



당신만의
가장 빠른
중고사냥꾼

2020 1학기 소프트웨어 공학
1조 고준서 이은호 이진권 전제영 황유나

Requirements Specification

Contents

<1> Preface -----	4
1.1 Readership -----	4
1.2 Version history -----	4
<2> Introduction -----	5
2.1 Needs & Background -----	5
2.2 Glossary -----	7
2.3 System overview -----	10
2.4 Expected effects -----	11
<3> User scenario -----	13
3.1 Buyer -----	13
3.2 Seller -----	19
3.3 Administrator -----	22

<4> Requirements -----	24
4.1 Functional Requirements -----	24
4.2 Non-functional Requirements -----	41
<5> System Architecture -----	43
<6> System requirement specification -----	48
<7> System models -----	57
<8> System evolution -----	58
<9> Appendices -----	59
9.1 Hardware & software requirements	
9.2 DB requirements	
<10> Index -----	61

Preface

해당 문서의 예상 독자, version history 를 다룬다.

>> Readership

- User Scenario Readership

사용자를 예상 독자로 상정한다. 일반적인 사용자가 따르게 될 시스템의 사용 시나리오를 자연어 중심으로 표현하여, 사용자가 보다 쉽게 이해하고, 요구사항을 구체화할 수 있도록 돕는다.

- System Requirements & System Architecture Readership

도표 중심의 구체적인 자료로, 개발자와 다양한 Stakeholder 가 시스템 전반에 대한 상세한 지식과 작동 원리를 알 수 있도록 보조한다. 이후 고객과의 계약에도 참고 자료로 사용될 수 있도록 명확하게 작동한다.

>> Version history

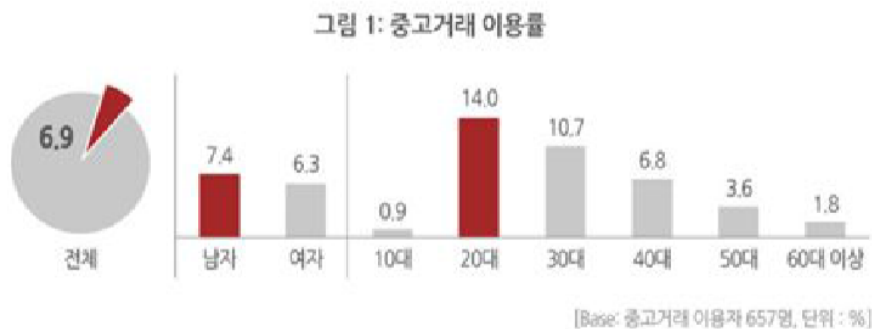
Revised on	Version	Description	Modified by
2020-05-13	1.0	첫번째 버전입니다.	1 조 전원

Introduction

해당 시스템의 개발 배경 및 needs 를 밝히고, 작동 방식 및 원리를 간단하게 나타낸다.

>> Needs & Background

중고 거래 시장은 국내외에서 빠르게 성장하고 있으며, 특히 스마트폰 platform 을 중심으로 발전하고 있다. 그럼에도 불구하고, 사용자층의 확대는 상대적으로 더디게 이루어지고 있는데, 이는 중고 거래 특유의 낮은 신뢰도와 번거로움 때문으로 보인다.



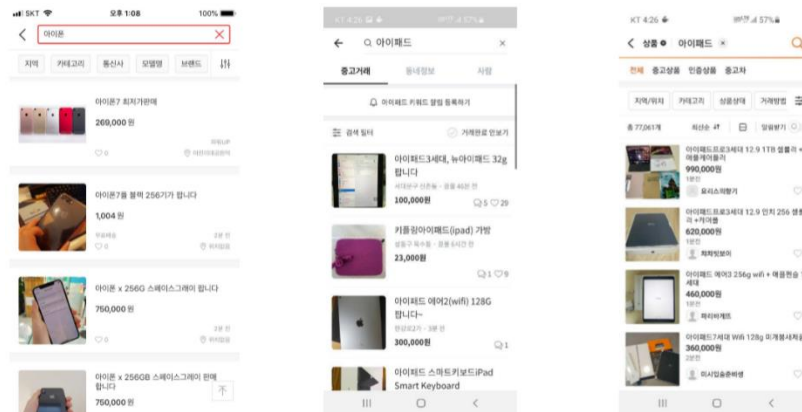
[Figure 1 : 2019 중고거래 이용자 현황]

결과적으로 20 대 중심의 사용자가 형성되고 있으며 그조차도 14% 정도의 높지 않은 비율을 보이고 있다. 상대적으로 높은 연령대에서는 중고 거래 자체를 부정적으로 인식하는 경우가 많기 때문에, 오히려 20 대 내에서 중고거래 시스템에 대한 진입장벽을 낮추는 것을 목표로 한다.



[Image 1 : 기존 3 대 중고거래 어플의 최신 전략]

기존 업계에서는 '검색 -> 개별 연락 -> 입금 및 거래' 로 이루어지는 기본적인 사용자 경험의 틀은 바꾸지 않은 채 입금 과정에 시스템이 개입해서 사기를 예방하는 것에 집중하고 있는 상황이다. 이는 결과적으로 사기 범죄의 위험성은 낮출 수 있지만, 중고 거래 플랫폼의 사용을 더욱 복잡하게 만든다. 근본적으로, 중고 거래 자체의 인터페이스 자체를 변화시켜 User 에게 보다 간편하고 새롭게 느껴지도록 하는 시도가 필요하다.



[Image 2 : 기존 3 대 중고거래 어플의 UI 비교]

시장의 기존 어플들의 UI 를 보면 거의 구분이 되지 않으며, 차별성이나 개성도 거의 존재하지 않는다. 결국 중고 물품을 원하는 사용자는 여러 사이트를 돌아다니며 실시간으로 매물을 확인해야 하며, 원하는 매물을 찾다고 해도 판매자와 개별적인 연락을 하며 세부적인 거래 조건을 확인해야 하고, 도중에 경쟁으로 인해 구매 절차가 취소되는 경우도 많다. 이와 같은 불편함을 해소시켜줄 새로운 UX 에 대한 시장의 Needs 가 충분하다고 생각된다.

>> Glossary

Term	Description
로그인	개인정보에 맞는 아이디와 비밀번호를 검색하면 접속할 수 있도록 하는 기능이다.
회원가입	사이트에 가입하도록 하며 개인정보를 수집하는 단계이다.
상품업로드	중고거래이기 때문에 상품의 업로드 기능을 제공해준다. 세밀하게 상품을 업로드 하도록 한다. 업로드 하며 상품의 삭딜에 있을 시간을 설정한다. 상품을 올릴 때, 해당 조건이면 몇명의 사람이 해당 상품을 구매할지 정보를 보여준다.
상품	상품은 다음과 같은 정보를 포함한다. 판매 요청 양식 ID, 상품 카테고리 ID, 상품 상태, 삭딜 구매자 목록, 거래를 거절/취소한 구매자 ID 목록, 거래를 승낙/거절한 이유와 이횃수 목록
판매 요청 양식 위시리스트 (구매 요청 양식)	<p>위시리스트(플랫폼 내 시스템에서는 '구매 요청 양식'이라는 용어로 통용)와 판매 요청 양식은 회원이 구매/판매를 위해 작성하는 양식이며, 다음과 같은 정보가 공통적으로 포함되어있다. 판매(구매) 요청 양식 ID 상품 카테고리 ID, 회원 ID, 가격, 상품 세부 정보. 타임스탬프 등</p> <p>위시리스트에는 포함되지 않고, 판매 요청 양식에만 포함되어있는 정보들도 있으며, 다음과 같다. 품목 사진, 거래 세부 정보.</p>
상품 상태	<p>각 상품은 등록된 후 다음과 같은 여러가지 상태를 거치게된다.</p> <p>- 삭딜 구매자 대기 상태 - 1: 처음 상품이 등록되거나, 이전 삭딜 구매자 목록에 구매자가 더는 남지 않아 새로운 구매자 목록을 기다리는 상태</p>

- 삭딜 구매자 대기 상태 - 2: 현재 구매자가 없어 새로운 구매자를 기다리는 상태
- 삭딜 구매자 목록 생성 상태: 서열화 모듈이 구매자 목록을 생성하는 상태
- 삭딜 팝업 상태: 현재 삭딜 구매자 목록의 최상단에 위치한 구매자에게 삭딜 팝업을 표시한 상태
- 일반딜 상태: 삭딜 지속 시간이 끝나, 상품의 일반딜 검색이 가능해진 상태
- 등록 보류 상태: 문제가 있다고 판단되어 상품의 등록이 보류된 상태
- 거래 중: 판매자와 구매자가 거래중인 상태, 거래중인 구매자를 제외한 다른 구매자에게 이 상품이 보이지 않아야 한다.
- 거래 완료: 거래가 완료된 상태

카테고리 상품을 분류하는 기준, 카테고리는 여러 층위로 이루어져 있으며, 하위 카테고리는 각각 상위 카테고리의 '카테고리 ID' 아래에 저장된다.

카테고리의 가장 마지막 층위에, 개별 상품으로 식별 가능한 정도로 세분화된 상품이 있다. 상품에는 각각 ID가 있어 식별 가능하며, 각 '상품 ID'는 색상, 사이즈 등의 부차적인 정보를 담은 여러 '옵션 ID'를 가진다. 각 옵션 ID는 해당 상품을 대표하는 썸네일 이미지와, 이를 호출할 수 있는 '썸네일 이미지 ID'를 가진다.

삭딜 등록된 상품은 일정 시간동안 삭딜 상태를 유지한다. 삭딜 지속 시간동안 판매되지 않은 상품은 일반딜로 넘어간다.

삭딜 구매자 목록 삭딜 팝업을 표시할 구매자의 목록이다.

상품이 등록되거나 이전 삭딜 구매자 목록에 더는 구매자가 남아있지 않을 때, 상품은 '삭딜 구매자 대기 상태 - 1'에 돌입해 서열화 모듈에 '삭딜 구매자 목록'을 요청한다.

삭딜 지속 시간 삭딜이 얼마나 지속될 지를 나타내는 단어이다.

'사전에 설정된 삭딜 지속 시간'을 '삭딜 구매자 목록'의 구매자 수로 나누어 각 구매자가 $1/n$ 만큼의 시간을 할당받아, 그 시간만큼 삭딜 팝업을 확인할 수 있게 한다.

만약 삭딜 지속시간이 남았는데, 삭딜 구매자 목록에 구매자가 남아있지 않으면, 상품이 서열화 모듈에 요청을 전송해 새로운 삭딜 구매자 목록을 생성한다.

서열화 모듈 서열화 모듈은 판매 상품의 카테고리 ID와 가격, 상품 세부 정보, 거래 세부 정보 등을 전송받아, 해당 상품의 카테고리 ID와 같은 값을 가진 구매 요청 양식들을 모아 가격, 지역, 상품 세부 정보 등의 조건으로 필터링한 뒤, 각 구매 요청을 거래가 이루어질 확률이 높은 순으로 배치하고, 이 중 상위 X % 만큼만 목록에 넣은 뒤, 무작위로 순서를 할당해 삭딜 구매자 목록을 생성한다.

만약 적절한 구매자를 찾을 수 없다면, '삭딜 구매자 대기 상태 - 2'에 돌입해, 삭딜 지속 시간이 모두 지날 때까지 구매자를 기다리게 된다.

삭딜 팝업 삭딜의 정보를 단말기를 통해 구매자에게 전하는 단계와 그 방식의 이름이다.

삭딜 팝업은 사진, 가격, 상품 세부 정보, 거래 세부 정보, 신고 버튼을 포함하고 있으며, 구매자는 각 세부 정보를 터치해 마음에 든/들지 않은 요소를 표시할 수 있으며, 좌/우로 스와이핑해 거래를 승낙하거나 거절할 수 있다.

일반딜	일반적인 중고장터와 동일한 방식의 검색-거래 방식이다.
채팅	구매자와 판매자가 연결되면 둘의 아이디를 통해 채팅을 할 수 있도록 한다.
크롤러	다른 중고장터 사이트의 상품들의 가격이나 조건들을 크롤링하여 긁어온다는 뜻이다.
Database	백엔드에 위치하여 모든 로그와 정보를 기록해두는 저장소이다.
레포트	신고한 내용에 대한 유저가 매니저에게 보내는 내용인 동시에 매니저가 처리 내역에 대해 유저에게 보내는 내용이다.

>> System Overview

 <p>기존 사용자도 쉽게 적용할 수 있는 직관성</p>	 <p>기존 소개팅 어플의 UX 를 일부 차용하는 신선함</p>	 <p>가격 비교 사이트의 알고 리즘을 활용한 간편함</p>
--	--	--

중고 거래 플랫폼 '삭'은 기존 어플들의 사용성과 유사한 면도 많지만, '스 와이핑'과 '가격 비교' 두 가지 기능을 추가하여 보다 간편하면서도 신선한 UX를 제공한다는 특징을 가지고 있다.



[Image 3 : 시스템 UI 예시]

첫째 : 물품의 등록 및 검색 방식을 카테고리 위주로 변환하여 일관된 format 으로 물품에 관련한 정보가 저장되고 공유 및 비교될 수 있도록 한다.

둘째 : 다른 중고 거래 사이트의 매물에 대한 정보를 크롤링하고, 간단한 키워드 위주의 NLP 를 거쳐 일관된 형식의 데이터로 변환한 뒤 DB 에 저장한다. 이렇게 정리된 정보는 사용자의 물품 구매 및 판매에 도움이 될 수 있도록 활용한다.

셋째 : 공통된 format 에 맞게 등록한 구매자와 판매자의 거래 조건을 매칭하여, 이를 기반으로 '스 와이핑을 통한 간편한 거래' 인터페이스를 구축한다.

>> Expected Effects

1. 소비자의 진입 장벽 낮추기

간편한 가격 비교 및 개별 연락의 요소를 최소화한 거래 과정을 통해 중고 거래에 대한 거부감을 해소하고, 궁극적으로 진입장벽을 낮춘다. 이는 중고 거래 사용자의 주축이 되는 20 대를 겨냥한 시스템으로, 특히 모바일 플랫폼에 특화되어 있기에 현재의 사용 트렌드에 부합한다고 볼 수 있다.

