

|  |  |
| --- | --- |
| 题目 | 基于区块链的学分认证系统 |
| 专业 | 软件工程（区块链方向） |
| 班级 | B190601 |
| 姓名 | 陈超荣 |
| 学号 | 2019000101180 |

|  |
| --- |
| 本人承诺本作业是本人独立撰写完成的。凡涉及他人的观点和材料，均做了注释，如有抄袭或其他学术不端行为，本人承担由学术不端行为所导致的相应责任和处分。 |

# 目录

[1 目录 0](#_Toc60601993)

[1. 概述 3](#_Toc60601994)

[1.1 背景 3](#_Toc60601995)

[1.2 风险、问题 3](#_Toc60601996)

[2 系统分析与设计 4](#_Toc60601997)

[2.1 系统总结结构 4](#_Toc60601998)

[2.2 开发环境的配置 6](#_Toc60601999)

[2.3 运行环境的配置 6](#_Toc60602000)

[2.4 系统功能描述 7](#_Toc60602001)

[2.4.1 功能汇总 7](#_Toc60602002)

[2.4.2 授权系统 7](#_Toc60602003)

[2.4.3 学分加减系统 8](#_Toc60602004)

[2.4.4 代币转账销毁添加系统 9](#_Toc60602005)

[2.4.6查询系统 11](#_Toc60602006)

[2.5 系统功能流程 13](#_Toc60602007)

[3 系统实现与测试 13](#_Toc60602008)

[3.1 系统采用的关键技术 13](#_Toc60602009)

[3.2 关键程序流程 13](#_Toc60602010)

[3.3 关键代码分析 15](#_Toc60602011)

[3.4 接口测试用例 16](#_Toc60602012)

[3.5 用户界面测试的检查表 19](#_Toc60602013)

[4 系统界面及操作 20](#_Toc60602014)

[4.1 系统访问与登录 20](#_Toc60602015)

[4.2 用户注册 20](#_Toc60602016)

[4.3 设置管理员 21](#_Toc60602017)

[4.4 设置相关部门 22](#_Toc60602018)

[4.5 添加学分 23](#_Toc60602019)

[4.6 扣除学分 24](#_Toc60602020)

[4.7 查询加分记录 25](#_Toc60602021)

[4.8 查询扣分记录 25](#_Toc60602022)

[4.9 查询发行代币记录 26](#_Toc60602023)

[4.10 查询学分 26](#_Toc60602024)

[4.11 查询相关部门 27](#_Toc60602025)

[4.12 查询管理员 27](#_Toc60602026)

# 概述

## 背景

目前正在使用的系统：

现有的学生学分记录信息存储在高校的中央数据库中，而系统管理员拥有可以编辑、修改或删除记录的权利。这使得学分的任意篡改成了可能。学生无法知道学分增减情况。各相关部门信息核对繁琐。

现存的问题：

当前，学生并不能知晓自身学分的多少，学生对自己的信息没有知情权。假如相关部门渎职，在本应该添加学分的时候，没有进行添加。学生无法知晓此事，也无法获取证据来进行申诉。学生对自己的信息没有监督权。学分的用处过于单一，学分只能用来作为毕业的其中一个指标，价值极低。

方案：基于区块链的学分认证系统

理由：

 区块链是一种点对点的分布式账本，用于存储信息并跟踪交易。区块链社区的每个成员都有自己的信息副本。随后将信息记录到称为块的单元中，并通过强大的加密技术进行保护，从而创建数据链。区块链系统架构不允许更改块，因此每个动作和事件都可以追溯到其起源。

​ 区块链可以存储有关各方之间的协议，其凭证，交易以及以数字形式呈现的任何其他信息的数据。由于此信息是分布式且高度安全的，因此任何欺诈活动的尝试都可以被区块链社区的成员看到。这为区块链集成到的任何类型的生态系统创建了信任和透明性。

学生可以知晓自己学分数值，并且能够查询加分与扣分的记录。学生能够对学校的行为进行监督，学校的行为会被保存在区块链之中，并且不能被篡改。学生如果发现学分异常的数据，可以与学校进行反馈处理。

