**《管理信息系统》课程作业**

# 在线协作网盘信息系统文档

**摘要：**内容文字用楷体，五号字

**关键词：**楷体；五号字；不多于五个词；各词之间用分号分隔

## 介绍

## 系统规划

1. 项目需求

随着互联网技术的发展，企业和组织越来越需要一种在线协作平台及网盘来方便员工进行远程办公以及团队合作。某企业内部要求开发一个项目在线协作、存储及验收的系统。以下是这样一个项目的必要性：

1. 方便远程办公

现在很多企业已经将其业务转向了线上，在全球范围内招聘员工或者让员工在家办公，其中涉及到数据存储、共享、实时编辑等问题。因此，建立一个在线协作平台及网盘可以方便所有员工在任何地点进行远程办公，从而提高工作效率。

2. 提高团队协作效率

在线协作平台及网盘不仅可以方便员工个人的远程办公，也可以实现团队内部的协作。通过文件共享和实时编辑等功能，员工可以更容易地互相交流信息、分享文档和数据，并且快速完成对共同项目的任务分配。

3. 安全可靠

在企业级别使用在线协作平台及网盘时，安全性和数据保护是非常重要的。该项目应该具有强大的数据加密和权限管理能力，确保只有授权的用户才能访问敏感数据。

4. 灵活的存储管理

随着公司业务不断扩展，数据量也会不断增加。因此，该项目应该提供灵活的存储管理功能，允许企业根据实际需求调整存储容量，并确保数据可以被安全地备份和恢复。

5. 可定制化的工作流程

不同的企业和组织需要不同的工作流程和协作方式。在线协作平台及网盘应该提供可定制化的工作流程来满足用户的不同需求，例如自定义任务分配、审批流程等。

**预期收益**

1. 提高工作效率：使得员工在任何地点都能够使用设备进行远程办公，并且快速完成任务。
2. 改善团队协作：员工可以轻松地共享文件和数据，并且实时编辑相同的文档或表格，增强团队之间的沟通和合作能力。
3. 保证数据安全可靠：确保企业的敏感数据不被未经授权的人访问。在备份和恢复方面也有较强的能力，可靠性更高。
4. 提高客户满意度：更加快速地响应客户的需求，提供更优质的服务。如果团队与客户之间能够进行更好的协作和沟通。

**系统目标**

1. 支持文件上传、下载、编辑与查看；根据团队、项目、个人的不同分配不同的存储空间。
2. 提供实时协作和沟通工具：系统旨在提供实时协作和沟通功能，使团队成员能够在同一平台上共享想法、讨论问题并迅速做出决策。通过提供即时聊天、在线会议和协作工具，系统促进了团队成员之间的交流和合作。
3. 管理权限和访问控制：系统将提供灵活的权限管理和访问控制功能，以确保只有经授权的人员可以访问特定文件和数据。管理员可以根据团队、项目和个人的需求设置权限级别，并限制对敏感信息的访问。这样可以保护企业的机密信息，并确保数据的安全性。
4. 管理权限和访问控制：系统将提供灵活的权限管理和访问控制功能，以确保只有经授权的人员可以访问特定文件和数据。管理员可以根据团队、项目和个人的需求设置权限级别，并限制对敏感信息的访问。这样可以保护企业的机密信息，并确保数据的安全性。
5. 提供易用性和个性化定制：系统将注重用户友好的界面设计和易用性，以确保用户能够快速上手并高效地使用系统。此外，系统还将提供一些个性化定制选项，使用户能够根据自己的喜好和工作需求进行界面和功能的定制，提供更加个性化的体验。
6. 初步关联图
7. 可行性论证

1. 经济可行性

在线协作平台及网盘是一种基础设施项目，其实施涉及多个方面的经济成本。然而，从长远来看，该项目可以在减少基础设施建设成本、提高工作效率、改善团队协作、增强数据安全可靠性等方面带来显著的经济收益。因此，该项目具有良好的经济可行性。

2. 技术可行性

在线协作平台及网盘需要使用先进的互联网技术，如云计算、大数据、人工智能等。这些技术已经得到了广泛的应用和发展，并且在很多企业中有着成功的案例。因此，该项目具有较强的技术可行性。

3. 时间可行性

在线协作平台及网盘的开发、测试和部署需要一定的时间，但是这个时间不会超过其他类似项目的时间。另外，在软件开发周期内，可以采用敏捷开发和迭代开发的方式，来确保项目可以按时交付。因此，该项目具有良好的时间可行性。

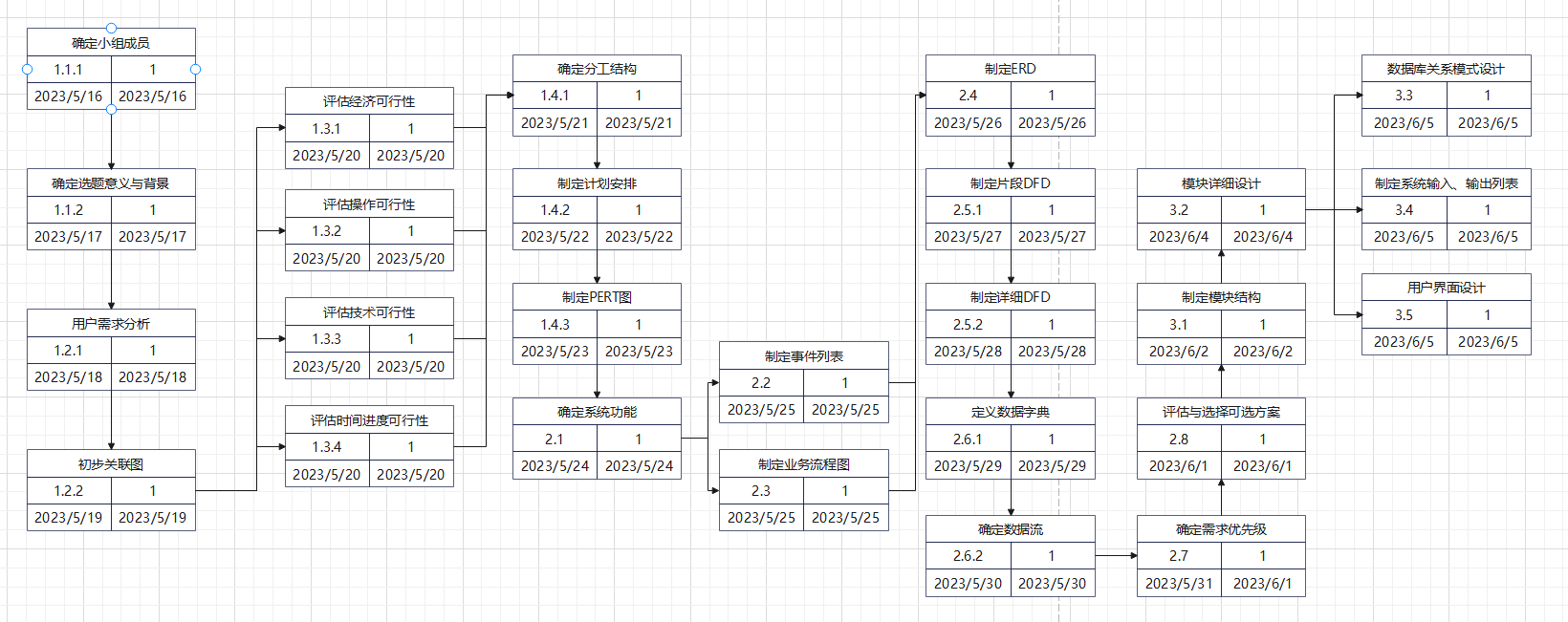
4. 社会可行性

随着互联网技术的发展，越来越多的人开始意识到远程办公和线上协作的优势。在线协作平台及网盘可以促进企业团队之间的协作，提高工作效率，并且符合节约能源减少污染等方面的社会价值观。因此，该项目具有较强的社会可行性。

