

CrowdBC

(下一代安全去中心化的任务众包系统)

电子邮箱: limjnu@gmail.com

摘要

本系统主要研发一套基于区块链的安全众包系统,针对传统的众包平台,雇主(任务发布者)与工作者(任务接收者)几乎都需要依赖可信的第三方来处理任务派发、交易及任务评估等工作,虽然大多数情况下这类系统都运作良好,但是这类系统与传统的金融机构一样受制于“基于信用的模式”,而且中心化服务器会产生单点故障、隐私泄露等问题,同时任务中间过程会产生高昂的手续费。本系统研发的目的在于将传统众包领域中任务发布、接收、评估等过程以智能合约的形式写入区块链,雇主与工作者无需可信第三方中介结构自动完成交易,整个交易的数据具备不可篡改性及可追溯性。此外,本系统利用加密方式来有效保护用户版权及隐私。

本系统不仅可以有效解决在无可信第三方服务机构情况下的任务众包问题,而且可以有效减少了用户的交易成本。

第一章 系统概述

1.1 系统背景

1.1.1 术语定义及说明

序号	术语	名称	备注
1	Crowdsourcing	众包	众包指的是一个公司或机构把过去由员工执行的工作任务，以自由自愿的形式外包给非特定的（而且通常是大型的）大众网络的做法
2	Blockchain	区块链	区块链是分布式数据存储、点对点传输、共识机制、加密算法等计算机技术的 新型应用模式。所谓共识机制是区块链系统中实现不同节点之间建立信任、获取权益的数学算法
3	聚众链(CrowdBC)	下一代去中心的安全众包系统	本系统构建系统的名称（基于区块链的众包系统）

1.1.2 系统产生背景

近年来，越来越多的企业开始选择众包作为其完成任务的一种工作模式。众包的概念首次由 JeffHoew 在 2006 年美国《连线》杂志上提出，众包植根于每个人拥有对其他人有价值的知识或技能，属于电商模式的一种，通过聚集全球各个地方的劳动者，为企业和个人等提供解决方案，涵盖的领域包括工程制造、网页开发、艺术设计等。在目前众多的众包系统中，例如 Freelancer、oDesk、Amazon Mechanical Turk、UBER 以及国内的猪八戒网，雇主将任务发布在第三方众包服务平台，工作者通过竞争来为雇主提供有偿服务。

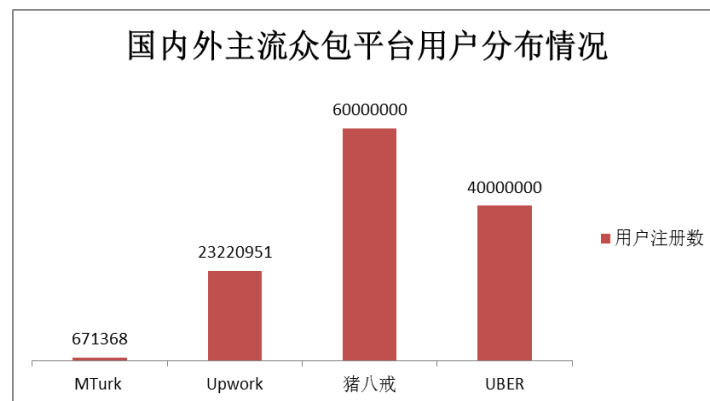


表 1.1 国内外较大众包系统的用户分布情况

随着全球化的推进，外包服务应用蓬勃发展，在全球化 3.0 时代每个人都能以个体为单位参与全球合作与竞争，似乎把外包发挥到极致就成了众包，但二者之间却是有着本质的不同。

表 1.1 众包与外包的区别

类别	众包（Crowdsourcing）	外包（Outsourcing）
地域范畴	不限制区域，任何地方都可以接收任务	基于中心化的区域，局限于一定的范围内
工作时间	任何时间、任何地点都可以工作	工作者完成任务在一定工作时间范围内支持
工作类别	一个任务可接触任何领域，同时会支持多语言	限定在某一类被雇佣的群体
工作报酬	根据任务完成的质量来给予报酬	按照工作者数量以及工作时间
工作成本	由众包平台来链接需求方和开发者，去外包公司化，省去固定成本，故而众包往往比外包更便宜	需要一些固定成本，包括运营人员管理、办公场地、市场费用等

专业的众包平台的出现，解决某些特定行业内的现实需求，通过互联网完成任务获取报酬的形式，突破地域、时间限制，使用抽奖悬赏、在线招标，点对点雇佣等形式，促进了各种资源的优化利用。例如猪八戒网，从 2006 年成立，到现在已经累积了过千万威客群体，累积交易额近 50 亿；近一年的活跃威客数超过 160 万，交易金额超过 20 亿。

1.2 需求分析

在传统的众包模式中，雇主和工作者几乎都需要依赖可信的第三方来处理任务派发、交易及任务评估等工作。虽然大多数情况下这类系统都运作良好，但是这类系统与传统的金融机构一样受制于“基于信用的模”，而且中心化服务器会产生单点故障问题。最近几年，有许多事故产生是由于服务过于集中化管理而导致出现问题，主要针对系统可用性以及隐私泄露问题。

□ 单点故障

2014 年，UBER-中国由于一台服务器故障而出现阻止用户关闭和停止订单的问

题，只能由车主来进行订单停止，导致服务使用出现故障。

❑ 隐私泄露

Freelancer 在 2015 年 10 月，OAIC 组织发现 Freelancer 违背隐私法而将一位前使用用户的私人信息：IP 地址，假名和其他细节公布出来。

❑ 中心化信任模式

传统众包模式中另外一个问题在于现有众包过程中的任务金额由第三方平台来保存，并且默认雇主和工作者相信第三方平台作为中介机构不会吞没任务金额，然后情况并非总是如此。

❑ 成本高昂

由于第三方中介机构的存在，雇主会增加支出成本，工作者则会减少收益成本，例如 oDesk，作为买家和承包商之间的中介，买家可直接与劳动力签订合同，利用 oDesk 平台支付给自由职业者并与其交流，oDesk 会从中抽取 5%-20% 不等的佣金。

目前传统的众包解决方案都需要依赖可信第三方来设计相应的信任机制、激励机制与结果质量评估，随着越来越多的第三方出现信任危机，开发一套不需要依赖不可信第三方的众包系统成为非常必要的事情。为了解决上述问题，本系统将区块链技术与众包模式相结合，通过制定合约来解决无需可信第三方的问题，并且基于该模型下，雇主和工作者可以利用本发明提供的方案进行公平交易，雇主获得想要的解决方案，工作者从中获得有偿服务。

1.3 相关工作分析

目前出现的众包平台本质上是中心化的管理机制，包括 Upwork、UBER 以及国内的猪八戒网、威客网站等等，这些平台接收任务发布者发布的任务，主要面向一些计算机难以解决而人比较容易解决的问题，同时会对发布者收取一定额度的保证金，然后将任务公布在平台中，所有系统中的相关工作者都可以接收任务并进行完成，完成之后提交答案并汇总的过程依赖于该平台来完成，同时每笔任务需要收取一定额度的佣金，大约按照 5%-20% 的比例进行收取。

❑ The DAO

区块链作为比特币的底层实现技术，具有去中心化、不可篡改等特性，以太坊（Ethereum）是一个平台和一种编程语言，使开发人员能够建立和发布下一代分布式

区块链应用。在以太坊上实现的应用目前接近三万多种,其中 The DAO(Decentralized Autonomous Organization, 去中心化的自治组织)众筹项目属于其中一种比较大项目,最高融资达到 1.6 亿美元。众筹属于众包的一种模式,通过发起一个众筹任务,其他所有人都是可以参与到该任务中。The DAO 不受任何第三方的控制,属于完全中心化的管理模式,是否投资一个项目由参与该众筹项目的人共同来进行决策。

❑ Microwork.io

Microwork 是国外一家初创公司,面向全球的自由职业者,基于区块链来建立任务发布机制,最具创新性功能是在于没有第三方帮助的情况下完成合同的能力。虽然对于在平台上发布的每种类型的工作来说显然不起作用,但是有些任务可以通过智能合同来检查数据完成的状态来确认。

1.4 特色及主要技术难点

本系统的主要特色在于以下几个方面:

❑ 去中心化

本系统目的在于建立基于区块链去中心化的众包系统,该系统无需依赖第三方中心化服务器和管理商,不存在单点故障问题,系统的安全性保证由成千上万个分布式的节点共同来维护。

❑ 智能合约

智能合约是以一套数字形式定义的承诺,包括合约参与方可以在上面执行这些承诺的协议,本系统将用户管理、任务的发布、任务的接收、任务的评价等通过智能合约的形式来自动执行与完成,且合约本身在区块链平台无法被篡改。

❑ 信誉度系统

为了有效激励工作者在系统中接收任务,本系统建立基于信誉度的管理机制,为每位工作者分配一个信誉值,信誉值初始值为 60(最大 100),随着工作者接收任务的完成情况来不断更新,该信誉值体现在接收任务的成功概率上,信誉值越大,则工作者接收任务的几率更大。

❑ 匿名保护

本系统针对传统众包系统中存在的用户个人隐私泄露的问题,利用一对公私钥地址来代表用户的个人信息,无需注册相关真实身份信息,例如姓名、身份证、电

话等敏感信息，通过该公私钥地址即可完成在该系统中的任务众包过程。

主要的技术难点：

□ 智能合约

智能合约虽然是图灵完备的语言，但是由于合约的发布和执行都涉及交易费，无法将所有的过程都在智能合约中实现，同时，需要将系统的功能逻辑和业务逻辑进行区分，如何设计几种有效的智能合约来描述任务众包的过程成本本系统的难点之一。

