



# Used2Block

## REQUIREMENTS SPECIFICATION

Student Number	Name
2015314213	이재원 Jae Won Lee
2013313737	김무성 Mu Sung Kim
2015318385	김현우 Hyun Woo Kim
2015310250	김동환 Dong Hwan Kim
2016315146	이상우 Sang Woo Lee

# Content

<b>1. Preface</b>	8
<b>1.1. Readership</b>	8
1.1.1. User Requirement Readership	8
1.1.2. System Requirement Readership	8
<b>1.2 Document Structure</b>	8
1.2.1. Introduction	9
1.2.2. Glossary	9
1.2.3. User Requirements Specification	9
1.2.4. System Architecture	9
1.2.5. System Requirements Specification	9
1.2.6. System Models	9
1.2.7. System Evolution	10
1.2.8. Appendices	10
1.2.9. Index	10
<b>2. Introduction</b>	11
<b>2.1. Needs</b>	11
<b>2.2. System Overview</b>	13
<b>2.3 Expected Effects</b>	14
2.3.1. 품질 보증	14
2.3.2. 신뢰성 강화	14
2.3.3. 사용자 중심	14

2.3.3. 안정성 강화 .....	14
2.3.4. 빠른 거래 속도 .....	14
<b>3. Glossary .....</b>	<b>16</b>
<b>4. User Requirements Definition .....</b>	<b>17</b>
<b>4.1. Functional Requirements .....</b>	<b>17</b>
4.1.1. Sign Up.....	17
4.1.2. Login/ Logout .....	18
4.1.3. User's Profile Management.....	18
4.1.4. Creating Wallet.....	18
4.1.5. Searching.....	19
4.1.6. Selling Product.....	20
4.1.7. Buying Product .....	20
4.1.8. Manage product .....	21
4.1.9. Verifying Transaction .....	21
<b>4.2 Non-functional Requirements .....</b>	<b>23</b>
<b>4.2.1 Product Requirement.....</b>	<b>23</b>
4.2.1.1. Security.....	23
4.2.1.2. Reliability.....	23
4.2.1.3. Speed.....	23
4.2.1.4. Usability. ....	24
<b>4.2.2 Organization Requirement .....</b>	<b>24</b>
4.2.2.1. Delivery Requirement.....	24

4.2.2.2. Implementation Requirement.....	24
<b>4.2.3 External requirements.....</b>	<b>24</b>
4.2.3.1. Interoperability .....	24
4.2.3.2. Ethical Requirement .....	25
4.2.3.3. Legislative Requirement .....	25
<b>5. System Architecture .....</b>	<b>26</b>
5.1 Overall Architecture .....	26
5.2. Frontend Architecture .....	27
5.3. Backend Architecture .....	28
5.4. ETH Blockchain System.....	29
5.5. Application System .....	30
5.5.1 Transaction Management.....	30
5.5.2 User Management.....	30
5.5.3 Product Management .....	30
5.5.4 Cryptocurrency Management.....	30
<b>6. System Requirements Definition .....</b>	<b>31</b>
6.1. Functional Requirements .....	31
6.1.1. Sign Up (Register) .....	31
6.1.2. Login.....	32
6.1.3. Logout .....	32
6.1.4. User's Profile Management.....	33
6.1.5. Creating Wallet.....	34

6.1.6. Searching.....	35
6.1.7. Selling Item .....	35
6.1.8. Buying Product .....	36
6.1.9. Validation .....	37
6.1.10. Manage Item.....	38
6.1.11. Item Page .....	38
6.1.12. Verifying Transaction.....	39
<b>6.2 Non-functional Requirements.....</b>	<b>41</b>
<b>6.2.1. Product Requirement .....</b>	<b>41</b>
6.2.1.1 Security.....	41
6.2.1.2. Reliability.....	41
6.2.1.3. Speed.....	42
6.2.1.4. Usability.....	42
<b>6.2.2. Organization Requirement .....</b>	<b>43</b>
6.2.2.1. Delivery Requirement.....	43
6.2.2.2. Implementation Requirement.....	43
<b>6.2.3. External Requirements .....</b>	<b>44</b>
6.2.3.1. Interoperability .....	44
6.2.3.2. Ethical Requirement .....	44
6.2.3.3. Legislative Requirement.....	45
<b>6.3 System Scenario .....</b>	<b>46</b>
<b>6.3.1. 서비스 가입/접속/접속_종료 시나리오.....</b>	<b>46</b>

6.3.1.1. 서비스 가입 .....	46
6.3.1.2. 로그인 .....	47
6.3.1.3. 로그아웃 .....	48
<b>6.3.2 가상 Wallet 관리 시나리오 .....</b>	<b>49</b>
6.3.2.1. Wallet 관리 .....	49
<b>6.3.3 거래 시나리오 .....</b>	<b>501</b>
6.3.3.1. 판매자의 상품 등록 .....	501
6.3.3.2. 구매자의 검색 및 구매 주문 .....	512
6.3.3.3. Blockchain 거래 검증 과정 .....	523
6.3.3.4. 해당 거래 모델에 대한 참조용 도식 .....	534
<b>7. System Model .....</b>	<b>556</b>
<b>7.1. Context Models .....</b>	<b>556</b>
7.1.1. Context Diagram .....	556
<b>7.2. Process Diagram .....</b>	<b>567</b>
7.2.1. Sign Up Process .....	567
7.2.2. Overall Process .....	578
<b>7.3. Interaction Models .....</b>	<b>59</b>
<b>7.3.1. Use Case Diagram .....</b>	<b>59</b>
<b>7.3.2. Tubular description for Each Use Case .....</b>	<b>60</b>
7.3.2.1. Sign Up .....	60
7.3.2.2. Login .....	59
7.3.2.3. Search .....	60

7.3.2.4. View Item Details .....	60
7.3.2.5. Purchase Item .....	61
7.3.2.6. Write Review.....	61
7.3.2.7. Post Item .....	62
7.3.2.8. Delete Item.....	62
7.3.2.9. Verify Transaction.....	62
<b>7.4. Behavioral Models .....</b>	<b>63</b>
7.4.1. Data-Driven Modeling.....	63
<b>8. System Evolution.....</b>	<b>64</b>
8.1. 경매 시스템 .....	64
8.2. 품질 보증 시스템 .....	64
8.3. 상품 추천 시스템.....	65
8.4. 개인정보 보호를 위한 hash 값 활용 .....	65
8.5 모바일 어플리케이션 연동 .....	65
<b>9. Appendix .....</b>	<b>66</b>
<b>10. Index.....</b>	<b>67</b>
10.1. Figures.....	67
10.2. Tables.....	67
10.3. Diagrams .....	68
<b>11. References.....</b>	<b>70</b>





# 1. Preface

This chapter comprises of expected readership of the document and gives overall introduction to remaining chapters.

## 1.1. Readership

This document aims two kinds of readership.

### 1.1.1. User Requirement Readership

Users of our system are expected to read the user requirement. The corresponding chapter is written from their point of view. A user would not use our system when the user requirement is full of formal and technical specification. Thus, natural language is used to write the specification by developers in the team with potential users in mind.

### 1.1.2. System Requirement Readership

Developers of our system is expected to read and understand the system requirement deeply. The operational constraints and system functionalities are represented using structural language. The system requirement would be used in system design and implementation as a reference, therefore it must be done in a completely formal document.

## 1.2 Document Structure

### 1.2.1. Introduction

The domain is defined. The market analysis is the main highlight of this chapter. The structure of the system and functionalities are explained briefly. Moreover, expected effects of the system implementation is briefed.

### 1.2.2. Glossary

The technical terminologies used throughout the document are defined with examples based on our system. This chapter is for all stakeholders of our system to read, thus the terminologies are explained in detail.

### 1.2.3. User Requirements Specification

Functional and non-functional requirements of our system are demonstrated from user's perspective. Diagrams are used along with natural language to help user's understanding.

### 1.2.4. System Architecture

Architecture of our system is demonstrated. The functionalities are explained based on subsystems.

### 1.2.5. System Requirements Specification

Based on the user requirements specification, functional and non-functional requirements are defined. Diagrams in this chapter would be a good reference throughout system design, architecture design and implementation of our system.

### 1.2.6. System Models

Components of our system, relationship among the components and

external environment are expressed in diagrams.

#### 1.2.7. System Evolution

Limitation of system implementation and operational constraints are explained. Expected changes in the market are mentioned with possible solutions.

#### 1.2.8. Appendix

Materials that are exempted from the document are combined into this chapter.

#### 1.2.9. Index

Diagrams, charts and figures used in the document are listed with indexes.

## 2. Introduction

This chapter defines the domain of our system. The market analysis is explained with figures. The structure of the system and functionalities are briefed.

### 2.1. Needs

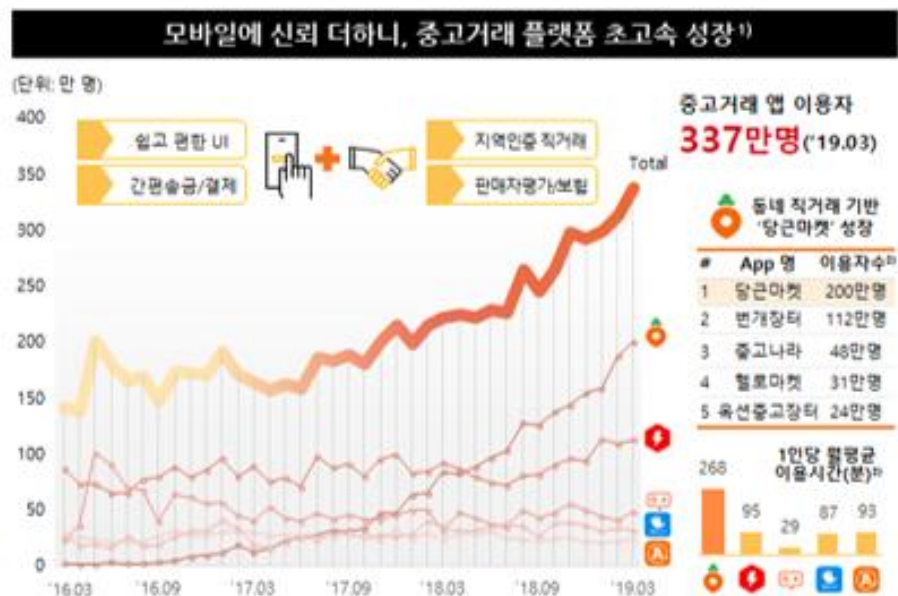


Figure 2-1. 모바일 중고거래 시장 조사(닐슨 코리아)

미디어 조사기관인 닐슨 코리아에 따르면, 현재 중고거래 앱 이용자는 점차 증가하여 337만명에 이른다고 한다. Figure 1처럼 특히 지역 기반 거래를 중요시하는 당근마켓의 경우 사용자가 200만명에 이른다는 통계자료도 있다. 또한 번개장터의 경우 2018년 연간 거래액이 2591억에 이르렀고 우리나라에서 제일 큰 규모의 중고나라의 경우 2018년 연간 거래액이 3421억으로 조사되었다. 중고나라의 모바일 앱은 550만건의 다운로드 수 기록도 보유하고 있다.

시장이 커짐에 따라 이에 관련된 문제도 드러나고 있다. 현재 중고시장에서의 큰 문제는 판매자를 신뢰할 수 없다는 것이다. 정보의 비대칭성 때문에 판매하는

제품이 허위 매물인지 아닌지 판단할 수 있는 수단이 전혀 없기 때문이다. 또한 해당 판매자가 실제로 매물을 보냈는지, 또는 벽돌을 보냈는지 확인할 방법도 없다.

기존 중고나라에서는 안전거래 제도가 도입이 되었다. 중고나라 사업자가 거래 중계자 역할을 대신해 거래를 참여하고, 구매자가 선 대금 결제 후 제품 수령으로 구매 확정 버튼을 누르면 판매자에게 대금을 지급하는 제도이다. 중고나라 사업자는 이 제도를 통해 거래에 대한 수수료를 받는다. 예를 들자면, 신용카드 결제 시 사업자는 판매금액의 3.74%의 수수료를 챙겨간다고 한다.

이 방법의 문제점은 돈을 보유하는 번거롭고 긴 과정 때문에 이를 판매자가 기피하기도 하고, 안전거래라고 할 지라도 물건이 도착하기 전까지는 상태가 괜찮은지 확인할 수 있는 방도가 없다. 흔히 레몬이라는 속어로 비유되기도 한다.



Figure 2-2. 모바일 중고거래 어플리케이션의 성장 추이

만약 거래 과정에서 분쟁이 생겼더라도, 중고거래는 개인과 개인의 거래이기 때문에 법적으로 보호받기도 힘든 상황이다. 사기를 당해 물건을 못 받았다면 사기죄로 처벌할 수 있지만, 물건에 판매자가 언급하지 않은 결함이 있거나 혹은 문제가 발생했을 경우, 현재 한국에서는 소비자가 보호받을 방법이 전혀 없는 상황이다. 이런 레몬 시장에서는 구매자 스스로가 하자가 있는 제품의 중고거래를 대비에 신중에 신중을 기해야 할 뿐이다.

중고나라 외에도 당근마켓에서는 직거래를 통해 문제를 해결하려고 했다. 현재 사는 지역의 매물만 보여줌으로써 직거래를 유도하고, 상품을 직접 눈으로 확인

하며 구매하는 과정을 거쳐 안정성을 획득한다. 하지만, 세탁기, 공기청정기 등 가전제품과 같은 무거운 물건의 경우, 직거래를 통해 거래하기 어려운 게 사실이다. 실제로 구매자와 판매자가 서로 시간을 맞추어 만나는 것도 번거롭다는 의견들이 많았다.

