****

**本 科 毕 业 论 文（设计）**

|  |  |
| --- | --- |
| 课题名称 | 基于区块链的网约车路径预警系统的设计与实现 |
| 学 院 | 计算机科学与网络工程学院 |
| 专 业 | 计算机科学 |
| 班级名称 | 计科174 |
| 学生姓名 | 李诗云 |
| 学 号 | 1706100092 |
| 指导教师 | 彭滔 |
| 完成日期 |  |

教 务 处 制

基于区块链的网约车路径预警系统的设计与实现

计科174 李诗云

指导教师:彭滔

摘要 本研究针对现有网约车平台安全事故频发的问题，旨在运用区块链技术特性，实现可靠的网约车路径预警系统。可以在杜绝人为因素影响的前提下，保证记录的真实性，并通过自动化且不可逆的预警机制，提高网约车模式的安全系数。文中的设计主要通过以太坊公链，实现不可篡改、公开透明的信息记录。Solidity智能合约自动进行核心逻辑的施行。Web3进行区块链与前后端的交互 。ReactNative结合高德地图API实现易迁移的移动端APP开发。

关键词 区块链；网约车；智能合约；

**ABSTRACT** Aiming at the frequent safety accidents of the existing ride-hailing platform, this study aims to realize a reliable ride-hailing path warning system by using the technical characteristics of block chain.The authenticity of the records can be guaranteed on the premise of eliminating the influence of human factors, and the safety factor of online car-hailing mode can be improved through automatic and irreversible early warning mechanism.The design in this paper is mainly through the Ethereum public chain to achieve tamper-proof, open and transparent information records.Solidity smart contracts automatically implement core logic.Web3 interacts with the blockchain and front and back ends.ReactNative is combined with Amap API to realize mobile terminal APP development that is easy to migrate.

**KEY WORDS** blockchian； online car-hailing；smart contract

目录

[1 前 言 2](#_Toc70929080)

[1.1 课题研究背景 2](#_Toc70929081)

[1.2 现有问题 2](#_Toc70929082)

[1.3 项目创新 3](#_Toc70929083)

[1.4 国内外研究现状 3](#_Toc70929084)

[1.4.1 网约车安全 3](#_Toc70929085)

[1.4.2 区块链技术 4](#_Toc70929086)

[1.5 文章结构 5](#_Toc70929087)

[2 相关概念和技术介绍 5](#_Toc70929089)

[2.1 区块链技术 5](#_Toc70929090)

[2.1.1 概念介绍 5](#_Toc70929091)

[2.1.2 技术特点 6](#_Toc70929092)

[2.1.3 交易原理 7](#_Toc70929093)

[2.2 智能合约 7](#_Toc70929094)

[2.2.1 概念 7](#_Toc70929095)

[2.2.2 部署原理 8](#_Toc70929096)

[2.2.3 运行原理 8](#_Toc70929097)

[2.2.4 局限性 8](#_Toc70929098)

[3 系统方案设计 9](#_Toc70929099)

[3.1 开发工具环境 9](#_Toc70929100)

[3.1.1 以太坊测试网Ropsten 9](#_Toc70929101)

[3.1.2 Solidity 9](#_Toc70929102)

[3.1.3 Remix-ide 9](#_Toc70929103)

[3.1.4 Web3JS 9](#_Toc70929104)

[3.1.5 MetaMask 10](#_Toc70929105)

[3.1.6 ReactNative 10](#_Toc70929106)

[3.1.7 高德地图API 10](#_Toc70929107)

[3.2 移动端APP设计 11](#_Toc70929108)

[3.2.1 地图展示模块 11](#_Toc70929109)

[3.2.2 路径规划模块 12](#_Toc70929111)

[3.2.3 交易打包和事件监听模块 14](#_Toc70929112)

[3.3 区块链和智能合约模块设计 14](#_Toc70929113)

[3.3.1 行程信息记录模块 14](#_Toc70929114)

[3.3.2 紧急联系人设置模块 16](#_Toc70929115)

[3.4 路径脱离判断模块 16](#_Toc70929116)

[3.4.1 计算辅助模块 17](#_Toc70929117)

[3.4.2 脱离路径事件模块 21](#_Toc70929122)

[4 系统实现与测试 23](#_Toc70929123)

[4.1 系统实现 23](#_Toc70929124)

[4.2 系统测试 23](#_Toc70929125)

[4.2.1 测试环境配置 23](#_Toc70929126)

[4.2.2 测试方案 23](#_Toc70929127)

[4.2.3 操作流程 23](#_Toc70929128)

[4.2.4 功能测试 23](#_Toc70929129)

[5 结语 23](#_Toc70929130)

[6 参考文献 23](#_Toc70929131)

[7 致谢 23](#_Toc70929132)

基于区块链的网约车路径预警系统的设计与实现

* 1. 前 言
     1. 课题研究背景

近年来，城市化推进、基础交通设施的完善有利于孕育大量的出行需求；互联网技术的飞速发展也为出行行业的发展奠定基础。传统出行模式下，用户原地等待，司机沿途接客的出行方案，已经难以适应市场需求的发展。

在此背景下，网约车服务应运而生，并飞速发展，短短数年内已经完成了对传统出租车模式的反超。根据中国互联网络信息中心(CNNIC)的统计， 截至2020年6月，我国网约车用户规模为3.40亿。受疫情影响，较2019年中减少6415万。而在疫情前，网约车用户规模呈持续上升趋势，2019年6月网约车用户规模则一度达到4.04亿人，较2018年中增长约1600万人。其捷高效的服务模式已经在很大程度上改变了人们的出行习惯。

现代网约车平台同时集合了用户叫车，分配派单，行程结算，订单管理，支付管理，信用管理，综合运营等多种功能于一体。一站式为司机和用户解决营业和出行问题。较传统打车模式，有效匹配了司机和用户的需求，减少了司机空载率和乘客的等待时间，最大化的节省了双方的时间和沟通成本。成为了人们打车出行的首选模式。

* + 1. 现有问题

然而近年来多次发生的网约车抢劫杀人案、侵犯案、失踪案，动机卑劣，性质恶劣社会危害极大，引起了广泛的重视，将网约车推向了舆论的风口浪尖，对网约车业务发展造成了严重的打击。在此背景下保障网约车安全，完善相关法律，建设合理的监管体系的呼声越发响亮。

刑事案件外，平台管理方面也频频传出负面消息。2019年失信司机、报废车辆通过后门重新营业，却拒绝接受有合法营运资质的网约车公司接入事件。运营部门操作后台直接修改司机信用评分等被曝光出的事件，也在一次又一次地加剧网约车平台与用户之间的信任危机。如何平台数据的真实性，如何确保用户的知情权，如何建立良好的信任体系，也已成为了亟待网约车平台解决的问题。

* + 1. 项目创新

本文将立体式的运用区块链技术的优势特性，针对应用场景设计对应模式，以满足网约车模式中存在的安全需求。

区块链公链的特性完美符合网约车行程中对于订单坐标记录的需求。其具有任何人都可参与发送交易、确认交易、参与共识过程的架构；具有公开透明，去中心化，去信任成本的优势。采用公有链记录社会公示所需的重要信息，如司机信息，车辆情况，行程坐标等记录，在密码学上保证了其无法篡改，能够很好实现无需第三方机构信用背书的社会监督，同时做到减少成本及提高公信力。

同时创新性的辅以智能合约进行行程路径是否脱离的判断，较传统后台，智能合约部署在区块链中，拥有内容公开透明，不可篡改，一旦部署永久运行的特性。脱离了人为影响的判断机制，杜绝了平台暗箱操作的可能，保证了记录的真实性和判断的客观性，运用该种技术能够使网约车平台的具有相当的社会公信力。同时智能合约的内容由区块链执行，较传统后台，能够节省大量的服务端资源。

结合Web3监听智能合约中触发的路径脱离事件，第一时间对可能存在的威胁做出反应，自动触发应急机制。将区块链与APP立体结合，在模式设计的层面上，提供了一种有效的乘客安全保障方案。

* + 1. 国内外研究现状
       1. 网约车安全

对于网约车安全的探索也在一直进行中，现有的研究主要给出了两个方向的解决方案。

一是完善相关法律，通过国家立法，规范化网约车的经营管理流程。夯实平台责任提升合规率；同时完善监管平台，实现与网约车平台信息共享，进一步提高监管能力和水平，通过国家监管对平台进行督促和补充。

