



# **coconow 在线协作编程平台**

## **需求规格说明书**

作者：攻壳机动队

时间：June 12, 2022

版本：3.0.0

# 目录

第一章 简介 .....	3
1.1 背景 .....	3
1.2 目标 .....	3
1.3 范围 .....	3
1.4 术语 .....	3
1.5 参考文献 .....	4
第二章 综合描述 .....	5
2.1 产品前景 .....	5
2.2 产品功能 .....	5
2.3 用户特征 .....	5
2.4 一般性限制 .....	5
2.5 假设和依赖 .....	6
第三章 详细需求 .....	7
3.1 功能需求 .....	7
3.1.1 功能需求总体说明 .....	7
3.1.2 功能需求详细说明 .....	8
3.2 外部接口需求 .....	16
3.2.1 用户界面 .....	16
3.2.2 软件接口 .....	17
3.2.3 通讯接口 .....	17
3.3 性能需求 .....	17
3.3.1 总体性能需求 .....	17
3.3.2 数据库特性需求 .....	17
3.4 质量属性 .....	17
3.5 其它需求 .....	18

# 第一章 简介

## 1.1 背景

对于程序员来说一个功能完备且贴合实际编程需求的平台对项目的开发有着事半功倍的作用。现在的软件开发多是在 IDE（集成开发环境）上进行，不同的开发平台会侧重于不同的功能，例如 Visual Studio，它的强大功能和多功能性使其成为众多 IDE 顶级的竞争对手，但也因此有笨重、不适合初学者的缺点；Eclipse 主要是为 Java 开发人员构建的，提供了一个由插件组成的大型生态系统，轻量级且没有正式的最低系统要求，但用户界面沉闷，不适用于大型项目的特点也对其产生一定限制。

虽然这些主流的编程平台确实解决了编程环境的大部分问题，但在一些方面却并不能够尽善尽美，比如团队项目中，为了把握好代码的实现过程，经常会出现线下会议编程，或是结对编程等编程形式，而大部分 IDE 缺少的实时共享性就可以一定程度上简化完成这些需求所做的操作，衍生例如远程结对编程，多人线上协作编程的工作形式。

目前支持实时协同的 IDE（如 AWS Cloud9 等）或多或少也都有其局限性：仅支持编辑文件不支持代码检查，与某特定实例绑定使用，收费等等，因此，干净、便捷、功能相对完备的线上实时共享编码平台是有开发可行性且有必要开发的。

## 1.2 目标

解决团队开发中需要多人共同编程的情况，在这个由于各种原因无法线下合作的大环境下使线上共享编程成为一个便利的开发手段；

预期用户：敏捷开发项目工作团队，在校有合作编程相关需求的学生。

## 1.3 范围

待开发的软件名称：coconow

软件的总体功能描述：一个综合了编辑代码、代码编译、聊天室等多种功能且可多人实时共享代码的线上协作编程平台。

## 1.4 术语

未注册用户	该系统的未注册的人员
注册用户	所有注册该系统的人员
小组	由组长和组员组成的一个项目开发团队
组长	一个开发项目的发起人，拥有项目开发过程中对小组控制的最高权限
组员	参与到某个开发项目中，除组长之外的人员

## 1.5 参考文献

- [1] 清华大学软件需求规格说明 Version 1.0
- [2] cocode-在线结对编程平台 \$交付 ppt\$ .pdf

## 第二章 综合描述

### 2.1 产品前景

如今,在线应用已成为一个发展的趋势,受各方面影响,信息化系统的应用将不断深化,人们对多人在线协作编程平台的需求也在不断激增,具体体现为对办公场景的需求的多样化使得信息化系统需适应不同场景的需要。本产品能够提供线上多人协作完成项目、数据同步以及文件编辑等功能,同时让使用者在任何地点都可参与,对办公场景要求不高,具有轻量化的特点,这种便利程度让本产品的发展前景一路向好。

### 2.2 产品功能

本产品的功能包括:

