

|  |  |
| --- | --- |
| **题目** | 基于区块链的公益献血活动 |
| **专业** | 软件工程（区块链方向） |
| **班级** | B190601 |
| **姓名** | 聂萍萍 |
| **学号** | 2019000102095 |

目录

[一、 背景 4](#_Toc4084)

[1. 发展历史 4](#_Toc29202)

[2、 发展现状 6](#_Toc22898)

[二、 传统献血平台 10](#_Toc19096)

[传统献血流程图： 10](#_Toc20490)

[传统献血结构图： 11](#_Toc2295)

[三、 传统献血业务 12](#_Toc18069)

[四、 传统献血存在的问题 13](#_Toc18424)

[ 信息缺失、混乱 13](#_Toc31430)

[ 效率较低 13](#_Toc19559)

[ 血浆易浪费、抛弃 14](#_Toc11411)

[ 难以溯源 14](#_Toc4946)

[五、 应用分析 14](#_Toc30204)

[六、 链改后的献血平台 19](#_Toc17213)

[链改后献血流程图： 19](#_Toc3565)

[链改后献血结构图： 20](#_Toc8180)

[七、 链改后献血业务 22](#_Toc14159)

[ 身份核查 22](#_Toc25989)

[ 体格检查 22](#_Toc22784)

[ 血液检测 23](#_Toc10729)

[ 献血 23](#_Toc5400)

[ 领证 23](#_Toc23035)

[ 信息上链 23](#_Toc22382)

[八、 链改后的优势 25](#_Toc28752)

[ 数据共享 25](#_Toc15158)

[ 减少错误不可篡改 26](#_Toc14676)

[ 可溯源 26](#_Toc19251)

[ 监管透明 26](#_Toc14044)

[九、 功能页面展示 27](#_Toc32321)

[1、献血功能页面结构图 27](#_Toc32357)

[2、 首页 28](#_Toc23212)

[3、 献血自助登记 29](#_Toc20653)

[4、 献血流程指引 29](#_Toc11097)

[5、 献血地点查询 31](#_Toc3512)

[6、 检测结果查询 32](#_Toc22052)

[7、 献血注意事项 33](#_Toc6706)

[8、 信息证书上链 34](#_Toc29252)

[十、 关键技术及方法 35](#_Toc18636)

[十一、 结束语 53](#_Toc1713)

[十二、 总结与体会 54](#_Toc12154)

1. **背景**
2. **发展历史**

输血作为一种有效的治疗手段发展迄今已有百余年历史。在古代，人们对血液的理解是未知且毫无科学依据的。同时，输血救人也并不是一躇而就的，而是经过了漫长且曲折的道路才发展至今。

无偿献血是指公民自愿、无报酬地提供自身血液的行为，它是国际卫生组织、国际红十字会推崇的献血形式。1946年，红十字会与红新月会首先提出了无偿献血的原则。1948年国际红十字会组织向各国呼吁，医疗献血采取无偿献血免费输血的原则。1973年在德黑兰召开的第22届国际红十字会大会指出，出于人道主义动机志愿的献血，不领取任何报酬的无偿献血，才是血液需要最安全有效的道路。并且督促成员国加速自愿无报酬献血事业的发展。经过几十年的不懈努力，世界上很多国家都从过去的有偿献血制度，逐渐向义务性无偿献血制度过渡，最终实现了公民无偿献血。如原西德、日本、瑞士、加拿大、澳大利亚等国家都先后基本实现了公民无偿义务献血。

回顾新中国输血事业发展的历程，一直伴随着献血制度的与时俱进。1921年至1932年，北京协和医院在临床开展直接输血，自此中国输血疗法的新大门终于打开了。在战争一茬接一茬的年代，输血疗法成功地挽救了无数人地生命，此后更像火箭一样在国内开始飞速发展。从1944年开始，我国在昆明、南京、沈阳等多个城市相继建立起血库。1958年我国第一个初具规模的现代化血站在天津的中国医学科学院输血及血液学研究所诞生（该血站是现在位于成都的中国医学科学院输血研究所的前身）。该血站从一成立就开始进行采血、制备血液成分和生产血浆蛋白制品（包括白蛋白、丙种球蛋白和凝血酶原复合物等），并且提供临床应用。那时所需的血源大部分是来自城市里地大专院校师生、工厂职工以及附近的农民，实施的是有偿献血制度。

随着我国经济建设和医疗卫生事业的发展，临床用血量逐年攀升。在上世纪六十年代和七十年代，由于当时缺乏完善的管理体制及卫生监督体系，部分地区不规范使用采血用具，导致病毒交叉感染，血源组织发生混乱，输血事故不断发生，造成了非常严重的后果和影响。在此情况下，卫生部专门组织了一个调查组，在中国医学科学院输血研究所肖星甫所长率领下到中南、西南和东北地区调查输血工作现状，发现不少地区献血管理混乱、卖血成风，有的医院甚至不按规定采血，而是随便降低体检标准，甚至不体检，不做血液化验。有些地区血霸横行，对献血者敲诈勒索。当时，使用的输血技术和输血器具也落后于国外，操作规程执行不严，病人输血反应多，甚至因血液被污染造成死亡。对于以上出现的问题，卫生部写成报告呈报国务院。1978年11月24日，国务院颁发了《国务院批转卫生部关于加强输血工作的请示报告的通知》（国发[1978]242号），通知指出“实行公民义务献血制度，是改变我国输血工作落后面貌，解决医疗和战备储备用血的一项根本办法”，之后又在1998年10月1日起开始实施《中华人民共和国献血法》。自此无偿献血制度正式启用，同时对医疗临床用血的安全制定了相应的法律法规。

1. **发展现状**

血液一直被视为活力与健康之源，输血更是现代医学治病救人的重要手段。外伤性失血、产后大出血、严重烧伤、各种血液病和施行大型手术都离不开输血治疗。

研究表明，一个人身体内部的全部血液，其实只有80%左右循环流动，其余20%在体内储存用于补充，献出少量血液，不会影响身体健康。而且定期献血的人，造血功能比不献血的人旺盛。从新陈代谢的角度讲，红细胞平均寿命为120天，白细胞为9至13天，血小板为8至9天。正常情况下，每人每天约有40毫升的血液细胞衰老死亡，一个正常健康成人身体的血液总量大约是体重的8%，一个50公斤体重的人约有4000毫升血液，如一次献血200毫升至400毫升，只占人体总量的5至10%，身体能很快自动调节予以补充血液，人体本身有旺盛的新陈代谢能力，献血后，骨髓受到刺激，会加速血细胞的生成。献血200毫升，只需两周左右时间，各种血液成分就能恢复到献血前的水平。因此，献血对人体正常生理活动有利。人体血液中55%是血浆，而血浆中91%至92%是水分。所以，对健康献血适龄者（18至55周岁）来说，献血不会影响身体健康。医学研究还表明，定期适量献血，有降低血液粘稠度的作用，可使血浓度降低，有利于心脏，更有利于长寿，不易衰老。

据世界卫生组织（WHO）统计，献血人数占一国人口总数的4%，即可以满足全国临床用血的需要。瑞士每年参加献血的人数约占全国人口的9%，美国每年参加献血的人数约占8%，其次是日本，约占7%。但是据世界卫生组织（WHO）的数据表明，在全球每年1.125亿次献血总量中，超过一半来自总人口占世界人口19%的高收入国家。其中，高收入国家的献血率达4.54%，中等收入国家达1.01%。现在，中国只是初步达到中等收入国家的水平，也刚好达到世界卫生组织（WHO）推荐的1%的标准。从中国的情况可以看到，1%的人献血提供了全国人民的用血需求，也表明，这个献血量的供大大低于求，随时都有可能造成临床医疗用血的“血荒”。

目前，无偿献血已经成为衡量一个国家社会文明程度的标志之一。各国政府都十分重视和关心无偿献血，无偿献血的宣传活动在各国开展得丰富多彩。现在，世界上很多国家已达到医疗用血全部或大部分来自无偿献血。如经济发达的美国、日本、加拿大、德国、澳大利亚，也有经济欠发达的阿尔及利亚、坦桑尼亚、尼泊尔、缅甸等，都实行了无偿献血制度。并且世界各国都还非常重视无偿献血的意义和血液生理知识的宣传教育。在日本、法国、德国等国家号召人们参加献血的宣传画、标语口号等到处可见，有关献血的小册子在血站或红十字会可以自由索取，献血的好人好事经常通过电视台、电台、报纸进行报道，无偿献血已经成为一种良好的社会风气。

许多国家也还非常重视对青少年的献血知识教育。日本允许家长带孩子到血站，父母献血时孩子在一旁观看。澳大利亚等国家在中小学设有献血知识课，学校定期带学生到血液中心采血现场参观，让学生了解献血的全过程，使他们从小就懂得献血是每个人应尽的社会义务和献血不影响健康的道理，长大后成为一名自愿的光荣献血者。

而在中国，献血作为我国公共卫生事业的重要组成部分,无偿献血事业的发展意义重大。安全、足量地保证临床用血并确保献血者的健康是采供血机构的重要职责。自1998年10月1日《中华人民共和国献血法》实施以来,经过各级政府,卫生行政部门,采供血机构以及其它一些社会组织的不懈努力,我国无偿献血目前已经基本能够满足临床用血。各大城市血站招募血源，完成了从有偿献血、义务献血到无偿献血制度的转变。据国家卫健委发布的数据显示，2018年我国无偿献血人次接近1500万，血液采集量稳步增长。这不仅对推进中国输血安全具有重大战略意义，也体现了人们思想观念的转变与社会文明的进步。

据介绍，目前全国已经超过28个省成立了省级的临床用血质控中心，并把医疗用血作为医疗质量评价的重要指标。通过不断推广临床合理用血的理念和经验，患者术中出血和输血不良反应的发生有效减少，保障了患者的用血安全。

随着社会的进步与发展，临床输血安全得到全社会的广泛关注和有关管理部门的高度重视。输血是现代医学不可或缺的重要手段，但与多数临床治疗方法一样存在一定的风险。例如献血填写过程的漏填和错填让很多血源的信息不明确，造成血源的溯源信息不准确以及档案管理的不完整性。

安全是血液管理和使用的底线和生命线。多年来我国不断加强血站质控体系建设，积极推进信息化建设，已建立高危献血人员屏蔽制度和血液的冷链管理制度，每年还要组织全国范围内的安全技术核查。2015年以来，血液核酸检测实现了全覆盖，有效缩短了病毒的窗口期，进一步提升了我国血液管理的安全水平。