二是加速信用平台建设，根据《重庆市网络预约出租汽车经营服务管理暂行办法》第四十条规定：各有关部门应当按照职责建立网约车平台公司和驾驶员信用记录，并纳入全国信用信息共享平台。同时将网约车平台公司行政许可和行政处罚等信用信息在全国企业信用信息公示系统上予以公示。通过与信用挂钩，降低问题人士进入网约车行业的概率，同时大幅提高违规成本，以达到保障用户安全的目的。

而本文设计的基于区块链的网约车路径预警系统，将在技术层面为满足网约车模式中存在的安全需求提供新的思路。

* + - 1. 区块链技术

国外对区块链的探索大多来自于实务界的不断探索，强调数据编程应以法律为基础，这有利于区块链的智能合约与实际生活相结合，提供更好的服务。国外的许多企业开始认识到，区块链技术的兴起对网约车行业可能造成巨大冲击，开始研究区块链在网约车领域的应用，纷纷发布了区块链战略书。

2019年网约车领域的领头企业Uber，寄希望于使用区块链技术解决目前网约车行业缺乏安全标准，缺乏透明度，平台维护费用高昂的问题。通过Uber区块链提高司机与客户的联系效率，并使用区块链进行身份管理，同时解决目前存在的效率问题与信用问题。

早在 2016年区块链就获得国内各行各业的关注，在网约车领域对于区块链技术应用也在不断进行探索。快的打车创始人陈伟星于2018年致力于打造以VV Share为核心的共享经济体，以区块链网约车应用VV Go打开相关市场，公布了详细的白皮书和模式设计。希望能形成囊括衣食住行的区块链生态。

杭州趣链科技有限公司在2017年申请了一项《一种基于区块链技术的智能网约车系统及网约方法》专利，该专利试图通过以一种基于区块链技术的智能网约车系统及网约方法，来打破目前中心化服务网约车平台的垄断。其中支付模块，评价模块，行程记录模块的设计与区块链密切结合，费用的计算逻辑，评价逻辑将通过智能合约实现直接部署在区块链上。

但遗憾的是，目前还没有完整的区块链网约车项目落地。

* + 1. 文章结构

第一章主要介绍论文的研究背景和国内外研究现状，并介绍了论文的主要工作内容和创新点。

第二章介绍了系统设计和实现中涉及的主要技术，包括区块链技术介绍和智能合约技术的介绍，对二者的技术特点和基本原理进行了分析。

第三章中介绍了系统设计和实现中将要使用到的具体环境，框架和工具。确定系统中功能的功能需求，完成功能模块的划分，描述了各个功能模块的实现思路。

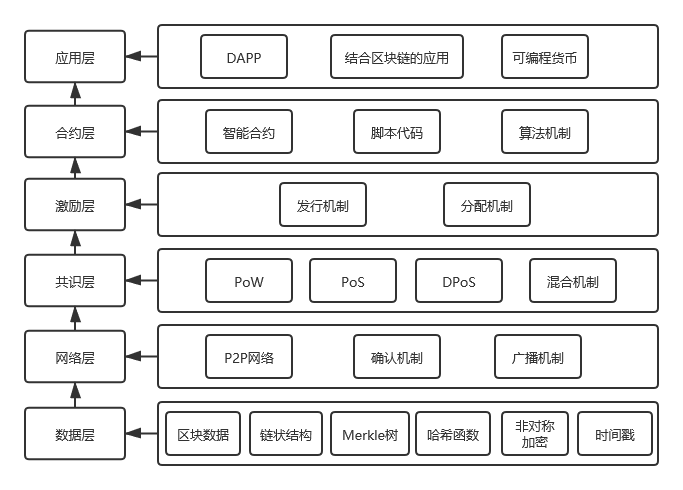
第四章中介绍了各个模块具体实现的关键部分，并对系统进行功能测试。

第五章对本文工作进行总结，并提出项目未来的工作方向。

1. 1. 相关概念和技术介绍
      1. 区块链技术
         1. 概念介绍

区块链技术最早在2008年由中本聪团队以“区块”和“链”的概念提出，其本质是一个由节点共同维护的分布式账本。区块链技术为在一个分布式环境下保证数据透明度和可靠性提供了解决方案。

目前达成共识区块链的基础模型，自下而上分为六层，分别是数据层、网络层、共识层、激励层、合约层和应用层。数据层主要封装数据区块和哈希函数、时间戳等技术，保证区块链的基本结构，网络层主要包含分布式网络体系、数据广播机制和验证机制，共识层则是分布式网络中维持对数据的共识的各类机制，激励层主要包含经济激励的发行和分配机制，合约层主要包括智能合约、脚本代码和算法，应用层则是区块链技术的各类应用场景。



区块链分层架构模型

区块链的数据结构是一种链式结构，由记录节点生成的符合规则的区块以类似链表的形式组成。区块由区块头和区块体组成，区块体包含交易信息和附加信息；区块头的生成规则较为复杂，对整个区块链系统的可靠性起着决定性作用。其必须通过上一个区块的hash值，再由哈希函数和区块链所使用的共识机制附加码构成。通过这样的链式生成规则，即可在密码学上保证其不可篡改，不可伪造的特点。

* + - 1. 技术特点

一、去中心化。区块链网络无需第三方管理机构或硬件设施进行管理和维护，所有节点都享有同等的权利，各个节点实现了信息自我验证、传递和管理。去中心化的特点使得人为因素对平台的影响被消除。同时平台的维护成本也因所有节点得以共同参与验证和维护工作，而得到了降低。

二、保障记录真实不可篡改。区块链技术通过加密算法、共识机制、时间戳等，保证每个区块的唯一性与可回溯性。每个区块都保留了上一区块的哈希值，并且每块的区块数据内容又与其哈希值唯一对应，这种遗传性使得诚实链上的区块相互关联，确保信息不可篡改。

三、公开透明。区块链使用去中心化的分布式账本，通过网络层中的信息广播机制，当交易发生时，将交易信息加密后向所有节点广播，所有节点均对交易进行记录，获取链上完整的交易信息。同时，后加入的节点也可通过区块链网络同步所有已存在的区块。该机制保障了区块链的公开透明性。

* + - 1. 交易原理

在区块链中，用户与区块链间的交互都是以交易的形式进行的。一个合法的交易需要包括以下的信息和流程。1.交易的序号nonce，写入发起交易的地址from、交易指向的地址to、交易的货币数额value；2.使用发起交易地址的私钥对交易信息进行签名；3.系统对交易信息进行验证，把这笔交易入到本地的txpool中（就是缓存交易池）；4.把交易信息广播给其它节点。5.经过其他节点确认后，交易生效，返回记录交易的地址。交易一旦经过确认则永久生效不可篡改，同时任何节点都可以同步交易的信息。在以太坊中，为了支付打包节点的矿工费（gas），还需要附加以下信息：愿意支付的gas单价Price，允许交易使用的最大gas值GasLimit。附加了这两项信息，交易池中的交易才会被矿工进行打包。矿工会更倾向于打包Price高的交易以获取更高的矿工费。为了防止Price低的交易滞留在交易池中长时间得不到打包，以太坊中引入了优先级的概念：每个区块生成时，滞留在交易池的交易优先级会得到提高，当交易的优先级达到一定值时，必须先打包该交易，否则交易池中其他交易的打包将无法进行。

* + 1. 智能合约
       1. 概念

智能合约的定义是：一套以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。本文的中智能合约主要指以太坊中的智能合约。

以太坊中智能合约是一段依托于区块链网络运行的代码和数据的集合，即区块链分层架构模型中合约层负责处理的部分。以太坊的智能合约与早期区块链脚本的主要区别是其具有图灵完备，可以让我们像使用任何高级语言一样来编写几乎可以做任何事情的程序。提前制定规则并部署到网络后，即可在对应条件发生时自动执行所制定的规则。

* + - 1. 部署原理

部署智能合约到以太坊上时，以太坊客户端首先会调用SOLC智能合约编译器将合约编译成EVM字节码，之后通过RPC接口将字节码文件发送到以太坊网络进行广播，超过一半以上节点验证通过之后，编译后的字节码会被写入到区块链账本中，之后会通过一笔交易在区块链网络中创建出一个智能合约。该笔交易包含几个特殊信息，分别是创建者地址、智能合约内容、智能合约的部署地址。智能合约部署地址是由创建者地址和发送的交易数作为输入，通过Kecca-256加密算法重新计算生成的地址。

