

# Software Requirement Specifications (SRS)

MediChA.I.n

Ver. 1.0



201411295 이상훈

201511272 양재민

201511295 조범석

## Contents

<b>1</b>	<b>소개 (Introduction)</b> .....	<b>4</b>
1.1	목적 (Purpose) .....	4
1.2	범위 (Scope) .....	4
1.3	용어 (정의, 머리글자) 및 약어 정의 (Definitions, acronyms and abbreviations) 5	
1.4	참고자료 (References) .....	9
1.5	개요 (Overview) .....	10
<b>2</b>	<b>전체 시스템 개요 (Overall description)</b> .....	<b>10</b>
2.1	제품 관점 (Product perspective) .....	10
2.1.1	시스템 인터페이스 (System interfaces) .....	10
2.1.2	사용자 인터페이스 (User interfaces) .....	10
2.1.3	하드웨어 인터페이스 (Hardware interfaces) .....	11
2.1.4	소프트웨어 인터페이스 (Software interfaces) .....	11
2.1.5	통신 인터페이스 (Communications interfaces) .....	12
2.1.6	메모리 제약사항 (Memory constraints) .....	12
2.1.7	운영 (Operations) .....	12
2.1.8	사이트 적용 요건 (Site adaption requirements) .....	12
2.2	제품 기능 (Product functions) .....	13
2.3	사용자 특성 (User characteristics) .....	14
2.4	제약사항 (Constraints) .....	15
2.5	가정 및 의존성 (Assumptions and dependencies) .....	15
2.6	단계별 요구사항 (Apportioning of requirements) .....	15
<b>3</b>	<b>상세 요구사항 (Specific requirements)</b> .....	<b>15</b>
3.1	외부 인터페이스 요구사항 (External interface requirements) .....	16
3.1.1	사용자 인터페이스 (User interfaces) .....	16

<b>3.2</b>	<b>System features</b> .....	18
3.2.1	Web System .....	18
3.2.2	Blockchain .....	24
3.2.3	Deep learning.....	26
<b>3.3</b>	<b>성능 요구사항 (Performance requirements)</b> .....	28
<b>3.4</b>	<b>설계 제약사항 (Design constraints)</b> .....	29
<b>3.5</b>	<b>소프트웨어 시스템 속성 (Software system attributes)</b> .....	29
3.5.1	신뢰성 (Reliability).....	29
3.5.2	가용성 (Availability).....	29
3.5.3	보안성 (Security).....	30
3.5.4	유지 보수성 (Maintainability).....	30
3.5.5	이식성 (Portability).....	30
<b>3.6</b>	<b>이외 요구사항 (Other requirements)</b> .....	30

# 1 소개 (Introduction)

## 1.1 목적 (Purpose)

- 프로젝트 진행과 프로젝트 개발에 어려움이 없도록 요구사항을 뽑아내고 프로젝트에 사용할 자원들에 대해 명세를 하기 위함이다.
- 이 명세서의 대상은 제품 개발을 진행할 개발자와 제품 진행 총괄 말을 프로젝트 담당자, 제품을 사용할 사용자이다.

## 1.2 범위 (Scope)

- 소프트웨어의 이름은 **MediChA.I.n**이고, 개인 의료 데이터를 블록체인에 저장하고 블록체인을 통해 접근할 수 있는 이미지를 학습하는 딥 러닝 시스템을 만들어 새로 들어오는 의료 이미지 데이터의 신뢰성을 계산하고 질병 발생 여부를 예측하여 기존 의료 데이터와 예측 결과를 새로운 블록에 담아 블록체인을 구성하는 시스템이다.
- 이 소프트웨어는 블록체인을 이용해 의료 데이터를 저장하려는 의도는 많은 곳에서 일어나고 있지만, 아직 실용성이 뚜렷하지 않은 상황에서 블록체인에 이상한 데이터가 들어올 경우 딥 러닝으로 거르거나, 경고메시지를 보내줄 수 있다는 장점이 있다. 또한 공공 데이터 사용을 통한 초기 블록 데이터 구축을 진행할 수 있다.
- 초기에는 공공 의료 이미지 데이터로 학습을 진행하고 학습이 완료된 이후에는 입력되는 데이터를 블록이 생성되는 과정에서 분석 및 예측 또는 검증하여 모델을 개선시키고, 결과 데이터를 예측해 블록을 만들어 기존 블록체인에 연결해준다.
- 사용자들은 웹 사이트를 통해 접속해 의료 데이터와 이미지를 입력시킬 수 있으며, 의료 데이터는 블록체인 상에 보관되어 위, 변조가 안되는 신뢰성 있는 데이터로 네트워크 참여자에게 권한에 맞게 공개된다.
- 학습 모델들은 블록체인 노드로 보내져 블록체인 데이터를 확인해 학습되고 모두 학습된 모델들은 다시 통합 딥 러닝 서버로 돌아가 각각의 학습된 모델들은 다시 종합된 학습 모델로 개선된다.
- 노드 별로 지정 블록 데이터를 통해 학습을 하고 통합학습모델이 있는 서버와 모델 만을 통신해 주고받기 때문에 통합학습모델이 있는 서버는 데이터를 갖지 않고도 학습을 할 수 있게 된다. (Federated Learning)

### 1.3 용어 (정의, 머리글자) 및 약어 정의 (Definitions, acronyms and abbreviations)

#### - Acronyms

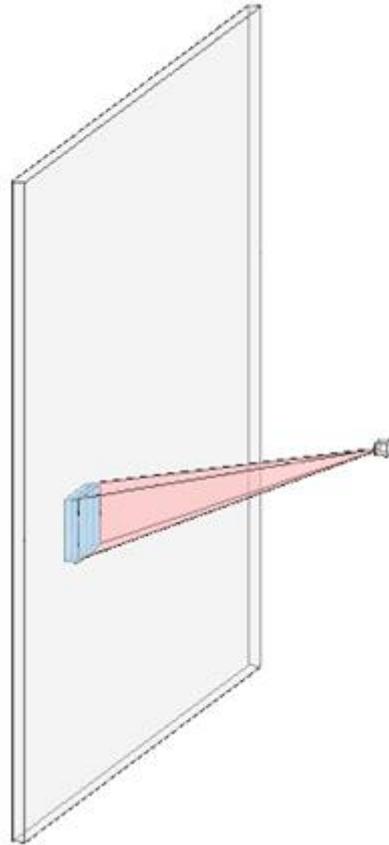
- SW: Software
- IPFS: Inter Planetary File System

