***基于区块链的成绩认证***

***◎著 陈超荣***



***江西软件职业技术大学***

目录

[一、需求分析 2](#_Toc61214363)

[1.背景与现状 2](#_Toc61214364)

[2.概述 4](#_Toc61214365)

[3.区块链技术与成绩认证的应用案例 5](#_Toc61214366)

[二、区块链技术与成绩认证的匹配度分析 7](#_Toc61214367)

[三、基于区块链的成绩认证方案设计 9](#_Toc61214368)

[1.业务设计 9](#_Toc61214369)

[2.架构设计 12](#_Toc61214370)

[3.交互设计 16](#_Toc61214371)

[四、关键技术 21](#_Toc61214372)

[1.以太坊与智能合约 21](#_Toc61214373)

[2.web3js 22](#_Toc61214374)

[五、合约内容 23](#_Toc61214375)

[1.1基于区块链的毕业证书 23](#_Toc61214376)

[1.2基于区块链的学分认证 26](#_Toc61214377)

[六、前后端代码 30](#_Toc61214378)

[6.1必要的准备： 30](#_Toc61214379)

[6.2基于区块链的毕业证书： 31](#_Toc61214380)

[6.3基于区块链的学分认证： 39](#_Toc61214381)

[七、前端操作 45](#_Toc61214382)

[7.1基于区块链的毕业证书 45](#_Toc61214383)

[7.2. 基于区块链的学分认证 46](#_Toc61214384)

[八、总结和体会 50](#_Toc61214385)

## 一、需求分析

### 1.背景与现状

**学历造假**

早在文学经典《围城》中，钱钟书先生便描绘了主人公方鸿渐在海外留学，一事无成，花钱买了一张“克莱登大学”的学历，归国后竟能够以此假学历招摇撞骗。今天，这一问题仍然未得到很好的解决，假证、假学历在人才市场泛滥，其造假成本极低，几百元就能定制一张大学毕业证书，对企业识人用人造成了严重困扰。

**一考定终身**

在升学过程中，学生和家长都面临一个困扰：一考定终身。学生生涯中屈指可数的几次大型考试在很大程度上决定了人生的发展路径。过去教育评价方式单一，衡量受教育者的标准只局限于考试这种存在弊病的固定形式。人才培养方式过于单一，无法真正做到对个体因材施教，难以发挥每个人的优势。

**学习社区积极性低**

在日常的学习工作中，我们总会求助全球知识网络，小到基于在线课程的学习交流平台——同时上这门课程的学员可以通过分享笔记、互动讨论等方式交流心得，大到专业性强的各类学习论坛，如以Github为首的编程类论坛、经管之家等经济类论坛。这类论坛会集了专业能力极强的专家，他们乐于在自己擅长的专业领域里与其他有相同兴趣的人讨论难题，传道解惑。更广泛的是各类问答型社区，以知乎、Quora等知名社区为主导，这类社区由于讨论话题的全面性带来较低的门槛，网民参与度极高，而一些热门答案往往引起极高的关注度。然而，现有的学习社区存在参与度不足、社区环境治理难、抄袭现象严重等问题，严重阻碍了社区的健康发展。

当前，学习社区为了激励回答者的回答热情，提升社区活跃度，一般会给优质资料上传者或者答疑者社区币、积分等形式的虚拟奖励。优质资料上传者和答疑者会因为奖励积累得到等级的提升等荣誉，也可以用积分等兑换相应奖励。然而，这一激励系统存在一些问题：其一，在社区里得到的积分不能变现，只能内部消费；其二，由于这些社区都是相互孤立的，A社区的积分对于B社区来说毫无价值；其三，万一哪天社区不运营了，这些积分将全部作废。因为这些问题，相关学习者的积极性非常低，现有学习社区的活跃度也受到很大影响。

**版权维权困难**

2019年初，翟某学术不端事件一石激起千层浪。以高学历自居的翟某，竟不知中国学术期刊著名网站“知网”为何物，其博士、硕士期间的论文均涉嫌抄袭，这一行为不仅伤害了广大寒窗苦读的学子，更是伤害了呕心沥血完成作品的原作者。近来一段时间，学术圈频频出现的“暴雷”现象更是让人寒心。踏实做研究的学者得不到应有的权益，侵犯别人版权的抄袭者却逍遥法外。学术版权问题在中国学术界乃至世界学术界都是极为重要且需要规范的问题

如今，学术出版的数字化使得学术作品能够更快地触及更多的读者，这本是一件知识普及的好事，却难逃更易被盗用的下场。虽然近年来相关检测学术不端的工具也取得了一定的发展，但还是难以杜绝学术版权被频繁侵犯的现象。并且，就算相关出版机构在筛选时发现了学术不端行为，大多也只是私下撤稿，对相关人员的学术声誉影响很小。

在版权保护方面，虽然我国不断取得进步，但要想真正实现对个体版权的有效保护仍存在不小的难度。从版权登记成本来看，版权登记在提交所有证明材料的前提下，最短需要30个工作日，并且以单件最低100元为成本，才能完成版权登记。对于大规模出产的学术作品来说，这一耗费时间、金钱成本的流程的实操性很低。若不采取版权登记，只靠版权自动获取的政策，后续追责难度又会有很大提升。一旦牵扯版权法律纠纷，传统司法实践中证据验真会带来繁复的手续，维权成本会进一步提升。

**学术激励困难**

致力于学术研究的学者大都需要阅读大量的学术期刊才能写出好的作品，但由于绝大多数学术期刊都是需要付费的，而且近年来其价格一直在大幅增长，这给学者带来不小的经济压力。不少业内人士认为，由于大多数科学研究离不开政府的资助，付费学刊让科学成为一个需要纳税人与科学家双重付费的产品。造成这种局面的原因，是学术出版公司的垄断。

研究显示，如今几乎所有学术领域都至少有一半的研究成果出版于全球前5大出版集团，而一些领域，比如心理学，甚至有71%的论文发表于5大出版集团的学刊。与一切垄断一样，巨大的市场份额给这几大出版集团带来了惊人的利润。五大出版集团之一的爱思维尔的市值已涨到350亿美元，其科学出版业务的净利润率更是高达39%。相比之下，脸书与谷歌等科技巨头的净利润率仅有20%左右，苹果公司也只有30%出头。与此同时，相对于利润极高的出版集团而言，真正发表作品的学术研究者收到的酬劳却是杯水车薪，甚至难以收回前期在学术期刊上的投入。在当前国际学术版权垄断的背景下，广大学者对于寻求更为民主化的学术版权价值交换方式的呼声日渐高涨。

### 2.概述

**什么是区块链？**

简单来说，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。



**区块链如何工作？**

区块链是一个平台，可确保所存储信息的完整性并维持生态系统成员之间的交互。以下是其工作方式的高级概述：

1.每个成员都维护自己的区块链节点，其中包含附加到网络的所有事件和数据的完整历史记录，包括凭证，身份，证书等。

2.网络的每次更新都需要在链的末尾创建一个新块。区块链协议规定了如何记录，验证和分发这些块。

3.采用共识机制来验证每个创建的区块，区块链网络的成员将决定是否将其添加到链中。

4.一旦创建并确认了一个块，就无法将其撤销。区块链上的所有条目都是永久且安全地存储的。这允许社区成员跟踪交易的完整历史记录以及区块链中的任何其他修改。

5.智能合约是网络成员之间已编程了条件的一种特殊类型的协议，可确保在各方收到协议之前，它们都已得到满足。智能合约消除了第三方和中间人参与协议解决的需要。

6.加密货币交易在区块链生态系统中发挥着非常重要的作用，激励社区的所有成员做出宝贵的贡献并参与整个系统的开发。

区块链技术的这些关键支柱为其在包括教育在内的不同行业中使用奠定了基础。它有可能创造一个全球环境，使学习材料，出版物，学生学分和成绩单易于获得。它还可以为问责，激励以及教师，学生和其他参与者之间的交流引入新的创新方式。

### 3.区块链技术与成绩认证的应用案例

区块链应用在教育领域仍然很少，主要是因为它们与金融科技解决方案和加密货币的发展紧密相关。但是，许多创新型初创公司和全球公司开始向全球学校提供区块链支持的教育服务。以下是一些具体示例。



##### （1）区块证书：教育证书管理

[Blockcerts](https://www.blockcerts.org/)是一个开放标准的平台，用于存储和验证数字证书（包括成绩单和证书）。它的区块链网络可以存储从文凭到转移记录到成绩分数的任何内容。这将创建一个通用的生态系统，用于创建，发布，查看和验证有关学生的学术历史的信息。使用该平台，大学可以确保学生记录的可信度，同时潜在的雇主可以查看整个学生档案，包括他们的奖励和成就，并对候选人的优势，能力，知识和经验有更好的了解。

##### （2）APPII：简历制作人向雇主展示经过验证的教育功绩

[APPII](https://appii.io/)是职业经历验证者和招聘平台。候选人可以通过增加学历，学历，成就和经验轻松制作简历。区块链网络用于通过与教育机构共享来验证候选人的学历。机器学习还用于将简历中提供的所有数据与需要相应技能的工作机会进行匹配。由于该信息已通过区块链技术和密码学进行验证和保护，因此雇主对候选人的知识和经验更加信任。

##### （3）ODEM：教育产品和服务市场

[ODEM](https://odem.cloud/)是基于区块链的在线教育平台，提供可负担得起的学习课程，可以根据学生的喜好进行定制。学生可以直接与学术专业人士交流，并商定最佳的学习体验。该平台使用具有区块链功能的支付系统和智能合约来激励和奖励学生的学习努力，并激励教师也开设更好的课程。学生获得的所有证书都安全地存储在区块链上。因此，可以轻松地在社交媒体上访问和共享它们，或在数字CV中作为证明提供它们。

##### （4）索尼全球教育：数字笔录管理

[索尼全球教育部](https://blockchain.sonyged.com/)允许多个教育机构在分布式分类帐中共享其学生的成就。区块链网络充当每个学生终生数字笔录的存储，以后可以在大学之间进行转移时使用。索尼的意图是进一步开发平台，并为下一代教育生态系统构建基础设施，在该生态系统中，师生双方都可以合作并获得加密货币的奖励。

##### （5）BitDegree：具有游戏化体验的在线教育平台

[BitDegree](https://www.bitdegree.org/)提供有关软件和技术的在线课程。鼓励学生使用代币奖学金来学习。该平台具有一个游戏化的环境，可以为每个学习者分配一个角色，并通过完成任务来导航虚拟世界。在学习和完成课程方面取得显著成就将获得他们的奖励和认可，这些潜在的雇主也在平台上注册。所有学术证书都存储在区块链上，并累积在学生的个人资料中。希望以崭新的眼光聘请新技术专家的公司可以跟踪学生的学习进度，并为最合适的候选人提供他们的团队一席之地。

