功能：**提供用户自定义选项，如界面布局调整、功能模块选择等，增强用户体验的个性化与灵活性。

通过上述开放性要求的落实，深圳大学本科生参与竞赛交流平台将能够更好地适应未来变化，持续为用户创造更多价值。

4.1.4 系统可扩充性要求

为了确保深圳大学本科生参与竞赛交流平台在未来能够顺利地进行功能扩展和性能提升，同时保持与现有及未来系统的兼容性，以下是本项目的系统可扩充性要求：

- **遵循标准协议：**平台应遵循业界主流的标准和协议，如 RESTful API、OAuth 2.0 等，以确保与上级平台系统和其他业务系统的信息交换便捷性和安全性。
- **模块化架构：**平台应采用模块化架构设计，使得各个功能模块可以独立开发和维护，便于未来的功能扩展和升级。
- **灵活的定制能力：**系统应提供灵活的定制能力，允许对不同级别的用户需求进行层次化和模块化的定制，以满足不同用户群体的特定需求。
- **统一接口：**系统应提供与深圳大学现有平台统一的接口，确保将来能够无缝地与当前系统实现互连互通。
- **数据模型的可扩展性：**数据模型应设计得足够灵活，以支持未来可能增加的数据类型和结构，同时保证数据的一致性和完整性。
- **性能可扩展性：**系统应设计为可水平扩展的架构，以便在用户数量增加或数据量增大时，可以通过增加硬件资源或服务实例来提升系统性能。
- **服务的可替换性：**系统应设计为服务组件可替换，以便在技术更新或服务升级时，可以轻松替换旧的服务组件而不影响整个平台的运行。
- **易于集成：**系统应提供清晰的集成指南和工具，使得第三方服务和应用能够轻松集成到平台中。
- **用户界面的可定制性：**用户界面应允许一定程度的定制，以适应不同用户的操作习惯和视觉偏好。
- **文档和培训：**提供详细的开发文档、API 文档和用户手册，以及必要的培训支持，以帮助用户和开发者理解和使用平台。
- **支持多语言：**系统应支持多语言，以适应不同语言背景的用户，提高系统的国际化水平。
- **安全性考虑：**在设计可扩展性的同时，系统必须考虑到安全性，确保在扩展过程中不会引入新的安全漏洞。

4.1.5 界面友好性要求

在软件开发中，界面友好性（User Interface, UI）是提升用户体验的关

键因素。为了进一步优化界面友好性，我们可以从以下几个方面进行深入探讨和改进：

1. 直观与一致性

- 界面应符合用户习惯，设计直观，提供清晰的导航和操作指引。
- 保持设计元素的一致性，如图标、字体和颜色。

2. 响应与性能优化

- 界面应对用户操作做出快速响应，并适配各种设备，确保操作流畅。
- 优化加载时间，尤其是大数据和图像密集型应用。

3. 易用与简洁

- 操作应易于学习和使用，提供简洁的界面设计，减少用户的认知负担。
- 通过合理分组与结构引导用户完成关键任务。

4. 反馈与容错性

- 系统应提供及时且明确的操作反馈，并允许用户撤销或纠正错误。
- 针对错误操作提供清晰的提示和解决方案。

5. 可访问与灵活性

- 确保界面对所有用户友好，提供无障碍选项，如屏幕阅读器支持和高对比度模式。
- 允许用户自定义界面布局或操作方式，满足不同需求。

4.2 安全设施需求

4.2.1 硬件安全主要在于服务器硬件的安全

- 确保服务器电力供应稳定，实现不间断电源保障。
- 由于硬件安全问题难以避免且不可预测，为规避风险，需确保系统冗余性。通过部署集群或至少保留一台可随时启用的闲置服务器，以便在硬件严重故障导致宕机时迅速恢复运行，最大限度减少故障停机时间。
- 选择信誉良好、知名度高的服务器供应商，以确保在硬件遇到问题时能获得优质的售后服务支持。

4.2.2 网络安全

- 为确保网络服务连续性，网络连接需具备冗余性。当一条网络接入出现故障时，另一条备用网络连接应立即启用，以维持不间断服务。即便在仅有一条网络接入的条件下，也应通过技术手段确保服务的持续可用性。
- 网络阻塞和掉包问题在国内尤为普遍。为解决这一问题，需在程序设计中融入响应、队列管理、应答确认及数据校验等机制，以有效提升数据传输的有效性和稳定性。
- 网络监控是保障网络稳定性和质量的关键。应对网络状态进行持续监控，并进行历史数据分析。条件允许时，可对整个网段的IP及端口流量进行记录与分析，但为节省存储空间，建议采用间断性历史数据分析策略，并定期清理旧数据。
- 为防范网络攻击，需利用防火墙对内网进行严密保护，确保端口安全。

4.3 安全性需求

4.3.1 系统安全

在系统层级上，为防止外部入侵及系统数据丢失，需采取以下安全措施：

首要任务是确保操作系统的稳定与安全。建议选择性能与安全性均表现优异的Unix系统或其他经过安全验证的操作系统，并尽量减少不必要的对外服务接口。同时，实施严格的帐户管理策略，确保帐户安全无虞。

其次，建立系统事件监控机制至关重要。通过详细记录并分析系统日志，可以实时掌握系统运行状况，及时发现潜在问题。同时，对网络事件进行监控也是必不可少的环节，有助于防范网络攻击。

此外，还可以借助专业的安全工具进行系统安全评估，以进一步提升系统的安全防护能力。

4.3.2 程序安全

- 制定严格的开发协议规范，以防止在程序与系统对接过程中发生密码、源代码及软件架构的非必要泄露。
- 为应对可能的恶意连接耗尽攻击，需优化 web 服务。具体措施包括：减少非动态页面数量，对无法静态化的程序实施智能判断，并设定单位时间内的提交次数上限，以有效抵御此类攻击。
- 针对特定类型的网络攻击，应在防火墙层面加强防护。这包括主动关闭不必要的服务端

口，切断多余连接，以及采取其他必要措施来及时中止攻击。

- 对于持续连接的情况，需进行特别处理，以确保系统资源的合理利用和安全性。
- 在数据库层面，需加强数据保护措施，防止因程序缺陷导致的数据丢失或损坏。这要求开发者在数据库设计和编程过程中，充分考虑数据的安全性和完整性。

4.3.3 数据安全

为确保数据安全，应实施有效的备份与恢复机制。具体措施包括：

1. 对数据进行细致分类，根据数据的重要性和敏感性，制定差异化的备份策略。
2. 设定合理的备份时间频率，确保关键数据能够得到及时、定期的备份。
3. 选择可靠的备份媒介，如磁盘阵列、磁带库或云存储等，以提高备份数据的可靠性和可访问性。
4. 根据数据的恢复需求，设定合理的恢复时间目标（RTO）和恢复点目标（RPO），确保在数据丢失或损坏时能够迅速恢复业务运行。

通过实施这些措施，可以最大程度地保障数据的安全性和可用性。

4.3.4 内容安全

在数据流层面，对敏感信息进行传输和存储时需实施加密处理，有效防范人为窃取数据的风险。在用户操作层面，系统需具备阻止危险操作的能力，通过发出警告并提供错误恢复机制，确保用户操作的安全性。此外，在用户完成操作并提交后，系统应迅速进行数据同步，以确保数据的一致性和完整性。