## 风险、问题

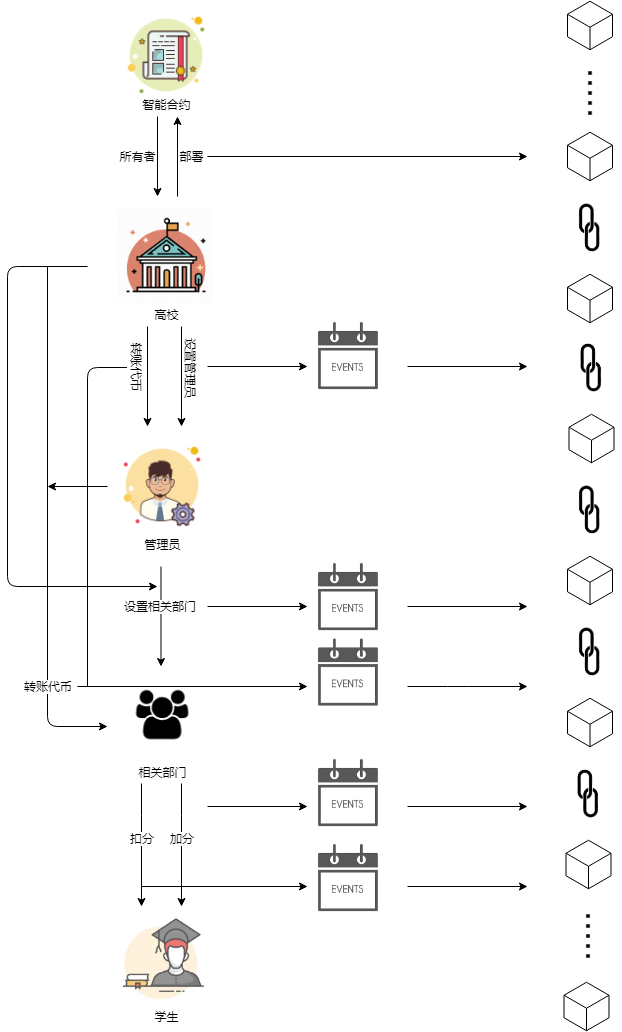
目前基于区块链的学分认证仍然会存在一些风险和问题。

其一，区块链技术无法保证渎职问题，高校要对管理员与相关部门的操作行为与线下的事实进行确认。

其二、要将数据重新整理导入新的系统需要进行大量的工作。费时费力。

# 系统分析与设计

## 系统总结结构



## 开发环境的配置

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 处理器 | 英特尔 Core i7-8750H @ 2.20GHz 六核 | 英特尔 Core i5-8750H @ 2.20GHz 六核 |
| 操作系统 | Window 10 64位 | Window 10 32位 |
| 主板 | 16 GB ( 海力士 DDR4 2666MHz / 金士顿 DDR4 2400MHz ) | 金士顿 DDR4 2400MHz |
| 内存 | 16GB | 8GB |
| 主硬盘 | 西数 WDC PC SN520 SDAPNUW-256G-1002 ( 256 GB / 固态硬盘 ) | 西数 WDC PC SN520 SDAPNUW-256G-1002 ( 256 GB / 固态硬盘 ) |
| 显卡 | GTX 1060 | GTX 1060 |
| 显示器 | 京东方 BOE0747 ( 15.5 英寸 ) | 京东方 BOE0747 ( 15.5 英寸 ) |
| 软件 | Remix  Nodejs14.4.0  Geth1.9.23  MetaMask8.1.9  Webstorm(编译器) | Remix  Nodejs12.15.3  Geth1.9.23  MetaMask8.1.9  Webstorm(编译器) |

## 运行环境的配置

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 处理器 | 英特尔 Core i7-8750H @ 2.20GHz 六核 | 英特尔 Core i5-8750H @ 2.20GHz 六核 |
| 操作系统 | Window 10 64位 | Window 10 32位 |
| 主板 | 16 GB ( 海力士 DDR4 2666MHz / 金士顿 DDR4 2400MHz ) | 金士顿 DDR4 2400MHz |
| 内存 | 16GB | 8GB |
| 主硬盘 | 西数 WDC PC SN520 SDAPNUW-256G-1002 ( 256 GB / 固态硬盘 ) | 西数 WDC PC SN520 SDAPNUW-256G-1002 ( 256 GB / 固态硬盘 ) |
| 显卡 | GTX 1060 | GTX 1060 |
| 显示器 | 京东方 BOE0747 ( 15.5 英寸 ) | 京东方 BOE0747 ( 15.5 英寸 ) |
| 软件 | Nodejs14.4.0  Geth1.9.23  MetaMask8.1.9 | Nodejs14.4.0  Geth1.9.23  MetaMask8.1.9 |

## 系统功能描述

### 功能汇总

|  |  |
| --- | --- |
| 授权系统 | |
| 模块名称 | 功能简述 |
| *设置管理员* | *合约所有者设置指定账户为管理员* |
| *设置相关部门* | *合约所有者或者管理员设置指定账户为相关部门* |
| 学分加减系统 | |
| 模块名称 | 功能简述 |
| *添加学分* | *为指定账户添加学分* |
| *扣除学分* | *扣除指定账户的学分* |
| 代币转账销毁添加系统 | |
| 模块名称 | 功能简述 |
| *代币转账* | *代币转账参数： 目标账户地址、转账数额、转账原因* |
| *代币销毁* | *代币销毁参数：销毁代币的账户地址、销毁数额、销毁原因* |
| *代币添加* | *代币添加参数：增加代币的数额* |
| *查询系统* | |
| *模块名称* | *功能描述* |
| *查询代币名称* | *查询此合约代币的名擦* |
| *查询代币符号* | *查询此合约代币的符号* |
| *查询代币位数* | *查询此合约代币的位数* |
| *查询代币总发行量* | *查询此合约代币的总发行量* |
| *查询账户的余额* | *查询指定账户的代币余额* |

