Họ Tên: Lê Thị Trà  
Lớp:TH27.14   
MSV:2722230374

**1. Giới thiệu**

**1.1. Mục đích đề tài**

Blockchain là một cơ chế lưu trữ dữ liệu an toàn, minh bạch, không thể bị thay đổi. Mục đích của bài toán này là mô phỏng các chức năng của Blockchain, bao gồm:

* **Giao dịch**: Tạo và xác minh giao dịch
* **Mining (Đào khối)**: Cơ chế Proof of Work (PoW)
* **Minting**: Tạo coin mới
* **Phí giao dịch (Gas Fee)**: Trả phí xử lý giao dịch
* **Phần thưởng Mining**: Những người mining nhận thưởng bằng coin

**1.2. Công nghệ sử dụng**

* **Node.js**: Chạy mã JavaScript trên backend
* **Thư viện crypto**: Mã hóa SHA-256
* **Thư viện elliptic**: Ký và xác minh giao dịch

**2. Cấu trúc Blockchain**

Blockchain gồm nhiều khối (Block), mỗi khối là một danh sách các giao dịch và liên kết với nhau bằng mã băm (Hash).

**2.1. Lớp Block**const crypto = require("crypto");

SHA256 = message => crypto.createHash("sha256").update(message).digest("hex");

class Block {

constructor(timestamp = "", data = []) {

this.timestamp = timestamp;

this.data = data;

this.hash = this.getHash();

this.prevHash = "";

this.nonce = 0;

}

getHash() {

return SHA256(this.prevHash + this.timestamp + JSON.stringify(this.data) + this.nonce);

}

mine(difficulty) {

while(!this.hash.startsWith(Array(difficulty + 1).join("0"))) {

this.nonce++;

this.hash = this.getHash();

}

}

}

* **getHash()**: Sinh mã băm SHA-256.
* **mine(difficulty)**: Cơ chế Proof of Work (PoW) để tìm giá trị nonce hợp lệ.

**3. Quản lý Blockchain**

Blockchain quản lý danh sách các khối và giao dịch.

**3.1. Lớp Blockchain**

class Blockchain {

constructor() {

this.chain = [new Block(Date.now().toString(), [])];

this.difficulty = 1;

this.transactions = [];

this.reward = 297;

}

getLastBlock() {

return this.chain[this.chain.length - 1];

}

addBlock(block) {

block.prevHash = this.getLastBlock().hash;

block.hash = block.getHash();

block.mine(this.difficulty);

this.chain.push(block);

}

}

* **addBlock()**: Thêm khối mới vào Blockchain sau khi mining.

**4. Quản lý giao dịch**

**4.1. Lớp Transaction**

class Transaction {

constructor(from, to, amount, gas = 0) {

this.from = from;

this.to = to;

this.amount = amount;

this.gas = gas;

}

}

* **from**: Địa chỉ người gửi.
* **to**: Địa chỉ người nhận.
* **amount**: Số tiền giao dịch.
* **gas**: Phí giao dịch.

**4.2. Ký giao dịch**

const EC = require("elliptic").ec;

const ec = new EC("secp256k1");

class Transaction {

sign(keyPair) {

this.signature = keyPair.sign(SHA256(this.from + this.to + this.amount + this.gas), "base64").toDER("hex");

}

}

* **Ký giao dịch**: Dùng khóa riêng ECC.

**5. Mining và Phần thưởng**

**5.1. Mining giao dịch**

Blockchain.prototype.mineTransactions = function(rewardAddress) {

const rewardTransaction = new Transaction("MINT", rewardAddress, this.reward);

this.addBlock(new Block(Date.now().toString(), [rewardTransaction, ...this.transactions]));

this.transactions = [];

};

* **mineTransactions()**: Thêm giao dịch thưởng vào Blockchain.

**6. Kiểm tra số dư**

Blockchain.prototype.getBalance = function(address) {

let balance = 0;

this.chain.forEach(block => {

block.data.forEach(transaction => {

if (transaction.from === address) {

balance -= transaction.amount + transaction.gas;

}

if (transaction.to === address) {

balance += transaction.amount;

}

});

});

return balance;

};

* **getBalance()**: Kiểm tra số dư tài khoản.

**7. Kết luận**

* Hệ thống hoạt động đầy đủ các tính năng cơ bản của Blockchain.
* Có thể mở rộng với Smart Contract, lưu trữ MongoDB.