

# University Community based Group purchasing platform

Software Engineering Team 2

## Requirement Specification

Student Number	Name
2015310295	현수빈
2014312749	남기현
2014312978	변희성
2016311956	이준희
2016315178	이기요

# Contents

1. Preface.....	8
1.1. Objective.....	8
1.2. Readers.....	8
A. User Requirement Readers .....	8
B. System Requirement Readers .....	8
1.3. Document Structure .....	9
A. Introduction .....	9
B. Glossary.....	9
C. User Requirements Definition.....	9
D. System Architecture.....	9
E. System Requirements Specification.....	9
F. System Models.....	9
G. System Evolution.....	10
H. Appendices .....	10
I. Index .....	10
2. Introduction .....	11
2.1 Objective .....	11
2.2 Needs .....	11
2.3 System Overview.....	12
2.4 Expected Effect.....	13

3. Glossary.....	14
4. User Requirement Specification.....	15
4.1 Functional Requirement .....	15
A. Sign up.....	15
B. Login.....	15
C. 포인트 충전.....	15
D. 판매 상품 등록 .....	15
E. 검색.....	15
F. 공유.....	16
G. 상품 상세정보.....	16
H. 마이페이지.....	16
I. 상품 구매(결제).....	16
4.2 Non-functional Requirement .....	17
4.2.1 Product requirements .....	17
4.2.1.a Usability Requirement .....	17
4.2.1.b Efficiency requirement .....	17
4.2.1.c Dependability requirement .....	17
4.2.1.d security requirement.....	17
4.2.1.e Performance Requirement.....	18
4.2.2 Organizational requirements.....	18
4.2.2.a Environmental Requirement.....	18

4.2.2.b Operational Requirement .....	18
4.2.3 External Requirements .....	18
4.2.3.a Regulatory Requirement .....	18
4.2.3.b Safety/Security Requirement.....	19
5. System Architecture .....	20
5.1 Overall Architecture .....	20
5.2 Point management System.....	21
5.3 Item search System.....	22
5.4 Group purchase System .....	23
6. System Requirement Specification .....	25
6.1 Functional Requirement .....	25
A. Sign up .....	25
B. Log in.....	27
C. 포인트 충전 .....	28
D. 판매 상품 등록.....	29
E. 검색.....	30
F. 공유.....	31
G. 상품 상세 정보.....	32
H. 마이 페이지.....	33
I. 상품 구매.....	34
6.2 Non-functional requirement .....	35

6.2.1 Product requirements .....	35
6.2.1.a Usability Requirement .....	35
6.2.1.b Efficiency requirement .....	35
6.2.1.c Dependability requirement .....	35
6.2.1.d security requirement.....	35
6.2.1.e Performance Requirement.....	36
6.2.2 Organizational requirements.....	36
6.2.2.a Environmental Requirement.....	36
6.2.2.b Operational Requirement.....	36
6.2.3 External Requirements.....	36
6.2.3.a Regulatory Requirement .....	36
6.2.3.b Safety/Security Requirement.....	37
6.3 Scenario example.....	37
6.3.1 구매 Scenario .....	37
6.3.1.a Initial Assumption .....	37
6.3.1.b Normal flow of event .....	37
6.3.1.c What can go wrong.....	37
6.3.1.d System state on completion .....	38
6.3.2 검색 scenario .....	38
6.3.2.a Initial Assumption.....	38
6.3.2.b Normal flow of events.....	38

6.3.2.c What can go.....	38
6.3.2.d System state on completion.....	38
7. System Models.....	39
7.1 Context Model.....	39
7.1.a Context Model.....	39
7.2 Interaction Model.....	40
7.2.a Use case Diagram.....	40
7.2.b Tabular Description.....	41
7.3 Sequence diagram.....	44
7.3.a 검색.....	44
7.3.b Log-in.....	45
7.3.c 구매.....	46
7.4 Behavioral Models.....	47
7.4.a Sign up Event-Driven Modeling.....	47
8. System Evolutions.....	48
8.1. Popular product recommendation.....	48
8.2 Real-time tracking of delivery status.....	49
8.3 OAuth implementation.....	50
8.4 Sorting and filtering.....	50
9. Appendices.....	51
9.1. Objectives.....	51

9.2. DB requirements .....	51
9.3. User-system requirements .....	52
9.4. Development process .....	52
10. Index .....	53
10.1 Figure .....	53
10.2 Table .....	53
10.3 Diagram .....	54
11. References.....	54

# 1. Preface

## 1.1. Objective

Preface 는 본 문서의 예상되는 독자들과 전반적인 구조, 그리고 각 부분의 역할에 대해 설명한다. 또한, 각 버전 관리 정책, 버전 변경 기록, 그리고 문서의 변경사항들을 설명한다.

## 1.2. Readers

본 Requirement Specification 은 reader 에 따라 User Requirement 와 System Requirement 로 구성되어 있다.

### A. User Requirement Readers

시스템이 사용자에게 제공하는 서비스에 대하여 설명한다. Non-functional system requirements 도 설명한다. 도식과 자연어 등을 사용해 사용자가 이해하기 쉽게 설명한다.

### B. System Requirement Readers

개발자의 관점에서 본 프로젝트에서 개발해야 하는 시스템의 특정 기능이 어떻게 구현되어야 하는지 설명한다. Non-functional system requirements 도 설명한다. 시스템 개발에 참여하는 모든 developer 를 독자로 설정한다



## 1.3. Document Structure

### A. Introduction

현재 개발하고자 하는 시스템의 필요성과 해당 시스템을 둘러싼 needs 를 설명하고 , 시스템의 대략적인 구조와 기능에 대해 설명한다. 또한 시스템을 개발을 통해 target user 가 얻을 수 있는 기대효과를 설명한다.

### B. Glossary

해당 시스템의 domain 의 전문적인 용어들을 설명하고 용어가 사용되는 문맥에 대해서 간략하게 서술한다. 시스템의 개발자에 국한되지 않고 시스템의 사용자 또한 독자로 고려해 전문적인 용어들을 이해하기 쉬운 용어로 재정의한다.

### C. User Requirements Definition

시스템의 functional, non-functional requirements 들을 서술한다. 사용자에게 친근한 자연어로 기술하고, 쉬운 이해를 위해 다이어그램과 같은 도식을 함께 사용하여 설명한다.

### D. System Architecture

예상되는 시스템의 구조를 보여준다. 각 시스템의 architecture 의 기능도 설명한다. 또한, sub-system 이 전체적인 시스템의 분포에 어떤 식으로 할당되었는지 설명한다.

### E. System Requirements Specification

User Requirements Definition 에 간략히 서술되어 있는 요구사항을 기반으로, functional requirement, non-functional requirement 그리고 기타 요구사항을 자세히 서술한다. 더욱 세세한 requirement 분석과 기능의 대략적인 흐름을 알기 위해 scenario 를 이용한다.

### F. System Models

시스템의 component 와 system 의 관계, 그리고 system 을 둘러싼 environment 간의 관계를 설명한다. 효과적인 설명을 위해 다이어그램 같은 도식을 함께 사용한다.

## G. System Evolution

시스템 개발과정에서 발생할 수 있는 한계점과 예상되는 change 에 대해 서술한다. 시스템의 구조에 영향을 주는 change 와 한계점을 미리 설명함으로써 후에 발생할 문제의 대응 방안을 세운다.

## H. Appendices

개발되는 시스템에서 참고자료로 사용된 문서나 정보들을 기술한다. Hardware 와 database 에 대해서도 설명한다.

## I. Index

본 문서에서 사용된 그림, 표, 다이어그램 등의 색인을 기술한다.

## 2. Introduction

### 2.1 Objective

본 시스템이 반영하고 있는 needs 와 필요성에 대해서 서술한다. 또한, 시스템의 대략적인 기능과 본 시스템의 기능이 target user 에게 주어지는 기대효과를 설명한다.

### 2.2 Needs

최근 중국에서는 '타오바오'와 '핀뒤티' 같은 공동 구매 어플이 큰 인기를 끌고 있다. 더욱이 '핀뒤티'는 중국에서 가장 많이 사용되던 '타오바오'를 제치고 신형 강자로 떠올랐다. '핀뒤티'의 가장 큰 장점은 저가 용품을 공동 구매를 통해 더 싼 가격으로 구매한다는 점이다. 구매하고자 하는 이용자가 특정 수를 넘으면 가격이 내려가는 기능을 내세우고 있는데 공동구매의 가장 큰 문제인 구매 인원 모집을 SNS 를 이용해 공유를 활성화함으로써 해결하였다. 이러한 공동구매 어플의 성공은 공동구매의 이점을 잘 보여준다.

