仲恺农业工程学院

毕 业 设 计

基于区块链的农产品溯源平台的设计与实现

姓 名 陈振聪

院（系） 信息科学与技术学院

专业班级 计算机科学与技术183班

学 号 201810214302

指导教师 王潇 胡鑫（校外）

职称/职务 讲师 工程师

论文答辩日期 2022年 5月14日

仲恺农业工程学院教务处制

**Design and Implementation of Agricultural Products Traceability Platform Based on Blockchain**

Chen Zhencong

College of Information Science and Technology

Zhongkai University of Agriculture and Engineering

Guangzhou, China

**Supervisor: Lecturer Wang Xiao**

**Engineer Hu Xin（Extramural Tutor）**

**学生承诺书**

本人郑重承诺：所呈交的毕业论文（设计）是本人在导师的指导下独立进行研究所取得的成果。除了文中已用特别标志加以标记的引述内容之外，本论文（设计）不含有任何其他个人或集体已经发表或撰写的研究成果。对本文研究做出过重要贡献的个人或集体，均已在文中以明确的方式标明。若在毕业论文（设计）的各项检查、评比中被发现有抄袭、剽窃或其他的违规行为，本人愿按学校有关规定接受处理，并承担相应的法律责任。

学生（签名）：

年 月 日

摘 要

在现如今的信息化时代中，社会经济文化的发展快速增长，中国国民经济在不断增长。消费者能购买到的农产品越来越多，品种越来越丰富。在农产品出产的过程中一旦出现食品安全问题，不仅会造成巨大的经济损失，也会危害民众的身体健康，引发一系列的社会问题。建立起有效的、安全的农产品溯源平台能够让消费者清楚的看见农产品从生长到收获再到加工的全过程，能有效解决消费者与生产者之间的信息不对等的问题。而在区块链技术下的农产品溯源系统能有效防止信息被篡改，保障信息的真实性、可靠性。

本农产品溯源平台以区块链技术Hyperledger Fabric为核心并部署智能合约，采用基于Spring Boot的若依框架开发。实现了管理员对用户的管理、农户对农作物详情及生长过程的管理、物流公司对运输的管理、原料厂商对农作物的质量检测、生产厂商对农产品的加工管理、消费者对农作物进行溯源等功能。

关键词：农产品溯源 区块链 Hyperledger Fabric 智能合约 Spring Boot

Abstract

In today's information age, social economy and culture are developing rapidly. China's national economy is also growing. Consumers can buy more and more agricultural products, varieties are becoming richer and richer. People's eyes are no longer limited to what they can eat, but also want to know whether the food is safe. Once food safety problems occur in the production of agricultural products, they will not only cause huge economic losses, but also harm people's health and cause a series of social problems. Establishing effective and safety of agricultural products traceability platform can let consumers clearly see whole process from growth to harvesting and processing of agricultural products, can effectively solve the problem of trust between consumers and producers, and under the block chain technology of agricultural products traceability system can effectively prevent information been tampered with, guarantee the authenticity of information.

The agricultural product traceability platform is based on Hyperledger Fabric blockchain technology and smart contract deployment. It adopts Spring Boot framework to develop and MySQL database to realize data persistence. The realization of the administrator to user management, farmers to crop management, raw material manufacturers to crop quality testing, manufacturers to agricultural products processing management, logistics companies to transport management, consumers to crop traceability.

**Key words：**agricultural products traceability; blockchain; Hyperledger Fabric; smart contract; Spring Boot

目 录

[1 绪论 1](#_Toc102999284)

[1.1 系统研究背景 1](#_Toc102999285)

[1.2 系统开发的目的与意义 1](#_Toc102999286)

[1.3 国内外研究现状 2](#_Toc102999287)

[2 系统开发平台及使用技术 2](#_Toc102999288)

[2.1 系统开发平台 2](#_Toc102999289)

[2.2 系统相关技术介绍 3](#_Toc102999290)

[2.2.1 Spring Boot 3](#_Toc102999291)

[2.2.2 Mybatis 3](#_Toc102999292)

[2.2.3 FastDFS 3](#_Toc102999293)

[2.2.4 Redis 4](#_Toc102999294)

[2.2.5 区块链技术 4](#_Toc102999295)

[2.2.6 Hyperledger Fabric 4](#_Toc102999296)

[3 需求分析 5](#_Toc102999297)

[3.1 系统应用需求 5](#_Toc102999298)

[3.2 平台业务流程设计 5](#_Toc102999299)

[3.3 可行性分析 7](#_Toc102999300)

[3.3.1 技术可行性 7](#_Toc102999301)

[3.3.2 经济可行性 7](#_Toc102999302)

[3.3.3 操作可行性 8](#_Toc102999303)

[4 系统设计 8](#_Toc102999304)

[4.1 系统项目架构 8](#_Toc102999305)

[4.2 系统功能模块设计 9](#_Toc102999306)

[4.3 数据库设计 10](#_Toc102999307)

[4.3.1 数据库E-R图 10](#_Toc102999308)

[4.3.2 数据库表结构 11](#_Toc102999309)

[4.4 Fabric区块链设计 15](#_Toc102999310)

[4.4.1 区块链设计 15](#_Toc102999311)

[4.4.2 智能合约设计 15](#_Toc102999312)

[4.4.3 系统内部溯源流程设计 18](#_Toc102999313)

[5 系统实现 19](#_Toc102999314)

[5.1 Fabric区块链网络搭建 19](#_Toc102999315)

[5.2 用户登录 22](#_Toc102999316)

[5.3 系统管理模块实现 23](#_Toc102999317)

[5.3.1 用户管理 23](#_Toc102999318)

[5.3.2 角色管理 24](#_Toc102999319)

[5.3.3 日志管理 25](#_Toc102999320)

[5.4 农作物管理模块实现 27](#_Toc102999321)

[5.4.1 农作物信息管理 27](#_Toc102999322)

[5.4.2 农作物生长记录管理 28](#_Toc102999323)

[5.5 农产品加工模块实现 29](#_Toc102999324)

[5.6 质量检测模块实现 30](#_Toc102999325)

[5.7 运输管理模块实现 31](#_Toc102999326)

[5.8 农产品溯源模块实现 32](#_Toc102999327)

[6 系统测试 33](#_Toc102999328)

[6.1 系统管理模块测试 33](#_Toc102999329)

[6.2 农作物管理模块测试 34](#_Toc102999330)

[6.3 农产品加工模块测试 35](#_Toc102999331)

[6.4 运输管理模块测试 36](#_Toc102999332)

[6.5 农产品溯源模块测试 37](#_Toc102999333)

[7 总结与展望 37](#_Toc102999334)

[参考文献 39](#_Toc102999335)

[致谢 40](#_Toc102999336)

# 1 绪论

## 1.1 系统研究背景

农产品是人们生活中不可缺少的一部分，因此人们最关心的就是农产品的质量安全。近年来，以习近平同志为核心的党中央高度重视食品药品安全工作，习近平总书记多次作出重要指示，强调要坚持“最严谨的标准、最严格的监管、最严厉的处罚、最严肃的问责”[1]。在苏丹红鸡蛋、毒豆芽、毒大米等事件被曝光后，民众对于买到农产品可能危害自己身体的担忧愈加严重。这是由于消费者与生产商之间的信息不对称，农产品只有在消费者购买之后才能确定其质量好坏，农产品生产过程中是否存在农药超标使用和激素滥用情况用户也无从得知。为了更好保障消费者的权益，打消消费者的顾虑以及提高消费者的购买意愿，农产品供应链应该变得更加透明，并且能够可不断追踪和认证农产品生产到出售的全流程信息。这些功能可以由农产品溯源系统提供。

在我国大力推进科技创新的战略背景下，已经出现了运用大数据、物联网等信息技术构建的农产品溯源系统，但是这种传统的农产品溯源系统在实践过程中容易出现一些问题，导致直接影响了农产品溯源的结果，例如数据没有去中心化，溯源数据的数据控制管理权都掌握在企业方，一些员工可能会对数据进行直接篡改，又或者数据库可能被不法分子入侵，导致数据被盗窃利用、恶意破坏，致使市场上的高危害食品泛滥，不可控，甚至出现大规模的食品安全问题。

“区块链”的概念首次在2008年被提出，此后的二十多年里，区块链技术被越来越多的人知晓、了解，这是一种由多方参与共同维护全网一致的分布式账本，它使用自动脚本代码的智能契约来编程和操作数据，采用块链数据结构来验证和存储数据，并使用密码技术来保障数据传输和访问的安全性[2]。由于区块链系统具有分布式高冗余存储、时序数据且不可篡改和伪造、去中心化信用、自动执行的智能合约、安全和隐私保护等优势，使得区块链技术不仅仅可以使用在数字加密货币领域，同时在经济、金融和社会系统中也存在广泛应用前景[3]。将区块链技术应用在农产品溯源领域，可以解决农产品溯源中可能出现的过于中心化、信息易篡改等问题，因此设计并实现一个基于区块链技术的农产品溯源平台是非常必要的。

## 1.2 系统开发的目的与意义

基于区块链技术的农产品溯源平台通过提供一条清晰、透明的数据区块链给予消费者了解农产品生产全过程情况的机会，将动植物的养殖和种植过程、农产品生产加工过程、农产品运输过程、零售商对食品包装的过程等关键溯源数据上传至区块链上，实现从农场到餐桌的全过程检验监督。

采用区块链技术来进行农产品溯源系统的开发可保障数据的可信问题，它能更加方便消费者对农产品安全信息的查询，避免各个角色之间繁琐的人工对账，提高业务处理效率。同时由于区块链去中心化的特点，它的数据是由多方参与者共同进行管理和维护，每个参与者都可提供节点并存储链上的数据，并且共同保证数据的可靠性。这样能有效防止数据被一方篡改，以确保溯源信息的真实性、可靠性，也方便有关部门从中介入进行溯源查询。

## 1.3 国内外研究现状

欧盟作为食品溯源系统的先驱，为了应对疯牛病的食品安全问题，建立起一个对牛、牛肉及牛肉制品的验证和注册体系，对牛肉及牛肉制品进行溯源。加拿大于2004年开始推行国家食品溯源体系，并承诺在该体系下，能保证80%的国产食品从农产品原料到零售均可获得溯源[4]。

我国溯源系统的研究始于2002年，并且目前已建立起相对比较完善、涵盖面广、具有统一性的农产品溯源系统体系[5]。但传统的农产品溯源体系还是存在一些问题，一方面因为农产品溯源系统还未完全普及，各个省份之间关于农产品溯源的系统体制也不径相同。另一方面国内的农产品溯源平台比较多且杂，这些系统在设计时采用的识别方式、数据存储方式、网络部署方式等都不完全一致。由于没有统一的标准，不同溯源平台的溯源信息也难以达到共享，难以进行跨系统查询，并且使用的是传统的农产品溯源系统，其数据安全性还有待提升。