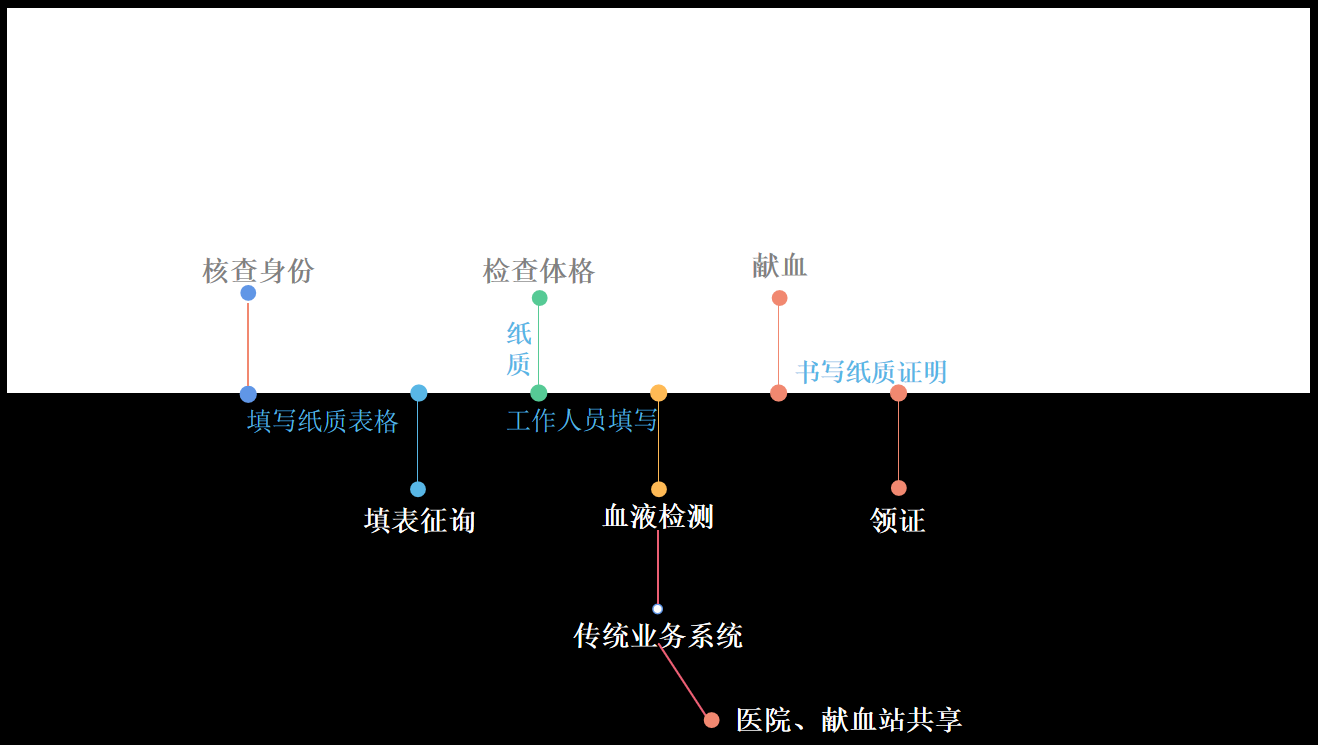
但是我国的献血存在一些问题。比如血站使用的输血技术和输血器具落后于国外，操作规程执行不严，病人输血反应多，无法排除所提供血液携带病原体的可能，甚至因血液被污染造成死亡。在涉及医疗事故时，血站要作为承担法律责任方，而无偿献血登记表作为一份最原始的记录就成为的法律依据之一。因而它的内容资料十分的重要。

但是对于血液的获取、存储及供应的管理方面，目前的管理系统都只是比较粗放式的服务器或客户端类型的查询式，这会导致血液获取、存储及供给管理的不够精细，造成造成血液的浪费。血液的获取方也就是血站等需要了解所采集血液的血型、品种及数量需求的类型，并且需要了解目前存储的血液的血型、品种及数量及其随时间的变化情况，血液的使用方也就是医疗机构需要持续的了解血液的血型、品种及数量的真实数据，医疗管理机构需要了解血液的真实的类型和数量的变化。如果采用单一服务器诸如服务器或客户端类型管理系统，将可能存在服务器宕机、停电等问题，对于单一服务器管理数据也存在着被人篡改的可能。因此，我们提出采用基于区块链模式的献血管理系统，将需要管理的数据去中心化存储，新献血管理系统将可以避免服务器宕机而数据不在线，也避免了数据被篡改而不真实的问题。

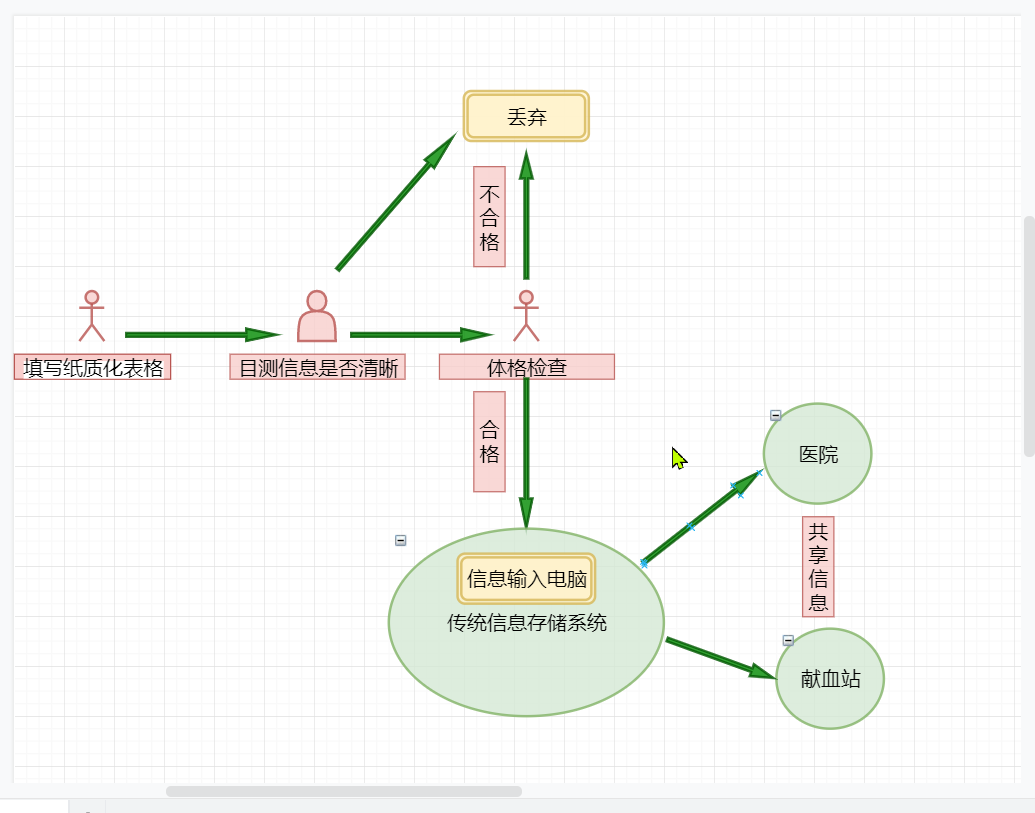
新的献血系统的目的主要在于提供一种基于区块链的血液管理方法，至少解决由于现有技术采用集中式数据库对多个不同部门之间的血液数据进行管理处理，容易被篡改或血液数据存在安全隐患的技术问题。它不仅仅能对献血者的个人信息进行简单的登记管理、反馈，提供给献血者便利的服务，而且还能为献血站和医院提供一个献血采集的依据，比如说什么血型紧缺等信息。

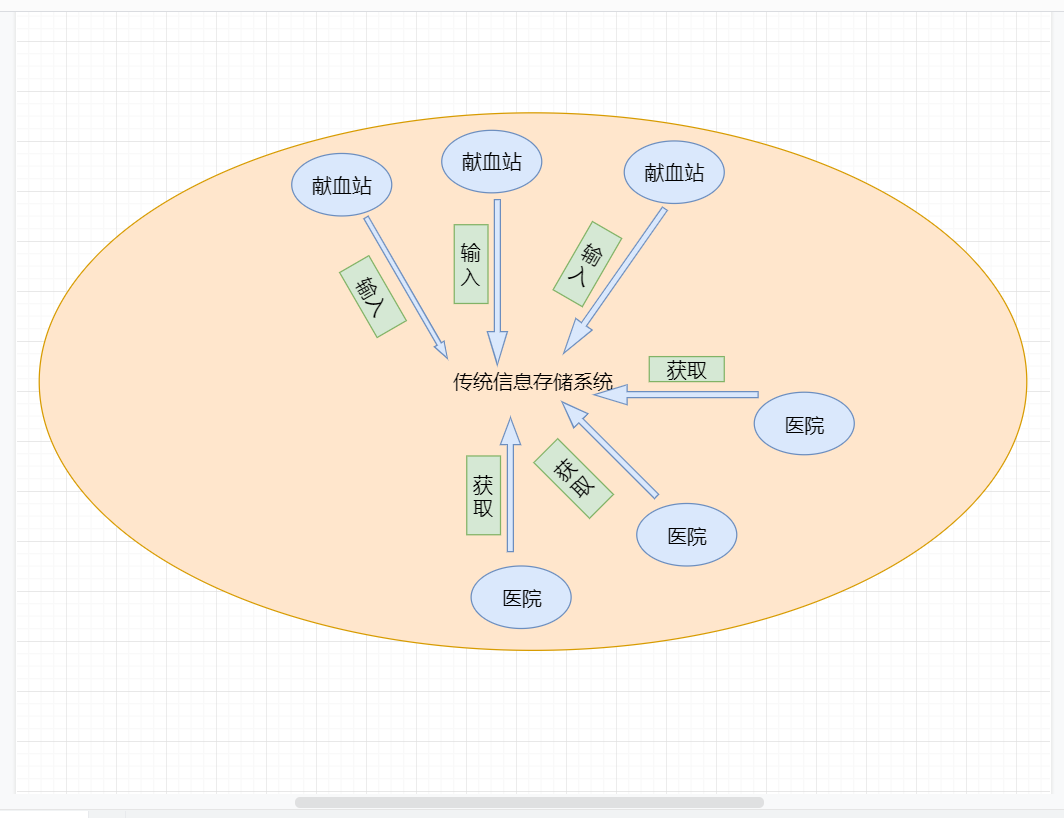
1. **传统献血平台**

**传统献血流程图：**



**传统献血结构图：**





1. **传统献血业务**

传统的信息存储系统是采用纸质表格和传统的联网系统。传统献血的流程是：首先，献血者需要先进行基本的身份核查，确认是否为本人；然后接着进行基本信息的填写（如年龄，性别，身份证号码等等），基本信息的填写采用的是纸质表格的方式，由相关的工作人员进行监督，献血者本人进行填写；填写完成之后进行对应的体格检查（如身高、体重、血压等等），检查是否符合献血的要求，相关的工作人员则将检查的信息以手写的方式填入表中，不符合条件的人不能进行献血，符合条件的人则随后进行血液检测；血液检测检查的是该血液是否携带病毒，确认该血液是否健康，血液检测完成后，相关的工作人员将血液信息和身份信息一并输入到传统的联网系统，该传统的联网系统主要由医院和血液站两方合作，随后血液检测合格的献血者进行献血，献血完成后由相关医护人员书写纸质证书给献血志愿者，以表示感激。

