# **KSU Medical BC**

# **Design Document**

# **Team# 2**

# **Team members**

Soham Bapat

Brad Jenkins

Court Robertson

Devyn Wilkins

Oluwafisayo Oke

Peter Twene

Tyler Whitfield

Zeke Whyte

**Date:** (03/27/2019)

**Department:**Software Engineering

**Course:** Fundamentals of Blockchain and Smart Contracts

**Professor:** Dr. Reza Parizi

**TABLE OF CONTENTS**

**1. INTRODUCTION**

1.1 Purpose

1.2 Scope

1.3 Overview

1.4 Reference Material

1.5 Definitions and Acronyms

**2. SYSTEM OVERVIEW**

**3. SYSTEM ARCHITECTURE**

3.1 Architectural Design

3.2 Technical Design

3.3 Design Rationale

**4. DETAILED DESIGN**

4.1 Class diagrams

**5. HUMAN INTERFACE DESIGN**

5.1 UI design

5.2 UX design

### 

### 

### **1.** **Introduction**

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project.

## **1.1** **Purpose**

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to be built. The Software Design Document provides information necessary to provide description of the details for the software and system to be built

## **1.2** **Scope**

This project will consist of using Agile development to build a real open source Ethereum Blockchain Application (Dapp). The project will be completed by May 2019. The overall advantage of using this application is the fact that medical records can be virtually stored, accessed and modified. When medical record are stored like that ,it makes it more difficult for unauthorized individuals to access or even edit them.

## **1.3** **Overview**

This Software Design Document is organized in sections and subsections. These include: System Overview, System Architecture, Data Design, Component Design, Human Interface design and Requirement Matrix

## **1.4** **Reference Material**

## The only reference template currently used in this project are school-provided documents via D2L, which primarily consist of IEEE templates.

## **1.5** **Definitions and Acronyms**

|  |  |
| --- | --- |
| **Term** | **Meaning** |
| **SDD** | **Software Design Document** |
| **IEEE** | **Institute of Electrical and Electronics Engineers** |
| **Dapp** | **Decentralized application** |

**2.** **SYSTEM OVERVIEW**

The purpose of this project is to develop and to build a real open source Ethereum Blockchain Application (Dapp). This application would be able to manage and store patient data. The system will be used by doctors and hospitals to access and add to their patient’s data with the permission of the patient in question.The system will be designed using the Ethereum blockchain,Truffle Framework and web3.js.

**3.** **SYSTEM ARCHITECTURE**

## **3.1** **Architectural Design***.*

The overall purpose of the healthcare blockchain product is to establish a patient-centered protocol, enabling a secure intermodal exchange of healthcare information, as deemed appropriate by the consumer(patient), to various providers.

The security aspect was recently expressed by the National Coordinator for Health Information Technology in issuing a shared interoperability roadmap defining the essential components of interoperability, which include:

* Ubiquitous secure network infrastructure
* Verifiable identity and authorization all participants
* Consistent representation of authorization to access electronic health information…

Ideally,each subsystem will operate independently of the others, allowing only those so authorized to interact with each other and only in an approved form. The objective of this structure is to maximize security while democratizing accessibility. There are essentially two major components for completing an effective interface: the hardware component and the software component.

The hardware component is a public-facing entity, a remote procedure call server, acting as an interface with a private-permissioned Ethereum blockchain. The software component acts as an access control to maintain the integrity of data, verifying the validity of any data requestor. These subsystems, effecting this desired functionality, will collaborate with each other as follows:

The hardware component, the remote procedure call server, being a public-facing entity, assures interaction with only authorized blockchain nodes, a HIPAA-compliant data storage unit and a key-authorizing entity.

Interaction occurs when a data request is initiated to the storage unit which decrypts the relevant database requested. Then the information is re-encrypted for transmission, using the public key of the requesting party. This public key is also the contract public key, controlling interface from the blockchain to the HIPAA-protected data.

The software component, a software-access control entity, using its HIPAA-compliant database to maintain data integrity, is responsible for verifying authorization for requestors. This accepts inbound traffic forwarded only from the HIPAA-compliant data base upon validating a transaction-request as qualified. This software also maintains a “customer list” of permissioned requestors.

All patient data information inquiries flow through these two subsystems before any release of patient information, thus enhancing patient data confidentiality.