目前区块链技术在农产品溯源上的应用也渐渐出现，有采用区块链技术，通过“链上+链下”双模存储机制和基于智能合约的供应链信息管理构建的粮油食品全供应链信息安全管理原型系统[6]。有对农产品溯源信息双链存储模式和智能合约进行设计，然后采用联盟区块链技术构建从田间到餐桌的农产品全产业链质量安全可信溯源系统[7]。

# 2 系统开发平台及使用技术

## 2.1 系统开发平台

表1 系统开发环境配置

|  |  |  |
| --- | --- | --- |
| 项目 | 详情 | 备注 |
| 操作系统 | Windows 10 家庭版  Centos 7 |  |
| 浏览器 | Google Chrome | 版本 100.0.4896.127 |
| 服务器 | Tomcat 9.0.38 | 轻量级应用服务器 |
| 系统开发语言 | Java  go | JDK1.8  go1.17.7 |
| 数据库 | MySQL 8.0.21 | 关系型数据库管理系统 |
| 开发工具 | IntelliJ IDEA 2019.1  Visual Studio Code | JAVA开发工具  前端代码开发工具 |

表2 硬件开发平台

|  |  |  |
| --- | --- | --- |
| 项目 | 详情 | 备注 |
| 处理器 | Intel（R）Core(TM) i5-7300HQ 2.30GHz | 4核4线程 |
| 内存 | Google Chrome |  |
| 硬盘 | 固态硬盘128GB |  |
| 显卡 | NVIDIA GeForce GTX 1050Ti | 显存大小4GB |

## 2.2 系统相关技术介绍

### 2.2.1 Spring Boot

Spring Boot简化了Spring开发框架的开发、配置、调试、部署工作，仅需在Maven、Gradle配置文件中进行少量配置，即可在开发过程中使用所需的框架[8]。

### 2.2.2 Mybatis

Mybatis是一个可以自定义SQL、存储过程和高级映射的持久层框架，能通过简单的XML或注解的方式实现 SQL 的灵活配置。

### 2.2.3 FastDFS

FastDFS是一个开源的轻量级分布式文件系统，它的主要功能包括：文件存储，文件同步和文件访问，以及高容量和负载平衡，它的优势是在于解决了大数据量存储和负载均衡等问题，特别适合以中小文件为载体的在线服务[9]。在本文所设计的系统中需要一些图片数据来辅助农户呈现自己的农作物生长情况，因此选用FastDFS技术来存储一些中小文件数据。

### 2.2.4 Redis

Redis是一个灵活的高性能key-value数据结构存储，可以用来作为数据库、缓存和消息队列。它有着以下几个优势：高性能，并且支持 Pipelining 命令，可一次发送多条命令来提高吞吐率，减少通信延迟；支持数据持久化；支持主从复制，配置文件中只需要一行来设置它。

### 2.2.5 区块链技术

区块链技术是一种通过去中心化、高信任的方式集团维护一个可靠数据库的技术方案，数据库中的每个区块包含系统的全部数据信息，使用数字签名验证信息的有效性和完整性，通过密码学原理链接到下一个区块并形成一条主链[10]。区块链结构示意图如图1所示。

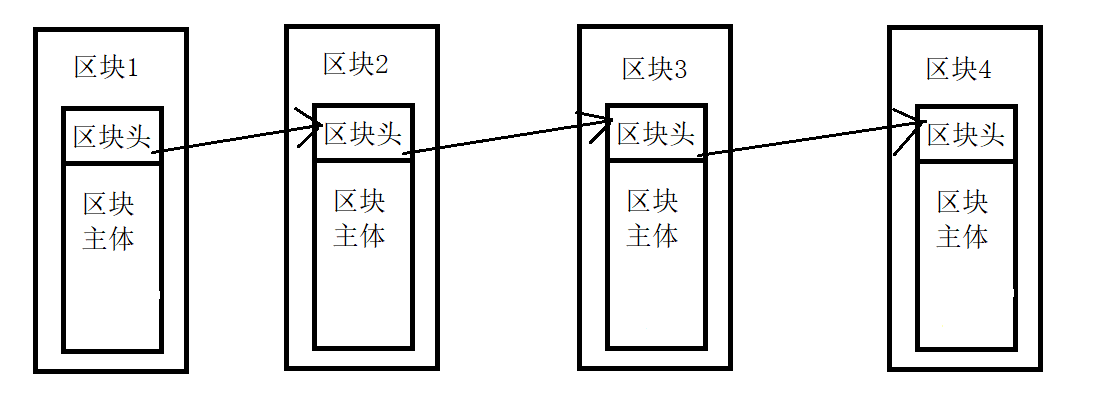


图1 区块链结构示意图

区块头包含的属性有：版本号、时间戳、哈希指针、随机数以及merkle树根哈希。其中，哈希指针用于各个区块之间的链接；区块主体存储由一笔笔交易构成的merkle树，merkle树用于存储每一笔交易数据转化后的哈希值。

### 2.2.6 Hyperledger Fabric

Hyperledger Fabric是来自Linux Foundation的开源项目，这是一个模块化区块链框架，也是企业区块链平台实际采用的标准。Fabric区块链采用了基于账户的模型，基于账户的模型可方便地查询交易余额或业务状态数据[11]。智能合约也更适合于在基于账户的模型之上构建, 其针对状态数据更易处理复杂的业务逻辑[11]。

# 3 需求分析

## 3.1 系统应用需求

本系统是基于区块链技术实现的农产品溯源平台。建立一个农产品质量安全追溯信息化体系需要包括农作物生产安全管理、农产品质量安全追溯管理、农产品加工安全追溯管理和物流运输管理等。通过给农产品添加唯一识别编码，完善农产品加工信息的添加识别，规范农产品溯源过程中所出现的信息，如质检报告、运输报告等，建立农产品质量追溯数据库，实现食用农产品全生产过程的可溯源性。通过这种高可靠性的质量安全保证体系，对农产品供应商来说，能提高农产品厂商的品牌形象；对农产品的销售商来说，这种可靠的、可追溯质量的农产品能让消费者有更高的认可度，提高消费者的购买意愿，提高商品的市场价值；对消费者来说，能对自己所食用的农产品进行溯源，保障了消费者对自己所购买商品的知情权，能够更好的进行反馈和维权。由此本系统从用户的角度来讲需要包含以下几个方面：

（1）管理员，负责管理所有用户，拥有最高访问权限，管理员能够创建登录该系统的账号并分配其相对应的访问权限，可以完成对平台的登录日志管理和操作日志管理。

（2）农户，负责管理农作物，可以管理作物的种植信息，能够对自己的农作物生长全过程提供足够全面的信息，对农作物生长的周期记录，查看作物详情，生长过程详情，能够及时通知物流公司进行运输。

（3）原料厂商，负责从农户手中收集农作物并进行质量检测，能够对农作物进行库存管理，能够及时通知物流公司进行运输。

（4）生产厂商，负责对农产品的加工操作进行管理，能够安排工人对农作物进行加工，对加工完的农产品添加产品信息，能够及时通知物流公司进行运输。

（5）零售商，负责生产唯一的溯源ID并出售商品，能够对运输的商品进行签收操作，并一键生成溯源ID。

（6）物流公司，负责管理运输订单，能够在收到订单请求后向司机派送订单任务，记录农产品在各个环节中的运输情况。

（7）消费者，能根据唯一ID进行农产品的全过程溯源。

## 3.2 平台业务流程设计

本平台整个业务流程思路：由各个商家采集并获取农产品从生产到销售中间的所有环境的源头数据；商家在将数据进行处理后上传到指定区块链上，区块链上的数据将得到保护，不被篡改；消费者通过使用农产品溯源ID在农产品进行农产品溯源，即可查看农产品的生产到销售过程。区块链追溯业务流程设计图如图2所示。

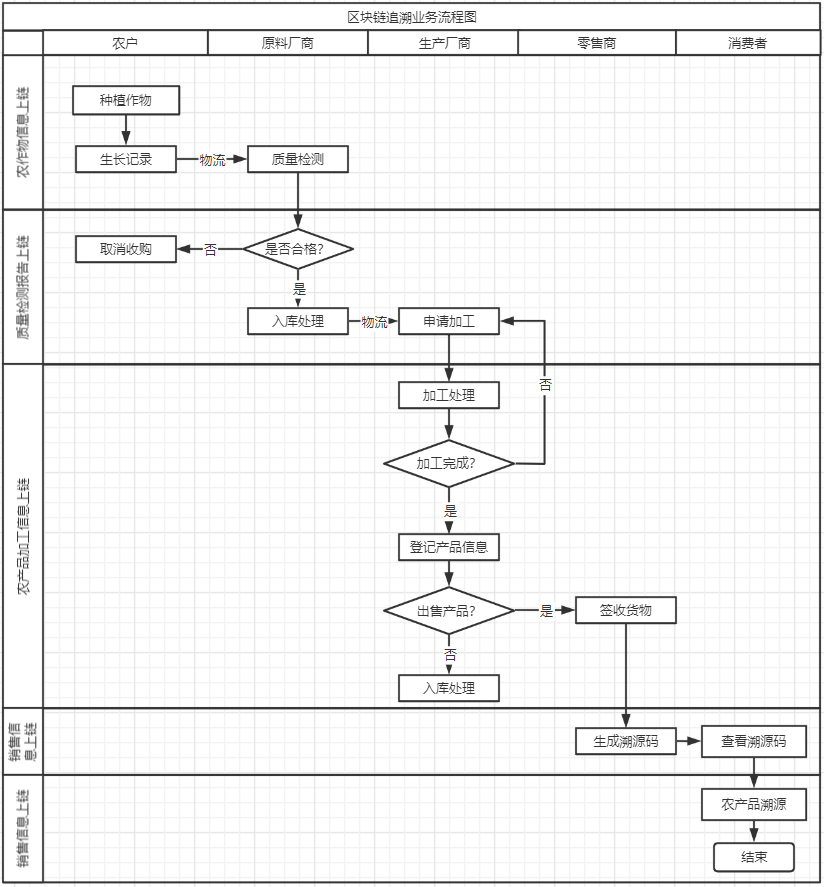


图2 区块链追溯业务流程设计图

（1）农户新增农作物种植信息，并上传至区块链中，后续农户对农作物的生长情况进行定期记录，并将记录信息上传至区块链上。待农作物完全成熟时，选择一家原料厂商添加运输订单，由物流公司通知司机进行配送；

（2）配送订单完成时，原料厂商对收到的农作物进行质量检测，并进行入库。选择一家生产厂商添加运输订单，由物流公司通知司机进行配送；

（3）配送订单完成时，生产厂商库里增加一项未加工的农产品信息，生产厂商负责人添加农产品加工任务，生产厂商员工进行加工任务记录，并将记录上传至区块链。待农作物加工完成后，生产厂商负责人将产品信息填写完毕，农产品状态变为已加工。选择一家零售商添加运输订单，由物流公司通知司机进行配送；