传统信息存储系统对于输入的献血者的信息进行存储，以便医院需要对应的血源的人员信息，而基本的系统结构为纸质表格，经过工作人员目测信息填写是否清晰合理，接着经过体格检测和血液检测决定是否将信息输入到电脑，存储到电脑，存储献血者的信息。而医院可以通过需求可以向信息存储系统进行特定信息的提取，或向献血站提出医院所紧缺的血液信息。献血站可以进行输入信息增加献血信息数据，也可以通过医院提出的需求而增加献血。

1. **传统献血存在的问题**

* **信息缺失、混乱**

传统献血信息存储系统以填写表格登记的形式会因为部分献血者或相关工作人员的字迹模糊或者纸张的原因以及填表过程的错填而造成出现信息缺失和信息错误的现象。

* **效率较低**

由于工作人员的主观因素大家的工作效率较难以统一；传统填表过程中采用纸质表格加电子登记系统的方式，纸质填表的效率不及电子表单，因而效率较低下。

* **血浆易浪费、抛弃**

血站可能会由于部分献血者的身份信息不明确，或信息模糊，甚至血源未知而不使用，从而造成出现血浆的浪费、抛弃的现象。

* **难以溯源**

利用传统的献血系统所留下的身份信息血液信息进行血源追溯，可能会因为部分身份信息不明、信息错乱而无法进行溯源，最后导致追溯不成功。

1. **应用分析**

目前市场上还没有完善的基于区块链的公益献血服务系统。无法形成献血者、献血站、医院的一个完整的献血数据体系。而区块链技术可以为政府、第三方机构的献血服务提供统一可信的服务平台，形成完整的献血服务链条，积累大量的数据信息，构建诚实可信的献血信用平台，为献血者、献血站、医院提供献血服务，其市场潜力巨大。

什么是区块链？区块链是比特币的一个重要概念，利用块链式数据结构来验证与存储数据；利用分布式节点共识算法，来生成和更新数据；利用密码学的方式，保证数据传输和访问的安全；利用由自动化脚本代码组成的智能合约，来编程和操作数据的分布式基础架构与计算方式。

区块链技术作为比特币和其他虚拟货币底层核心技术，是一种开放的分布式分类帐本，它的实质是不同的节点共同参与的分布式数据库，是一个开放式的公共账簿（没有所谓的中心化，是大家一起维护的一个）账簿。可以有效地以可验证和永久的方式记录双方之间的交易信息，而分类帐本身也可以被编程以自动触发交易。然后从数据包形成区块，中间有一个加密的哈希值计算（密码学）技术，把不同时间段的交易信息连接起来，形成了区块链。

从数据角度来看，区块链是一种几乎不可能被更改的分布式数据库，这里的”分布式”不仅体现为数据的分布式存储，也体现为数据的分布式记录（即由所有参与者共同维护所有的数据信息）。

从技术角度看，区块链并不是一种单一的技术，而是多种技术整合的结果，这些技术以新的结构组合在一起，形成一种新的数据记录，存储和表达的方式。

区块链技术具有五个基本特点。

特点一：分布式的数据库，去中心化、去信任的机制。区块链上的每一方都可以访问整个数据库及其完整的历史记录。没有单一的一方控制数据或信息。每一方都可以直接验证其交易合作伙伴的记录，而不需要中间人。任何人都可以参与到区块链网络中去，每一台设备都能作为一个节点，每个节点都允许获得一份完整的数据库拷贝信息，节点之间会基于一套共识机制，通过竞争计算共同维护整个区块链。去中心化是区块链技术的一个颠覆性特点，它无需中心化代理，实现了一种点对点的直接交互，使得高效率、大规模、无中心化代理的信息交互方式成为了现实。

特点二：点对点的传输。在区块链的点对点的传输网络中，每一个用户既是一个节点，也具有服务器的功能，它们之间是平等的，网络中的所有节点都可以互相传输，整个网络中没有任何的一个中心点，网络中的任何两个点都可以进行数据传输。传输通信直接在对等体的节点之间发生，而不是通过中心节点，每个节点可以存储并转发信息到所有其他的节点中去。在这个点对点的传输过程中，解决了信息不对等以及信任的问题，传统的互联网中心化节点的作用被区块链自身的系统和分散的节点所取代，价值传递有望在用户和用户之间，节点和节点之间进行。

特点三：透明的匿名性。区块链的运行规则是公开透明的，所有的数据信息也是公开的，每笔交易都是对所有节点公开可见，由于节点之间是去信任的，因此节点不需要公开身份，每个参与的节点都是匿名的。任何有权访问系统的用户都可以看到每个事务及其关联值。区块链上的每个节点或用户都有一个唯一的30以上的字母、数字组成的地址，用于标识自身。用户可以选择保持匿名或向他人提供其身份证明。而区块链的节点则加以发生在这些地址之上。

特点四：记录的不可篡改性，且可溯源。一旦在数据库中输入事务并更新了帐户，则不能更改记录，因为它们链接到它们之前的每个交易记录（因此称为“链”）。采用各种不同的算法以确保数据库中的记录是永久的、按时间顺序排序的，并且对于网络上的所有其他节点都是可以访问的。单个节点甚至多个节点对数据库的修改无法影响其他节点的数据库，区块链中的每一笔交易都通过密码学方法与两个相邻的两个区块串联，因此可以追溯每一笔交易的所有记录。

特点五：计算逻辑。分类帐本的数字性质意味着区块链交易可以关联到计算逻辑、本质上是可编程的。因此，用户可以设置自动触发节点之间交易的算法和规则。通过区块链的广播机制，一方写入数据，多方进行共识，然后各自写入账本，一定是能够保证写入的数据的一致性的。区块链的智能合约是多方同步，多方一致的，而且智能合约不存在随机性，因此同样的输入，一定对应同样的输出，多方之间能够保证逻辑一致。

因为有了数据一致和逻辑一致以及信息的不可篡改性，所以区块链非常适用于多个互不信任的主体之间的信息共享，以及实现穿透式的监管方式。一方面是数据的多方同步，信息的不可篡改，就可以有效避免不按照标准运营业务的情况，减少谎称数据丢失、或者提前修改数据的情况。另一方面是多方逻辑是一致的，将业务标准使用智能合约写到区块链上，不符合标准的业务数据无法上链，业务无法开展，就可以有效的避免不符合规则的业务的产生。

在献血应用系统中引入区块链技术，可以借助区块链的去中心化同步记账、献血者身份查验、献血信息数据加密和数据不可篡改等特征，可以确保血液信息安全可靠，可信任且可追溯，使各参与方共同建造、共同维护、共同监督，从而满足献血者的知情权、监督权，增强献血的客观性与可信度。

区块链与献血的匹配度分析如下所示。

**（1）打破信息壁垒**：区块链采用分布式数据库架构，不需要跨部门、跨地区的数据集中，不需要构建多级数据管理中心。区块链构建新型的献血协作模式，提高其协同的流通效应，促进献血者、献血站、医院间信息的流动。与中心数据库集中共享模式比较起来，这种模式建设难度更低，实施可行性更高。

**（2）可信的存在性证明：**发证机构通过对电子献血证文件的签名，将发证记录保存在区块链上，可证明其献血证的合法存在，而电子献血证文件经过哈希运算备份保存在区块链上，数据安全性高，不易作假，从而减少了献血证的伪造现象。

**（3）献血者可监督血液的使用过程：**区块链通过交易打包的形式，保存了不可篡改的血液交易记录。献血者可以查看自己的血液在何时何地被查验和使用过，知道自己的血液用途，避免了血液在不知情的状况下被他人使用，大大降低了使用血液的风险。

**（4）可审计性高：**电子献血证文件经过哈希运算备份保存在区块链上，数据安全性高。由于加盖了时间戳使得献血证的使用记录可追溯，可为审计部门提供更为可信的数据支撑，避免了重复认证，减少了资源浪费。区块链还有一个技术便是不可篡改性，不用经过第三方批准，便能**证明自己就是自己**，在用血时可报销，可立即使用，这是无形中对献血者的最大肯定，自己的献血信息变成了一种隐形财富，更坚定了献血者的爱心忠诚度。比如说献血者在A地献了血，去B地医院用血，便要用各种资料来证明“我是我”，是我献的血，是我本人要用，若是亲友要用，更要各种证明“我妈是我妈”，是我妈要用我的血。这种繁琐，不灵活的操作手段，有了区块链技术，便都解决了，因为除了区块链技术的可追溯性，它的数据不可篡改性，更能证明你就是你。

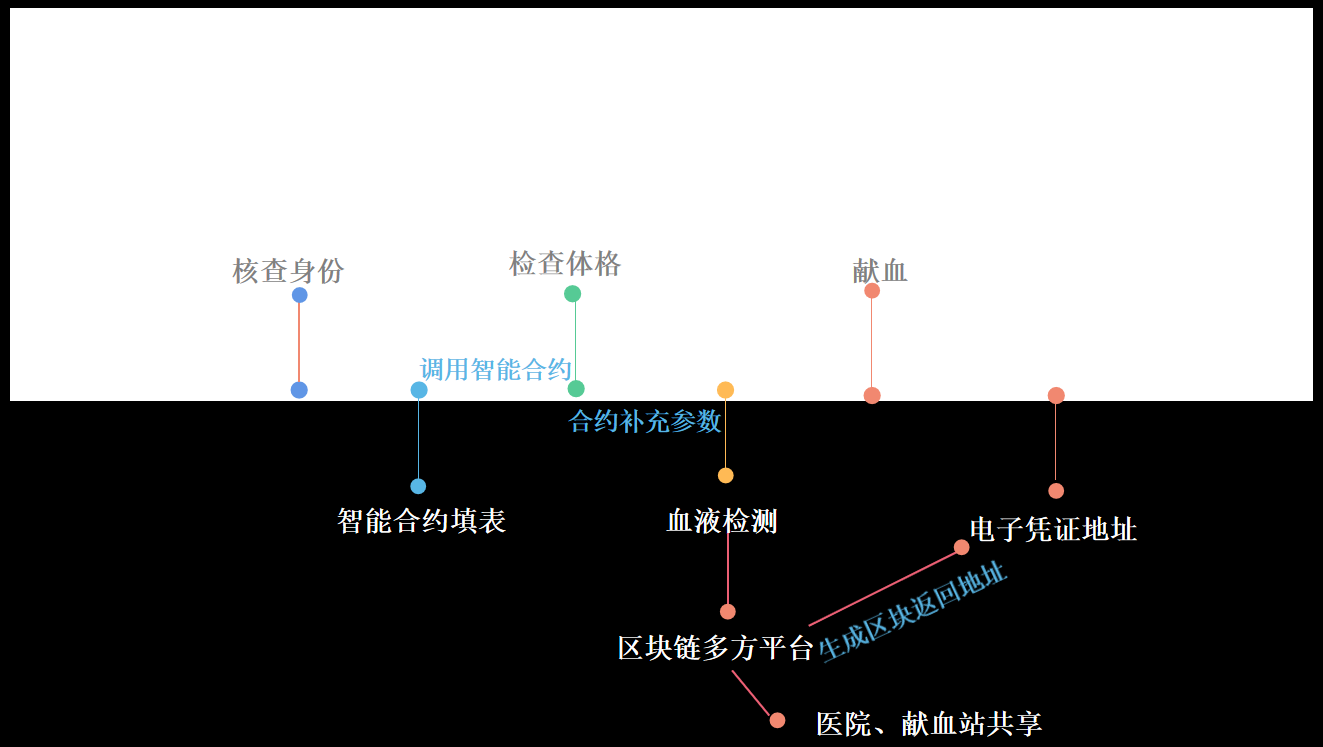
**（5）信息互信互通：**区块链进行献血者的身份认证，实现献血者的信息自由流动，积累大量应用数据，献血证在各个医院都可以使用。比如说原来我们的献血证，一旦跨地区便失效不能使用了，但区块链从可追溯的特性上可改变这一情况，发证机构通过对电子献血证文件的签名，将发证记录保存在区块链上，可证明其献血证的合法存在，献血证只要在区块链上做了公证，便成了全球可信任的证件。

