

---

# BLOOD WALLET

---

## REQUIREMENTS SPECIFICATION



Student Number	Name
2017310794	서성윤
2014311338	모경준
2014310967	곽희주
2017311290	김진범
2018312494	박상현
2020318179	Snaedis Perla Sigurdardottir

# Contents

1. Preface	6
1.1. Readership	6
A. User Requirements Readership	6
B. System Requirements Readership	6
1.2. Document Structure	7
2. Introduction	9
2.1. Needs	9
2.2. System Overview	11
2.3. Expected Effects	13
3. Glossary	14
4. User Requirements Definition	15
4.1. Functional Requirements	15
A. Sign up/Login	15
B. Scan Certificate	16
C. Posting	17
D. Donate Certificate	18
E. Mypage	18
F. Preference of donation	18
4.2. Non-functional Requirements	19
A. Product requirements	19

## Requirements Specification

B. Organizational requirements	20
C. External requirements	21
5. System Architecture	22
5.1. Frontend Architecture	22
5.2. Backend Architecture	23
5.3. Certificate Scan System	24
5.4. Certificate Donation System	25
5.5. Donation History System	26
5.6. Ranking Post System	26
6. System Requirements Specification	27
6.1. Functional Requirements – Frontend	27
A. Ranking	27
B. Show Post Detail	28
C. Donate Certificates	28
D. Scan a Certificate	29
E. Show Donation History	30
F. Mypage	31
6.2. Functional Requirements – Backend	32
A. Login	32
B. Certificate Scan System	33
C. Certificate Donation System	33

## Requirements Specification

D. Donation History System	34
6.3. Non-functional Requirements	35
A. Product requirements	35
B. Organization requirements	36
C. External requirements	37
6.4. Scenario Examples	38
A. Sign Up Scenario	38
B. Certificate Donation Scenario	39
7. System Models	40
7.1. Context models	40
A. Context Diagram	40
B. Process Diagram	41
7.2. Interaction models	41
A. Use case Diagram	41
B. Tabular Description of Use case Diagram	42
7.3. Behavioral models	48
A. Certificate Scan System DFD	48
B. Certificate Donation DFD	48
C. Donation History System DFD	49
8. System Evolution	50
8.1. Limitation and Assumption	50

## Requirements Specification

8.2. Evolution System	51
A. 종이 헌혈 증서	51
B. 헌혈 통합 어플리케이션	51
9. Appendices	52
9.1. Hardware requirements	52
9.2. Database requirements	53
A. User	53
B. Post	53
C. Blood Certificate Issue	53
D. Authority	53
E. Donation	53
F. Keyword	53
10. References	54

# 1. Preface

This chapter defines the expected readership of the document, and briefly introduces the content of each chapter. This chapter also describes version history including a rationale for the creation of a new version and a summary of the changes made in each version.

## 1.1. Readership

본 문서는 다양한 독자에게 읽힐 것을 상정하고 있다. 따라서 각 부분을 서술하는 데 있어 어떠한 독자층을 상정하고 있는지를 설명한다.

### A. User Requirements Readership

This document is divided into user requirements and system requirements based on the differences between the two reader groups. User requirements are based on the user's point of view. The system requirements are very detailed.

### B. System Requirements Readership

The readers of this document are mainly system architects, software designers, developers, and testers, and can also provide some help for software users.

## 1.2. Document Structure

### A. Introduction

개발하고자 하는 시스템을 둘러싼 니즈를 설명하고, 시스템의 개략적인 구조와 기능에 대해 설명한다. 또한 시스템을 개발함으로써 얻을 수 있는 목표 효과에 대해서 설명한다.

### B. Glossary

문서 전반에서 사용되고 있는 기술적 용어들에 대해 정의하고, 해당 용어가 어떤 맥락에서 사용되는지 기술한다. 본 문서는 시스템 개발자뿐 아니라 사용자, 이해당사자 등의 독자 또한 대상으로 포함하고 있으므로, 가능한 자세하게 용어에 대해 설명한다.

### C. User Requirements Definition

시스템의 기능적, 비 기능적 요구사항을 사용자의 입장에서 설명한다. 간략한 도식과 함께 시스템의 사용자가 문서를 이해하기 쉽도록 자연어를 사용하여 서술한다.

### D. System Architecture

시스템의 구조를 개괄적으로 기술하고, 시스템의 기능이 각 서브시스템에 어떻게 할당되어 분포하는지를 설명한다.

### E. System Requirements Specification

User Requirements Definition 에 간략히 서술되어 있는 요구사항을 기반으로, 기능적 요구사항과 비 기능적 요구사항, 기타 요구사항을 자세히 서술한다. 이때 본 챕터는 시스템 설계 단계와 구현 단계를 포함한 개발 프로세스 전반에서 사용되어야 하므로 도표 등을 사용해 체계적으로 표현한다.

### F. System Models

시스템의 각 컴포넌트 간의 관계, 시스템을 둘러싼 외부 환경과의 관계 등을 다이어그램으로 표현한다.

G. System Evolution

시스템의 개발상 한계에 대해 서술하고, 시스템의 운영 과정에서 발생할 수 있는 환경의 변화를 예상하고 그에 대한 대응 방안을 서술한다.

H. Appendices

본문에서 생략된 참고자료 등을 기술한다. 하드웨어, 데이터베이스 요구사항, 개발 환경 요구사항 등이 포함되어 있다.

I. Index

본 문서에서 사용된 그림, 표, 다이어그램 등의 색인을 기술한다.



## 2. Introduction

이 챕터에서는 본 시스템을 둘러싼 니즈를 설명하고, 시스템의 기능을 비롯해 본 시스템이 다른 시스템과 어떻게 상호작용하는지를 간략히 서술한다.

### 2.1. Needs

헌혈 증서 제도는 한국에 1975 년에 도입되었다. 헌혈을 한 사람은 헌혈 증서를 받게 되는데 헌혈 증서는 나중에 수혈을 받을 때 헌혈 증서 한 장당 혈액 한 팩의 값을 공제받을 수 있다. 혈액 한 팩의 가격은 진료의 보험 적용 가능 여부에 따라 다른데 적게는 7416₩에서 109,018₩까지 된다. 헌혈 증서의 가치도 마찬가지다.

“헌혈환급 적립금”은 헌혈 증서를 사용해서 혈액 투석을 받을 경우 비용은 보건복지부에서 부담하는데 이를 위해 마련해둔 기금 제도이다. 혈액원이 기증자로부터 혈액을 받으면 보건복지부에 한 건당 1500₩을 내게 되어 있다. 따라서 환자가 혈액투석을 받으면 그 혈액의 비용은 국민건강보험과 헌혈환급적립금에서 부담을 하게 되어있는 것이다. 이것이 혈액관리법에서 말하는 헌혈환급적립금 제도이다.

헌혈 증서는 종이로 관리되어지는데 이는 여러 문제점들을 야기한다. 첫번째로는 관리성 측면이다. 헌혈혜택을 받으려면 헌혈 증서가 반드시 필요한데 분실하거나 시간이 오래되어 바코드 부분이 지워지면 그 효력이 없어진다는 문제가 있다. 이런 상황에도 불구하고 적십자는 헌혈 증서 재발급 민원들을 외면하고 있다.

또 다른 문제점은 헌혈 증서의 낮은 사용률이다. 아래 표는 2010 년도부터의 헌혈환급적립금 변화 추이를 보여준다. 아래 표를 보면 헌혈 증서 예치금 환급율은 10%언저리에서 머무르고 있으며 매년 헌혈환급적립금은 의미없이 쌓여만 가는 실정이다.

## Requirements Specification

연도	헌혈환급적립금 연간 총액	헌혈 증서 환급 건수	헌혈 증서 적립금 환급율	누적 헌혈 환급 적립금
2010 년	6,538,805,000 원	403,854 건	15.2%	4,570,622,305 원
2011 년	6,414,237,500 원	383,684 건	14.7%	8,277,730,912 원
2012 년	6,677,265,000 원	357,922 건	13.1%	12,617,472,313 원
2013 년	7,266,430,000 원	350,380 건	12.0%	17,843,268,357 원
2014 년	7,839,590,000 원	310,300 건	10.2%	23,485,920,534 원
2015 년	8,069,867,500 원	319,646 건	10.4%	29,179,422,907 원
2016 년	7,570,040,000 원	282,171 건	9.8%	34,940,174,522 원
2017 년	7,801,010,000 원	293,445 건	10.0%	40,739,604,455 원
2018 년	5,112,312,500 원	187,275 건	9.8%	44,691,286,810 원

Table 1: 연도별 헌혈 환급 적립금

또한 혈액관리법 제 15 조 제 3 항에 의하면 헌혈환급적립금의 용도가 명시되어 있는데 그중 혈액원 혈액관리업무의 전산화에 대한 지원이라는 항목이 있다. 하지만 지난 몇 5 년간 전산화에 투입된 예산은 0 원인 것으로 나타났다. 즉 시스템 전산화는 정책적으로도 가능한 지점에 있다고 볼 수 있다.

따라서 위에 언급한 문제를 해결하기 위해 현재 헌혈 증서 시스템과 공존할 수 있는 전산화할 수 있는 시스템의 개발이 필요해졌다.

## 2.2. System Overview

Blood Wallet 어플리케이션은 크게 1) 헌혈 증서의 저장, 2) 기부 요청 글 포스팅, 3) 헌혈 증서 기증자와 수령자 연결, 4) 헌혈 증서 기부의 기능을 가진다.