（4）配送订单完成时，零售商对农产品进行签收，签收成功后即可生成农产品唯一溯源ID；

（5）消费者根据农产品的溯源ID进行农产品溯源，即可查看农产品从生产到销售的全过程。

## 3.3 可行性分析

### 3.3.1 技术可行性

本系统前端采用基于Vue的若依前端框架RuoYi-Vue，使用到了Vue和Element UI。前端使用框架可以大量减少重复代码的书写，并且这个框架免费。区块链技术使用的是Hyperledger Fabric，Hyperledger Fabric是来自Linux Foundation的开源项目，这是一个模块化区块链框架，也是企业区块链平台实际采用的标准，通过建立智能合约进行供应链的追踪和跟踪，Hyperledger Fabric也有成功应用在供应链的例子，供应链行业是供应商、制造商和零售商的全球分布式网络。Hyperledger Fabric 网络可以通过在网络中提高事务的透明度和可追踪性来改进供应链流程。在Fabric网络上，具有分类账访问权限的公司可以查看相同的不可变数据，实施责任制并降低造假的风险，因此本系统的开发在技术上是可行的。

### 3.3.2 经济可行性

Fabric相比于比特币、以太坊这些区块链平台，一个很重要的优势是Fabric 可以利用不需要本地加密货币的共识协议来推动智能合约的执行，这意味着Fabric在达成共识和执行智能合约时不会有额外的开销。Fabric对服务器的硬件要求也没有比特币和以太坊高，在服务器部署方面可以选择多机部署Fabric网络以提升系统安全性、稳定性，也可以使用单机部署，并让各个节点使用不同的端口，以此来节约成本。

开发技术方面，Fabric区块链开发平台是开源免费的，系统开发所需的Mysql数据库、Redis缓存中间件、RuoYi框架、FastDFS文件管理系统等也都是开源的，无需额外预算，因此本系统的开发在经济上是可行的。

### 3.3.3 操作可行性

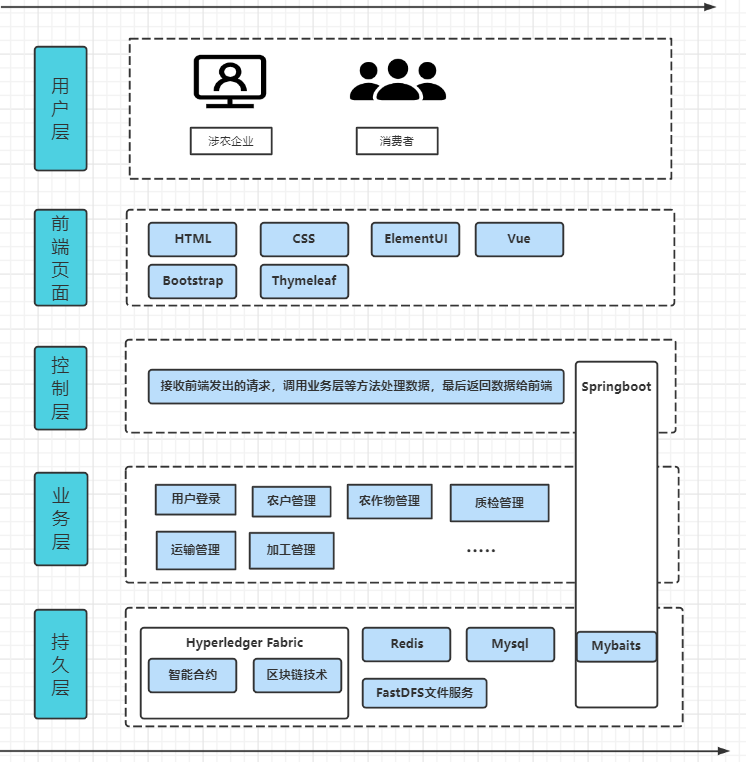
本系统采用基于Spring Boot的前后端分离系统，前端使用RuoYi-Vue框架进行简易开发，区块链服务器选择部署在Linux上，按照Hyperledger Fabric官方文档进行部署，并进行适当地修改。最终前台页面给消费者进行溯源，后台界面由各参与者进行编辑，最终呈现的出来的系统应该对所有参与者的学习成本都不算高，因此本系统在操作上是可行的。

# 4 系统设计

## 4.1 系统项目架构

本系统使用了基于Spring Boot的权限管理系统，若依管理系统进行辅助开发，实现了前后端分离的开发模式。系统前端采用了Vue框架技术进行开发，系统后端的区块链相关功能模块方面使用node.js进行开发，基础数据管理方面使用Spring Boot整合Mybatis，数据库使用开源的Mysql数据库，为了方便一些中小文件的存储，选择在服务器部署FastDFS文件服务来存储图片信息，图片路径及相关信息存储在Mysql数据库里。区块链技术使用Hyperledger Fabric,智能合约采用Golang语言编写。部署区块链的服务器运行环境为Centos7操作系统，需要先安装curl，docker，docker-compose，Golang，Python，node.js。结合农产品的生产过程及溯源的要求，系统总体架构图如图3所示。

本系统逻辑架构设计一共分为五层。持久层采用Fabric部署区块链服务器来存放溯源数据，并使用智能合约和Fabric的共识机制对溯源的数据进行验证、更新，把溯源关键数据上传至区块链网络中。Mysql数据库存放基础数据，包括用户数据、日志数据以及一些不需要上传至区块链的农作物管理数据。本平台使用FastDFS来存储图片等小文件，并将路径存储在Mysql数据库里。业务层主要负责整个溯源系统的农产品溯源数据采集、溯源信息查询、用户信息的增删改查等业务功能，为企业、消费者提供操作支持以及可靠的溯源信息。控制层负责接受前端发出的页面请求，调用业务层的方法处理数据，最后把返回的数据打包发送给前端页面。

图3 系统总体架构图

## 4.2 系统功能模块设计

农产品溯源平台根据上述需求分析可以划分为系统管理模块、农作物管理模块、农产品加工管理模块、质量检测模块、运输管理模块、农产品溯源模块。本系统的系统功能模块图如图4所示。

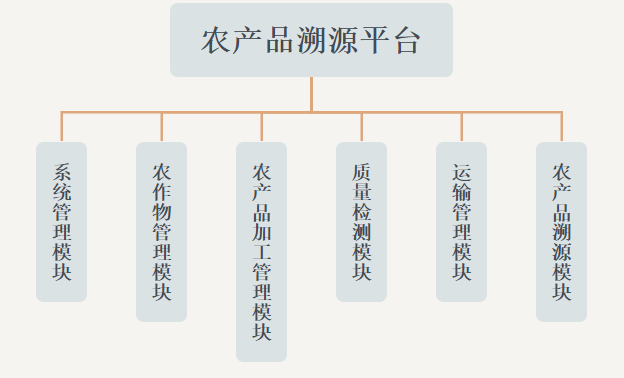


图4 系统功能模块图

上述功能模块描述如下：

（1）系统管理模块，包括用户管理、日志管理等。用户管理由管理员进行操作，主要用于维护系统的用户信息，如操作权限、启用状态等；日志管理负责记录各个角色进行的数据操作，如登录记录、操作记录等。

（2）农作物管理模块，农作物管理由农户进行操作，主要用于管理农作物的种植信息，对农作物生长过程进行详细记录，包括产品名称、生产地、生长周期，湿度、温度、光照等状况。

（3）农产品加工管理模块，包括产品信息管理、加工任务管理等。由生产厂商的负责人对产品信息进行记录，并发布农产品的加工任务，生产厂商的员工登记加工操作记录，包括具体的加工方式、操作时间、涉及产品等。

（4）质量检测模块：由原料厂商接收到农作物后，对其进行质量检测，将结果记录并上传至区块链中。

（5）运输管理模块：由物流公司管理各个环节用到的运输订单，向驾驶员派送订单任务，需要记录每次运输的产品批次、出发地、目的地以及到达时间。

（6）农产品溯源模块：供消费者根据产品的溯源ID进行溯源，查看产品从生长到销售的全过程信息。

## 4.3 数据库设计

### 4.3.1 数据库E-R图

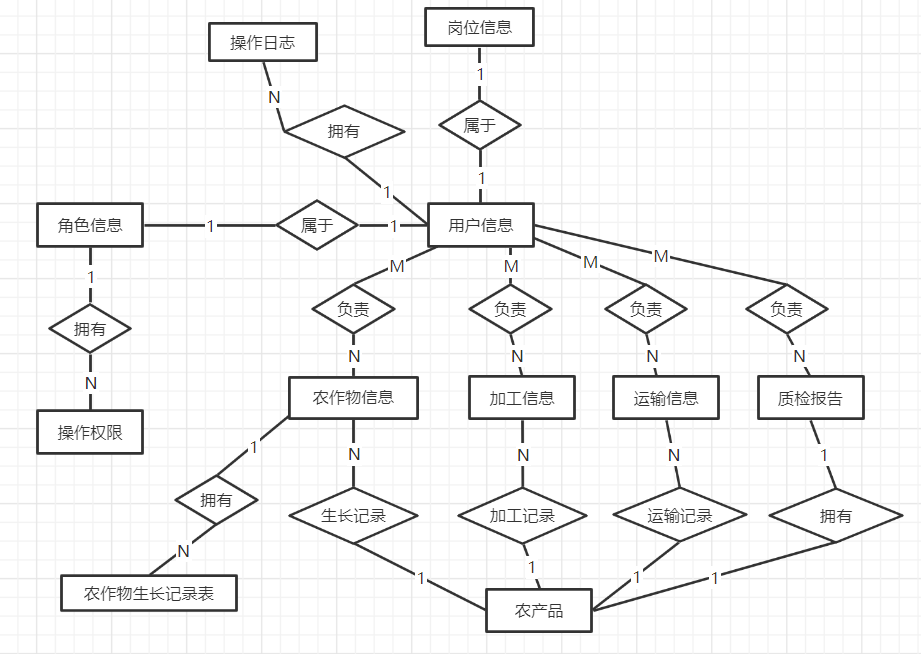


图5 系统全局E-R图

根据上述功能模块分析，本农产品溯源平台的系统全局E-R图如图5所示。用户信息实体包括农户、加工商、销售商、运输公司等实体，为方便系统管理分出岗位信息实体，根据岗位进行权限分配。农产品在农作物生长阶段，可由多个农户进行多次记录，是多对多关系，同理加工信息、运输信息、质检报告也是多对多关系。农作物在生长过程中需要多次记录生长情况，因此农作物信息和农产品是一对多的关系，农产品在生产到出售过程中需要多次运输，多次加工，因此加工信息与农产品、运输信息与农产品也是一对多的关系。