* + - 1. 运行原理

智能合约的调用需要两个关键值，用于描述智能合约的ABI或字节码和智能合约具体部署到地址，ABI或字节码可以在编译器编译智能合约通过后得到。调用智能合约的本质就是发送一笔到智能合约部署地址的交易。相较于上文提到的区块链中交易的发送，智能合约的调用将交易指向的地址to，改为了需调用智能合约的部署地址to；交易的货币金额value，改为了试图调用的写方法的字节码。其余的输入值与发送交易所需的一致，打包与确认的流程也一致。

由于不存在中心节点，所以智能合约调用时，并不是由区块链直接执行。而是由试图记账的以太坊节点从区块链中取出合约代码，然后由本地的以太坊虚拟机（EVM）进行执行，将返回结果写入本次调用产生的区块中，完成整个调用的过程。之后进行广播，由整个网络的节点对运算结果相互认证，以保证智能合约执行的正确性。

* + - 1. 局限性

智能合约目前还不支持浮点型的使用。这是由于智能合约将运行在多种设备上，而不同设备对于浮点数有不同的定义，考虑到浮点数容易引起各个设备上的截断误差不一致而导致的验证失败，为了维护公共账本一致性，目前智能合约目前还不支持浮点数。

本文期望使用智能合约进行可追溯的路径脱离判断，而判断不可避免的要进行较为复杂数学运算。在不支持浮点数的条件下，如何实现该算法，如何保障计算的精度，就成为了本项目的一大挑战。

* 1. 系统方案设计
     1. 开发工具环境
        1. 以太坊测试网Ropsten

本文主要在以太坊上进行区块链模块的开发。开发阶段将会使用以太坊测试网络Ropsten代替主网络(Mainnet)进行的交易和部署。

以太坊测试网（testnet）用于模拟以太网主网的行为，拥有几乎和以太坊主网等效的功能。在Ropsten测试网络中，用户可以在拥有小于四个eth时申请免费eth用于模拟交易和支付交易费用(gas)。因此可以在testnets上开发和测试自己的智能合约，测试发币等等，作为上链前的评估环境。测试网络上的代币只能在测试环境中使用。

* + - 1. Solidity

Solidity是以太坊官方推荐语言，也是目前最流行、版本最稳定的智能合约语言。Solidity运行在Ethereum虚拟机（EVM）之上，是一个图灵完备的成熟语言。用这种语言可以创建合约来编码任意状态转换功能，可以用简单的几行代码来实现系统逻辑。

* + - 1. Remix-ide

Remix-ide是一款在线ide，主要用于solidity和yul语言的在线编译，调试和部署。支持目前发布的所有solidity版本，同时提供虚拟机环境，web3注入环境和第三方web3Provider环境的多种部署功能。支持如单步调试在内的丰富插件。已经成为solidity在线开发的首选。

* + - 1. Web3JS

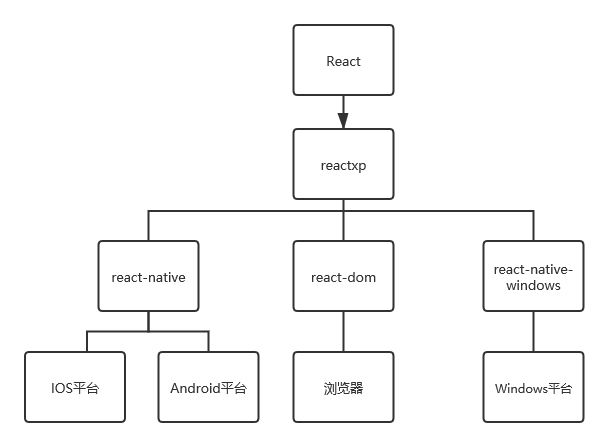
Web3JS是一个轻量级的JavaScript库，用于连接以太坊网络的客户端。当Solidity合约编译好并且发送到网络上之后，可以使用以太坊的Web3API来调用它，使平台节点能与区块链网络节点进行交互，底层实现上，它通过RPC 调用与本地节点通信。Web3JS可以与任何暴露了RPC接口的以太坊节点连接。

* + - 1. MetaMask

MetaMask是一个浏览器插线形式的轻量级数字货币钱包。对以太坊有很好的支持，可以一键链接到以太坊主网络和Ropsten、kovan、Rinkeby等测试网络，同时支持自定义网络和自定义RPC。相较于其他钱包，MetaMask的优势在于轻量和提供友好的图形界面，基础功能齐全且封装程度高。

* + - 1. ReactNative

ReactNative是FaceBook发布的一个JS库，使用户只使用JavaScript也能编写原生移动应用，提供了一种便利的多平台开发方式。它在设计原理上和React一致，通过声明式的组件机制来搭建丰富多彩的用户界面。区别仅仅在于React将虚拟DOM映射为浏览器的DOM树来实现网页的构建，而ReactNative将虚拟DOM映射为原生组件来实现高性能的原生应用。



React跨平台示意图

* + - 1. 高德地图API

本文中地图展示，路径规划功能的实现，主要依靠高德地图WebAPI。高德Web服务API向开发者提供HTTP接口，开发者可通过这些接口使用各类型的地理数据服务。主要使用到了地理/逆地理编码，路径规划两个功能。

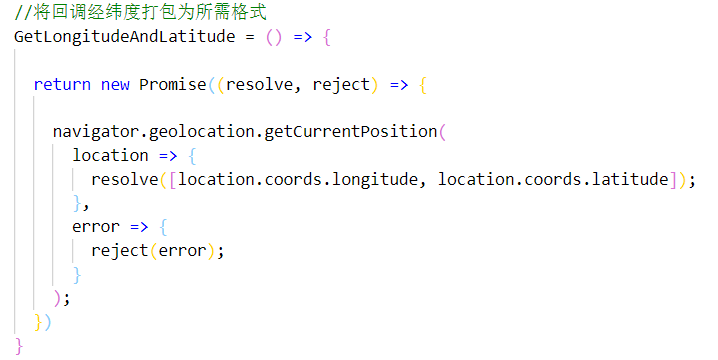
* + 1. 移动端APP设计
       1. 地图展示模块

地图展示模块设计目的在于获得用户的位置信息以及向用户提供友好的地图界面。获取位置信息的方法有两个：一是通过WebAPI提供Geolocation下的getCurrentPosition()方法；二是通过原生的GIS方法获取。

方法二由于是原生方法，所以具有略胜一筹的性能，但缺点是需要配置原生文件。安卓平台需要在工程文件中配置GIS信息，之后设置LocationManager对象，通过 (LocationManager) getSystemService(Context.LOCATION\_SERVICE)方法获取位置信息；而IOS平台需要在工程的info.plist文件中增加Privacy - Location Always and When In Use Usage Description和Privacy - Location When In Use Usage Description两个Key，导入<CoreLocation/CoreLocation.h>包后设置CLLocationManager对象来获取位置信息。

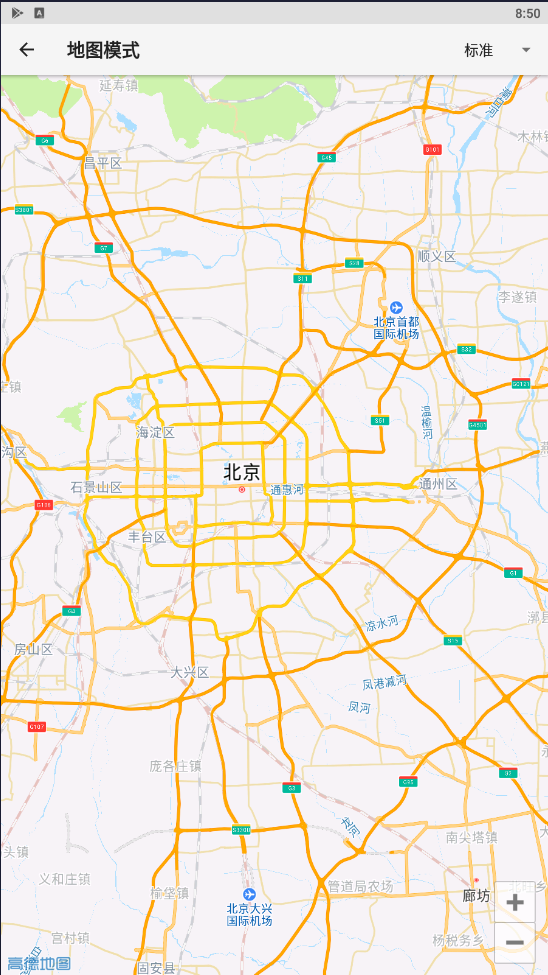
方法一通过WebAPI的封装，将自动根据所处平台采用合适的方法，无论是浏览器，还是安卓端、IOS端，都仅需通过Geolocation.getCurrentPosition() 方法直接获取位置信息。