#### - Definitions:

- Blockchain: 블록이라고 하는 소규모 데이터들이 P2P 방식을 기반으로 생성된 체인 형태의 연결고리 기반 분산 데이터 저장 환경에 저장해 누구라도 임의로 수정할 수 없고 누구나 변경의 결과를 열람할 수 있는 분산 컴퓨팅 기술 기반의 원장 관리 기술
- Node(노드): Blockchain 시스템에서 P2P 방식 네트워크의 한 일원으로 구성원을 뜻한다. Blockchain은 Node 간의 합의를 통해 블록을 생성하고 체인을 구성한다.
- IPFS: 분산형 파일 시스템에 데이터를 저장하고 인터넷으로 공유하기 위한 프로토콜
- 체인코드 (스마트 컨트랙트): 스마트 컨트랙트는 계약 당사자가 사전에 협의한 내용을 미리 프로그래밍하여 전자 계약서 문서 안에 넣어두고, 이 계약 조건이 모두 충족되면 자동으로 계약 내용이 실행되도록 하는 시스템을 일컫는다. 하이퍼레저 패브릭에서는 이를 체인코드라 부른다.
- 주소: Blockchain 상에서 개인 고유성을 나타내기 위한 16진수 문자열이며, 사용자를 식명화해서 블록체인 상에 데이터를 올린다. 이를 통해 개인 정보를 보호할 수 있다.
- Server: 클라이언트에게 네트워크를 통해 정보나 서비스를 제공하는 컴퓨터 시스템으로 컴퓨터 프로그램(server program) 또는 장치(device)를 의미. 이 문서에서는 Server Program까지 통틀어서 일컫는 단어이다.
- Deep Learning: 인공 신경망에서 발전한 형태의 인공지능으로, 뇌의 뉴런과 유사한 정보 입출력 계층을 활용하여 데이터를 학습한다. 사람이 가르치지 않아도 컴퓨터가 스스로 사람처럼 학습할 수 있는 인공지능 기술.
- Federated Learning: 지금까지 인공지능을 만드는 방식은 데이터를 모두 한 곳에 끌어모아 모델이 학습하도록 훈련시키는 형태였다. Federated

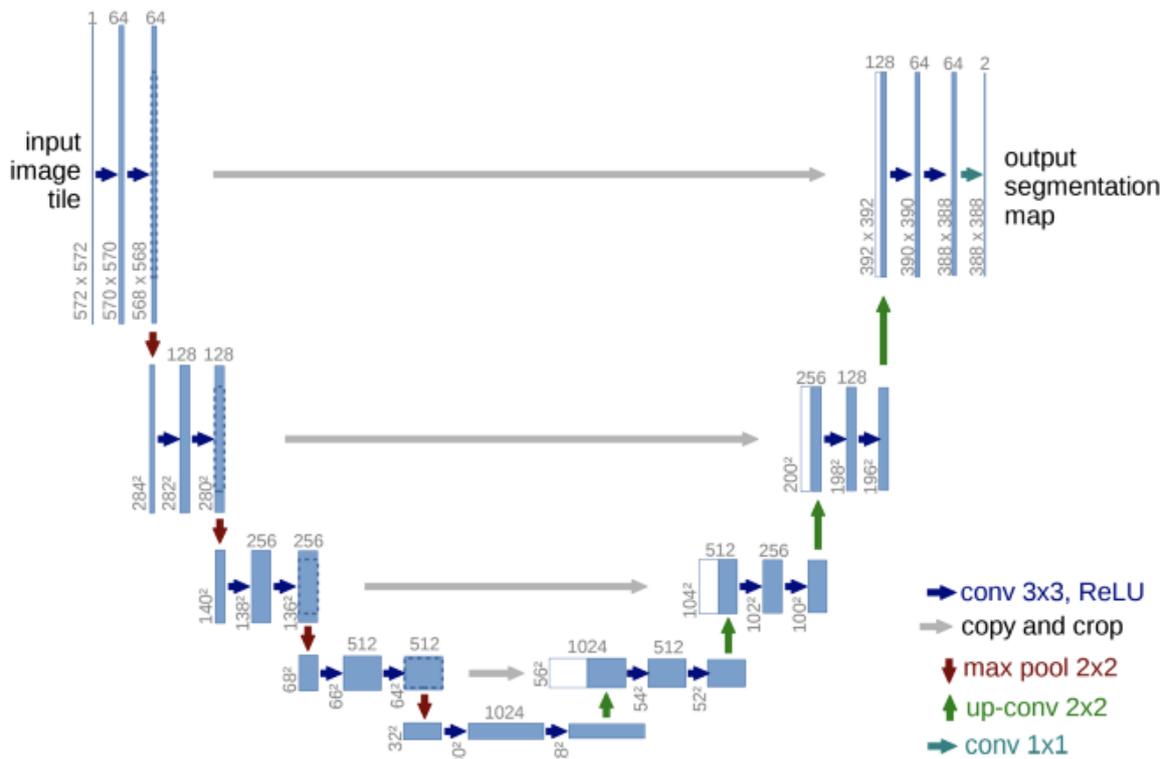
learning은 데이터를 하나의 서버로 끌어 모으는 것이 아니라 모델을 각 데이터가 저장된 곳으로 보낸다. 각 저장소는 마스터 모델을 자신의 데이터에 적용에 활용하고 저장소의 모델이 배운 업데이트된 내용만 서버로 보낸다. 데이터는 전혀 이동하지 않으며, 서버는 각 저장소의 데이터를 보지 못하는 방식.

- Validation: 모델이 학습을 하게 되면 training data에 overfitting되는 결과가 나올 수 있다. 이를 해결하기 위해서 validation data를 이용하여 모델이 얼마나 fitting이 되었는지 확인하는 과정.
- Web System: Web Server와 Web Application Server로 구성되는 Client-side 시스템.
- Web Server: 클라이언트에게 시스템의 정보를 보여주는 정적 서버.
- Web Application Server: 동적 웹 서버. 데이터를 업로드하거나 이미지를 분석한다.
- Convolution Neural Network
  - ◆ 합성곱 신경망 신경망이라고 하고 이미지 인식에서 많이 사용 된다.
  - ◆ 고양이의 실험에서 아이디어를 얻었다.
  - ◆ 이미지를 입력으로 받으면 일부분만 처리하기 위해 filter라는 개념을 사용한다.
  - ◆ Filter가 하는 일은 궁극적으로 한 값을 만든다.
  - ◆ 그림을 점차적으로 보듯이 filter를 sliding시킨다.
  - ◆ Stride: filter를 sliding할 때 몇 칸씩 띄워서 sliding 할 지 나타내는 말이다.



- U-net: 데이터를 학습하고 예측하는 모델.
  - ◆ U-net은 의료이미지 분야에서 Image segmentation을 목적으로 제안된 Fully-Convolutional Network 기반 모델이다.
  - ◆ U-net은 전반적인 컨텍스트 정보를 얻기 위한 네트워크와 정확한 localization을 위한 네트워크가 대칭형태로 구성되어 있다.
  - ◆ Contracting path와 expanding path 구성으로 23-layers fully convolution network 구조이다.
  - ◆ Contracting path
    - 입력 이미지의 context 포착을 목적으로 구성되었다.
    - 두 번의 3x3 convolution layer(activation function: Relu)
    - 2x2 max-pooling하여 feature map의 크기가 절반으로 줄어든다.
    - 각 Down-sampling마다 채널수를 두배로 늘린다.
  - ◆ Expanding Path

- 세밀한 localization을 위한 구성이다.
- 높은 차원의 채널을 가지는 up-sampling 앞은 레이어의 feature 맵을 결합한다.
- 즉 contracting path의 최종 feature map으로부터 보다 높은 해상도의 segmentation을 얻기 위해 몇 차례의 up-sampling을 진행한다.
- 각 step 마다 2x2 up-convolution을 수행한다.
- 3x3 convolution 연산을 두번씩 반복한다.(activation function: relu)
- 각 up-sampling마다 채널수를 반으로 줄인다.
- 마지막 레이어에 1x1 convolution 연산을 수행한다.