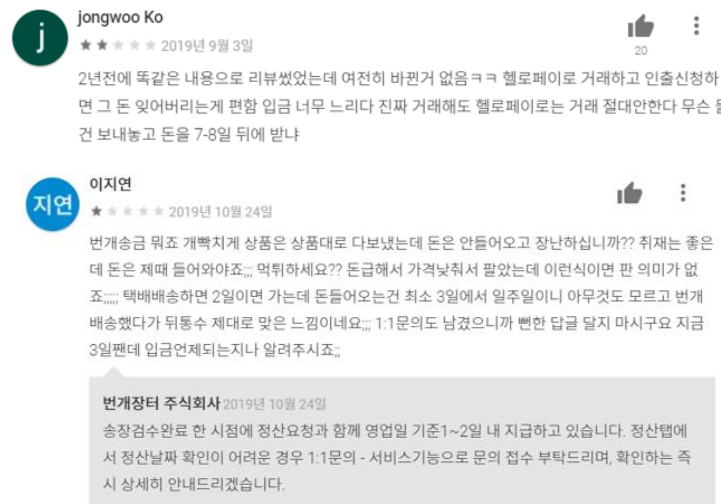


Figure 2-3. 기존 중고거래 앱의 문제점-자체 화폐 시스템의 불편함(Google 플레이스토어)

## 2.2. System Overview

Used2Block은 PoS 알고리즘을 차용한 새로운 PoC 알고리즘을 중고 시장 거래에 도입함으로써 기존 중고거래시장에 주요 문제점이었던 신뢰성을 강화한다. 또한 안전거래 과정을 생략하고 PoC를 통해 거래를 증명하기 때문에 안전거래의 주요 문제점이었던 거래 속도를 높일 수 있다.

PoC란 Proof of Collective Intelligence의 준말로 집단 지성의 증명이라는 뜻이다. 즉, 중고 거래에 집단 지성을 이용해 해당 거래가 안전한 지 여부를 판단하는 과정을 도입하여 기존의 1대 1 거래가 아닌 1대 N 중고 거래를 선보인다. 해당 과정에 참여한 인원은 모두 가상화폐를 보상으로 받게 된다.

해당 Blockchain은 Oracle 서버와 연동되어 사용자가 쉽게 사용할 수 있도록 Hybrid 형태로 구성된다. 사용자는 Blockchain을 직접적으로 이용하지 않고 Used2Block 서비스를 이용해 기존의 웹 브라우저를 통해 Blockchain에 접근할 수 있다. 참고로 사용자는 구매자, 판매자와 검증자를 통틀어 일컫는다.

## 2.3 Expected Effects

### 2.3.1. 품질 보증

- 거래 검증 과정을 통해 판매자의 신뢰도를 증명함으로써 물건의 품질도 같이 증명 가능하다.

### 2.3.2. 신뢰성 강화

- Blockchain 기술 기반의 판매자 거래 내역 기록을 통해 판매자의 신뢰성을 가늠할 수 있다.
- 거래 검증이라는 프로세스를 통해 한번 더 판매자의 신뢰성을 검증할 수 있다.

### 2.3.3. 사용자 중심

- 판매자 신뢰도에 대한 검증이 이루어지는 과정에서, 검증에 참가한 사용자들에게 인센티브가 주어진다.
- 해당 검증에 대한 수수료 비용은 서비스 제공자가 아닌 검증자들에게 돌아간다.

### 2.3.3. 안정성 강화

- 로그를 남기는 가상 화폐 활용하여 거래에 대한 지속적인 기록이 가능하다.
- 거래 검증에 참여하는 인원이 많아질 수록 시장이 더 안정적이다.

### 2.3.4. 빠른 거래 속도

- 기존의 안전 거래가 아닌 Used2Block 서비스의 Consensus Algorithm을 통해 거래를 빠르게 처리한다.

- 이로써 사용자의 편의성 증대한다.



### 3. Glossary

용어	설명
Blockchain	온라인 거래 명세가 기록된 블록을 연결한 것으로, 누구나 접근 가능한 공공 거래 장부이다.
Smart contract	스마트 계약, 소프트웨어 프로토콜로 미리 프로그래밍을 한 조건을 등록하고, 이를 만족시키면 계약이 자동으로 이루어지도록 하는 기술이다.
(Private) Key	Transaction에 대한 내용을 Sign하기 위한 암호 키이다. 해당 Key를 통해 Transaction에 대한 소유권을 주장할 수 있다.
Hashing	특정 데이터에 대한 값을 함수를 통해 변형시켜 고정된 사이즈의 값으로 만드는 과정이다
Hash값	hashing 과정을 통해 변환된 값이다.
Solidity	이더리움 Blockchain 플랫폼의 smart contract 작성 및 구현에 사용되는 프로그래밍 언어이다.
Web3.js	모든 자료와 정보가 분산화, 분권화된 네트워크 구조. 서버가 없는 인터넷 분산형 웹. 후술할 javascript를 기반으로 한다.
Javascript	웹 브라우저 내에서 주로 사용되는 언어이다.
Node.js	네트워크 어플리케이션 개발에 사용되는 소프트웨어 플랫폼이다.
사용자(user)	Used2Block을 이용하는 모든 사람을 일컫는다.
구매자(buyer)	사용자 중에서 물건을 구입하기를 희망하는 사람을 일컫는다.
판매자(seller)	사용자 중 물건을 판매하기를 희망하는 사람을 일컫는다.
검증인(Validator, Validator)	구매자와 판매자 사이의 거래에 개입하여, 거래에 대한 신뢰도를 검증하는 역할을 가지는 사용자. 구매자도 판매자도 아닌 제 3자가 검증인이 된다.
쿼리	데이터베이스와 정보 시스템의 질의를 하여 정보 수집을 하기 위해 쓰이는 프로그래밍 언어이다.
SQL	데이터베이스 관리를 위해 만들어진 프로그래밍 언어이다.
SHA-256	Hashing 알고리즘의 하나로 입력값을 32바이트로 변형시키는 방법이다.
PoS : Proof of Stake	Blockchain에서 쓰이는 합의 알고리즘의 하나로 온라인 거래를 증명하기 위해 신뢰성을 지닌 Stakeholder를 선정하는 방법이다.
Tit-for-tat	게임 이론에서 '반복되는 죄수의 딜레마'의 전략 중 하나. A와 B라는 경기자가 있고 A가 이 전략을 따른다면, A는 처음에는 B에게 협력하고 그 이후에는 B의 바로 전 선택에 반응한다.
DApp : Decentralized App	기존의 중앙화된 서버를 이용하는 방식이 아닌 분산형 정보 저장소를 이용하는 분산형 Application이다.

Table 3-1. Glossary

## 4. User Requirements Definition

해당 챕터에서는 시스템이 핵심으로 제공하는 기능적 요구사항인 서비스와 부가적으로 충족시켜야 하는 비기능적 요구사항 두 가지로 나누어 설명한다. 해당 챕터에서는 사용자를 대상으로 작성되는 부분으로 구체적인 기술은 지양하며 자연어와 시각 자료를 이용해 설명한다.

### 4.1. Functional Requirements

#### 4.1.1. Sign Up

사용자가 서비스를 이용하기 위해 서버에 회원으로 등록하는 기능이다. 회원 가입시에는 사용자가 이메일, 전화번호, 주소, 아이디, 비밀번호를 입력해 진행한다.

#### 4.1.2. Login/ Logout



Figure 4-1. Used2Block의 로그인 화면

회원으로 가입된 사용자가 시스템에 인증 받는 절차이다. 인증 과정 이후 해당 사용자는 Used2Block 서비스를 이용할 수 있다.

#### 4.1.3. User's Profile Management

사용자 자신의 프로필 정보를 확인하고 수정할 수 있는 기능을 제공한다. 해당 페이지에서는 현재 가상 화폐의 잔고, 거래 내역, 개인 정보가 포함된다. 만약 자신의 계정이 존재하지 않는다면 Sign In 페이지로 이동해 가입을 진행하게 된다.

#### 4.1.4. Creating Wallet

해당 서비스를 이용하기 위해서는 가상 화폐의 소유권을 나타내는 Address가

필요하다. 이를 위해 해당 사용자에게 지갑 기능을 담당하는 Address와 해당 지갑을 안전하게 보호할 수 있는 Key를 제공한다. 사용자의 편의성을 위해 Key값을 사용자가 입력하는 단어들의 조합으로 구성하여 제공할 수 있도록 한다.

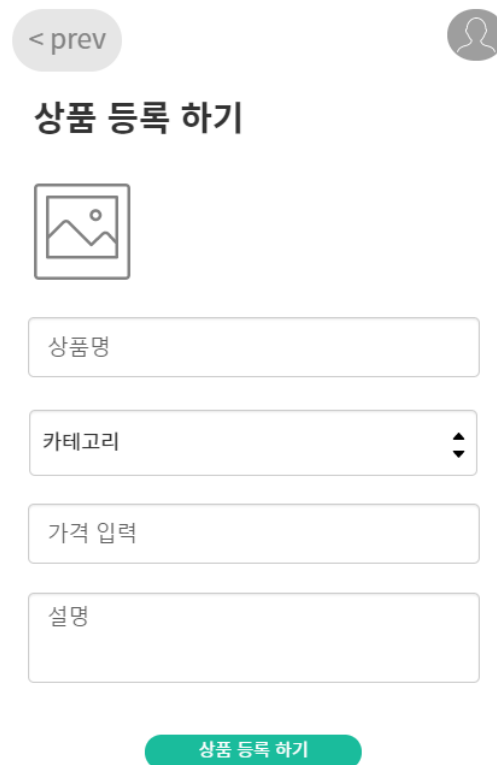
#### 4.1.5. Searching



Figure 4-2. Used2Block의 검색 화면


사용자가 구매하고 싶은 물건 혹은 검증하고 싶은 물건을 검색할 수 있는 기능이다. 사용자는 원하는 검색어를 입력함으로써 원하는 물건에 대한 리스트를 얻을 수 있다.

#### 4.1.6. Selling Product



< prev

상품 등록 하기



상품명

카테고리

가격 입력

설명

상품 등록 하기

Figure 4-3. Used2Block의 판매자 상품 등록 화면

판매자가 팔고 싶은 물건을 등록할 수 있는 기능이다. 판매하고자 하는 상품을 올릴 때, 판매자는 상품명, 카테고리, 가격을 입력하고 설명을 같이 기재한다. 이후 “상품 등록 하기” 버튼을 클릭하면 해당 상품은 Blockchain 거래 시스템에 등록된다.

#### 4.1.7. Buying Product

판매자가 올린 제품을 구매하는 기능이다. 구매하고자 하는 상품을 선택한 뒤, 수령인, 연락처, 배송지를 입력한다. 이후 Blockchain 검증 거래 서비스를 이용하고 싶다면 해당 서비스에 대한 체크박스를 클릭한 뒤 검증 기간과 판매자에게 지급할 수수료를 입력한다. 이후 판매자가 요청을 수락하면 거래 검증이 시작된다. 이후 거래 검증이 문제 없이 종료되면 결제가 이루어진다.

#### 4.1.8. Manage product

자신이 구매하거나 판매 또는 검증한 물건을 관리하는 기능이다. 이전에 판매, 구매 또는 검증을 했던 물품의 리스트가 나타나며 해당 물품을 클릭하면 물품에 대한 정보와 자세한 거래 정보가 기재된다.

#### 4.1.9. Verifying Transaction

[< prev](#)



### 주문 검증

상품 상태	판매 중
상품 번호	3233535
등록 일시	2019-10-05



#### 쿠쿠 밥솥

User description Here

25,000원   판매자 : 백산수



판매자	백산수
계좌	신한 110-223-2424242
전화 번호	010-0000-000

No.	내용	출처
1	더치트에 검색됨	
2		
3		

신고 하기

Figure 4-3. Used2Block의 주문 검증 화면

Blockchain 위에 등록된 거래에 대해 검증하는 서비스이다. 해당 서비스는 판매자, 구매자가 이용하는 것이 아닌 제 3자가 참여하게 된다. 제 3자는 거래에 대한 정보, 즉 판매자가 올린 사진, 판매자의 계좌 번호, 전화 번호 등 다양한 정보를 통해 해당 판매자의 신변을 확보하게 된다. 이를 통해 제 3자는 판매자에 대한 타 플랫폼에서의 거래내역과 사기 내역을 조회하고 해당 거래를 검증할 수 있다. 거래를 검증하는 과정에서 문제를 발견해내면 인센티브가 지급되며 해당 거래는 파기된다.

## 4.2 Non-functional Requirements

### 4.2.1 Product Requirement

#### 4.2.1.1. Security

해당 시스템은 사용자의 가상화폐를 관리하고 있으므로 특히 보안에 신경써야 한다. 가상화폐에 대한 Private Key는 서버에 안전하게 저장되어야 하여야 한다. 또한 검증 과정 이후 해당 판매자에 대한 정보를 Blockchain에 저장할 시 정보가 유출되지 않도록 판매자의 계좌와 전화번호를 모두 Hash값으로 저장한다.

#### 4.2.1.2. Reliability

해당 시스템은 Blockchain 위에서 작동하는 Smart Contract와 서버가 연동되는 시스템이기 때문에 두 시스템이 상호 작용함에 있어 안정성을 갖춰야 한다. Blockchain 위에서 작동하는 Smart Contract가 수수료 문제로 정지될 수 있기 때문에 해당 사항을 고려해 수수료를 지속적으로 측정해야한다.

#### 4.2.1.3. Speed

Blockchain을 이용하는 시스템이기 때문에 거래에 대한 Speed는 Blockchain의 거래 속도와 관련이 크다. 때문에 Blockchain 상에서 거래에 대한 수수료를 한도 내에서 최대한 높게 설정하여 거래가 빨리 이루어 질 수 있도록 해야한다.