**（6）安全性高：**区块链技术具有**安全性**，可以确保血液的安全性。这无形中解决了无偿献血的一个重要的问题：**献血者的血液安全性。献血站和医院不敢保证这一点。因为献血者换了献血地点，他们的献血间隔无法查询。**而有了区块链后，献血者的信息不光是一家献血站有，而是多家献血站拥有，按间隔期坚持献血的献血者，无论到哪一个献血机构，都会被数据显示为安全间隔期，可以献血。而患有不同疾病而不能献血的献血者，也会被提前在数据中预测，他的血液将不予采用，这不仅保证了血液的安全性，也从根本上提高了工作效率。

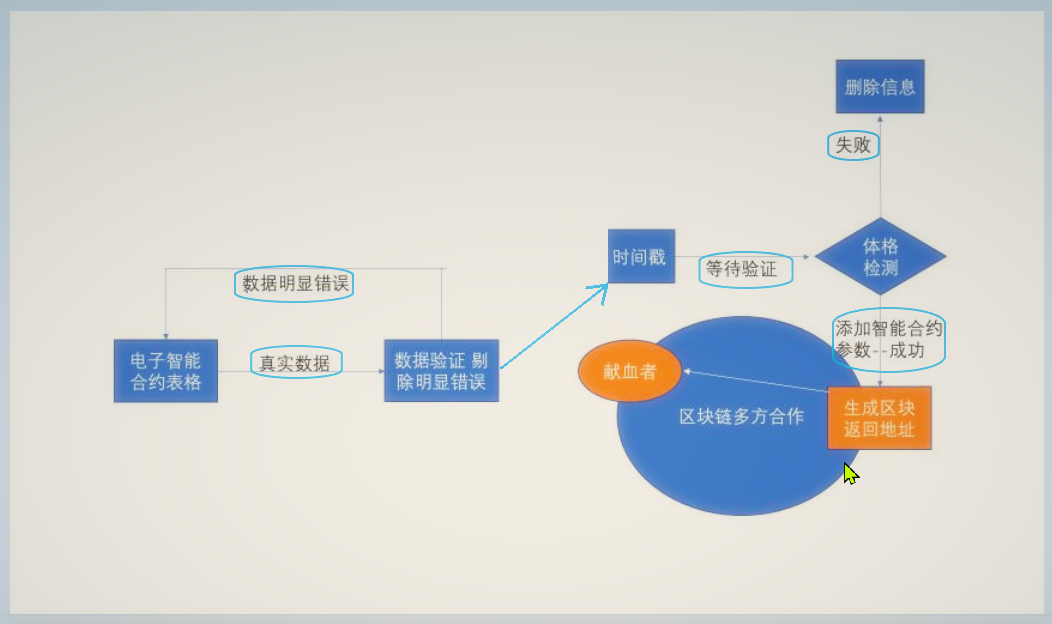
**（7）国家政策支持：**2019年对于区块链技术发展来说无疑是一个重要拐点。伴随着2019年10月24日国家高层对于区块链技术发展现状和趋势进行的第十八次集体学习，区块链技术的集成应用在新的技术革新和产业变革中起的重要作用被强调。指出要把区块链作为核心技术自主创新的重要突破口，明确主攻方向，加大投入力度。目前，全球主要国家都在加快布局区块链技术发展。而我国在区块链领域拥有良好基础，要加快推动区块链技术和产业创新发展，积极推进区块链和经济社会融合发展。自此，区块链作为一项前沿技术开始被上升到国家高度，在这样的背景下，近年来国家和各地方针对区块链出台了许多相关政策。所有，区块链+献血系统会有国家政策的支持，其前景广阔。

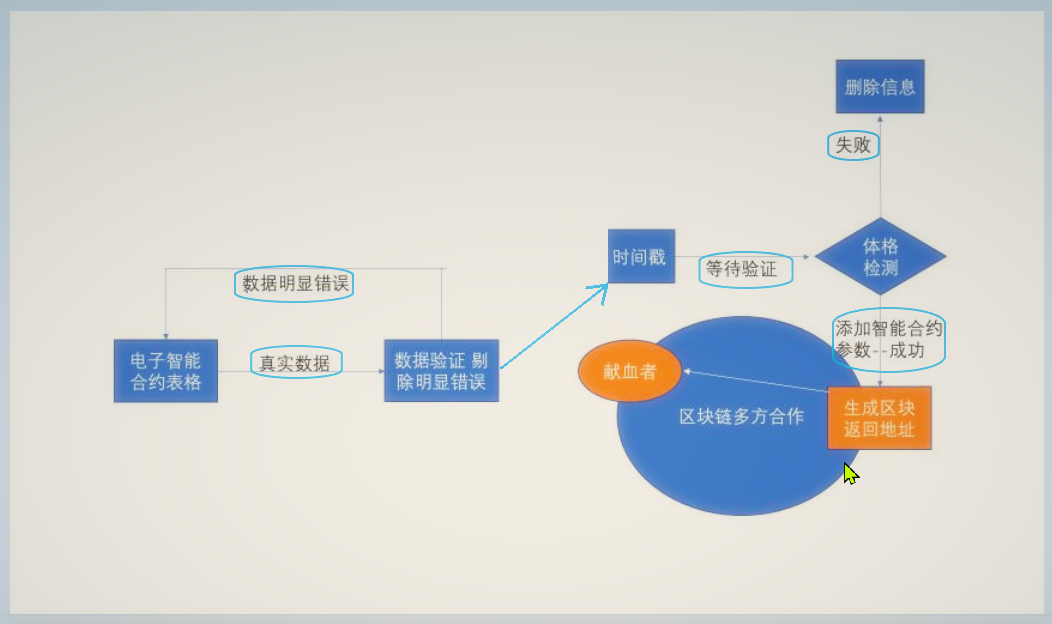
1. **链改后的献血平台**

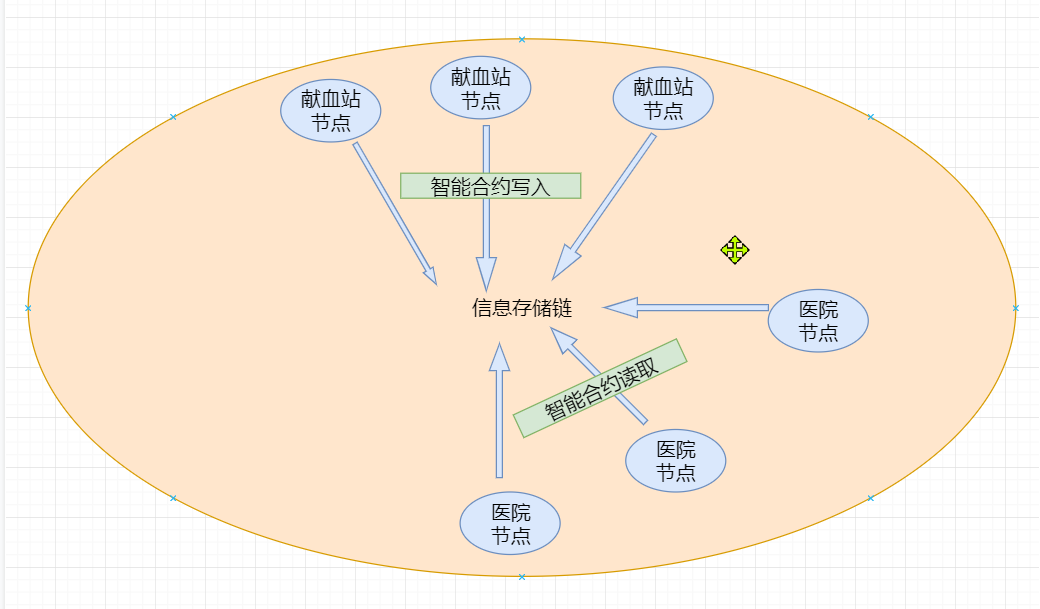
**链改后献血流程图：**



**链改后献血结构图：**







1. **链改后献血业务**

**1、**

* **身份核查**

先对献血者进行身份核查，核查基本信息（姓名、证件等等），合格者才能进行智能合约填表。

* **体格检查**

对献血者进行体格检查，检查身高、体重、血压等，献血者完成体检后，在智能合约中进行参数补充。智能合约会对输入的数据进行剔除、验证等。合格的数据会加入参数中，不合格的参数则会被剔除。

* **血液检测**

体检完成的且合格的献血者随后进行血液检测，在智能合约中进行参数补充。智能合约会对输入的数据进行剔除、验证等。合格的数据会加入参数中，不合格的参数则会被剔除。

* **献血**

献血者的体格检查、血液检测都合格后，就可以进行献血。

* **领证**

献血完成后，献血者可以领取自己的电子凭证地址和献血证书。献血者可以通过采用交易哈希查询的方式进行监督查询自己血液的用途。

* **信息上链**

利用区块链进行对智能合约的填表操作，将献血者的身份信息、体格检查信息、血液检测等信息通过智能合约的方式进行上链。

**2、**

1. 采用智能合约的方式制定对应的规则，判断献血者的数据是否有效，无效的数据进行剔除，有效的数据则进一步加盖时间戳，确保时间信息的准确性。
2. 智能合约会检测献血者的身份信息是否和对应基于证件的信息相符合，并加入判断条件来减少误差，不符合的数据进行剔除，符合的数据则进行下一步。智能合约会利用区块链的特性，增强签名属性，做到一签一献。智能合约也可以利用区块链的可溯源性，可以方便的找到当初献血者的身份信息、血液检测等详细信息。智能合约的实现方式是可以采用 require 属性值对身份、体格、血液信息做限制，然后通过事件信息来对献血成功的进行对应的记录。