Term	Description
1.1 Sign In	로그인하여 서비스를 이용한다. 환자 회원인 경우 환자 고유 주소를 보여준다.
1.2 Sign Out	로그인이 되어있는 상태에서 로그아웃 한다.
1.3 Sign Up	회원 가입한다. 환자에게는 환자 고유 주소를 부여하고 이를 보

	여준다.
1.4 Upload Image	이미지를 업로드 한다. 이 행동은 의료진만 가능하다.
1.5 Detect Strange	이상한 이미지가 들어왔는지 감지한다.
1.6 Get Info	의료 정보를 요청해 가져온다.
1.7 Authorize	로그인할 때의 회원 정보를 바탕으로 시스템 접근 권한을 얻는다.
2.1 Check Authorized Data	의료진 권한을 가진 사용자가 환자 고유 주소를 바탕으로 환자의 개인 데이터를 조회한다.
2.2 Check Self Data	의료진이 아닌 환자인 사용자가 환자 고유 주소를 이용하여 자신의 데이터를 조회한다.
2.3 Upload Data	의료진 권한이 있는 사용자가 환자 고유 주소를 이용하여 환자의 데이터를 업로드한다.
2.4 Upload To IPFS	IPFS에 이미지를 업로드 한 뒤 사진에 이미지에 해당하는 해시 테이블을 받는다.
3.1 Send Model	딥러닝 서버에서 각 노드에게 모델을 전송한다.
3.2 Learning Model	각 노드 별로 모델을 학습한다.
3.3 Send Model from Node to Server	각 노드에서 정해진 블록들을 모두 학습하면 모델을 전송한다.
3.4 Modify Model	딥러닝 서버에서 각 노드에서 보내온 모델들을 종합한다.
3.5 Validate Model	딥러닝 서버에서 종합한 모델을 검증한다.
3.6 Check Loss	서버에서 각 노드에서 나온 validation loss를 이전의 validation loss와 비교하여 master 모델을 교체할지 결정하고 특정 횟수 만큼 개선이 이루어지지 않으면 training을 중지한다.
3.7 Fetch Model	딥러닝 서버에서 웹서버에 분석 모델을 전송한다.

## 1.4 참고자료 (References)

- Wikipedia (<https://ko.wikipedia.org/>)
- IEEE Std 830 – 1998
- Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, Jason Roseland, TOWARDS FEDERATED LEARNING AT SCALE: SYSTEM DESIGN, 1902.01046v2, 2019

## 1.5 개요 (Overview)

- 이 명세서에는 IEEE Std. 830-1998에 따라 작성되어 있으며 이후 전체 시스템을 살펴보고 (2. 전체 시스템 개요), 상세 요구사항 (3. 상세 요구사항)을 분석하며 기능적 요구사항, 비기능적 요구사항 등을 살펴보도록 한다.

## 2 전체 시스템 개요 (Overall description)

### 2.1 제품 관점 (Product perspective)

- 기존 의료 이미지 데이터로 학습된 딥 러닝 모델이 새로운 의료 이미지 데이터를 분석하고 그 결과를 포함하여 새로운 블록을 생성하는 시스템
- 기존의 제품들보다 블록체인을 활용하기 때문에 신뢰성과 보안성이 올라가고, 딥러닝을 통해 의사가 진단하는 부분에서 도움을 준다.

#### 2.1.1 시스템 인터페이스 (System interfaces)

- Web Server, Deep learning Model, Blockchain system
- Web System: 사용자가 웹에 사진을 업로드하거나 원하는 정보를 웹 상에서 사용자에게 보여준다. 입력된 데이터는 Blockchain에 입력되고 분석되는 정보 또한 입력된다. 사용자가 원하는 정보도 Blockchain에서 얻어온다. 이미지가 업로드 되면, 서버에 존재하는 학습된 모델로 예측을 하고 Blockchain으로 체인 코드 (스마트 컨트랙트)를 호출해 데이터를 담는다.
- Deep learning Model: Blockchain에 저장되어 있는 데이터를 기반으로 학습하고, 충분히 학습된 모델은 새로 들어오는 Blockchain 데이터를 검사하여 예측을 한다. Blockchain에서 데이터를 얻어와서 예측된 데이터를 추가하여 Blockchain 시스템에게 입력한다.
- Blockchain System: 사용자가 입력한 데이터와 Deep learning Model에서 예측된 결과 데이터를 Block으로 저장한다. 사용자가 원하는 정보를 Web System에게 보내주고, 사용자가 새로 입력한 데이터를 기반으로 생성된 Block의 데이터를 통합 Deep Learning model에게 보내준다.

#### 2.1.2 사용자 인터페이스 (User interfaces)

- 사용자들은 각자의 신분에 맞게 웹 사이트에서 서비스를 이용할 수 있어야 한다. 크게 회원가입, 로그인, 데이터 조회, 업로드로 나뉘며 자세한 사용자 인터페이스는 3.1.1에 기록한다.

### 2.1.3 하드웨어 인터페이스 (Hardware interfaces)

- Web Server은 서버 컴퓨터에서 실행된다. 요구되는 포트는 하나이고 이 외의 요구사항은 없다. 서버 컴퓨터는 네트워크로부터 받은 데이터를 Web Server에게 전달해준다.
  - 지원하는 기기의 종류: Window, MacOS 운영체제 등 Chrome 웹 사이트를 지원하는 모든 기기에서 Chrome을 통해 HTTP 프로토콜로 접속 가능
- Blockchain Node 시스템은 최소 3개의 서버 컴퓨터에서 실행되며, 각각의 서버 컴퓨터는 네트워크를 통해 서로 통신하며 합의 알고리즘을 구성한다.
  - P2P 통신을 통해 네트워크가 이뤄지며 따라서 네트워크 연결을 위한 포트 필요
- Deep Learning Server는 서버 컴퓨터에서 실행되며 GPU가 요구된다. Deep Learning Server는 각 노드들과 모델들을 주고받으며 각 노드에서 업데이트된 모델들을 취합하여 최종 모델을 웹 서버에 전송한다.

### 2.1.4 소프트웨어 인터페이스 (Software interfaces)

- Blockchain System
  - Name: Hyperledger Fabric
  - Version: 2.0
  - 역할: 프라이빗 블록체인 네트워크를 구성한다.
  - Source: Github (<https://github.com/hyperledger/fabric>)
- Deep Learning System
  - Name: Tensorflow
  - Version: 버전 2.0 이상
  - 역할: 딥러닝 모델을 생성할 때 사용한다.
  - Source: Tensorflow ([https://www.tensorflow.org/api\\_docs/python/tf](https://www.tensorflow.org/api_docs/python/tf))
- Deep Learning Federated
  - Name: Tensorflow Federated

- Version: 버전 0.10.0 이상
- 역할: 딥러닝 모델들을 학습하고 취합할 때 사용한다.
- Source: Tensorflow Federated Document  
([https://www.tensorflow.org/federated/federated\\_learning](https://www.tensorflow.org/federated/federated_learning))

#### Web System

- 1) Web Server
  - ◆ Name: HTML, Javascript, CSS
  - ◆ Version: 5, 7, 3
  - ◆ 역할: 웹 사이트의 정적 정보를 클라이언트에게 보여주기 위함
- 2) Web Application Server
  - ◆ Name: Node.js Server
  - ◆ Version: Node 12.8 LTS Version
  - ◆ 역할: 클라이언트와 웹 서버의 연결, 동적 정보 제공을 위함.