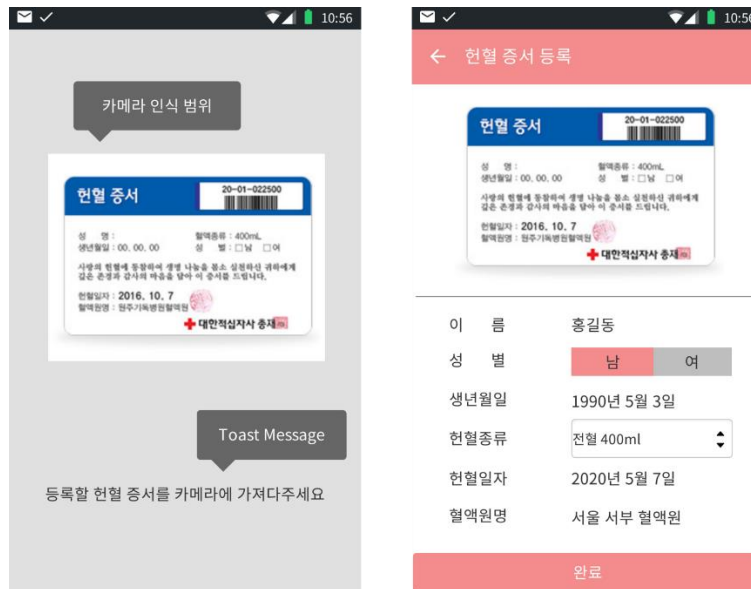


Figure 1: 시스템 목업 - Ranking

Blood Wallet App 은 헌혈 증서를 관리하는 서비스를 제공한다. 구글의 OCR(광학 문자 인식) API 를 이용해 사용자들의 헌혈 증서를 스마트폰 카메라로 스캔 하고 관련된 정보들을 데이터베이스에 저장한다. 관련된 정보들에는 성별, 헌혈 일시, 헌혈 증서 코드, 헌혈 종류, 성명, 혈액원 등이 있다.

경제적 어려움으로 인해 헌혈 증서 기증이 필요한 사람들은 그들의 사연을 어플리케이션 커뮤니티에 포스팅할 수 있다. 포스팅은 사진과 텍스트 모두 사용가능 하고 받고 있는 진료, 필요한 헌혈 증서 수 등을 적게 된다.

유저들은 기증 선호도를 설정할 수 있다. 여기서 선호도란 기증자가 기증하고 싶어하는 특정 집단을 선정할 수 있게 하는 것이다. 예를 들어 환자의 진료 종류, 성별, 나이대 등 원하는 정보들을 설정할 수 있다. 선호도 설정이 키워드를 기반으로 사용자들의 기부 요청 글에 등급을 매겨 기증자들의 선호도와 상통하는 사연을 추천해준다.

## Requirements Specification

Blood Wallet 은 직접 헌혈 증서를 기부할 수 있게 하는 플랫폼을 제공한다. 현재 시스템에서는 물리적으로 해당 헌혈 증서를 가지고 있다는 것이 곧 양도 받음을 증명하는 것이었다. 하지만 헌혈 증서가 전산화 되면 현재 시스템과 같이 양도를 받았다는 것을 증명해야 한다. 이를 보완하기 위해 Kakao 자회사 Klaytn 블록체인 네트워크를 활용해서 구현된다. 사용자는 원하는 사연을 골라 기증을 하면 기증자의 헌혈 증서는 수령자에게 전달된다.

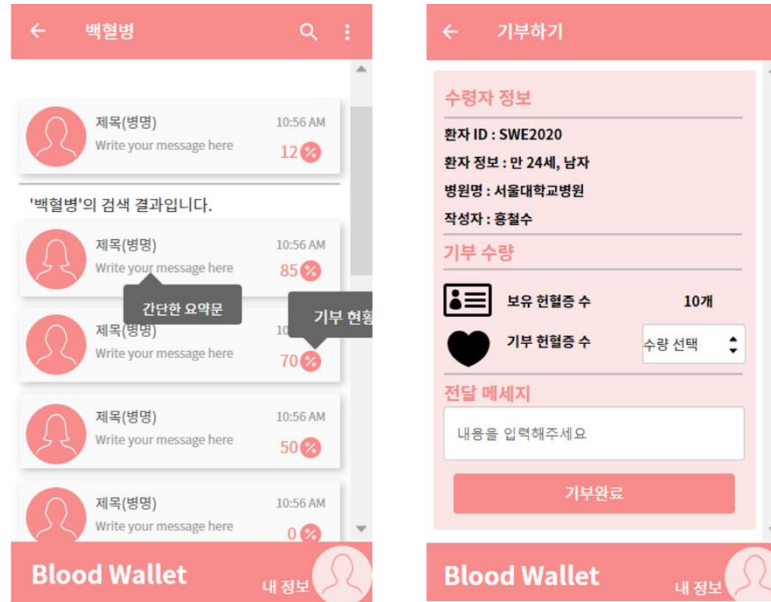


Figure 2: 시스템 목업 – Donation

## 2.3. Expected Effects

### A. 헌혈 증서 관리성 향상

Blood Wallet 어플리케이션 사용으로 인해 앞서 다뤘던 문제점들은 대부분 해결될 것으로 기대한다. 종이로 관리되던 헌혈 증서가 어플리케이션 내의 데이터베이스와 블록체인 상에서 관리되므로 종이를 사용함으로써 발생하는 분실, 훼손 등의 문제점들이 해결될 것이라고 기대한다.

### B. 헌혈 증서 사용 활성화

이 어플리케이션은 매우 간편한 헌혈증서의 기부를 제공함으로써 그동안 낮았던 헌혈 증서의 사용률을 높일 수 있다. 기존 종이 헌혈 증서를 사용할 때에는 관리도 어렵고 기부를 하려고 해도 단체에 직접 헌혈증서를 보내야 했기 때문에 불편함 때문에 기증도 쉽지 않았다. 하지만 이 어플리케이션을 통해 손쉬운 기부와 사용자들의 관심에 적합한 조건의 사용자들을 추천해줌으로써 더 높은 참여도를 이끌어 낼 수 있다. 헌혈 증서의 활성화로 그저 의미없이 쌓여만 가던 헌혈환급적립금도 어느정도 해소 가능할 것으로 기대한다. 헌혈증서의 사용이 활성화되면 헌혈증서의 불법 매매 또한 자연스럽게 없어질 것으로 기대한다.

### 3. Glossary

This chapter specifies the terms used in this documentation.

Term	Description
<b>Login</b>	A function that let the user enter the application with a private customized information and setting
<b>Search</b>	A function that let the user to toggle through the products and find the most appropriate ones
<b>Ranking</b>	사용자가 헌혈 증서 기부를 할 때 관심있게 보는 항목. 질병의 종류, 나이 등의 키워드를 기반으로 Post 를 정렬하는 것을 말한다.
<b>Database</b>	A backend term which represents a large collection of data (in this case Reviews)
<b>API</b>	A set of higher-class functions and interfaces developed by advanced programmers that can be imported and use easily in different projects
<b>NLP</b>	Natural Language Processing
<b>OCR</b>	Optical Character Recognition

Table 2: Glossary

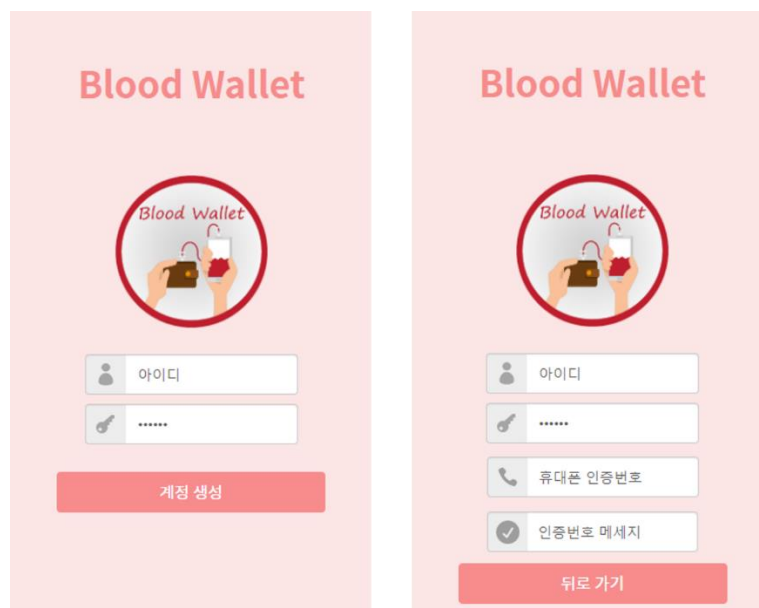
## 4. User Requirements Definition

This chapter lists the functional and non-functional requirements of the application. The functional requirements describe the services that the system provides to users and thus are important for the user to know in order to use the system. Therefore, we use natural language and visual materials to explain each requirement.

### 4.1. Functional Requirements

#### A. Sign up/Login

This requirement is where users can sign up or login to the platform using a specific ID and Password. Before users can start using Blood Wallet, they have to create an account using a unique ID and PW as well as putting in some personal information. This information will be stored in our database. Later on, when the users want to login, the system will authenticate the given (ID, PW) pair by checking if it can match it with a one from the database.



The image displays two mobile application screens for 'Blood Wallet'. Both screens feature a pink background and a circular logo at the top center showing a hand holding a wallet with a red lock icon. The left screen is for account creation, with input fields for '아이디' (ID) and a password (masked with dots), and a red button labeled '계정 생성' (Create Account). The right screen is for login, with input fields for '아이디' (ID), a password (masked with dots), '휴대폰 인증번호' (Mobile Phone Verification Number), and a checkbox for '인증번호 메시지' (Verification Number Message), and a red button labeled '뒤로 가기' (Go Back).