또한 검증과정에 있어서의 Speed도 중요하다. 해당 과정이 길어지면 기존 안전거래 서비스가 가진 문제점을 해결할 수 없기 때문에 검증 과정 시간을 최대한 줄일 수 있는 검증 알고리즘을 사용해야 한다.



#### 4.2.1.4. Usability

Blockchain 에 익숙하지 않은 사용자를 위해서 최대한 사용하기 쉽도록 만들어져야 한다. 거래 검증 과정이라든지, 가상 화폐 관리 방법 등 기존의 중고 시장 앱과는 많은 기능적 차이가 존재하기 때문에 해당 기능들에 대해 익숙하지 않은 사용자가 쉽게 사용할 수 있도록 해야한다.

### 4.2.2 Organization requirement

#### 4.2.2.1. Delivery requirement

해당시스템은 Generic Product 로 소비자에게 최대한 빨리 전달되어야 한다. Waterfall Model 을 사용해 프로젝트를 설계하지만 Delivery 는 최대한 이른 시간 안에 진행되어야 Time-to-Market 을 만족할 수 있다.

#### 4.2.2.2. Implementation requirement

먼저 Blockchain 상에서 동작하기 위해서는 이더리움과 연동되는 Solidity 를 사용해야한다. 해당 언어를 통해 Smart Contract 를 만들고 그와 연동되는 Web3.js API 를 지원하는 Javascript 와 Node.js 를 사용해 웹과 서버를 구축해야 한다.

### 4.2.3 External requirements

#### 4.2.3.1. Interoperability

해당 시스템은 Blockchain 위에 저장된 정보를 사용하기 때문에 서버와의 상호 호환성이 매우 중요하다. 서버에서 사용하는 Oracle과 Blockchain이 연동되어 정보를 주고 받고 웹페이지에 Blockchain 위의 정보를 운용할 수 있어야 한다.

또한 화폐 간 상호 운용성도 중요하다. 해당 시스템의 가상화폐가 단순히 고립된 화폐가 아니라 비트코인, 이더리움, 리플 등 다양한 화폐와 교환될 수 있도록 해야한다.

#### 4.2.3.2. Ethical requirement

해당 시스템을 사용함에 있어 사용자는 거래에 기록되는 개인정보를 유출해서는 안된다. 해당 정보는 개인정보보호법 제 3 조 2 항<sup>1)</sup>에 따라 보호되기 때문에 이를 유출하는 행위는 금지되어야 한다. 또한 사용자 간의 거래에 있어 악의적으로 잘못된 정보를 입력해서는 안된다. 이 또한 사기죄에 해당하며 사용자의 혼란을 막기 위해 금지되어야 한다.

#### 4.2.3.3. Legislative requirement

해당 시스템은 사용자가 거래에 입력하는 정보를 거래 검증인에게 공개한다. 이 과정에서 당사자의 동의가 없다면 제 3자에게 제공하는 것은 금지되기 때문에 일부 정보를 공개하기 이전에 사용자의 동의가 필수적이다. 이를 통해 개인정보의 유출을 방지하여 개인의 사생활을 보호해야한다.

해당 시스템은 가상화폐를 제작하고 유통하기 때문에 금융감독원의 허가가 필요하다. 해당 가상화폐는 ICO를 통해 모금을 진행하게 되는데 허가없는 자금 모금 사업은 법률 위반으로 법적인 처벌을 피할 수 없다. 때문에 가상화폐에 대한 금융감독원의 허가를 받아 ICO를 진행해야 한다.

---

<sup>1</sup> 개인정보보호법 제3조 2항은 2019년 1월 1일 시행본으로 아래와 같은 내용을 포함하고 있다.

*네트워크 카메라: 일정한 공간에 지속적으로 설치된 기기로 촬영한 영상정보를 그 기기를 설치·관리하는 자가 유무선 인터넷을 통하여 어느 곳에서나 수집·저장 등의 처리를 할 수 있도록 하는 장치*

## 5. System Architecture

이 챕터에서는 시스템의 전체적인 구조를 기술한다. Requirement Engineering 과정에서 도출한 시스템의 전체적인 구조와 각 서브시스템의 구성, 서브시스템 간의 관계를 설명하며, 각 구조는 도표를 첨부하여 이해를 돕는다.

### 5.1 Overall Architecture

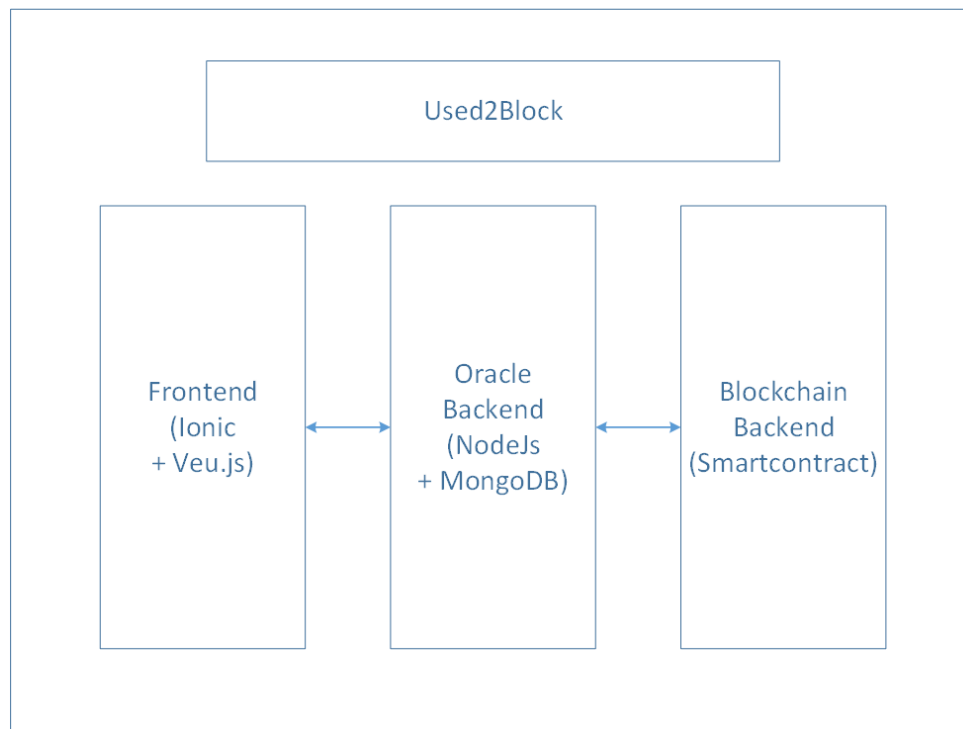


Figure 5-1. Used2Block의 Overall architecture

기본적으로 Used2Block 서비스는 크게 세가지 Component로 구성된다. User에게 서비스를 진행할 Frontend, Blockchain Backend의 보조 역할을 담당할 Oracle Backend, 그리고 모든 거래가 발생하는 Blockchain Backend이다.

## 5.2. Frontend Architecture

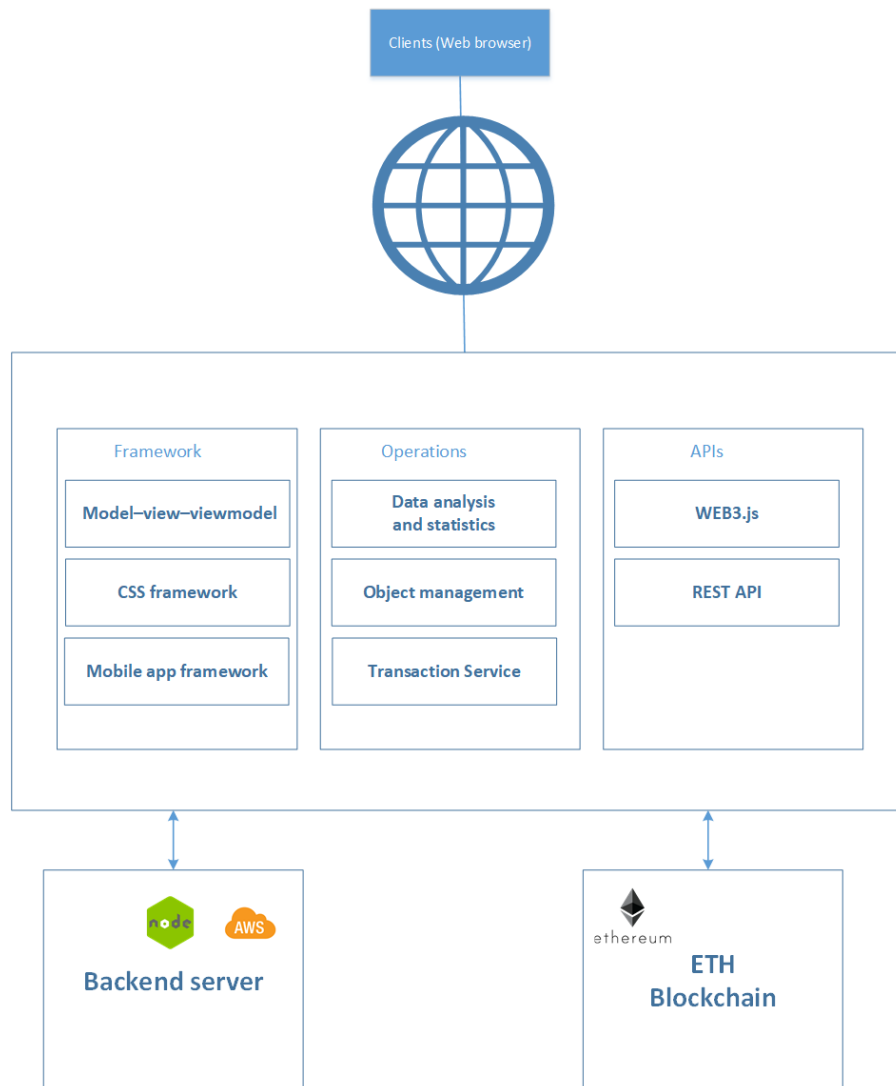


Figure 5-2. Used2Block의 Frontend Architecture

Frontend Architecture는 사용자가 시스템과 Interaction하고 데이터를 요청하는 데 필요한 기능을 제공한다. Used2Block에서의 주된 Operation은 두 가지 종류의 서버와 Interaction하게 된다. 사용자, 상품 관리, 상품 거래라는 Object Management 부분과 사용자의 거래 기록을 분석하는 Data Analysis 부분은 Backend Server와 Interaction하게 되고, Transaction Service는 ETH Blockchain과 Interaction할 수 있도록 설계되었다. 또한 API를 통해 각 서버로부터 얻은 정보는 Framework를 이용해 사용자에게 전달되도록 한다.

### 5.3. Backend Architecture

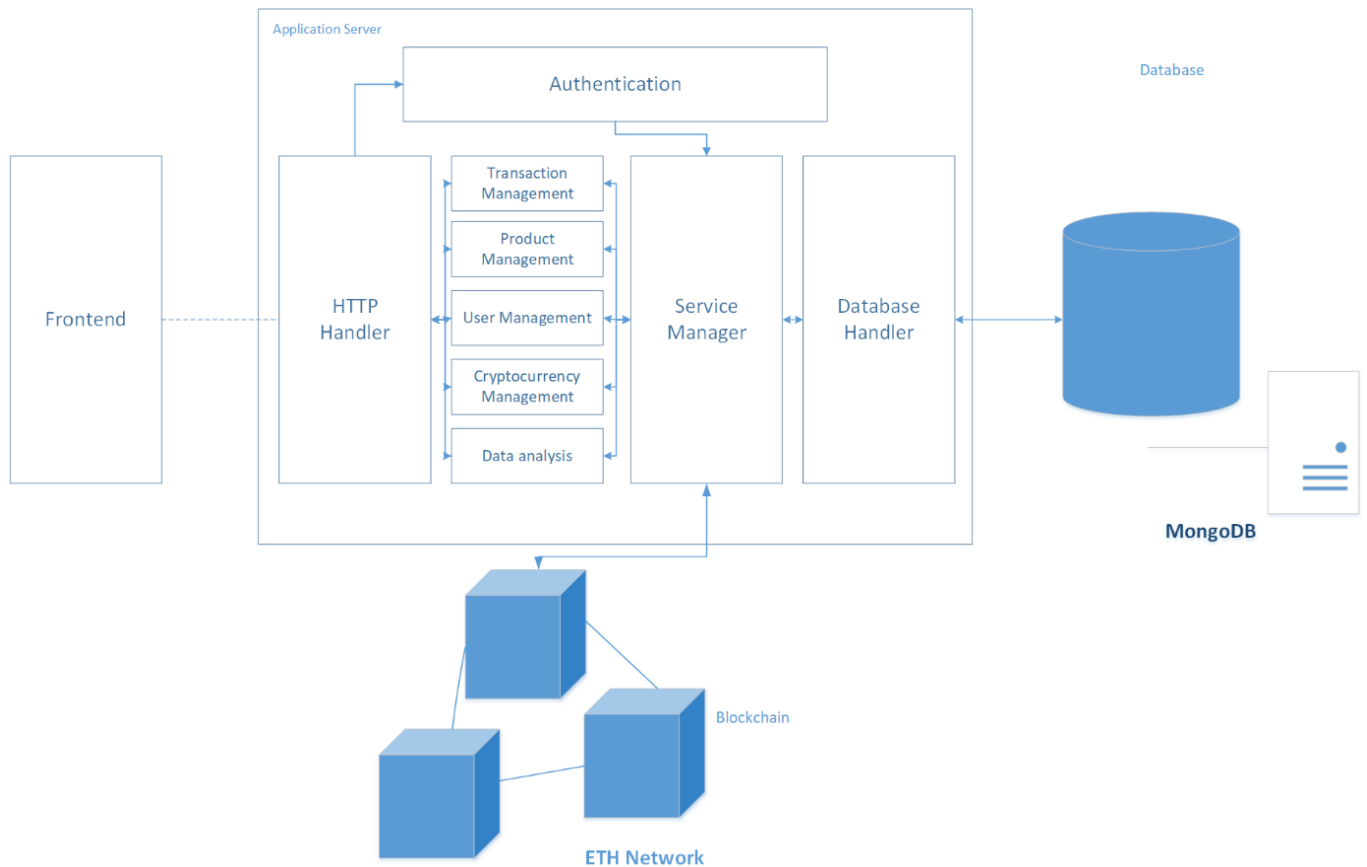


Figure 5-3. Used2Block의 Backend Architecture

Backend의 구조는 Figure 5-3와 같이 구성되어 있다. Backend는 HTTP 요청을 받아 Handling하는 기능을 담당하도록 설계되었다. 기본적으로 Backend는 ETH Network로부터 정보를 갖고 오거나 입력하는 역할을 담당하며, 부가적인 정보를 저장하기 위해 Database를 사용한다.