##### （6）学科：个性化学习管理系统

[Disciplina](http://disciplina.ru/)使用区块链技术来跟踪学生的学习进度。根据学生在课堂上的表现，参加课外活动和其他成就的程度为每个学生分配分数。在平台上注册的学校和学院使用这些分数来个性化学习过程并为每门课程制定特殊计划。Disciplina还旨在简化学生与教师之间的交流，并引入方便的工具进行评估，评分和评论。招聘机构和雇主也可以在平台上注册以找到潜在的候选人。

## 二、区块链技术与成绩认证的匹配度分析

教育行业面临的挑战已经拖了多年。这包括低效的基于纸张的记录保存流程，缺乏透明度，不良的学生和老师问责制，没有真正动机促使学生学习和在课堂上表现良好，以及由于伪造而对教育功绩和学位缺乏信任。区块链有能力通过提供以下详细描述的切实解决方案来影响教育并应对这些挑战。



##### （1）增强和透明的记录保存

使用不可变的区块链存储，注册服务商的学校办公室只需创建一次学生记录。此后，教育系统的所有参与者都可以使用它，甚至可以在各个机构之间共享它。每个任务，任务，参加学校和课外活动都作为块附加到学生的数字记录中，并在学生的个人资料中查看。因此，教师，辅导员和学校管理人员对学生的学习过程有鸟瞰图。

除了维护学生记录外，区块链系统还可用于监督和促进学校，学院和大学的认证，保护知识产权并消除文凭和成绩单的伪造。基于区块链的教育记录可以轻松访问并快速转移到其他机构和组织，包括潜在的雇主。

学生要做的就是共享一个链接或他们的数字地址，以访问他们的个人资料。这种类型的系统可确保凭据和学生记录的完全真实性，因为带有签名的完整更改历史记录存储在区块链上。

##### （2）智能合约推动的学生问责制

教师可以使用区块链智能合约与学生就完成任务达成数字协议。协议中详细说明了所有作业要求，包括说明，条件，截止日期和截止日期。学生完成任务并满足协议中的所有条款后，可以自动授予他们访问课程下一部分的权限，或获得学分和结业证书。

最终，智能合约可以在传统教室设置中提供个性化学习中发挥重要作用，并且还可以创建传统教育系统中缺少的激励因素。

##### （3）激励学生和老师取得更大的成就

加密货币和令牌化学分可用于创建有效的奖励系统，以激励学生，并通过创建新的学习材料和参与研究来鼓励教师为教育过程做出贡献。可以对智能合约进行编程，以针对完成的任务以加密货币颁发奖励。通过向表现优异的人给予特别的奖励，教师将能够进一步激励学生。

最终，区块链技术可以充分利用现代教育中的游戏化元素，并将其提升到一个新的水平。代币化的学分可用于偿还学生贷款，购买学校用品，学习材料以及支付其他教育费用。完成某些专业并获得大学文凭的奖励可以大大减少在没有完成大学意愿的情况下辍学的学生人数。

##### （4）学习成绩跟踪和个性化

归功于区块链数据存储提供的透明性，教育者拥有更多可用于监视和控制学生表现的资源。在使用智能合约及其输出作为确定学生苦苦挣扎的工具的同时，学校可以提供完全定制的学习环境。整个学区的机构都可以访问记录在区块链上的学生成绩数据，并用于改善教育实践。

在全球范围内，这种系统可以汇总地区、省和国家/地区级别的绩效统计信息，并提高学校，学院和大学之间的健康竞争，其最终目标是提供最佳的教育水平。

##### （5）介绍可验证的终身学生成绩单

我们所有人都经历了从教育机构获取成绩单的麻烦。这是一个非常耗时的过程，涉及多个方面来验证凭据并编制完整的学术记录。区块链可以在需要时为学生提供在线数字成绩单。这份可核实的学生终生成绩单可以包含有关所有教育成就的信息，并简化凭据验证，从而使学生在学校之间的转学更加容易。

##### （6）为学生提供教育成就档案袋

一个教育性的区块链平台可以允许学生创建自己的数字档案袋，以存储他们所有的教育价值，例如：

①获得专业/副修及学位

②经实践证明的能力

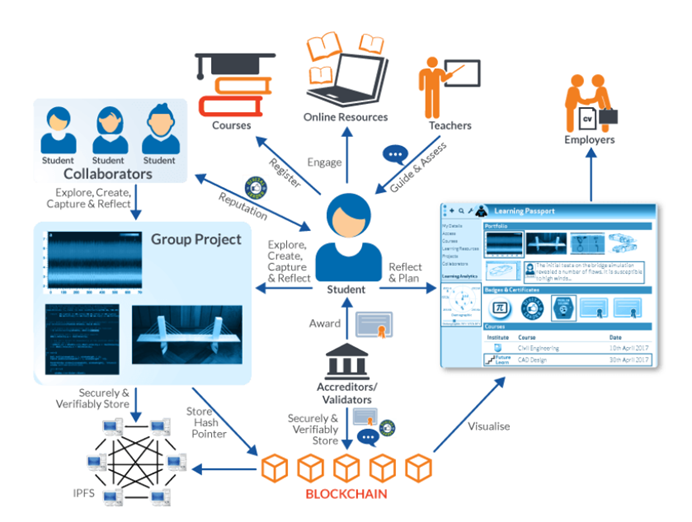
③完成的课程证书

④取得成就的微证书

⑤额外的学分和奖励

⑥测试成绩和出勤记录。

可以公开共享该档案袋，以证明学生有大学录取的资格，或者由寻找工作机会的学生发送给雇主。最终，可以建立一个潜在候选人的通用数据库，并且公司可以寻找表现最好的学生，从而为在学校取得更好的表现提供了更多的动力。



## 三、基于区块链的成绩认证方案设计

### 1.业务设计

#### 1.1基于区块链的毕业证书



基于区块链的可信毕业证书用于高校给学生发放区块链毕业证书，以取代纸质证书，并通过区块链保存不可篡改的发证、查证记录，使各参与主体共同建造、共同维护、共同监督，从而满足各参与主体的监督权、增强毕业证书的客观性与可信度。

从业务功能上来说，可信区块链毕业证书应用包括登记、发证、查验几个环节。

登记环节由学生在高校网站指定页面将账户地址提交给学校，完成实名登记，申请区块链毕业证书；

发证环节由学生完成毕业论文达到毕业条件时，由学校发放区块链毕业证书；

查验环节由学生向雇主提供指向的数字文凭的链接，雇主在区块链对证书进行查询，验证证书的真伪。

1.登记

实现区块链毕业证书所有者的申请工作。学生根据个人在高校网站指定页面将账户地址提交给学校，完成实名登记、申请区块链毕业证书的业务。

2.发证

实现学生区块链毕业证书的发放。学生完成毕业论文达到毕业条件时，由学校发放区块链毕业证书。

具体流程如下：

（1）学校为学生申请一个唯一标识（证书ID）

（2）学校将学生信息将唯一标识绑定在一起，区块链完成共识过程，实现个人证书的记账、生成基于区块链的唯一性的证书。

3.查验

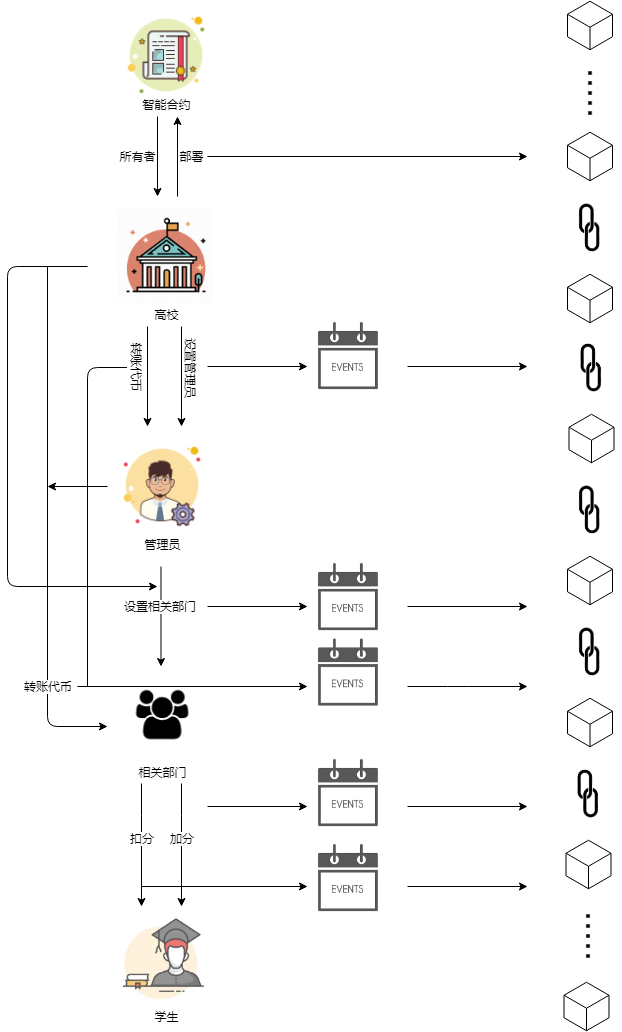
雇主通过区块链对学生证书的真伪性进行确认。

具体流程如下：

（1）学生向雇主提供指向的数字文凭的链接。

（2）雇主在区块链对毕业证书进行查询，验证毕业证书是否为高校官方所发放。

#### 1.2基于区块链的学分认证

****

基于区块链的学分认证用于高校给学生进行学分认证，通过区块链保存不可篡改的加分、扣分记录，使学校，学生共同建造、共同维护、共同监督，从而满足学生的知情权和监督权、增强学分认证的客观性与可信度。

​ 从业务功能上来说，包含以下几个步骤：

（1）高校创建合约之后，可以增加几个备用的管理员账户，这几个管理员账户和所有人一样拥有操控合约的最高权限，添加管理员的操作将记录到合约的事件当中。

（2）管理员对相关部门进行登记、认证，添加相关部门的操作将记录到合约的事件当中。经过认证之后相关部门就可以对学生的学分进行增加和扣除的处理，但是相应的操作都会被记录到合约的时间当中。

（3）学生完成了指定的任务，达到了添加学分的条件，相关部门将AheadCoin奖励给学生（相关部门将AheadCoin发送到学生的账户上），并且将此操作记录到合约的事件当中，学生可以立即查询到AheadCoin的余额和加分的数值和原因。

（4）学生违反了某些规定，触发了扣分的条件，相关部门将AheadCoin从学生账户上扣除，并且将此操作记录到合约的事件当中，学生可以立即查询到AheadCoin的余额和扣分的数值和原因

### 2.架构设计

内容架构由用户层、应用层、合约层、网络层和数据层组成。



#### 4.1用户层

区块链技术通过密码学及共识算法解决了链上数据的信任问题，其中身份权限问题可通过数字身份解决。数字身份模块负责合约各参与方的身份管理和身份认证，确认用户对资源的访问、操作权限，保障合约访问控制策略的可靠运行。数字身份模块包括管理人员准入和访问控制功能组件。