11월 플랫폼별 월활성이용자수(MAU) 규모 증감폭

	동기대비 증감폭(%)	활성이용자수 (억명)
타오바오(淘寶)	-1.4	5.4
핀뒤티(拼多多)	205.2	2.8
징둥(京東)	18.5	2.6
웨이핀후이(唯品會)	14.8	0.9
덴마오(天貓)	-4.1	0.8
샤오홍수(小紅書)	142.5	0.4
쑤닝이거우(蘇寧易購)	-2.7	0.2
모구제(蘑菇街)	-49.6	0.1

[자료=지광빅데이터(極光大數據)] [그래픽=홍종현 미술기자]  NEWS PIM

Figure 1: MAU 규모 증감폭

공동 구매는 현재 한국에서도 상당히 많이 이용되고 있지만 이렇다 할 만한 공동 구매 전문 사이트를 찾기 힘들다. 대부분은 공동구매는 구매 인원 모집이 어려워 카페나 SNS 를 통해 이루어진다. 카카오톡에서도 '톡탈'이라는 2인 공동구매 기능을 새로 선보였는데 2명만 모으면

낮은 가격에 물건을 구매할 수 있으며 이 점은 공동 구매의 가장 큰 문제인 인원 모집을 해결하고자 했다. 하지만 이러한 기능은 공동 구매의 가장 큰 장점인 구매자 수와 가격 간의 관계를 살리지 못했다. 구매자를 2 인으로 제한하고 가격 하락이 그다지 효과적이지 않아 구매자가 느낄 수 있는 공동구매의 매력이 반감된다. 또한 판매자가 겪을 수 있는 다른 문제점은 배송이었다. 공동구매라 하더라도 각 구매자의 주소로 개별 배송을 해야 하는데 이 점을 시스템 개발하는데 가장 큰 어려움으로 뽑았다. 공동 구매 수량이 늘어날 수록 배송 비용은 더 늘어나고 특히 물건의 크기와 상관없이 개별 배송을 하는 것은 판매자에게도 불편한 점이다. 이렇듯 공동구매의 이점을 인지하고 있지만 여러가지의 문제점으로 공동 구매 사이트가 발전되지 않았다. 구매자들은 더 낮은 가격에 합리적으로 구매할 권리가 있으며 그렇기 때문에 공동 구매 사이트의 발전은 필요하다.

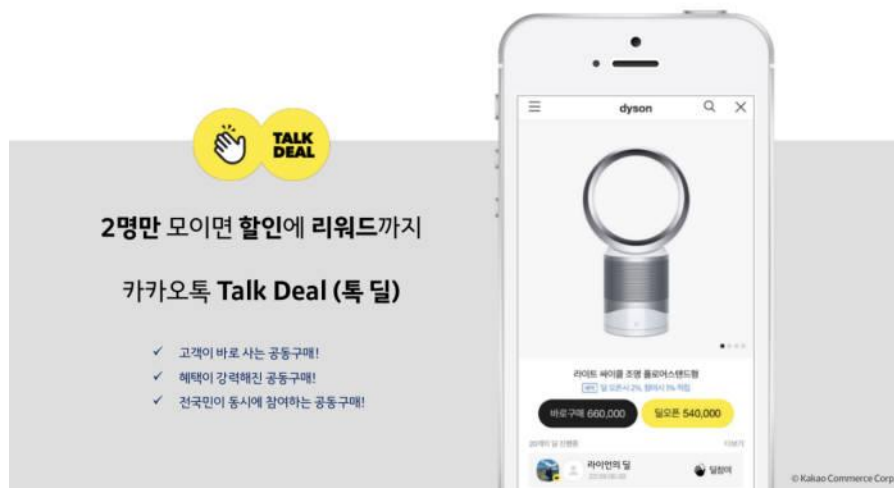


Figure 2: 카카오톡 Talk Deal

## 2.3 System Overview

본 시스템은 대학 커뮤니티를 기반으로 같은 학교 학생들끼리 공동구매를 통해 도매 물품을 원하는 만큼 분할 구매할 수 있는 환경을 제공한다. 도매 물품의 경우 가격은 저렴하지만 개인이 구매하기에는 수량이 많은 경우가 많다. 앞서 2.2 Needs 에서 서술한 바와 같이 중국 시장에는 '핀뒤뒤'와 같은 공동구매 플랫폼이 잘 활성화되어 있으나 국내에는 도매 물품을 개개인에게 분할로 판매하는 서비스는 존재하지 않는다.

구매자는 판매자가 올린 물품에 공동구매 참여를 하여 일반 구매가격보다 저렴하게 구입을 할 수 있으며, 배송비 또한 절감되는 효과를 받을 수 있다. 판매자의 경우 해당 플랫폼이 활성화됨에 따라 새로운 고객을 계속해서 확보해 갈수 있다는 장점이 있다.

## 2.4 Expected Effect

### A. 구매자

구매자는 매일 소모되는 생필품을 보다 싼 가격에 구매할 수 있다. 휴지나 즉석밥 같은 자주 소모되고 고갈되지만 자취생들에게는 필수인 제품들의 구매를 낮춘다면 구매자들은 조금 더 효과적인 재정 관리가 가능하다. 또한, 대학교 별 커뮤니티라는 점을 이용해 빠른 공유를 기대할 수 있다.

### B. 판매자

판매자는 공동구매 시 가장 큰 걸림돌이 되는 개별 배송의 문제를 해결하였다. 생필품처럼 자잘한 물건을 개별 포장 시 많은 노력과 시간이 허비되는데 미리 설정된 특정 배송 장소에 제품들을 배송하면 구매자가 나눠가지는 형식이니 자원을 아낄 수 있다. 또한 판매자는 대량 주문을 통해 금전적 효과를 기대할 수 있다.

### 3. Glossary

Term	Description
Login	사용자가 ID와 PW를 이용해 우리 서비스에 접근할 수 있게 하는 과정이다. 일반적인 웹 사이트 로그인과 같은 의미이다.
반응형 웹페이지	단말의 종류와 화면 크기에 실시간 반응하여 웹 페이지를 최적화된 화면 구성으로 보여주는 웹이다.
Database	Database란 정보를 저장하는 보관 장소로 원하는 데이터를 빠르게 검색할 수 있는 장점이 존재한다.
키워드	본 문서에서 키워드는 해당 제품의 특성을 나타내며, 이를 통하여 소비자가 상품 검색을 쉽게 할 수 있도록 도와준다.
Point	본 플랫폼에서 소비자가 사용하는 일종의 가상화폐이다.
Frontend	User-side에서 동작하는 웹페이지를 지칭한다.
Backend	주로 서버나 database 관련된 작업을 말한다.
API	Application Programming Interface의 줄임말. 프로그램 사이의 인터페이스 역할을 한다.
oAuth	OAuth는 인터넷 사용자들이 비밀번호를 제공하지 않고 다른 웹사이트 상의 자신들의 정보에 대해 웹사이트나 애플리케이션의 접근 권한을 부여할 수 있는 공통적인 수단으로서 사용되는, 접근 위임을 위한 개방형 표준이다..

Table 1: Glossary

## 4. User Requirement Specification

### 4.1 Functional Requirement

#### A. Sign up

사용자에게 필요한 정보를 요청하는 기능. 구매자의 경우 자신의 나이 및 성별을 입력하며 학교 이메일 인증을 통해 자신이 학생이라는 것을 인증함으로써 구매자 신분을 획득한다. 판매자의 경우 판매자 전용 웹 페이지가 존재하며 구매자와 마찬가지로 가입을 통해 판매자 권한을 획득한다.

#### B. Login

사용자가 로그인을 통해 회원으로서 사이트에 접속하는 과정. 아이디 및 패스워드를 입력해 정보가 맞으면 각자의 권한(구매자, 플랫폼 관리자, 판매자)에 따라서 사이트에 접속하게 된다.

#### C. 포인트 충전

실제 결제를 통해 현금을 해당 플랫폼 내 재화인 포인트 형식으로 전환할 수 있다. 결제창을 통해 포인트 충전이 가능하며 포인트는 마이 페이지 및 상단에서 확인이 가능하다.

#### D. 판매 상품 등록

판매자 권한을 가진 사람은 자신이 팔고자 하는 상품을 등록할 수 있다. 이 때 상품의 이름, 사진, 등록자, 상품의 가격, 수량에 따른 할인율, 최소 배달 개수 및 상품 공동구매 기간을 설정할 수 있다. 판매 상품이 등록되면 메인 창 및 검색 창에서 정렬될 수 있다.

#### E. 검색

사용자는 이 기능을 통하여 상품을 검색할 수 있다. 검색 조건은 이름, 카테고리, 판매 회사 등 다양한 조건이 될 수 있으며 검색 시 기간 순으로 재정렬이 가능해야 한다. 상품을 누르면 해당 상품의 자세한 정보를 담은 페이지로 연결된다.

## F. 공유

공동구매 사이트에서 내가 구매한 상품을 다른 사람들과 공유할 수 있는 기능이다. 구매가 확정된 후에 '공유하기' 또는 '메인 화면으로' 창이 뜨는데, '공유하기' 항목을 선택할 경우 카카오톡 API 를 통하여 구매내역을 공유할 수 있다.

## G. 상품 상세정보

사용자가 특정 상품을 클릭하여 상품에 대한 자세한 정보를 볼 수 있는 기능이다. 사용자는 메인 메뉴에서 상품을 선택하거나 검색한 후, 상품 페이지로 들어와 판매자가 등록한 상품에 대한 자세한 설명을 볼 수 있다. 상품 상세정보에는 또한 '공동구매 남은 시간', '상품 가격' 등을 확인할 수 있다. 이후 공동구매 버튼을 통해 공동구매를 진행할 수 있다.

