**Requirement Specification**

****

**ARIT Market**

**Group #5**

|  |  |  |  |
| --- | --- | --- | --- |
| **학번** | 2017310887 | **이름** | 손하민 |
| **학번** | 2016314609 | **이름** | 이홍균 |
| **학번** | 2015312193 | **이름** | 임현묵 |
| **학번** | 2017311337 | **이름** | 한상미 |

Contents

[1. Preface 5](#_Toc23095497)

[1.1. Objective 5](#_Toc23095498)

[1.2. Readership 5](#_Toc23095499)

[**A. User Requirements Readership** 5](#_Toc23095500)

[**B. System Requirements Readership** 5](#_Toc23095501)

[1.3. Document Structure 5](#_Toc23095502)

[**1.3.1** **Preface** 5](#_Toc23095503)

[**1.3.2** **Introduction** 5](#_Toc23095504)

[**1.3.3** **Glossary** 6](#_Toc23095505)

[**1.3.4** **User Requirements Definitions** 6](#_Toc23095506)

[**1.3.5** **System Architecture** 6](#_Toc23095507)

[**1.3.6** **System Requirements Definitions** 6](#_Toc23095508)

[**1.3.7** **System Modeling** 6](#_Toc23095509)

[**1.3.8** **System Evolution** 7](#_Toc23095510)

[2. Introduction 8](#_Toc23095511)

[2.1. Objective 8](#_Toc23095512)

[2.2. Needs 8](#_Toc23095513)

[2.3. ARIT 11](#_Toc23095514)

[2.4. Expected Effects 12](#_Toc23095515)

[3. Glossary 13](#_Toc23095516)

[3.1. Objective 13](#_Toc23095517)

[3.2. Term Definition 13](#_Toc23095518)

[4. User Requirements Definitions 15](#_Toc23095519)

[4.1. Objective 15](#_Toc23095520)

[4.2. Functional Requirements 15](#_Toc23095521)

[**4.2.1** **Sign up** 15](#_Toc23095522)

[**4.2.2** **Sign in** 15](#_Toc23095523)

[**4.2.3** **Search** 15](#_Toc23095524)

[**4.2.4** **My page** 16](#_Toc23095525)

[**4.2.5** **Post** 16](#_Toc23095526)

[**4.2.6** **View product** 17](#_Toc23095527)

[4.3. Non-functional Requirements 17](#_Toc23095528)

[**4.3.1** **Performance Requirements** 17](#_Toc23095529)

[**4.3.2** **Security Requirements** 18](#_Toc23095530)

[**4.3.3** **Software Quality Attributes** 18](#_Toc23095531)

[**4.3.4** **Business Rules** 19](#_Toc23095532)

[5. System Architecture 20](#_Toc23095533)

[5.1. Objective 20](#_Toc23095534)

[5.2. Overall Architecture 20](#_Toc23095535)

[5.3. Post System 21](#_Toc23095536)

[5.4. List System 22](#_Toc23095537)

[5.5. AR Measurement System 22](#_Toc23095538)

[5.6. AR Arrangement System 23](#_Toc23095539)

[6. System Requirements Specifications 24](#_Toc23095540)

[6.1. Objective 24](#_Toc23095541)

[6.2. Functional Requirements 24](#_Toc23095542)

[**6.2.1** **Sign up** 24](#_Toc23095543)

[**6.2.2** **Sign in** 24](#_Toc23095544)

[**6.2.3** **Search** 25](#_Toc23095545)

[**6.2.4** **My page** 26](#_Toc23095546)

[**6.2.5** **Post** 26](#_Toc23095547)

[**6.2.6** **View product** 28](#_Toc23095548)

[6.3. Non-functional Requirements 29](#_Toc23095549)

[**6.3.1** **Performance Requirements** 29](#_Toc23095550)

[**6.3.2** **Security Requirements** 29](#_Toc23095551)

[**6.3.3** **Software Quality Attributes** 30](#_Toc23095552)

[**6.3.4** **Business Rules** 30](#_Toc23095553)

[6.4. Scenario 31](#_Toc23095554)

[**6.4.1** **LogIn** 31](#_Toc23095555)

[**6.4.2** **Buyer’s state** 31](#_Toc23095556)

[**6.3.3** **Seller’s state** 32](#_Toc23095557)

[**6.3.3** **dealing state** 32](#_Toc23095558)

[7. System Models 33](#_Toc23095559)

[7.1. Objective 33](#_Toc23095560)

[7.2. Context Models 33](#_Toc23095561)

[**A. Context Diagram** 33](#_Toc23095562)

[**B. Process Diagram** 33](#_Toc23095563)

[7.3. Interaction Models 34](#_Toc23095564)

[**A. Use-Case Diagram** 34](#_Toc23095565)

[**B. Tabular Description for each Use-Case** 34](#_Toc23095566)

[**C. Sequence Diagram** 37](#_Toc23095567)

[8. System Evolution 38](#_Toc23095568)

[8.1. Objective 38](#_Toc23095569)

[8.2. Convenient Product Selection by Sellers 38](#_Toc23095570)

[8.3. Increased Seller’s Confidence 38](#_Toc23095571)

[8.4. Addition of Categories 38](#_Toc23095572)

[9. INDEX 39](#_Toc23095573)

[9.1. Tables 39](#_Toc23095574)

[9.2. Figures 40](#_Toc23095575)

[9.3. Diagrams 40](#_Toc23095576)

# **1. Preface**

## **1.1. Objective**

Preface에서는 이번 보고서에서 예상 독자를 정의하고, 문서의 구조, 각 부분의 역할에 대해 기술한다.

## **1.2. Readership**

### **A. User Requirements Readership**

User Requirements는 본 소프트웨어를 사용할 예상 유저를 예상 독자로 간주한다. 예상 독자의 편의성을 고려하여 자연어와 다이어그램 형태를 이용하여 Requirements을 기술한다.

### **B. System Requirements Readership**

System Requirements는 소프트웨어를 개발하는 개발자나 시스템의 설계자를 주요 독자로 간주한다. Requirements를 세부적으로 기술하고 구조화된 문서 형태로 정리한다.

## **1.3. Document Structure**

### **1.3.1 Preface**

Preface에서는 이번 보고서에서의 예상 독자를 정의하고, 문서의 구조, 각 부분의 서술 내용에 대해서 기술한다.

### **1.3.2 Introduction**

Introduction 에서는 이 소프트웨어의 필요성에 대해서 기술한다. 어떤 배경을 바탕으로 이 소프트웨어가 제안되었는지 설명하고, ARIT 서비스의 기능, 이 서비스가 Social Commerce 시장에 어떠한 영향을 가져올 지에 대해서 설명한다.

### **1.3.3 Glossary**

Glossary에서는 이 보고서에서 전반적으로 사용될 전문 용어들에 대한 정의를 간략하게 정의한다. 전문성이 없는 기본 독자들이 이 보고서를 이해하는데 어려움이 없도록 한다.

### **1.3.4 User Requirements Definitions**

User Requirements Definitions 에서는 이 소프트웨어에서 제공하는 기능들에 대해서 이 소프트웨어를 사용할 예상 독자들을 상대로 기술한다. Functional requirement와 non-functional requirement로 나누어 기능을 설명하며, functional 은 이 소프트웨어의 실제 사용 가능한 기술 등을 설명하고, non-functional 에서는 소프트웨어의 실제 기술보다는 이 시스템의 기본적인 performance, security측면에 대해서 설명한다. 전문성이 없는 독자들의 이해를 돕기 위해 자연어나 다이어그램을 사용하도록 한다.

### **1.3.5 System Architecture**

System Architecture에서는 이 어플의 전체적인 구조에서 대해서 기술한다. 다이어그램을 통해서 각 functions들 사이의 관계에 대해서 기술하고, UI에 대해서도 전체적인 구조를 서술한다.