- (1) 账号信息管理: 注册账号、登录账号、修改密码等信息
- (2) 项目管理: 可以创建项目, 并能将所需文件导入项目中, 同时也可以将已完成的项目文件导出
- (3) 实时展示: 满足多人 (1~5 人) 实时编辑同一个项目内的文件, 并实时显示编辑者输入的位置, 该文件可以是代码文件或文本文件 (至少完成一种代码文件)
- (4) 内容纠错与提示: 能根据选择代码文件的类型 (如.js 文件, .c 文件, 至少完成一种) 进行语法高亮、检查
- (5) 代码编译: 能对(4)中选择的代码类型的文件进行编译运行并反馈结果和错误信息
- (6) 用户信息交流: 同一项目中的成员可以通过聊天窗口互相发送文本信息交流
- (7) 项目用户邀请: 创建项目的人为该项目的组长, 组长通过搜索 ID 的方式添加成员, 被邀请者自动加入

### 2.3 用户特征

用户主要分为以下 5 类, 每种用户的特征如下:

- (1) 未注册用户: 浏览网站、注册账号
  - (2) 组员: 进行代码开发、文件管理、发送接收即时消息
  - (3) 组长: 在组员的基础上, 可以创建/删除整个项目, 管理项目人员
- 其中(2)(3)合称为注册用户

### 2.4 一般性限制

- (1) 操作系统: 需支持 Windows7 以后的版本
- (2) 界面语言: 中文/英文
- (3) 使用可维护的主流编程语言开发, 例如: C++ python java 等
- (4) 使用 MySQL 等主流数据库
- (5) 一个项目内满足 1-5 人在线编码无崩溃

## 2.5 假设和依赖

本项目是否能够成功实施，主要取决于以下条件：

- (1) 了解现有的软件开发流程
- (2) 相对稳定的项目团队，保证项目的进度和质量
- (3) 掌握能够使用本产品的技术，保证系统的性能以及项目的成功
- (4) 选择能够满足系统运行条件的硬件环境和软件环境

## 第三章 详细需求

### 3.1 功能需求

#### 3.1.1 功能需求总体说明

主要功能	详细功能	功能描述	用户
账号管理	用户注册	未注册用户填写用户名、密码等信息。验证上述信息的有效性后向用户提供的联系方式发送验证码。用户输入正确的验证码后成为注册用户。	未注册用户
	用户登录	注册用户输入账号密码，在数据库中检查账号密码的存在性和正确性。若账号密码正确且账号未被冻结，则完成登录；反之，视为登录失败。	注册用户
	忘记密码	注册用户输入账号和注册时填写的联系方式，验证账号有效性。如果账号有效，向联系方式发送重置密码链接。用户访问该链接后，输入两次一致且有效的新密码。正确输入后写回数据库完成密码修改。	注册用户
	修改密码	注册用户在登录状态下输入原密码和两次一致且有效的新密码。正确输入后写回数据库完成密码修改。	注册用户
	退出登录	完成登出，无法使用登录状态的所有功能	注册用户
代码开发	修改代码	任意组内成员修改代码，即时同步其他人的修改内容，同时进行代码检查、语法高亮和代码提示。	注册用户
	运行代码	立刻保存所有人的修改内容，禁止所有人继续修改，启动编译器/解释器，运行项目	注册用户
信息交流	接发即时文本消息	对于文本消息，以一个较短的时间间隔，响应组员发送消息的动作，同步所有人的消息，并展示在页面上。	注册用户
文件管理	导入文件	组员选择文档文件、上传，服务器比对 MD5 值后，如果新文件不存在于服务器，则保存该文件，记录上传者信息并公开给相关成员。	注册用户
	修改文件	组员在可进行修改的文档的编辑窗口进行编辑。当组员选择保存后，服务器写回后重新计算 MD5 值。文档不允许并发写，但允许并发读。	注册用户
	删除文件	组员选择可进行删除的文档后，在服务器端删除文件和相关 MD5 记录。处理完成后给出反馈。	注册用户
	查看文件	组员或组长选择可查看的文档，在窗口进行查看。	注册用户
	下载文件	组员或组长选择一到多个文件下载到本地	注册用户
成员管理	添加组员	项目组长向服务器提出邀请组员的请求，提交被邀请者的账号，数据库验证账号之后将被邀请用户加入项目，给予组员权限（对项目内所有文件的操作权限）。	组长
	移除组员	项目管理者将项目内组员移出项目，收回其组员权限。	组长