##### （1）管理人员准入

为保证合约管理人员行为所属人可追溯，管理人员身份与实体身份一一对应，做到管理人员行为所属人可追溯。

##### （2）访问控制

权限控制功能组件是合约安全的重要部分，功能包括用户身份管理、用户角色管理和权限配置。按操控资源的权限不同定义多个角色，按角色授权不同用户操控资源的权限及安全规则，使用户只能在有限范围内进行写入、读取和使用数据资源，而在其读取范围外的数据则会进行加密保护。

合约的参与成员之间权限、职责、资源、合作形式都形成共识，在彼此信任的平台上，实现授权管理、授权部门、发放代币和扣除代币等业务。保证各个业务操作都清晰明确。

###### a.权限控制模型

学分认证合约建议使用RPI（Role-Permission-Interface）权限控制模型①，设置角色、权限、接口四类对象。

账号和角色的对应关系是1:N，一个账号能对应多个角色，每个账号使用独立且拥有唯一的公私钥对，发起交易时使用私钥进行签名，实现交易的可控及后续监管的追溯。

权限的大小按照普通用户、相关部门、系统管理员及所有者递增。用户只能调用该权限下合约的接口

###### b.角色和权限设计

①合约所有者 拥有认证管理人员和相关部门的权限。拥有发放和销毁令牌的权限，并且能自主增发令牌。拥有查看每笔交易和查看身份的权限，一般只参与授权，不参与运作管理

②管理员 权力等同于合约所有者

③相关部门 拥有发放和销毁令牌的权限，但是不能自主增发令牌。不参与授权、参与运作管理

④普通用户 拥有查看每笔交易和查看身份的权限，不参与授权、不参与运作管理

#### 4.2应用层

##### 4.2.1主要功能组件

##### （1）业务引擎

业务引擎功能组件负责向各参与方包括业务查询服务。

业务查询为各参与方提供转账记录查询，销毁记录查询，授权记录查询，用户余额查询，授权信息查询。可通过账户地址查询用户余额，授权信息。

##### （2）权限配置

权限配置功能组件为相关合约方法配置操作权限，只有授权用户才能够操作相关方法。

##### （3）服务模块的工作思路

一旦进行了交易，交易的内容将被永久保存，为后续审核，溯源、评价等提供基础依据。

##### 4.2.3服务模块的用途

##### （1）信息溯源

合约的事件中记录了相关人员的信息，可以通过事件来获取当时的用户交易信息。

##### （2）事故定责

如果相关部门操作失误，或者没有履行变动学分的责任，可以通过信息溯源追责相关负责人，对信息进行纠正。

#### 4.3合约层

#### （1）合约服务模块

智能合约是一种可自动执行的数字化协议，可按照预设合约条款自动执行。智能合约实现身份认证、配置管理、权限管理等功能。合约所有者在部署时生效。

##### （2）合约访问接口

合约访问接口为应用层发起的查询、交易、配置和监控等操作提供接口服务。

##### （3）合约安全

区块链具有不可篡改的特点，智能合约一旦发布极难修改，合约的安全决定了程序的安全。

智能合约本质上是代码程序，难免会有考虑缺失导致的漏洞。合约安全威胁主要包括可重入攻击、调用深度攻击、交易顺序依赖攻击、整数溢出攻击等。为保障智能合约的安全，应保证开发安全和全面的安全审计。

##### a.安全开发

智能合约安全开发包括开发工具、编译工具及安全编程规范等方面的保障。

①.智能合约应采用规范的工具进行开发和编译，使用安全统一的编译器选项，以便保证编译器所提供的最新的安全保护机制。

②.智能合约的编写应基于严格的安全编程规范。常见的安全编码规范包括弃用一些存在注入或内存破坏危险的第三方函数，采用智能合约的checks-effects-interactions原则进行合约编写等。

③.在智能合约编码中尽可能少地暴露被外部调用的接口。建议采用进程间隔离的技术将智能合约进程与系统隔离，通过socket通信方式远程调用智能合约。

##### b.安全审计

在发布智能合约之前，做好全面的安全审计工作，有效防范已知风险。第一，选择对应的虚拟机语言扫描工具对合约代码进行静态扫描，扫描事先建模的漏洞类型；第二，为了避免静态扫描可能产生误报的情况，应邀请安全专家进行交互式代码审计，避免重入攻击、未授权访问攻击、Solidity开发安全等。

##### c.安全测试

智能合约部署前必须经过大量的测试，在模拟环境中对智能合约进行攻击测试，挖掘可被利用的漏洞。

#### 4.4网络层

区块链的信息传播采用P2P网络，P2P网络是分布式自组织网络，依赖附近的节点来进行信息传输，需要互相暴露对方的IP，若网络中存在一个攻击者，就很容易给其他节点带来安全威胁。针对P2P网络的攻击有日食攻击、窃听攻击、BGP劫持攻击、节点客户端漏洞、拒绝服务攻击等。 P2P网络安全应做到以下几方面： ​

（1）在网络传输过程中，使用可靠的加密算法进行传输，防止恶意攻击者对节点网络进行流量窃取或劫持。如开启 JSONRPC 的节点强制使用 HTTPS 传输，而不是 HTTP 协议。

（2）加强对网络中传输数据的有效性、合理性、安全性进行验证，防止出现整数溢出等情况导致出现数据错误。

（3）加强节点网络安全性。对重要操作和信息客户端节点做必要的验证。

#### 4.5数据层

区块数据是分布在多个节点上的链式结构数据，单一或少部分节点的区块数据篡改不会影响整个区块链的运行。区块链的数据安全风险主要包括：利用数据不可删除的特性进行恶意信息攻击，利用大量垃圾信息进行资源滥用攻击。 针对以上风险，采用多层次的安全保障机制实现数据安全，包括权限控制、数据加密和隐私保护。

（1）权限控制 结合角色设计和权限控制，实现对数据读写权限的过滤，有权力的用户才能操控合约。可对系统资源采用最小权限原则来进行权限管理。

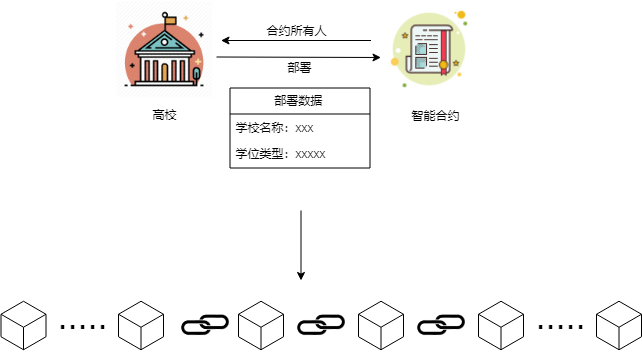
（2）数据加密 加密算法保证区块链的安全和不可篡改性。采用算法复杂度较高的对称加密、非对称加密算法，用密码信封完成加密数据分享。

（3）隐私保护 采用承诺系统、零知识证明、同态加密等方式规避隐私暴露。

### 3.交互设计

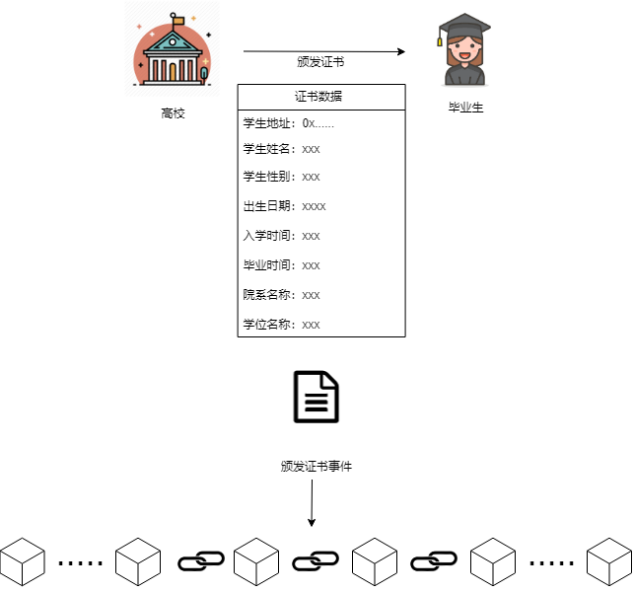
**3.1基于区块链的毕业证书**

**1.合约部署阶段**

****

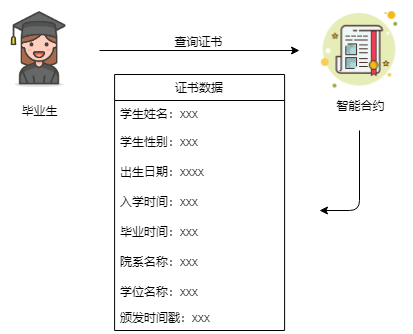
高校部署合约，部署数据为学校名称和学位类型，部署完毕之后，合约保存在区块链之中，高校成为合约的所有人。只有合约的所有人才能颁发证书。

**2.证书颁发阶段**

****

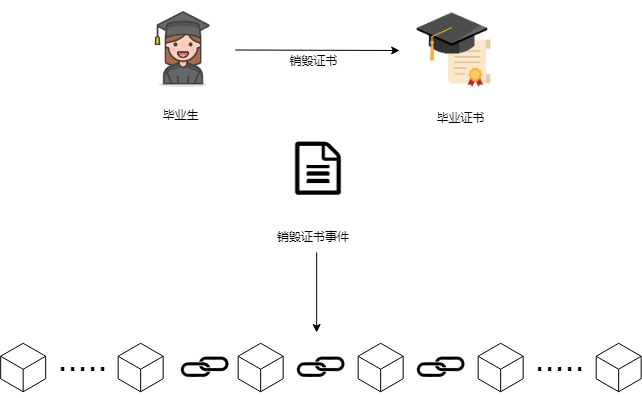
毕业生修完学分、完成学业达到毕业条件之后，高校根据学生信息为其颁发基于区块链的不可替代性的毕业证书，毕业生成为此证书的所有人。高校账户使用私钥签名交易，颁发证书的操作会被记录到合约中的颁发证书事件中，并且永远保存在区块链中，不可被篡改。

**3.证书查询阶段**

****

学生或者是用人单位可以通过学生毕业证书的编号查询证书的相关信息。学生输入证书的编号，如果证书存在，合约会返回相应的证书的信息。证书的相关信息有：学生姓名、学生性别、出生日期、入学时间、毕业时间、院系名称、学位名称以及证书颁发的时间戳。