### **1.3.6 System Requirements Definitions**

System Requirements Definitions 에서는 위의 user requirements definitions에서 설명한 functional, non-functional requirements 들을 개발자나, 시스템의 구조를 설계할 설계자들에게 설명하기 위하여, 좀 더 구체적이고 전문적으로 requirements를 기술한다.

### **1.3.7 System Modeling**

System Modeling 에서는 위에서 기술된 Architecture 와 requirements를 기반으로 만들어진 소프트웨어 안에서 각 기능들과 architecture 가 어떻게 상호작용하여 UI에 나타나는지에 대해서 기술한다. 독자들의 이해를 돕기 위해서 시각적인 다이어그램을 활용하여 관계를 나타낸다.

### **1.3.8 System Evolution**

System Evolution에서는 hardware의 변화나, user needs의 변화로 생길 변경사항들에 대해서 기술한다. 또한 이 소프트웨어에서 예상 변화에 맞춰서 발전시킬 방향성에 대해서 기술한다.

# **2. Introduction**

## **2.1. Objective**

Introduction 에서는 이 소프트웨어의 필요성에 대해서 기술한다. 어떤 배경을 바탕으로 이 소프트웨어가 제안되었는지 설명하고, ARIT 서비스의 기능, 이 서비스가 Social Commerce 시장에 어떠한 영향을 가져올 지에 대해서 설명한다.

## **2.2. Needs**

제품을 구매하는 방법에 있어서, 인터넷이 제대로 활용되지 않았던 시절에는 온라인 마켓보다는 오프라인 마켓의 비중이 월등히 컸고, 안정성에 있어서도 온라인 시장은 많은 문제점을 가지고 있었다.

하지만 점차 인터넷이 활성화 되고 많은 거래들이 온라인을 통해서 이루어지면서, 이제는 쇼핑에 있어서도, 오프라인에서 제품을 사고 팔기 보다는 온라인으로 제품을 거래하는 일들이 많아지면서, 소셜 커머스(Social Commerce)의 비중이 많이 늘고 있다.

온라인 상으로 제품을 사고 파는 일은 신제품 거래에만 영향을 준 것은 아니다. 신뢰성과 접근성 등의 문제로 쓰던 제품을 사고 파는 일인 secondhand Goods, 이른바 중고거래도 많은 영향을 받았다. 온라인 거래가 활성화되지 않던 때에는 원거리로 중고거래를 하는 것은 상상하기 힘든 일이었고, 신뢰성의 문제로 지인들 사이에서만 중고거래가 이루어지는 식이었다.

하지만 온라인 거래가 활성화 되면서, 판매자와 구매자의 물리적인 거리가 아무리 멀어도, 구매자들은 온라인으로 편리하게 판매자가 팔고자 하는 제품들을 접할 수 있고, 판매자들도 보다 많은 예비 구매자들에게 자신의 중고품을 알릴 수 있게 되었다.

대표적인 예가 ‘중고 나라’라고 불리는 사이트이다. ‘중고 나라’는 2003년에 네이버 카페로 시작하여서, 현재 약 1700만명이 넘는 회원을 보유하고 있는 자타공인 독보적인 중고거래 사이트라고 할 수 있다. 또한 ‘중고 나라’는 2016년 공식 모바일 앱을 출시하면서, 모바일 시장에서도 독보적인 중고거래 앱으로 자리 잡을 수 있었고, 실제 우리 주변에서도 웹이나 앱을 통한 중고거래를 해본 사람을 심심찮게 볼 수 있고, 대부분의 거래를 중고거래로 하는 사람들도 볼 수 있다.

하지만 중고거래의 문제점이 없는 것은 아니다. 온라인 거래의 특성 상, ‘면대면 대화’를 통해서 이루어지기 보다는 온라인으로 맺어진 관계를 바탕으로 판매자는 제품을 보내고, 구매자는 돈을 보내는 거래방식으로 이루어지기 때문에, 온라인 중고 거래에서 발생하는 사기 사례 등은 손쉽게 찾아 볼 수 있다. ‘중고 나라’등 많은 중고 거래 사이트 및 앱에서 사기를 예방하기 위해서 판매자의 신뢰도를 측정할 수 있는 많은 장치를 만들어 둔 상태이지만, 신뢰성에 관한 측면은 온라인 중고거래에서 해결해야하는 문제 중 하나로 항상 대두되고 있다.

중고거래의 문제는 이것만 있는 것이 아니다. 온라인으로 이루어지는 거래 특성 상, 구매자 들은 구매하고자 하는 물건을 실제로 보지 못하고 거래를 해야 되는 입장이기 때문에, 구매를 완료한 후에, 제품이 배달이 왔을 때, 개인적인 기호에서 오는 불만족, 즉 생각했던 크기보다 물건이 작거나 클 수 있고, 디자인이 생각보다 맘에 들 지 않는 경우, 이 발생했을 때에 이를 처리할 수 있는 방법이 많이 않다는 약점을 가지고 있다. 좀 더 구체적인 고객들의 Requirements를 파악하기 위하여 각 조원의 주변사람들 126명을 대상으로 설문조사를 시행하였다.

최대한 많은 조사 대상을 모으고 자료를 수집하기 위하여, 설문 조사를 기반으로 조사를 실시하였고, 조사 대상의 다양성을 위하여 최대한 겹치지 않는 조원들의 구성원들을 선택하였다.

첫째로, 중고 거래를 포함하여 온라인으로 제품을 거래해 본 적이 있는 질문에 34명이 자주 한다에, 74명이 자주는 아니지만 경험은 있다에 체크를 함으로써 현재 20대의 많은 고객들이 온라인으로 거래를 한다는 것을 알 수 있었다.

두번째로 맘에 드는 제품을 발견하였지만 거래를 하지 않은 적이 있냐는 질문에 34명이 없다, 92명이 있다라고 하였고, 많은 사람들이 온라인 거래를 선호하지만 아직까지 온라인 거래의 문제점들이 완전히 사라지지는 않았음을 알 수 있었다.

마지막으로 거래를 하지 않은 적이 있다면, 어떤 이유 때문에 거래를 하지 않았냐는 질문에, 44명이 사기에 대한 두려움 때문으로 가장 많았고, 26명 물건을 실제로 볼 수 없는 등 제품의 정보가 부족하다는 이유가 두번째로 많았다.

그래서 우리는 실제로 볼 수 없다는 점에 대한 문제점, 특히 온라인 상에 나와있는 크기를 정확하게 판단하기 힘든 점에서, 이 시스템을 개발 하기로 결정하였고, 논의 결과 현재 신기술로 대두되고 있는 AR(Augmented reality)를 기반으로 소프트웨어를 개발하기로 결정하였다.

AR(Augmented reality), 즉 증강현실은, 가상 현실(VR)의 한 분야로 실제로 존재하는 환경에 가상의 사물이나 정보를 합성하여 마치 원래의 환경에 존재하는 사물처럼 보이도록 하는 컴퓨터 그래픽 기법이다. 우리는 만약 이 AR을 중고 거래에 도입한다면, 판매자는 자신이 팔고자 하는 제품을 올릴 때 쉽게 크기를 측정하여 올릴 수 있고, 구매자는 자신이 사고자 하는 제품의 크기를 AR을 통해서 가시적으로 확인하고 구매를 할 수 있다는 점에서 좀 더 온라인 중고 거래의 신용도 및 활용도를 높여줄 것으로 기대하고 있다.

## **2.3. ARIT**

우리는 중고 거래로 거래가 이루어지기 힘든 항목 중 하나가 전자제품에 관한 거래라고 생각하였다. 그 중에서도 개인용 컴퓨터(PC)나 노트북, 등을 거래할 때 중요한 제품의 성능이 너무나도 전문적인 지식으로 되어 있어서 PC나 노트북에 전문적인 지식이 없는 사람들은 거래를 할 때 불편함이 많고, 크기도 빼놓을 수 없는 거래할 때에 고려해야할 항목이지만 화면 상에 있는 단편적인 수치로는 정확하게 크기를 예측하기 힘들다는 문제점이 있었다. 이 문제점을 해결하기 위하여 우리는 중고거래 중에서도 전자제품, 특히 PC나 노트북을 거래하는 데에 초점을 맞추고, 이에 대한 편의성을 고려하여 소프트웨어을 개발하고자 한다.