□ 任务结果评估

本系统将任务发布出去之后，需要对任务的结果进行评价，一般发布的任务是机器难以实现的，需要通过人来现实，同样对任务结果的评估也需要人来完成，如何保证结果评价的公平性问题，不会出现任务完成实际情况比较好，工作者得到差点评价，或者工作者实际结果完成不好，却想获得更多的奖励，这些问题是本系统中需要着重考虑的。

第二章 系统设计与实现

2.1 系统总体方案

本系统是基于区块链的任务众包服务系统，为自由职业者提供众包服务，与传统的方案不同，该平台无需依赖可信第三方服务平台，任务发布者与接收者之间通过智能合约的形式达成一致。在该系统中，通过以太坊上的智能合约为用户之间达成一致性约定。本系统主要分为三层结构：应用层、区块链层、数据存储层。应用层作为用户接口操作层，将用户的输入信息与区块链平台进行对接，包括用户注册管理、任务管理两个主要部分。区块链层在本系统中选择以太坊作为系统实施平台，主要运行几种智能合约：用户注册管理合约、用户个人信息汇总合约、发布者-工作者接收合约。数据存储层在系统方案设计中采用分布式云端数据库，用于存储在众包过程中产生的大量数据，对于任务的元数据信息，本系统将其保存在区块链平台，能够有效扩展系统的存储容量，并且云端数据被篡改能够通过区块链平台进行识别。

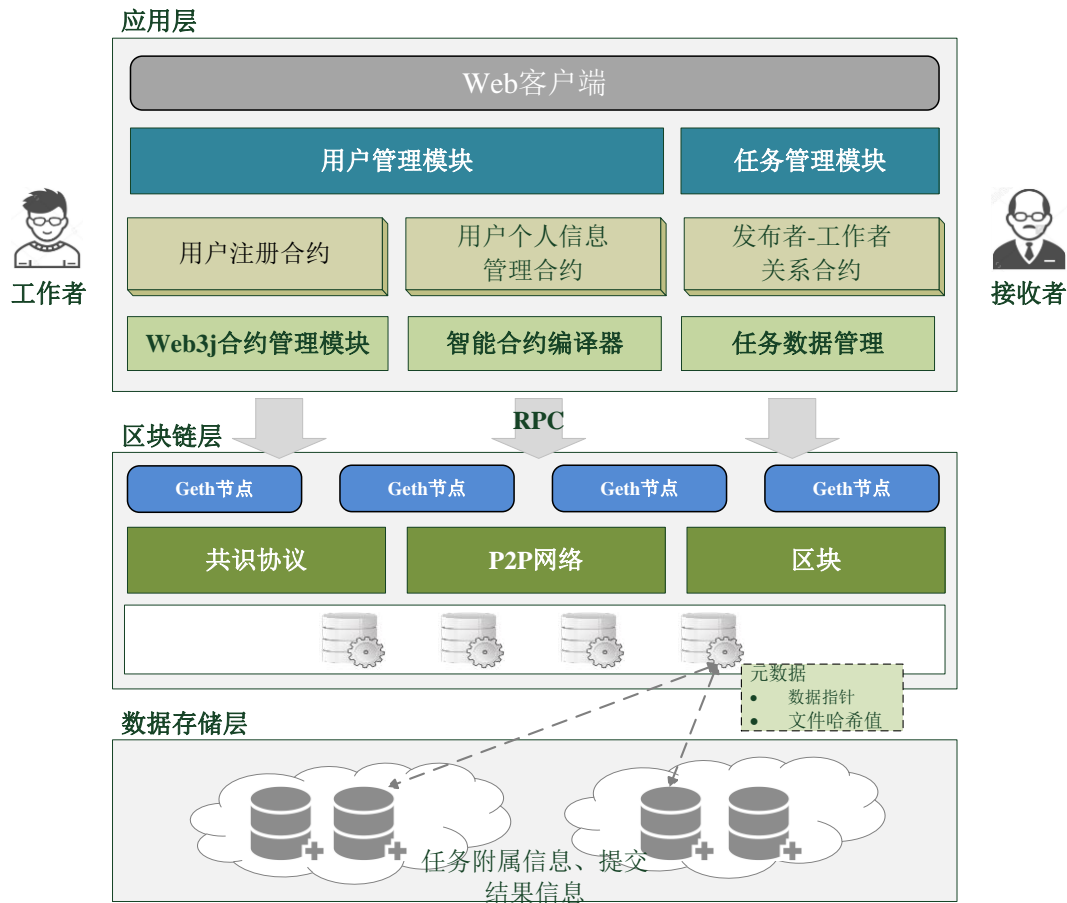


图 2.1 本系统实现的总体架构图

2.2 系统实现原理

本系统实现原理主要按照系统的总体架构，主要分为三层结构：应用层、区块链层、数据存储层。

□ 应用层

应用层属于面向用户的接口层，主要功能包括：用户管理模块、任务管理模块、Web3j 智能合约管理模块、智能合约编译模块以及数据库管理模块。应用层通过 Web 浏览器的形式运行起来，无需远端 Server，只需要本地部署一个 Tomcat 7.0 及以上的进程即可运行。发布者、接收者在使用该平台之前首先进行注册，注册主要是将用户账号，密码，信誉值以及钱包地址等相关信息导入到本系统中，同时通过智能合约在区块链中完成注册。此后，发布者通过应用层金额口提交任务，并且保证任务提交时将奖励锁定在区块链中，在完成时间之内无法被取走。工作者可以通过应用层查询到当前待完成的任务列表，并按照需求进行投标，如果工作者与发布者达成智能合约，

该合约中规定任务完成的时间、需求以及奖励等信息，并以交易的形式写入区块链中。

本系统设计了三套通用的智能合约模板：用户注册合约（**Register Contract**）、用户汇总合约（**User Summary Contract**）、发布者-工作者关系合约（**Requester-Worker Relationship Contract**），该模板被公布在 **GitHub** 中，所有人都能够获取，如果工作者和发布者了解智能合约可以根据需要达成的要求修改模板信息，附带数字签名信息，直接通过以太坊钱包将合约公布在以太坊平台中，并等待以太坊节点确认，确认过程被认为是一种基于安全密码学的交易行为，可以触发任务向下一个状态进行转变。另外，个人用户可以不需要懂智能合约代码，利用本系统的 **Web** 客户端也可以完成交易。本系统开发基于 **Web3j** 的智能合约管理模块，**Web3j** 支持以太坊的 **JSON-RPC** 客户端 **API** 的实现，利用 **Java** 即可完成对智能合约的读写操作。

❑ 区块链层

区块链层属于中间层，本系统将以太坊（**Ethereum**）作为区块链平台，以太坊可以建立和发布新一代的分布式应用平台，支持图灵完备的智能合约。应用层经过编译后的合约通过 **RPC** 接口发送到区块链层，矿工(**Geth** 节点)对合约有效性进行确认，确认通过则写入区块链数据层。**CrowdChain** 将元数据信息被写入区块链层，详细数据被写入到下层的数据存储层，区块链层基于以太坊平台提供的共识机制来保证所有节点所存储信息的一致性。

为描述状态机的运行过程，本系统建立基于任务的状态机描述，每一个新建的任务可以被看成是一个新的状态机，由已经公布在区块链的交易合约作为输入，来触发任务状态向下一个任务状态迁移，发布者和工作者可以通过任务状态查找到所有已公布但是还未被解决的任务进行处理。任务的状态主要包括：待上传、未接收、已接收、评估中、取消和完成状态。任务被发布者提交以后在未被确认以太坊节点之前处于待上传状态，确认通过之后，该任务信息被写入区块链并进入未接收状态，工作者可以在平台中搜索到所有关于待上传和未接收的任务列表，工作者通过提供初步方案竞争来获取该任务的处理权，但如果在一时间内没有工作者处理，则该任务超时需要重新进行提交。在已接收状态时，如果有工作者提交方案，任务将会进入评估状态，所有评估完成则任务生命周期结束。

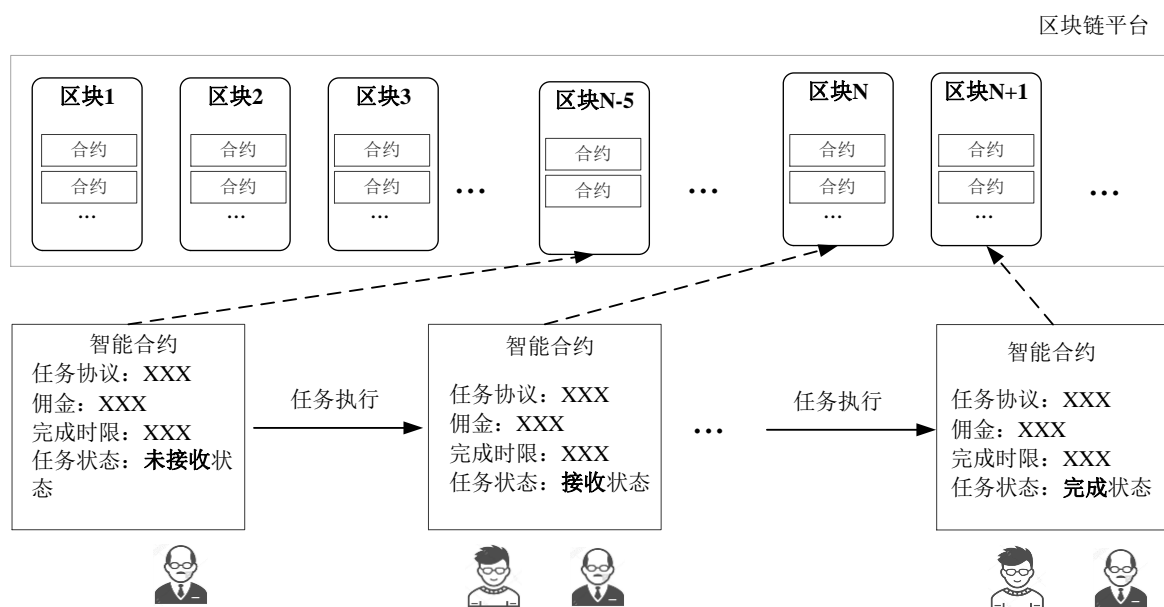


图 2.2 系统中任务流转图

□ 数据存储层

数据存储层属于系统的最底层，本系统通过将元数据存储在区块链层，详细数据存在存储层的方式有利于减少区块链中的数据大小，提升区块链上的读写效率。我们采用开放式接口的方式支持云平台存储方式数据存储方式，用户无需信任该层数据的真实性，在区块链层已经记录所有数据的哈希值，可以在该层来核对数据的完整性。在这种设计模式下，不仅可以提高数据的存储空间，而且有效减少发布者和接收者之间的交易成本。