项目管理	项目创建	注册用户申请创建项目，提交项目相关信息，包括项目名、项目所需资源、基础项目说明等。服务器通过项目创建请求验证之后，维护项目信息将用户设置为该项目组长，给予组长权限。	组长
	项目删除	组长删除整个项目。	组长

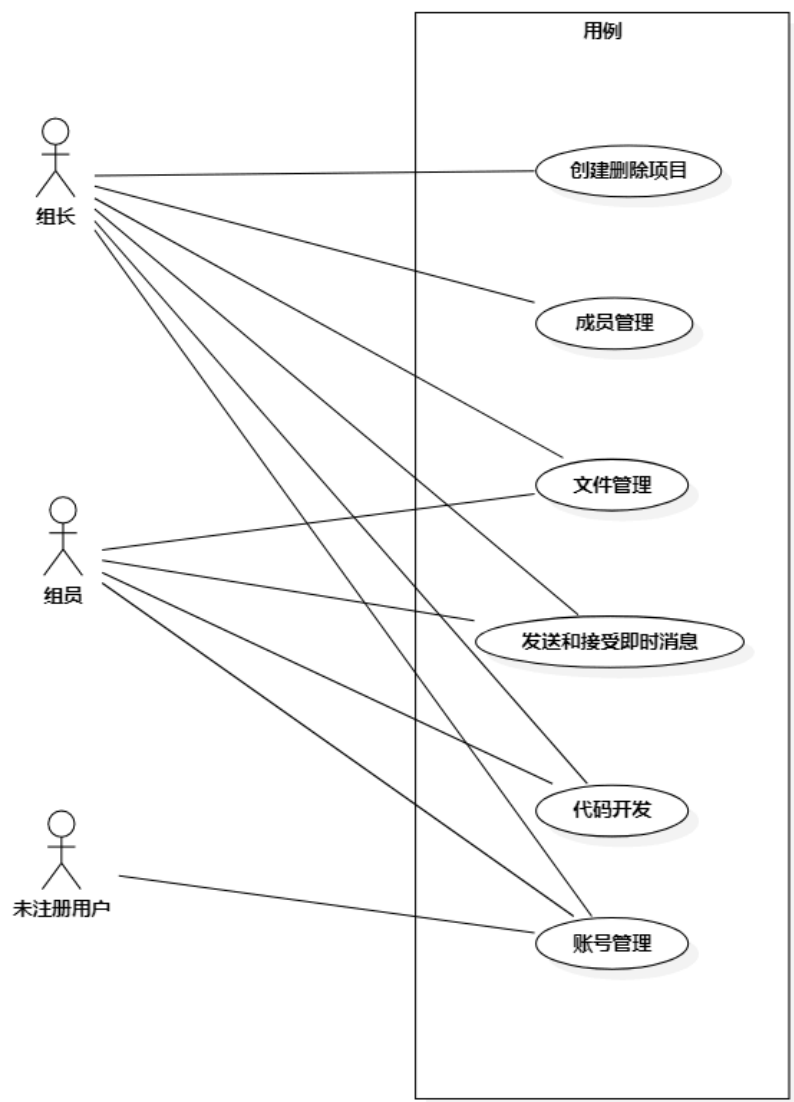


图 1 总体需求用例图

3.1.2 功能需求详细说明

账号管理部分用例和用例图

用例 1	用户注册
前提条件	非注册用户访问注册页面
基本路径	1. 非注册用户填写用户名、密码和联系方式



	2. 检查用户名是否重复。如果用户名已存在，要求重新输入 3. 检查密码是否合法，如果密码非法，要求重新输入 4. 等待用户请求发送验证码 5. 向联系方式发送验证码 6. 用户输入验证码 7. 检查验证码正确性。如果错误，回到 4 8. 数据库写回用户信息
后置条件	非注册用户成为注册用户

用例 2	用户登录
前提条件	用户访问登录页面
基本路径	1. 用户填写用户名、密码 2. 检查用户名是否存在且与密码匹配，是否被冻结 3. 通过上述验证后，跳转到相关页面，登录成功；否则回到 1 4. 若连续 5 次由于输入正确的账号和错误的密码而导致的登录失败被视为非法操作，采取封禁 IP、冻结账号等操作，24 小时后解禁
后置条件	用户成功登陆