综上所述，从经济、技术、时间和社会四个方面论证了在线协作平台及网盘项目的可行性。结论：项目可行。

1. 项目计划与PERT图

|  |  |  |  |
| --- | --- | --- | --- |
| 标识号 | 阶段、活动和任务 | 最可能的持续时间 | 先行任务 |
| 1 | 项目计划与系统规划 | | |
| 1.1 | 项目计划 | | |
| 1.1.1 | 确定小组成员 | 1 |  |
| 1.1.2 | 确定选题意义与背景 | 1 | 1.1.1 |
| 1.2 | 系统规划 |  |  |
| 1.2.1 | 用户需求分析 | 1 | 1.1.2 |
| 1.2.2 | 初步关联图 | 1 | 1.2.1 |
| 1.3 | 确定项目可行性 | | |
| 1.3.1 | 评估经济可行性 | 1 | 1.2.2 |
| 1.3.2 | 评估操作可行性 | 1 | 1.2.2 |
| 1.3.3 | 评估技术可行性 | 1 | 1.2.2 |
| 1.3.4 | 评估时间进度可行性 | 1 | 1.2.2 |
| 1.4 | 制作项目进度表 | | |
| 1.4.1 | 确定分工结构 | 1 | 1.3.1 1.3.2 1.3.3 1.3.4 |
| 1.4.2 | 制定计划安排 | 1 | 1.4.1 |
| 1.4.3 | 制定PERT图 | 1 | 1.4.2 |
| 2 | 系统分析 | | |
| 2.1 | 确定系统功能 | 1 | 1.4.3 |
| 2.2 | 制定事件列表 | 1 | 2.1 |
| 2.3 | 制定业务流程图 | 1 | 2.1 |
| 2.4 | 制定ERD | 1 | 2.2 2.3 |
| 2.5 | 制定DFD | | |
| 2.5.1 | 制定片段DFD | 1 | 2.4 |
| 2.5.2 | 制定详细DFD | 1 | 2.5.1 |
| 2.6 | 数据库分析 | | |
| 2.6.1 | 定义数据字典 | 1 | 2.5.2 |
| 2.6.2 | 确定数据流 | 1 | 2.6.1 |
| 2.7 | 确定需求优先级 | 1 | 2.6.2 |
| 2.8 | 评估与选择可选方案 | 1 | 2.7 |
| 3 | 系统设计 | | |
| 3.1 | 制定模块结构 | 1 | 2.8 |
| 3.2 | 模块详细设计 | 1 | 3.1 |
| 3.3 | 数据库关系模式设计 | 1 | 3.2 |
| 3.4 | 制定系统输入、输出列表 | 1 | 3.2 |
| 3.5 | 用户界面设计 | 1 | 3.2 |

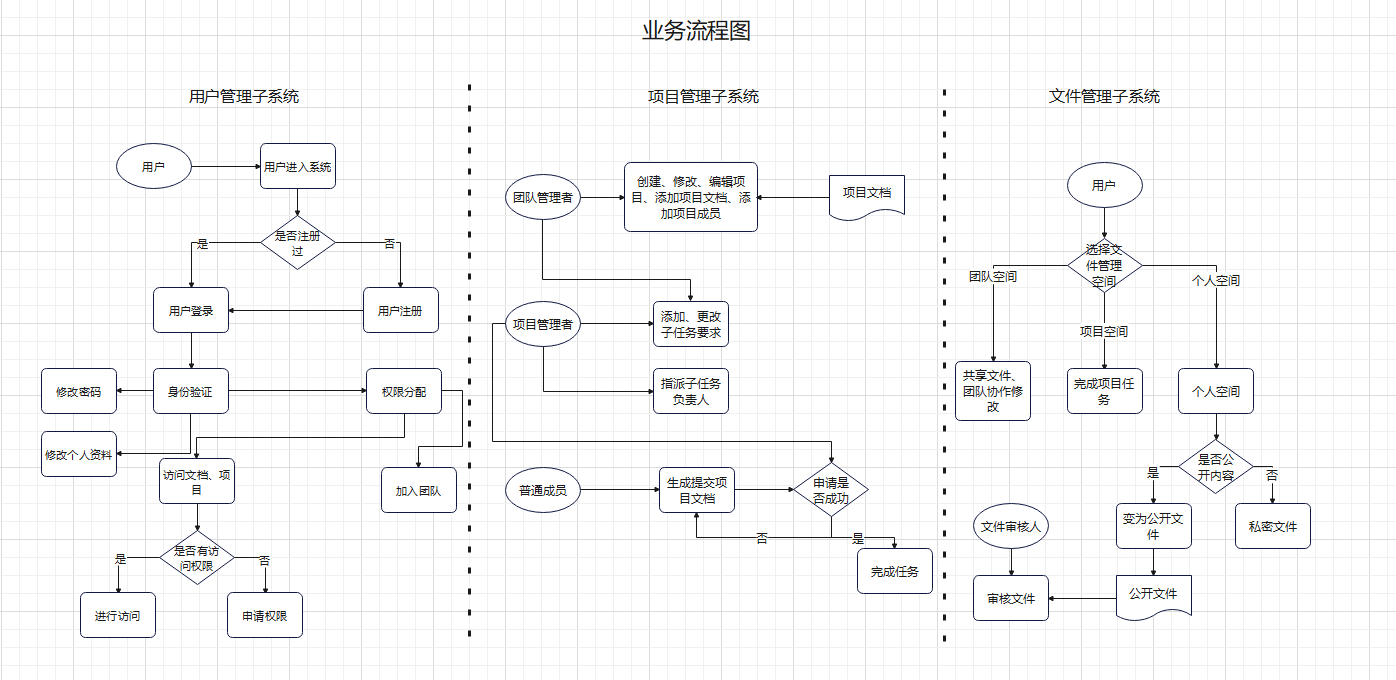


## 系统分析

1. 系统功能分析
2. 业务流程图与事件列表

1、业务流程图

为了能够反映共享协作网盘系统内部各功能之间的工作关联关系，根据共享协作网盘管理的特点和实际工作业务流程，总结并绘制的业务流程图如图所示：



如上图业务流程图所示，系统的主要相关角色包括文档系统中的用户和文件审核人，其中，用户（包括普通成员、项目管理者与团队管理者）与文件审核人。其中，用户可以选择加入团队、加入项目、上传修改文档等，当用户将文档转为公开文档后，文件审核人会对其进行审核。在项目管理子系统中，团队管理者可以创建团队、指定项目管理者等。管理者负责子任务完成的监督，审核普通成员提交完成文档的结果。若审核不通过，则普通成员重新生成项目文档。

