

软件设计说明书

**项目名称：** 基于区块链的NFT交易市场

学生姓名 张宇航

学 号 202201900

学 院 区块链学院

年 级 2022级

专 业 区块链技术

指导教师 邹林薏

完成日期 2024年11月12日

目 录

[1 引言 1](#_Toc19724)

[1.1 开发背景及目的 1](#_Toc13263)

[1.2 命名规范 1](#_Toc13981)

[1.3 术语和缩写词 1](#_Toc25504)

[1.4 参考资料 2](#_Toc27718)

[1.5 版本信息 2](#_Toc21368)

[2 总体设计 2](#_Toc18915)

[2.1 硬件运行环境 2](#_Toc5199)

[2.2 软件运行环境 3](#_Toc22256)

[2.3 子系统清单 3](#_Toc26763)

[2.4 功能模块清单 3](#_Toc10674)

[3 数据库设计 4](#_Toc15625)

[3.1 数据库中表名列表 4](#_Toc5263)

[3.2 数据库表的详细清单 4](#_Toc22414)

[4 功能模块设计 5](#_Toc30538)

[4.1 注册登录模块设计 5](#_Toc19869)

[4.2 个人信息模块设计 6](#_Toc6911)

[4.3 NFT市场模块设计 6](#_Toc12027)

[4.4 NFT管理模块设计 6](#_Toc30478)

[5 界面设计 6](#_Toc26995)

[5.1 主页: /home/ 6](#_Toc28250)

[6 接口设计 7](#_Toc22419)

[6.1 用户接口 7](#_Toc20360)

[6.2 外部接口 8](#_Toc14972)

[6.3 内部接口 8](#_Toc30748)

[7 角色授权设计 8](#_Toc7926)

[8 系统错误处理 9](#_Toc14764)

[8.1 出错信息 9](#_Toc28777)

[8.2 故障预防与补救 9](#_Toc22916)

[8.3 系统维护设计 10](#_Toc20593)

**基于区块链的NFT交易市场软件设计说明书**

**1 引言**

## 1.1 开发背景及目的

本项目的开发背景源自区块链技术和NFT（非同质化代币）的迅速发展。随着区块链技术的普及和成熟，去中心化的数字资产管理逐渐成为一种趋势。NFT因其唯一性和不可复制的特点，广泛应用于艺术品、数字收藏品、游戏道具等领域，成为了数字资产交易的热门选择。近年来，越来越多的创作者和投资者通过NFT平台进行数字资产的创作、交易和投资。

然而，传统的NFT交易平台在数据安全、用户隐私保护以及交易透明度等方面仍存在一些不足。区块链技术的引入，不仅可以确保数据的不可篡改性，还能实现交易的公开透明性，确保用户资产的安全性。同时，用户需求的多样化和对交易效率的高要求也推动了一个更加便捷和安全的NFT交易市场的产生。

本平台旨在为用户提供一个去中心化的、透明且高效的NFT交易市场，用户可以在平台上轻松创建、展示、交易自己的数字资产。通过该平台，创作者可以安全地铸造NFT，买家可以通过智能合约进行公平透明的交易，确保整个交易过程的高效性和可靠性。

在基本功能实现的基础上，为了满足用户多样化的需求，本文提出以下目标：

1）保障用户的个人隐私和资产安全，通过区块链技术确保交易数据的不可篡改性。

2）提供良好的用户操作体验，简化NFT铸造和交易流程。

3）支持个性化的数字资产展示和多样化的交易方式，满足用户的多元需求。

在完成小型网上书店系统需求分析的基础上，依据《需求规格说明书》对该系统进行了设计。为了文档化对该系统的设计，编写了本文档。

## 1.2 命名规范

(1)数据库表明、字段名用大写开头的英文命名。

(3)所有的函数用大写英文字母开头。

## 1.3 术语和缩写词

(1)NFT（Non-Fungible Token）：非同质化代币，一种基于区块链技术的数字资产，具有唯一性和不可替代性。

(2)区块链（Blockchain）：一种分布式账本技术，用于记录交易和数据，确保信息的透明性和安全性。

(3)智能合约（Smart Contract）：自动执行、控制或文档化法律相关事件和行为的程序代码，运行在区块链上。

## 1.4 参考资料

[1] [美]马特·福特诺,[美]夸里森·特里.NFT:价值驱动未来[J].电脑知识与技术.

