



厦 门 理 工 学 院

XIAMEN UNIVERSITY OF TECHNOLOGY

毕 业 设 计

中文题目 基于联盟链的学术出版平台设计与实现

英文题目 **Design and Implementation of an Academic Publishing**
Platform Based on Consortium Blockchain

院 系：	计算机与信息工程学院
年级专业：	2020 级计算机科学与技术
姓 名：	廖建德
学 号：	2013206118
指导教师：	李远敏
职 称：	副教授

2024 年 5 月

毕业设计（论文）诚信声明书

本人郑重声明：在毕业设计（论文）工作中严格遵守学校有关规定，恪守学术规范；我所提交的毕业设计（论文）是本人在李远敏指导教师的指导下独立研究、撰写的成果，设计（论文）中所引用他人的文字、研究成果，均已在设计（论文）中加以说明；在本人的毕业设计（论文）中未剽窃、抄袭他人的学术观点、思想和成果，未篡改实验数据。

本设计（论文）和资料若有不实之处，本人愿承担一切相关责任。

学生签名：廖建德

2024 年 5 月 15 日

厦门理工学院学位论文著作权使用声明

本人同意厦门理工学院根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门理工学院图书馆及其数据库被查阅、借阅。本人同意厦门理工学院将学位论文汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

（）1. 经厦门理工学院教学工作委员会审查核定的保密学位论文，于 年 月 日解密，解密后适用上述授权。

（）2. 不保密，适用上述授权。

（请在以上相应括号内打“√”或填上相应内容。保密学位论文应是已经厦门理工学院教学工作委员会审定过的学位论文，未经厦门理工学院教学工作委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

年 月 日

摘 要

随着信息技术的不断发展，传统的学术出版方式已经面临着诸多挑战。学术失范造成了学术领域严重的道德问题，其主要表现为学风浮躁、抄袭剽窃等问题，这些问题严重影响了学术研究的质量和可信度。

首先对整个学术论文出版流程进行了分析，其中功能需求包括了通过区块链浏览器查询联盟链信息、学术组织的创建、投稿审评、出版交易、版权认证保护、智能合约调用等。基于前述需求分析，设计了基于联盟链的学术出版平台的总体架构，搭建了基于 Hyperledger Besu 的联盟链基础网络，详细说明了系统数据库的表和字段并列举出相关字段含义；分别对用户鉴权管理、学术组织、投稿审评、出版交易、版权认证等模块进行与区块链智能合约技术相融合，对处理流程和相关类进行了详细设计；此外，介绍了各个子模块之间的交互关系和数据流动，本文给出了部分核心模块的接口设计，并以文字的方式阐述了相关接口调用信息。继而，在随后的章节阐述了整个系统的部署环境和最终的运行结果，展示了最终的用户交互界面。

本系统使用了区块链联盟链技术，solidity 语言以太坊智能合约开发技术、ERC20 发行 token 和 ERC721 铸造 NFT 协议、Hyperledger Besu 联盟链协议，MetaMask 区块链钱包、IPFS 分布式存储和 MySQL 数据库存储技术、docker 容器技术、web 前端 vue 框架和后端 go 语言开发等技术设计并实现了一个能够为学术研究提供更加公正、透明、不可篡改和可追溯的学术出版平台，推动学术发展进步。

关键词：学术出版；联盟链；智能合约；IPFS；NFT

ABSTRACT

In today's digital age, traditional academic publishing methods are facing numerous challenges due to the continuous development of information technology. Academic misconduct has led to serious ethical issues in the academic field, mainly manifested as academic misconduct, plagiarism, and other problems, which significantly affect the quality and credibility of academic research.

Firstly, an analysis of the entire academic paper publishing process is conducted, with functional requirements including querying consortium chain information through blockchain browsers, creating academic organizations, manuscript submission and review, publishing transactions, copyright authentication protection, and smart contract invocation. Based on the aforementioned requirements analysis, the overall architecture of an academic publishing platform based on consortium blockchain is designed. A consortium blockchain foundation network based on Hyperledger Besu is established, detailing the tables and fields of the system database and listing the meanings of relevant fields. Authentication management, academic organizations, manuscript submission and review, publishing transactions, copyright authentication, and other modules are integrated with blockchain smart contract technology. Detailed designs of processing flows and related classes are provided. In addition, the interaction relationships and data flow between various sub-modules are introduced, and partial core module interface designs are presented in text format along with relevant interface invocation information.

Furthermore, the subsequent chapters elaborate on the deployment environment of the entire system and the final operational results, demonstrating the final user interaction interface.

This system utilizes blockchain consortium technology, Solidity language for Ethereum smart contract development, ERC20 token issuance and ERC721 NFT protocol, Hyperledger Besu consortium chain protocol, MetaMask blockchain wallet, IPFS distributed storage, MySQL database storage technology, Docker container technology, web front-end Vue framework, and back-end Go language development, etc. These technologies are designed and implemented to provide a more fair, transparent, tamper-proof, and traceable academic publishing platform, promoting the advancement of academic development.

Key Words: Academic Publishing; Consortium Chain; Smart Contract; IPFS; NFT

目 录

第 1 章	引言	1
1.1	研究背景	1
1.2	国内外研究现状	1
1.3	主要研究内容	2
第 2 章	相关技术简介	3
2.1	区块链	3
2.2	Web 开发	4
2.3	总结	5
第 3 章	需求分析	6
3.1	需求概述	6
3.2	可行性分析	6
3.3	系统功能需求分析	7
3.4	总结	12
第 4 章	系统设计	13
4.1	系统概要设计	13
4.2	系统详细设计	14
4.3	数据库设计	17
4.4	总结	23
第 5 章	系统实现	24
5.1	Hyperledger 节点搭建联盟链	24
5.2	MetaMask 钱包连接联盟链	26
5.3	IPFS 系统	27
5.4	区块链浏览器	28
5.5	智能合约接口功能实现	28
5.6	前后端功能实现	31
5.7	总结	35
第 6 章	系统测试	36
6.1	用户基础功能模块测试	36
6.2	组织模块测试	37
6.3	投稿模块测试	39
6.4	审稿模块测试	40
6.5	出版交易模块测试	41
6.6	总结	43
第 7 章	结论与展望	44
7.1	结论	44
7.2	进一步工作的方向	44
参考文献		45
致 谢		46

第1章 引言

1.1 研究背景

近年来，区块链技术在全球范围内快速发展，并在数字金融、物联网、智能制造、供应链管理等多个领域展现了广阔的应用前景。2018年11月14日，在中央全面深化改革委员会第五次会议上，审议通过了《关于深化改革培育世界一流科技期刊的意见》^[1]。同时《关于深化改革培育世界一流科技期刊的意见》指出，科技期刊传承人类文明，荟萃科学发现，引领科技发展，直接体现国家科技竞争力和文化软实力。要以习近平新时代中国特色社会主义思想为指导，全面贯彻党的十九大和十九届二中、三中全会精神，全面把握创新发展规律、科技管理规律和人才成长规律，立足国情、面向世界，提升质量、超越一流，走出一条中国特色科技期刊发展道路^[2]。目前条件下中国期刊编辑界面临着极大的挑战，只有聚合人才，提高组织效率，更新管理模式，加快各学术领域的资源整合，深化产业链合作，提高技术维护的能力，同时重视运营销售方式，重视数字化出版及交易，才能在未来知识出版、期刊发展中，获得建设世界一流期刊的契机^[3]。

1.2 国内外研究现状

近年来，国家宏观管理部门以及各种相关学术研究部门等都制订和实施了很多针对性和目的性较强的学术评价政策和准则，这些政策和准则对引导学术评价、推动学术评价的良好发展起到了深远的正面影响。根据学术评价的不同评价方式对其进行分类，可以分成两类，定性评价和定量评价两种，其中定性评价方式以同行评议为代表，定量评价方式以科学计量为核心。两种方式都有其存在的质疑和不足之处^[4]。现存的采编系统从根本上存在一些问题：一是现行系统是中心化存储结构，作者的稿件信息和审稿人的评议信息存储在同一中心化服务器上，一旦服务器崩溃，所有信息均无法找回，基于新型分布式区块链存储的去中心化特性^[5]，能有效解决此问题。二是上述信息有被篡改的可能性。三是评议流转过程中，稿件和评议信息缺乏保护机制^[6]。

另外，现行的同行评议制度，也有一些漏洞和缺陷。例如：审稿过程中审稿人与作者由于身份差造成无法平等的交流，审稿人因无需具名而草率对待，审稿人权力寻租等^[7]。由于制度的不透明，地位的不对等，信息的不对称，往往会导致各个角色之间产生信任危机^[8]。为了维护学术出版诚信，促进学术公平、公正、透明，有必要寻求一种全新的技术，从系统上解决以上的问题。此外，期刊的影响力、声望和地位涉及多个主客观因素，需要应用一系列指标进行多方面评估^[9]。

目前，国内外有关区块链在学术期刊应用的研究集中在实践或探索可能维度方面：一

是对现有应用案例的梳理总结^[10]；二是提出新的出版模式，包括智能合约合作出版模式^[11]、代币驱动的出版模型、去中心化的新型治理模式^[12]等；三是站在全局的角度进行应用展望；四是关于区块链革新学术出版的细分环节的研究，包括利用通证激励用户参与并由学术社群决定是否发表^[13]，把评议拓展到出版流程上游，从而使评议更自动化、公开和可查证，用智能合约保障评议信息的写入和上链^[14]，形成作者—学术共同体—读者的交互评价模式，将区块链技术应用用于期刊版权保护，利用区块链技术建设学术出版信任机制^[15]等。

区块链从读取权限的维度分类，可分为公有链和联盟链。公有链通常应用于虚拟货币、互联网金融等应用场景。例如比特币，NFT 等属于公有链。联盟链一般是某一行业或领域中一些机构共同维护的区块链，联盟链只允许联盟内的机构进行数据读写和记录交易。联盟链适用于特定的行业和领域，例如版权机构联盟等。联盟链与公有链相比，具有交易确认时间短、单位时间交易数量大等特点，对安全和性能的要求也比公有链高。这一特性使联盟链可以实现企业级的应用。

区块链技术作为重构生产关系的新技术，促进了整体的生态进化。区块链在学术出版界的应用，国际上已有先例。对比之下，我国的应用发展还比较单一，暂时只在诚信建设、学术版权、评价方式、网络出版信任危机治理^[15]等方面有所应用。而国际上已有部分国家将区块链技术应用在期刊管理和出版流程中。

1.3 主要研究内容

1. 区块链为学术出版提供去中心化和可溯源

区块链技术将推动新的学术传播机制的形成和发展，为新时代学术期刊评价提供新视角。区块链没有中央系统，数据以块集的形式存在，每个块集包含一组事务，涉及一个时间戳和前一个块集的哈希值。通过 IPFS 去中心化网络存储，保证数据不丢失。对投稿者的论文通过加密算法分布式储存于区块链，为学术期刊的评价提供去中心化、不可篡改的溯源依据。

2. 联盟链的共识协议

部署多个节点共同监督。在基于区块链的学术出版体系中，每次做出链上写操作，必须由一个可靠的共识协议来做出决策，让分散在链上的各区块节点形成一致意见，并对达成共识的协议进行管束。

3. 智能合约

智能合约是一套以数字形式定义、合约参与方可以执行的承诺协议。一个合约就是经计算机执行、记录和处理的在区块链里的程序，目的是保护计算机的数据，避免数据丢失^[16]。利用合约完成无需信任度保障的交易，所有操作内容在写在合约里，不可更改，达成条件自动执行，提高信任度和效率。

第2章 相关技术简介

2.1 区块链

1. 区块链技术原理概述

区块链技术是一种革命性的分布式账本技术，它采用了一系列创新性的方法来解决传统中心化数据库系统所面临的一些问题。首先，区块链是一种去中心化的技术，意味着数据不再存储在单一的中心服务器上，而是分布在网络的多个节点上。这种去中心化的结构消除了单点故障的风险，使得数据更加安全和可靠。其次，区块链的数据结构是以链式连接的数据块的形式存在的，每个数据块包含了多个交易的信息。每当有新的交易发生时，这些交易将被打包成一个新的数据块，并被加入到链的末端。这种链式结构确保了交易的顺序性和不可逆性，一旦数据被写入到区块链中，就无法被篡改或删除，从而保证了数据的完整性和可信度。另外，区块链技术还借助于密码学哈希函数和加密技术来确保数据的安全性。每个数据块都包含了前一个数据块的哈希值，这种链接机制使得数据的修改会导致后续数据块的哈希值发生变化，从而保证了数据的不可篡改性。同时，区块链还使用了分布式共识机制来确保网络的安全性和稳定性，例如工作量证明（Proof of Work）和权益证明（Proof of Stake）等。总之，区块链技术通过其去中心化、不可篡改、可追溯和安全的特性，为各种应用场景提供了一种全新的解决方案。它不仅在金融领域有着广泛的应用，还可以用于供应链管理、智能合约、数字资产管理等各个领域，为现代社会的数字化转型提供了强大的支持和保障。

2. 联盟链和 Hyperledger Besu

联盟链是区块链的一种形式，与公有链相比，联盟链由已知的参与方共同管理，提供了更高的可控性和隐私性。Hyperledger Besu 是一种基于以太坊协议的企业级联盟链平台，它是 Linux 基金会旗下的 Hyperledger 项目的一部分。Hyperledger Besu 提供了可扩展性和灵活性，适用于构建企业级联盟链网络。作为企业级区块链解决方案的一部分，Hyperledger Besu 可以满足企业对于联盟链平台的高度可控、高度安全的需求，并提供了强大的工具和功能，如智能合约支持、权限控制、隐私保护等，使得企业能够更加灵活地搭建和管理自己的区块链网络，应用于不同的业务场景中，从而实现数字化转型和业务优化。

3. 智能合约的编写与部署

智能合约是一种以代码形式存在于区块链上的自动执行合约规定操作的计算机程序。它们可以在区块链网络上运行，无需第三方介入即可执行合同条款。Remix 是一种基于网页的以太坊智能合约集成开发环境，它为开发人员提供了一个方便的平台，可以编写、编译、部署和调试智能合约。通过 Remix，开发人员可以在浏览器中直接编写智能合约代码，并在实时编辑器中查看代码的效果。Remix 还提供了一个交互式调试器，可以让开发人员

在测试智能合约时进行单步调试，以确保合约的正确性和可靠性。此外，**Remix** 还集成了 **Solidity** 编译器，可以帮助开发人员将智能合约代码编译成可在以太坊虚拟机上运行的字节码。**Remix** 为智能合约的开发提供了一个便捷而强大的工具，使开发人员能够更轻松地创建、测试和部署智能合约，从而推动了区块链应用的发展和普及。

4. IPFS

IPFS（**Inter Planetary File System**）是一种点对点的分布式文件存储系统，通过内容寻址的方式为文件分配唯一的哈希值作为地址，确保了数据的唯一性和安全性。与传统的 **Web** 服务器不同，**IPFS** 将文件分布在网络的各个节点上，使得用户可以从最近的节点获取文件，提高了数据的访问速度和可用性。同时，**IPFS** 实现了版本控制，使得用户可以轻松地追踪和管理文件的历史记录，确保了数据的完整性和可追溯性。综合而言，**IPFS** 为构建更加去中心化、安全、高效的 **Web** 提供了一种新的解决方案，推动了数据存储和传输方式的革新。

5. MetaMask 钱包

MetaMask 钱包是一款基于浏览器插件的以太坊钱包，它允许用户在网页浏览器中直接与以太坊区块链交互。作为一种数字钱包，**MetaMask** 提供了一种安全、方便的方式来管理以太币（**Ether**）和其他以太坊资产，以及与去中心化应用（**DApps**）进行交互。用户可以通过 **MetaMask** 创建和管理以太坊账户，并进行加密货币的发送、接收和存储。此外，**MetaMask** 还提供了一个内置的 **DApp** 浏览器，用户可以通过它访问各种以太坊上的去中心化应用，如加密货币交易所、游戏和金融服务等。**MetaMask** 钱包的安全性和易用性使其成为了许多以太坊用户的首选钱包之一，为用户提供了便捷的数字资产管理和 **DApp** 使用体验。

2.2 Web 开发

1. Vue2

Vue.js 作为一种备受欢迎的 **JavaScript** 前端框架，专注于构建交互式的 **Web** 界面和单页面应用程序（**SPA**）。其简洁、灵活和高效的特性使开发人员能够快速构建具有丰富交互功能的用户界面。通过巧妙优化和重构代码，并结合现有的 **UI** 组件库和工具，开发人员在使用 **Vue.js** 时能够降低代码的相似度，从而减少查重率。此外，**Vue.js** 还提供了一套响应式的数据绑定机制，能够自动追踪数据变化，并实时更新 **DOM**。这种特性也有助于降低代码的重复性，进一步减少查重率。

2. Docker

Docker 是一种容器化技术，旨在简化应用程序的开发、部署和管理过程。它允许开发人员将应用程序及其依赖项打包成一个称为容器的可移植单元。这些容器可以在任何支持 **Docker** 的环境中运行，无需重新配置或重新安装应用程序的依赖项。**Docker** 提供了一

种轻量级、灵活和高效的解决方案，可以帮助开发人员快速构建、测试和部署应用程序。通过使用 **Docker**，开发人员可以更加方便地创建一致性的开发和生产环境，提高了应用程序的可移植性、可扩展性和安全性。

3. Go 语言

是一种由 **Google** 开发的开源编程语言。它具有简洁、高效、可靠的特点，旨在提高开发人员的生产力和代码的可维护性。**Go** 语言支持并发编程，通过轻量级的协程（**goroutines**）和通道（**channels**）机制，使得编写并发程序变得简单和高效。此外，**Go** 语言还具有垃圾回收、静态类型、编译型语言等特性，使得它在处理大规模系统和高性能应用程序方面表现优异。**Go** 语言拥有丰富的标准库和活跃的社区支持，被广泛应用于网络服务、分布式系统、云计算、容器化等领域。其简单易学的语法和强大的性能使得 **Go** 语言成为了许多开发人员的首选语言之一。

2.3 总结

将这些技术有机地结合起来，以构建一个创新的平台。首先，利用区块链技术和 **Hyperledger Besu** 搭建一个安全、高效的联盟链网络，确保平台数据的透明性和不可篡改性。然后，利用智能合约和 **Remix** 开发智能合约，用于管理用户账户、实现数据交换和执行业务逻辑。同时，利用 **IPFS** 作为分布式存储系统，存储和检索用户上传的学术作品和相关数据，确保数据的安全性和可用性。为了提供良好的用户体验，采用 **MetaMask** 钱包作为平台的支付和身份认证工具，使用户可以方便地进行交易和使用平台服务。通过 **vue2** 和 **go** 语言编写传统 **web** 交互页面逻辑。最后，我会利用 **Docker** 容器化技术，部署平台的各个组件，提供可靠、高效、安全的运行环境。综合运用这些先进技术，我将打造一个公正、透明、高效的学术出版平台，推动学术领域的发展和进步。

第3章 需求分析

本章将介绍与本研究相关的技术背景，包括区块链的基本概念和工作原理、Web 开发技术的应用现状以及相关技术在学术出版平台中的具体应用。这些技术的综合应用为基于联盟链的学术出版平台的设计与实现提供了理论基础和技术支持。

