The System and Software Architecture Description for 2MB

Oct 29 2019

2013210063 Moon Gi Jun

Table of Contents

Version control

Date	Author	Changes	Version
Oct-29 2019	ALL Members	Initial draft	0.1
Oct-31 2019	Kwangrok R.	Component Model modify	0.2
Nov-2 2019	Su Jung B. Sun Wo C.	Behavior Model modify	0.3

List of Figures

1 Introduction

The section covers the purpose of the system and software architecture description document as well as all the references used in the preparation of this document.

1.1 Purpose of SSAD

The System and Software Architecture Description describes how the WebMail system will be implemented in the University of Southern California (USC) user computing environment which meets. The document contains two major sections, system analysis and architectural design of the proposed system.

The system analysis section focuses on the high-level architectural components and interactions that satisfy the system requirements. The intended audience for the section is the domain expert, ISD system developer, who validates the faithfulness of the transformation.

The architectural design section focuses on how the software is implemented using specific software frameworks, components, and technologies. The section describes the implementers such that it is clear for them to proceed to the construction phase. This document is included in the package delivered at Life-Cycle Objective (LCO) milestone.

1.2 Standards and Conventions

MBASE framework is used for the overall software engineering process.

Unified Modeling Language (UML) is used for diagram illustrations. Class diagrams, interaction diagrams, state diagrams are used for graphical presentation in standard notation.

1.3 References

System and Software Requirements Definition 1.0
Easy WinWin negotiation report (10/02/2019)
Stevens, Pooley, “Using UML”, Addison-Wesley

2 Architectural Analysis

2.1 Component Model

2.1.1 COM-01: BlockChain

This component is essential to safely store and manage clinical records.

Identifier	**COM-01**	
Defining quality	A Repository for safe storage of medical records	
Name	BlockChain	
Attributes	a) nodes	
Behaviors	a) Receive transaction requests from the application interface. b) Check the input medical data is valid. c) Send results to the user interface.	
Relationships	a) Patient UI (COM-02) b) Medical institution UI (COM-03) c) Certification Authority (COM-04) d) Verifier (COM-05)	
Roles	Keep the medical data safe	
Constraints	Dependencies	a) Patient UI (COM-02) b) Medical institution UI (COM-03) c) Certification Authority (COM-04) d) Verifier (COM-05)
	Candidate Key	
	Cardinality	1. each interface between server and client

2.1.2 COM-02: Patient User Interfaces

This component is essential for patients to conveniently check their medical records.

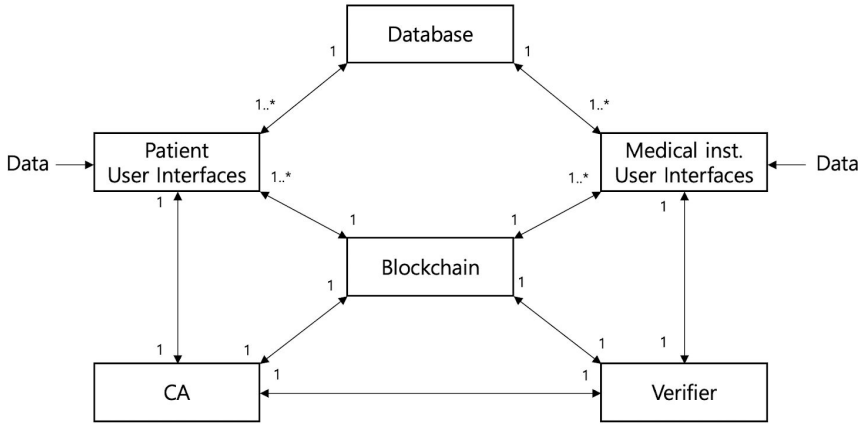
Identifier	**COM-02**

Defining quality	A user interface that allows the patient to access our BlockChain data.
Name	Patient UI
Attributes	a) Session ID b) User Name
Behaviors	a) Access through patient UI b) Check user's medical record
Relationships	a) BlockChain (COM-01) b) Certification Authority (COM-03) c) Database (COM-07)
Roles	Enable patients to access their medical records.

2.1.3 COM-03: Medical inst User Interfaces

This component is essential for making patient care data easy to insert and manage in medical institutions.

Identifier	**COM-03**
Defining quality	A user interface that allows the medical institution to access the blockchain
Name	Medical institution UI
Attributes	a) Session ID b) Medical institution name
Behaviors	a) Access through Institution UI b) Writing medical records c) Check user's medical record
Relationships	a) Blockchain (COM-01) b) Verifier (COM-05) c) Database (COM-06)
Roles	Enable medical institutions to store and view medical records.

State Group	

2.1.4 COM-04: CA

This component is an authorized certification system for patient medical data.

Identifier	**COM-04**
Defining quality	Authenticate if the information correct
Name	Certification Authority
Attributes	a) private key
Behaviors	a) Send private key
Relationships	a) BlockChain (COM-01) b) Patient UI (COM-02) c) verifier (COM-05)
Roles	Middle-man between patient interface and blockchain

2.1.5 COM-05: Verifier

This component is a system that verifies whether appropriate data are available when viewing medical records.

Identifier	**COM-05**
Defining quality	Verify that the information is correct

Name	Verifier
Attributes	a) validity
Behaviors	a) check if the data is corrected
Relationships	a) BlockChain(COM-01) b) Medical Institution UI (COM-03) c) Certification Authority (COM-04)
Roles	Middle-man between medical institution interface and blockchain

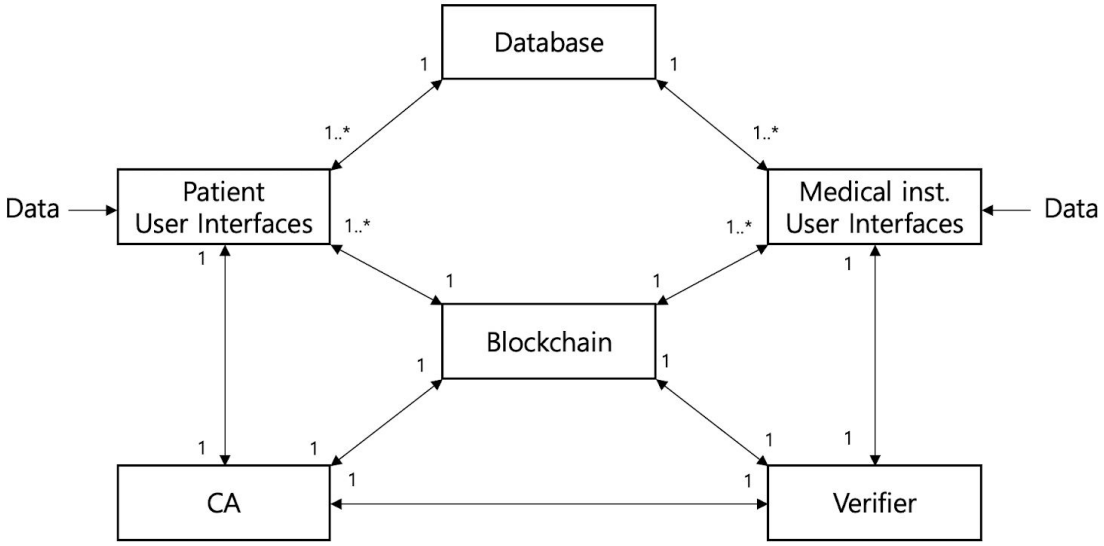
2.1.6 COM-06: Database

This component saves the information of the members so that the functions such as login can be activated.

