

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 花俊毅

学 号 202202531

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](#_Toc22157)

[1.1 开发背景及目的 1](#_Toc9718)

[1.2 命名规范 1](#_Toc25596)

[1.3 术语和缩写词 2](#_Toc152)

[1.4 参考资料 2](#_Toc21606)

[1.5 版本信息 2](#_Toc4873)

[2 总体设计 3](#_Toc17450)

[2.1 硬件运行环境 3](#_Toc22556)

[2.2 软件运行环境 3](#_Toc6902)

[2.3 子系统清单 3](#_Toc14410)

[2.4 功能模块清单 3](#_Toc18651)

[3 数据库设计 5](#_Toc20207)

[3.1 数据库中表名列表 5](#_Toc269)

[3.2 数据库中的表关系 5](#_Toc6802)

[3.3 数据库表的详细清单 5](#_Toc1938)

[4 功能模块设计 7](#_Toc30808)

[4.1 会员购书系统设计 7](#_Toc4454)

[4.2 书评管理系统设计 7](#_Toc3081)

[4.3 订单管理系统设计 7](#_Toc29418)

[5 界面设计 9](#_Toc344)

[5.1 网站设计母板 9](#_Toc30354)

[5.2 主页: / 10](#_Toc14408)

[6 接口设计 11](#_Toc17962)

[6.1 用户接口 11](#_Toc15847)

[6.2 外部接口 11](#_Toc25394)

[6.3 内部接口 11](#_Toc15290)

[7 角色授权设计 13](#_Toc20250)

[8 系统错误处理 14](#_Toc7041)

[8.1 出错信息 14](#_Toc19049)

[8.2 故障预防与补救 14](#_Toc19016)

[8.3 系统维护设计 14](#_Toc2581)

**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

随着区块链技术的迅猛发展和NFT（非同质化代币）在数字艺术及收藏品领域的广泛应用，市场对一个高效、安全且易于使用的NFT交易平台的需求日益增加。现有市场缺乏一个集中化的交易平台，使得用户在交易NFT时面临信息不对称、交易安全性不足及操作复杂等问题。因此，本项目旨在开发一个基于区块链的NFT交易市场，以提供安全、便捷和透明的交易体验。该平台将支持用户创建、购买、出售和管理NFT，并提供实时交易信息和数据分析服务，以满足用户多样化的需求。

在实现基本交易功能的基础上，为了增强用户体验，平台还将实现以下目标：

1）保障用户的数字资产安全和个人隐私。

2）提供直观友好的用户界面，提高平台的可操作性。

3）通过智能合约实现自动化交易，提升交易效率。

## 1.2 命名规范

为了保持代码和数据库设计的一致性和可读性，本项目采用以下命名规范：

1）数据库表名与字段名，数据库表名和字段名采用大写英文字母开头的英文命名，单词之间不使用下划线或其他符号连接。例如：

表名：UserProfile，TransactionRecord

字段名：UserID，NFTPrice

2）函数命名，所有函数的名称使用大写英文字母开头的英文命名，单词之间不使用下划线或其他符号连接，采用驼峰式命名规则。例如：

CreateNFT

GetTransactionHistory

ValidateUserInput

3）变量命名，变量名使用小写字母开头的英文命名，后续单词首字母大写（驼峰式），便于区分。例如：

userInput

transactionAmount

nftMetadata

4）常量命名，常量使用全大写字母，单词之间以下划线分隔。例如：

MAX\_TRANSACTION\_LIMIT

DEFAULT\_NFT\_PRICE

5）接口命名，接口命名以I开头，后接功能性描述的单词，首字母大写。例如：

IUserService

INFTMarketplace

## 1.3 术语和缩写词

1）NFT：非同质化代币，代表数字资产的唯一性和所有权。

2）区块链：一种分布式账本技术，用于记录和存储交易信息。

3）智能合约：一种自动执行的合约，允许在满足特定条件时自动完成交易。

## 1.4 参考资料

[1] 中央财经大学. 区块链技术及其应用[M]. 北京：中央财经大学出版社，2022

[2] 李华. NFT市场研究与发展趋势[J]. 电子商务, 2023, 35(3): 12-15.