2. 다른 중고 거래 시스템과의 차별 요소 확보

거의 구분이 되지 않는 기존 어플들의 UX/UI 에서 탈피하여 스와이핑에 기반한 새롭고 신선한 거래 생태계 구축이 가능하다. 동시에 기존 어플들의 익숙함은 어느 정도 계승하여 사용자들이 쉽게 적응할 수 있도록 배려한다. 결과적으로 사용자 수가 곧 경쟁력으로 직결되는 중고 거래 플랫폼 특성상, 기존 사용자를 유지한 채 새로운 사용자층을 확대하는 것이 business object 라고 할 수 있다.

Scenario

시나리오를 통해 사용자의 시스템 사용 경험을 유추하고 세분화하여, 요구 사항을 구체화하고, 이를 위해 구현해야 하는 function 이 무엇인지 정리할 수 있다.

>> 구매자 <<

>> Normal Deal

Actor	<ul style="list-style-type: none">일반사용자(판매자/구매자)시스템 매니저데이터베이스
시나리오 1	처음 접속자의 회원가입
<ol style="list-style-type: none">처음 어플에 접속하면 회원가입을 할지 로그인을 할지 선택할 수 있다.아이디가 없을 경우 회원가입을 하도록 유도한다.회원가입절차는 아이디, 비밀번호, 전화번호(인증), 거래가능지역, 약일을 받을 수 있는 시간을 설정한다.시스템은 데이터베이스에 사용자 정보를 저장한다.그 후 로그인창으로 나가지는 것이 아니라 바로 약 화면 안으로 들어간다.	
시나리오 2	구매자 로그인

1. 스마트폰 어플리케이션을 기반으로 하기 때문에, 이전에 자동로그인을 설정해놓았다면 별도의 로그인 절차 없이 접속하도록 한다.
2. 일반사용자는 자동로그인이 설정되지 않았다면 아이디와 비밀번호를 입력한다.
3. 시스템은 데이터베이스의 로그인정보와 비교하여 로그인을 가능하도록 한다.

시나리오 3

시스템 매니저 로그인

1. 시스템 매니저는 다른 별도의 어플리케이션을 사용한다.
2. 다른 어플리케이션을 통해 회사에서 부여받은 계정으로 로그인하여 시스템을 관리한다.
3. 그 후 시스템을 관리하는 과정은 시스템 관리자의 시나리오를 따른다.

시나리오 4

구매희망상품 검색하기

1. 일반, 검색창 밑을 옵션으로 선택한다.
2. 구매자는 구매희망상품의 품목명, 모델명 (이 정보는 데이터베이스에 이미 저장되어 있다) 을 클릭하여 검색창 필터링을 시행한다.
 - a. 이 경우 찾고자 하는 품목명과 모델명이 없다면 임의 입력할 수 있도록 한다.
3. 가격이나 제품의 상태와 같은 옵션(조건)들에 대해서도 버튼 클릭을 통해 검색 결과를 정교화할 수 있다.
 - a. 이 경우에도 마찬가지로 찾고자 하는 옵션들에 대해 미리 정의되어 있지 않다면 구매자가 직접 임의 입력할 수 있도록 한다.
 - b. 이때, 구매자가 필터링한 옵션들에 대해 완전 일치하는 결과가 나오지 않는 경우 구매자에게 '완전 일치하는 상품이 없습니다' 라고 에러 메시지를 띄운 뒤, 구매자에게 '유사 검색 결과를 보시겠습니까?' 라고 묻는다. 만약 '예'를 선택했다면 구매자가 입력한 키워드 개수와 최대한 많이 겹치는 키워드를 지닌 상품을 상단부터 위치시킨다.
4. 일반 밑에서는 한 화면당 세 개 품목에 대해 그 사진과 간단한 정보가 보여지는데,

<p>관심이 가는 상품에 대해 더 자세한 정보를 조회하기 위해서는 품목 사진을 클릭하면 된다.</p>	
<p>시나리오 5</p>	<p>위시리스트에 관심 상품 보관하기</p>
<ol style="list-style-type: none"> 1. 구매자는 관심 있는 상품을 위시리스트에 넣어 보관할 수 있다. 2. 구매자가 작성한 위시리스트는 '구매 요청 양식'의 형태로 내부 데이터베이스에 저장된다. 3. 구매자는 초기화면에서 해당 상품에 대한, 플랫폼 내 + 외부 사이트 크롤링 평균 거래 가격 추이를 관찰할 수 있다. 	
<p>시나리오 6</p>	<p>바로 거래하기</p>
<ol style="list-style-type: none"> 1. 구매자는 관심 있는 상품에 대해서 위시리스트에 보관하지 않고 바로 판매자에게 연락을 취해 거래를 시도할 수 있다. 2. 판매자에게 삭딜과 마찬가지로 스와이핑 메시지가 전달된다. 3. 판매자에게 제시된 구매자의 제한된 정보(희망 거래 지역 및 신뢰도 등급)이 제시된다. 4. 거래가 성사되면 삭딜과 동일하게 거래 성사 메시지와 함께 연락처 정보가 양측에 전달된다. 5. 어느 한쪽의 일방적인 거래 파기의 경우, 당사자는 레포트를 작성하여 시스템 매니저에게 세부 사항을 전달할 수 있다. 이 정보는 신고 레포트 데이터베이스에 저장됨과 동시에, 당사자 (구매자 혹은 판매자) 정보 데이터베이스에도 업데이트된다. 	

>> Shark Deal

Actor	<ul style="list-style-type: none"> • 일반사용자(구매) • 데이터베이스 • 시스템 매니저
시나리오 1	회원가입
<p>6. 처음 어플에 접속하면 회원가입을 할지 로그인을 할지 선택</p> <p>7. 아이디가 없을 경우 회원가입을 하도록 유도한다.</p> <p>8. 회원가입절차는 아이디, 비밀번호, 전화번호(인증), 거래가능지역, 약일을 받을 수 있는 시간을 설정한다.</p> <p>9. 시스템은 데이터베이스에 사용자 정보를 저장한다.</p> <p>10. 그 후 로그인창으로 나가지는 것이 아니라 바로 약 화면 안으로 들어간다.</p>	
시나리오 2	로그인
<p>4. 스마트폰 어플리케이션을 기반으로 하기 때문에, 이전에 자동로그인을 설정해놓았다면 별도의 로그인 절차 없이 접속하도록 한다.</p> <p>5. 일반사용자는 자동로그인이 설정되지 않았다면 아이디와 비밀번호를 입력한다.</p> <p>6. 시스템은 데이터베이스의 로그인정보와 비교하여 로그인을 가능하도록 한다.</p>	
시나리오 3	위시리스트 제출

7.스와이프 딜 탭을 클릭한다.삭딜을 사용하기 위해서 먼저 위시리스트를 작성하여야 한다.
기존 위시리스트가 있다면 작성하지 않고 넘어갈 수 있다.

8. 위시리스트는 구매자가 구매하길 원하는 상품에 대한 정보를 담고 있다. 많은 옵션이 있고
대부분 클릭을 통해서 위시리스트를 입력할 수 있다. 기초적인 옵션은 시스템 매니저가
구성했다.

9. 옵션은 단계별로 나누어져서 관련분야로 점점 좁혀진다.

(ex. 핸드폰 -> 핸드폰 브랜드 -> 해당 브랜드 핸드폰 기종 -> 기종 사양)

10.기존의 옵션 외에 다른 정보를 입력할시에는 해시태그를 마지막에 입력할 수 있다.

11.선택할 옵션이 해당 단계에 없을 경우 다음 옵션 단계로 넘어갈 수 있다.

12. 입력한 정보는 추후에 수정이 가능하며 스와이프 딜에서 선호하는 옵션을 클릭하면
자동으로 위시리스트 데이터 베이스 업데이트된다.

시나리오 4

알림 설정

13. 위시리스트 작성이 끝나면 알림 설정의 권한을 요구한다. 삭딜을 원활하게 하기 위해선
핸드폰 알림 설정을 해야함을 알린다.

14. 설정하지 않을 시에 위시리스트에 맞는 물건이 매칭되었을 때 빠른 시간에 답장하지
못하면 구매하지 못할 수 있음을 알린다.

15. 알림 설정을 할 시에 알림을 받지 않을 시간을 설정할 수 있다.

시나리오 5

스와이프 요청하기

1. 자신이 매칭받은 물건에 대한 정보를 알림을 통해 받는다.
2. 알림에서 자세하기 보기 옵션을 클릭한다.
3. 물건의 정보를 보고 옵션에 대한 선호를 피드백할 수 있다.

4. 이 피드백은 자신의 삭딜 매칭에 영향을 준다.
5. 원하는 물건은 구매 예약을 할 수 있고 마음에 들지 않으면 다른 매칭을 기다릴 수 있다.
6. 스와이프 요청을 하면 물건마다 따로 설정되어 있는 구매자 대기열에 들어가게 된다.

시나리오 6	삭딜 매칭
<ol style="list-style-type: none"> 7. 알림을 켜놓게 되면 매칭이 되는 순간 구매자에게 대기화면을 통해 해당 물건, 매칭 성사를 알려준다. 8. 스와이프를 취소하거나 판매자와 대화하는 화면으로 넘어갈 수 있다. 	
시나리오 7	판매자와 대화
<ol style="list-style-type: none"> 4. 매칭이 성사되면 거래에 대한 짧은 채팅을 진행한다. 5. 불성실하게 답변하면 다음 대기열의 구매자에게 매칭이 넘어갈 수 있다. 	

>> 판매자 <<

Actor	<ul style="list-style-type: none"> • 일반사용자(판매 / 구매) • 전문판매자(돈을 내고 광고하는 전문판매자) • 시스템 매니저 • 데이터베이스
시나리오 1	처음 접속자의 회원가입
<ol style="list-style-type: none"> 1. 처음 어플에 접속하면 회원가입을 할지 로그인을 할지 선택할 수 있다. 2. 아이디가 없을 경우 회원가입을 하도록 유도한다. 3. 회원가입절차는 아이디, 비밀번호, 전화번호(인증), 거래가능지역, 약일을 받을 수 있는 시간을 설정한다. 4. 시스템은 데이터베이스에 사용자 정보를 저장한다. 5. 그 후 로그인창으로 나가지는 것이 아니라 바로 약 화면 안으로 들어간다. 	
시나리오 2	로그인
<ol style="list-style-type: none"> 1. 스마트폰 어플리케이션을 기반으로 하기 때문에, 이전에 자동로그인을 설정해놓았다면 별도의 로그인 절차 없이 접속하도록 한다. 2. 일반사용자는 자동로그인이 설정되지 않았다면 아이디와 비밀번호를 입력한다. 3. 시스템은 데이터베이스의 로그인정보와 비교하여 로그인을 가능하도록 한다. 	
시나리오 3	판매상품 올리기

1. 판매하기 탭을 클릭한다.
2. 시스템 매니저와 AI 가 구성한 질문에 답변하고 요청하는 사진을 업로드한다.
3. 거래가능지역을 설정하고,
4. 질문을 통해 시스템이 자동으로 내용을 구성한다.
5. 허위 사실을 기재했을 경우 있을 불이익에 동의하는 창에 동의한다.
6. 24 시간동안 삭월에 올라가게 되고, 현재의 조건으로 올릴 경우 몇 명이 구매하고 싶다고 표시하고 있는지 보여준다.
7. ===시스템이 자동으로 처리하고, 거래되면 알림을 판매자한테 보낸다.===
8. 그 후의 과정은 거래 시나리오로 이어진다.

시나리오 4

전문 판매자 상품 올리기

1. 판매하기 탭을 클릭한다.
2. 시스템 매니저와 AI 가 구성한 질문에 답변하고 요청하는 사진을 업로드한다.
3. 거래가능지역을 설정한다.
4. 질문을 통해 시스템이 자동으로 내용을 구성한다.
5. 허위 사실을 기재했을 경우 있을 불이익에 동의하는 창에 동의한다.
6. 광고가 AI 에 의해서 자동적으로 사용자에게 보여진다.
7. 그 후 과정은 거래 시나리오로 이어진다.

시나리오 5

시스템 매니저 로그인

1. 시스템 매니저는 다른 별도의 어플리케이션을 사용한다.
2. 다른 어플리케이션을 통해 회사에서 부여받은 계정으로 로그인하여 시스템을 관리한다.

3. 그 후 시스템을 관리하는 과정은 시스템 관리자의 시나리오를 따른다.

시나리오 6

전문 판매자 로그인

1. 일반 계정으로 가입한 후 전문 판매자 계정으로 전환을 신청하면 전문 판매자 계정으로 전환
2. 전문판매자의 경우 삭딜 사이에 전문 판매자의 상품을 광고해주는 방식이기 때문에 일정의 광고료를 받는다.
3. 시간이나, 빈도에 따라 광고료가 다르게 자동으로 측정되고 판매자는 해당 서비스를 결제한다.
4. 전문판매자 계정 로그인 절차는 동일하다. 단 전문판매자 계정으로서는 구입을 할 수 없다.
5. 로그인 후 판매 절차는 일반적인 판매절차와 동일하다. 시스템이 자동으로 판매에 올려놓은 상품 중 광고서비스를 결제한 제품에 대해서 일정확률로 일반사용자의 삭딜에 띄워준다.

시나리오 7

삭딜 등록

1. 물품 등록과 함께 삭딜이 시작되며, 판매자에게 현재 리스트에 있는 사람의 수와 몇 번째 순서가 진행되고 있는지 실시간으로 정보를 확인할 수 있도록 한다.

시나리오 8

구매성사시 알림

- 1 .삭딜과 일반딜 모두 거래가 성사되면 바로 거래할 수 있도록 알림을 보낸다.
- 2 알림을 클릭하면 거래상황으로 넘어간다.
- 3 알림을 일정시간 이상 클릭하지 않으면 거래가 파기되고 시스템은 판매자에게 패널티를 부여한다.(어떤 패널티?)

