附件1

基于区块链的人力资源管理系统计划书

——初稿

|  |  |
| --- | --- |
| **专业** | 区块链（软件工程） |
| **班级** | B190601 |
| **姓名** | 王紫繁 |
| **学号** | 2019000101017 |

|  |
| --- |
| 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**

[一、 方案简述 4](#_Toc9002)

[二、 人力资源背景 4](#_Toc24035)

[三、 传统人力资源招聘现状 4](#_Toc32761)

[四、 传统人力资源市场中存在的问题 5](#_Toc18982)

[求职者方面： 5](#_Toc9407)

[（一） 信用体系不健全 5](#_Toc19011)

[（二） 信息泄露严重 6](#_Toc22461)

[（三） 价值变现困难 6](#_Toc7027)

[招聘者方面： 7](#_Toc6483)

[（四） 人力资源招聘信息采集不精确 7](#_Toc23645)

[（五） 背调程序有困难 8](#_Toc22066)

[（六） 成本管理不科学 8](#_Toc29960)

[五、 区块链与人力资源管理的匹配度 9](#_Toc13301)

[（一） 利用区块链的透明性进行招聘 9](#_Toc9103)

[（二） 利用区块链数据不可篡改性进行背调 9](#_Toc5008)

[（三） 利用区块链去中心化进行成本管理 10](#_Toc31566)

[六、 结合区块链技术的人力资源分析 10](#_Toc63)

[（一） 求职者方面分析 10](#_Toc21152)

[（二） 招聘者方面分析 11](#_Toc14034)

[七、 基于区块链的人力资源管理方案架构 12](#_Toc15209)

[（一） 技术架构 12](#_Toc4243)

[（二） 功能架构 12](#_Toc17655)

[（三） 方案设计架构 13](#_Toc13844)

[（四） 信息上链 14](#_Toc31245)

[八、 基于区块链的人力资源管理方案业务分析及具体实施 15](#_Toc24996)

[业务分析： 15](#_Toc25611)

[具体实施： 16](#_Toc30272)

[（一） 开发环境的配置 16](#_Toc25747)

[（二） 运行环境的配置 16](#_Toc5944)

[（三） 方案部分代码 17](#_Toc283)

[（四） 方案功能描述 20](#_Toc3561)

[（五） 方案功能操作流程图 24](#_Toc22604)

[（六） 方案代码功能测试 25](#_Toc30253)

[（七） 系统界面及操作 27](#_Toc30495)

[九、 方案执行预期效果 31](#_Toc12151)

[十、 结语 33](#_Toc13333)

1. **方案简述**

本方案是基于区块链的人力资源管理，将先进的区块链技术与人力资源结合，对于传统人力资源信息技术而言是一种升级和补充，给我国人力资源服务教育市场发展带来新的市场机遇和挑战，基于“区块链”技术的人力资源市场生态中，企事业单位与求职者之间更加透明化且趋于真实性。利用区块链技术的去中心化、不可篡改、溯源清晰等特性，解决目前传统人力资源信息技术产生的大量信息不安全等问题，将人力资源管理模式带到了新高度，大大提高人力资源管理效率。

1. **人力资源背景**

改革开放以来，特别是20世纪90年代以来，我国人才事业得到了长足的发展，人才规模不断扩大，我国劳动力总量达，但是目前来看，无论是从人才的总量、高级人才的储备以及人才与经济结构发展有效配置等问题上仍然存在着不少的问题。人才是国家、企业成长发展的最核心和最基础的资源，我国经济正处于转型升级的关键时期，人才资源是企业转型升级最重要的战略资源，人才资源短缺是目前企业转型升级遇到的最大瓶颈之一，我国人才的需求还处于快速增长时期，人力资源服务机构发挥人才资源和人力资源专业服务的优势上还有巨大的发展空间。

目前，我国已经初步形成了多层次、多元化的人力资源市场服务体系，人力资源服务内容也由最初的职业介绍、培训和流动人口档案管理等延伸至完整的人力资源服务产业链，包括政策咨询、求职招聘、劳动人事代理、就业指导、职业培训、创业指导、社会保障、劳务派遣、人才测评、人才搜寻、管理咨询和服务外包等多种业务。

1. **传统人力资源招聘现状**

当前大量的人力资源需求导致招聘市场简历造假泛滥，社会信用体系建设尚未成熟等，人力资源行业面临的最大痛点是信任缺失，包括信息真实性、信息孤岛、候选人能力评价维度单一、主观等一系列问题。同时传统的招聘方式中第三方人资企业的准确性、时效性、保障性普遍差，冗杂繁多的简历资料需通过第三方人资企业进行审查， 人工筛选处理时间长，第三方收集简历也是造成大量个人信息泄露的源头。

由于现有的社会信用环境及体系不健全，导致求职者自身的信用观念、雇佣方的信用信息核实出现企业在招聘过程中需要通过大量的简历筛选、多轮面试及详细的背景调查来避免学历和职业经历造假等风险。 社招中对求职者的过往业绩难以核实其真实性，学历成为筛选的基准之一，从而使得求职者真正的职业价值难 体现。而且由于招聘求职市场信息量大，信息不对称程度较高，导致市场上较常出现企业招不到合适的人才， 者又往往求职无果的人岗难以匹配的现象，从而产生对猎头行业的需求，导致企业需要增加大笔招聘支出，还有入职后快速离职的现象。

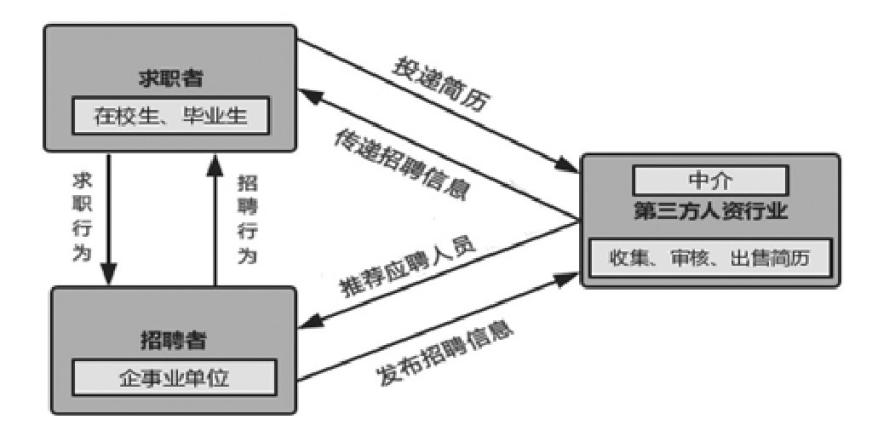


图1传统人力资源招聘市场模式

1. **传统人力资源市场中存在的问题**

求职者方面：

1. **信用体系不健全**

一方面，由于互联网求职平台门槛低，只需要手机号码验证便可注册登录，求职者的信息真实性无法得到保障，同时，企事业单位在招聘网站上面注册虽然有一定的门槛,会经过网站系统认证，但事实表明，招聘网站上经系统认证后的企业也会出现信息不对称，虚假地址，夸大薪资等现象。另一方面，求职者会出现简历内容夸大、学历学术造假等问题，使得企事业单位对求职者没有一个正确的认知与了解，并且企事业单位在招聘网站上发布招聘信息有限，通常以最简洁的排版列出所需的招聘信息，这种简单的信息经过美化加工，甚至会“误导”求职者，使求职者无法全面了解企事业单位的岗位内容。