## 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 | 张三 | 2023-10-21 | 文档创建 |
| V1.1 | 李四 | 2023-11-19 | 增加了需求分析内容 |
| V1.2 | 王二 | 2023-12-05 | 优化了数据库设计结构 |
| V1.3 | 花俊毅 | 2024-09-25 | 修订软件需求规格说明书 |

**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 | Intel(R) Core(TM) i5-13500H 2.60 GHz |
| 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 11 家庭中文版22H2 |
| 数据库 | MySQL-8.4.3 |

## 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 |
| NFT管理系统 | 支持用户创建、查看、编辑和管理NFT，实现NFT的上架与下架功能。 |
| 交易管理系统 | 提供NFT的购买和出售功能，支持查看交易记录，实现安全的交易流程。 |
| 系统管理系统 | 管理员可通过该系统进行用户信息审核、NFT审核和系统配置管理。 |
| 数据分析系统 | 提供实时交易数据分析、平台数据统计，支持图表展示交易动态和趋势。 |

## 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 | 用户从游客注册为会员，填写注册信息并通过验证。 |
| 登录系统 | 已注册用户输入账号和密码登录系统。 |
| 找回密码 | 会员丢失密码后，通过身份验证重新设置密码。 |
| 查看个人信息 | 会员登录后可查看其个人信息和账户信息。 |
| 修改个人信息 | 会员登录后对个人资料进行管理，包括昵称、联系方式等。 |
| 确认订单并付款 | 会员提交订单后，确认订单内容并完成付款。 |
| 创建NFT | 会员创建NFT，填写NFT信息并上传相关文件。 |
| 浏览NFT市场 | 用户浏览NFT市场上的NFT，查看详细信息。 |
| 购买NFT | 用户选择NFT并进行购买交易，支付成功后NFT所有权转移。 |
| 出售NFT | 会员将拥有的NFT设定价格后上架市场，等待其他用户购买。 |
| 查看交易记录 | 会员可以查看个人的NFT交易记录，包括购买和出售信息。 |
| 管理用户信息 | 管理员管理平台用户信息，审核注册信息、禁用违规用户。 |
| 审核NFT | 管理员审核用户上传的NFT，确保符合平台规则。 |
| 数据分析和统计 | 系统对平台交易数据进行统计分析，生成交易报告和趋势图表。 |
| 系统设置管理 | 管理员可进行系统设置和配置，包括安全设置、通知管理等。 |

**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 | User | 用户信息表，用于存储用户的注册信息及基本资料 |
| 2 | Transaction | 交易记录表，记录NFT的买卖交易信息 |

## 3.2 数据库中的表关系

数据库中的表关系如图3-1所示。

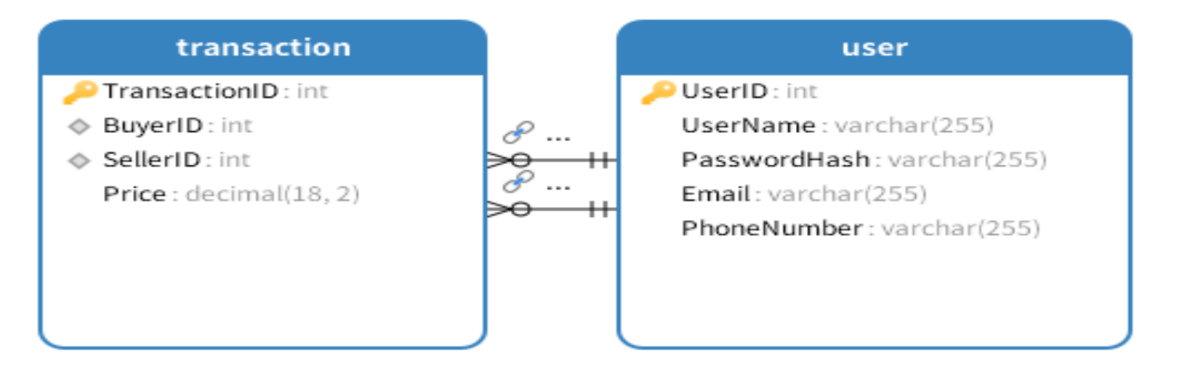


图 3-1 数据库表关系图

## 3.3 数据库表的详细清单

**3.3.1 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 |
| UserID | 用户ID | INT | 否 | 主键 |
| UserName | 用户名 | varchar(50) | 否 | / |
| PasswordHash | 密码哈希 | varchar(255) | 否 | / |
| Email | 电子邮箱 | varchar(100) | 否 | / |
| PhoneNumber | 电话号码 | varchar(15) | 是 | / |

**3.3.2 交易记录表**

交易记录如表3-3所示。

表 3-3 交易记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |
| TransactionID | 交易ID | INT | 否 | 主键 |
| BuyerID | 买家ID | INT | 否 | 外键 (User) |
| SellerID | 卖家ID | INT | 否 | 外键 (User) |
| Price | 价格 | DECIMAL(18,2) | 否 | / |