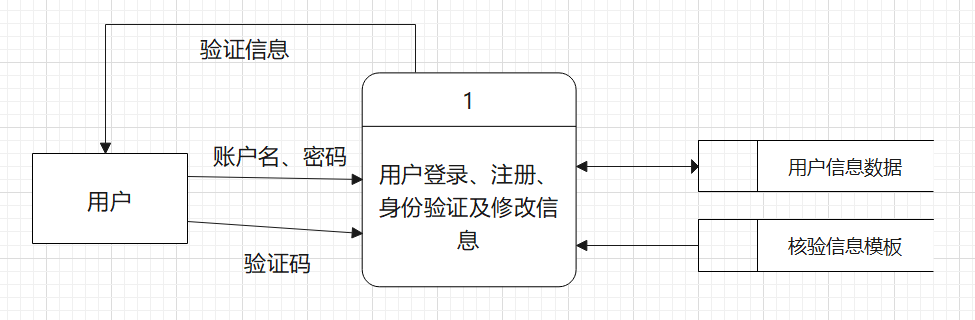
2、事件列表

本系统需要处理的共享协作网盘业务主要涉及事件在以下事件列表中较为清晰明确的体现：

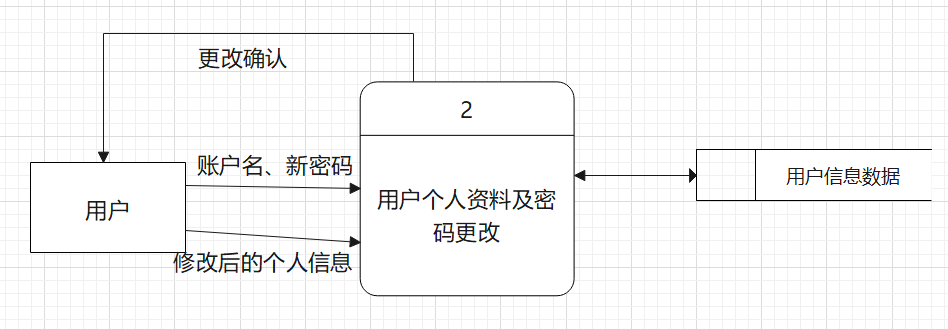


1. ER图
2. 数据流程图

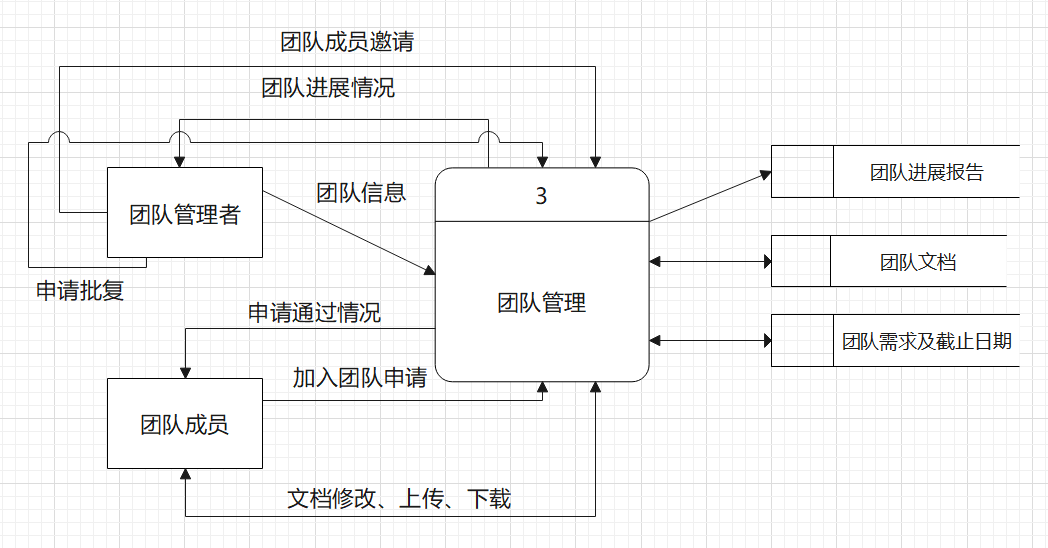
由于关联图只能帮助理解系统范围的概貌，反映系统与系统环境的交互作用，但它并没有描述发生在系统内部的有关处理过程的任何细节，所以我们要进一步采用 DFD 片段和详细 DFD 来描述系统内部的处理细节。



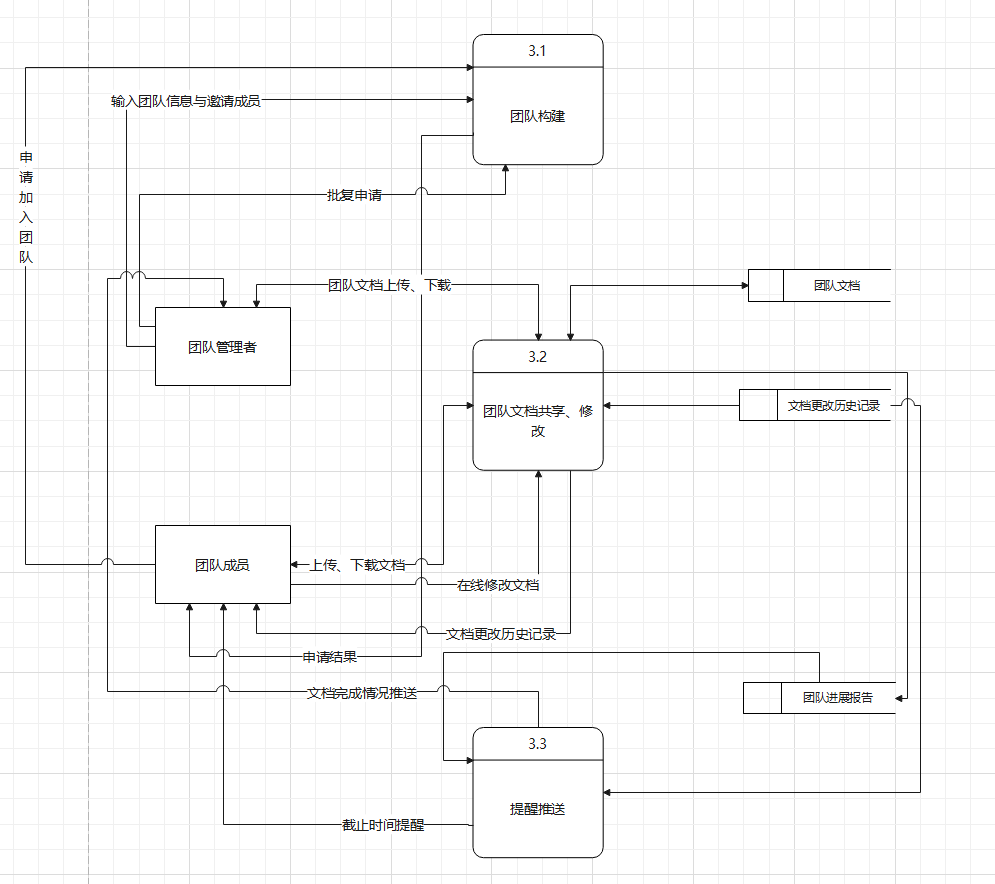
首先，用户进行登录以进入系统，如果在用户信息数据中没有查询到该用户的用户名，则需要进行注册，注册时需要对用户信息数据进行添加。登录密码正确后，由系统发送核验信息进行身份验证，用户将验证码输入系统以成功登录系统。