### 授权系统

|  |  |
| --- | --- |
| 模块名称 | *设置管理员* |
| 功能描述 | *合约所有者设置指定账户为管理员* |
| 接口与属性 | *1.function setAdministrators(address \_AdminAdd,bool \_bool) public {*  *require(\_owner == msg.sender, "ERC20: 非所有者，您没有权限设置管理员");*  *administrators[\_AdminAdd] = \_bool;*  *emit SetAdministrators(msg.sender,\_AdminAdd,\_bool,uint32(block.timestamp));*  *}*   1. *输入参数：账户地址、true/false* 2. *设置管理员成功后，此操作记录到合约SetAdministrators事件中，数据为（方法调用人的地址、目标地址、当前时间戳）* |
| 补充说明 | *1.注意输入参数的数据类型*  *2.第二个参数的是否决定目标账户是否能成为管理员*  *3.权限：操作人的身份须是合约所有者* |

|  |  |
| --- | --- |
| 模块名称 | *设置相关部门* |
| 功能描述 | *合约所有者或者管理员设置指定账户为相关部门* |
| 接口与属性 | *1. function setDepartment(address \_departAdd,string memory \_departName,bool \_bool) public {*  *require(administrators[msg.sender] == true || \_owner == msg.sender, "ERC20: 非管理员或者所有者，您没有权限设置部门");*  *departments[\_departAdd].departmentName = \_departName;*  *departments[\_departAdd].departmentTF = \_bool;*  *emit SetDepartment(msg.sender,\_departAdd,\_bool,uint32(block.timestamp));*  *}*  *2. 输入参数：账户地址、true/false*  *3. 设置相关部门成功后，此操作记录到合约SetDepartment事件中，数据为（方法调用人的地址、目标地址、当前时间戳）* |
| 补充说明 | *1.注意输入参数的数据类型*  *2.第二个参数的是否决定目标账户是否能成为相关部门*  *3.权限：操作人的身份须是合约所有者或者管理员* |

### 学分加减系统

|  |  |
| --- | --- |
| 模块名称 | 添加学分 |
| 功能描述 | *给指定账户添加学分* |
| 接口与属性 | *1.function transfer(address recipient, uint256 amount,string memory reason) public virtual override returns (bool) {*  *\_transfer(\_msgSender(), recipient, amount, reason);*  *return true;*  *}*  *function \_transfer(address sender, address recipient, uint256 amount, string memory reason) internal virtual {*  *require(sender != address(0), "ERC20: transfer from the zero address");*  *require(*  *departments[sender].departmentTF == true ||*  *administrators[sender] == true ||*  *\_owner == sender,*  *"ERC20: 非相关部门，您没有权限转账");*  *require(recipient != address(0), "ERC20: transfer to the zero address");*  *\_beforeTokenTransfer(sender, recipient, amount);*  *\_balances[sender] = \_balances[sender].sub(amount, "ERC20: 转账金额超过余额");*  *\_balances[recipient] = \_balances[recipient].add(amount);*  *emit Transfer(sender, recipient, amount, reason);*  *}*  *2. 输入参数：添加学分账户地址、学分数值、加分原因*  *3. 返回值为 ture*  *4. 转账成功后，此操作记录到合约Transfer事件中，数据为（发送人地址、接收人地址、发送数额、发送原因）* |
| 补充说明 | *权限：操作人的身份须是合约所有者、管理员或者是相关部门* |

|  |  |
| --- | --- |
| 模块名称 | *扣除学分* |
| 功能描述 | *扣除指定账户的学分* |
| 接口与属性 | *1.function burn(address account, uint256 amount,string memory reason) public override virtual returns(bool){*  *\_burn(\_msgSender(),account,amount,reason);*  *}*  *function \_burn(address sender,address account, uint256 amount,string memory reason) internal virtual {*  *require(*  *departments[sender].departmentTF == true ||*  *administrators[sender] == true ||*  *\_owner == sender,*  *"ERC20: 非相关部门，您没有权限销毁");*  *require(account != address(0), "ERC20: burn from the zero address");*  *\_beforeTokenTransfer(account, address(0), amount);*  *\_balances[account] = \_balances[account].sub(amount, "ERC20: 销毁金额超过余额");*  *\_totalSupply = \_totalSupply.sub(amount);*  *emit Burn(sender,account, amount,reason);*  *}*  *2. 输入参数：扣除学分账户地址、学分数值、扣分原因*  *3. 返回值为 ture*  *4. 学分扣除成功后，此操作记录到合约Burn事件中，数据为（操作人地址、扣分账户地址、扣分数额、扣分原因）* |
| 补充说明 | *权限：操作人的身份须是合约所有者、管理员或者是相关部门* |