**4 功能模块设计**

## 4.1 NFT管理系统设计

**4.1.1 类设计**

NFT管理系统主要用于处理与NFT相关的操作，如创建、查看、修改、删除NFT以及管理NFT的交易等。该系统允许会员进行NFT的购买、管理和展示等操作，类图设计如下：

1）NFT 类：用于表示NFT对象，包含NFT的基本信息，如ID、名称、描述、价格、所有者等。

2）用户 类：用于表示用户对象，包含用户的基本信息及与NFT的交互信息，如购买记录、NFT拥有记录等。

3）NFT管理 类：用于管理NFT的创建、编辑、删除及其状态变化。

4）交易记录 类：记录每一笔NFT交易的信息，如买方、卖方、交易价格、时间等。

## 4.2 交易管理系统设计

**4.2.1 类设计**

该子系统主要处理NFT的买卖交易及相关的业务操作，包括购买、支付、订单生成和订单状态管理等，类图设计如下：

1）订单 类：用于表示一笔NFT交易订单，包含订单的基本信息，如订单号、NFT项、交易金额、买卖双方信息、支付状态等。

2）支付 类：处理支付操作，记录支付信息，包括支付方式、支付状态等。

3）买卖双方 类：分别代表买方和卖方，保存与交易相关的用户信息和交易历史。

4）交易历史 类：记录所有已完成的交易信息，并用于生成交易报告或查询交易详情。

## 4.3 系统管理系统设计

**4.3.1 类设计**

系统管理子系统主要包括管理用户订单的功能，如订单提交、查看、取消、评价、确认订单等，类图设计如下：

1）订单管理 类：处理订单的生成、查询、更新及删除等操作。

2）订单状态 类：表示订单的各个状态，如待付款、已付款、已发货、已完成等。

3）用户管理 类：管理平台用户的注册、登录、权限控制等功能。

4）评价系统 类：处理订单的用户评价，允许用户对已完成的订单进行评分和评论。

## 4.4 数据分析系统设计

**4.4.1 类设计**

数据分析子系统主要用于分析交易数据、用户行为数据以及其他系统性能相关的数据。它的功能包括生成报告、统计信息、分析用户趋势等，类图设计如下：

1）交易数据 类：用于收集和存储交易记录，提供对交易数据的详细分析，如交易频率、交易金额等。

2）用户行为 类：记录用户在平台上的行为数据，如浏览记录、购买记录、活跃度等。

3）分析报告 类：基于数据分析结果生成可视化的报告，帮助平台决策和优化运营。

4）趋势分析 类：用于分析数据趋势，如用户增长趋势、销售趋势等，预测未来的业务发展方向。

**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 网站设计母板

本系统在设计前端页面时使用的“母版”的设计思想，即系统绝大多数页面的布局都继承自该母版页面，也就是说，系统的每个页面的头部和左侧的布局和功能完全相同，只是中间部分会有布局和功能上的变化。如图5-1所示。