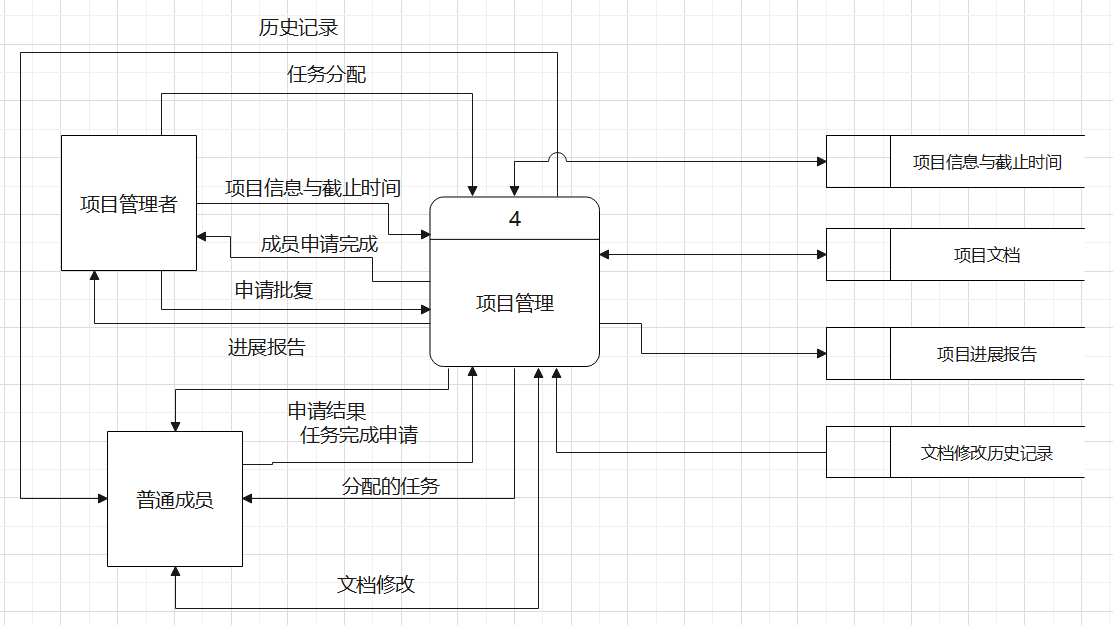


用户通过输入新的信息更改原始信息、密码等，首先从用户信息数据中显示原始信息，在接受更改确认后，将用户信息数据更新。

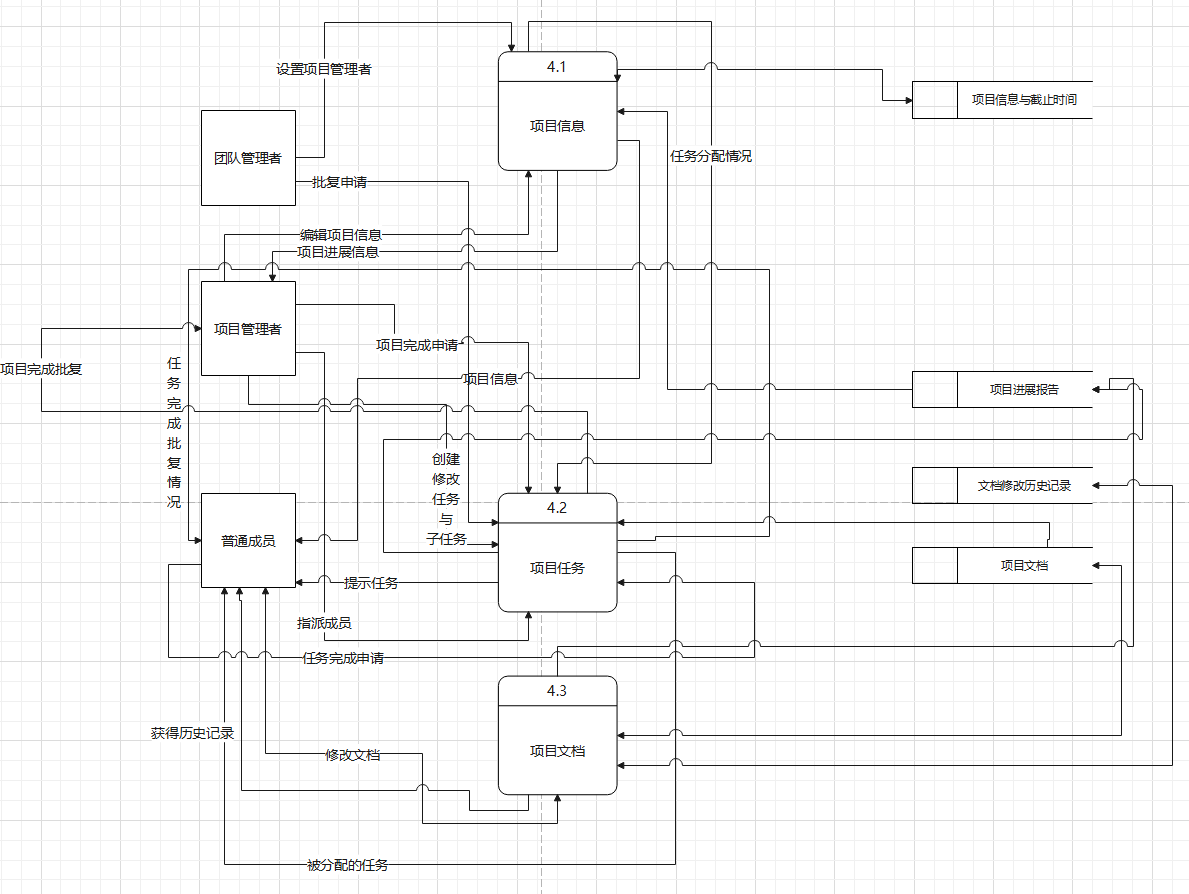


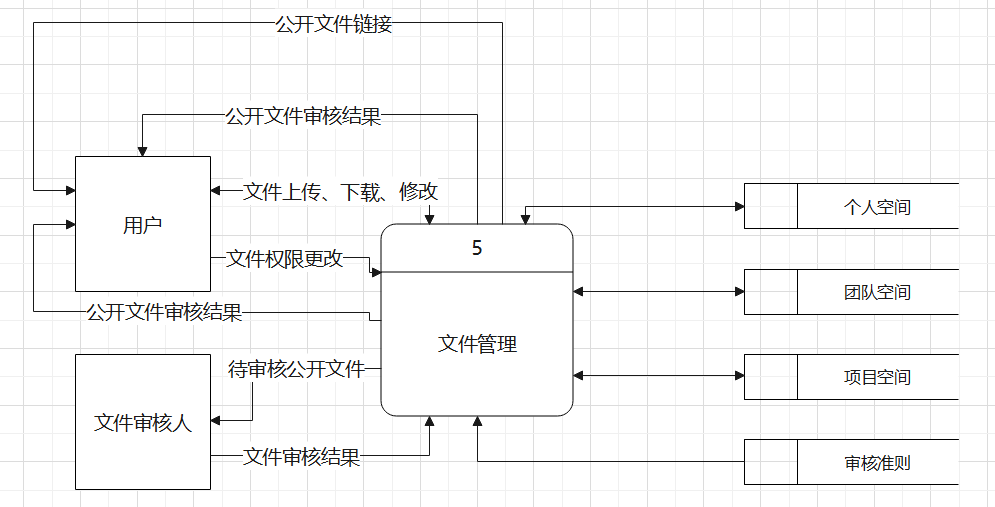
在团队管理功能中，团队管理者可以创建、修改、编辑项目说明与截止日期、添加项目文档等，并可以添加项目成员，针对团队成员的加入申请进行批复。团队成员可以选择加入团队，上传、下载和共享文件，并支持团队成员之间的协作、编辑与标注评价；能够跟踪文件的历史记录，并提供数据备份和恢复功能来保障数据的安全和可靠性；系统可以自动对于团队空间进行归档处理。由于一系列过程比较复杂，还可以进一步细化，所以绘制了以下详细流程图：





在项目管理部分，项目管理者负责管理创建、修改、编辑项目的任务和子任务，包括任务要求、时间需求与资源使用情况；可以指派任务与子任务的负责人；普通成员通过提交文档或说明以申请完成任务；项目管理者通过审核完成情况来选择通过申请或驳回申请。支持团队成员之间的协作和沟通，并提供一些自动工具和报告来跟踪项目的进展情况。能够跟踪文件的历史记录，并提供数据备份和恢复功能来保障数据的安全和可靠性。用户可以从系统中获取用户待完成的任务，如用户管理中的待完善资料，项目管理中的任务截止日期以及根据截止日期生成待完成项目列表与日期列表。由于一系列过程比较复杂，还可以进一步细化，所以绘制了以下详细流程图：





在文件管理区域，分为团队空间、项目空间与个人空间进行管理；用户可以进行上传、下载和共享修改文件，文件能够进行实时预览；个人空间可以进行私密与公开管理，对于公开文件，其他用户可通过链接访问评论；对于公开数据，文件审核人可以进行文件审核。

1. 数据字典
2. 系统解决方案及其决策
3. 硬件环境

集中式计算系统将所有的计算和存储资源集中在一个中央服务器上，用户通过终端设备远程访问和使用系统。在硬件环境上，可以考虑以下方面：

服务器：选择高性能的服务器，具备强大的处理能力和存储容量，以支持大规模文件上传、下载和编辑操作，以及并发用户的需求。服务器应具备高可靠性和可扩展性，以满足系统的性能和容量需求。

网络设备：选择高速、稳定的网络设备，包括交换机、路由器和防火墙等，以确保用户能够快速访问系统并进行实时协作。网络设备应支持高带宽和低延迟的数据传输，以提供流畅的用户体验。

存储设备：选择高容量、高可靠性的存储设备，如硬盘阵列（RAID）或存储区域网络（SAN），以满足系统对大量文件和数据的存储需求。存储设备应具备数据备份和恢复功能，以保证数据的安全性和可靠性。

客户端设备：用户通过终端设备访问系统，可以选择常见的台式机、笔记本电脑或移动设备。客户端设备应具备适当的计算能力和网络连接能力，以支持用户的日常工作和远程访问需求。

硬件环境的选择应根据系统的需求、预期负载和可扩展性考虑。在实际部署中，可能需要进行性能测试和负载测试，对需要的硬件环境进行选择。

1. 开发环境和系统软件环境

开发环境：

1. IDE（集成开发环境）：选择使用IntelliJ IDEA对后端进行开发，VSCode对前端进行开发。

2. 编程语言和规范：使用Python作为后端开发语言，使用FastAPI的Flask框架作为后端开发的规范与框架。

系统软件环境：

1. 操作系统：使用Linux CentOS7作为服务器操作系统。

2. 数据库管理系统（DBMS）：考虑数据的安全性、性能和可扩展性，选择使用已有服务使用的MySQL。

选择合适的开发环境和系统软件环境应综合考虑系统需求、开发团队的技术能力和项目约束。确保所选环境能够满足开发需求、提高开发效率，并与系统软件环境相兼容，以便实现系统的目标和预期收益。