## 5.4. ETH Blockchain System

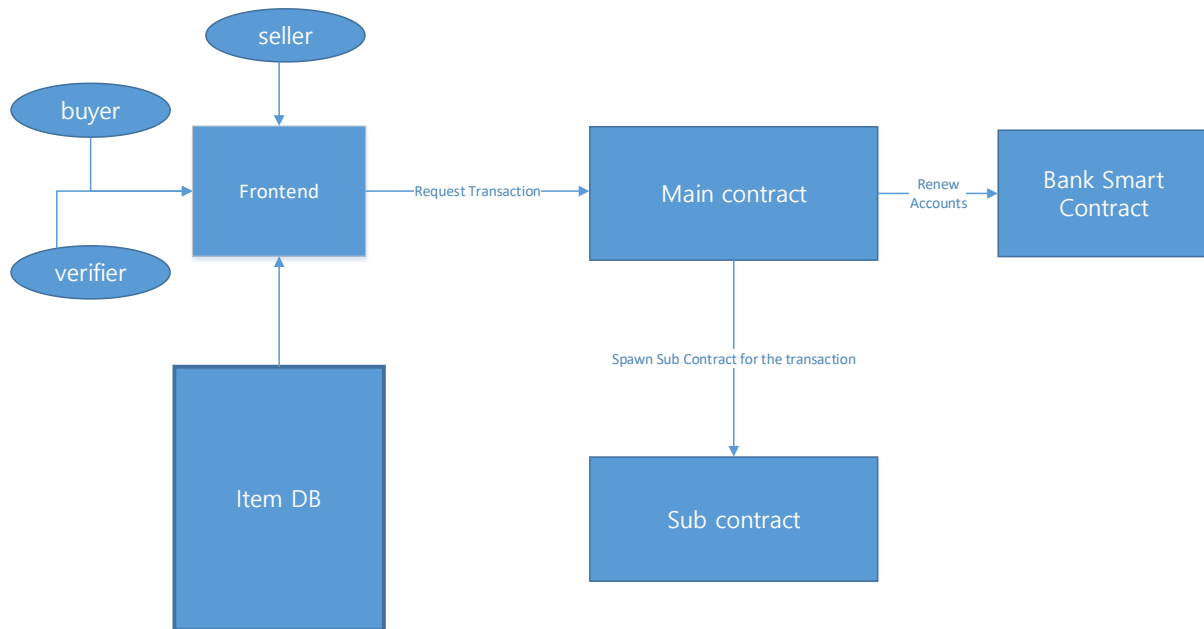


Figure 5-4. Used2Block의 거래 과정 모식도

Blockchain 시스템은 기본적으로 Smart Contract 로 구성된다. Main Smart Contract 는 구매자 혹은 판매자로부터 Transaction Request 를 받아 Request 에 대한 정보를 담은 Sub Smart Contract 를 생성한다. 해당 Sub Smart Contract 에서 Validator 가 Seller 의 과거 이력을 확인하는 검증 작업이 일어난다. 검증 작업 이후 신뢰 거래가 이루어졌을 경우 Bank Smart Contract 상의 구매자의 가상화폐가 판매자에게 전송된다. 모든 작업이 종료되면 Sub Smart Contract 는 소멸한다.

## 5.5. Application System

### 5.5.1 Transaction Management

어플리케이션 시스템의 거래 관리는 Blockchain 상에서 사용자가 이용했던 거래 내역 데이터를 가져오는 역할 또는 거래를 진행하는 경우 Blockchain 위에 기록하는 역할을 한다.

### 5.5.2 User Management

User Management 는 데이터베이스 상의 사용자에 대한 정보를 관리하는 기능이다. 사용자가 Used2Block 서비스를 사용함에 있어 필요한 계정을 생성하거나 로그인을 시도하는 경우 데이터베이스로부터 정보를 Write 하거나 Read 한다. 또한 거래를 진행하면서 거래 상대들로부터 받은 평판 등을 데이터베이스로부터 가져올 수도 있다.

### 5.5.3 Product Management

Product Management 는 판매자가 상품 판매를 원할 때 상품의 사진, 상품명, 카테고리, 희망 가격, 설명을 입력하였을 때 Blockchain 위에 해당 상품을 올리거나, 이미 등록된 상품을 Blockchain 시스템으로부터 가져오게 된다.

### 5.5.4 Cryptocurrency Management

Cryptocurrency Management 는 사용자가 가진 가상화폐를 관리하는 역할을 한다. 해당 사용자의 가상 화폐에 대한 정보는 Blockchain 으로부터 가져오며, 환전을 요청했을 경우 Blockchain 에 사용자의 가상 화폐에 대한 갱신을 요청한다.

## 6. System Requirements Definition

해당 챕터에서는 시스템이 핵심으로 제공하는 기능적 요구사항인 서비스와 부가적으로 충족시켜야 하는 비기능적 요구사항 두가지로 나누어 설명한다. 개발자를 대상으로 작성되는 부분인만큼 구체적인 기술과 구현에 관한 자연어, 시각 자료, 도표 등을 이용해 상세히 설명한다. 각 System Requirement들은 보다 상세한 내용 서술과 어떤 Input을 받고 어떤 동작을 하며, 어떤 Output을 산출하는지 등에 관해 기술된다. 또한 해당 요구사항의 조건과 제약을 함께 표현한다.

### 6.1. Functional Requirements

#### 6.1.1. Sign Up (Register)

Name(이름)	Sign up
Description(설명)	처음으로 Used2Block 서비스 이용하는 사용자는 회원으로서 서버에 등록된다. 이후 사용자는 고유 식별자를 부여받아 서비스를 이용할 수 있다.
Inputs(입력)	이메일, 전화번호, 주소, 아이디, 비밀번호
Source of Input(근원지)	사용자(User)
Outputs(출력)	사용자 계정(account)
Destination for Output(도착지)	Used2Block 데이터베이스
Action(행동, 처리)	주어진 입력 값을 통합해서 고유한 사용자의 데이터를 데이터베이스에 저장한다.
Requirements(요구사항)	이메일과 전화번호는 정해진 형식에 맞춰서 입력되어야 한다. (ID@mail_domain), (+82-10-0000-0000) 비밀번호는 보안상 쉽게 유출되지 않



	는 조합의 값이어야 한다. (영문, 숫자, 특수기호)
Pre-condition(전치조건)	먼저 입력된 정보와의 중복이 없어야 한다.
Post-condition(후치조건)	사용자의 입력 값은 손실 없이 저장되어야 한다.

Table 6-1. Sign up Function.

### 6.1.2. Login

Name(이름)	Login
Description(설명)	Sign Up 과정을 거쳐 회원으로 가입된 사용자가 시스템에 인증 받는 절차이다. 인증 과정 이후 해당 사용자는 Used2Block 서비스를 이용할 수 있다.
Inputs(입력)	아이디, 비밀번호
Source of Input(근원지)	사용자(User)
Outputs(출력)	Used2Block 웹 페이지 (메인)
Destination for Output(도착지)	User Terminal (웹 브라우저)
Action(행동, 처리)	데이터베이스 내의 데이터와 입력 값을 비교한 뒤, 인증과정을 수행하여 해당 사용자를 식별한다.
Requirements(요구사항)	경우에 따라 추가 인증기능이 필요할 수 있다.
Pre-condition(전치조건)	입력된 사용자의 데이터가 기존의 데이터베이스에 존재해야 한다.
Post-condition(후치조건)	인증이후 Logout 기능의 수행 전까지 서비스는 제공된다.

Table 6-2. Login Function

### 6.1.3. Logout

Name(이름)	Logout
Description(설명)	시스템에 인증을 받고 서비스를 이용하던 사용자가 사용을 모두 마친 후

	안전하게 Used2Block 서비스를 종료하는 행위를 의미한다. 이후 계정은 다음의 Login 기능의 동작까지 안전하게 보호된다.
Inputs(입력)	Logout button의 signal
Source of Input(근원지)	사용자(User)
Outputs(출력)	Logout message
Destination for Output(도착지)	User Terminal (웹 브라우저)
Action(행동, 처리)	서비스 종료 명령을 받은 즉시 인증된 사용자의 인증을 종료하고 기존의 비인증 상태로 복귀한다.
Requirements(요구사항)	사용자가 Login 상태여야 한다.
Pre-condition(전치조건)	사용자가 Used2Block 시스템에 로그인한 상태여야 한다.
Post-condition(후치조건)	이후 재 인증과정 전까지 해당 사용자 개인의 어떠한 서비스도 수행되지 않는다.

Table 6-3. Logout Function

#### 6.1.4. User's Profile Management

Name(이름)	User Profile
Description(설명)	사용자 자신의 프로필 정보를 확인하고 수정할 수 있는 기능을 제공한다. 해당 페이지에서는 현재 가상 화폐의 잔고, 거래 내역, 개인 정보가 포함된다.
Inputs(입력)	서비스 요청,
Source of Input(근원지)	사용자, Used2Block 데이터베이스
Outputs(출력)	사용자 정보, 가상 화폐 정보, 거래 내역,
Destination for Output(도착지)	Used2Block 웹 페이지
Action(행동, 처리)	서비스 요청을 받아 인증 받은 해당

	사용자의 정보(Output)를 데이터베이스로부터 받아와 제공한다.
Requirements(요구사항)	정보요구는 인증을 받은 상태여야 하며, 정보의 제공은 즉시 이루어져야 한다.
Pre-condition(전치조건)	가입하고 인증 받은 사용자의 상태여야 한다. 사용자의 정보가 데이터베이스에 손실 없이 저장되어 있어야 한다.
Post-condition(후치조건)	변경사항이 있을 시 모두 즉시 반영한다.

Table 6-4. User Management Function

#### 6.1.5. Creating Wallet

Name(이름)	Creating Wallet
Description(설명)	가상 화폐는 소유권을 나타내는 Address가 필요하다. 이를 위해 해당 사용자에게 지갑 기능을 담당하는 Address와 해당 지갑을 안전하게 보호할 수 있는 Key를 제공해야 한다. 여기서 Key값은 사용자가 입력하는 단어들의 조합으로 구성하여 제공하며 Address는 식별가능한 조합의 일련번호로 시스템이 제공한다.
Inputs(입력)	Sign Up 신호, Key 입력,
Source of Input(근원지)	사용자(User), Used2Block 데이터베이스
Outputs(출력)	완료문구, Address, Key value
Destination for Output(도착지)	User Terminal, Used2Block 데이터베이스
Action(행동, 처리)	Sign up과 동시에, 식별가능한 Address를 제공하고, Key 값을 추가로 입력 받는 프로세스를 통해서 해당 사용자의 Wallet을 만든다.

Requirements(요구사항)	Address 값은 Unique해야한다.
Pre-condition(전치조건)	Sign Up과정에서 오류가 없어야 한다.
Post-condition(후치조건)	Address와 Key는 함께 사용자 데이터와 함께 저장되고 관리되어야 한다.

Table 6-5. Creating Wallet Function

#### 6.1.6. Searching

Name(이름)	Searching
Description(설명)	사용자가 구매하고 싶은 물건 혹은 검증하고 싶은 물건을 검색할 수 있는 기능이다. 사용자는 원하는 검색어를 입력함으로써 원하는 물건에 대한 리스트를 얻을 수 있다.
Inputs(입력)	검색하고 싶은 값
Source of Input(근원지)	사용자(User)
Outputs(출력)	검색된 품목들의 List, Message.
Destination for Output(도착지)	User Terminal (웹 브라우저, Used2Block page)
Action(행동, 처리)	사용자가 찾기를 원하는 값을 입력 받아 데이터베이스 상에서 저장된 품목의 정보와 비교한다. 일치하는 항목들을 사용자에게 다시 출력한다. 일치하는 모든 품목은 일치하는 여부와 등록순서에 따라서 정렬되어 출력한다. 만일 일치하는 목록이 없을 시, '목록이 없음' 또한 출력한다.
Requirements(요구사항)	쿼리의 처리는 SQL언어로 작동한다.
Pre-condition(전치조건)	-
Post-condition(후치조건)	-

Table 6-6. Search Function

#### 6.1.7. Selling Item

Name(이름)	Selling Item (상품 등록, 판매요청)
----------	----------------------------

Description(설명)	시스템 상에서 사용자가 팔고 싶은 물건을 올릴 수 있는 기능이다. 물건은 Blockchain 거래 시스템에 등록되어 다음 작업을 위해 대기한다.
Inputs(입력)	상품명, 카테고리, 가격, 설명, 사진에 대한 등록하기 버튼의 입력
Source of Input(근원지)	사용자 중에서 판매자
Outputs(출력)	등록된 상품 정보, 상품 데이터
Destination for Output(도착지)	User Terminal, Used2Block 데이터베이스
Action(행동, 처리)	입력된 데이터를 안전하게 저장하고, 저장된 결과를 Interface에 맞게 출력한다. 저장된 데이터는 즉시 검색에 적용되고, 노출되며, 등록된 페이지는 등록자에게도 노출되어 검증 과정을 거친다.
Requirements(요구사항)	상품의 등록자가 명시된 최소한의 정보들은 모두 입력하게 제약하고, 추가로 정보를 입력하는 것 또한 수용한다.
Pre-condition(전치조건)	-
Post-condition(후치조건)	해당상품이 중복되어 등록되지는 않았는지 점검해야 한다.