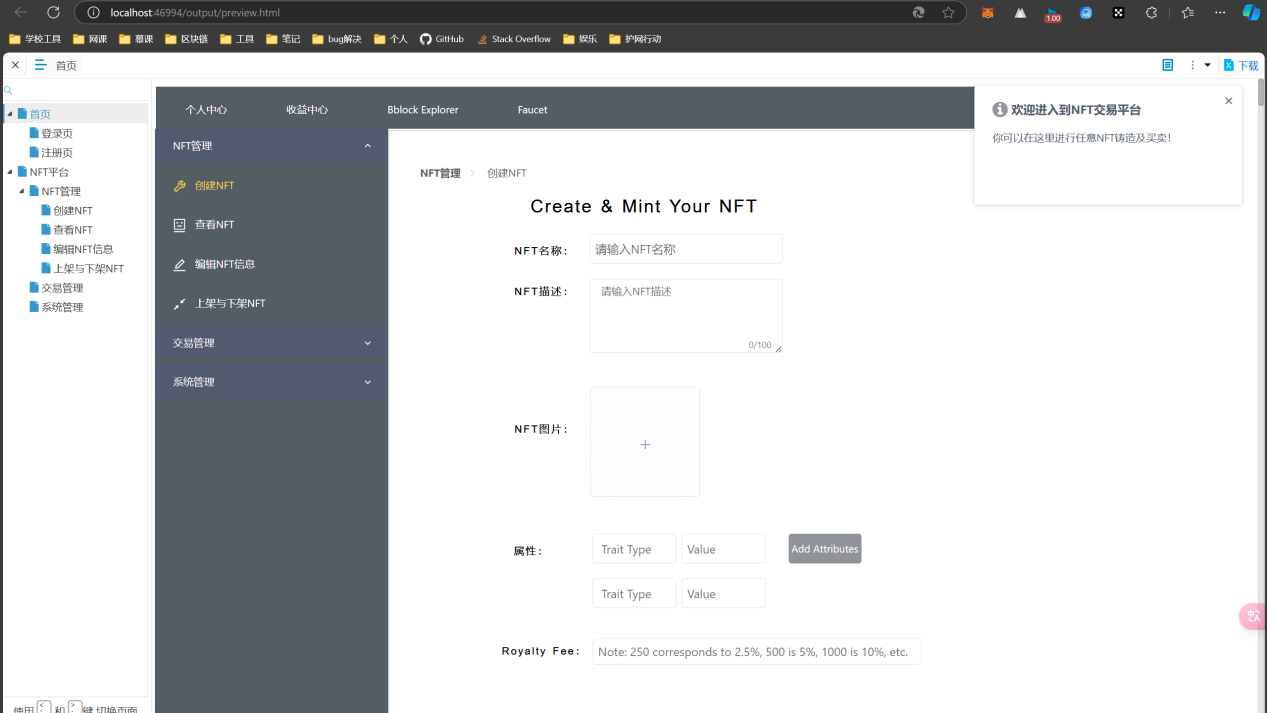


图 5-1 网站设计母版图

因此，本节之后其他功能模块的设计介绍将不包括头部和左侧的设计功能。

**5.1.1 页面组成**

(1)头部：包括上中下两部分，上部包括欢迎信息和登录注册链接（游客可见）或注销链接（已登录用户可见）；中部包括网站 Logo、网站全局搜索框、个人账户按钮、收益中心按钮，下部为区块链浏览器。

(2)左侧：左侧主要包括一些系统功能模块跳转的按钮 。

**5.1.2 调用描述**

游客在使用系统时，点击头部的登录超链接或注册超链接，将跳转到登录页面/login/。

用户已注册后，点击头部的注销超链接时，系统将会注销已登录的用户信息，同时弹出“已注销”通知，页面恢复为游客使用状态。

点击头部的 Logo，将跳转到系统主页/。

点击头部的个人中心查看按钮，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“收益中心”按钮，将跳转到/income/，以查看用户当前收益信息。

点击导航栏最左侧的主页按钮，将跳转到系统主页/。

点击导航栏中的“创建NFT”按钮，将跳转到/mint，该页面将提供创建自定义NFT的界面。

点击导航栏中的“更多”按钮，将弹出以下拉菜单，该下拉菜单包括“Faucet”、“Block Explorer”等按钮。

## 5.2 主页: /

**5.2.1 参数**

主页不需要特殊的请求参数。

**5.2.2 调用背景**

主页是系统的默认页面，游客和已登录用户在访问系统时会首先进入主页。该页面提供了平台的主要介绍、当前热门NFT、最新交易信息等，为用户提供系统概览和快速导航。

**5.2.3 页面组成**

主页由以下主要部分组成：

1）Banner 区域：显示平台的欢迎标语或活动宣传图，位于页面顶部。

2）热门NFT：展示当前交易市场中最受欢迎的NFT，用户可点击查看详情。

3）最新交易：实时展示NFT的最新交易记录，点击可查看交易详情。

4）推荐收藏：系统根据用户的偏好和历史记录推荐的NFT列表。

5）底部导航：提供“关于我们”、“服务条款”、“隐私政策”等链接。

**5.2.4 调用描述**

1）游客或用户打开系统主页，默认显示当前平台的热门NFT、最新交易信息和推荐收藏。

2）用户点击热门NFT中的某个项目，将跳转到该NFT的详情页 /nft/{id}/。

3）点击最新交易中的某条记录，将跳转到该交易的详情页面 /transaction/{id}/。

4）点击底部导航中的“关于我们”链接，跳转到系统介绍页面 /about/。

**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 Vue3 框架配合 Element-UI 和其他组件库来搭建前端页面。通过 Vue-CLI 4.5.0 脚手架工具进行项目结构和界面路由的构建。页面设计和交互精心设计，力求让用户操作流畅、体验友好。例如：