3.1 需求概述

基于联盟链的学术出版平台旨在构建一个透明、安全、高效的学术作品发布和管理系统，以解决传统学术出版方式存在的问题，提高学术作品的质量和可信度。主要功能模块包括：用户基础功能，创建管理委员会、期刊和会议，投稿审核出版，代币 token 激励，NFT 版权证书生成与交易。

本系统旨在构建一个基于联盟链技术的学术出版平台，通过智能合约、IPFS 分布式存储和 Hyperledger Besu 联盟链等技术，实现学术作品的安全存储、版权认证和交易。用户通过 Web 浏览器客户端前端界面上传学术作品并与智能合约交互，使用 MetaMask 钱包界面进行数字签名认证和支付。用户可以通过区块链浏览器界面验证交易记录和区块信息，确保交易的透明度和可追溯性。需求分析图如图 3-1 所示：

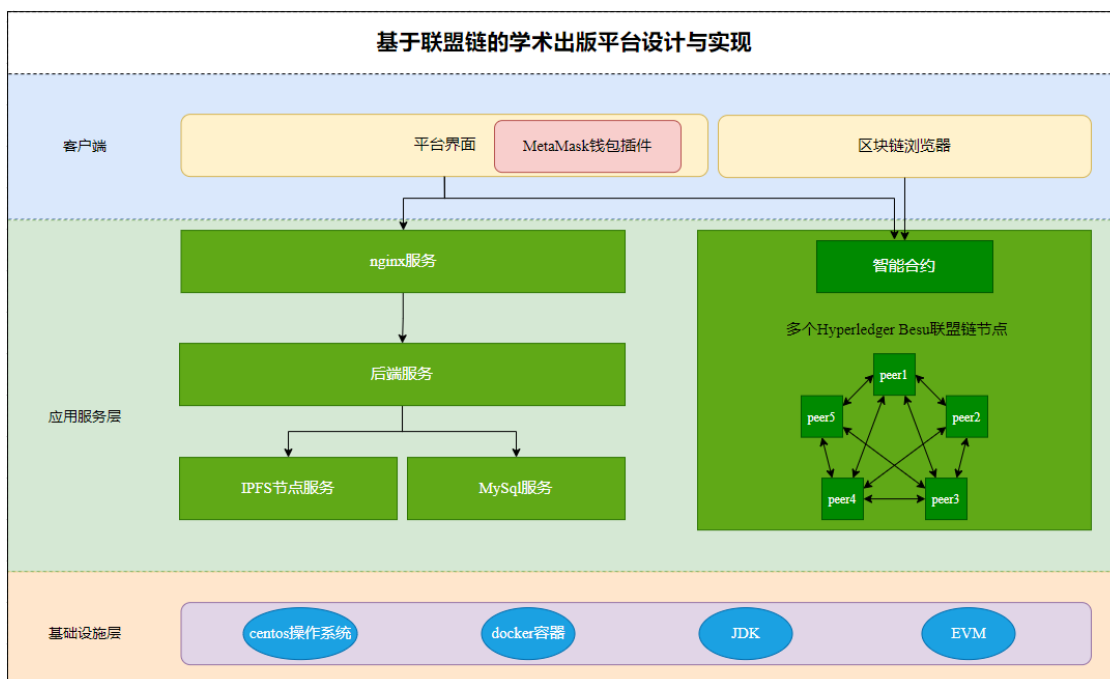


图 3-1 需求分析图

3.2 可行性分析

所选择的技术方案在解决学术出版平台面临的问题上具有显著的潜力和优势。联盟链

技术提供了高度可控和安全的环境，确保数据的隐私和权限控制；智能合约的使用使得合约规定的操作自动执行，提高了平台的自动化和可靠性，并且增强了交易的透明度和可追溯性；IPFS 分布式存储为平台提供了分布式、去中心化的存储方式，保障了数据的安全性和可靠性，同时提升了数据的访问速度和可用性。结合传统的 Web 前端和后端技术，如 Vue、Go 语言和 MySQL 数据库，能够为学术出版平台提供稳定、高效的运行环境。综合而言，这些技术方案的选择为学术出版平台的设计和实现提供了坚实的技术基础，有望解决传统学术出版领域所面临的诸多挑战，为学术交流和研究提供更加公正、透明和高效率的平台。

3.3 系统功能需求分析

在一个基于联盟链的学术出版平台中，区块链技术可以被广泛应用，以确保系统的透明性、可追溯性和安全性。平台会有用户鉴权管理来维护平台的信息安全，选择在哪个委员会下的会议或期刊下进行学术作品的投稿也可以通过智能合约来实现。当用户投稿至对应组织下时，组织应对齐进行审评工作，审评内容通过智能合约存储至联盟链。一旦论文投稿者的投稿通过审评，投稿者选择出版，智能合约可以自动将投稿进行出版发布，生成 NFT 证书，同时记录出版的时间和相关信息，确保出版过程的公正和合法性。这样一来，基于联盟链的学术出版平台将会更加高效、透明和安全。

1. 用户鉴权管理

一个平台必须是由广大用户的参与，用户可通过注册登录进入平台页面体验平台功能，用户应提供信息，并进行连接 MetaMask 钱包账户地址绑定验证，通过注册账户体验平台功能。维护平台秩序离不开用户的分级管理，用户管理是至关重要的一环，它涉及到对用户身份、权限和行为的有效管理。平台创建管理员可以查看已注册用户信息以及权限，对用户进行分角色管理。为此平台需提供如下功能：

用户注册登录：提供用户注册页面，要求填写用户名、密码、邮箱等信息，并进行连接 MetaMask 钱包账户地址绑定验证是否一致。已注册用户可以使用用户名和密码登录系统，系统验证密码准确性。

用户管理：管理员可以查看已注册用户列表，包括其角色和权限，进行用户管理操作，如授权用户可创建组织，管理组织权限等。角色包含：管理员、主席、成员、普通用户。用例图如图 3-2 所示：

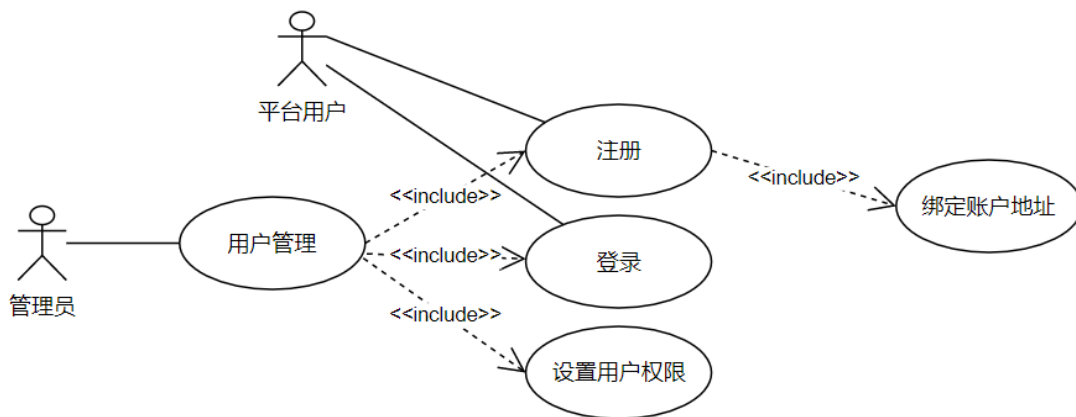


图 3-2 用户管理用例图

2. 组织管理

委员会、会议和期刊作为学术出版的重要组成部分，对于学术界的发展和学术交流起着至关重要的作用。作为学术界的评审和发布机构，承担着筛选和传播学术成果的重要责任。通过严格的审稿流程和专业的编辑团队，它们确保了学术作品的质量和可信度，为学术界提供了可靠的知识来源，它们的决策和发言往往能够引领学术潮流和影响学术发展方向。为此平台需提供如下功能：

创建管理学术委员会：管理委员会各项信息：名称、描述信息以及成员结构。

创建管理会议和期刊：创建会议或期刊时选择已创建的委员会，在委员会下可以创建学术会议和期刊。

Issues 规定可投稿时间：在会议或期刊内可以添加增加 Issue，设定投稿日期。用户即可在主页面进入对应会议或期刊的详情页面投稿信息。用例图如图 3-3 所示。

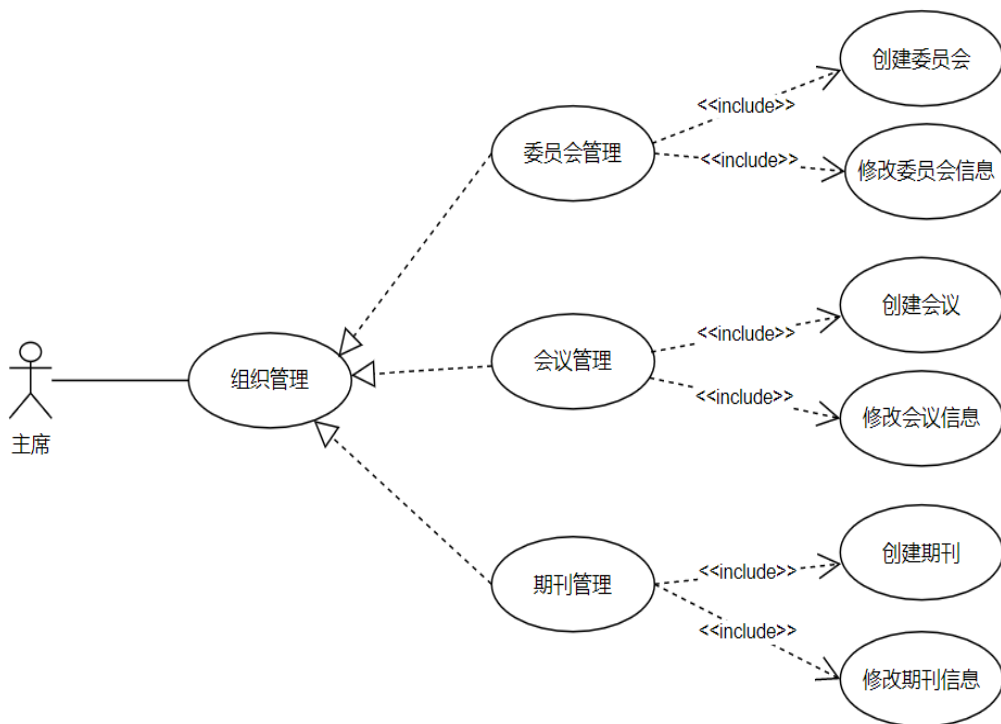


图 3-3 组织管理用例图

3. 投稿

用户使用平台的主要目的就是在平台上进行投稿出版，并且版权可以得到有效的保护。

投稿：选择时间范围内可投稿的期刊或会议投稿，作者可以选择将投稿提交到特定的学术期刊或会议，论文上传时，上传后通过调用智能合约的存储 hash 接口，使用 MetaMask 钱包确认签名将论文的 hash 值保存至联盟链上，填写标题、摘要、作者、关键词、论文所属学科分类、通讯作者等信息，选择论文类型，提交投稿，等待审核；若还未开始评审，可对投稿自由修改。用例图如图 3-4 所示：

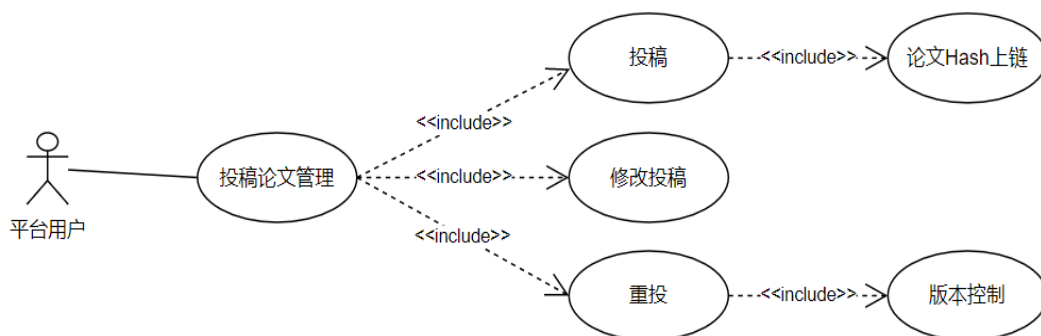


图 3-4 投稿论文管理用例图

4. 审评

当用户投稿至对应组织下时，对应组织的主席可分配审稿人审稿，主席将在其他审稿人评审完成后进行最终评审，如不通过需要修改或重新投稿，原版本投稿将保留。

分配审稿人：投稿至对应组织后，自动分配给对应组织主席进行审评，主席可分配组织内部的成员对投稿进行审评。

审评：审评人可以对投稿进行不同等级的评级，以及给出审评意见，审评内容将被通过调用智能合约接口，使用 MetaMask 钱包确认签名保存至联盟链；主席拥有最终决定权，根据审评人员审评内容做出决定，对投稿进行最终审评。审评通过，论文文件将存储与 IPFS 去中心化分布式文件存储系统中，一定时间后分散至各个节点，防止论文的丢失，确保论文的安全存储。

版本控制：投稿人可以在已审核投稿列表查看投稿，查看详情可以看到所有审评人给出的评审意见和建议，如果要求修改，则需对此次投稿进行修改，这将是一个新的投稿版本，在新的投稿中可以看见旧的投稿版本列表，也可查看详情，修改完成的投稿会再次出现在对应主席的待审核列表，等待进行第二轮审评。用例图如图 3-5 所示：

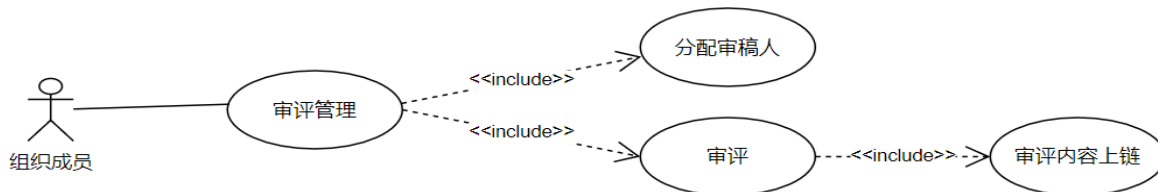


图 3-5 审评管理用例图

5. 出版交易

一旦论文投稿者的投稿通过审评，投稿者可将论文进行出版发布，生成 NFT 证书，存储至 IPFS 系统，同时在联盟链上记录出版的时间和相关信息，确保出版过程的安全公正和合法性。这样一来，这些信息都记录在联盟链上，可通过区块链浏览器查询相关信息。基于联盟链的学术出版平台将会更加高效、透明和安全。为此平台需提供如下功能：

NFT 证书出版：投稿人查询已审核通过后的投稿详情，选择出版论文，平台将自动生成版权证书，证书内容包括：作者姓名、论文标题、投稿时 hash 存储至联盟链上的交易地址、该交易所在的区块链地址、出版日期等信息，确保作品的唯一性和版权归属。投稿人可以根据自主对学术作品的质量和重要性评估，制定合适的出版定价以及论文下载定价。调用智能合约接口生成 NFT 证书，用户在 MetaMask 钱包要求账户对此次交易进行签名认证生成 NFT，等待联盟链确认交易完成后表示 NFT 证书生成成功，论文出版成功。

查询支付：平台其他用户可在主页面查找到该论文，如果需要下载查看，通过 MetaMask 钱包进行签名认证向作者账户地址转移 Token，等待联盟链确认交易完成后即可下载查看论文，下次再次查看无需再次转账购买。

交易市场：论文出版成功后，在后台可查询到用户所拥有的 NFT 证书，可以修改论文查看 Token 价格已经购买 NFT 证书版权的 Token 价格；如将 NFT 证书上架版权平台交易市场，上架交易市场后，所有用户可在平台主页交易市场查看论文 NFT 售卖信息，用户如果有充足的 Token 余额可进行购买版权证书，通过 MetaMask 钱包对 NFT 合约进行授权可花费的 Token 以及将 NFT 转给购买者的两次交易进行签名认证，等待联盟链确认交易完成后，NFT 将发生转移，购买者在后台可以查看到购买到的 NFT，此时该 NFT 证书对应的论文版权为此购买者，其他用户下载查询论文付费时，转移的 Token 将转向此购买者，即版权拥有者。用例图如图 3-6 所示：

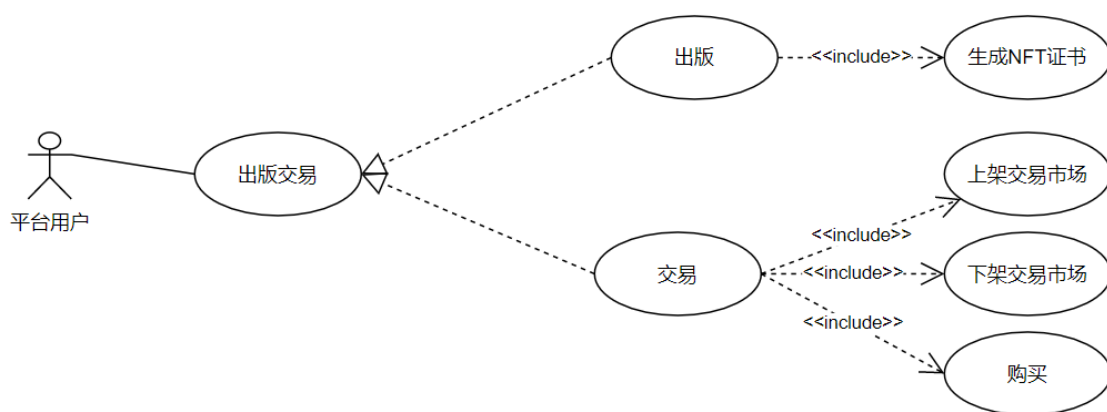


图 3-6 出版交易管理用例图

6. 区块链浏览器

区块链浏览器是用于查看和浏览区块链上的数据和交易信息的工具。区块链浏览器是区块链应用的重要组成部分，所有的基于区块链的应用都离不开区块链浏览器的信息查询，区块链上的所有数据都是公开透明的，区块链浏览器可以向用户提供查询在区块链上

发生的所有交易信息，包括部署的合约代码和地址，账户的转账、铸造 NFT、存储论文 Hash 等交易信息。用户通过复制区块链交易地址，可以查询对应地址所在的区块链位置，发生的时间以及调用者的账户地址，这些数据都是公开透明可查且不可篡改的。为此区块链浏览器需提供如下功能

区块和交易浏览：提供对最新区块和交易的实时浏览功能，包括区块的高度、时间戳、大小等信息，以及交易的发送者、接收者、金额等信息。用户可以通过区块链浏览器浏览区块链上的历史交易记录，以及每个区块中包含的交易信息。

查询：提供搜索功能，用户可以通过输入账户地址或交易哈希来快速定位到相关信息。

区块链状态监控：提供区块链网络的状态监控功能，包括节点数量、网络哈希率、区块间隔等关键指标。可视化展示区块链的运行状态，用户可以了解网络的健康状况和稳定性。

区块链数据统计和分析：提供区块链数据的统计和分析功能，包括交易数量、交易额、活跃地址数等指标的图表展示。用户可以通过数据统计和分析了解区块链的发展趋势和用户活动情况。用例图如图 3-7 所示：