Identifier	**COM-06**
Defining quality	Provides access to data stored in blockchain when correct login information is entered
Name	Database
Attributes	a) Login database b) Query translator
Behaviors	a) Receive login query b) Send login result c) Query translator d) Update information
Relationships	a) Patient UI (COM-02) b) Medical Institution UI (COM-03)
Roles	Check and save the information from users entered from interface

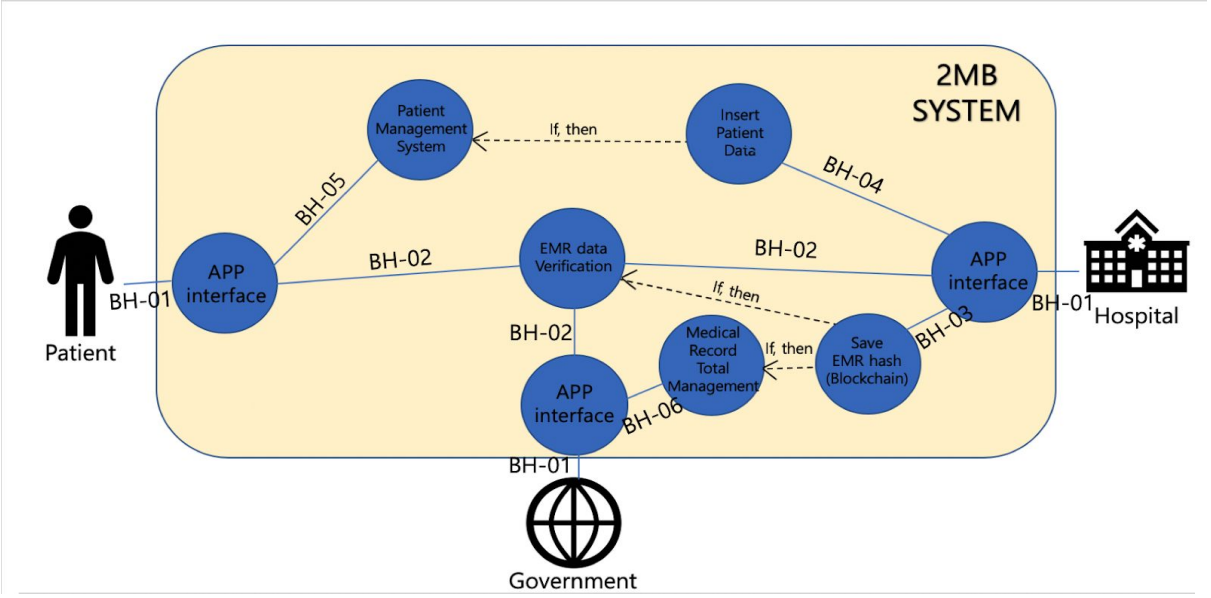
The Component Model is shown below

Figure 2-1 Component Model Diagram



2.2 Behavior Model

Figure 2-2 Use-Case Diagram - Behavior Model



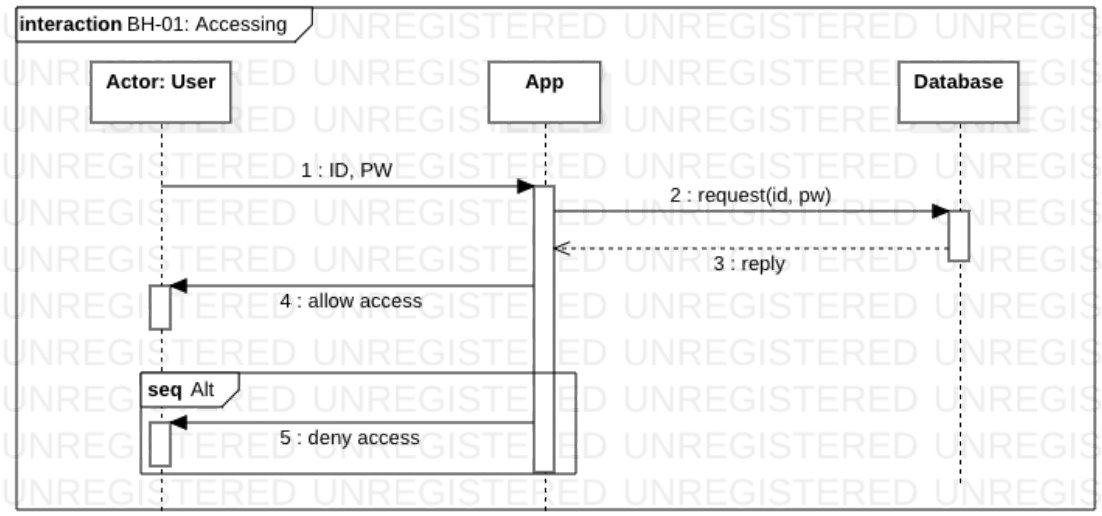
2.2.1 BH-01: ‘Accessing’

This behavior is triggered when users access the application.

Behavior Identifier	BH - 01
Trigger	User logs in.

Preconditions	Have to Download the application on an applicable device.
Postconditions	Depends on the type of the User there will be different types of interface appear to the User.
Inputs	User ID and Password.
Outputs	Access denied for unmatched ID & PW and allow access for correct ID & PW.

Figure 2-3 Sequence diagram - Accessing

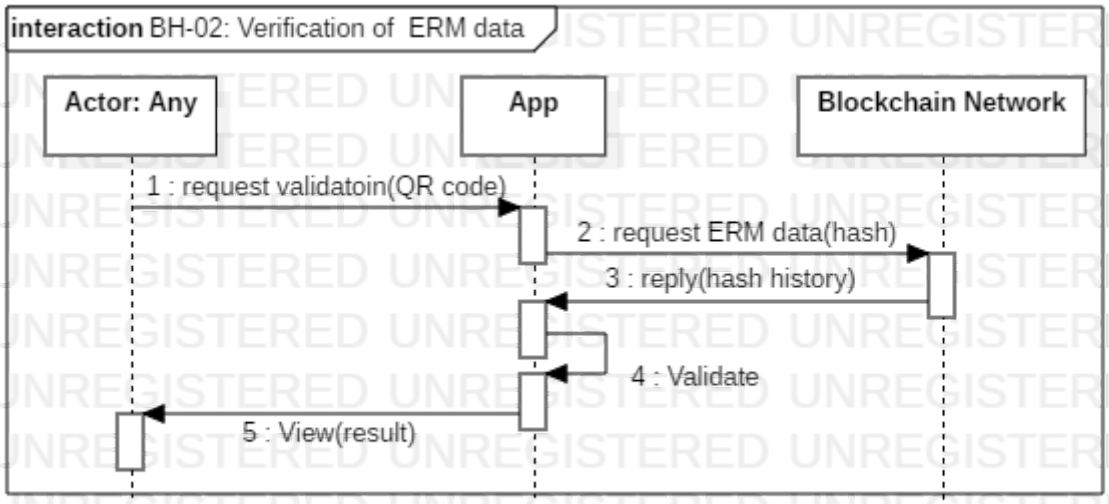


2.2.2 BH-02: 'Verification of EMR data'

This behavior is triggered when hospitals manage medical data.