1）用户注册：用户可以通过填写注册表单并点击“提交”按钮完成注册。

2）登录验证：用户在登录页面输入账户信息，系统调用后端 API 进行验证，并在验证成功后跳转至用户主页。

3）NFT详情页：用户点击某个NFT项目时，页面通过动态路由加载该NFT的详细信息，包括持有人、交易记录、价格等。

## 6.2 外部接口

在数据存储和数据处理方面，本项目涉及多种外部接口，以确保系统的高效性和可靠性。具体接口设计如下：

1）后端接口：项目后端使用 Spring Boot 2.4.5 框架搭建，与 MySQL Server 8.4.3 进行数据存储和交互。所有数据操作通过 RESTful API 进行，确保前后端之间的数据交互流畅。

用户数据接口：通过 /api/users/ 提供用户的增删查改操作接口。

NFT数据接口：通过 /api/nfts/ 实现NFT的创建、查询、更新和删除。

交易数据接口：通过 /api/transactions/ 记录每笔NFT交易的数据，提供交易历史查询。

2）邮件发送接口：使用 SMTP 协议和 163 邮箱服务器进行邮件通知功能。SMTP 通过系统配置文件设置，所有邮件均通过 163 服务器的安全连接进行发送。

账户激活邮件：在用户注册后，系统自动发送包含激活链接的邮件。

交易通知：当交易完成后，系统会向交易双方发送交易确认邮件。

## 6.3 内部接口

系统各模块均通过内部接口进行数据交互，实现模块间的高效通信。具体设计如下：

1）ORM 数据交互接口：系统采用 ORM 框架管理数据库连接和查询操作。各模块在处理用户请求时，通过 ORM 模块对 MySQL 数据库进行增删查改操作。例如，用户模块通过 ORM 框架实现用户数据的增删改查操作，确保数据操作的稳定性和一致性。

2）Session 管理：用户登录时，系统生成 Session 数据并进行验证和管理。在页面跳转和请求参数传递时，通过 Session 和 GET 请求参数保证用户操作的连贯性和安全性。

3）参数传递接口：页面跳转或数据提交时，系统通过 GET 请求和表单提交实现参数传递。例如：用户点击 NFT 项目后，系统通过 GET 请求将 NFT 的 ID 传递到详情页面并显示相关数据。在提交订单时，系统通过表单提交订单数据至后端，完成支付和交易记录更新。

**7 角色授权设计**

本项目的使用角色有三类：游客、普通用户（会员）、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 | ● | ● | ○ |
| 注册与登录 | ○ | ● | ● |
| NFT 创建和上架 | ● | ● | ○ |
| 交易管理 | ● | ● | ○ |
| 系统配置 | ● | ○ | ○ |
| 数据分析与报表 | ● | ○ | ○ |
| ●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

1）输入有效性和安全性检查：对游客和会员输入的所有内容进行有效性和安全性检查。通过前端和后端验证机制，确保数据符合预期格式，减少错误输入的发生几率。

2）异常捕获与统一提示：系统对程序运行中的异常情况进行捕获，并以统一的格式向用户（游客和会员）展示错误提示信息。该提示将使用用户友好的语言，以便用户理解问题的原因并采取相应措施。

3）访问权限控制：当用户（游客或会员）尝试访问权限以外的资源时，系统将用户重定向至统一的错误提示页面，并告知其没有相应权限。

## 8.2 故障预防与补救

1）权限控制机制：系统采用统一的权限控制机制，确保不同角色只能访问与其权限相对应的功能和数据，进一步保障系统的安全性和稳定性。

2）数据加密：对涉及用户隐私和系统核心数据进行加密处理，确保数据在传输和存储过程中不被非法访问，减少潜在的黑客攻击影响。

3）定期备份：实施定期的数据库备份策略，包括海量备份和增量备份，以便在系统发生故障时迅速恢复数据，减少数据丢失的风险。

## 8.3 系统维护设计

1）模块化和分层设计：在系统编码过程中采用模块化和分层设计原则，将系统功能分解为独立的模块，提高模块内部的内聚性，降低模块之间的耦合度，从而增强系统的可读性和可维护性。

2）结合面向对象的设计思想：在功能模块划分上，尽量符合面向对象的设计思想，确保系统架构简洁、逻辑清晰，使后续的功能扩展和维护更加便捷。

3）注释与命名规范：编码时遵循命名规范，为标识符赋予清晰的含义，并在必要处添加适量注释，以帮助后续的开发人员和维护人员理解代码逻辑，提升代码的可维护性。