学号\_\_\_\_20194824\_\_\_\_ 密级\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**东北大学本科毕业论文**

基于联盟链的存证数据

加密管理系统设计与实现

学 院 名 称 ：计算机科学与工程学院

专 业 名 称 ：计算机科学与技术

学 生 姓 名 ：齐彦乔

指 导 教 师 ：信俊昌 教授

二○二三年六月

**郑 重 声 明**

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名： 日期：

摘 要

传统数据存证加密管理系统大多与第三方平台合作，依赖中心化管理方式，第三方平台在有关法律事务处理流程中存在大量重复数据，使得查询数据耗时长，效率低。数据存储在第三方平台也存在丢失和篡改的风险，安全性和可靠性得不到保障，且难以实现数据共享。而区块链技术具有去中心化，数据难篡改的特点，联盟链又是许可的区块链技术。基于以上特点，联盟链可以有效解决传统存证系统以上的痛点。本课题的研究目的就是将联盟链与电子存证系统下相结合，实现更加安全的新型存证系统。

本课题从当前研究现状出发，简要分析了该项目的发展前景。接着对该系统将要使用的关键技术进行了介绍。辅以系统用例图对系统进行了功能性需求分析与非功能性需求分析。在系统设计章节，运用分层思想将系统分为区块链网络层，业务逻辑层，用户界面层三层。又对系统进行了功能模块划分，将系统分为登录模块，个人信息管理模块，证据管理模块以及证据分享模块。实现各个模块后，整合为完整的系统。最后对系统进行了模块测试，检测出可能存在的问题并分析解决。

采用联盟链可以有效解决传统存证系统查询时间慢，存在丢失与篡改风险的痛点。联盟链与存证系统相结合可以实现更为安全的新型电子存证加密管理系统

关键词：区块链；存证；数据共享；Fabric

**ABSTRACT**

Most of the traditional data storage systems cooperate with third-party platforms and rely on centralized management methods. Third-party platforms have a lot of duplicate data in the process of legal affairs, which makes querying data time-consuming and inefficient. Data storage on third-party platforms also has the risk of loss and tampering, security and reliability cannot be guaranteed, and data sharing is difficult to achieve. The blockchain technology has the characteristics of decentralization and difficult data tampering, and the alliance chain is a permissioned blockchain technology. Based on the above characteristics, the alliance chain can effectively solve the pain points above the traditional certificate deposit system. The research purpose of this topic is to combine the alliance chain with the electronic certificate deposit system to realize a more secure new certificate deposit system.

Starting from the current research status, this topic briefly analyzes the development prospect of the project. Then the key technologies to be used by the system are introduced. Supplemented by the system use case diagram, the functional requirements analysis and non-functional requirements analysis of the system are carried out. In the system design chapter, the system is divided into three layers: blockchain network layer, business logic layer, and user interface layer by using layered thinking. The system is also divided into functional modules, and the system is divided into login module, personal information management module, evidence management module and evidence sharing module. After each module is realized, it is integrated into a complete system. Finally, the module test is carried out on the system, the possible problems are detected and analyzed and solved.

**Key words:** Blockchain; Deposit evidence; Data sharing; Fabric

**目 录**

[摘要……………………………………………………………….………………………I](#_Toc422211181)

ABSTRACT…………………………………………………………………………….II

1绪论…………………………………………………………………………………….1

1.1研究背景 ……………………………………………………………………………1

1.2国内外充填采矿发展现状…………………………………………………….….…3

1.3国内外充填采矿研究现状……………………………………………….……….…6

1.4本文研究内容和技术路线…………………………………………………..……. 10

1.4.1本文的研究内容……………………………………………………….………. 10

1.4.2本文的研究方法和技术路线…………………………………………………....11

2尾砂胶结充填体力学特性研究………………………………………………....12

[2.1尾砂胶结充填体的物理力学性能及胶凝机理…………………………………... .12](#_Toc422211202)

[2.1.1尾砂胶结充填体的物理力学性能……………………………………………. ..13](#_Toc422211203)

[2.1.2尾砂胶结充填体的胶凝机理………………………………………………….. .15](#_Toc422211204)

[2.2尾砂胶结充填体的力学实验……………………………………………….......... .16](#_Toc422211205)

[2.2.1尾砂物理力学参数测试………………………………………………………. .16](#_Toc422211206)

[2.2.2尾砂胶结充填体力学实验……………………………………………………...17](#_Toc422211207)

[2.3尾砂胶结充填体破坏规律分析………………………………………………........19](#_Toc422211208)

[2.4本章小结……………………………………………………………………........…20](#_Toc422211209)

(各章的名称黑体四号，其余宋体小四号，行间距为1.5倍)

**……**

结论……………………………………………………………………………………..57

参考文献……………………………………………………………………………… 59

附录……………………………………………………………………………………. 62

致谢……………………………………………………………………………………. 72

1 绪 论

* 1. 研究背景
     1. 选题意义

数字资产是现代信息技术高速发展得产物，自然需要对其存储并且防篡改，同时还需要方便对其共享，验证。传统的数据存证系统虽然具有易创建，易存储，成本低等特点，是具有证明力度且可靠得电子数据证据。但是，目前的电子凭证还存在诸多的问题，一是传统的数据存证系统大多与第三方平台合作，依赖中心化管理，第三方平台在有关法律事务处理流程中存在大量重复数据，使得查询数据耗时长，效率低。二是数据存储在第三方平台存在丢失和篡改的风险，安全性和可靠性得不到保障，且难以实现数据共享。而区块链技术加密技术完善与去中心化特点，恰好可以解决传统存证系统以上的痛点。

区块链是一种记录交易历史信息的分布式存储数据库技术，具有去中心化，匿名化等特点，解决了不同节点之间数据共享，安全，信任等问题，在电子货币，金融科技，医疗，能源，互联网等领域发展迅速。在金融领域，区块链技术在精简结算流程，降低交易成本等方面具有极大潜力，在私募股权，追溯交易，查验数据资产归属等方面也体现了其重要性。在医疗领域的应用更是突出，使用区块链可以提供可靠的存储解决方案，确保其存储数据的原始性和可验证性，同时保护患者隐私。在数据存证方面，有人提出利用比特币来存储数据。他们建议对比特币进行修改，利用其挖掘资源以实现归档数据的分布式存储。在电子卫生系统中，曾提出通过联盟区块链实现安全的与具有隐私保护的个人健康信息共享。

基于传统数据存证系统的痛点与区块链去中心化，防篡改的特点，本次毕设提出一个可行的解决方案。在数据存证系统中采用联盟链技术为基础，通过密码学知识保障系统中数据的安全性和隐私性，使节点之间的信任度得到提升。

1.2 国内外研究现状

* + 1. 区块链技术研究现状

在国外，甘俊等人[1]通过改进了PBFT主节点选取方式提高了共识效率。Ge Yu等人[2]通过提出一种动态分组拜占庭容错机制提高区块链的可信度。方轶[3]在PBFT算法上提出了基于EIGamal算法的签名改进方案，实现了节点的动态加入退出。SHI liucheng等人[4]设计了具有唯一性和匿名性的算法领导者，提高了区块链的可信的。Marko Vukoli等人[5]通过对比基于PoW的区块链与基于BFT状态机复制的区块链，提高区块链的可拓展性。Stefanos Leonardos等人[6]利用权重改进PoS共识算法。在国内，丁庭深[7]提出了基于信用等级划分机制，提高了区块链间的通信效率，降低了区块链间的通信开销。

* + 1. 区块链应用研究现状

应用方面，宁梦月等人[8]意在解决数字知识产权在传统情况下无法进行有效保护的问题，提出基于Polkadot的多链异构数字知识版权保护模型系统。Zheng等人[9]利用连续时间马尔可夫链（CTMC）模型来模拟基于实用拜占庭容错共识算法的医疗区块链网络的时间响应。赵海鸿等人[10]实现了一个基于区块链的去中心化的数据分享方式电子合同管理方案。Antonio Clim等人[11]为了确保移动医疗应用数据的无重复性、安全性、完整性，提出了结合区块链技术与移动医疗应用。周茂君等人[12]提出区块链与商品溯源系统相结合，摒弃传统中心化存储的方案，可以有效地提高电商整体信用，可以对恶意行为进行有效地跟踪处罚。

在电子存证领域，田林等人[13]同样提出将区块链与电子病例存证系统相结合，该文将数据存储到区块链当中，解决旧有的中心化存储方式存在的数据易泄露，共享困难等问题。韦智勇等人[14]通过对电子病历存证系统引进区块链，使新系统能够更好的防止信息泄露或篡改的同时还能实现数据共享。

* 1. 本文研究内容和技术路线
     1. 研究内容

本次课题的目的在于设计并实现一个以联盟链为基础的存证数据加密管理系统，其主要解决传统数据存证系统的安全性与可靠性的不足的痛点。

实现以联盟链为基础的分布式存储系统。利用联盟链对“记账者”有一定限制与区块链自身数据不可篡改的特点，能够实现相对保密，有效保障系统安全性和数据隐私性，并能通过共识机制达成一致性的分布式存储系统。

本系统采用分层思想架构，分为三层，从上至下依次为：用户界面层，业务逻辑层，区块链网络层。用户界面层，可分为为UI层与服务请求层， UI层负责PC端界面展示，服务请求层根据用户的行为发起HTTP请求，请求逻辑层中的服务并根据服务的响应进行界面的更新。业务逻辑层，根据应用层发起的请求提供对应的服务。并有智能合约负责与存储层进行交互，可向区块链网络层创建，修改与读取账本数据。 区块链网络层，基于联盟链实现的“部分去中心化”的分布式存储数据库，存储着系统交互产生的数据信息，该系统可分为数据层与网络通信层。数据层包含联盟链的每个区块的逻辑存储结构的具体信息（每个区块的信息，加密算法，上传下载的关键信息等），网络通信层负责每个区块间的信息通信。可与逻辑层中的智能合约服务交互。

* + 1. 技术路线

用户界面层使用UmiJS框架（JavaScript语言的上层框架）编写PC端界面，UmiJS为业界知名web端框架，并内置全局状态方案。使用axios（promise封装的ajax）发起http请求与用户鉴权。

业务逻辑层使用Express框架（Node语言的上层框架）编写服务端提供的服务，区块链网络层选择为Hyperledger Fabric，其提供了node-sdk，便使用node作为web服务器语言。

区块链网络层使用Hyperledger Fabric搭建以联盟链为基础的分布式存储系统，因Hyperledger fabric技术更为成熟，系统会更加稳定。数据层的节点用docker创建，使用docker-compose实现对docker集群的编排。智能合约则使用Go语言实现。

* 1. 本文组织结构

第一章 绪论，阐述了本课题的研究的意义，研究现状，进一步确定了本文的研究内容与技术路线，最后说明全文的组织架构。

第二章 基于联盟链的存证数据加密管理系统的相关技术介绍，介绍系统使用的区块链，Express，等技术。

第三章 基于联盟链的存证数据加密管理系统的系统需求分析，对系统进行了功能性分析与非功能性分析，最后对该系统的可行性进行了分析。

第四章 基于联盟链的存证数据加密管理系统的系统设计，经过系统分析之后对系统进行进一步的设计，为后续的系统实现打好基础。

第五章 基于联盟链的存证数据加密管理系统的系统实现，阐述系统的具体实现。

第六章 基于联盟链的存证数据加密管理系统的系统测试，系统实现过后，需要对系统进行测试。只有经过测试，系统才会愈加完善。

2 相关技术介绍

2.1 区块链

区块链是一种创新的数据传输方式，集合了密码学[15]、分布式存储[16]、智能合约、点对点传输[17]和共识算法[18]等多种新型技术。它采用分布式的存储结构，使得数据不再集中存储，实现了去中心化的特点。在区块链系统中，任何一个节点都可以对数据进行维护，同时共识机制确保节点行为的协调，确保数据处理的一致性。常见的共识机制有工作量证明（PoW）[19]，权益证明（POS）[20]，权威证明（POA），股份授权证明机制（DPS）[21]，实用拜占庭容错机制（PBFT）[22]等。智能合约是在可复制、共享的账本上运行的计算机程序，通过智能合约，用户可以进行约定，并由智能合约对区块链进行数据访问和同步，解决了用户之间的信任问题。区块链中的密码学技术将打包好的数据块通过哈希函数处理成链式结构，使得链上数据具备不可篡改和可追溯等特点，提高了数据的安全性。此外，区块链还采用非对称加密方式进一步增强数据的安全性。

整个区块链体系可以分为应用层、激励层、合约层、共识层、网络层和数据层六个层次结构，每一层都具有其核心功能。数据层是区块链系统的底层数据结构，存放着区块上的所有数据信息。网络层包括了P2P组网技术、传播机制和验证机制等技术，确保节点之间的信息交流和传递。共识层封装了共识算法和共识机制，保证高度分散的节点对区块数据的有效性达成一致共识。激励层将经济因素融入到区块链中，包括发行机制和分配机制，鼓励节点参与安全验证工作。合约层包括各种智能合约、算法、脚本和代码，体现了区块链的可编程性。应用层封装了实际场景和案例，类似于网站和应用程序，展示了区块链的应用价值。

2.2 Hyperledger Fabric 框架

Hyperledger Fabric是一个许可区块链的分布式账本技术。与公共非许可区块链技术，例如比特币（Bitcoin）[23][24][25]和以太坊（Ethereun）等的共同点是，它们都有一个账本，使用智能合约对账本进行受控制的访问，与公共非许可区块链技术之间最大的不同点是，Fabric是许可的，参与者需要向可信赖的成员服务提供者（MSP）注册，而不是像公共非许可区块链技术中参与者之间是匿名或完全不信任的。因为所有参与者是已知的，所以消除了Sybil[26]攻击的风险。