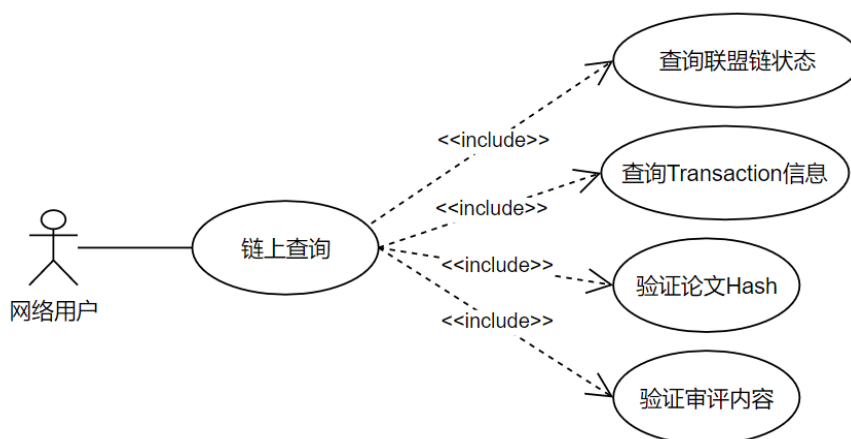


图 3-7 链上查询用例图

7. 智能合约

智能合约是区块链上的自动化合约，它们通过预先编程的规则和条件执行特定的功能。对于用户需要生成 NFT 证书资产，以及交易功能，通过编写 ERC-20、ERC-721 和交易市场合约，实现下面是功能需求：

ERC-20 合约：

代币发行和转账：提供代币的发行和转账功能，包括创建新的代币、转移代币所有权等。确保代币的发行总量和每次转账的数量符合合约规定。

代币余额查询：允许用户查询特定地址的代币余额，以及总发行量等信息。

转账事件通知：当代币转账发生时，能够触发相应的事件通知，以便其他合约或应用程序能够及时响应。**授权和转账代理：**允许用户授权其他地址代表其转账代币，以及撤销该授权。

合规性和安全性：确保合约的实现符合 ERC-20 标准，以保证与其他 ERC-20 兼容应用的互操作性。

ERC-721 合约：

非同质化资产发行和转移：提供非同质化资产（版权证书）的发行和转移功能，每个资产都有唯一的标识符。确保资产的唯一性和不可替代性，防止重复发行或双重所有权。

资产所有权查询：允许用户查询特定资产的所有权和相关信息，包括当前所有者、历史所有者等。

转移事件通知：当资产转移发生时，能够触发相应的事件通知，以便其他合约或应用程序能够及时响应。

元数据支持：支持元数据，包括资产的名称、描述、图像等信息，以增强用户体验和资产的可视化展示。

合规性和安全性：确保合约的实现符合 ERC-721 标准，以保证与其他 ERC-721 兼容应用的互操作性。

交易市场合约：

基本交易功能：允许用户将自己的 NFT 资产进行定价发布订单或取消自己发布的订单，用户以指定 Token 价格购买 NFT 资产。

事件功能：包括成交事件、新订单事件、价格更改事件和订单取消事件。

3.4 总结

本章对相关技术的介绍和分析，为后续章节的需求分析和系统设计奠定了坚实的基础。接下来将详细阐述平台的具体需求，并对系统的功能和性能进行全面分析，以确保平台能够有效满足学术出版的需求，并在实际应用中展现出其优越性和可靠性。

第4章 系统设计

本章将介绍基于联盟链的学术出版平台的系统设计。系统设计是实现一个功能完善、可靠和安全的平台的关键步骤。在本章中，将详细讨论系统的整体架构、功能模块设计、数据流程和安全机制。通过对各个组件的深入分析与设计，确保平台能够满足用户需求，并在学术出版过程中提供高效的管理和操作体验。本章内容将为后续的系统实现提供明确的实现蓝图和指导原则。

4.1 系统概要设计

在本平台中，划分了多个功能模块，包括用户登录注册、管理员管理用户、创建委员会、期刊和会议、学术作品投稿、审稿、投稿版本查询、出版定价和生成 NFT 证书以及 NFT 交易等模块。每个模块都有其独特的功能和作用，共同构成了一个完整的学术出版生态系统。通过本系统，用户可以方便地进行学术作品的投稿、审稿和出版，确保数据的安全性和可追溯性。区块链技术将被广泛应用于用户身份验证、数据存储和智能合约执行等方面，以保障平台的透明度、安全性和可信度。同时也能够通过 NFT 技术确保作品的版权归属和交易安全。管理员可以高效地管理用户和学术活动，保障平台的正常运行和安全性。系统需求的模块框架结构如图 4-1 所示：

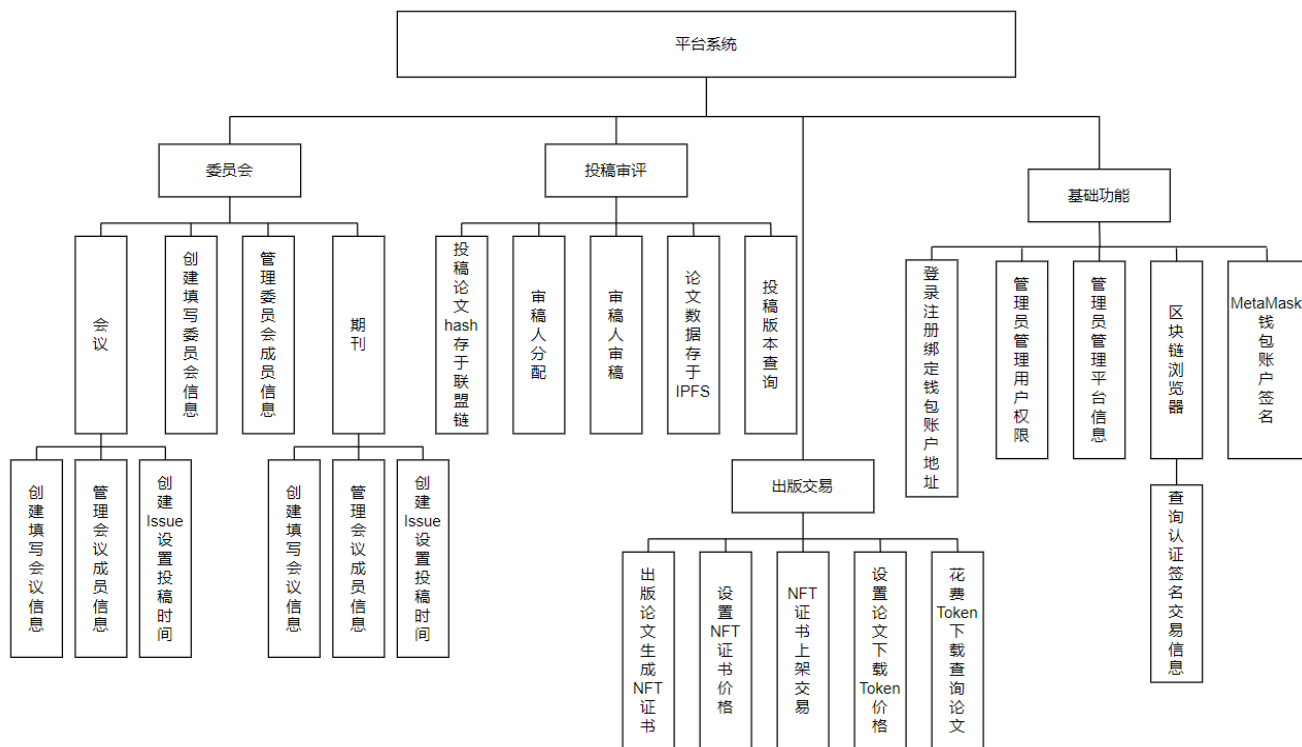


图 4-1 系统架构图

4.2 系统详细设计

系统各个模块相互协作，共同构建了一个完整的基于联盟链的学术出版平台。每个模块都有自己的功能和责任，通过相互协作实现系统的整体功能。

1. 用户鉴权管理模块设计

在系统设计中，在注册时，用户需要填写其账户地址，并将 MetaMask 钱包连接的账户地址复制填入相应字段，同时补充其他相关信息。系统验证用户输入的信息是否完整且符合格式规范，并确保用户名在平台上尚未被使用。

管理员拥有平台所有功能，可以在管理员选择查看用户列表、修改用户角色权限：角色分为管理员，主席，成员，普通用户。还可管理所有委员会、期刊和会议信息，对不符合规定的团队或投稿论文进行解散删除。

2. 组织模块设计

拥有创建权限的用户可以选择创建或管理委员会、期刊或会议。输入相应的信息，如名称、描述、成员等信息。如图 4-2 所示：

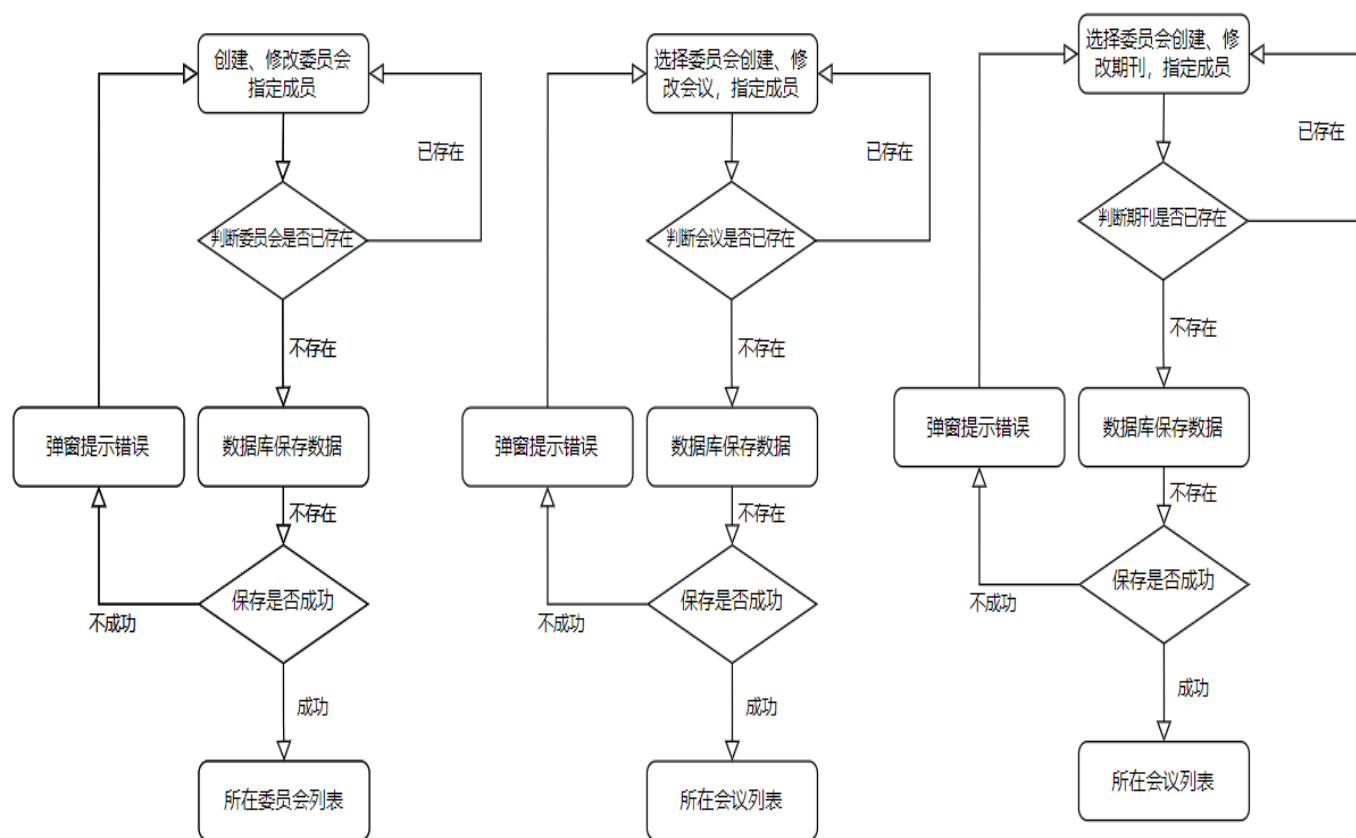


图 4-2 委员会期刊会议流程图

3. 投稿模块设计

系统包括文件上传模块，用户通过该模块可以方便地选择要上传的文件，还包含了自动计算文件哈希值的功能，并与智能合约 API 进行交互，以将文件哈希值安全地存储到合约联盟链中。在用户上传文件的过程中，系统与 MetaMask 紧密集成，以确保对链上

操作的签名确认。一旦用户确认签名，系统会等待联盟链将交易打包并执行。在成功执行后，系统将接收到联盟链返回的相应交易地址哈希值和联盟链区块地址哈希值。系统会整合用户填写的信息和联盟链返回的信息，并将其有效地提交至后端。后端系统则设计了相应的接口，以接收并处理前端提交的信息，并将文件数据和相关信息安全地存储到服务器中。对于修改版本，进行版本号递增的逻辑确保版本正确稳定的更迭。如图 4-4 所示：

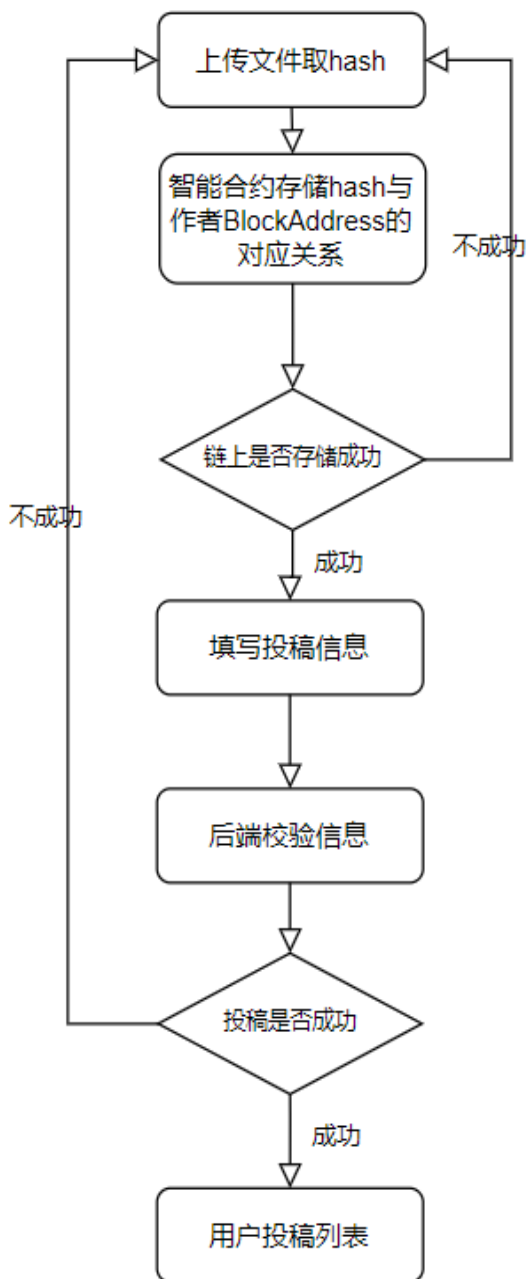


图 4-4 投稿流程图

4. 审稿模块设计

主席即组织者通过系统选择并分配审稿人员，审稿人员收到任务后，在系统中进行审稿。审稿完成后，系统自动将审稿结果整合并调用智能合约 API 将其安全地存储到合约联盟链中。在此过程中，MetaMask 钱包前面验证，以确保审稿人员对链上操作进行签名

确认。系统根据审稿结果自动判断投稿的通过与否。若投稿通过，系统将文件上传至 IPFS 分布式文件系统，以确保文件的永久存储，并将投稿标记为待出版状态。若投稿未通过，则系统通知作者需重新修改投稿或重新提交。如图 4-5 所示：

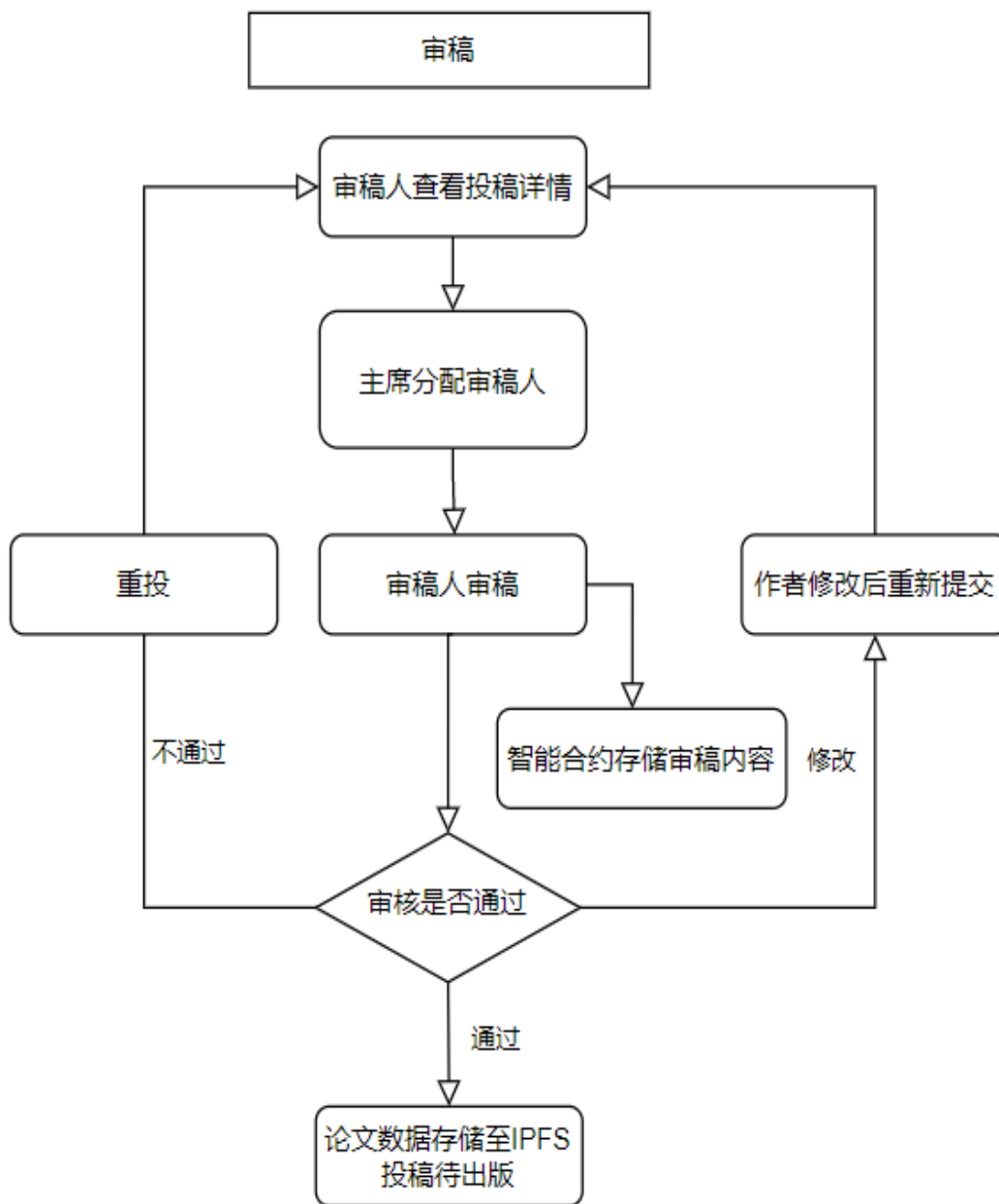


图 4-5 审稿流程图

5. 出版交易模块设计

投稿审核通过，作者即可查看待出版的投稿。作者选择出版后，根据投稿信息自动生成证书图片。作者确认后，设置 NFT 证书价格并查看所需 Token。再次确认出版后，将证书图片保存至 IPFS 分布式文件存储系统。此过程中，MetaMask 钱包进行签名认证，同意签名认证后返回响应数据，其中 tokenId 为 NFT 的唯一标识，代表其独一无二性。