### 4.3.2 数据库表结构

以下将列出一些与系统业务相关的数据库表的结构。

（1）农作物信息表（crops）：记录农作物种植的相关信息，表结构如表3所示。

表3 农作物信息表（crops）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 是否唯一 | 备注 |
| crops\_id | varchar(255) | 否 | 是 | 农作物ID，主键 |
| crops\_name | varchar(255) | 否 | 否 | 农作物名称 |
| crops\_type | varchar(255) | 否 | 否 | 作物类型 |
| address | varchar(255) | 否 | 否 | 原产地 |
| telephone | varchar(64) | 否 | 否 | 手机号 |
| user\_name | varchar(64) | 否 | 否 | 农户 |
| is\_bagging | varchar(2) | 否 | 否 | 是否套袋 |
| fertilizer | varchar(32) | 否 | 否 | 肥料名称 |
| plant\_annual | varchar(16) | 否 | 否 | 种植年份 |
| seecding\_period | varchar(32) | 否 | 否 | 育苗周期 |
| irrigate\_period | varchar(32) | 否 | 否 | 灌溉周期 |
| weeding\_period | varchar(32) | 否 | 否 | 除草周期 |

（2）农作物生长记录表（crops\_growth）：记录农作物生长过程中的数据，例如湿度、温度、生长阶段、图片记录等，表结构如表4所示。

表4 农作物生长记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 是否唯一 | 备注 |
| crops\_id | varchar(255) | 否 | 否 | 农作物ID |
| crops\_growth\_id | varchar(255) | 否 | 是 | 生长记录ID，主键 |
| temperature | varchar(32) | 否 | 否 | 温度 |
| humidness | varchar(32) | 否 | 否 | 湿度 |
| illumination | varchar(32) | 否 | 否 | 光照强度 |
| comment | varchar(255) | 否 | 否 | 生长情况介绍 |
| picture\_path | varchar(255) | 否 | 否 | 图片路径 |

（3）农产品加工信息表（crops\_process）：用于记录农产品加工前的数据，如农作物名称、原产地、负责人、入库状态、加工状态等，表结构如表5所示。

表5 农产品加工信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 是否唯一 | 备注 |
| crops\_id | varchar(255) | 否 | 是 | 农作物ID，主键 |
| crops\_name | varchar(255) | 否 | 否 | 农作物名称 |
| origin | varchar(255) | 否 | 否 | 原料厂商 |
| principal | varchar(255) | 否 | 否 | 负责人 |
| process\_state | varchar(1) | 否 | 否 | 加工状态，0为未加工，1为已加工 |
| storage\_state | varchar(1) | 否 | 否 | 入库状态，0为未入库，1为已入库 |

（4）农产品加工任务表（process\_task）：用于记录对农作物的加工任务，包括操作人员、加工类型、操作内容等，表结构如表6所示。

表6 农产品加工任务表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 是否唯一 | 备注 |
| task\_id | varchar(255) | 否 | 是 | 任务ID，主键 |
| task\_name | varchar(64) | 否 | 否 | 任务名称 |
| task\_principal | varchar(64) | 否 | 否 | 操作人姓名 |
| telephone | varchar(64) | 否 | 否 | 操作人电话号码 |
| content | varchar(255) | 否 | 否 | 任务内容 |
| remark | varchar(255) | 是 | 否 | 备注 |

（5）农产品表（product）：用于记录农产品详细信息，如农作物ID、农产品ID 、农产品名称等数据，表结构如表7所示。

表7 产品表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 是否唯一 | 备注 |
| product\_id | varchar(255) | 否 | 是 | 农产品ID，主键 |
| product\_batch | varchar(255) | 否 | 否 | 农产品批次ID |
| crops\_id | varchar(255) | 否 | 否 | 农作物ID |
| principal | varchar(255) | 否 | 否 | 负责人 |
| telephone | varchar(50) | 否 | 否 | 负责人电话号码 |
| workshop | varchar(50) | 否 | 否 | 生产车间 |
| time | time | 否 | 否 | 生产完成时间 |
| net\_content | varchar(50) | 否 | 否 | 净含量，如500g/袋 |
| storage\_way | varchar(50) | 否 | 否 | 保存方法 |
| storage\_deadline | varchar(50) | 否 | 否 | 保质期 |
| remark | varchar(255) | 是 | 否 | 备注 |

（6）质量检测表（product\_quality\_check）：用于记录农产品批次的质检结果，其表结构如表8所示。

表8 质量检测表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 是否唯一 | 备注 |
| quality\_id | int | 否 | 是 | 自增ID，主键 |
| product\_id | varchar(255) | 否 | 是 | 农产品ID |
| product\_batch | varchar(255) | 否 | 否 | 农产品批次ID |
| organization | varchar(64) | 否 | 否 | 质检机构 |
| time | time | 否 | 否 | 质检时间 |
| result | varchar(32) | 否 | 否 | 质检结果 |
| remark | varchar(255) | 否 | 否 | 备注 |
| picture | varchar(255) | 否 | 否 | 质检证书路径 |

（7）运输表（transport）：用于记录农产品的流通行为中出现的数据，如物流编号、出发点与目的地、运输司机及其手机号，其表结构如表9所示。

表9 运输表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 是否为空 | 是否唯一 | 备注 |
| transport\_id | varchar(255) | 否 | 是 | 物流ID，主键 |
| product\_id | varchar(255) | 否 | 是 | 农产品ID |
| product\_batch | varchar(255) | 否 | 否 | 农产品批次ID |
| driver\_id | varchar(64) | 否 | 否 | 驾驶员ID |
| driver\_name | varchar(64) | 否 | 否 | 驾驶员名字 |
| driver\_dept | varchar(32) | 否 | 否 | 所属部门 |
| license\_plate\_number | varchar(32) | 否 | 否 | 车牌号 |
| telephone | varchar(64) | 否 | 否 | 驾驶员电话号码 |
| depart | varchar(255) | 否 | 否 | 始发地 |
| depart\_time | time | 否 | 否 | 出发时间 |
| arrive | varchar(255) | 否 | 否 | 目的地 |
| arrive\_time | time | 否 | 否 | 到达时间 |

## 4.4 Fabric区块链设计

### 4.4.1 区块链设计

Fabric系统主要由两个应用构成：Orderer和Peer，其中Orderer负责交易共识并生成区块，Peer节点负责模拟执行交易和记账，这种划分可以使整个平台拥有更好的弹性和扩展性[12]。搭建Fabric网络首先需要生成公私钥和证书，通过fabric-ca服务来生成相关证书，Fabric网络中的成员通过提供基于数字证书的信息来表明自己的身份。然后需要创建一个通道（channel）并生成创世区块，再把一个orderer排序节点和五个peer节点加入通道中，五个peer记账节点对应五个组织Org1、Org2、Org3、Org4、Org5，并作为锚节点，负责代表组织和其他组织进行信息交换。最后给每个peer节点部署安装智能合约并实例化。

### 4.4.2 智能合约设计

智能合约的实现，本质是通过赋予对象数字特性，把对象程序化并部署在区块链上，成为网上共享的资源，使单一用户无法对其直接进行修改，只能通过外部事件触发合约的生成、发布、执行来接受、存储、执行数据，进而改变区块链网络中的数字对象的数值和状态[13]。

根据功能模块分析，需要将农产品的种植信息、加工信息、质量检测信息、产品信息以及运输信息上传至区块链网络上，由此需要设计五个智能合约来实现上述功能模块。以下为智能合约的部分设计字段。

表10 智能合约crops存储字段

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为key | 说明 |
| cropsID | string | Y | 作物ID |
| cropsName | string | N | 作物名称 |
| address | string | N | 所在地 |
| registerTime | string | N | 种植日期 |
| farmerName | string | N | 农户名字 |
| farmerTel | string | N | 联系电话 |
| fertilizerName | string | N | 肥料名称 |
| platMode | string | N | 种植方式 |
| baggingStatus | string | N | 是否套袋种植 |
| growSeedingCycle | string | N | 育苗周期 |
| irrigationCycle | string | N | 灌溉周期 |
| applyFertilizerCycle | string | N | 施肥周期 |
| weedCycle | string | N | 除草周期 |
| remarks | string | N | 备注 |
| cropsGrowID | string | N | 生长记录ID |
| cropsBakId | string | N | 作物ID |
| recordTime | string | N | 记录时间 |
| cropsGrowPhotoUrl | string | N | 图片记录url |
| temperature | string | N | 温度 |
| growStatus | string | N | 生长情况 |
| waterContent | string | N | 水分 |
| illuminationStatus | string | N | 光照情况 |

表11 智能合约operate存储字段

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为key | 说明 |
| cropsID | string | N | 作物ID（作为溯源ID） |
| operationID | string | Y | 操作ID |
| operationPeopleName | string | N | 操作人 |
| operationPeopleTel | string | N | 操作人电话 |
| time | string | N | 时间 |
| workContent | string | N | 工作内容 |
| remarks | string | N | 备注 |

表12 智能合约productTest存储字段

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为key | 说明 |
| id | string | Y | 主键 |
| productBatch | string | N | 农产品批次ID |
| productName | string | N | 农产品名称 |
| organization | string | N | 质检机构 |
| time | string | N | 质检时间 |
| result | string | N | 质检结果 |
| picture | string | N | 质检报告url |
| remarks | string | N | 备注 |

表13 智能合约product存储字段

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为key | 说明 |
| productID | string | Y | 农产品ID |
| productBatch | string | N | 农产品批次ID |
| cropsID | string | N | 作物ID（作为溯源ID） |
| principal | string | N | 负责人 |
| telephone | string | N | 联系方式 |

续表13 智能合约product存储字段

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为key | 说明 |
| factory | string | N | 工厂 |
| workshop | string | N | 车间 |
| time | string | N | 生产日期 |
| netContent | string | N | 净含量 |
| remarks | string | N | 备注 |

表14 智能合约transport存储字段

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 数据类型 | 是否为key | 说明 |
| transportID | string | Y | 物流ID |
| cropsID | string | N | 作物ID（作为溯源ID） |
| driverID | string | N | 驾驶员ID |
| driverName | string | N | 驾驶员名字 |
| driverDept | string | N | 所属部门 |
| transportToChainTime | string | N | 物流信息上链时间 |
| depart | string | N | 始发地 |
| arrive | string | N | 目的地 |

### 4.4.3 系统内部溯源流程设计

在存储农产品溯源信息的过程中，使用Hyperledger Fabric SDK将农产品不同环节的信息，如农作物种植信息、农产品加工信息、运输信息等通过invoke函数发送到背书节点，等待背书节点返回一个交易提案，排序节点将这个响应信息和交易进行排序打包区块，并广播到共识网络，再由peer节点验证区块中的交易并检查区块数据是否正确，验证通过后添加区块到通道对应的链上，更新状态数据库State Database，溯源信息输入时序图如图6所示。