## H. 마이페이지

My page 에서 로그인 한 학생들은 개인 정보 및 구매 상품 내역, 구매한 상품 중 현재 공동구매 중인 상품 내역을 보여준다. 남은 point 를 확인할 수 있는 기능도 있으며 point 가 부족할 경우 B.포인트 충전으로 연결해준다. 또한 실시간 알림 기능을 통하여 구매가 확정되거나 수량부족으로 인해 구매가 연기 / 취소 될 시 유저에게 실시간으로 알려준다.

## I. 상품 구매(결제)

사용자가 상품 상세정보를 확인한 후, 상품을 구매할 수 있는 기능이다. 여기서 사용자는 먼저 공동구매 진행 내역 및 가격, 남은 포인트를 확인한 후, 상품을 원하는 만큼 장바구니에 담는다. 이후, '구매하기' 버튼을 누를 시 구매한 만큼 point 를 차감한다. point 가 부족할 경우, B.포인트 충전으로 연결해준다. 결제 완료 시 주문이 완료되었는지 확인하고 메인 메뉴로 돌아갈 수 있다.



## 4.2 Non-functional Requirement

### 4.2.1 Product requirements

#### 4.2.1.a Usability Requirement

해당 시스템은 사용자의 공동구매 과정에 편리한 기능을 제공하여야 한다. 현재 참여중인 공동구매 현황, 관심있는 물품에 대한 즐겨찾기 기능, 카테고리를 설정하여 원하는 상품을 찾을 수 있는 기능을 제공하여야 한다. 플랫폼 이용자가 쉽게 이해할 수 있도록 웹사이트의 구성요소들을 적절히 배치하여야 하고 개인정보에 관련된 수정사항도 쉽게 변경할 수 있도록 마이페이지 기능도 제공하여야 한다. 공동구매의 특성상 참여할 수 있는 기간이 정해져 있으므로 공동구매 기간이 종료되었을 경우 사용자에게 관련 정보를 전달해 줄 수 있는 알림 기능도 제공한다.

#### 4.2.1.b Efficiency requirement

해당 시스템은 실시간으로 공동구매 현황을 업데이트하여 사용자에게 제공할 의무가 있으므로 서버와의 통신에 있어서 대기 시간을 최소화하여야 한다. 대기시간을 최소화하기 위하여 서버와 효율적인 통신을 해야 한다.

#### 4.2.1.c Dependability requirement

해당 시스템은 사용자가 구매한 상품에 대한 목록을 제공하여 어떤 물건을 구매하려 하거나(장바구니), 구매했는지를 확인할 수 있는 기능을 제공해야 한다. 또한, 장바구니 목록이나 구매 완료 목록은 임의로 손실되거나 변경되어서는 안 된다.

#### 4.2.1.d security requirement

해당 시스템은 보안성이 뛰어나야 한다. 관리자 계정에 다른 불법적인 접근을 차단해야 한다. 또한 거래 정보가 유출되어서는 안 된다.

#### 4.2.1.e Performance Requirement

해당 시스템은 PC / 휴대용 기기 등 어떤 기기에서도 주문을 할 수 있는 웹 서비스이기 때문에, 이용 환경에서 성능 차이가 있어서는 안 된다. 판매자가 물건의 정보를 업로드 하거나, 구매자가 물건을 구매할 때 performance 측면에서 문제가 없어야 한다. 구매하는 사람이 많아 트래픽이 급증하더라도 적당 선에서 제대로 된 처리속도를 보장할 수 있어야 한다.

### 4.2.2 Organizational requirements

#### 4.2.2.a Environmental Requirement

해당 시스템은 웹 페이지로만 구현되지만, 사용자가 PC / 휴대용 기기 등 모든 상황에서 쉽게 접속할 수 있는 환경을 제공해야 한다. 구현 시 이러한 요구사항을 충족시키기 위해 반응형 웹페이지를 구현한다.

#### 4.2.2.b Operational Requirement

해당 시스템은 구매자에게 판매 중인 상품의 정보를 보여주고 해당 상품의 남은 시간/남은 수량을 실시간으로 알려 준다. 판매자에게는 상품 등록, 수정 및 구매자의 주문 내역을 확인할 수 있는 인터페이스를 제공해야 한다.

### 4.2.3 External Requirements

#### 4.2.3.a Regulatory Requirement

해당 시스템은 회원가입이 필요하기 때문에 사용자에게 이름, 소속 학교, 휴대폰 번호 등 개인정보를 요구한다. 회원가입을 위해서는 개인정보 제공에 대한 사용자의 동의를

받아야 한다. 플랫폼 제공자는 고객의 개인정보를 철저하게 관리하고 보호해야할 의무가 있다. '주민번호 수집 원칙적 금지(공포 '13.8.6, 시행 '14.8.7)' 에 따라 회원가입시 사용자의 주민등록번호를 수집해서는 안 된다. 따라서 사용자 인증의 경우 휴대폰 인증 및 이메일 인증을 이용한다. 판매자의 경우 사업자 등록번호 조회 등을 통하여 관련된 법규에 적합한지 확인하고 가입을 허가한다.

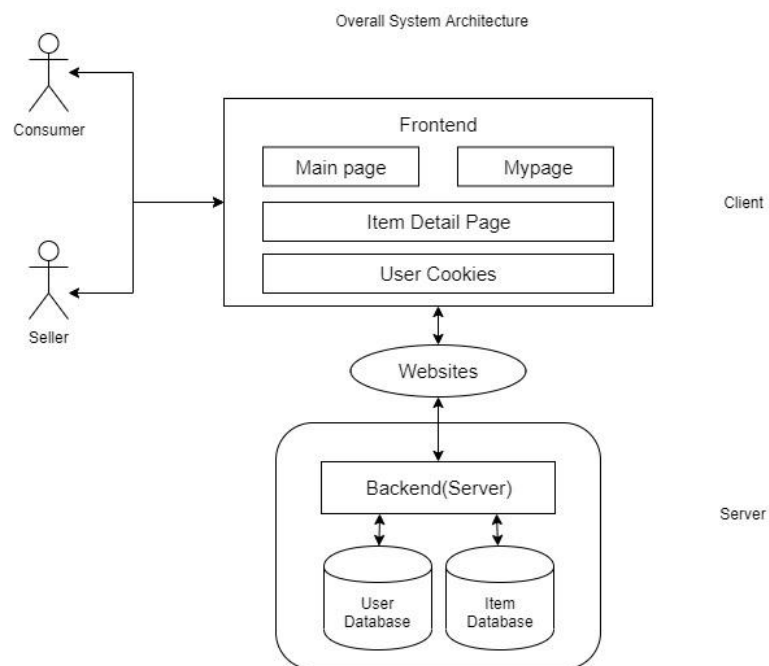
#### 4.2.3.b Safety/Security Requirement

해당 시스템은 거래를 제공하는 서비스이므로 개인정보가 외부로 유출되어서는 안된다. 관리자의 허가 받지 않은 외부/내부의 접근으로부터 안전해야 하며, 판매자의 업로드 오류나 거래할 때 중복 결제 등의 오류가 발생하지 않도록 해야 한다. 중요한 데이터(특히, 시스템 내의 재화인 포인트)들은 암호화하여 저장해야 하며, 거래 시 임의로 변경되어서는 안된다.

## 5. System Architecture

해당 챕터에서는 전체적인 시스템의 구조에 대해 기술한다. 시스템 전반에 걸친 개괄적인 구조와 이를 구성하는 각 서브시스템들의 구성, 서브시스템이 동작하는 flow-chart 등을 이용하여 시스템의 동작 방식에 대한 설명을 한다. 각 구조에 대해 독자의 이해를 돕고자 다이어그램을 첨부하였다.

