<基于区块链的信息共享系统>

软件需求规约

版本 <1.1>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2018年10月19日 | 1.0 | 需求规约初稿 | 第四组全员 |
| 2018年10月20日 | 1.0.1 | 需求规约初稿修改 | 第四组全员 |
| 2018年10月25日 | 1.0.2 | 数据挖掘需求规约细化 | 第四组全员 |
| 2018年12月9日 | 1.1 | 需求变更 | 全员 |

目录

[1. 简介 4](#_Toc534837782)

[1.1 目的 4](#_Toc534837783)

[1.2 定义、首字母缩写词和缩略语 4](#_Toc534837784)

[1.3 参考资料 4](#_Toc534837785)

[2. 整体说明 5](#_Toc534837786)

[2.1 产品总体效果 5](#_Toc534837787)

[2.2 产品功能 5](#_Toc534837788)

[2.3 用户特征 5](#_Toc534837789)

[2.4 约束 5](#_Toc534837790)

[2.5 假设与依赖关系 5](#_Toc534837791)

[3. 具体需求 5](#_Toc534837792)

[3.1 功能 5](#_Toc534837793)

[3.1.1 用例图 6](#_Toc534837794)

[3.1.2 EHR管理 用例规约 6](#_Toc534837795)

[3.1.3 数据挖掘 用例规约 6](#_Toc534837796)

[3.2 易用性 6](#_Toc534837797)

[3.2.1 用户培训时间 6](#_Toc534837798)

[3.2.2 图形标准 7](#_Toc534837799)

[3.2.3 系统可维护性 7](#_Toc534837800)

[3.3 可靠性 7](#_Toc534837801)

[3.4 性能 7](#_Toc534837802)

[3.5 可支持性 7](#_Toc534837803)

[3.5.1 编码标准与命名约定 7](#_Toc534837804)

[3.5.2 维护访问权限 8](#_Toc534837805)

[3.6 设计约束 8](#_Toc534837806)

[3.6.1 编程语言 8](#_Toc534837807)

[3.6.2 编程工具和框架 8](#_Toc534837808)

[3.6.3 兼容性约束 8](#_Toc534837809)

[3.6.4 测试约束 8](#_Toc534837810)

[3.7 联机用户文档和帮助系统需求 8](#_Toc534837811)

[3.7.1 用户手册 8](#_Toc534837812)

[3.7.2 帮助系统要求 8](#_Toc534837813)

[3.8 接口 8](#_Toc534837814)

[3.8.1 用户界面 8](#_Toc534837815)

[3.8.2 软件接口 9](#_Toc534837816)

[3.9 适用的标准 9](#_Toc534837817)

软件需求规约

# 简介

## 目的

本文档旨在定义软件总体要求，作为用户和软件开发人员之间相互了解的基础。同时，提供性能要求、初步设计和对用户影响的信息，作为软件人员进行软件结构设计和编码的基础。最后，本文档将作为项目管理、软件总体测试和质量保证的依据。

## 定义、首字母缩写词和缩略语

**区块链（英语：blockchain或block chain）：**是借由密码学串接并保护内容的串连交易[记录](https://zh.wikipedia.org/wiki/%E8%AE%B0%E5%BD%95" \o "记录)（又称区块）。每一个区块包含了前一个区块的[加密散列](https://zh.wikipedia.org/wiki/%E5%AF%86%E7%A2%BC%E9%9B%9C%E6%B9%8A%E5%87%BD%E6%95%B8" \o "密码散列函数)、相应时间戳记以及交易数据（通常用[默克尔树](https://zh.wikipedia.org/w/index.php?title=%E9%BB%98%E5%85%8B%E7%88%BE%E6%A8%B9&action=edit&redlink=1)算法计算的散列值表示），这样的设计使得区块内容具有难以篡改的特性。用区块链所串接的分布式账本能让两方有效纪录交易，且可永久查验此交易。

**Smart Contract:** 智能合约（英语：Smart contract ）是一种旨在以信息化方式传播、验证或执行[合同](https://zh.wikipedia.org/wiki/%E5%90%88%E5%90%8C" \o "合同)的计算机协议。智能合约允许在没有第三方的情况下进行可信交易。这些交易可追踪且不可逆转。智能合约概念于1994年由[Nick Szabo](https://zh.wikipedia.org/wiki/Nick_Szabo" \o "Nick Szabo)首次提出。智能合同的目的是提供优于传统合同方法的安全，并减少与合同相关的其他[交易成本](https://zh.wikipedia.org/wiki/%E4%BA%A4%E6%98%93%E6%88%90%E6%9C%AC" \o "交易成本)。