成功出版生成 NFT 证书后，作者可将证书上架到平台 NFT 交易市场进行售卖。若不想再卖，也可选择从交易市场下架 NFT。下架 NFT 时，系统会在联盟链上留下相应

痕迹。上架成功后，在平台交易系统可看到 NFT 售卖列表，用户只需有足够 Token 余额即可进行购买。如图 4-6 所示：

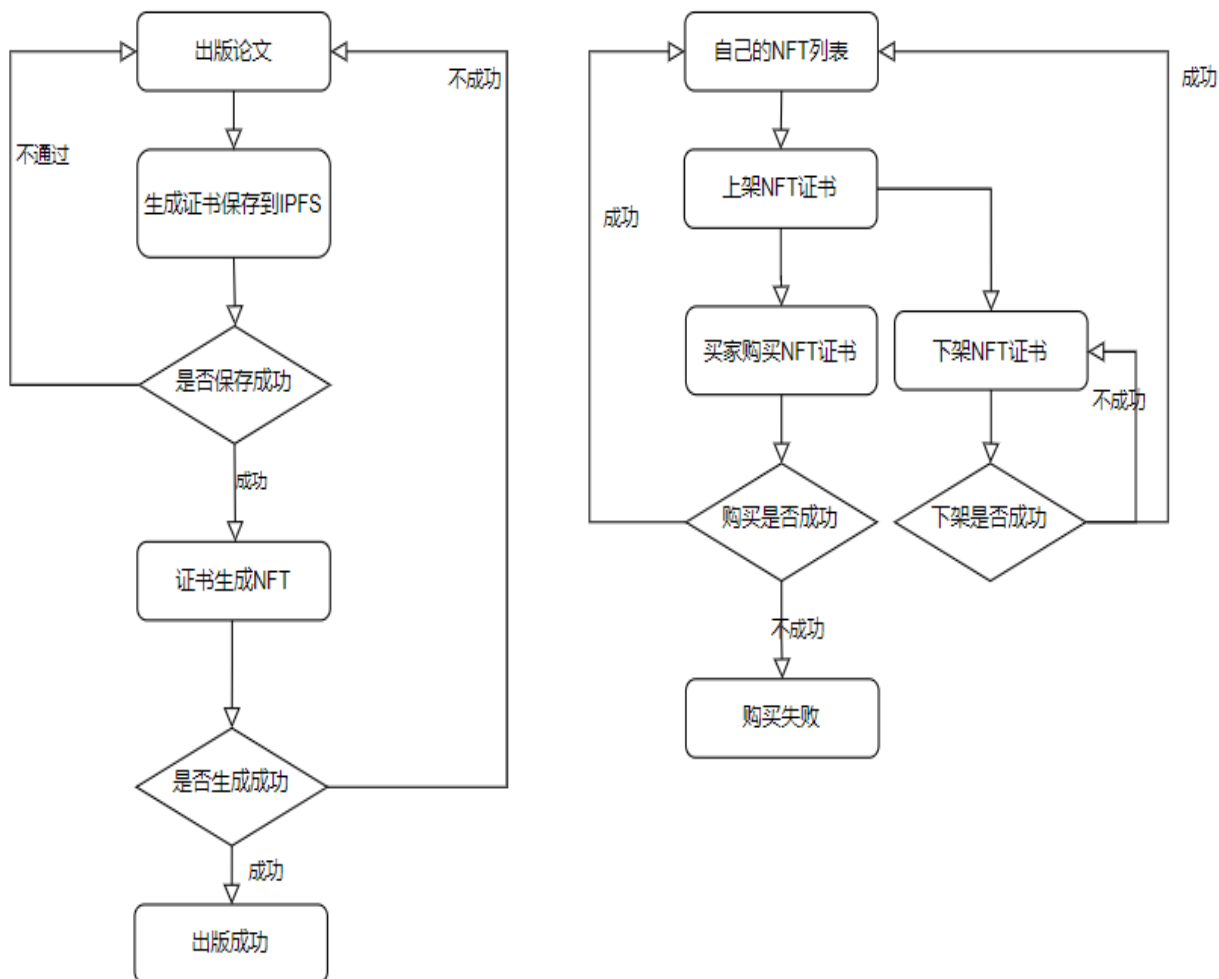


图 4-6 生成与交易 NFT 流程图

6. 小结

本节详细介绍了系统的主要功能模块的详细设计，描述了功能的实现流程，展示了各个模块的流程图，设计了用户绑定 MetaMask 钱包账户，投稿审评时的与联盟链的交互过程：投稿论文取 hash 值调用智能合约接口存储上链、审核通过将论文数据存储在 IPFS 系统、出版生成 NFT 证书、上架版权证书 NFT 交易、查阅论文的 token 转账，说明了账户签名调用智能合约的时机。

4.3 数据库设计

1. 逻辑结构设计

本系统包含 17 张表，根据系统功能，主要包括用户实体，投稿论文实体，委员会、会议、期刊实体，角色权限实体，拥有权限的用户可以创建多个委员会、会议、期刊组织，在不同组织担任职位，所有用户可以在任意组织下进行投稿。主要数据库的逻辑结构 ER

图如图 4-7 所示：

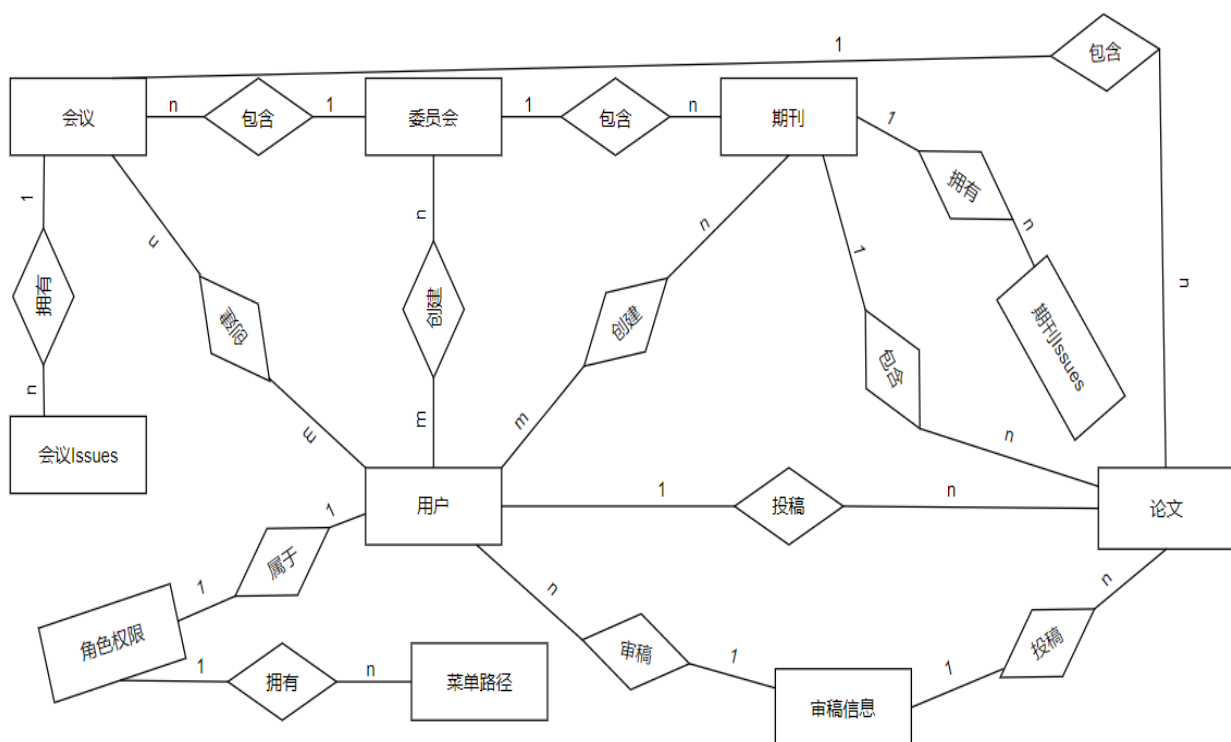


图 4-7 数据库 ER 图

2. 实体设计

本系统包含的功能主要有用户管理，委员会、期刊、会议管理、投稿、审稿、出版等，为此将确定系统中涉及的所有业务对象和数据，并将它们抽象成数据库中的实体。每个实体将有其相应的属性，这些属性描述了实体的特征和属性。下面将列出系统所有的实体以及它们的属性：根据本平台的功能需求以及用户所需的必要信息，用户包含主要信息有：用户 id，用户名，区块链账户地址等，具体用户实体如图 4-8 所示：



图 4-8 用户实体

系统围绕着投稿论文为核心,实现完整的平台功能,所有论文包含的信息内容比较多,重要的信息包括:论文 id、版本号、论文存储至 IPFS 后的生成 cid、论文 hash、存论文 hash 时的区块链地址、NFT 证书的基本信息以及存储位置信息,上述信息保证论文投稿的内容详细以及安全保证,实体如图 4-9 所示:

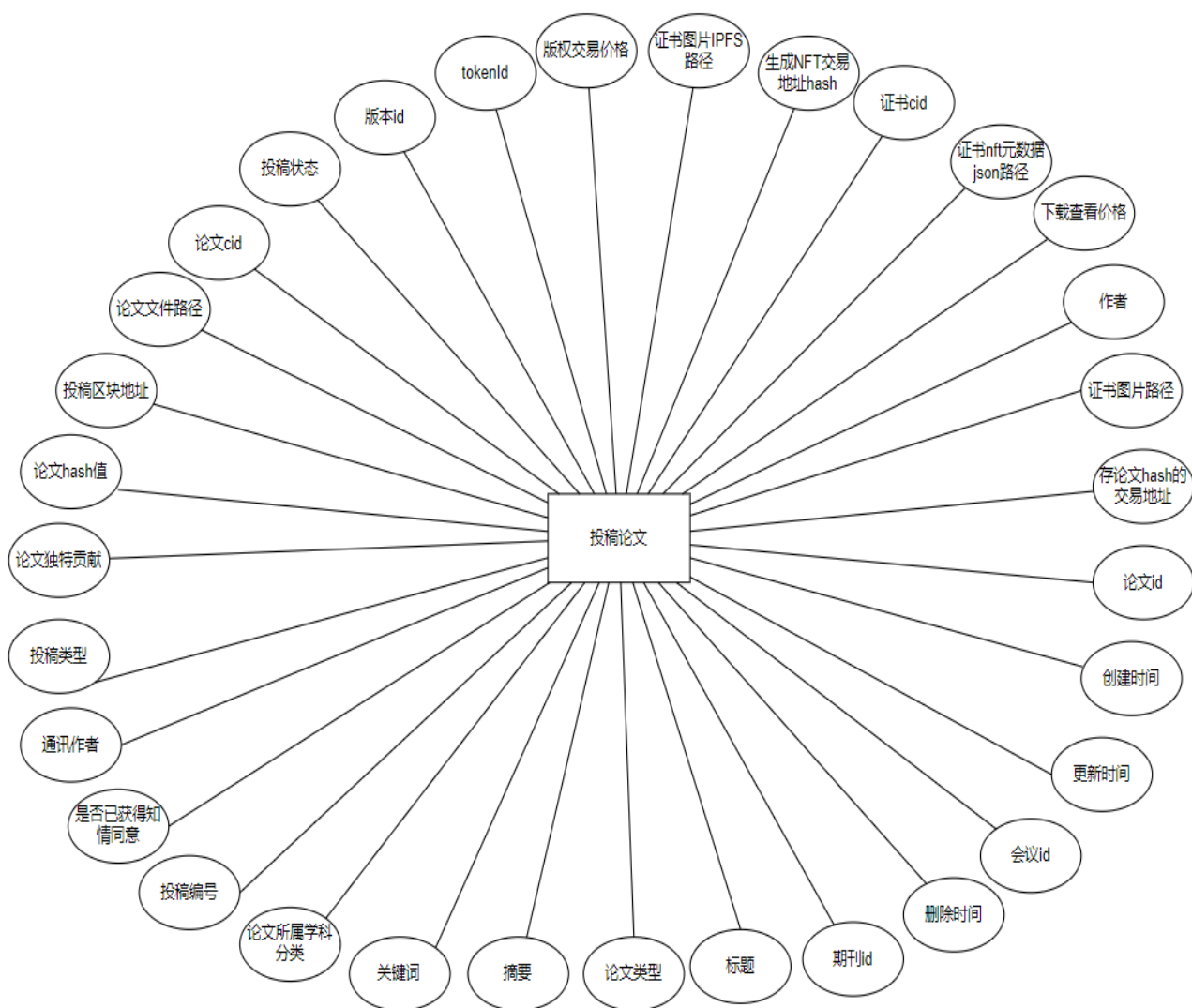


图 4-9 投稿论文实体

投稿内容需要评委进行审评,需要记录用户与投稿之间的关系,记录用户对投稿做出的审评记录,因此应有一个审评信息表,主要字段包含:审稿人 id、审稿人意见,审稿状态以及投稿是否为久版本标识。

论文需要找到对应的组织下进行投稿,通过创建委员会,在委员会下创建期刊或会议,不同组织对应不同信息描述,不同组织下包含不同主席和成员,可以设置投稿时间范围。

委员会包含主要字段有:委员会名称、描述、创建人 id。

会议包含主要字段有:会议名称、描述、创建人 id、会议开始时间和结束时间、地址、所在委员会 id。

期刊包含主要字段有:期刊名称、描述、创建人 id、期刊投稿开始时间和结束时间、

所在委员会 id。

用户管理需要设定不同的角色以及可访问的后台菜单路径。

角色主要信息实体包括：角色 id、角色名称。

菜单主要信息实体包括：菜单 id、父菜单 id、菜单名称、路由名称、前端对应路径。

3. 物理设计

通过对系统的分析与设计，需要对应的实体进行具体的物理设计，所需的所有表名罗列出如下：

表 4-1 系统所使用的表

序列	表名	说明
1	users	记录用户信息
2	authorities	存储角色权限信息
3	authority_menus	记录角色权限和菜单的关联信息
4	base_menus	存储基本菜单信息，用于权限管理
5	committees	记录委员会信息
6	conference_issues	存储会议期刊信息
7	conferences	存储会议信息
8	journal_issues	记录期刊信息
9	journals	存储期刊信息
10	paper_viewers	记录论文查看者信息
11	papers	存储论文信息
12	reviews	存储审稿意见
13	user_authority	记录用户和角色权限的关联信息
14	user_committee	记录用户和委员会的关联信息
15	user_conference	记录用户和会议的关联信息
16	user_journal	记录用户和期刊的关联信息
17	user_paper	记录用户和论文的关联信息

用户信息表 users，用户数量理论上无上限，id 可以用存储较大数字的 bigint 类型设为主键，字段信息内容长度都不会太长，字符类型统一用 191 长度，时间用时间类型，性别只有两个状态用一个长度的数字标识，如表 4-2 所示：

表 4-2 用户信息表

序号	字段名称	字段类型	长度	允许空值	主键	备注
1	id	bigint		NO	YES	用户 id
2	created_at	datetime	3	NO		创建时间
3	updated_at	datetime	3	YES		修改时间

续表 4-2

序号	字段名称	字段类型	长度	允许空值	主键	备注
4	deleted_at	datetime	3	YES		删除时间
5	authority_id	bigint		NO		用户角色 id
6	sex	tinyint		NO		性别 1 男 2 女
7	username	varchar	191	NO		用户登录名
8	password	varchar	191	NO		用户登录密码
9	first_name	varchar	191	NO		姓氏
10	last_name	varchar	191	NO		名字
17	research	varchar	191	NO		研究方向
18	block_chain_address	varchar	191	NO		区块链账户地址
19	affiliation	varchar	191	YES		所属机构

投稿信息表 papers，投稿数量理论上无上限，id 可以用存储较大数字的 bigint 类型，字段信息内容长度不长的字符类型统一用 191 长度，对于需要填写内容较长的标题和摘要部分用存储长度较大的 text 字段类型，时间用时间类型，投稿信息表 papers 如表 4-3 所示：

表 4-3 投稿信息表

序号	字段名称	字段类型	长度	允许空值	主键	备注
1	id	bigint		NO	YES	投稿论文 id
2	created_at	datetime	3	NO		创建时间
3	updated_at	datetime	3	YES		修改时间
4	deleted_at	datetime	3	YES		删除时间
5	conference_id	bigint		YES		会议 id
6	journal_id	bigint		YES		期刊 id
7	title	varchar	191	NO		标题
8	paper_type	varchar	191	NO		论文类型
9	abstract	text		NO		摘要
10	key_words	text		NO		关键词
11	subject_category	varchar	191	NO		论文所属学科分类
12	cor_author	varchar	191	NO		通讯作者
13	manuscript_type	varchar	191	NO		投稿类型
14	unique_contribution	varchar	191	NO		论文独特贡献
15	hash	varchar	191	NO		论文 hash 值
16	block_address	varchar	191	NO		投稿区块地址
17	filepath	varchar	191	NO		论文文件路径

续表 4-3

序号	字段名称	字段类型	长度	允许空值	主键	备注
18	manuscript_id	varchar	191	NO		投稿编号
19	cid	varchar	191	NO		论文 cid
20	status	varchar	191	NO		投稿状态
21	version_id	bigint		NO		版本 id
22	image_token_id	bigint		YES		证书图片 token_id
23	copyright_trading_price	bigint		YES		版权交易价格
24	image_uri	varchar	191	YES		证书图片 IPFS 路径
25	image_cid	varchar	191	YES		证书图片 cid
26	json_uri	varchar	191	YES		证书 nft 元数据 json 路径
27	download_price	bigint		YES		下载价格
28	transaction_hash	varchar	191	YES		交易地址
29	authors	varchar	191	YES		作者
30	token_id	varchar	191	YES		token id
31	image_url	varchar	191	YES		证书图片路径
32	paper_transaction_hash	varchar	191	NO		存论文 hash 的交易地址

审稿信息用于关联 users 表和 papers 表, 需要存储审稿人 id 与投稿论文 id, 使审稿人与投稿论文建立联系, old_version 标识是否为历史版本以标识是否还需评审, 审评意见可以输入较多内容, 使用 text 字段类型, 表 reviews 如表 4-4 所示:

