**基于区块链的喜剧人**

**投票系统**

**详**

**细**

**设**

**计**

**说**

**明**

**书**

目录

[1引言 3](#_Toc22189)

[1.1编写目的 3](#_Toc23242)

[1.2背景 3](#_Toc24794)

[2概述 3](#_Toc18209)

[2.1基线 3](#_Toc23511)

[2.2范围 3](#_Toc29767)

[3模块命名规则 4](#_Toc24493)

[4模块汇总 4](#_Toc5641)

[4.1模块汇总表 4](#_Toc1338)

[4.2块关系图 5](#_Toc19721)

[5子系统模块设计 6](#_Toc4331)

[5.1用户管理模块 6](#_Toc8455)

[5.1.1功能描述 6](#_Toc5216)

[5.1.2相关模块 6](#_Toc27262)

[5.1.3类图 6](#_Toc17944)

[5.1.4 User类 7](#_Toc3608)

[5.2信息维护模块 9](#_Toc1945)

[5.2.1功能描述 9](#_Toc16580)

[5.2.2相关模块 9](#_Toc5547)

[5.2.3 实现方法 10](#_Toc368)

[5.2.4 属性 10](#_Toc31649)

[5.3投票模块 11](#_Toc7614)

[5.3.1功能描述 11](#_Toc27082)

[5.3.2相关模块 12](#_Toc295)

[5.3.3 实现方法 12](#_Toc20605)

# 1引言

## 1.1编写目的

此说明书是基于区块链的喜剧人投票系统的详细设计，其编写目的在于：在概要设计的基础上，针对该项目的每个模块、每个功能的实现方法进行详细的设计和划分，按照详细设计说明书的要求，进一步地明确系统结构，为项目开发人员后期的编码工作做准备。

## 1.2背景

在前一阶段的概要设计文档中，已经通过顺序图解决了该系统功能的实现流程，包括系统的各模块间的关系设计等。本文档就是在前一阶段成果的基础上，继续对系统的各模块进行详细的划分和说明。

# 2概述

## 2.1基线

以需求分析和概要设计为基准。

## 2.2范围

本设计文档提供给系统设计开发人员，包括详细设计人员和项目组成员。

# 3模块命名规则

驼峰标识，由单词的全拼组成，除首单词外每个单词的首字母大写，例如：medcineManager

# 4模块汇总

## 4.1模块汇总表

提示：这里模块是指相对独立的软件设计单元，例如对象类、函数包等等。

|  |  |  |
| --- | --- | --- |
| **子系统A** | | |
| 模块名称 | 功能简述 | |
| 注册 | 新用户注册 | |
| 登录 | 可选择传统的用户名密码登陆方式和一键登录方式 | |
| 用户角色配置 | 配置用户的角色信息 | |
| **子系统B** | | |
| 模块名称 | 功能简述 | |
| 查看参赛队伍信息 | 可按类别查看参赛队伍信息，也可定向查找指定参赛队伍的信息 | |
| 查看投票结果 | 可实时查看当前的投票结果 | |
| **子系统C** | | |
| 模块名称 | | 功能简述 |
| 添加选民 | | 授予特定选民投票权 |
| 参赛队伍信息维护 | | 可以对参赛队伍信息进行增删改操作 |
| 投票通道 | | 可以对喜爱的参赛队伍投票 |

## 

## 4.2块关系图

区块链投票系统

投票模块

信息维护模块

用户管理模块

参赛队伍信息维护

投票权维护

投票

投票结果查看

用户登录

用户角色管理

用户注册

# 5子系统模块设计

## 5.1用户管理模块

### 5.1.1功能描述

本系统的资源仅对本系统的用户开放，因而，想要获取本系统的资源，首先应在本系统中完成用户注册或者用户登录功能才可以，另外，本系统的菜单是根据用户角色动态加载的，所以，为不影响用户的使用体验，还应对用户的角色进行相应的配置。

### 5.1.2相关模块

用户管理模块

用户角色管理

用户登录

用户注册

角色查询

角色修改

角色添加

### 5.1.3类图

|  |
| --- |
| User |
| +id：String  +nonce：Integer  +username：String  +password：String  +salt：String  +userAddress：String  +roles：List<String>  +roleList:List<String>  +permissions:List<String> |
| +register()  +login()  +loginByUserName()  +logout()  +getUserInfo()  +addUserRoles() |