Figure 3: Login/Signup process

### B. Scan Certificate

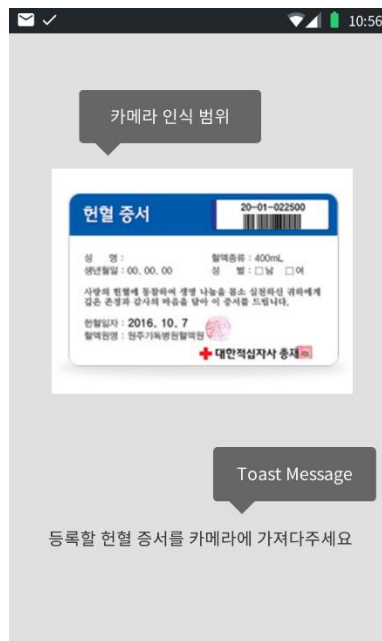


Figure 4: User Requirement - Ranking

This requirement enables the user to scan in his blood certificates. We will use OCR (Optical Character Recognition) API from Google Cloud to scan and then store the information in the database so that the user can view his certificates in the application.



## C. Posting

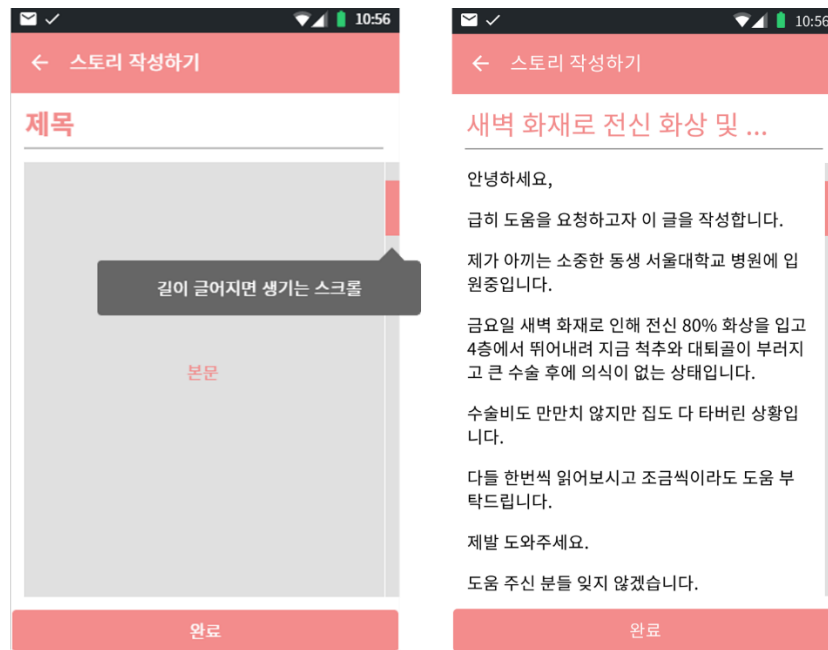


Figure 5: User requirement – Posting

If a user wishes to receive a donation then this requirement makes it possible for him to make a post, explaining his situation and asking for donations. 이 Post 는 데이터베이스에 저장되고 다른 사용자들이 기부하고자 하는 사연을 찾을 때 확인할 수 있다.

## D. Donate Certificate



Figure 6: User requirement – Donate Certificate

If a user wishes to Donate a certificate, he can browse through the posts from users that wish to receive a donation and choose to donate to one or more of them.

## E. Mypage

사용자의 개인 정보를 확인할 수 있다. 기본적인 정보 뿐만 아니라 사용자가 보유하고 있는 헌혈 증서 개수를 알 수 있다.

## F. Preference of donation

This function allows the user to set up preferences, made possible by an NLP api, to narrow down the search of a recipient if he has some specific preferences, like age or type of disease.

## 4.2. Non-functional Requirements

### A. Product requirements

#### 1. Usability

성공적인 어플리케이션을 위해 필수적인 요구사항이다. 시스템은 직관적인 사용자 인터페이스를 제공해야 하며, 동시에 단순한 조작으로 사용자 친화적이어야 한다. Blood Wallet에서는 Navigator View의 버튼들로 구현하였기 때문에, 사용자는 쉽고 빠르게 어플리케이션의 여러 기능들을 확인할 수 있다.

또한 사용자에게 복잡한 블록체인 경험을 막고자 Fee delegation을 통해 본 어플리케이션이 블록체인을 이용하고 있는 것을 숨길 수 있도록 설계하였다. 이로써 기부 거래의 투명성과 쉬운 사용법을 제공한다.

#### 2. Performance

블록체인 네트워크를 통해 물리적인 현물 증서를 처리하기 때문에 전통적인 서버 구조보다 느린 Latency를 처리해야 한다. 블록체인을 통해 투명성을 확보할 수 있지만 사용자에게 어플리케이션의 큰 Latency를 줄이기 위한 알고리즘이 필요하다. 또한 현물 증서를 촬영하여 등록할 때, 이미지 처리 시 여러 예외 케이스가 발생할 수 있기 때문에 이를 고려하여 사용자의 만족도를 최대화해야 한다.

#### 3. Dependability

Blood Wallet은 모든 트랜잭션이 Atomic하게 처리되도록 함으로써 Dependability를 최대화하였다. 만약 현물 증서를 전달하거나 현물 증서 기부 요청 글을 작성하는 중간에 시스템이 비정상적으로 종료될 때, 서버와 블록체인 네트워크는 네트워크 연결을 잃기 때문에 비동기적인 수정과 트랜잭션 발생은 시스템에 반영되지 않는다. 또한 모바일 기기의 문제로 어플리케이션이 종료된다면 서버가 빠르게 다시 네트워크 연결을 시도하여 정상적인 상태로 되돌아간다. 그리고 블록체인 네트워크는 카카오 자회사

Klaytn 을 이용하고, 한글을 인식률이 높은 Clova OCR API 를 도입함으로써 헌혈 증서 등록 시스템과 헌혈 증서 기부 트랜잭션 관리의 높은 정확성과 신뢰도를 얻을 것이다.

#### 4. Security

헌혈 증서 기부를 블록체인 네트워크를 통해 수행하기 때문에 Public key 와 Private Key 가 필요하다. 이때 Private Key 는 개인의 고유한 비밀 키이기 때문에 서버에서 관리하지 않고 모바일 기기 로컬 스토리지에 저장하여 관리한다.

### B. Organizational requirements

#### 1. Operational

현재 지속적으로 운영할 수 있는 하드웨어 자원이 없으므로, 상용 클라우드 서버에 배포할 것을 전제로 한다. 따라서 다양한 클라우드 서버에 배포할 수 있도록 hardware-dependent 한 구현을 자제하고, 서버의 주소가 바뀌어도 대응할 수 있게 시스템을 설계해야 한다.

#### 2. Development

보다 빠른 기간에 시스템 개발이 가능하도록 Parallel Development 를 도입한다. 이를 위해 개발 프로세스를 plan-driven 으로 설계하고, 개발 단위를 프론트엔드와 백엔드로 나누며, 프론트엔드와 백엔드는 서로 독립적으로 구동할 수 있게 설계해야 한다.

### C. External requirements

#### 1. External System Policy

본 시스템은 헌혈 증서를 기부하는 시스템이므로 헌혈 증서의 양도가 어떠한 경제적인 거래로 이어질 수 있다. 또한 보건 복지부는 혈액관리법 제 2 조 제 4 호에 언급된 무상 헌혈을 원칙으로 하기 때문에 헌혈 증서 전산화 추진은 심도 깊은 논의가 필요하다고 언급하였다. 따라서 사용자의 기부가 어떠한 경제적 이득으로 이어지지 않도록 주의해야 한다.

#### 2. Personal Information Protection

본 시스템은 헌혈 증서 관리 및 기부 기능을 제공하기 위해 회원제로 운영한다. 따라서 사용자의 개인정보를 수집하며, 개인 정보가 제대로 보호되지 않을 경우 민형사상의 책임을 질 가능성이 있다. 따라서 가입 시 사용자 식별에 필요한 최소한의 정보만을 제공받고, 개인정보가 유출되지 않도록 충분한 노력을 기울여야 한다.

## 5. System Architecture

이번 챕터에서는 시스템의 개괄적인 Architecture 를 기술한다. 시스템의 전체적인 구조와 각 서브시스템의 구성, 서브시스템간 관계를 대략적으로 설명하며, 각 구조는 다이어그램을 첨부하여 이해를 돕는다.