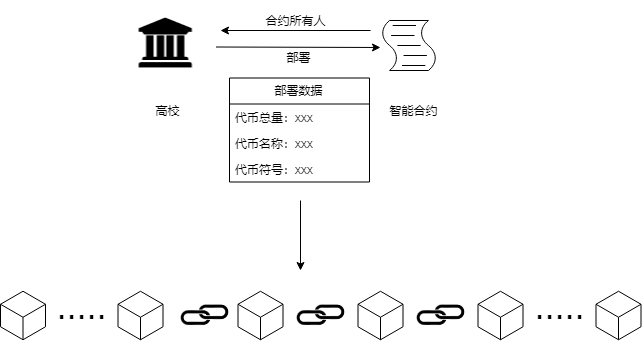
1. **证书销毁阶段**

****

只有毕业证书的所有人才能销毁毕业证书，学生通过毕业证书的编号获取指定证书，用私钥签署交易，对毕业证书进行销毁。此次销毁的操作会被记录到合约中的颁发证书事件中，并且永远保存在区块链中，不可被篡改。

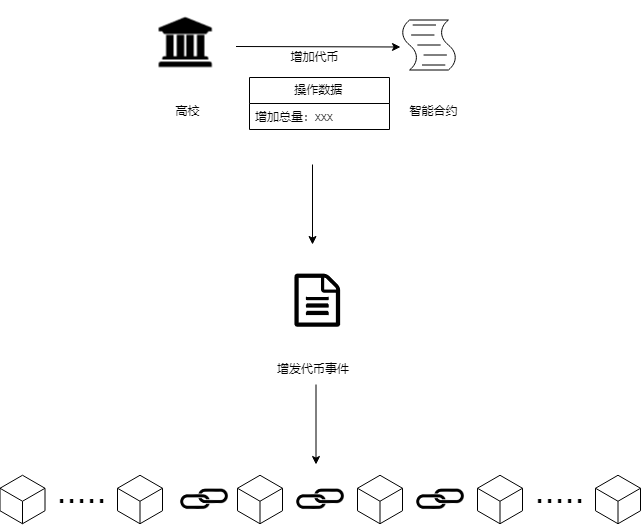
**3.2基于区块链的学分认证**

**1.合约部署阶段**



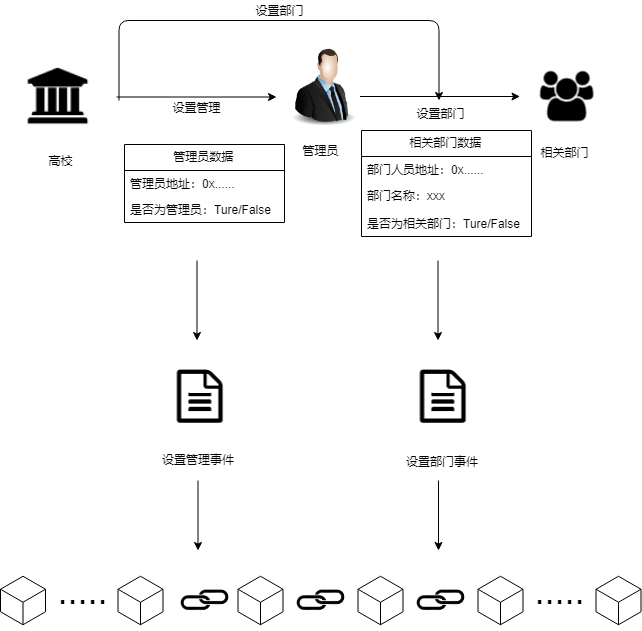
高校部署合约，部署数据为代币的总发行量、代币名称和代币符号，部署完毕之后，合约保存在区块链之中，高校成为合约的所有人。

**2.代币增加发行阶段**



高校或是管理员根据需要进行代币的发行，输入数据为额外发行的代币数量，高校或者是管理员账户使用私钥签名交易，完成代币的发行。代币发行的操作会被记录到合约中的增发代币事件中，并且永远保存在区块链中，不可被篡改。

**3.设置管理员、相关部门阶段**

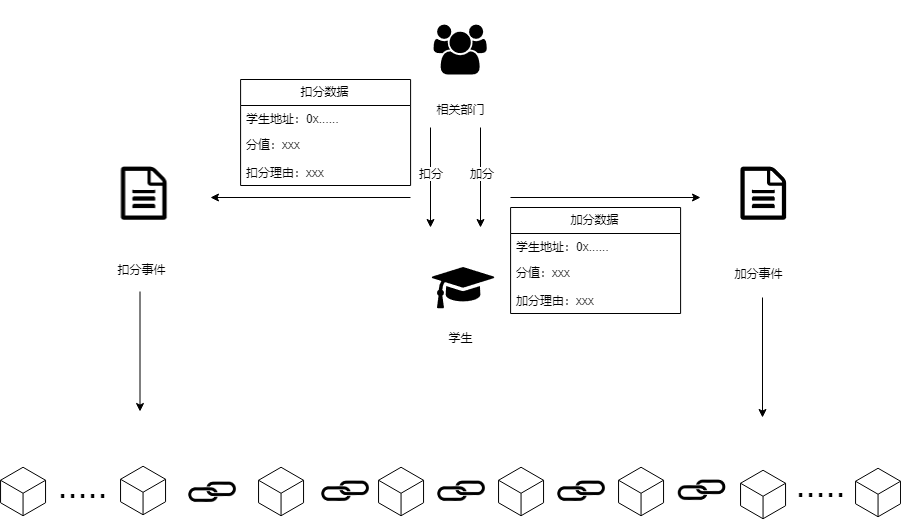


高校根据需要设置管理员作为备用账户，管理员和高校主账户拥有同等的权利。高校输入数据：目标账户地址以及是否让其成为管理员，接着使用私钥签名交易，完成设置，如果是true，目标账户被设置为管理员；如果是false，则目标账户不具有管理员权限。设置管理员的操作会被记录到合约中的设置管理事件中，并且永远保存在区块链中，不可被篡改。

高校根据需要设置相关部门。高校输入数据：目标账户地址、部门的名称以及是否让其成为相关部门，接着使用私钥签名交易，完成设置，如果是true，目标账户被设置为相关部门；如果是false，则目标账户不具有相关部门的权限。设置相关部门的操作会被记录到合约中的设置部门事件中，并且永远保存在区块链中，不可被篡改。

高校或者管理员需要预先将一定量的代币转账给相关部门，让其转账给学生，来对学生进行加分的操作。

**4.学分添加、扣除阶段**



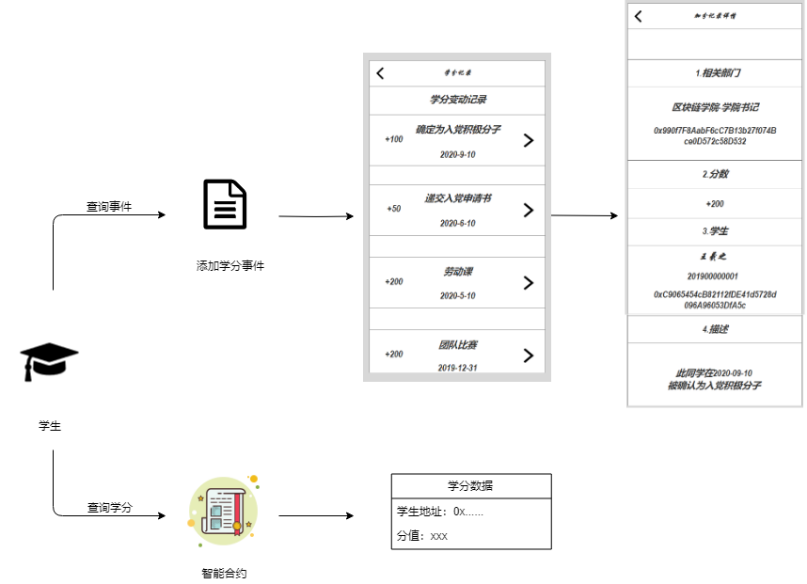
如果学生完成了指定的任务，达到了添加学分的条件，相关部门可将Ahead Coin奖励给学生（相关部门将一定数量的Ahead Coin发送到学生的账户上）来对学生进行加分处理。相关部门输入数据：学生账户、加分数额、加分原因，接着用私钥签名交易，将加分数额的代币发送到学生账户上，完成加分操作。此操作会被记录到合约的加分事件中，并且永远保存在区块链中，不可被篡改。

学生可以立即查询自己的账户余额来检查自己的学分是否已被添加。

如果学生违反了某项规定，达到了扣除学分的条件，相关部门可以从学生账户上扣除代币来对学生进行扣分处理。相关部门输入数据：学生账户、扣分数额、扣分原因，接着用私钥签名交易，将扣分数额的代币从学生账户上扣除，完成扣分操作。此操作会被记录到合约的扣分事件中，并且永远保存在区块链中，不可被篡改。

相关部门不能随意加分或者是扣分，因为通过事件，我们可以进行溯源，查询到之前的交易记录，不论是什么操作都会被记录下来。如果相关部门渎职，那么行为被发现只是时间问题。

**5.查询学分、学分变动记录**

****

学生输入数据：学生账户，点击查询学分，即可读取该账户学分的数值。学生输入数据：学生账户，点击查询学分变动记录，即可读取该账户变动记录。可以得知执行添加学分的相关部门、加分数值以及加分原因。

这个操作做到了，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。

## 四、关键技术

### 1.以太坊与智能合约

以太坊的核心元素是以太坊虚拟机（Ethereum Virtual Machine，EVM），它是智能合约的执行环境。EVM分散储存在以太坊网络的每个节点上，智能合约代码被对外隔离，并分布在每个节点上执行，因此以太坊EVM又被称为世界电脑。合同代码不是用图灵完备的高级程序语言编写的，而是由简单的、基于堆栈的低级程序语言编写的，看起来就像JVM的字节码（Java虚拟机）。每个以太坊节点都运行EVM，这意味着对于以太坊网络的参与者，每个节点都参与验证新块是否有效以及计算是否已正确，都是运行EVM代码的独立实例。由于每个节点都参与计算，虽然不一定是最高效的模型，但它具有很高的加密安全性。

从技术上讲，EVM以状态转换作为函数的运作模式，其工作原理是将一串参数输入EVM，以获取整个以太坊网络的新区块状态和gas数量，具体过程为输入（block\_state，gas，memory，transaction，message，code，stack，pc）→EVM→输出（block\_state，gas）。其中block\_state是以太坊网络的全局状态，包括所有账户、账户余额和长期存储；gas是运行这些计算所需的费用，由计算的类型和工作量决定；memory是执行内存；transaction代表交易；message是有关交易的元数据；code就是代码本身；stack和pc是与执行相关的堆栈和程序计数器。这一串参数被输入到EVM以生成整个以太坊网络的新block\_state和账户拥有的新gas数量。