Table 6-7. Item Registration Function

#### 6.1.8. Buying Product

Name(이름)	Buying Item (구매 주문)
Description(설명)	판매자가 올린 제품을 구매하는 기능이다.
Inputs(입력)	상품, 수령인, 연락처, 배송지
Source of Input(근원지)	사용자(User - 구매자)
Outputs(출력)	구매 확정 페이지, 거래내역
Destination for Output(도착지)	User Terminal, Used2Block 데이터베이스

	스
Action(행동, 처리)	구매하고자 하는 상품과 구매자의 배송 정보를 입력하고 나면 구매 상품의 거래를 확정한다.
Requirements(요구사항)	-
Pre-condition(전치조건)	-
Post-condition(후치조건)	-

Table 6-8. Buying Item Function

#### 6.1.9. Validation

Name(이름)	Buying Item with Validation
Description(설명)	상품을 구매할 때, Blockchain 검증 서비스를 추가로 선택하는 경우에 제공되는 서비스다. 보통의 거래와 달리, 구매자가 동의하고, 판매자에게 지급할 수수료를 입력한 후, 판매자가 요청을 수락해야 한다. 그 후 거래 검증이 문제없이 종료되고 나서 결제가 이루어진다.
Inputs(입력)	검증 서비스 체크박스 시그널, 수수료, 요청 수락,
Source of Input(근원지)	구매자, 판매자
Outputs(출력)	거래 검증 페이지
Destination for Output(도착지)	User Terminal, Used2Block 데이터베이스
Action(행동, 처리)	검증서비스 요청과 데이터를 받고, 판매자의 동의를 얻은 후 해당 거래를 검증 대기 상태로 변경한다.
Requirements(요구사항)	-
Pre-condition(전치조건)	검증 서비스 이외의 정보 모두 이미 확인되고 성립된 상태이다.
Post-condition(후치조건)	Verifying Transaction의 기능으로 전달

	한다.
--	-----

Table 6-9. Validation Function

#### 6.1.10. Manage Item

Name(이름)	Manage Item
Description(설명)	사용자와 관계된 상품들을 관리하는 기능이다. 이전의 판매, 구매, 검증 기록이 제공된다.
Inputs(입력)	Manage Page Request
Source of Input(근원지)	사용자(User), Used2Block 데이터베이스
Outputs(출력)	Item history list
Destination for Output(도착지)	User Terminal
Action(행동, 처리)	이용자와 관계된 상품들을 모두 저장된 데이터베이스로부터 검색해서 불러온다.
Requirements(요구사항)	SQL쿼리를 통한 검색의 경우 결과의 검색이 느리지 않도록, 사용자에게 interactive하도록 한다.
Pre-condition(전치조건)	사용자와 관련된 상품들에 관한 기록이 데이터베이스에 저장되어 있어야 한다.
Post-condition(후치조건)	해당 상품들에 대한 추가 요청이 있을 경우 Product Page로 연결되어야 한다.

Table 6-10. Item Management Function

#### 6.1.11. Item Page

Name(이름)	Item Page
Description(설명)	상품의 정보를 제공하는 페이지이다. 판매자가 올린 상품이 하나의 개체로서 저장되고 관리된다. 이 개별 Item의 페이지들은, 사용자가 구매 또는 검증을 위한 검색을 할 시에 접근된다.

Inputs(입력)	Item page request
Source of Input(근원지)	사용자 (User)
Outputs(출력)	Item Information
Destination for Output(도착지)	User Terminal
Action(행동, 처리)	이용자가 요청한 상품의 정보를 저장된 데이터베이스로부터 검색해서 불러온다.
Requirements(요구사항)	-
Pre-condition(전치조건)	사용자와 관련된 상품들에 관한 기록이 데이터베이스에 저장되어 있어야 한다.
Post-condition(후치조건)	-

Table 6-11. Item Page

#### 6.1.12. Verifying Transaction

Name(이름)	Verifying Transaction
Description(설명)	Blockchain 검증 거래 서비스를 요청한 경우에 제3자가 참여하여 거래를 검증한다. 제3자는 거래에 대한 정보 중 판매자가 등록한 사진, 계좌번호, 전화번호 등을 이용하여 해당 판매자의 거래 내역과 사기 내역을 조회하고 거래를 검증한다. 타 플랫폼에서의 기록 또는 Blockchain에서의 거래내용을 통해서 찾아내면 인센티브가 지급되고 거래는 안전하게 파기된다.
Inputs(입력)	판매자 사기 정보.
Source of Input(근원지)	사용자(User), Blockchain,
Outputs(출력)	거래 파기 페이지.
Destination for Output(도착지)	사용자 (User)
Action(행동, 처리)	검증자가 판매자의 사기 내역 정보를 입력하면 확인을 통해서 판매자와 구



	매자 사이의 거래를 즉시 파기한다. 검증은 Blockchain에 기록된 거래 내역과 웹 페이지와 연결되어 있는 기존의 다른 서비스에서 검증된 결과를 통해서 이루어진다. 제3자는 이를 통해서 인센티브를 지급받을 수 있어야 한다.
Requirements(요구사항)	사기 내역 정보는 그 근거가 충분해야 한다. 타 사이트(ex: 더 치트, 경찰청 데이터베이스 등) 또는 Blockchain에서 찾아낸 사기내역을 의미한다.
Pre-condition(전치조건)	Blockchain 검증 거래 서비스를 요청해야 한다.
Post-condition(후치조건)	이후 검증에 이용된 판매자의 개인 데이터는 더 이상 노출되지 않는다.

Table 6-12. Transaction Verifying Function

## 6.2 Non-functional Requirements

### 6.2.1. Product Requirement

#### 6.2.1.1 Security

사용자 요구사항에 언급했던 것처럼, 가상화폐를 관리하고 있으므로 특히 보안에 신경 써야 한다. 가상화폐의 Private Key 는 서버에 안전하게 저장되어야 한다. 또한 검증 과정 이후 해당 판매자에 대한 정보를 Blockchain 에 저장할 때에 정보가 유출되지 않도록 판매자의 계좌와 전화번호를 역시 Hash 값으로 저장한다.

여기서 암호 해시는 데이터를 함수를 이용해 고정된 사이즈의 고유 값으로 변경하는 기술이다. 즉, 내용보다 해당 개체에 대한 식별 정보로 요약하는 것이다. SHA-256 을 사용해 총 256 바이트가량의 고유 값이 만들어야 한다. 해시 알고리즘을 통해서 Raw data 에 대한 유출을 막을 수 있다.

#### 6.2.1.2. Reliability

해당 시스템은 Blockchain 위에서 작동하는 Smart Contract 와 서버가 연동되는 시스템이기 때문에 두 시스템이 상호 작용함에 있어 안정성을 갖춰야 한다. Used2Block 플랫폼 정보를 저장하고 있는 메인 데이터베이스는 특히 Validation 과정에서, Product 의 정보를 즉각적으로 Block Chain 검증 과정에 제공해 주어야 하고, Smart Contract 결과는 검증과 동시에 거래내역 등의 정보를 서버에 제공하고 다시 기록해서 실시간으로 동기화할 수 있어야 한다.

Blockchain 위에서 작동하는 Smart Contract 가 거래를 진행하는 데 있어 부족한 수수료 문제로 Operation 이 정지될 수 있기 때문에 해당 사항을 고려해 Blockchain 상 Transaction 에대한 수수료를 지속적으로 측정해야 한다.

#### 6.2.1.3. Speed

여기서의 Speed 는 시스템 Performance 를 의미하는 것과는 다르다. 이는 시스템의 활발한 사용을 측정하기 위해 쓰이는 척도이며, Blockchain 의 거래 속도와 관련이 크다. 이를 위해서는 먼저 이더리움 Blockchain 상에서 거래에 대한 수수료를 최대한 높게 설정하여 거래가 빨리 이루어 질 수 있도록 해야 한다.

또한 검증과정에 있어서의 Speed 도 중요하다. 해당 과정이 길어지면 기존 안전거래 서비스가 가진 느린 거래와 같은 문제를 해결할 수 없기 때문에, 검증 과정 시간을 최대한 줄일 수 있어야 한다. 이를 위해 검증 과정을 빠르게 처리할 만큼의 인센티브 즉, 수수료 책정이 필요하다. 이는 판매자가 검증을 기다리면서 얻을 이익(수수료)이며 상품의 가치에 따라 적절한 수준이다. 동시에 구매자가 요청함에 있어서 부담되지 않는 수준이다.

종합적으로 거래에 대한 빠른 Processing 을 위해서 검증 수수료는 전체 시장의 유동성, 가상화폐 가치, 거래되는 상품의 가격에 따라 탄력적으로 책정된다. 검증 과정에 있어 사용자의 반응 속도를 고려해 거래 당 1-2 시간 내외의 threshold 를 목표로 한다.

#### 6.2.1.4. Usability

Blockchain 에 익숙하지 않은 사용자를 위해서 사용하기 쉽도록 만들어져야 한다. Blockchain 원리에 대한 사용자가 알아야 할 만큼의 정보가 Tutorial 과 같은 방식으로 제공되어야 한다. 또한 복잡한 작동 원리보다 서비스 이용에 필요한 거래 검증 과정, 가상 화폐 관리 방법과 같은 기존의 중고 시장 앱과는 다른 기능을 교육하고 익숙하게 만드는 데에 기여해야 한다.

시스템 구현에 있어서, 복잡한 원리 보다 각 Seller, Buyer, Validator 역할에서 어떤 인터페이스를 어떻게 이용할 지에 대한 개별적인 Instruction 정보가 전달되어야 한다. 앞서 언급한 System Functionality 에 관한 작동법을 첫 로그인 이후부터 제공하며, 사용자 의사로 '그만 볼래요'와 같은 체크박스를 통해 동의할 때까지 인터페이스를 활용해서, 불편함을 감소시켜야 한다.

## 6.2.2. Organization requirement

### 6.2.2.1. Delivery requirement

해당 시스템은 generic product 로서 Time-To-Market 이 중요하다. 플랫폼 서비스인 만큼 최대한 빨리 서비스를 시작해서 더 많은 고객들을 수용해야 하는 필요가 있다. 기존의 서비스 제품들과 경쟁하기 위해서 Waterfall model 을 사용해 꼼꼼하게 검증하고 설계하지만, Delivery 는 최대한 이른 시간 안에 진행되어야 시장을 선점할 수 있다.

구체적인 발매 시간은 Working System 을 만든 이후 Alpha, Beta 그리고 Release Test 를 마치고 바로 출시될 수 있게 해야 한다. 또한 플랫폼 서비스의 초반 사용자 유치를 위해서 지속적인 홍보와 설명, 그리고 가상화폐 리워드와 같은 초기 서비스 참여의 유인을 두어 초기 참여자들의 Feedback 을 받을 수 있도록 한다.

### 6.2.2.2. Implementation requirement

먼저 Blockchain 상에서 동작하기 위해서는 이더리움 네트워크와 연동되는 Solidity 를 사용한다. 해당 언어를 통해 Smart contract 를 만들고 Smart contract 와 연동되는 Web3.js API 를 지원하는 Javascript 와 NodeJS 를 사용해 웹과 서버를 구축해야 한다. 내부 데이터베이스의 검색과 관리를 위해서는 SQL 을 사용한다.

Frontend 웹 UX/UI 개발을 위해서 React, Vue.js, Ionic 등의 개발 툴을 이용한다. Backend 의 데이터베이스 관리를 위해서는 MongoDB 를 사용하고 Blockchain ETH-net 은 Remix 를 이용해서 구현 및 관리한다.

### 6.2.3. External requirements

#### 6.2.3.1. Interoperability

해당 시스템은 Blockchain 위에 저장된 정보를 사용하기 때문에 서버와의 상호 호환성이 매우 중요하다. 서버에서 사용하는 Oracle 과 Blockchain 이 연동되어 정보를 주고받고 웹페이지에 Blockchain 위의 정보를 운용할 수 있어야 한다. Reliability 를 확보하기 위한 노력과 연결되는 요구사항이다.

또한 화폐 간 상호 운용성도 중요하다. 해당 시스템의 가상화폐가 단순히 고립된 화폐가 아니라 비트코인, 이더리움, 리플 등 다양한 화폐와 교환될 수 있도록 해야 한다. 해당 비율을 현재 가치를 비교하여 산정될 예정이다.

#### 6.2.3.2. Ethical requirement

해당 시스템을 사용함에 있어 사용자는 거래에 기록되는 개인정보를 유출해서는 안된다. 해당 정보는 개인정보보호법에 따라 보호되기 때문에 해당 정보를 유출하는 행위는 금지되어야 한다. 또한 사용자 간의 거래에 있어 악의적으로 잘못된 정보를 입력해서는 안된다. 해당 행위는 사기죄에 해당하며 사용자의 혼란을 막기 위해 금지되어야 한다.

구체적인 처벌에 대한 방침은 Tit-for-Tat 원칙을 선택한다. 선량한 사용자와는 지속적인 유대관계를 유지하며, 수수료와 같은 유인들로 서로의 관계를 견고히 한다. 하지만 사용자의 고의적인 서비스 오용, 개인정보 유출 등의 범죄가 발견되면 해당 개인정보의 계정을 영구 정지시키고 형사처벌과 관련한 수사에 협조한다.