### 代币转账销毁添加系统

|  |  |
| --- | --- |
| 模块名称 | 代币转账 |
| 功能描述 | *给指定账户转账代币* |
| 接口与属性 | *1.function transfer(address recipient, uint256 amount,string memory reason) public virtual override returns (bool) {*  *\_transfer(\_msgSender(), recipient, amount, reason);*  *return true;*  *}*  *function \_transfer(address sender, address recipient, uint256 amount, string memory reason) internal virtual {*  *require(sender != address(0), "ERC20: transfer from the zero address");*  *require(*  *departments[sender].departmentTF == true ||*  *administrators[sender] == true ||*  *\_owner == sender,*  *"ERC20: 非相关部门，您没有权限转账");*  *require(recipient != address(0), "ERC20: transfer to the zero address");*  *\_beforeTokenTransfer(sender, recipient, amount);*  *\_balances[sender] = \_balances[sender].sub(amount, "ERC20: 转账金额超过余额");*  *\_balances[recipient] = \_balances[recipient].add(amount);*  *emit Transfer(sender, recipient, amount, reason);*  *}*  *2. 输入参数：接受代币账户地址、代币数值、转账原因*  *3. 返回值为 ture*  *4. 转账成功后，此操作记录到合约Transfer事件中，数据为（发送人地址、接收人地址、发送数额、发送原因）* |
| 补充说明 | *权限：操作人的身份须是合约所有者、管理员或者是相关部门* |

|  |  |
| --- | --- |
| 模块名称 | *销毁代币* |
| 功能描述 | *销毁指定账户的代币* |
| 接口与属性 | *1.function burn(address account, uint256 amount,string memory reason) public override virtual returns(bool){*  *\_burn(\_msgSender(),account,amount,reason);*  *}*  *function \_burn(address sender,address account, uint256 amount,string memory reason) internal virtual {*  *require(*  *departments[sender].departmentTF == true ||*  *administrators[sender] == true ||*  *\_owner == sender,*  *"ERC20: 非相关部门，您没有权限销毁");*  *require(account != address(0), "ERC20: burn from the zero address");*  *\_beforeTokenTransfer(account, address(0), amount);*  *\_balances[account] = \_balances[account].sub(amount, "ERC20: 销毁金额超过余额");*  *\_totalSupply = \_totalSupply.sub(amount);*  *emit Burn(sender,account, amount,reason);*  *}*  *2. 输入参数：被销毁代币的账户地址、代币数值、销毁原因*  *3. 返回值为 ture*  *4. 学分扣除成功后，此操作记录到合约Burn事件中，数据为（操作人地址、被销毁代币的账户地址、代币数值、销毁原因）* |
| 补充说明 | *权限：操作人的身份须是合约所有者、管理员或者是相关部门* |

|  |  |
| --- | --- |
| 模块名称 | 增发代币 |
| 功能描述 | *增加代币的总发行量* |
| 接口与属性 | *1.function addTotalSupply(uint256 amount) public returns(bool){*  *require(*  *administrators[\_msgSender()] == true ||*  *\_owner == \_msgSender(),*  *"ERC20: 非管理员，您没有权利增加代币");*  *\_mint(\_msgSender(),amount);*  *return true;*  *}*  *function \_mint(address account, uint256 amount) internal virtual {*  *require(account != address(0), "ERC20: mint to the zero address");*  *\_beforeTokenTransfer(address(0), account, amount);*  *\_totalSupply = \_totalSupply.add(amount);*  *\_balances[account] = \_balances[account].add(amount);*  *emit Add(address(0), account, amount,"增加代币");*  *}*  *2. 输入参数：增发代币的数额*  *3. 返回值为 ture*  *4. 代币增发成功后，此操作记录到合约Add事件中，数据为（空账户地址、调用人地址、增发代币的原因（增加代币））* |
| 补充说明 | 1. *权限：操作人的身份须是合约所有者或者管理员* 2. *代币增发成功之后，增发部分的代币归方法调用人所有* |

### 2.4.6查询系统

|  |  |
| --- | --- |
| 模块名称 | *查询代币名称* |
| 功能描述 | *查询此合约的代币名称* |
| 接口与属性 | *1. function name() public view returns (string memory) {*  *return \_name;*  *}*  *2. 返回值为 代币名称* |
| 补充说明 | *权限：所有人均可以执行此方法* |