表 4-4 审稿信息表

序号	字段名称	字段类型	长度	允许空值	主键	备注
1	id	bigint		NO	YES	id
2	created_at	datetime	3	NO		创建时间
3	updated_at	datetime	3	YES		修改时间
4	deleted_at	datetime	3	YES		删除时间
5	reviewer_id	bigint				审稿人 id
6	paper_id	bigint				论文 id
7	comment	text				审稿意见
8	status	varchar	191			审稿状态
9	old_version	tinyint	1			是否历史版本

4.4 总结

通过本章对系统设计的详细阐述,明确了基于联盟链的学术出版平台的整体架构和各个模块的功能。本章为系统的具体实现提供了详细的设计蓝图和指导原则,接下来的章节将进一步介绍系统的具体实现过程以及测试结果,验证系统设计的合理性和功能的完善性。

第5章 系统实现

在系统设计完成后，接下来的关键步骤是系统的具体实现。本章将详细描述基于联盟链的学术出版平台的系统实现过程，包括开发环境的搭建、主要功能模块的实现以及智能合约的部署等。通过 Hyperledger besu 搭建联盟链节点网络，使用的开发工具包括 Visual Studio Code、Remix、Goland 编译器，主要的开发语言包括：Solidity、JavaScript、Go 语言。通过详细的步骤讲解和技术细节说明，展示如何将设计转化为实际应用，以确保平台的功能完善和运行稳定。

5.1 Hyperledger 节点搭建联盟链

在 docker 容器中配置好对应的 Hyperledger 节点，并且设置好对应的端口映射和内部 docker 网络。配置好联盟链的创世文件，其他每个节点都需要以此文件为创世文件才能接入该联盟链网络中。

1. 创世文件内容和启动配置文件内容如图 5-1。

配置说明：

- blockperiodseconds: 出块时间。
- epochlength: 重置所有投票的块数。
- extraData: 包括最初的签名者
- gasLimit: 天然气价格，当天然气是免费的，限制区块和合约大小就不那么重要了。在自由气体网络中，增加区块大小的限制，并将合约大小的限制设置为最大值。
- alloc: 初始账户和该账户所拥有的 coin。

2. 启动配置文件如图 5-2 所示。

[illegible]

图 5-1 创世文件配置

```
data-path="/Clique-Network/node-1/data"
genesis-file="/Clique-Network/cliqueGenesis.json"
network-id=1337
rpc-http-enabled=true
rpc-http-api=["CLIQUE","ETH","NET","WEB3","TXPOOL","ADMIN","TRACE","DEBUG"]
host-allowlist=["*"]
rpc-http-cors-origins=["all"]
rpc-http-host="0.0.0.0"
rpc-ws-enabled=true
rpc-ws-api=["ADMIN","CLIQUE","ETH","NET","WEB3","DEBUG","TXPOOL"]
rpc-ws-host="0.0.0.0"
min-gas-price=0

rpc-http-port=8545
rpc-ws-port=8546

nat-method="NONE"
p2p-host="172.16.170.218"
p2p-port="30303"
```

图 5-2 启动配置文件

参数说明要求:

- `--data-path=data`: 保存的数据存到 `data` 目录。
- `--genesis-file=./cliqGenesis.json`: 创世文件每个节点必须保持一致。
- `--network-id 123`: 全部节点的 `network id` 要一致,就是创世文件里的 `chainId`, `chainId` 优先级比较高。

- `--host-allowlist=""`: 可以连接的主机, *代表所有主机。
- `--rpc-http-host=0.0.0.0`: 可以让其他主机连接(可以在网络上访问到)。
- `--p2p-host=192.168.28.3`: 修改 enode url 上的 ip 地址, 默认是 127.0.0.1, 这里用的是 docker 容器里的虚拟机 ip, 如果要对外暴露 ip 要设成主机 ip。
- `--bootnodes=<Node Enode URL>`: 用来发现 Node 节点的, 等于哪个的 enode, 就连接那个节点。
- `--min-gas-price=0`: 汽油费设置为 0。
- `--rpc-http-api=["CLIQUE","ETH","NET","WEB3","TXPOOL","ADMIN","TRACE","DEBUG"]`: 开放 api。
- `--rpc-ws-api=["ADMIN","CLIQUE","ETH","NET","WEB3","DEBUG","TXPOOL"]`: 开放 api。
- `--rpc-ws-enabled=true`: 启动 ws。

3. 执行启动命令, 启动联盟链节点

5.2 MetaMask 钱包连接联盟链

搭建完联盟链后, 配置浏览器插件 MetaMask Web3 钱包, 连接上联盟链网络, 如图 5-4 所示:

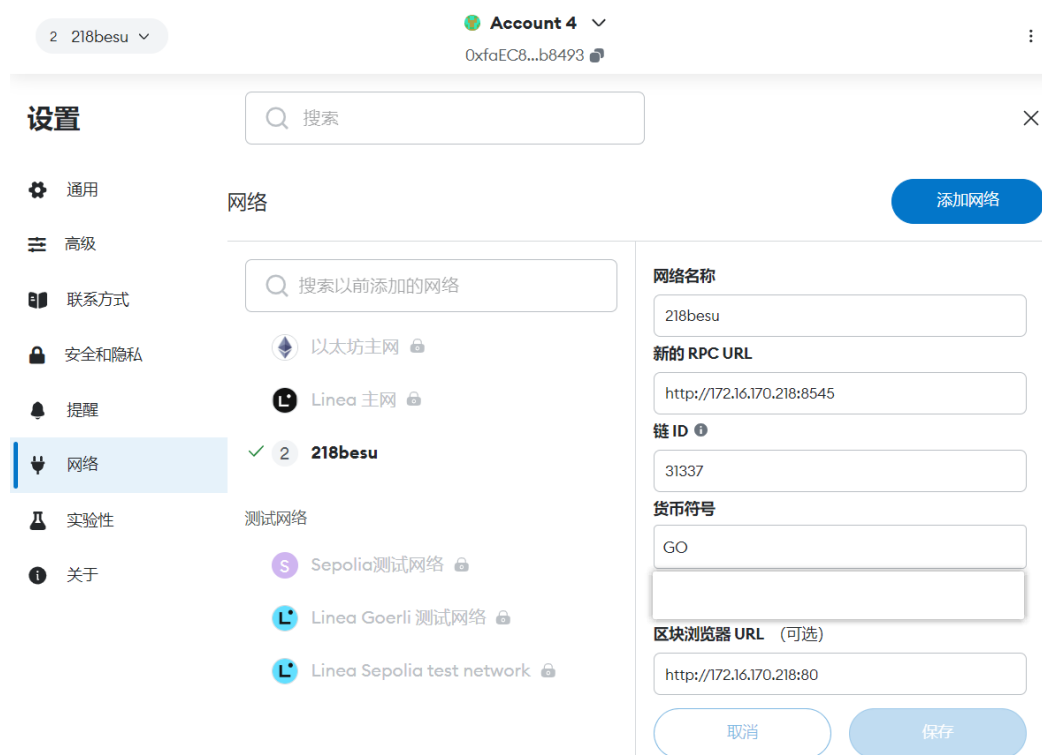


图 5-4 连接联盟链网络

5.3 IPFS 系统

IPFS（Inter Planetary File System，星际文件系统）是永久的、去中心化保存和共享文件的方法，这是一种内容可寻址、版本化、点对点超媒体的分布式协议。进入 IPFS 官网下载 IPFS 的源代码，选择适合的版本的进行下载，编译安装，运行脚本 `./install.sh`，执行 `ipfs init` 进行初始化，默认配置文件内容。启动 IPFS: `ipfs daemon`，如图 5-5 所示：

```
[root@localhost ~]# ipfs daemon
Initializing daemon...
Kubo version: 0.20.0
Repo version: 13
System version: amd64/linux
GoLang version: go1.19.8

Computed default go-libp2p Resource Manager limits based on:
- 'Swarm.ResourceMgr.MaxMemory': "135 GB"
- 'Swarm.ResourceMgr.MaxFileDescriptors': 4096

These can be inspected with 'ipfs swarm resources'.

Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/127.0.0.1/udp/4001/quic
Swarm listening on /ip4/127.0.0.1/udp/4001/quic-v1
Swarm listening on /ip4/127.0.0.1/udp/4001/quic-v1/webtransport/certhash/uEiBQZzejrnuMaeXFyx_9xYzExmkUAuLMoejnLP_g-vzjKg/certhash/uEiCnz2k8Ke-MAjpb_k1LK9MQ8223MmujcYugNTQIJF4F5g
Swarm listening on /ip4/172.16.170.195/tcp/4001
Swarm listening on /ip4/172.16.170.195/udp/4001/quic
Swarm listening on /ip4/172.16.170.195/udp/4001/quic-v1
Swarm listening on /ip4/172.16.170.195/udp/4001/quic-v1/webtransport/certhash/uEiBQZzejrnuMaeXFyx_9xYzExmkUAuLMoejnLP_g-vzjKg/certhash/uEiCnz2k8Ke-MAjpb_k1LK9MQ8223MmujcYugNTQIJF4F5g
Swarm listening on /ip4/172.16.239.1/tcp/4001
Swarm listening on /ip4/172.16.239.1/udp/4001/quic
Swarm listening on /ip4/172.16.239.1/udp/4001/quic-v1
Swarm listening on /ip4/172.16.239.1/udp/4001/quic-v1/webtransport/certhash/uEiBQZzejrnuMaeXFyx_9xYzExmkUAuLMoejnLP_g-vzjKg/certhash/uEiCnz2k8Ke-MAjpb_k1LK9MQ8223MmujcYugNTQIJF4F5g
Swarm listening on /ip4/172.17.0.1/tcp/4001
Swarm listening on /ip4/172.17.0.1/udp/4001/quic
Swarm listening on /ip4/172.17.0.1/udp/4001/quic-v1
Swarm listening on /ip4/172.17.0.1/udp/4001/quic-v1/webtransport/certhash/uEiBQZzejrnuMaeXFyx_9xYzExmkUAuLMoejnLP_g-vzjKg/certhash/uEiCnz2k8Ke-MAjpb_k1LK9MQ8223MmujcYugNTQIJF4F5g
Swarm listening on /ip4/172.18.0.1/tcp/4001
Swarm listening on /ip4/172.18.0.1/udp/4001/quic
Swarm listening on /ip4/172.18.0.1/udp/4001/quic-v1
Swarm listening on /ip4/172.18.0.1/udp/4001/quic-v1/webtransport/certhash/uEiBQZzejrnuMaeXFyx_9xYzExmkUAuLMoejnLP_g-vzjKg/certhash/uEiCnz2k8Ke-MAjpb_k1LK9MQ8223MmujcYugNTQIJF4F5g
Swarm listening on /ip4/172.31.1.1/tcp/4001
Swarm listening on /ip4/172.31.1.1/udp/4001/quic
Swarm listening on /ip4/172.31.1.1/udp/4001/quic-v1
```

图 5-5 启动 IPFS 节点

启动 IPFS 后，可以访问提供的 UI 界面，获取 IPFS 的状态以及配置信息，如图 5-6 所示：

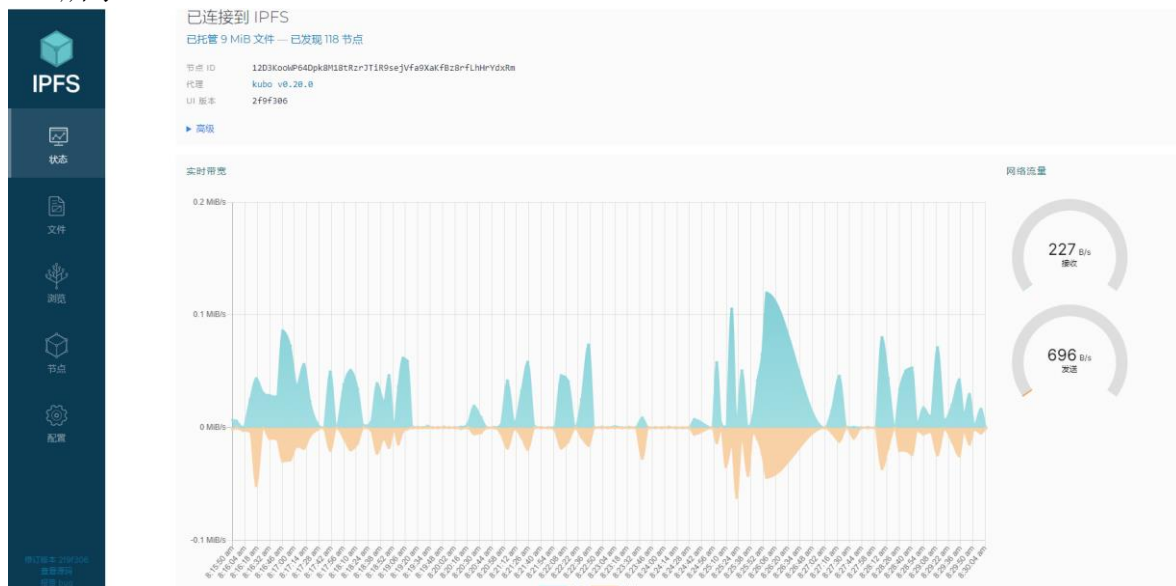


图 5-6 IPFS UI 交互页面

5.4 区块链浏览器

Chainlens Blockchain Explorer 支持公共和私有 EVM 网络，Chainlens 提供了整个网络的概览，包括区块信息、合约元数据、交易搜索等；从 GitHub 下载，配置相关文件，运行 docker-compose 启动浏览器服务，如图 5-7 所示：

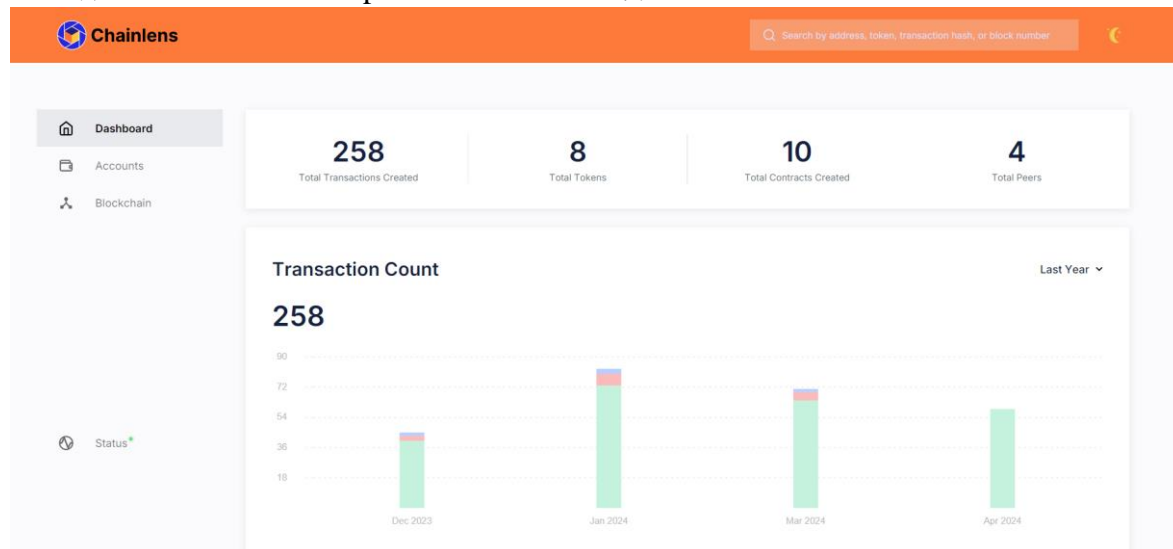


图 5-7 区块链浏览器 UI 查询页面

5.5 智能合约接口功能实现

Remix IDE 被用于每个知识水平的用户在智能合约开发的整个过程中。它不需要设置，促进了快速的开发周期，并具有丰富的插件和直观的 GUI。用 solidity 语言编写智能合约代码，包括三个文：基于 ERC20 协议的 Token 发行合约、基于 ERC721 协议的 NFT 铸造合约以及整合以上两个合约的 NFT 交易合约。

1. 基于 ERC20 协议的 Token 发行合约接口功能

主要有八个接口功能：

- `totalSupply()` public view returns (uint256): 查询代币的总供应量。
- `balanceOf(address account)` public view returns (uint256): 查询指定账户的代币余额。
- `transfer(address recipient, uint256 amount)` public returns (bool): 将代币从调用者账户转移到指定账户。
- `allowance(address owner, address spender)` public view returns (uint256): 查询某个账户允许另一个账户可以转移的代币数量。
- `approve(address spender, uint256 amount)` public returns (bool): 授权某个账户可以转移指定数量的代币。
- `transferFrom(address sender, address recipient, uint256 amount)` public returns (bool): 从指定账户转移代币到另一个账户，需满足授权条件。

- `storeHash(string memory hash) public`: 函数允许用户将哈希值存储到智能合约中, 将哈希值与调用者的地址相关联。每个哈希值只能由一个地址存储, 如果哈希值已经存在, 则只有之前存储了该哈希值的地址才能更新它, 或者该哈希值未被存储过。
- `getRecipientByHas`: 函数允许用户根据哈希值查询对应的账户地址。

2. 基于 ERC721 协议的 NFT 铸造合约

- `balanceOf(address owner) external view returns (uint256)`: 查询指定账户拥有的 ERC721 代币数量。
- `ownerOf(uint256 tokenId) external view returns (address)`: 查询指定 ERC721 代币的所有者地址。
- `approve(address to, uint256 tokenId) external`: 授权另一个地址管理指定的 ERC721 代币。
- `getApproved(uint256 tokenId) external view returns (address)`: 查询被授权管理指定 ERC721 代币的地址。
- `setApprovalForAll(address operator, bool approved) external`: 设置指定地址是否具有管理所有 ERC721 代币的权限。
- `isApprovedForAll(address owner, address operator) external view returns (bool)`: 查询指定地址是否被授权管理所有 ERC721 代币。
- `transferFrom(address from, address to, uint256 tokenId) external`: 将指定 ERC721 代币从一个地址转移到另一个地址。
- `safeTransferFrom(address from, address to, uint256 tokenId) external`: 安全地将指定 ERC721 代币从一个地址转移到另一个地址, 触发接收地址的 ERC721 接收函数。
- `safeTransferFrom(address from, address to, uint256 tokenId, bytes memory data) external`: 安全地将指定 ERC721 代币从一个地址转移到另一个地址, 并传递额外的数据, 触发接收地址的 ERC721 接收函数。

3. NFT 交易合约

- `buy(uint256 _tokenId) external`: 购买指定 `tokenId` 的 ERC721 代币。
- `cancelOrder(uint256 _tokenId) external`: 取消指定 `tokenId` 的订单, 将 ERC721 代币退还给卖家。
- `changePrice(uint256 _tokenId, uint256 _price) external`: 修改指定 `tokenId` 的订单价格。
- `isListed(uint256 _tokenId) public view returns (bool)`: 检查指定 `tokenId` 的 ERC721 代币是否已经被列出销售。
- `onERC721Received(address operator, address from, uint256 tokenId, bytes`

calldata data) external returns (bytes4): 接收 ERC721 代币, 并将其加入市场销售列表。

- getOrderLength() external view returns (uint256): 获取当前市场上订单的数量。
- getAllNFTs() external view returns (Order[] memory): 获取当前市场上所有的订单。
- getMyNFTs() external view returns (Order[] memory): 获取调用者在市场上正在出售的订单。