客户端首先要使用组织的证书服务注册与登记，并获取在网络中用于确认身份的加密材料，加入该网络中可正常运行的通道。客户端通过SDK生成交易提案并签名后提交至背书节点。背书节点验证签名后，根据交易提案执行链码生成交易结果。这些结果与签名会作为提案响应返回给客户端。客户端验证背书节点的签名并比较这些提案响应是否相同，当满足用户指定的背书策略后发将交易发送给排序节点。排序节将接收到的交易排序，并将其打包成区块发送给该通道中的所有peer节点。peer节点对区块内的交易进行验证，验证通过后更新账本。

2.3 Express框架

Express是一款开源，灵活，基于Node的web开发框架。以面向切面编程的思想设计框架的结构。该框架解决原生node开发web服务器时，代码过于冗杂的痛点。该框架简单但因其设计思想为面向切面编程，所以故拓展性强，开发者可开发各种中间件以实现不同功能[27]。譬如JWT中间件可实现鉴权功能，Upload中间件可实现后端解析formData中的文件的功能，CORS跨域中间件可实现跨域功能。

2.4 JWT

JSON Web Token（JWT）是一个使用JSON在各方之间安全地传输信息的一个开放标准。它由头部（Header）、有效载荷（Payload）以及签名（Signature）组成[28]，上述三部分之间用点（.）连接。

头部（Header）包含了token的类型与算法名称。token类型通常为“JWT”，算法名称可选择HMAC或RSA[29]。载荷信息可以存储签发者，面向的用户或过期时间戳等，也可自定义字段。签名是为了保证信息是可被验证且可信的，通过使用Base64URL编码前两部分内容再使用头部包含的加密算法加密后生成。

在使用JWT进行鉴权的认证过程中，当用户注册或登录成功后，后端将会根据用户的登录以及个人信息生成token，该token通常会被设置为cookie或者HTTP请求头中的authorization属性，并在前端发起的HTTP请求含有的token过期或错误时返回预设的响应码，当前端收到后端的响应码为预设的JWT鉴权错误响应码，前端将跳转到登录界面以获取新的token。如果token被设置为cookie，前端将不需要进行多余操作。如果token预期成为HTTP请求头中的authorization属性，那么前端需要拦截除注册、登录外的所有请求，为其请求头设置authorization属性。

2.5 本章小结

本章对本系统用到的相关技术进行了介绍，包括区块链网络使用的Hyperledger Fabric框架，业务逻辑层使用基于Node的快速，开放，极简的Express框架。用户展示层使用基于TS的企业级React应用框架UmiJS，并采用JSON Web Token标准鉴权。

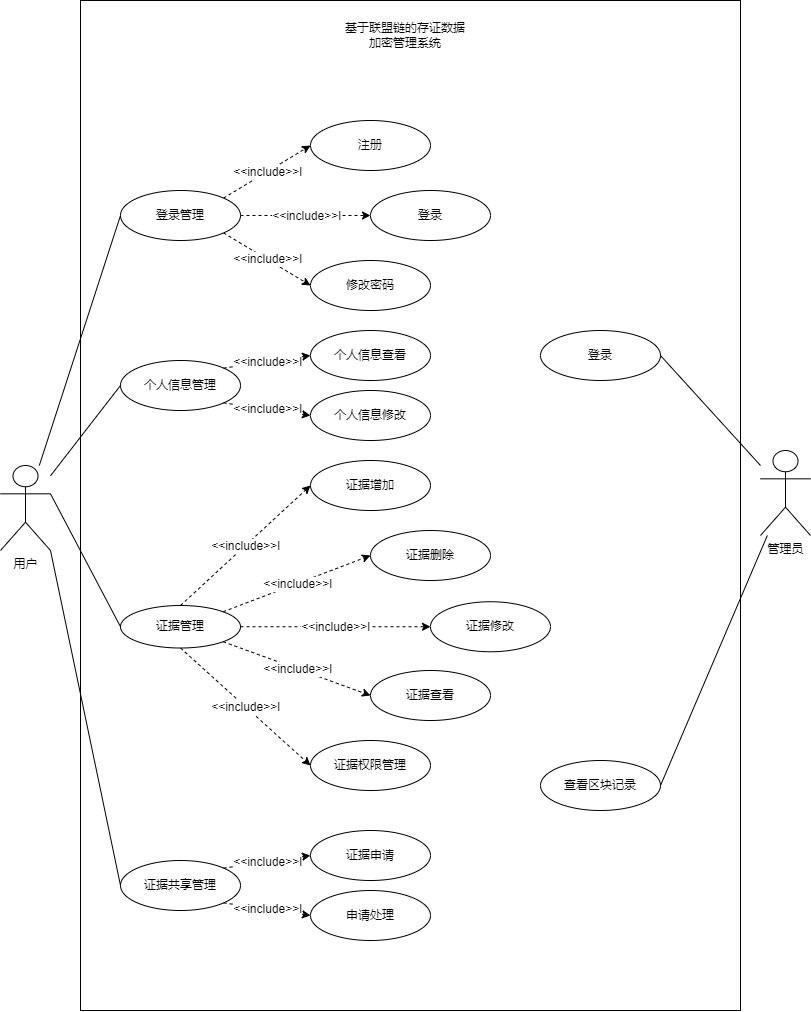
3 系统需求分析

3.1 系统需求概述

基于联盟链的存证数据加密管理系统的基础是存证数据加密管理系统，所以要实现传统存证数据加密管理系统的基本功能，例如注册、登录、修改密码、创建证据、删除证据、修改证据信息、查询证据、证据信息与文件加密、证据共享等。本课题的核心则是要将联盟链技术与存证数据加密管理系统相结合，在传统存证数据加密管理系统的基础上充分利用联盟链的特点，解决传统存证数据加密管理系统核心痛点。

3.2 功能性需求分析

依据需求给出本系统的用例图，如图3.1所示

图3.1系统用例图

在用户使用电子存证加密管理系统时，可以根据用例快速，简单地处理自己的需求。系统包括常规用户与管理员两种参与者。参与者都需要账号进行登录才能使用系统的各种功能。常规用户需要通过注册创建账号，注册成功后即可凭借账号登录进入系统以充分且独立的使用系统的各种功能。而管理员的账户密码是系统预设，无需注册。常规用户在登录进入系统后可以进行个人信息的管理，其中包含对个人信息的查看以及对个人信息的修改。用户也可对证据进行管理，用户可以创建、删除、查询证据，修改、加密证据信息以及对证据进行权限管理。另外，该系统也提供证据共享功能，用户可申请证据库中的证据信息，或申请证据的下载，用户也可处理其他用户对我的证据进行的申请，进一步提高了用户间进行协作的能力，增强了系统的互动性，灵活性。

系统功能性需求如表3.1所示

系统功能性需求表3.1

|  |  |  |
| --- | --- | --- |
| 需求名称 | 需求说明 | 参与者 |
| 注册 | 用户通过邮箱创建账户用于登录。 | 用户 |
| 登录 | 用户注册后，可登录进入系统 | 用户 |
| 修改密码 | 用户可修改自己的密码，修改后需重新登录 | 用户 |
| 个人信息查看 | 查看已经录入的个人信息 | 用户 |
| 个人信息修改 | 修改已经录入的个人信息 | 用户 |
| 证据创建 | 通过上传的文件与用户填入的信息创建证据，并可选择其加密方式 | 用户 |
| 证据删除 | 删除指定的证据 | 用户 |
| 证据修改 | 修改指定的证据 | 用户 |
| 证据查看 | 查看指定的证据的详细信息或下载该证据 | 用户 |
| 证据权限管理 | 对指定的证据的权限进行管理，可设置为私有，完全公开，只可下载或只可查看证据信息 | 用户 |
| 证据申请 | 在证据库（包含所有用户公开的证据）中申请其他用户公开的证据。 | 用户 |
| 申请处理 | 当B用户申请A用户公开的证据时，A用户可选择同意或拒绝 | 用户 |

接下来将以用例表的方式对电子存证加密管理系统的用例图的用例进行必要的说明。

注册功能的用例表如表3.2所示

表3.2注册用例表

|  |  |
| --- | --- |
| 用例名称 | 注册 |
| 用例描述 | 用户通过邮箱创建账户用于登录。 |
| 参与者 | 用户 |
| 前置条件 | 用户还没有创建账户。 |
| 后置条件 | 如果学生注册账号后，可凭借该账号进入系统以使用各种功能。 |
| 基本路径 | 1. 用户进入系统注册界面。 2. 用户输入邮箱，密码以及再次确认密码。 3. 系统对输入的邮箱，密码以及再次确认密码进行有效性检查，检测输入是否合法。 4. 系统验证邮箱是否已被注册。 5. 用户可通过该账号密码进行登录 |
| 拓展点 | 用户输入邮箱，密码或再次确认密码不合法，系统将显示对应的错误信息。  系统验证邮箱错误，系统将显示“邮箱验证失败”错误信息。  若邮箱已被注册，系统将显示“邮箱已注册”错误信息。 |

登录功能的用例表如表3.3所示

表3.3登录用例表

|  |  |
| --- | --- |
| 用例名称 | 注册 |
| 用例描述 | 用户登录可以选择填写邮箱与密码，或者采取邮箱验证方式登录 |
| 参与者 | 用户 |
| 前置条件 | 用户已通过系统的注册功能创建账号。 |
| 后置条件 | 用户登录进入系统后，可以使用系统的各种功能 |
| 基本路径 | 1. 用户进入系统登录界面。 2. 用户可以选择输入自己的邮箱，密码或邮箱，邮箱验证码。 3. 系统对输入的邮箱，密码进行有效性检查，检测输入是否合法 4. 系统对输入的邮箱，密码或者邮箱验证码进行验证。 5. 系统记录并显示当前登录用户。 6. 用户可使用各种系统功能。 |
| 拓展点 | 用户输入邮箱，密码或再次确认密码不合法，系统将显示对应的错误信息。  系统未找到该邮箱的账号信息，系统将显示“邮箱未注册”错误信息。  系统验证邮箱错误，系统将显示“邮箱验证失败”错误信息。  系统登录失败，系统将显示“密码或账户错误”错误信息 |

修改密码功能用例表如表3.4所示

表3.4修改密码用例表

|  |  |
| --- | --- |
| 用例名称 | 修改密码 |
| 用例描述 | 用户登录可以选择填写邮箱与密码，或者采取邮箱验证方式登录 |
| 参与者 | 用户 |
| 前置条件 | 用户已凭借系统注册的账号登录进入系统。 |
| 后置条件 | 用户修改密码后会跳转至系统登录界面，需要重新登录以判断当前用户是否拥有权限继续操作系统。 |
| 基本路径 | 1. 系统进入修改密码界面。 2. 用户输入邮箱、邮箱验证码、新密码与再次确认新密码。 3. 系统对输入的邮箱，密码进行有效性检查，检测输入是否合法。 4. 系统对输入的邮箱，新密码以及再次确认新密码进行验证 5. 系统跳转至系统登录界面 |
| 拓展点 | 用户输入邮箱，新密码或再次确认新密码不合法，系统将显示对应的错误信息。  系统未找到该邮箱的账号信息，系统将显示“邮箱未注册”错误信息。  系统验证邮箱错误，系统将显示“邮箱验证失败”错误信息。  系统修改密码失败，系统将显示“密码修改失败”错误信息 |

个人信息查看功能用例表如表3.5所示

表3.3个人信息查看用例表

|  |  |
| --- | --- |
| 用例名称 | 个人信息查看 |
| 用例描述 | 查看已经录入的个人信息。 |
| 参与者 | 用户 |
| 前置条件 | 用户已凭借系统注册的账号登录进入系统。 |
| 后置条件 | 用户可在我的信息界面查看个人信息。 |
| 基本路径 | 1. 系统进入我的信息界面。 2. 用户可在我的信息界面查看已经录入的个人信息。 |

个人信息修改功能用例表如表3.6所示

表3.6个人信息修改用例表

|  |  |
| --- | --- |
| 用例名称 | 个人信息查看 |
| 用例描述 | 查看已经录入的个人信息。 |
| 参与者 | 用户 |
| 前置条件 | 用户已凭借系统注册的账号登录进入系统。 |
| 后置条件 | 用户可在我的信息界面查看已经修改后的个人信息 |
| 基本路径 | 1. 系统进入我的信息界面。 2. 用户输入修改后的个人信息。 3. 系统验证修改后的个人信息。 4. 用户可在我的信息界面查看已经修改后的的个人信息。 |
| 拓展点 | 用户输入修改后的个人信息不合法，系统将显示对应的错误信息。  系统验证修改后的个人信息不合法，系统将显示对应的错误信息。 |