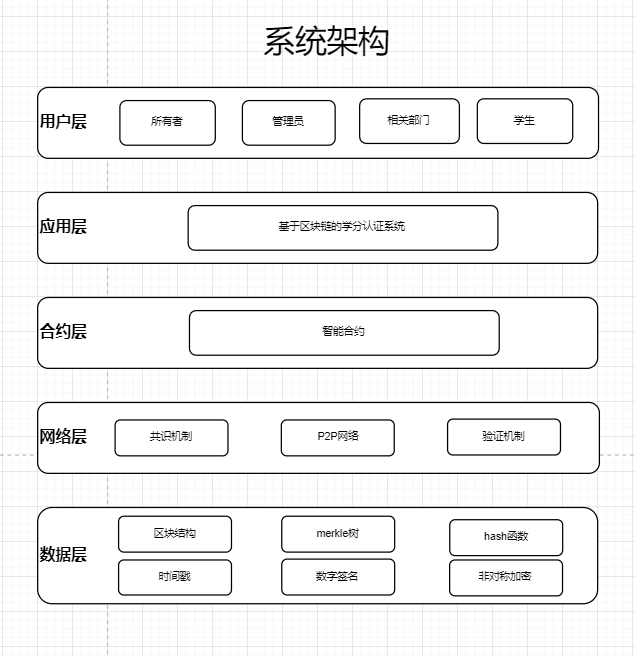
|  |  |
| --- | --- |
| 模块名称 | *查询代币符号* |
| 功能描述 | *查询此合约的代币符号* |
| 接口与属性 | *1. function symbol() public view returns (string memory) {*  *return \_symbol;*  *}*  *2. 返回值为 代币符号* |
| 补充说明 | *权限：所有人均可以执行此方法* |

|  |  |
| --- | --- |
| 模块名称 | *查询代币位数* |
| 功能描述 | *查询此合约的代币位数* |
| 接口与属性 | *1. function decimals() public view returns (uint8) {*  *return \_decimals;*  *}*  *2. 返回值为 代币位数* |
| 补充说明 | *权限：所有人均可以执行此方法* |

|  |  |
| --- | --- |
| 模块名称 | *查询代币总发行量* |
| 功能描述 | *查询此合约的代币总发行量* |
| 接口与属性 | *1. function totalSupply() public view override returns (uint256) {*  *return \_totalSupply;*  *}2. 返回值为 代币总发行量* |
| 补充说明 | *权限：所有人均可以执行此方法* |

|  |  |
| --- | --- |
| 模块名称 | *查询账户代币余额* |
| 功能描述 | *查询指定账户拥有的的代币余额* |
| 接口与属性 | *1. function balanceOf(address account) public view override returns (uint256) {*  *return \_balances[account];*  *}2. 返回值为 代币符号* |
| 补充说明 | *权限：所有人均可以执行此方法* |

## 系统功能流程



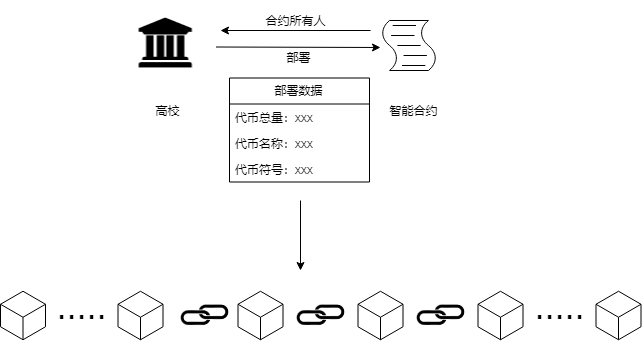
# 系统实现与测试

## 系统采用的关键技术

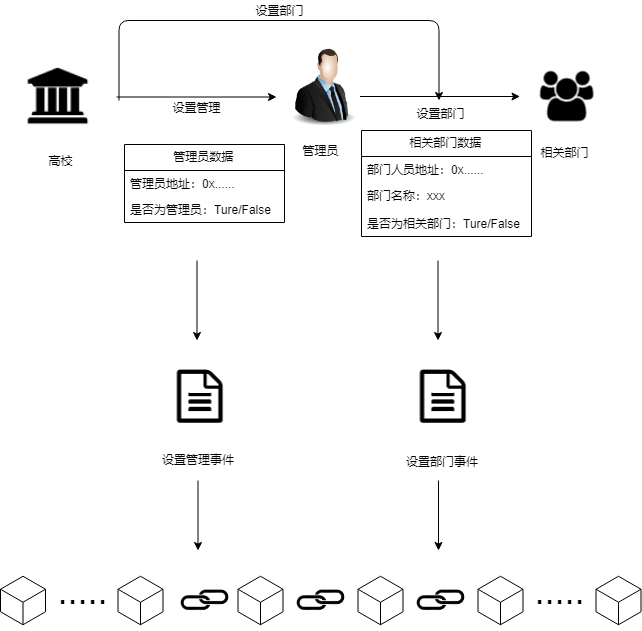
[以太坊，智能合约，web3js]