이 소프트웨어는 크게 두 부분에서 메인 기능을 볼 수 있다. 처음은 구매자나 판매자가 제품을 팔기 위해 글을 올리고, 제품을 사기 위해 글을 확인하는 카테고리 부분이다. 이 부분은 다른 많은 중고 거래 어플과 비슷하지만, 좀 더 PC 쪽에 집중하여서, 카테고리를 세분화하고 전문적인 지식이 없는 고객들도 쉽게 제품을 보고 구매할 수 있도록 편의성을 도모하였다. 자신이 본 제품이나 사고자 하는 제품을 담을 수 있는 개인용 장바구니도 만들어서, 쉽게 물건을 비교 해 볼 수 있도록 하였다.

다음은 이 소프트웨어의 핵심 기술인 AR 기술이다. 판매자는 자신이 제품을 올릴 때 사이즈를 쉽게 측정할 수 있도록 AR 서비스를 제공한다. 판매자는 그저 핸드폰 화면에 사진이 팔고자 하는 물건을 위치시켜서 사이즈를 측정하면, 자동으로 측정된 사이즈를 어플의 데이터베이스에 전송 시켜, 제품을 팔고자 하는 게시글을 올릴 시에 자동으로 업로드 된다. 또한 구매자는 AR화면으로 저장되어 있는 사이즈를 가져와서 자신이 두고자 하는 책상, 혹은 선반 등에 위치시켜 봄으로써 수치 상으로만 나와있는 제품의 크기를 좀 더 가시적으로 확인해 볼 수 있도록 하였다.

## **2.4. Expected Effects**

본 소프트웨어의 가장 큰 기대 효과는 전자 제품 거래의 편의성이다. 상대적으로 다른 제품에 비해 가격이 있는 전자 제품은 다른 제품들 보다 더 온라인 중고 거래를 할 때에 고민이 되고 쉽게 결정하기 힘든 것이 사실이다. 하지만 이 어플을 통해서 좀 더 신뢰도를 높이고, 크기를 AR을 통하여 시각적으로 확인함으로써, 거래를 할 때에 불안 요소를 제거하고 거래에 도움을 줄 수 있도록 하였다.

# **3. Glossary**

## **3.1. Objective**

Glossary에서는 이 보고서에서 전반적으로 사용될 전문 용어들에 대한 정의를 간략하게 정의한다. 전문성이 없는 기본 독자들이 이 보고서를 이해하는데 어려움이 없도록 한다.

## **3.2. Term Definition**

|  |  |
| --- | --- |
| Term | Definition |
| AR(Augmented Reality) | 증강현실(增強現實, augmented reality, AR)은 가상현실(VR)의 한 분야로 실제로 존재하는 환경에 가상의 사물이나 정보를 합성하여 마치 원래의 환경에 존재하는 사물처럼 보이도록 하는 컴퓨터 그래픽 기법이다. |
| Datebase | 데이터베이스(database, DB)는 여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합이다. 작성된 목록으로써 여러 응용 시스템들의 통합된 정보들을 저장하여 운영할 수 있는 공용 데이터들의 묶음이다. |
| Firebase server | 파이어베이스는 14년도에 구글에서 인수한 모바일, 웹 애플리케이션 개발플랫폼이다. 16년도 부터 구글에서 서비스를 대폭 강화한 Baas(Backend as a service) 서비스 이므로, 백엔드를 대신해 줘서 프론트엔드에 집중할 수 있게 서비스를 지원해준다. |
| VR | 가상현실(假想現實, virtual reality, VR)은 컴퓨터 등을 사용한 인공적인 기술로 만들어낸 실제와 유사하지만 실제가 아닌 어떤 특정한 환경이나 상황 혹은 그 기술 자체를 의미한다. 이때, 만들어진 가상의(상상의) 환경이나 상황 등은 사용자의 오감을 자극하며 실제와 유사한 공간적, 시간적 체험을 하게 함으로써 현실과 상상의 경계를 자유롭게 드나들게 한다. 또한 사용자는 가상현실에 단순히 몰입할 뿐만 아니라 실제하는 디바이스를 이용해 조작이나 명령을 가하는 등 가상현실 속에 구현된 것들과 상호작용이 가능하다. |
| 소셜 커머스 | 소셜 커머스(Social commerce)는 소셜 미디어와 온라인 미디어를 활용하는 전자상거래의 일종이다. 소셜 커머스라는 용어는 야후에 의해 2005년에 처음 소개되었다. 이는 제품정보 등에 대한 사용자의 평가나 공유 목록 같은 온라인 협업 쇼핑 도구의 집합을 설명하기 위해서였다. 소셜 커머스는 크게 소셜 링크형, 소셜 웹형, 공동구매형, 오프라인 연동형의 네가지로 분류할 수 있다. |

# **4. User Requirements Definitions**

## **4.1. Objective**

고객(Customer)이 이해하기 쉽도록 쓰여지는 Requirements definition이다. 자연어(natural language)와 다이어그램(diagrams)등을 이용하여 이해하기 쉽고, 읽고나서 단번에 파악할 수 있도록 한다.

## **4.2. Functional Requirements**

### **4.2.1 Sign up**

사용자가 앱에서 자신의 계정을 만들고, 그 계정으로 활동하며, 시스템의 서비스를 사용하기 위해서는 회원가입을 해야 한다. 정보를 요청하고 필요한 정보가 모이면 회원가입을 허가한다. 필요한 정보는 생년월일, 중고거래를 하기 위한 연락처, 주소, 이메일, 닉네임, 비밀번호 등이다.

### **4.2.2 Sign in**

Sign up과정을 거친 사용자는 시스템에 등록되어 설정한 닉네임과 비밀번호로 로그인이 가능하게 된다. 이 로그인 과정을 넘기고 나서부터, 시스템의 서비스를 제공받을 수 있고, 게시물 작성 및 열람이 가능해진다.

### **4.2.3 Search**

사용자가 자신이 원하는 물건을 찾기 위해 사용하는 기능이다. 단어 및 필터를 조합하여 자신이 원하는 물품을 취급하고 있는 게시물을 편리하고, 정확하게 찾을 수 있다. 제공하는 필터로는 판매 중인지 아니면 전체 게시물을 검색할지를 선택할 수 있으며, 최신순으로 정렬이 가능하다.

### **4.2.4 My page**

사용자의 정보는 언제나 바뀔 수 있기에, 이를 수정할 수 있게 하는 기능이 필요하다. 이를 가능하게 해주는 것이 My page 기능이다. 사용자의 정보 수정 만이 아니라 사용자의 게시물도 한눈에 볼 수 있도록 ‘내가 쓴 글’ 기능을 지원하여, 사용자의 게시물을 통합 관리할 수 있도록 한다.

### **4.2.5 Post**

사용자는 자신의 물품을 판매하기 위해서 게시글을 작성하게 된다. 여기서 본 시스템인 ‘ARIT’는 다양한 툴을 제공하여, 물품이 좀 더 효과적으로 구매자들에게 전달될 수 있도록 판매자를 보조하는 역할을 한다.

#### A. AR Measurement

‘ARIT’의 핵심기능 중 하나인, 물품의 크기를 인식하는 기능이다. 판매자는 물품의 크기를 게시글에 게재하도록 되어있는데, 모든 판매자가 물품의 사이즈를 수치로 알고 있는 경우는 적다. 직접 자 등을 이용하여 측정하거나, 다른 곳에서 정보를 알아볼 필요없이, 본 시스템은 AR측정 기능을 도입하여, 곧바로 제품의 크기(높이, 너비, 길이)를 측정할 수 있도록 한다.