시나리오 9	구매가 성사되지 않고 삭딜이 종료된 경우
	<ol style="list-style-type: none"> 1. 이후 삭딜이 종료되었는데에도 매칭이 이루어지지 않으면 종료 알림과 함께 구매자들이 거래를 거부한 피드백을 확인할 수 있다. 2. 판매자에게 두가지 선택권이 주어진다. <ul style="list-style-type: none"> -첫째 : 피드백을 반영해 거래 조건을 변경하고 다시 삭딜에 올린다. -둘째 : 삭딜을 포기하고 일반 상품으로 그대로 전환시킨다.
시나리오 10	일반 딜로 넘어간 경우
	<ol style="list-style-type: none"> 1. 구매자는 관심 있는 상품을 위시리스트에 보관할 수 있다. 2. 구매자가 상품 리스트를 검색하고 마음에 드는 매물에 '관심' 버튼을 누른다. 3. 판매자에게 삭딜과 마찬가지로 스와이핑 메시지가 전달된다. 4. 구매자의 제한된 정보(희망 거래 지역 및 신뢰도 등급)이 제시된다. 5. 거래가 성사되면 삭딜과 동일하게 거래 성사 메시지와 함께 연락처 정보가 양측에 전달된다.

>> 관리자 <<

Actor	<ul style="list-style-type: none"> • 판매자/구매자 • 관리자: 관리자는 플랫폼/AI 가 해결할 수 없는 일이나, 사람이 직접 수행하고, 그 결과를 보고해야 하는 일들을 도맡아 수행한다.
Keyword	물품, 삭탈, 플랫폼(일단 시스템의 사람이 아닌 모든 구성요소를 플랫폼이라고 지칭했습니다.), 거래, 요청 양식, 발신, 수신,
-	
시나리오 1	관리자가 물품/회원 관리에 대해 수행할 일
	<ol style="list-style-type: none"> 1. 판매자/구매자는 판매자가 등록한 물품이나 구매자/판매자의 행위가 플랫폼이 정한 규칙에 위배된다고 판단될 시, 플랫폼을 통해 관리자에게 해당 물품과 판매자/구매자를 신고할 수 있다. 2. 관리자는 플랫폼을 통해 판매자와 구매자의 신고를 수신할 수 있다. 3. 관리자는 플랫폼의 규칙에 따라 판매자가 등록한 물품의 등록을 취소할 수 있다. 이때, 관리자는 플랫폼을 통해 해당 판매자에게 이 사실을 알려야 한다. 4. 관리자는 일정 기준 이상 플랫폼의 규칙을 어긴 판매자/구매자의 회원 등록을 취소할 수 있다. 이 때, 해당 회원의 정보를 이후 재가입을 방지할 목적으로 블랙리스트에 등록할 수 있다. 5. 관리자는 플랫폼이 등록을 보류한 물품의 등록을 취소하거나 허가할 수 있다. 6. 구매자와 판매자는 플랫폼의 오류를 신고할 수 있다.
시나리오 2	플랫폼이 항상 수행할 일
	<ol style="list-style-type: none"> 1. 플랫폼은 오류가 발생해 정보를 수신하거나 발신하지 못할 시, 수행하던 작업을 멈추고, 어떤 문제가 어디서, 가능하다면 왜 발생했는지 관리자에게 알리고, 판매자/구매자에게 오류코드를 알린다. <ol style="list-style-type: none"> a. 이 때, 관리자의 판단 아래 점검을 하게 될 수 있다
시나리오 3	플랫폼이 물품 판매/구매 등록에 대해 수행할 일
	<ol style="list-style-type: none"> 1. 플랫폼은 판매자/구매자가 판매할/구매할 물품을 등록하겠다고 발신한 사항을 수신하면, 해당 판매자/구매자에게 양식을 발신한다. 플랫폼은 판매자/구매자가 양식을 발신하면, 제대로 작성했는지 확인한다. <ol style="list-style-type: none"> a. 양식이 제대로 작성되지 않았을 시, 제대로 작성하지 않은 사항을 판매자/구매자에게 발신한다. b. 양식이 제대로 작성되었을 시, <ol style="list-style-type: none"> i. 플랫폼은 동일한 판매자/구매자나 동일한 지역, 동일한 물품이 짧은 시간 단위로 등록을 시도할 시, 해당 상품의 등록을 보류하고 이 사실을 관리자에게 알린다. ii. 플랫폼은 미리 제공된 정보를 사용해 불법적이거나 부적절한 것으로 판단되는 물품이 등록을 시도할 시, 등록을 보류하고 이 사실을 관리자에게 알린다.

- iii. 플랫폼은 위의 사항에 해당하지 않는 물품을 플랫폼에 등록하고, 등록 요청을 발신한 판매자/구매자에게 해당 물품의 등록 사실을 알린다.

시나리오 4

플랫폼이 삭딜에 대해 수행할 일

1. 플랫폼은 판매할 물품이 정상적으로 등록되면, 이미 위시리스트를 통해 등록된 구매 요청을 확인한다.
 - a. 적합한 구매 요청이 없을 시, 24 시간 동안 삭딜 구매자 대기 상태를 유지하며,
 - i. 해당 물품에 대한 구매 요청이 발생하면, 해당 구매자에 대해 삭딜 상태를 유지한다.
 - ii. 해당 물품에 대한 구매 요청이 남은 시간 안에 발생하지 않을 시, 판매자에게 이를 알리고, 일반 딜로 전환한다.
 - b. 적합한 구매 요청이 있을 시, 가장 적합한 구매자를 n 명 선정해 순서가 있는 목록으로 만들어, $24/n$ 시간 동안 해당 목록의 구매자에 대해 삭딜 상태를 유지한다. 삭딜 상태가 유지되는 동안, 구매자는 등록된 단말기를 통해 양식에 작성된 물품과 그에 대한 정보로 생성된 삭딜 팝업을 확인할 수 있고, 정보들의 체크 활성화/비활성화 상태를 전환할 수 있으며, 거절하거나 취소하거나 아무 반응을 보이지 않을 수 있다.

.삭딜 상태 중 구매자가 거래를 수락할 시, 구매자가 체크한 사항을 저장하고, 해당 구매자의 해당 물품 구매 요청을 거래 중으로 전환하고, 해당 물품을 삭딜 거래 중인 상태로 전환한다

1. 삭딜 상태 거래 중, 거래가 취소되면, 취소한 구매자를 저장하고, 해당 물품이 등록된 지 24 시간이 지났는지 확인한다.

.삭딜 상태 중, 구매자의 반응 없이 $24/n$ 시간이 지날 경우, 해당 구매자에 대한 삭딜 상태를 취소한 뒤, 해당 구매자에게 이를 알리고, 순서가 있는 목록의 다음 구매자에 대해 $24/n$ 시간 동안 삭딜 상태를 유지한다.

.삭딜 상태 중, 구매자가 거래를 거절할 시, 거절한 구매자와 해당 구매자가 체크한 사항을 저장하고, 순서가 있는 목록의 다음 구매자에 대해 $24/n$ 시간 동안 삭딜 상태를 유지한다.

.삭딜 상태 중, $24/n$ 이 다 지나지 않은 상태에서, 더는 순서가 있는 목록에 구매 요청이 없을 시,

1. 거래를 취소/거절한 구매자를 제외한 구매자의 구매 요청 목록에서 다시 가장 적합한 구매자를 n 명 선정해, 남은 시간/ n 시간 동안 삭딜 상태를 유지한다.
2. 더이상 거래를 취소/거절한 구매자를 제외한 구매자의 구매 요청이 없을 시, 남은 시간만큼 삭딜 구매자 대기 상태를 유지한다.

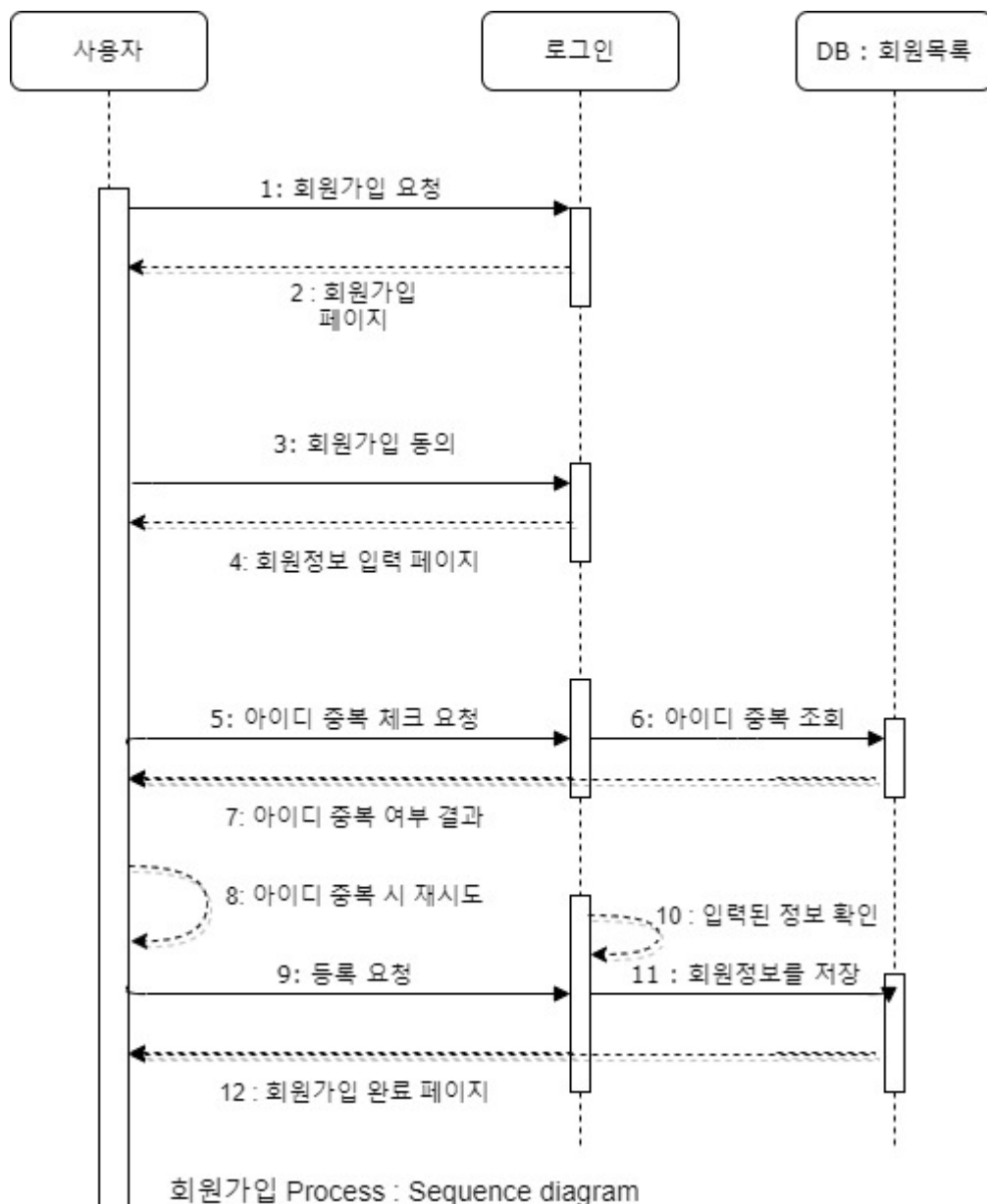
a. 구매자가 해당 물품에 대한 구매 요청을 등록하면, 해당 구매자에 대해 삭딜 상태를 유지한다.

b. 구매 요청이 남은 시간 안에 발생하지 않을 시, 판매자에게 이를 알리고, 일반 딜로 전환한다.

Functional Requirements

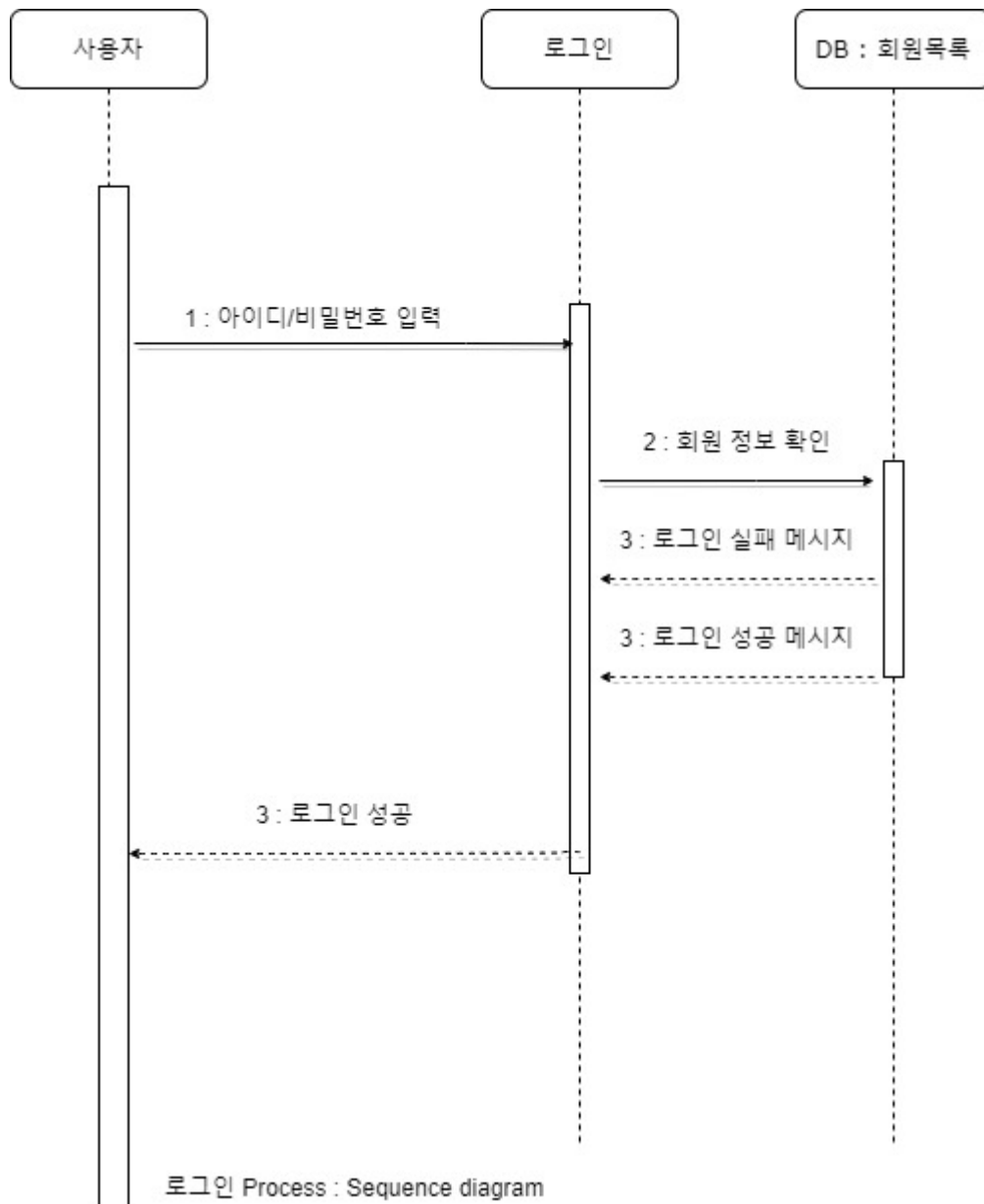
다양한 stakeholder 의 각 단계별 구체적인 요구사항들을 다이어그램을 통해 정리한다.