以太坊EVM的设计目标有5个：简单、高效、确定性、专用化和安全性。EVM设计简单，可以轻松证明智能合约的安全性，这也有助于保护平台本身。EVM组件尽可能紧凑，以提高空间效率。EVM具有确定性，即相同的输入状态应始终产生相同的输出状态。确定性的虚拟机必然会限制应用范围，例如以太坊的HTTP请求不可用。EVM具有专用的内置函数，例如可以轻松处理20字节地址加密的加密函数、用于自定义加密的模块化指数算法、读取区块数据、读取交易数据的函数，以及与block\_state交互的函数。以太坊EVM的安全性在于每次计算都要预先消耗gas，这增加了DoS攻击的成本，使得攻击者无法发动大规模的无效合约。EVM的主要编程语言是Solidity，智能合约用Solidity写好后，通过Solidity Compiler（solc）编译并生成EVM代码。合约语言的复杂性通过Solidity Compiler进行管理，但在架构层面，Solidity仍然是一种简单的基于堆栈的语言。

智能合约是在以太坊EVM上自动执行的合约代码，一般包括合约所有人、合约对象、合约条款、合约算法、合约触发条件等内容。对于可信电子证照应用，数据共享规则被转换为智能合约并部署在区块链上之后，常规共享条款和违约处理条款就可以自动履行，且执行过程由区块链完整记录，其执行状态可被随时查看和审计，从而提供一个公平、公正、公开的合约执行环境。此外，通过智能合约还可对参与方身份进行权限检查，针对交易者身份进行访问控制。

### 2.web3js

Web3.js是以太坊提供的一个JavaScript库，它封装了以太坊的RPC通信API，提供了一系列与区块链交互方法，使JavaScript与以太坊交互变得简单。

以太坊节点通过JSON-RPC提供外部访问接口，Web3.js是基于JavaScript和Node.js的JSON-RPC封装。

JSON-RPC是一个无状态、轻量级的远程过程调用协议（RPC），与传输协议无关，可使用socket、HTTP或者其它协议，它使用JSON（RFC4627）作为数据格式。

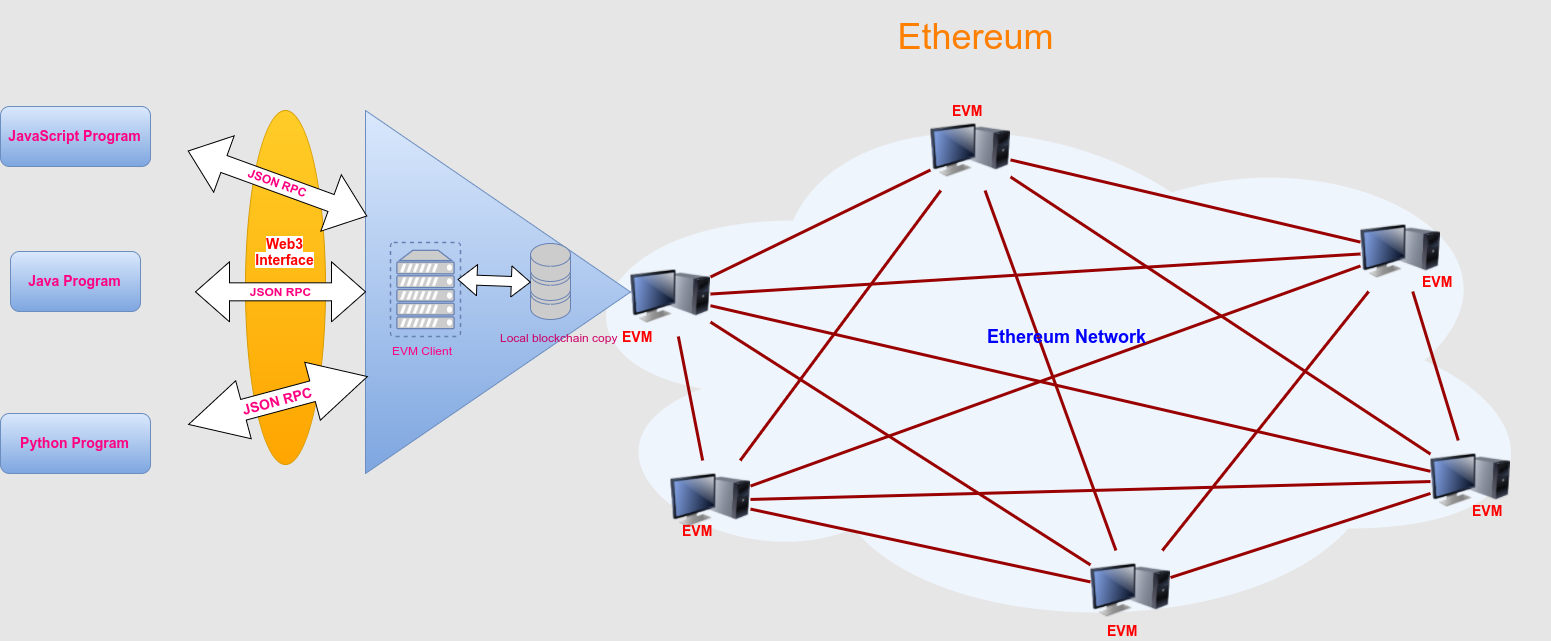
开发基于以太坊的区块链应用，一般有两个方面:

智能合约开发 - writing code that gets deployed to the blockchain with the Solidity programming language.

开发与区块链交互的网页或者客户端 - writing code that reads and writes data from the blockchain with smart contracts.

[Web3.js](https://web3js.readthedocs.io/en/1.0/) 帮助你完成第二个方面的工作，它是能帮助你完成发送以太币、读写智能合约的数据、创建智能合约等功能的一系库。

以下是客户端如何与以太坊交互的示意图



Web3.js 通过 [JSON RPC](https://en.wikipedia.org/wiki/Remote_procedure_call)（ "Remote Procedure Call"）远程过程调用来与区块链交互。Ethereum以太坊是由一些包含区块链上所有数据和代码的节点组成的点对点网络。 Web3.js 允许我们通过JSON RPC来向某个以太坊节点发送请求以读写数据，这有点像使用 jQuery with a JSON API 去读写 web server的数据。

## 五、合约内容

### 1.1基于区块链的毕业证书

1.学校部署毕业证书合约时会自动初始化赋值：学校的名称和学生的学位。（例如：清华大学、硕士研究生）

|  |
| --- |
| constructor() public certificate721("清华大学", "硕士研究生") {} |
| constructor (string memory name\_, string memory symbol\_) public {  \_name = name\_;  \_symbol = symbol\_;  \_host = msg.sender;  \_registerInterface(\_INTERFACE\_ID\_ERC721);  \_registerInterface(\_INTERFACE\_ID\_ERC721\_METADATA);  \_registerInterface(\_INTERFACE\_ID\_ERC721\_ENUMERABLE);  } |

2.将ERC721代币中唯一的TokenId和学生的信息绑定在一起，生成了唯一的毕业证书。

|  |
| --- |
| struct certificate {  string \_studenSex;  string \_studenBirthday;  string \_beginTime;  string \_endTime;  string \_academy;  string \_title;  uint32 \_timestamp;  }  mapping(uint256 => certificate ) public certificates; |

3.调用totalSupply方法可以查询到这个学校颁发的证书数量。

|  |
| --- |
| function totalSupply() public view override returns (uint256) {  return \_tokenOwners.length();  } |

4.调用name和symbol方法查询到学校的名字和证书的类型。

|  |
| --- |
| function name() public view override returns (string memory) {  return \_name;  }  function symbol() public view override returns (string memory) {  return \_symbol;  } |

5.高校通过调用addItem方法来给学生颁发毕业证书，往方法里面传入的参数为学生的基本信息（如：姓名，性别，出生日期，入学时间，毕业时间，学院，学位），生成的证书会打上时间戳。调用addItem方法时会被记录在合约的事件（event）里面。

|  |
| --- |
| function addItem(  address player, string memory tokenURI,  string memory studenSex,  string memory studenBirthday,  string memory beginTime,  string memory endTime,  string memory academy,string memory title  )  public onlyHost  returns (uint256)  {  \_tokenIds.increment();  uint256 newItemId = \_tokenIds.current();  \_mint(player, newItemId);  \_setTokenURI(newItemId, tokenURI);    certificates[newItemId].\_studenSex = studenSex;  certificates[newItemId].\_studenBirthday = studenBirthday;  certificates[newItemId].\_beginTime = beginTime;  certificates[newItemId].\_endTime = endTime;  certificates[newItemId].\_academy = academy;  certificates[newItemId].\_title = title;  certificates[newItemId].\_timestamp= uint32(block.timestamp);  return newItemId;  } |

6.调用getItem方法，可以通过tokenId（即证书标号）查询到证书的详细信息。

|  |
| --- |
| function getItem(uint256 \_ItemId) public view returns (  string memory, string memory,string memory,  string memory,string memory,uint32  ) {  return (  certificates[\_ItemId].\_studenSex,  certificates[\_ItemId].\_studenBirthday,  certificates[\_ItemId].\_beginTime,  certificates[\_ItemId].\_endTime,  certificates[\_ItemId].\_academy,  certificates[\_ItemId].\_timestamp);  } |

7.证书的所有者可以通过burn方法来销毁自己的证书。调用burn方法时会被记录在合约的事件（event）里面。

|  |
| --- |
| function burn(uint256 tokenId) public returns(bool \_bool) {  require(\_isOwner(\_msgSender(), tokenId), "ERC721: 证书只能被所有者销毁");  \_burn(tokenId);  certificates[tokenId].\_studenSex = "";  certificates[tokenId].\_studenBirthday = "";  certificates[tokenId].\_beginTime = "";  certificates[tokenId].\_endTime = "";  certificates[tokenId].\_academy = "";  certificates[tokenId].\_timestamp = 0;    return \_bool;  } |
| function \_burn(uint256 tokenId) internal virtual {  address owner = ownerOf(tokenId);  \_beforeTokenTransfer(owner, address(0), tokenId);  if (bytes(\_tokenURIs[tokenId]).length != 0) {  delete \_tokenURIs[tokenId];  }  \_holderTokens[owner].remove(tokenId);  \_tokenOwners.remove(tokenId);  emit Burn(\_msgSender(), tokenId, uint32(block.timestamp));  } |

8.查询数据

调用balanceOf方法，可以查询学生在这个学校所拥有的证书数量。

调用ownerOf方法通过tokenId（即证书标号）查询到证书的所有人。

调用tokenURI方法可以通过证书编号查询到证书所有人的名字。

|  |
| --- |
| function balanceOf(address owner) public view override returns (uint256) {  require(owner != address(0), "ERC721: balance query for the zero address");  return \_holderTokens[owner].length();  }  function ownerOf(uint256 tokenId) public view override returns (address) {  return \_tokenOwners.get(tokenId, "ERC721: owner 查询不到证书");  }  function tokenURI(uint256 tokenId) public view override returns (string memory) {  require(\_exists(tokenId), "ERC721Metadata: URI 查询不到证书");  string memory \_tokenURI = \_tokenURIs[tokenId];  if (bytes(\_baseURI).length == 0) {  return \_tokenURI;  }  if (bytes(\_tokenURI).length > 0) {  return string(abi.encodePacked(\_baseURI, \_tokenURI));  }  return string(abi.encodePacked(\_baseURI, tokenId.toString()));  } |