#### 2.1.5 통신 인터페이스 (Communications interfaces)

- Client는 Http를 통한 서버로의 접속을 하고, 이 서버는 JSON-RPC Call을 통해 블록체인에 통신을 요청한다. 블록체인 상에서 데이터가 들어오면 학습 모델을 TCP를 통해 받아와 학습을 한 후 결과를 기록, Client에 응답해준다.

#### 2.1.6 메모리 제약사항 (Memory constraints)

- 블록 생성을 하는 노드의 메모리는 충분한 연산 처리와 통신을 통해 블록을 생성할 수 있어야 한다
  - RAM 2GB 이상 필요
- 딥러닝 시스템은 학습을 하기 위한 계산을 위해 충분한 메모리가 필요하다.
  - RAM 3GB 이상 필요

#### 2.1.7 운영 (Operations)

- 사용자 그룹은 1) 의료진 그룹 2) 환자 그룹으로 나누어진다.

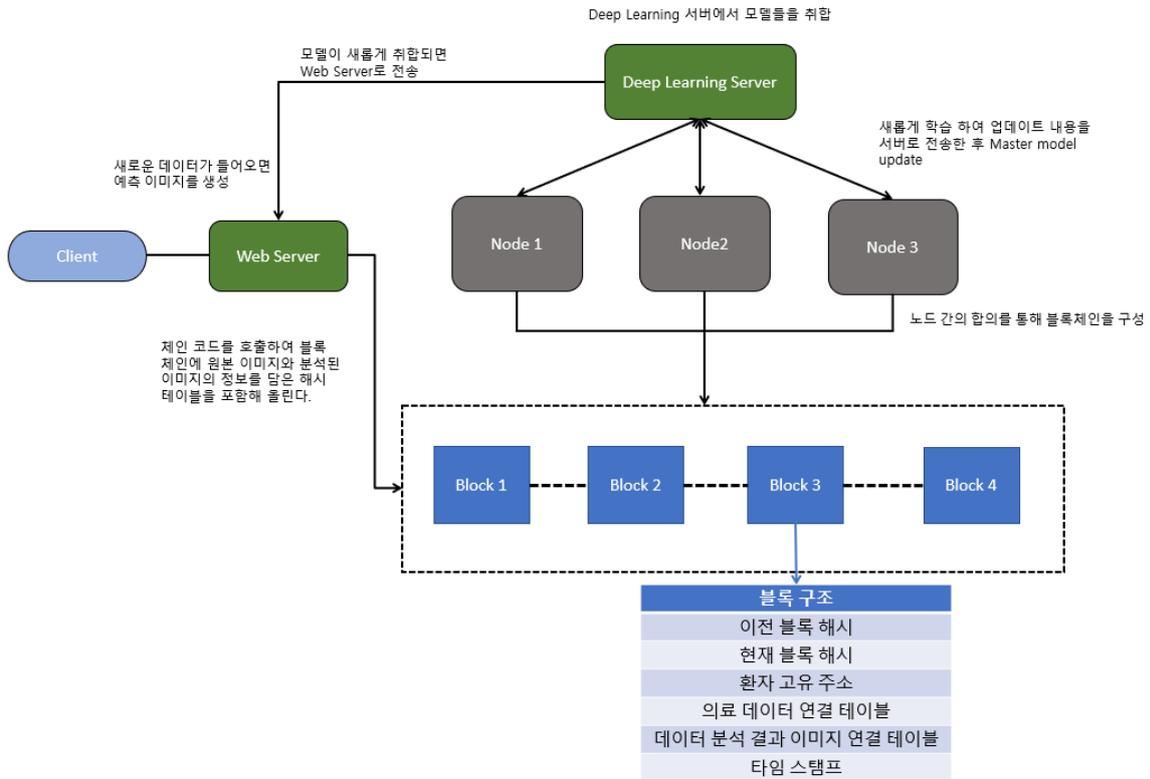
#### 2.1.8 사이트 적용 요건 (Site adaption requirements)

- Chrome 웹 사이트를 통해 접속할 수 있기 때문에 Chrome에 특화된 특징이 있다.

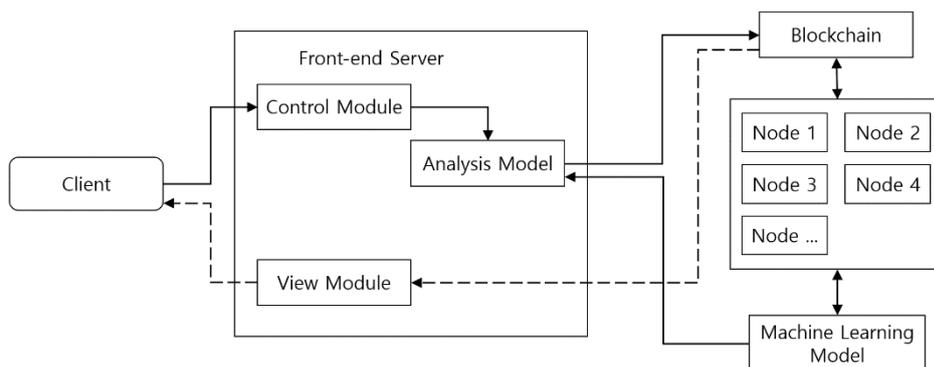
## 2.2 제품 기능 (Product functions)

기능적 요구사항	Operation
1.1 Sign In	1. signIn()
1.2 Sign Out	2. signOut()
1.3 Sign Up	3. signUp()
1.4 Upload Image	4. uploadImage()
1.5 Detect Strange	5. detectStrange()
1.6 Get Info	6. getInfo()
1.7 Authorize	7. getAuthorize()
2.1 Check Authorized Data	8. checkAuthorizedData()
2.2 Check Self Data	9. checkMyData()
2.3 Upload Data	10. uploadData()
2.4 Upload To IPFS	11. uploadToIPFS()
3.1 Send Model	12. sendModel()
3.2 Learning Model	13. fitting()
3.3 Send Model from Node to Server	14. sendToServer()
3.4 Modify Model	15. modifyModel()
3.5 Validate Model	16. validateModel()
3.6 Check Loss	17. checkLoss()
3.7 Fetch Model	18. fetchModel()

그림을 통한 설명



### 아키텍처 다이어그램



### 2.3 사용자 특성 (User characteristics)

- 이 제품은 환자에 대한 데이터를 시스템에 입력할 의료진, 자신의 의료 데이터를 조회할 환자들로 사용자가 정해진다.
- 의료진들은 환자의 병에 대해 분석해야 되는 의무를 가지고 있고, 이미지를 잘 추출할 수 있는 능력이 있어야 한다. 또한 환자의 개인 정보는 본인이 소지하

지 않아야 한다.