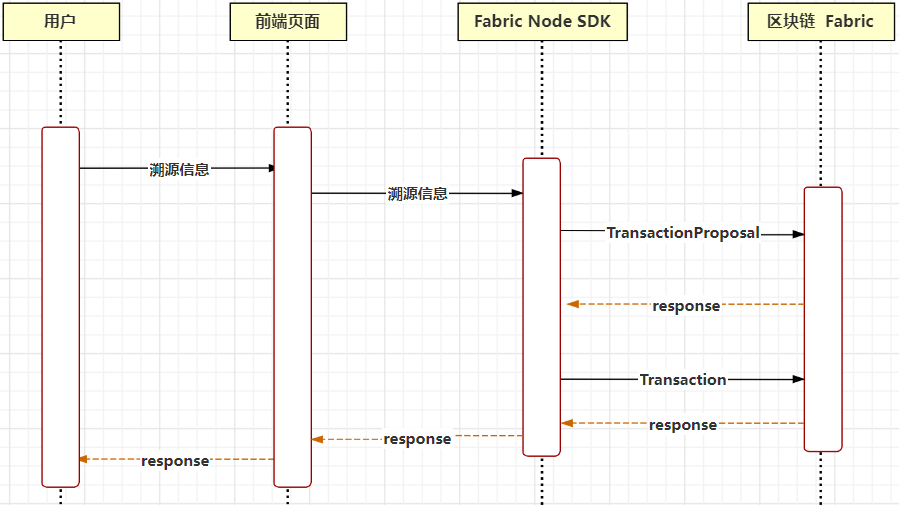


图6 溯源信息输入时序图

进行农产品信息溯源时，使用Shim接口的函数GetState和GetQueryResult，通过检索key值来一次遍历区块，获取对应key值的账本数据。通过此方法得到的时json格式的数据，在客户端把数据处理后即可展示农产品溯源信息。

# 5 系统实现

## 5.1 Fabric区块链网络搭建

本农产品溯源平台的区块链服务搭建在Centos 7 操作系统中，需要先安装docker和docker-compose,因为Fabric网络的peer节点、orderer节点、智能合约都需要运行在docker容器中。

（1）环境准备：安装docker、docker-compose、node.js 8.x、Golang,并配置好环境变量，版本信息截图如图7所示。

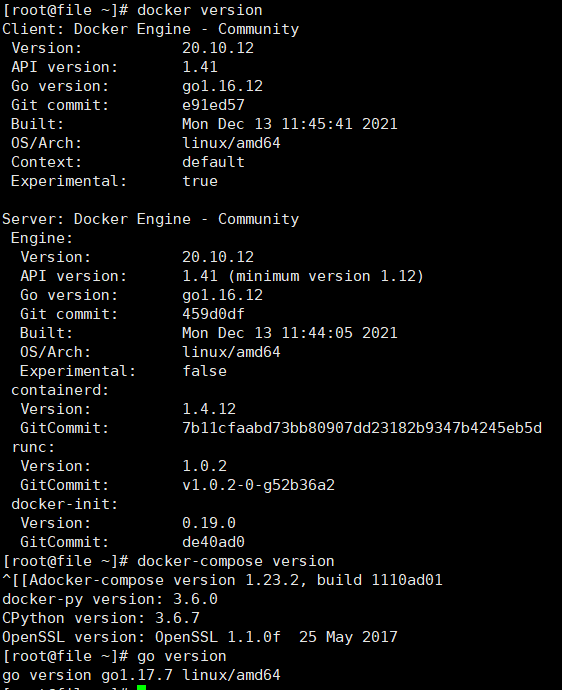


图7 docker、docker-compose、Golang版本信息截图

（2）拉取Fabric源码，本系统使用的是Fabric1.2.0版本，使用git clone命令拉取源码时选择切换到1.2.0版本分支。

（3）通过docker pull命令拉取搭建Fabric网络所需的镜像文件，fabric-peer、fabric-orderer、fabric-ca、fabric-tools、fabric-ccenv、fabric-baseimage、fabric-baseos、fabric-couchdb。使用docker images命令查看已下载的docker镜像文件，镜像文件截图如图8所示。

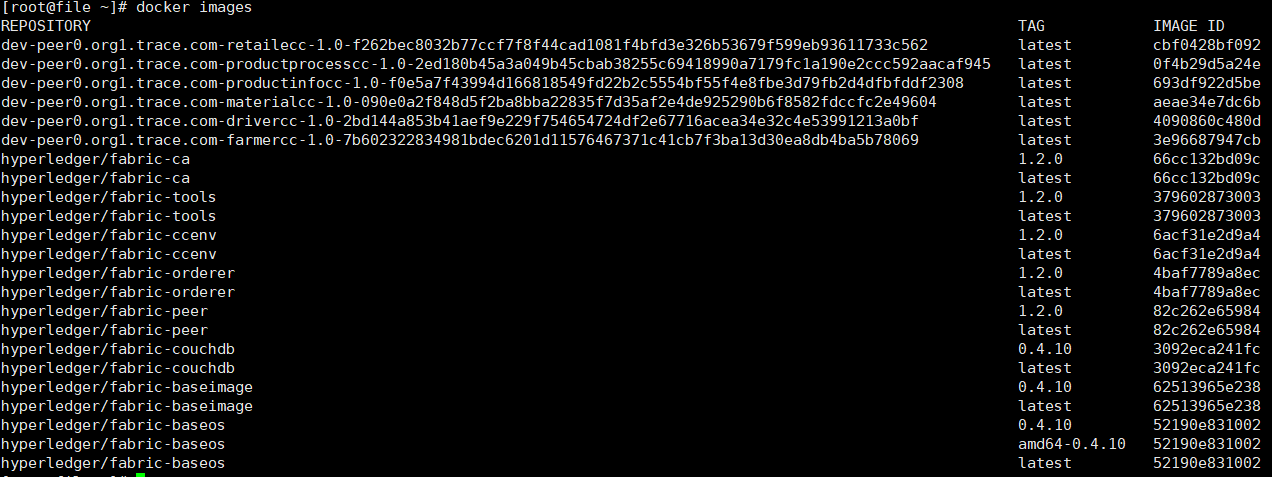


图8 docker镜像文件截图

（4）配置Fabric源码的四个文件configtx.yaml、crypto-config.yaml、docker-compose-cli.yaml、docker-compose.yaml。根据配置文件的内容在network目录下分别执行../bin/cryptogen generate --config=./crypto-config.yaml、../bin/configtxgen -profile Two OrgsOrdererGenesis -channelID byfn-sys-channel -outputBlock ./channel-artifacts /genesis. block、docker exec cli peer channel create -o orderer.trace.com:7050 -c tracechannel -f ./ channel-artifacts/channel.tx，生产组织orderer与组织peer的证书与密钥，创世区块文件、创建通道并配置环境变量。最后通过docker-compose -f docker-compose.yml up -d命令启动Fabric区块链网络，docker-compose-cli.yaml配置各个节点域名和容器使用情况、docker-compose.yaml配置各peer节点的端口使用情况，相关配置截图如图9、图10所示。

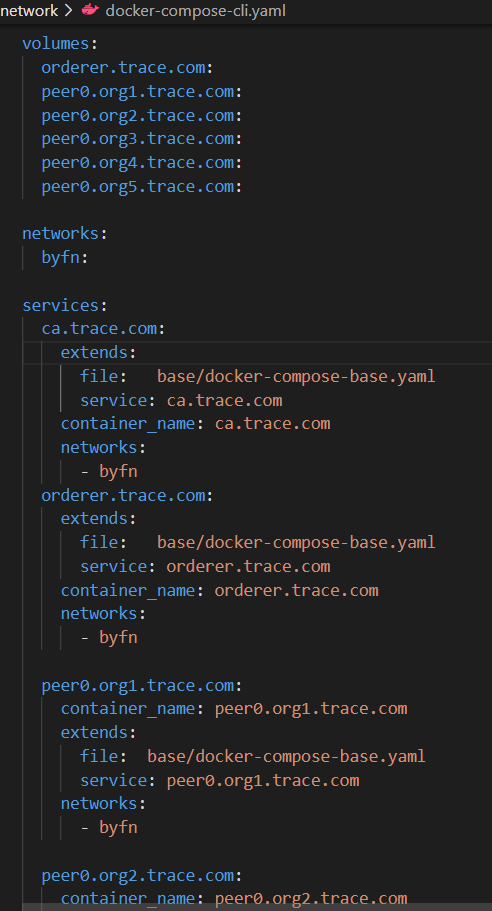


图9 docker-compose-cli.yaml部分配置信息截图

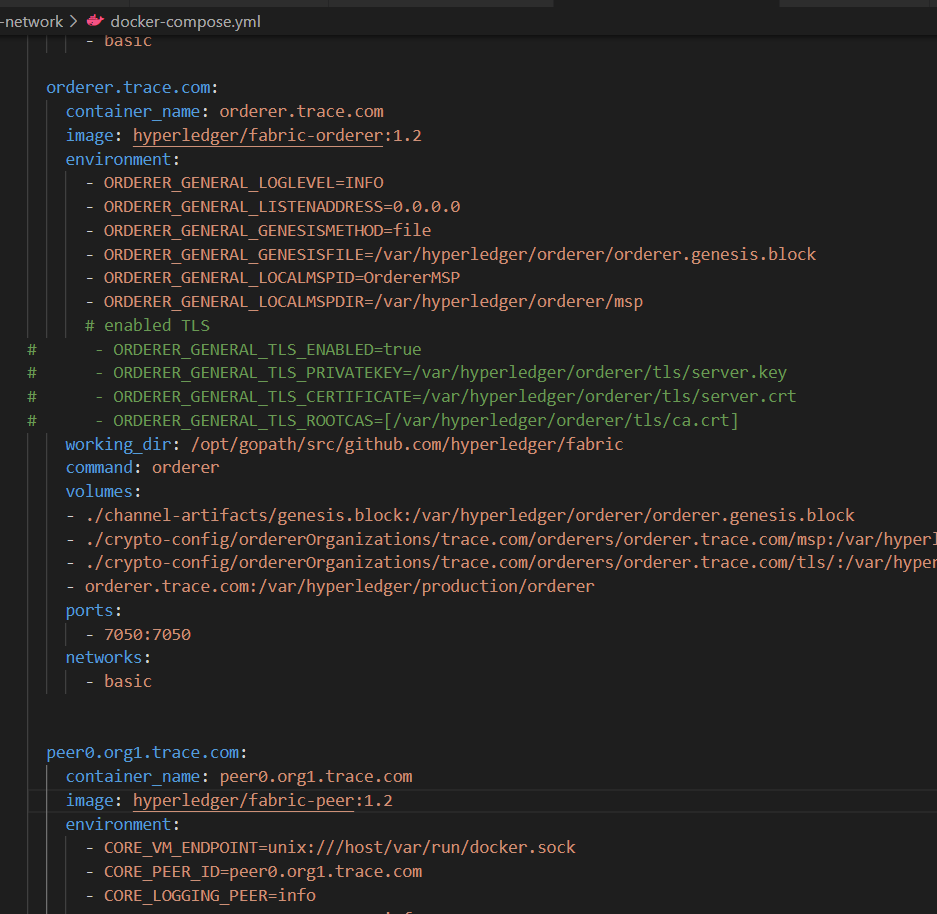


图10 docker-compose.yaml部分配置信息截图

（5）使用docker exec命令将各个peer节点的证书加入通道，更新锚节点。docker exec cli peer channel join -b tracechannel.block、docker exec cli peer channel update -o orderer.trace.com:7050 -c tracechannel -f ./channel-artifacts/Org1MSPanchors.tx。