1. 系统需求优先级及自动化

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 功能 | 优先 | 低等自动化 | 中等自动化 | 高级自动化 |
| 审查公开资源内容 | 必需 | 系统提供基本的资源内容筛选功能，例如根据关键词进行搜索和过滤。 | 系统能够自动识别和标记一些常见的违规内容，如敏感词汇或不当图片 | 系统能够利用机器学习和自然语言处理技术，自动识别和过滤大量的违规内容，提供更准确和智能的审查功能 |
| 上传个人与项目文件 | 必需 | 系统提供基本的文件上传功能，用户可以手动选择文件并上传到系统中 | 系统能够自动检测和处理一些常见的文件格式问题，如文件格式转换或校验 | 系统能够根据团队成员的角色和权限，自动控制团队管理操作的访问和执行，提高管理流程的自动化程度 |
| 添加、修改、删除团队 | 必需 | 系统提供基本的团队管理功能，管理员可以手动添加、修改和删除团队信息 | 系统能够自动验证和处理一些团队管理的合规性，如检查团队名称的唯一性或成员的权限设置 | 系统能够利用机器学习和自然语言处理技术，自动识别和过滤大量的违规内容，提供更准确和智能的审查功能 |
| 添加、修改、删除项目 | 必需 | 系统提供基本的项目管理功能，管理员可以手动添加、修改和删除项目信息 | 系统能够自动验证和处理一些项目管理的合规性，如检查项目名称的唯一性或项目状态的合法性 | 系统能够自动验证和处理一些项目管理的合规性，如检查项目名称的唯一性或项目状态的合法性 |
| 提交文件以完成任务 | 必需 | 系统提供基本的任务提交功能，用户可以手动选择文件并将其提交给相关人员 | 系统能够自动验证和处理一些常见的提交问题，如文件格式的兼容性或大小限制 | 系统能够根据任务要求和文件内容，自动检测和分析文件的完整性和质量，提供更准确和智能的提交功能 |
| 审核文件，通过/驳回任务 | 必需 | 系统提供基本的任务审核功能，管理员可以手动审核并标记任务的通过或驳回状态 | 系统能够自动检测和处理一些常见的审核问题，如文件的完整性或合规性 | 系统能够利用机器学习和智能算法，自动分析文件的内容和任务要求，自动判断文件是否符合要求，并自动标记任务的通过或驳回状态。系统还可以根据历史审核数据和模式识别，提供更准确和智能的审核功能 |
| 查看项目当前进度 | 必需 | 系统提供基本的项目进度查看功能，用户可以手动查看项目的当前进度，如已完成和待完成的任务数量 | 系统能够自动汇总和计算项目的进度信息，提供可视化的图表或指标，以便用户更直观地了解项目的当前进展 | 系统能够根据任务的完成情况和时间进度，自动预测项目的完成时间，并提供预警和提示，帮助用户及时调整和管理项目进度 |
| 协同编辑文档 | 必需 | 系统提供基本的协同编辑功能，多个用户可以同时编辑同一个文档，但需要手动处理并合并冲突 | 系统能够自动检测和解决一些常见的协同编辑冲突，例如同时修改同一段落的内容，系统会提供冲突解决工具或提示 | 系统能够自动检测和解决一些常见的协同编辑冲突，例如同时修改同一段落的内容，系统会提供冲突解决工具或提示 |
| 查看文档历史记录 | 必需 | 系统能够自动检测和解决一些常见的协同编辑冲突，例如同时修改同一段落的内容，系统会提供冲突解决工具或提示 | 系统能够自动记录文档的修改历史，包括版本号、修改时间和修改内容的摘要，用户可以方便地查看和比较不同版本的文档 | 系统能够自动跟踪和保存文档的详细修改历史，包括每个用户的具体修改操作和变更内容，用户可以精确地查看和还原文档的任意历史版本 |
| 提供文档备份功能 | 必需 | 系统提供基本的文档备份功能，用户可以手动选择并备份文档到指定的位置 | 系统提供基本的文档备份功能，用户可以手动选择并备份文档到指定的位置 | 系统能够实现自动化的文档备份和恢复流程，包括定期自动备份文档、备份版本的管理和存储、自动检测备份完整性等。系统还可以提供灾难恢复功能，如数据灾难时自动恢复备份的文档 |