证据管理模块用例表如表3.7所示

表3.7证据管理模块用例表

|  |  |
| --- | --- |
| 用例名称 | 证据管理 |
| 用例描述 | 用户可通过该用例，对其所拥有的证据进行创建，删除，修改，查询，加密，权限管理操作。 |
| 参与者 | 用户 |
| 前置条件 | 用户已凭借系统注册的账号登录进入系统。  我的证据界面将对用户展现该用户创建的所有证据的列表，每个列表项的最右侧为操作列，操作列包含修改，删除，下载按钮。并在列表右上角展示可创建证据的按钮。 |
| 后置条件 | 用户可在我的证据界面对证据进行操作。 |
| 基本路径 | 1. 进入我的证据界面。 2. 用户可点击创建按钮并输入证据类型，证据描述，证据是否私有，证据加密，证据加密方式，证据文件上传以创建证据，可在我的证据界面查看已经创建后的证据。 3. 用户可点击列表项的操作列的删除按钮，用户鉴权后，可在我的证据界面查看到要删除的证据的状态为已删除。 4. 用户可点击列表项的操作列的修改按钮，用户鉴权后，用户可在对应列表项进行证据信息修改。用户点击确认修改按钮后，可在我的证据界面看到修改后的证据信息。 5. 用户可点击列表项的操作列的下载按钮，用户鉴权后，即可下载该证据文件 |
| 拓展点 | 用户输入的信息不合法，系统将显示对应的错误信息。  用户管理证据失败，系统将显示对应的错误信息 |

证据共享模块用例表如表3.8所示

表3.8证据分享模块用例表

|  |  |
| --- | --- |
| 用例名称 | 证据分享 |
| 用例描述 | 用户可通过该用例，在证据库界面中申请其他用户公开的证据。也可在他人申请界面中处理他人的申请 |
| 参与者 | 用户 |
| 前置条件 | 用户已凭借系统注册的账号登录进入系统。  证据库界面将显示所有用户公开的证据。  我的申请界面将显示我对其他用户的公开证据的申请。  他人申请界面将显示其他用户对登录用户所拥有证据的申请。 |
| 基本路径 | 进入证据库界面，用户点击目标证据对应的列表项的操作列中的申请按钮发起证据的下载或信息查看申请。用户随后可在我的申请界面查看我的申请目前的进度。处理者可在他人申请界面处理他人对处理者的公开证据的申请。  若处理者同意该申请，证据库将开放申请者对该证据进行下载的权限。  若处理者拒绝，申请者将不具备权限 |

3.3 非功能性需求分析

一个完善的系统除了功能性需求之外，还需要一些基本的非功能性需求，包括观感需求，易用性需求，安全性需求。

1. 观感需求

前端界面的设计简洁明了，风格统一，减少不必要的元素，方便用户使用。用户需要进行功能性操作时，能够快速定位至功能入口，用户需要数据录入时，需要对输入进行规范。同时还要考虑浏览器适配问题，保证各个浏览器能够成功运行

1. 易用性需求

系统的所有功能性操作的流程要符合用户习惯以及使用的期望，且操作简单不复杂，这将极大提高用户对该系统的接受度，提高转化率。

1. 安全性需求

数据不能被授权用户以外的任何人访问，授权用户可以不受阻止的访问数据，以及产品的强壮度必须得到保证。

3.4 可行性分析

3.4.1 技术可行性

在以联盟链为基础的数据加密存证系统能否相较于传统数据加密系统更安全更可靠这一层面考虑，针对数据存证系统，公有链相对复杂且难以保证数据隐私性。有文献证明基于区块链的数据库能够做到相对保密，能够更好的保障系统安全和数据隐私性[14][15]，联盟链作为区块链的一种，对于用户有一定限制，但仍然有区块链的特点，更为符合数据存证系统的要求。

软件实现方面，各架构层逻辑清晰，实现技术皆采用开源，社区较为繁荣，已被时间、业界证明为稳定、成熟的。

在联盟链实现方面，业界已有成熟的技术，且经过实践与市场的沉淀，已被证实为可靠的，稳定的。

3.4.2 经济可行性

主要开发IDE为VSCODE以及免费linux系统Ubuntu与开源联盟链框架HyperledgerFabric、开源Web服务器Node框架Express与开源前端构建框架UmiJS，因此经济损耗小，对CPU等硬件要求也不大。经济可行性高。

3.5 本章小结

本章主要对于基于联盟链的电子存证管理系统进行了初步的系统分析。

首先通过系统用例图阐述了常规用户与管理者与电子存证加密管理系统之间的关系。并依此对系统的功能性需求进行了合理分析，并对各项功能性需求进行了较为详细的说明，理清了开发的侧重点。在非功能性需求分析中，感官需求需要前端界面设计要简洁，设计风格统一，减少不必要元素。易用性需求要求系统所有的操作流程符合用户的操作习惯。安全性需求要求用户只能访问到自己权限以内的数据。

通过本章的分析，发现该系统开发流程长，设计技术较为繁复。由于实现的功能较为固定，所以也应重视其他非共功能性需求，例如运行效率，界面美观等。

4 系统设计

* 1. 系统总体设计

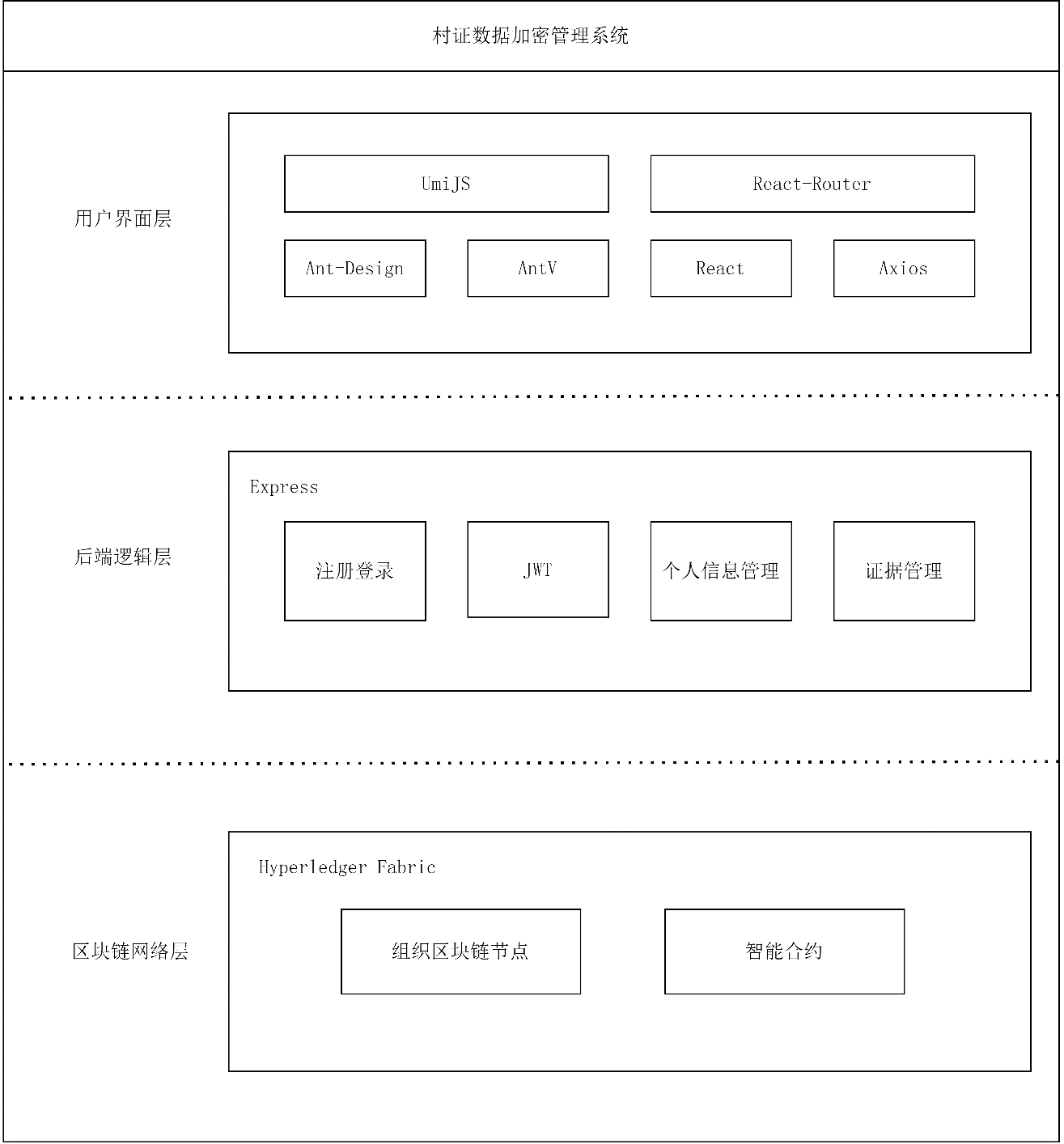
系统架构如图4.1所示

图4.1系统架构图

通过对基于联盟链的存证数据加密管理系统的整体分析，运用分层思想将整个系统分为区块链网络层，业务逻辑层与用户界面层。

区块链网络层主要利用联盟链的不可篡改等特点来对整个系统的数据进行存储，提高系统的安全性。系统采用Hyperledger Fabric构建区块链网络层，采用状态数据库LevelDB存储数据。主要包含组织区块链节点与智能合约两大部分工作。

业务逻辑层需要与区块链网络层进行交互，并且需要对用户界面层发起的HTTP请求进行解析并响应，是数据与业务逻辑的结合处，是该系统运转的核心。主要包含注册登录，JWT，个人信息管理，证据管理等功能。

用户界面层为供用户进行操作的PC端图形界面，依赖于业务逻辑层所提供的对数据进行访问的能力，给用户展示相关的数据与对这些数据进行操作的入口。

* 1. 区块链网络层设计

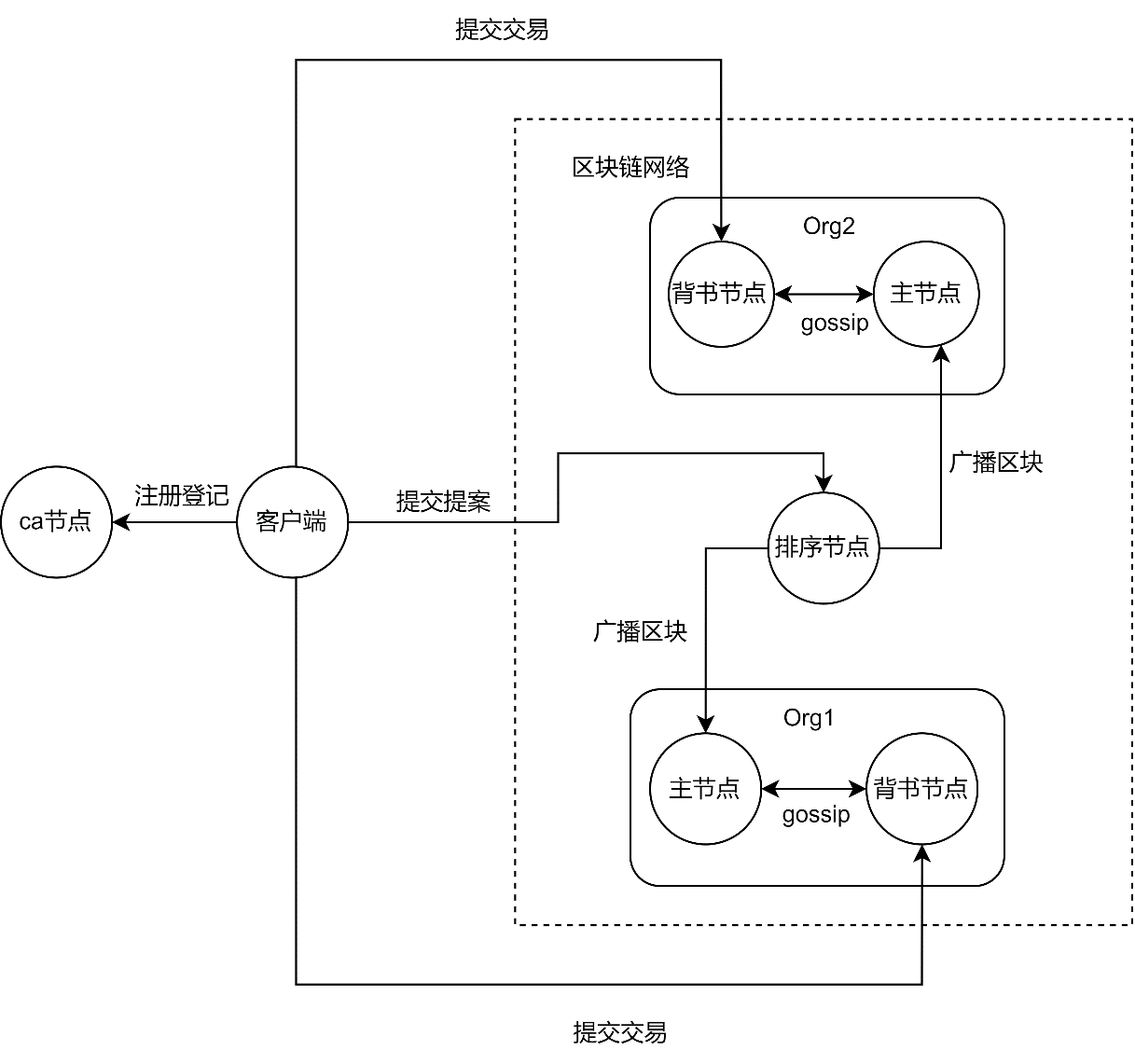
区块链网络层架构如图4.2所示

图4.2区块链网络层架构

区块链网络层的设计主要在于网络结构，帐本与智能合约。

网络结构为区块链网络的拓扑结构，采用1个ca节点，1个orderer节点，2个组织，每个组织含有1个背书节点与1个peer节点的网络配置。

账本与智能合约可抽象为数据与一系列的可更新与查阅该数据的函数。根据该系统的功能模块的设计，账本将存储元数据，用户，证据，证据申请四个对象。元数据存储证据对象的总数以及证据申请对象的总数，用于生成新的证据对象的与证据申请对象的id。用户对象是对登录模块，个人信息模块中需要的用户数据进行数据抽象的结果。证据对象是对证据管理模块，证据权限模块的证据这一客体进行数据抽象的结果。证据申请对象是对证据分享模块的用户间进行证据共享的过程进行数据抽象的结果。智能合约则是对这些数据对象进行增删改查的操作。