#### 6.2.3.3. Legislative requirement

해당 시스템은 사용자가 거래에 입력하는 정보를 거래 검증인에게 공개한다. 이 과정에서 당사자의 동의가 없다면 제 3 자에게 제공하는 것은 금지되기 때문에 일부 정보를 공개하기 이전에 사용자의 동의가 필수적이다. 개인정보를 공개하는 동의와 불법적인 유출을 막겠다는 동의 내역서가 필요하다.

해당 시스템은 가상 화폐를 제작하고 유통하기 때문에 금융감독원의 허가가 필요하다. 해당 가상화폐는 ICO 를 통해 모금을 진행하게 되는데 허가 없는 기금 모금 사업은 법률 위반으로 법적인 처벌을 피할 수 없다. 때문에 가상화폐에 대한 금융감독원의 허가를 받아 ICO 를 진행해야 한다.

## 6.3 System Scenario

### 6.3.1. 서비스 가입/접속/접속\_종료 시나리오

#### 6.3.1.1. 서비스 가입

##### - Initial Assumption

사용자가 가입한 계정이 없거나 혹은 새로운 계정으로 가입하려 한다.  
아이디, 전화번호의 조합은 기존의 데이터 베이스에 없다.

##### - Normal Flow of Events

사용자는 원하는 아이디, 비밀번호, 전화번호, 계좌번호, 주소, 이메일 등의 시스템 기능 요구사항 6.1.1 에서 정의된 입력 값을 넣는다. 웹에서는 데이터베이스로 해당 값들을 조직하여 저장하고 관리한다. 가입시 동시에 가상화폐 관리를 위한 Address 를 생성하고 그 고유 비밀번호 또한 입력 받는다.

##### - What Can Go Wrong

서비스 가입 시 생성한 정보를 사용자가 분실할 수 있다. 낮은 확률로 데이터베이스에 잘 못 저장될 수 있다.

##### - System State on Completion

이용자는 이후 해당 아이디와 비밀번호를 통해서 서비스 사용의 식별, 인증을 받을 수 있다.

#### 6.3.1.2. 로그인

- Initial Assumption

사용자가 먼저 만든 계정이 있다.

- Normal Flow of Events

사용자는 자신의 정보를 입력하고 로그인 버튼을 클릭한다. 데이터베이스에 있는 일치하는 정보가 있는 것을 확인하고, 사용자는 Used2Block 에 접속하여 사용할 수 있게 된다.

- What Can Go Wrong

사용자가 데이터베이스에 존재하지 않는 정보를 입력하는 경우에는, 올바르지 않은 ID 와 패스워드라는 메시지를 보여주도록 한다.

서버가 정상적으로 작동하지 않는다. 연결이 끊겨 있거나 데이터 베이스에 정보가 누락되어 있다. 이미 다른 User Terminal 에서 인증 받은 상태로 남아있다.

- System State on Completion

사용자가 서비스에 접속하여서 Used2Block 서비스를 이용한다.



### 6.3.1.3. 로그아웃

#### - Initial Assumption

사용자는 인증된 사용자의 계정으로 Used2Block 서비스를 이용하고 있었다.

#### - Normal Flow of Events

모든 서비스를 마친 뒤, 사용자는 서비스 접속 종료를 요청한다. 요청된 입력을 통해서 서버는 사용자 계정 인증을 만료하고, 해당 계정에 대한 사용자의 접근 권한을 중단한다. 이용자는 다시 메인 페이지 혹은 로그인 페이지로 이동한다.

#### - What Can Go Wrong

정상적인 로그아웃이 일어나지 않을 경우에 계속해서 해당 User Terminal 에 인증된 상태로 유지되어 보안을 위협받는다.

#### - System State on Completion

다음 인증 시까지 해당 계정은 안전하게 보호된다.

## 6.3.2 가상 Wallet 관리 시나리오

### 6.3.2.1. Wallet 관리

#### - Initial Assumption

회원가입 시 생성된 가상화폐 관리용 Address 와 사용자만이 알고 있는 key 값이 존재한다.

#### - Normal Flow of Events

Wallet 페이지 열람 요청을 사용자가 요구한다. 요구된 요청을 시스템은 처리한다; 해당 사용자의 거래내역, 잔고, 잔고변동내역 등의 시스템 요구사항에서 명시된 정보들을 페이지에 출력한다. 이용자는 원하는 정보를 얻고 페이지를 떠나거나 다른 작업을 수행한다.

#### - What Can Go Wrong

사용자가 key 값을 분실하여 접근할 수 없다.  
거래내역 정보를 Blockchain 또는 데이터베이스로부터 읽어올 수 없다.

#### - System State on Completion

페이지 종료 시점까지 Wallet 페이지를 통해서 사용자는 개인의 잔액, 거래내역, 잔고 변동 내역 등을 확인할 수 있다.

### 6.3.3 거래 시나리오

#### 6.3.3.1. 판매자의 상품 등록

##### - Initial Assumption

등록하고자 하는 물건이 중복되어 등록되지 않는다. 따라서 판매자는 등록 전에 확인이 필요하다.

##### - Normal Flow of Events

판매자는 상품명, 카테고리, 가격, 설명, 사진 등 시스템 요구사항에서 명시했던 입력 값들을 입력한다.

입력된 값들을 이용해서 새로운 판매 상품이 등록되고 이는 개별적인 Item Page 로 관리된다. 데이터 베이스에 저장된 상품 정보는 다른 구매자, 판매자, 검증자 통칭해서 사용자에게 검색되고 열람 될 수 있다.

##### - What Can Go Wrong

상품정보가 올바르지 않을 수 있다.  
중복된 상품이 등록되어 시스템 전체의 공간을 불필요하게 차지할 수 있다.  
상품 등록에 제한이 없지만 바람직하지 않은(법률적, 윤리\_보편적 관점에서 어긋나는) 거래 상품이 등록될 수 있다.

##### - System State on Completion

완성된 상품 페이지는 접근이 가능하며, 검색을 통해서도 정보 접근이 가능하다. 판매자는 다시 사용자 페이지, 또는 새로운 상품 페이지로 연결된다.

#### 6.3.3.2. 구매자의 검색 및 구매 주문

##### - Initial Assumption

구매자가 원하는 물건이 있고, 판매자는 팔고 싶은 동일한 물건을 등록했다.

##### - Normal Flow of Events

구매자는 사고자 하는 물건을 Used2Block 데이터 베이스에서 검색한다.

검색을 통해 얻어진 상품 중 원하는 상품을 골라서 Product Page 를 열람, 상품 정보를 획득한다.

원하는 상품임을 확인한 구매자는 구매 주문을 넣는다. 이 때, 구매자의 선택과 판매자의 동의에 따라 거래가 검증거래 또는 일반 거래로 이루어질지 선택된다. 검증거래의 경우 별도의 검증 과정을 거친다. (이는 다음 시나리오 부분에서 서술된다)

##### - What Can Go Wrong

판매자가 판매를 원하지 않아서 거래가 성사되지 않는다. 또는 다른 요인으로 인해서 판매자가 판매 거래를 확인, 수락하지 않는다. 허위 매물이 발생한다. 이를 검증 과정 없이 놓칠 수 있다.

##### - System State on Completion

Blockchain\_검증과정으로 이관되거나 간편 거래가 성립되어 구매자와 판매자 모두 참여한 거래가 시작된다.

### 6.3.3.3. Blockchain 거래 검증 과정

#### - Initial Assumption

Buyer 와 Seller 가 합의하에 거래가 진행되었다. 해당 거래에 대한 검증 기간, 판매자가 올린 사진, 판매자의 전화번호, 계좌 등이 기재된다.

#### - Normal Flow of Events

구매자와 판매자의 거래 시작 그리고 검증 요청에 따라 다시 상품은 검증을 위한 대기 상태에 놓이며 검색될 수 있다. 3rd Party, 즉 거래 검증인들은 해당 거래에 기재된 판매자에 대한 정보를 통해 사기 기록을 검색을 진행한다. 사기기록은 타 서비스 플랫폼의 정보 혹은 해당 Blockchain 에 기록된 거래내역을 통해서 이루어진다.

검증 기간이 종료될 때까지 해당 판매자에 대한 사기 관련 기록이 없어 report 가 발생하지 않으면 해당 거래는 정상적으로 진행된다.

또는 검색을 통해 해당 판매자에 대한 사기 기록이 발견되면 해당 거래를 파기한다.

검증에 소요되는 수수료를 구매자가 일부 지불하고, 판매자와 검증인은 시스템으로부터 수수료를 가상화폐로 받는다.

#### - What Can Go Wrong

3rd party 가 거래를 검증하는 도중, Buyer 또는 Seller 가 해당 거래를 취소할 수 있다. 이럴 경우를 대비해 거래 검증 도중에는 취소가 불가능하게 한다.

3rd party 가 잘못된 정보를 Report 할 수 있다. 해당 내용에 대한 근거를 기록하도록 해 사용자들이 해당 report 를 한번 더 판단할 수 있도록 한다.

거래 검증에 아무도 참여하지 않을 수 있다. 이럴 경우 해당 거래는 미 검증 상태로 자동 파기된다.

#### - System State on Completion

거래 검증 과정이 정상적으로 진행되어 Buyer 와 Seller 가 서로를 신뢰하게 되고 초기 Blockchain 에 기재된 내용으로 거래를 진행한다.

#### 6.3.3.4. 해당 거래 모델에 대한 참조용 도식

### Normal Transaction Process

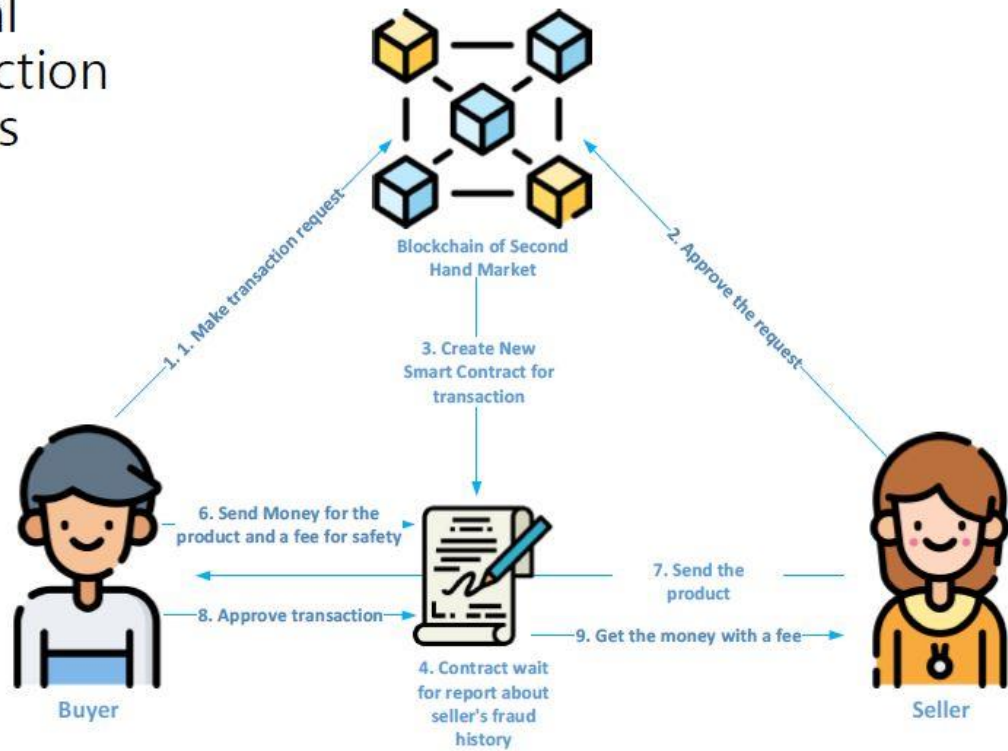


Figure 6-1. 일반 거래

1. 거래자의 거래 요청 – 2. 판매자의 승인 – 3. Smart Contract 문서 생성– 4. 송금 – 5. 배송 – 6. 구매자의 상품 확인 후 거래의 승인 – 7. 판매자의 수금.

## Detecting Abnormal Transaction Process

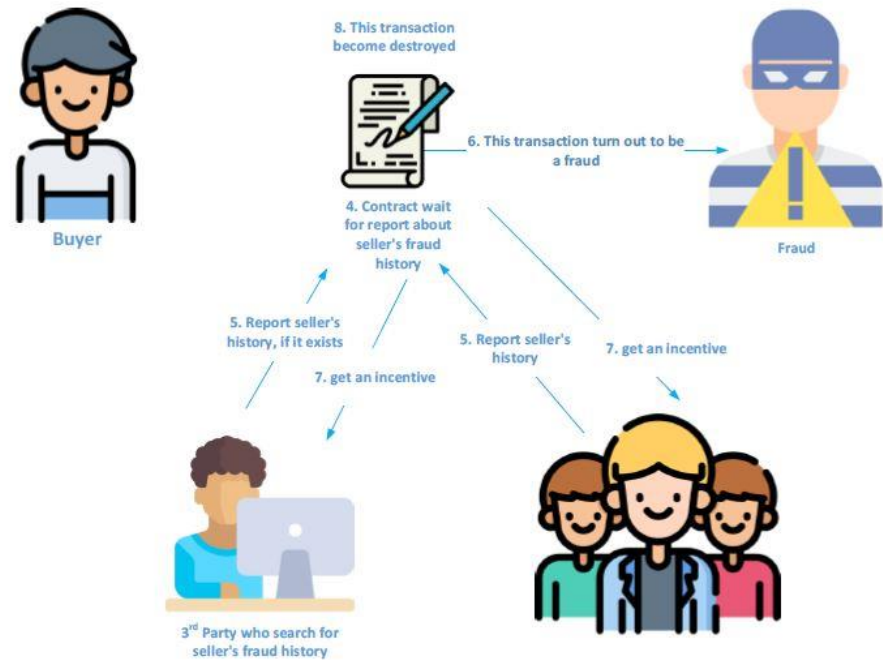


Figure 6-2. 검증 거래 모형

1. 거래자의 거래 요청 - 2. 판매자의 승인 - 3. Smart Contract 문서 생성- 4. 판매자의 사기 관련 검증 대기 - 5. 제3자의 검색 및 검증 - 6. 성공 시 인센티브 부여/거래파기 - 7. 구매자의 상품 확인 후 거래의 승인 - 8. 판매자의 수금.