1. 系统实施方式

由于是对互联网企业及其相关企业内部使用，更注重安全性与快捷性，同时考虑使用人员具有一定的技术水平与自定义需求，因此使用自行开发的实施方案。

1. 结论

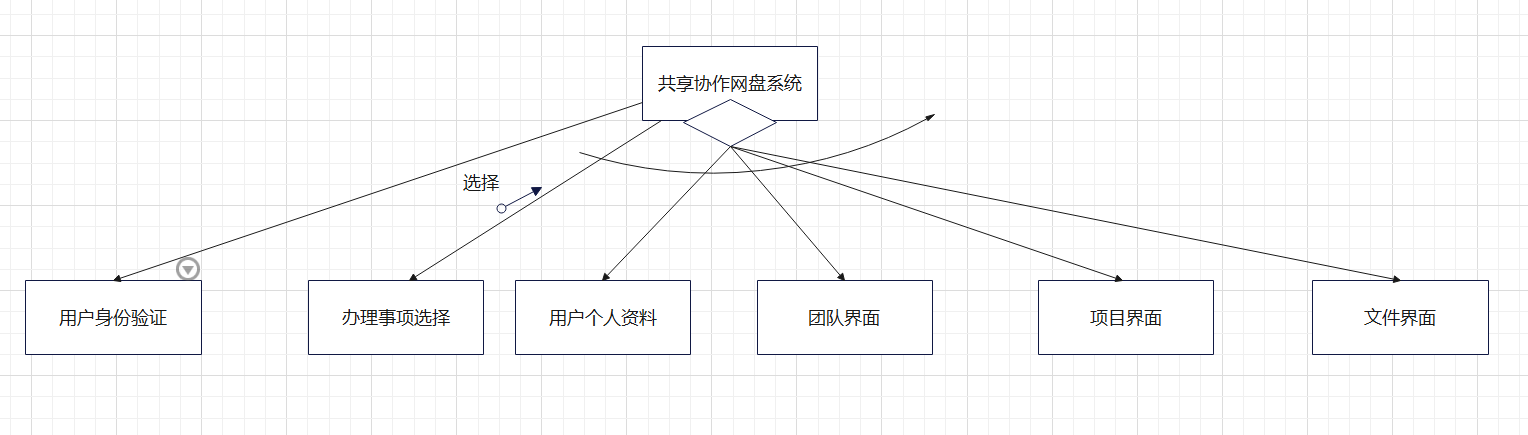
环境与需求范围及自动化如上，采用自行开发的实施方案进行开发。

## 系统分析

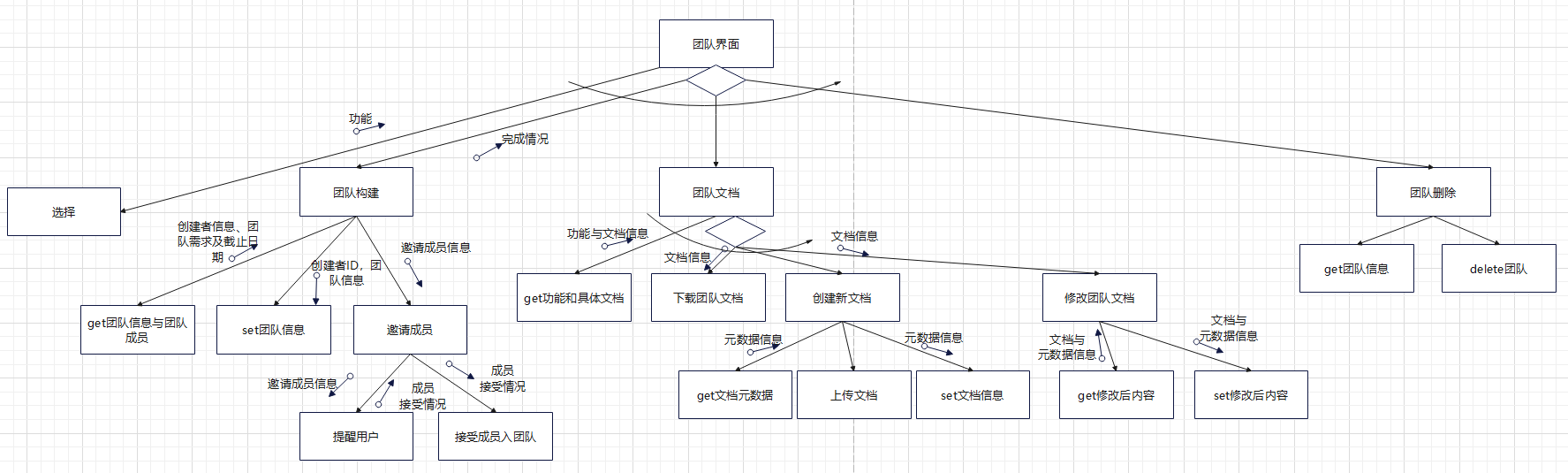
1. 模块结构图与部分设计

由于空间原因，将所有模块拆分成各个部分进行展示。

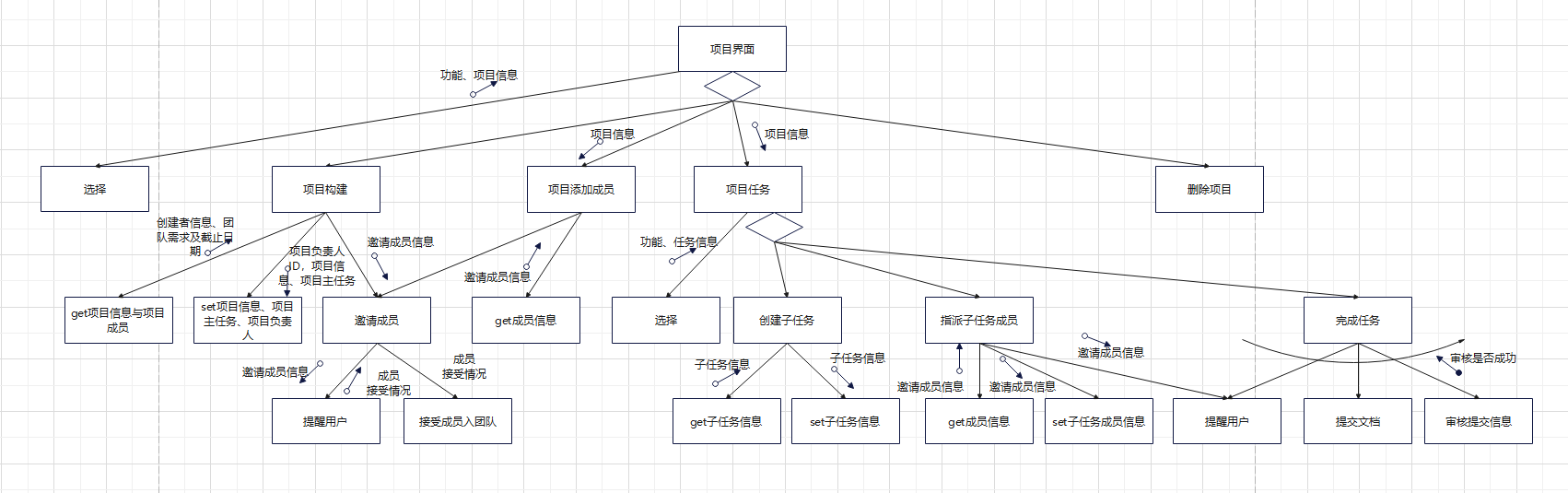
**总模块**



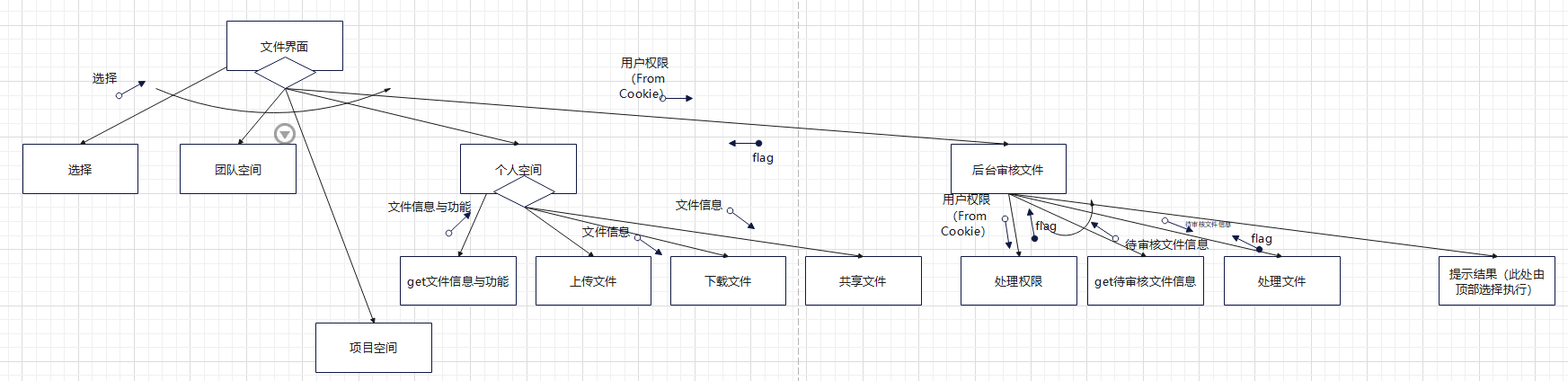
**团队界面**

****

**项目界面**

****

**文件界面**

****

**部分模块内部处理逻辑：**

1. **审查公开资源内容**

**def reviewPublicResource(resource):**

**if detectViolation(resource):**

**markAsViolated(resource)**

**notifyAdmin(resource)**

**def detectViolation(resource):**

**# 检测资源是否包含违规内容的逻辑**

**if violationDetected:**

**return True**

**else:**

**return False**

**def markAsViolated(resource):**

**# 标记资源为违规的逻辑**

**resource.setViolationStatus(True)**

**def notifyAdmin(resource):**

**# 发送通知给管理员的逻辑**

**message = "资源：" + resource.getName() + " 包含违规内容，请及时处理。"**

**admin.notify(message)**

1. **上传个人与项目文件**

**def uploadFile(user, project, file):**

**if isAllowedStorage(user, project, file):**

**storeFile(user, project, file)**

**updateStorageUsage(user, file)**

**def isAllowedStorage(user, project, file):**

**# 检查用户、项目和文件的存储权限的逻辑**

**If user.hasStoragePermission() and project.hasStorageSpace(file.getSize()):**

**return True**

**else:**

**return False**

**def storeFile(user, project, file):**

**# 存储文件的逻辑**

**storageLocation = project.getStorageLocation()**

**storageLocation.store(file)**

**def updateStorageUsage(user, file):**

**# 更新用户存储空间使用情况的逻辑**

**user.updateStorageUsage(file.getSize())**

1. **添加、修改、删除团队**