* 1. 业务逻辑层设计

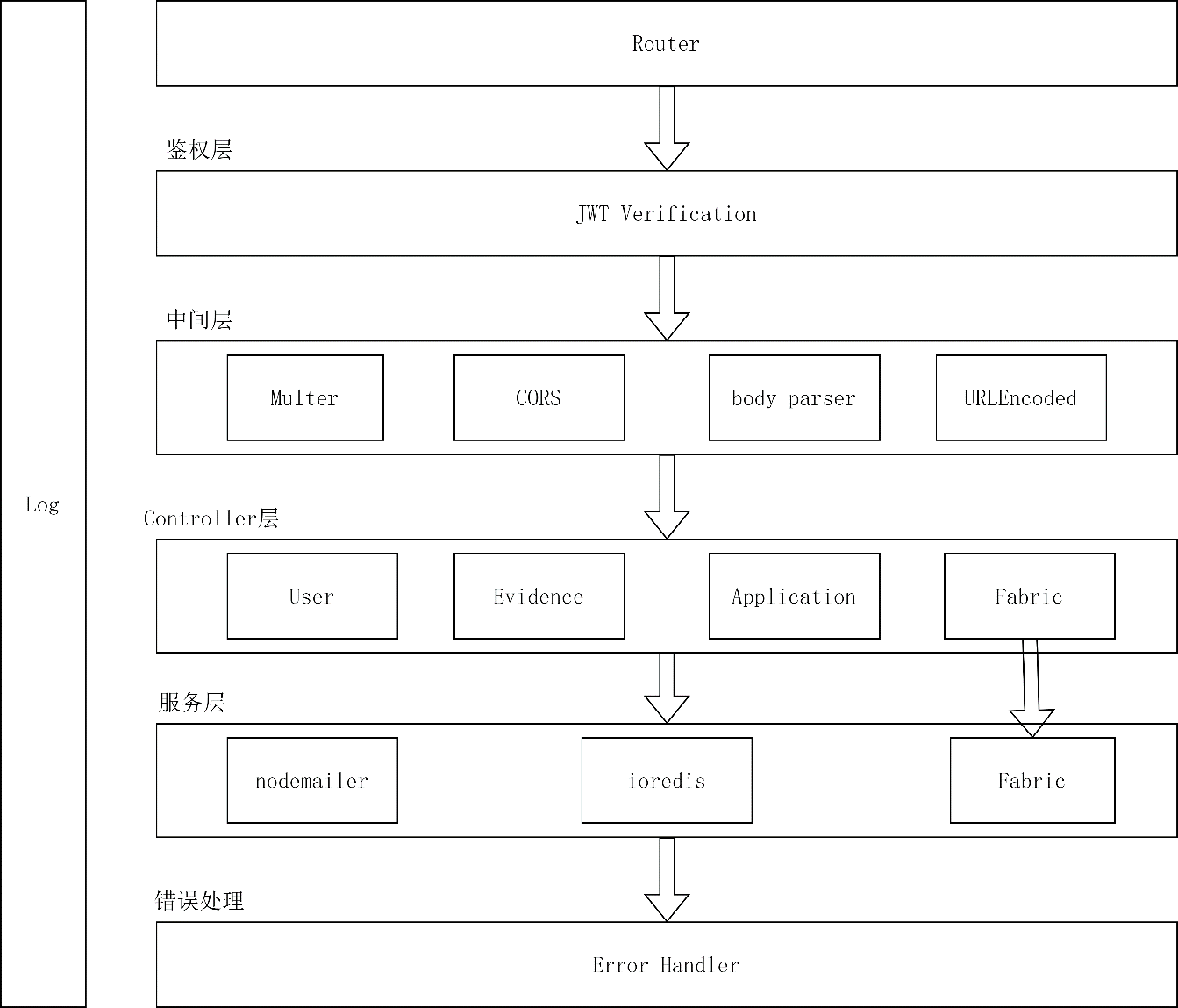
业务逻辑层架构如图4.3所示

图4.3业务逻辑层架构

业务逻辑层需要区块链网络层进行数据交互，并且提供给用户对数据进行访问的能力。将业务逻辑层架构为为鉴权层，中间层，Controller层，服务层以及错误处理。

鉴权层是业务逻辑层一切接口的基石，所有接口都需要经过鉴权层的鉴权，只有鉴权成功的接口才能返回正常的响应，否则将返回状态码为304的异常响应。鉴权方法为JWT[28]，每个接口的HTTP请求的请求头都将携带属性为authorization，值为登录或注册成功后，后端响应给前端的token。

中间层是Express为满足各种需求，响应在到达Controller层真正进行业务处理之前，对HTTP请求进行预处理的业务逻辑层层次。Multer在请求体为formData时，自动获取请求体formData中的文件数据。CORS提供系统跨域服务。body pauser可解析请求体，最终将请求体用对象来表示。urlencoded可处理HTTP请求在路由中的参数，最终将参数用对象来表示。

最终将Controller层抽象为FabricController，UserController，EvidenceController，ApplicationController。

FabricController负责与区块链网络层进行数据交互，并将交互后的结果进行解析。因网络IO时间占比较大，故对数据进行缓存以提高系统响应速度，缩短响应时间。是整个业务逻辑层访问数据的基础，其他controller皆依赖于FabricController。

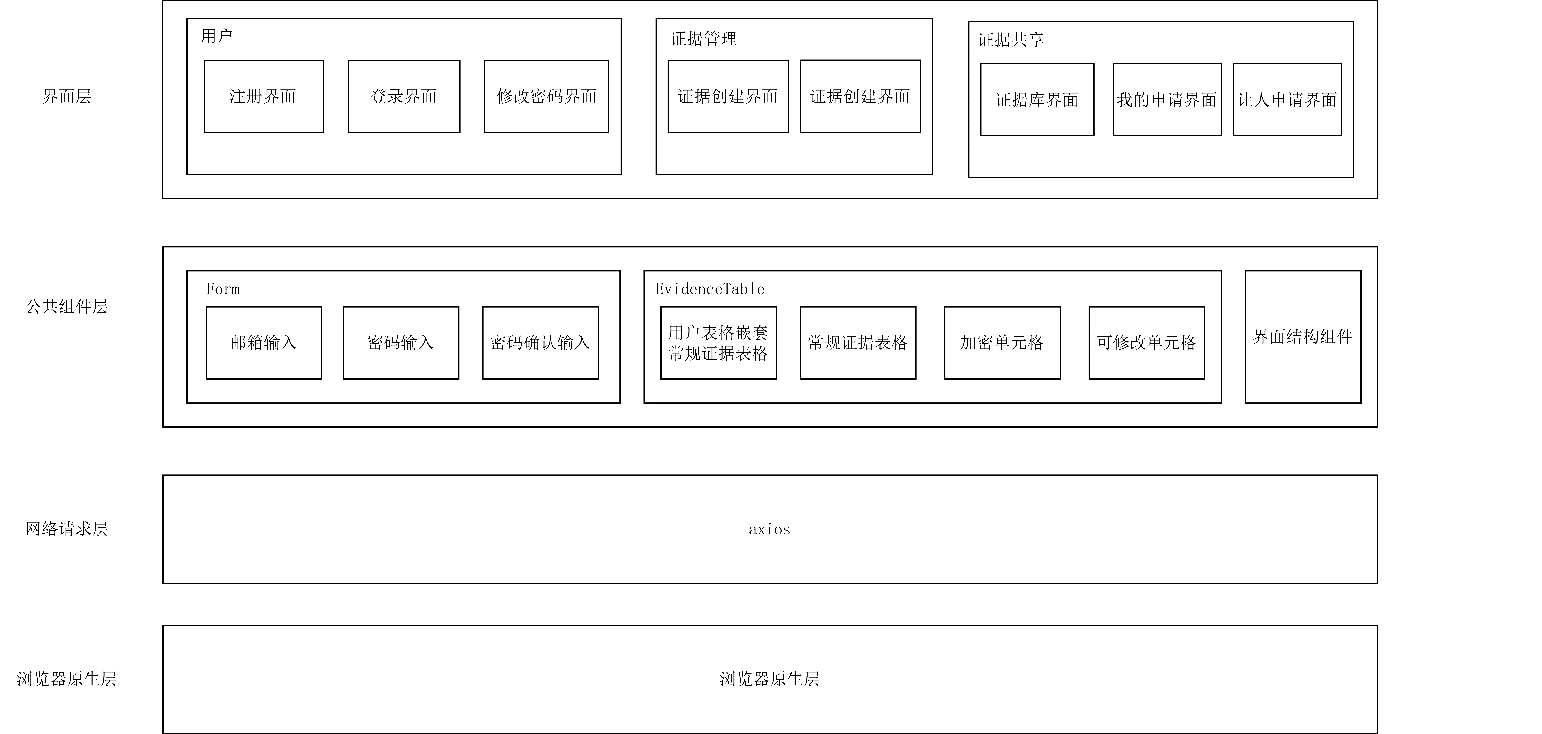
UserController负责管理与用户相关的业务，如用户的注册，登录，修改密码，查看与修改个人信息等。

EvidenceController负责管理与证据相关的业务，其中包括对证据的增删改查，证据的加密解密。

ApplicationController负责管理证据的共享业务。ApplicationController主要是对区块链网络层的账本中的证据申请对象进行处理，而EvidenceController是对账本中的证据对象进行处理，两者所处理的数据不同。

* 1. 用户界面层设计

用户界面层架构如图4.4所示

图4.4用户界面层架构图

对于一个系统，用户界面层是用户唯一可感知的系统分层，是系统与用户交互的桥梁，一个系统的好坏除了性能指标外，是否操作友好也是关键评判标准。界面简洁，设计风格统一是用户界面层的目标。致力于通过用心设计的界面，尽力简化的操作流程，能够使用户能拥有良好的使用体验，轻松地完成操作。

用户界面层整体架构如图4.4所示。用户界面层由界面层，公共组件层，网络请求层以及浏览器原生层。

界面层使用UmiJs实现与用户相关的注册界面，登录界面与忘记密码界面，分别可以使用系统的注册，登录与忘记密码功能。接着是与证据管理相关的证据创建界面与我的证据界面，证据创建界面可以进行证据的创建，我的证据界面可以对证据进行删除，修改，下载，也可以加密证据信息或证据文件。与证据共享相关的界面为证据库界面，我的申请界面以及他人申请界面。证据库界面内可以看到其他用户所有的公有证据，可以对这些证据申请查看证据信息或申请证据下载。我的申请界面可以看到我对其他证据发出的申请的进度。他人申请界面则是处理其他人对我的证据发出的申请，可以选择同意绝。个人信息界面用于展示个人信息，并可用于修改个人信息。

用户在注册登录成功后应先进入系统主界面，即控制台界面，在控制台界面，用户可以查看个人信息，我的证据，证据库，证据共享卡片。可通过点击卡片跳转至对应功能的界面。

公共组件层则是界面层可复用的能力，包含逻辑复用组件，样式复用组件，格式化函数等。将多个界面需要的能力进行分析并解耦后，发现有多个界面展示能力或数据输入能力是多个界面同时需要的，故将这种多个界面都需要的通用能力进行抽离、解耦，使其在多个界面中可重复使用。

其中的逻辑复用组件包括表单组件与证据表格组件。表单组件中的邮箱输入框，密码输入框以及确认密码输入框，这些组件对输入的数据进行特定的有效性检查，当有效性检查未通过，将无法提交表单。证据表格组件将按特定的顺序显示证据信息，并对加密后的数据进行提示。并在修改时，可将表格的单元格变化为对应数据的输入框。

样式复用组件则是将除了注册，登录，忘记密码界面的其他界面的页面结构进行统一分析后又经过化归所得到的组件，避免每个界面都将设置同样的样式，书写重复的代码。

网络请求层曾主要使用axios发起HTTP请求来实现数据的通信。axios是使用Promise封装XMLHttpRequest的一个JavaScript库。并支持响应拦截，请求拦截，请求取消，错误处理等功能。该系统使用了上述的响应拦截，请求拦截，错误处理功能。使用响应拦截与请求拦截是为了搭配JWT进行用户鉴权。拦截响应查看响应的状态码是否为304，如果为304说明鉴权失败，则无条件跳转到登录界面以重新鉴权。拦截请求目的是为了在发起的所有请求中都自动的在请求头中携带Authorization属性进行鉴权。如果不使用这个能力的话，将在所有请求发起时手动携带，这将过于繁琐，且不利于系统的拓展性。错误处理功能则提供了对约定好的错误码进行处理的入口。

浏览器原生层则是用户界面层使用的浏览器的原生能力，其中主要客户端存储能力与编码URL中特殊字符的能力。客户端存储能力localStorage则是用户界面层存储JWT中的token所需要的能力。端存储能力包含两个原生API，包含localStorage与sessionStorage。localStorage存储的数据可以长期保留，而sessionStorage存储的数据当页面关闭时会自动清除。根据上述的比较，localStorage更适合存储用于鉴权的token，否则每次页面关闭都将重新申请，对后端资源造成浪费。