考虑到前端APP的扩展性与兼容性，此处选用了方法一来获取位置。



封装后的getCurrentPosition方法

地图展示方面，此处使用高德提供的react-native-amap3d包来进行实现。该包提供了一个简单的地图组件和基本的拖拽，缩放，添加标记点功能。



amap3d基本图层

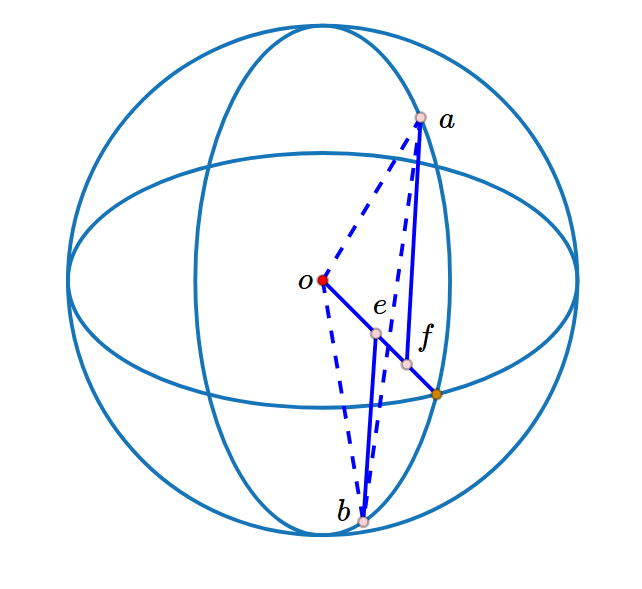
amap3d基本图层可以通过传入属性来调整显示内容，如coordinate属性调整聚焦位置，zoom属性调整缩放比例等。还可以监听地图上的点击、拖动等事件。

我们将通过地图标记（MapMaker）来标记用户和接单司机的位置，通过海量点（MultiPoint）数组标出周围所有司机的位置。

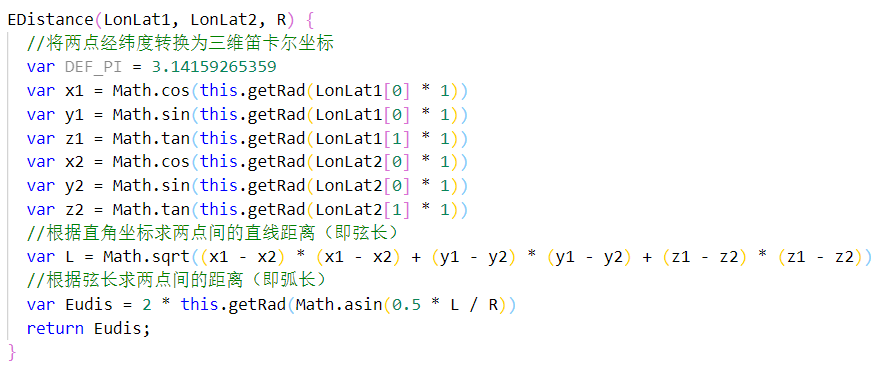
* + - 1. 路径规划模块

路径规划模块将要实现根据用户选择的起止地点，为其推荐合适的路线。这里选用高德WebAPI的路线规划功能，该方法以两个经纬度坐标为输出，以坐标数组的形式返回合适的路径。得到推荐路径后，还需要进行二次处理。通过屏幕绘制实现数据数组到地图路线的映射，以完成直观的路径引导。将数组分段，防止一次性渲染过长路线导致的卡顿。

行程过程中还需定期获取当前坐标，写入区块链记录，同时根据已行进的坐标点及时更新路径，将已经通过的路径点在地图上移除。此处就引出了一个核心问题，如何判断用户是否经过路径点。通过实际坐标与推荐点的坐标间的距离是一个行之有效的方案。经纬度坐标系近似于一个球体，两个经纬度坐标的距离不能通过平面距离公式获得。而是需要将经纬度坐标转化为三维笛卡尔坐标系，将地球近似的看作一个球体，两个经纬度坐标通过正余弦函数转化作球面上的两点，通过获得两点之间的弦长，代入通过弦长求弧长公式，所得弧长即为所求的距离。



求球面两点间的弧长示意图



封装后的求球面两点间弧长的函数

获得了坐标点间的距离之后，就可以以此进行是否经过路径点的判断了。需要注意的是行程中，推荐路线与实际位置往往存在一点的偏差，需要预留一点的判断空间。

为了保障用户的安全，我们需要在用户脱离既定路线时做出反应。真正触发紧急事件时的数据十分重要，也是本文的重点，此部分将由区块链和智能合约模块中的路径脱离判断模块完成，同时完成脱离的判断，关键数据的永久化记录和紧急事件的触发。而前端模块中，需要处理的情况是路径偏离量在可接受范围内未触发紧急事件时，提供为用户返回推荐路线推荐合适的路线。

* + - 1. 交易打包和事件监听模块

该模块主要负责使用web3js进行前端和区块链之前的交互。web3js可以在前端组件中直接使用，但是需要用户安装数字货币钱包，钱包作为web3节点提供者（Provider）即可正常进行交易的打包发送与区块链事件的监听。为了降低用户使用的门槛，让初次尝试区块链APP的用户也能使用。这里我们将建立一个简单的nodejs后台，仅负责提供默认的Web3Provider和地址以帮助没有数字货币钱包的用户使用APP，初次之外进行区块链交易的简单打包和转发，而不负责区块链上的任何逻辑，以保证该模块的纯粹性。

交易打包模块，需要内置以部署合约的地址和智能合约编译后的ABI，通过web3js建立连接到智能合约的对象。还需要收集交易必须的信息，当用户未输入时提供默认值。考虑到坐标的上传可能较为频繁，可能发生短时间内多笔交易等待发送的情况，该模块还需要设置合适的nonce值，以便于交易形成队列而不是相互覆盖。

事件监听模块用于监听区块链上触发的事件，以便于做出相应的应对。在交易打包模块建立的智能合约对象的基础上，引入WebSocket协议下的Provider以实现持久化的监听。

* + 1. 区块链和智能合约模块设计
       1. 行程信息记录模块

行程信息记录模块将通过区块链技术实现信息记录的不可篡改，公开透明性，同时为与APP模块的交互做好准备。

为了实现该目的，将在智能合约中完成以下任务:

1.构造合适的结构

contract Order{

address Manager; //用于记录行程的拥有者

uint[] Latitude; //用于记录坐标纬度

uint[] Longitude; //用于记录坐标经度

uint[] PositionTime; //用于记录坐标上传的时间

uint256 OrderCreatedTime; //订单的创建时间

uint256 OrderTime; //行程开始的时间

uint256 OrderIndentifier; //随机生成的识别码

uint16 OrderStatus; // 行程的状态

uint[2] StartingPoint; //行程的开始坐标

uint[2] Destination; //行程的目的地

}

2．实现对应的初始化，改变行程状态，记录行程信息等方法。

初始化方法将用仅部署时触发一次的construct（）实现，需要记录部署时合约的msg.sender属性，确定合约的归属者；同时生成随机的识别码，如何在区块链中生成安全的随机数是一个很复杂的问题。此处将使用合约部署时的区块难度附加此时的时间戳作为输入，通过keccak256哈希算法得到的强抗碰撞性值作为识别码。

改变行程状态方法用于明确方法的调用时机，将通过OrderStaus状态的改变，结合solidity修饰符（modifier）实现智能合约中的状态管理。

记录行程信息方法将接受两个参数，分别是纬度值和经度值，将数据打上获取时间的时间戳后永久记录到区块链中。

3．提供查询数据的接口

虽然区块链完全公开透明，任何节点都可以获得完整的信息记录，但直接通过区块链地址来获取数据较为繁琐，需要定位区块，验证区块合法性，将区块中的数据转化为需要的格式。所以在此将添加若干接口用于快速直观的获得所需的信息。

此处将提供查看完整位置记录和查看完整合约结构的两个接口。

4．定义状态改变事件并提供事件接口。

定义事件并提供事件接口，便于区块链外的模块监听发生的事件，以做出相应处理。

此处将定义状态改变事件和位置信息记录成功事件。

* + - 1. 紧急联系人设置模块

紧急联系人设置模块用于记录紧急联系人的联系方式和用户的基本信息，用于紧急事件触发时进行相应处理。该部分变动较小，为了避免每一份行程信息记录合约中都存在相同的信息，造成资源的浪费，故单独设立一个模块。