1. **信息泄露严重**

传统招聘环节中第三方人资企业起着非常重要的作用，而企事业单位与大学生就显得非常被动，大学生通过 互联网上企事业单位投放的招聘信息，将自己的个人简历传递至第三方网站，第三方通过收集简历、筛选简历、出售简历获取盈利，这也是将个人信息泄露的源头。许多人资企业对求职者的信息保护不到位，造成信息泄露，被用于交易，轻则会使得求职者遭受垃圾信息和垃圾电话的骚扰，对其生活产生不必要的麻烦；重则发生互联网诈骗，使得求职者财产损失或者威胁到其生命安全。

1. **价值变现困难**

相比大学生的简历学历以及教育经历造假的情况来看，求职者过往的工作经验更难以被证实检验，就目前社会上还没有更加完善的信用生态体系，企事业单位也不会花费大量的人力财力去调查一个人过往的经历，使得企事 业单位难以对求职者的职业价值进行评定，一定程度上会影响求职的成功率。《中国互联网络发展状况统计报告》显示，截至2017年12月，我国网民规模达7.72亿，普及率达到55.8%，超过全球平均水平（51.7%）4.1个百分点，超过亚洲平均水平（46.7%）9.1个百分点，占全球网民总数的五分之一。2018届全国普通高校毕业生预计820万人，大量的数据表明，网上求职者普遍倾向于年轻化群体。如下图所示：

以2018届校园招聘为例，offer拒收，就业协议违约率高达9%,大学生违约率较高，使得招聘流程转化率低，不能实现求职者和雇主双方的有效匹配。

招聘者方面：

1. **人力资源招聘信息采集不精确**

2020年全国普通高校毕业生874万人，仅仅从高校毕业生这个数据，可见整个人力资源行业所需采集的信息量之大。由于人力资源行业涵盖农民工、技术人员、毕业生等多个主体，各个主体的流动性、分散性、信息完整性等并不一致，很容易出现部分人力资源的信息没有采集或采集后没有成功录入的问题，导致人力资源信息的不完善。

此外，许多主体对于信息采集的重要性并没有形成正确认识，导致部分求职者对待信息采集工作不认真，提供模糊的甚至是虚假的信息，使得采集到的部分信息失真，难以发挥信息采集的真正作用。同时，错误信息也给高校和企业人才对接带来麻烦，造成双方资源的浪费。因此，人力资源信息采集不精准在一定程度上阻碍了人力资源的精准就业，从而也阻碍了人力资源的后续管理。

1. **背调程序有困难**

背调属于人力资源术语，意为背景调查。目前，在一部分招聘工作过程中，应聘者虚增薪酬、夸大工作业绩、虚报任职背景、虚构教育培训经历等让企业难辨真伪，常常造成企业筛选工作成本高、选择效果差等问题。而优秀者曾经的工作经历由于没有可查的记录，在应聘新职位时也很难被知晓、信任，造成是“金子”也 难以在短短的简历或急促的面试中“发光发热”。 据不完全估计，有多达75%的求职者在求职申请或简历中包含表述不准确、篡改历史，甚至出现虚假信息的问题，导致用人单位越来越难以找到既 符合岗位条件又值得信任的人才。同时，由于人力和资本等机会成本的限制，用人单位无法对求 职者展开全面深入的调查，只能通过应聘者简历和简约的面试情况来快速判断一个人是否符合岗位要求。

此外，求职者也只能凭着岗位简介或企业的招聘讲座等形式来判断是否适合该岗位，难以对岗位需求、工作环境和企业文化进行全面的了解。这很容易导致应聘者入职后才发现与岗位并不匹配，从而造成人才资源的浪费，也给用人单位带来较大损失。

1. **成本管理不科学**

目前，大部分高校的专业课程主要以理论教学为主，而让学生独自动手实践的课程较少，这使得部分学生理论与实践严重脱节。但是，企业对学生的实操能力要求较高，并且需要学生能够较快适应工作环境。现实情况往往是，理论学习不扎实且实操经验不丰富，导致求职者上岗后需要从头学起，给企业带来不少的资源浪费，也增加了不少的成本。

此外，各种人力资源进入工作岗位，许多企业面临管人难的问题，如员工的迟到、早退、怠工、非正常离职，上下级关系恶化，工作失误造成公司重大损失，性骚扰等人力资源问题。传统的管理方式，常常是通过公司制度、组织文化、社会道德和领导魅力等“软”性管理方式体现出来，而缺乏从技术、智能、量化等方面记录每个员工的工作履历数据。岗位基本信息、工作绩效、奖罚等级、员工态度等信息缺乏“透明性量化”“分布式留痕”“激励性机制”，这不仅造成人力资源管理成本的上升，而且也不利于形成积极向上的文化氛围。

1. **区块链与人力资源管理的匹配度**
2. **利用区块链的透明性进行招聘**

区块链具有去中心化、分布式存储、数据的时间序列性且不可篡改和否认、智能合约的自动执行、安全和隐私保护等特点。因此，区块链可以帮助我们透过现象看本质，通过纷杂的数据提炼出关键、有价值的信息。我们主要通过财务记录了解一个人的历史，从而推测这个人的能力、道德及诚信程度，但这些记录可以是任何类型的记录——包括教育和职业背景。区块链直接击中学历、履历、简历造假的要害”，可以维护并确保候选人的文凭、参考文献、出版物、考试成绩、医疗证明、犯罪记录和培训证明等信息的真实性。此外，招聘或聘用单位可以使用区块链技术验证候选人的简历，保证数据的透明、可验证性，从技术上解决人力资源领域的诚信问题。

1. **利用区块链数据不可篡改性进行背调**

通过区块链技术，可以安全、真实地访问数据库里面的信息。数据库信息包括人的初始教育、技能、培训和工作经历等可靠的、不可篡改的记录。区块链技术直接增强了工作人员精准寻找适合岗位人才能力，也增加了潜在员工展示其最佳技能的能力，从而提高组织的人才竞争力。个人履历数据一经上链，将不能被篡改，从技术上保证了个人数据的真实性。区块链所代表的“信任证件”，如果用动物世界的语言来表达，就是一本“血统 文件”。一个人职业发展的每一个阶段，都有可能成为区块链账本的一个分录。 区块链分布式账本只有取得各个参与者的同意，才能进行修改。一个人从一个职业生涯转到另一个职业生涯的情况，可以较为容易地进行跟踪。区块链技术帮助企业“猎才”、人才寻觅“英主”，搭建企业、人才之间的“鹊桥”，确保参与方彼此之间的信任，有效避免应聘人员简历造假、企业虚假招聘及平台管理繁琐等弊端。

1. **利用区块链去中心化进行成本管理**

利用区块链技术，实现去中心化，有效降低人力资源管理的成本。

一是区块链让组织管理更加简洁。通过区块链，可以让技术取代许多复杂的工作，节省大量的时间和费用等。去中心化减少了众多人员，压缩了环节，可以降低企业人力的搜索成本和协调成本。二是增强员工的工作积极性。区块链是一串使用密码学方法相关联产生的数据块，每个数据块中包含了一次比特币网络交易的信息，用于验证其信息的有效性和生成下一个区块。类似于“挖矿”，只是这个“挖矿”在人力资源领域拼的不是算力，而是在职场上的表现力。只要你是优秀者、卓越者，就能时时刻刻、清清楚楚得到量化的奖励。

1. **结合区块链技术的人力资源分析**
2. **求职者方面分析**
3. **信息安全**

在传统的招聘环节中求职者通过互联网上的企业事业单位投放的招聘信息，将自己的个人简历传递至第三方网站，而第三方网站通过收集简历、筛选简历、出售简历获取盈利，这也是将个人信息泄露的源头。大量人资企业对求职者的信息保护不到位，造成求职者信息泄露，并且被用于交易，轻则会使得求职者遭受垃圾信息和垃圾电话的骚扰，对其生活产生不必要的麻烦；重则发生互联网诈骗，使得求职者财产损失或者威胁到其生命安全。