#### B. Upload photo

몇 백자의 글보다 한 장의 사진이 더 이해가 쉽다. 제품의 사진을 게시글에 올릴 수 있게 하여, 제품에 대한 자세한 정보를 전달할 수 있도록 한다.

#### C. Provide form

판매자의 연락처 및 택배거래 여부 등을 표시할 수 있게 한다. 또한 판매자가 제품의 사양 등을 효과적으로 전달하기 위한 정보 및 항목에 대한 가이드라인을 제공한다. 대표적인 항목으로는 ‘제조사’, ‘제품 링크’등이 있다.

### **4.2.6 View product**

사용자는 구매를 하기 위해 게시글을 열람하고, 게시글 안에서 가볍게 판매하는 사용자와 댓글로 소통한다. 본 시스템은 구매자의 제품에 대한 이해도를 높이기 위해 여러 기능을 제공한다.

#### A. AR Arrangement

‘ARIT’의 핵심기능 중 하나인 AR배치 기능은 증강현실 기술을 이용하여 구매자가 있는 실제 공간에 판매자가 판매하는 물품을 배치한다. 구매자가 제품에 대한 정보 및 사양을 알아본다고 하더라도, 크기는 실제로 보지 않는 이상 감을 잡기 힘들다. 특히 노트북과 같이 크기에 민감한 제품이나, 데스크탑 본체의 사이즈 등 크기가 사진으로 구분이 잘 가지 않는 것들은 특히 그렇다. 이때, 제품의 크기를 육면체 등의 모양으로 보여주어, 제품의 크기를 실공간에서의 부피를 알 수 있게 한다. 육면체의 크기는 판매자가 AR 측정 기능을 통해 측정한 값을 기반으로 한다.

#### B. Secret Comments

댓글창은 구매자와 판매자가 소통할 수 있는 공간이다. 댓글창은 누구에게나 보인다. 하지만 구매자는 판매자만 볼 수 있는 글을 작성하고 싶을 수 있을 것이다. 이를 기능으로 구현한데 비밀 댓글이다. 오직 댓글을 작성한 사용자와 게시판을 작성한 사용자만 비밀 댓글을 읽을 수 있도록 한다.

## **4.3. Non-functional Requirements**

### **4.3.1 Performance Requirements**

Due to the fast nature of today’s world, performance is even more important. Because of this, efficiency must be emphasized. The app should be efficient and fast, allowing for easy and simple interaction between customer and seller. Signing up and singing in must be done efficiently. Though the increase of the number of users may increase the amount of time for verification, organized and efficient searches must be done in order to ensure that users do not feel frustrated with the sign up and log in process. Though this feeling of frustration may be a subjective feeling, rather than objective, for the general public, this feeling must be avoided. All these things must be implemented in order to ensure that customers can close our app feeling satisfied rather than disappointed and frustrated.

The biggest challenge to achieve high efficiency is to make sure that data is sent from the server to the client efficiently and also to make sure that data from the database is efficiently found. As firebase is being used, and as this online opensource database is run by google, there will be minimal issues on the server side of the network. Now, because this database is online, this means that the network of the client is also important. As the network of the client cannot be controlled, the database efficiency must be focused. Sufficient indexes must be created in order to allow reliable and efficient accesses to the database. There must be minimal accesses to data that is unimportant. The only possible exception to this should be when checking sign in requests, as all users must be checked against the corresponding group of data.

### **4.3.2 Security Requirements**

Security and privacy must be emphasized. Pictures or profiles or other user information that is displayed may contain sensitive materials that users were unaware of. Security is also important for the information that is entered during the sign-up process. There is bound to be personal information during this stage, as they enter information of themselves. Because of this, adequate precautions must be taken. Unauthorized access to the system must not be allowed. That is, there must be an accurate and dependable way to check user. There should be barriers in place to protect user information. Data must be stored in a secure location where the stealing of information can be prevented. In order to prevent user data loss, there must be relevant and up to date back up of data. This may be even more particularly important in events in which user information is exposed.

### **4.3.3 Software Quality Attributes**

As the largest and important software that is to be implemented is the integration of augmented reality, the software must focus on the *portability and flexibility, usability and robustness, maintainability,* and perhaps most importantly, the *correctness* of the system.

First, the *portability and flexibility* of the system. The app must allow augmented reality across the maximum number of devices that supports augmented reality. The importance of this is apparent in the fact that app must allow the maximum number of customers to use our app.

For *usability and robustness*, the app must be easy to use. Augmented reality may be a new concept for many users, and thus, a way to introduce users to the usability should be implemented.

For *maintainability*, the code should be easy to understand to allow developers to maintain the system as a whole.

For *correctness*, the measurements must be close to being accurate to allow sellers to accurately register the size of the product and allow buyers to accurately see the size of the product.

### **4.3.4 Business Rules**

Largely, the people involved will be: users and admins. The users are further divided into producers and consumers. Users are able to be involved in both options. A user can post their product to be viewed by other users, or they are allowed to only view and purchase products. Based on their history of selling and purchasing products, users may increase their credit. There is no credit system per say, yet simply having and showing that a user has successfully sold an object gives assurance to the user that is buying another product from the same person. This is particularly important in the purchase of second-hand materials, as there may be problems with scams. In order to combat this, users are able to look at products that have already been sold, which in turn allows users to see that a seller has successfully sold a product without scams. Furthermore, in order to allow contact between users, contact information of the seller is given to the buyer if they show potential interest.

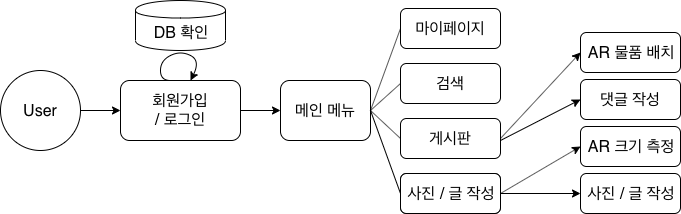
The second of the group are the admins. Though there is not a special reporting function in place, admins should make sure that there is no inappropriate information being uploaded to the database. This includes pictures as well as information that is being entered by the user. Admins should be given the authority to delete any information that they deem necessary – even to the point of deleting user data.

# **5. System Architecture**

## **5.1. Objective**

System Architecture는 시스템 아키텍처에 대한 설명으로 전체 시스템이 어떻게 돌아가는 지에 대한 flowchart와 세부 시스템 각각에 대한 설명으로 이루어져 있다. 해당 시스템을 통해 할 수 있는 기능들과 구조를 대략적으로 설명한다.

## **5.2. Overall Architecture**



**[diagram 1] overall flow chart**

전체적인 시스템이 돌아가는 흐름은 [diagram 1]과 같다. 먼저 사용자가 새로운 사용자라면 회원가입을 통해 user database에 회원 정보를 등록하고 가입한다. 이미 회원가입이 되어있는 사용자라면 로그인을 통해 메인 메뉴 창으로 이동한다. 메인 메뉴로는 마이페이지, 검색, 게시판, 사진/글 작성의 4가지 메뉴가 있다. 사용자는 원하는 메뉴를 선택해 원하는 기능을 이용할 수 있다.

## **5.3. Post System**



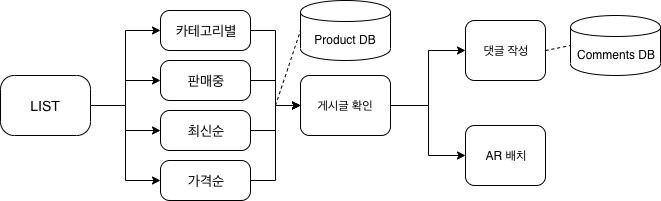
**[diagram 2] post system flow chart**

Post system은 사용자가 게시글을 작성할 수 있는 시스템으로 사용자가 게시글을 업로드 할 때는 [diagram 2]와 같은 흐름을 따라 실행된다.