（6）给每个peer节点安装上智能合约，并实例化。docker exec cli peer chaincode install -n farmercc -v 1.0 -l golang -p github.com/chaincode/farmercc/go/、docker exec cli peer chaincode instantiate -o orderer.trace.com:7050 -C tracechannel -n farmercc -l golang -v 1.0 -c '{"Args":[""]}' -P "OR ('Org1MSP.member','Org2MSP.member')"、docker exec cli peer chaincode invoke -o orderer.trace.com:7050 -C tracechannel -n farmercc -c '{"function":"initLedger","Args":[""]}'。实例化后智能合约即可生效。

以上Fabric网络的构建和智能合约的初始化操作就完成了。接下来如果需要要使系统前端连接上区块链服务器，还需要通过node enrollAdmin.js、node registerUser.js来安装用户密钥。

## 5.2 用户登录

登录页面如图11所示，如果是没有账号的消费者，可以直接点击消费者溯源按钮进入农产品溯源页面，根据溯源编号进行溯源。有账号的相关参与者根据自己的账号密码，输入验证码进行登录。如果密码错误或用户不存在则登录失败，如果登录成功，系统会添加一条登录日志进数据库，用户则进入系统界面。系统界面会根据用户的权限等级，提供相应的菜单选项，如果是管理员则拥有所有菜单功能选项。



图11 登录页面

## 5.3 系统管理模块实现

### 5.3.1 用户管理

用户管理界面中展示所有的用户，管理员能够根据用户名称、手机号码来查找指定用户，用户管理界面如图12所示。点击“新增”按钮能够进行新增一名用户，在该页面填写用户相关信息，并给予某个角色的权限，新增用户界面如图13所示；点击“重置”按钮将用户密码重置；点击状态的开关选项可以控制用户的可用状态。



图12 用户管理界面



图13 新增用户界面展示

### 5.3.2 角色管理

角色管理页面中展示所有角色的信息，管理员能根据角色名称、权限字符等信息进行查找角色，角色管理界面如图14所示。点击“新增”按钮能添加一名新角色，根据实际需求给该角色赋予相对于的菜单管理权限。点击状态选项开关能够随时启用或关闭一位角色；点击“数据权限分配”按钮能进行数据权限分配，决定该角色所能访问的数据范围，数据权限修改界面如图15所示；点击状态的开关选项可以控制角色的可用状态。

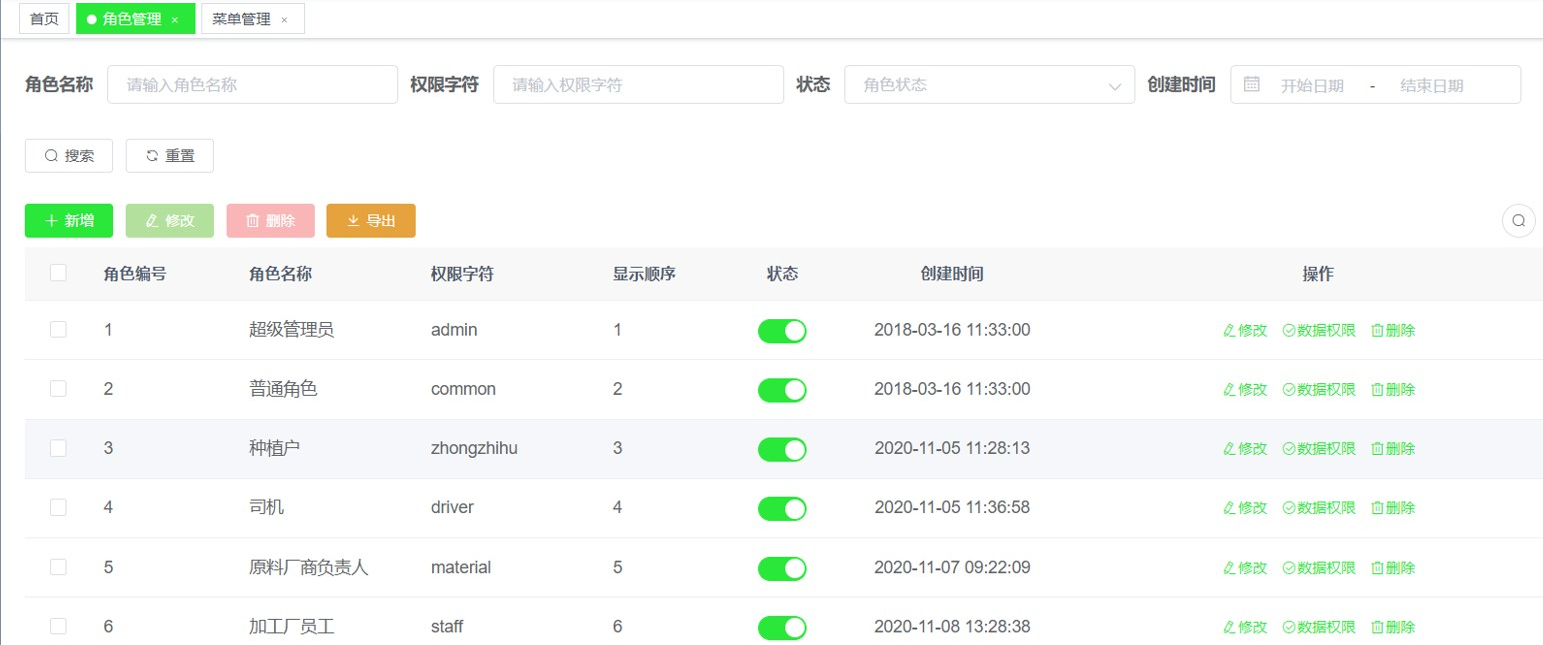


图14 角色管理界面



图15 数据权限修改界面

### 5.3.3 日志管理

日志管理模块分为登录日志与操作日志，登录日志用于记录登录情况，管理员能根据登录地址、用户名称或登录状态进行查找操作，登录日志界面如图16所示。操作日志用于记录在哪个系统模块，哪位操作人员进行了什么操作，管理员能根据系统模块、操作人员、操作类型或操作时间进行查找操作记录，操作日志界面如图17所示，点击查看详情按钮能进一步查看该操作所使用的方法以及请求参数，操作日志详情界面如图18所示。



图16 操作日志页面



图17 操作日志界面



图18 操作日志详情界面

## 5.4 农作物管理模块实现

农户点击左侧农作物管理菜单选项进入农作物管理页面，农作物管理界面如图19所示。点击“添加作物”能添加一条新的种植信息；点击“周期记录”能对作物的生长情况进行记录；点击“过程详情”能查看作物的全部生长记录。在作物成熟之后，可点击“通知运输”能联系物流公司进行配送任务。



图19 农作物管理界面

### 5.4.1 农作物信息管理

点击添加作物按钮能够添加农作物信息。填完所需信息后点击确定，能够将作物上传到区块链上，并添加到后台数据库中。添加完农作物信息后，能通过点击作物详情按钮查看农作物的详细信息，农作物详细数据界面如图20所示。



图20 农作物详细信息数据界面

### 5.4.2 农作物生长记录管理

点击周期记录按钮，能对当前农作物的生长情况进行登记，包括温度、湿度、生长阶段等并进行拍照记录，点击确认后即可上传到区块链上，农作物生长过程界面如图21所示。



图21 农作物生长过程界面

本模块的图片上传功能用到了FastDFS文件服务系统，上传图片时先向FastDFS的Tracker Server发送上传请求，Tracker Server查询可用的Storage Server的ip和端口返回给客户端，客户端上传文件到Storage Server，storage将图片信息保存到磁盘，并返回一个fileid，客户端把fileid存入数据库和区块链上。读取照片时也是先向Tracker Server发送请求，把接受到的storage的ip和端口与fileid组合即可从Storage Server上下载图片。FastDFS上传下载的工作流程图如图22所示。

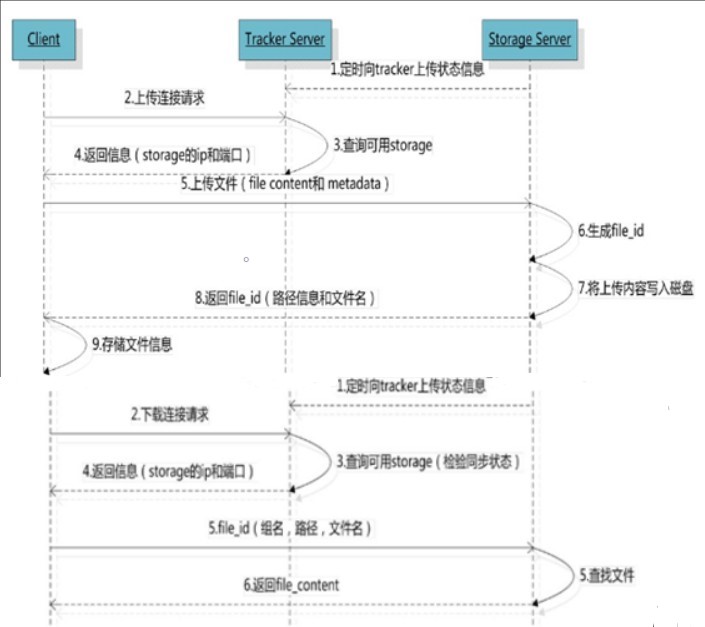


图22 FastDFS工作流程图

## 5.5 农产品加工模块实现

进入农产品加工模块首先需要使用生产厂商的账号进行登录，如果是厂商负责人则进入农产品加工管理界面，农产品加工管理界面如图23所示。由负责人点击“任务分发”按钮发布加工任务，发布加工任务后，作物会进入“加工中”状态，等待员工上报任务进度。负责人点击“员工工作量”可查看员工对农产品的加工完成进度，员工工作证明界面如图24所示。待产品完成加工后厂商负责人点击“填写产品基本信息按钮”添加农产品信息，通知运输后即可将产品出库。