结合区块链技术将使求职者的信息安全问题得到缓解，求职者将简历信息储存在IPFS中，IPFS的存储机制是类似于共享模式的应用，它的去中心化的特点让招聘时的第三方没有了可控之机，让求职者简历信息的泄露问题从源头得到解决，从而达到保护求职者简历信息安全的效果。同时IPFS机制是分布式的存储，求职者的隐私安全也得以保证。

1. **信息可信度**

由于互联网求职平台门槛低，只需要手机号码验证便可注册登录，求职者的信息真实性无法得到保障。当求职者信息提交到区块链上，其信息将难以篡改，即使修改的信息也会有清晰的记录。同时一些企事业单位在招聘网站上发布招聘信息有限，通常以最简洁的排版列出所需的招聘信息，这种简单的信息经过美化加工，经常会“误导”求职者，使求职者无法真实且全面了解企事业单位的岗位内容。同样招聘公司将招聘信息提交到区块链上，使信息安全共享，一经提交信息难以篡改。区块链技术将较好的解决双方的信息真实问题，建立健全求职者与招聘方之间的信用体系。

1. **价值变现**

相比大学生的简历学历以及教育经历造假的情况来看，求职者过往的工作经验更难以被证实检验，就目前社会上还没有更加完善的信用生态体系，企事业单位也不会花费大量的人力财力去调查一个人过往的经历，使得企事业单位难以对求职者的职业价值进行评定，一定程度上会影响求职者求职的成功率。在区块链技术下，求职者工作经历也将记录在链上，同时求职者之前所在的公司可对求职者的工作业绩等进行评价，保持求职者经历的真实性，使求职者的过往工作经验得到证实，从而达到提升价值的效果。

1. **招聘者方面分析**
2. **招聘信息公开透明**

人力资源行业涵盖农民工、技术人员、毕业生等多个主体，各个主体的流动性、分散性、信息完整性等并不一致，很容易出现部分人力资源的信息没有采集或采集后没有成功录入的问题，导致人力资源信息的不完善。由于区块链具有去中心化、分布式存储、数据的时间序列性且不可篡改和否认、智能合约的自动执行、安全和隐私保护等特点。人力资源的信息上链后，信息将公开透明，任何员工的有关信息都可在链上进行查验，同时也可通过大量纷杂的数据提炼出需要、关键、有价值的信息。

1. **背景调查方便**

通常招聘方需要在背调上耗费大量资源，且背调困难颇多，在大量招聘工作过程中，应聘者虚增薪酬、夸大工作业绩、虚报任职背景、虚构教育培训经历等让企业难辨真伪，常常造成企业筛选工作成本高、选择效果差等问题。而且优秀者曾经的工作经历由于没有可查的记录，在应聘新职位时也很难被知晓。结合区块链技术，所有信息均储存在链上，数据库信息包括人的初始教育、技能、培训和工作经历等可靠的、不可篡改的记录。个人履历数据一经上链，将不能被篡改，从技术上保证了个人数据的真实性。区块链数据库便于对大量应聘数据进行处理，招聘方可在应聘者许可下直接获得其工作经历数据，减少了大量人力和资金资源上的消耗。

1. **节约管理成本**

区块链技术具有去中心化的特点，利用去中心的特点可以有效节约人力资源的管理成本。结合区块链技术，大量数据在链上，可直接进行查验，可以替代许多复杂的工作，减少了大量的时间和金钱的花费。同时去中心化减少了人员调查消耗，压缩了环节，可以降低招聘方人力资源搜索成本。

1. **基于区块链的人力资源管理方案架构**
2. **技术架构**

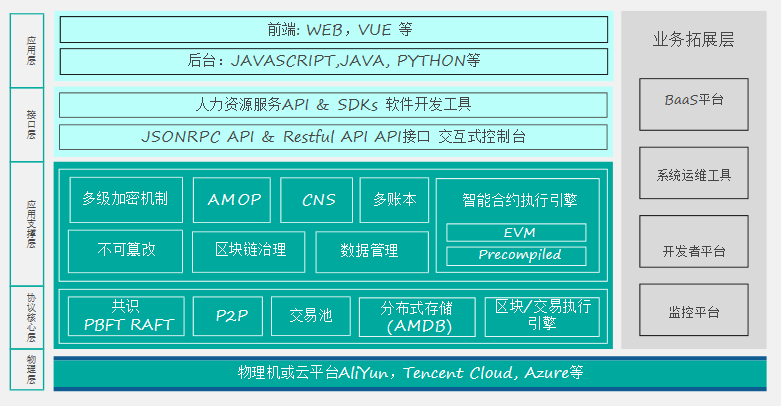


图2 基于区块链的人力资源管理的技术构架图

1. **功能架构**

结合区块链技术该功能架构分为３个层次：

1. 技术层：包括工作证明，p2p验证机制，共识机制等的区块链技术支持。
2. 交易层：交易层实现区块链的核心业务，整个区块链交易过程，包括地址格式、交易格式、智能合约、全局账本、共识机制和激励机制等。
3. 应用层：主要用于对链上数据进行验证和查询。

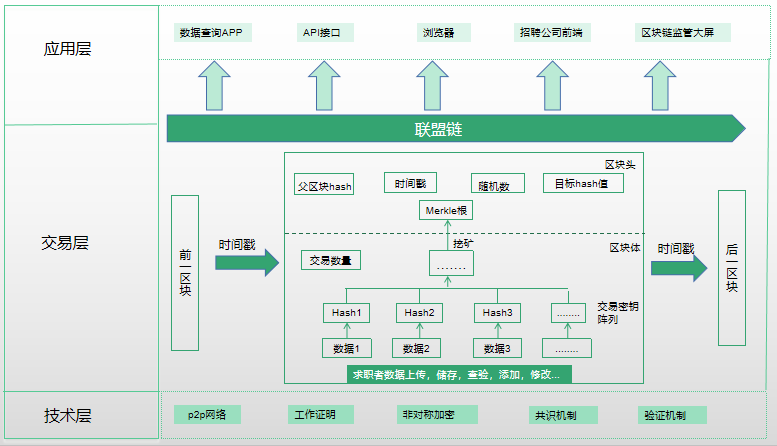


图3 基于区块链的人力资源管理的功能构架图

1. **方案设计架构**

现今企业招聘管理工作在发布招聘需求后，会消耗大量的人力和物力资源来筛选适合岗位的人才，而区块链的使用去除了第三方的介入，会帮助企业使得传统招聘管理工作更加有效和便捷，它甚至会记录这个岗位的应聘者过往的学习经历、职业生涯以及工作经历，区块链技术会验证其真实性，避免了传统简历造假的情况。同时，因为区块链的不可篡改性，保证了应聘者过往经历的真实性，将职业经历看作分布式账本中一笔“数据账”，招聘方单位就可以制定更加有效的方法和内容标准去评测应聘者。