먼저 제품의 사진을 등록할 땐 카메라 앱을 열어서 제품의 사진을 직접 찍거나 갤러리의 존재하는 사진을 올릴 수 있도록 한다. 글을 작성할 땐 제품의 카테고리를 선택하고 제품의 상세 정보를 입력한다. 상세 정보엔 제품의 판매 가격, 구입 시기, 정가, 제품명(모델명) 등의 양식이 존재하며 이에 맞추어 입력해야한다. 또 게시자가 자유롭게 글을 작성할 수 있는 자유 글 입력 칸도 있다. AR로 크기를 측정하는 기능을 선택하면 카메라 앱을 열고 AR로 제품의 크기 정보를 측정할 수 있다.

정보 입력을 다 했으면 입력한 해당 제품의 정보들과 게시자의 정보가 product database에 등록된다. 게시자의 정보는 user database와도 연결되어 마이페이지에서 자신의 글을 확인할 때 사용된다.

## **5.4. List System**



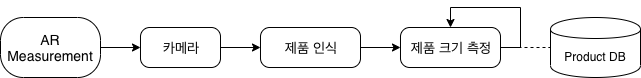
**[diagram 3] list system flow chart**

List system은 사용자가 게시글을 확인하는 시스템으로 정렬 기능, 댓글 기능, AR 배치 기능이 포함되어 있다. 시스템은 [diagram 3]과 같은 흐름으로 실행된다.

먼저 사용자가 게시판 메뉴를 들어가면 중고 판매 대상인 제품들의 전체 목록이 띄워진다. 사용자가 원하는 제품을 확인하기 위해 4가지 정렬 기능을 제공하는데 이는 카테고리별, 판매중 여부, 최신순, 가격순이다.

사용자가 원하는 게시글을 클릭하면 제품의 상세 정보를 열람할 수 있다. 그리고 구매를 원하면 게시글에 댓글을 작성하여 판매자와 연락한다. 또 AR 배치 기능을 이용하여 제품의 실제 크기를 가늠해볼 수 있다.

## **5.5. AR Measurement System**

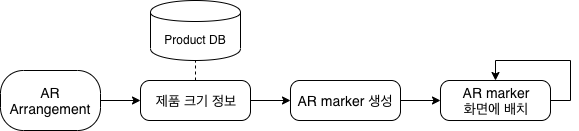


**[diagram 4] AR measurement system flow chart**

AR Measurement system은 제품의 실제 크기를 측정하는 시스템으로 [diagram 4]와 같은 흐름으로 작동된다.

먼저 사용자의 스마트폰의 카메라 어플을 접근하고 이를 통해 제품의 위치를 인식하여 제품의 크기(길이, 너비, 높이)를 측정한다. 측정을 계속 시도하고 성공하면 해당 크기 정보를 product database에 저장한다.

## **5.6. AR Arrangement System**



**[diagram 5] AR arragement system flow chart**

AR Arrangement system은 제품의 실제 크기를 이용해 사용자의 화면에 배치해 주는 시스템으로 [diagram 5]와 같은 흐름으로 작동한다.

먼저 제품의 크기 정보(길이, 너비, 높이)를 product database로부터 가져온 후 해당 크기와 같은 AR marker를 생성해낸다. 그 후 AR marker를 가지고 사용자가 원하는 곳에 배치해볼 수 있다.

# **6. System Requirements Specifications**

## **6.1. Objective**

시스템의 기능과 서비스, 작동 제약 조건 등을 자세하게 명시한다.

## **6.2. Functional Requirements**

### **6.2.1 Sign up**

|  |  |
| --- | --- |
| 이름 | Sign up |
| 기능 | 사용자가 개인 정보를 입력하여 회원가입을 하는 기능이다. |
| 설명 | 사용자가 중고거래를 하기위해 게시물 등을 작성해야 하며, 개인의 주소 및 연락처 등의 정보가 필요하다. 이를 위해 사용자의 기본적인 정보를 모아서 데이터 베이스를 형성한다. |
| 입력 | 사용자의 이메일, 생년월일, 기본 주소, 연락처, 아이디(닉네임)을 받는다. 그 뒤 개인의 패스워드를 설정하여 회원으로서 로그인 할 수 있도록 한다. |
| 출력 | 회원가입이 정상적으로 완료되면, 축하문구와 안내사항이 적혀 있는 화면으로 넘어간다. 그리고 ‘계속 하기’ 버튼을 누르면 홈페이지로 돌아가고, 사용자 이메일로 회원가입 완료 안내 문구를 보낸다. |
| 처리 | 먼저 DB에서 사용자의 이메일을 찾고, 없으면 계속 진행한다. 만약 있다면 이미 있는 계정이라고 안내하고 중지한다. 사용자 정보는 DB에 저장되며, 패스워드는 해시 처리한후 DB에 저장하여 유사시에 사고를 막는다. |
| 조건 | 사용자가 주의 사항 등을 읽고 동의해야 하며, 아이디와 이메일은 중복될 수 없다. |

**[table 1] Sign up requirements**

### **6.2.2 Sign in**

|  |  |
| --- | --- |
| 이름 | Sign in |
| 기능 | 이메일 및 아이디로 로그인하는 기능이다. |
| 설명 | 회원가입을 한 사용자가 본 시스템의 서비스를 이용하기 위해서는 로그인을 해야 한다. 게시글 및 댓글을 작성할 때 누가 작성했는지 표시하기 위해서 로그인 기능이 필요하다. |
| 입력 | 사용자가 회원가입때 설정했던 아이디 또는 이메일과, 비밀번호를 받는다. |
| 출력 | 로그인이 완료되면 기본적으로 마이페이지 또는 홈페이지가 나오며, 오른쪽 위에 로그인이 되어 사용자의 닉네임이 시스템의 특정부분에 표시된다. |
| 처리 | DB에서 이메일 또는 아이디를 찾고, 만약 있다면 비밀번호를 매칭한다. 만약 아이디 또는 이메일이 없다면 잘못되었다고 표기한다. 비밀번호가 매칭이 되면 로그인이 되고, 아니면 비밀번호가 틀리다고 표시한다. |
| 조건 | 로그인은 연속으로 5회 실패 시, 계정이 잠기게 된다. |

**[table 2] Sign in requirements**

### **6.2.3 Search**

|  |  |
| --- | --- |
| 이름 | 검색 |
| 기능 | 검색하는 단어를 기반으로 단어가 포함된 게시글을 찾는다. |
| 설명 | 사용자는 로그인 후에 수많은 게시글을 보게 된다. 자신이 원하는 특정 물품을 찾기 위해서는 검색기능이 필수이다. 사용자가 검색기능을 편리하게 이용할 수 있도록 한다. |
| 입력 | 사용자가 원하는 물품을 찾기 위한 단어를 받는다. 또한 필터링을 하기 위한 값을 받고, 이를 조합하여 기준으로 게시글을 특정한다. 또한 정렬을 위한 기준도 선택할 수 있다. |
| 출력 | 입력된 값을 기준으로 특정된 게시글을 정렬 값에 맞게 정렬하여 화면에 나열한다. |
| 처리 | 사용자가 입력한 단어가 들어간 게시글을 먼저 DB에서 찾는다. 그 뒤 필터링하여 검색된 게시글을 2차로 처리한다. 검색이 완료된 게시글의 타임 스탬프와 판매 상태 등을 체크하여 사용자의 화면에 나열한다. |
| 조건 | 검색하기위란 단어는 최소 2글자이상 이어야 한다. |

**[table 3] Search requirements**

### **6.2.4 My page**

|  |  |
| --- | --- |
| 이름 | My page |
| 기능 | 사용자의 정보 및 게시글을 한 화면에 표시하여 관리할 수 있게 한다. |
| 설명 | 사용자의 정보 수정할 수 있게 하는 기능이다. 또한 사용자의 게시물도 한눈에 볼 수 있도록 ‘내가 쓴 글’기능을 지원하여, 사용자의 게시물을 통합 관리할 수 있도록 한다. |
| 입력 | 사용자의 정보 수정 여부 및 수정 정보, 또는 게시글을 선택한다. |
| 출력 | 사용자의 정보를 수정했다면, 수정된 정보를 반영하여 다시 마이페이지를 띄우고, 게시글을 선택했다면 사용자의 게시글로 이동하여 추후에 있을 행동을 준비해 준다. |
| 처리 | 사용자 정보 수정전에 비밀번호를 다시 묻고, 맞다면 DB에서 사용자의 정보를 수정한다. 게시글 선택시에는 사용자의 게시글을 화면에 띄우도록 한다. |
| 조건 | 사용자 정보 수정 시 비밀번호가 틀리면 수정을 할 수 없다. |

