

基于区块链的教育资源共享平台

网络技术挑战赛 A 系列

2022 年 4 月 21 号

目 录

1. 所在系列及赛项	1
2. 引言	2
2.1 背景及应用领域	2
2.2 解决的问题	2
2.3 实现的基本功能	3
3. 设计思路与方案	4
3.1 设计思路	4
3.2 总体设计	4
3.3 系统详细设计	6
3.3.1 系统架构	6
3.3.2 调用关系	7
4. 系统实现	8
4.1 应用技术介绍	8
4.1.1 以太坊与智能合约	8
4.1.2 IPFS	10
4.1.3 React 与 Node.js	11
4.2 系统的预期效果	12
5. 系统创新性	16

1. 所在系列及赛项

本作品属于中国高校计算机大赛——网络技术挑战赛 A 系列。

2. 引言

2.1 背景及应用领域

随着高等学校数字化校园建设的迅速发展,高校优质教学资源共建共享取得了初步成效。但正是教育资源“开放”的特性,使资源创作者、服务提供方、资源使用者缺乏约束,并且无法对资源产权进行清晰的界定和划分,故出现了资源版权保护弱、资源存储有风险、资源质量低等现实问题。

区块链技术因虚拟数字加密货币的诞生与发展而逐步得到专家学者们的关注,其本身的去中心化、防篡改、可追溯、分布式技术的特性区别于其他技术有着革命性的意义,其应用场景也早已从最初的虚拟货币逐步延伸到现在的各个领域之中,为其提供去中心化的解决方案。

第二代以以太坊为代表的区块链技术,首次实现了图灵完备性,使得以太坊应用越来越多元。以太坊秉持简洁、通用、模块化、无歧视原则进行设计供用户使用,并且在其上提供各种模块让用户搭建应用,即智能合约,其提供业务无穷无尽,进而解决了传统区块链应用的可扩展性不足的缺陷。

2.2 解决的问题

本系统将针对教育资源平台现存在的资源存储风险、资源创建激励机制欠缺、优质教育资源缺乏以及资源版权确权等问题,设计一个

教育资源共享平台。本系统将选择以太坊为平台开发环境部署智能合约，结合 IPFS（InterPlanetary File System/星际文件系统）、Nodejs（Koa、Egg、web3.js、ipfs-http-client）、React 对平台总体架构进行构建。解决资源版权保护弱、资源存储有风险、资源质量低等现实问题。

2.3 实现的基本功能

系统实现的基本功能大致如下：

- 1、设计用户系统并结合以太坊的账户功能，解决资源版权保护弱的问题
- 2、使用 IPFS 技术，实现资源的分布式存贮，以此解决资源的存储风险
- 3、实现用户评论信息上链，保证资源评论有迹可循

3. 设计思路与方案

3.1 设计思路

随着我国教育数字化建设的迅速发展，中国教育资源建设取得了初步成效然而目前的教育资源建设仍存在以下问题：

1、资源版权保护弱。部分数字作品的发表导致资源版权的认定及后续维权带来极大困难。很多资源的作者担心自己编写内容公开后版权得不到有效保护，从而导致很多优质教学资源得不到更广泛的利用。

2、资源存储有风险。资源在存储过程中可能受到网络攻击导致服务器被破坏或数据丢失。此外，由于教学资源的共享，用户隐私也受到一定的威胁。

3、资源评论环境恶劣。由于网络的虚拟性，部分用户的评论具有攻击性，甚至会攻击作者本身。

本项目从解决上述问题的角度出发，面对当下网络的虚拟性导致的较为恶劣的讨论氛围，决定采用区块链（以太坊）与智能合约技术进行用户的管理与认证；利用 IPFS 等技术来实现文件的分布式存储；通过将评论信息上传至区块链，保证信息有迹可循，有据可查。