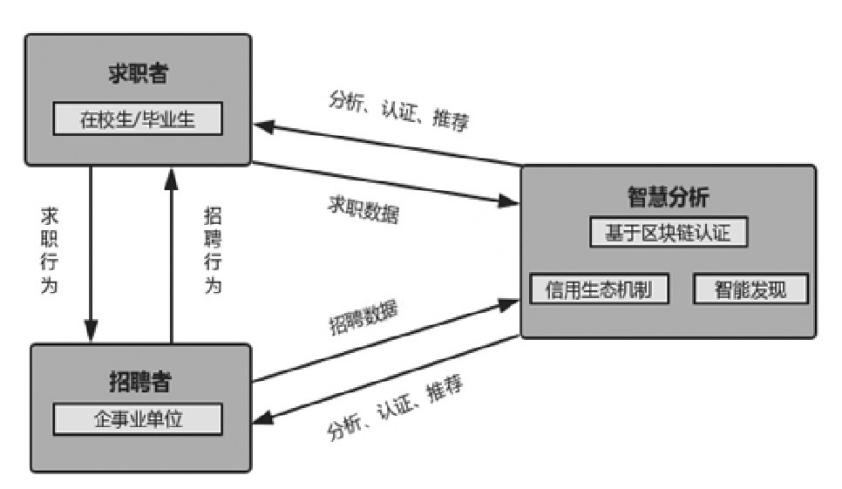
****

图4 基于区块链技术的人力资源管理模式

该模式结构框架由求职者、招聘者和区块链智慧分析三部分构成，具有以下优点：

1. 基于大数据分析求职者的简历以及各方面信息，同时系统也录入招聘方即企事业单位的各种信息，为求职者提供精准的招聘信息，为招聘方提供合适岗位的人才，实现精准匹配。
2. 结合区块链技术，对求职者资料（包括简历，学历，证书及工作经历等）进行真实性认证，对招聘方发布的招聘信息（包括岗位的薪资、公司地址等）安全认证，记录整个招聘活动的环节，建立良好的数字型人力资源管理市场。
3. 解决了价值变现困难的问题，利用区块链技术的分布式架构特点，在招聘环节中可以记录求职者或者招聘方双方的行为，给求职和招聘双方带来较好的诚信交互。
4. **信息上链**

基于区块链技术的人力资源主要有求职者和招聘者或招聘公司两方：

结合区块链技术，求职者信息上链可以省掉中介职能，这些职能包括数据核对、提供收据和采购订单等。通过区块链的IPFS储存机制将求职者信息储存在链上，便于招聘方查验，使招聘方完成交易和其他管理任务所花费的时间大大减少。缩短交易时间，让员工花更多时间在其他重要业务职能上。

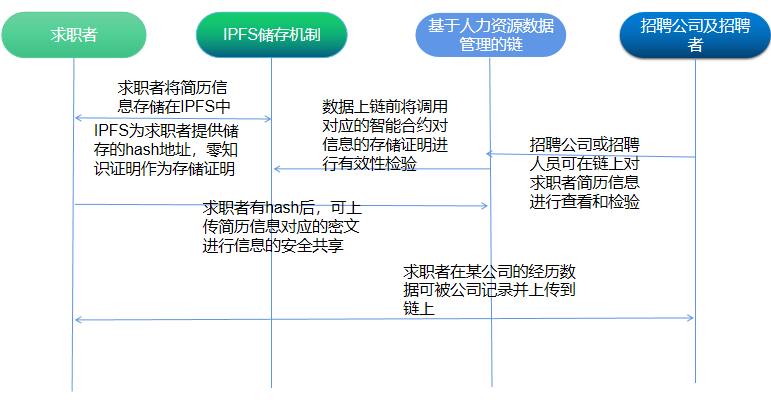


图5 基于区块链的人力资源管理的信息上链图

1. **基于区块链的人力资源管理方案业务分析及具体实施**

**业务分析：**

基于区块链的人力资源管理系统包括求职者简历信息和招聘方招聘信息上链、招聘方公司对简历信息进行查验、求职者工作经历信息的真实性判断等业务需求。

该系统参与交易的主体分别是求职者和多个招聘方公司：首先是求职者需要在基于人力资源数据管理的链上注册用户，每个求职者用户地址将会拥有一个如同身份证一样的“永久数字ID”，该ID对应存储着所有可以用数字衡量的指标，即应聘者的简历信息等各种数据，如可直观看到技能水平、培训经历、教育认证、诚信、 能力优势、个人价值和工作经验等可靠记录。通过区块链技术的声誉系统，招聘方企业能够对应聘者进行快速初步筛选。同时基于区块链的人力资源中招聘方有权对求职者在公司的各种经历进行评价，其中工作记录包括如迟到、上下级关系恶化、受罚、工作失误造成公司重大损失等，求职者的工作记录将保持真实性不可篡改，并且该求职者到下一公司进行求职时，其下一公司可对其工作经历进行查看，则该求职者的工作记录权转移到其下一工作公司。确认权利转移后的下一新公司即可对该职员工作记录进行评价，而该求职者所在的前一公司则再无权利对其工作进行评价。

招聘方在得到求职者的允许后可对求职者简历进行查验，根据求职者储存简历信息对应的身份ID进行查询，得到该求职者的资料包括简历，学历，证书及工作经历等。

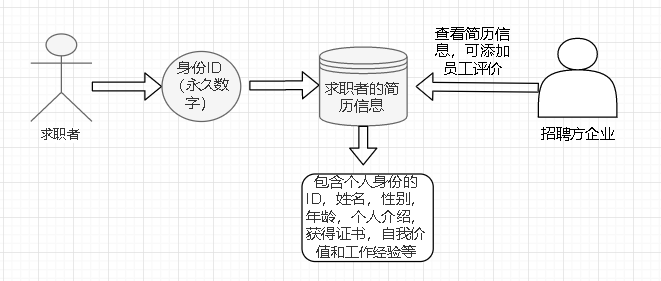


图6 求职方简历信息上链的业务分析图

其次，招聘方公司可在基于人力资源数据管理的链上发布招聘信息，便于求职者清晰的进行选择合适自己的工作职位，招聘信息包括招聘职位，对求职者工作要求，学历，薪资等。求职者都可在链上进行查看招聘信息，当查看到恰好符合自我信息的招聘即可直接进行接收，使招聘方公司快速找到合适的人才。

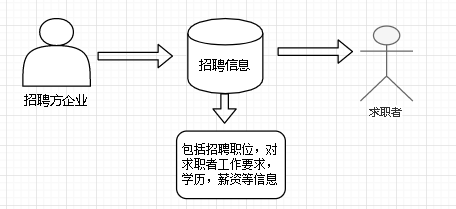


图7 招聘业务分析图

**具体实施：**

【代码加代码的具体分析，及具体的合约交互实施代码解析，方案设计各方面功能，最后交互效果图】

1. **开发环境的配置**

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 计算机硬件配置CPU | Intel(R) Core i7 2.60GHz | Intel(R) Core i5 3.50GHz |
| 内存 | 8GB | 8GB |
| 显卡 | 集成 | 集成 |
| 网卡 | 千兆以太网 | 千兆以太网 |
| 软件配置 | Windows 10系统 | Windows 10系统 |
| 软件 | Remix，solidity0.6.0版本 | Remix，solidity0.5.0版本 |

1. **运行环境的配置**

|  |  |  |
| --- | --- | --- |
| 类别 | 标准配置 | 最低配置 |
| 计算机硬件配置CPU | Intel(R) Core i7 2.60GHz | Intel(R) Core i5 3.50GHz |
| 内存 | 8GB | 8GB |
| 显卡 | 集成 | 集成 |
| 网卡 | 千兆以太网 | 千兆以太网 |
| 软件配置 | Windows 10系统 | Windows 10系统 |
| 软件 | Remix，solidity0.6.0版本 | Remix，solidity0.5.0版本 |
|  | Web3js v 1.3.1 | Web3js v 1.3.0 |

1. **方案部分代码**

合约Resume.sol中部分主要方法代码如下：

//员工自我添加简历，并有权利是否让他人查验

     function AddSelfData(

        uint staffId,

        string memory \_name,

        uint \_age,

        string memory \_gender,

        string memory \_education,

        string memory \_selfvalue,

        string memory \_certification

    ) public onlyStaff{

        staffId = staff\_id;