**[table 4] Mypage requirements**

### **6.2.5 Post**

#### A. AR Measurement

|  |  |
| --- | --- |
| 이름 | AR 측정 |
| 기능 | AR 기술을 이용하여 물품의 크기를 측정할 수 있는 기능이다. |
| 설명 | 판매자는 물품의 크기를 게시글에 게재하도록 되어있는데, 모든 판매자가 물품의 사이즈를 수치로 알고 있는 경우는 적다. AR측정 기능을 도입하여, 곧바로 제품의 크기(높이, 너비, 길이)를 측정할 수 있도록 한다. |
| 입력 | 사용자는 스마트폰을 이용하여 길이를 측정한다. 카메라로 기준위치를 잡고, 그 뒤 도달할 위치를 잡으면 길이 측정이 된다. |
| 출력 | 길이, 너비, 높이 3가지의 값이 게시글에 표시된다. |
| 처리 | AR 기능이 실시간으로 길이를 측정하고, DB에 전달한다. |
| 조건 | 스마트폰이 AR기능을 지원해야한다. |

**[table 5] AR Measurement requirements**

#### B. Upload photo

|  |  |
| --- | --- |
| 이름 | 사진 올리기 |
| 기능 | 사진을 게시글에 올릴 수 있다. |
| 설명 | 제품의 사진을 게시글에 올릴 수 있게 하여, 제품에 대한 정보를 효과적으로 전달할 수 있도록 한다. |
| 입력 | 사용자가 올리고 싶은 그림을 선택한다. |
| 출력 | 사용자가 선택한 사진이 포함되어 게시글이 나타난다. |
| 처리 | 게시글이 포함된 DB에 사진도 같이 포함되어 저장된다. |
| 조건 | 용량 제한 |

**[table 6] Upload photo requirements**

#### C. Provide form

|  |  |
| --- | --- |
| 이름 | 양식 제공 |
| 기능 | 사용자가 게시글을 작성할 때, 게시글에 양식이 제공되고, 사용자는 그에 맞추러 글을 쓴다. |
| 설명 | 사용자가 물품을 판매하기 위해 게시글을 쓸 때, 필수적으로 써야 할 정보를 제공한다. |
| 입력 | 사용자는 제공되는 양식에 맞추어 글을 쓴다. |
| 출력 | 양식에 맞게 작성이 완료된 게시글을 화면에 표시한다. |
| 처리 | 양식에 맞게 작성된 글은 게시글이 있는 DB에 통합되어 저장되고 관리된다. |
| 조건 | - |

**[table 7] Provide form requirements**

### **6.2.6 View product**

#### A. AR Arrangement

|  |  |
| --- | --- |
| 이름 | AR 배치 |
| 기능 | 구매자가 스마트폰 AR을 사용하여 원하는 공간에 직접 구매하고자 하는 제품을 배치해볼 수 있다. |
| 설명 | 구매자가 제품의 정보를 알아본다고 하더라도 실제 크기는 직접보지 않는 이상 감이 잡히지 않는다. 그렇기 때문에 실제 있는 공간에 제품의 크기만큼의 부피를 표현해 주는 기능이다. |
| 입력 | 구매자의 현재 공간을 AR 기술을 사용하여 인식한다. 또한 자동적으로 판매자가 입력한 제품의 크기가 입력된다. |
| 출력 | 구매자가 원하는 공간에 배치가 완료 되어있는 직육면체의 화면을 AR을 통해 사용자가 확인할 수 있다. |
| 처리 | 사용자의 스마트폰에서 AR을 처리한다. 주변 환경을 읽고 그 위에 판매자의 게시글로부터 가져온 제품의 크기를 이용하여 직육면체의 가상 물체를 배치한다. 그 결과를 실시간으로 사용자의 스마트 폰에 표시한다. |
| 조건 | 사용자의 스마트폰이 AR을 지원해야함. |

**[table 8] AR Arrangement requirements**

#### B. Secret Comments

|  |  |
| --- | --- |
| 이름 | 비밀 댓글 |
| 기능 | 댓글을 쓴 사용자와 글쓴이만이 확인할 수 있게 한다. |
| 설명 | 구매자는 판매자에게 민감한질문을 해야 하는 경우가 있다. 이때 구매자가 간편하게 질문을 하기 위한 비밀 댓글 기능이다. |
| 입력 | 구매자는 일반 댓글과 같이 댓글을 단다. 이때 ‘비밀 댓글’박스를 체크하면 비밀 댓글 활성화가 된다. |
| 출력 | 댓글의 글쓴이와 게시판 글쓴이는 댓글과 다르지 않게 보인다. 위에 ‘비밀 댓글입니다.’라는 문구가 보인다. 다른 사용자에게는 자물쇠 표시가 보이며, 댓글 내용은 보이지 않는다. |
| 처리 | 댓글을 DB에 넣는 과정을 할 때, 먼저 비밀 댓글인지를 체크한다. 비밀 댓글이면 비밀 댓글이라는 표시를 한다. 사용자가 게시판의 글을 읽으려고 접속 시, 댓글이 비밀 댓글인지를 확인하여 댓글을 표시한다. 비밀 댓글이 아닌 경우 정상적으로 표시한다. |
| 조건 | - |

**[table 9] Secret comments requirements**

## **6.3. Non-functional Requirements**

### **6.3.1 Performance Requirements**

Everything must be done with purpose and efficiency to ensure that users do not feel frustrated when dealing with the app. Unnecessary loops must be avoided checking with data base. The app should be efficient and fast, allowing for easy and simple interaction between customer and seller. Signing up and singing in must be done efficiently. Though the increase of the number of users may increase the amount of time for verification, organized and efficient searches must be done in order to ensure that users do not feel frustrated with the sign up and log in process. As for specific requirements:

A maximum of five (5) seconds for sign up

A maximum of three (3) seconds for sign in

A maximum of three (3) seconds for searching for products

A maximum of seven (7) seconds to receive details of products

A maximum of five (5) seconds for opening the camera to be ready for AR operations to be done by the user

A maximum of sixty (60) seconds delay for database issues during working hours, and a maximum of three hundred (300) seconds during non-working hours

These maximum seconds for delay are implemented in order to ensure an efficient working system and the best experience for our customers.

### **6.3.2 Security Requirements**

Security and privacy are an integral part of the system. Pictures or profiles or other user information that is displayed may contain sensitive materials that users were unaware of. Security is also important for the information that is entered during the sign-up process. There is bound to be personal information during this stage, as they enter information of themselves. Because of this, adequate precautions must be taken. Unauthorized access to the system must not be allowed. That is, there must be an accurate and dependable way to check user. As such, external communication between system and the server should done on a **secure network**. Data is to be stored on google’s **firebase server**. Hacking is highly unlikely. Though firebase data does not provide encryption, encrypted data is unnecessary as google takes many precautions for hacking. Furthermore, in order to prevent user data loss, there must be relevant and up to date back up of data. This may be even more particularly important in events in which user information is exposed. Firebase allows auto backup for data, so this must be enabled.

### **6.3.3 Software Quality Attributes**

As the largest and important software that is to be implemented is the integration of augmented reality, the software must focus on the *portability and flexibility, usability and robustness, maintainability*, and perhaps most importantly, the *correctness* of the system.