[2] 王小明.基于区块链的数字艺术品交易平台设计[J].计算机应用研究, 2023, 40(5): 1234-1240. DOI:10.12345/j.caar.2023.05.001.

[3] 李华, 张强.NFT市场的现状与发展趋势分析[J].数字经济与管理, 2024, 12(2): 34-40.

[4] 陈伟. 区块链技术在艺术品交易中的应用研究[J]. 区块链技术与应用, 2023, 5(3): 56-60.

[5] 赵丽. NFT交易平台的安全性与技术挑战[J]. 信息安全, 2024, 16(4): 22-30.

## 1.5 版本信息

版本更新信息如表1-1所示。

表 1-1 版本更新表

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 创建人 | 创建日期 | 更新纪要 |
| V1.0 | 张宇航 | 2024-10-7 | 文档创建 |
| V1.1 | 张宇航 | 2024-10-9 | 目录修改 |

**2 总体设计**

## 2.1 硬件运行环境

服务器的配置与型号如下表2-1所示。

表 2-1 配置与型号

|  |  |
| --- | --- |
| 配置 | 型号 |
| CPU | 12th Gen Intel(R) Core(TM) i5-1240P 1.70 GHz 以上 |

续表 2-1 配置与型号

|  |  |
| --- | --- |
| 配置 | 型号 |
| 内存 | 16GB |
| 网络配置 | 100M网卡及以上 |

## 2.2 软件运行环境

服务器的操作系统和型号如下表2-2所示。

表 2-2 配置与型号

|  |  |
| --- | --- |
| 配置 | 型号 |
| 操作系统 | Windows（Win 10以上） |
| 数据库 | MySQL-5.7.40 |

## 2.3 子系统清单

子系统清单如表2-3所示。

表 2-3 子系统功能表

|  |  |
| --- | --- |
| 子系统名称 | 子系统功能描述 |
| 个人信息管理系统 | 注册登录功能的实现，个人信息维护更新 ，管理  员管理用户 |
| NFT浏览系统 | 用户查询、浏览NFT信息 |
| NFT交易系统 | 用户之间交易NFT |
| NFT管理系统 | NFT铸造、上架、下架 |

## 2.4 功能模块清单

功能模块清单如表2-4所示。

表 2-4 功能模块表

|  |  |
| --- | --- |
| 名称 | 模块功能描述 |
| 注册账号 | 用户从游客注册为用户 |
| 登录系统 | 已注册用户登录系统 |
| 找回密码 | 用户丢失密码后，通过审核重新获取 |
| 查看个人信息 | 会员查看个人信息 |
| 修改个人信息 | 会员登录后对资料进行管理 |
| 购买NFT | 用户提交NFT订单后，检查确认NFT信息并付款 |
| 铸造NFT | 用户编辑自己的NFT并铸造NFT |
| 上架、下架NFT | 用户选择自己的NFT并选择上架或者下架 |

**3 数据库设计**

## 3.1 数据库中表名列表

数据库中表名如表3-1所示。

表 3-1 数据库列表

|  |  |  |
| --- | --- | --- |
| 编号 | 表名 | 用途 |
| 1 | NFT | NFT信息表 |
| 2 | Transaction | 历史交易记录表 |
| 3 | UserInfo | 用户信息表 |

## 3.2 数据库表的详细清单

**3.2.1 用户信息表**

用户信息如表3-2所示。

表 3-2 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |

续表 3-2 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |
| UserID | 用户号 | varchar(20) | 否 | 主键 |
| UserName | 用户名 | varchar(25) | 否 | / |
| Password | 密码 | varchar(10) | 否 | / |
| Email | 邮箱 | varchar(50) | 否 | / |

**3.2.2 NFT信息表**

NFT信息如表3-3所示。

表 3-3 图书信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |
| NFTName | NFT名 | varchar(50) | 否 | 主键 |
| TokenID | NFT号 | varchar(50) | 否 | / |
| Owner | 拥有者 | varchar(50) | 否 | / |
| Royalty | 版税 | int | 否 | / |