Behavior Identifier	BH - 02
Trigger	Verification of EMR data
Preconditions	EMR data should be stored in BlockChain
Postconditions	Show the message that shows whether the EMR is true or not
Inputs	EMR data
Outputs	Whether EMR data is true or not

Figure 2-4 Sequence Diagram - Verification of ERM data

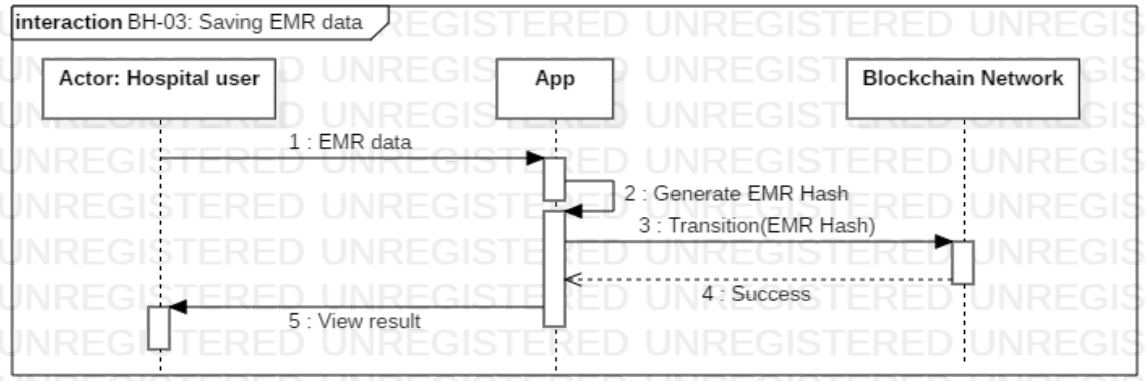


2.2.2.1 BH-03: 'Saving EMR data'

This behavior is triggered when saving EMR input Data in hospitals.

Behavior Identifier	BH - 03
Trigger	Saving EMR input Data
Preconditions	Depends on the type of the User there will be different types of interface appear to the User. Types of User: **Hospital Account**, Patient Account
Postconditions	ERM data should be stored in the blockchain.
Inputs	Any type of data.
Outputs	- System response which indicates whether the operation is normal or abnormal. - The input hash data will be in the blockchain DB.

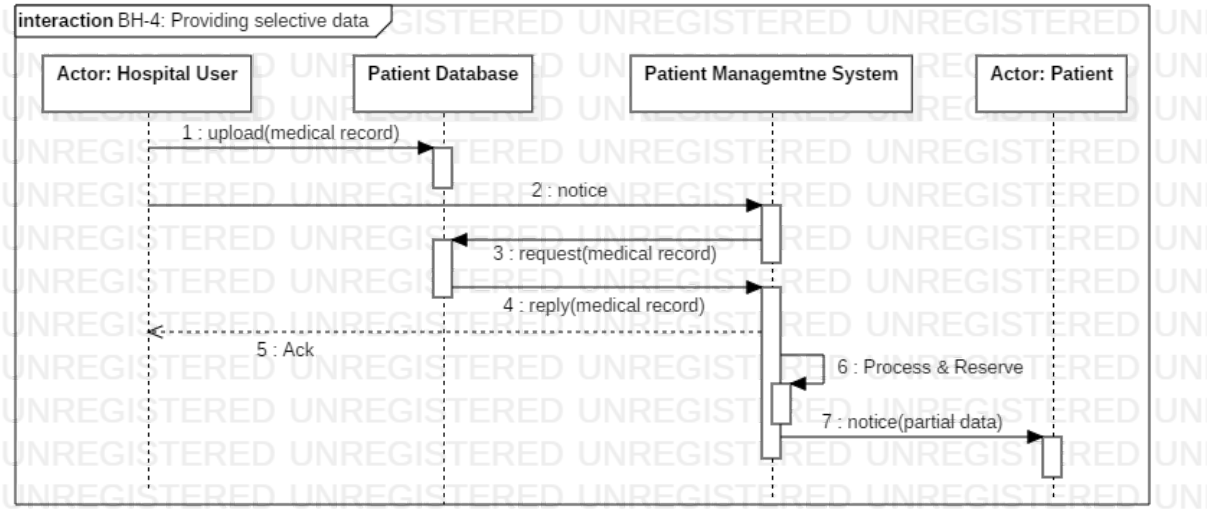
Figure 2-5 Sequence Diagram - Saving EMR data



2.2.2.2 BH-04: 'Providing selective data'

Behavior Identifier	BH - 04
Trigger	The data which is the entire patient's medical record to the system doesn't have to be served to the patient. The hospital selectively chooses the data they want to provide to the patient.
Preconditions	Depends on the type of the User there will be different types of interface appear to the User. Types of User: **Hospital Account**, Patient Account
Postconditions	message showing whether the data is successfully saved or not
Inputs	Any type of data that include medical record
Outputs	- System response which indicates whether the operation is normal or abnormal - Save the record to system DB

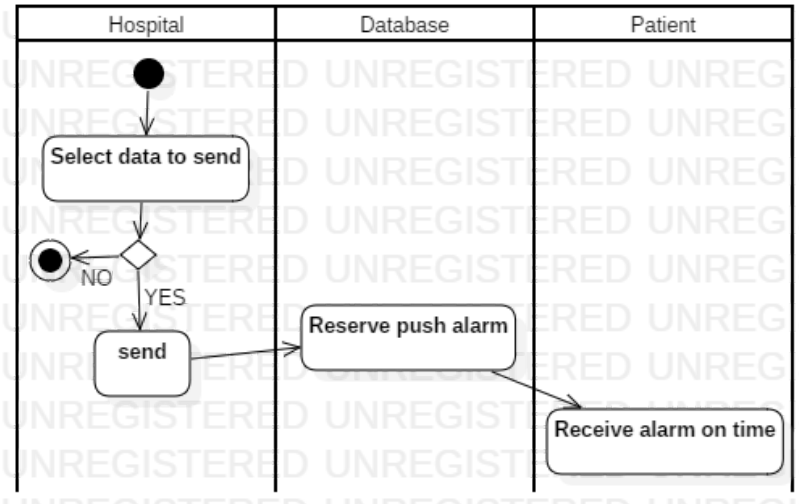
Figure 2-6 Sequence Diagram - Providing selective data



2.2.2.3 BH - 05: 'Providing medical service'

Behavior Identifier	BH - 05
Trigger	Providing medical service
Preconditions	Sending a message that is based on the data that the hospital gave us selectively
Postconditions	BH that puts up pop-up windows
Inputs	Patient's medical record
Outputs	Pop-up window will appear