用例 3	忘记密码
前提条件	用户尝试登录但账号密码不匹配
基本路径	1. 用户提供用户名和注册时填写的联系方式 2. 检查账号是否有效 3. 对于有效账号，向相应联系方式发送重置密码链接 4. 用户访问链接 5. 用户输入两次一致且有效的密码 6. 若正确输入，则写回数据库，重置成功；否则，回到 4
后置条件	用户重置密码成功

用例 4	修改密码
前提条件	用户已登录且访问修改密码页面
基本路径	1. 用户输入原密码和两次新密码 2. 验证原密码的有效性 3. 若原密码有效且两次新密码一致且有效，则写回数据库，修改成功；否则，回到 1
后置条件	用户修改密码成功

用例 5	退出登录
前提条件	用户已登录
基本路径	1. 用户选择“退出登录”功能 2. 修改服务器内的用户状态，完成退出登录操作
后置条件	用户成功退出，无法使用登录状态的所有功能

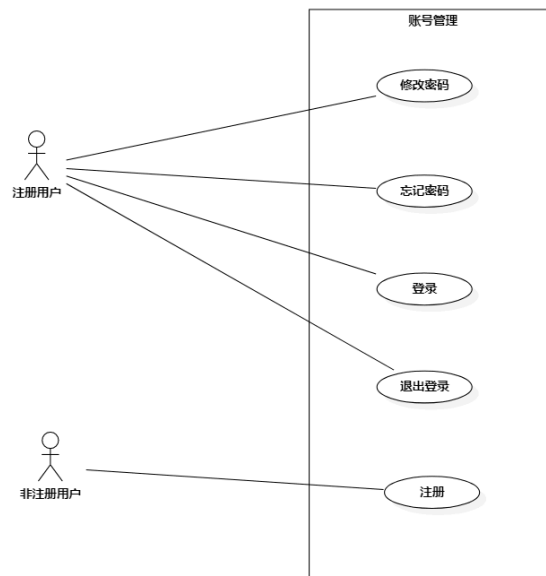


图 1 账号管理用例图

## 代码开发部分用例

用例 1	修改代码
前提条件	组员已经登录，组员在项目中
基本路径	1. 组员可以看到项目的目录结构和文件内容 2. 组员可以调整目录结构和修改代码文件 3. 组员的修改操作会反映在项目视图中 4. 项目视图会随着任何一位组员的修改而即时变化 5. 组员点击“保存项目”会将当前的项目视图保存到项目存储中
后置条件	保存后的项目可以运行或调试

用例 2	运行代码
前提条件	组员已经登录，组员在项目中
基本路径	1. 组员点击“运行代码”则用例开始 2. 立刻保存所有人的修改内容 3. 启动编译器/解释器，运行项目 4. 运行代码时禁止所有人修改代码
后置条件	运行/调式的结果用于指导后续的修改