本系统解决了上述的问题，有效的解决资源的版权问题、存储问题与评论环境问题，给每个用户一种较为舒适的使用体验。

3.2 总体设计

基于区块链的教育资源共享平台是一个面向 Web 端开发的资源

共享平台。服务端通过 node.js (web3.js 与 ipfs-http-client) 与智能合约和 IPFS 进行交互。

系统大致可以分为部分：合约交互模块、ipfs 交互模块、用户数据模块、web 前端界面。如图 3-1 所示。

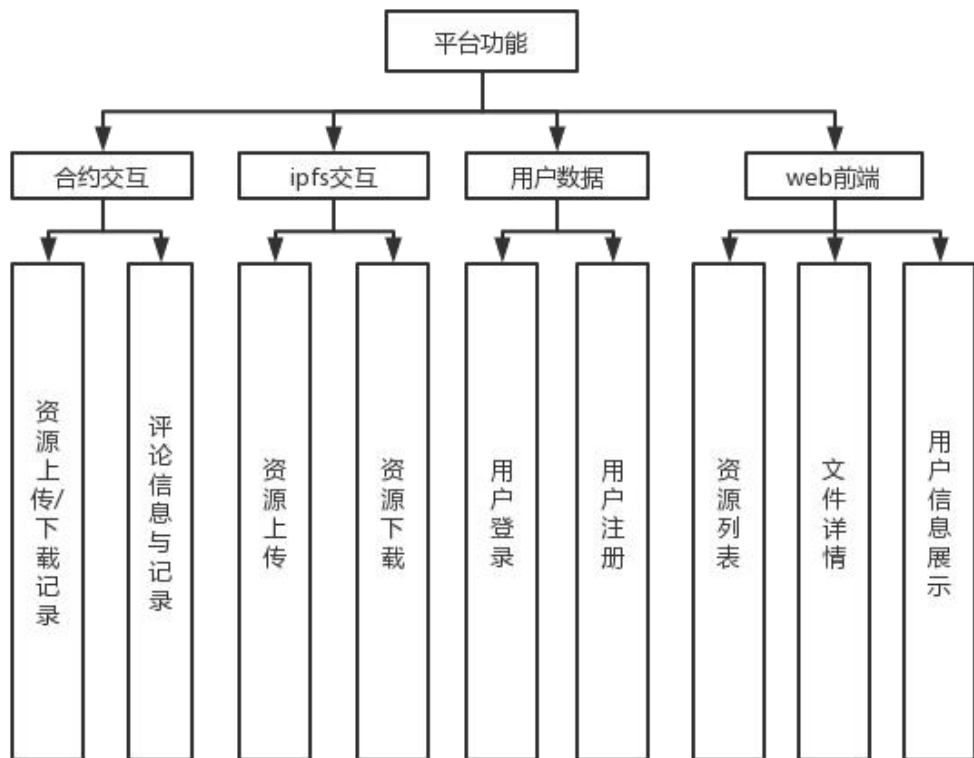


图 3-1 系统功能

1、合约交互模块：使用 web3.js 与以太坊和智能合约进行交互，通过以太坊区块存储资源的上传/下载记录与评论信息。

2、Ipfs 交互模块：使用 ipfs-http-client 与 IPFS 进行交互，实现资源的上传与下载。用 IPFS 来实现文件分布式存储。

3、用户数据模块：使用 MySQL 数据库来存储用户数据，实现用户的登录注册等功能。

4、Web 前端模块：前端界面展示资源列表，并且能展示每个资源的详细信息。在用户信息中可以展示用户的基本信息，用户的资源上传记录与资源下载记录。

3.3 系统详细设计

3.3.1 系统架构

本系统实现了资源的分布式存储、资源的下载、评论信息上链与用户系统。系统架构图如图 3-2 所示。

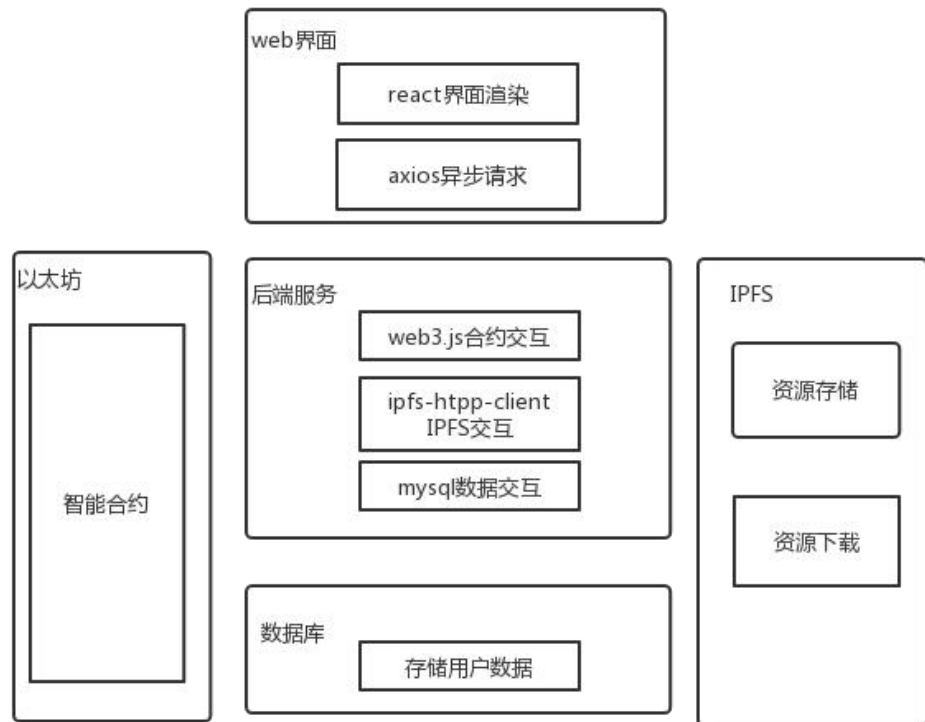


图 3-2 系统架构图