- 환자는 자신의 병에 대한 알아야 할 권리를 가져야 하며, 기본적인 웹사이트 접속, 이용을 할 줄 알면 자신의 의료 데이터에 접근할 수 있어야 한다. 하지만 다른 환자의 데이터에는 접근하지 않는 것을 원칙으로 한다.

## 2.4 제약사항 (Constraints)

- 규제 정책: 의료법에 의거하여, 의료진과 환자는 개인 정보 또는 의료 정보를 보호해야할 의무가 있다.
- 하드웨어 제한: 블록체인과 딥 러닝이 충분히 돌아갈 사양 좋은 컴퓨터 필요하다.
- 다른 applications에서의 인터페이스: 기기 중 모바일 기기로도 웹 사이트를 접속할 수 있기 때문에 이에 따른 처리 또한 필요하다.
- 감사 기능: 블록에 데이터가 정상적으로 들어가 블록이 정상적으로 쌓이는지에 대한 확인, 학습된 딥러닝 모델에 데이터가 정상적인 결과를 내는지에 대한 확인 필요. 서버와 클라이언트 간의 원활한 통신이 이루어지는지 확인.
- 안전 고려사항: 블록체인을 구성하는 주체에 대한 안전성 필요
- 언어 요구사항: Node.js와 Python 언어 간 자연스러운 연계 필요

## 2.5 가정 및 의존성 (Assumptions and dependencies)

- 소프트웨어의 버전 변경으로 인한 기능 추가 시 SRS 요구사항 영향 미침.
- Chrome에서의 지원이 끊길 경우 SRS 즉시 변경

## 2.6 단계별 요구사항 (Apportioning of requirements)

- 사용하는 SW의 업데이트에 따라 블록의 생성 현황을 지켜볼 수 있는 웹사이트를 도입할 수 있다.

# 3 상세 요구사항 (Specific requirements)

### 3.1 외부 인터페이스 요구사항 (External interface requirements)

#### 3.1.1 사용자 인터페이스(User interfaces)

- 1) 회원가입 하기 위한 사용자 인터페이스

## Sign Up

ID:

PW:

Auth:  

Doctor

Patient

- 1. 웹 사이트를 이용하기 위한 회원가입이다. 아이디와 비밀번호를 입력하고 자신의 신분을 선택한 뒤 Confirm 버튼을 누른다
- 2. 개인 고유 주소를 받으며 이 주소를 통해 블록체인 네트워크를 사용하고, 주소는 개인이 간직해야 하며, 환자의 고유함을 입증하기 위해 다른 사람에게 노출되면 안 된다.

- 2) 로그인을 하기 위한 사용자 인터페이스

# MediCh.A.I.n

ID:

PW:

Sign In

Sign Up

- 1. 아이디와 비밀번호를 입력한 뒤 로그인을 한다.

- 3) 의료데이터 이미지 업로드를 위한 사용자 인터페이스

## MediChA.I.n

 조범석  
주소: 1abCegHdqeczdwqw

이미지 업로드 및 분석

블록 생성

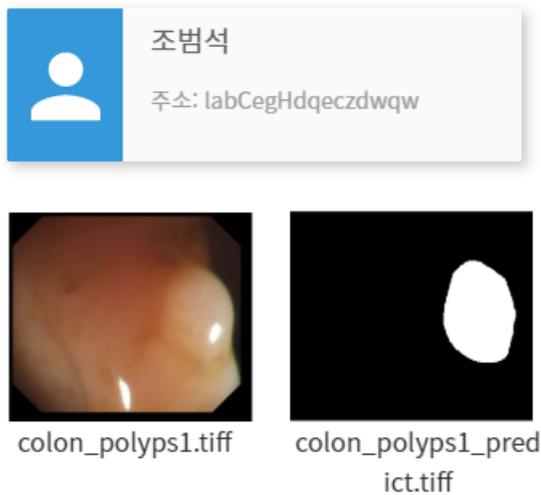
파일 업로드

확인하기

- 1. 파일 업로드 버튼을 눌러 이미지를 선택한다

- 2. 이미지를 업로드 하고 분석을 하며 예측 결과를 도출한다
- 3. 블록에는 업로드된 이미지, 예측이 끝난 이미지의 IPFS 와 연결할 수 있는 해시테이블과 환자의 정보가 익명화되어 올라간다.
- 확인 버튼을 눌러 블록체인에 올라간 정보를 조회할 수 있다.

- 4) 의료 정보 조회를 위한 인터페이스



- 1. 원본 이미지와 예측 이미지를 통해 용종의 위치를 파악한다.

## 3.2 System features

### 3.2.1 Web System

#### 3.2.1.1 Introduction/Purpose of feature

- 사용자가 이 SW에 원하는 기능을 입력하고 결과를 받아볼 수 있도록 Client-side 화면을 웹에 띄워준다. Block-chain으로 사용자가 입력한 데이터를 전송하거나 Block-chain으로부터 사용자가 보낸 요청에 대한 응답을 받아서 사용자에게 보여준다.

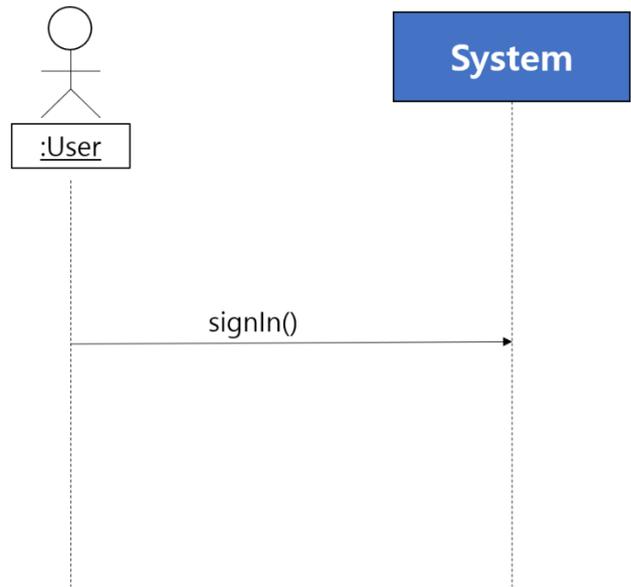
#### 3.2.1.2 자극/반응 순서

##### 3.2.1.2.1 Sign In

### 1.1 Sign In

Use Case: 1. Sign In

1. ID와 비밀번호를 입력한다.
2. 사용자의 개인 화면을 보여준다.
3. 사용자가 환자면 환자 고유 주소를 보여준다.

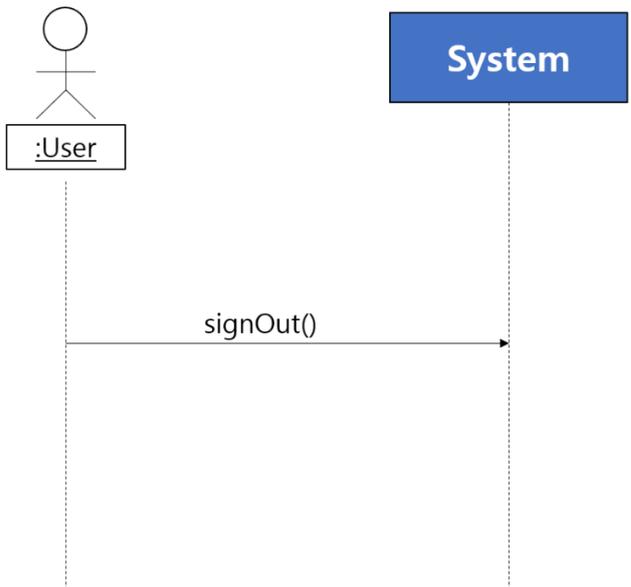


### 3.2.1.2.2 Sign Out

#### 1.2 Sign Out

Use Case: 2. Sign Out

1. 로그아웃 버튼을 누른다.
2. 로그인 화면으로 돌아간다. 개인 정보를 볼 수 없다.