First, the *portability and flexibility* of the system. The app must allow augmented reality across the maximum number of devices that supports augmented reality. The importance of this is apparent in the fact that app must allow the maximum number of customers to use our app. As the minimum requirement for the support of google’s ARcore is **android 7.0**, all devices that is capable of installing the software should be allowed to download our app.

For *usability and robustness*, the app must be easy to use. Augmented reality may be a new concept for many users, and thus, a way to introduce users to the usability should be implemented. For users that access the app for the first time, there should be a **manual** or a help option to allow users to more easily understand the functions of the system.

For *maintainability*, the code should be easy to understand to allow developers to maintain the system as a whole. **Refactoring** should be done frequently, and the lead developers must not be changed during the development of the system.

For *correctness,* the measurements must be close to being accurate to allow sellers to accurately register the size of the product and allow buyers to accurately see the size of the product. The measurements should be no more than a **percent error of 10%** to allow users to view their product as accurately as possible. As the measurements are dependent on the user holding their camera, however, the percent error is fairly generous.

### **6.3.4 Business Rules**

The people involved are users and admins. Because of this, roles must also be assigned. A user can post their product to be viewed by other users, or they can view and purchase products. Based on their history of selling and purchasing products, users may increase their credit. There is no credit system per say, yet simply having and showing that a user has successfully sold an object gives assurance to the user that is buying another product from the same person. This is particularly important in the purchase of second-hand materials, as there may be problems with scams. In order to combat this, users are able to look at products that have already been sold, which in turn allows users to see that a seller has successfully sold a product without scams. Furthermore, in order to allow contact between users, contact information of the seller is given to the buyer if they show potential interest. Due to this, users are given the authority to post or to view products and write secret comments.

For the admins, as admins should make sure that there is no inappropriate information being uploaded to the database, including pictures as well as information that is being entered by the user, admins should be given the authority to delete any information that they deem necessary – even to the point of deleting user data. This can be achieved by allowing the admins to **access the firebase project**. If the admins see information that is inappropriate, they are able to simply delete the data from the firebase – even allowing the deleting of users, effectively deleting them from the app.

## **6.4. Scenario**

### **6.4.1 LogIn**

#### A. Normal Flow

If new customer visit our application first time, he would be get into sign in Page. If he write down proper information in the blank, his personal information would be stored in the firebase database server. And if he already signed in, he can log in from the login page and get into the market or his personal pages.

#### B. What can go wrong

If he write down wrong information when he sign in or log in, error msg would be pop up to his screen and make him to wirte down right information.

### **6.4.2 Buyer’s state**

#### A. normal Flow of buying

When he get into the market page, he could choose the category from the listView of the market, and he can specify the product that he is looking for. In the category that he choose to look for, he can see the list of the products that he want to buy. If he choose one of the product from the list, new page of the product pop up and show the information of the product. And we want to see the size of the product by AR system, we click the button of the “AR mode” from the product page, application get the camera from his phone and he can see the camera scene and the model of the product that have the same size of the product.

#### B. What can go wrong

When he get into the category of his wants, there would be no products that seller posted in these category. If so, pop up msg of “There is no product you want” and show some alternative products that similar with he wants.

### **6.3.3 Seller’s state**

#### A. normal Flow of selling

If he want to sell the product, he can show the product to others by just write the post on the category of the product. When he write the post. He write down the title, picture, performance of the product, and main text. Just same as other secondhand application. But when he write down the size of the product. He doesn’t need to measure of the product by himself. He can measure the size by take some pictures, would be a video, and our “AR measurement system” get the size of the product automatically.

#### B. What can go wrong

“AR System” would have some mistake to measure the size. Then he or the system would reset the “AR Mode” and restart the measuring the size.

### **6.3.3 dealing state**

#### A. normal Flow of dealing

If buyer find the product that he want to buy, he can post the ‘secret comment’ in the post section, and other buyers can’t look the comment that he wrote, and just the buyer and the seller of the product can see the comment and they can give and have the information for dealing the product.

#### B. What can go wrong

Because of personal reason, the deal would be cancelled. If so, system erase the comment and make the state of the post to “on sale” again.

#### C. Final state

If the deal end, the state of the post would be changed to “Sold out”. And If there is no problem in the dealing, dealer and seller get one history record that could be the one of the most powerful evidence when make the other deals in the future.

# **7. System Models**

## **7.1. Objective**

System Models에서는 각 시스템들의 관계와 구성요소를 diagram과 table로 표현하여 나타낸다.

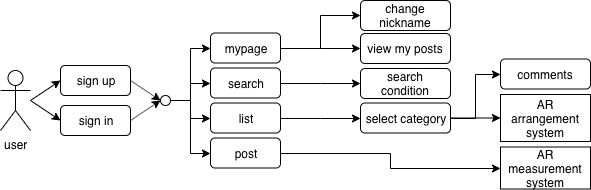
## **7.2. Context Models**

### **A. Context Diagram**



**[diagram 6] context diagram**

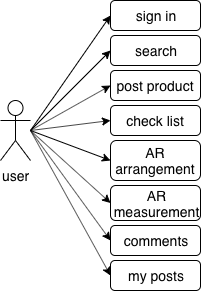
### **B. Process Diagram**



**[diagram 7] overall process diagram**

## **7.3. Interaction Models**

### **A. Use-Case Diagram**



**[diagram 8] use-case diagram**

### **B. Tabular Description for each Use-Case**

**B.1 SIGN IN**

|  |  |
| --- | --- |
| **USE CASE** | **Sign in** |
| **ACTOR** | User, User DB |
| **DESCRIPTION** | 사용자가 어플리케이션을 사용하기 위해 로그인을 한다. 로그인 할 때 입력한 정보와 데이터베이스 안에 저장된 사용자의 정보가 일치하는지 판단한다. |
| **TRIGGER** | 사용자가 ID와 PW을 입력하고 로그인 버튼을 누른다. |
| **RESPONSE** | 입력한 ID와 PW 쌍이 User DB에 존재하는지 확인하고 존재한다면 해당 회원의 접속을 허락한다. |
| **COMMENTS** |  |

**[table 10] Login use-case**

**B.2 SEARCH**

|  |  |
| --- | --- |
| **USE CASE** | **Search** |
| **ACTOR** | User, Product DB |
| **DESCRIPTION** | 사용자가 보고싶은 게시글이 특정되어 있을 때 검색 조건(게시자, 게시글 제목/내용)을 입력하면 해당 조건을 만족하는 게시글이 리턴된다. |
| **TRIGGER** | 사용자가 검색 조건을 입력하고 검색하기 버튼을 누른다. |
| **RESPONSE** | 입력한 검색 조건을 쿼리문을 통해 만족하는 제품 목록을 Product DB에서 검색하고 해당 목록을 리턴한다. |
| **COMMENTS** |  |

**[table 11] Search use-case**

**B.3 POST PRODUCT**

|  |  |
| --- | --- |
| **USE CASE** | **Post Product** |
| **ACTOR** | User, User DB, Product DB |
| **DESCRIPTION** | 사용자가 중고 판매를 할 제품을 등록한다. 제품의 정보를 입력하면 해당 제품의 정보와 게시자(판매자)의 정보가 Product DB에 저장된다. |
| **TRIGGER** | 사용자가 제품의 정보를 입력하고 업로드 버튼을 누른다. |
| **RESPONSE** | 입력한 제품의 상세 정보가 Product DB에 각각의 entity에 해당하는 곳에 저장된다. 이때 게시자(판매자)의 정보도 저장되는데 foreign key로 User DB에서 빌려온다. |
| **COMMENTS** | 해당 양식을 만족하지 못하면 다시 입력하라는 경고문을 띄운다. |

**[table 12] Post Product use-case**

**B.4 CHECK LIST**

|  |  |
| --- | --- |
| **USE CASE** | **Check List** |
| **ACTOR** | User, Product DB |
| **DESCRIPTION** | 사용자가 제품 목록을 확인하고 목록을 카테고리별, 최신순, 판매중, 가격순으로 정렬할 수 있다. |
| **TRIGGER** | 사용자가 원하는 정렬 조건으로 맞춘 후 정렬 버튼을 누른다. |
| **RESPONSE** | 해당 정렬 조건에 맞추어 Product DB에 쿼리문을 보내 제품 목록을 다시 가져온다. |
| **COMMENTS** |  |