## 关键程序流程

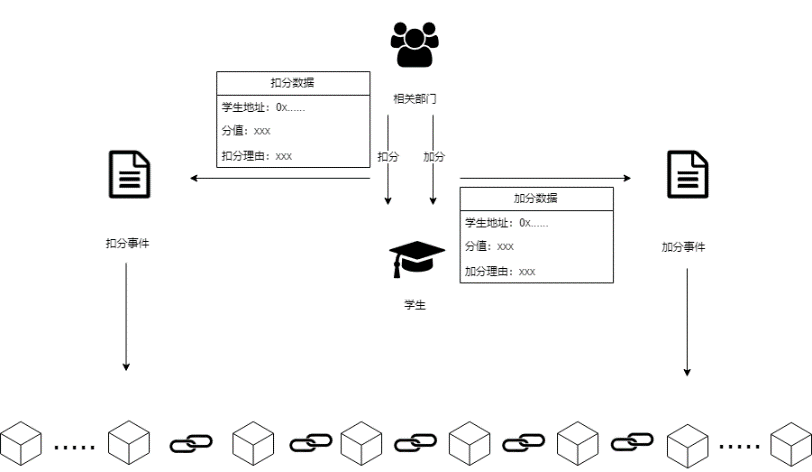
1.合约部署



2. 授权



3.添加删除学分



## 关键代码分析

|  |  |
| --- | --- |
| 程序名称 | *代币转账* |
| 功能描述 | *转账代币的时候设置权限，只有合约所有者、管理员或者相关部门才能转账*  *转账之后，将转账操作记录到合约的时间Transfer中。数据为（代币发送方，代币接收方，代币数值，发送原因）* |
| 代码块 | *function \_transfer(address sender, address recipient, uint256 amount, string memory reason) internal virtual {*  *require(sender != address(0), "ERC20: transfer from the zero address");*  *require(*  *departments[sender].departmentTF == true ||*  *administrators[sender] == true ||*  *\_owner == sender,*  *"ERC20: 非相关部门，您没有权限转账");*  *require(recipient != address(0), "ERC20: transfer to the zero address");*  *\_beforeTokenTransfer(sender, recipient, amount);*  *\_balances[sender] = \_balances[sender].sub(amount, "ERC20: 转账金额超过余额");*  *\_balances[recipient] = \_balances[recipient].add(amount);*  *emit Transfer(sender, recipient, amount, reason);*  *}* |
| 1.防止学生将代币转发给其他人，因为代币的多少决定着学生学分的高低。  2.进行转代币账的操作会被记录到合约的Transfer事件当中。后续可以通过事件进行溯源，滥用权限进行转账加分操作很容易被发现。 | |

|  |  |
| --- | --- |
| 程序名称 | *代币销毁* |
| 功能描述 | *销毁代币的时候设置权限，只有合约所有者、管理员或者相关部门才能销毁*  *销毁之后，将销毁操作记录到合约的时间Burn中。数据为（方法的调用人、销毁代币的账户、销毁数额、销毁原因）* |
| 代码块 | *function \_burn(address sender,address account, uint256 amount,string memory reason) internal virtual {*  *require(*  *departments[sender].departmentTF == true ||*  *administrators[sender] == true ||*  *\_owner == sender,*  *"ERC20: 非相关部门，您没有权限销毁");*  *require(account != address(0), "ERC20: burn from the zero address");*  *\_beforeTokenTransfer(account, address(0), amount);*  *\_balances[account] = \_balances[account].sub(amount, "ERC20: 销毁金额超过余额");*  *\_totalSupply = \_totalSupply.sub(amount);*  *emit Burn(sender,account, amount,reason);*  *}* |
| 1.只有合约所有者、管理员、相关部门才能销毁代币，因为代币的多少决定着学生学分的高低。  2.进行代币销毁的操作会被记录到合约的Burn事件当中。后续可以通过事件进行溯源，滥用权限进行销毁扣分操作很容易被发现。 | |

|  |  |
| --- | --- |
| 程序名称 | *代币增发* |
| 功能描述 | *增发代币的时候设置权限，只有合约所有者或者管理员才能增发代币*  *增发之后，将增发操作记录到合约的时间Add中。数据为（空账户地址、增发代币的账户、增发数额、增发原因）* |
| 代码块 | *function addTotalSupply(uint256 amount) public returns(bool){*  *require(*  *administrators[\_msgSender()] == true ||*  *\_owner == \_msgSender(),*  *"ERC20: 非管理员，您没有权利增加代币");*  *\_mint(\_msgSender(),amount);*  *return true;*  *}*  *function \_mint(address account, uint256 amount) internal virtual {*  *require(account != address(0), "ERC20: mint to the zero address");*  *\_beforeTokenTransfer(address(0), account, amount);*  *\_totalSupply = \_totalSupply.add(amount);*  *\_balances[account] = \_balances[account].add(amount);*  *emit Add(address(0), account, amount,"增加代币");*  *}* |
| 1.只有合约所有者、管理员才能进行增发代币。因为新发行的那部分代币会存入发行人的账户中。  2.进行代币增发的操作会被记录到合约的Add事件当中。后续可以通过事件进行溯源，可以知道是哪个账户发行了代币。 | |