## 7. System Model

In this chapter, the relationship between the various system components, the entire system, and the environment surrounding the system is described by a variety of diagrams.

### 7.1. Context Models

#### 7.1.1. Context Diagram

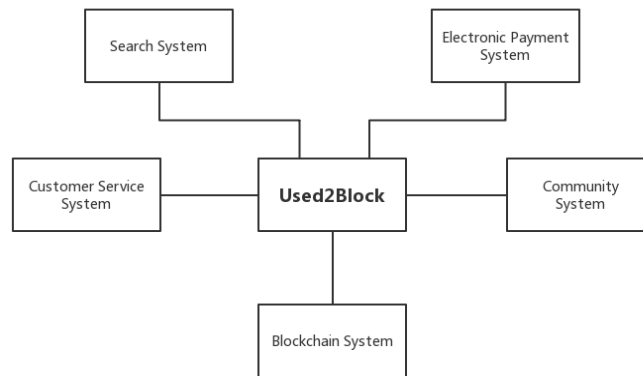


Diagram 7-1: Overall context diagram



## 7.2. Process Diagram

### 7.2.1. Sign up process

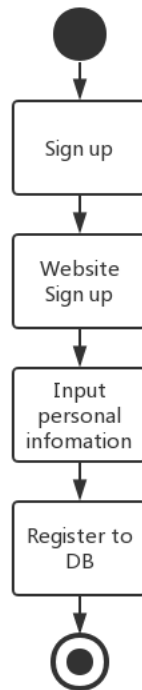


Diagram 7-2: Sign up process

### 7.2.2. Overall process

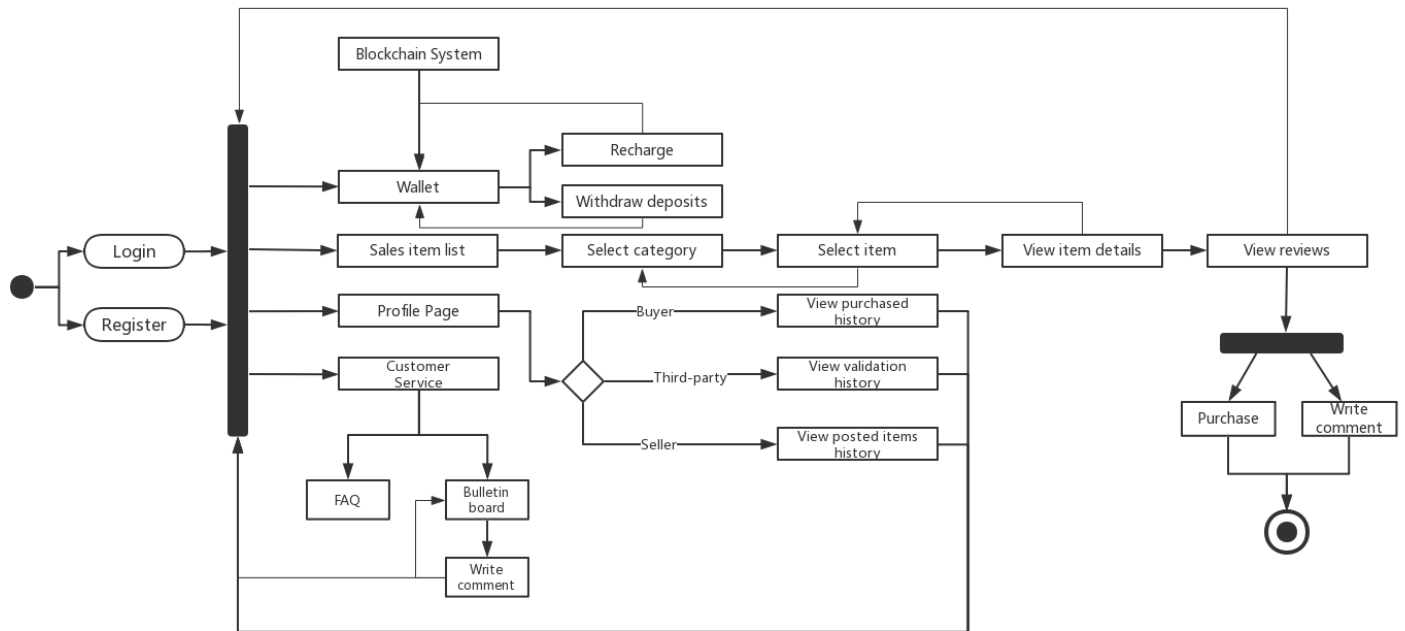


Diagram 7-3: Overall process

## 7.3. Interaction Models

### 7.3.1. Use case Diagram

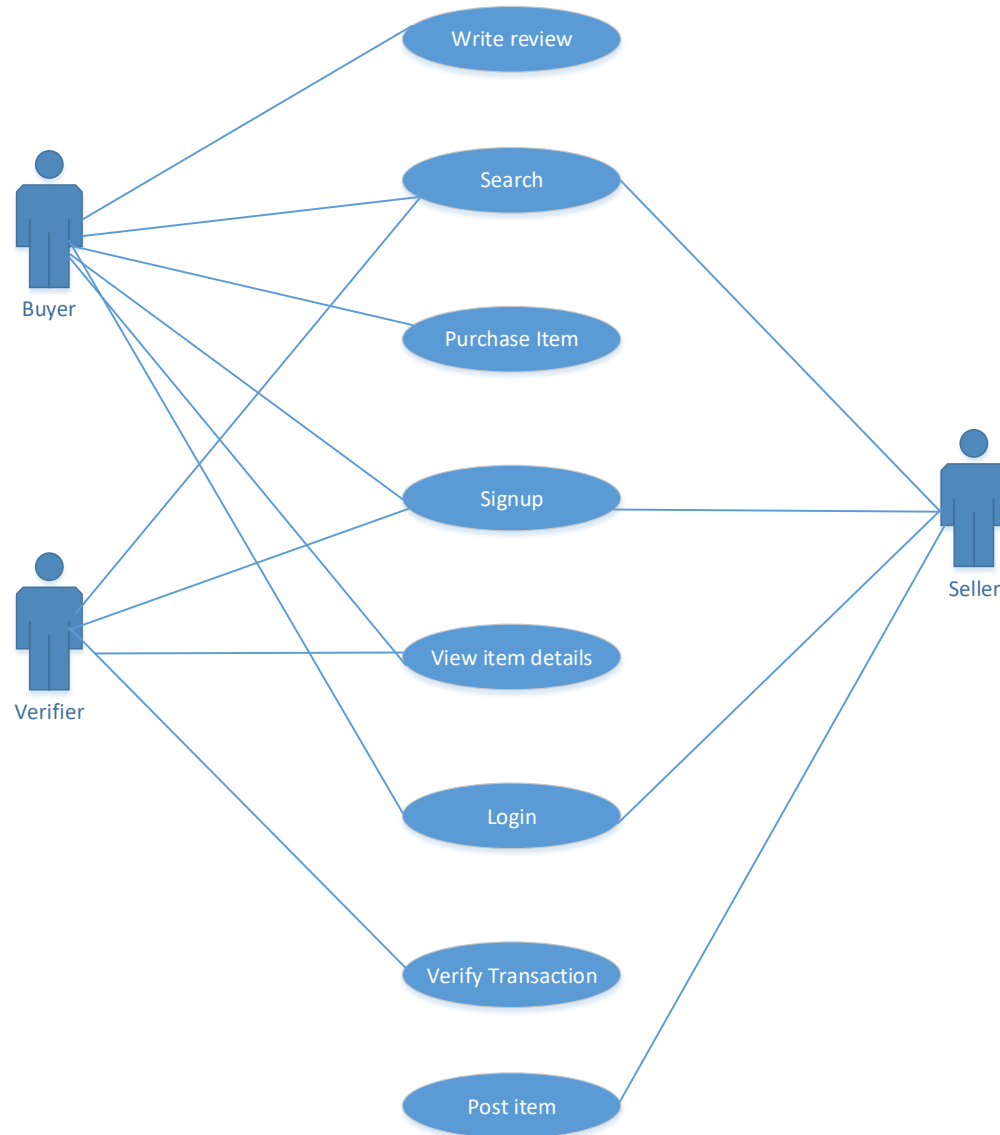


Diagram 7-4: Use case process

### 7.3.2. Tubular description for each use case

#### 7.3.2.1. Sign up

Use Case	Sign up
<b>Actor</b>	Buyer, Seller, Validator
<b>Description</b>	<ul style="list-style-type: none"><li>- Ability to request and receive personal information from the user to obtain membership and save this information in the user database</li><li>- When a user registers for the Used2block service, they must enter their mobile number in their member information.</li></ul>
<b>Data</b>	User's ID and password.
<b>Stimulus</b>	When running the service, request to register at the login screen, and then enter the necessary information to execute the membership.
<b>Response</b>	<ul style="list-style-type: none"><li>- Print out the information needed to join the form on the screen.</li><li>- If the form entered by the user is the correct information, a completion message will be displayed after updating to the user database.</li></ul>

Table 7-1. Tubular Table: Sign up

#### 7.3.2.2. Login

Use Case	Login
<b>Actor</b>	Buyer, Seller, Validator
<b>Description</b>	It is a process of determining whether the connected user is consistent with user information registered in the system.
<b>Data</b>	User's ID and password
<b>Stimulus</b>	Login request on login screen when running the service
<b>Response</b>	<ul style="list-style-type: none"><li>- ID / PW input form required for login on login page</li></ul>

- The information entered by the user is compared with the information stored in the user database.
- If it matches, the login success message is displayed, if not, the failure message is displayed.

Table 7-2. Tubular Table: Login

#### 7.3.2.3. Search

Use Case	Search
<b>Actor</b>	Buyer, Seller, Validator
<b>Description</b>	A user searches for a product by name, category, or product description.
<b>Data</b>	User's search keywords
<b>Stimulus</b>	Users enter their search criteria and press the search button.
<b>Response</b>	<ul style="list-style-type: none"> <li>- Modified search criteria, the database SQL query and returns processing to the list of items that match after executing the query.</li> <li>- If no products match the search criteria, an empty list is returned, and an error message is issued stating that no products are found.</li> </ul>

Table 7-3. Tubular Table: Search

#### 7.3.2.4. View item details

Use Case	View item details
<b>Actor</b>	Seller, Validator
<b>Description</b>	Check the product details listed in the recommendation list, ranking, search, etc.
<b>Data</b>	Product information category.
<b>Stimulus</b>	The user clicks on a product list entry.
<b>Response</b>	The product ID is used to retrieve the product information with the corresponding ID from the database.

Table 7-4. View item details

7.3.2.5. Purchase item

Use Case	Purchase item
<b>Actor</b>	Buyer
<b>Description</b>	The buyer purchase an item of a seller.
<b>Data</b>	Buyer's address and a period of verification.
<b>Stimulus</b>	The user clicks on one of the shopping lists.
<b>Response</b>	<ul style="list-style-type: none"> <li>- Go to the corresponding product page.</li> <li>- If the corresponding product is deleted or different from the stored condition, the connection is interrupted and an error message is displayed.</li> </ul>

Table 7-5. Purchase item

7.3.2.6. Write review

Use Case	Write review
<b>Actor</b>	Buyer
<b>Description</b>	The user writes a review for a product.
<b>Data</b>	User's review, etc.
<b>Stimulus</b>	The user clicks a review button for a specific product and enters a review title and description.
<b>Response</b>	<ul style="list-style-type: none"> <li>- Registers a user's new review in the review database and executes the analysis for the review in the Review Analysis System.</li> <li>- If the review is too short or uses slang, an error message is generated without registering the review.</li> </ul>

Table 7-6. Write review

#### 7.3.2.7. Post item

Use Case	Post item
<b>Actor</b>	Seller
<b>Description</b>	Service that Seller proceeds to sell product.
<b>Data</b>	User's product price, quantity, media files, etc.
<b>Stimulus</b>	The seller enters product-uploading page and registers a specific product with descriptions
<b>Response</b>	When the Seller uploads the product details, the product is stored onto DB with the product name, price, specifications, etc.

Table 7-7. Post item

#### 7.3.2.8. Delete item

Use Case	Delete item
<b>Actor</b>	Seller
<b>Description</b>	The user deletes the items that have been posted.
<b>Data</b>	Reason for deletion.
<b>Stimulus</b>	When the user selects the item to be removed, click the button of the product to be removed.
<b>Response</b>	The item information of the item selected by the user is deleted from the item list.

Table 7-8. Delete item

#### 7.3.2.9. Verify transaction

Use Case	Verify transaction
<b>Actor</b>	Validator
<b>Description</b>	The Validator validate the history of past transactions of a seller.
<b>Data</b>	Date of past transactions, product URL, etc.
<b>Stimulus</b>	After the Validator finds the transaction history of a seller, click the Post button to inform a buyer of the history.

- Response**
- If the report is successful, buyer decide whether to continue a transaction or not.
  - If there is no report, or the transaction destroyed.