常用的联系方式都可以作为紧急联系人的联系方式，如电子邮件，电话号码，节点地址等。此处选用电子邮件和节点地址作为记录方式。

用户的基本信息主要用于向紧急联系人说明情况，此处仅记录用户姓名或昵称。

contract UserPage{

address Manager;

string UrgencyMail;

string Username;

address UrgencyAddress;

}

该模块逻辑较为简单，仅需完成记录和返回记录值即可，也无需定义事件或提供易读接口。故将实现记录信息和返回信息两个方法。

* + - 1. 路径脱离判断模块

该模块是整个设计的重点，将实现无法篡改，可追溯，满足条件自动执行的行程中路径脱离判断。主要的难点在于如何在无法使用三角函数的前提下，计算两点间的距离；如何在不支持浮点数的情况下完成计算，并保持结果的准确性。

由于路径的上传存在时间间隔，我们得到的实际位置点并不是连续的。若通过是否经过每一个路径点，即是否与每一个路径点的距离都小于过某值进行判断，会存在实际经过了某路径点，但由于上传间隔正好通过了判断范围，导致该点被视作未通过的误判发生。

故此处将通过点到直线的距离作为判断依据。即取上一个经过的路径点与下一个要通过路径点，分别作为起止点作线段，过实际位置点作该线段的垂线，该垂线的长度即为所求，以此作为判断依据。该方案大幅扩大了判定的范围，能有效减少上传时间间隔造成的误判。

* + - * 1. 计算辅助模块

智能合约中，写类型方法会造成永久化的记录并花费大量的资源，需要支付较多的gas，而读取数据、进行运算仅需执行节点和确认节点进行，需要支付的gas较少。而路径判断模块中的需要重复进行大量的数学运算，将这一部分计算的过程放在写类型方法中，将造成大量资源的浪费。故将这一部分提取而出，放入一个不进行任何写操作的智能合约中，为其他部分提供计算辅助。

为了实现路径脱离判断功能，计算辅助模块需要实现上文中提到的点到点距离的计算，点到直线距离的计算和数据平方根的计算。

在实现具体算法前，需要解决的问题是solidity目前还不支持浮点型，无论是作为输入的经纬度坐标值，还是计算的过程中，都难以避免的会出现小数。智能合约的编写中，常以有理数的形式表示小数，以完成无浮点型的运算。然而输入的经纬度坐标值和三角函数运算的结果往往都难以有理数化，需要采取其他的解决方案。

由于经纬度值是外部输入，故可以通过在输入前将小数点右移若干位的方式将小数化为整型，此后在计算和返回值时进行额外处理，将小数点回移，以保障计算的正确性。

而内部三角函数和其他计算导致的小数，就不得不进行截断操作了。因此，为了防止多次数据截断造成的精度损失，我们需要尽可能的增大输入的有效位数以及减少会造成小数的算法的使用。

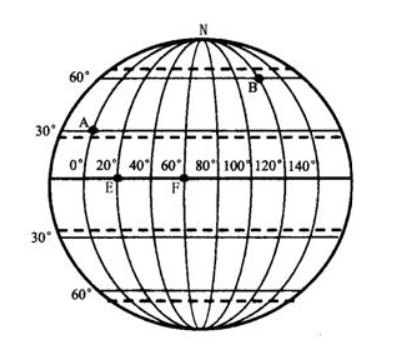
* + - * 1. 点到点距离的计算

该函数负责完成经纬度坐标点间距离的计算。

本文在移动端APP的路径规划模块中，已经说明并实现了一种求球面两点间弧长的方案。但由于solidity目前还不支持浮点型，该方案中的弧度方法，三角函数难以保留小数部分的信息，截断造成的误差经过多次计算的放大后，结果与实际值已存在极大的偏差，无法满足后续计算的需求。为此，需要设计一种新的计算方案。

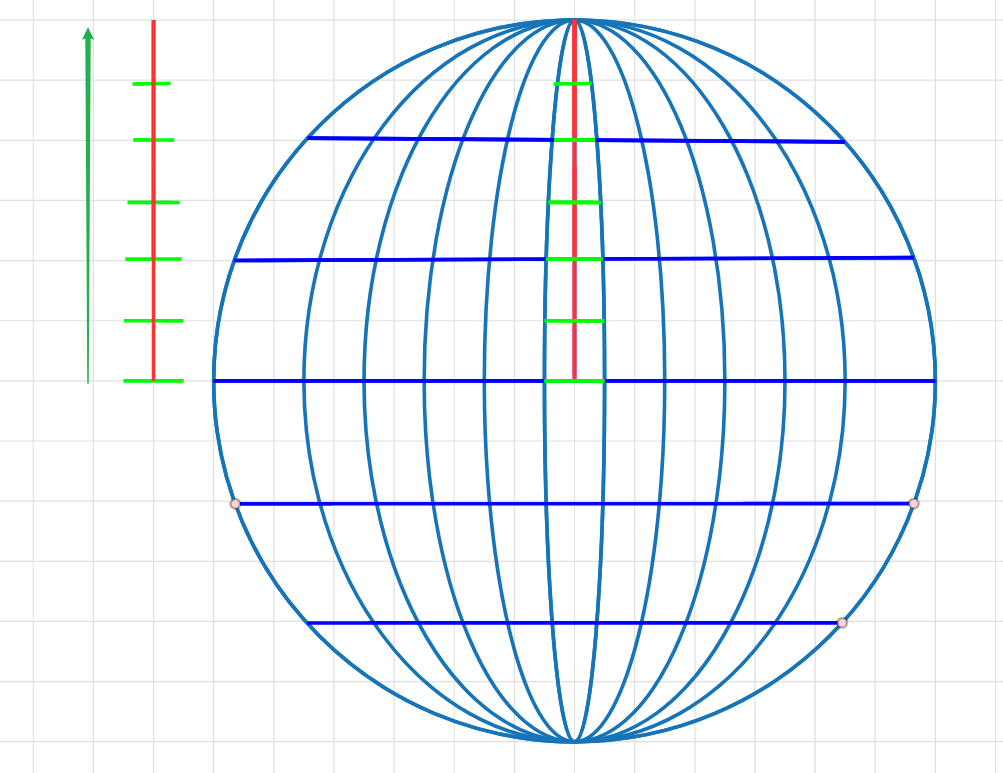
由于该部分实际上处理的是当前位置点，上一个经过的路径点和下一个要通过的路径点，三点间的实际距离往往是数米到数十米的量级，地球表面弧度对其造成的影响可以忽略不计，所以点到点连线的距离即可视作实际距离。因此，我们无需建立三维坐标系，将问题简化为平面坐标系下点到点距离的计算。

这引出了第二个问题：如何将经纬度坐标系转化为平面坐标系。



经纬度坐标系

平面坐标系中两点的距离即为两点坐标差平方和，然而这种方法以x，y轴等长为前提。在经纬度坐标系中，经度固定时，纬度每变更一度，两点间的距离是固定的，约为110km；纬度固定时，经度每变更一度，两点间的相差的距离却是变化的。如图3-2所示，纬度固定经度变化时，两点间的距离在赤道最大，随着经度绝对值的增加而减小，在经度绝对值为90时达到最小值0。



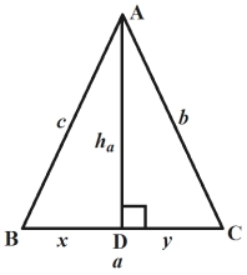
经纬度变化图

此处采用一个折中的方式，虽然我们无法在不使用浮点数的条件下计算出不同纬度下，经度变化一度相差的距离。但可以提前计算该值，建立一个映射表，在需要转换经度轴时通过查表得到所需的值，从而完成从经纬度坐标系到平面坐标系的转换。需要考虑的是映射表的粒度，需要划分合适的粒度使通过查表法得到的数组与实际数据接近。