사용자 공통 - 회원가입 / 로그인



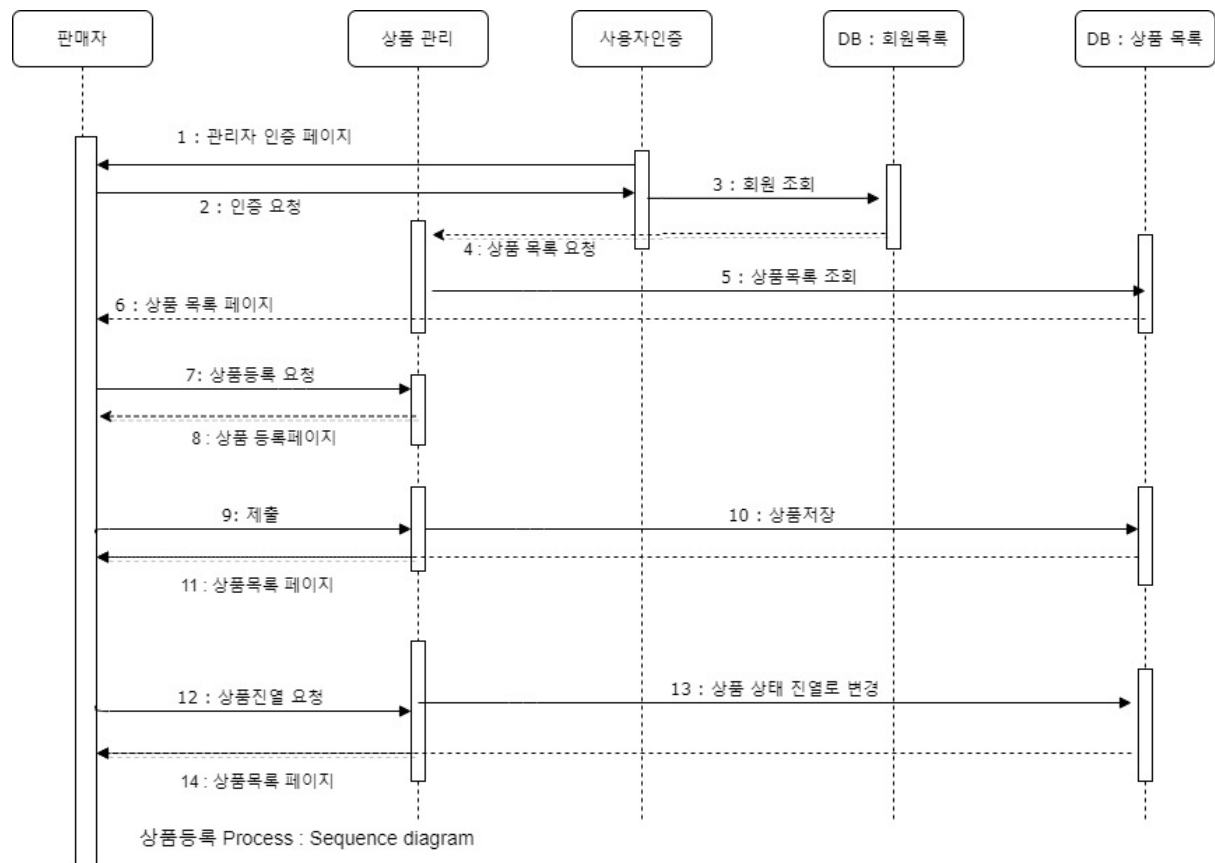
[Diagram 1 : 회원가입 – Sequence diagram]

다이어그램 설명 : 회원가입하는 과정을 나타내는 시나리오이다.



[Diagram 2 : 로그인 – Sequence diagram]

판매자 - 판매상품 등록



[Diagram 3 : 상품 등록 – Sequence diagram]

□ 이벤트 흐름

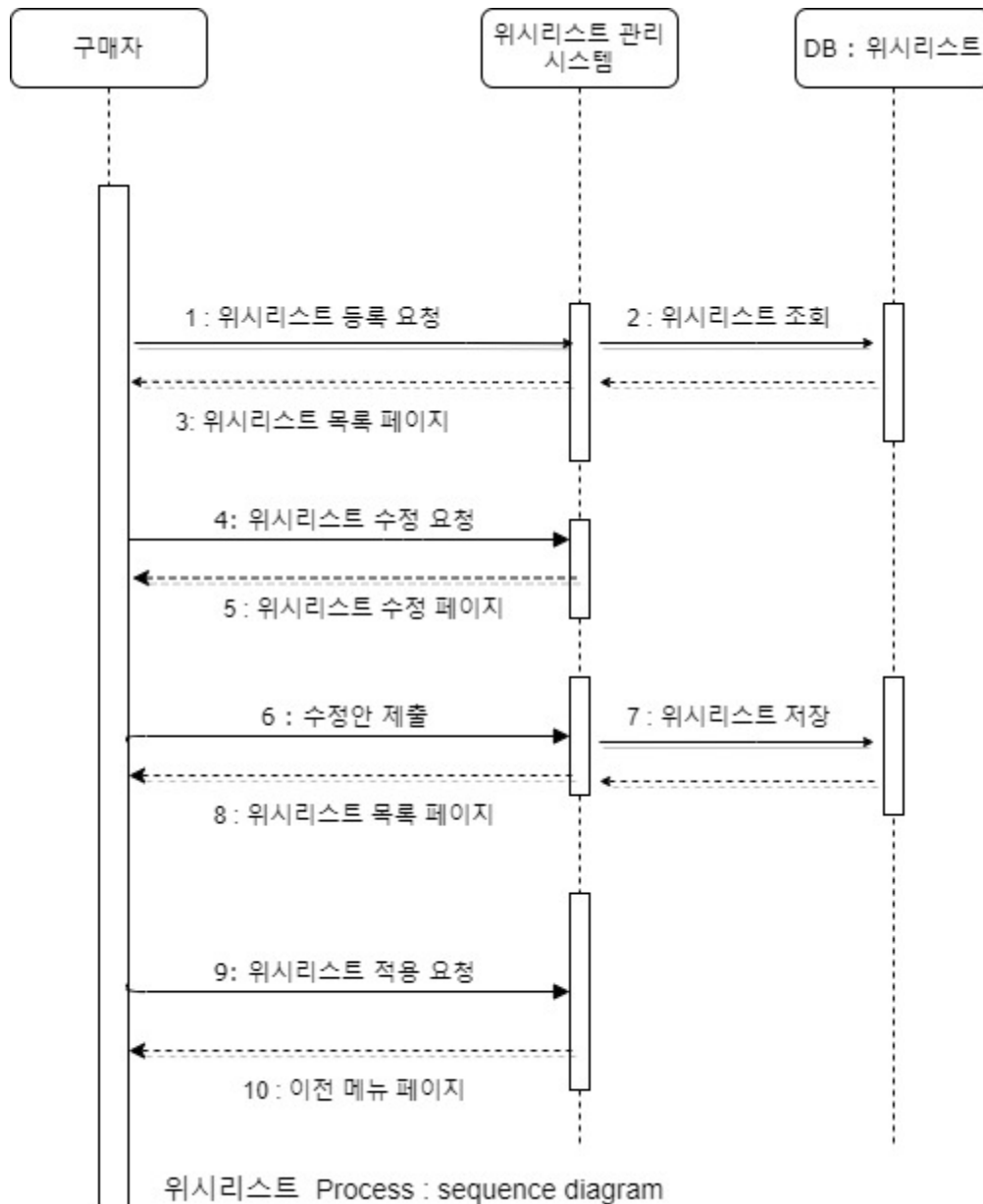
- 정상 흐름

- (1) 관리자 인증을 요구한다(시스템)
- (2) 아이디와 비밀번호를 입력하여 인증을 요청한다(액터)
- (3) 회원 TABLE 에서 권한이 관리자임을 확인한다(시스템)
- (4) 상품목록을 요청한다(시스템)
- (5) 상품 TABLE 에서 상품목록을 조회한다(시스템)
- (6) 상품목록을 보여준다(시스템)
- (7) 관리자는 상품등록을 요청한다(액터)
- (8) 상품등록 페이지를 보여준다(시스템)
- (9) 상품을 모두 입력하고(상품명, 상세설명, 썸네일이미지, 상품사진, 가격, 카테고리 등) 제출한다(액터)
- (10) 상품 TABLE 에 상품을 저장한다(시스템)
- (11) 상품목록을 보여준다(시스템)
- (12) 상품을 진열하도록 요청한다(액터)
- (13) 상품 TABLE 에서 상품상태를 진열로 변경한다(시스템)
- (14) 상품목록을 보여준다(시스템)

- 선택 흐름

- ▶ 관리자가 아닌 경우 오류메시지와 함께 메인 페이지로 돌아간다.
- ▶ 카테고리를 입력하지 않고 등록 요청을 했을 경우 오류메시지를 보여준다.
- ▶ 카테고리가 하나도 없을 경우 오류메시지를 보여주고 카테고리를 하나라도 등록할 때까지 상품등록 페이지를 보여주지 않는다.
- ▶ 상품옵션이 없을 경우 상품상태를 진열로 바꿀 수 없다.

구매자 - 위시리스트 작성 (구매하고자 하는 물건의 조건)



[Diagram 4 : 위시리스트 작성 및 수정 – sequence diagram]

□ 이벤트 흐름

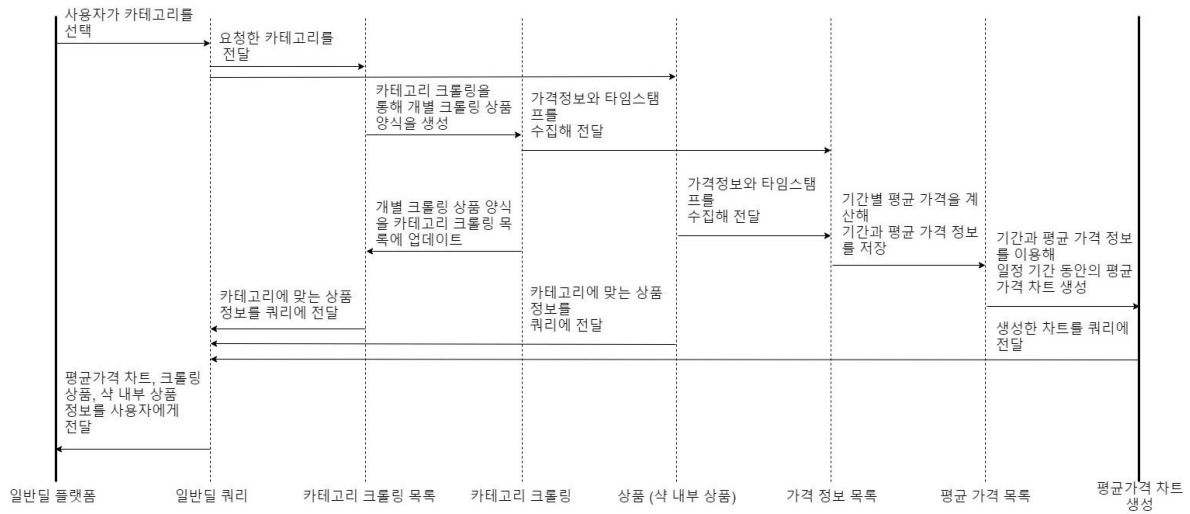
- 정상 흐름

- (1) 위시리스트 등록 요청한다(액터)
- (2) DB 에서 위시리스트 조회한다(시스템)
- (3) 기존 위시리스트 목록을 보여준다(시스템)
- (4) 수정을 요청한다(액터)
- (5) 수정 페이지를 보여준다(시스템)
- (6) 수정안을 제출한다(액터)
- (7) 수정된 위시리스트를 DB 에 저장한다(시스템)
- (8) 위시리스트 목록을 보여준다(시스템)
- (9) 위시리스트 적용 요청한다.(액터)
- (10) 이전 메뉴로 돌아간다.(시스템)

- 선택 흐름

- ▶ 기존 위시리스트가 없을 경우 빈 위시리스트를 보여준다.
- ▶ 수정안을 제출하지 않고 마무리할 경우 저장되지 않는다는 메시지를 보여준다.