Table 7-9. Verify transaction

## 7.4. Behavioral Models

### 7.4.1. Data-Driven Modeling

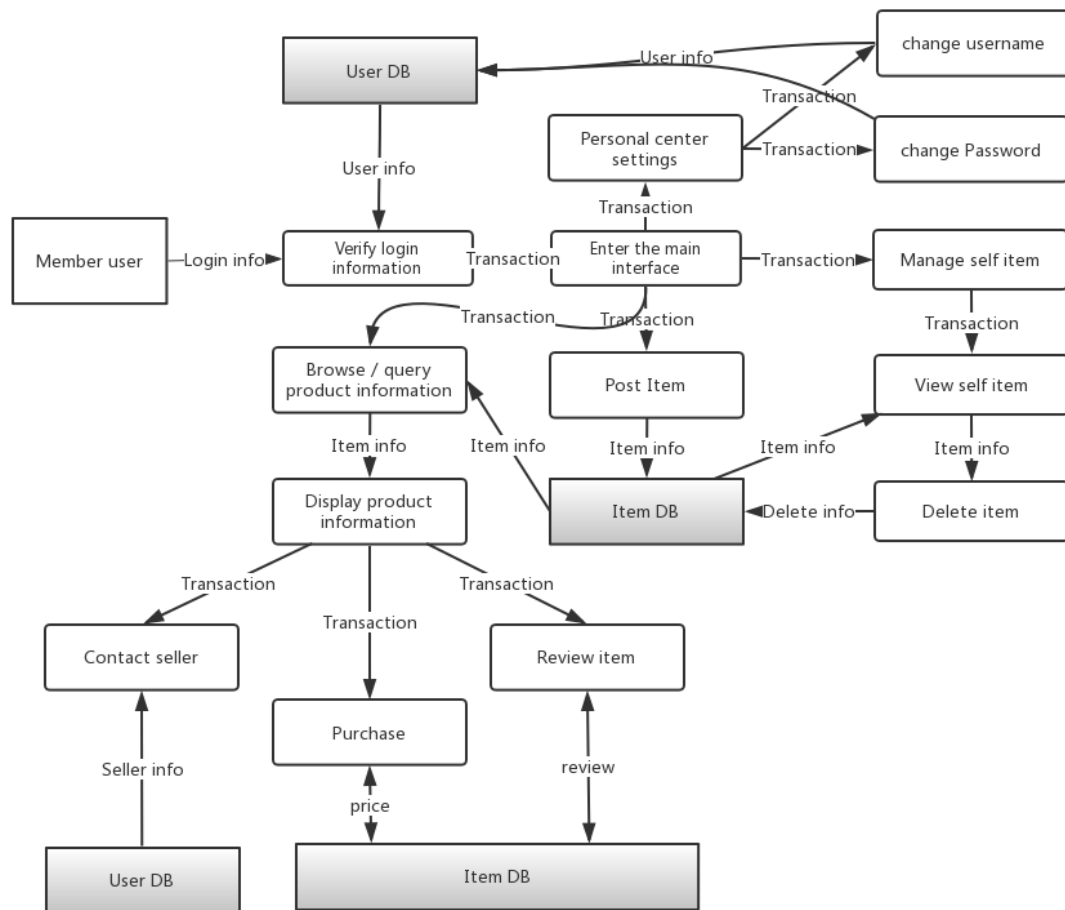


Diagram 7-6: Member user DFD



## 8. System Evolution

### 8.1. 경매 시스템

기본적으로 구매자는 같은 물건을 더 낮은 가격에 사기를 원하고, 판매자는 더 높은 가격에 판매하기를 원한다. 판매자 측이 기업이 아니라 일반인이기 때문에 구매자의 욕구를 충족해주는 것은 힘들 수도 있으며, 물건 가격을 협상하는 과정에서 스트레스를 받을 수도 있다.

이러한 경우에는 판매자가 물건을 올릴 때, 경매 기능을 사용할 수 있도록 한다. 구매를 원하는 구매자들이 경매를 참가해 경매 참가자들은 자신이 구입하고 싶은 가격을 제출하여 구매가를 갱신한다. 판매자는 자신이 희망한 판매가와 경매 최고 갱신액 중 원하는 가격으로 자신의 상품을 판매할 수 있다.

### 8.2. 품질 보증 시스템

구매자는 물건을 사더라도 최상의 상태인 물건을 사기를 원하지만 인터넷으로 거래되는 중고거래 특성 상 물건 상태를 직접 확인하기는 어렵다.

이러한 단점을 해소하기 위해, 구매자가 원하는 경우에 한하여

- i) 전자기기 같이 A/S 센터에서 수리를 받는 물품에 대해서는 A/S센터에서 수리 로그를 받아서 Blockchain 상 공개하도록 한다.
- ii) 판매자가 판매를 원하는 상품을 Validator에게 배송한다. 전문가 수준에 해당하는 Validator가 상품에 대한 감정을 진행하고 적절한 감정가를 제시한다. 이 방법은 추가적인 인건비를 필요로 하며, 거래 과정의 복잡도와 시간이 증가하게 되지만 고가의 물품에 대한 품질을 보증할 수 있다는 장점이 있다.

### 8.3. 상품 추천 시스템

구매자가 특정 물건을 구매하기 위해 카테고리에서 물건을 찾거나 검색 기능을 활용하여 물건을 찾을 것이다. 해당 정보를 기반으로 로그를 분석하여 구매자가 원하는 물건을 추정한다. 추정한 데이터를 바탕으로, 신뢰도가 높은 방식으로 판매자와 원하는 가격대의 물건들을 나열해 추천 상품을 보여준다.

### 8.4. 개인정보 보호를 위한 hash 값 활용

서비스 초기에는 사용자들의 거래 내역이 없는 상태이기 때문에, 제 3자인 Validator 를 통해 Seller 의 번개장터, 중고나라 등의 타 플랫폼에서의 거래 기록을 확인하여 Seller 의 신뢰도를 확인하는 과정을 거친다. 이 과정에서는 Seller 의 개인정보가 유출될 위험이 크다.

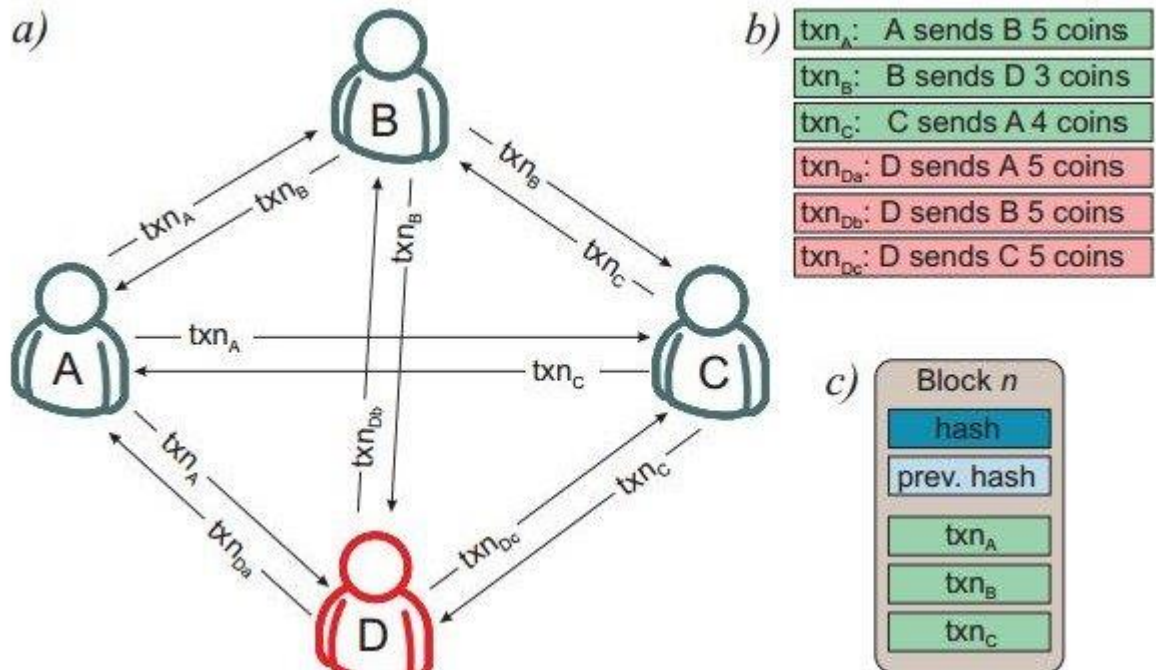
따라서 거래 검증 과정마다 판매자에 대한 검증 여부와 판매자의 개인정보에 대한 Hash값을 Blockchain에 기록하여, hash값을 통해 seller의 신뢰도를 검색 및 확인하는 방식으로 변경한다.

### 8.5 모바일 어플리케이션 연동

Used2Block 은 웹 기반의 플랫폼이다. 일반 사용자들이 일반 PC 보다는 휴대전화나 태블릿 PC 등의 모바일 기기를 많이 사용하는 추세에 따라, Used2Block 모바일 어플리케이션을 출시하여 PC 를 통한 접근 뿐만 아니라 모바일 기기에서의 접근성을 높일 예정이다.

## 9. Appendices

### 9.1 Explanation for Blockchain



Blockchain 은 누구나 열람할 수 있는 장부에 거래 내역을 기록하고 각 거래를 합의 알고리즘을 통해 증명하는 분산형 데이터 저장 기술이다. 해당 정보는 여러 노드에 분산되어 저장되며 노드들은 모두 거래를 증명하는데 참여한다. 이를 통해 데이터를 대조할 수 있고 데이터의 신뢰성을 검증할 수 있다.

### 9.2 Diagram for safe transaction 안전거래



## 10. Indexes

### 10.1. Figures

Figure 2-1. 모바일 중고거래 시장 조사(닐슨 코리아) -----	10
Figure 2-2. 모바일 중고거래 어플리케이션의 성장 추이 -----	11
Figure 2-3. 기존 중고거래 앱의 문제점-자체 화폐 시스템의 불편함(Google 플레이스토어) -----	12
Figure 4-1. Used2Block의 로그인 화면 -----	17
Figure 4-2. Used2Block의 검색 화면 -----	18
Figure 4-3. Used2Block의 판매자 상품 등록 화면 -----	19
Figure 4-4. Used2Block의 주문 검증 화면 -----	20
Figure 5-1. Used2Block의 Overall architecture -----	25
Figure 5-2. Used2Block의 Frontend Architecture -----	26
Figure 5-3. Used2Block의 Backend Architecture -----	27
Figure 5-4. Used2Block의 거래 과정 모식도 -----	28
Figure 6-1. 일반 거래 -----	55
Figure 6-2. 검증 거래 모형 -----	56
Figure 9-1. Blockchain 기본 구조 -----	68
Figure 9-1. 안전거래 기본 구조 -----	68

### 10.2. Tables

Table 3-1. Glossary -----	15
---------------------------	----

Table 6-1. Sign up Function -----	30
Table 6-2. Login Function -----	31
Table 6-3. Logout Function -----	32
Table 6-4. User Management Function -----	33
Table 6-5. Creating Wallet Function -----	34
Table 6-6. Search Function -----	34
Table 6-7. Item Registration Function -----	35
Table 6-8. Buying Item Function -----	36
Table 6-9. Validation Function -----	37
Table 6-10. Item Management Function -----	37
Table 6-11. Item Page -----	38
Table 6-12. Transaction Verifying Function -----	39
Table 7-1. Tubular Table: Sign up -----	61
Table 7-2. Tubular Table: Login -----	61
Table 7-3. Tubular Table: Search -----	62
Table 7-4. View item details -----	62
Table 7-5. Purchase item -----	63
Table 7-6. Write review -----	63
Table 7-7. Post item -----	63
Table 7-8. Delete item -----	64
Table 7-9. Verify transaction -----	64
10.3. Diagrams	
Diagram 7-1: Overall context diagram -----	57

Diagram 7-2: Sign up process .....	58
Diagram 7-3: Overall process .....	59
Diagram 7-4: Use case process .....	60
Diagram 7-6: Member user DFD .....	65

## 11. References

DR. Gavin Wood. (2019) ETHEREUM: A secure decentralized generalized transaction ledger byzantium version

Rosic, A., Blockgeeks, Andonov, S., Mihaylov, M., Dictor, V., & @georgenguyen, G. N. (2019, August 16). Smart Contracts: The Blockchain Technology That Will Replace Lawyers. Retrieved from <https://blockgeeks.com/guides/smart-contracts/>.

Saleh, F. (2018). Blockchain Without Waste: Proof-of-Stake. SSRN Electronic Journal.

Sunny King. (2012) PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake

개인정보 보호법 제 3조 2항.

네이버도 손 못대는 1300만 회원 중고나라...사기범죄 '온상'. (2015, July 14). 머니투데이. Retrieved from <https://news.mt.co.kr/mtview.php?no=2015071310053109767>

모바일 중고마켓 '번개장터', 2018년 연간 거래액 2591억 규모. (2019, February 28). Platum. Retrieved from <https://platum.kr/archives/116293>

불황에 크는 중고시장, 20兆까지 몸집 커졌다. (2018, August 8). 조선일보. Retrieved from

[https://biz.chosun.com/site/data/html\\_dir/2018/08/28/2018082800303.html](https://biz.chosun.com/site/data/html_dir/2018/08/28/2018082800303.html)

50만 원대 제품, 반값 중고 매물 믿어도 되나? (2019, August 22)

<https://1boon.kakao.com/ziksir/5d5e308b3fc43135364965d0>

인터넷서 개인끼리 중고물품 사고팔 때... 안전거래 사이트 '먹튀 사기' 속출 .  
(2013, November 18). 조선일보. Retrieved from

[http://news.chosun.com/site/data/html\\_dir/2013/11/18/2013111800158.html](http://news.chosun.com/site/data/html_dir/2013/11/18/2013111800158.html)