**EHR：**电子健康纪录（Electronic Health Record），又称为电子健康档案，简称EHR，是电子化的个人健康纪录（病历、心电图、医疗视频等），电子健康纪录可以经由计算机或网上访问，可以包含现今与过去个人的健康信息。除此之外，电子健康纪录也能够包括医学相关的引用数据、医疗处置、药物使用、人口统计数据、其他与非医疗的管理数据等等。然而，一个理想的电子健康纪录系统，现今还没有一个软件或是供应商所能够建置出来。

**SPSS：**SPSS是统计产品与服务解决方案（Statistical Product and Service Solutions）的简称，为[IBM公司](https://zh.wikipedia.org/wiki/IBM%E5%85%AC%E5%8F%B8" \o "IBM公司)的一系列用于[统计学](https://zh.wikipedia.org/wiki/%E7%B5%B1%E8%A8%88%E5%AD%B8" \o "统计学)分析运算、[数据挖掘](https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E6%8C%96%E6%8E%98" \o "数据挖掘)、预测分析和决策支持任务的软件产品及相关服务的总称，有[Windows](https://zh.wikipedia.org/wiki/Windows" \o "Windows)和[macOS](https://zh.wikipedia.org/wiki/MacOS" \o "MacOS)等版本。

**ATM：**自动调优模型（Auto-Tuned Models，简称ATM）, 一个自动化机器学习系统，基于Auto-sklearn实现的平台。它提出了一种贝叶斯优化和多臂赌博机选择策略相结合的算法，将算法选择和超参数调优这两个阶段异化为选择超参数分支和超参数调优，而且在实践中对来自OpenML平台的420个分类任务，得到了较好的结果。

**SQL：**是一种[特定目的编程语言](https://zh.wikipedia.org/wiki/%E7%89%B9%E5%AE%9A%E7%9B%AE%E7%9A%84%E7%A8%8B%E5%BC%8F%E8%AF%AD%E8%A8%80" \o "特定目的编程语言)，用于管理[关系数据库管理系统](https://zh.wikipedia.org/wiki/%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F" \o "关系数据库管理系统)（RDBMS），或在[关系流数据管理系统](https://zh.wikipedia.org/wiki/%E5%85%B3%E7%B3%BB%E6%B5%81%E6%95%B0%E6%8D%AE%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F" \o "关系流数据管理系统)（RDSMS）中进行流处理。SQL基于[关系代数](https://zh.wikipedia.org/wiki/%E5%85%B3%E7%B3%BB%E4%BB%A3%E6%95%B0_(%E6%95%B0%E6%8D%AE%E5%BA%93)" \o "关系代数 (数据库))和[元组关系演算](https://zh.wikipedia.org/wiki/%E5%85%83%E7%BB%84%E5%85%B3%E7%B3%BB%E6%BC%94%E7%AE%97" \o "元组关系演算)，包括一个[数据定义语言](https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E5%AE%9A%E4%B9%89%E8%AF%AD%E8%A8%80" \o "数据定义语言)和[数据操纵语言](https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E6%93%8D%E7%BA%B5%E8%AF%AD%E8%A8%80" \o "数据操纵语言)。

## 参考资料

* + 1. 高级软件开发课程第4组，<基于区块链的电子病历共享信息系统需求规约>，1.0版
    2. 高级软件开发课程第4组，<基于区块链的电子病历共享信息系统立项建议书>
    3. 维基百科，条目：区块链
    4. 维基百科，条目：智能合约
    5. RUP软件需求规约文档
    6. Edward Choi, Andy Schuetz, Walter F. Stewart and Jimeng Sun. *Using recurrent neural network models for early detection of heart failure onset*. JAMIA. 2017.