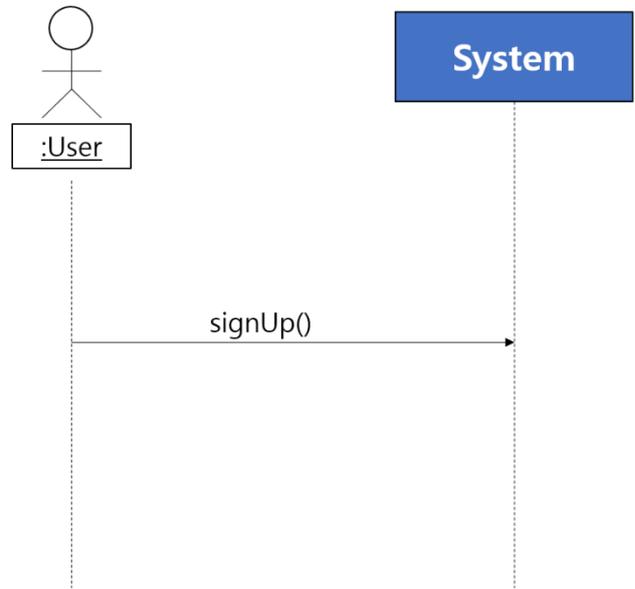


### 3.2.1.2.3 Sign Up

### 1.3 Sign up

#### Use Case: 3. Sign Up

1. 중복된 ID가 존재하면 회원을 생성하지 않는다.
2. ID, 비밀번호, 이름 중 하나라도 입력되지 않은 정보가 있으면 회원을 생성하지 않는다.
3. 환자 회원을 생성하면 환자 고유 주소를 보여준다. 환자 사용자가 주소를 확인하면 로그인 화면으로 돌아간다.

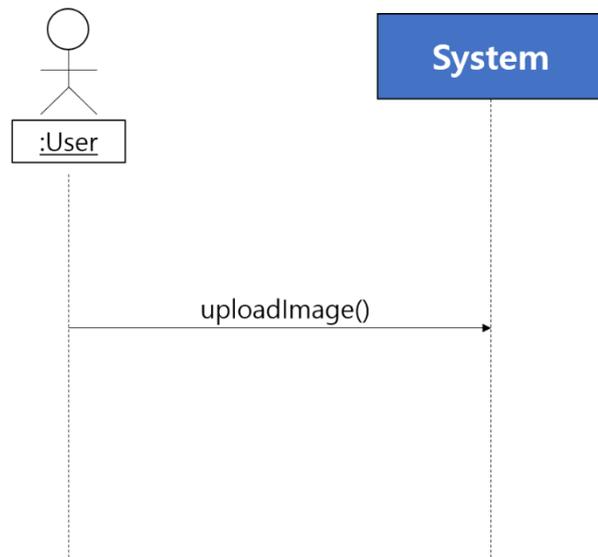


### 3.2.1.2.4 Upload Image

#### 1.4 Upload Image

#### Use Case: 4. Upload Image

1. 이미지를 웹 서버에 업로드하면 분석한다.
2. 이미지 분석 결과와 원본 이미지를 함께 블록체인에 업로드한다.

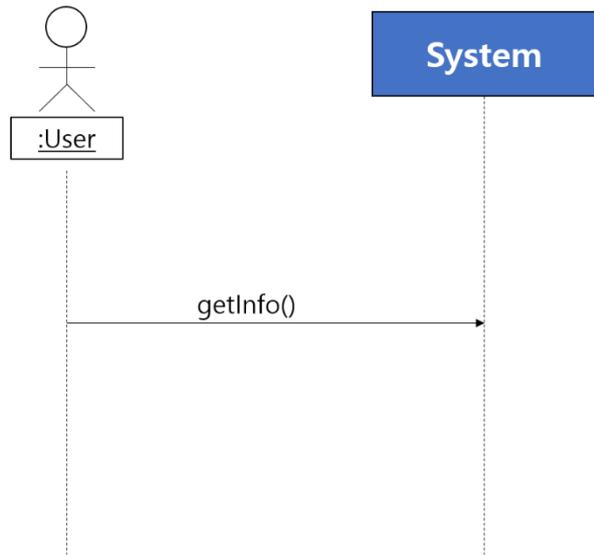


### 3.2.1.2.5 Get Info

### 1.5 Get Info

Use Case: 5. Get Info

1. 환자 고유의 주소를 이용하여 환자의 개인 정보를 요청한다.
2. 블록체인으로부터 환자 개인 정보를 반환한다.
3. 블록체인으로부터 반환된 데이터를 보여준다.



### 3.2.1.3 Associated functional requirements

#### 3.2.1.3.1 Sign In

Name	signIn()
Responsibilities	사용자가 로그인하면 ID와 권한을 포함한 개인 정보를 보여준다. 사용자가 환자면 환자 고유 주소를 보여준다.
Type	System
Cross References	
Notes	
Exceptions	ID와 비밀번호가 일치하지 않을 경우 로그인되지 않는다.
Output	개인 정보를 보여준다.
Pre-conditions	사용자 정보가 회원 정보 데이터베이스에 등록되어 있어야 한다.
Post-conditions	개인 정보를 보여준다.

#### 3.2.1.3.2 Sign Out

Name	signOut()
Responsibilities	시스템에 로그인되어 있는 사용자가 로그아웃 한다. 로그인 화면으로 돌아가고 개인 정보를 볼 수 없다.
Type	System
Cross References	

Notes	
Exceptions	N/A
Output	로그인 화면으로 돌아간다.
Pre-conditions	사용자가 로그인 되어있어야 한다.
Post-conditions	로그인 화면으로 돌아가고 개인 정보를 볼 수 없다.