图23 农产品加工信息界面



图24 员工工作证明界面

在生产厂商负责人派发任务后，厂商里的员工登录系统就能看到任务，生产厂商员工界面如图25所示，员工根据任务的具体内容完成任务后，上报自己的工作进度，同时这个操作内容也会上传到区块链上，供用户溯源。



图25 生产厂商员工界面

## 5.6 质量检测模块实现

原料厂商在收到农作物后，进行质量检测，并把质检报告上传到区块链上，添加质检结果界面如图26所示。



图26 添加产品质检结果界面

## 5.7 运输管理模块实现

在上述模块完成种植或加工后，需要运输服务，由其他企业点击通知运输并选择物流公司以及司机，运送订单就会发送到物流管理界面，物流信息管理界面如图27所示。使用司机的账号登录进系统就能进入物流管理界面，点击“开始配送”进行订单配送，运输完成时点击完成配送，运输记录就会上传到区块链上。



图27 物流信息管理界面

## 5.8 农产品溯源模块实现

农产品最后运输到销售商时，销售商在签收农产品后能够生产该产品的唯一溯源ID，零售商产品管理界面如图28所示。



图28 零售商产品管理界面

后台根据农作物ID、产品ID以及零售商的基本信息随机生成唯一溯源ID。根据产品溯源ID，用户在系统首页可用选择消费者溯源入口进行农产品溯源，可查看农产品从生产到销售的全过程记录，农产品溯源界面如图29所示。系统先以消费者输入的溯源ID为key值，在区块链上搜索对应的区块并获得农作物ID、产品ID，再以这两个ID作为key值再次遍历区块，获取对应key值的账本数据，并把数据处理好展示到页面上。

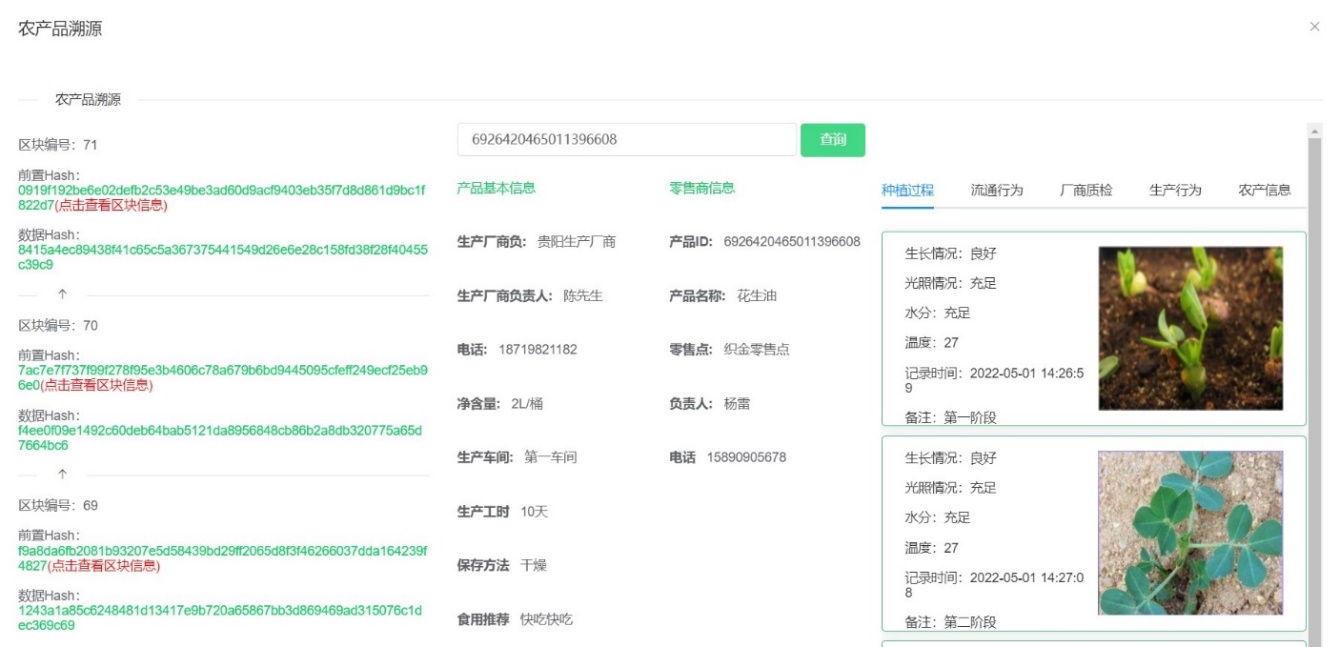


图29 农产品溯源界面

# 6 系统测试

## 6.1 系统管理模块测试

表15 用户登录测试表

|  |  |
| --- | --- |
| 测试编号 | 001 |
| 测试项目 | 用户登录 |
| 测试目的 | 测试用户能否登录进农产品溯源平台 |
| 测试依据 | 用户能够顺利进入农产品溯源平台 |
| 测试步骤 | 1、用户进入登录页面；  2、输入正确的账号密码，输入验证码；  3、点击登录按钮。 |
| 预期结果 | 成功登录，用户登陆后进入系统界面。 |
| 测试结果 | 与预期结果一致，进入系统界面。 |

表16 用户权限控制测试表

|  |  |
| --- | --- |
| 测试编号 | 002 |
| 测试项目 | 检测用户的权限控制 |
| 测试目的 | 测试不同的用户登录进系统拥有不同的操作权限 |
| 测试依据 | 不同的用户拥有不同的权限，登录成功后展示不同的菜单功能选项。 |
| 测试步骤 | 1、使用管理员账号登录本系统，查看菜单选项；  2、退出系统，使用普通账号farmer登录本系统，查看菜单选项；  3、退出系统，使用普通账号driver登录本系统，查看菜单选项。 |
| 预期结果 | 1、管理员账号登录进系统后，加载所有的菜单选项；  2、普通账号farmer登录进系统后，加载“农作物管理”菜单选项；  3、普通账户driver登录进系统后，加载“物流管理”菜单选项。 |
| 测试结果 | 与预期结果一致。 |

## 6.2 农作物管理模块测试

表17 农作物信息管理测试表

|  |  |
| --- | --- |
| 测试编号 | 003 |
| 测试目的 | 验证添加农作物功能是否正常，并通过“作物详情”按钮显示出来。 |
| 测试依据 | 农户可以实现添加农作物及查看作物详情的功能，并将数据上传到区块链上。 |
| 测试步骤 | 1、在农作物管理界面，点击“添加作物”按钮并输入相关信息；  2、点击保存；  3、点击“作物详情”按钮查看农作物详情。 |
| 预期结果 | 1、步骤2后系统弹出“数据上链成功”的提示；  2、步骤3后用户能够查看到农作物的详细信息。 |
| 测试结果 | 与预期结果一致。 |

表18 农作物生长记录测试表

|  |  |
| --- | --- |
| 测试编号 | 004 |
| 测试项目 | 农作物生长记录相关功能测试 |
| 测试目的 | 验证农户的周期记录功能是否正常，检查图片上传功能是否正常，并通过“过程详情”按钮查看该农作物所有的生长记录。 |
| 测试依据 | 农户可以实现周期记录作物的功能，并将农作物的生长过程上传到区块链上。 |
| 测试步骤 | 1、在农作物管理界面，点击“周期记录”按钮并输入相关信息；  2、点击“图片上传”，选择图片进行上传；  3、点击“确定”；  4、点击“过程详情”按钮查看农作物。 |
| 预期结果 | 1、步骤2后系统提示“图片上传成功”；  2、步骤3后系统将农作物生长记录的数据上传到区块链上，并提示“图片上传成功”；  3、步骤4后，系统显示该作物的所有周期记录。 |
| 测试结果 | 图片格式为.png上传时，提示图片上传失败，将图片换成.jpg格式后，与预期结果一致。 |

## 6.3 农产品加工模块测试

表19 农产品加工任务发布与上报功能测试表

|  |  |
| --- | --- |
| 测试编号 | 005 |
| 测试项目 | 测试农产品加工任务发布与上报功能 |
| 测试目的 | 1、验证生产厂商负责人能否正常发布加工任务；  2、验证生产厂商员工能否正常上报加工任务内容。 |
| 测试依据 | 农产品加工的流程由负责人发布加工任务，员工完成加工任务并上报，加工记录将上传到区块链供消费者溯源。 |
| 测试步骤 | 1、使用生产厂商负责人的账户登录系统；  2、进入农产品加工管理界面，点击“任务发布”，并填写相关信息，点击确认；  3、使用生产厂商员工的账号登录系统；  4、进入工作记录界面，根据负责人发布的任务信息，进行任务上报。  5、点击确认上报任务内容，即农产品加工记录。 |
| 预期结果 | 1、步骤2后，所有生产厂商员工的工作记录界面增加一项任务信息；  2、步骤5后，农产品的加工记录数据上传到区块链上，系统提示“数据上链成功”。 |
| 测试结果 | 与预期结果一致。 |

表20 农产品加工任务查看测试表

|  |  |
| --- | --- |
| 测试编号 | 006 |
| 测试项目 | 测试查看加工记录的功能 |
| 测试目的 | 验证生产厂商负责人能否查看农产品的所有加工情况。 |
| 测试依据 | 在生产厂商员工上报任务内容后，负责人能够查看员工的上报情况，并根据这些内容判断产品是否能够出库并进行运输。 |
| 测试步骤 | 1、使用生产厂商负责人的账户登录系统；  2、进入农产品加工管理页面，点击“员工工作量”按钮。 |
| 预期结果 | 系统展示包含该产品所有加工记录的弹窗。 |
| 测试结果 | 与预期结果一致。 |

## 6.4 运输管理模块测试

表21 用户通知运输功能测试表

|  |  |
| --- | --- |
| 测试编号 | 007 |
| 测试项目 | 用户通知运输功能测试 |
| 测试目的 | 测试用户发起运输通知后，司机能否收到运输订单。 |
| 测试依据 | 用户可以发起运输通知，让司机对货物进行运送。 |
| 测试步骤 | 1、使用农户账号登录进系统，对一项农作物数据点击“通知运输”；  2、选择目的地和司机，点击确定；  3、使用司机账号登录进系统，进入物流管理界面。 |
| 预期结果 | 物流管理界面增加了一项运输订单。 |
| 测试结果 | 与预期结果一致。 |

表22 用户运输功能测试表

|  |  |
| --- | --- |
| 测试编号 | 008 |
| 测试项目 | 用户运输功能测试 |
| 测试目的 | 检测运输订单能否顺利完成。 |
| 测试依据 | 司机用户能够完成订单，将数据传输到下一个目的地，并将数据上传至区块链上。 |
| 测试步骤 | 1、使用司机账号登录进系统，进入物流管理界面；  2、对一项未完成的订单，点击“开始运输”；  3、对运输中的订单，点击“完成配送”；  4、使用目的地的相关账号登录系统，查看相应的管理界面。 |
| 预期结果 | 点击“开始运输”后，订单进入运输中状态；点击“完成配送”订单提示配送完成，运输数据上传至区块链上，系统提示“数据上链成功”；切换账号后查看相对于的数据增加。 |
| 测试结果 | 与预期结果一致。 |