### 5.1. Frontend Architecture

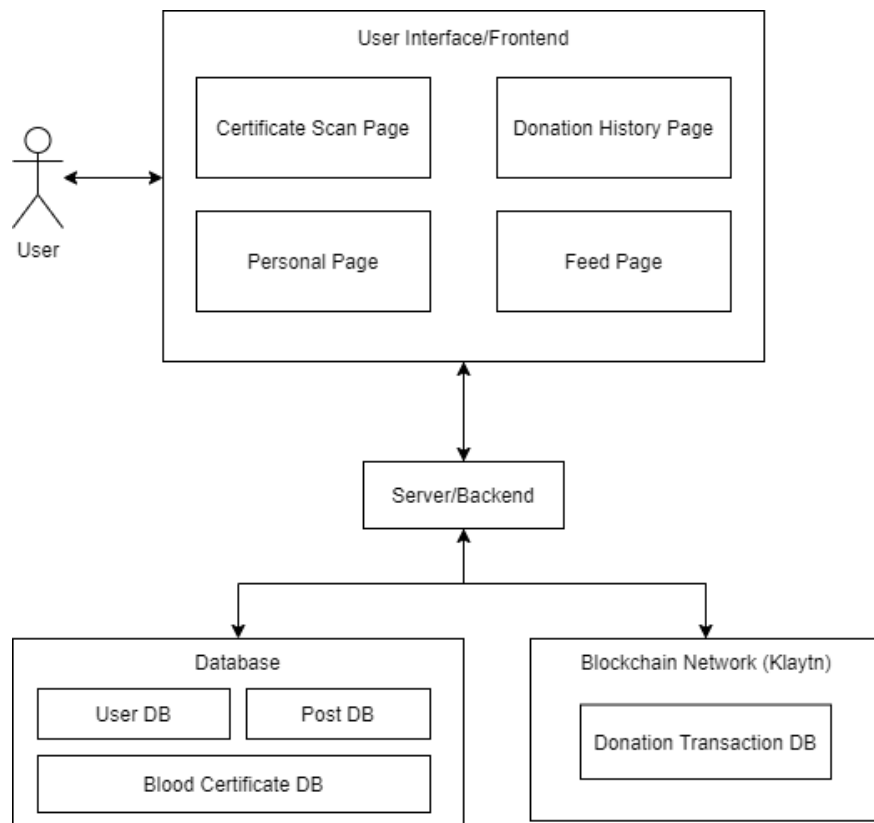


Diagram 1: Frontend Architecture

프론트엔드 구조는 사용자와 상호작용하는 User interface 이고 사용자의 요청을 인식하고 데이터를 수집한다. Blood Wallet 에는 가장 중심이 되는 인터페이스는 총 네 개로 구성되어 있다. 이는 헌혈 증서를 촬영하여 시스템에 등록할 수 있는 Certificate Scan Page, 헌혈 증서 기부 기록을 조회할 수 있는 Donation History Page, 사용자의 개인 정보를 확인할 수 있는 Personal Page, 마지막으로 여러 사용자들의 기부 요청 글을 확인할 수 있는 Feed Page 이다. Diagram. 3 은 프론트엔드를 중점적으로 묘사하고자 백엔드/서버에 관한 세부 정보는 생략한다.

## 5.2. Backend Architecture

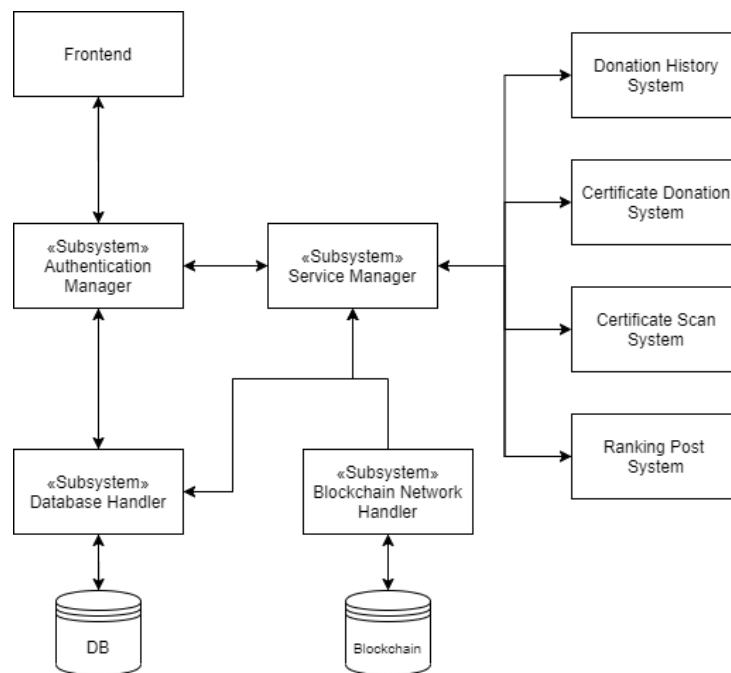


Diagram 2: Backend Architecture

백엔드 시스템은 위와 같이 구성되어 있다. 사용자 권한을 관리하는 Authentication Manager 와 데이터베이스 및 블록체인 네트워크를 통해 하부 시스템들의 데이터를 획득 및 전달하는 Service Manager, 그리고 Blood Wallet 에서 기부 기능을 가능케하는 네 가지 System 이 존재한다.

### 5.3. Certificate Scan System

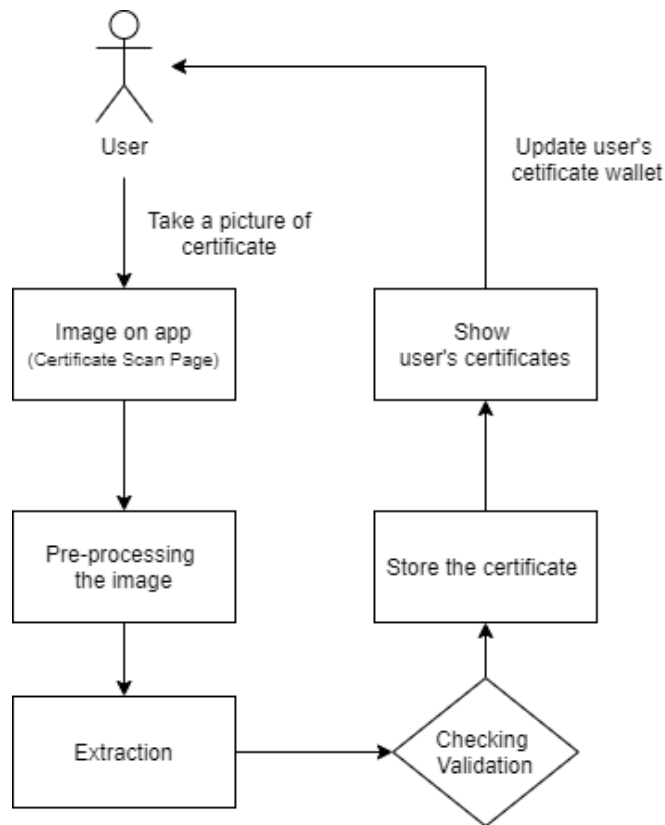


Diagram 3: Certificate Scan System Architecture

Certificate Scan System 은 헌혈 증서를 시스템에 전산화하는 시스템이다. 먼저 헌혈 증서를 촬영하고 본 시스템은 촬영한 이미지를 인자로 취한다. 이 이미지는 전처리 과정을 이후 이미지 속의 텍스트를 추출한다. 이 추출한 데이터를 기반으로 해당 헌혈 증서의 유효성을 판단하고 유효하다면 이를 사용자 계정에 반영한다.



## 5.4. Certificate Donation System

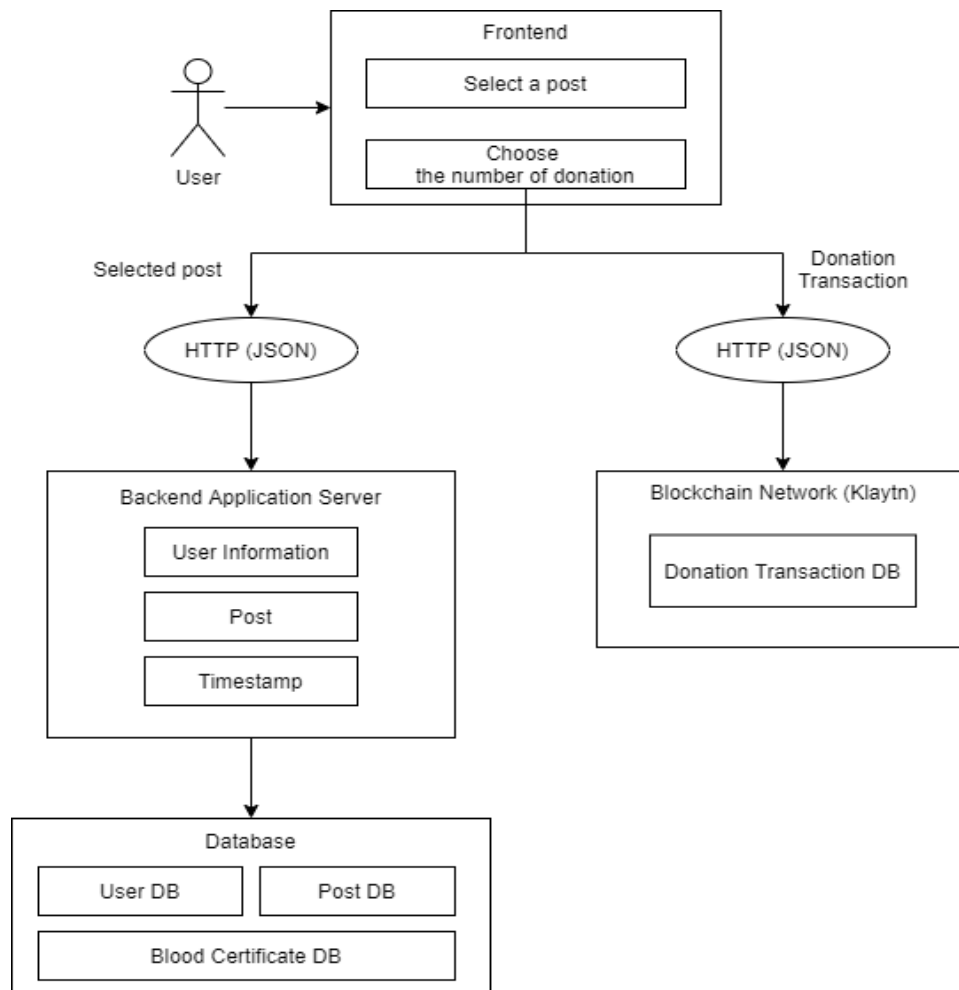


Diagram 4: Certificate Donation System Architecture

Certificate Donation System 은 사용자가 다른 사용자에게 헌혈 증서 기부를 구현한 시스템이다. 사용자는 기부하고자 하는 기부 요청 글을 선택하고 기부 버튼을 통해 헌혈 증서를 기부하게 된다. 이때 서버에 기부 시각, 사용자가 선택한 기부 요청 글, 사용자 정보를 전송하고 이를 데이터베이스에 반영한다. 이와 동시에 헌혈 증서 기부 거래 기록은 클레이튼(Klaytn) 블록체인 네트워크에 배포된 스마트 컨트랙트에 반영한다. 이를 통해 전산화되어 양도되는 헌혈 증서는 물리적인 헌혈 증서의 거래와 동일한 유효성을 가지고 보다 투명하게 사용자 간의 양도 기록을 관리할 수 있다.