2.3 系统功能设计

□ 用户注册管理

在用户使用本系统进行任务发布及接收之前，需要预先进行注册，注册的信息包括：用户名、密码、描述等，同时系统为新注册用户设定相关参数的默认值，包括已接收任务数量、已完成任务数量、信誉值等。

模块编号	001
模块名称	用户注册
输入	用户名、密码、确认密码、钱包地址、描述
处理	将以上用户信息输入通过前台传到后台 Controller，后台利用

	Web3j 将信息写入到用户注册合约（URC），同时生成一个新的用户汇总合约（USC）
输出	注册成功/失败

□ 用户登陆管理

用户登陆之前需要进行注册，注册完成每个用户都会有对应的用户名及密码，而且此信息通过加密的形式保存在区块链中，登陆过程的验证也是通过 Web3j 进行接口调用来验证用户是否在用户注册合约（URC）中进行过登陆，验证成功则登陆任务管理界面，失败则提示用户用户名及密码错误。

模块编号	002
模块名称	用户登陆
输入	用户名、密码
处理	将以上用户信息输入通过前台传到后台 Controller，后台利用 Web3j 将信息发送至区块链平台进行确认，确认过程会调用合约中的已注册用户函数，判断是否已经注册，并判断用户名及密码是否正确
输出	登陆成功/失败

□ 用户密码修改

本系统支持用户进行密码修改，如果用户需要重新设定密码，可以登陆用户密码修改界面，输入旧密码成功以后进行密码修改，同样的新密码会同步到区块链中。

模块编号	003
模块名称	修改密码
输入	用户名、旧密码、新密码
处理	将以上用户信息输入通过前台传到后台 Controller，后台利用 Web3j 将信息发送至区块链平台，首先会确认该用户名是否存在，如果存在则进一步判断旧密码与注册密码是否相等，如果相等则重新修改区块链中保存的密码

输出	修改成功/失败
----	---------

□ 用户账户管理

一个用户支持有多个账户，在发布任务和接收任务时支持选择不同账户进行支付。查看用户个人账户，会显示用户相关的个人信息，包括注册时间、名称、已接收任务数量、信誉值等基本信息参数。

模块编号	004
模块名称	显示用户账户信息
输入	无
处理	根据当前已登陆用户名，读取区块链中用户汇总合约（USC）中的用户信息、包括注册时间、名称、已接收任务数量、信誉值等
输出	读取成功/失败

□ 发布任务

已注册的用户可以在平台中发布一个任务信息，发布任务信息包括：任务押金、任务奖金、任务描述、任务完成时间、任务所需工作者数量、最低信誉值要求值。发布任务的过程需要将用户输入的信息同步至区块链中，任务初始状态为待接收状态，其他满足要求的工作者可以进行任务接收。

模块编号	005
模块名称	任务发布管理
输入	任务押金、任务奖金、任务描述、任务完成时间、任务所需工作者数量、最低信誉值要求值
处理	任务发布的过程是将用户前段输入的任务信息同步至区块链的过程，新发布的任务会通过 Web3j 新产生一个“发布者-工作者关系合约”，合约初始状态为“待接收”状态，同时发布一个任务会将工作者的押金进行锁定，在任务完成时间结束之前，押金不能被撤回

输出	发布成功/失败、发布的任务地址
----	-----------------

□ 接收任务管理

工作者可以进行任务接收，工作者会查找适合接收的任务，查看任务当前的状态，如果可以则点击接收功能，任务接收过程中需要工作者输入对应的密钥信息，为了保证交易的公平性，系统支持两种选择来让用户进行任务接收，一种是工作者使用押金作为输入，另一种是使用信誉值作为输入，如果任务完成之后获得正面评价则押金或者信誉值被赎回，否则将被系统进行扣除（押金转入对应任务发布者账户）。

模块编号	007
模块名称	任务接收管理
输入	任务地址、工作者任务押金、工作者私钥地址
处理	任务接收过程对应“发布者-工作者关系合约”中的 <code>receiveTask</code> 函数，该函数首先对工作者的资质进行判断，主要看信誉值是否满足要求，同时确认当前任务是否可以继续接收工作者，如果可以则将其签名信息放入到任务接收池中，接收成功并且任务数量已达到最大值后，任务状态向“已接收”转变
输出	接收成功/失败

□ 任务查询管理

系统中可以查询到当前所有已发布到区块链中的任务及其状态信息，或者根据任务地址进行查询状态查询。

模块编号	008
模块名称	任务查询管理
输入	任务地址、任务状态、发布时间
处理	任务查询过程会依据用户输入的查询条件，利用 <code>Web3j</code> 遍历所有的任务信息，用户注册合约（ <code>URC</code> ）中保留了所有已发布任务的 <code>ID</code> ，可以根据 <code>ID</code> 来找到对应任务的信息并进行过滤
输出	符合条件的任务信息

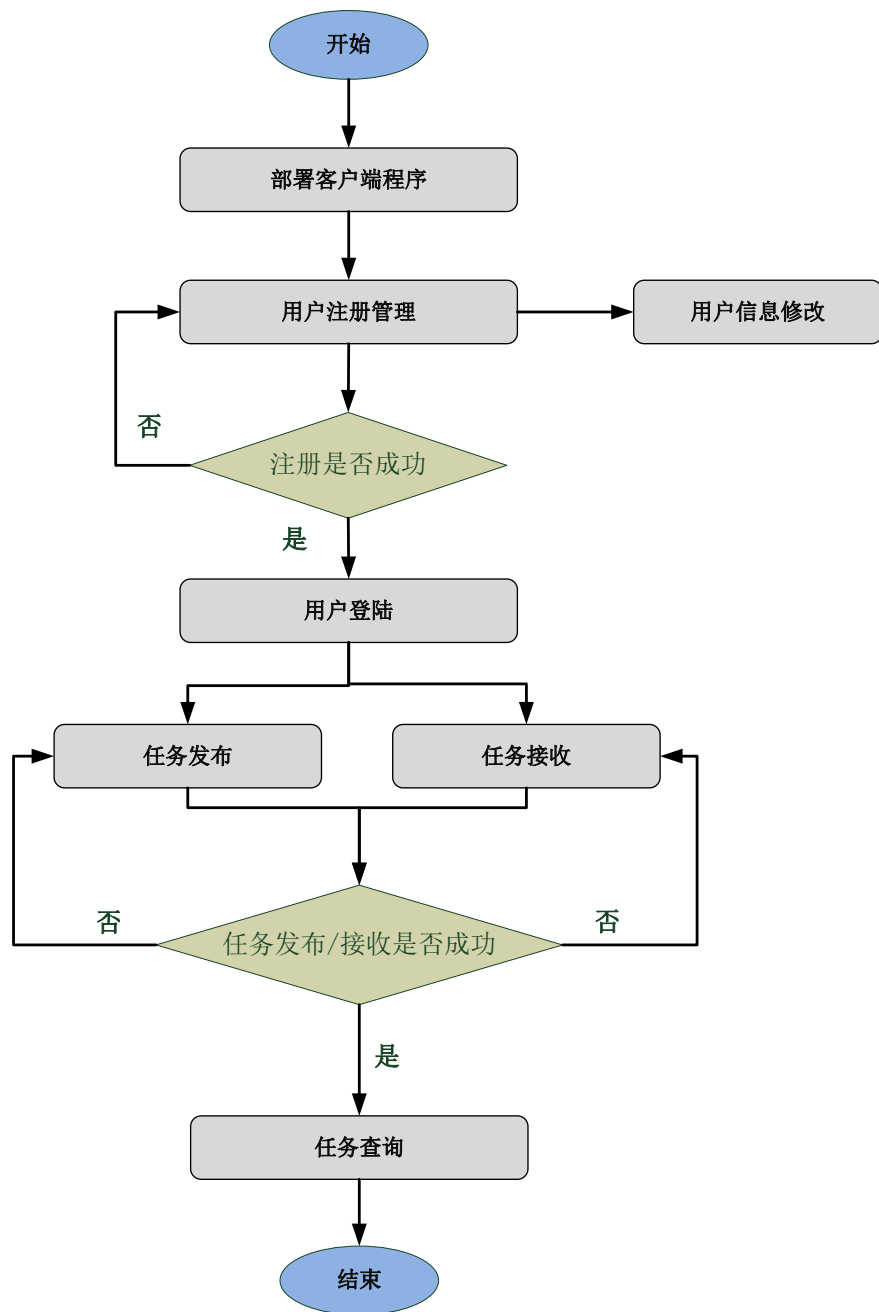
□ 任务状态管理

如果工作者已完成任务提交或者任务完成时间已到，那么发布者可以对任务状态进行操作，主动发起对任务的评价。

模块编号	009
模块名称	任务状态管理
输入	查询条件：任务地址、任务状态、任务关键字
处理	所有已发布的任务通过 mapping 池保存在用户注册合约中（URC），通过 URC 合约可以找到所有任务的 ID，然后依据 ID 逐一进行遍历即可找到任务信息，根据关键字过滤任务得到用户查询的地址信息
输出	满足条件的任务列表