구매자 - 카테고리 검색



[Diagram 5: Categorized search & Crawling - Event trace diagram]

▶ 카테고리 검색/ 크롤링 ETD

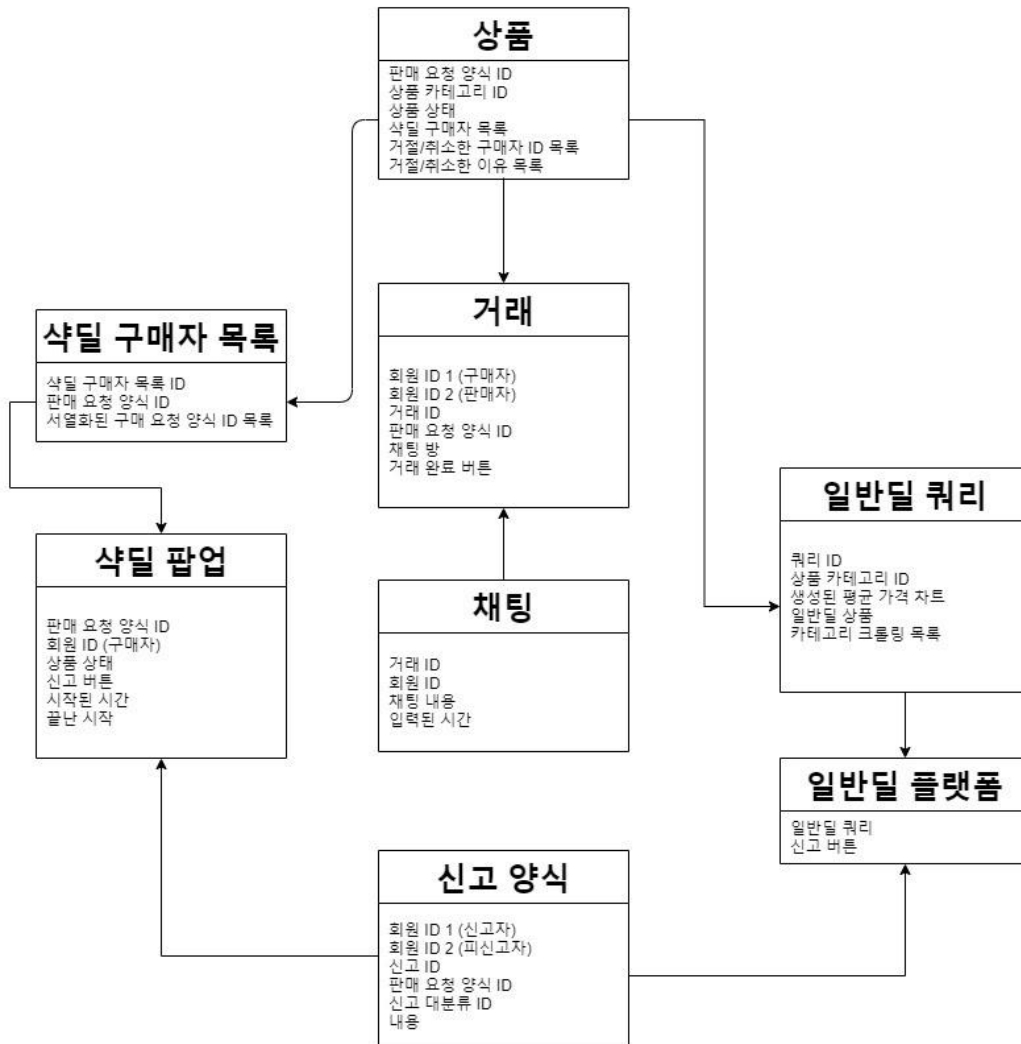
1. 구매자가 검색창에서 카테고리를 선택, '일반딜 쿼리'에 전달
2. 요청한 카테고리를 '카테고리 크롤링 목록'과 '상품(삭 내부 상품)'에 전달
3. '카테고리 크롤링 목록'이 '카테고리 크롤링'을 통해 개별 '크롤링 상품 양식'을 요청
4. '카테고리 크롤링'을 통해 생성된 '크롤링 상품 양식'들을 '카테고리 크롤링 목록'에 업데이트
4. '상품(삭 내부 상품)'의 가격 정보와 수집한 시각의 타임스탬프를 '가격 정보 목록'에 전달
5. '카테고리 크롤링' 중 수집한 제품들의 가격 정보와 수집한 시각의 타임스탬프를 가격 정보 목록에 전달
6. '가격 정보 목록'이 기간별 평균 가격을 계산해, 기간과 평균 가격 정보를 '평균 가격 목록'에 저장
7. '평균 가격 차트 생성' 모듈에서 '평균 가격 목록'정보를 이용해 일정 기간동안의 평균 가격 추이 차트를 생성

9. '일반딜 쿼리'에서 사용자의 단말기로 위의 정보들을 전달

1. 구매자가 구매할 제품의 정보를 '위시리스트'에 작성하면, '구매 요청 양식'이 생성됨
2. 판매자가 판매할 제품의 정보를 '판매 요청 양식'에 작성
3. '판매 요청 양식'이 서버로 전송되면, '서열화' 모듈에 필요한 정보를 전송하고, '삭딜 구매자 목록'을 요청

4. '서열화' 모듈이 '구매 요청 양식'에 필요한 정보를 전송하고, 알맞은 '구매 요청 양식' 요청
5. '구매 요청 양식'이 카테고리, 상품 상태 등의 조건으로 필터링해 카테고리에 알맞은 구매 요청들을 '서열화' 모듈에 전달
6. '서열화' 모듈이 해당 '판매 요청 양식'과 가장 유사한 순으로 수집한 구매 요청을 서열화하여, 그 중 상위 X%를 선별해 '삭딜 구매자 목록' 생성, 무작위로 순서를 배정
7. '삭딜 구매자 목록'과 '판매 요청 양식 ID'가 '상품' 데이터에 전송됨
8. '삭딜 구매자 목록'이 무작위로 배정한 순서가 가장 먼저인 구매자에게 '삭딜 팝업' 생성
9. '삭딜 팝업'이 구매자의 단말기에 삭딜 팝업을 전송
- 10-1. (구매자가 거래 승낙시) 승낙 여부와 구매자가 표시한 세부사항 전달
- 11-1. 상품의 상태를 거래 중으로 전환하고, 구매자의 승낙 여부와 구매자가 표시한 세부사항을 판매자에게 전달
- 12-1. (판매자가 거래 승낙시) 거래 시작
- 12-2. (판매자가 거래 거부시) 상품 삭제
- 10-2. (구매자가 거래 거절/취소/시간 초과시), 승낙 여부와 구매자가 표시한 세부사항 전달
- 11-2-1. (시간이 남아있을 시) 거절/취소한 구매자 ID 를 저장한 뒤, 8 번부터 재시행

구매자 & 판매자 - 일반딜



[Diagram 7: 일반딜 - Relation diagram]

구매자가 상품에 대해 거래 의사를 표하면 거래가 시작됩니다.

거래는 크게 '삭'에 등록된 상품과 크롤링 상품으로 나눌 수 있으며, '삭'에 등록된 상품은 삭딜과 일반딜로 나눌 수 있습니다.

삭딜 상품은 한 번에 오직 한 명의 구매자에게만 표시되며, 해당 구매자에게 할당된 시간이 지나면 다른 한 명의 구매자에게만 표시되는 방식으로 진행됩니다.

삭딜 상품은 각각 '사전에 지정된 삭딜 상태 지속 시간'을 가지며, 이 지속 시간은 등록된 시점부터 흐르기 시작하여, 판매자의 '삭딜 승인 시간'을 제외한 어느 때나 계속해서 흐릅니다.

만약 삭딜 상태 지속 시간이 모두 소진되면, 해당 상품은 '일반딜' 상품으로 전환됩니다.

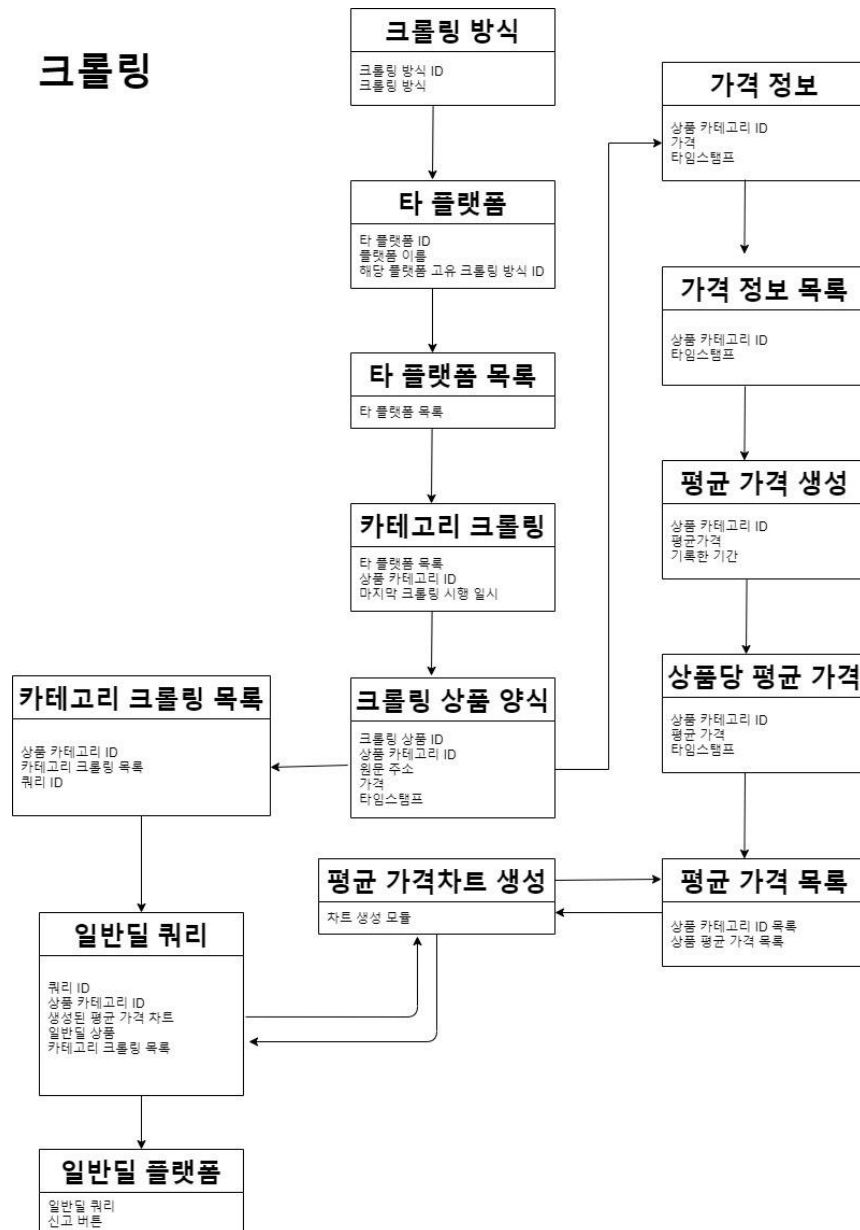
일반딜 상품은 '삭'에서 해당 카테고리의 상품을 검색한 누구에게나 표시되며, 누구나 구매 의사를 표할 수 있습니다.

크롤링 상품은 '삭'이 크롤링을 통해 수집한 외부 중고거래 플랫폼의 상품으로, '삭'은 가격과 수집할 수 있었던 일부 상세 정보, 해당 상품의 외부 플랫폼 게시물로의 링크를 구매자에게 제공합니다.

구매자 혹은 판매자는, 거래 과정 중 거래를 취소할 수 있으며, 만약 최초로 등록한 구매/판매 요청 양식의 내용과 다른 거래를 요구받을 시, 해당 내용을 관리자에게 신고할 수 있습니다.

관리자는 해당 내용의 경중을 고려해 사용자에게 페널티를 부여할 수 있습니다.

관리자 - 크롤링 및 DB 관리



[Diagram 8: Crawling - Relation diagram]

'삭'은 일반딜 상품 검색 시 외부 상품도 크롤링하여 함께 표시해줍니다. 크롤링과 검색은 다음과 같은 과정을 통해 이루어집니다.

회원은 누구나 검색을 할 수 있습니다. 검색은 상기한 바와 같이 대분류에서 소분류로 카테고리를 선택하는 방식이나 키워드를 검색하는 방식으로 진행할 수 있으며, 회원이 검색을 요청할 시, 일반딜 쿼리가 생성됩니다. 쿼리는 평균 가격 차트, 크롤링 상품, 삭 내부 상품, 총 세 가지 정보를 구매자 단말기로 전송해야 합니다.

이를 위해, 쿼리는 '카테고리 크롤링 목록'과 '상품(삭 내부 상품)'에 해당 정보를 요청합니다.

카테고리 크롤링 목록은 평균 가격 차트와 크롤링 상품, 두 가지 정보를 생성하기 위해 다음과 같은 작업을 진행합니다.

우선, '카테고리 크롤링' 모듈을 통해 크롤링 된 상품 각각에 대한 양식을 생성합니다. 이때, '카테고리 크롤링' 모듈은 여러 종류의 타 플랫폼 각각에 적합한, '사전에 정의된 크롤링 방식'을 해당 플랫폼의 ID 를 통해 불러와 크롤링을 진행합니다.

크롤링을 통해 생성된 '크롤링 상품'의 양식에는 해당 양식의 '크롤링 상품 양식 ID', 해당 상품의 카테고리 ID, 상품 사진, 원문 링크, 가격, 수집 가능한 정보들, 타임스탬프가 기록됩니다. 여기서, 수집 가능한 정보들이란, 해당 상품 원문에 기술된 상품 및 거래 상세 정보를 NLP 를 통해 '삭'에서 통용되는, 태그 형식의 상세 정보로 바꾼 것을 의미합니다.