### 1.2基于区块链的学分认证

1.学校部署统计学分合约，生成合约实例之后，会初始化得到一些代币，代币的多少对应为学分的多少，如果代币数量不足，可以通过addTotalsupply来增加代币的数量，调用addTotalsupply方法时会被记录在合约的事件（event）里面，被永久保存在区块链中。部署合约的人为合约的所有人。

|  |
| --- |
| contract credit\_ccr20 is ERC20\_ccr20 {  constructor(uint256 initialSupply) public ERC20\_ccr20("Ahead", "AD") {  \_mint(msg.sender, initialSupply);  }  } |
| 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.所有人可以通过setAdministrators方法，增加几个备用的管理员账户，管理员账户和所有者一样有操控合约的最高权限。调用setAdministrators方法时会被记录在合约的事件（event）里面。

|  |
| --- |
| 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));  } |

3.管理员可以通过addDepartment方法，给相关部门进行身份认证，再将代币转账给相关部门，之后，相关部门就可以按照加分要求将代币转账给学生，以此给学生加学分，学生账户代币数量的多少对应为学生学分的多少。调用addDepartment方法时会被记录在合约的事件（event）里面。

|  |
| --- |
| 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));  } |

4.合约里面规定了只有合约创建者、管理员以及相关部门才能进行转账，也就是只有经过认证的部门才能给学生加学分，并且学生不能将学分转赠给其他人。每次转账也就是每次加学分的时候，都会被event记录下来。

合约里面规定了只有合约创建者、管理员以及相关部门才能进行代币销毁，也就是只有经过认证的部门才能给学生扣分，每次销毁代币也就是每次扣分的时候，都会被event记录下来。

|  |
| --- |
| 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);  } |
| 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);  } |

5.任何人都可以查询账户余额和学分变动记录，这样学分的信息就会公开透明，学生能够知晓学分的多少，以及加分扣分的理由。

|  |
| --- |
| interface IERC20 {  function totalSupply() external view returns (uint256);  function balanceOf(address account) external view returns (uint256);  function transfer(address recipient, uint256 amount, string memory reason) external returns (bool);  function burn(address account, uint256 amount,string memory reason) external returns(bool);    event Transfer(address indexed from, address indexed to, uint256 value,string reason);  event Burn(address indexed sender,address indexed from,uint256 value,string reason);  event Add(address indexed from, address indexed to, uint256 value,string reason);    event SetDepartment(address indexed setFrom,address indexed setTo,bool status,uint32 setTime);  event SetAdministrators(address indexed setFrom,address indexed setTo,bool status,uint32 setTime);  } |

## 六、前后端代码

### 6.1必要的准备：

1.必要文件：jquery和web3的js文件

|  |
| --- |
| <script type="text/javascript" src="js/jquery-3.5.1.min.js"></script>  <script type="text/javascript" src="js/web3.min.js"></script> |

2.点击按钮，连接以太坊账户。账号变更的话，前端显示的账户会变化。

|  |
| --- |
| <button class="enableEthereumButton">连接以太坊账户</button> |
| if (typeof window.ethereum !== 'undefined') {  console.log('MetaMask 已安装!');  }  console.log("isMetaMask：",ethereum.isMetaMask);  let accounts = [];  $(".enableEthereumButton").click(function () {  getAccount()  });  async function getAccount() {  accounts = await ethereum.request({ method: 'eth\_requestAccounts' });  const account = accounts[0];  $('.showAccount').html(account);  contract.methods.balanceOf(accounts[0]).call({from:accounts[0]}).then(  function (result) {  $('.ShowThisAccountBalance').html(result)  }  );  }  ethereum.on('accountsChanged', function (accounts) {  getAccount()  });  ethereum.on('chainChanged', (chainId) => {  console.log("ChainID为：",chainId.toString())  }); |

**3．加载智能合约的实例**

|  |
| --- |
| let web3 = new Web3(Web3.givenProvider || "ws://localhost:8546");  const contractABI = […];  var contract = new web3.eth.Contract(contractABI, "0xA830D086E03b351311cbDA3DC9b143290bDE6e7b"); |

### 6.2基于区块链的毕业证书：

**1. 获取证书的发布量、学校名称和学位名称。**

点击获取获取证书的发布量，js中通过jquery获取getTotalSupply的点击事件，接着执行web3js方法：contract.methods.totalSupply()，通过连接上的账户读取合约数据，并将结果展示在前端页面上。

点击获取获取获取学校名称，js中通过jquery获取getName的点击事件，接着执行web3js方法：contract.methods.name()，通过连接上的账户读取合约数据，并将结果展示在前端页面上。

点击获取获取学位名称，js中通过jquery获取getSymbol的点击事件，接着执行web3js方法：contract.methods.symbol()，通过连接上的账户读取合约数据，并将结果展示在前端页面上。

|  |
| --- |
| <P><button class="getTotalSupply">获取证书的发布量</button><span class="showTotalSupply"></span><br></P>  <p><button class="getName">获取学校名称</button><span class="showName"></span><br></p>  <p><button class="getSymbol">获取学位名称</button><span class="showSymbol"></span><br></p> |
| $(".getTotalSupply").click(function () {  contract.methods.totalSupply().call({from:accounts[0]}).then(  function (result) {  $('.showTotalSupply').html(result)  }  );  });  $(".getName").click(function () {  contract.methods.name().call({from:accounts[0]}).then(  function (result) {  $('.showName').html(result)  }  );  });  $(".getSymbol").click(function () {  contract.methods.symbol().call({from:accounts[0]}).then(  function (result) {  $('.showSymbol').html(result)  }  );  }); |

2.颁发证书

输入学生的信息，jquery通过input的id获取指定的值，再将指定的值传入web3js的方法contract.methods.addItem()里面，在此之前进行权限的判断，如果不是合约的所有人，不能调用颁发证书方法。等待交易的时候会加载动画。交易成功之后，前端会显示true，代表证书颁发成功。

|  |
| --- |
| <p>  学生账户地址:<br>  <input type="text" value="" id="addPlayer" placeholder="请输入学生账户地址"><br>  学生姓名：<br>  <input type="text" value="" id="addTokenURI" placeholder="请输入学生姓名"><br>  学生性别：<br>  <input type="text" value="" id="studenSex" placeholder="请输入学生性别"><br>  出生时间：<br>  <input type="text" value="" id="studenBirthday" placeholder="请输入出生时间"><br>  入学时间:<br>  <input type="text" value="" id="beginTime" placeholder="请输入入学时间"><br>  毕业时间:<br>  <input type="text" value="" id="endTime" placeholder="请输入毕业时间"><br>  院系:<br>  <input type="text" value="" id="academy" placeholder="请输入院系"><br>  学位:<br>  <input type="text" value="" id="title" placeholder="请输入学位"><br>  <button onclick="addItem()">颁发证书</button>  <span class="showadd"></span>  <!--逮虾户-->  <div id="loading"><img width="10%" src="img/accelerate.gif"></div>  </p><hr> |
| /\*2.addItem\*/  function addItem() {  let player=document.getElementById('addPlayer').value;  let tokenURI=document.getElementById('addTokenURI').value;  let studenSex=document.getElementById('studenSex').value;  let studenBirthday=document.getElementById('studenBirthday').value;  let beginTime=document.getElementById('beginTime').value;  let endTime=document.getElementById('endTime').value;  let academy=document.getElementById('academy').value;  let title=document.getElementById('title').value;  $('.showadd').html("");  console.log("学生证书数据为："+player,tokenURI,studenSex,studenBirthday,beginTime,endTime,academy,title);  contract.methods.host().call({from:accounts[0]}).then(  function (result2) {  if (result2.toLowerCase() == accounts[0]){  document.getElementById("loading").style.display="";  contract.methods.addItem(player,tokenURI,studenSex,studenBirthday,beginTime,endTime,academy,title).send({from:accounts[0]}).then(  function (result) {  console.log("add\_result:",result);  document.getElementById("addPlayer").value="";  document.getElementById("addTokenURI").value="";  document.getElementById("studenSex").value="";  document.getElementById("studenBirthday").value="";  document.getElementById("beginTime").value="";  document.getElementById("endTime").value="";  document.getElementById("academy").value="";  document.getElementById("title").value="";  $('.showadd').html(result.status);  document.getElementById("loading").style.display="none";  }  )  } else {  $('.showadd').html("您没有权限");  }  }  );  } |

2.查询证书

输入证书的编号，jquery通过id获取input的id获取指定的值，再将指定的值传入web3js的方法contract.methods.getItem()里面，在此之前进行判断，判断证书是否存在，如果不存在，前端显示，证书不存在；如果证书存在，通过连接上的账户读取合约数据，并将结果展示在前端页面上。

|  |
| --- |
| <p>  <input type="text" value="" id="getItem" placeholder="请输入证书编号">  <button onclick="getItem()">读取证书</button>  <span class="showItem"></span>  </p> |
| function getItem() {  let resItemId = document.getElementById('getItem').value;  console.log("查询TokenId为：" + resItemId);  contract.methods.getItem(resItemId).call({from: accounts[0]}).then(  function (result2) {  if (result2[3] == 0){  $('.showItem').html("<p>"+ "证书不存在" + "</p>")  } else {  contract.methods.tokenURI(resItemId).call({from: accounts[0]}).then(  function (result) {  $('.showItem').html("<p>"  + "学生名字：" +result + " "  + "学生性别：" +result2[0] +" "  + "出生时间：" + result2[1] +" "  +"入学时间：" + result2[2] +" "  +"毕业时间：" + result2[3] + " "  +"院系：" + result2[4] + " "  +"时间戳：" + result2[5] + " "  +"</p>")  });  }  });  } |

**3.销毁证书**

输入证书的id，jquery通过input的id获取指定的值，再将指定的值传入web3js的方法contract.methods.burn ()里面，在此之前进行证书存在性的判断，如果证书存在才能进行下一步处理，接着是对用户权限的判断，如果不是证书的所有人，不能调用销毁证书方法。等待交易的时候会加载动画。交易成功之后，前端会显示true。之后证书会被销毁。

|  |
| --- |
| <p>  <input type="text" value="" id="burn" placeholder="请输入证书编号">  <button onclick="burn()">销毁证书</button>  <span class="showburn"></span>  <div id="loading2"><img width="10%" src="img/accelerate.gif"></div>  </p> |
| function burn() {  let resItemId = document.getElementById('burn').value;  $('.showburn').html("");  console.log("查询TokenId为：" + resItemId);  contract.methods.getItem(resItemId).call({from: accounts[0]}).then(  function (result2) {  if (result2[3] == 0){  $('.showburn').html("<p>"+ "证书不存在" + "</p>")  }  else{  contract.methods.ownerOf(resItemId).call({from: accounts[0]}).then(  function (result3) {  if (result3.toLowerCase() == accounts[0]){  /\* 逮虾户 显示 \*/  document.getElementById("loading2").style.display="";  contract.methods.burn(resItemId).send({from: accounts[0]}).then(  function (result) {  console.log("burn\_result:",result);  $('.showburn').html(result.status);  /\* 逮虾户 隐藏 \*/  document.getElementById("loading2").style.display="none";  }  );  } else {  $('.showburn').html("<p>"+ "您不是证书所有者" + "</p>")  }  }  );  }  });  } |