# 整体说明

## 产品总体效果

本产品（BlockHealth）基于区块链技术，构建基于区块链技术的分布式系统，完成有效地共享医疗机构之间的EHR的功能，并在不涉及任何第三方的情况下提供必要的隐私保护，解决信息隐私保护和信息共享之间的矛盾问题，并使病历信息被合理安全地使用。

## 产品功能

本产品面向医疗机构所维护的分布式系统的维护人员和开发人员以及医疗机构的医疗人员，提供了统一的SQL请求功能和SQL操作记录查询的功能；面向医疗数据的研究人员提供了数据挖掘的功能；面向医疗机构从业人员提供了EHR的加密查阅和修改功能。

## 用户特征

|  |  |  |
| --- | --- | --- |
| 角色分类 | 角色名称 | 角色描述 |
| 高级用户 | 医疗数据研究者 | 该类用户将进行医疗数据的数据挖掘。  该类用户对平台的连续使用时间较长，数据读I/O操作的数据量大，对平台的性能需求较高。 |
| 普通用户 | 医疗机构从业人员（如医生等） | 该类用户将进行EHR的查阅、更新。  该类用户对平台的可靠性要求高，对性能需求不高；对平台的易用性依赖程度较高（尤其是图形界面）。 |
| 其他用户 | 医疗机构系统维护人员 | 该类用户将查看SQL操作记录。  该类用户对平台具有一定的维护作用。 |

## 约束

1. 产品应兼容多种医疗机构的数据库系统支持，兼容多种数据库的接入。
2. 开发语言为Sodility、Javascript，采用面向对象方法的开发方法，采用3-tier架构。
3. 系统必须保证数据的安全。

## 假设与依赖关系

假设针对区块链分布式系统、对开发语言Sodility开发经验不足，功能设计不够完善，都会影响本项目的开发流畅性。项目时间设计不足都会影响到后续开发进程。

# 具体需求

## 功能

1. 本产品面向医疗机构从业人员提供EHR管理功能；
2. 本产品面向医疗数据的研究人员提供数据挖掘的功能；
3. 本产品面向医疗机构系统的维护人员，提供了SQL操作记录查询的功能。