其中后端服务代码使用 `node` 进行书写，`web3`、`ipfs-http-client` 等都是 `node` 中较为成熟的与以太坊和 IPFS 交互的 `js` 库。智能合约使用 `Solidity` 语言进行书写，并且使用 `truffle` 进行部署，使用 `ganache` 在本机内存中模拟区块链。

3.3.2 系统调用关系

在本系统中用到了 node、react 和 axios 等多项技术，这些之间的调用关系如图 3-3 所示。

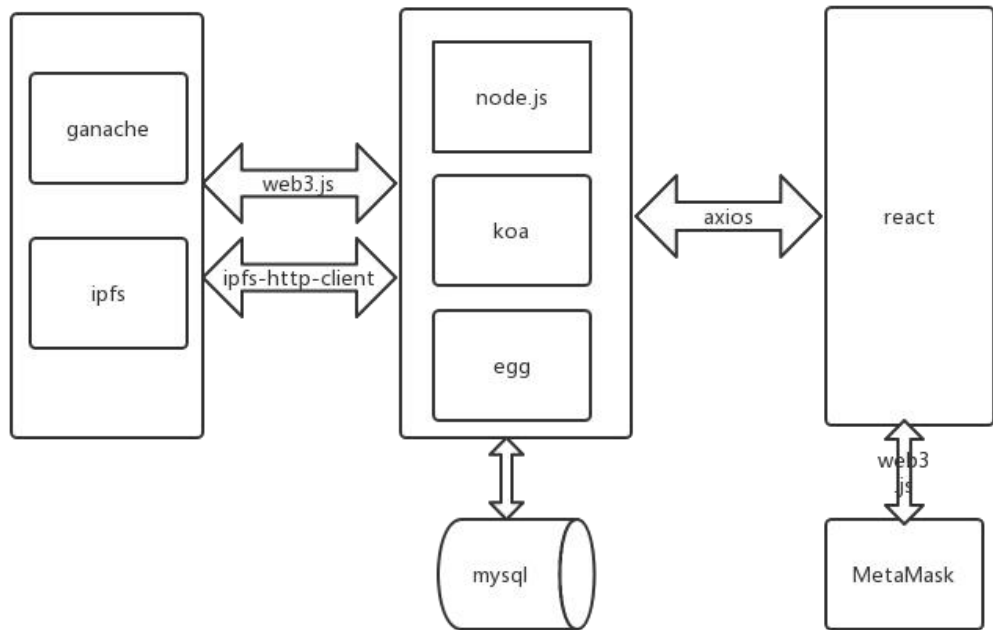


图 3-3 引用关系

4. 系统实现

该系统基于区块链的去中心化、防篡改、可溯源等特性，选择以太坊为平台部署智能合约，结合 IPFS（InterPlanetary File System/星际文件系统）、Nodejs（Koa、Egg、web3.js、ipfs-http-client）、React 对平台总体架构进行构建。

本系统的开发环境如下：

表 4-1 系统环境与对应版本

环境	版本号
Ubuntu	18.0.4
Node.js	16.14.2
React	>=16.8
GO 语言	1.14.2
Solidity 语言	>=0.4.0
Ipfs	0.8.0

4.1 应用技术介绍

4.1.1 以太坊与智能合约

以太坊的主要特点就是其图灵完备性，支持包括循环在内的所有类型的运算。除此之外以太坊还支持 8 种交易状态，同时还对区块链结构的其他特性进行了一些改进。以太坊代表具有内置图灵完备编程语言的区块链。它提供了一个抽象层，使任何人都可以创建自己的所有权，交易格式和状态转换功能规则。