## 接口测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| 接口A  transfer的函数原型 | function transfer(address recipient, uint256 amount, string memory reason) external returns (bool); | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 典型值  {  "address recipient": "0x206e7B4702Df5382f502F0114Ce6c79b40AEfdAD",  "uint256 amount": {  "\_hex": "0x0a"  },  "string reason": "xxx"  } | | True | True |
| 边界值  {  "address recipient": " 0x6d5af3704f8F03B0fB9BF6a5A96a725f29657866",  "uint256 amount": {  "\_hex": "0x0"  },  "string reason": "xxx"  }或  {  "address recipient": " 0x6d5af3704f8F03B0fB9BF6a5A96a725f29657866",  "uint256 amount": {  "\_hex": "<账户代币余额>"  },  "string reason": "xxx"  } | | True | True |
| 异常值1  {  "address recipient": " 0x0000000000000000000000000000000000000000",  "uint256 amount": {  "\_hex": "0x01"  },  "string reason": "xxx"  } | | Error：接收者不能是零账户地址 | Reason provided by the contract: "ERC20: transfer to the zero address". |
| 异常值2  {  "address recipient": "0x206e7B4702Df5382f502F0114Ce6c79b40AEfdAD ",  "uint256 amount": {  "\_hex": "0.1"  },  "string reason": xxx"  } | | Error：类型异常 | Error: invalid number value (arg="", coderType="uint256", value="0.1", version=4.0.47) |
| 异常值3：  {  "address recipient": "0x206e7B4702Df5382f502F0114Ce6c79b40AEfdAD",  "uint256 amount": {  "\_hex": "0xc8"(注：超过账户代币余额)  },  "string reason": "0"  } | | VM error: revert  "ERC20: 转账金额超过余额". | VM error: revert  "ERC20: 转账金额超过余额". |
| 接口B  burn的函数原型 | function burn(address account, uint256 amount,string memory reason) external returns(bool); | | |
| 输入/动作 | | 期望的输出/相应 | 实际情况 |
| 典型值  {  "address account": " 0x6d5af3704f8F03B0fB9BF6a5A96a725f29657866",  "uint256 amount": {  "\_hex": "0x01"  },  "string reason": "xxx"  } | | True | True |
| 边界值  {  "address account": " 0x6d5af3704f8F03B0fB9BF6a5A96a725f29657866",  "uint256 amount": {  "\_hex": "0x0"  },  "string reason": "xxx"  }或  {  "address account": " 0x6d5af3704f8F03B0fB9BF6a5A96a725f29657866",  "uint256 amount": {  "\_hex": "<账户代币余额>"  },  "string reason": "xxx"  } | | True | True |
| 异常值1  {  "address account": " 0x6d5af3704f8F03B0fB9BF6a5A96a725f29657866",  "uint256 amount": {  "\_hex": <数值超过账户代币余额>  },  "string reason": "xxx"  } | | VM error: revert  "ERC20: 销毁金额超过余额". | VM error: revert  "ERC20: 销毁金额超过余额". |
| 异常值2  {  "address account": " 0x0000000000000000000000000000000000000000",  "uint256 amount": {  "\_hex": "0x01"  },  "string reason": "xxx"  } | | 被销毁代币的地址不能是零账户地址 | Reason provided by the contract: "ERC20: burn from the zero address". |
| 异常值3  {  "address recipient": "0x206e7B4702Df5382f502F0114Ce6c79b40AEfdAD",  "uint256 amount": {  "\_hex": "0.1"  },  "string reason": "xxx"  } | | Error：类型异常 | Error: invalid number value (arg="", coderType="uint256", value="0.1", version=4.0.47) |