### 用例图

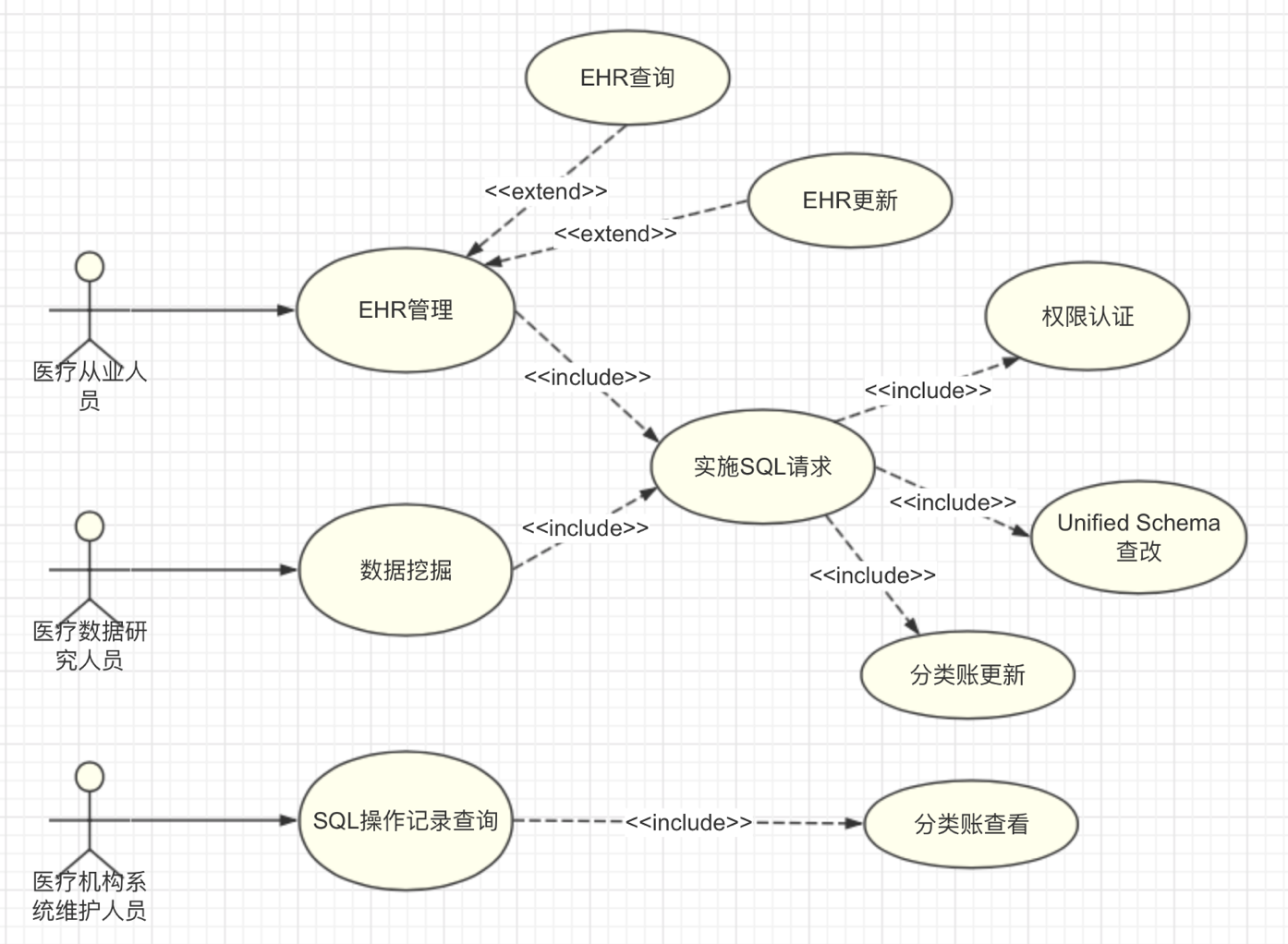


图1

如图1所示，该图通过Use Case模型介绍了产品的整体功能需求。针对三类产品使用人员（医疗机构数据库系统维护人员、医疗数据研究者和医疗机构从业人员），分别提供了不同的功能。详细功能定义与用例规约将在以下部分阐述。

### EHR管理 用例规约

EHR管理用实现了为医疗从业人员提供EHR管理功能。该功能为BlockHealth平台的核心功能。医疗从业人员通过用户图形界面，登陆EHR管理平台，完成EHR查询操作。EHR的所有管理操作，均会记录在分类账目中。

### 数据挖掘 用例规约

数据挖掘功能是BlockHealth平台的扩展功能。基于现状，该平台的应用于数据挖掘的数据集在现实世界中分为四大类：包括身体指数、生活习惯、生理状态和历史疾病。

## 易用性

### 用户培训时间

1. 普通用户，即医疗机构从业人员，在首次使用该平台即可掌握。
2. 高级用户，医疗数据研究者具有数据分析经验，参考用户文档，培训时间在30～60分钟；高级用户中无经验者的培训时间不可控，一般不超过5天。
3. 其他用户，即医疗机构数据库系统维护人员具有开发和维护经验，参考用户文档在30分钟左右可掌握SQL相关的接口使用。

### 图形标准

产品的图形使用符合Microsoft 的GUI标准要求。

### 系统可维护性

系统设置了降级模式，在系统维护期间可以启用。在此期间，普通用户只能查看本地数据库的EHR，高级用户功能全部受限。

## 可靠性

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *可用时间* | *MTBF(day)* | *MTTR(min)* | *精确度* | *最高缺陷率(bugs/KLOC)* |
| *Min* | *94.01%* | *150* | *15* | *93.00%* | *0.40* |
| *Max* | *99.99%* | *400* | *120* | *99.99%* | *4.01* |

（以上数据为估约数据）

## 性能

初步分析，本产品主要的性能瓶颈集中在多种数据库的性能表现。合理的性能需求参考了多种数据库的性能指标，得到以下数据：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *对事务的响应时间（ms/千件）* | *吞吐量（件/s）* | *容量(事务数)* | *资源利用情况（内存GB/高级用户平台）* |
| *Avg* | *3000* | *2000* | *6000* | *0.5* |
| *Max* | *50000* | *3000* | *10000* | *8.0* |