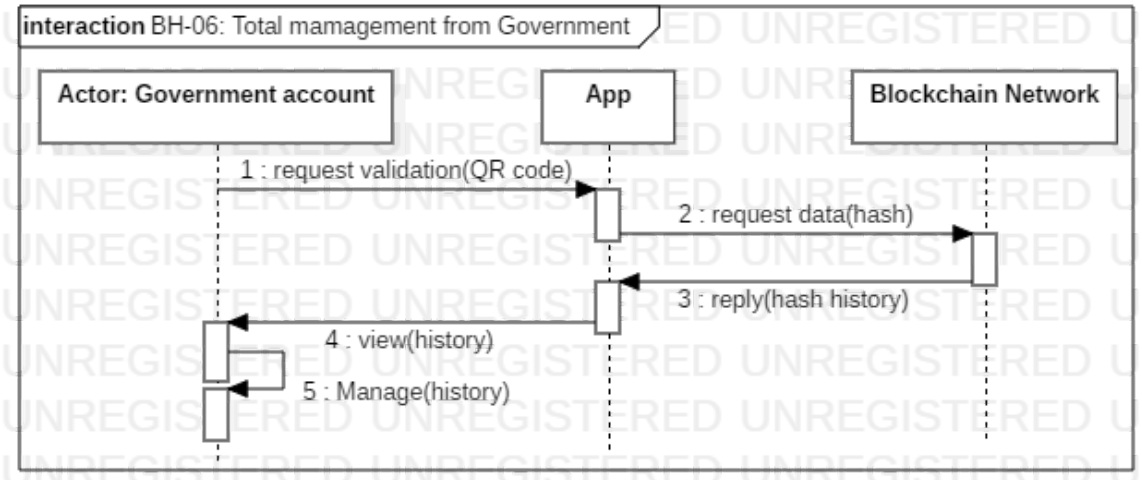
Figure 2-7 Activity Diagram - Providing medical service



2.2.3 BH - 06: ‘Total management from Government’

Behavior Identifier	BH - 06
Trigger	Total management from Government
Preconditions	EMR data is stored in the blockchain.
Postconditions	Detecting the change of medical record
Inputs	EMR hash
Outputs	Whether changes happen or not

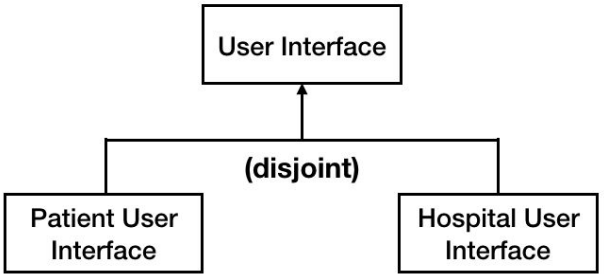
Figure 2-8 Sequence Diagram - Total management from Government



2.3 Enterprise Model

2.3.1 Component Classification

Figure 2-9 Component Classification



2.3.2 Behaviors In Service

Figure 2-10 Service(Hospital) Component Classification

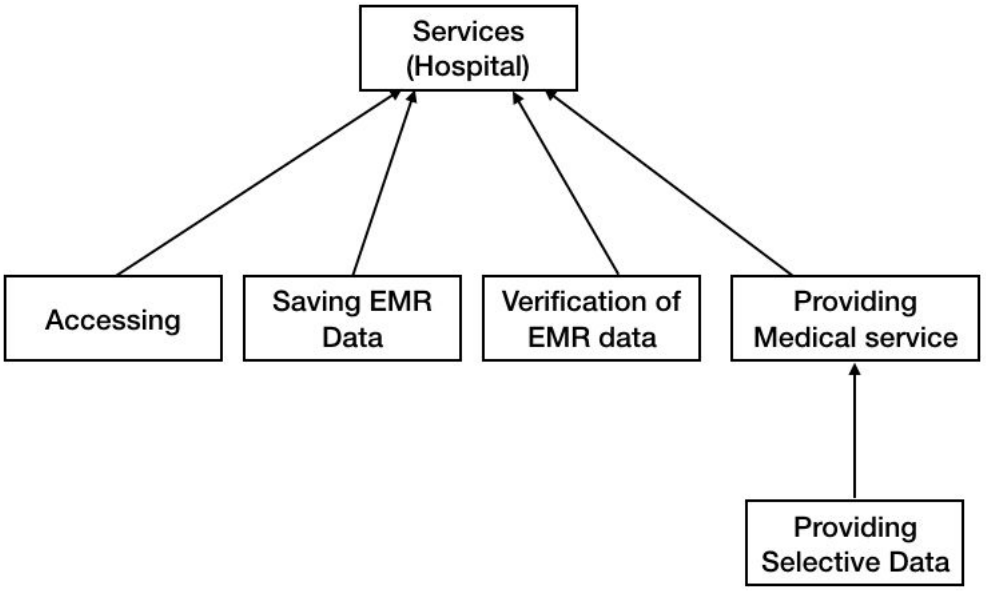
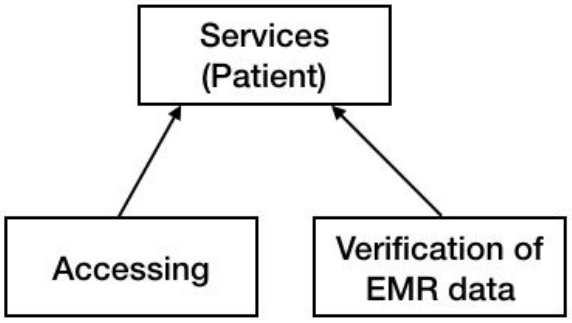


Figure 2-11 Service(Patient) Component Classification



3 System Design

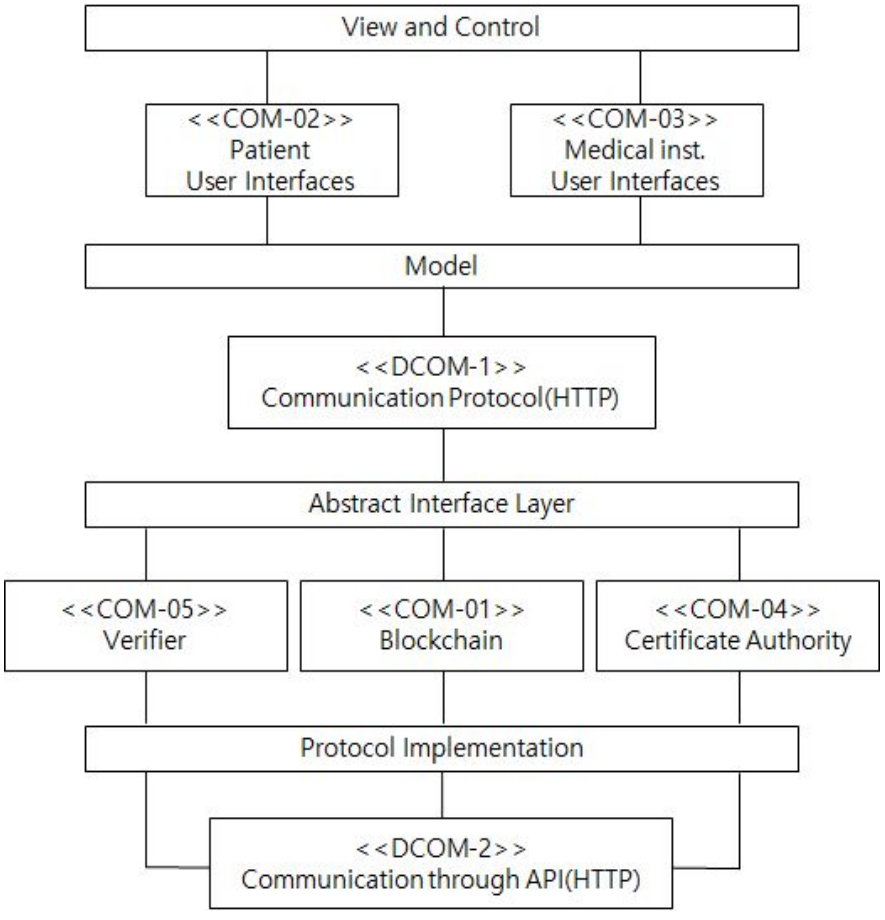
3.1 Architectural Views

The architectural views provide a high-level blueprint for implementation from different perspectives. They act as a bridge that transforms from system analysis result in implementation.