4.5 功能模块设计

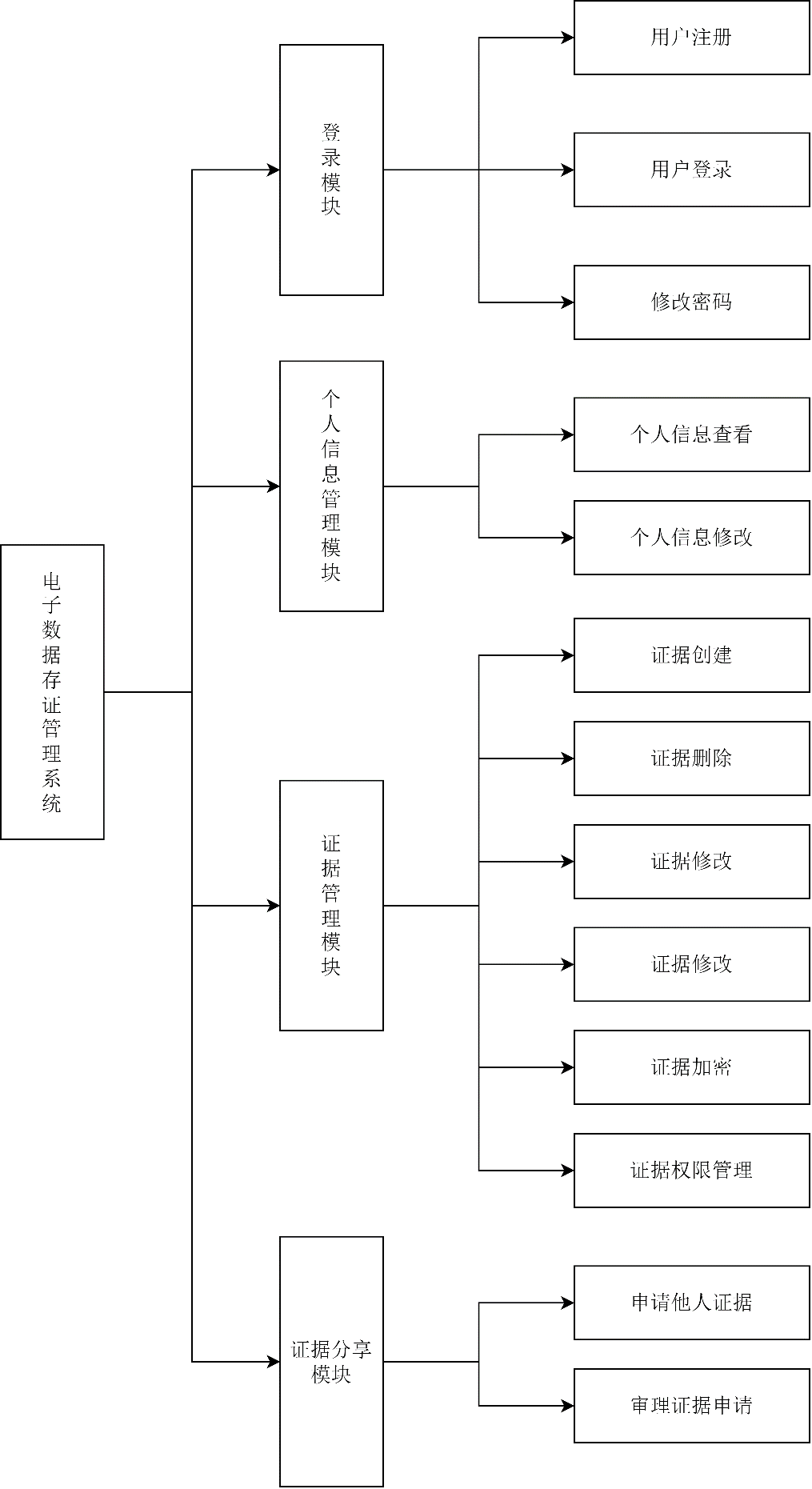
系统功能模块设计如图4.2所示

图4.2 系统功能模块设计图

电子数据存证加密管理系统的功能模块可划分为登陆模块、个人信息管理模块、证据管理模块以及证据共享模块。登录模块主要含有注册、登录、修改密码功能。个人信息管理主要含有个人信息管理以及个人信息修改功能。证据管理模块主要含有证据创建、证据删除、证据修改、证据加密以及证据权限管理功能。证据共享模块主要含有申请他人证据以及处理证据申请功能。

4.5.1 登录模块设计

该系统的登录模块主要包含以下功能：

1. 用户注册：用户初次使用该系统时，需要注册账号用来登录。
2. 用户登录：用户注册后，可凭借该账号登录进入该系统以使用相关功能。
3. 修改密码：当用户登录后可修改密码，修改密码成功后会跳转到登录页面，此时用户需要重新登录才可继续使用该系统。当用户忘记密码时，可通过验证邮箱方式修改密码。

用户登录数据结构如表4.1所示

表4.1用户登录数据结构表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 字段描述 |
| id | string | 账号id |
| password | string | 账号密码 |
| identity | number | 账号身份  0代表管理员  1代表常规用户 |

4.5.2 个人信息管理模块设计

该系统的个人信息管理模块主要包含以下功能：

1. 个人信息查看：用户登录后可以查看自己账号的详细信息，其中包括自己创建的总证据数、各种证据类型数量，与总证据数的比例等。
2. 个人信息修改：用户可以选择修改自己账户的部分信息，其中包括性别，生日，身份证号等。但不包括账号创建时间。

用户信息数据结构如表4.2所示

表4.2用户信息数据结构表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 字段描述 |
| id | string | 账号id |
| createTime | number | 创建时间 |
| updateTime | number | 更新时间 |
| idCardNumber | string | 身份证号 |
| name | string | 用户姓名 |
| gender | number | 用户性别 |

4.5.3 证据管理模块设计

该系统的证据管理模块主要包含以下功能：

1. 证据创建：用户登录后可进入证据创建页面完成对应表单创建对应证据的数据抽象。其中表单包括对证据类型的选择，例如图片，文档等、证据权限的设置、证据文件的上传等。
2. 证据查看：用户登录后可在个人证据界面查看到自己创建的所有证据的详细信息以及对证据进行操作的入口。证据查看功能也包括了证据的下载。
3. 证据删除： 用户登录后可在个人证据界面查看到自己创建的所有证据的详细信息以及证据删除按钮。点击后，需要进行密码验证，验证成功则将在数据库中将该证据的isDelete字段设为1，用户在7天内可选择恢复。若7天后用户不选择恢复的话，证据将自动从数据库中删除。
4. 证据修改：用户登录后可在个人证据界面查看到自己创建的所有证据的详细信息以及证据修改按钮。点击后，即可在证据对应信息展示处修改。确认修改后需要进行密码验证，验证成功将在数据库修改该证据信息。
5. 证据加密管理：用户在创建证据时可选择该证据文件的加密方式，也可选择不加密。证据创建成功后在个人证据界面将显示证据所有字段的明文数据。可在个人证据界面批量选择证据字段的加密方式，确认选择字段的加密方式后，需要进行密码验证，验证成功则将自己的所有证据对应字段的值统一加密。

证据权限管理：证据在创建时可选择将该证据公有或私有。若选择将该证据设置为共有证据，其他所有用户将在证据库界面查看到该证据经过该证据的创建者选择性加密后的信息。若选择将该证据设置为私有证据，只有自己在个人证据界面查看到该证据。

证据数据结构如表4.3所示

表4.3证据数据结构表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 字段描述 |
| id | number | 证据id |
| creatorId | string | 创建者的用户id |
| description | string | 证据描述 |
| type | string | 证据类型 |
| name | string | 证据文件名称 |
| extension | string | 证据文件拓展名 |
| size | number | 证据文件内存大小 |
| encryption | string | 证据文件加密方式 |
| createTime | number | 创建时间 |
| updateTime | number | 更新时间 |
| isDelete | number | 是否已删除  1表示已删除  0表示未删除 |
| isPrivate | number | 是否私有  1表示私有  0表示公开 |
| access | map<userId,string[]> | 用户权限字典  键：用户id  值：用户所拥有的权限 |

4.5.4 证据共享模块设计

该系统的证据共享模块主要包含以下功能：

1. 证据申请：用户可在证据库界面申请查看目标证据目标字段的权限或申请下载目标证据的权限。申请成功后等待目标证据的创建者处理该申请，目标证据的申请者同意后，将获取相对应权限。
2. 处理申请：在他人申请界面可处理其他用户对自己证据相关权限的申请。可选择同意或拒绝，若选择同意申请则需要填写该证据对于申请者的有效时间。

证据申请数据结构如表4.4所示

表4.3证据申请数据结构表

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 字段描述 |
| id | number | 证据申请的id |
| evidenceId | number | 证据id |
| done | number | 流程是否已完成  1表示已完成  0表示待处理 |
| code | number | 表示申请是否通过  1表示已通过申请  0表示该申请未通过 |
| expire | number | 若申请已通过，表示  该权限的过期时间 |
| type | string | 申请的类型  “download”表示申请下载  其他值为证据字段名称 |
| applicantId | string | 申请人的id |
| transactorId | string | 处理人的id |
| createTime | number | 创建时间 |
| endTime | number | 处理人处理该申请的时间 |

4.6 本章小结

本章首先根据分层思想将系统划分为区块链网络层，业务逻辑层，用户界面层，并依次阐述了对应分层的设计，分层思想将系统解耦，有助于降低开发的复杂度。接着对该系统进行功能模块划分，将该系统划分为登录模块，用户信息管理模块，证据管理模块与证据共享模块，并依次阐述了对应功能模块的所需要实现的具体功能以及实现该功能需要的账本中的数据结构。常用的分治，分层方法可以有效地降低系统耦合度，复杂度、减轻工作量、节省人力，同时也能提高工作效率，为以后的测试，维护，拓展打好根基。

5 系统实现

5.1 智能合约的实现

智能合约主要实现对账本数据中的个人信息，证据，证据申请的初始化以及创建，删除，修改，查询。以及一些工具函数，比如isAuthorized判断该用户是否有权限使用该系统，encrypt对数据进行加密，getId当新建证据或证据申请时，获取新证据的id。

智能合约涉及的主要函数如表5.1所示

表5.1智能合约函数表

|  |  |
| --- | --- |
| 函数名 | 函数描述 |
| initLedger | 初始化账本 |
| createUser | 创建用户 |
| updateUser | 更新用户信息 |
| getUser | 查询用户信息 |
| createEvidence | 创建证据 |
| deleteEvidence | 删除证据 |
| updateEvidence | 更新证据信息 |
| getEvidence | 查询证据信息 |
| getEvidencesOfUser | 查询用户的所有证据 |
| getAllEvidences | 获取所有用户的公有证据 |
| encryptEvidences | 加密证据的相关信息 |
| decryptEvidencesField | 解密证据已加密字段值 |
| createApplicaton | 创建对证据的申请 |
| shouldCreateApplication | 判断当前对某证据的申请是否通过  不通过的情况有已申请，处理者未批准或  申请通过，权限未过期 |
| updateApplication | 当处理者处理后用于更新证据申请的信息 |
| getApplicantsApplications | 获取所有申请者id为参数的证据申请 |
| getTransactorsApplications | 获取所有处理者id为参数的证据申请 |
| isAuthorized | 判断某用户是否有权限使用该系统 |
| encrypt | 对证据文件或对证据信息进行加密 |

5.2 登录模块的实现

登录模块是该系统的基石，系统一切功能都将基于注册或登陆后前端存储的鉴权数据token，其他接口都将使用该token进行身份认证，目的是为了鉴别当前用户是否有该接口的权限。登录模块主要含有注册，登录，修改密码三个功能。主要需要实现用户界面层的注册，登录，修改密码界面，业务逻辑层的UserController以及区块链网络层账本中的user数据与智能合约对账本中user数据的处理。

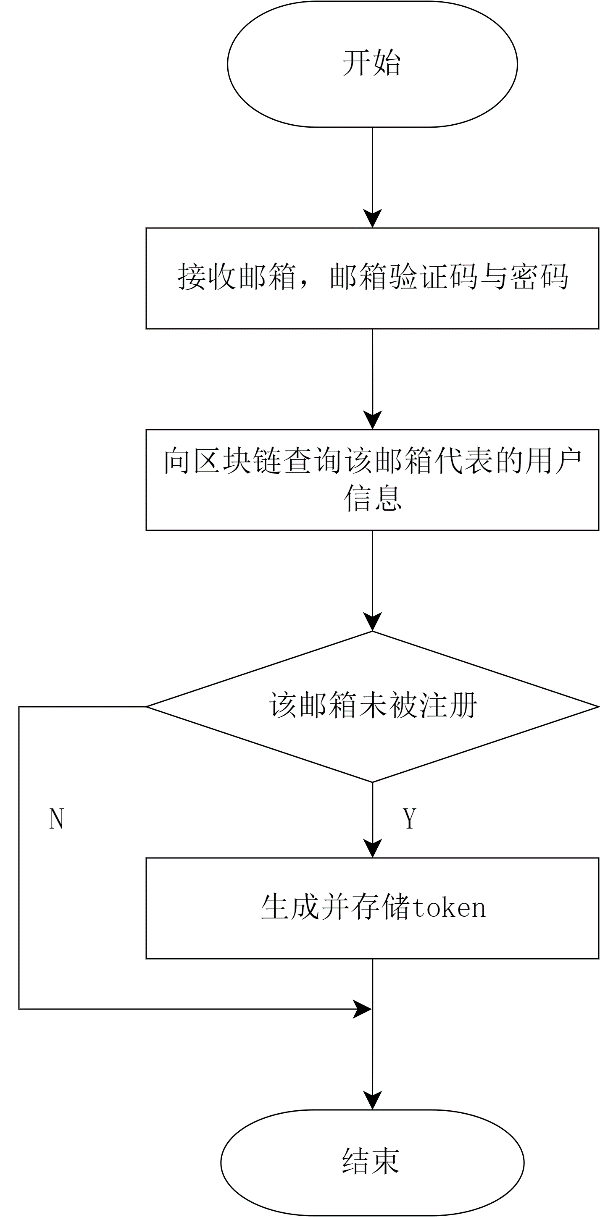
该系统的用户鉴权则是采用JWT技术，当注册或登陆成功后，后端会通过用户的账号等信息来生成token，并返回给前端存储。前端随后向后端发起的任意HTTP请求都将携带该token作为请求头的authorization属性。后端收到前端发起的除了注册与登录的任意HTTP请求也都会先判断该token是否有效，如果该token无效，则会返回给前端设定好的状态码，该系统选用304，并附带token已过期的提示信息。反之，则会正常处理该请求。前端也将会对所有的HTTP响应做出拦截，当检测到状态码为304，即为鉴权失败时，将无条件跳转到登录界面，需要用户重新登录以获取新的token。