经度到距离映射表（单位：度/米）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 十位/个位 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 0 | 110604 | 110587 | 110537 | 110452 | 110335 | 110183 | 109998 | 109779 | 109528 | 109242 |
| 1 | 108924 | 108572 | 108187 | 107769 | 107318 | 106835 | 106319 | 105771 | 105190 | 104578 |
| 2 | 103934 | 103258 | 102550 | 101811 | 101042 | 100241 | 99410 | 98549 | 97657 | 96736 |
| 3 | 95786 | 94806 | 93797 | 92760 | 91694 | 90601 | 89480 | 88332 | 87157 | 85955 |
| 4 | 84727 | 83473 | 82194 | 80890 | 79561 | 78208 | 76831 | 75431 | 74008 | 72562 |
| 5 | 71094 | 69605 | 68094 | 66563 | 65011 | 63439 | 61848 | 60239 | 58611 | 56965 |
| 6 | 55301 | 53621 | 51925 | 50213 | 48485 | 46743 | 44986 | 43216 | 41433 | 39636 |
| 7 | 37828 | 36009 | 34178 | 32337 | 30486 | 28626 | 26757 | 24880 | 22996 | 21104 |
| 8 | 19206 | 17302 | 15393 | 13479 | 11561 | 9640 | 7715 | 5788 | 3860 | 1930 |

* + - * 1. 点到直线距离的计算

该函数负责完成经纬度坐标点到另两个经纬度坐标点连线的距离的计算。

脱离判断示意图

图3-1中A点即为当前位置，B点为上一个经过的路径点，C点为下一个要通过的路径点，ha即为所求长度。

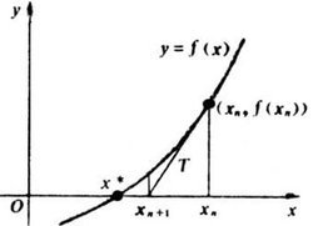
将借助海伦公式，其中，求ha的值，具体过程如下

1. 计算BC,AC,AB的长度，分别记为a，b，c。
2. 若b2>=c2+a2或c2>=b2+a2，即三点未构成三角形，则返回b，c中较短的一段。
3. 计算a，b，c之合，取结果的1/2记作p，即三角形周长的一半。
4. 计算，结果记为S，即三角形的面积。
5. 返回2S/a的值，即为所求线段ha的长度。
   * + - 1. 数据平方根的计算

该模块将使用实现牛顿迭代法结合位运算，进行高效准确的数据平方根计算。

通过简单循环来寻找数据的平方根，复杂度约为O(log2n）。循环是智能合约中消耗gas最多的部分，我们需要一种迭代次数较小的方法来实现数据平方根的计算，否则计算会将交易附带的gas被消耗殆尽，使得计算被终止。

设r是f(x) = 0的根，选取x0作为r初始近似值，过点(x0,f(x0) )作曲线y = f(x)的切线L，L的方程为y = f(x0)+f'(x0)(x-x0)，求出L与x轴交点的横坐标，称x1为r的一次近似值。

过点(x1，f(x1))作曲线y = f(x)的切线，并求该切线与x轴交点的横坐标，称x2为r的二次近似值。重复以上过程，得r的近似值序列，其中，称为r的n+1次近似值，上式称为牛顿迭代公式。

牛顿迭代法图例

根据以上原理可得，牛顿迭代法计算平方根的迭代公式如下:

由于已知智能合约最大支持uint256格式的数据，我们可以通过位运算处理不同位数的输入，截去对计算无用的位数，提前放大r值来加速迭代的进程。将迭代此处控制到七次内。使得计算的复杂度从O(log2n)降低到O(1)，满足了在智能合约中进行运算的需求。

* + - 1. 脱离路径事件模块

该模块负责在脱离路径事件触发后做出处理。主要任务包括监听区块链事件，在事件触发后永久化记录事件信息，向紧急联系人告知事件信息。

* + - * 1. 智能合约部分

在为了实现预期功能，需要在智能合约中加入一个结构体用于记录事件信息。

struct ErrorReport

{

uint startlat; //用于记录出发位置

uint startlng;

uint endlat; //用于记录原定目标

uint endlng;

uint poilat; //用于记录现在位置

uint poilng;

uint errorTime; //用于记录事件发生的时间

}

ErrorReport[] report; //结构体数组，用于容纳复数的错误报告

永久化记录与判断路径脱离函数绑定，在事件触发时自动执行，将错误信息写入结构体中即可实现。此外，还需定义路径脱离事件，便于区块链外监听事件。

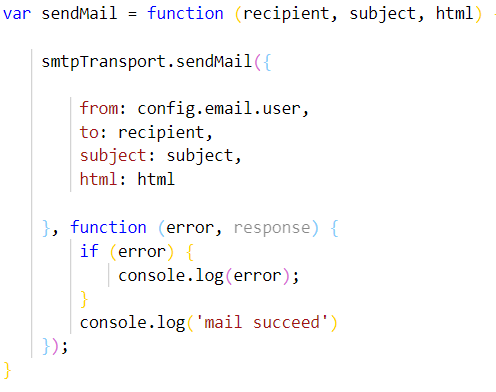
* + - * 1. 合约外部分

该部分负责向紧急联系人告知事件信息，主要逻辑如下：

1. 后台监听区块链中事件
2. 路径脱离事件发生，事件信息写入区块链，发送错误信息给监听者
3. 后台从错误信息中取得用户地址，以此为输入，通过紧急联系人模块提供的接口获得该用户的紧急联系人信息。
4. 通过nodemailer向紧急联系人发送通知邮件。

路径脱离事件的主要逻辑都在区块链中完成，但区块链无法直接实现收发邮件的功能，所以通知紧急联系人还需要通过区块链外部分的合作。

将在3.2.3节中提到的简易nodejs后台中完成这些功能。nodejs有已经封装好的邮件方法包，此处选用了nodemailer包中的stmp协议包nodemailer-smtp-transport，仅需少量配置即可完成电子邮件的发送。



封装后的发送邮件方法

* 1. 系统实现与测试
     1. 系统实现

该部分将介绍设计思路具体实现后的成果，移动端APP主要以APP截图的形式展示，逻辑主要以核心代码的形式展示。

* + - 1. 移动端APP实现
         1. 地图展示模块

在该模块主要实现了地图展示，位置选取和标注，位置信息显示，位置搜索和跳转功能。如图4-1所示，1处为用户信息按钮，2处为当前所在市，3处为用户位置和用户选取位置标记，4处为当前位置栏，5处为目标位置栏。

APP将根据用户位置，自动跳转到相应位置，同时使用地理逆编码功能显示当前位置的具体信息。用户位置和用户选中的地点分别以小绿旗和水滴状标签显示。点击下方的当前位置栏或目标位置栏将跳转到搜索页面，通过关键词搜索位置并跳转，便于用户选中远处的位置。

地图支持监听拖动事件、点击事件和重绘事件。拖动事件结束时，取地图聚焦位置为当前地点；点击事件取事件地点为目标地点；重绘事件执行地图和路径信息的重绘。



移动端APP首页

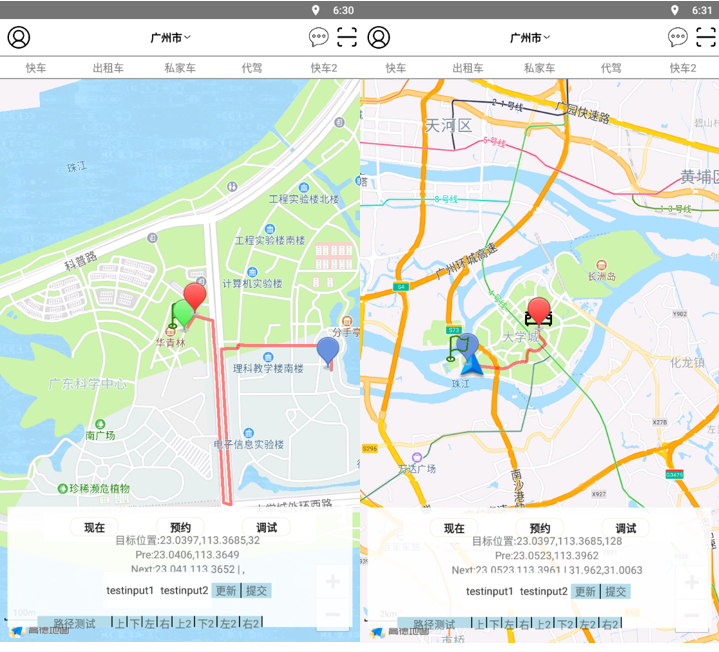
搜索页面图4-2所示，支持在搜索框中输入关键词，返回匹配位置列表，点击相应位置即可在主页面跳转到目标位置。