（3）并且最后的证件采用区块链凭证，这不但可以降低成本，还可以增强实用性。献血者可以登陆区块链平台查找属于自己的私钥的中间的献血记录和献血证书，还可以通过采用交易哈希查询的方式进行监督查询自己血液的用途。而传统的信息信息系统是将医院、献血者、献血站三者结合，采用传统信息化的手段，而其中的每一步都可以遭受任何人的更改以及删除，具有非同步性，而链改后的区块链献血平台将医院、献血者、献血站三者结合，保持数据同步，并且进一步简化整个登记流程，减少纸质证件成本，增加溯源能力。

（4）新的信息存储系统利用区块链平台整合献血者、医院、献血站成为区块链的节点，让公证处等权威机构成为记账节点，记账节点具有利用智能合约写入数据形成区块的功能，将献血者的血液信息和身份信息写入区块链形成哈希，而医院等其他节点同步获取数据，**与此同时，每家献血站和医院还会生成一个副本，备份着所有献血站的献血者的献血数据。这样的结果就是，每家献血站和医院，可以看到所有献血站上传献血者的信息。献血者也可以查看自己的血液去向，被哪家医院使用，用途是什么，这也便解决了献血者对血液不透明的各种揣测。**并且基于不可更改性，保障信息不丢失，医院可以采用智能合约调取区块链中间信息。而对应的献血者可以通过献血行为获取联盟链积分，提高公信力，后期将有利于维护提高社会公信力建设。

但是任何参与区块链的节点都需要先完成身份注册，通过验证后，其数字身份才能被允许。具体来说，以类似会员制的方式实现一种区块链节点的准入机制，通过会员制的注册服务对区块链参与者的身份和权限进行管理。由政府部门统一部署和维护，并让这些节点有执行整个区块链平台的共识记账功能，参与节点通过区块链客户端完成会员注册后，系统会自动完成注册节点公私钥的分配，并生产合法的数字身份即交易地址。

1. **链改后的优势**

* **数据共享**

医院、献血站等多方进行协作，它们之间通过数据同步进行一系列的数据交互，相互监督，多方合作管理整个血液数据，保持供给和需求之间的相互平衡，缓解目前需求高于供给的现状。

* **减少错误不可篡改**

利用区块链和智能合约对数据进行存储，能进一步的减少献血者的身份、体格、血液等信息的错误和误差等。且由于区块链的不可篡改，上链的信息不能篡改，这可以为溯源进一步提供了基础，比如说献血者溯源、血液信息溯源等。

* **可溯源**

血液在献血者、献血站、医院、患者之间进行转换，形成交易。而通过追踪交易，可以溯源整个血液流转过程，从而形成溯源。因为血液每一次的献血类似于ERC20类的非同质代币，因此血液的传送过程为血液代币化的转账过程，如果某人用了别人的血得了其他疾病，那么我们就可以利用血液流转过程形成的交易链进行溯源。

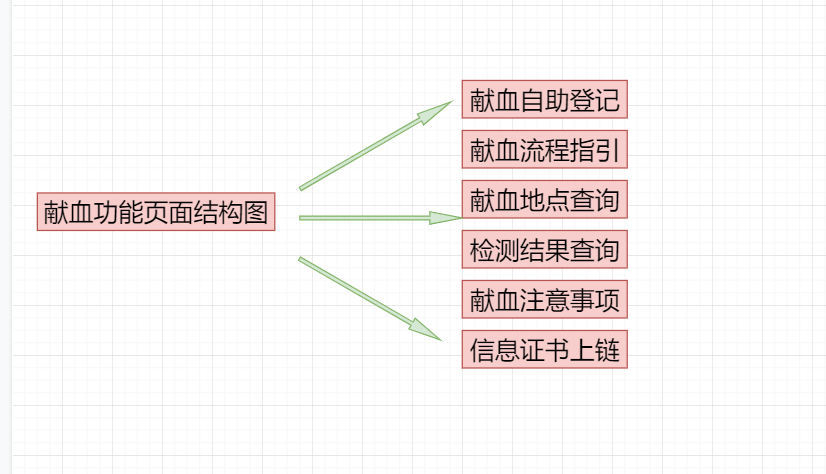
* **监管透明**

由于整个过程中间具有溯源和不可篡改性质，所以对应的区块链平台进一步提高了监管能力，透明化了整个献血市场和相关活动，使献血者可以在阳光下监督自己的血液是否进入对应的需求方。

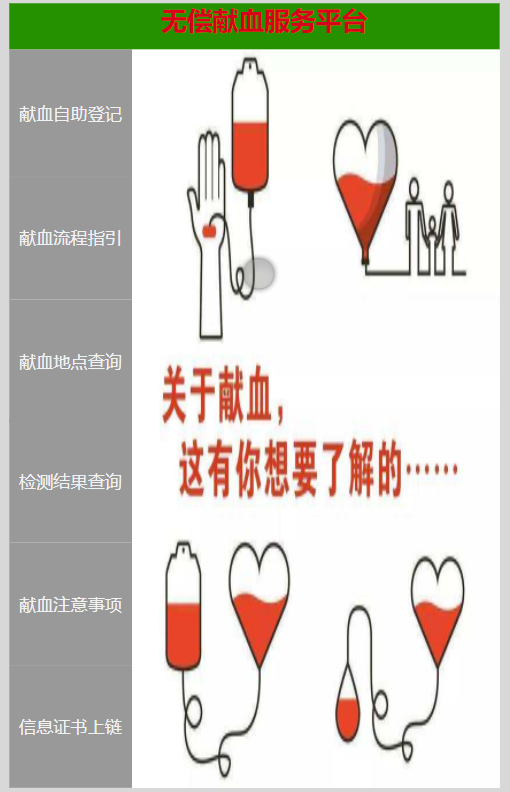
1. **功能页面展示**

**1、献血功能页面结构图**

新献血服务平台主要分为六个页面。分别是献血自助登记、献血流程指引、献血地点查询、检测结果查询、献血注意事项、信息证书上链。



1. **首页**



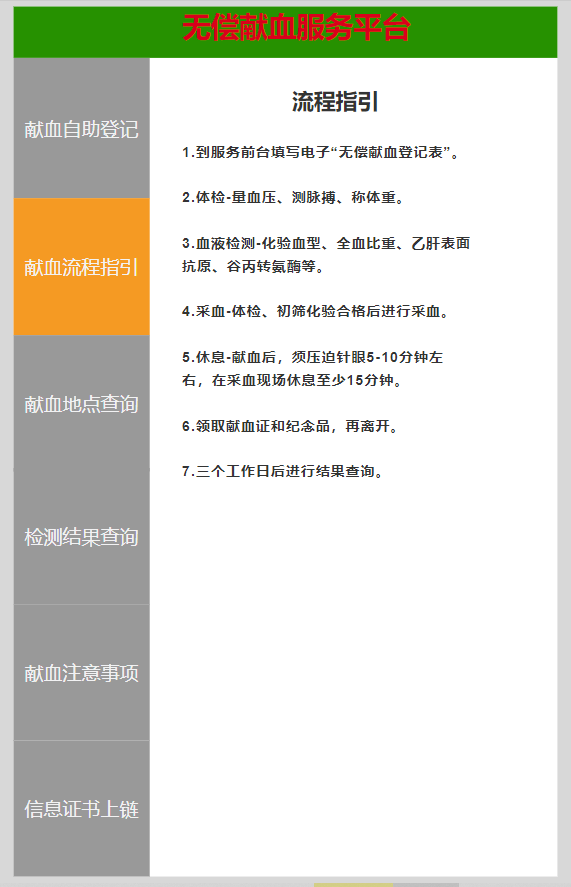
1. **献血自助登记**

该服务平台提供“献血自助登记”功能，当献血者进入献血自动登记页面时，可以进行登记操作。填写的内容包括有：姓名、性别、联系电话、证件类型、证件号码、通讯地址、职业、献血量等其他信息。



1. **献血流程指引**

该服务平台提供“献血流程指引”功能，当献血者完成自助登记后，若不熟悉献血流程可以进行查询。该服务平台将整理好的献血流程在系统中直接显示。主要分为六个步骤：第一，到服务前台填写电子“无偿献血登记表”；第二，体检（量血压、测脉搏、称体重）；第三，血液检测（化验血型、全血比重、乙肝表面抗原、谷丙转氨酶等）；第四，采血（体检、初筛化验合格后进行采血）；第五，休息（献血后，须压迫针眼5-10分钟左右，在采血现场休息至少15分钟）；第六，领取献血证和纪念品，再离开；第七，献血完成后，在三个工作日之后可以在“检测结果查询”功能中进行结果查询。



1. **献血地点查询**

该服务平台提供“献血地点查询”功能，献血者可以查看献血站，献血者在点击献血地址进入后，可以查看到具体的献血点详情信息。主要包括有：名称、地址、工作时间、热线电话等。



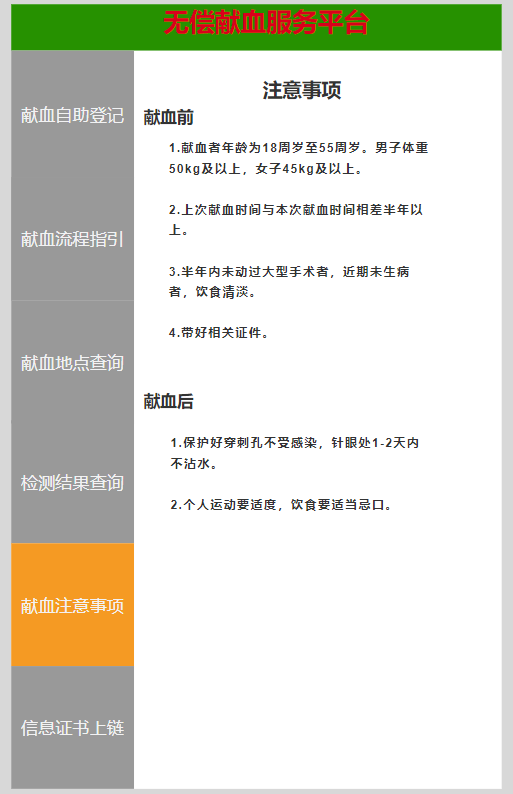
1. **检测结果查询**

该服务平台提供“献血结果查询”功能，献血者在进入献血检测结果查询页面后，需要填写“姓名”、“证件类型”、“证件号码”以及“献血号”信息，填写完成后，该服务平台可以查询出献血者的献血检测结果。主要的信息包括献血序号、献血时间、献血方式、献血量、检测结果等信息。