3.1.1 System Topology

The system topology describes the logical software framework that shows and organizes the software components introduced in the SSAD 2.1 Component Model into a hierarchy of layers.

Figure 3-1 System Topology



3.1.2 Component Specifications

Component Identifier	DCOM-1
Defining Quality	A protocol used to verify the identity of users and request access to Blockchain.
Name	Communication Protocol
Attributes	a) Server IP address (Domain) b) Server Port number
Behaviors	a) Process user requests b) Deliver user validity
Relationships	a) COM-1 Blockchain b) COM-2 Patient User Interface c) COM-3 Medical institute User Interface d) COM-4 Certificate Authority
State Groups	
Constraints	
Implementation	Express JS Web Interface Implementation (Node.JS)

Component Identifier	DCOM-2
Defining Quality	A protocol which handles verifying processes and an Access Control List(ACL)
Name	Communication through API
Attributes	a) Server IP address (Domain) b) Server Port number

Behaviors	a) Process authentication request b) Manage Access Control List(ACL)
Relationships	a) COM-1 Blockchain b) COM-4 Certificate Authority c) COM-5 Verifier
State Groups	
Constraints	Only authenticated users can access Blockchain
Implementation	a) Express JS Web Interface Implementation (Node.JS) b) TypeScript Web Interface Implementation

3.1.3 Framework and Protocol Specifications

This section describes the frameworks and protocols used in our proposed architecture and how they are integrated into our application development context.

Android Studio (JAVA)

Android studio with the highest SDK version will be selected to develop this application. It will adapt the highest updated version in order to support any further updates for complexness and flexibility. The simple design will be adapted to make the User Interface as simple as possible for the users since our clients are mainly patients and hospitals.

Node.js

Node.js a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

TCP/IP

This network protocol is used as a communication protocol between different machines and servers within the LAN and accessible through the Internet infrastructure.

HTTP

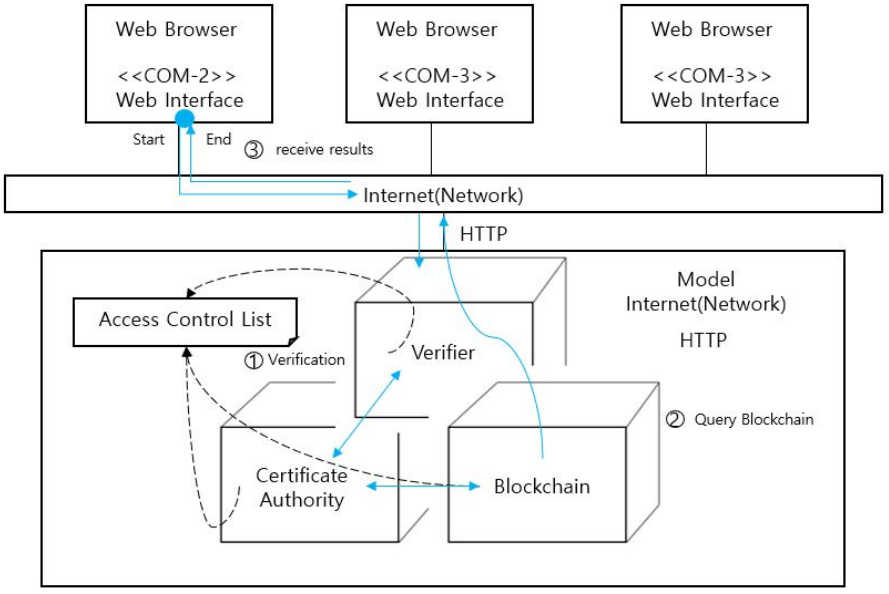
HTTP is the underlying protocol used by the World Wide Web and this protocol defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

Blockchain

A blockchain is a decentralized, distributed, and oftentimes public, digital ledger that is used to record transactions across many computers so that any involved record cannot be altered retroactively, without the alteration of all subsequent blocks.