2.4 系统软件流程

本系统软件运行的整体流程图如下，从系统开始，首先需要部署客户端程序，客户端程序为运行在 Tomcat 下的 Web 页面程序，用户可以下载官方的 Apache-tomcat 程序。打开页面首页之后，如果用户还未进行过注册，首先参照用户注册管理的要求进行注册，同样的支持用户进行信息修改。注册成功后用户进行登陆进入相应的任务管理界面，任务管理主要包括三个部分：发布任务、接收任务、查询任务。如果是发布者可以选择任务发布功能，填写任务信息将任务公布出去，如果是工作者则进行接收任务界面，选择相应的任务，输入密钥信息接收该任务。成功之后，发布者/工作者可以在任务查询页面查询任务信息。



2.5 系统运行指标

❑ 最大用户注册数

为保证系统运行质量，本系统设置了最大注册用户数，默认设置为100000，可支持系统发布者进行修改。

❑ 最大并发用户数

由于服务端为以太坊服务端，以太坊是分布式的区块链平台，最小支持上万用户的并发访问。

❑ 每秒能处理的请求/事务的数量

由于以太坊区块大小的限制，每秒能处理的交易数据达到100000笔。

❑ Response Time响应时间

系统每次请求的响应时间与以太坊操作时间相匹配，写入或者修改交易信息需要秒级的响应时间，查询类服务毫秒级时间。

❑ 用户信誉度

用户信誉度的取值范围为[0, 100]，初始值为60。

❑ 工作者同时最多接收任务数

系统为保证完成质量，工作者同一时间点最多接收2笔任务。

❑ 任务信息大小

由于任务信息保存在云端数据库，任务信息的大小不受限制，哈希值保存在区块链中。

❑ 交易确认时间

以太坊每14秒产生一个区块，因此交易确认最小时间为14秒。

第三章 系统测试与分析

3.1 测试总体方案

在 Windows7 系统下搭建系统运行环境，并测试整个系统所实现的所有功能。

3.2 测试环境与配置

□ 测试环境

操作系统	Windows7
客户端	firefox 浏览器 53.0.3
服务端	以太坊客户端 geth1.4.18 以太坊钱包 Ethereum Wallet0.8.9 Java 服务器 Tomcat8.5 Java 开发环境 Eclipse Jee Neon4.6（含 maven）

□ 环境配置步骤

[1] 安装以太坊客户端 **geth1.4.18**、以太坊钱包 **Ethereum Wallet0.8.9**

1、从官方网站下载 **geth1.4.18** win64 客户端，解压缩即可运行。

下载地址：<https://github.com/ethereum/go-ethereum/releases/>

为方便起见，可从我们上传的 ExecutableFiles 文件夹找到 geth-win64 文件夹，双击 startup.bat 即可启动。

2、从官方地址下载 **Ethereum Wallet0.8.9** win64 客户端，解压缩即可运行。

下载地址：<https://github.com/ethereum/mist/releases/>

为方便起见，可从我们上传的 ExecutableFiles 文件夹找到 Ethereum-Wallet-win64-0-8-9 文件夹，双击 Ethereum-Wallet.exe 即可启动。

[2] 配置创世块

配置自己的创世块是为了区分公有链，同一个网络中，创世块必须是一样的，否则无法联通，此方法在 windows 和 Ubuntu 下通用。新建一个名为 piccgenesis.json 的 json 格式文件，输入如下内容并保存到与 geth 客户端相同的目录下。

```

piccgenesis.json*  X
架构: <未选择架构>
1  {
2    "nonce": "0x0000000000000042",
3    "mixhash": "0x0000000000000000000000000000000000000000000000000",
4    "difficulty": "0x4000",
5    "alloc": { },
6    "coinbase": "0x0000000000000000000000000000000000000000",
7    "timestamp": "0x00",
8    "parentHash": "0x0000000000000000000000000000000000000000000000000",
9    "extraData": "PICC GenesisBlock",
10   "gasLimit": "0xffffffff"
11 }
12

```

mixhash	与nonce配合用于挖矿，由上一个区块的一部分生成的hash。注意他和nonce的设置需要满足以太坊的Yellow paper, 4.3.4. Block Header Validity, (44)章节所描述的条件。.
nonce	nonce就是一个64位随机数，用于挖矿，注意他和mixhash的设置需要满足以太坊的Yellow paper, 4.3.4. Block Header Validity, (44)章节所描述的条件。
difficulty	设置当前区块的难度，如果难度过大，cpu挖矿就很难，这里设置较小难度
alloc	用来预置账号以及账号的以太币数量，因为私有链挖矿比较容易，所以我们不需要预置有币的账号，需要的时候自己创建即可以。
coinbase	矿工的账号，随便填
timestamp	设置创世块的时间戳
parentHash	上一个区块的hash值，因为是创世块，所以这个值是0
extraData	附加信息，随便填，可以填你的个性信息
gasLimit	该值设置对GAS的消耗总量限制，用来限制区块能包含的交易信息总和，因为我们是私有链，所以填最大。

图 3.2.0

[3] 创建钱包

1、启用以太坊客户端 geth:

当看到命令行中出现 Listening on [::]:30303 和 Welcome to the Geth JavaScript console!的提示，说明已经启动成功。

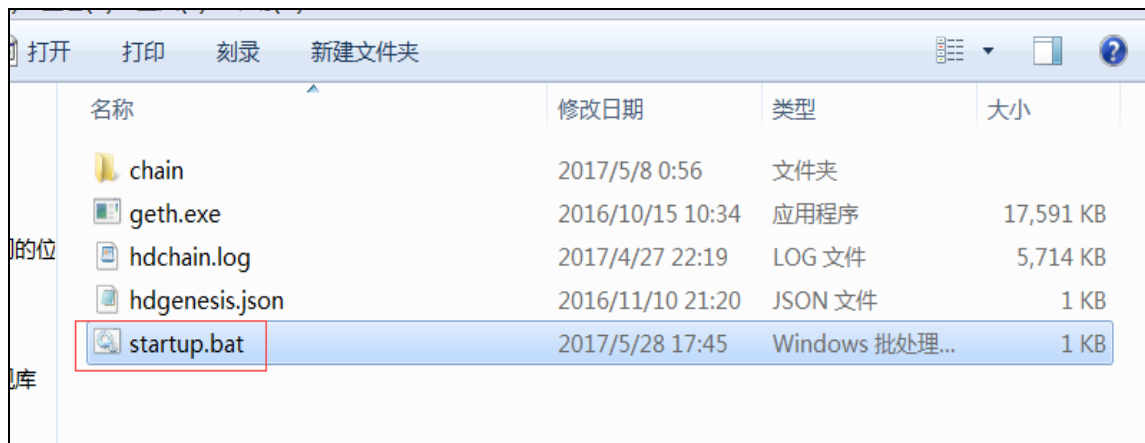


图 3.2.1

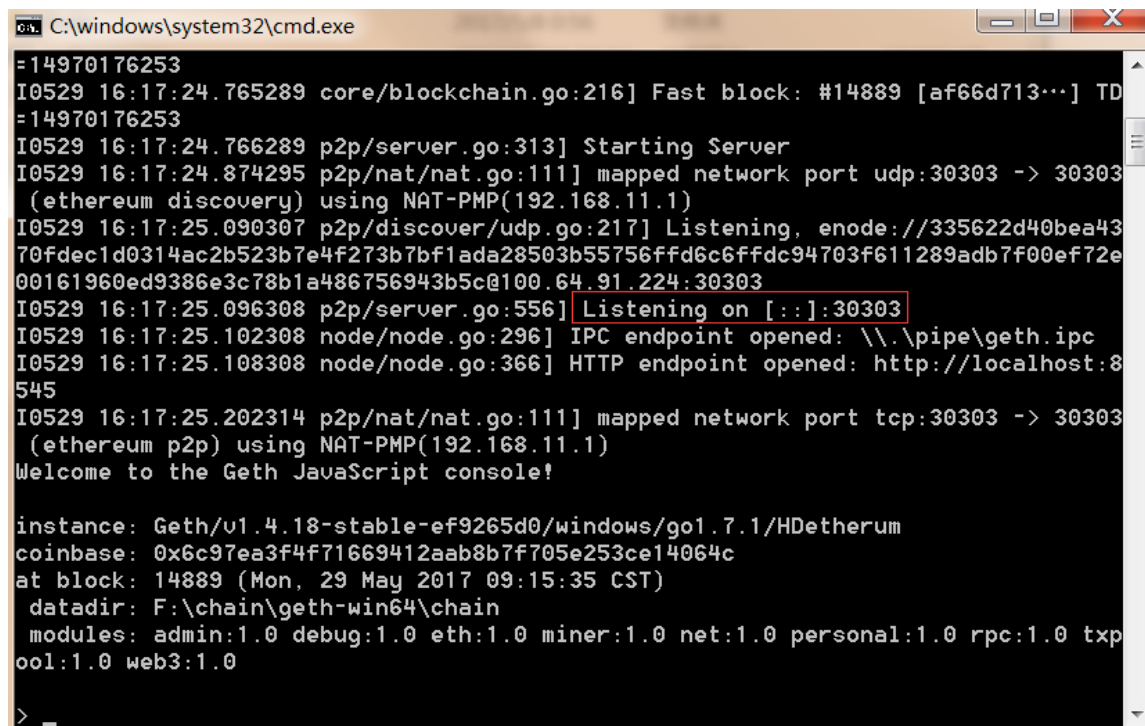


图 3.2.2

2、启动以太坊钱包 Ethereum-Wallet:

点击“ADD ACCOUNT”按钮，添加一个钱包，程序会弹出一个对话框，提示输入密码，确认密码后，账号即创建成功。

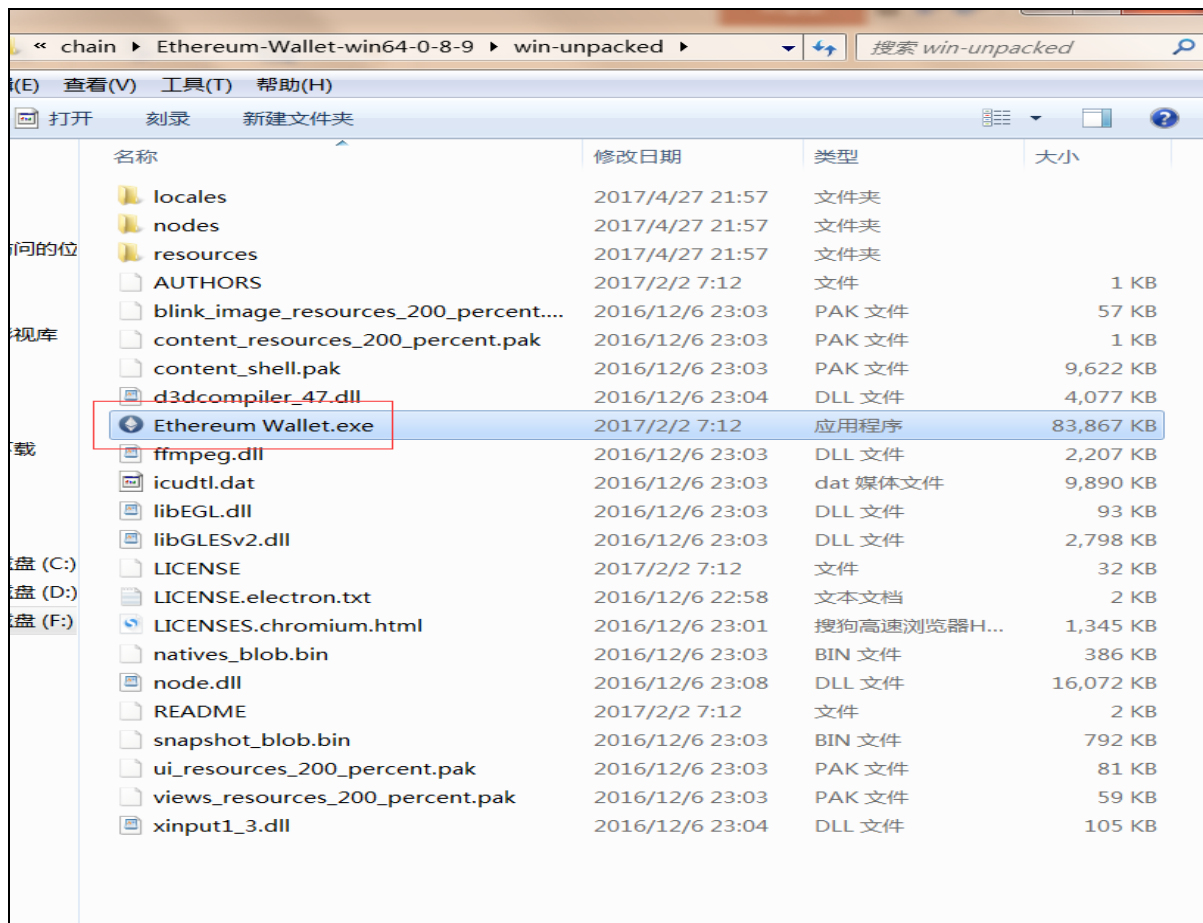


图 3.2.3

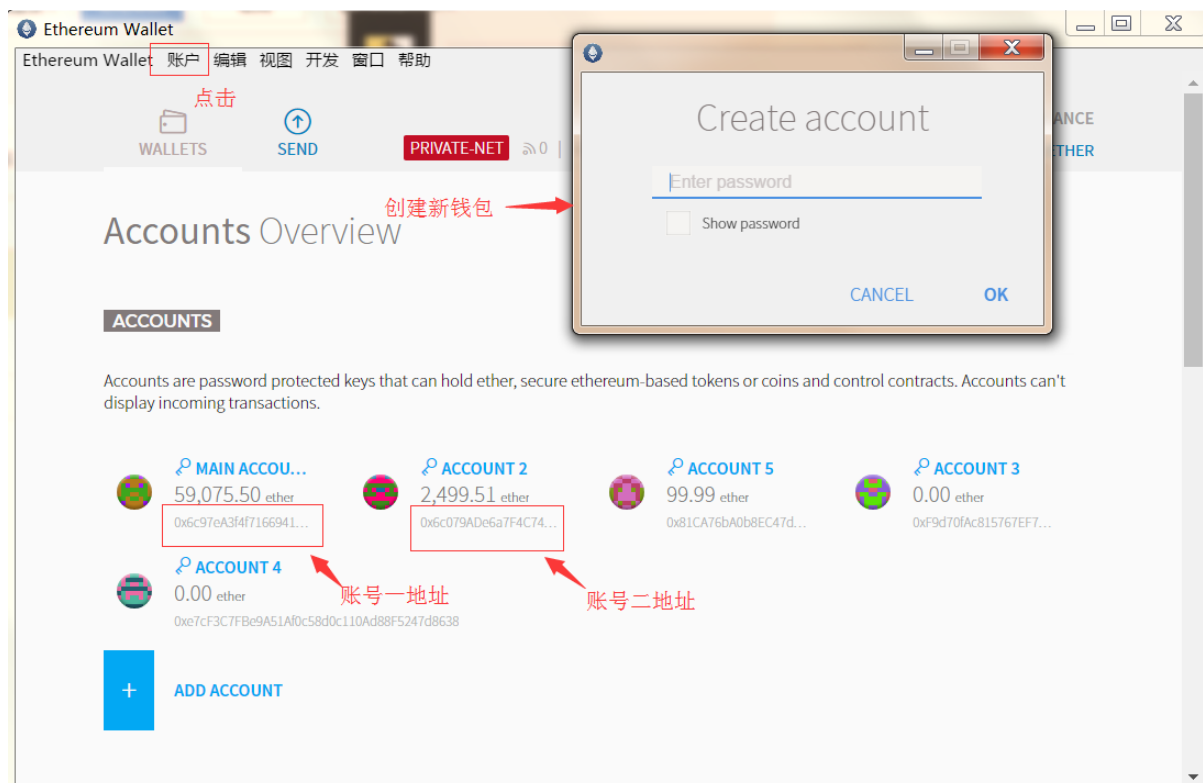


图 3.2.4

[3] Eclipse 部署 Tomcat8.5

Window->Preferences->Server->Runtime Environments->Add, 选择 Tomcat 版本和安装目录。

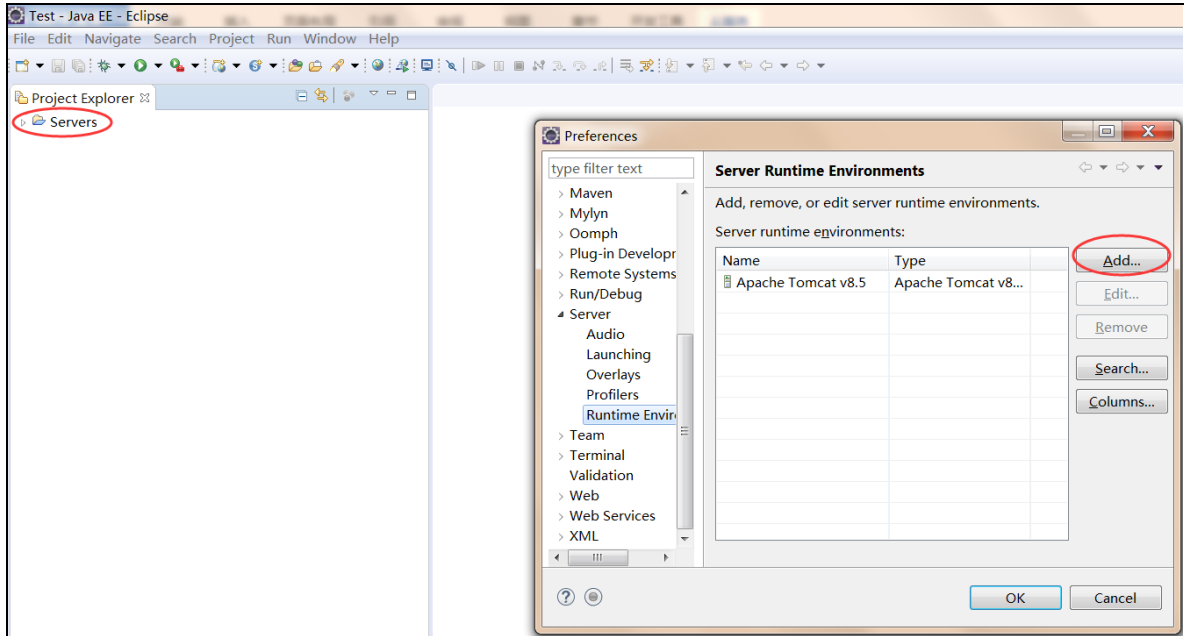


图 3.2.5

[4] Eclipse 导入系统工程

- 1、打开源码包，CrowdChain 为我们的工程文件夹。
- 2、进入 eclipse 将源码包以 maven 工程的形式导入，具体步骤略。

[5] 导入数据库

本系统先基于本地数据库进行数据存储，后续将迁移将数据保存在云端，因此需要参考如下方式导入数据库：

- 1、打开源码包，将 mysql 文件夹下的 CrowdChain.sql 文件导入 MySQL 数据库，内含一个数据库 CrowdChain、一张数据表 tb_file，用于保存任务附件和答案附件。
- 2、打开工程下的 mybatis-config.xml，配置数据库具体信息。