1. **献血注意事项**

该服务平台提供“献血注意事项”功能，献血者可以查看献血注意事项的内容。献血前主要有四点：第一，献血者年龄为18周岁至55周岁，男子体重50kg及以上，女子45kg及以上；第二，献血者上次献血时间与本次献血时间要相差半年以上；第三，献血者要半年内未动过大型手术者，近期未生病者，饮食清淡；第四，献血者要带好相关证件。献血后主要有两点：第一，保护好穿刺孔不受感染，针眼处1-2天内不沾水；第二，个人运动要适度，饮食要适当忌口。



1. **信息证书上链**

该服务平台提供“信息证书上链”功能，主要对献血者的身份信息、血液信息、证书的电子凭证地址进行上链。身份信息的内容主要包括有：姓名、性别、联系电话、证件类型、证件号码、通讯地址、职业、献血量等其他信息。血液信息的内容主要包括有：献血序号、献血时间、献血方式、献血量、检测结果等信息。



1. **关键技术及方法**

在这个基于区块链的公益献血系统中，首先要解决的就是献血数据的接入方式问题，即献血数据如何“上链”。在区块链网络中，区块数据将被多个节点备份，因此不能将献血直接存储在链上，以免造成大量冗余，影响区块链的运行效率。区块链上只保存献血的部分数据，保证献血的信息在存储和流转的过程中只有拥有合法访问权限的用户才可以获取，而献血数据本身则存储在云端上。

1. 献血站将献血数据上传至云服务器，同时将基于献血信息内容的Hash地址保存在区块链上。区块数据仅记录数据的数字指纹，而非数据内容本身。
2. 公证机构给献血者发放献血证书及电子凭证地址哈希，通过加密通道访问区块数据和数字指纹查验数据的真伪，为数据提供公证服务。
3. 献血者可以登陆区块链平台查找属于自己的私钥的中间的献血记录和献血证书，还可以通过采用交易哈希查询的方式进行监督查询自己血液的用途。

且因为区块链具有无法篡改和时间戳功能，所以这些重要的献血数据可以被全世界的计算机同步并永久保存。只要区块链节点存在，这些数据就不会丢失并且可以被持续地传播和保存。

代码实现如下：

//SPDX-License-Identifier: UNLICENSED

pragma solidity ^0.7.2;

contract Donations {

//定义献血结构体

struct blood {

string name; //献血者姓名

address hash; //返回的address地址哈希值

uint248 quantity; //献血量

uint256 when ; //献血次数

}

//记录所有在结构体中所输入的的献血数据信息

blood[] private allDonations;

//将献血数据储存在区块链上，包括献血者姓名 、返回

//的address地址哈希值 、献血量、献血次数

function save(string memory name, uint248 quantity, uint256 time) public {

allDonations.push(blood({

name: name,

hash: msg.sender,

quantity:quantity,

when: time

}));

}

//查询当前献血总次数。访问当前数据的长度

function getSize() public view returns (uint256){

return allDonations.length;

}

//根据编号查询献血的具体内容（结构体所包含的信息）

function getRandom(uint random) public view returns (string memory, address, uint248, uint256) {

if(allDonations.length==0){

return ("", msg.sender, 0, 0);

}else{

blood storage result = allDonations[random];

return (result.name, result.hash,result.quantity, result.when);

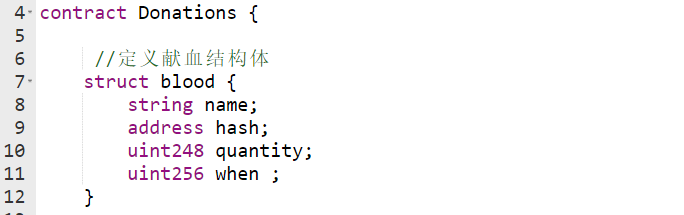
}

}

}

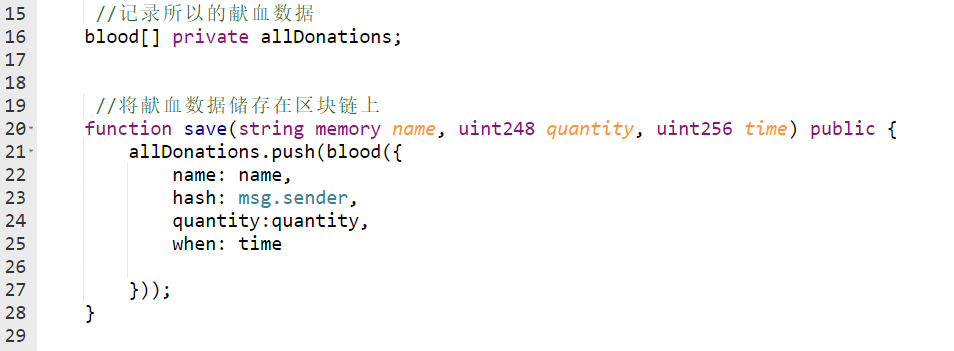
详细解释：

1. 定义献血结构体



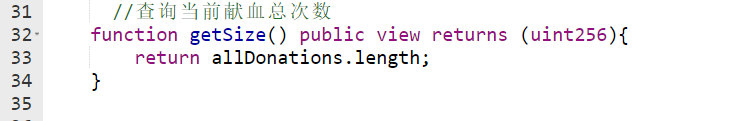
第4至第12行代码定义了一个献血的结构体（献血者姓名、address地址哈希值 、献血量、献血次数），该结构体主要用于简单地记录献血者在什么地点献了多少血，总共献了多少次血。

1. 将献血数据记录、存储在区块链上



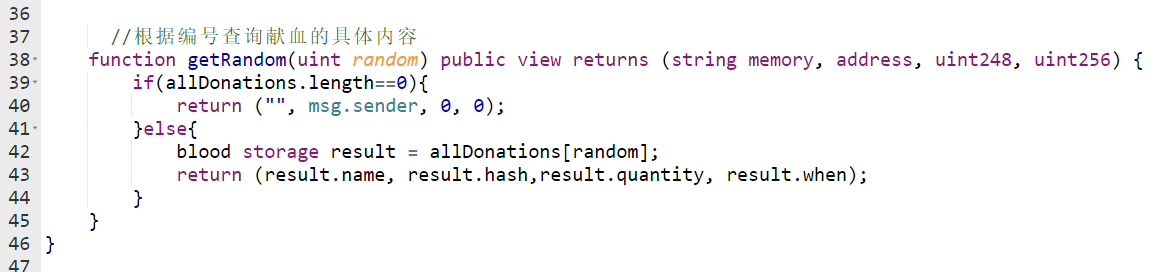
第16行代码是个数组，在输入献血信息之后，合约会通过第20至28行代码将输入的献血信息（献血者姓名、address地址哈希值 、献血量、献血次数）组装成一个结构体，并存入该数组当中。

3、获取献血次数



第32至34行代码是为了访问当前数组的长度，即是访问献血者的献血的总次数。

4、获取献血数据信息

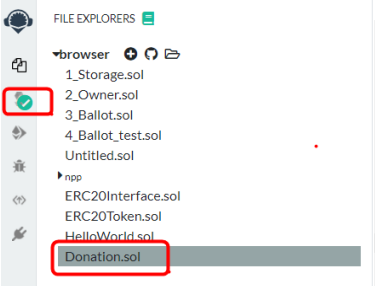


第38至47行代码是根据输入的数组下标，返回指定的结构体，然后将结构体内存储的信息返回给函数调用者，即通过输入定义的献血结构体信息，返回献血结构体内容存储的献血数据信息。