## 5.5. Donation History System

Donation History System 은 클레이튼 블록체인 네트워크에 배포된 스마트 컨트랙트로부터 사용자 간 기부 기록을 사용자에게 제공하는 시스템이다. 스마트 컨트랙트의 스토리지에 사용자가 기부한 기록과 기부 받은 기록을 이용하고, 동시에 데이터베이스에서 해당 기록이 기부 요청 글과 매칭되는지 검색하여 사용자에게 기부 기록 뿐만 아니라 사용자가 기부한 사연 (기부 요청글)을 제공한다.

## 5.6. Ranking Post System

Ranking Post System 은 키워드를 기반으로 기부 요청 글의 피드를 점수를 계산하여 내림차순으로 제공하는 시스템이다. 본 시스템은 두 가지 방식으로 사용자에게 제공될 수 있다. 첫째는 사용자가 설정한 선호하는 키워드와 관련된 Ranking Post System 을 먼저 자동적인 키워드에 관한 ranking 을 제공해주는 것이고, 둘째는 사용자가 원하는 키워드를 검색함으로써 수동적으로 Ranking Post System 을 제공해주는 것이다.

사용자는 Ranking Post System 을 통해 수많은 기부 요청 글을 사용자의 관심사와 관련된 Post 를 우선적으로 확인함으로써 더 활발한 기부를 진행할 수 있다.

## 6. System Requirements Specification

이 챕터에서는 유저 요구사항 명세에서 정의된 요구사항을 바탕으로 세부적인 Functional requirements 와 Non-functional requirements 를 명시한다. 각 requirements 는 Name, description, inputs, source, outputs, destination, action, requirements, pre-condition, post-condition, side effects 를 포함해 최대한 상세히 기술한다.

### 6.1. Functional Requirements – Frontend

#### A. Ranking

Name	Ranking Function	
Description	기부 요청 Post 를 매겨진 점수 내림차순으로 띄운다.	
Inputs	Keyword_List	사용자 선호 키워드 문자열 리스트
Source (Input)	User input	
Outputs	업데이트된 기부 요청 Post List	
Destination (Output)	Android App View	
Action	사용자가 기존에 설정한 선호 키워드 혹은 새롭게 입력한 키워드를 백엔드에 보내, 백엔드가 해당 키워드들을 바탕으로 정렬된 Post List 를 전송한다. 이를 App View 상에 띄워 사용자에게 우선순위를 두어 Post 를 제공한다.	
Requirements	사용자 선호 키워드가 명시되어야 한다.	
Pre-condition	-	
Post-condition	-	
Side effects	프론트엔드는 업데이트된 Ranking Post List 에 따라 달라진다.	

Table 3: Ranking Function

## Requirements Specification

### B. Show Post Detail

Name	Show Post Detail Function	
Description	기부 요청 Post 의 세부 내용을 보여준다.	
Inputs	Post_ID	기부 요청 Post 의 String 고유 ID
Source (Input)	User input	
Outputs	기부 요청 Post 의 제목 및 세부 내용, 모금 현황 정보 객체	
Destination (Output)	Android App View	
Action	Ranking 되어 나타난 기부 요청 Post 에서 특정한 기부 요청 Post 가 사용자에게 의해 선택되면 Post 의 ID 가 백엔드에 전달되고 백엔드는 데이터베이스에서 해당 Post 의 제목, 글 내용, 모금 현황(%)을 전송한다. 이를 App View 상에 띄워 사용자에게 제공한다.	
Requirements	-	

Table 4: Show Post Detail

### C. Donate Certificates

Name	Donate Certificate	
Description	특정한 기부 요청 Post 에 헌혈 증서를 기부한다.	
Inputs	User_ID	사용자의 아이디
	Donatory_ID	기부 받는 사용자의 아이디
	CertificateID_List	기부하는 헌혈 증서 ID
Source (Input)	사용자의 버튼 클릭으로 발생한 프론트엔드 요청	
Outputs	성공/실패 메시지	
Destination (Output)	Android App View	
Action	User_ID 와 Donatory_ID 를 이용하여 각 사용자의 Public Key 를 데이터베이스에서 얻는다. 이후 클레이튼 네트워크에 배포된 스마트 컨트랙트에 Fee delegation call 을 통해 각 헌혈 증서마다 기부 트랜잭션 기록한다. 만약 기부하는 헌혈 증서 만큼의 트랜잭션이 성공적으로 블록에 담기게 되면 성공 메시지를 반환하고 그렇지 않으면 실패 메시지를 반환한다.	
Requirements	User_ID, Donatory_ID 와 이에 대응하는 Public Key 가 모두 존재.	

Table 5: Donate Certificates

## D. Scan a Certificate

<b>Name</b>	<b>Scan a Certificate</b>	
<b>Description</b>	사용자의 헌혈 증서를 촬영하여 계정에 추가한다.	
<b>Inputs</b>	Name	성명
	Sex	성별
	Birth Date	생년 월일
	Type	헌혈 종류
	Donation Date	헌혈 일자
	Blood Bank Name	혈액원명
	Serial Number	헌혈 증서의 고유 번호
<b>Source (Input)</b>	User input	
<b>Outputs</b>	Boolean 으로 나타낸 유효 여부	
<b>Destination (Output)</b>	Android App View	
<b>Action</b>	<p>사용자가 등록하지 않은 헌혈 증서를 등록할 때, 입력한 헌혈 증서의 정보를 백엔드에 넘긴다. 백엔드를 통해 헌혈 증서 발급 데이터베이스에서 해당 헌혈 증서의 유효 여부 및 사용자 본인 확인을 판단한다. 만약 헌혈 증서가 데이터베이스에 존재하지 않거나 해당 헌혈 증서가 이미 등록되었거나, 타인의 헌혈 증서일 시, 실패 메시지를 띄운다. 그렇지 않다면 성공 메시지를 띄운다.</p>	
<b>Requirements</b>	모든 input 이 존재해야 한다.	
<b>Pre-condition</b>	-	
<b>Post-condition</b>	-	
<b>Side effects</b>	-	

Table 6: Scan a Certificate

## E. Show Donation History

Name	Show Donation History Function	
Description	사용자가 이전에 기부했거나 기부 받은 내역을 조회하여 보여준다.	
Inputs	User ID	사용자의 ID
Source (Input)	User Input	
Outputs	Transaction List	
Destination (Output)	Android App View	
Action	사용자가 버튼을 누르면 백엔드에 사용자의 ID 를 전달하고 백엔드는 넘겨 받은 사용자 ID 에 매칭되는 Public Key 를 획득한다. 해당 Public key 를 통해 클레이튼 스마트 컨트랙트에 조회하여 사용자의 거래 내역 리스트를 전달한다.	
Requirements	사용자의 ID 와 Public Key 가 존재해야 한다.	
Pre-condition	None.	
Post-condition	None.	
Side effects	업데이트된 기부 거래 내역 리스트를 프론트엔드에 나타난다.	

Table 7: Donation History Function

## F. Mypage

Name	Mypage Function	
Description	사용자의 고유한 정보를 표시하는 페이지	
Inputs	ID	사용자의 ID
	Password	사용자의 PW
Source (Input)	User authentication 과 my page 버튼 클릭	
Outputs	Authenticated user information object	
Destination (Output)	Android App View	
Action	사용자의 정보를 제공하는 페이지이다. 이 페이지에서 사용자의 이름, 나이, 관심 키워드, 보유 헌혈 증서 개수 등을 확인할 수 있다. 따라서 이 페이지는 로그인을 한 사용자만 접근이 가능하다.	
Requirements	사용자의 아이디와 비밀번호가 명시되어야 한다.	
Pre-condition	권한이 없는 사용자는 접근이 불가능하다.	
Post-condition	로그인을 한 권한이 있는 사용자만 접근 가능하다.	
Side effects	프론트엔드가 사용자 개인 정보를 포함하도록 변화한다.	

Table 8: Mypage

## 6.2. Functional Requirements – Backend

### A. Login

<b>Name</b>	<b>Login Function</b>	
<b>Description</b>	사용자의 권한을 확인한다.	
<b>Inputs</b>	ID	사용자의 ID
	Password	사용자의 PW
<b>Source (Input)</b>	User input	
<b>Outputs</b>	Authenticated user information object	
<b>Destination (Output)</b>	Android App View	
<b>Action</b>	<p>사용자의 ID 를 통해 데이터베이스에서 사용자의 정보를 확인한다.          이후 사용자의 비밀번호를 해쉬하여 데이터베이스와 대조하여 로그인 성공 여부를 판단한다. 만약 로그인에 성공했다면 추가적인 사용자의 정보를 포함하는 user object 를 생성한다.</p>	
<b>Requirements</b>	사용자의 아이디와 비밀번호가 명시되어야 한다.	
<b>Pre-condition</b>	권한이 없는 사용자는 접근이 불가능하다.	
<b>Post-condition</b>	로그인을 한 권한이 있는 사용자만 접근 가능하다.	
<b>Side effects</b>	프론트엔드가 사용자 개인 정보를 포함하도록 변화한다.	