        Staffs[staffId].username = \_name;

        Staffs[staffId].age = \_age;

        Staffs[staffId].gender =\_gender;

        Staffs[staffId].education =\_education;

        Staffs[staffId].selfvalue = \_selfvalue;

        Staffs[staffId].certification = \_certification;

        Resumedatas[staffId].resumeState == State.Add;

        emit Add(staffId);

    }

//上一公司在员工工作期间可对该员工工作等方面进行评价

    function UpdateResumeData(

        uint staffId,

        string memory \_entryTime,

        string memory \_resignTime,

        string memory \_performance,

        string memory \_code,//分数，奖金等

        string memory \_evaluate

        ) public posted(staffId) onlyLastCompany{

    Resumedata memory temp\_resumedata = Resumedata({

      staff\_id:staffId,

      username:Staffs[staffId].username,

      age:Staffs[staffId].age,

      gender:Staffs[staffId].gender,

      selfvalue:Staffs[staffId].selfvalue,

      education:Staffs[staffId].education,

      certification:Staffs[staffId].certification,

      entryTime:\_entryTime,

      resignTime:\_resignTime,

      performance:\_performance,

      code:\_code,

      evaluate:\_evaluate,

      staffAddress:address(0x0),

      resumeState:State.Posted,

      companyAddress:address(0x0)

      });

       Resumedatas[staffId]= temp\_resumedata;

       Resumedatas[staffId].resumeState == State.Updated;

       emit Updated(staffId);

}

//查询求职者简历信息

function ViewResumeData(uint32 staffid)

public

view

returns (

string memory,

string memory,

string memory,

string memory,

string memory,

address

)

{

return (

Resumedatas[staffid].entryTime,

Resumedatas[staffid].resignTime,

Resumedatas[staffid].performance,

Resumedatas[staffid].evaluate,

Resumedatas[staffid].code,

Resumedatas[staffid].companyAddress

);

}

}

合约Recruiters .sol中部分主要方法代码如下：

// 招聘方公司登录后可添加招聘信息，公开给所有用户查看

    function AddRecruitMsg(

        uint256 \_recruitdataID,

        string memory \_Jobposition,

        string memory \_gender,

        uint256 \_age,

        uint256 \_salary,

        string memory \_companyName,

        string memory \_companyPosition

    ) public onlyCompany {

        RecruitData memory temp\_recruitdata =

            RecruitData({

                recruitdataID: \_recruitdataID,

                Jobposition: \_Jobposition,

                age: \_age,

                gender: \_gender,

                salary: \_salary,

                companyName: \_companyName,

                companyPosition: \_companyPosition,

                recruitState: State.Add,

                companyAddress: address(0x0)

            });

        RecruitDatas[\_recruitdataID] = temp\_recruitdata;

        RecruitDatas[\_recruitdataID].recruitState = State.Add;

        emit Add(\_recruitdataID);

    }

//求职者查询招聘信息

    function ViewRecruitMsg(uint256 \_recruitdataID)

        public

        view

        returns (

            uint256,

            string memory,

            uint256,

            string memory,

            uint256,

            string memory,

            string memory

        )

    {

        return (

            RecruitDatas[\_recruitdataID].recruitdataID,

            RecruitDatas[\_recruitdataID].Jobposition,

            RecruitDatas[\_recruitdataID].age,

            RecruitDatas[\_recruitdataID].gender,

            RecruitDatas[\_recruitdataID].salary,

            RecruitDatas[\_recruitdataID].companyName,

            RecruitDatas[\_recruitdataID].companyPosition

        );

    }

}

1. **方案功能描述**
2. **功能汇总**

**从合约Resume.sol进行分析：**

|  |  |
| --- | --- |
| **子系统A 求职者** | |
| 模块名称 | 功能简述 |
| *添加简历信息* | *定义一个功能“AddResumeData”，允许求职者进行添加自我简历信息* |
| *发布简历信息* | *定义一个函数“PostResumeData”，该函数允许求职者将信息标记为已成功发布状态* |
| **子系统B 求职者所在上一方公司** | |
| 模块名称 | 功能简述 |
| *进行员工工作评价* | *定义函数'UpdateResumeData'，允许求职者所在上一方公司扩充员工简历信息* |
| *转移给下一方公司权利* | *定义函数'TransferResumeData'，该函数允许上一方公司将信息标记为已转移* |
| **子系统C 求职者当前所在公司** | |
| 模块名称 | 功能简述 |
| *接收简历信息* | *定义一个函数“receiveResumeData”，该函数允许员工当前所在公司可标记为已确认接收* |
| *员工评价* | *定义一个函数“ModifyResumeData”，该函数允许员工当前所在公司对员工工作进行评价* |

**从合约Recruiters.sol进行分析：**

|  |  |
| --- | --- |
| **子系统A 公司方** | |
| 模块名称 | 功能简述 |
| *添加招聘信息* | *定义一个功能“AddRecruitMsg”，允许公司方进行添加招聘信息* |
| *发布招聘信息* | *定义一个函数“PostRecruitMsg”，该函数允许公司方将信息标记为已成功发布状态* |
| **子系统B 求职者方** | |
| 模块名称 | 功能简述 |
| *对招聘信息进行查看* | *定义函数'ViewRecruitMsg'，允许求职者对招聘方的招聘信息进行查看* |
| *确认进行应聘* | *定义一个功能“ReceiveRecruitMsg”，求职者确定自我信息符合该招聘信息后确认进行应聘* |

1. **单个系统功能分析**

**在合约Resume.sol中：**

**系统A 求职者方**

|  |  |
| --- | --- |
| 模块名称 | *添加简历信息* |
| 功能描述 | *定义一个功能“AddResumeData”，允许求职者进行添加自我简历信息* |
| 接口与属性 | 定义函数*AddResumeData*，传了七个参数，分别为 uint代表员工个人身份的ID staffId，string求职者姓名，uint 年龄，string，string性别，string 学历，string 个人价值，string 获得证书  使用*AddResumeData*函数添加自我简历信息后，其身份ID对应代表其简历信息 |
| 补充说明 | *该函数仅可求职者方调用* |

|  |  |
| --- | --- |
| 模块名称 | *发布简历信息* |
| 功能描述 | *定义一个函数“PostResumeData”，该函数允许求职者将信息标记为已成功发布状态* |
| 接口与属性 | 定义函数*PostResumeData*，传了一个参数，为uint代表员工个人身份的ID staffId  检查该简历信息的状态是否成功发布 |
| 补充说明 | *该函数仅可求职者方调用* |

**系统B 求职者所在上一方公司**

|  |  |
| --- | --- |
| 模块名称 | *进行员工工作评价* |
| 功能描述 | *定义函数'UpdateResumeData'，允许求职者所在上一方公司扩充员工简历信息* |
| 接口与属性 | 定义函数*UpdateResumeData*，六个参数分别是 uint员工个人身份的ID staffId，string 员工入职时间，string 员工离职时间，string 工作业绩，string 工作分数或获得奖金，string 具体评价  使用*UpdateResumeData*函数进行员工评价后，该员工的简历信息会得到更新 |
| 补充说明 | *该函数仅可求职者所在上一方公司调用* |

|  |  |
| --- | --- |
| 模块名称 | *转移给下一方公司权利* |
| 功能描述 | *定义函数'TransferResumeData'，该函数允许上一方公司将信息标记为已转移* |
| 接口与属性 | 定义函数*TransferResumeData*，传了一个参数，为uint代表员工个人身份的ID staffId  检查该简历信息的状态是否转移成功 |
| 补充说明 | *该函数仅可求职者所在上一方公司调用* |