A diagram included is a basic example of an information-process flow, showing the clinical generation of patient data, its transmission to the blockchain, making it available to all qualified users, and the protection of essential components. This diagram shows how requests for patient data are processed and specific data released via the Blockchain:

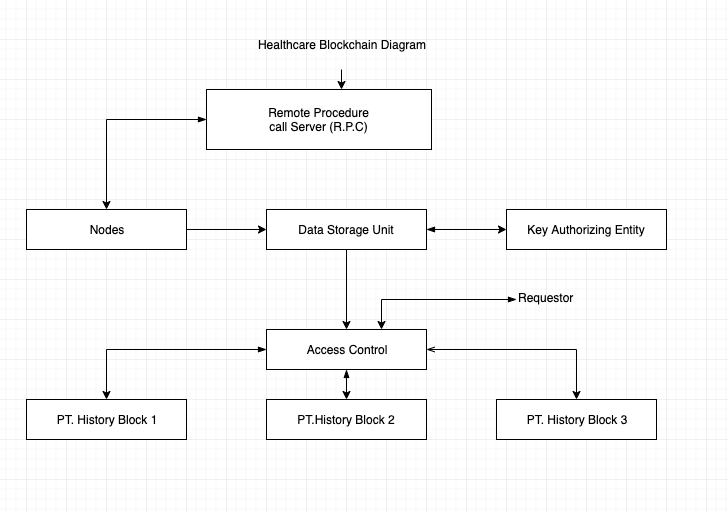
The Remote Procedure Call Server receives a request, verifies it via a recognized node and sends that request on to a Key Authorizing entity, which verifies/authorizes the request. The verified request is then sent to the Data Storage Unit which in turn decrypts the request and sends it to the Access Control Unit.

The Access Control Unit verifies that the requestor has access rights to the stored data, using a Customer(Patient) List Database. Once this is verified, the requestor has access to the specific Block of information specified, which is sent via the requestor’s public key.

Each Block of Data is based on each encounter-record account number covering the date of encounter, diagnosis, medical/surgical procedures, if any, physician notes, lab reports, progress notes, imaging reports, nurses’ notes, etc. This is all transmitted as authorized.

## 

## **3.2 Technical Design**



## **3.3 Design Rationale**

The design of this application is created by keeping the user in mind. In this new type of medical records database system, the user is looking for a system that is more accessible and secure than non-blockchain systems. Medical records being on a blockchain create a protection around the stored data and only authorized user can gain access to it through a key.

Anyone requiring access to the data stored on the blockchain needs to go through the Access Control Unit(ACU) in order to maintain security and privacy. The Access Control Unit uses a patient list database which helps it approve or reject access to the private medical records.