Table 9: Login Function



## Requirements Specification

### B. Certificate Scan System

Inputs	User ID	사용자 ID
	Certificate Data	헌혈증서의 정보를 포함하는 객체
Source (Input)	Backend Application	
Outputs	등록 헌혈 증서 유효 여부	
Destination (Output)	Blood Certificate Issue Database	
Action	헌혈 증서 데이터 객체를 헌혈 증서 발급 데이터베이스와 비교하여 해당 헌혈 증서의 유효성을 판단한다. 더불어 사용자의 ID 를 통해 얻은 사용자의 정보와 헌혈 증서의 정보를 대조하여 실제 사용자의 헌혈 증서가 맞는지 확인한다.	
Requirements	-	
Pre-condition	-	
Post-condition	-	
Side effects	-	

Table 10: Certificate Scan System

### C. Certificate Donation System

Inputs	Transaction Object	기부 트랜잭션 객체
Source (Input)	Backend Application	
Outputs	파싱된 블록체인 네트워크 전송 Response Object	
Destination (Output)	User Database	
Action	클레이튼 블록체인 네트워크에 Fee delegation call 을 통해 기부 트랜잭션을 발생한다. 이후 Fee delegation 된 트랜잭션이 이후 성공적으로 네트워크에 반영되었는지 response object 를 파싱한 결과를 사용자 데이터베이스에 반영한다.	
Requirements	Transaction Object 에 sender, receiver, certificate code 모두 존재.	
Pre-condition	-	
Post-condition	-	
Side effects	사용자 데이터베이스에서 사용자의 보유 헌혈 증서가 변경된다.	

Table 11: Certificate Donation System

## D. Donation History System

Inputs	User ID	사용자의 ID
Source (Input)	Backend Application	
Outputs	사용자의 기부 트랜잭션 리스트	
Destination (Output)	Fronted Application	
Action	사용자의 ID 에 해당하는 Public Key 를 사용자 데이터베이스에 조회하고, 얻은 Public Key 를 클레이튼 스마트 컨트랙트에 조회하여 해당 Public Key 로 발생한 기부 트랜잭션을 전달 받는다.	
Requirements	User ID 와 매칭되는 Public Key 가 존재해야 한다.	
Pre-condition	-	
Post-condition	-	
Side effects	기부 트랜잭션 데이터가 업데이트될 경우 현재 표시되고 있는 기부 내역 정보에 변동이 발생한다. 따라서 프론트엔드에서의 다음 요청시 업데이트된 기부 내역을 리턴해야 한다.	

Table 12: Donation History System

## 6.3. Non-functional Requirements

### A. Product requirements

#### 1. Usability

성공적인 어플리케이션을 위해 필수적인 요구사항이다. 시스템은 직관적인 사용자 인터페이스를 제공해야 하며, 동시에 단순한 조작으로 사용자 친화적이어야 한다. Blood Wallet에서는 Navigator View의 버튼들로 구현하였기 때문에, 사용자는 쉽고 빠르게 어플리케이션의 여러 기능들을 확인할 수 있다.

또한 사용자에게 복잡한 블록체인 경험을 막고자 Fee delegation을 통해 본 어플리케이션이 블록체인을 이용하고 있는 것을 숨길 수 있도록 설계하였다. 이로써 기부 거래의 투명성과 쉬운 사용법을 제공한다.

#### 2. Performance

블록체인 네트워크를 통해 물리적인 헌혈 증서를 처리하기 때문에 전통적인 서버 구조보다 느린 Latency를 처리해야 한다. 블록체인을 통해 투명성을 확보할 수 있지만 사용자에게 어플리케이션의 큰 Latency를 줄이기 위한 알고리즘이 필요하다. 또한 헌혈 증서를 촬영하여 등록할 때, 이미지 처리 시 여러 예외 케이스가 발생할 수 있기 때문에 이를 고려하여 사용자의 만족도를 최대화해야 한다.

#### 3. Dependability

Blood Wallet은 모든 트랜잭션이 Atomic하게 처리되도록 함으로써 Dependability를 최대화하였다. 만약 헌혈 증서를 전달하거나 헌혈 증서 기부 요청 글을 작성하는 중간에 시스템이 비정상적으로 종료될 때, 서버와 블록체인 네트워크는 네트워크 연결을 잃기 때문에 비동기적인 수정과 트랜잭션 발생은 시스템에 반영되지 않는다. 또한 모바일 기기의 문제로 어플리케이션이 종료된다면 서버가 빠르게 다시 네트워크 연결을 시도하여

정상적인 상태로 되돌아간다. 그리고 블록체인 네트워크는 카카오 자회사 Klaytn 을 이용하고, 한글을 인식률이 높은 Clova OCR API 를 도입함으로써 헌혈 증서 등록 시스템과 헌혈 증서 기부 트랜잭션 관리의 높은 정확성과 신뢰도를 얻을 것이다.

#### 4. Security

헌혈 증서 기부를 블록체인 네트워크를 통해 수행하기 때문에 Public key 와 Private Key 가 필요하다. 이때 Private Key 는 개인의 고유한 비밀 키이기 때문에 서버에서 관리하지 않고 모바일 기기 로컬 스토리지에 저장하여 관리한다.

### B. Organization requirements

#### 1. Operational

현재 지속적으로 운영할 수 있는 하드웨어 자원이 없으므로, 상용 클라우드 서버에 배포할 것을 전제로 한다. 따라서 다양한 클라우드 서버에 배포할 수 있도록 hardware-dependent 한 구현을 자제하고, 서버의 주소가 바뀌어도 대응할 수 있게 시스템을 설계해야 한다.

#### 2. Development

보다 빠른 기간에 시스템 개발이 가능하도록 Parallel Development 를 도입한다. 이를 위해 개발 프로세스를 plan-driven 으로 설계하고, 개발 단위를 프론트엔드와 백엔드로 나누며, 프론트엔드와 백엔드는 서로 독립적으로 구동할 수 있게 설계해야 한다.

### C. External requirements

#### 1. External System Policy

본 시스템은 헌혈 증서를 기부하는 시스템이므로 헌혈 증서의 양도가 어떠한 경제적인 거래로 이어질 수 있다. 또한 보건 복지부는 혈액관리법 제 2 조 제 4 호에 언급된 무상 헌혈을 원칙으로 하기 때문에 헌혈 증서 전산화 추진은 심도 깊은 논의가 필요하다고 언급하였다. 따라서 사용자의 기부가 어떠한 경제적 이득으로 이어지지 않도록 주의해야 한다.

#### 2. Personal Information Protection

본 시스템은 헌혈 증서 관리 및 기부 기능을 제공하기 위해 회원제로 운영한다. 따라서 사용자의 개인정보를 수집하며, 개인 정보가 제대로 보호되지 않을 경우 민형사상의 책임을 질 가능성이 있다. 따라서 가입 시 사용자 식별에 필요한 최소한의 정보만을 제공받고, 개인정보가 유출되지 않도록 충분한 노력을 기울여야 한다.

## 6.4. Scenario Examples

### A. Sign Up Scenario

#### 1. Initial Assumption

사용자는 본 시스템에 계정을 가지고 있지 않고 새롭게 가입하고자 한다.

#### 2. Normal flow of events

사용자는 전화번호를 입력하고 인증 번호를 입력하여 인증을 완료한다. 이후 이름, 생년월일, 성별, 나이, 아이디, 비밀번호를 입력한다. 이때 데이터베이스는 아이디의 중복 여부를 확인하여 아이디가 유일성을 가질 수 있도록 한다. 이후 자동적으로 Caver 모듈을 통해 사용자 아이디에 매핑되는 Public Key 와 Private Key 를 생성한다.

#### 3. What can go wrong

사용자가 이름, 생년월일 등 헌혈 증서의 소유권을 확인할 수 있는 개인 정보를 임의로 입력할 가능성이 있다. 이를 본인 인증 모듈로 해결할 수 있으나, 해당 서비스는 사업자 등록을 전제 하기 때문에 전화번호 인증으로 인증 과정을 간소화 하였다.

또한 이미 계정이 있는 사용자이면 이미 데이터베이스에 정보가 있기 때문에 에러 메시지를 생성한다.

#### 4. System state on completion

사용자는 사용자 권한을 얻게 되고 이후 Blood Wallet 이 제공하는 서비스를 이용할 수 있게 된다.

### B. Certificate Donation Scenario

#### 1. Initial Assumption

사용자는 본 시스템에 이미 등록되어 있고, 로그인을 통해 시스템에 접속하였다. 사용자는 자신이 보유하고 있는 헌혈 증서를 기부하고자 하지만 누구에게 기부할지 알고 있지 않다.

#### 2. Normal flow of events

사용자는 본 시스템에 접속해 아이디와 비밀번호를 입력하여 로그인을 하면 총 6 개의 버튼을 확인할 수 있다. 그 중 "Donate" 버튼을 누르면 여러 헌혈 증서 기부 요청 글 리스트가 표시된다. 사용자는 그 중 자신이 기부하고자 하는 Post 를 선택하면 전체 글과 "기부하기" 버튼을 확인할 수 있다. "기부하기" 버튼을 누르면 데이터베이스에서 사용자의 보유 헌혈 증서 개수를 검색하고 사용자가 해당 사연자에게 최대 자신 보유량까지 선택적으로 기부할 수 있도록 한다. 사용자는 최종적으로 기부를 하게 되면 기부 트랜잭션이 블록체인 네트워크에 반영된 후 성공적으로 기부되었음을 나타내는 메시지와 함께 기부를 완료하게 된다.