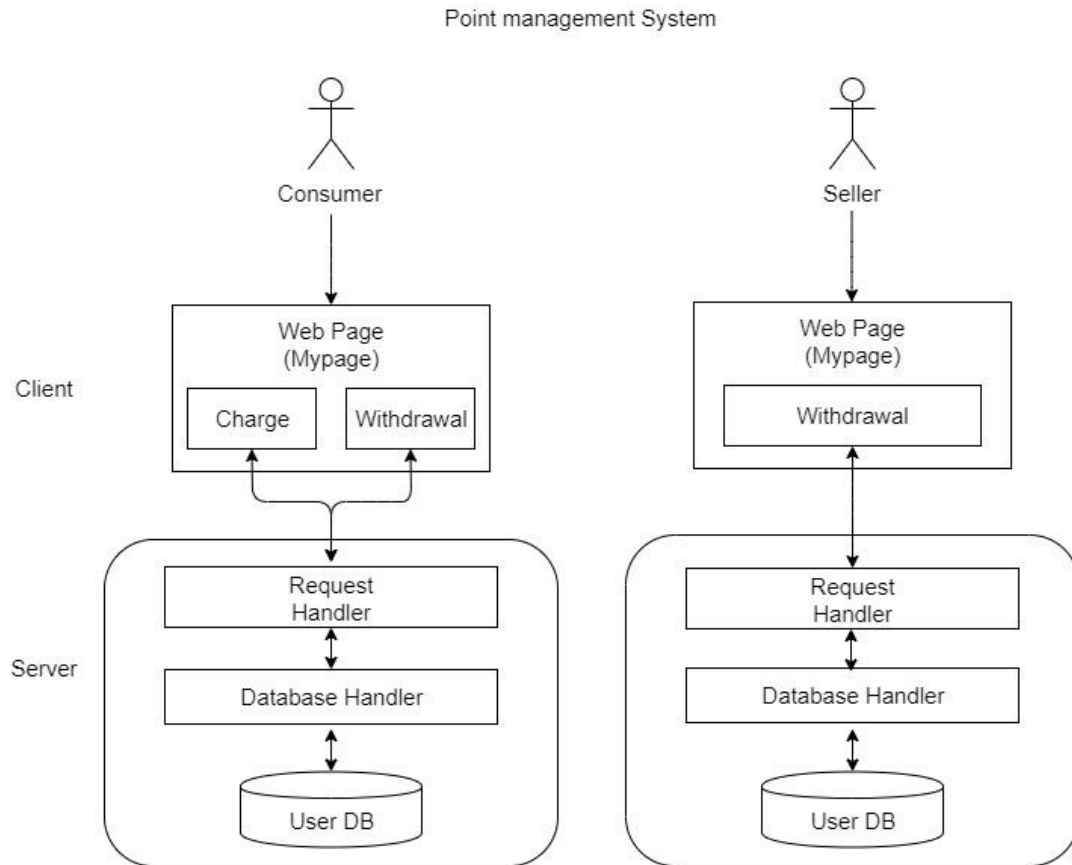
### 5.1 Overall Architecture



**Diagram 1: Overall Architecture**

Frontend는 웹 기반 사이트로써 사용자가 공동구매를 진행함에 있어서 편리하게 이용할 수 있도록 도와준다. 웹사이트는 크게 main page, mypage, item detail page 로 나뉘고 유저로부터 입력 받은 request를 서버로 전송해준다. Backend(서버)에서는 유저의 요청에 따라 데이터 베이스에 알맞은 쿼리를 전송하여 데이터 베이스를 업데이트 해주고 그에 따른 응답을 frontend로 보내어 실시간으로 유저에게 정보를 전달해 준다.

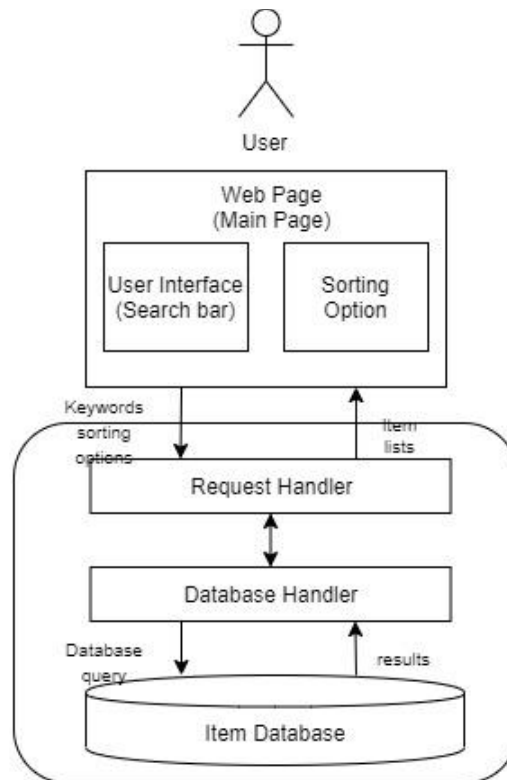
## 5.2 Point management System



**Diagram 2: Point management System**

우리가 개발하고자 하는 웹 서비스는 사이트내 가상 화폐로 포인트를 이용한다. 따라서 소비자(consumer)는 공동구매에 참여하기 전 먼저 포인트를 충전하고 이를 활용하여 구매하고자 하는 물품에 입찰을 할 수 있게 된다. 잔여 포인트에 대해 소비자가 출금을 원할 경우가 생길 수 있으므로 출금에 대한 처리도 가능하여야 한다. 판매자(seller)는 물품 판매의 수익으로 포인트를 얻게 된다. 따라서 이를 인출할 수 있는 기능만 필요하게 된다. 서버는 출금 또는 충전에 대한 요청을 받아서 데이터 베이스에 적절한 쿼리를 전송하고 업데이트 해줌으로써 포인트 관리를 할 수 있게 된다.

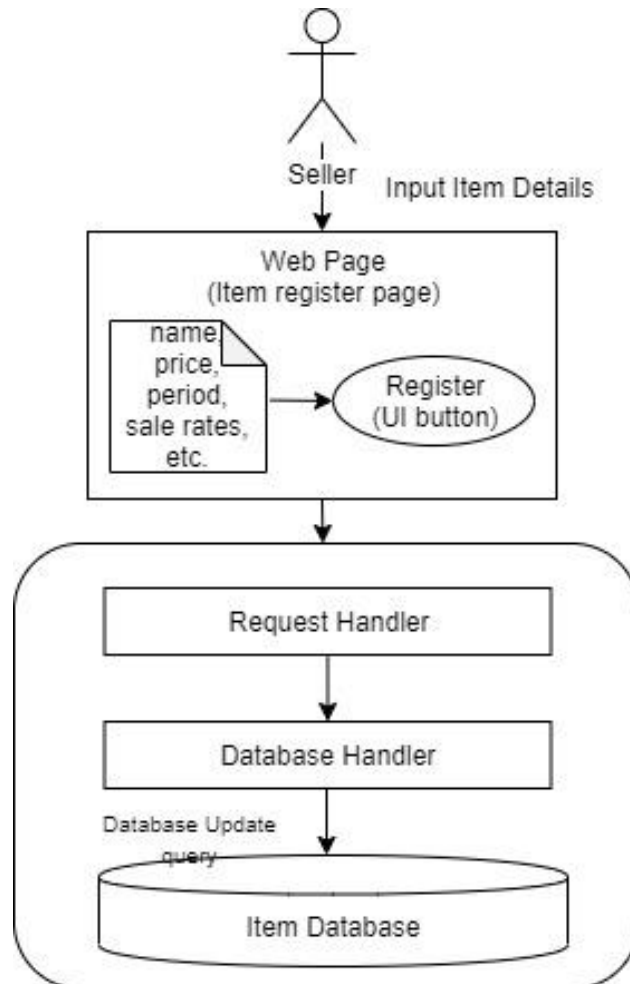
### 5.3 Item search System



**Diagram 3: Item search System**

소비자가 공동구매가 진행중인 물품들에 대하여 특정 물품을 쉽게 찾을 수 있도록 도와줄 수 있는 검색 시스템이 필요하다. 웹사이트의 메인 페이지 검색창을 통하여 찾고자 하는 물품에 대한 키워드를 입력하고 정렬 옵션을 설정하면 request handler가 이를 통해 데이터 베이스에 해당되는 쿼리를 전송해주고 쿼리에 따라 적절한 물품 리스트를 유저에게 return하여 보여준다.

## 5.4 Group purchase System



**Diagram 4: Group purchase System 1**

판매자는 위와 같이 판매할 물품을 등록할 수 있다. 판매할 물품의 이름, 가격, 공동구매를 진행할 기간 그리고 참여율에 따른 할인율을 입력하고 등록 버튼을 클릭하면 이에 따른 적절한 operation을 통하여 item 데이터베이스에 상품을 업데이트 해준다.

아래의 다이어그램은 소비자가 공동구매를 참여하는 과정에 대한 순서도를 나타낸 것이다.