## 用户界面测试的检查表

|  |  |
| --- | --- |
| 检查项 | 测试人员的类别及其评价 |
| 窗口切换、移动、改变大小时正常吗？ | 正常 |
| 各种界面元素的文字正确吗？（如标题、提示等） | 正确 |
| 各种界面元素的状态正确吗？（如有效、无效、选中等状态） | 正确 |
| 各种界面元素支持键盘操作吗？ | 支持 |
| 各种界面元素支持鼠标操作吗？ | 支持 |
| 对话框中的缺省焦点正确吗？ | 正确 |
| 数据项能正确回显吗？ | 可以 |
| 对于常用的功能，用户能否不必阅读手册就能使用？ | 能 |
| 执行有风险的操作时，有“确认”、“放弃”等提示吗？ | 有 |
| 操作顺序合理吗？ | 合理 |
| 有联机帮助吗？ | 有帮助 |
| 各种界面元素的布局合理吗？美观吗？ | 合理 美观 |
| 各种界面元素的颜色协调吗？ | 协调 |
| 各种界面元素的形状美观吗？ | 美观 |
| 字体美观吗？ | 美观 |
| 图标直观吗？ | 直观 |
|  |  |
| … |  |

# 系统界面及操作

## 系统访问与登录

* **通过网址直接访问**

在地址栏键入访问地址http://127.0.0.1:8080/erc20.html并回车，浏览器中会出现系统登录页面。



## 用户注册

用户注册步骤如下：

* 打开用户注册对话框

打开系统登录页面，点击“注册”按钮，弹出“用户注册”对话框。

* 提交注册流程

输入用户信息，点击“注册”后，提示用户注册流程启动。



## 设置管理员

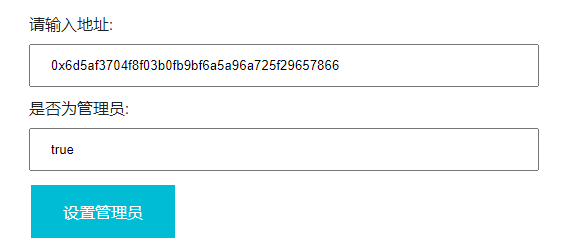
**设置系统的管理员**

1.将账户地址添加到地址栏文本框中。

2.在是否为管理员中填入true，其可获得管理员权限；在是否为管理员中填入false，取消其管理员权限，

* 点击设置管理员按钮，

即可设置管理员信息



## 设置相关部门

**设置系统的相关部门**

1. 将账户地址添加到地址栏文本框中。
2. 将部门名称添加到名称的文本框中
3. 在是否为**相关部门**中填入true，其可获得**相关部门**权限；在是否为**相关部门**中填入false，取消其**相关部门**权限，

* 点击设置**相关部门**按钮，

即可设置**相关部门**信息



## 添加学分

**为学生添加学分**

1.将账户地址添加到地址栏文本框中。

2.将加分数值添加到加分数值的文本框中

3.将加分原因添加到加分原因的文本框中

* 点击设置**加分**按钮，

即可**为学生添加学分**



## 扣除学分

**扣除学生学分**

1.将账户地址添加到地址栏文本框中。

2.将扣分数值添加到扣分数值的文本框中

3.将扣分原因添加到扣分原因的文本框中

* 点击设置扣**分**按钮，

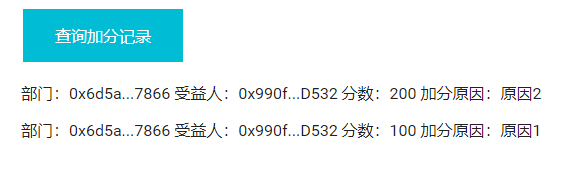
即可**扣除学生学分**



## 查询加分记录

* 点击查询加分记录按钮，

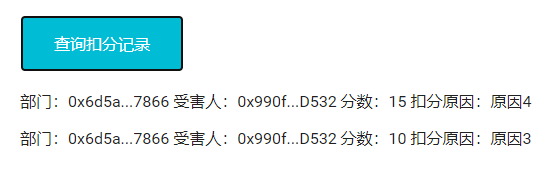
即可获取学分添加记录



## 查询扣分记录

* 点击查询扣分记录按钮，

即可获取学分扣除记录



## 查询发行代币记录

* 点击查询增发代币记录按钮，

即可获取代币增发记录



## 查询学分

**查询指定账户的学分**

将账户地址添加到地址栏文本框中。

* 点击查询**加分**按钮

即可**查询指定账户的学分**



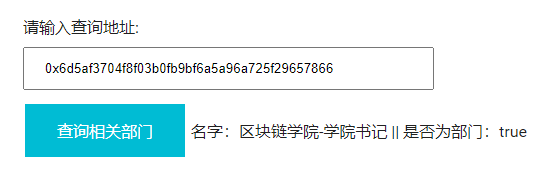
## 查询相关部门

**查询指定账户是否为相关部门**

将账户地址添加到地址栏文本框中。

* 点击查询相关部门按钮

即可**查询指定账户是否为相关部门**



## 查询管理员

**查询指定账户是否为管理员**

将账户地址添加到地址栏文本框中。

* 点击查询管理员按钮

即可**查询指定账户是否为管理员**