图5.1注册流程图

当用户初次使用该系统时需要注册，注册需要先对邮箱进行验证。验证方式为后端向用户在注册表单内填入的邮箱发送验证码，用户需要在注册表单内填入收到的验证码以及登录用的密码。向后端发送表单后，后端需要确认该邮箱未注册，确认该邮箱未注册后则注册成功，否则注册失败。该系统的注册流程图如图5.1所示。

用户注册成功后，可凭借注册成功的邮箱密码或者验证邮箱的方式进行登录。

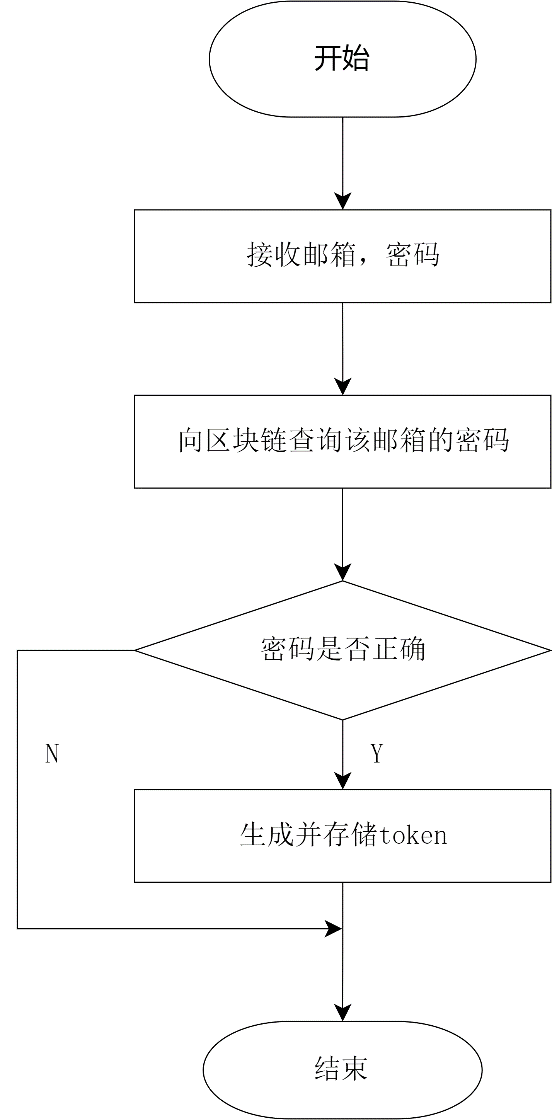
该系统的登录流程图如图5.2所示。

图5.2登录流程图

用户在登录后可以选择修改密码，修改密码首先需要验证邮箱，验证成功后输入两次相同的新密码后提交修改，后端验证两次输入的新密码相同后invoke智能合约的updateUser函数修改用户密码。

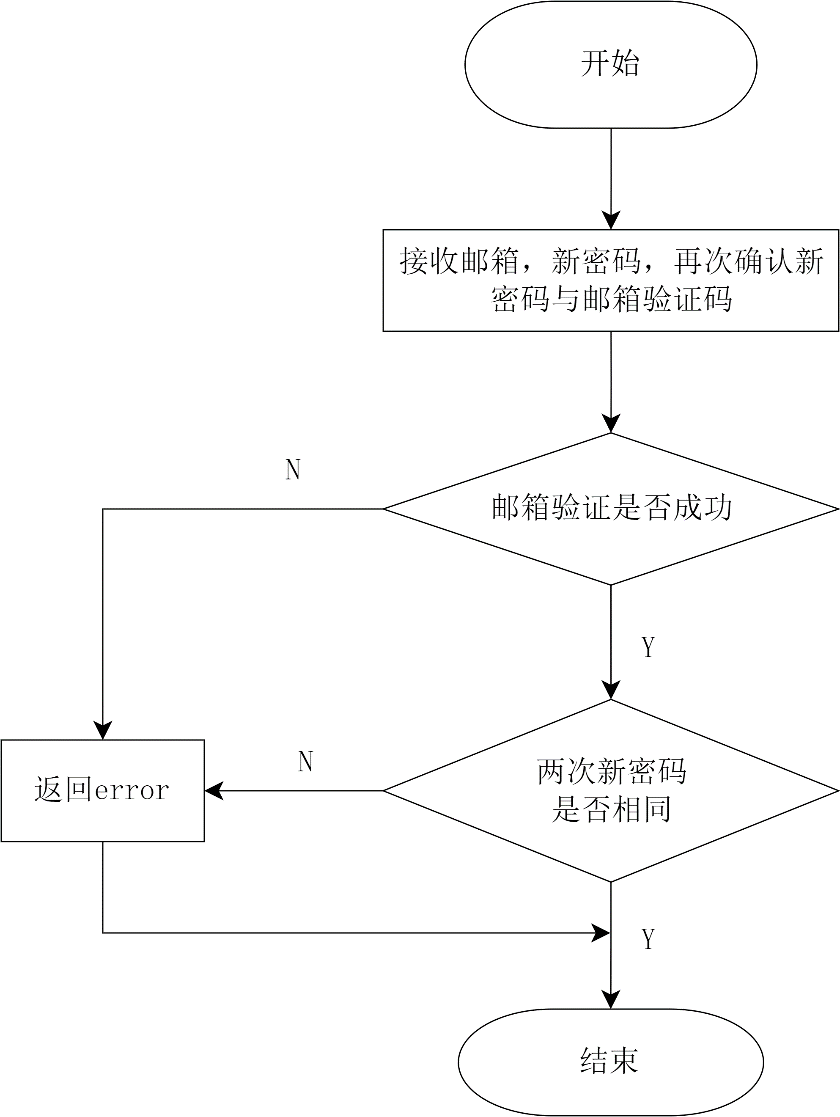
该系统的修改密码流程图如图5.3所示。

图5.3修改密码流程图

5.3 个人信息管理模块的实现

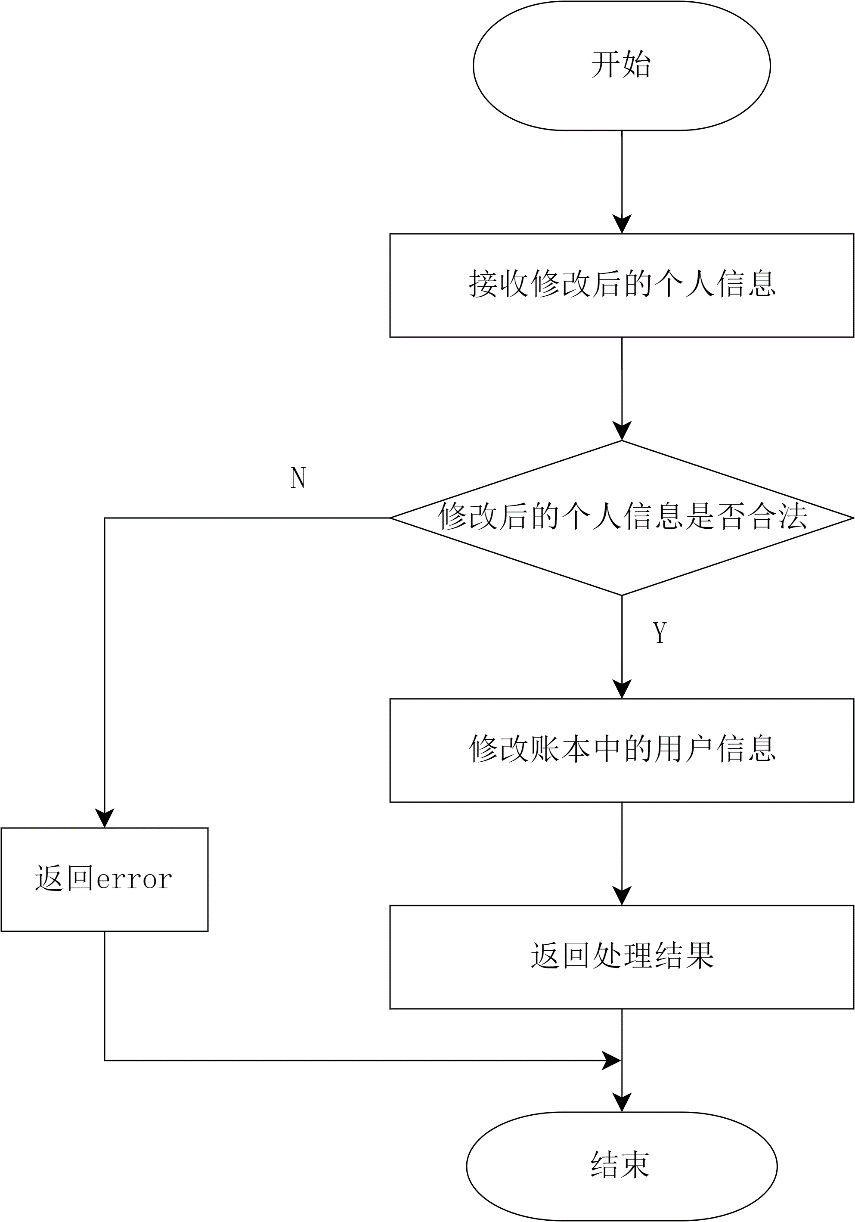
个人信息模块主要包含个人信息查看，个人信息修改这两个功能。主要需要实现用户界面层的操作台界面的个人信息模块，业务逻辑层的UserController与FabricController中关于修改用户信息的部分，区块链网络层智能合约中个人信息相关的部分。该模块只可对账本中的用户信息进行查看与修改，而证据管理模块与证据分享模块可对账本中的对应数据进行删除。

图5.4修改个人信息流程图

该系统的修改个人信息流程图如图5.4所示。用户需要在修改个人信息表单内填写修改后的个人信息。在填写表单数据时，前端会检测用户输入的数据是否合法，若不合法将不可提交表单并显示对应的错误信息。用户提交合法的表单数据至后端后，后端对应的Controller将对该表单数据及逆行再次检测。若检测成功，将向区块链发起修改个人信息的请求，区块链处理后将处理结果返回给后端，后端将该结果处理后再将后端处理的结果响应给前端，前端将显示修改个人信息成功与否。

5.4 证据管理模块的实现

证据管理模块是该系统的核心，该功能模块主要包含证据的创建，删除，修改，下载，加密等功能。主要需要实现用户界面层的证据创建界面与个人证据管理界面，业务逻辑层的EvidenceController与FabricController中证据相关的部分，区块链网络层中证据相关的部分。

证据的创建，删除，修改，查看功能的实现的逻辑流程均为前端发起功能对应的HTTP请求，后端查验token成功后调用EvidenceController对应的处理方法。在对应的处理方法中，invoke或query区块链智能合约中的函数，invoke方法会对账本进行修改，query则是对账本进行查询，在智能合约对应的函数中对账本中的数据进行创建，删除，修改或查询。区块链将结果返回给后端，后端根据其结果响应前端。

以创建证据功能为例，用户进入证据创建界面填写对应的表单。表单内容包含证据类型选择，证据描述，证据加密算法，证据是否私有，证据文件上传。证据类型选择限制了上传的文件的格式，证据是否私有则控制了该文件的基础权限。证据加密算法控制了该证据文件将被采取何种加密算法加密。若证据选择私有，则其他用户无法在证据库界面查询到该证据，若证据选择公开，则证据可被其他用户在证据库界面查询到并申请相关权限，这是为了能让用户有选择的将证据公开，而不是所有证据都只能公开。

当用户填写好表单后向后端发起HTTP请求，由于此次请求涉及文件上传，故使用formData数据格式作为HTTP的POST请求的请求体。后端接受到formData格式的请求体后将使用解析formData的中间件Multer将formData中的文件数据解析为Buffer，将其他非文件的key-value形式的数据解析为对象。将其中代表文件的数据Buffer根据文件格式转换为对应格式编码的字符串，例如证据为图片文件，会将buffer读取为base64编码格式的字符串，之后将对应格式编码的字符串按照用户选择的加密算法加密。将加密后的字符串写入文件，并将前端传入的其他非文件的数据处理后invoke智能合约中的createEvidences函数并传入相关参数。区块链处理后端发起的请求，随后将处理后的结果返回给后端，后端根据区块链层调用成功与否响应前端对应的提示信息。