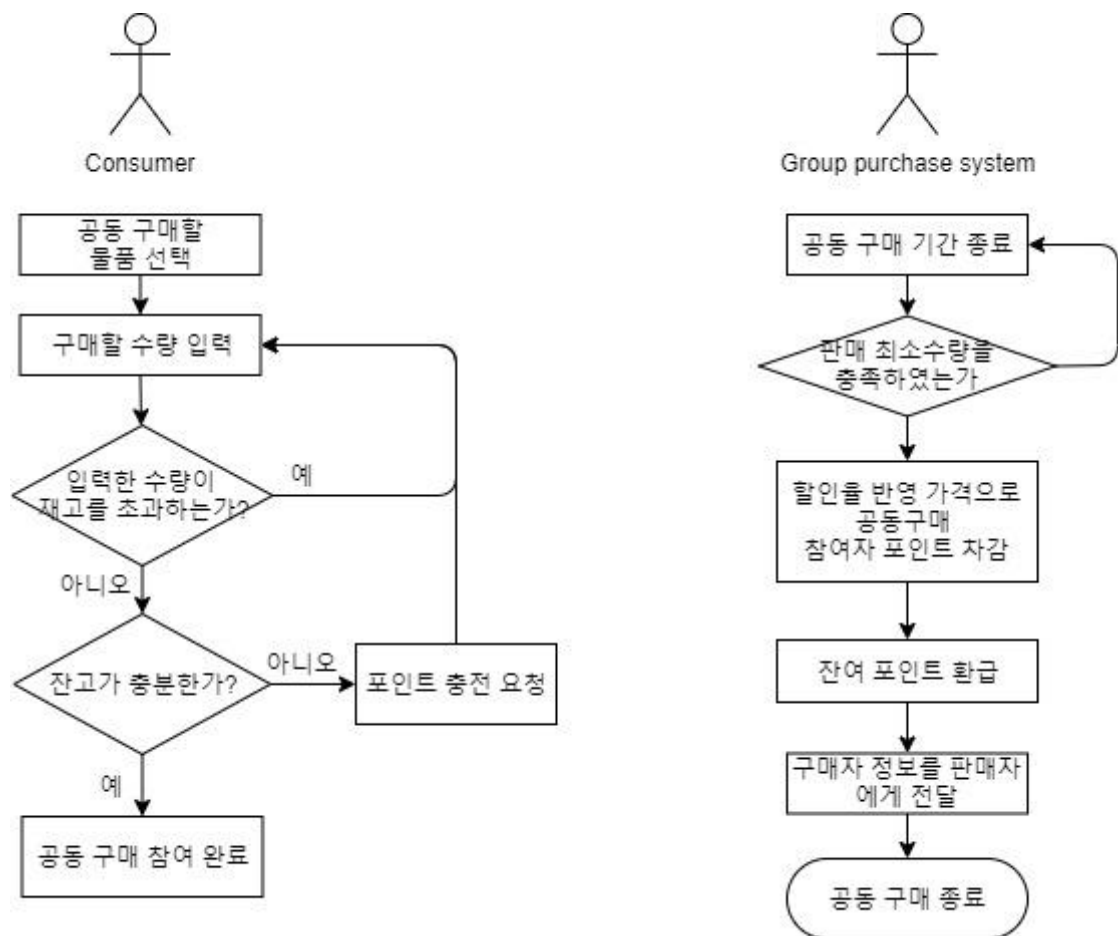


Diagram 5: Group purchase System flow chart



## 6. System Requirement Specification

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

### 6.1 Functional Requirement

#### A. Sign up

<b>이름</b>	Sign up
<b>설명</b>	사용자가 이 사이트의 회원임을 등록하는 과정
<b>Inputs</b>	판매자/구매자 신분 ID: 사용자가 이용할 아이디 Password: 사용자가 사용할 패스워드 <구매자일 시> 학교 선택: 어느 학교를 다니는지 선택 이메일: 이메일 인증을 통해 선택한 학교에 다니고 있음을 입증
<b>Outputs</b>	인증된 회원 정보

<b>Action</b>	아이디 중복 확인 및 비밀번호가 적합한지 확인하고 이메일 인증이 완료되었을 경우 회원의 내용을 DB 에 저장. 이러한 조건을 만족하지 못했을 시 에러 메시지 출력.
<b>Requirements</b>	아이디 중복 확인 및 비밀번호가 적합한지 확인. 구매자 신분일 경우 이메일 인증으로 학생 인증.
<b>Pre-condition</b>	인증 받지 못한 회원은 로그인 창에서 막힘
<b>Post-condition</b>	인증 받은 회원은 사이트에 접근 가능
<b>Side effects</b>	-

**Table 2: Functional Requirement – Sign Up**

B. Log in

<b>이름</b>	Log in
<b>설명</b>	사용자가 메인 페이지에 접근하는 과정
<b>Inputs</b>	ID: 사용자가 이용할 아이디 Password: 사용자가 사용할 패스워드
<b>Outputs</b>	신분에 맞는 사이트
<b>Action</b>	ID 및 Password 입력 후 로그인 버튼을 누르면 DB 에서 확인, 일치하는 정보가 있으면 통과시키고 아니면 에러 메시지와 함께 통과시키지 않음.
<b>Requirements</b>	-
<b>Pre-condition</b>	로그인 창
<b>Post-condition</b>	성공 시 다음 창, 실패 시 그대로.
<b>Side effects</b>	-

Table 3: Functional Requirement – Log in

### C. 포인트 충전

<b>이름</b>	포인트 충전
<b>설명</b>	사용자가 사이트에 사용할 재화를 충전한다.
<b>Inputs</b>	사용자가 충전할 금액
<b>Outputs</b>	DB 에 사용자가 충전한 금액 갱신
<b>Action</b>	사용자가 충전할 금액을 입력할 시 실제로 그 금액만큼 충전이 되어 DB 에 저장된다.
<b>Requirements</b>	-
<b>Pre-condition</b>	충전되기 전 금액으로 표시
<b>Post-condition</b>	충전된 후 금액으로 표시
<b>Side effects</b>	-

**Table 4: Functional Requirement – Point Charge**

#### D. 판매 상품 등록

<b>이름</b>	판매 상품 등록
<b>설명</b>	자신이 팔고자 하는 상품 등록
<b>Inputs</b>	상품의 이름, 상품 사진, 등록자, 상품의 가격, 수량에 따른 할인율, 최소 배달 개수 및 상품 공동구매 기간 등의 상품 상세 정보
<b>Outputs</b>	DB 에 판매자가 등록한 상품 상세정보 저장
<b>Action</b>	등록이 정상적으로 되면 메인 화면으로 이동한다. 이 상세 정보는 메인 창 및 검색 창에 상품에 정렬되어 표시되며 판매자의 경우 마이 페이지에 등록된다.
<b>Requirements</b>	신분이 판매자이어야 함
<b>Pre-condition</b>	-
<b>Post-condition</b>	상품이 검색창에 정렬 및 판매자 마이 페이지에 표시
<b>Side effects</b>	-

**Table 5: Functional Requirement – Item register**

E. 검색

<b>이름</b>	검색
<b>설명</b>	검색한 상품을 정렬 설정대로 보여주는 과정
<b>Inputs</b>	찾고자 하는 물품 이름
<b>Outputs</b>	물품 이름이 들어간 상품 정렬
<b>Action</b>	물품 이름 입력 후 검색 버튼을 누를 시 이름에 맞는 상품 정렬된 채로 정렬. 이 물품의 경우 공동구매 남은 기간 순이나 최근 등록 순으로 정렬 가능.
<b>Requirements</b>	-
<b>Pre-condition</b>	정렬 전 메인 페이지
<b>Post-condition</b>	물품이 정렬된 페이지
<b>Side effects</b>	같은 이름으로 입력하지 않을 시 물품을 못 찾을 수 있다.

**Table 6: Functional Requirement – Search items**

F. 공유

<b>이름</b>	공유
<b>설명</b>	구매한 상품을 카카오톡을 통해 공유
<b>Inputs</b>	공유하고자 하는 물품
<b>Outputs</b>	카카오톡 방에 구매한 상품 상세 내역 공유
<b>Action</b>	물품 구매 완료 후 공유 버튼 표시,
<b>Requirements</b>	구매가 완료된 후에 그 물품에 대해 공유 가능
<b>Pre-condition</b>	정렬 전 메인 페이지
<b>Post-condition</b>	검색대로 물품이 정렬된 페이지
<b>Side effects</b>	같은 이름으로 입력하지 않을 시 물품을 못 찾을 수 있다.

**Table 7: Functional Requirement - Share**

G. 상품 상세 정보

<b>이름</b>	상품 상세 정보
<b>설명</b>	상품의 상세한 정보를 볼 수 있는 창
<b>Inputs</b>	상품 정렬 창에서 상품 내역 클릭
<b>Outputs</b>	상품의 이름, 상품 사진, 등록자, 상품의 가격, 수량에 따른 할인율, 최소 배달 개수 및 상품 공동구매 기간 등의 상품 상세 정보 등 상품의 자세한 내역 출력 및 공동구매 버튼 페이지
<b>Action</b>	상품 정렬 창에서 상품을 클릭하면 상품의 상세 내역들을 확인할 수 있다. 공동구매 버튼을 통해 구매창으로 이동할 수 있다.
<b>Requirements</b>	-
<b>Pre-condition</b>	상품 정렬 창
<b>Post-condition</b>	상품 상세내역 표시 창
<b>Side effects</b>	-

Table 8: Functional Requirement – Item details



## H. 마이 페이지

<b>이름</b>	마이 페이지
<b>설명</b>	개인에 대한 정보를 보여주는 페이지
<b>Inputs</b>	마이 페이지 이동 버튼 클릭
<b>Outputs</b>	개인 정보 및 구매자의 경우 구매 상품 내역, 구매한 상품 중 현재 공동구매 진행 중인 상품 내역, 남은 포인트 확인, 포인트 충전 버튼, 실시간 알림 창 보여주는 페이지
<b>Action</b>	마이 페이지 이동 버튼을 클릭하면 바로 마이 페이지로 연결되어 outputs 에 나오는 상세 내역들을 조회 가능하다.
<b>Requirements</b>	자신과 관련된 정보들이어야만 함
<b>Pre-condition</b>	마이 페이지 제외 창
<b>Post-condition</b>	마이 페이지 창
<b>Side effects</b>	-

**Table 9: Functional Requirement – My page**

## I. 상품 구매

<b>이름</b>	상품 구매
<b>설명</b>	상품 구매를 하는 페이지
<b>Inputs</b>	상품 상세 정보에서 공동구매 클릭
<b>Outputs</b>	구매할 상품의 개수 및 가격, 남은 포인트가 확인 가능한 창, 구매 버튼 창
<b>Action</b>	상품 상세 정보에서 공동구매 버튼을 클릭할 시 구매 창이 뜨며, 여기서 상품의 개수를 지정한 후 상품 구매를 할 수 있다. 이 때 자신이 보유한 포인트가 구매할 상품 가격보다 높다면 구매 완료 창이 뜨고 아닐 시에는 포인트 충전으로 연결하겠냐는 문구를 띄운다.
<b>Requirements</b>	자신의 포인트가 구매하고자 하는 상품의 총 가격보다 커야 함
<b>Pre-condition</b>	상품 상세 정보 창
<b>Post-condition</b>	상품 구매 창
<b>Side effects</b>	-

Table 10: Functional Requirement - Purchase

## 6.2 Non-functional requirement

### 6.2.1 Product requirements

#### 6.2.1.a Usability Requirement

해당 시스템은 공동구매/즐거찾기/검색 기능을 제공한다. 개인정보 수정, 포인트 충전 등의 서비스가 있는 마이페이지 기능을 제공한다. 공동구매 기간 종료 알림 기능도 제공한다. 이런 모든 기능들은 처음 온 사람들도 편하게 사용할 수 있게 사용자에게 직관적인 UI/UX 를 제공해야 한다.