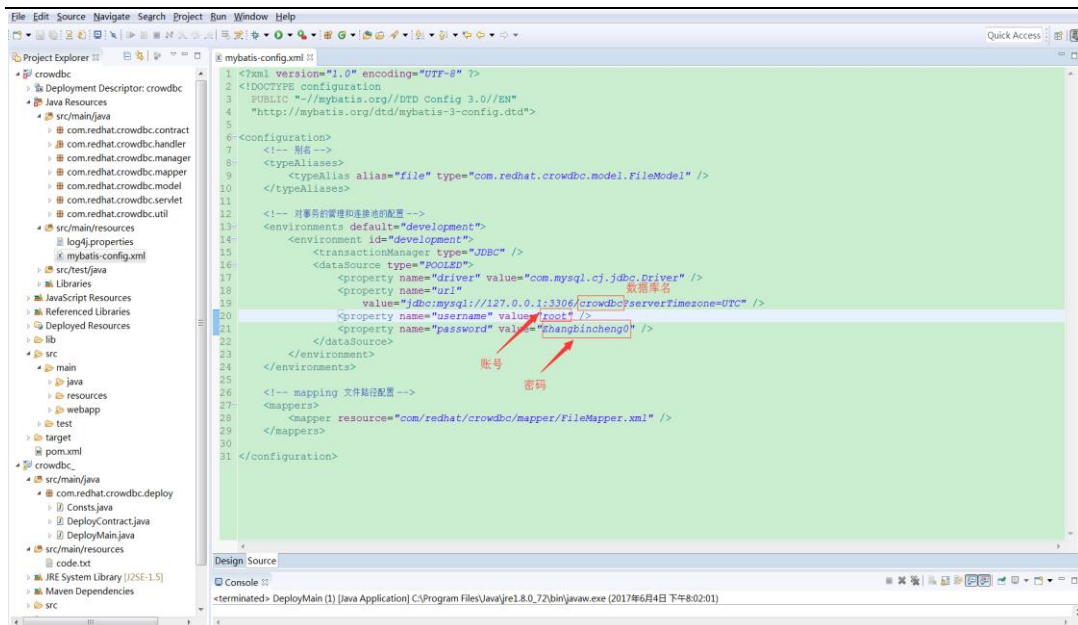


图 3.2.6

[6] 部署 Register 合约

1、首先启动 geth 客户端（方法同上），在 geth 命令行输入 miner.start(1)进行挖矿。

2、运行 CrowdChain_下的 DeployMain.java，运行过程将会读取 resources 下的 code.txt，其内容为 Register 合约编译后的二进制代码。

3、将运行输出的 contractAddr 保存下来，并传值给 CrowdChain 工程 util 包下 Consts.java 的 REGISTER_ADDR，同文件下的 RWRC_BINARY 则是 RWRC 合约编译后的二进制代码，具体如下图所示：

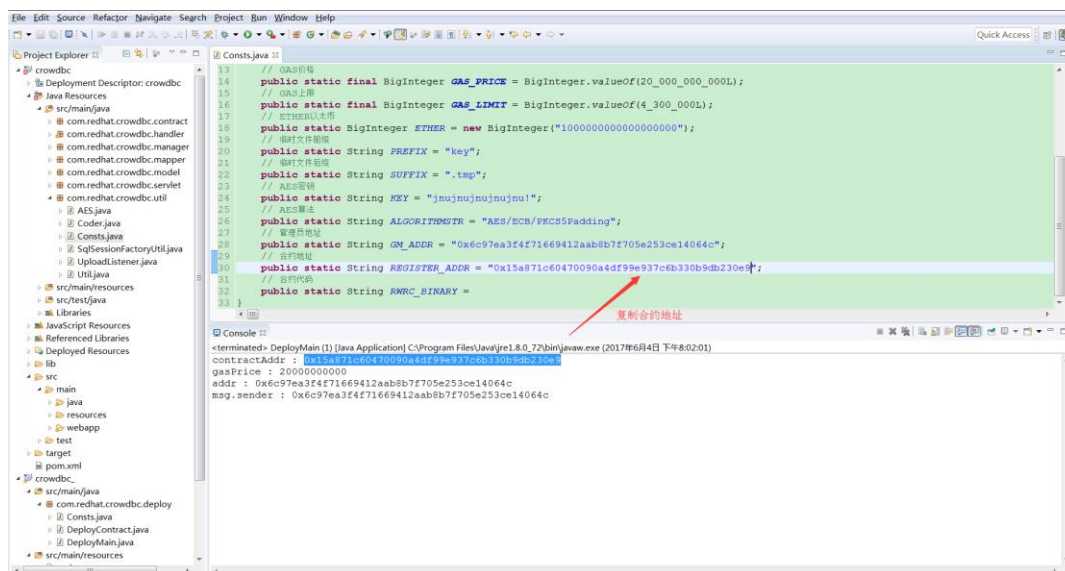


图 3.2.7

[7] 启动 CrowdChain Web 工程

本地测试 URL: <http://localhost:8080/CrowdChain/index.html>

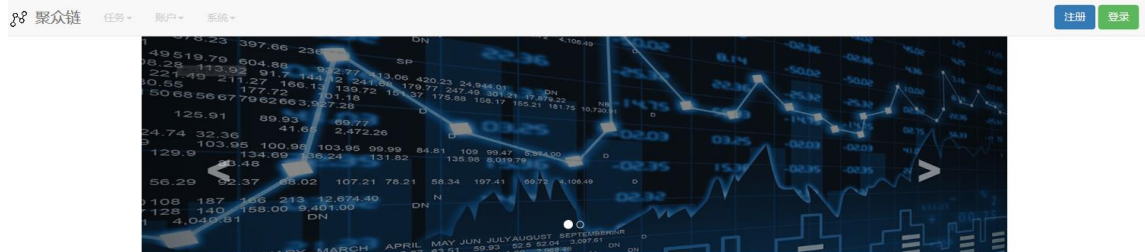
3.3 测试用例设计

序号	测试场景	功能要求
1	进入主界面	以秒级单位进入主页面
2	查看导航条	查看导航条中包含的内容
3	注册账号	实现个人账号的注册
4	登录账号	通过登录界面成功登陆账号
5	查看个人账号信息	通过弹出的账号信息界面查看账号的相关信息
6	修改密码	进入密码修改界面并进行密码修改
7	通过标签检索任务	进入任务平台,通过类型标签选择感兴趣的 任务
8	接受任务	进入任务平台,选择一个任务查看其详情后 选择接受该任务
9	查看已接受任务	进入任务管理页面,并能查看到已接受的任 务
10	提交任务解决方案	在管理任务界面中能够成功以附件形式提 交一个解决方案
11	发布任务	进入任务发布界面,并能够以附件形式发布 一个任务
12	查看已发布任务	进入任务管理列表点击发布任务查询,能够 查看到已发布任务
13	查看已发布任务的进度	进入任务管理界面并选择一个任务,查看其 完成进度
14	查看解决方案详情	查看一个解决方案的内容
15	解决方案评分	进入解决方案评分界面,并能够为一个方案 进行评分
16	账户信息更新	完成工作后个人账户中的信息能够及时更 新
17	个人简介更新	能够在个人信息面板中修改个人简介
18	账号登出	能成功登出账号

具体测试:

1.系统启动测试:

场景名称	进入主界面
测试目的	测试主界面功能是否正常
基本要求	在秒级单位内能够进入主界面并显示所有内容
用例描述	运行系统并进入主界面
测试情况	通过



聚众链

基于区块链的下一代安全众包系统

传统的众包平台中雇主（任务发布者）与工作者（任务接收者）几乎都需要依赖可信的第三方来处理任务派发、交易及任务评估等工作，虽然大多数情况下这类系统都运作良好，但是这类系统与传统的金融机构一样受制于“基于信用的模式”，而且中心化服务器会产生单点故障、隐私泄露等问题，而且中间会产生高昂的手续费。

本系统将众包领域中任务的发布、接收、评估等过程以智能合约的形式写入区块链，雇主与工作者无需可信第三方中介结构自动完成交易，而且区块链的数据无需依赖中心数据库，不存在单点故障问题，并且数据具备不可篡改性及可追溯性。此外，利用加密方式有效保护用户隐私及数据。本系统不仅可以有效解决在不可信第三方机构情况下的任务众包问题，并可以有效减少用户的交易成本。



核心



去中心化

系统基于以太坊平台运行，以太坊是分布式系统，单个节点故障不影响系统运行，同时，本系统客户端运行也无需依赖远端服务器，本地搭建Tomcat服务即可运行



智能合约


智能合约约定交易双方（雇主和工作者）的权利和义务，且一旦写入交易中的约定不可篡改，用户可以追溯合约中的所有交易中的行为，并获得相应的回报




信誉度系统

每个工作者分配一个信誉值，信誉的大小与工作者的完成表现相关，越大信誉值工作更有可能接收到工作


开始使用




注册



发布任务



接收任务



提交任务

Copyright © 2017

图 3.3.1 进入主界面测试

2.查看导航条

场景名称	查看导航条
测试目的	查看导航条内容

基本要求	点击导航栏目，显示导航条所包含内容
用例描述	点击导航栏目，查看导航条包含的内容
测试情况	通过



图 3.3.2 进入主界面测试

3.注册功能测试

场景名称	个人账号注册
测试目的	测试能否成功注册账号
基本要求	能够在秒级单位内完成账号注册
用例描述	根据要求随机输入一组账号密码进行注册
测试情况	通过

注册账户

用户名

test111

必填

密码

.....