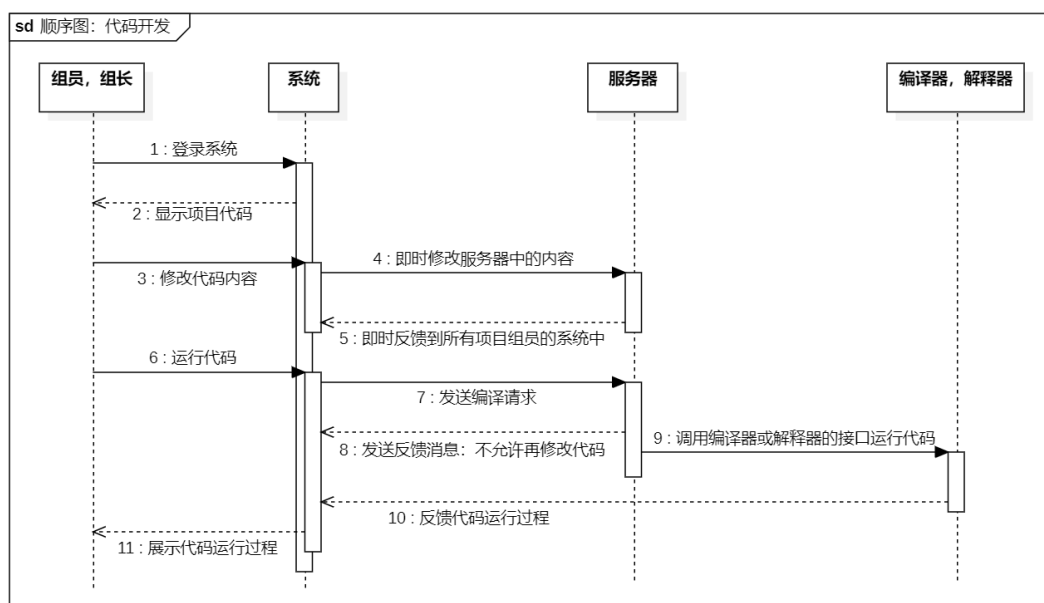


图 3 代码开发顺序图

## 聊天部分用例和用例图

用例 1	查看聊天信息
前提条件	登录用户且已经被加入聊天室
基本路径	1. 项目组长已经开设了聊天室 2. 用户打开特定聊天室时检索一定时间内的历史信息。 3. 用户通过前端页面代码页中的嵌入式窗口部件查看加载出来的消息。
后置条件	无

用例 2	查看成员资料
前提条件	登录用户且已经被加入聊天室
基本路径	1. 用户在聊天窗口内点击用户头像 2. 该用户的个人资料从服务器中检索到 3. 资料页面浮动展示在 UI 中
后置条件	无

用例 3	过滤消息
前提条件	聊天室在线且聊天记录非空
基本路径	1. 用户在聊天嵌入窗口中选择过滤选项 2. 系统从已经加载的消息中按条件过滤 3. 用户从变更的窗口中选择性查看特定消息，过滤结果为空则对话提示并返回原界面
后置条件	无

用例 4	查看聊天历史记录
------	----------

前提条件	聊天室已被创建
基本路径	<ol style="list-style-type: none"> <li>1. 用户发送任意消息</li> <li>2. 系统将消息存入数据库记录中</li> <li>3. 用户在聊天窗口中选择查看历史记录</li> <li>4. 系统自动设置分页检索数据库中未被清理且可以公开的消息</li> <li>5. 用户在聊天窗口看到历史消息</li> </ol>
后置条件	无

用例 5	发送文本消息
前提条件	登录用户且已经被加入聊天室
基本路径	<ol style="list-style-type: none"> <li>1. 用户发送文本消息</li> <li>2. 系统构造消息对象并存档在数据库记录中</li> <li>3. 系统将消息发送到目标成员前端</li> <li>4. 目标成员看到他人发送的消息</li> </ol>
后置条件	无

用例 6	创建聊天室
前提条件	登录用户小组成员
基本路径	<ol style="list-style-type: none"> <li>1. 注册用户在菜单栏中选择申请聊天室，并选择要加入聊天室内的成员</li> <li>2. 请求转发给网站管理方</li> <li>3. 管理方接受或拒绝请求，后续可考虑软件自动判断</li> <li>4. 接受请求后，添加聊天窗口</li> </ol>
后置条件	无

用例 7	删除聊天室
前提条件	选择删除聊天室的用户为组长或聊天室内成员
基本路径	<ol style="list-style-type: none"> <li>1. 注册用户选择删除聊天室</li> <li>2. 注册用户确认删除聊天室</li> <li>3. 请求转发给网站管理方</li> <li>3. 管理方接受或拒绝请求，后续可考虑软件自动判断</li> <li>4. 接受请求后，删除聊天室</li> </ol>
后置条件	无