### 3.2.1.3.3 Sign Up

<b>Name</b>	<b>signUp()</b>
Responsibilities	사용자가 입력한 ID, 비밀번호, 이름, 권한 정보대로 회원을 생성한다.
Type	System
Cross References	
Notes	
Exceptions	<ol style="list-style-type: none"> <li>1. 중복된 ID가 존재하면 회원을 생성하지 않는다.</li> <li>2. ID, 비밀번호, 이름 중 하나라도 입력되지 않은 정보가 있으면 회원을 생성하지 않는다.</li> <li>3. 환자 회원을 생성하면 환자 고유 주소를 보여준다. 환자 사용자가 주소를 확인하면 로그인 화면으로 돌아간다.</li> </ol>
Output	회원을 생성하면 로그인 화면으로 돌아간다.
Pre-conditions	회원가입 화면에 들어와야 한다.
Post-conditions	회원 정보를 데이터베이스에 저장해준다.

### 3.2.1.3.4 Upload Image

<b>Name</b>	<b>uploadImage()</b>
Responsibilities	이미지 데이터를 업로드하면, 웹 서버에서 이미지를 분석하고 원본 이미지와 이미지 분석 결과를 함께 블록체인에 업로드한다.
Type	System
Cross References	
Notes	
Exceptions	이미지 분석 결과 적합한 의료 이미지가 아니면 업로드하지 않고 경고 화면을 보여준다.
Output	<p>업로드에 성공하면 성공화면을 보여준다.</p> <p>업로드에 실패하면 실패화면을 보여준다.</p>
Pre-conditions	<p>의료진 회원으로 로그인 되어 있어야 한다.</p> <p>업로드 화면에서 이미지를 업로드해야 한다.</p>

Post-conditions	데이터가 블록체인에 업로드된다.
-----------------	-------------------

### 3.2.1.3.5 Detect Strange

Name	detectStrange()
Responsibilities	이미지 데이터를 업로드하면, 이상한 이미지인지 판단한다.
Type	System
Cross References	
Notes	
Exceptions	1. 이미지 분석 버튼을 누른다
Output	이상한 이미지일 경우 에러 메시지를 출력해준다.
Pre-conditions	의료진 회원으로 로그인 되어 있어야 한다. 업로드 화면에서 이미지를 업로드해야 한다.
Post-conditions	맞는 이미지이면 그대로 진행, 이상한 이미지이면 에러가 출력된다.

### 3.2.1.3.6 Get Info

Name	getInfo()
Responsibilities	환자 고유 주소를 이용하여 블록체인으로부터 환자의 개인 정보를 가져온다.
Type	System
Cross References	
Notes	
Exceptions	유효하지 않는 환자 고유 주소가 입력되면 정보를 반환하지 않는다.
Output	환자의 개인 정보를 받는다.
Pre-conditions	환자 정보 조회 화면에서 유효한 환자 고유 주소를 입력한다.
Post-conditions	환자 고유 주소가 가리키는 모든 블록의 정보를 받는다.

### 3.2.1.3.7 Authorize

Name	getAuthorize()
Responsibilities	로그인할 때 회원 정보를 바탕으로 시스템 접근 권한을 얻는다. 환자 사용자인 경우 환자 개인 정보만 보는 권한을 얻는다. 의료진 사용자인 경우 모든 환자의 개인 정보를 보는 권한과 데이터를 업로드하는 권한을 얻는다. 로그아웃하면 모든 권한이 사라진다.

Type	System
Cross References	
Notes	
Exceptions	N/A
Output	환자는 개인 화면에 조회 버튼만 활성화되고 의료진 사용자는 조회와 업로드 버튼이 모두 활성화된다.
Pre-conditions	로그인을 한다.
Post-conditions	환자 사용자는 개인 정보를 조회하는 기능만 접근할 수 있어진다. 의료진 사용자는 모든 환자의 개인 정보를 조회하는 기능과 데이터 업로드 기능을 접근할 수 있어진다.

### 3.2.2 Blockchain

#### 3.2.2.1 Introduction/Purpose of feature

- 개인 의료 데이터를 보관하기 위한 블록체인이다. 크게 하이퍼레저 패브릭 블록체인과 이미지 데이터를 보관할 수 있는 IPFS 로 이루어져 있으며, 한 시스템으로의 의료진의 접근과 환자들의 자신의 데이터에 대한 주권을 부여해주기 위한 목적을 가지고 있다.

#### 3.2.2.2 Associated functional requirements

##### 3.2.2.2.1 Check Authorized Data

Name	checkAuthorizedData()
Responsibilities	의료진은 블록체인 상 환자 고유의 주소(고유 ID)를 입력해 환자의 이미지 데이터 분석 결과들을 확인할 수 있어야 한다.
Type	System
Cross References	
Notes	
Exceptions	1. 의료진은 환자 고유의 주소 (고유 ID)를 입력해 환자의 의료 정보를 조회한다. 2. 절대 웹 상에서 다운로드할 수 없다.
Output	의료 이미지 정보 웹 사이트로 이동한다.
Pre-conditions	의료진은 로그인을 통해 조회 권한을 얻는다.
Post-conditions	블록체인에 접근해 정보를 출력해준다.

##### 3.2.2.2.2 Check Self Data

Name	checkSelfData()
Responsibilities	환자는 블록체인에 저장된 환자 개인의 의료 정보를 조회할 수 있어야 한다. 환자는 환자 고유의 주소(고유 ID) 와 본인의 이미지 데이터 및 분석 결과들을 한 화면에서 확인할 수 있다.
Type	System
Cross References	
Notes	
Exceptions	1. 환자는 자신의 의료 정보 페이지로 이동해 의료 데이터를 확인한다.
Output	의료 이미지 정보 웹 사이트로 이동한다.
Pre-conditions	환자는 로그인을 통해 본인 정보의 조회 권한을 얻는다.
Post-conditions	블록체인에 접근해 정보를 출력해준다.

### 3.2.2.2.3 Upload Data 의료 종사자가 이미지 데이터를 업로드 할 수 있어야 한다.

Name	uploadData()
Responsibilities	의료진 권한이 있는 사용자가 환자 고유 주소를 이용하여 환자의 데이터를 업로드한다.
Type	System
Cross References	System functions: R 2.3
Notes	
Exceptions	1.
Output	2.
Pre-conditions	
Post-conditions	블록체인 네트워크가 정상적으로 돌아가야함.

### 3.2.2.2.4 Upload To IPFS

Name	uploadToIPFS()
Responsibilities	IPFS를 통해 의료 이미지를 저장한다.
Type	System
Cross References	System functions: R 2.3
Notes	
Exceptions	웹서버에서 받은 이미지를 체인코드 상에서 IPFS에 업로드 한다.

Output	이미지에 접근 가능한 해시 테이블
Pre-conditions	IPFS 네트워크 또한 정상적으로 돌아가야함.
Post-conditions	IPFS에 이미지가 업로드 된다.

### 3.2.3 Deep learning

#### 3.2.3.1 Introduction/Purpose of feature

- 의료 데이터 분석을 하려면 의료 관계자가 일일이 데이터를 확인해가면서 판독을 해야 한다. 이는 시간적인 소모와 노동력이 소모가 된다. 이를 해결하기 위해서 딥 러닝을 이용한다. 하지만 의료데이터는 개인 정보가 매우 중요하기 때문에 중앙에 데이터를 집중하여 학습시키기에는 개인정보 문제가 생길 수 있다. 따라서 블록체인에 분산해 저장된 데이터들을 학습하여 새로운 데이터가 들어오면 분석 결과를 도출해 낼 수 있도록 한다.