编译、部署、测试合约：

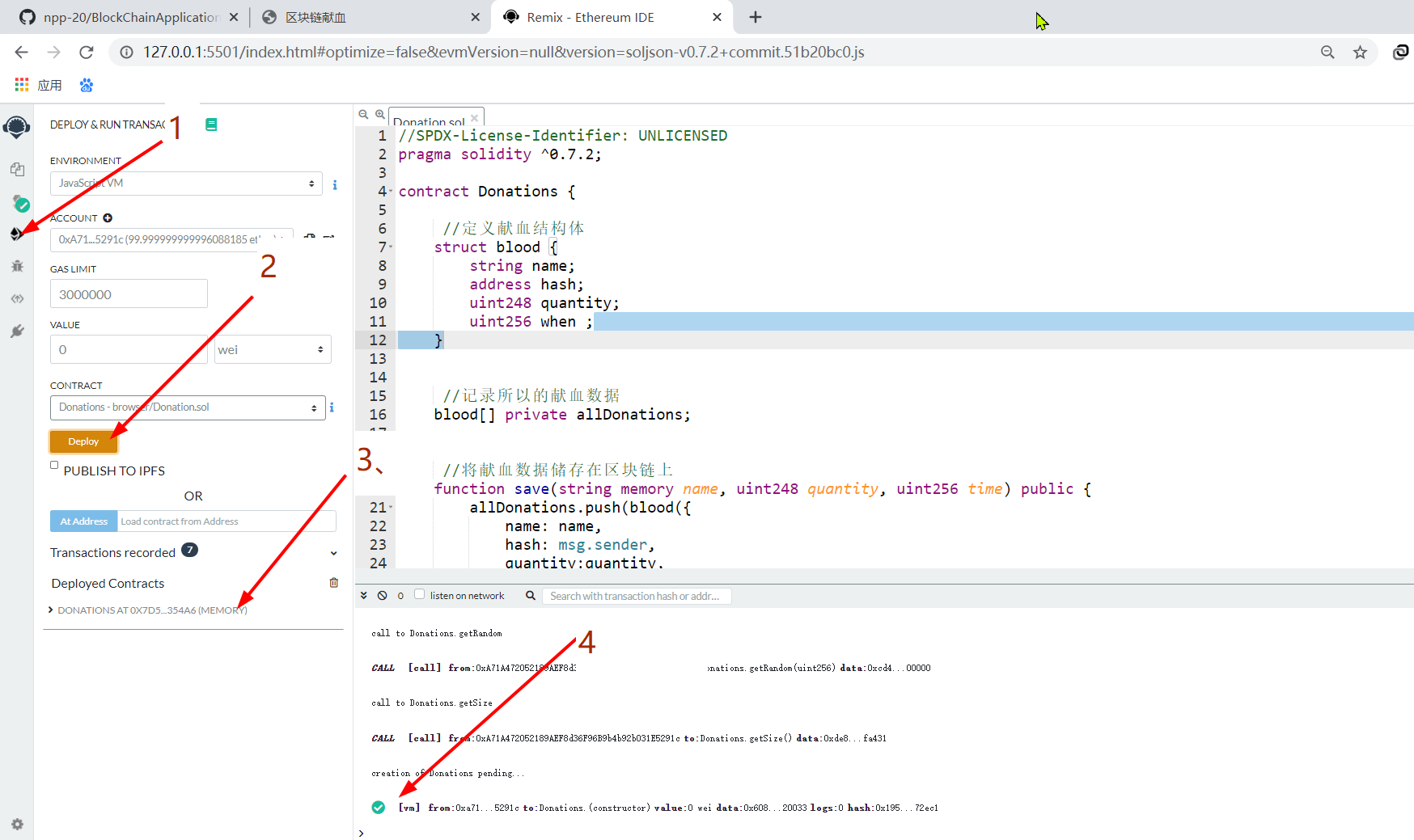
1. 编译合约

出现绿色图标，合约编译成功



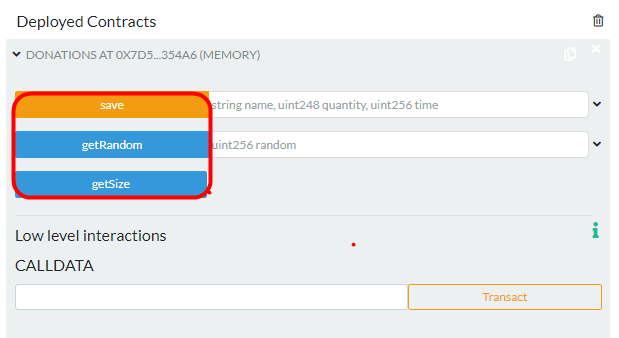
1. 部署合约

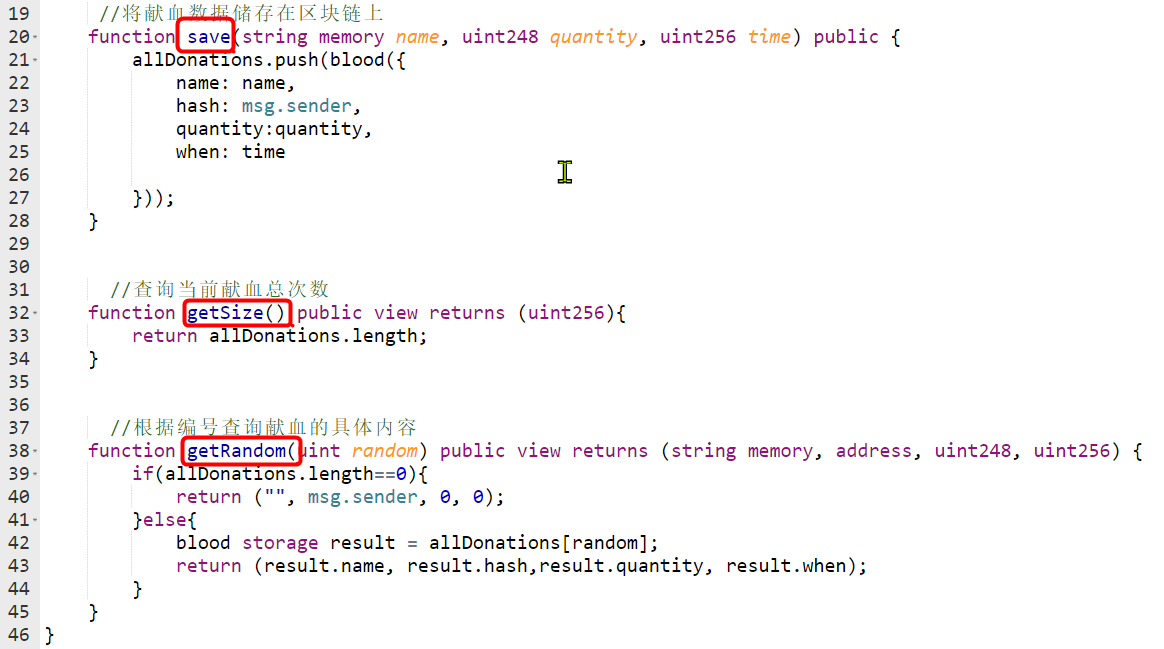
第一步点击黑色小图标，第二步点击“Deploy”按钮。出现了第3和4，则部署成功。



1. 测试合约

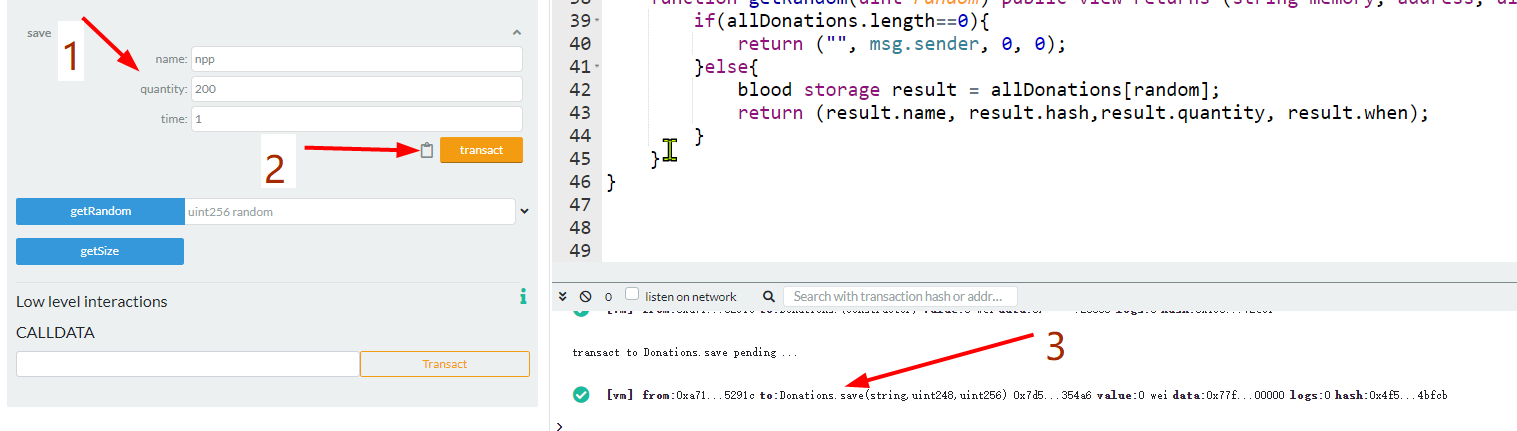
（1）合约包括了save、getRandom、getSize这三个函数方法，而部署出现的三个内容与合约所写的三个方法一致，表示三个函数方法成功。





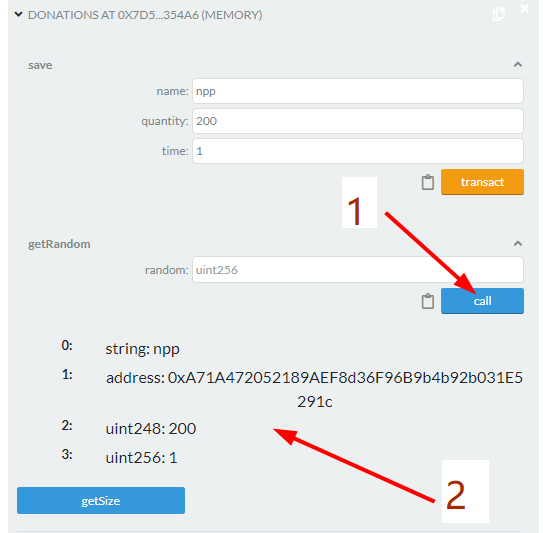
（2）在save方法输入献血者姓名、献血量及献血次数，然后transact，出现3则成功。

示例：name填入npp；quantity填入200；time填入1.



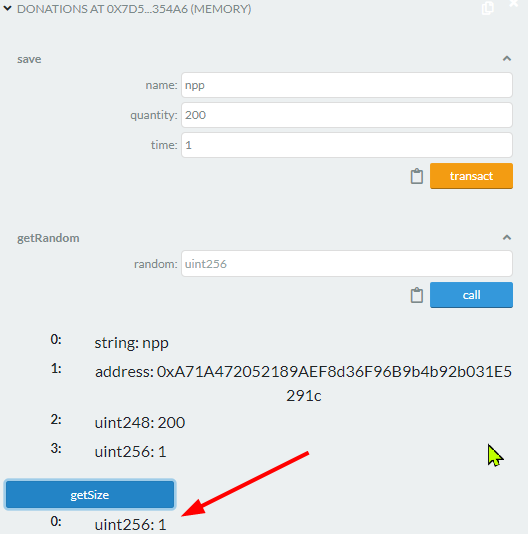
（3）利用getRandom方法返回在save方法输入的数据信息。点击call，出现了与在save方法输入的数据信息内容一致，并返回一个address地址，即哈希值，则成功。

示例：上一步所填入的npp，200，1，call显示内容为string：npp；adress：0xA71A472052189AEF8d36F96B9b4b92b031E5291c；uint248：200；uint256：1.



（4）利用getsize方法返回献血的总次数。

示例：上两个步骤输入一次，所以getsize返回的献血的总次数为1。

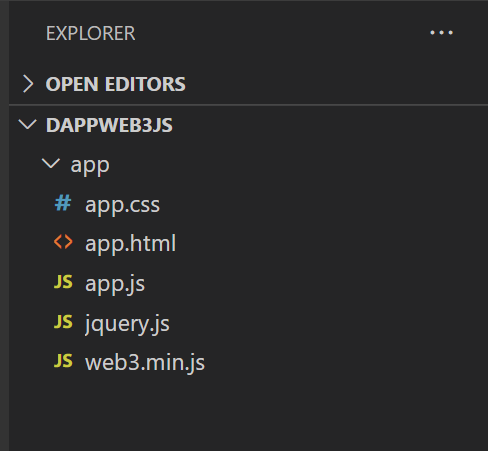


前端展示：页面与代码

（1）前端页面：



（2）文件目录：主要是css、html和js文件。



（3）app.css代码：

body {

    background-color: #725BA4;

    color: #FCE8DF;

    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;

    text-align: center;

}

.text,

.hidden {

    display: none;

}

.error {

    color: red;

}

.input-group,

.display-message {

    display: flex;

    flex-direction: column;

    align-items: center;

}

.input-group,

button {

    margin-top: 5px;

}

#loader {

    border: 4px solid #f3f3f3;

    /\* Light grey \*/

    border-top: 4px solid #3498db;

    /\* Blue \*/

    border-radius: 50%;

    width: 25px;

    height: 25px;

    animation: spin 2s linear infinite;

    display: none;

}

@keyframes spin {

    0% {

        transform: rotate(0deg);