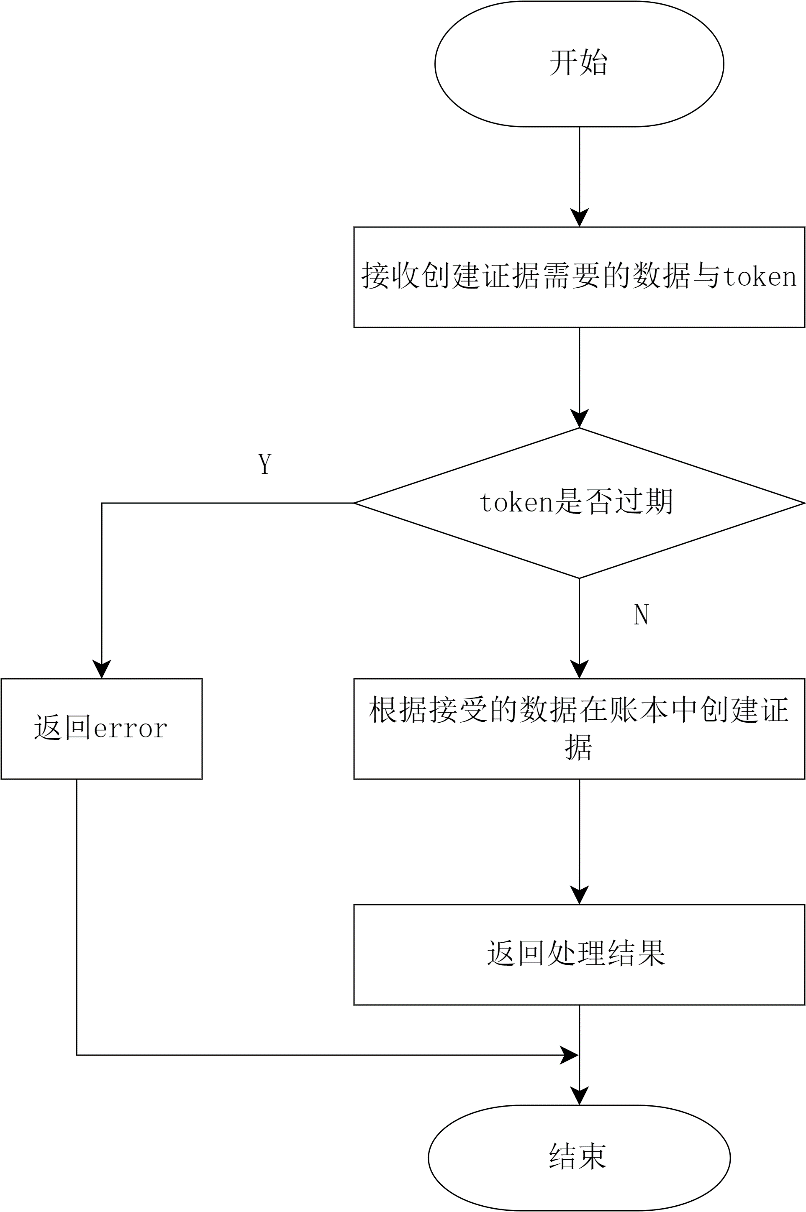
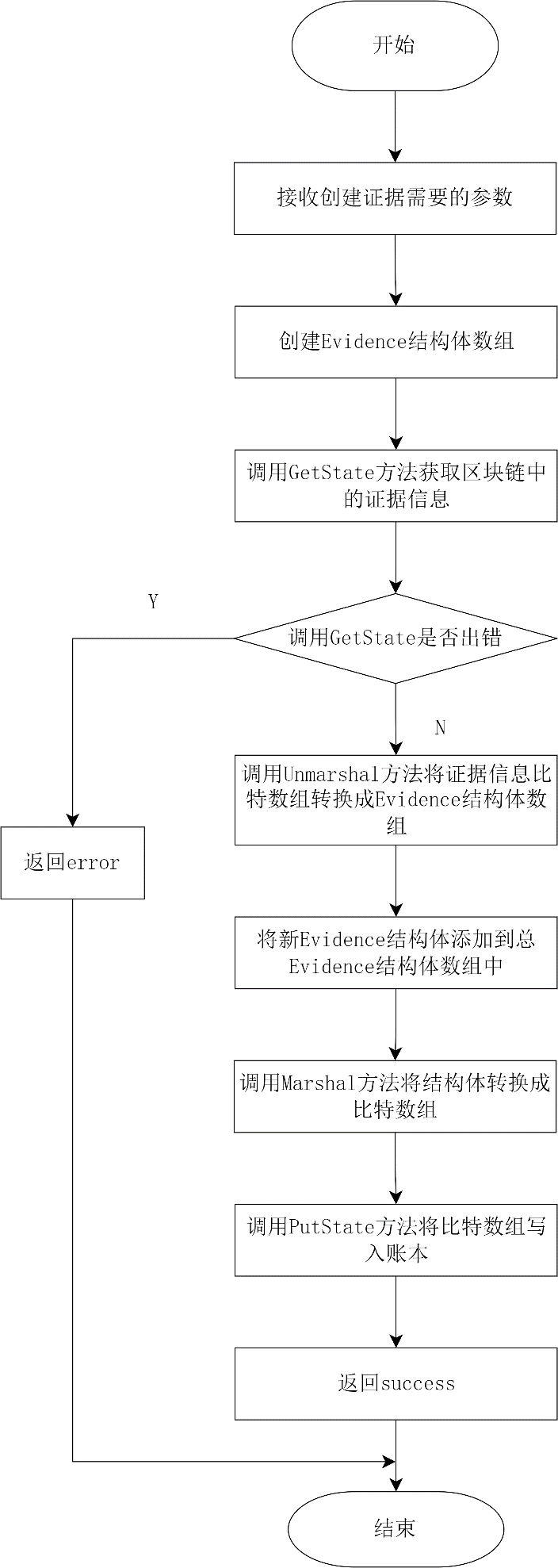
该系统的创建证据流程图如图5.5所示。

图5.5创建证据流程图

区块链收到后端发起的请求后，首先会执行链码中的Invoke函数，之后根据请求的参数去调用智能合约中对应的createEvidence函数，智能合约中的函数可通过第一个预设参数stub调用其核心方法GetState并传入比特数组获取账本数据，写入区块链账本则需要调用另一核心方法PutStatue以key-value形式写入账本，其value也是需要比特数组对象。Go语言内置包encoding/json提供了Marshal方法实现将比特数组转换为结构体，与其相对应的Unmarshal方法将结构体转化为比特数组。

智能合约创建证据流程图如图5.6所示

图5.6智能合约创建证据流程图

5.5 证据共享模块的实现

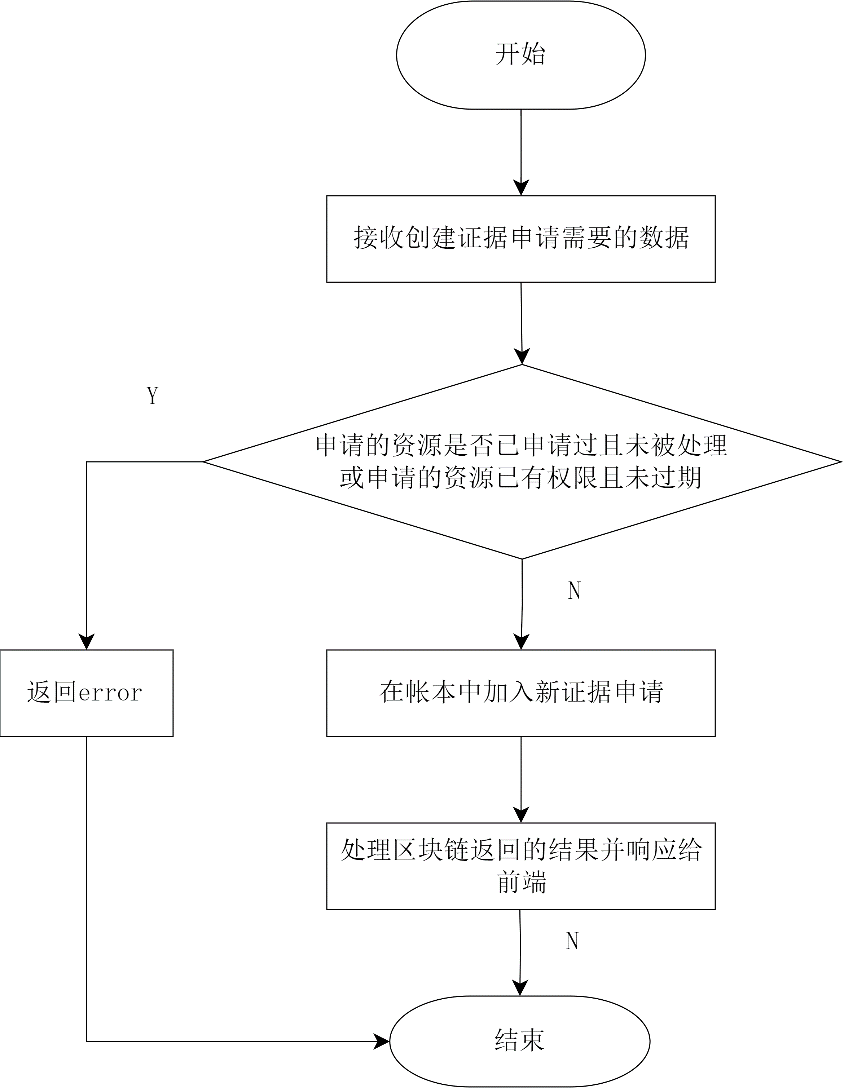
证据共享模块是为了用户间能够安全的进行证据信息共享。该功能模块主要包含向公开证据发出申请与处理他人申请两个功能。主要需要实现用户界面层的证据库界面，我的申请界面以及申请处理界面。证据库界面显示所有公开的证据，用户可在该页面向其他公开证据发出申请。我的申请界面显示我向其他人的公开证据发出的申请，可查看我发出的申请的进度，申请通过后，可在证据库界面与我的申请界面查看自己申请的信息。申请处理界面显示其他人向我的公开证据发出的申请，可选择同意或决绝。业务逻辑层则需要实现ApplicationController，其中包含对申请的创建，删除，处理等功能。区块链网络层则需要实现智能合约中与申请相关的函数，包含对账本中的证据申请进行创建，删除，处理等。

图5.7申请证据流程图

申请证据流程图如图5.7所示。申请证据功能流程如下，申请者发送创建证据申请表单，如果后端查询到被申请的证据是否已申请过且处理者未处理，或者申请者已经有被申请资源的权限且未过期。则申请失败。反之，后端对应的ApplicationController将会处理该请求，ApplicationController会向区块链发起Invoke createApplication请求，区块链收到请求后将结果响应给后端，后端处理区块链返回的结果并将处理后的结果响应给前端。

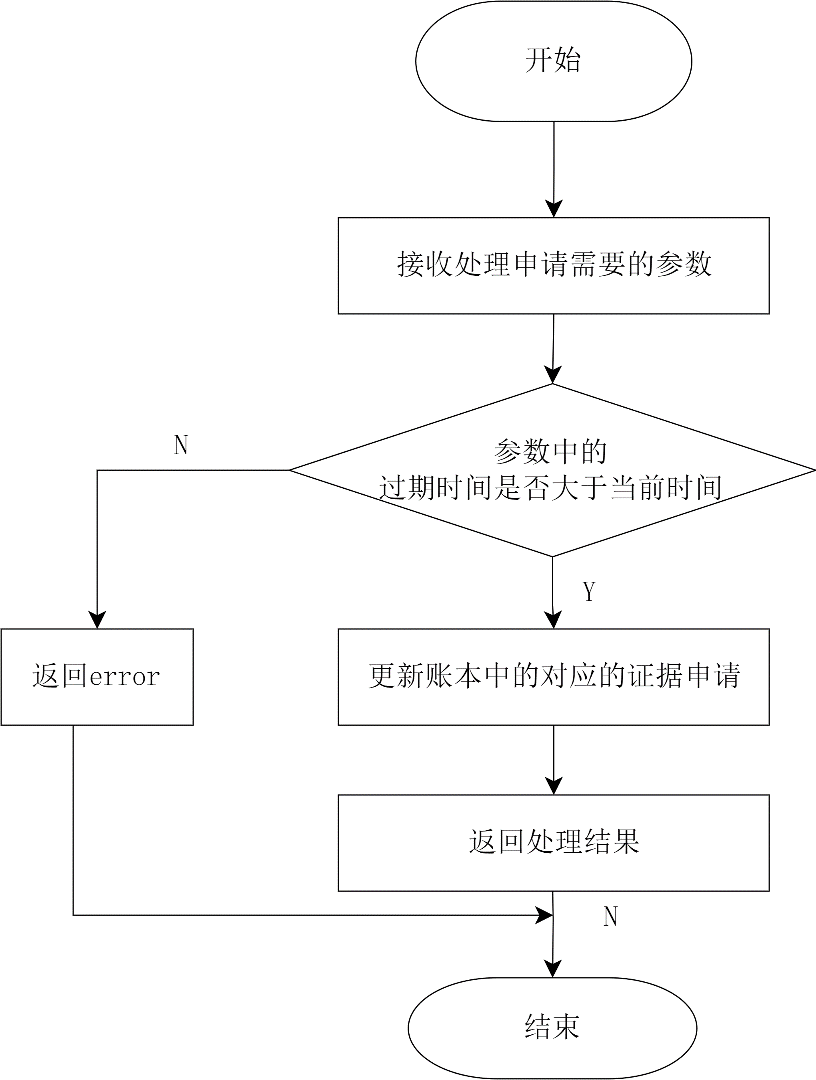


图5.8处理申请流程图

处理证据申请流程图如图5.8所示。处理证据申请流程如下，若同意申请则需要填写权限过期时间，前端会检验过期时间是否大于当前时间，若拒绝则不填写表单。前端发起处理申请的HTTP请求后，若发出同意申请的请求，后端将再次检验权限过期时间是否大于当前时间。若过期时间小于当前时间，将返回处理错误，并返回提示信息。反之，后端对应ApplicationController将会处理该请求，ApplicationController会向区块链发起Invoke updateApplication请求，区块链收到请求后将结果响应给后端，后端处理区块链返回的结果并将处理后的结果响应给前端。