1、以太坊账户：以太坊状态由帐户组成，其中每个帐户都有一个 20 字节的地址和状态转换。世界状态是地址和帐户状态之间的映射。以太坊支持两种类型的账户：外部拥有的（由私钥控制）和合约账户（由其合同代码控制）。以太坊账户由四个字段组成：随机数，以太币余额，合约代码哈希和存储根。

2、以太坊交易和消息：交易有两种类型，一种是导致消息调用的交易，另一种是创建新帐户的交易。交易被定义为从外部拥有的帐户发送的签名数据包。每个交易都由消息的接收者，标识发送者的签名，要发送的以太量，可选数据字段，STARTGAS 和 GASPRICE 值组成。

3、以太坊区块链：以太坊区块链和比特币区块链有许多相似之处，主要区别在于以太坊区块不仅包含区块编号，难度，随机数等，还包含交易列表和最新状态。对于交易列表中的每笔交易的新状态都是基于前一个交易状态来创建。

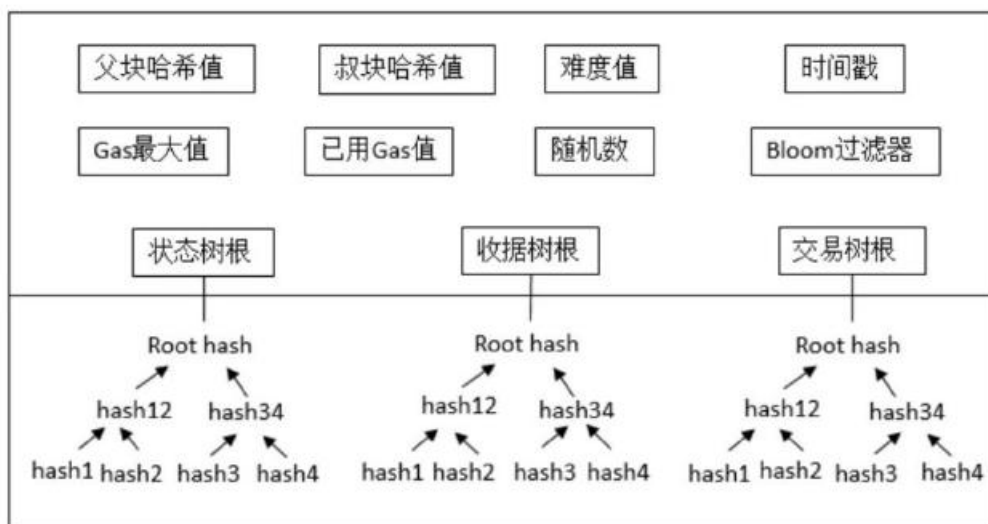


图 4-1 以太坊区块数据结构

4、以太坊网络中的每个节点都在 EVM 下运行并执行其指令。智

能合约被转换为 EVM 代码，然后由节点执行。用于编写智能合约的最受欢迎的编程语言之一是 Solidity。

合约通常指双方或多方之间就某件事情该如何处理的协议，而智能合约亦是如此，两者的区别在于，普通合约是用语言文字书写，由人来执行，而智能合约是用代码书写，用计算机运行，这使得智能合约能够脱离人为影响自动执行。而运行在区块链上的智能合约，由区块链不可篡改的特性加上区块链共识的安全机制保证，使得区块链上的智能合约能够无需任何三方信任就能执行，这对于实际应用中来说意义重大。智能合约通过提供不可逆的交易来促进支付公平，像是一个系统中的某个协议参与者，一旦成功部署到区块链上，它就不可逆地执行提前写好的合约，无法由外力干预。

以太坊是第一个开源的以智能合约为主的区块链平台，支持用户可以使用智能合约在平台上进行开发，搭建自己的区块链应用。以太坊应用了图灵完备的技术开发语言 Solidity 用作智能合约的开发语言，其强大的功能使得基于以太坊可以开发多种多样的应用。

Solidity 是一种语法类似于 Java Script 的高级面向对象语言，支持多种复杂的自定义类型异常发生后，执行会返回到异常临界点处，这种方法其实提高了一定的容错率，同时还可以恢复到数据正常的阶段。Solidity 语言包括文件布局和合约结构、数据类型和状态变量、函数调用、函数可见性、常函数、事件这些和其他语言相近的使用方法。