**系统C 求职者当前所在公司**

|  |  |
| --- | --- |
| 模块名称 | *接收简历信息* |
| 功能描述 | *定义一个函数“receiveResumeData”，该函数允许员工当前所在公司可标记为已确认接收* |
| 接口与属性 | 定义函数*receiveResumeData*，传了一个参数，为uint代表员工个人身份的ID staffId  检查该简历信息的状态是否确认已接收 |
| 补充说明 | *该函数仅可求职者当前所在公司调用* |

|  |  |
| --- | --- |
| 模块名称 | *员工评价* |
| 功能描述 | *定义一个函数“ModifyResumeData”，该函数允许员工当前所在公司对员工工作进行评价* |
| 接口与属性 | 定义函数*ModifyResumeData*，六个参数分别是 uint员工个人身份的ID staffId，string 员工入职时间，string 员工离职时间，string 工作业绩，string 工作分数或获得奖金，string 具体评价  使用*ModifyResumeData*函数进行员工评价后，该员工的简历信息会得到更新 |
| 补充说明 | *该函数仅可求职者当前所在公司调用* |

**在合约Recruiters.sol中：**

**系统A 公司方**

|  |  |
| --- | --- |
| 模块名称 | *添加招聘信息* |
| 功能描述 | *定义一个功能“AddRecruitMsg”，允许公司方进行添加招聘信息* |
| 接口与属性 | 定义函数*AddRecruitMsg*，七个参数分别是 uint 招聘信息序号ID，string招聘的职位及说明，uint年龄，uint薪资范围，string性别，string公司名称，string公司位置  使用*AddRecruitMsg*函数添加招聘信息后，其序号ID分别对应其招聘信息 |
| 补充说明 | *该函数仅可公司方调用* |

|  |  |
| --- | --- |
| 模块名称 | *发布招聘信息* |
| 功能描述 | *定义一个函数“PostRecruitMsg”，该函数允许公司方将信息标记为已成功发布状态* |
| 接口与属性 | 定义函数*PostRecruitMsg*，传了一个参数，为uint招聘信息序号ID  检查招聘信息的状态是否成功发布 |
| 补充说明 | *该函数仅可公司方调用* |

**系统B 求职者方**

|  |  |
| --- | --- |
| 模块名称 | *对招聘信息进行查看* |
| 功能描述 | *定义函数'ViewRecruitMsg'，允许求职者对招聘方的招聘信息进行查看* |
| 接口与属性 | 定义函数*ViewRecruitMsg*，传了一个参数，为uint招聘信息序号ID  使用*ViewRecruitMsg*函数对招聘方招聘信息进行查询为call方法，不需要花费金额，会返回可供求职者查看的招聘具体需求等信息 |
| 补充说明 | *该函数平台多方均可调用，只起到查询作用* |

|  |  |
| --- | --- |
| 模块名称 | *确认进行应聘* |
| 功能描述 | *定义一个功能“ReceiveRecruitMsg”，求职者确定自我信息符合该招聘信息后确认进行应聘* |
| 接口与属性 | 定义函数*ReceiveRecruitMsg*，传了一个参数，为uint招聘信息序号ID  求职者查看招聘信息后，认为符合自我即可使用ReceiveRecruitMsg方法进行确认应聘该招聘职位 |
| 补充说明 | *该函数仅可求职者方调用* |

1. **关键代码分析**

**合约Resume.sol中：**

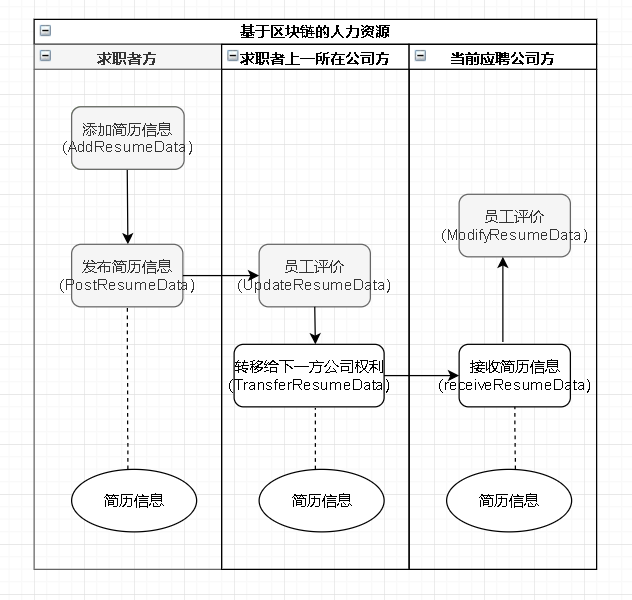
|  |  |
| --- | --- |
| **程序名称** | **添加简历信息** |
| **功能描述** | **定义一个功能“AddResumeData”，允许求职者进行添加自我简历信息** |
| 该功能方法是整个流程中比较关键的第一步，求职者方成功登录系统后首先需进行的操作便是添加简历信息，使用*AddResumeData*方法，传了七个参数，代表求职者个人身份的ID，姓名，性别，年龄，个人介绍，获得证书，自我价值 | |
| **程序名称** | **员工评价** |
| **功能描述** | **函数'UpdateResumeData'，允许求职者所在上一方公司扩充员工简历信息；函数'ModifyResumeData'，允许员工当前所在公司对员工工作进行评价** |
| 员工评价功能是整个流程中重要的一个操作，在该平台中求职者上一所在公司方和其当前所在公司均可对员工工作进行评价，求职者上一所在公司方使用*UpdateResumeData方法，传递*六个参数分别为 uint员工个人身份的ID staffId，string 员工入职时间，string 员工离职时间，string 工作业绩，string 工作分数或获得奖金，string 具体评价，  进行员工的评价操作，评价的信息将直接记录在员工的简历上，对求职者今后求职有一定的影响作用  员工离职后其所在的新公司接收可修改其简历信息的权利后即可使用ModifyResumeData方法对简历信息进行修改 | |

**合约Recruiters.sol中：**

|  |  |
| --- | --- |
| **程序名称** | **添加招聘信息** |
| **功能描述** | **定义一个功能“AddRecruitMsg”，允许公司方进行添加招聘信息** |
| 该功能方法是整个流程中比较关键的第一步，求职者方成功登录系统后首先需进行的操作便是添加简历信息，使用AddRecruitMsg方法，传了七个参数，代表招聘信息的序号ID，招聘的职位及说明，年龄，性别，薪资范围，公司名称，公司位置 | |

1. **方案功能操作流程图**

以基于区块链的人力资源中求职者方简历信息为例，其功能操作流程图如下：



1. **方案代码功能测试**
2. **采用的关键技术**

实现：[以太坊，智能合约，web3js，本地geth]

测试：[智能合约，web3js，本地geth]