### 5.1.4 User类

#### 5.1.4.1方法

##### 5.1.4.1.1方法—用户注册:public Result register()

**5.1.4.1.1.1 功能**

用户注册

**5.1.4.1.1.2 参数**

User类对象：待注册用户的信息

##### 5.1.4.1.2方法—用户登录:public Result loginByUsername()

**5.1.4.1.2.1 功能**

传统的用户名密码登陆方式

**5.1.4.1.2.2 参数**

User类对象：待登录用户的信息

**5.1.4.1.2.3 算法**

采用shiro框架对待登录用户作认证授权。

###### 5.1.4.1.3方法—用户一键登录:public Result login()

**5.1.4.1.3.1 功能**

用户一键登录

**5.1.4.1.3.2 参数**

userAddress：用户的以太坊地址

##### 5.1.4.1.4方法—用户登出:public Result logout()

**5.1.4.1.4.1 功能**

用户登出系统

**5.1.4.1.4.2 参数**

无

##### 5.1.4.1.5方法—获取用户信息:public Result getUserInfo()

**5.1.4.1.5.1 功能**

用于向前端发送当前用户的角色、权限以及一些基础信息

**5.1.4.1.5.2 参数**

无

##### 5.1.4.1.6方法—配置用户角色信息:public Result addUserRole()

**5.1.4.1.6.1 功能**

用于添加或修改用户的角色信息

**5.1.4.1.6.2 参数**

User类：前端传来的用户信息，包含基本信息和角色等信息

#### 5.1.4.2属性

private String id;

private Integer nonce =(Integer)(int)(random()\*1000000);

private String userAddress;

private String username;

private String password;

private String salt;

private List<String> roles;

private List<String> permissions;

private List<String> roleList;

**5.1.4.2.1 属性id**

User类的唯一标识

**5.1.4.2.2 属性nonce**

随机数，用于一键登录时前端要请求meta Mask签名时的签名消息内容

**5.1.4.2.3 属性userAddress**

用户注册时绑定的以太坊地址

**5.1.4.2.4 属性username**

用户的用户名

**5.1.4.2.5属性password**

对用户的明文密码加盐操作后的密文

**5.1.4.2.6属性salt**

用于对用户明文密码进行加盐操作的盐

**5.1.4.2.7 属性roles**

User类对象所拥有的角色列表

**5.1.4.2.8属性permissions**

User类对象所拥有的权限列表

**5.1.4.2.9 属性roleList**

User类的所有角色列表

## 5.2信息维护模块

### 5.2.1功能描述

负责喜剧人参赛队伍信息的维护，其中包括录入、修改、删除、查询操作；本系统用户的初始状态中，并不具备投票权，因而，为不影响用户的投票体验，需要为用户添加投票权。

### 5.2.2相关模块

信息维护模块

参赛队伍信息维护

投票权维护

信息查询

信息删除

信息修改

信息添加

**5.2.3 实现方法**

该模块的功能均需要调用以太坊智能合约中的函数来实现，具体的函数如下。

**5.2.3.1 方法—添加或修改参赛队伍信息:public addCandidate()**

**5.2.3.1.1 功能**

用于向智能合约中添加或修改参赛队伍信息

**5.2.3.1.2 参数**

Proposal结构体：参赛队伍信息

**5.2.3.2 方法—删除参赛队伍信息:public deleteCandidate()**

**5.2.3.2.1 功能**

用于将智能合约中指定的参赛队伍信息删除

**5.2.3.2.2 参数**

team：参赛队伍名称；names：参赛队伍队员姓名

**5.2.3.3 方法—查询参赛队伍信息:public findAll()**

**5.2.3.3.1 功能**

按参赛队伍的类别查询所有参赛队伍信息

**5.2.3.3.2 参数**

kind：参赛队伍的类别

**5.2.3.4 方法—添加投票权:public giveRightToVote()**

**5.2.3.4.1 功能**

给指定选民添加投票权（仅1票）

**5.2.3.4.2 参数**

address：选民的以太坊地址

**5.2.4 属性**

struct Proposal {

bytes32 team;

bytes32 names;

bytes32 masterWork;

bytes32 work;

bytes32 itsType;

bytes32 cover;

bytes32 discription;

uint voteCount; // number of accumulated votes

}

**5.2.4.1 属性team**

参赛队伍名称

**5.2.4.2 属性names**

参赛队伍的队员姓名

**5.2.4.3 属性masterWork**

参赛队伍的代表作

**5.2.4.4 属性work**

参赛队伍的参赛作品名称

**5.2.4.5 属性itsType**

参赛队伍的参赛作品的类型

**5.2.4.6 属性cover**

参赛队伍的海报的url

**5.2.4.7 属性discription**

参赛队伍的个性宣言

**5.2.4.8 属性voteCount**

参赛队伍的得票数

## 5.3投票模块

### 5.3.1功能描述

本模块通过调用智能合约中的相应函数，完成对喜剧人参赛队伍的投票功能以及查询当前得票最高的队伍名称的功能。

### 5.3.2相关模块

投票模块

投票结果查看

投票

**5.3.3 实现方法**

**5.3.3.1 方法—投票功能：public vote()**

**5.3.3.1.1 功能**

给指定的参赛队伍投票

**5.2.3.4.2 参数**

index：参赛队伍的索引

**5.3.3.2 方法—查看投票结果：public winningProposal()**

**5.3.3.2.1 功能**

查看当前得票最高的队伍名称、队员姓名、当前票数

**5.3.3.2.2 参数**

无