#### 3.2.3.2 Associated functional requirements

##### 3.2.3.2.1 Send Model

<b>Name</b>	sendModel()
Responsibilities	서버에서는 master model을 각 노드에 전송할 수 있어야한다.
Type	System
Cross References	System functions: R 3.1
Notes	
Exceptions	1. 생성되어 있는 모델이 없을 경우 모델을 생성하여 전송한다.
Output	N/A
Pre-conditions	서버에서는 노드에 전송할 모델을 가지고 있다.
Post-conditions	각 노드에서 모델을 전송 받는다.

##### 3.2.3.2.2 Learning Model

<b>Name</b>	fitting()
Responsibilities	각 노드에서 학습할 블록들이 생성되어 있고 이 블록에 있는 데이터를 이용하여 학습한 후 모델을 업데이트한다.
Type	System
Cross References	System functions R 3.2
Notes	
Exceptions	1. 생성된 블록이 없을 경우 학습을 진행하지 않는다.

Output	N/A
Pre-conditions	서버에서 모델을 전송 받고 데이터 블록이 준비되어 있다.
Post-conditions	학습 후 업데이트된 모델이 생성된다.

### 3.2.3.2.3 Send Model from Node to Server

<b>Name</b>	sendToServer()
Responsibilities	각 노드에서 설정한 epoch만큼 학습이 완료되면 서버에 학습한 모델을 전송한다.
Type	System
Cross References	System function R 3.3
Notes	
Exceptions	N/A
Output	N/A
Pre-conditions	각 노드에서 학습을 한 결과를 모델에 업데이트 한다.
Post-conditions	서버에서는 각 노드에서 보내온 모델들의 업데이트 내용을 받는다.

### 3.2.3.2.4 Modify Model

<b>Name</b>	modifyModel()
Responsibilities	서버에서는 각 노드에서 보내온 모델들의 업데이트 내용들을 사용하여 새로운 master 모델을 업데이트 한다.
Type	System
Cross References	System function R 3.4
Notes	
Exceptions	N/A
Output	N/A
Pre-conditions	각 노드에서 학습을 하면서 업데이트한 내용들이 서버에 있다.
Post-conditions	업데이트한 내용들을 사용하여 새로운 master 모델을 생성하여 validation 노드로 전송한다.

### 3.2.3.2.5 Validate Model

<b>Name</b>	validateModel()
Responsibilities	Master 모델이 정확한 예측을 하는지 검증을 한다.
Type	System

Cross References	System function R 3.5
Notes	
Exceptions	N/A
Output	N/A
Pre-conditions	Master 모델이 validation을 담당하는 노드에 전송된다.
Post-conditions	Validation Loss 값을 구하고 Server에 전송한다.

### 3.2.3.2.6 Check Loss

<b>Name</b>	checkLoss()
Responsibilities	서버에서 각 노드에서 나온 validation loss를 이전의 validation loss와 비교하여 master 모델을 교체할지 결정하고 특정 횟수 만큼 개선이 이루어지지 않으면 training을 중지한다.
Type	System
Cross References	System function R 3.6
Notes	
Exceptions	N/A
Output	N/A
Pre-conditions	Validation 노드에서 전송한 Loss 결과 값을 Server에 전송된다.
Post-conditions	서버에서 Master 모델을 교체할지 결정하고 training을 중지할지 결정한다.

### 3.2.3.2.7 Fetch Model

<b>Name</b>	fetchModel()
Responsibilities	딥러닝 서버에서 웹서버에 분석 모델을 전송한다.
Type	System
Cross References	System function R 3.7
Notes	
Exceptions	N/A
Output	N/A
Pre-conditions	모델 학습이 완료된 상태이다.
Post-conditions	웹서버에 딥러닝 서버에서 전송한 모델이 저장된다.

## 3.3 성능 요구사항 (Performance requirements)

- 의료 이미지 업로드가 30 초 내로 빠르게 되어야한다
- 딥러닝 예측이 5 분 안에 이미지 생성까지 빠르게 되어야 한다
- 사용자는 자신의 개인 정보를 안전하게 유지할 수 있어야한다
- 의료 정보 조회가 1 분 내로 완료된다
- 환경 구성
  - 하이퍼레저 패브릭을 활용해 의료 이미지 데이터 링크와 개인 데이터로 블록체인을 구성할 수 있어야 한다
  - 학습할 수 있는 환경이 구성되어야 한다

### 3.4 설계 제약사항 (Design constraints)

- 하드웨어 적으로 메모리 한계에 부딪혀 블록 생성 시간이 느려지거나, 학습 시간이 느려질 수 있다.
- 데이터의 종류에 따라 데이터 크기가 클 수록 training 시간과 prediction 시간이 길어질 수 있다.

### 3.5 소프트웨어 시스템 속성 (Software system attributes)

#### 3.5.1 신뢰성 (Reliability)

- 블록체인 네트워크가 원활하게 돌아가 신뢰성, 안전성을 가질 수 있어야 한다.
- 이를 위해서 합의에 필요한 충분한 노드가 필요하다.
- 적어도 3개의 노드가 있어야 안전한 네트워크라고 정의한다.
- 웹 사이트를 통해 접근 시 안전한 보안 프로토콜이 적용되어야 한다.
- 학습이 완료 후, 새로운 데이터가 업로드 되면 prediction이 정확하게 이루어져야한다.

#### 3.5.2 가용성 (Availability)

- 웹 사이트를 제공하는 서버가 재시작 될 때 원활하게 돌아가기 위해 빠른 연결이 필요하다.

- 블록체인 네트워크가 끊임없이 돌아가야 전체 시스템에 대한 가용성 수준을 보장하기 때문에 꾸준한 모니터링이 필요하다.

### 3.5.3 보안성 (Security)

- 신뢰성과 마찬가지로 웹 사이트에 악의적인 접근 (예를 들면 잘못된 데이터 입력 - 시스템 Operation이나 SQL 삽입)을 방지해야 한다.
- 사용자의 정보를 보호할 수 있는 암호화 알고리즘이 적용되어야 한다.
- 블록체인 노드는 정해진 서버 IP 이외의 서버와는 통신하지 말아야 한다.
- 딥 러닝 서버는 training 하는 데이터에 대해서 일절 열람할 수 없도록 하여야 한다.

### 3.5.4 유지 보수성 (Maintainability)

- 소프트웨어는 쉽게 유지보수 되어야 하기 때문에 각 시스템의 업데이트에 다른 시스템이 영향을 받지 않게 관리한다.

### 3.5.5 이식성 (Portability)

- 블록체인 노드가 구동되는 서버, 딥러닝 서버, 웹 서버는 모두 Ubuntu 18. 04 를 사용한다.
- 원활한 코드 구성을 위해 증명된 Node.js Python 언어를 사용한다.
- 웹 사이트로의 접근은 Chrome을 사용할 수 있는 모든 OS에서 가능하도록 한다.

## 3.6 이외 요구사항 (Other requirements)

- 변경 사항이 있을 경우 SRS의 버전 관리가 진행되어야 한다.