#### 6.2.1.b Efficiency requirement

서버 대기 시간을 최소화하기 위하여 서버에 접근 가능한 구매자의 system 수를 적당한 수의 인원으로 한정한다. 인원이 초과하였을 경우 대기 시간을 부여한다. 서버와 통신을 주고받을 데이터를 최적화하여 불필요한 지연을 없애야 한다.

#### 6.2.1.c Dependability requirement

사용자에게 장바구니 기능 / 구매했는지 확인할 수 있는 기능일 제공해야 한다. 이는 사용자마다 각각의 DB 에 저장되어 이 DB 의 내용은 손실되거나 변경되어서는 안된다.

#### 6.2.1.d security requirement

해당 시스템은 거래를 제공하는 서비스이므로 사용자의 거래관련 개인정보 유출 방지를 위하여 보안성이 뛰어난 시스템을 구현해야 한다. 또한 플랫폼 관리자(운영자)의 계정에 대한 관리를 철저하게 하여 관리자를 제외한 다른 불법적인 접근을 차단하여야 하므로 이를 위해 관리자 계정에 대한 다중 접근 제한이 필요하다. 거래와 관련된 데이터베이스를 전송할 때 암호화를 하여 외부에 거래정보가 유출되지 않도록 하여야 한다.

#### 6.2.1.e Performance Requirement

어떤 사용 환경에서도 해당 시스템의 성능이 저하되어서는 안 된다. 특히 사용자들은 모바일을 자주 사용하므로, 모바일에서의 성능 관리가 매우 중요하다. 구매자 동시접속으로 트래픽이 몰릴 때를 대비하여, 서버와 주고받는 데이터를 최소화해야 한다.

### 6.2.2 Organizational requirements

#### 6.2.2.a Environmental Requirement

해당 시스템은 Bootstrap 을 사용하여, 사용자가 어떤 환경에서라도 접근성이 좋도록 반응형 웹 사이트를 구현한다. 특히, 쇼핑몰 같은 경우 모바일을 많이 이용하므로, 개발 시 모바일 환경을 우선순위로 웹 사이트를 개발한다.

#### 6.2.2.b Operational Requirement

공동구매 사이트로써, 상품의 남은 시간/수량을 실시간으로 알려 줘야 한다. 이를 위해, 웹 데이터 실시간 처리 API 를 통해 서비스를 제공한다. 상품이 매진되었을 경우, 알람으로 실시간 정보 제공을 해야 한다.

### 6.2.3 External Requirements

#### 6.2.3.a Regulatory Requirement

회원가입 시 사용자에게 개인정보 습득 권한 획득 요청을 해야 한다. 회원가입 정보를 제외한 사용자의 다른 정보에 접근을 차단해야 한다.

#### 6.2.3.b Safety/Security Requirement

해당 시스템은 개인정보 유출을 차단할 의무가 있다. 관리자는 서버에 접근하는 모든 사용자들을 관리할 수 있는 권한을 가진다. 거래 시에는 atomicity 를 유지해야 하기 때문에, 중간에 오류가 발생할 경우 모든 프로세스를 취소하고 거래 전으로 되돌아가야 한다. 특히, 시스템 내의 재화인 포인트는 현금과 1:1 가치이기 때문에 사용자의 조작을 막기 위해 암호화되어야 하며, 이를 위해 외부의 보안 API 를 사용할 수 있다.

## 6.3 Scenario example

### 6.3.1 구매 Scenario

#### 6.3.1.a Initial Assumption

유저가 이미 회원가입을 통해 계정을 생성하였고 시스템에 로그인 한 상태이다. 구매자는 이미 원하는 물품을 선택해 제품 상세 정보 페이지에 접속한 상태이다.

#### 6.3.1.b Normal flow of event

제품의 상세정보 페이지에 들어온 유저는 원하는 수량을 선택해 구매 버튼을 누른다. 제품은 유저 본인이 개인으로 주문하던 타 유저와 함께 공동 구매를 진행하던 무조건 최소 구매 수량이 넘어야 주문에 성공한다. 구매 버튼을 누르면 현재 유저가 소유한 포인트에서 제품의 가격에 맞게 포인트가 차감된다. 제품 구매 기간까지 주문 제품의 개수가 일정량 증가할 때마다 제품의 할인을 적용한다. 제품 구매의 제한기간이 지났을 때 제품의 최소 구매 수량을 넘는다면 주문에 성공한다.

#### 6.3.1.c What can go wrong

유저가 현재 보유하고 있는 포인트가 구매 금액보다 적을 시 구매에 실패한다. 이 경우 오류 메시지가 뜬다.

#### 6.3.1.d System state on completion

주문에 성공한다면 할인된 금액만큼 다시 유저의 포인트로 돌아온다. 제품 구매가 성공하면 성공하였다는 알림 메시지가 뜬다.

### 6.3.2 검색 scenario

#### 6.3.2.a Initial Assumption

사용자는 회원가입과 로그인을 통해 시스템에 접속하였다. 유저는 구매하고자 하는 상품이 있어야 하며, 상품의 이름을 통해 검색해야 한다.

#### 6.3.2.b Normal flow of events

메인 페이지 상단에 위치한 검색창에 원하는 상품의 이름을 검색한다. 기본으로 설정된 검색 정렬 조건은 최근 등록된 순이 원한다면 남은 기간 순, 즉 공동 구매 종료까지 남은 기간이 적은 순서대로 정렬 가능하다.

#### 6.3.2.c What can go

유저가 검색한 상품이 존재하지 않는다. 이 경우 상품이 존재하지 않는다는 오류 메시지를 보여준다.

#### 6.3.2.d System state on completion

유저가 검색한 이름의 상품이 유저가 설정한 정렬 조건에 따라 나타난다.