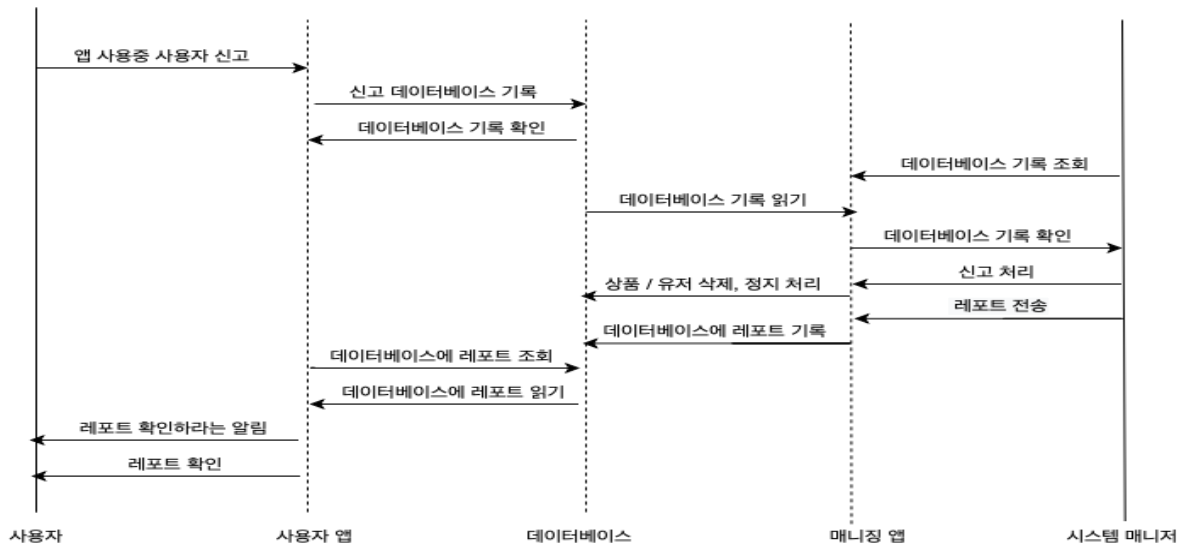
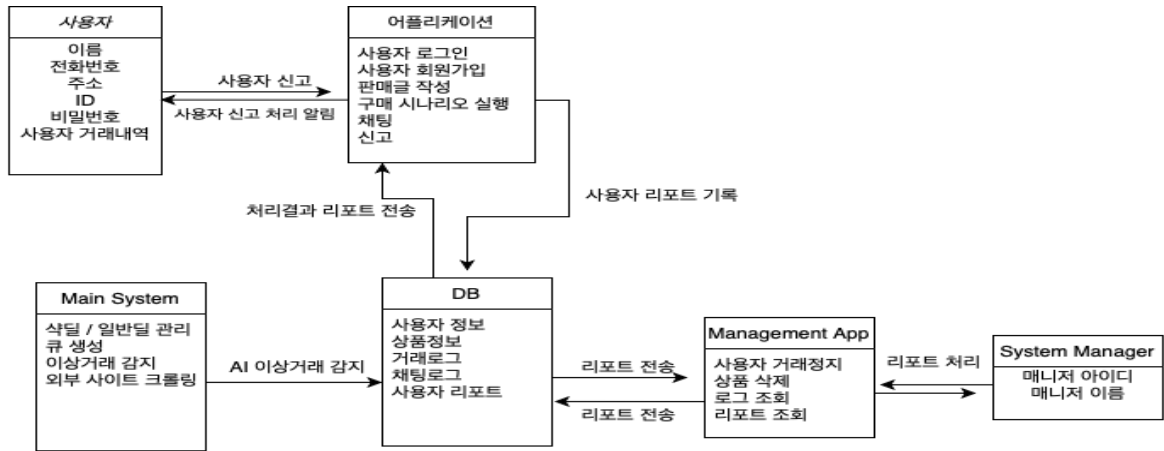
이후, 크롤링 목록과 상품(삭 내부 상품)은 '가격 정보 목록'에 양식 ID 와 가격 정보, 타임 스탬프를 전달합니다. 가격 정보 목록은 해당 정보를 포함해 여태까지 수집한 기간별 평균 가격을 '평균 가격 목록'에 저장합니다. 평균 가격 목록은 기간과 평균 가격 정보를 '평균가격 차트 생성' 모듈에 전달하고, 모듈은 이를 이용해 '해당 상품의 일정 기간의 평균 가격 차트'를 생성해 '일반딜 쿼리'에 전달합니다.

한편, '카테고리 크롤링' 모듈은 '각 크롤링 상품의 양식'을 '카테고리 크롤링 목록'에 업데이트합니다. 이 때, 원래 목록에 있었지만, 더는 유효하지 않은 상품들과 거래 완료 여부도 수집하여 저장합니다. 카테고리 크롤링 목록은 업데이트된 검색 조건에 맞는 크롤링 상품 목록을 일반딜 쿼리에 전달합니다.

상품(삭 내부 상품)도, 검색 조건에 알맞은 상품 정보를 쿼리에 전달합니다.

일반딜 쿼리는 평균가격 차트, 삭 내부 상품, 크롤링 상품 순으로 각각 전달받는 대로 사용자 단말기로 전송합니다.

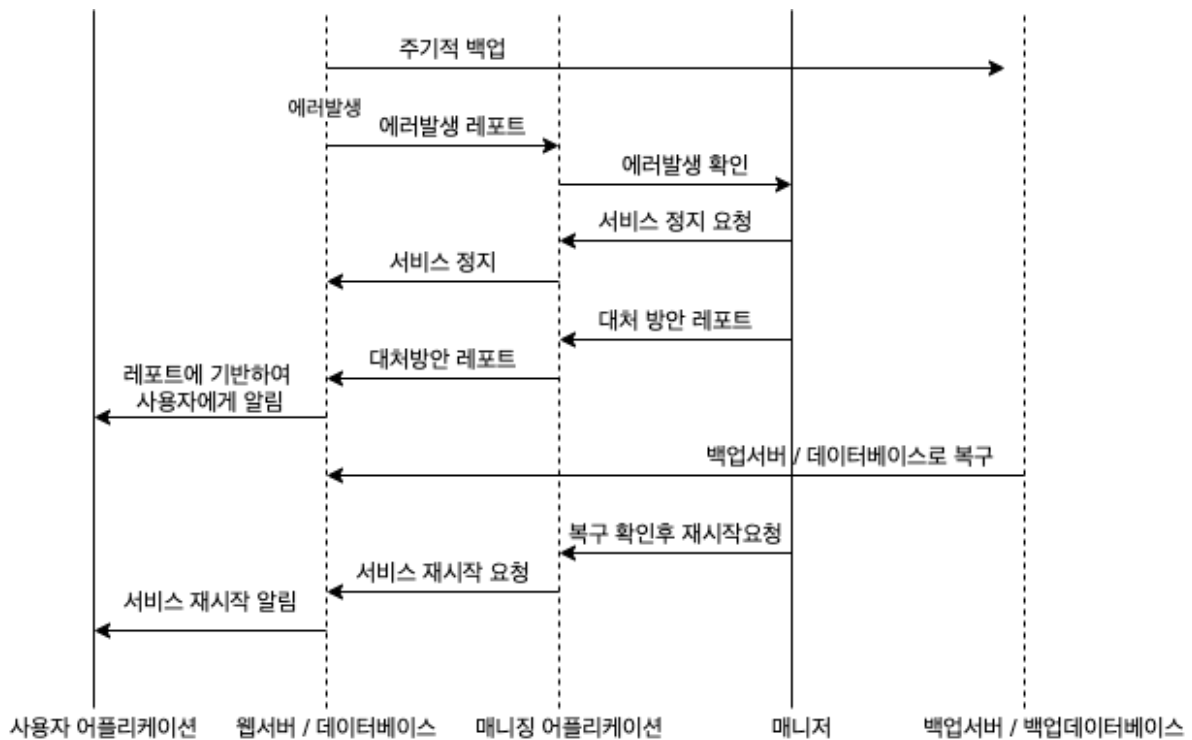
관리자 – System managing



[Diagram 9&10 : System managing -Class diagram & Relation diagram]

사용자가 사용자 혹은 상품에 대해서 신고를 접수하면 데이터베이스에 이를 기록한다. 시스템 매니저가 데이터베이스에 신고된 내역을 확인하면 기록을 전송한다. 기록을 확인하면서 해당 유저나 상품에 대한 처리를 보내면 매니징 앱이 데이터베이스에 정보를 기록한다. 다시 유저가 처리 내역을 확인하고 싶어서 조회를 신청하면 데이터베이스의 정보를 가져다가 레포트를 역으로 보여준다.

관리자 – 오류 처리



[Diagram 11 : 오류처리 – Event trace diagram]

에러시 복구 다이어그램입니다. 주기적으로 백업서버를 두어 중요작업시점 혹은 시간을 정하여 백업을 미리 해두어야 합니다. 특정 시점에서 에러 발생시 가장 최근의 백업으로 데이터베이스를 돌리고 어느 시점으로 돌렸는지 사용자들에게 알리고 그 시점 이후의 거래 내용등을 다시 복구하여야 하고 에러발생 레포트에 포함된 에러 발생 시점과 기록된 로그를 점검하여 에러 발생 요인을 찾아서 대처방안을 다시 리포트하고 백업을 한 후 재시작을 하는 것까지의 과정입니다.

Non-Functional Requirements

해당 시스템의 비 기능적 요구사항들을 정리합니다.

- Usability - 대부분 거래에 대해서 서버는 다중읽기를 지원하면 된다. 동시에 사용하는 경우는 여러명의 사용자가 동시에 "거래" 동작을 수행했을 때이다. 아무래도 다른 웹사이트보다 동시에 거래하는 상황이 많지 않기 때문에 이런 부분에서만 적절하게 처리를 해주면 된다.
- Performance - 속도의 측면보다는 안정적이고 공정하게 시스템이 돌아가야 하기 때문에, 실시간성이나 빠른 속도보다는 정확성과 안정성에 초점을 맞춘 프로그램을 만들어야 한다. 다만 크롤링은 다른 시스템에 정보를 요청하고 받아와야 하기 때문에, 시간이 많이 소비된다. 사용자가 많이 사용하지 않는 시간에 외부 시스템을 크롤링하여 서버에 저장해두고 저장한 정보를 제공해줌으로서 성능 개선을 이뤄야 한다. 다만, 이렇게 실시간적 크롤링이 아니라 서버에 저장하고 보여주는 것은 시세 정보와 같은 다량의 데이터를 처리해야하는 상황에 국한하고 그 이외에 상품을 보여주는 것은 어느정도의 실시간성을 유지해야한다.
- Dependability - 시스템의 상호작용 간에 유저-시스템, 유저-유저 간의 검증 과정을 적극적으로 도입해 오류나 착오가 생기지않도록 조치하여야 하며, 중요한 정보의 처리는 내부에서 진행하고 해당 처리의 결과만 전송하는 방식으로 시스템이 오류를 일으켜도 피해를 최소화할 수 있게 해야한다.
- Security - 개인정보 보호를 위해서 철저하게 해킹이나 프로그래밍상 오류로 개인정보가 유출되지 않도록 적절하게 처리해야한다. 암호화를 적절하게 수행하여야 한다. 해싱 알고리즘과 https 보안을 통해 서버와 주고 받는 정보고 일반 사용자에게 실수로라도 노출되지 않으면서 이를 중간에 가로챌 수 없도록 설계를 해주어야 한다.

또한 메신저 사용 내역도 마찬가지로 외부로 쉽게 노출되지 않도록 관리해야 하며, 일정 기간이 지난 뒤에는 폐기될 수 있도록 조치한다.

- Maintainability - 이후 지속적인 유지보수가 요구될 가능성이 높기 때문에, 서버 및 데이터베이스와 어플 인터페이스가 독립적으로 작동할 수 있도록 해야 한다. 각 function 도 마찬가지로 독립적으로 세분화한다. 이를 통해 어쩔 수 없이 퍼포먼스를 희생해야 할 수도 있을 것이다.

개발 방법론 - Agile

이유: 소규모의 팀원이 짧은 기간안에 만들어야 한다는 조건을 충족시키기 위해 Agile 을 선택하였다. 그리고, 이런 프로그램의 경우 잦은 변경과 요구사항을 반영할 수 있어야 한다는 점에서 여러번의 버전을 릴리스 할 수 있는 Agile 방법론을 선택하여 개발을 진행하려고 한다.

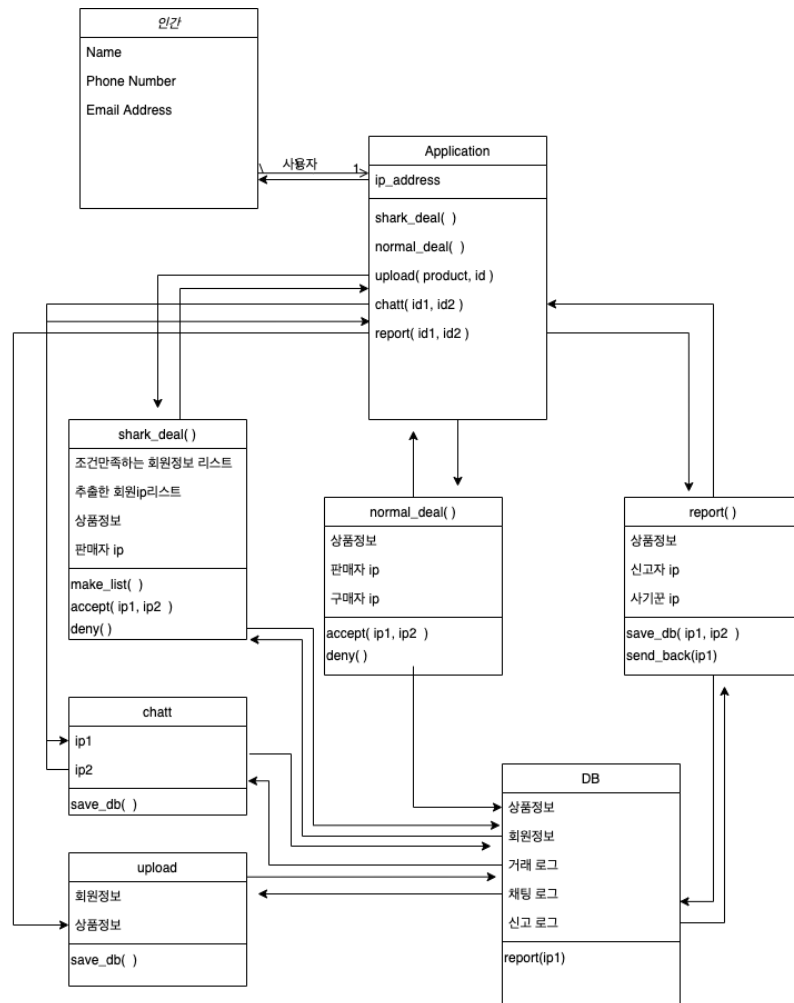
외부에서 가져올 수 있는 오픈소스 (COTS)

- 자연어 처리
 - BeautifulSoup4
 - Nltk
 - Sklearn
 - KoNLP
 - Khaii
- 시계열 자료 분석 및 시각화
 - Matplotlib
 - Prophet
- 어플리케이션 디장인
 - 일반 중고거래 사이트 데이터베이스구조 (구매)
 - 스마트메이커 (app-design)
- 서버 구축
 - Heroku (웹서버)

System architecture

특정 기능이 아닌 시스템 전반의 세부적인 구조에 대해 다이어그램을 통해 설명합니다.