搜索页面

* + - * 1. 路径规划模块

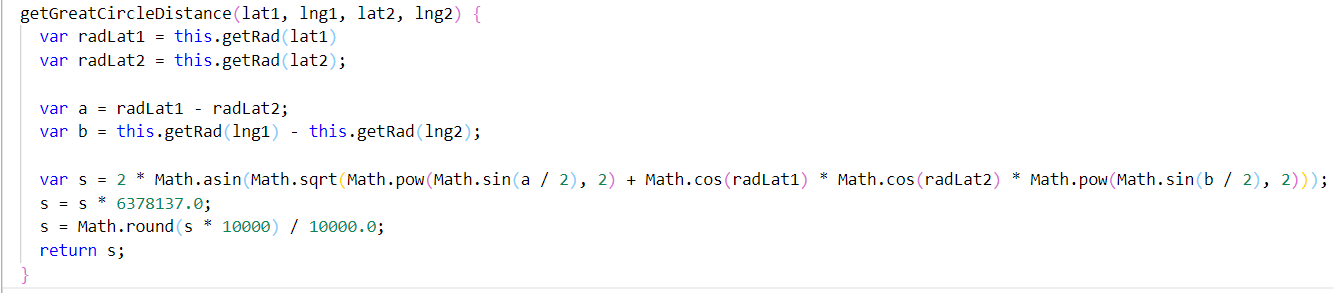
该模块主要实现了推荐路线的获取和展示。通过高德WebAPI，根据用户选择的起止地点，得到推荐路线的坐标数组，进行二次处理，将数组切割为合适的长度，防止一次行渲染过长路线导致的卡顿。使用地图展示模块中的重绘事件，实现数据数组到地图路线的映射，以完成直观的路径引导。



路径规划效果图



路径规划部分代码



封装后的求两点距离函数

* + - * 1. 交易打包和事件监听模块

该模块设计理念是仅实现交易的打包和封装，将逻辑交给智能合约处理。所以无需复杂的功能实现。

然而为了提高编码效率，实现代码复用，根据需求利用web3js和ethereumjs-tx封装了一个工具类，提供以下接口：

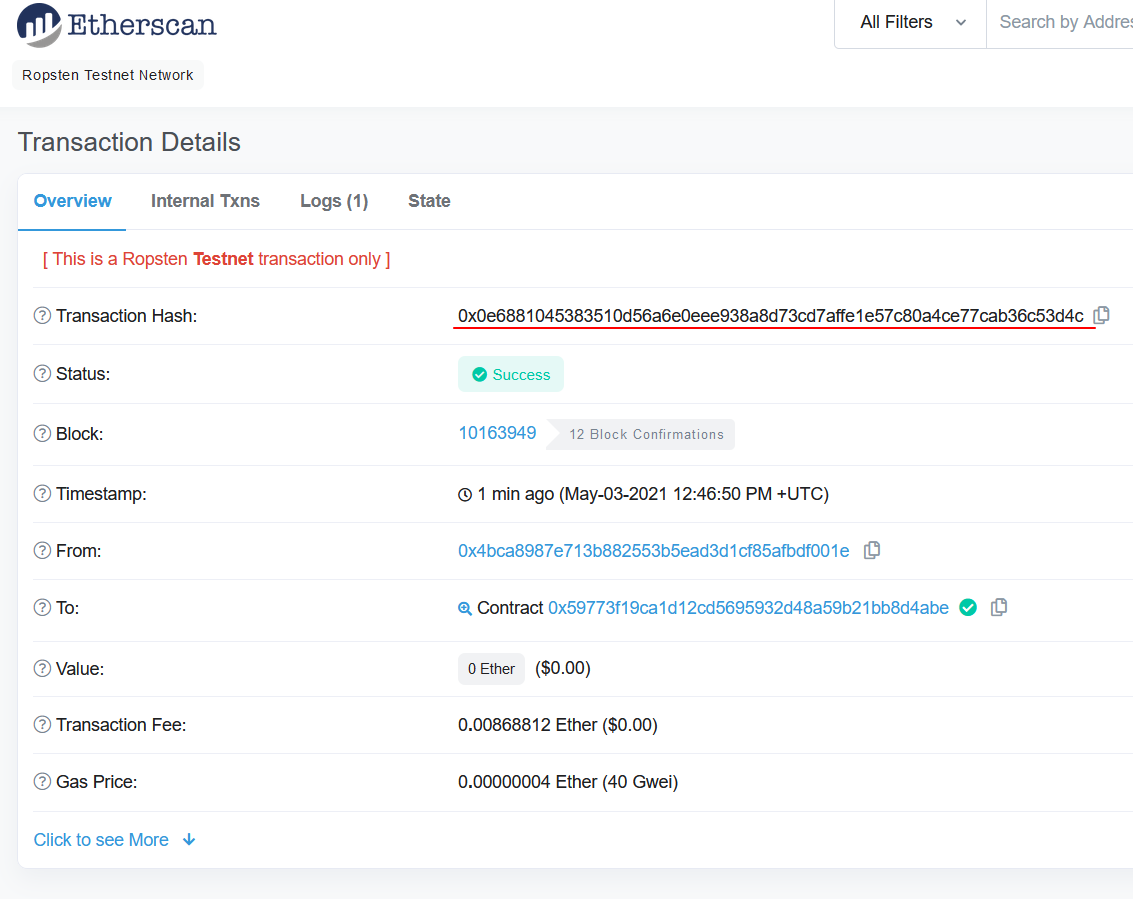
1. 提供默认的Web3HttpProvider，以便于进行合约操作。
2. 提供默认的Web3WebSocketProvider，以便于监听合约事件。
3. 输入ABI和合约地址，返回可操作的合约对象
4. 输入账户，密钥，目标地址和金额自动完成交易创建、签名和广播。
5. 输入账户，密钥，合约地址和调用方法自动完成交易创建、签名和广播。