4.1.2 IPFS

IPFS (InterPlanetary File System) 是一个基于内容寻址的、分布式的、新型的超媒体传输协议。IPFS 支持创建完全分布式的应用, 它旨在使网络更快、更安全、更开放。IPFS 是一个分布式文件系统, 它的目标是将所有计算机设备链接到同一个文件系统, 从而形成一个全球统一的存储系统。

IPFS 与传统的 http 协议基于域名访问文件并从单一服务器下载不同, 它是基于文件内容的哈希值去索引文件, 使得获取文件的速度更高效。由于 IPFS 基于点对点网络, 因此在 IPFS 网络中, 不存在中心化的服务器, 每个用户 (节点) 都是对等的, 每个用户既是服务提供方, 也是服务的接受方, 用户之间可以自由地分享文件。

4.1.3 React 与 Node.js

React 是用于构建用户界面的 JavaScript 库, 起源于 Facebook 的内部项目, 该公司对市场上所有 JavaScript MVC 框架都不满意, 决定自行开发一套, 用于架设 Instagram 的网站。于 2013 年 5 月开源, 现在是最为主流的前端开发框架之一。

React 框架特点:

1、声明式设计: React 使创建交互式 UI 变得轻而易举。为你应用的每一个状态设计简洁的视图, 当数据变动时 React 能高效更新并渲染合适的组件

2、组件化: 构建管理自身状态的封装组件, 然后对其组合以构成复杂的 UI。

3、高效: React 通过对 DOM 的模拟, 最大限度地减少与 DOM 的

交互。

4、灵活：无论你现在使用什么技术栈，在无需重写现有代码的前提下，通过引入 React 来开发新功能。

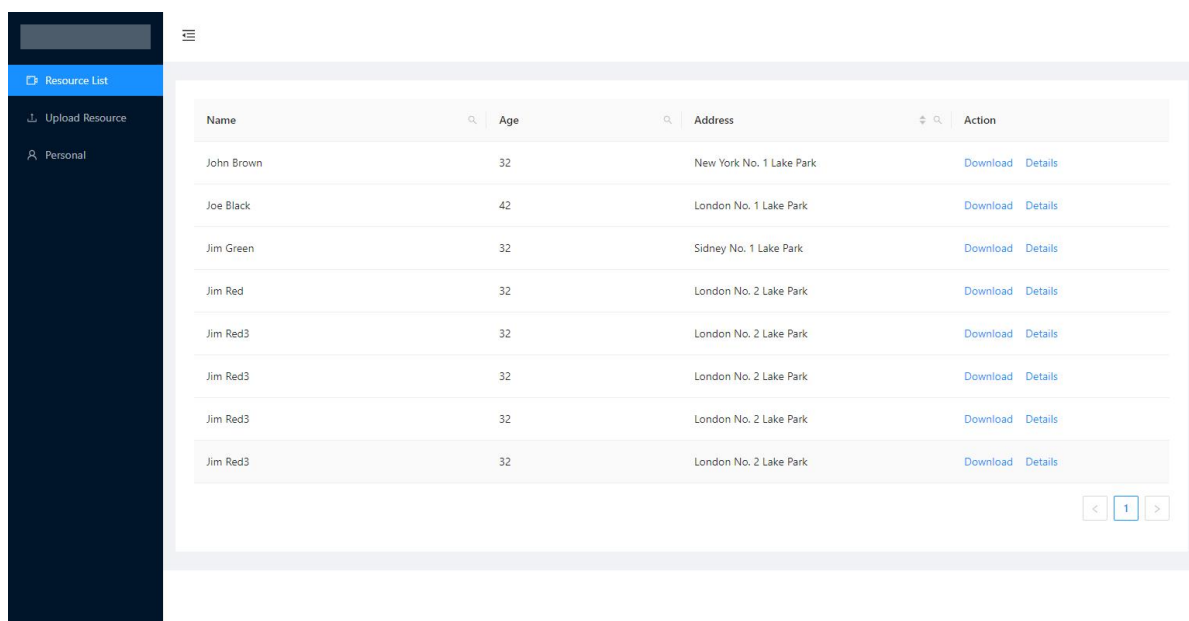
Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境，使用了一个事件驱动、非阻塞式 I/O 模型，让 JavaScript 运行在服务端的开发平台，它让 JavaScript 成为与 PHP、Python、Perl、Ruby 等服务端语言平起平坐的脚本语言。