프론트엔드 아키텍처

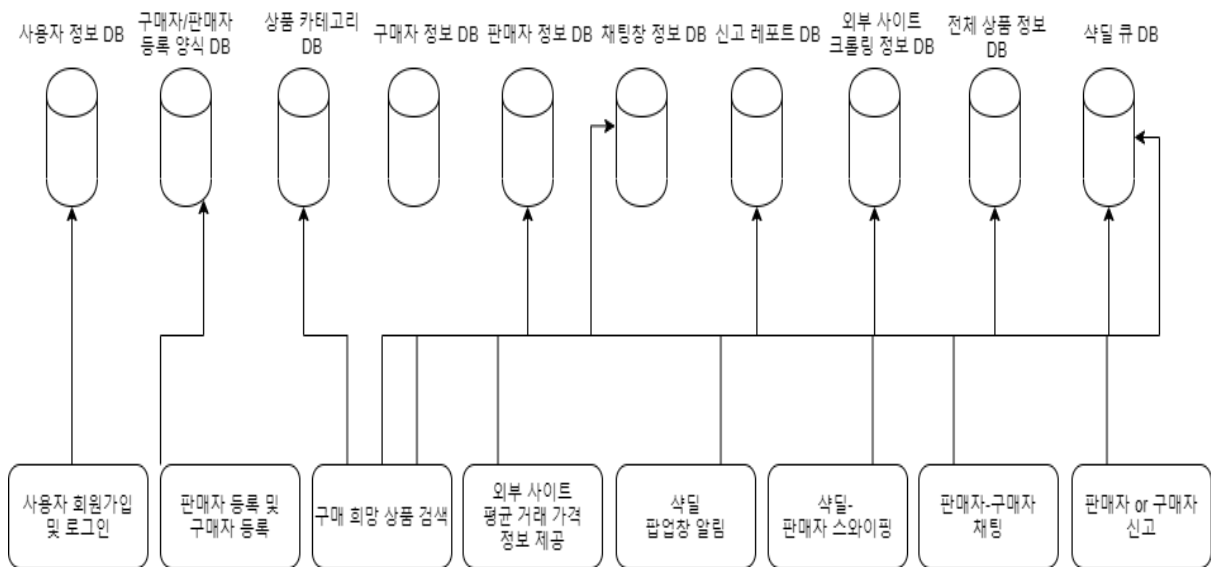


[Diagram 12 : Front-end architecture – Class diagram]

아래는 프론트 엔드의 작동방식을 서술해놓은 것이다. 프론트엔드는 서버에 정보를 입력하고 받아오는 방식으로 작동한다. 사용자가 application 을 통해서 여러가지 기능을 요청하면 그에 따른 호출이 이뤄지고 호출을 통해 기능이 작동하도록 한다.

서버에서 많은 작업을 수행하면서도, 어플리케이션에 정보를 전달하여 어플리케이션이 직접 기능들을 호출하도록 만들어서 어플리케이션에 높은 자유도를 준다.

데이터베이스 아키텍처



[Diagram 13 : Database and Main Functionality Relationship]

위 diagram 은 플랫폼 내 주요 기능과 데이터베이스 간의 연결 관계를 보여준다.

-사용자 회원가입을 통해 사용자 정보 데이터베이스의 내용은 업데이트 되며 사용자 로그인 시도 시 역시 이 데이터베이스를 참조한다.

-판매자 또는 구매자 등록 기능 수행을 위해서는 양식 데이터베이스의 데이터를 참조한다.

-구매자가 구매 희망 상품을, 일반 딜을 통해 찾고자 할 때는 상품 카테고리 데이터베이스를 참조하며 어떤 판매자가 올린 품목인지 확인하기 위해 판매자 정보 데이터베이스 역시 참조한다.

이는 삭제에 대해서도 동일하게 적용되는데, 구매자가 어떤 품목에 대해 구매 의사를 표현했을 때 이 정보는 삭제 큐 데이터베이스에 저장된다.

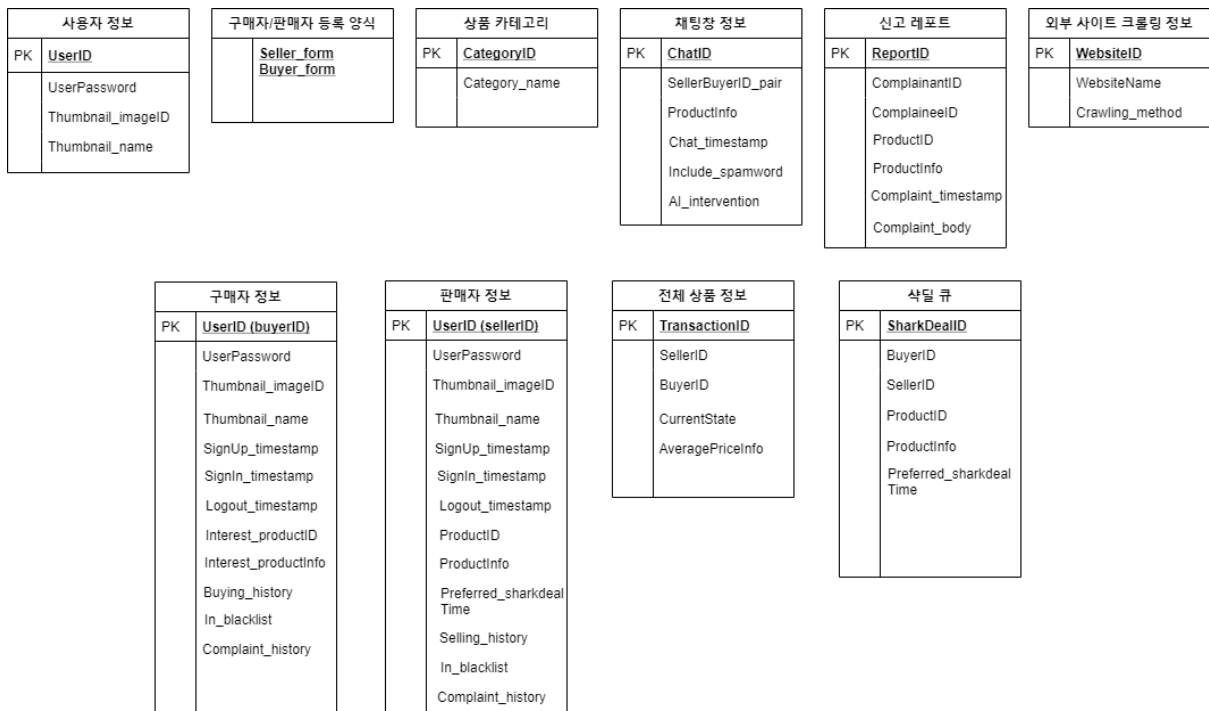
-외부 사이트에 존재하는 동일 품목에 대한 평균 거래 가격에 대한 정보를 제공하는 기능을 수행하기 위해서는 이와 관련된 데이터베이스에 저장되어 있는 내용을 사용한다.

-삭딜 팝업창 알림은 판매자의 알림 설정에 따라 수행되므로 판매자 정보 데이터베이스에 있는 내용을 이용한다. 팝업창 알림을 통해 생성된 구매 희망자 큐의 순서에 따라 판매자에게 구매희망자 정보를 하나씩 제공하기 위해서 삭딜 큐 데이터베이스를 참조한다.

-판매자와 구매자의 채팅 기록은 채팅 데이터베이스에 저장된다.

-부당 거래 또는 일방적 거래 파기와 같은 내용은 판매자 또는 구매자가 작성한 신고 레포트의 내용에 포함되며 이는 신고 레포트 데이터베이스에 저장된다.

Entity Relationship Diagram



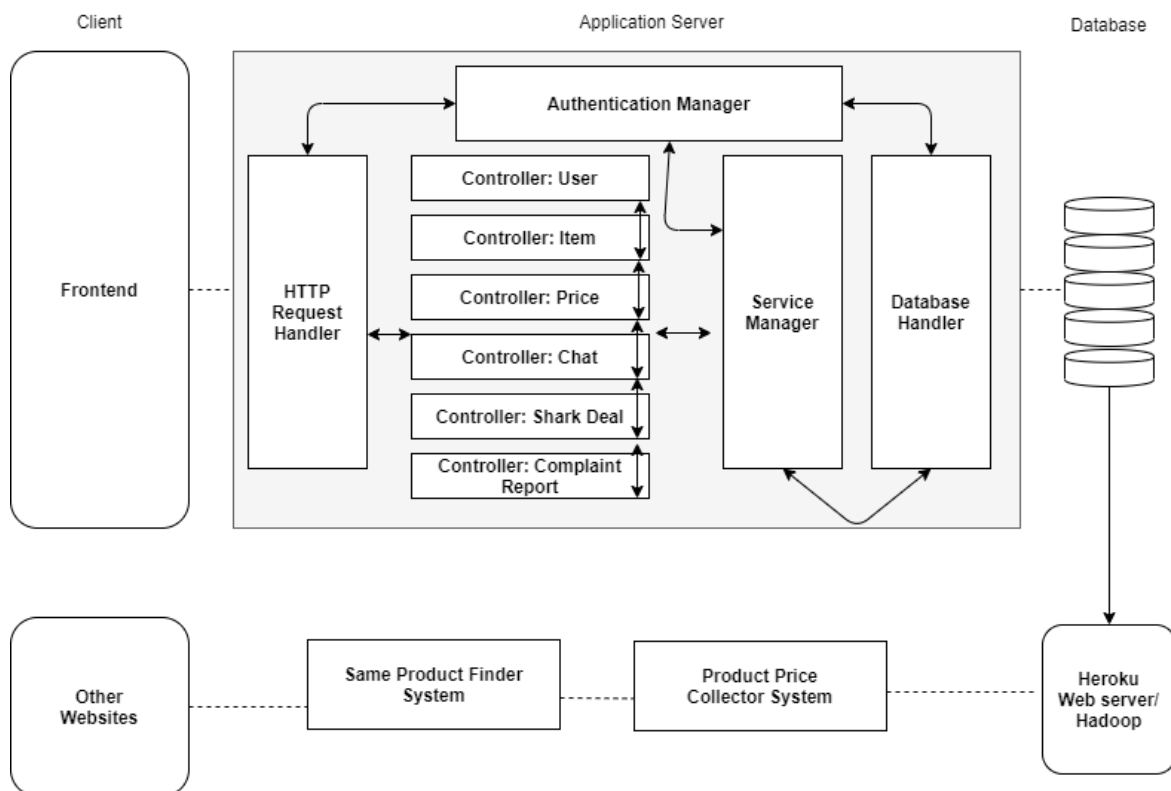
[Diagram 14 : Entity Relationship Diagram for System Database]

총 10 개의 데이터베이스에 포함되어 있는 정보가 시스템의 주요 기능 수행 시 관여한다. 구매자 정보 데이터베이스와 판매자 정보 데이터베이스는 사용자 정보 데이터베이스로부터 조건에 맞는 (구매자인지 판매자인지) 정보를 불러온다. 구매자/판매자 등록 양식 데이터베이스는 사용자가 구매자 또는 판매자 등록을 할 때 사용하는 양식을 보관하는 용도로 쓰인다. 채팅창 정보

데이터베이스의 include_spamword 는 유해 정보가 포함되어 있는지에 대한 여부를 표현해주며, AI_intervention 은 시스템 매니저의 수동적인 관리를 요하지 않고도 AI 가 알아서 부당거래, 유해 콘텐츠 요소 등을 파악하여 조치를 취한 결과를 표현해준다. 외부 사이트 크롤링 정보

데이터베이스는 외부 사이트 각각의 특징을 파악하여 그에 맞춘 크롤링 방법에 대한 정보를 포함한다. 구매자와 판매자 정보 데이터베이스에는 모두 사용자에게 대한 개인 정보와 과거 거래 이력, 신고 이력, 블랙리스트 포함 여부와 같은 정보를 담는다. 전체 상품 정보는 상품의 현재 상태 (삭질 대기 중, 거래 완료 등)을 나타내며 이 결과가 Front-end 에도 반영이 되도록 한다. 삭질 큐는 판매자와 그가 올린 품목, 구매 희망자 정보를 담고 있으며 이를 통해 판매자에게 대기열에 있는 구매희망자 한명씩을 매칭해줄 수 있다.

웹서버 아키텍처



[Diagram 15 Backend Architecture]

크게 client 가 볼 수 있는 창인 Front-end 적인 측면과 Application Server, 그리고 이에 관여되는 database 로 나누어 정리할 수 있다. Application 내에서는 웹 서비스로 작용하고자 할 경우 HTTP

Request Handler 가 필요하며 사용자 정보, 채팅과 같은 주요 기능에 대한 Controller 가 포함된다. 이와 같이 시스템 내부에서의 Controller, Handler 를 정의할 수 있으며 우리 플랫폼의 경우 외부 사이트에서의 가격 정보도 크롤링 해와야 하므로 이를 위한 시스템을 두 개로 정의할 수 있다. 외부 사이트와 우리 플랫폼 내에서 발생한 데이터는 Heroku Web Server 와 Hadoop 을 이용하여 관리 가능하다.

System Requirement Specification

작성한 architecture 가 갖추어야 할 기능적, 비기능적 요구사항에 대해서 서술합니다.

Front-end Architecture Requirement

이 아키텍처는 DB 와 웹서버의 여러 기능들 그리고 사용자 단말기 어플리케이션의 정보를 주고 받는 방향과 각 오브젝트의 함수들을 정의해 놓은 구조이다.

[기능적 요구사항]

1. 앞서 정의된 내용 참조.

[비기능적 요구사항]

1. 첫째, 데이터베이스로부터의 정확한 호출이다. 많은 정보들이 데이터베이스로부터 조회되고 입력된다. 이 과정에서 정확한 정보가 정확한 ip 로 전달되어야 한다. 만약 잘못된 ip 로 정보가 간다거나 요청한 정보를 제대로 제공하지 못할 경우 소비자는 여기서 사용에 어려움을 겪게 된다. 혹은 이런 잘못된 정보들이 오가면서 불필요한 트래픽을 발생시키곤 한다. 혹은 불필요한 정보를 처리할 때 발생하는 과부하나 시스템 오류를 피할 수 없다. 사용자가 전문가가 아닌 일반인이 많기 때문에 이런 에러의 발생을 최소화하여 프론트엔드를 구성하여야 한다.
2. 둘째, 정보의 암호화이다. 소비자가 보내는 정보를 중간에 탈취당할 수도 있고, 동시에 암호화하지 않아서 보관할 경우 사용자가 이를 유출하게 될 수도 있다. 사용자가 다양한 다른 사용자와 연결을 주고 받는 어플리케이션에서 사용자

하나가 단순히 해당 사용자만의 정보를 지니는 것이 아니라 다른 상호작용하는 사용자의 정보까지 지니고 있을 가능성이 크기 때문에 이 부분에서 안정성을 보장하여야 한다.