**4.查询证书所有者。**

输入证书的编号，jquery通过id获取input的id获取指定的值，再将指定的值传入web3js的方法 里面，在此之前进行判断，判断证书是否存在，如果不存在，前端显示，证书不存在；如果证书存在，通过连接上的账户读取合约数据，并将结果展示在前端页面上。

|  |
| --- |
| <p>  <input type="text" value="" id="ownerOf" placeholder="请输入证书编号">  <button onclick="ownerOf()">查询证书所有者</button>  <span class="showOwnerOf"></span>  </p><hr> |
| function ownerOf() {  let resItemId = document.getElementById('ownerOf').value;  console.log("查询TokenId为：" + resItemId);  contract.methods.getItem(resItemId).call({from: accounts[0]}).then(  function (result2) {  if (result2[3] == 0){  $('.showOwnerOf').html("<p>"+ "证书不存在" + "</p>")  } else {  contract.methods.ownerOf(resItemId).call({from: accounts[0]}).then(  function (result) {  console.log("所有者:",result);  $('.showOwnerOf').html(result)  }  );  }  });  } |

5.查询证书发行记录

点击获取证书颁发记录按钮，jquery相应点击方法，接着执行web3js方法：获取，并将结果展示在前端页面上。

|  |
| --- |
| <p><button class="getEventAdd">获取证书颁发记录</button><span class="showEventAdd"></span><hr></p> |
| $(".getEventAdd").click(function () {  contract.getPastEvents('Add', {  fromBlock: 0,  toBlock: 'latest'  }, function(error, events){  console.log("-------------------------------------");  $('.showEventAdd').html("");  events.forEach(element => {  /\*  \* 转成JSON格式字符串，再将字符串转化json对象：  \* \*/  var jsonData = JSON.stringify(element.returnValues);  var json = JSON.parse(jsonData);  var json = eval("(" + jsonData + ")");  var json = (new Function("return " + jsonData))();  console.log("Event add：",json);  $('.showEventAdd').prepend(  "<p>"  + "from：" + json.from.slice(0,6) + "..." + json.from.substring(38)  + " to：" + json.to.slice(0,6) + "..." + json.to.substring(38)  + " tokenId：" + json.tokenId  + " 时间戳：" + timeConverter(json.addTime,1) + "</p>"  );  });  });  }); |

6.查询证书颁发记录

点击查询获取证书颁发记录按钮，js中通过jquery获取getEventTansfer的点击事件，接着执行web3js方法：获取合约事件contract.getPastEvents，从创世纪块读取到当前区块，将获得的数据经过格式化展示在前端页面上。

|  |
| --- |
| <p><button class="getEventAdd">获取证书颁发记录</button><span class="showEventAdd"></span><hr></p> |
| $(".getEventAdd").click(function () {  contract.getPastEvents('Add', {  fromBlock: 0,  toBlock: 'latest'  }, function(error, events){  console.log("-------------------------------------");  $('.showEventAdd').html("");  events.forEach(element => {  /\*  \* 转成JSON格式字符串，再将字符串转化json对象：  \* \*/  var jsonData = JSON.stringify(element.returnValues);  var json = JSON.parse(jsonData);  var json = eval("(" + jsonData + ")");  var json = (new Function("return " + jsonData))();  console.log("Event add：",json);  $('.showEventAdd').prepend(  "<p>"  + "from：" + json.from.slice(0,6) + "..." + json.from.substring(38)  + " to：" + json.to.slice(0,6) + "..." + json.to.substring(38)  + " tokenId：" + json.tokenId  + " 时间戳：" + timeConverter(json.addTime,1) + "</p>"  );  });  });  }); |

### 6.3基于区块链的学分认证：

1.发行代币

输入发行的代币数量，jquery通过input的id获取指定的值，再将指定的值传入web3js的方法里面，等待交易的时候会加载动画。交易成功之后，前端会显示true，代表代币发行成功。

|  |
| --- |
| <p>  <br>请输入增发的代币数:<br>  <input type="text" value="" id="addTotalSupply">  <button onclick="addTotalSupply()">增加总发行量</button>  <span class="showaddTotalSupply"></span>  <!--逮虾户-->  <div id="loading"><img width="10%" src="img/accelerate.gif"></div>  </p> |
| function addTotalSupply() {  let res=document.getElementById('addTotalSupply').value;  document.getElementById("addTotalSupply").value="";  console.log("add为："+res);  /\* 逮虾户 显示 \*/  document.getElementById("loading").style.display="";  contract.methods.addTotalSupply(res).send({from:accounts[0]}).then(  function (result) {  console.log("add\_result:",result);  $('.showaddTotalSupply').html(result.status);  /\* 逮虾户 隐藏 \*/  document.getElementById("loading").style.display="none";//隐藏  }  );  } |

2.设置管理

输入目标账户地址和是否，jquery通过input的id获取指定的值，再将指定的值传入web3js的方法里面，等待交易的时候会加载动画。交易成功之后，前端会显示true，代表设置管理成功。

|  |
| --- |
| <p>  <br>请输入地址:<br>  <input type="text" value="" id="setAdministratorsAdd">  <br>是否为管理员:<br>  <input type="text" value="" id="setAdministratorsBool">  <button onclick="setAdministrators()">设置管理员</button><hr>  <span class="showsetAdministrators"></span>  <!--逮虾户-->  <div id="loading3"><img width="10%" src="img/accelerate.gif"></div>  </p> |
| function setAdministrators() {  let resAdd=document.getElementById('setAdministratorsAdd').value;  let resBool=document.getElementById('setAdministratorsBool').value;  document.getElementById("setAdministratorsAdd").value="";  document.getElementById("setAdministratorsBool").value="";  console.log("新增管理员数据为："+resAdd,resBool);  /\* 逮虾户 显示 \*/  document.getElementById("loading3").style.display="";  contract.methods.setAdministrators(resAdd,resBool).send({from:accounts[0]}).then(  function (result) {  console.log("add\_result:",result);  $('.showsetAdministrators').html(result.status);  /\* 逮虾户 隐藏 \*/  document.getElementById("loading3").style.display="none";//隐藏  }  );  } |

3.设置部门

输入目标账户地址、部门名称和是否，jquery通过input的id获取指定的值，再将指定的值传入web3js的方法里面，等待交易的时候会加载动画。交易成功之后，前端会显示true，代表设置部门成功。

|  |
| --- |
| <p>  <br>请输入地址:<br>  <input type="text" value="" id="setDepartmentAdd">  <br>请输入部门名字:<br>  <input type="text" value="" id="setDepartmentName">  <br>是否为相关部门:<br>  <input type="text" value="" id="setDepartmentBool">  <button onclick="setDepartment()">设置相关部门</button>  <span class="showsetDepartment"></span>  <!--逮虾户-->  <div id="loading2"><img width="10%" src="img/accelerate.gif"></div>  </p> |
| function setDepartment() {  let resAdd=document.getElementById('setDepartmentAdd').value;  let resName=document.getElementById('setDepartmentName').value;  let resBool=document.getElementById('setDepartmentBool').value;  document.getElementById("setDepartmentAdd").value="";  document.getElementById("setDepartmentName").value="";  document.getElementById("setDepartmentBool").value="";  console.log("新增部门数据为："+resAdd,resName,resBool);  /\* 逮虾户 显示 \*/  document.getElementById("loading2").style.display="";  contract.methods.setDepartment(resAdd,resName,resBool).send({from:accounts[0]}).then(  function (result) {  console.log("add\_result:",result);  $('.showsetDepartment').html(result.status);  /\* 逮虾户 隐藏 \*/  document.getElementById("loading2").style.display="none";//隐藏  }  );  } |

4.加学分

输入目标账户地址、学分数值和加分原因，jquery通过input的id获取指定的值，再将指定的值传入web3js的方法里面，等待交易的时候会加载动画。交易成功之后，前端会显示true，代表学分添加成功。

|  |
| --- |
| <p>  <br>请输入学生地址:<br>  <input type="text" value="" id="recipient">  <br>请输入加分数值:<br>  <input type="text" value="" id="amount">  <br>请输入加分原因:<br>  <input type="text" value="" id="reason">  <button onclick="transfer()">加分</button>  <span class="showtransfer"></span>  <!--逮虾户-->  <div id="loading4"><img width="10%" src="img/accelerate.gif"></div>  </p> |
| function transfer() {  let resAdd=document.getElementById('recipient').value;  let resAmount=document.getElementById('amount').value;  let resReason=document.getElementById('reason').value;  document.getElementById("recipient").value="";  document.getElementById("amount").value="";  document.getElementById("reason").value="";  console.log("转账数据为："+resAdd,resAmount,resReason);  /\* 逮虾户 显示 \*/  document.getElementById("loading4").style.display="";  contract.methods.transfer(resAdd,resAmount,resReason).send({from:accounts[0]}).then(  function (result) {  console.log("transfer\_result:",result);  $('.showtransfer').html(result.status);  /\* 逮虾户 隐藏 \*/  document.getElementById("loading4").style.display="none";//隐藏  }  );  } |

5.扣学分

输入目标账户地址、学分数值和扣分原因，jquery通过input的id获取指定的值，再将指定的值传入web3js的方法里面，等待交易的时候会加载动画。交易成功之后，前端会显示true，代表学分扣除成功。

|  |
| --- |
| <p>  <br>请输入学生地址:<br>  <input type="text" value="" id="BurnFrom">  <br>请输入扣分数值:<br>  <input type="text" value="" id="BurnAmount">  <br>请输入扣分原因:<br>  <input type="text" value="" id="BurnReason">  <button onclick="burn()">扣分</button>  <span class="showburn"></span>  <!--逮虾户-->  <div id="loading5"><img width="10%" src="img/accelerate.gif"></div>  </p>  <hr> |
| function burn() {  let resAdd=document.getElementById('BurnFrom').value;  let resAmount=document.getElementById('BurnAmount').value;  let resReason=document.getElementById('BurnReason').value;  document.getElementById("BurnFrom").value="";  document.getElementById("BurnAmount").value="";  document.getElementById("BurnReason").value="";  console.log("销毁的数据为："+resAdd,resAmount,resReason);  /\* 逮虾户 显示 \*/  document.getElementById("loading5").style.display="";  contract.methods.burn(resAdd,resAmount,resReason).send({from:accounts[0]}).then(  function (result) {  console.log("burn\_result:",result);  $('.showburn').html(result.status);  /\* 逮虾户 隐藏 \*/  document.getElementById("loading5").style.display="none";//隐藏  }  );  } |