## 6.5 农产品溯源模块测试

表23 农产品溯源功能测试表

|  |  |
| --- | --- |
| 测试编号 | 009 |
| 测试项目 | 农产品溯源功能测试 |
| 测试目的 | 测试能否根据溯源ID进行农产品溯源 |
| 测试依据 | 消费者能够根据产品溯源ID对产品进行溯源。 |
| 测试步骤 | 1、进入登录界面，点击“消费者溯源”；  2、进入农产品溯源界面，输入产品溯源ID；  3、点击“查询”。 |
| 预期结果 | 系统展示溯源结果，包括农产品信息、销售商信息、种植过程信息、物流信息、农产品加工过程信息、质检报告、农作物信息。 |
| 测试结果 | 与预期结果一致。 |

# 7 总结与展望

随着社会的不断发展，食品安全问题正在渐渐进入大众的视野，各国也都有建立农产品溯源的一些相关政策与措施，使用IT技术来搭建一个农产品溯源平台是十分必要的，它不仅能够在一定程度上减小食品安全事件的发生，还能打造品牌效应，提供市场价值。本农产品溯源平台通过使用区块链技术Hyperledger Fabric来保障溯源数据的可靠性，可以把农产品生产过程更加公开、透明地展示给政府与消费者看。

在本次系统设计中，使用了目前比较流行的技术，例如若依框架，进行前后端分离的开发，在后端开发时学习了如何Spring Boot与Mybatis，在开发前端时使用若依框架自带的一些组件，进行vue页面的开发，第一次接触前端代码开发遇到了很多诸如接口出现跨域、连接不上后端等问题。区块链服务器的部署也是让我吃尽苦头，只能不断翻看文献，查看别人开源的项目来进行学习。经过这几个月的学习与开发，完成毕业设计，让我重新对大学这四年学习的技能与知识做了一次总结，也补充了很多课程里没有的内容。

本次毕业设计让我感到了自己的渺小，在搭建区块链服务器的时候，有时都会坐一整天没有进度，不停地翻看论坛、文献，还是感觉在做无用功。但哪怕只是在慢慢前进，我也还是在进步，最终把这个系统完成了。

本系统可以优化的地方还有很多，如区块链在上传时，是否可以加一个数据校验再进行上传，毕竟数据一旦上传了就无法再进行修改；还有运输管理模块，原本我打算做一个物流追踪系统，能够把商品的运送路径也记录下来，但是由于目前的知识储备不够，时间也不允许我再慢慢学习，希望在后期的学习中我能有毅力完善迭代本系统的功能。

# 参 考 文 献

1. 李晓理,卜坤,翟玉鹏,王康.基于人工智能技术的重大活动食品安全与风险评估综述[J].北京工业大

学学报,2021,47(05):530-539.

1. 柳祺祺,夏春萍.基于区块链技术的农产品质量溯源系统构建[J].高技术通讯,2019,29(03):240-248.
2. 袁勇,王飞跃.区块链技术发展现状与展望[J].自动化学报,2016,42(04):481-494.
3. 刘俊华,金海水.国外农产品质量快速溯源的现状和启示[J].物流技术,2009,28(11):251-253.
4. 马懿,林靖,李晨等.国内外农产品溯源系统研究现状综述[J].科技资讯,2011(27):158.
5. 许继平,孙鹏程,张新等.基于区块链的粮油食品全供应链信息安全管理原型系统[J].农业机械学报,

2020,51(02):341-349.

1. 刘双印,雷墨鹥兮,徐龙琴等.基于区块链的农产品质量安全可信溯源系统研制[J/OL].农业机械学报:

1-12[2022-05-07].http://kns.cnki.net/kcms/detail/11.1964.S.20220426.1729.016.html

1. 王永和,张劲松,邓安明等.Spring Boot研究和应用[J].信息通信,2016(10):91-94.
2. bojiangzhou.用FastDFS一步步搭建文件管理系统[DB/OL].https://www.cnblogs. com/chiangchou/p/

fastdfs.html. 2017-10-18.

1. 杨现民,李新,吴焕庆等.区块链技术在教育领域的应用模式与现实挑战[J].现代远程教育研究,2017

(02):34-45.

1. 邵奇峰,金澈清,张召等.区块链技术:架构及进展[J].计算机学报,2018,41(05):969-988.
2. 柳正龙. 两步搞定Hyperledger主打区块链解决方案Fabric[DB/OL]. https://r dc.hundsun.com/port

al/article/661.html.2022-04-22.

1. 贺海武,延安,陈泽华.基于区块链的智能合约技术与应用综述[J].计算机研究与发展,2018,55(11):24

52-2466.

# 致 谢

光阴荏苒，岁月如梭。论文的撰写也进入尾声，这份包含了我四年所学的毕业设计也终于完成了，这也代表着我要对我的大学生活说再见了。这四年的生活没有想象中的多姿多彩，甚至中途因为突如其来的疫情，与同学相处变成与“网友”相处，上课形式也变成了一个人在家里上网课了。但是因为这疫情，在回校后我更加珍惜留在学校的时间，也很高兴能在仲园度过四年。

在这四年中，我有太多想感谢的人了，首先是我们四年大学生活的引路人，方龙师兄和潘敏桃师姐，他们给初来乍到的我们很多建议和指引，告诉我们可以做些什么，应该做些什么，让我对这四年的规划有了一个大致的思路。

其次非常感谢我的毕业设计指导老师，王潇老师。第一次上潇姐的课是大一的离散数学，当时我对大学老师的印象就是那种上完课就溜，不会多讲一句话。但是潇姐对学生是很负责的，怕我们挂科甚至主动给我们加课讲解习题，所以我在找导师的时候再一次找到了潇姐，在老师的耐心指导下我完成了论文选题、项目撰写、论文报告，过程中也获得了很多的修改意见，顺利的完成毕业设计，非常感谢王潇老师对我的指导。

感谢我的一位高中舍友，他和我一起来到了仲恺农业工程学院，他加入了社联，当了副社，而我在第一次部门面试失败就没再尝试，四年都在宿舍度过。他和我分享他在大学的所见所闻，处处照顾我的情绪，教会我为人处事，在我最失落的时候，他陪我喝酒聊天，告诉我生活还得往前看，衷心祝愿他以后的研究生生活也能过得顺风顺水。

最后感谢我的舍友、我的老师、我在这大学四年结识的每一个人，给我这平平无奇的大学四年生活点缀了缤纷的色彩，感谢！

**仲恺农业工程学院**

**本科毕业论文(设计)成绩评定表**

|  |  |  |  |
| --- | --- | --- | --- |
| 学 院 | 信息科学与技术学院 | 专业班级 | 计算机科学与技术183班 |
| 姓 名 | 陈振聪 | 学 号 | 201810214302 |
| 毕业论文(设计)题目 | 基于区块链的农产品溯源平台的设计与实现 | | |
| 校内指导教师姓名 | 王潇 | 职称/职务 | 讲师 |
| 校外指导教师姓名 | 胡鑫 | 职称/职务 | 工程师 |
| 指导教师评语及评分：  **系统以SpringBoot为开发环境，以区块链技术Hyperledger Fabric为核心并部署智能合约，实现了一个农产品溯源平台，对基于区块链的农产品溯源平台的开发进行了有益的探索，具有一定的实践意义。该生在整个毕业设计过程中能积极和老师进行沟通，态度端正，工作勤奋，基本理论和专业知识掌握扎实，有很强的分析问题和解决问题的能力。论文能正确引用归纳参考文献，有一定难度，有一定的工作量，对各个功能模块的论述清楚，语句通顺，整体结构符合撰写规范。就系统功能而言，数据层和网络层的设计还有待加强和进一步完善，建议学生今后能在这方面做进一步的有益探索。**  **评分：90。**  签名：  年 月 日 | | | |
| 评阅老师评语及评分：  **课题以区块链技术Hyperledger Fabric为核心并部署智能合约，采用基于Spring Boot的若依框架开发农产品溯源平台。实现了管理员对用户的管理、农户对农作物详情及生长过程的管理、物流公司对运输的管理、原料厂商对农作物的质量检测、生产厂商对农产品的加工管理、消费者对农作物进行溯源等功能。工作量达到本科毕业设计要求，论文撰写较为规范，同意修改后答辩。**  **评分：91。**  签名：  年 月 日 | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 答辩记录：  问题1：农产品溯源平台的区块链模型结构是什么？  答：区块链是区块的链式存储结构，区块包括区块头和区块体两大部分，区块头包含版本号、时间戳、哈希指针、merkle树根；区块主体存储由一笔笔交易构成的merkle 树。  问题2：如何实现区块链链上链下的存储？  答：通过Fabric提供给node.js的外部SDK调用链码读写区块链数据，链上使用CouchDB进行存储；链下使用Mysql数据库进行数据存储，将农产品溯源的详细信息进行记录，关键信息进行上链。  问题3：本系统的HACCP关键控制节点在哪？  本系统的HACCP关键控制节点在农户在周期记录农作物生长过程中对农作物的农药使用情况，原料厂商对农作物的农药残留进行检测，生产厂商在生产加工过程中对农产品加工记录，包括一些防腐剂、香精等添加剂的使用记录。  答辩秘书签名：  年 月 日 | | | | | | | | |
| 答辩小组评价意见及评分：  答辩小组通过对该设计的审核，认为该设计选题较独特，结构合理，层次分明，达到了毕业设计任务规范化要求；答辩时，思路较清晰、能准确地回答问题，系统界面较友好、功能较完善，有一定的实用性。  经答辩小组讨论答辩成绩评定为：82分。  答辩组长签名：  年 月 日 | | | | | | | | |
| 论文(设计)成绩 | 指导老师（40%） | | 评阅老师（20%）  折算分 | | 答辩小组（40%）  折算分 | | 总评分 | 等级 |
| 原始分 | 折算分 | 原始分 | 折算分 | 原始分 | 折算分 |
| 90 | 36.0 | 91 | 18.2 | 82 | 32.8 | 87 | 良好 |
| 学院答辩领导小组(委员会)审核意见 | 签章：  年 月 日 | | | | | | | |

注：1、论文(设计)成绩一栏中，折算分分别由指导老师（40%）、评阅老师（20%）、答辩小组（40%）给出的原始分乘以各自百分比例所得。总评分由折算分相加所得。

2、“等级”：90分以上为“优秀”、80―89分为“良好”、70―79分为“中等”、60―69分为“及格”、59分以下为“不及格”。