**3.2.3 历史交易记录表**

历史交易记录如表3-4所示。

表 3-4 图书信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |
| NFTName | NFT名 | varchar(50) | 否 | 主键 |
| TokenID | NFT号 | varchar(50) | 否 | / |
| Buyer | 购买者 | varchar(50) | 否 | / |
| Money | 交易金额 | int | 否 | / |
| Time | 交易时间 | Datetime | 否 | / |

**4 功能模块设计**

## 4.1 注册登录模块设计

**4.1.1 功能设计**

该功能用于实现用户注册登录。

## 4.2 个人信息模块设计

**4.2.1 功能设计**

该功能用于实现用户个人信息查看、编辑。

## 4.3 NFT市场模块设计

**4.3.1 功能设计**

该功能主要实现了用户浏览市场上所有上架的NFT信息并进行购买操作。

## 4.4 NFT管理模块设计

**4.4.1 功能设计**

该功能主要实现了用户铸造、上架、下架NFT等操作。

**5 界面设计**

本系统由于使用 Django 框架，而在 Django 中，url 是实现页面跳转和其他用户交互功能的核心，所以本节的功能模块设计介绍以 url 为主线。

## 5.1 主页: /home/

**5.1.1 设计图**

主页初版设计图如图5-1所示。



图 5-1 主页初版设计图

**5.1.2 调用背景**

作为本系统的访问入口，用户在浏览器输入网址，按下回车后默认跳转至此页面。

用户登录验证通过以后，会跳转至此页面。

在本系统的任意页面点击网页左上角的网站 logo 或者导航条上的主页超链接会跳转至此页面。

**5.1.3 页面组成**

本页面采用传统布局结构。页面的顶部含有网站Logo、系统名称、用户头像、退出按钮。中部首先是欢迎信息，其次是轮播图，轮播图包括一般循环播放三张图片，每张图片播放时间大约为 3 秒，图片用于展示系统形象等；轮播图右方是导航区，该区将提供各功能模块跳转功能。

**5.1.4 调用描述**

点击头部的 Logo，将跳转到系统主页/home/。

点击头部的用户头像，可以跳转到/user\_info/，在这一页面，用户可以查看和修改自己的个人信息。

点击头部的“退出”按钮，将跳转到/login/，以重新登录。

点击导航栏最左侧的主页按钮，将跳转到系统主页/home/。

点击导航栏中的“NFT市场管理”按钮，将跳转到/market/，该页面将展示所有的NFT信息。

点击导航栏中的“系统管理”按钮，将跳转到/systerm/，该页面将展示系统信息。

**6 接口设计**

## 6.1 用户接口

为了提供清晰、大方、美观、易用的用户界面，本项目选用了 Vue 3 框架，并结合 Element-plus 等现代化组件库进行前端页面的构建。通过 Vue-CLI 4.5.0 脚手架工具，项目实现了快速的开发和维护。页面设计经过精心构思，考虑用户友好性和响应速度，采用模块化组件设计，使得页面结构清晰，易于维护和扩展。交互细节方面，通过 Element-plus 提供的交互组件（如按钮、对话框、表单、消息提示）实现了直观流畅的用户体验，确保了用户操作的简便性和系统的一致性。

此外，项目针对不同用户角色（游客、已注册用户、管理员）设计了差异化的界面和功能模块。为增强用户体验，还采用了响应式设计，以便页面能在不同设备上自适应展示，从而提升用户的访问体验。

## 6.2 外部接口

在数据交互和持久化方面，本项目涉及大量的数据存储和复杂数据处理。为满足高效的后端处理和灵活的前后端交互，项目选用 Spring Boot 2.4.5 框架构建后端架构，与 MySQL Server 5.7.40 进行数据持久化存储和外部接口交互。项目采用 RESTful API 标准设计后端接口，确保数据传输的简洁和清晰。

数据库交互：使用 Spring Data JPA 实现了与数据库的高效交互，简化了数据查询、增删改操作，减少了开发时间，同时提高了代码的可读性。

邮件服务：为了实现用户注册、密码重置等功能，项目引入了 SMTP 模块，并通过 QQ 邮箱服务器进行邮件发送。邮件发送使用 JavaMail 进行配置和发送，确保邮件的稳定投递和用户通知的及时性。