必填

确认密码

.....

必填

钱包地址

0x6c97eA3f4f71669412Aab8b7f705E253Ce14064C

必填

简介

大学狗程序猿

取消

重置

注册

注册中

正在注册，请稍候...

若长时间没有反应，请稍后重试

图 3.3.3 注册功能测试

4.登录功能测试

场景名称	账号登录
测试目的	测试已注册的账号能否正常登录
基本要求	在秒级单位内能够成功登录账号
用例描述	进入登录界面并输入一个已注册的账号及其密码，并进行登录
测试情况	通过

登录

用户名

test111

密码

.....

取消

登录

登录中

登录成功！

现正跳往主页...

登录中

登录失败！

用户名与密码不匹配或服务器异常，请稍后重试...

图 3.3.4 登录功能测试

5.查看账号信息

场景名称	查看个人账号信息
测试目的	测试账号信息查询功能是否正常

基本要求	在秒级单位内能够显示个人信息列表
用例描述	点击主界面右上角用户名，选择查看账号信息并显示出账号信息界面
测试情况	通过

账户信息

×

用户名

test111

钱包地址

0x6c97ea3f4f71669412aab8b7f705e253ce14064c

个人简介

大学狗程序猿

注册时间

2017/6/4 下午8:10:01

已接工作数量

0

已完成工作数量

0

信誉值

60

关闭

图 3.3.5 查看账号信息测试

6.修改密码测试

场景名称	密码修改
测试目的	测试修改密码功能是否正常
基本要求	能够成功修改密码
用例描述	进入密码修改界面进行密码修改
测试情况	通过



图 3.3.6 修改密码测试

7.通过标签检索任务

场景名称	通过标签检索任务
测试目的	测试能否通过不同的任务标签来获得不同类型的任务
基本要求	能够通过标签来给任务分类
用例描述	进入任务平台，点击“类型”标签，查看不同类型的任务
测试情况	通过



图 3.3.7 通过标签检索任务

8.从任务平台上接受任务

场景名称	接收任务
测试目的	测试能否成功接收任务
基本要求	能在秒级时间内成功接收任务

用例描述	进入任务平台，查看一个任务并点击”接受”
测试情况	通过



请上传您的私钥以及输入您的密码

任务地址

0x3a426eea8a7f8926b12bdcd79e54c5e559147312

上传文件

浏览... UTC--2017-05-03T17-49-17.159825700Z--6c079ade6a7f4c74eebd85c272d2b3930

请上传您的私钥.

密码

.....

关闭 确认

图 3.3.8 接受任务测试

9.查看任务列表

场景名称	查看已接受任务列表
测试目的	测试任务是否已成功接受并显示在任务管理界面中
基本要求	任务管理列表中已成功显示刚接受的任务
用例描述	进入任务管理界面查看已接受的任务。
测试情况	通过

聚众链 任务 账户 系统 欢迎！test222

发布的任务 接收的任务

搜索

截止日期	押金	描述	最大工作者数量	最低信用限制	当前工作者数量	奖励金额	任务地址	任务状态	任务类型	操作
2017/06/06 00:00	100	详情	2	60	1	1000	0x3a426eea8a7f8926b12bdcd79e54c5e559147312	未领取完	品牌设计	提交方案

当前页面共有1个任务

前一页 1 2 3 4 5 后一页

图 3.3.9 查看已接受任务列表

10.提交解决方案测试

场景名称	提交任务解决方案
测试目的	测试任务方案提交中的 reset 和 submit 功能是否正常
基本要求	能够将解决方案通过网页成功上传到后台

用例描述	进入提交方案界面，在 solution 中输入相关内容，添加附件并点击提交
测试情况	通过

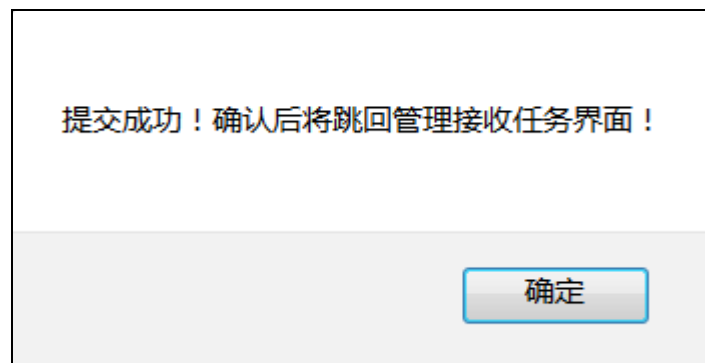


图 3.3.10 提交方案测试

11.发布任务测试

场景名称	发布任务
测试目的	测试任务发布界面能否正常工作以及任务发布功能能否正常运行
基本要求	发布任务的所以内容能成功上传到后台，并在“可接受任务列表”与“已发布任务列表”中显示出来
用例描述	打开任务发布界面，填写一个任务进行发布，并上传私钥
测试情况	通过



发布任务

品牌设计

任务描述

海报设计：将压缩包里的几张图片合成！

上传附件

浏览... qingyao.zip

请上传您的附件。

押金 ¥

100

奖励金额 ¥

1000

任务截止日期

2017-06-06 00:00

所需工作者数量

2

工作者所需最低信誉值

60

上传私钥

浏览... UTC-2017-05-03T17:48-46-721084800Z-6c97ea3f471669412aab8b7f705e253ce14064c

请上传您的私钥。

验证码

重置

提交



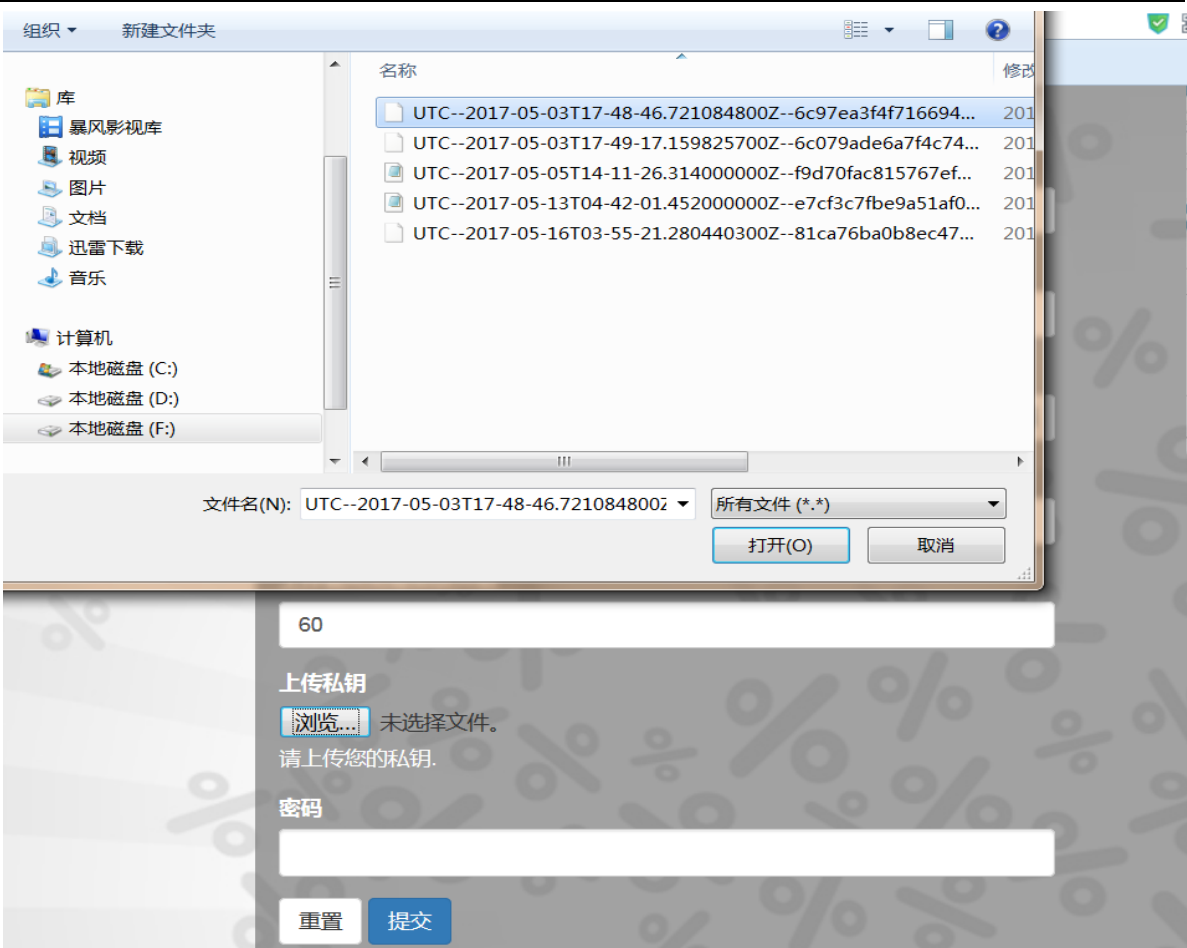


图 3.3.11 任务发布测试

12.查看已发布任务

场景名称	查看已发布任务
------	---------

测试目的	测试发布系统是否能正常显示在任务列表中
基本要求	发布的任务能在可接受任务中搜索得到，并且能在自己的任务发布列表中进行查看。
用例描述	打开任务管理界面，查看已发布任务中是否有显示刚发布的任务，再进入任务平台，查看该任务是否在平台上成功发布
测试情况	通过