（以上数据为基于10线程情况的估约数据）

## 可支持性

### 编码标准与命名约定

以下为编码规范及命名规范：

1. 名字中每个单词的首字母(比如类型或变量)要大写，单词之间通常不要使用下划线。比如， Health和UPrimitiveComponent是符合规范的，而lastMouseCoordinates或delta\_coordinates是不符合规范的。
2. 类型名称有一个额外的大写字母前缀，以便将类型名称和变量名称区分开来。比如，FSkin是类型名称，而Skin是FSkin的一个实例。
3. 类型和变量名称必须是名词。方法名称是动词，该动词描述了该方法的作用或者描述了无效的方法返回值。
4. 变量、方法及类的名称应该清晰、明确且具有描述性。名称的作用域越大，取一个符合标准的具有描述性的名称的重要性便越强。避免过度缩写。
5. 所有变量都应该一次仅声明一个，以便可以提供有关这个变量的含义的注释。同时，这也符合JavaDocs 风格的要求。您可以在变量前面使用多行或单行注释，可以留一个空行来给变量分类。
6. 所有返回布尔值的函数都应该询问返回值是真还是假这个问题，比如 "IsVisible()" 或 "ShouldClearBuffer()" 。所有布尔变量都必须以"b"字母为前缀(比如 "bPendingDestruction" 或 "bHasFadedIn")。
7. 过程(没有返回值的函数)命名时应该使用一个强动词后面加一个对象。如果方法的对象正是该方法所属的对象，那么方法将根据上下情境来获得对象，这是种例外情况。命名时要避免使用"Handle"和"Process"开头，这些动词表达的意思模糊不清。
8. 如果一个参数是通过引用传入函数且该函数要向此参数写入值，那么我们鼓励您使用 "Out" 作为函数参数名称的前缀，但这不是强制要求。这样便显而易见地表示出传入到该参数中的值会被该函数所代替。
9. 对于能够返回值的函数的名称，应该描述出该返回值的意思；名称应该可以清楚地表达出要返回的是什么值。这对于布尔函数尤为重要。考虑以下两个示例方法：

bool CheckTea(FTea Tea) {...} // true 代表什么意思？

bool IsTeaFresh(FTea Tea) {...} // 名称可以明确表示true代表茶是新鲜的

### 维护访问权限

普通用户在通过权限认证后具有访问和修改（受限制的）EHR的权限。

高级用户中，医疗机构的系统维护人员具有SQL接口的访问权限、医疗数据的研究人员具有数据挖掘接口的访问权限。

## 设计约束

### 编程语言

产品开发语言为Sodility、Javascript，支持多种数据库语言。

### 3.6.2 编程工具和框架

以太坊web3.js库，nodejs，npm，采用3-tier架构。

### 3.6.3 兼容性约束

产品平台支持多种数据库（MySQL、Oracle和Bigtable等），能够解析多种SQL语言。

### 3.6.4 测试约束

测试需在Windows、Linux、macOS多环境下完成。

## 联机用户文档和帮助系统需求

### 用户手册

用户手册需要提供详细的用户使用帮助说明，包括系统的基本介绍、功能。配置界面要求在每一步显示当前执行的操作，在每个设有选项处提供详细的功能说明。这些说明将每个选项的功能和选于不选的区别进行详述。

### 帮助系统要求

帮助系统要包含用户注册、用户登陆的管理；EHR管理、SQL相关功能的详细步骤；最后附数据库维护的基本知识和方法。

## 接口

### 用户界面

医疗从业人员用户所看到的界面的模块包括：用户登陆界面、EHR查询界面。专业人员将通过接口与系统进行交互。

### 软件接口

|  |  |
| --- | --- |
| 需求类别 | 软件接口 |
| 编号 | *BH.INIF.DATAMINING.001* |
| 需求名称 | 数据挖掘接口 |
| 优先级 | 低 |
| 描述 | 医疗数据研究人员通过该接口进行数据挖掘的功能使用。迭代一期不开放。 |

|  |  |
| --- | --- |
| 需求类别 | 软件接口 |
| 编号 | *BH.INIF.SQL.001* |
| 需求名称 | SQL接口 |
| 优先级 | 低 |
| 描述 | 医疗机构系统维护人员通过SQL对接入平台的数据库中的EHR进行SQL操作。迭代一期不开放。 |

|  |  |
| --- | --- |
| 需求类别 | 软件接口 |
| 编号 | *BH.INIF.BCLEDGER.001* |
| 需求名称 | 分类账获取接口 |
| 优先级 | 低 |
| 描述 | 医疗机构系统维护人员通过该接口得到分类账内容。迭代一期不开放。 |

## 适用的标准

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 标准名称 | 标准号 | |
| 1 | 信息处理——数据流程图、程序流程图、系统流程图、程序网格图和系统资源图的文件编辑GB符号及约定 | 526-89 | ISO 5807-1985 |
| 2 | 软件工程术语 | GB/T11457-89 |  |
| 3 | 软件工程标准分类方法 | GB/T15538-95 |  |
| 4 | 软件开发规范 | GB 8567-88 |  |
| 5 | 软件维护指南 | GB/T14079-93 |  |
| 6 | 计算软件需求说明编制指南 | GB 9385-88 | ANSI/IEEE 829 |
| 7 | 计算机软件可靠性和可维护性管理 | GB/T14394-93 |  |
| 8 | 计算机测试文件编制规范 | GB 9386-88 | ANSI/IEEE 830 |