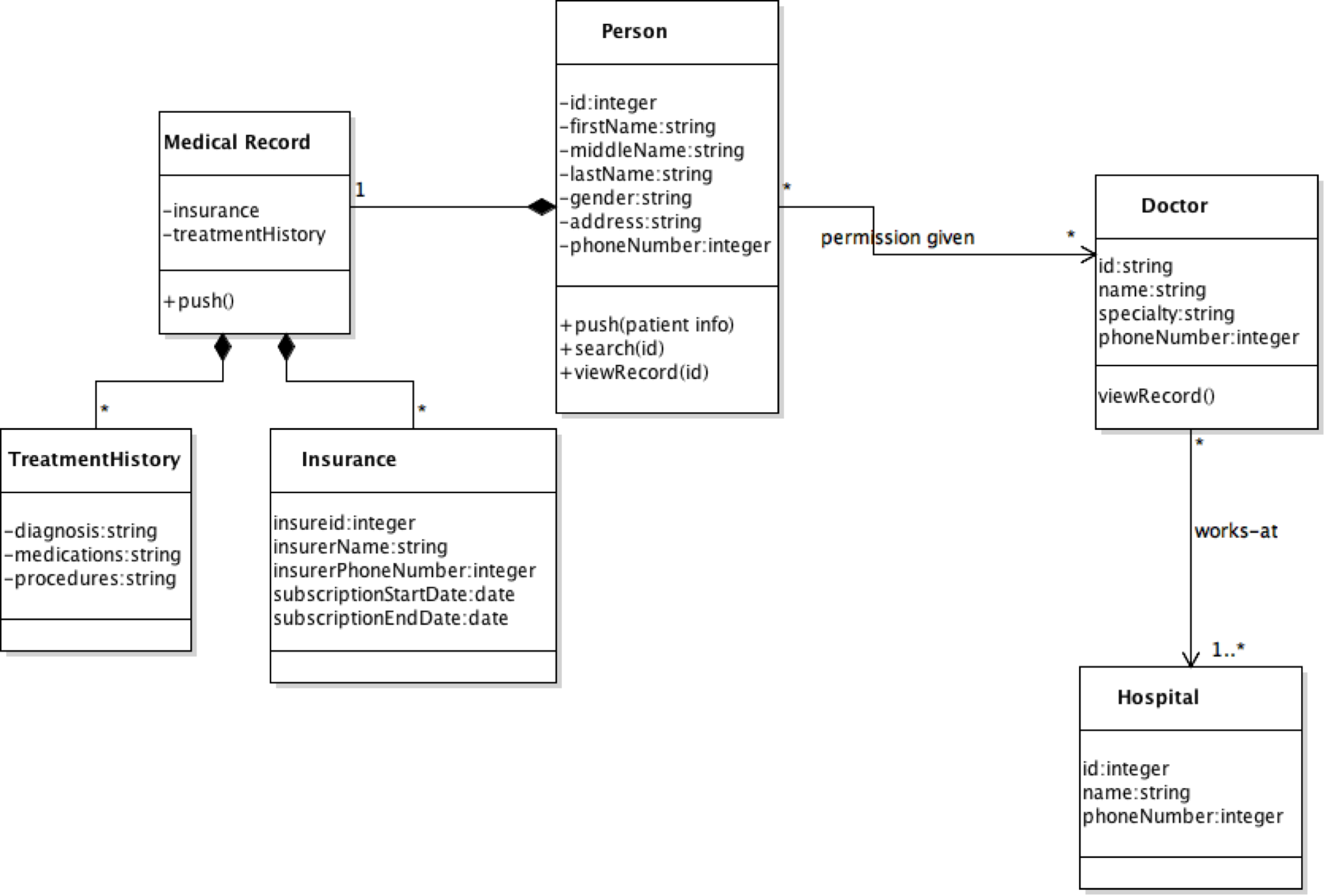
ACU is essentially a gate with a lock and only a person with the right key can get through it.

One other structure discussed was that there would be 2 ACUs back to back. This would double the level of security of the database as a person would require 2 keys to access the information but we decided against it having one key which wasn't secure enough for this project.

Our main focus is the ACU because it is the backbone of this system, so we are making sure that no errors or loopholes are present in it.

**4.** **DETAILED DESIGN**

## **4.1**. **Class diagrams**

**

### 

### **5.** **HUMAN INTERFACE DESIGN**

## **5.1** **UI design**

The healthcare blockchain project will utilize the Ethereum TestRPC (node.js required) and Web3.js to create a simple user interface to interact with the Ethereum smart contract. The TestRPC uses Ethereum-js to simulate client behavior and accelerate the development process. The front end will utilize a regular static HTML page applying Javascript for communicating with the blockchain as well as CSS for designing. Simple reusable UI components will be defined to manage interactions with other users in the network and will adopt an encapsulated set of behaviors/logic. One of the defined components will incorporate a display panel to view a user’s medical information and contain buttons to perform basic functions. Users will have MetaMask to create an instance of Web3 and serve as the token provider if necessary. Users will be able to take advantage of a list of features including user login, medical record addition/searching, user record change requests, access-permission change requests, and user override requests. More features may be included as development continues.

**5.2** **UX design**

A simple seamless merging of services is the main purpose of the project’s user experience. Fundamentally, the user experience will be a reflection of the function deliverables stated in the UI design and operates to satisfy a service level agreement with the users. A simple login/sign up screen with an introduction to the application and its included services will be initially displayed to the users. The user login feature utilizes a database link to pre-existing medical records established by third party providers. After logging in, users will be greeted with a welcome message and medical background image with information corresponding to their last login. This homepage will also include a list of accessible button that link to other pages(five as a placeholder) for navigational purposes. These page will grant the user access to their account information, request forms, medical records (pulled from the database) and any other user services that may be included later in development. Within each page, there will be a set of navigational tools to redirect the user to all other listed button links. Assuming the user has limited knowledge of the immutable nature of the blockchain, notifications and warning prompts will be utilized to notify the user of the inability to change/delete records prior to submitting any request executed in the request forms page.

**Mock Up**