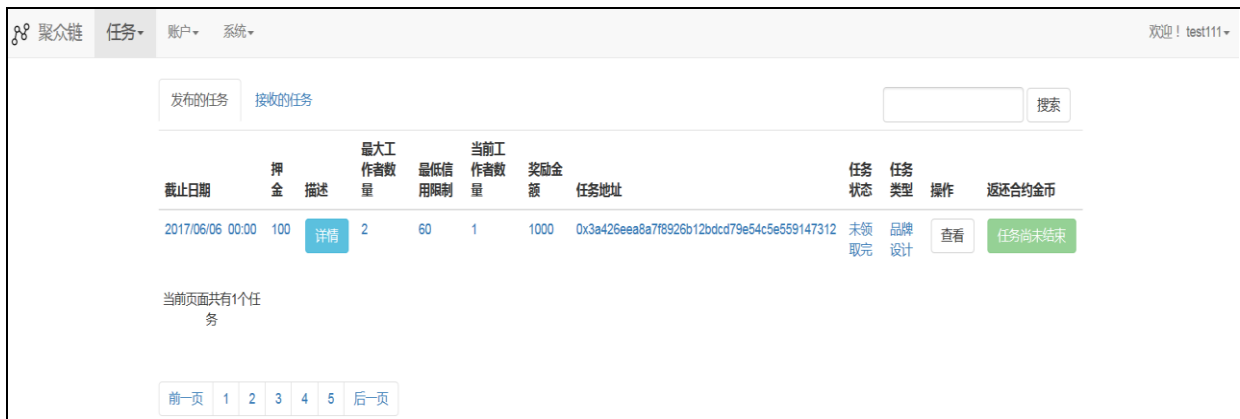


图 3.3.12 查看已发布任务列表

13.查看已发布任务进度

场景名称	查看已发布任务进度
测试目的	测试查看任务进度的功能是否正常运行
基本要求	点击“查看”按键后能显示任务完成的情况
用例描述	打开任务管理界面，点击已发布任务中的查看按钮并显示出任务完成情况
测试情况	通过

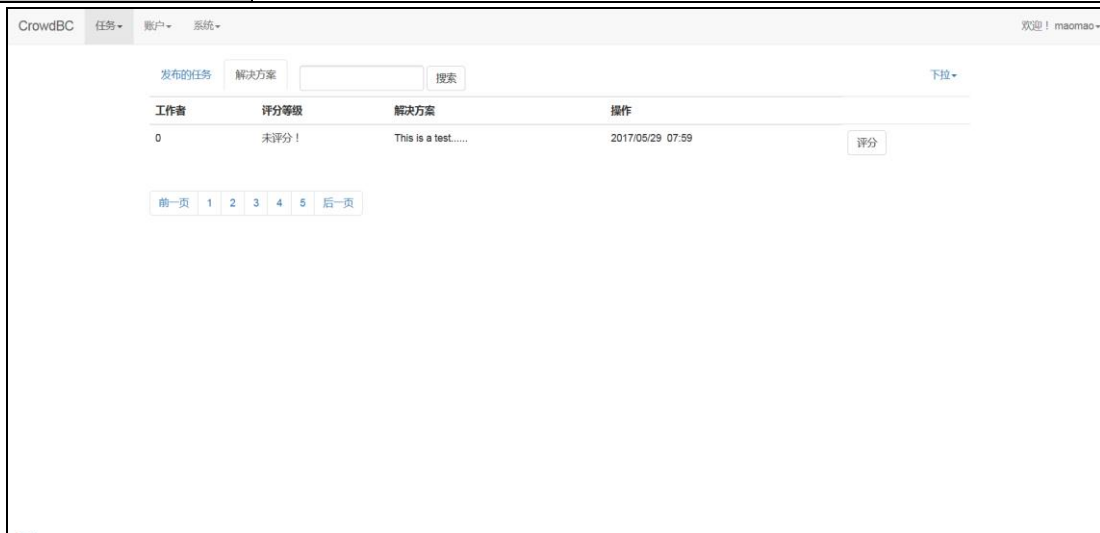


图 3.3.13 任务进度查询测试

14.查看解决方案详情

场景名称	查看解决方案详情
测试目的	测试查看已提交的解决方案的内容
基本要求	能够查看某个任务收到的解决方案的具体内容
用例描述	查看一个任务的解决方案，点击“详情”按键，显示解决方案的详细内容
测试情况	通过

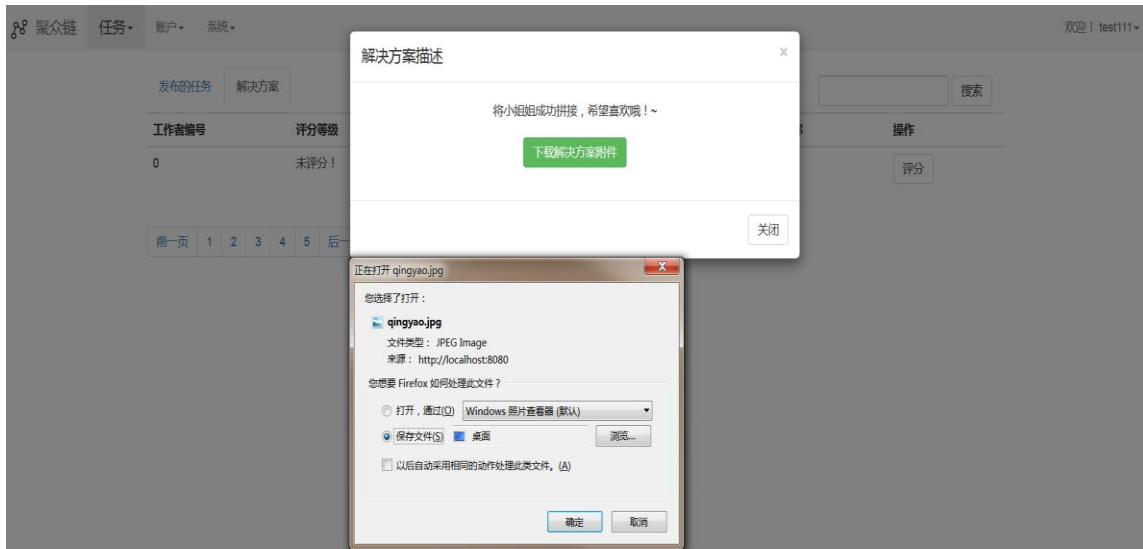


图 3.3.14 查看解决方案内容

15.解决方案评分测试

场景名称	解决方案评分测试
测试目的	测试给已提交的方案进行评分的功能是否正常运行
基本要求	能够为已提交的解决方案进行评估
用例描述	在解决方案界面中，点击评分按钮进行评估
测试情况	通过



图 3.3.15(1) 解决方案评分前



图 3.3.15(2) 解决方案评分后

16 帐户信息更新

场景名称	帐户信息更新
测试目的	测试当完成任务账号信誉值与完成任务数是否正常更新
基本要求	完成任务后账号信誉值与完成任务数会相应更新
用例描述	对比任务完成前后信誉值与完成任务数的值
测试情况	通过





图 3.3.16 帐户信息更新功能测试

17. 个人简介修改

场景名称	个人简介修改
测试目的	测试能否成功修改个人简介
基本要求	能够完成简介的修改并准确无误显示在个人信息面板上
用例描述	点击主界面右上角用户名，进入个人信息面板进行修改
测试情况	通过

修改简介

简介

大学狗程序猿

取消

修改



图 3.3.17 个人简介修改

18.登出功能测试

场景名称	登出
测试目的	测试能否成功登出账号
基本要求	能在秒级时间内成功登出
用例描述	点击主界面右上角用户名，点击登出按钮
测试情况	通过

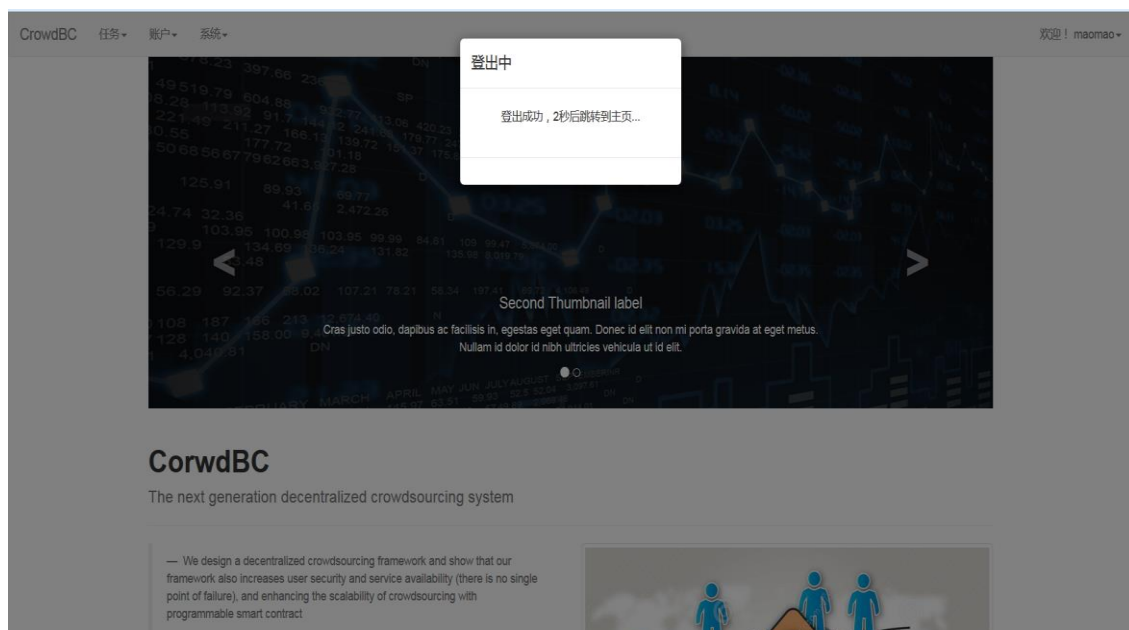


图 3.3.18 登出功能测试