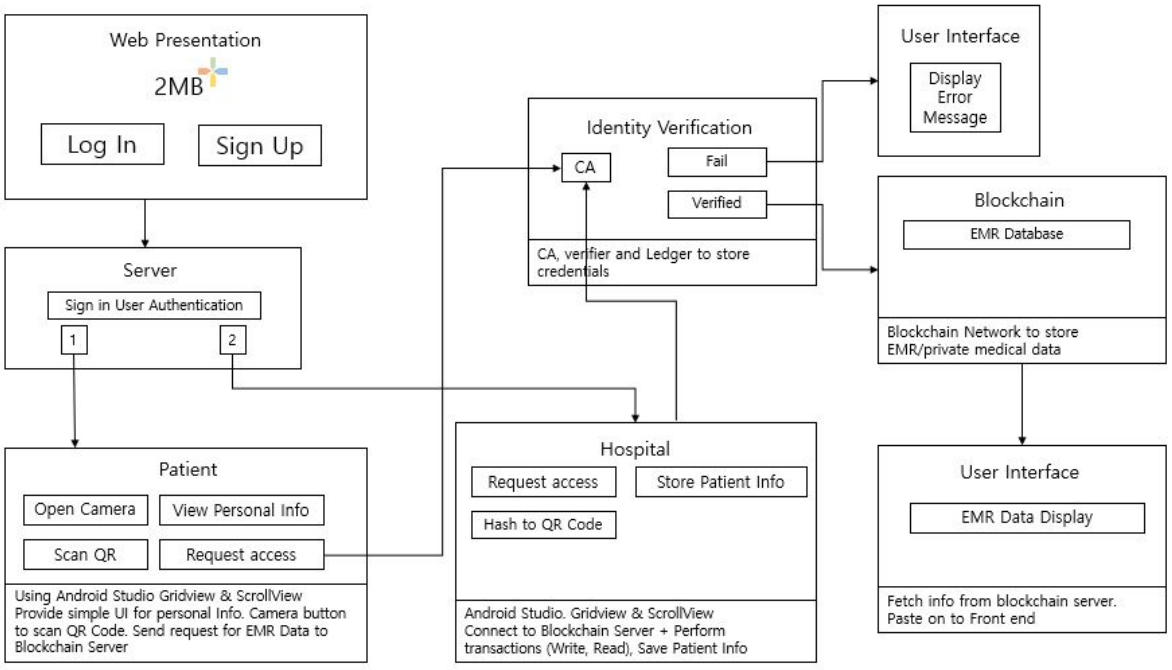
3.1.4 System Deployment View

Figure 3-2 System Deployment Diagram



3.1.5 Logical Component View

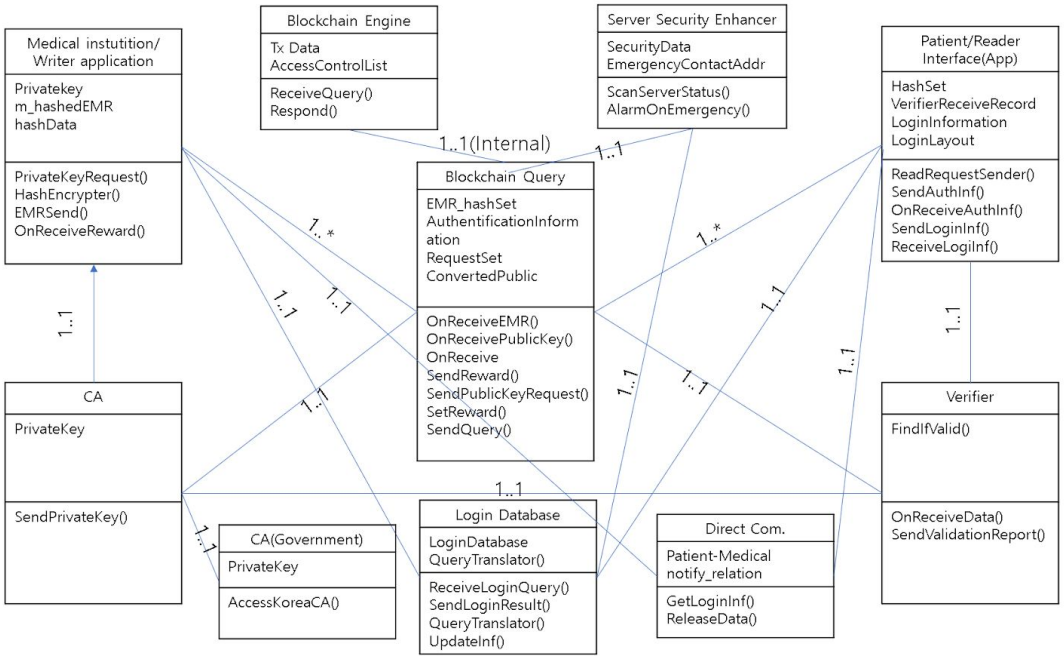
Figure 3-3 Logical Component View



3.2 Object Model

As the model of the project is held for the project needs all the connection internally without a complicated class relation model, the object model would be concluded with the figure below.

Figure 3-4 Object Model View



3.3 Operations Model

3.3.1 Critical Algorithms

No critical algorithm used in 2MB Project.

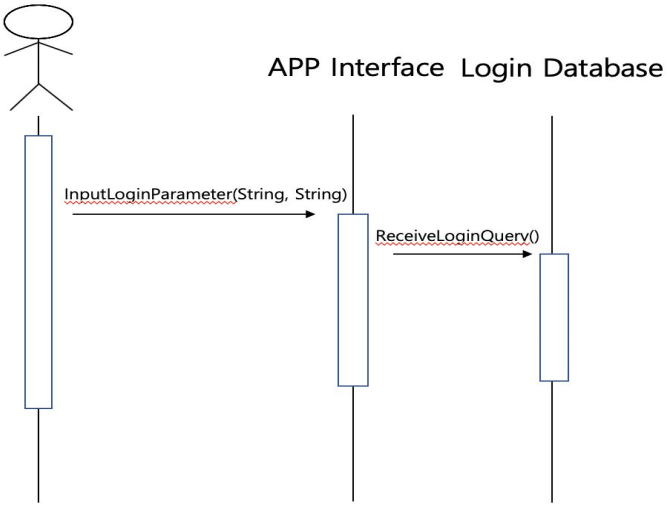
3.3.2 Operation Specifications

3.3.2.1 Access through App Interface

Identifier	**OP-01**
Initiator	SSAD 2.2.1 BH-01 Accessing
Passed parameters	a) User ID b) Password
Return Values	Access Token, Different Mode Page(Hospital, Patient) according to the type of user

Exception handling	Login Failed, invalid login page displayed
Guards	Parameters cannot be empty
Validation	Http Request to Login API to validate the user login
Message	login request { id, password }
Exits	
Constraints	
Relates to	SSAD 2.2.1 BH-01

Figure 3-5 Interaction for access OP-01



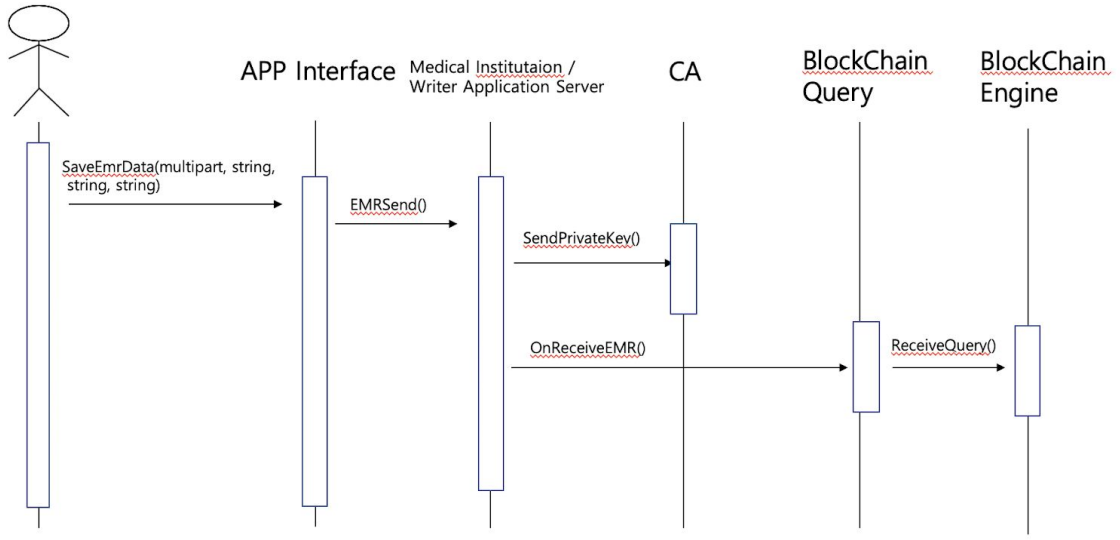
3.3.2.2 Hospital Mode

3.3.2.2.1 Saving EMR input Data into Server and BlockChain Engine

Identifier	**OP-02**
Initiator	SSAD 2.2.2.1 BH-01 Saving EMR data
Passed parameters	a) EMR Data b) Patient Information c) Doctor Information d) access token
Return Values	Result Page
Exception handling	Error page displayed with reasons shown

Guards	Parameters cannot be empty
Validation	a) Patient Information and doctor information must exist. b) Validate access token is valid.
Message	saving request { request }
Exits	
Constraints	
Relates to	SSAD 2.2.2.1 BH-01