封装后的交易发送函数



封装函数的调用

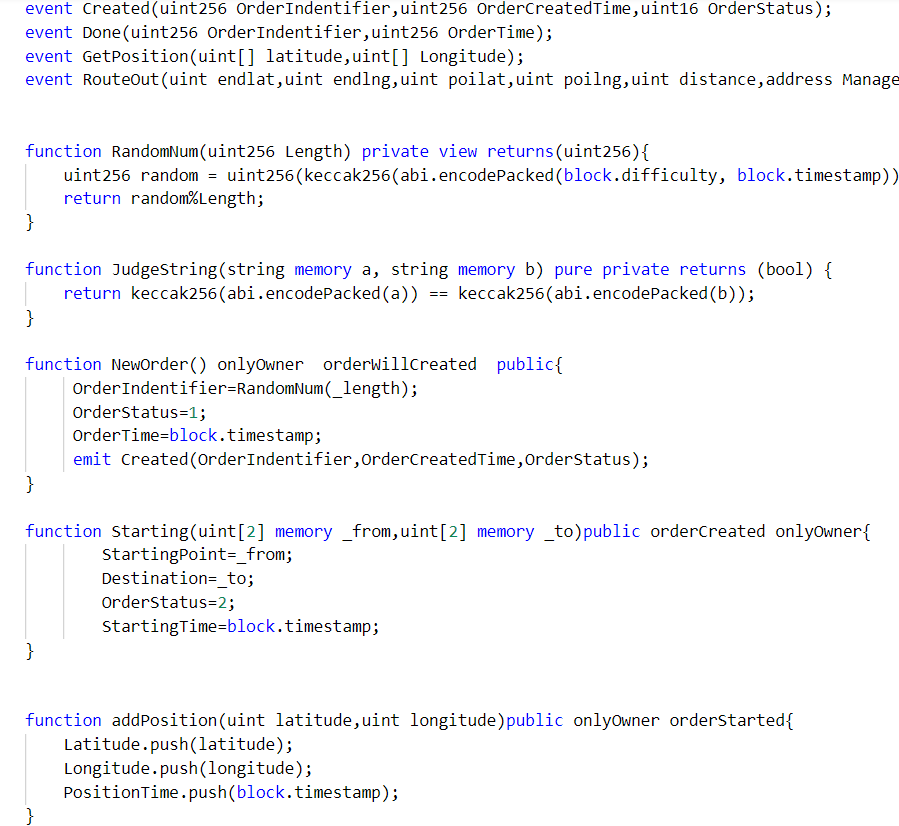


区块链浏览器上的交易信息

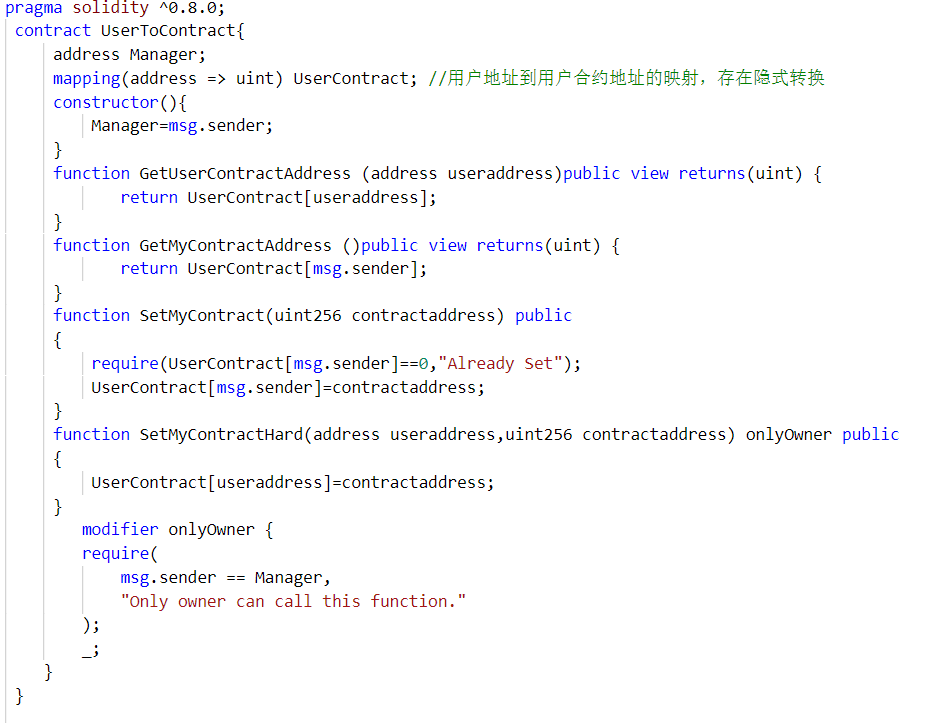
图4-7中演示了工具类中智能合约调用接口的使用，成功打包交易并返回了交易hash值，在区块链浏览器中我们可以看到，该交易已成功上链。

* + - 1. 区块链和智能合约实现

该部分中行程信息记录模块和紧急联系人设计模块已严格按照设计思路实现，由于这两部分的逻辑较为简单，在此不再赘述。



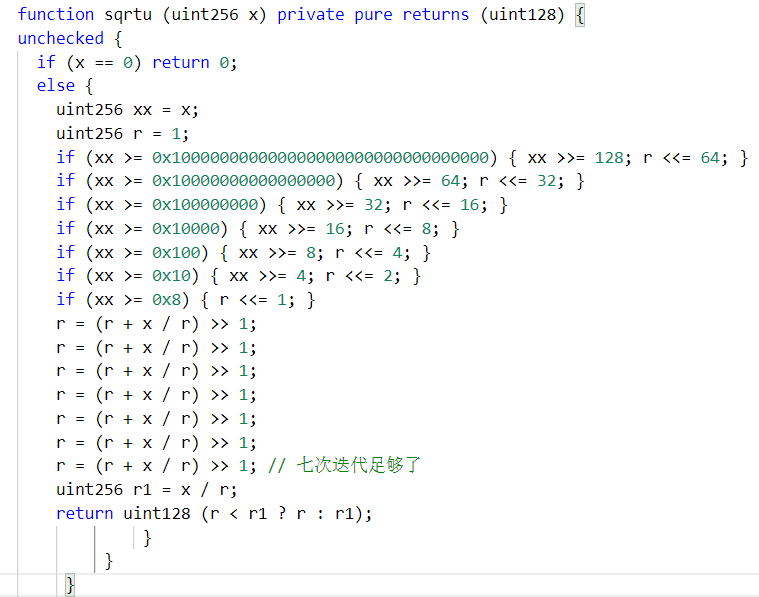
行程信息记录模块部分代码



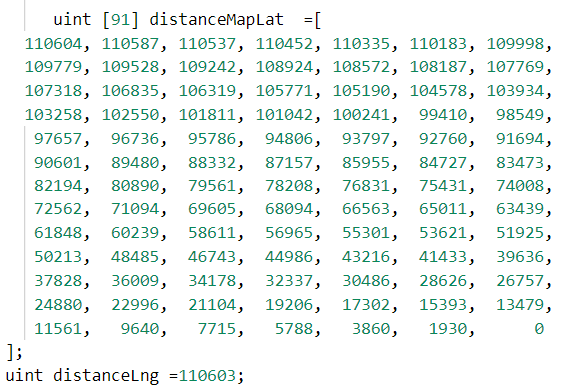
紧急联系人设置模块代码

* + - * 1. 路径脱离判断模块

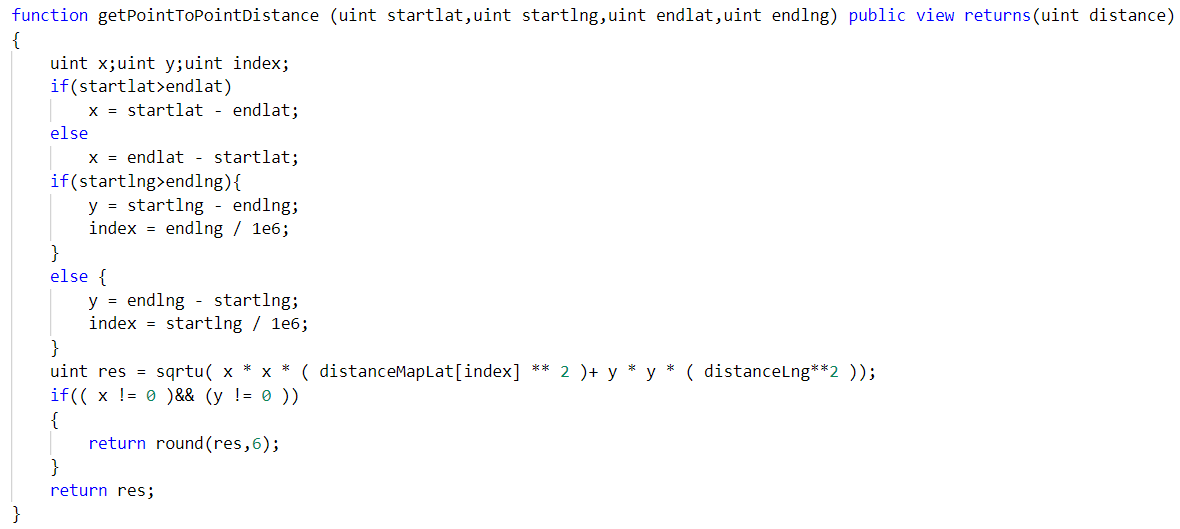
该部分主要负责实现设计中预期的算法，核心代码如下：



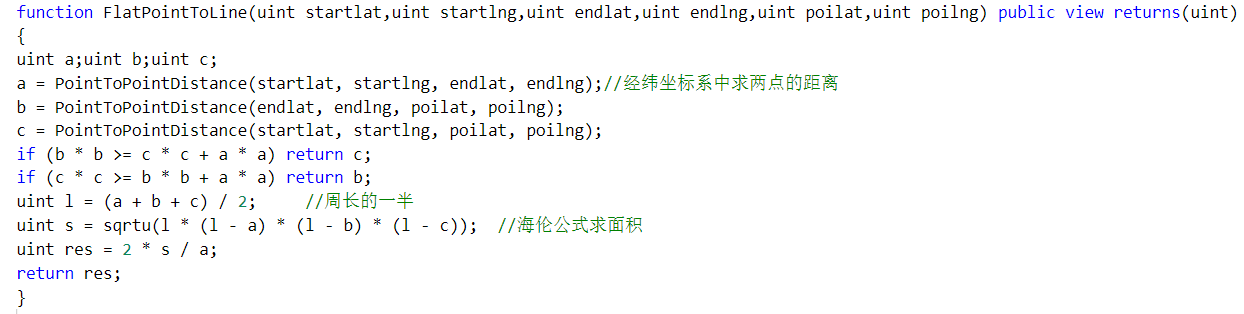
牛顿迭代法计算平方根



经纬度映射表



查表法实现距离计算



点到线距离计算

对实现后的智能合约进行测试，当测试坐标分布在赤道附近时，误差小于1%；

测试坐标的纬度分布在23度（广州）时，误差在2.5%左右；

* + - * 1. 脱离事件模块
    1. 系统测试
       1. 测试环境配置
       2. 测试方案
       3. 操作流程
       4. 功能测试
  1. 总结
  2. 参考文献
  3. 致谢