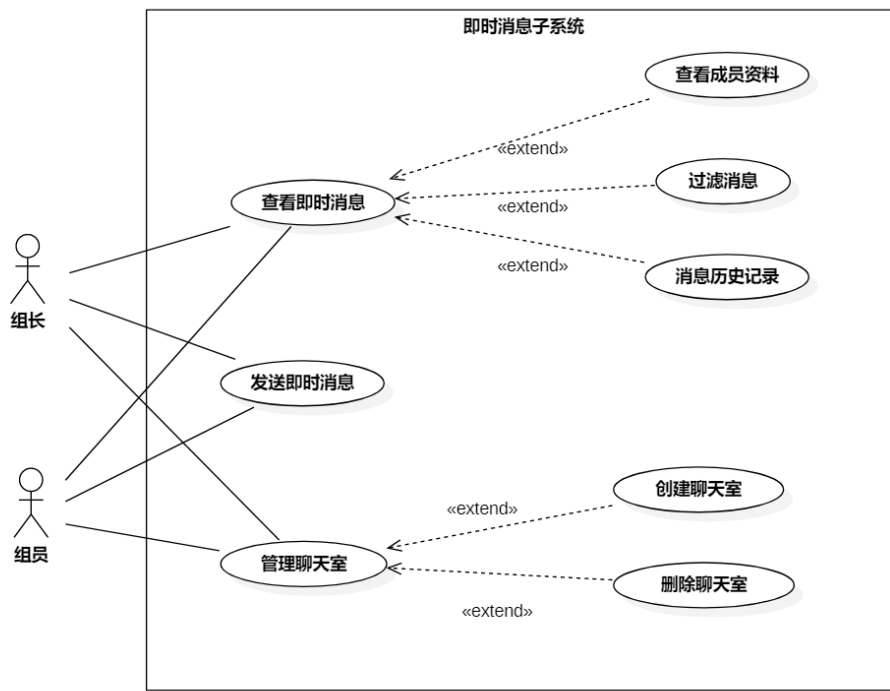


图 4 聊天部分用例图

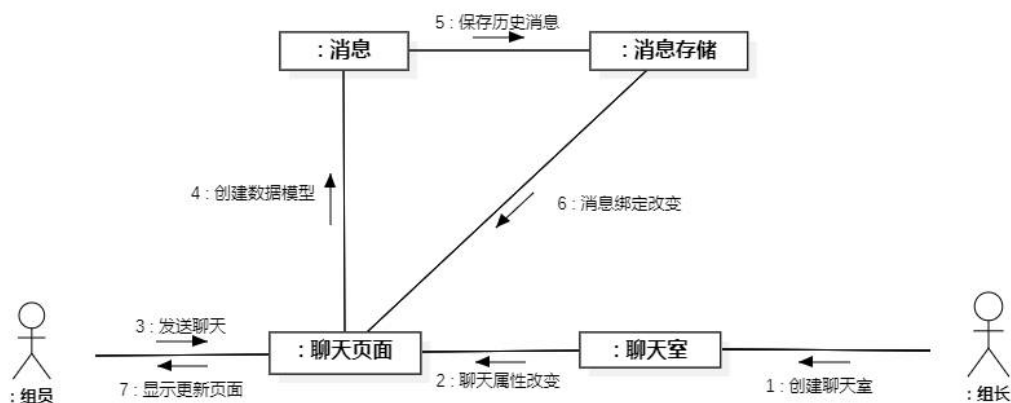


图 5 文字聊天协作图

## 文件管理部分用例和用例图

用例 1	导入文件
前提条件	用户进入要上传文件的目录位置
基本路径	<ol style="list-style-type: none"> <li>1. 用户选择上传文件</li> <li>2. 系统弹出本地文件选择框</li> <li>3. 用户选择要上传的文件</li> <li>4. 用户设置文件权限（默认为可修改、可删除、可查看）</li> <li>5. 用户确定上传</li> <li>6. 系统验证上文件合法性</li> </ol>

	7. 系统反馈上传操作结果
后置条件	服务器保存文件信息

用例 2	修改文件
前提条件	组员选择要修改的项目文件
基本路径	<ol style="list-style-type: none"> <li>1. 用户提出修改请求</li> <li>2. 系统验证当前文件状态 <ol style="list-style-type: none"> <li>a. 当前文件不支持修改</li> </ol> </li> <li>3. 系统反馈无法修改</li> <li>b. 当前文件支持修改 <ol style="list-style-type: none"> <li>3. 用户进入文件编辑界面，修改文件内容和权限设定</li> <li>4. 用户保存文件</li> <li>5. 系统反馈保存结果和问题原因 <ol style="list-style-type: none"> <li>a. 保存失败，用户检查网络 and 文件合法性重新保存</li> <li>b. 保存成功，退出编辑界面</li> </ol> </li> </ol> </li> </ol>
后置条件	服务器保存文件信息