## 7. System Models

### 7.1 Context Model

#### 7.1.a Context Model



Diagram 6: Context Model

## 7.2 Interaction Model

### 7.2.a Use case Diagram

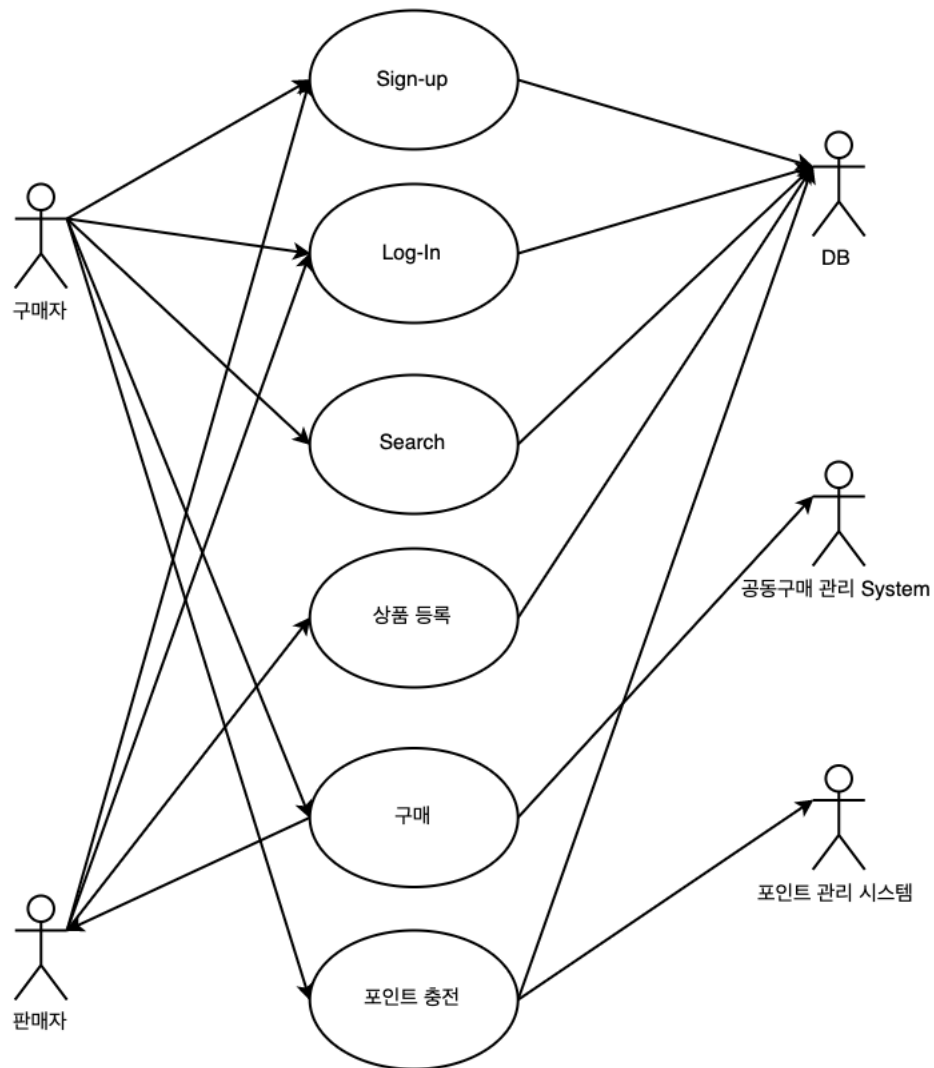


Diagram 7: Use case



## 7.2.b Tabular Description

### 7.2.b.1 Sign-up

Use Case	Sign-up
Actor	구매자, 판매자, DB
Description	구매자와 판매자가 시스템 이용을 위해 사용자 정보를 등록하는 과정이다. 이를 위해서는 먼저 유저가 판매자인지 구매자인지 구분한다. 이름, 성별과 같은 사용자 정보를 입력하고 아이디를 입력, 아이디 중복검사를 통과 해야 한다. 원하는 비밀번호를 입력한다. 가장 중요한 학교 인증은 학교 이메일로 인증번호를 전송하고, 인증에 성공하면 사용자의 정보가 데이터베이스에 저장된다.
Trigger	로그인 페이지에서 회원가입 버튼을 누른다
Success Response	사용자 정보를 입력하고 아이디 중복 확인 검사를 통과한 후, 학교 인증 (예. 성균관대학교 학생 일 시 <a href="mailto:이메일@skku.edu">이메일@skku.edu</a> 을 사용)에 성공한다면 사용자정보가 DB에 저장되고 로그인 페이지로 돌아간다.
Failure Response	인증번호 미입력이나 오류 시 “인증되지 않았습니다” 라는 경고 메시지 전송

Table 11: Tabular Desc – Sign up

### 7.2.b.2 Log-in

Use Case	Log-In
Actor	구매자, 판매자, DB
Description	사용자가 sign-up 하였다면 시스템을 이용하기 위해 DB에 저장된 정보와 사용자가 입력한 정보를 비교하여 사용자 인증을 시도한다.
Trigger	사용자가 ID와 PW 입력 후 로그인 버튼 클릭
Success Response	유저가 판매자라면 판매자 마이페이지로 이동, 구매자라면 메인 페이지로 이동한다

<b>Failure Response</b>	DB에 입력한 사용자 ID/PW가 들어있지 않다면 로그인 다시 시도
-------------------------	---------------------------------------

Table 12: Tabular Desc – Log in

#### 7.2.b.3 Search

<b>Use Case</b>	<b>Search</b>
<b>Actor</b>	구매자, DB
<b>Description</b>	사용자가 찾고자 하는 상품의 이름을 입력하면, 물품이 사용자가 원하는 검색 조건에 따라 정렬된다.
<b>Trigger</b>	사용자가 물품의 이름을 입력하고 검색을 누른다.
<b>Success Response</b>	
<b>Failure Response</b>	검색 조건에 일치하는 물품이 없다면 빈 페이지를 보여주고 오류 메시지를 전송한다.

Table 13: Tabular Desc – Search

#### 7.2.b.4 구매

<b>Use Case</b>	<b>구매</b>
<b>Actor</b>	구매자, 판매자, 공동구매 System, DB
<b>Description</b>	사용자가 원하는 상품을 찾는다면 포인트를 사용 제품의 주문을 넣는다면 판매자에게 데이터가 전송된다.
<b>Trigger</b>	원하는 제품 선택 후, 수량을 정한 후 구매 버튼을 클릭한다.
<b>Success Response</b>	개수 선택 후, 금액은 가지고 있는 포인트에서 차감된다. 이 후 유저에게 다른 사용자와 공유할 수 있는 공유하기 기능을 보여준다.
<b>Failure Response</b>	보유하고 있는 포인트가 부족하다면 구매가 불가능하다. 오류 메시지를 전송한다.

Table 14: Tabular Desc – 구매

#### 7.2.b.5 포인트 충전

<b>Use Case</b>	<b>포인트 충전</b>
<b>Actor</b>	구매자, 포인트 관리 System, DB
<b>Description</b>	구매자가 물품 구매를 위한 포인트를 충전한다. 구매자가 원하는 액수만큼 포인트를 충전할 수 있다. 충전된 포인트는 사용자 DB에 저장된다.
<b>Trigger</b>	마이페이지로 들어가 포인트 충전 클릭
<b>Success Response</b>	충전된 금액만큼 포인트로 입력되어 DB에 저장
<b>Failure Response</b>	-