    }

    100% {

        transform: rotate(360deg);

    }

}

（4）app.html代码：

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="utf-8">

    <script type="text/javascript" src="jquery.js"></script>

    <script type="text/javascript" src="web3.min.js"></script>

<link rel="stylesheet" href="app.css" />

</head>

<body>

    <h1>区块链献血平台</h1>

<h2>账户: <span class="showAccount"></span></h2>

<hr>

<button class="sendEthButton">发送交易</button>

<hr>

<button class="givemeEth">请求交易</button>

<hr>

<button class="getDanationsMsg">获取献血信息</button>

<h2>献血信息: <span class="showDanationsmessage"></span></h2>

<h2>交易哈希: <span class="transactionHash"></span></h2>

<h2>交易状态: <span class="status"></span></h2>

<hr>

<input id="danationstxt" value=""/>

<button id="addDanations">添加献血信息</button>

<button class="getMyDanationsAll">获取全部献血信息</button>

<h2>献血次数: <span class="Danationslength"></span></h2>

<h2>献血内容: <span class="DanationsContent"></span></h2>

<script type="text/javascript" src="app.js"></script>

</body>

</html>

（5）app.js代码：

let accounts = [];

//查看是否安装了MetaMask钱包，保证平台的交易环境。

if (typeof window.ethereum !== 'undefined') {

    console.log('MetaMask is installed!');

}

console.log("isMetaMask: " + ethereum.isMetaMask)

$(".献血平台").click(function () {

    getAccount()

}

)

//设置交易的转出账户、转入账户，设置gas和gasPrice，交易成功就打印交易哈希，失败就打印错误。

$(".sendEthButton").click(function () {

    ethereum

        .request({

            method: 'eth\_sendTransaction',

            params: [

                {

                    from: accounts[0],

                    to: '0xfe793b941f66CFb2b04943c9c1Ec1F4345d9a5f6',

                    value: '0x2386f26fc10000',

                    gasPrice: '12a05f200',

                    gas: '2dc6c0',

                },

            ],

        })

        .then((txHash) => console.log("交易哈希是", txHash))

        .catch((error) => console.error);

}

)

//得到此时使用的账户，当进行切换账户时，会进行显示，且打印处会显示账户已被更改。

//设置交易环境，当进行切换链时，打印处会显示链已经被更改。

async function getAccount() {

    accounts = await ethereum.request({ method: 'eth\_requestAccounts' });

    const account = accounts[0];

    $(".showAccount").html(account);

}

ethereum.on('accountsChanged', function (accounts) {

    console.log("accountsChanged");

    getAccount()

});

ethereum.on('chainChanged', (chainId) => {

    console.log("chainId", chainId);

});

let web3 = new Web3(Web3.givenProvider || "ws://localhost:8545");

console.log("web3", web3)

//复制合约的ABI文件。

var contractAbi = [

    [

        {

            "inputs": [

                {

                    "internalType": "string",

                    "name": "\_mystring",

                    "type": "string"

                }

            ],

            "name": "addDanations",

            "outputs": [],

            "stateMutability": "nonpayable",

            "type": "function"

        },

        {

            "inputs": [],

            "name": "giveMeEth",

            "outputs": [],

            "stateMutability": "payable",

            "type": "function"

        },

        {

            "inputs": [

                {

                    "internalType": "string",

                    "name": "initDanationsMessage",

                    "type": "string"

                }

            ],

            "stateMutability": "nonpayable",

            "type": "constructor"

        },

        {

            "inputs": [

                {

                    "internalType": "string",

                    "name": "newDanationsMessage",

                    "type": "string"

                }

            ],

            "name": "update",

            "outputs": [],

            "stateMutability": "nonpayable",

            "type": "function"

        },

        {

            "inputs": [],

            "name": "Danationsmessage",

            "outputs": [

                {

                    "internalType": "string",

                    "name": "",

                    "type": "string"

                }

            ],

            "stateMutability": "view",

            "type": "function"

        },

        {

            "inputs": [],

            "name": "getBalance",

            "outputs": [

                {

                    "internalType": "uint256",

                    "name": "",

                    "type": "uint256"

                }

            ],

            "stateMutability": "view",

            "type": "function"

        },

        {

            "inputs": [],

            "name": "getDanationsMsg",

            "outputs": [

                {

                    "internalType": "string",

                    "name": "",

                    "type": "string"

                }

            ],

            "stateMutability": "view",

            "type": "function"

        },

        {

            "inputs": [

                {

                    "internalType": "uint256",

                    "name": "i",

                    "type": "uint256"

                }

            ],

            "name": "getMyDanations",

            "outputs": [

                {

                    "internalType": "string",

                    "name": "",

                    "type": "string"

                }

            ],

            "stateMutability": "view",

            "type": "function"

        },

        {

            "inputs": [],

            "name": "getMyDanationsAll",

            "outputs": [

                {

                    "internalType": "string[]",

                    "name": "",

                    "type": "string[]"

                }

            ],

            "stateMutability": "view",

            "type": "function"

        },

        {

            "inputs": [

                {

                    "internalType": "uint256",

                    "name": "",

                    "type": "uint256"

                }

            ],

            "name": "myDanations",

            "outputs": [

                {

                    "internalType": "string",

                    "name": "",

                    "type": "string"

                }

            ],

            "stateMutability": "view",

            "type": "function"

        }

    ]

];

//复制合约的部署地址。

var contract = new web3.eth.Contract(contractAbi, "0x3866aaFD1D529b78Bfff47852e4eE7758499c414");

console.log("contract", contract)

//设置获取献血信息按钮；当点击时，可返回献血信息。

$(".getDanationsMsg").click(function () {

    contract.methods.getDanationsMsg().call({ from: accounts[0] }).then(

        function (result) {

            console.log(result)

            $(".showDanationsmessage").html(result)

        }

    );

}

)

//设置获取全部献血信息按钮；当点击时，可返回献血信息和献血的总次数。

$(".getMyDanationsAll").click(function () {

    alert("getMyDanationsAll")

    contract.methods.getMyDanationsAll().call({ from: accounts[0] }).then(

        function (result) {

            console.log("getMyDanationsAll", result)

            $(".Danationslength").html(result.length)

            $(".DanationsContent").html(result)

        }

    );

})、

//设置增加献血信息，设置添加条件为输入的长度不能小于3.

$("#addDanations").click(function () {

    alert("addDanations" + $("#danationstxt").val());

    let \_txt = $("#danationstxt").val()

    if (\_txt.length < 3) {

        alert("长度不能小于3")

    } else {

        contract.methods.addDanations(\_txt).send({ from: accounts[0] }).then(

            function (result) {

                console.log("result", result)

            }

        )

    }

});

//请求交易；从哪个账户发出，打印交易哈希和交易的状态。

$(".givemeEth").click(function () {

    alert("givemeEth")

    contract.methods.giveMeEth().send({ from: accounts[0], value: 12300\_000\_000\_000\_000 }).then(

        function (result, error) {

            console.log("giveMeEth==>", result);

            $(".transactionHash").html(result.transactionHash)

            $(".status").html(result.status)

            alert("givemeEth OK");

        }

    )

}

)

1. **结束语**

本文档研究了区块链技术在公益献血方面的应用，包括分析公益献血的发展历史和发展现状，分析传统献血平台的流程、业务、结构以及存在的问题，分析链改后献血平台的流程、业务、结构以及链改后的优势，设计链改后的献血功能页面图，进行了基于区块链的公益献血的应用分析，展示设计后的前端页面以及关键的技术和方法，也解释了相关代码含义。对于针对传统献血数据采集不准确、效率低、血浆易浪费、血液难以溯源等相关问题，设计了一种新型的、基于区块链技术的公益献血活动。利用运行在区块链上的智能合约，可以减少数据的错误且不可篡改，不仅提高了工作效率，同时还推动提高血液流转的透明度，并实现血液的可溯源。

1. **总结与体会**

老师布置本此课程设计的目的是通过实践使同学经历区块链应用设计开发的全过程，受到一次综合训练，以便比较全面的理解、掌握和综合运用所学的知识，结合具体的设计案例、理解并初步掌握系统分析、系统设计、系统开发的主要环节和步骤，以及报告编写的能力。所以这是一个为了使我们能够达到学以致用，更好的领会课程而安排的一个重要的课程设计，是培养我们综合素质和实践能力的一个教学过程。

为期一个月的课程设计结束了，在这次的课程设计中不仅检验了我所学的知识，也培养了怎样去写软件文档。课程设计是我们专业课程知识综合应用的实践训练，这是我们迈向社会，从事职业工作前一个必不可少的过程。通过这次我在多方面有所提高，巩固与扩充了《软件文档写作》等课程所学的内容，掌握了设计的方法和步骤，同时对区块链应用设计开发有了了解，独立思考的能力也有了提高。在这次设计过程中，体现出自己单独设计的能力以及综合运用知识的能力，体会了学以致用、突出自己劳动成果的喜悦心情，并从中发现自己平时学习的不足和薄弱环节，从而加以弥补。

作为一名学生，我觉得能做这样的课程设计是十分有意义的。在校园生活里，我们大多数接触的都是一些专业的基础理论课。我们在课堂上掌握的仅仅是专业的基础理论知识，如何去面对现实中的各种软件设计，如何把我们所学到的专业的基础理论知识运用到实践中去呢？而类似此次的课程设计作业就为我们提供了良好的实践平台。

在做本次课理设计的过程中，我感触最深的当属是去网上查阅了很多有关献血的知识。为了让自己的设计文档更加完善，更加符合老师的标准，查阅资料是十分重要的，同时也是必不可少的，我们做的是课程设计，而不是艺术家的设计。艺术家可以抛开实际，尽情在幻想的世界里翱翔。但我们不一样，我们是软件工程师，所写的软件文档设计是需要实现出来的，不切实际的构想永远只能是构想，永远无法应用到实际生活中去。只有把所学的理论知识与实践相结合，从理论到实践，才能够应用到社会上去，才能够提高自己的实际动手能力和独立思考能力。

作为一名学软件的学生来说，掌握一门或几门画图软件同样是必不可少的。虽然本次课程设计中老师没有要求用什么画图软件，但是我在整个课程设计写作过程中都用到了draw 和Axure ，用它们两个画图方便简洁，且易修改，速度快。我的画图设计，基本上都是在draw 和Axure 上设计出来的。特别是Axure 画图软件十分好用，它设计好各个页面的功能图，还可以进行预览，相当于一个小小的demo。

尽管这次作业的时间是漫长的，但我的收获还是很大的。对我来说，收获最大的是方法和能力，那些分析和解决问题的方法和能力。在整个过程中，我发现像我们这样的学生最缺少的便是经验，且没有感性的认识，空有理论知识，且有些东西会与实际脱节。

最后我想说，知识的获得是无止境，只要你有行动。