用例 3	删除文件
前提条件	组员选择要删除的项目文件
基本路径	<ol style="list-style-type: none"> <li>1. 用户提出删除请求</li> <li>2. 系统验证当前文件状态 <ol style="list-style-type: none"> <li>a. 当前文件不支持删除</li> </ol> </li> <li>3. 系统反馈无法删除</li> <li>b. 当前文件支持删除 <ol style="list-style-type: none"> <li>3. 系统反馈删除结果和问题原因 <ol style="list-style-type: none"> <li>a. 删除失败，用户检查网络重新删除</li> <li>b. 删除成功</li> </ol> </li> </ol> </li> </ol>
后置条件	服务器维护文件目录信息

用例 4	查看文件
前提条件	组员选择要查看的项目文件
基本路径	<ol style="list-style-type: none"> <li>1. 用户提出查看请求</li> <li>2. 系统验证当前文件状态 <ol style="list-style-type: none"> <li>a. 当前文件不支持查看</li> </ol> </li> <li>3. 系统反馈无法查看</li> <li>b. 当前文件支持查看 <ol style="list-style-type: none"> <li>3. 显示文件内容</li> </ol> </li> </ol>
后置条件	无

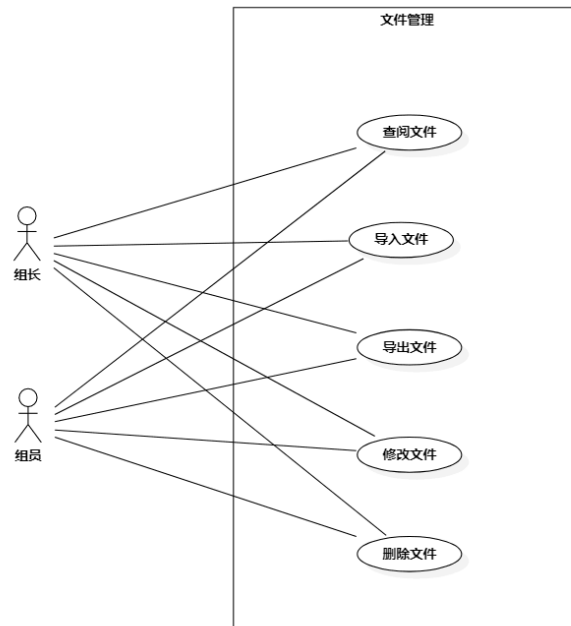


图 6 文件管理用例图

#### 成员管理部分用例和用例图

用例 1	添加组员
前提条件	组长查看组员列表
基本路径	<ol style="list-style-type: none"> <li>1. 组长选择添加组员</li> <li>2. 系统弹出被邀请人信息栏</li> <li>3. 组长填写被邀请人账号</li> <li>4. 系统验证账号合法性 <ol style="list-style-type: none"> <li>a. 没有该用户 <ol style="list-style-type: none"> <li>5. 系统反馈账号有误</li> </ol> </li> <li>b. 存在该用户 <ol style="list-style-type: none"> <li>5. 系统反馈添加成功</li> <li>6. 系统分配该用户组员权限（对项目内所有文件的所有操作权限）</li> <li>7. 在组员列表中添加显示新组员</li> </ol> </li> </ol> </li> </ol>
后置条件	系统向组员发送加入项目通知

用例 2	移除组员
前提条件	组长查看组员列表
基本路径	<ol style="list-style-type: none"> <li>1. 组长选择移除组员</li> <li>2. 系统显示可被移除的组员列表</li> <li>3. 组长选择要移除的组员</li> <li>4. 系统反馈移除成功</li> <li>5. 系统回收该用户组员权限</li> <li>6. 更新组员列表显示</li> </ol>
后置条件	系统向组员发送退出项目通知