4. 将合约部署至联盟链

合约编写完成后将三个合约通过 MetaMask 钱包签名部署至联盟链, 如图 5-8 所示:



图 5-8 部署合约

区块链浏览器可以便捷的查找到部署合约的地址以及 Transaction 信息, 如图 5-9 所示:

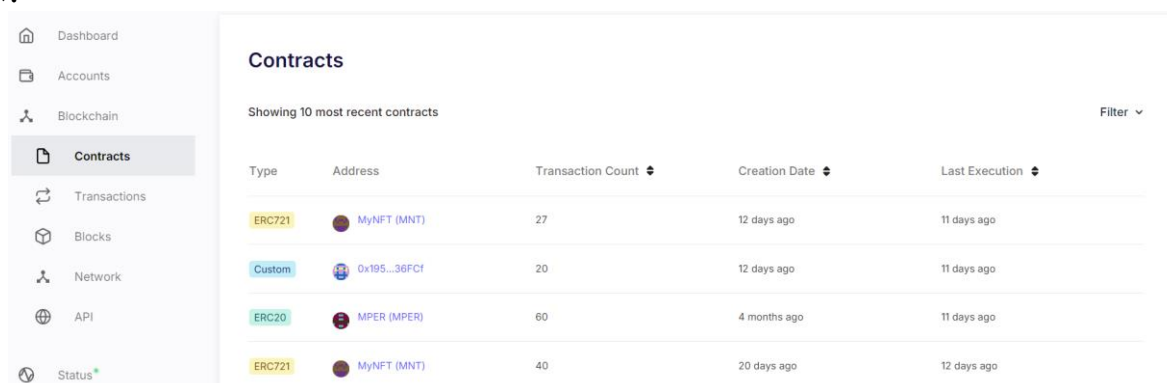


图 5-9 区块链浏览器查看合约部署信息

5.6 前后端功能实现

IPFS、区块链浏览器和智能合约已经启动并部署完成，这些基础的区块链开发环境就变得完善起来。传统的前后端开发可以轻松地调用这些接口来实现平台的各项功能。通过与 IPFS 集成，系统可以实现分布式文件存储和检索，保障了论文和相关资料的高效、安全地存储和获取。而借助区块链浏览器，用户可以方便地查看区块链上的交易记录和智能合约状态，确保了整个系统的透明度和可追溯性。智能合约的部署则为平台提供了自动化的业务逻辑处理，确保了数据的安全性和不可篡改性。这样一来，传统前后端可以借助这些基础设施，更加便捷地开发出功能丰富、安全可靠的学术出版平台。

1. 后端接口实现

后端根据需求实现不同模块所需的接口，采用了 Go 语言作为开发语言，结合了 Gin 框架和 Gorm ORM，利用 Goland IDE 进行开发和调试。在开发过程中，使用了 FoxAPI 进行接口测试，并生成了详细的接口文档。这样的技术组合不仅能够提高开发效率，还能保证后端代码的质量和稳定性。使用 docker 进行容器镜像打包，保存至阿里云，可以随时在不同服务器上拉取运行。

- 后端项目目录结构如图 5-10:

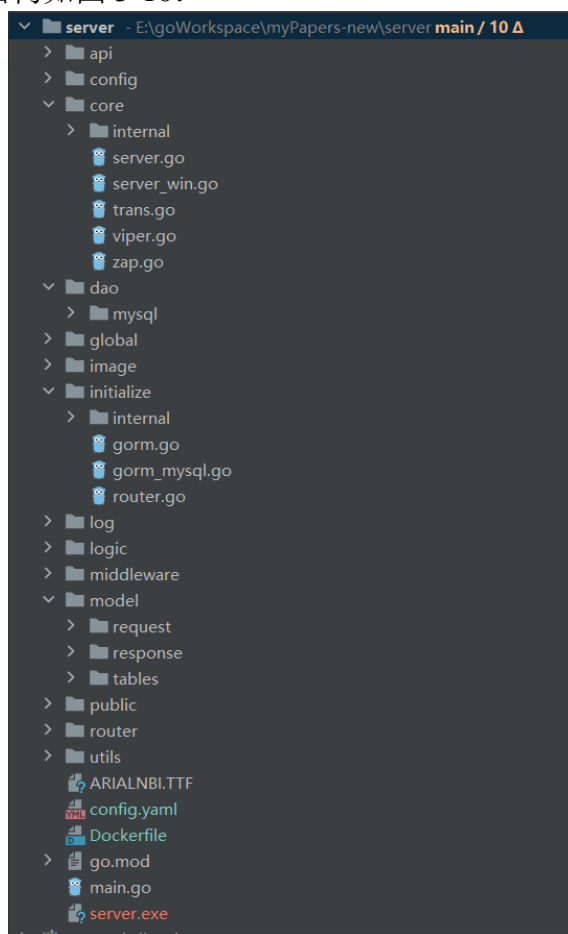


图 5-10 后端目录结构

- 分不同模块实现不同 Api 接口:

用户基础模块：包含所有网络应用基本的权限如表 5-1 所示：

表 5-1 用户基础模块 API 表

请求类型	接口名称	接口分组	接口路径
POST	注册	基础模块	/mypapers/user/register/
POST	登录	基础模块	/mypapers/user/login/
GET	获取自身信息	用户模块	/mypapers/user/getSelfInfo/
PUT	设置自身信息	用户模块	/mypapers/user/setSelfInfo/
POST	修改密码	用户模块	/mypapers/user/changePassword/
PUT	设置用户权限	用户模块	/mypapers/user/setUserInfo/
GET	获取所有用户	用户模块	/mypapers/user/list/
POST	创建 Issue	Issue 模块	/mypapers/journal/issue/create/
GET	查询 Issue	Issue 模块	/mypapers/journal/issue/list/
DEL	删除 Issue	Issue 模块	/mypapers/journal/issue/delete/
PUT	更新 Issue	Issue 模块	/mypapers/journal/issue/update/

组织管理模块：包含对委员会、会议、期刊的管理，如表 5-2 所示：

表 5-2 系组织管理模块 API 表

请求类型	接口名称	接口分组	接口路径
POST	创建	委员会模块	/mypapers/committee/create
PUT	更新	委员会模块	/mypapers/committee/update/
GET	详情	委员会模块	/mypapers/committee/detail/
GET	列表	委员会模块	/mypapers/committee/list/
DEL	删除	委员会模块	/mypapers/committee/delete/
POST	创建	会议模块	/mypapers/conference/create/
PUT	更新	会议模块	/mypapers/conference/update/
GET	详情	会议模块	/mypapers/conference/detail/
GET	列表	会议模块	/mypapers/conference/list/
DEL	删除	会议模块	/mypapers/conference/delete/
POST	创建	期刊模块	/mypapers/journal/create/
PUT	更新	期刊模块	/mypapers/journal/update/
GET	详情	期刊模块	/mypapers/journal/detail/
GET	列表	期刊模块	/mypapers/journal/list/
DEL	删除	期刊模块	/mypapers/journal/delete/

投稿审评模块：包含了平台核心功能投稿模块和审稿复杂的逻辑处理内容，如表 5-3 所示：

表 5-3 投稿审评模块 API 表

请求类型	接口名称	接口分组	接口路径
POST	分配	审稿模块	/mypapers/review/allot
POST	投稿	投稿模块	/mypapers/paper/submit/
GET	详情	投稿模块	/mypapers/paper/detail/
GET	自己的投稿列表	投稿模块	/mypapers/paper/selfList/
GET	期刊或会议投稿列表	投稿模块	/mypapers/paper/list/
PUT	更新	投稿模块	/mypapers/paper/update
DEL	删除	投稿模块	/mypapers/paper/delete
GET	生成证书	投稿模块	/mypapers/paper/honoraryCertificate/
POST	发布投稿	投稿模块	/mypapers/paper/publish/
POST	增加 paper 的可查看用户	投稿模块	/mypapers/paper/addPaperViewer/
GET	查看用户是否有权查看投稿	投稿模块	/mypapers/paper/checkPaperViewer/
PUT	修改 NFT 价格	投稿模块	/mypapers/paper/updatePrice/
GET	获取我的 NFT	投稿模块	/mypapers/paper/myNFTs
GET	列表	审稿模块	/mypapers/review/list
POST	审稿	审稿模块	/mypapers/review/submit

2. 前端实现

后端与智能合约接口都实现后,前端设计有好多交互界面调用接口展示信息给用户。前端采用了 Vue2 框架,并结合了 Web3.js 技术。Vue2 提供了组件化开发能力和响应式数据绑定,使得端界面的开发更加简洁高效。而 Web3.js 则为与区块链进行交互提供了便捷的工具,使得前端可以直接与区块链网络进行通信,实现诸如交易签名、合约调用等功能,为用户提供更加丰富的区块链应用体验。前后台界面如图 5-11, 5-12 所示:

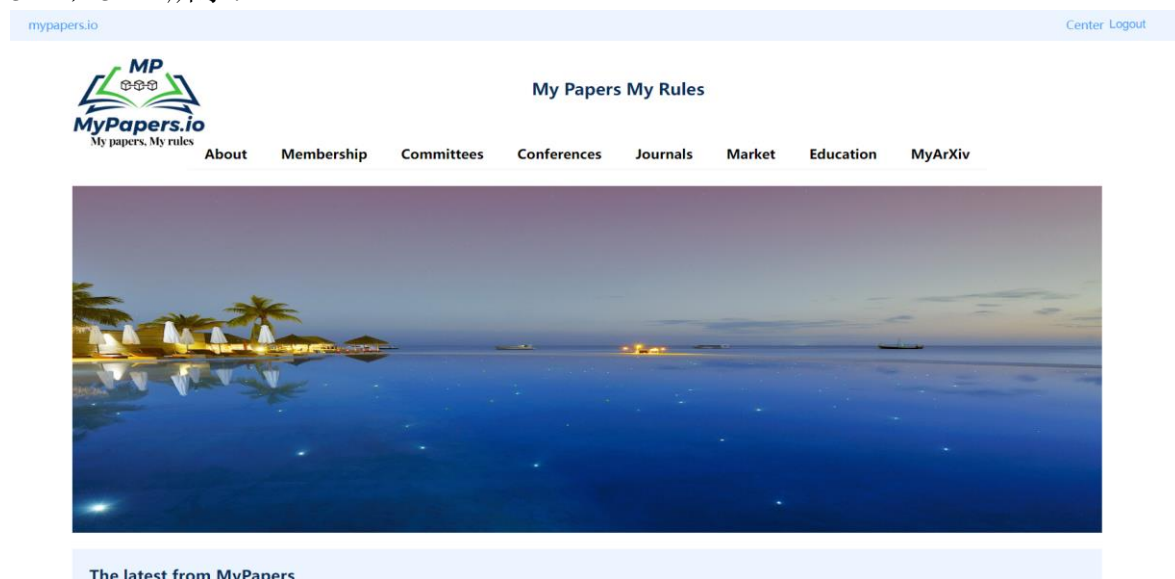


图 5-11 前端主界面

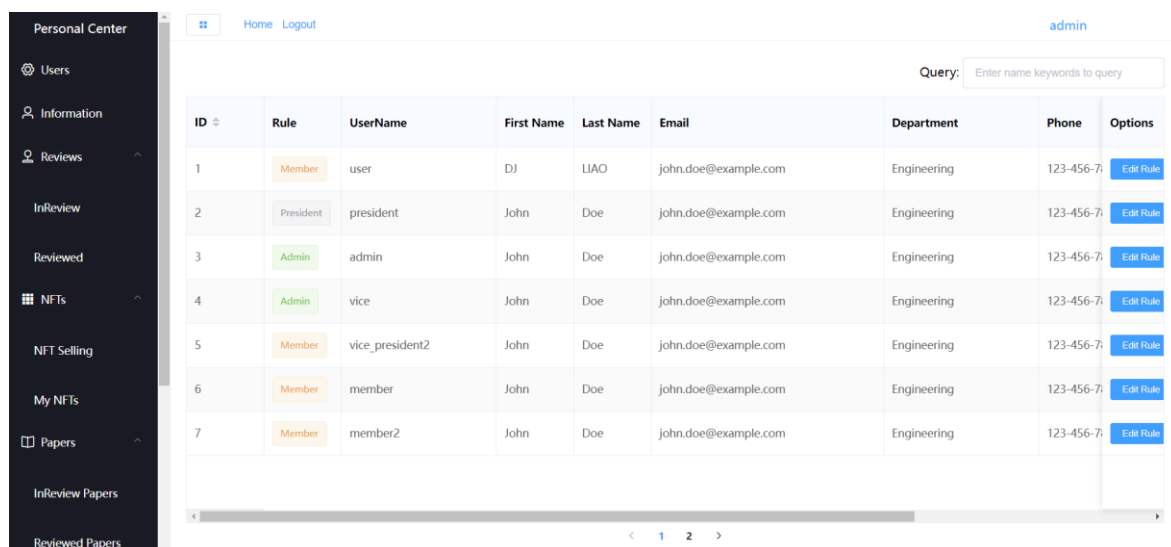


图 5-12 前后台界面

3. 投稿前端实现

在主页面的选择时间范围内可投稿的期刊或会议投稿，作者可以选择将投稿提交到特定的学术期刊或会议，上传文件按钮选择论文上传，上传后通过调用智能合约 `HashStored()` 函数接口，浏览器弹出 `MetaMask` 钱包确认签名将论文的 `hash` 值保存至联盟链上，表单收集标题、摘要、作者、关键词、论文所属学科分类、通讯作者等信息，选择论文类型，提交投稿调用后端 `SubmitPaper()` 函数接口，等待评审；跳转至后台投稿列表界面，对还未审核的投稿可进行自由修改信息。

4. 审评前端实现

投稿完成后会自动分配给对应会议或期刊的主席，主席可在后台待审核列表查看到投稿列表，点击分配按钮，表单输入审稿人用户名，确认分配调用后端 `AllotReviewers()` 接口；分配后投稿人不可再对投稿进行修改，审稿人同样可以在后台待审核列表查看到投稿列表，可查看详情，对论文和投稿信息进行查看后，进行审评，点击审评按钮，可以选择五种评价：接受，拒绝，小修，大修，拒绝并重投，并附带评论；主席对审稿人的审评进行评估，做出最终审评提出评审意见和建议，调用智能合约函数 `ReviewStored()` 将审评信息存入联盟，调用后端 `SubmitReview()` 接口，保存文件至 `IPFS` 系统，修改投稿论文状态。审评完成后投稿可在已审核列表查看，其投稿的状态包括已提交、待审稿、已接受、已拒绝等状态。

投稿人可以在已审核投稿列表查看投稿，进入详情可以看到所有审评人给出的评审意见和建议，如果要求修改，则会有修改按钮对此次投稿进行修改，这将是一个新的投稿版本，在新的投稿中可以看见旧的投稿版本列表，也可查看详情，修改完成的投稿会再次出现在对应主席的待审核列表，等待进行第二轮审评。

5. 出版交易

投稿人查询已审核通过后的投稿详情，点击出版按钮调用 `GetHonoraryCertificate` 接口，后端自动生成版权证书，前端展示证书内容：作者姓名、论文标题、投稿时

hash 存储至联盟链上的交易地址、该交易所在的区块链地址、出版日期。投稿人可以根据自主对学术作品的质量和重要性评估，制定合适的出版定价以及论文下载定价，输入表单。点击确认调用智能合约 `safeMint()` 接口铸造 NFT，浏览器弹出 MetaMask 钱包要求账户对此次交易进行签名认证生成 NFT，等待联盟链确认交易完成后表示 NFT 证书铸造成功，论文出版成功；此时可在主页面查找到该论文，如果需要下载查看，浏览器弹出 MetaMask 钱包要求账户对此次交易进行签名认证向作者账户地址转账，等待联盟链确认交易完成后即可下载查看论文。

论文出版成功后，在后台可查询到我的 NFT 界面，调用后端 `GetMyNFTs()` 接口以及合约 `ownerOf()` 接口查询已出版的 NFT 证书，可以调用合约接口 `changePrice()` 修改论文查看 Token 价格以及购买 NFT 证书版权的 Token 价格；点击出售按钮调用合约函数，将 NFT 证书上架版权平台交易市场，上架交易市场后，所有用户可在平台主页交易市场查看论文 NFT 售卖信息，用户如果有充足的 Token 余额点击购买，浏览器弹出 MetaMask 钱包要求账户对 NFT 合约进行授权可花费的 Token 以及将 NFT 转给购买者的两次交易进行签名认证，等待联盟链确认交易完成后，NFT 将发生转移，购买者在后台‘我的 NFT’界面可以查看到购买到的 NFT。

5.7 总结

通过对系统实现过程的详细描述，可以看到基于联盟链的学术出版平台在功能上已经达到了预期目标。各个模块之间的协调运作确保了系统的高效性和可靠性，智能合约的应用增强了平台的安全性和自动化水平。接下来的章节将对系统的测试和性能进行详细分析，以验证系统的实际运行效果和各项功能的稳定性，从而为平台的推广应用提供坚实的技术保障。

第6章 系统测试

系统测试是确保基于区块链技术的学术出版平台稳定性、安全性和可靠性的关键步骤。本章将详细介绍系统测试的范围、方法和实施过程，涵盖前端界面、后端逻辑、智能合约、IPFS 分布式存储、区块链节点等方面。通过系统测试，我们旨在验证系统的功能完整性、性能稳定性和安全可靠，为学术出版平台的正式上线提供有力的保障。

6.1 用户基础功能模块测试

1. 用户注册

填写基础信息，其中注意的是需要填写 Blockchain Addr，这个可以从 MetaMask 钱包里的账户地址拷贝，账户地址将作为用户在联盟链操作的凭据，所以是非常重要的，如图 6-1 所示：

The image displays a user registration form and a wallet overview. The registration form has the following fields:

Field	Value
UserName	Blingder
Password	*****
FirstName	JanDe
LastName	Liao
Email	2284486839@qq.com
Department	CS
Phone	15160679556
Address	xiamen, fujian
Education	college
Title	student
Research	blockchain
Blockchain Addr	0xfaEC8610B7310
Affiliations	XMUT
AffiliationType	education

Buttons: Register, Reset

The wallet overview shows:

- Account 4
- Balance: \$456.97 USD, 89849 GO
- Buttons: Buy & Sell, 发送, 兑换, 跨链桥, Portfolio
- Sections: 代币 (Tokens), 收藏品 (Collectibles), 活动 (Activities)
- Token 1: GO, \$456.97 USD, 89849 GO
- Token 2: MPER, 999500120 MPER
- Buttons: + 添加代币, 刷新列表