3. 셋째, 기능에 맞는 적절한 UI 이다. 프론트엔드의 여러 기술들은 사용자가 명령을 호출했을 때, 오류없이 해당기능이 나와야 한다. 그러기 위해선 해당 아키텍처에 맞는 적절한 UI 가 구상되어야 한다. 예를 들어 Application 의 함수를 호출하는 버튼을 기획하였을 때, 이 버튼이 다른 함수를 호출하는 것과 혼동되면 안된다.

거래시스템 종합 Architecture Requirement

[기능적 요구사항]

1. 앞서 정의된 내용 참조.

[비기능적 요구사항]

1. 확장성을 높이기 위해 카테고리 크롤링과 구매 요청 양식을 서열화한다. 하지만 사용자의 입장을 잘못 판단한 설부른 확장은 사용용이성을 떨어지게 할 수 있다. 크롤링과 서열화를 이용할 때 사용자의 입장에서 긍정적인 영향으로만 줄 수 있도록 체계적인 테스트 과정이 필요하다.
2. '삭'은 거래가 완료된 내부 상품과 크롤링 상품의 자료를 수집해, 이후 평균 가격 목록 및 카테고리, 상품 상세 정보 태그화에 활용하기 위해 저장하므로, 저장하는 과정에서 회원의 개인정보가 포함되지 않도록 익명화하거나, 개인정보가 담긴 정보를 제외하도록 설계해 security 에 중점을 두어야한다.
3. 평균 가격 목록은 외부에서 크롤링된 정보까지 포함하여 사용하므로, 해당 내용에 삭의 평균가격만을 보는 기능이나, 데이터를 선별해 추출할 수 있는 알고리즘을 통해 정보 조작의 가능성을 최대한 배제하고, 데이터의 투명성을 확보하여야한다.

Database Architecture Requirement

위 diagram 은 플랫폼 내 주요 기능과 데이터베이스 간의 연결 관계를 보여준다.

[Database and Main Functionality Relationship Diagram]

[기능적 요구사항]

1. 사용자 회원가입을 통해 사용자 정보 데이터베이스의 내용은 업데이트 되며 사용자 로그인 시도 시 역시 이 데이터베이스를 참조한다.
2. 판매자 또는 구매자 등록 기능 수행을 위해서는 양식 데이터베이스의 데이터를 참조한다.
3. 구매자가 구매 희망 상품을, 일반 딜을 통해 찾고자 할 때는 상품 카테고리 데이터베이스를 참조하며 어떤 판매자가 올린 품목인지 확인하기 위해 판매자 정보 데이터베이스 역시 참조한다. 이는 삭딜에 대해서도 동일하게 적용되는데, 구매자가 어떤 품목에 대해 구매 의사를 표현했을 때 이 정보는 삭딜 큐 데이터베이스에 저장된다.
4. 외부 사이트에 존재하는 동일 품목에 대한 평균 거래 가격에 대한 정보를 제공하는 기능을 수행하기 위해서는 이와 관련된 데이터베이스에 저장되어 있는 내용을 사용한다.
5. 삭딜 팝업창 알림은 판매자의 알림 설정에 따라 수행되므로 판매자 정보 데이터베이스에 있는 내용을 이용한다. 팝업창 알림을 통해 생성된 구매 희망자

큐의 순서에 따라 판매자에게 구매희망자 정보를 하나씩 제공하기 위해서 삭딜 큐 데이터베이스를 참조한다.

6. 부당 거래 또는 일방적 거래 파기와 같은 내용은 판매자 또는 구매자가 작성한 신고 레포트의 내용에 포함되며 이는 신고 레포트 데이터베이스에 저장된다.

7. 판매자와 구매자의 채팅 기록은 채팅 데이터베이스에 저장된다.

[비기능적 요구사항]

1. 외부 사이트의 정보를 크롤링하는 작업도 수행하고 동시에 플랫폼 내부의 정보도 꾸준히 데이터베이스에 업데이트 시켜줘야 하기 때문에 각 데이터베이스의 정보들이 독립적으로, 안정성 있게 저장되는 것이 중요하다 (dependability의 측면)
2. Performance의 측면에서도 외부 사이트 정보를 효율적으로 크롤링하는 것이 필요하다. 사용자가 많이 접속하지 않는 시간에 이와 같은 작업을 처리하여 종합적인 성능 개선을 이루어야 한다.
3. 무엇보다도, 사용자들이 거래 희망 지역과 같은 개인 정보를 입력하기 때문에 보안이 중요하다. 암호화 수행은 필수적으로 이루어져야 하며, https 보안을 통해 데이터 전송 과정에서의 보안 또한 신경 써야 한다.

[Entity Relationship Diagram]

[기능적 요구사항]

총 10 개의 데이터베이스에 포함되어 있는 정보가 시스템의 주요 기능 수행 시
관여한다.

1. 구매자 정보 데이터베이스와 판매자 정보 데이터베이스는 사용자 정보 데이터베이스로부터 조건에 맞는 (구매자인지 판매자인지) 정보를 불러온다.
2. 구매자/판매자 등록 양식 데이터베이스는 사용자가 구매자 또는 판매자 등록을 할 때 사용하는 양식을 보관하는 용도로 쓰인다.
3. 채팅창 정보 데이터베이스의 include_spamword 는 유해 정보가 포함되어 있는지에 대한 여부를 표현해주며, AI_intervention 은 시스템 매니저의 수동적인 관리를 요하지 않고도 AI 가 알아서 부당거래, 유해 콘텐츠 요소 등을 파악하여 조치를 취한 결과를 표현해준다.
4. 외부 사이트 크롤링 정보 데이터베이스는 외부 사이트 각각의 특징을 파악하여 그에 맞춘 크롤링 방법에 대한 정보를 포함한다.
5. 구매자와 판매자 정보 데이터베이스에는 모두 사용자에게 대한 개인 정보와 과거 거래 이력, 신고 이력, 블랙리스트 포함 여부와 같은 정보를 담는다.
6. 전체 상품 정보는 상품의 현재 상태 (삭딜 대기 중, 거래 완료 등)을 나타내며 이 결과가 Front-end 에도 반영이 되도록 한다.

7. 삭딜 큐는 판매자와 그가 올림 품목, 구매 희망자 정보를 담고 있으며 이를 통해 판매자에게 대기열에 있는 구매희망자 한명씩 매칭해줄 수 있다.

[비기능적 요구사항]

1. 앞서 언급한 [Database and Main Functionality Relationship Diagram] 의 비기능적 요구사항 참조.

Webserver Architecture Requirement

[기능적 요구사항]

크게 client 가 볼 수 있는 창인 Front-end 적인 측면과 Application Server, 그리고 이에 관여되는 database 로 나누어 정리할 수 있다.

1. Application 내에서는 웹 서비스로 작용하고자 할 경우 HTTP Request Handler 가 필요하며 사용자 정보, 채팅과 같은 주요 기능에 대한 Controller 가 포함된다.
2. 이와 같이 시스템 내부에서의 Controller, Handler 를 정의할 수 있으며 우리 플랫폼의 경우 외부 사이트에서의 가격 정보도 크롤링 해와야 하므로 이를 위한 시스템을 두 개로 정의할 수 있다.
3. 외부 사이트와 우리 플랫폼 내에서 발생한 데이터는 Heroku Web Server 와 Hadoop 을 이용하여 관리 가능하다.

[비기능적 요구사항]

1. 플랫폼 내부의 정보를 담고 있는 데이터베이스와 외부 사이트 크롤링 정보를 담고 있는 데이터베이스가 안정성을 유지하며 독립적으로 운영되어야 한다는 점에서 dependability 가 강조된다.

2. 한편으로는, 각 controller 가 서로 상호작용하며 데이터베이스에 저장되어 있는
공통된 정보들을 꾸준히 수정/업데이트하기 때문에 각 controller 간의
communication style 이나 format 은 통일되어야 하며 보안 수준이 높지 않아야
한다.

System models

시스템의 전반적인 구성요소들(객체)과 해당 요소들 사이의 관계에 대해 도식으로 표현한다.



[Diagram 16 : Overall Architecture]

삭딜, 일반딜 크롤링, 거래를 아우르는 다이어그램입니다. '삭'에서 등록된 상품은 삭딜을 거쳐 일반딜로 전환될 수 있으며, '삭'외부의 상품은 크롤링 상품으로 일반딜 검색시 표시됩니다. 구매자는 삭딜, 일반딜을 통해 물품을 거래할 수 있습니다.

System Evolution

시스템에 요구될 수 있는 변화 사항과, 새롭게 제기될 수 있는 user requirements 를 기술합니다.

신품과 비교하는 DB 구축

- 현재 시스템은 다른 중고 거래 플랫폼에 등록된 상품 정보를 크롤링하는 기능만을 포함하지만, 이후 중고 거래가 더욱 활성화된다면 신품 가격이나 해외 직구 가격 등 다양한 지표들과 비교할 수 있는 종합 가격 비교 어플을 구축할 수 있을 것.

이미지 학습을 통한 DB 구축 및 사기 방지

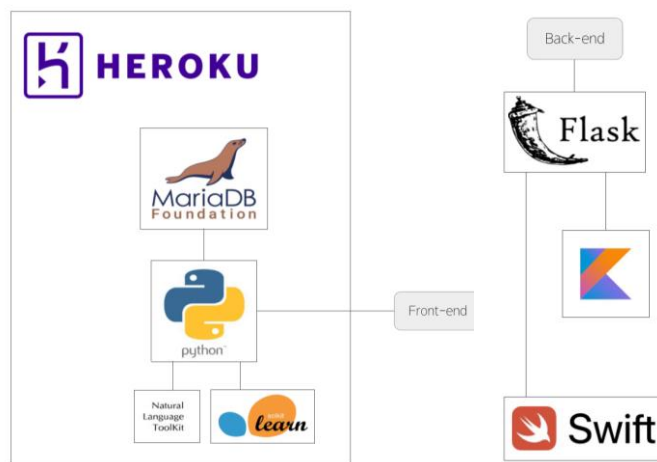
- 현재 시스템은 텍스트 기반의 자연어 처리를 통해 DB 를 구축하고 있으나, 거의 모든 중고 물품이 이미지 파일과 같이 등록되는 만큼, 이미지를 통해 상품의 종류와 상태를 파악할 수 있는 기능을 장기적으로 도입하는 것이 가능하며, 기존 인터넷상에 중복된 이미지가 있는지 점검하여 사진 도용을 통한 사기를 예방할 수 있음.

Appendices

시스템에 관한 세부적인 추가 내용을 기술합니다.

하드웨어 및 소프트웨어 requirements

- 앱 개발 프레임워크에 맞는 하드웨어가 요구됩니다. JavaScript 나 Native code 를 활용하는 경우가 많기 때문에, JSX 나 안드로이드의 최신 SDK 를 반영하는 수준의 하드웨어가 요구된다. 또한 추가적으로 디바이스와 DB 모두 실시간으로 온라인에 연결되어 있어야 합니다. DB 는 python 기반의 다양한 자연어 처리가 가능한 Heroku 서버와의 연동이 필수적이다.



[Image 4 : DB structure]

Database requirements

DB 를 구성하는 content 는 다음과 같다.

회원, User - 판매와 구매를 할 수 있는 일반 회원의 정보가 저장되는 엔티티

관리자, System Manager - 시스템 관리자의 정보가 저장되는 엔티티

상품 카테고리, Item Category - 상품 카테고리, 상품, 그 옵션과 썸네일 이미지가 저장되는 엔티티

상품, Item - 개별 상품과 그 세부사항, 상품의 현재 상태와 삭딜 구매자 목록이 저장되는 엔티티

거래, Trade - 거래를 진행중인 판매자와 구매자, 상품 ID 가 저장되는 엔티티

채팅, Chatting - 거래 중 채팅 내역이 저장되는 엔티티

카테고리 크롤링, Category Crawling - 여러 타 플랫폼에 최적화된 각각의 크롤링 방식과 크롤링한 상품들이 저장되는 엔티티

평균 가격, Average price - 각 상품의 기간별 평균 가격이 저장되는 엔티티

구매 요청 양식, Purchase request Form - 구매자들이 위시리스트에 등록한 상품들이 저장되는 엔티티

레포트, Report - 비정상적인 상품/거래/회원, 혹은 시스템 오류에 대한 신고들이 저장되는 엔티티

백업 DB, Back-up DB - 치명적인 시스템 오류에 대비해 중요한 정보를 백업해두는 엔티티

Index

[Image 1 : 기존 3 대 중고거래 어플의 최신 전략]

[Image 2 : 기존 3 대 중고거래 어플의 UI 비교]

[Image 3 : 시스템 UI 예시]

[Image 4 : DB structure]

[Diagram 1 : 회원가입 – Sequence diagram]

[Diagram 2 : 로그인 – Sequence diagram]

[Diagram 3 : 상품 등록 – Sequence diagram]

[Diagram 4 : 위시리스트 작성 및 수정 – sequence diagram]

[Diagram 5: Categorized search & Crawling - Event trace diagram]

[Diagram 6: 삭 deal - Event trace diagram]

[Diagram 7: 일반딜 - Relation diagram]

[Diagram 8: Crawling - Relation diagram]

[Diagram 9&10 : System managing -Class diagram & Relation diagram]

[Diagram 11 : 오류처리 – Event trace diagram]

[Diagram 12 : Front-end architecture – Class diagram]

[Diagram 13 : Database and Main Functionality Relationship]

[Diagram 14 : Entity Relationship Diagram for System Database]

[Diagram 15 Backend Architecture]

[Diagram 16 : Overall Architecture]