**def addTeam(teamName):**

**if isUniqueTeamName(teamName):**

**createTeam(teamName)**

**def isUniqueTeamName(teamName):**

**# 检查团队名称是否唯一的逻辑**

**existingTeams = getAllTeams()**

**for team in existingTeams:**

**if team.getName() == teamName:**

**return False**

**return True**

**def createTeam(teamName):**

**# 创建团队的逻辑**

**team = new Team(teamName)**

**saveTeamToDatabase(team)**

1. **添加、修改、删除项目**

**def addProject(projectName):**

**if isUniqueProjectName(projectName):**

**createProject(projectName)**

**def isUniqueProjectName(projectName):**

**# 检查项目名称是否唯一的逻辑**

**existingProjects = getAllProjects()**

**for project in existingProjects:**

**if project.getName() == projectName:**

**return False**

**return True**

**def createProject(projectName):**

**# 创建项目的逻辑**

**project = new Project(projectName)**

**saveProjectToDatabase(project)**

1. **提交文件以完成任务**

**def submitTask(user, task, file):**

**if isTaskAssignedToUser(task, user):**

**if meetsSubmissionRequirements(file, task):**

**markTaskAsSubmitted(task, user, file)**

**def isTaskAssignedToUser(task, user):**

**# 检查任务是否分配给指定用户的逻辑**

**assignedUser = task.getAssignedUser()**

**if assignedUser == user:**

**return True**

**else:**

**return False**

**def meetsSubmissionRequirements(file, task):**

**# 检查文件是否符合任务提交要求的逻辑**

**submissionRequirements = task.getSubmissionRequirements()**

**if file.meetsRequirements(submissionRequirements):**

**return True**

**else:**

**return False**

**def markTaskAsSubmitted(task, user, file):**

**# 标记任务为已提交的逻辑**

**task.setSubmitted(user, file)**

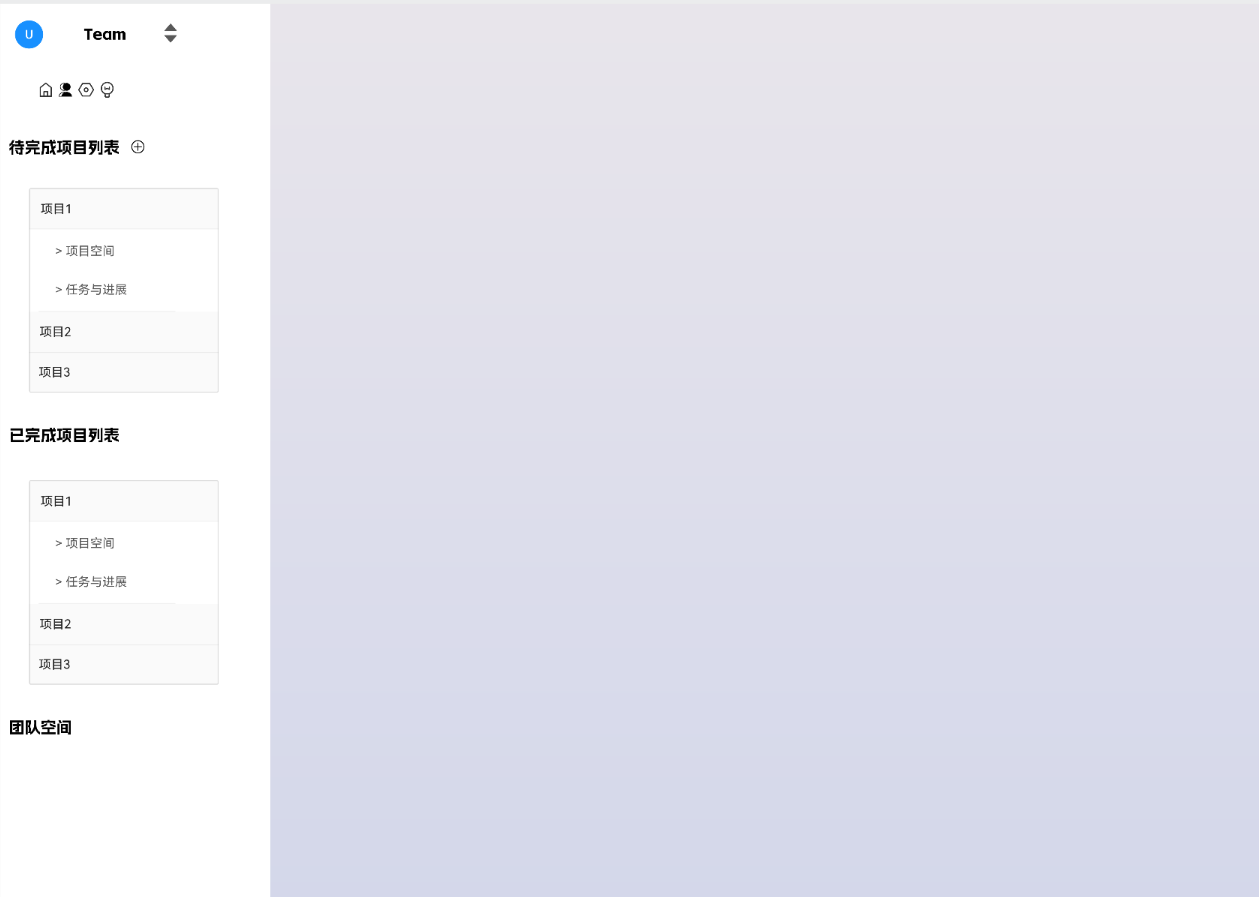
1. 数据库设计
2. 系统输入输出设计
3. 用户界面设计

用户界面设计作为用户和计算机系统之间所进行的所有交互活动的设计，应本着以用户为中心的设计思路。而本系统遵循具备可视性和提示性、一致性、反馈、防止出错和出错处理、允许撤销操作、最少记忆量的原则，对用户界面进行设计。 本文采用故事脚本的对话模型进行用户界面设计。