1. **接口测试用例**

|  |  |  |
| --- | --- | --- |
| 合约Recruiters.sol的函数原型 | function ViewRecruitMsg(uint \_recruitdataID) | |
| 输入/动作 | 期望的输出/相应 | 实际情况 |
| 典型值… 输入招聘信息序号ID进行查询 | 返回招聘公司所需招聘的职位，招聘具体需求，年龄，性别，工作年限，薪资范围，公司名称，公司位置，公司地址等相关信息 | 正确返回招聘公司所需招聘的职位，招聘具体需求，年龄，性别，工作年限，薪资范围，公司名称，公司位置，公司地址等相关信息 |
| 边界值… 0-无限大 | 成功返回招聘信息或无返回值 | 成功返回招聘信息 |
| 异常值… 负整数（如 -1） | 出现异常，弹出提示输入的值无效 | 错误，弹出提示输入的值无效 |
| 合约Resume.sol的函数原型 | Function AddSelfData(uint staffId,string memory \_name, uint \_age,string memory \_gender, string memory \_education, string memory \_selfvalue,string memory \_certification ) | |
| 输入/动作 | 期望的输出/相应 | 实际情况 |
| 典型值...（1,“姓名”，20，“女”，“本科”，“学习过Java，go语言”，“ETC4”) | 返回求职者添加成功身份ID为1的简历信息，和提交添加简历信息交易成功后所在区块信息 | 成功返回求职者添加成功身份ID为1的简历信息，和提交添加简历信息交易成功后所在区块信息 |
| 边界值…  年龄age为0 | 返回求职者添加成功身份ID为1的简历信息或者返回错误 | 返回错误，弹出提示输入的年龄信息无效 |
| 异常值… staffId为负整数（如 -1） age年龄不符合为负数或已超过一般常人年龄最高限制 | 出现异常，弹出提示输入的值无效 | 错误，弹出提示输入的值无效 |
| … |  |  |

1. **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） | 文字正确且大小合适 |
| 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？ | 可以 |
| 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？ | 有联动关系 |
| 各种界面元素的布局合理吗？美观吗？ | 布局较为合理，符合正常审美 |
| 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？ | 没有奇形怪状 |
| 字体美观吗？ | 统一字体 |
| 图标直观吗？ | 统一图标 |
| 各界面功能是否完善？ | 基本完善 |

1. **系统界面及操作**

**系统访问与登录**

* 访问方式：通过网址直接访问

系统启动后，在浏览器地址栏键入基于区块链的人力资源管理平台，访问http://127.0.0.1:8080/人力资源Dapp/Resume.html地址并回车，浏览器中会出现系统功能页面。

用户注册：系统用户未注册前，我们统一通过基于区块链的人力资源管理平台的职员管理员进行操作，并可通过输入需注册用户的地址信息进行申请相应工作管理的岗位职责权限。

用户注册步骤如下：

* 打开用户注册对话框

成功进入系统功能页面，输入注册的地址信息，点击“提交”按钮。

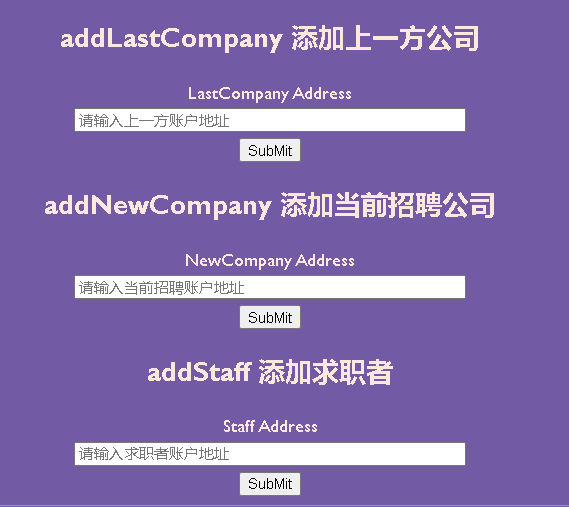


图8 与合约交互的求职者的简历信息注册界面图

* 提交注册流程

输入需注册的用户信息，并选择相应职权后，选择注册审核人，（通常而言注册审核人为系统管理员）点击“提交”后，提示用户注册流程启动。

**求职者方添加简历信息页面**

求职者输入自我信息，身份ID，姓名，性别，年龄等信息进行添加，求职者可根据自我隐私信息需要进行填写信息，性别我们提供了保密的选项，年龄方面可选择进行填写。



图9 与合约交互的求职者的简历信息注册界面图

**求职者方发布简历信息页面**

发布简历信息功能，通过输入身份ID将信息标记为已成功发布状态。

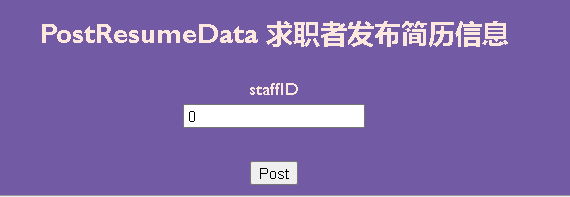


图10 与合约交互的招聘方的添加发布信息界面图

**求职者上一所在公司方评价员工页面**

求职者上一方所在公司在其工作期间有权对其工作等进行评价，输入该员工入职时间，工作业绩，获得重大活动奖金等信息，然后进行评价，该评价将直接更新到求职者的简历信息中。

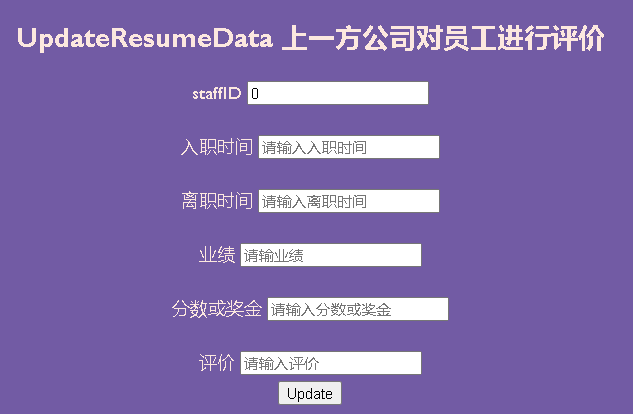


图11 上一方公司对职员评价界面图

**求职者上一所在公司方转移给下一方公司权利**

输入员工身份ID进行权利的转移，通过身份ID与选中的员工简历信息进行对应，消耗一定的币，标记该信息为已转移。

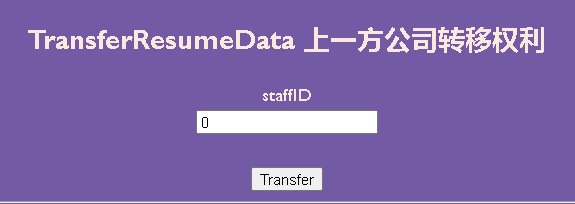


图12 上一方公司评价职员权利转移界面图

**求职者所在的当前公司方确认接收**

输入员工身份ID进行接收，通过身份ID与选中的员工简历信息进行对应， 消耗一定的币，标记该信息为已确认接收。



图13 当前公司评价职员权利转移界面图

**求职者所在的当前公司方进行员工评价**

求职者当前所在公司接收员工简历信息后，在该员工工作期间有权对其工作等进行评价，输入该员工入职时间，工作业绩，获得重大活动奖金等信息，然后进行评价，该评价将直接更新到求职者的简历信息中。

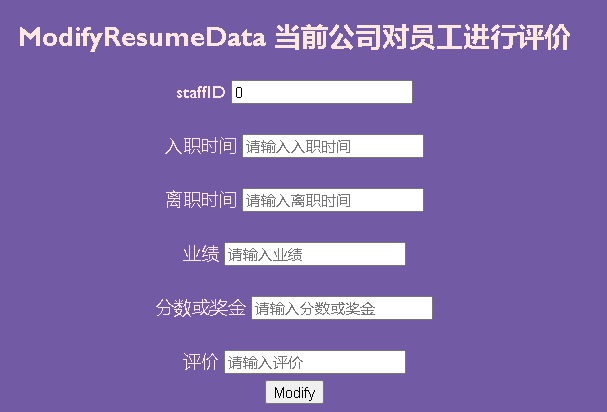


图14 当前公司对员工评价界面图

**查询求职者简历信息页面图**

该平台中的任一公司方都可对求职者的简历信息进行查看，通过求职者的身份ID进行获取信息，点击fetch Data按钮做查询，不用花费金额，返回求职者愿意公开提供的部分简历信息。