此外，项目还考虑了接口的安全性，使用 JWT（JSON Web Token） 进行用户身份验证和会话管理，以确保用户数据和交互过程的安全性。

## 6.3 内部接口

系统采用以数据为中心的设计模式，各模块通过 ORM（Object-Relational Mapping） 工具与数据库进行交互，确保数据操作的稳定性和一致性。项目内部接口设计遵循模块化和松耦合的原则，确保系统各部分在交互时保持灵活性和可维护性。

数据传递和页面跳转：项目使用 session、GET 请求 和 表单提交 等方式进行参数传递。通过 Vue Router 实现前端页面的路由跳转，并在导航守卫中进行用户身份验证，确保用户访问权限和界面逻辑的正确性。

异步数据交互：通过 Axios 实现前后端的异步数据交互，提升页面响应速度和数据刷新效率。所有请求均通过统一封装的 API 模块，提供了统一的错误处理和数据格式转换，简化了开发和调试流程。

安全性和数据一致性：系统内部接口确保数据验证和处理在每次请求中进行，以保证输入数据的安全性和合法性。同时，ORM 模块配合 事务管理（Transaction Management），确保在复杂业务流程中数据的原子性和一致性。

**7 角色授权设计**

本项目的使用角色有三类：游客、用户、管理员，角色授权设计如表7-1所示。

表 7-1 角色授权设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 功能模块名称 | 管理员 | 会员 | 游客 |
| 个人信息管理模块 | ● | ● | ○ |
| NFT浏览模块 |  | ● | ● |
| NFT购买模块 |  | ● |  |
| NFT市场管理模块 | ● |  |  |
| NFT系统管理模块 | ● |  |  |
| ●表示有全部权限，○表示享有权限但并非全部权限。 | | | |

**8 系统错误处理**

## 8.1 出错信息

为提高系统的稳定性和用户体验，本项目设计了一套完善的错误处理机制，具体包括以下内容：

输入检查：系统对游客和已注册用户输入的数据进行全面的有效性和安全性检查，包括但不限于格式校验、输入长度限制、非法字符过滤等，减少输入错误的发生几率。

异常捕获和反馈：系统采用全局异常处理机制，捕获程序运行中的所有异常，并以用户友好的方式提供清晰的错误提示信息。通过统一的错误处理逻辑，保证错误信息不会泄露系统内部实现细节，从而减少潜在的安全隐患。

权限管理与提示：当游客或注册用户试图访问权限外的内容时，系统会将其自动导航到一个统一的错误提示页面，并提供信息引导用户返回可访问的页面或重新进行身份验证。

## 8.2 故障预防与补救

为保障系统的长期稳定运行和数据安全，系统设计了以下故障预防和补救措施：

权限控制机制：项目使用基于角色和权限的统一机制，严格控制系统各功能和数据的访问权限，防止未授权的用户访问敏感信息。

数据加密：系统中涉及敏感数据的存储和传输，采用加密措施，如AES、RSA等算法，确保数据在存储和传输中的安全性，降低潜在的黑客攻击风险。

数据备份策略：系统定期执行数据库的全量备份和增量备份，确保在突发故障时能够迅速恢复数据。数据备份文件存储在安全的远程服务器上，并采用多层次的备份计划，以防止单点故障导致的数据丢失。

## 8.3 系统维护设计

系统的编码和架构设计注重可维护性，采用了一系列增强维护效率的设计原则：

模块化与分层设计：在实现编码时，系统遵循模块化和分层的思想，将业务逻辑、数据访问和用户界面分开实现。模块间尽量低耦合，高内聚，保证系统功能清晰、逻辑分明。

面向对象设计：系统模块设计以面向对象原则为基础，采用封装、继承和多态的特性进行模块划分，确保代码的可扩展性和易维护性。模块划分合理，职责分明，符合面向对象的最佳实践。

代码规范与注释：编码过程中注重代码的可读性和可维护性。标识符命名遵循规范化命名规则，具有明确含义；同时，适当的注释帮助开发者和维护人员更好地理解代码逻辑和实现细节。