Figure 3-6 Interaction for Saving EMR Data OP-02

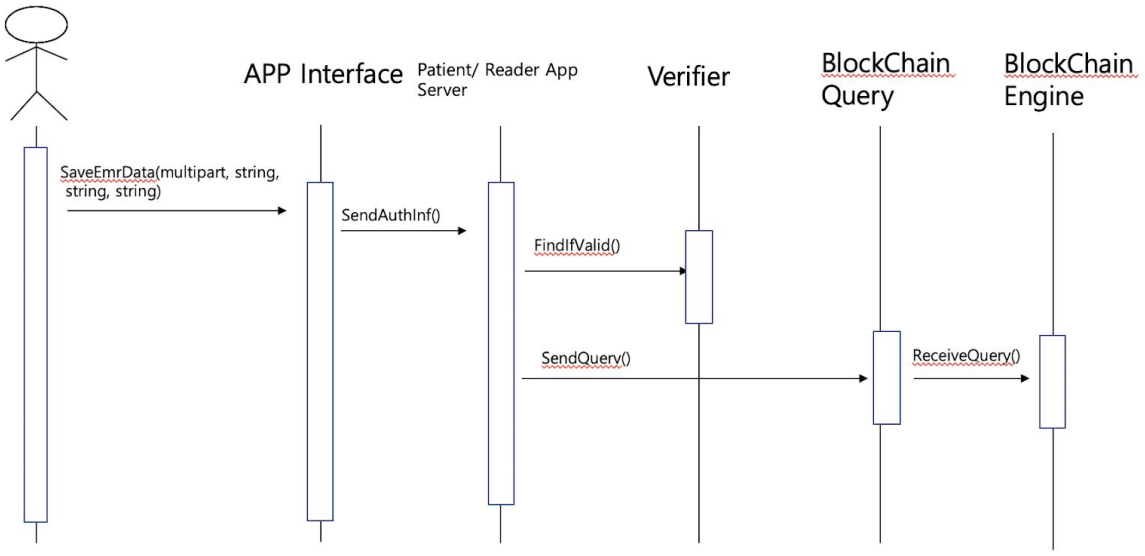


3.3.2.2.2 Authenticate EMR data

Identifier	**OP-03**
Initiator	SSAD 2.2.2 BH-02 Blockchain engine
Passed parameters	a) EMR Data b) access token
Return Values	Authentication Result Page
Exception handling	Error page displayed with reasons shown
Guards	Parameters cannot be empty
Validation	validate access token.

Message	authenticate request { request }
Exits	
Constraints	
Relates to	SSAD 2.2.2 BH-02

Figure 3-7 Interaction for Authentication EMR Data OP-03



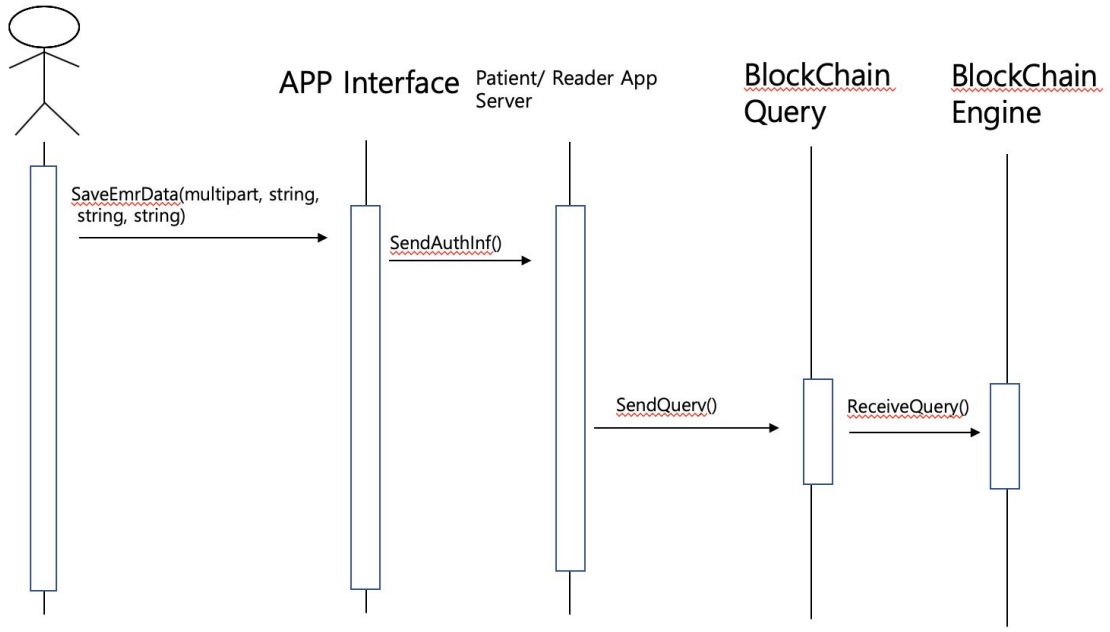
3.3.2.3 Patient Mode

3.3.2.3.1 Recommend appropriate medical services

Identifier	**OP-04**
Initiator	SSAD 2.2.2.3 BH - 05: ‘Providing medical service’
Passed parameters	a) access token
Return Values	send push notification with appropriate medical service information
Exception handling	No push notification sended when error occurred
Guards	Parameters cannot be empty
Validation	validate access token

Message	recommendation request { request }
Exits	
Constraints	
Relates to	SSAD 2.2.2.3 BH-05

Figure 3-8 Interaction for Recommendation Service OP-04

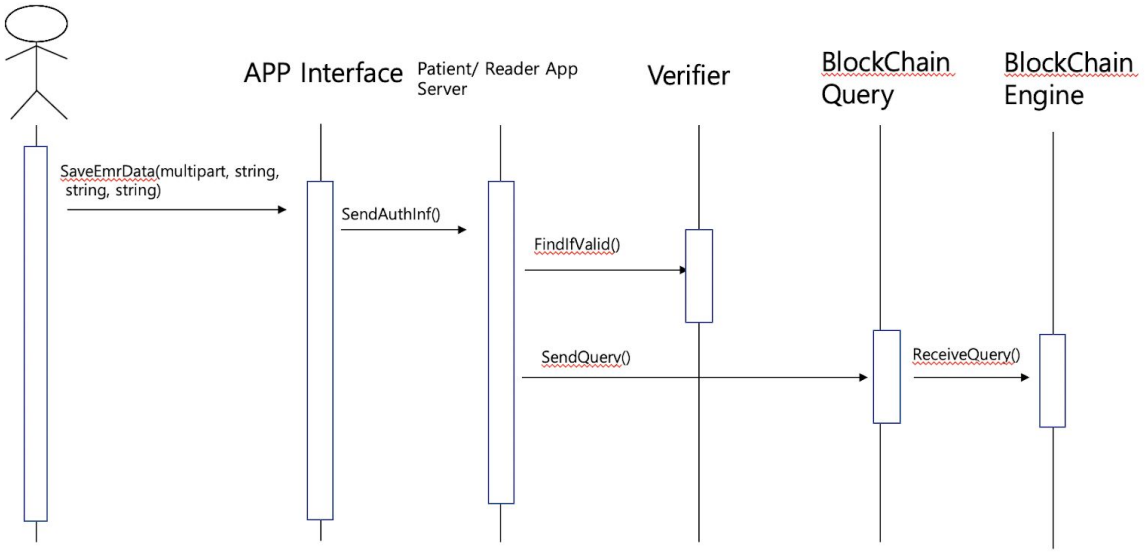


3.3.2.3.2 Authenticate EMR data

Identifier	**OP-05**
Initiator	SSAD 2.2.2 BH-02 Blockchain engine
Passed parameters	c) EMR Data d) access token
Return Values	Authentication Result Page
Exception handling	Error page displayed with reasons shown
Guards	Parameters cannot be empty
Validation	validate access token.
Message	authenticate request { request }