**[table 13] Check List use-case**

**B.5 AR ARRANGEMENT**

|  |  |
| --- | --- |
| **USE CASE** | **AR Arrangement** |
| **ACTOR** | User, AR Arrangement system, Product DB |
| **DESCRIPTION** | AR 배치 시스템이 사용자가 구매하고 싶은 제품의 크기 정보를 Product DB에서 가져와 사용자의 화면에 배치해준다. |
| **TRIGGER** | 사용자가 제품 배치 버튼을 누르면 실행된다. |
| **RESPONSE** | Product DB에서 가져온 제품의 크기 정보를 이용해 AR 배치 시스템이 직육면체 모양의 AR object를 사용자 화면에 띄운다. |
| **COMMENTS** |  |

**[table 14] AR Arrangement use-case**

**B.6 AR MEASUREMENT**

|  |  |
| --- | --- |
| **USE CASE** | **AR Measurement** |
| **ACTOR** | User, AR Measurement System, Product DB |
| **DESCRIPTION** | 사용자가 AR 측정 시스템을 통해 판매하고자 하는 제품의 크기 정보를 측정하고 해당 크기 정보를 Product DB에 저장한다. |
| **TRIGGER** | 사용자가 AR 측정 버튼을 클릭하여 AR 측정 시스템을 실행한다. |
| **RESPONSE** | AR 측정 시스템이 제품의 길이, 너비, 높이를 측정하여 Product DB에 전달한다. |
| **COMMENTS** |  |

**[table 15] AR Measurement use-case**

**B.7 COMMENTS**

|  |  |
| --- | --- |
| **USE CASE** | **Comments** |
| **ACTOR** | User, Comments DB, Product DB, User DB |
| **DESCRIPTION** | 사용자가 구매하고자 하는 제품 게시글에 댓글을 남겨 판매자와 연락을 취하도록 한다. 원한다면 비밀 댓글 기능을 통해 댓글 작성자와 판매자만 볼 수 있도록 한다. |
| **TRIGGER** | 사용자가 댓글을 작성하여 작성 버튼을 누른다. 비밀 댓글을 원하면 ‘비밀 댓글’ 박스를 체크하여 활성화 한다. |
| **RESPONSE** | 입력한 댓글 내용을 Comments DB에 저장하고 이때 어떤 제품에 대한 댓글인지 Product DB에서 foreign key로 가져와 같이 저장한다. 또 누가 작성했는지 역시 User DB에서 foreign key로 가져와 같이 저장한다. 비밀 댓글 여부 역시 저장하여 비밀 댓글 기능을 제공한다. |
| **COMMENTS** | 사용자가 댓글을 남기는 순간 판매자에게 PUSH 알림이 가도록 한다. |

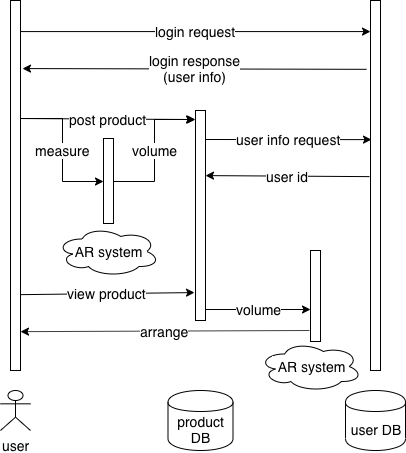
**[table 16] Comments use-case**

**B.8 MY POSTS**

|  |  |
| --- | --- |
| **USE CASE** | **My Posts** |
| **ACTOR** | User, Product DB |
| **DESCRIPTION** | 사용자가 자신의 글 목록을 확인할 수 있다. |
| **TRIGGER** | 사용자가 글 목록 메뉴를 클릭하면 자신의 글 목록이 띄워진다. |
| **RESPONSE** | Product DB에서 사용자의 id와 게시자 entity의 id가 같은 게시글의 목록을 쿼리문을 통해 가져온다. |
| **COMMENTS** | 자신의 글을 삭제하거나 수정할 수 있다. |

**[table 17] My Posts use-case**

### **C. Sequence Diagram**



**[diagram 9] overall sequence diagram**

# **8. System Evolution**

## **8.1. Objective**

System Evolution에서는 hardware의 변화나, user needs의 변화로 생길 변경사항들에 대해서 기술한다. 또한 이 소프트웨어에서 예상 변화에 맞춰서 발전시킬 방향성에 대해서 기술한다.

## **8.2. Convenient Product Selection by Sellers**

판매자가 제품을 올릴 시에, 카메라를 통하여 자신이 팔고자 하는 제품을 핸드폰에 위치시키면, AR을 통해 화면에 나타나는 제품들을 구체화 하고 그 구체화 된 물건을 클릭함으로써 좀 더 편리하게 제품을 판매하는 글을 올리기 쉽게 구현하고자 한다.

## **8.3. Increased Seller’s Confidence**

온라인 중고 거래의 문제점으로 항상 거두 되는 사기 문제를 낮추기 위하여, 판매자로 등록할 시에 본인 신원 확인 과정을 확실히 함으로써 신뢰성을 높인다. 또한 제대로 완료된 제품 거래가 쌓일수록, 그 판매자나 구매자에 대한 신뢰도가 늘어남으로써, 새로운 거래를 할 시에 서로에게 많은 정보를 제공할 수 있도록 하고자 한다.

## **8.4. Addition of Categories**

서비스를 하는 항목을 IT 계열의 전자제품에서 점차 늘려나가서 고객의 접근성을 늘리고 다양성을 높이고자 한다.

# **9. INDEX**

## **9.1. Tables**

[Table 1: Glossary 16](#_Toc9286211)

[Table 2: Sign up Requirements 2](#_Toc9286212)4

[Table 3: Sign in Requirements 2](#_Toc9286213)4

[Table 4: Search Requirements 2](#_Toc9286214)5

[Table 5: My page Requirements 26](#_Toc9286215)

[Table 6: AR Measurement Requirements 26](#_Toc9286216)

[Table 7: Upload photo Requirements 27](#_Toc9286217)

[Table 8: Provide form Requirements 27](#_Toc9286218)

[Table 9: AR Arrangement Requirements 28](#_Toc9286219)

[Table 10: Secret comments Requirements 28](#_Toc9286220)

[Table 11: Login use-case 34](#_Toc9286221)

[Table 12: Search use-case 34](#_Toc9286219)

[Table 13: Post Product use-case 35](#_Toc9286220)

[Table 14: Check List use-case 35](#_Toc9286221)

[Table 15: AR Arrangements use-case 36](#_Toc9286220)

[Table 16: AR Measurement use case 36](#_Toc9286221)

[Table 17: Comments use-case 36](#_Toc9286219)

[Table 18: My posts use-pase 37](#_Toc9286220)

## **9.2. Figures**

[Figure 1: 온라인 거래 경험 추이 10](#_Toc9286222)

[Figure 2: 온라인 중고 거래 신뢰도 현황 1](#_Toc9286223)0

[Figure 3: 온라인 중고 거래 신뢰도 주요 원인 1](#_Toc9286224)1

## **9.3. Diagrams**

[Diagram 1: overall flow chart 20](#_Toc9286231)

[Diagram 2: post system flow chart 21](#_Toc9286232)

[Diagram 3: list system flow chart 22](#_Toc9286233)

[Diagram 4: AR measurement system flow chart 22](#_Toc9286234)

[Diagram 5: AR arragement system flow chart 23](#_Toc9286235)

[Diagram 6: context diagram 33](#_Toc9286236)

[Diagram 7: overall process diagram 33](#_Toc9286237)

[Diagram 8: Use case diagram 34](#_Toc9286238)