图 6-1 用户注册

2. 管理用户

注册的用户默认权限有限，管理员可以修改用户权限，使其可以创建委员会、期刊和会议等操作，如图 6-2 所示：

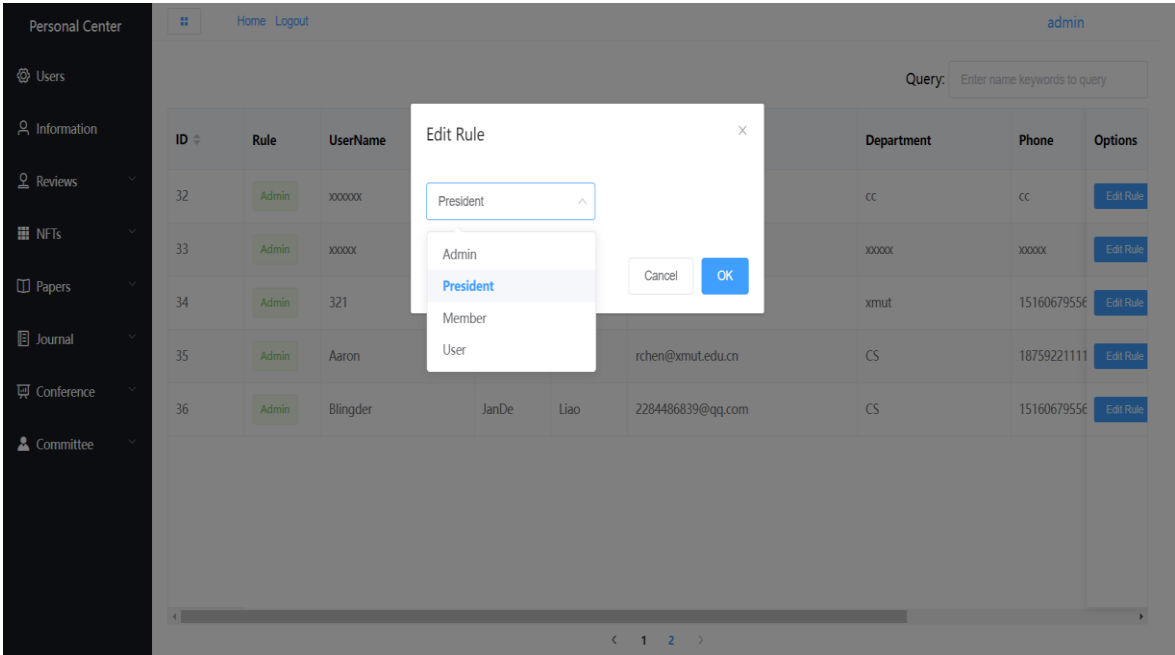


图 6-2 管理用户

6.2 组织模块测试

有了权限，用户可以在平台创建委员会、期刊和会议等学术组织实体。这些操作在流程上保持基本一致，包括填写基础信息、指定主席和成员，并设定运行的投稿日期等关键信息。此外，用户还可以在期刊和会议创建 Issue，以便于管理和跟踪学术作品的投稿流程，确保流程的顺畅进行，如图 6-3，6-4，6-5，6-6 所示：

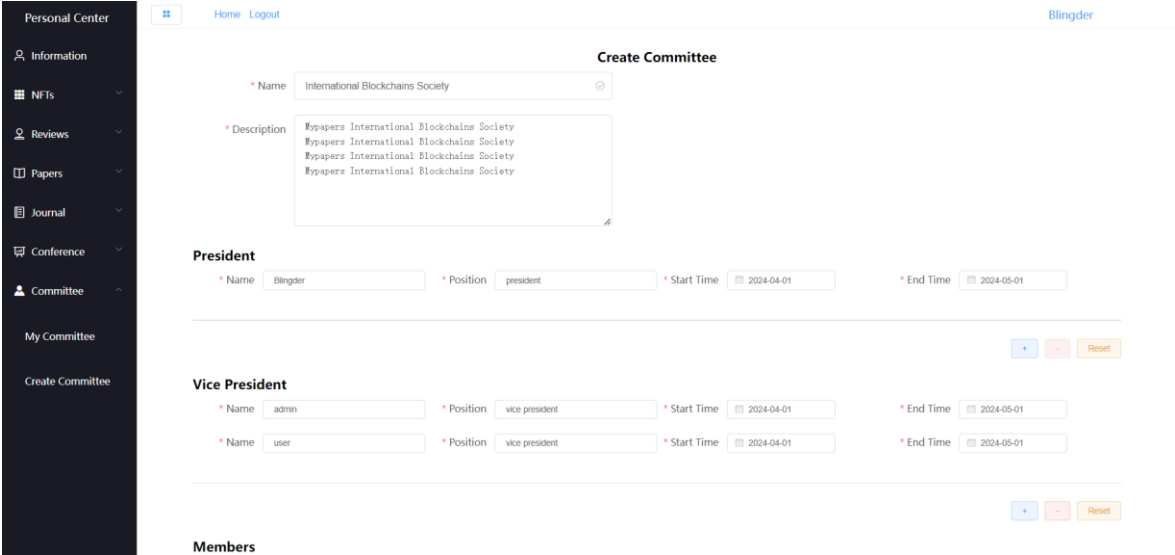


图 6-3 创建委员会



图 6-4 查看委员会详情

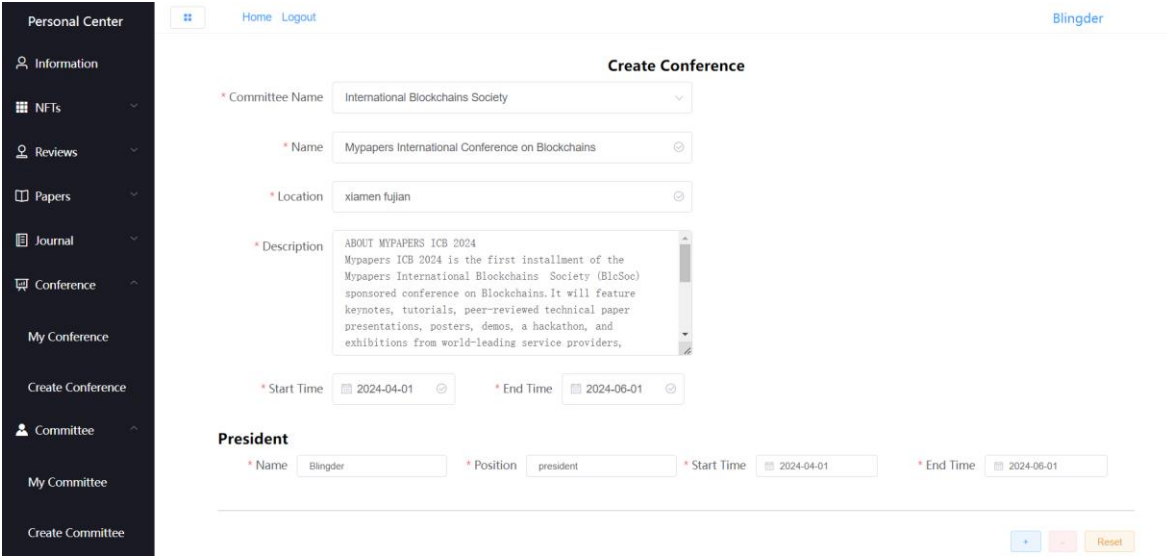


图 6-5 创建会议

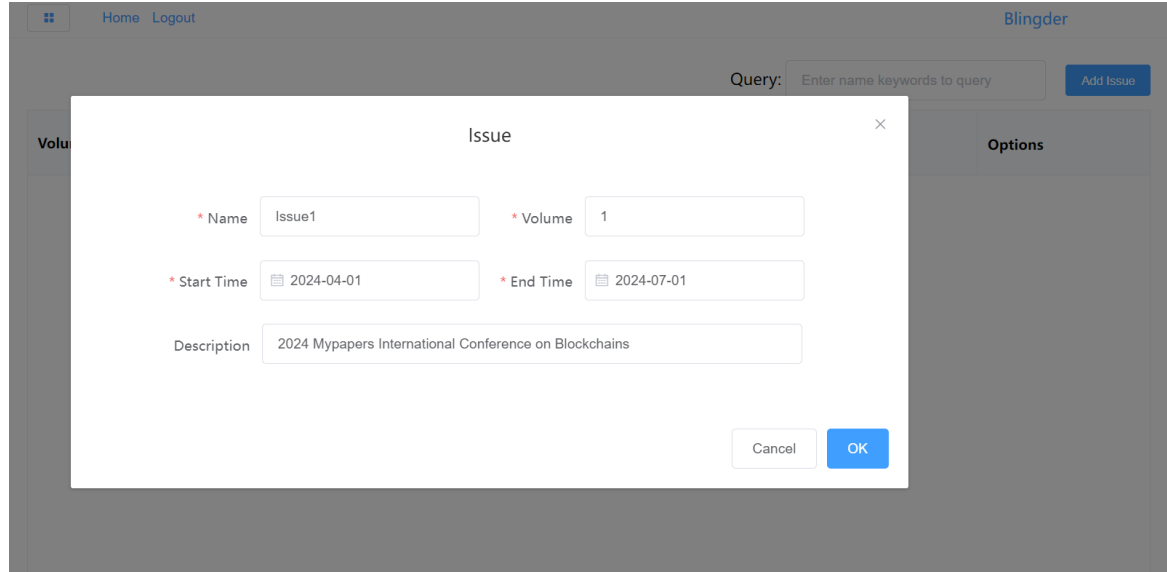


图 6-6 创建 Issue

6.3 投稿模块测试

创建完成会议后，可以在主页面找到会议查看其内容，点击 **Submit Manuscript** 按钮进行投稿，如图 6-7 所示：



图 6-7 主页面浏览会议详情

当提交论文文件时，会将论文取 hash，通过 MetaMask 钱包调用智能合约接口将论文对应的作者账户地址存到联盟链中，从而实现证明论文的作者是谁，上链操作后是不可篡改的，如图 6-8，6-9 所示：

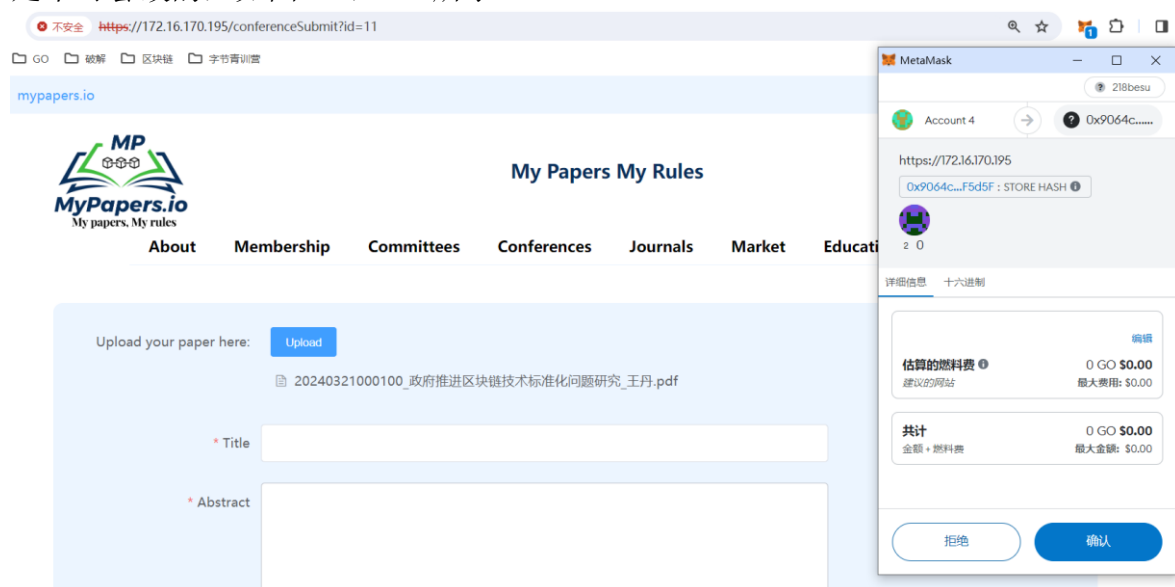


图 6-8 投稿

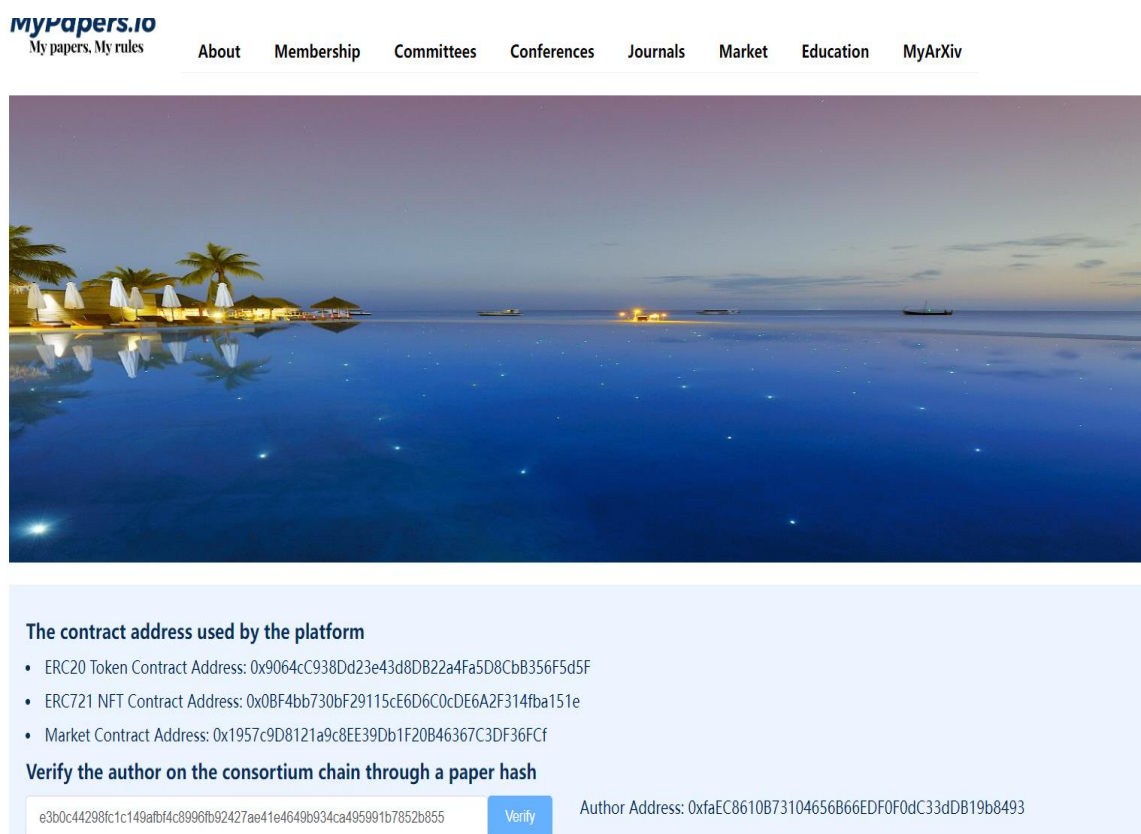


图 6-9 查询验证论文作者

6.4 审稿模块测试

投稿成功后，对应的会议主席在后台可以看到待审核的投稿列表，可以分配审稿人进行审稿，当主席的审核结果为需要修改时，投稿人需要修改后重新提交，之前的版本将保留可查看，如图 6-10，6-11，6-12 所示：

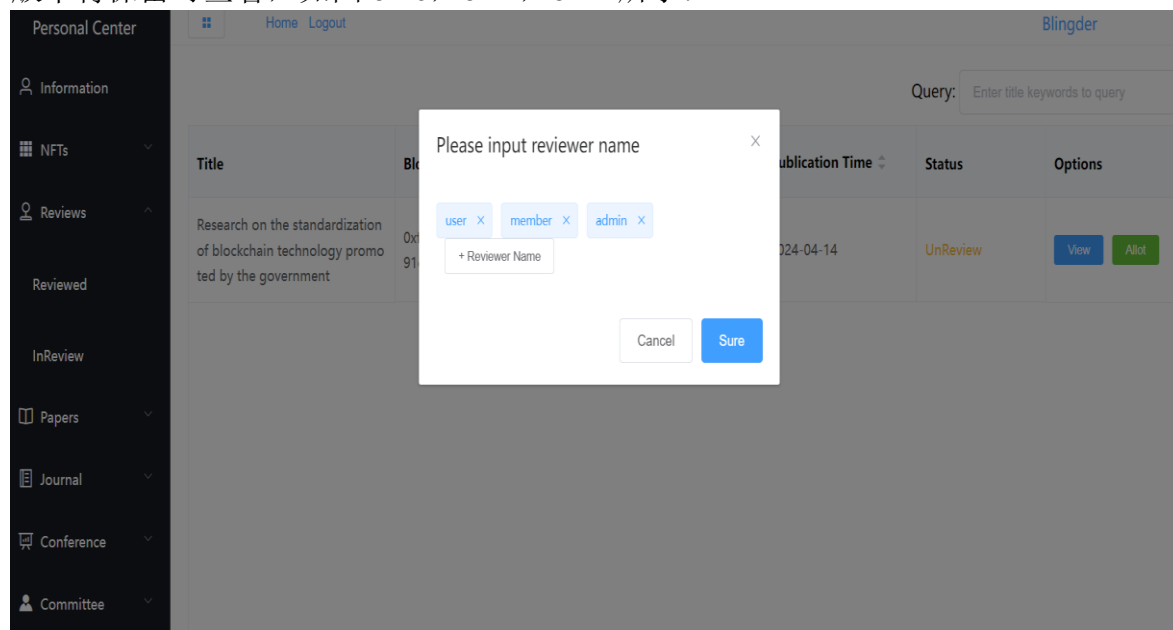


图 6-10 主席查看投稿分配审稿人

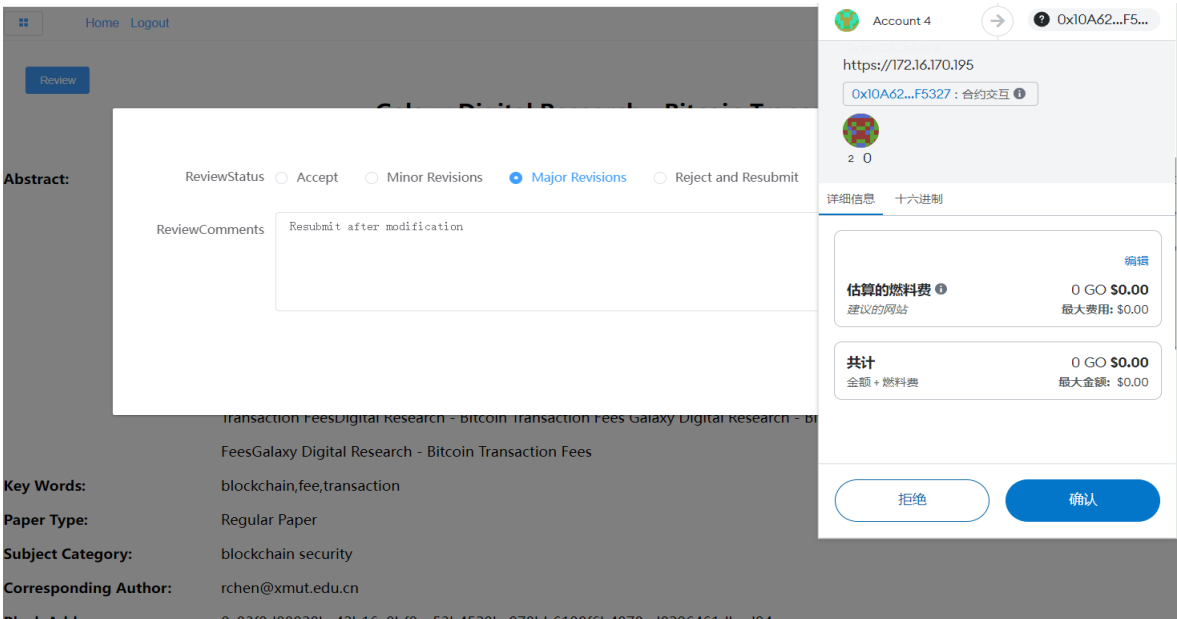


图 6-11 审稿

Historical Version:

Title	Block Address	Publication Time	Status	Options
Research on the standardization of blockchain technology promoted by the government	0xfd656a4e9e51acec2022c3ae544764aa1a2d499e678105cb91cc791d0f12f192	2024-04-14	Major Revisions	<button>View</button>

图 6-12 查看投稿历史版本

6.5 出版交易模块测试

当审核通过后，投稿者在后台的已审核列表中可以看到结果，此时可以设定查看论文的价格已经需要购买版权的价格，如图 6-13 所示：

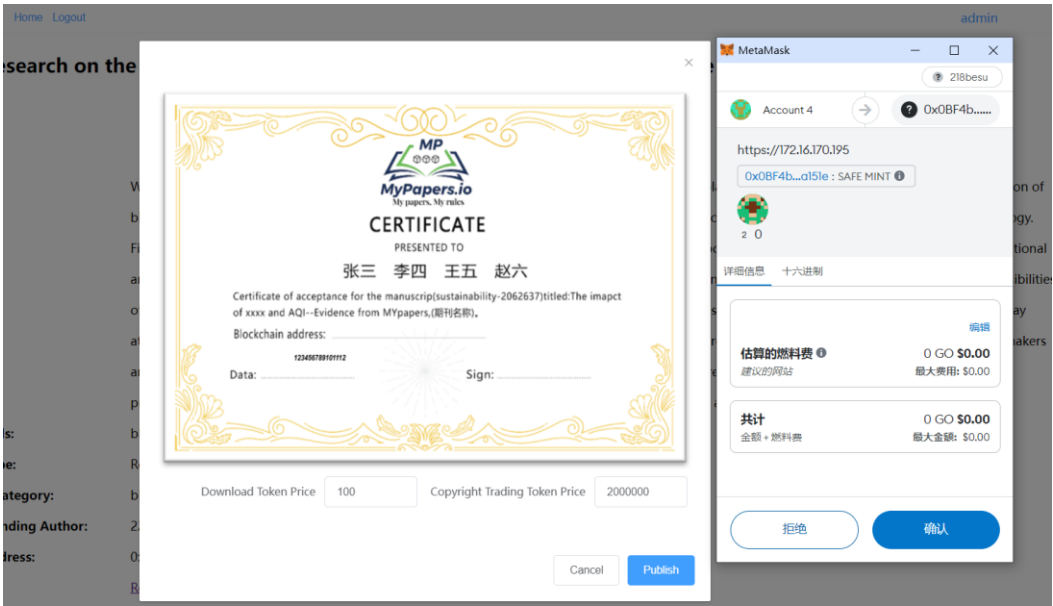


图 6-13 出版生成 NFT

出版完成后可以选择将 NFT 版权证书上架至版权交易市场, 此时还可以改动价格, 如图 6-14 所示:

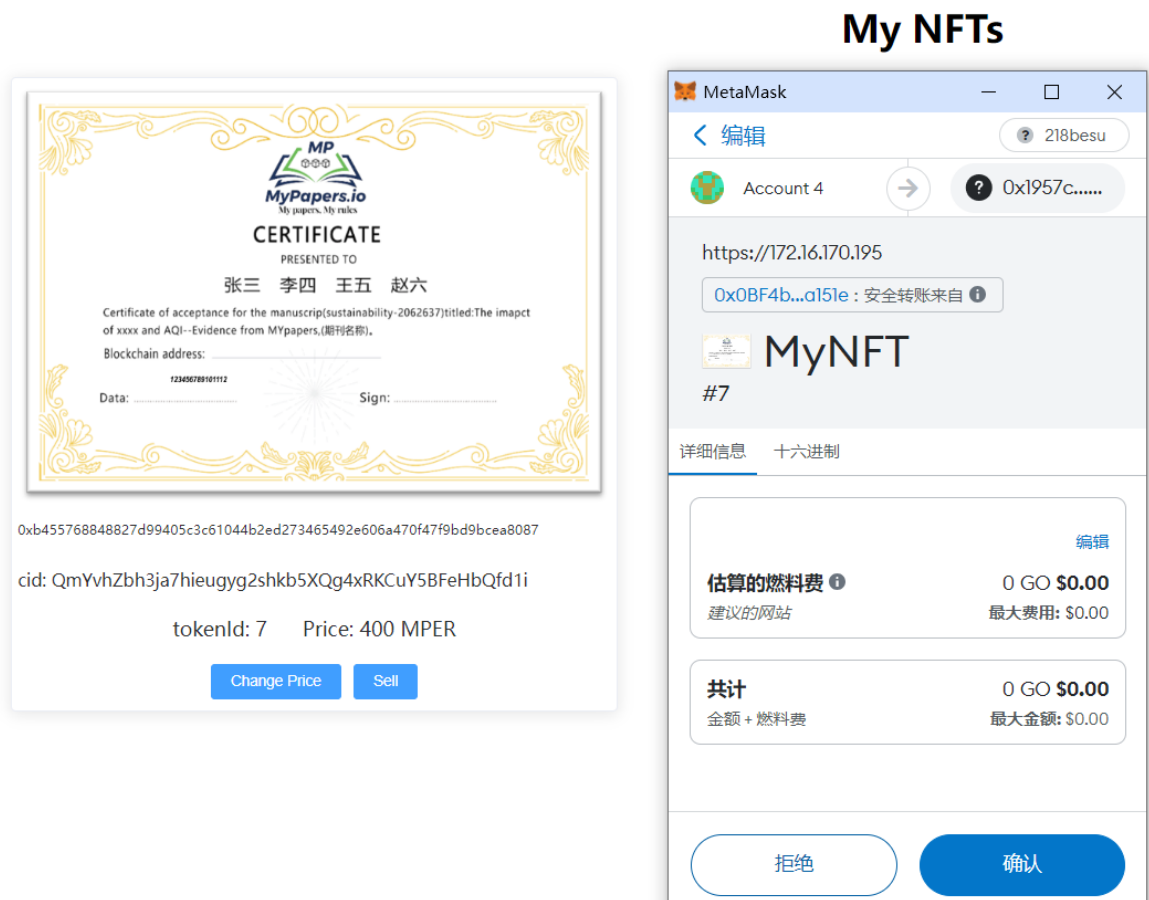


图 6-14 出售 NFT 版权

NFT 上架后所有平台用户可在前台页面的 NFT 交易市场上查看正在售卖的 NFT 版权证书, 如果余额足够将购买成功, 其中的 NFT 拥有权和 Token 转换都将由智能合约自动完成。保证了无需信用交易的高效实现, 如图 6-15, 6-16 所示:

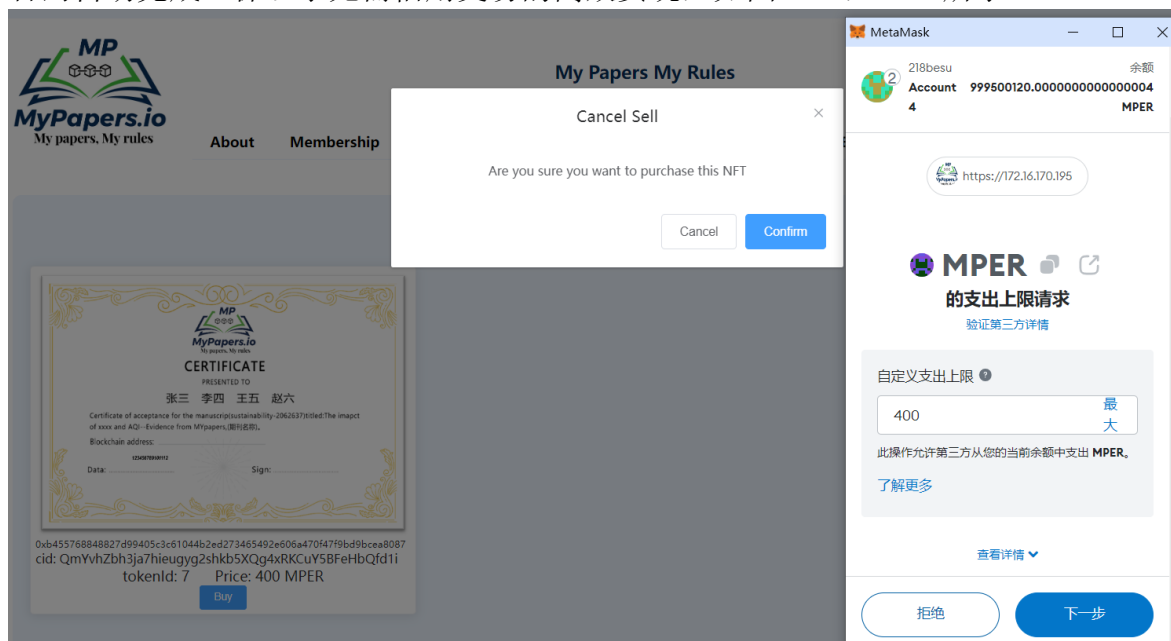


图 6-15 购买 NFT 版权

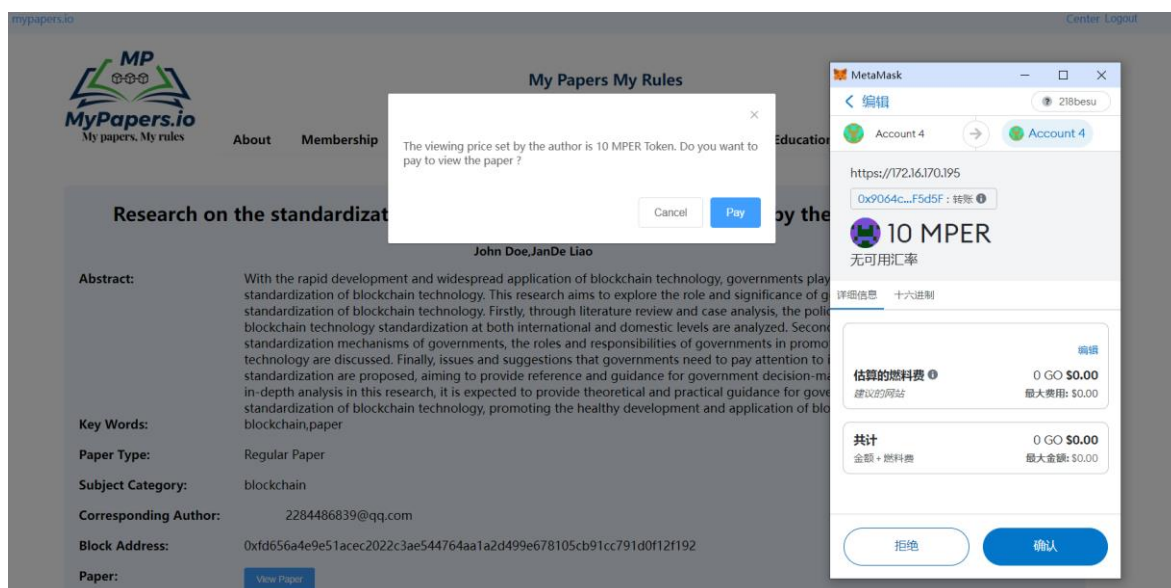


图 6-16 查看论文付费

需要通过 MetaMask 钱包账户签名的操作都被称为交易(Transaction)，都可以在区块链浏览器上查看到签名者进行了什么操作，联盟链中的内容不可被篡改，因而区块链浏览器上的数据都是真实可靠的，如图 6-17 所示：

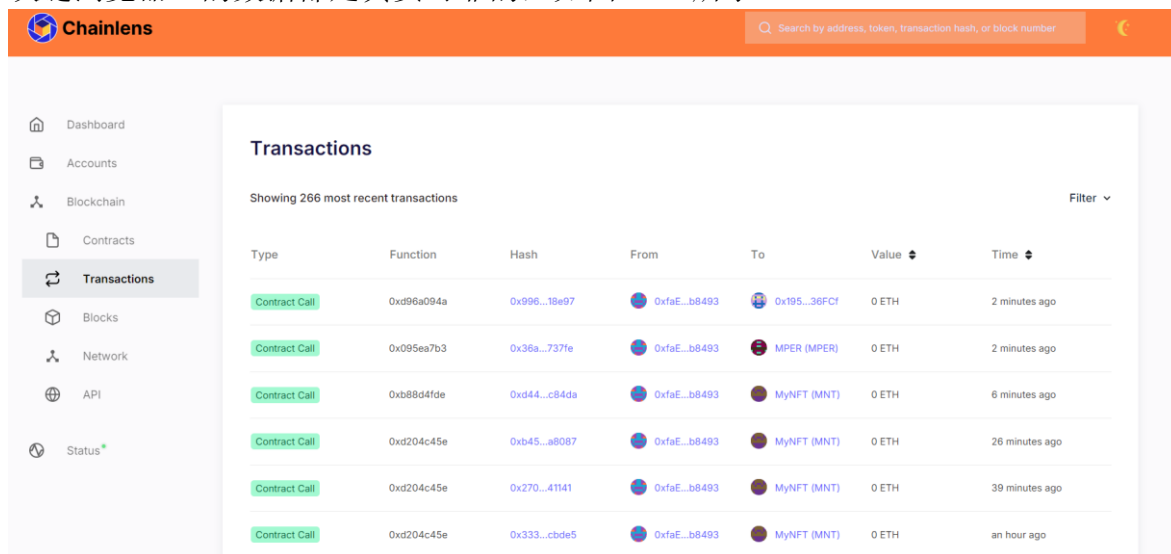


图 6-17 交易信息

6.6 总结

本章对系统测试的各个方面进行了详细的描述和分析。通过功能测试和性能测试，验证了基于联盟链的学术出版平台在实际运行中的可靠性和高效性。测试结果表明，系统在处理用户管理、论文投稿、审稿流程以及出版交易等功能上表现稳定，并且智能合约的应用有效提升了系统的安全性和自动化程度。本章为系统的实际应用提供了数据支持和理论依据，奠定了平台进一步推广和优化的基础。

第7章 结论与展望

7.1 结论

本毕业设计论文致力于基于联盟链技术的学术出版平台的设计与开发。通过对区块链技术和以太坊智能合约的深入学习，结合 Go 语言后端开发技术，成功地实现了一个具有可信度和透明性的学术出版平台原型。在设计过程中，充分考虑了学术出版领域的特点和需求，采用了联盟链技术来确保数据的安全性和可追溯性。通过智能合约实现了作者、审稿人和编辑之间的透明化、自动化的交互过程，极大地提高了出版流程的效率和可信度。通过本次设计与开发，本人深刻认识到区块链技术在学术出版领域的巨大潜力。它不仅可以解决学术领域中存在的信任问题，还可以提高出版流程的透明度和效率，推动学术研究的进步和发展。然而，我也意识到区块链技术在学术出版领域的应用还存在一些挑战和障碍，如性能、扩展性和标准化等方面的限制。未来，本人将继续努力，进一步完善和优化我们的学术出版平台，探索更多创新性的解决方案，为学术出版领域的发展做出更大的贡献。

7.2 进一步工作的方向

在未来，本人将继续深耕区块链领域，不断学习和探索新的技术，努力为社会创造更多有益的应用。同时，本人也希望能够与更多志同道合的伙伴们共同合作，共同推动区块链技术在各个领域的发展，为构建数字化时代的新生态贡献自己的力量。

参考文献

- [1] 冯粒, 袁勃. 习近平主持召开中央全面深化改革委员会第五次会议[OL]. 北京: 人民日报, 2018 (1).
- [2] 培育世界一流科技期刊四部门联合发文推动科技期刊改革发展[OL]. 北京: 新华社, 2018(1).
- [3] 李富山, 赵素婷, 汤宏波. 建设世界一流科技期刊的组织形式探讨[J]. 湖北科技学院学报, 2020,40(06):10-13.
- [4] 朱军文, 刘念才. 高校科研评价定量方法与质量导向的偏离及治理[J]. 教育研究, 2014,35(8):52-59.
- [5] 张小强, 曹馨予. 区块链技术赋能学术期刊深度融合发展的模式与路径研究[J]. 中国科技期刊研究, 2022,33(08):999-1012.
- [6] 谢文亮, 曹渝, 蒋美仕. 科技期刊基于区块链的审稿全流程监管机制研究[J]. 中国科技期刊研究, 2021,32(12):1490-1497.
- [7] Dadkhah M,Rahimnia F,Niyasr. Jourchain: using blockchain to avoid questionable journals[J]. Irish Journal of Medical Science,2022,191(3):1435-1439.
- [8] 郝淼闻. 基于区块链技术促进期刊同行评议制度的进化[J]. 科技传播, 2022,14(7):126-129.
- [9] Coleman A. Assessing the Value of a Journal Beyond the Impact Factor [J]. Journal of the American Society for Information Science and Technology,2007(08):1148-1161.
- [10] 张恬, 孟美任. 学术交流应用区块链技术的场景与案例剖析[J]. 中国科技期刊研究, 2019,30(5):469-475.
- [11] Duh E S,Duh A,Droftina U,et al. Publish-and-flourish: Using blockchain platform to enable cooperative scholarly communication[J]. Publications,2019,7(2):33.
- [12] Mackey T K,Shah N,Miyachi K,et al. A framework proposal for blockchain-based scientific publishing using shared governance[J].Frontiers in Blockchain,2019,2(1):19.
- [13] Wang Y,Zhao L B. Blockchain for scholarly journal evaluation: Potential and prospects[J]. Learned Publishing, 2021(4):682-687.
- [14] 余以胜, 朱佳雨, 许恩平. 基于区块链的学术评价系统模型构建研究[J]. 重庆大学学报, 2020,26(4):138-149.
- [15] 治丹丹. 基于区块链技术的科技期刊网络出版危机韧性治理[J]. 中国科技期刊研究, 2019,30(5):476-480.
- [16] 黎明. 基于区块链的学术评价系统模型构建分析[J]. 黑龙江科学, 2020(16):106-107.

致 谢

转眼间大学四年即将画上一个句号，衷心感谢所有在我学习和研究过程中给予我支持和帮助的人。

首先，我要感谢陈老师和李老师以及学长的耐心指导。他们的专业知识和指导帮助我在区块链领域迈出了坚实的一步，使我能够对区块链和以太坊世界有了全面的了解与认识。

此外，特别感谢北京大学肖臻老师的《区块链技术及应用》公开课以及成都信息工程大学区块链产业学院的梁培利老师的《DApp 设计与开发》课程。这两门课程为我提供了深入的学习资源，让我对区块链技术有了更深层次的理解。感谢 b 站上的李文周老师，他的 Go 语言后端开发课程为我提供了宝贵的学习资料，使我能够掌握必要的技术来实现我的项目。

最后，我要感谢所有在我学习过程中给予我鼓励和支持的家人和朋友。他们的支持和理解是我不断前行的动力源泉。

在这段研究旅程中，有了你们的帮助和支持，我才能够顺利完成这个基于联盟链的学术出版平台的开发。再次向大家表示最诚挚的感谢！

廖建德

2024 年 5 月 15 日