#### 3. What can go wrong

블록체인 네트워크에 트랜잭션을 기록하기 위한 수수료를 제 3 자를 통해 지불하도록 설계하였다. 이를 Fee delegation 이라고 하는데 이 과정에서 제 3 자의 계좌에 잔고가 부족할 수 있기 때문에 Fee delegation 을 하는 계좌의 잔고는 운영 가능하도록 관리해야 한다.

#### 4. System state on completion

사용자가 헌혈 증서를 기부하게 되면 블록체인에 기부를 한 주체, 기부를 받는 주체, 헌혈 증서 코드 총 세 가지 정보가 보관되며 이후 사용자가 기부 내역을 조회할 때 이를 이용하여 내역을 확인하게 된다.

## 7. System Models

이번 챕터에서는 각 시스템 컴포넌트와 전체 시스템, 그리고 시스템을 둘러싼 환경 사이의 관계를 여러 가지 Diagram 을 통해 기술한다.

### 7.1. Context models

#### A. Context Diagram

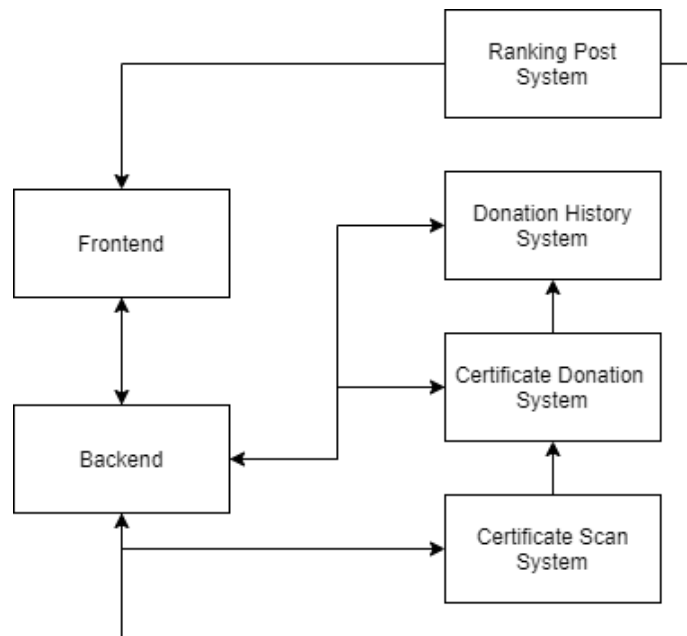


Diagram 5: Overall context diagram



# Requirements Specification

## B. Process Diagram

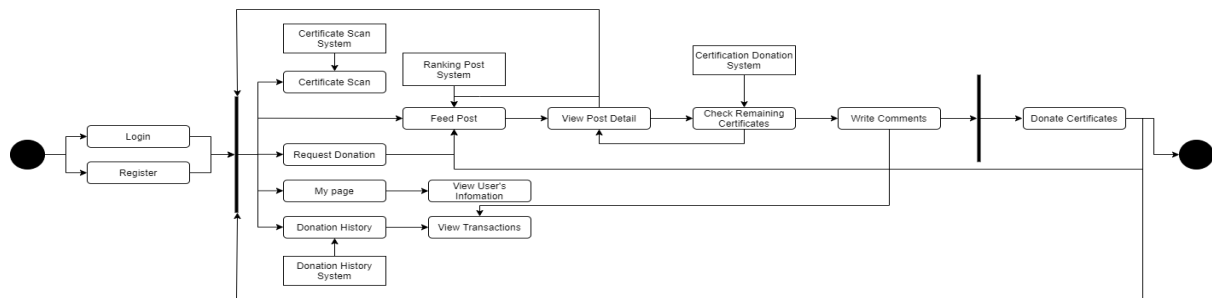


Diagram 6: Overall process diagram

## 7.2. Interaction models

### A. Use case Diagram

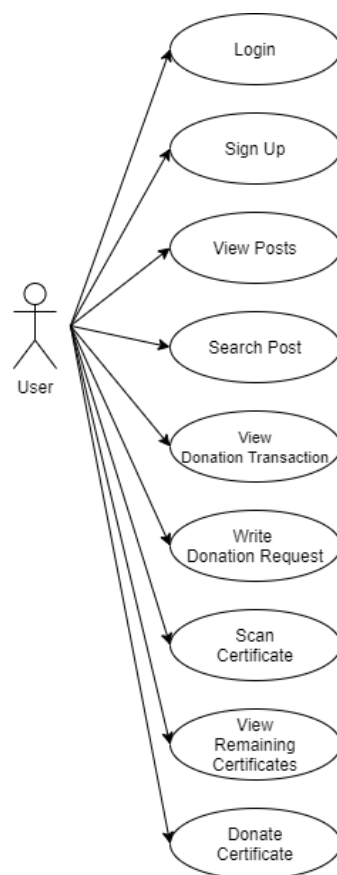


Diagram 7: Use case diagram

## B. Tabular Description of Use case Diagram

### 1. Login

<b>Use Case</b>	<b>Login</b>
<b>Actor</b>	User, Database(User)
<b>Description</b>	접속한 사용자가 시스템에 등록되어 있는 유저 정보와 일치하는지 판단하는 과정이다.
<b>Trigger</b>	사용자 ID 와 PW 를 입력한다.
<b>Success Response</b>	사용자 ID 가 데이터베이스에 존재하고, Password 의 Hash 값이 데이터베이스의 값과 일치하면 사용자 정보와 접속 권한을 포함한 객체를 돌려준다.
<b>Failure Response</b>	사용자 ID 가 데이터베이스에 없을 경우 Sign Up 페이지로 이동하며, Password 가 일치하지 않을 경우 오류 메시지를 발생시킨다.

## Requirements Specification

### 2. Sign Up

<b>Use case</b>	<b>Sign Up</b>
<b>Actor</b>	User, Database(User)
<b>Description</b>	사용자의 계정을 생성하는 과정이다.
<b>Trigger</b>	전화 번호 인증 및 성명과 같은 개인 정보 입력한다.
<b>Success Response</b>	전화 번호 인증을 성공하고, 아이디가 고유한 값을 가지며 모든 개인 정보 필드가 입력이 되면 사용자 정보를 데이터베이스에 반영하고 접속 권한을 포함한 객체를 생성한다.
<b>Failure Response</b>	이미 사용자 정보가 데이터베이스에 있거나 전화번호 인증에 실패하면 오류 메시지를 발생시킨다.

### 3. View Posts

<b>Use case</b>	<b>View Posts</b>
<b>Actor</b>	User, Database(Post), Post Ranking System
<b>Description</b>	헌혈 증서 기부 요청 글을 띄운다.
<b>Trigger</b>	사용자가 "Donate" 버튼을 누른다.
<b>Success Response</b>	사용자 선호 Keyword 를 통해 Post 를 정렬하여 보여준다.
<b>Failure Response</b>	-

## 4. Search Post

<b>Use case</b>	<b>Search Post</b>
<b>Actor</b>	User, Database(Post)
<b>Description</b>	검색어를 통해 기부 요청 글을 검색한다.
<b>Trigger</b>	사용자가 검색 키워드를 입력하고 검색 버튼을 누른다.
<b>Success Response</b>	검색 키워드를 포함한 글을 선별하여 돌려준다.
<b>Failure Response</b>	해당하는 글이 없을 경우 빈 목록을 돌려주고 검색 결과가 없다는 오류 메시지를 발생시킨다.

## 5. View Donation Transaction

<b>Use Case</b>	<b>View Donation Transaction</b>
<b>Actor</b>	User, Database(Post, User), Blockchain
<b>Description</b>	사용자의 기부 내역을 보여준다.
<b>Trigger</b>	사용자가 "기부한 내역" 혹은 "기부 받은 내역" 버튼을 누른다.
<b>Success Response</b>	사용자의 ID 를 통해 데이터베이스에서 Public Key 를 얻고 이를 스마트 컨트랙트에 조회하여 사용자와 관련된 거래 내역을 전달 받는다. 이후 각 트랜잭션과 Post 데이터베이스와 비교하여 각 트랜잭션이 어떤 Post 를 통해 진행되었는지 가공하여 돌려준다.
<b>Failure Response</b>	내역이 없을 시, 기부 내역이 없다는 오류 메시지를 발생시킨다.

## 6. Write Donation Request

Use case	Write Donation Request
Actor	User, Database(Post)
Description	사용자가 헌혈 증서 기부 요청 글을 작성한다.
Trigger	사용자가 기부 요청하기 버튼을 클릭하고 제목과 내용을 입력한다.
Success Response	Post 데이터베이스에 사용자의 새로운 글을 등록한다.
Failure Response	제목과 내용이 없으면 에러 메시지를 발생시킨다.

## 7. Scan Certificate

Use case	Scan Certificate
Actor	User, Database(User, Blood Certificate Issue)
Description	헌혈 증서를 촬영하여 등록한다.
Trigger	"헌혈 증서 스캔"을 클릭하여 헌혈 증서를 촬영하고 빠진 정보는 직접 기입한다.
Success Response	헌혈 증서 발급 데이터베이스에 해당 헌혈 증서가 시스템에 등록 되었음을 명시하고 사용자 데이터베이스에 헌혈 증서 보유 정보를 업데이트한다.
Failure Response	발급되지 않았거나 이미 등록되었거나 혹은 타인의 헌혈 증서일 경우 등록하지 않고 오류 메시지를 발생시킨다.