4.2 系统的预期效果

我们已经对本系统进行的预先的设计，下面将针对前端模块划分展示预期实现效果。

1、资源列表

资源列表的实现效果如图 4-2 所示：



Name	Age	Address	Action
John Brown	32	New York No. 1 Lake Park	Download Details
Joe Black	42	London No. 1 Lake Park	Download Details
Jim Green	32	Sidney No. 1 Lake Park	Download Details
Jim Red	32	London No. 2 Lake Park	Download Details
Jim Red3	32	London No. 2 Lake Park	Download Details
Jim Red3	32	London No. 2 Lake Park	Download Details
Jim Red3	32	London No. 2 Lake Park	Download Details
Jim Red3	32	London No. 2 Lake Park	Download Details

图 4-2 资源列表预期实现效果

2、资源详情

资源详情的实现效果如图 4-3 所示：

page / id / hash / [BACK](#) [Download](#)

Resource Information

UserName	Zhou Maomao	Telephone	1810000000
Live	Hangzhou, Zhejiang	Remark	empty
Address	No. 18, Wantang Road, Xihu District, Hangzhou, Zhejiang, China		
Rate	★★★★☆		

Comment

[Add Comment](#) 0 / 100

1 reply

Han Solo a few seconds ago
this is a test

图 4-3 资源详情预期实现效果

3、资源上传

资源上传的实现效果如图 4-4 所示：

☰

- Resource List
- Upload Resource**
- Personal

File Name:

Author Name:

Select Style:

Select Date:

Description:

Select:

Click or drag file to this area to upload
Support for a single or bulk upload. Strictly prohibit from uploading company data or other band files

Upload: [Upload](#)

图 4-4 资源上传预期实现效果

4、用户信息

用户信息的实现效果如图 4-5 所示：

The screenshot shows a web interface for user information management. On the left is a dark sidebar with a menu containing 'Resource List', 'Upload Resource', and 'Personal' (which is highlighted). The main content area is titled 'Personal Information' and includes an 'Edit' button. Below this, there are fields for personal details: name (Li Si), address (test address), sex (man), birthday (2000-06-02), position (xxxxxxx), education (xxxxxxx), and university (xxxxxxxxx). Below the personal information section are two tables: 'Upload Record' and 'Download Record'. The 'Upload Record' table has columns for Name, Age, and Address, and lists three entries: John Brown (32, New York No. 1 Lake Park), Jim Green (42, London No. 1 Lake Park), and Joe Black (32, Sidney No. 1 Lake Park). The 'Download Record' table has the same columns and lists one entry: John Brown (32, New York No. 1 Lake Park). Both tables have pagination controls at the bottom right, showing '< 1 >'. The 'Personal Information' section also has an 'Edit' button.

图 4-5 用户信息预期实现效果

5、登录注册

用户登录注册界面如图 4-6、4-7 所示：

The screenshot shows a login form titled 'Please Login'. It contains two input fields: 'Username' and 'Password'. Below the password field is a checkbox labeled 'Remember me' which is checked. At the bottom of the form is a blue 'Log in' button and a link that says 'Or register now!'. The form is centered on a light gray background.

图 4-6 用户登录界面

The image shows a user registration form titled "Register". It contains the following fields and elements:

- E-mail:** A text input field with an asterisk (*) indicating it is required.
- Password:** A text input field with an asterisk (*) and a toggle icon (an eye in a circle) to the right.
- Confirm Password:** A text input field with an asterisk (*) and a toggle icon (an eye in a circle) to the right.
- Nickname:** A text input field with an asterisk (*) and a smiley face icon (😊) to the right.
- Captcha:** A text input field with a label "Captcha:" to its left.
- Get captcha:** A button located to the right of the Captcha input field.
- Agreement:** A checkbox followed by the text "I have read the [agreement](#)".
- Register:** A blue button at the bottom of the form.

Below the Captcha input field, there is a small text note: "We must make sure that your are a human."

图 4-7 用户注册界面

5. 系统创新性

1、将传统的 web 开发模式与区块链技术结合

传统的 web 开发模式采用中央数据库存储各种数据，但在本系统中使用以太坊与智能合约代替了传统的事务性数据库，保证信息的安全、可溯源与不可抵赖。

2、使用 ipfs 实现文件的存储与共享

使用了 IPFS 来实现文件的分布式存储，避免了文件的存储风险，是文件相较于非问不是存储更加安全。

3、将 web 用户模式与以太坊账户结合

在本系统中只有当 web 用户相当于第一层授权，即只是给了用户使用该系统的权限，但是系统所展示的内容都是有用户所使用的以太坊账户决定的，只有用户的以太坊账户正确才能进行各种操作，这避免了用户因为用户名被盗所导致的损失。