5.6 本章小结

本章结合前两章的内容，详细介绍了智能合约中，登录模块，个人信息管理模块，证据管理模块，证据共享模块的流程与实现方式。第一节主要介绍了智能合约中的各个函数的名称与功能，第二节辅以流程图详细阐述了登录模块的注册，登录与忘记密码功能，第三节主要介绍了该模块的个人信息修改功能，第四节的证据管理模块是核心模块，以创建证据为例详细说明了从前端到后端，后端到区块链交互的全部流程。

6 系统测试

本章主要对该系统的测试环境，单元测试与系统测试进行说明，同时对用户界面进行了展示。

6.1 测试环境

操作系统：ubuntu 22.04.2 LTS

区块链网络：HyperLedger Fabric

开发语言：前端使用TypeScript，后端使用Node 16.19.0，智能合约使用Go 1.20

6.2 模块测试

6.2.1 登录模块测试

登录模块包含注册，登录，修改密码功能。测试用例如表6.1所示

表6.1登录模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试目的 | 测试内容 | 预期结果 | 测试结果 |
| A-1 | 用户能否注册 | 输入合法数据 | 新用户注册成功 | 通过 |
| A-2 | 用户能否注册 | 输入非法数据 | 注册失败并显示错误信息 | 通过 |
| A-3 | 用户能否登录 | 输入注册成功的账号 | 登录成功并进入主界面 | 通过 |
| A-4 | 用户能否修改密码 | 在修改密码界面内更改密码 | 修改密码成功并显示提示信息 | 通过 |
| A-5 | 用户能否使用新密码登录 | 输入新密码登录 | 登陆成功并进入主界面 | 通过 |

注册时用户需要输入合法的邮箱与两次相同的密码以及邮箱验证码，点击注册按钮完成注册。当输入的数据不合法或两次密码不相同或邮箱验证码错误，界面上方将显示具体信息。注册界面如图6.1所示

登陆时用户输入已注册的邮箱与密码点击登陆后即可进入系统主页。当输入的邮箱未注册或密码错误，界面上方将显示具体信息。登录界面如图6.2所示

6.2.2 个人信息管理模块测试

个人信息管理模块主要包含查看个人信息，修改个人信息功能。测试用例如表6.2所示。

表6.2个人信息管理模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试目的 | 测试内容 | 预期结果 | 测试结果 |
| B-1 | 用户能否查看个人信息 | 进入个人信息界面 | 显示个人信息 | 通过 |
| B-2 | 用户能否修改个人信息 | 输入合法数据 | 个人信息修改成功并显示提示信息 | 通过 |
| B-3 | 用户能否修改个人信息 | 输入非法数据 | 个人信息修改失败并显示错误信息 | 通过 |

图6.5查看个人信息界面

6.2.3 证据管理模块测试

证据管理模块主要包含证据的创建、删除、修改、查询、加密功能。测试用例如表6.3所示。

表6.3证据管理模块测试用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试目的 | 测试内容 | 预期结果 | 测试结果 |
| C-1 | 能否创建证据 | 输入合法文件 | 创建成功后跳转个人人证据界面 | 通过 |
| C-2 | 能否创建证据 | 输入非法文件 | 创建失败并显示错误信息 | 通过 |
| C-3 | 能否删除证据 | 点击删除 | 删除成功并更新证据列表 | 通过 |
| C-4 | 能否修改证据信息 | 输入合法数据 | 修改成功并更新信息 | 通过 |
| C-4 | 能否修改证据信息 | 输入非法数据 | 修改失败并显示错误信息 | 通过 |
| C-5 | 能否查看个人证据 | 进入个人证据界面 | 显示个人证据信息 | 通过 |
| C-6 | 能否加密证据信息 | 修改证据信息加密算法 | 修改后显示修改成功并更新证据信息 | 通过 |

6.2.5 证据共享模块测试

证据共享模块主要包含证据申请与处理申请功能。测试用例如表6.4所示

表6.4证据共享模块用例表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 测试目的 | 测试内容 | 预期结果 | 测试结果 |
| D-1 | 能否查看他人公开证据 | 进入证据库界面 | 显示所有用户公开证据 | 通过 |
| D-2 | 能否申请他人证据 | 点击申请下载或申请查看证据信息 | 点击申请后界面上面显示提示信息 | 通过 |
| D-3 | 能否查看我的申请的进度 | 进入我的申请界面 | 可看到我的申请的进度 | 通过 |
| D-4 | 能否处理他人申请 | 进入他人申请界面处理他人申请 | 点击同意后，申请人将获得权限。点击拒绝后，申请人获得提示信息 | 通过 |

证据库界面如图所示，可查看所有用户的公开证据，且含有申请按钮与解密按钮申请证据的权限

我的申请界面如图所示，可查看我的申请的进度

他人申请如图所示，可处理他人发起的申请

6.3 本章小结

测试是保障软件质量的一种有效的手段，本章详细介绍了各个模块的功能测试，对各个模块进行了用例测试与分析。分别对于前文提及的登录模块，个人信息管理模块，证据管理模块与证据分享模块进行了功能测试，经过不断的发现、分析、总结并解决问题，使系统越加完善。

7 结论

7.1 总结

本次课题提出了基于联盟链的电子存证加密管理系统，特点便是去中心化，难篡改，这些特点同样解决了传统存证管理系统的查询慢，安全性较低的痛点。分析了基于联盟链的电子存证加密管理系统的功能性需求，设计了存证管理系统应该具有的功能，接着分析了该系统的非功能性需求，要求系统的观感，易用性，安全性得到保障。分析后根据分层思想将整个系统分为区块链网络层，业务逻辑层与用户界面层。并按照其具体的功能划分为了登录模块，个人信息管理模块，证据管理模块，证据共享模块。模块实现后，将其整合为完善的系统。并对该系统的各个功能模块进行测试，将测试中出现的问题分析并解决。

具体工作如下：

1. 对国内外的电子存证管理系统进行了较为充分的调查，查阅了有关联盟链的资料，了解联盟链发展情况并学习了业界内成熟框架Hyperledger Fabric。设计了存证加密管理的方案，保障了证据的安全性，不可篡改。
2. 基于Hyperledger Fabric联盟链技术，完成了区块链网络的设计与搭建。使用Go语言编写智能合约，将其下载并实例化至背书节点，实现命令行提交交易。接着在后端运用SDK实现fabric服务层，实现后端调用SDK提交交易。
3. 设计并实现了基于联盟链的村正数据加密管理系统的各个功能模块。根据系统分析，结合Hyperledger Fabric、Express、UmiJS等技术对系统进行总体设计。在上述基础上实现该系统。
4. 完成系统之后对该系统进行了功能测试，对系统的各个模块完成测试，保障各个模块的运行结果与保持统一。

7.2 展望

按照预期实现了所有的功能开发，但仍有不足之处，该系统的共识算法使用的是Fabric测试共识算法solo。因资源限制，系统尚未部署在线上环境，整个系统在本地运行。在未来的不断改善过程中，将使用更加先进的PBFT（拜占庭容错共识算法），并在线上环境完成部署。

参考文献

1. 甘俊,李强 ,陈子豪,张超.区块链实用拜占庭 容错共识算法的改进[J].计算机应用,2019,39(07):2148-2155.
2. Ge Yu,Bin Wu,Xinxin Niu. Improved Blockchain Consensus Mechanism Based on PBFTAlgorithm[A]. International Association of Applied Science and Engineering, Proceedings of 2020 2nd International Conference on Advances in Computer Technology, InformationScience and Communications ,2020.
3. 方轶,邓建球,丛林虎,刘崇屹.一种基于环签名的 PBFT 区块链共识算法改进方案[J].计算 参考文献 75 机工程,2019,45(11):32-36.
4. SHI Liucheng,GUO Zhaozhong.Baguena: A Practical Proof of Stake Protocol with a Robust Delegation Mechanism[J].Chinese Journal of Electronics,2020,29(05):887-898.
5. Vukoli M . The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication[C]// International Workshop on Open Problems in Network Security. Springer International Publishing, 2016.
6. Stefanos Leonardos,Daniël Reijsbergen,Georgios Piliouras. Weighted voting on the blockchain: Improving consensus in proof of stake protocols[J]. International Journal of Network Management,2020,30(5).
7. 丁庭琛 , 陈世平 . 基于信用分级的 PBFT 共识算法改进方案[J]. 计算机系统应用,2020,29(09):255-259.
8. 宁梦月, 刘东苏. 基于区块链技术的数字知识产权保护方案研究[J]. 情报理论与实践, 2020: 1-11.
9. Zheng, Kai, et al. Model checking PBFT consensus mechanism in healthcare blockchainnetwork[J]. 2018 9th International Conference on Information Technology in Medicine andEducation (ITME). IEEE, 2018,877-881.
10. 赵海鸿,姚中原,祝卫华等.一种基于区块链的电子合同共享方案[J].应用科学学 报,2023,41(02):359-368.
11. CLIM A, ZOTA R D, CONSTANTINESCU R. Data exchanges based on blockchain inm-Health applications[J]. Procedia Computer Science, 2019,160.
12. 周茂君, 秦文琰. 区块链 3.0 时代电商信用体系的重构[J]. 学习与实践, 2019(11): 47-56.
13. 田林. 基于区块链的电子病历存证系统的设计与实现[D].山东师范大学,2019.45-55.
14. 韦 智 勇 , 周 立 广 . 基 于 区 块 链 的 电 子 病 历 存 证 系 统 设计与实现[J]. 软件工程,2020,23(09):56-58.
15. Chandrasekar C , Prabhakaran V . cryptographic algorithm[J]. Computer ence &Communications Dictionary, 2000:324-324.
16. Vermeulen A H , Atlas A B , Barth D M , et al. Distributed storage system with web services client interface[P]: US,13/791414 2015.
17. Datta S , Bhaduri K , Giannella C , et al. Distributed Data Mining in Peer-to-Peer Networks[J]. IEEE Internet Computing, 2006, 10(4):18-26.
18. Yuguang Y , Shuxin Z , Informatics D O . Review and Research for Consensus Mechanismof Block Chain[J]. Journal of Information Security Research, 2018.
19. Gervais A , Karame G O , Karl Wüst, et al. On the Security and Performance of Proof of Work Blockchains[C]// Acm Sigsac Conference on Computer & Communications Security. ACM, 2016.
20. Kiayias A , Konstantinou I , Russell A , et al. A Provably Secure Proof-of-Stake BlockchainProtocol[J]. 2017.
21. Jiacheng H , Xinhua X U , Shichun W . Improved scheme of delegated proof of stake consensus mechanism[J]. Journal of Computer Applications, 2019.
22. Sukhwani H , Martinez J M , Chang X , et al. Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric)[C]// 2017 IEEE36thSymposium on Reliable Distributed Systems (SRDS). IEEE, 2017.
23. Garay J, Kiayias A, Leonardos N. The bitcoin backbone protocol with chains of variable difficulty[C]//Advances in Cryptology–CRYPTO 2017: 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20–24, 2017, Proceedings, Part I 37. Springer International Publishing, 2017: 291-323.
24. Bonneau J, Miller A, Clark J, et al. Sok: Research perspectives and challenges for bitcoin and cryptocurrencies[C]//2015 IEEE symposium on security and privacy. IEEE, 2015: 104-121.
25. Zohar A. Bitcoin: under the hood[J]. Communications of the ACM, 2015, 58(9): 104- 113.
26. Zhao Y, Zhao J, Jiang L, et al. Privacy-preserving blockchain-based federated learning for IoT devices[J]. IEEE Internet of Things Journal, 2020, 8(3): 1817-1829.
27. (美)伊桑·布朗作；吴滠栩译.Node 与 Express 开发[M].2021 年 6 月第 1 版.北京：人民邮电出版社,2021:2-4.
28. 周虎. 一种基于 JWT 认证 token 刷新机制研究[J]. 软件工程,2019,22(12):18-20.
29. 陆江城.基于非对称加密算法的加密系统的研究与实现[D].广州:广东工业大 学,2020.

致谢

大学四年时光荏苒，一转眼走到了大学本科的尽头。这四年是离开家乡，独自拼搏的四年；也是展望未来，更进一步的四年。初入校时那青涩懵懂的样子还停留在脑海中，而一转眼，落下毕业论文的最后一个句号，大学生涯却也即将完美谢幕。

在此，要感谢我的父母，离家四年，父母也从未放下对我的关怀，每日每周每月都不间断地发来问候。

要感谢东北大学和计科1904班，虽然因为三年疫情，大家聚少离多，但大家从最一开始的不适应，到逐渐熟悉，最终完美融合，只需要一个过程。也感谢学校、所有老师教授们的栽培与教育，即便平时会有埋怨学校的食堂太油，吐槽快递站搬移，疫情封校太严等等，但离别时的母校也就会成为记忆中的母校，永远闪光。

要感谢我的导师信俊昌与王中阳老师，在毕设前期，忙于求职，老师给了建议与鼓励；毕设中后期，老师又每周与会面，指出工程项目中可能存在的问题。亦师亦友的陪伴与鼓励让感激。

当离开学校，希望的心中满怀感激。