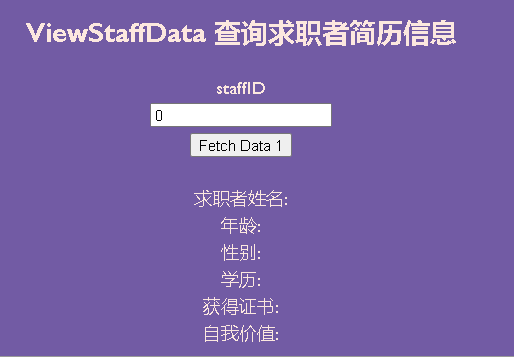


图15 查询求职者简历信息界面图

1. **方案执行预期效果**

基于区块链技术的人力资源管理方案将给人力资源管理平台带来重大转型和挑战。人力资源结合区块链技术将实现人才价值的可追溯管理，而人的价值管理产业潜力巨大，人力资源管理行业将会迎来全新局面。

给求职者带来的预期效果：

1. ****帮助求职者建立个人职业信用****

区块链技术将可能会使职员的“自主主义身份”概念成为现实，因为这些人将对已经由多方验证的数据拥有更大的输入和控制权。

上了基于区块链的人力资源链并经过验证的简历信息，职员的工作将越稳定、获得公司嘉奖表彰次数越多、工作能力表现越优秀、同事评价越高、提供的信息越准确、无不良工作记录（如迟到、上下级关系恶化、受罚、工作失误造成公司重大损失等），那么职员的职业信用值就越高，这样有利于更受到招聘企业的信任、好的评价，更有可能获得工作机会和较高的薪酬标准。

1. **求职者简历验证，提升个人价值**

结合区块链不可篡改性、透明性、智能合约等特性记录个人的职业档案相关数据，包括学历信息、职业历程、培训记录、职场所受奖惩等，甚至包括过往的工作绩效指标、晋升情况以及离职原因都记录在案，而不是任由一个求职者单方在简历上描述他们的工作情况，从而实现应聘方简历信息的真实性、不可篡改性、不可伪造性，真正解决了困扰网络在线招聘多年的招聘信息失真这一问题。充分验证后的成就，资格，经验和愿望将为潜在的雇主节省宝贵的时间和金钱，加速招聘过程，提高了求职者的就业能力，使求职者的个人价值水平得到更高层次提升。

给招聘方带来的预期效果：

1. **缩短招聘时间**

当求职者简历上了基于区块链的人力资源的链并且被验证后，招聘方企业HR就将无需再进行大量背调，节约了人力调查核实求职者背景成本，同时避免了验证过程过于冗长，可以有效缩短招聘周期。企业只需要通过区块链获取所需要的信息即可，区块链大大节省了企业HR的的人力成本和时间成本投入。

1. **降低招聘成本**

运用区块链技术的去中心化特点后，可以降低企业人力的搜索成本和协调成本。通过智能合约，区块链技术将提供无尽的人才供招聘方企业选择，每个人都可以上传并验证区块链平台上的个人简历，从而减少这些昂贵招聘机构存在的必要性。区块链技术有望为真正的人才提供无缝就业的方式，同时确保企业不再无谓地在人力资源领域浪费一分钱。

其次，可以节约企业大量的业务开支。通常人们会认为，省掉中介职能会导致员工失去无数的工作机会。实则不然，招聘方企业可以通过合并这些职能并培训这些员工，让员工在企业风险管理和财务分析中发挥作用，从而避免了大量的失业。

1. **简化招聘流程，轻松高效**

使用区块链技术，招聘方公司的人力资源部门将能够直接访问潜在员工的区块链数据库。这个数据库将包含职员的初始教育、技能、培训和工作经历等可靠的记录。这种有用的信息被称为“价值护照”，从而可以提高招聘方精准找到岗位适合人才的能力，也增加了潜在员工展示其最佳技能的能力，更好地匹配人员职位可以提高生产力。结合区块链技术帮助求职者在最适合他们技能的角色中工作。有效的避免了员工简历造假、企业虚假招聘及招聘平台管理繁琐等弊端，能够让企业人力资源管理更加安全透明。

同时在企业进行招聘时，区块链能够帮助人力资源部门更准确地判断招聘职位有哪些合适的人才，它能够简化当前的复杂招聘流程，提高招聘效率。链上的所有人都可以对自己或他人节点上存在的经历信息、能力信息进行测评，最终自动匹配到用人企业和岗位。

1. **结语**

近年来，区块链信息技术的成功提出和其快速发展， 作为对于传统人力资源信息技术的一种升级和补充，其发展将与其他新兴的信息技术相互交叉融合，该新兴信息技术的日益成熟将为我国乃至世界的经济和社会带来巨大改变。

人力资源是一个历史性的概念。制度经济学派的早期代表人物之一约 翰•洛克斯•康芒斯(John R.Commons),早在1919 年就已在其著作《产业荣誉》里使用“人力资源”一词,但与现在我们所理解的人力资源在含义上相差很远。目前,人们所理解的人力资源是指在一定范围内的人口总体所具有的劳动能力的总和,或者是指能够推动整个经济和社会发展的具有智力劳动和体力劳动能力的人们的总和。人力资源的特征主要具有：

一是不可剥夺性。不可剥夺性是人的价值意义的“内核”与外在展现,它同人 的生命密不可分,同人的尊严与权益相联系。

二是社会性。社会性反映出“人是社会关系的总和”的本质, 展现人的世界观、人生观、价值观、思维方式与行为模式,为人力资源开发运用提供依据。

三是时效性。人力资源的培养、运用、沉淀等工作同人的资源生命周期直接联系,人力资源不同的生命周期所具有的能力、使用价值是不同的。

四是能动性。能动性是指人的主观能动性,能够有目的、有计划地作用和反作用于外部世界,是人力资源作用发挥的前提。人力资源的政策应使其充分发挥正向能动作用,减少和避免负向能动作用。

人力资源具有不可剥夺性、社会性、时效性和能动性等特征，区块链具有去中心化、公开透明、不可篡改和可追溯等特征，人力资源需要区块链技术。人力资源管理存在招聘过程采集信息不精确、背调程序有困难和成本管理不科学等问题。破解人力资源管理中存在的问题，需要发挥区块链的技术优势，通过区块链透明性进行招聘、区块链数据不可篡改性进行背调、利用区块链去中心化进行成本管理。

综上所述，“区块链”技术融入人力资源市场将会给我国人力资源服务教育市场发展带来新的市场机遇和挑战，基于“区块链”技术的人力资源市场生态中，企事业单位与求职者之间会更加透明化且趋于真实性。对此，企业应该加强信用体系建设，淘汰失信者，改变传统招聘市场上信用秩序紊乱的现象，运用新的创新化思维，结合区块链这一新兴技术，对求职者进行更加全面的标准去评测。

附页1

【参考文献】

［1］ 于晓东，刘荣，陈浩.共享经济背景下的人力资源管理模式探索

［2］ 冯桂锋， 区块链与人力资源管理新路径 HR业态

［3］ 郑沛琪，司长宝.“区块链+大数据”图景下的招聘革新探究［J］.大数据时代

［4］ 张雨欣，邢苗条.“区块链”技术在人力资源市场中的应用探讨

［5］ 何永贵，冯缘.基于区块链技术的平台型企业人力资源管理体系研究

「6］ 孟韬，董政，关锤桥.区块链技术驱动下的企业管理与创新管理现代化