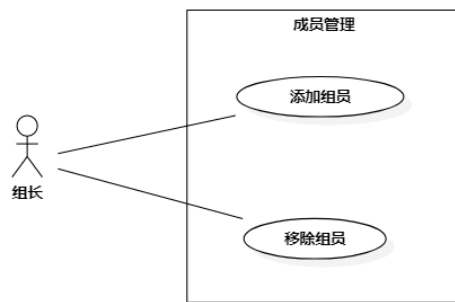


图 7 成员管理用例图

## 项目管理部分用例

用例 1	项目创建
前提条件	组长已登录
基本路径	1. 组长填写项目信息 2. 组长选择初始组员 3. 组长确认项目信息 4. 数据库保存项目信息，添加组员，初始化相关内容
后置条件	项目创建成功

用例 2	项目删除
前提条件	组长已登录
基本路径	1. 组长点击删除 2. 组长确认删除
后置条件	项目删除成功

## 3.2 外部接口需求

### 3.2.1 用户界面

(1) 使用浏览器的用户界面，遵循 MVC 设计模式，通过数据绑定与异步更新技术进行低耦合、低消耗的页面展示。

(2) 希望对常见的集成开发环境、编辑器如 Jupyter Notebook，VSCode 等进行借鉴，以更好地贴合开发者的开发习惯。

(3) 页面中主要展示代码编辑页面，此外能够展示项目文件目录，聊天窗口等。登录用户信息，交互请求等也可以收入菜单栏中。

(4) 无需适配移动端界面，但是需要确保常见比例显示效果合理。



### 3.2.2 软件接口

(1) 具有可以同常见的并选定的数据库系统交互的接口，可以存储项目文件，项目配置，成员状态，聊天记录等内容。同时数据还应保证安全性，文件不能被异常读取，账号密码等进行加密保护等。

(2) 具有上传本地文件至项目，将项目文件下载到本地的功能。

(3) 具有能够直接调用服务器载编译 / 解析器的软件接口，能够通过网页端的功能请求直接按照用户编写的脚本进行代码的编译 / 解释，并将执行结果反馈到网页端的模拟 shell 中。此外考虑到安全性问题，编译调试的代码在执行系统级操作时应该被禁止。(希望具体编译执行的实现中编译器与调试器尽可能使用已有软件包，不需要重复造轮子，但是需要完成前后端的数据对接。)

### 3.2.3 通讯接口

(1) Chrome (Webkit), Firefox (Gecko)作为主要浏览器能够被较好支持，二者没有较大的显示错误与区别。Safari, IE 11+ 等其他浏览器兼容功能使用或选择性放弃支持均可。

(2) 通信连接使用 HTTP/IP 协议，并应当尽量使用 Socket 长连接，确保协作与交流的不间断进行。

## 3.3 性能需求

### 3.3.1 总体性能需求

(1) 支持单文本文件 1MB，全项目 256 MB 级别的项目云存储。

(2) 账户系统至少应满足 3000 名注册用户的记录。

(3) 服务系统并发应至少满足 100 名用户同时编码，5 个项目同时进行调试。

(4) 对于通畅网络情况，实时编辑的网络时延不应高于 500 ms 。

(5) 服务器全局历史消息保存空间应至少有 5 GB 。

(6) 服务器应保证随时有足够的资源以响应实时的代码协作任务，合理进行服务器资源调度。

### 3.3.2 数据库特性需求

(1) 数据库系统应当至少实现二范式的设计以减少空间占用，提高处理效率。

(2) 数据库合理设计数据索引。

(3) 数据库可选实现并行优化请求以提高持久层的性能。

## 3.4 质量属性

(1) 可用性：要求系统至少能在 6:00 与次日 1:00 间提供不间断的服务。

- (2) 安全性：用户代码调试执行应有类似沙箱的机制防止对服务器造成损坏，此外还应防止 SQL 注入。
- (3) 可靠性：系统故障时应当自动保存用户未保存的所有代码文件。
- (4) 可维护性：关键的类与函数应当有文件注释，故障发生时应该至少有错误信息返回给用户以供查询。
- (5) 易用性：功能界面应当做到类似 Jupyter Notebook 的风格，功能划分简洁明了。

### 3.5 其它需求

- (1) 可选中文英文任一语言的支持。
- (2) 部署时应在 Ubuntu 至少应在 18.04 LTS 版本上顺利部署运行。
- (3) 可选用 Docker 进行环境打包。