6.查询学分数值

输入要查询学分的账户地址，点击查询学分按钮,触发getBalance()的方法，js获取输入的参数，传入web3js方法contract.methods.balanceOf()，通过连接上的账户读取数据，并将结果展示在前端页面上。

|  |
| --- |
| <p>  <br>请输入查询地址:<br>  <input type="text" value="" id="getBalance">  <button onclick="getBalance()">查询学分</button>  <span class="showBalance"></span>  </p> |
| function getBalance() {  let resAdd=document.getElementById('getBalance').value;  console.log("查询地址为："+resAdd);  contract.methods.balanceOf(resAdd).call({from:accounts[0]}).then(  function (result) {  console.log("学分为："+result);  $('.showBalance').html("学分为："+result)  }  );  } |

7.查询学分变动记录

点击查询加分记录按钮，js中通过jquery获取getEventTansfer的点击事件，接着执行web3js方法：获取合约事件contract.getPastEvents，从创世纪块读取到当前区块，将获得的数据经过格式化展示在前端页面上。

|  |
| --- |
| <p><button class="getEventTansfer">查询加分记录</button><span class="showEventTansfer"></span><br><hr></p> |
| $(".getEventTansfer").click(function () {  contract.getPastEvents('Transfer', {  // filter: {from: "0x990f7F8AabF6cC7B13b27f074Bce0D572c58D532"},  fromBlock: 0,  toBlock: 'latest'  }, function(error, events){  console.log("-------------------------------------");  $('.showEventTansfer').html("");  events.forEach(element => {  /\*  \* 转成JSON格式字符串，再将字符串转化json对象：  \* \*/  var jsonData = JSON.stringify(element.returnValues);  var json = JSON.parse(jsonData);  var json = eval("(" + jsonData + ")");  var json = (new Function("return " + jsonData))();  console.log("Event Transfer：",json);  $('.showEventTansfer').prepend(  "<p>"  + "部门：" + json.from.slice(0,6) + "..." + json.from.substring(38)  + " 受益人：" + json.to.slice(0,6) + "..." + json.to.substring(38)  + " 分数：" + json.value  + " 加分原因：" +json.reason + "</p>"  );  });  });  }); |

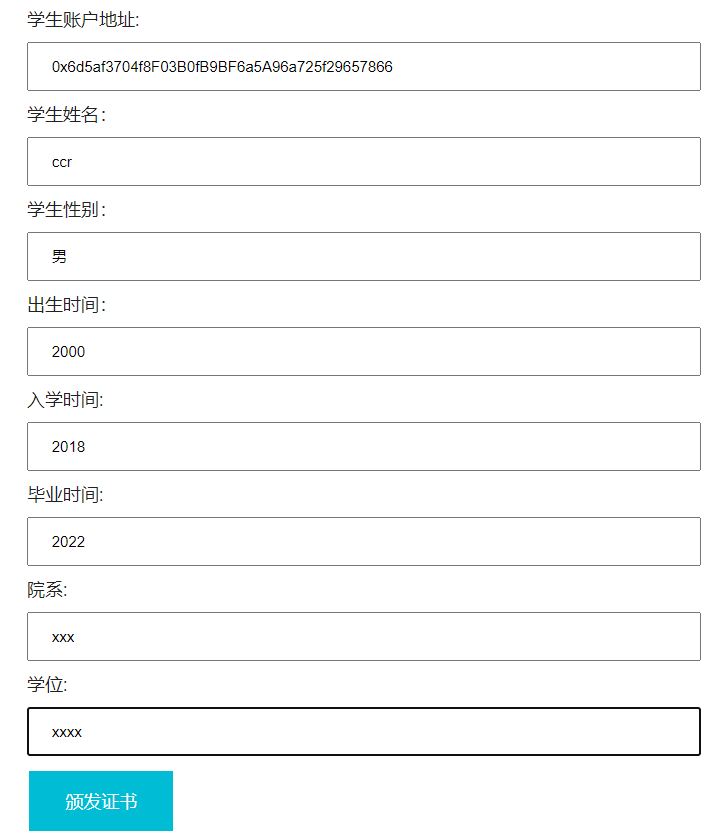
## 七、前端操作

### 7.1基于区块链的毕业证书

1.颁发证书

* 高校输入毕业生的数据，点击颁发证书

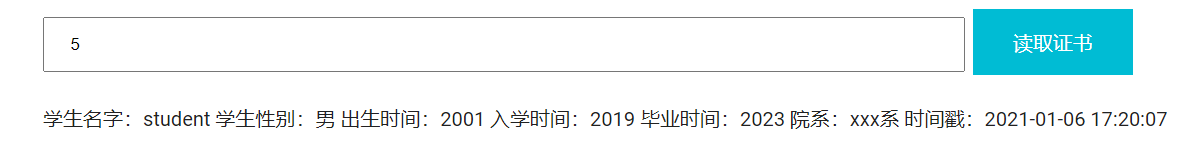
即可颁发证书



2.读取证书

* 输入证书的编号，点击读取证书

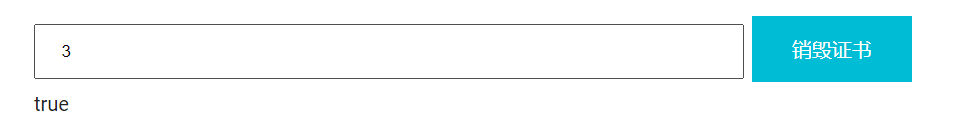
即可获取证书的详细信息



3.销毁证书

* 证书的所有者输入证书的编号，点击销毁证书

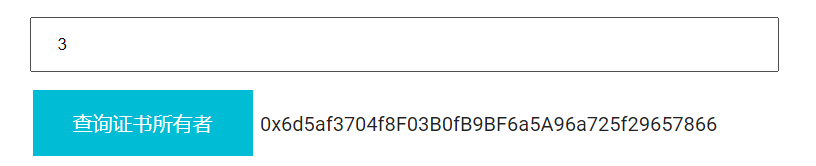
即可销毁证书



4.查询证书所有者

* 输入证书的编号，点击查询

即可读取证书所有者的地址



5.获取证书颁发记录

* 点击获取

即可读取证书颁发记录



6.查询证书销毁记录

* 点击获取

即可读取证书销毁记录



### 7.2. 基于区块链的学分认证

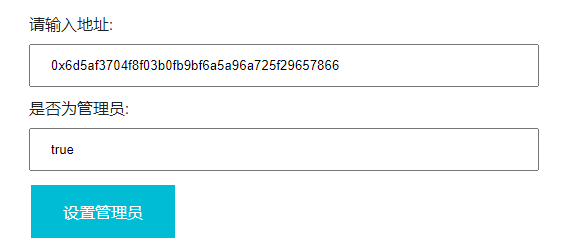
1.设置系统的管理员

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



2.设置相关部门

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



3.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**



4.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，

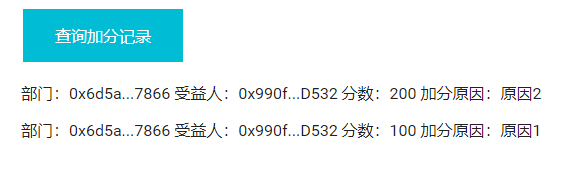
即可**扣除学生学分**



5.查询加分记录

* 点击查询加分记录按钮，

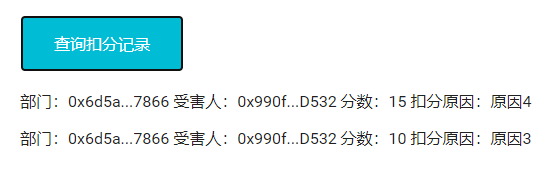
即可获取学分添加记录



6.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



7.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



8.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**



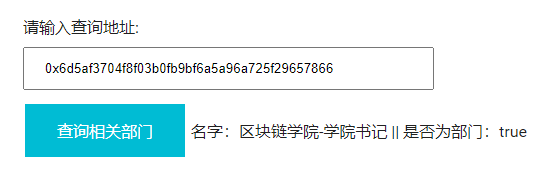
9.查询相关部门

**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**



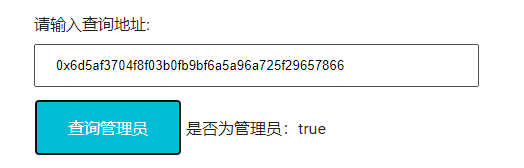
10.查询管理员

**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**



## 八、总结和体会

如果把人们在不同学习阶段取得的成果比喻成一枚枚徽章，那使用区块链技术就能实现“开放徽章”。“开放徽章”是用来呈现个人通过完成特定的项目、课程或者实践活动而获得的技能、兴趣和成就的一种“数字徽章”。区块链的防篡改系统是“开放徽章”最佳的存储空间。最终，见证我们学习成就的不只是那一张学历证书，还可以是一条长长的“徽章走廊”。

在区块链上，学习社区成员在学习交流过程中的价值由区块链通用积分体现。通过将自己在学习过程中的优秀作品、笔记、回答等发布到基于区块链的学习社区，利用其不可篡改的特点，所有过程得以长久保存，并得到相应积分奖励，此积分可以实现全球跨社区性流通。成员之间在进行答疑、合作等相互交流的时候，如答疑、合作等，可以通过发布积分悬赏的形式激励其他人参与，或者用自己的积分换取社区学习资料与服务。这一积分体系不仅注重参与量，还可将其他成员的认可纳入衡量标准，比如赞同、感谢、邀请等，进而保证内容质量更高。这样一来，学习社区成员在自我学习、互助交流过程中产生的价值能更好地体现，整个社区的参与度也会大大提升。

在区块链赋能教育后，将改变过去教育评价方式单一的局面，衡量受教育者的标准不再只局限于考试这种存在弊病的固定形式，而是延长至整个教育过程。孩子的每一次成长都会被真实记录并永久保存，教育评价体系由此变得多样化，与之对应的人才培养方式也变得更加个性化，真正做到对个体因材施教，最大限度地发挥每个人的优势。

可以看出，无论是学习轨迹记录，还是社区积分积累，教育领域是区块链激励制度的最佳运用地之一。展望未来，这种激励具有长期性，甚至会贯穿人的一生。不仅仅是在学校的时间里，在进入社会之后，在工作中也同样需要不断吸收新知识，做到时学时新。在实践运用中，可将这种激励落在实处，比如在体制内，可以尝试将这种学习积分以一定权重纳入考核评职称；在体制外，也可将雇员的学习经历纳入晋升加薪的考核体系，从而让终身学习成为每个人的信仰。