## 8. View Remaining Certificates

<b>Use case</b>	<b>View Remaining Certificates</b>
<b>Actor</b>	User, Database(User, Blood Certificate Issue), Blockchain
<b>Description</b>	사용자의 보유 헌혈 증서 개수를 확인한다.
<b>Trigger</b>	MyPage 를 클릭한다.
<b>Success Response</b>	사용자의 Public Key 를 통해 블록체인에 배포된 스마트 컨트랙트에 기부된 헌혈 증서를 조회하고 해당 헌혈 증서는 헌혈 증서 발급 데이터베이스에서 무효화 시키고 사용자 데이터베이스에서 보유 헌혈 증서 개수를 업데이트한다.
<b>Failure Response</b>	-

9. Donate Certificate

<b>Use case</b>	<b>Donate Certificate</b>
<b>Actor</b>	User, Database(User), Blockchain
<b>Description</b>	특정 기부 요청 글을 통해 사용자가 헌혈 증서를 기부한다.
<b>Trigger</b>	사용자가 특정 기부 요청 글을 클릭하고 기부하기를 선택하여 헌혈 증서를 기부한다.
<b>Success Response</b>	기부하는 사용자, 기부 받는 사용자의 Public Key, 헌혈 증서 고유 코드를 블록체인 스마트 컨트랙트에 트랜잭션을 올리고 이후 완료되면 사용자 데이터 베이스에서 보유 헌혈 증서의 개수를 업데이트한다.
<b>Failure Response</b>	트랜잭션 수행 과정 중 시스템 실패가 일어나면 기부 트랜잭션을 데이터베이스에 반영하지 않고 에러 메시지를 발생시킨다.

### 7.3. Behavioral models

이 장에서는 본 시스템에서 가장 중점으로 두는 하위 시스템인 'Ranking system' 과 'Text analysis system'에 초점을 두고 behavioral models 을 작성하였다.

#### A. Certificate Scan System DFD

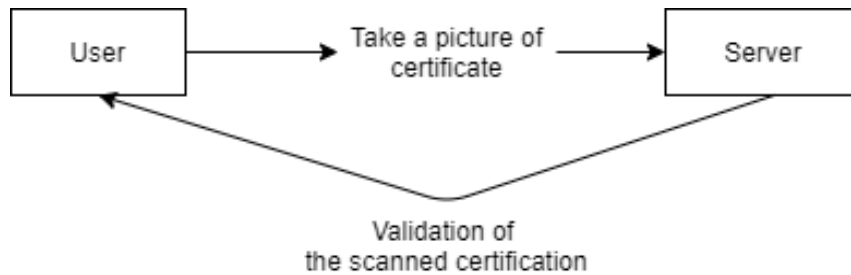


Diagram 8: Data flow diagram of certificate scan system

#### B. Certificate Donation DFD

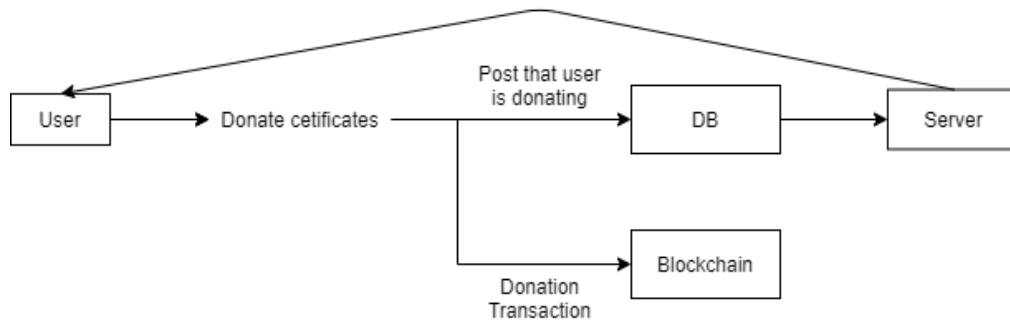


Diagram 9: Data flow diagram of certificate donation system



C. Donation History System DFD

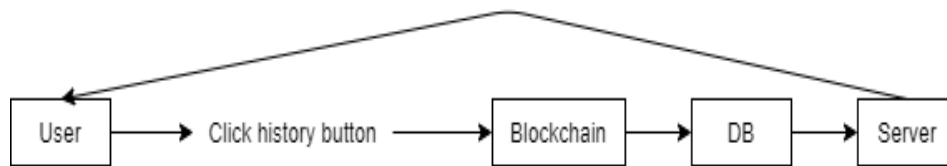


Diagram 10: Data flow diagram of donation history system

## 8. System Evolution

이번 챕터에서는 본 시스템이 가지고 있는 경계와 이로 인해 발생할 수 있는 한계에 대해 설명한다. 이후 발생 가능한 변화에 대해서도 예측하여, 해당 변화로 인해 발생하는 변경사항을 어떻게 반영할 수 있을지를 살펴본다. 이를 통해 시스템 설계자는 변화에 대응할 수 있는 시스템을 개발할 수 있다.

### 8.1. Limitation and Assumption

Blood Wallet 은 헌혈증의 간편한 관리를 가능하게 함으로써 현재 낮은 수준에서 머무르고 있는 환급율을 높일 수 있다. 즉 헌혈증의 사용을 더 활성화시킬 수 있다. 하지만 동시에 우려되는 점도 있다. 현재는 헌혈증을 직접 기부도 가능하지만 적십자 등의 혈액업무 관련 단체에 기부를 하면 단체에서 헌혈증을 기부 받는 사람에 대한 최소한의 검증을 한다. Blood Wallet 시스템 하에서는 그러한 소득정보, 가족관계정보 같은 세부적인 검증절차는 힘들 수 있기 때문에 헌혈증이 꼭 필요하지 않은 사람들에게 갈 수 있다는 점이 우려된다. 그럼에도 불구하고, 높아지는 헌혈증의 사용률을 고려해본다면 현 상황보다는 개선될 것으로 기대한다.

Blood Wallet 은 헌혈환급적립금이라는 제도 하에 제작되는 시스템이다. 하지만 무상 헌혈 정신과 위배된다는 이유 등으로 헌혈환급적립금 제도 자체에 대한 비판도 많고 나중에는 없어질 수도 있는 제도라는 점을 인식해야 한다. 제도가 사라진다고 해도 헌혈증에 대한 수혈비용 환급 기한은 제한되어 있지 않으므로 언제든지 수혈비용 청구 시 환급해야 할 의무가 있기 때문에 사용가치는 계속 있다. 다만 그 역할이 축소될 가능성이 있기 때문에 추후 다른 기능들의 추가를 고려해볼 필요가 있다.

## 8.2. Evolution System

### A. 종이 헌혈 증서

현재 종이로 발급되어지는 헌혈 증서에 초점을 맞추고 있지만, 전산화를 하면 굳이 신규발급을 종이로 할 이유가 없어진다. 따라서 종이 헌혈 증서 신규발급 자체가 없어질 수 있다는 점을 인식해야 한다. 이런 경우에도 현재 발급되어 있는 헌혈증을 데이터베이스에 저장하는 역할로써 OCR(광학문자해독) 모듈은 계속해서 필요하지만, 혈액관리자 입장에서 헌혈증을 종이로 발급하는 것이 아니라 데이터베이스를 바로 수정하게 할 수 있는 시스템도 필요할 것으로 예측된다. 즉 헌혈 증서 발급 기관 입장의 시스템과 짝을 이뤄서 운용되어질 수 있다.

### B. 헌혈 통합 어플리케이션

Blood Wallet 은 매년 헌혈자수가 150 만명 정도 되는 것을 고려하면 서비스 시작부터 큰 수의 사용자를 확보할 수 있을 것으로 예상되기 때문에 추가적인 기능을 제공하면 사회적 효용이 큰 폭으로 늘어날 수 있다. 이 앱은 헌혈 증서 관리 뿐 아니라 더 나아가서 헌혈에 대한 정보를 제공, 헌혈에 대한 접근성 향상, 헌혈 독려 등의 기능이 있는 헌혈 통합 앱으로 발전 가능하다.

## 9. Appendices

이번 챕터에서는 앞서 본문에서 다루지 못한 시스템 개발에 관련된 세부 정보를 다룬다. 시스템 운영 환경에 필요한 하드웨어 요구사항, 시스템에 사용되는 데이터베이스의 개략적 구조 등에 대해 기술한다.

### 9.1. Hardware requirements

본 시스템은 안드로이드 애플리케이션으로 개발되고 있다. 그러므로 본 시스템을 사용하기 위해서는 최소 API16: Android OS 4.1 (Jelly bean) 이상을 지원하는 모바일 디바이스를 갖추어야 하며, 백엔드 서버에서 기부 요청 글 및 기부 트랜잭션을 업데이트하므로 네트워크에 연결되어 있어야 한다.

## 9.2. Database requirements

본 시스템은 NoSQL Database 를 사용해 데이터를 저장한다. 데이터베이스를 이루는 각 엔티티는 다음과 같다. 세부적인 데이터베이스 설계는 Design Architecture 문서에서 다루게 된다.

### A. User

시스템의 사용자 정보를 담고 있는 엔티티

### B. Post

헌혈 증서 기부 요청 글의 정보를 담고 있는 엔티티

### C. Blood Certificate Issue

발급된 헌혈 증서의 정보를 담고 있는 엔티티

### D. Authority

사용자의 권한 정보를 담고 있는 엔티티

### E. Donation

헌혈 증서 기부 정보를 담고 있는 엔티티

### F. Keyword

검색 시스템에 필요한 키워드에 대한 정보를 담고 있는 엔티티

## 10. References

임술. (2018). 헌혈자 수혈받으면 헌혈수가 2500 원씩 적립...불필요한 적립금만 445 억원.

*MEDI:GATE NEWS.*