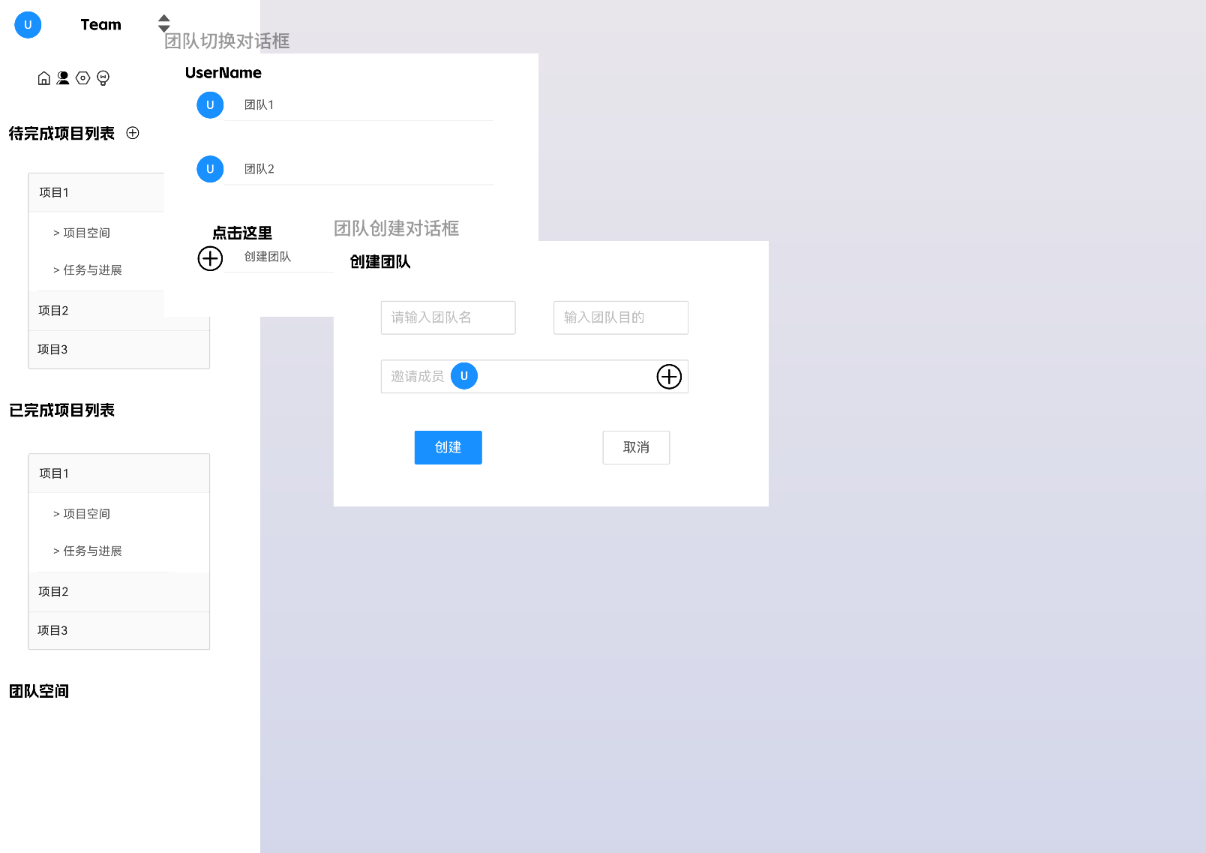
**登录/注册界面**

****

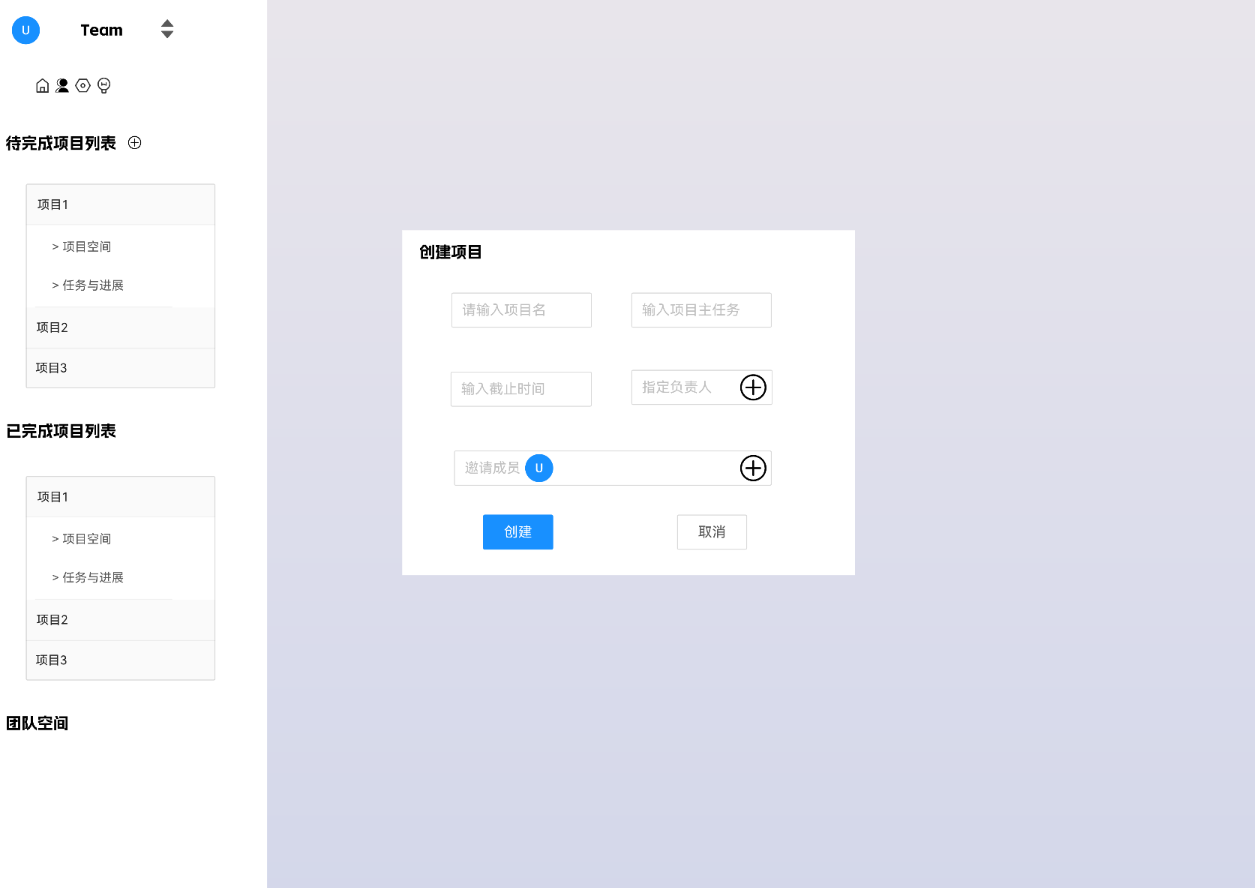
**侧边功能栏**

****

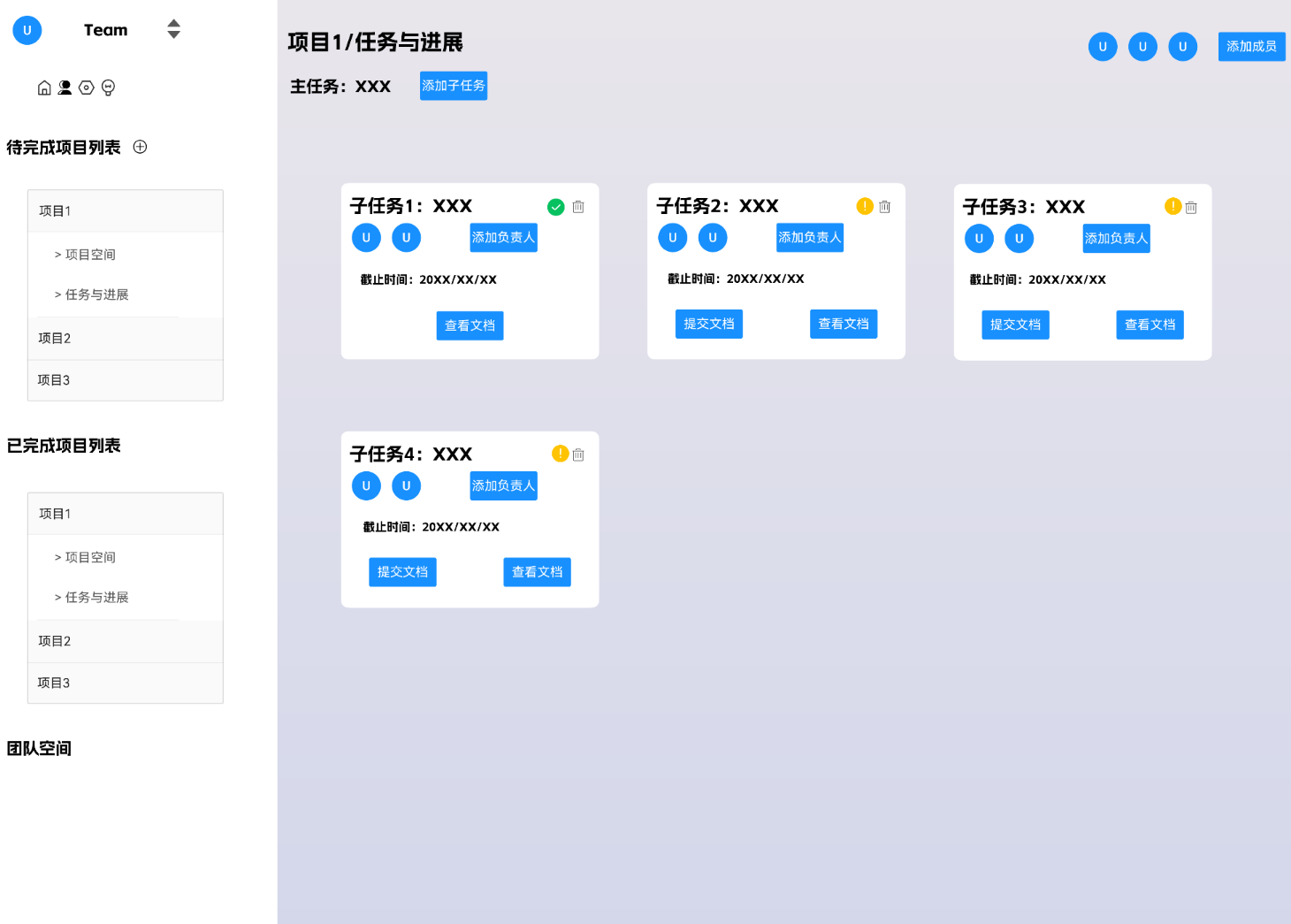
**创建团队**

****

**创建项目**

****

**项目具体界面**

****

**文档界面**

**x**

## 总结

## 附录

二级以下标题：用“（一）、1、（1）”等按顺序标示，黑体，左边起首空两格，小四号字

【论文正文】用宋体、小四号字、两端对齐。全文均是1.5倍行距。

以上说明，删除。

（1）文章标题：黑体，四号（居中，上面空三行，下面空一行）；文章副题：楷体加黑，四号；

（2）节标题：楷体，四号，粗体（居中，上下各空一行）；

（3）二级标题：黑体，小四号，（靠左，顶格，上面空一行）；

（4）正文：宋体，小四号；

（5）参考文献四个字用校四号黑体；参考文献一律用小四号楷体；

（6）行距：固定值，18磅。