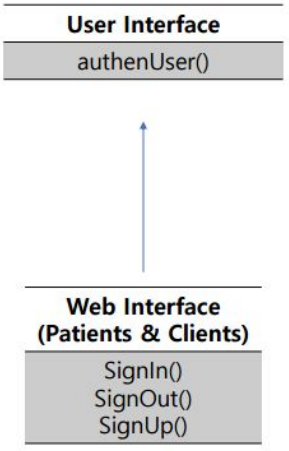
Exits	
Constraints	
Relates to	SSAD 2.2.2 BH-02

Figure 3-9 Interaction for Authenticate EMR Data OP-05



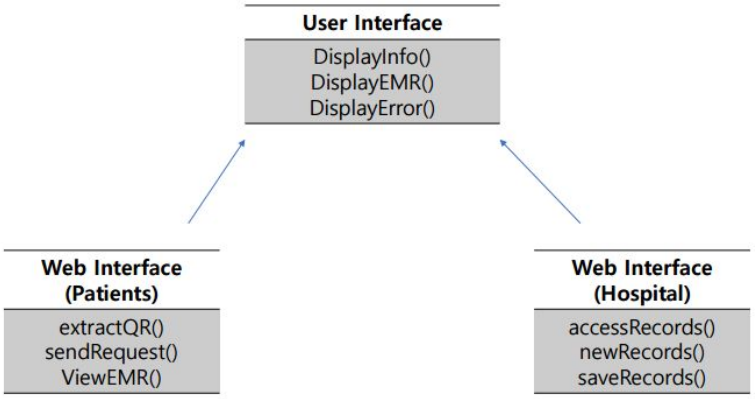
3.4 Class Model

Figure 3-10 Class Model No.1



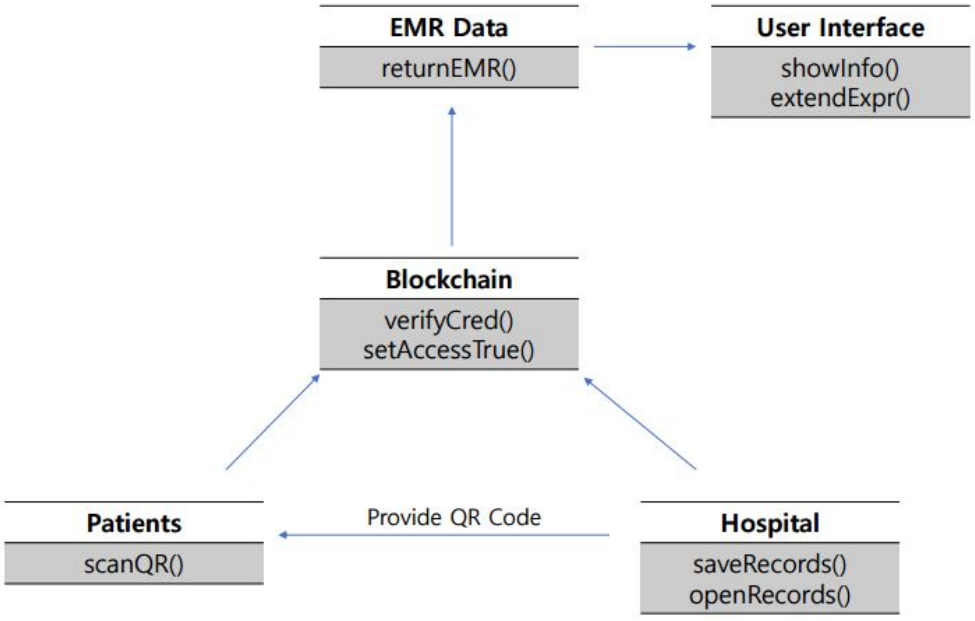
Class Model No.1

Figure 3-11 Class Model No.2



Class Model No.2

Figure 3-12 Class Model No.3



Class Model No.3

3.5 Configuration Model

Hycon in its latest version (2.0 at the current status) has to be downloaded from https://hycos.io/. Hycon's development is two-front that is committed to increasing scalability and throughput on the blockchain and providing functional and straightforward user platforms for everyone to adopt.

The Android Studio in version 3.5.1 or higher is available from https://developer.android.com/ has been chosen to install on the current version of the

Android device. The Android Studio provides the fastest tools for building apps on every type for the device, allows to generate multiple build variants for different devices from a single project, etc.

To support Android apps Android Bundle, which is a new delivery format will help to deliver an optimized version of APK by including what user needs, reducing the app size.

The latest version of SQL Server is available from https://www.postgresql.org/download/. The login must be associated with a SQL Server database user. The user account is used to control activities performed in the database. If no user account exists in a database for a specific login, the user that is using that login cannot access the database even though the user may be able to connect to SQL Server. The single exception to this situation is when the database contains the "guest" user account.

4. Common Definition Language

EMR

Electronic Medical Record (EMR) is a computerized chart of patients on traditional paper since the mid-1990s when personal computer PCs became common.
The major items include the patient's personal information, medical history, examination results, treatment results, surgical records, hospitalization records, and outpatient care.

BM QR code

Blockchain Medical(BM) QR code is a hash value which means a unique document ID. the hospital of 2MB network includes the BM QR code into the EMR document and the client of the hospital can submit BM QR code to his or her application. After that, the patient can examine a result of tracking the issued EMR document’s edit record.

Private Blockchain

Private blockchain is also called "Permissioned Ledger." There are pre-designated participants who can participate in the reading, writing and agreement processes, and specific subjects can be newly added or removed as necessary. In addition, depending on design purposes, private blockchains can be designed in different versions.

Front and Back End

In software engineering, the terms front end and back end refer to the separation of concerns between the presentation layer (front end), and the data access layer (back end) of a piece of software, or the physical infrastructure or hardware. In the client-server model, the client is usually considered the front end and the server is usually considered the back end, even when some presentation work is actually done on the server itself.

Hash

Hash function refers to a function that maps data of a fixed length to any data with any length. The value of a fixed-length resulting from the application of these hash functions is called a hash value.

5. Appendix

A. Reference

[1] Hycon 2.0 - Blockchain, https://hycon.io/ko/home_kr/#whitepaper
[2] Android Studio - Google Developers,
https://developer.android.com/studio/intro/?gclid=CjwKCAjwxt_tBRAXEiwAENY8hZQvHQ1w4X5rmmjLVeXjzFaR-O24gbqrozzlf-jcOFD11-xKZZtvSBoCLdIQAvD_BwE
UML Links :
http://www.1st-spot.net/topic_methodologies.html
http://sds.hss.cmu.edu/courses/Syllabi/ids/271/umlfaq.asp
http://www.rational.com/uml/resources/documentation/index.jsp
Node.js: https://nodejs.org/ko/
TypeScript: https://www.typescriptlang.org/
Express.js: https://expressjs.com/ko/

B. Vendor documents

No vendor documents identified.