Table 15: Tabular Desc – 포인트 충전

## 7.3 Sequence diagram

### 7.3.a 검색

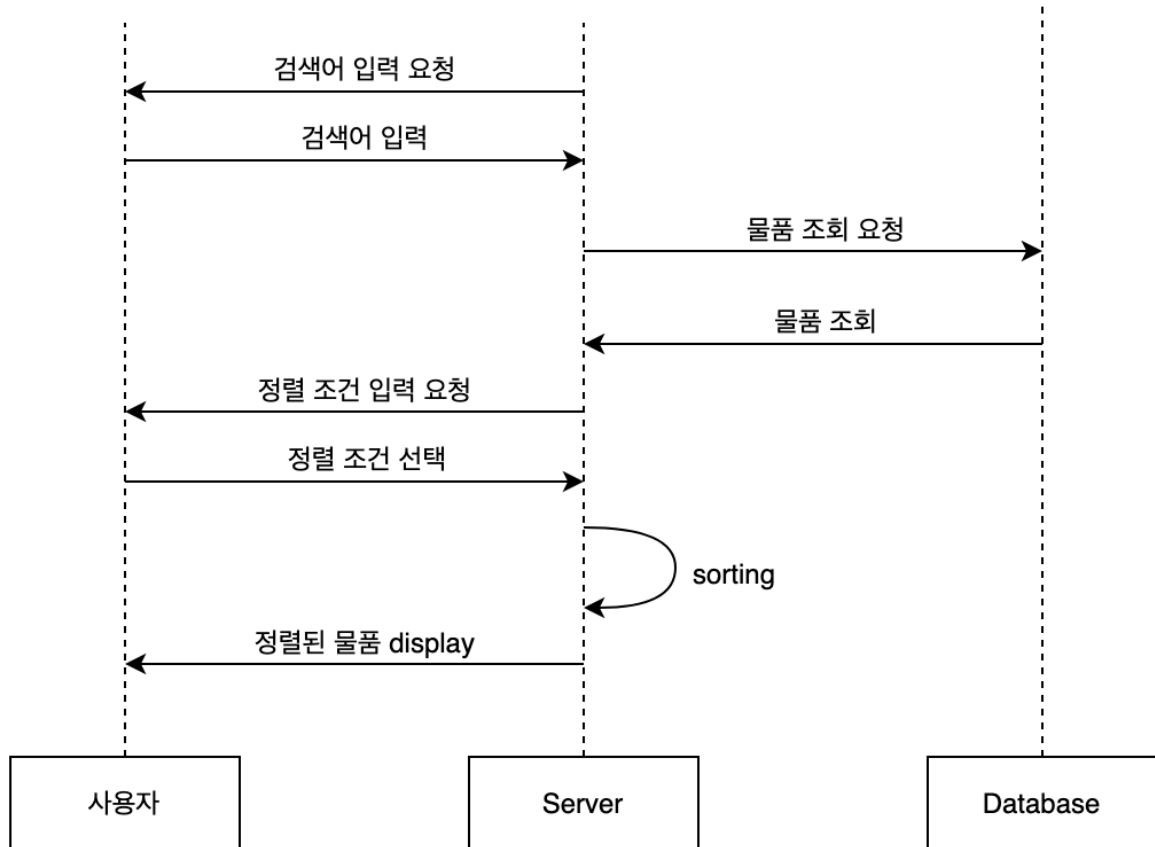


Diagram 8: Sequence Diagram – 검색

### 7.3.b Log-in

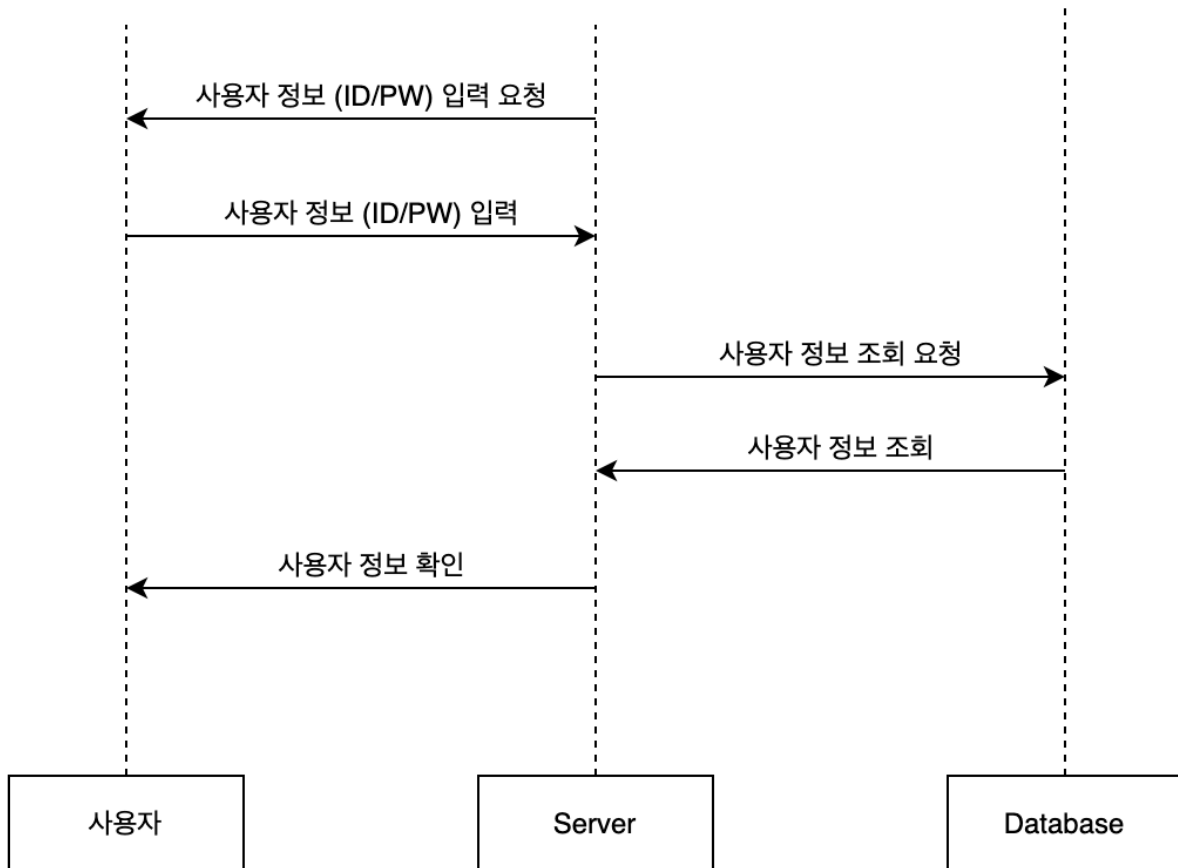


Diagram 9: Sequence Diagram – Log in

### 7.3.c 구매

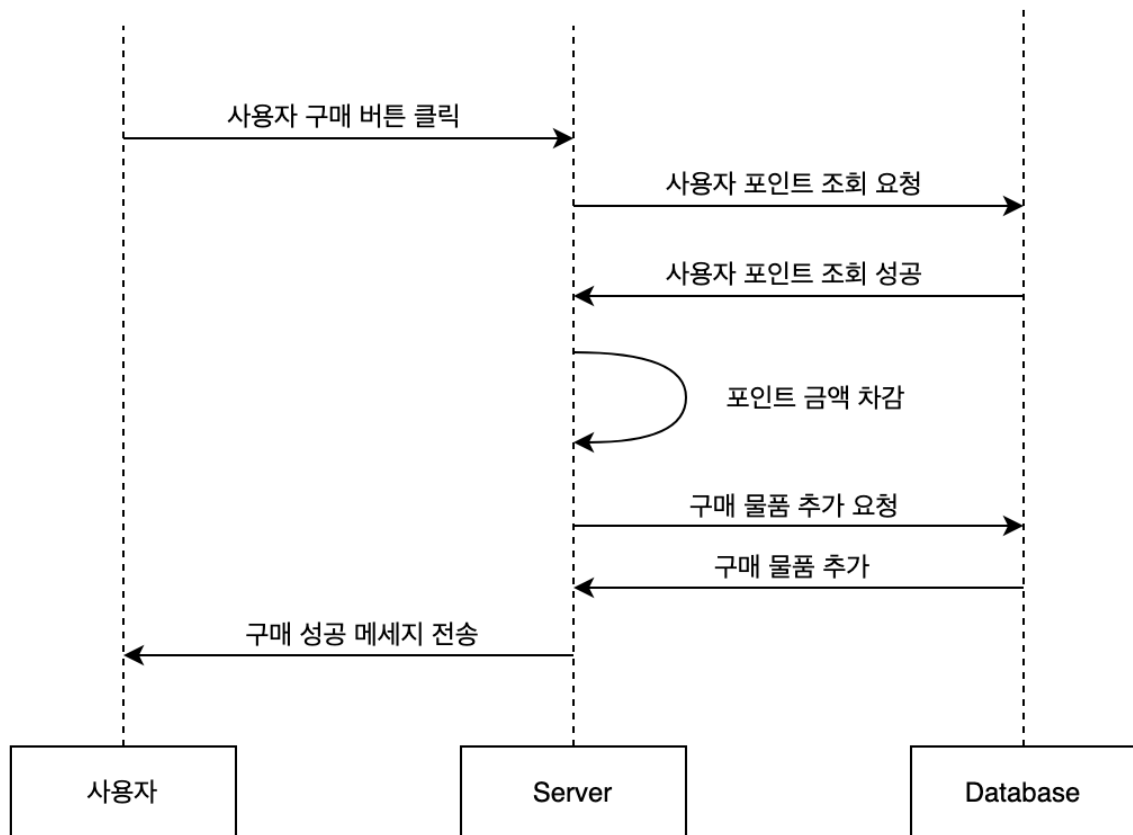


Diagram 10: Sequence Diagram – 구매

## 7.4 Behavioral Models

### 7.4.a Sign up Event-Driven Modeling

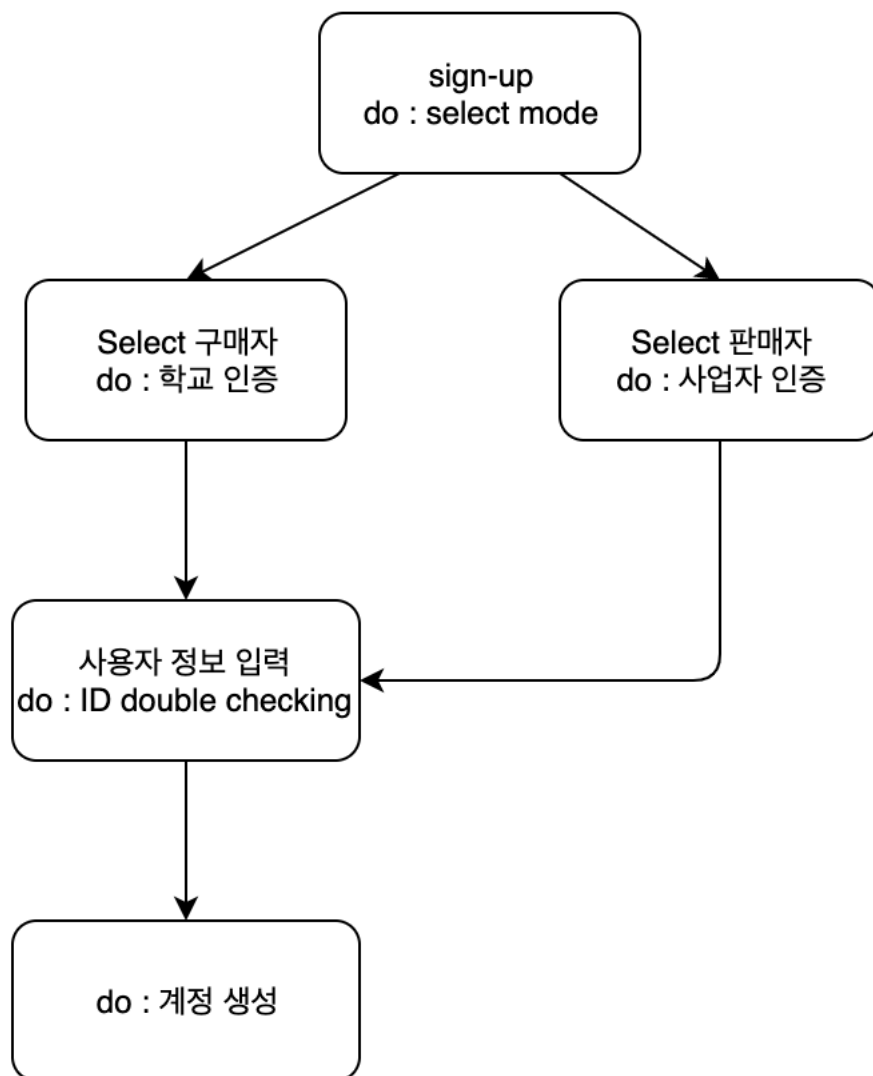


Diagram 11: Sign up Event-Driven Model

a. 포인트 충전 DFD

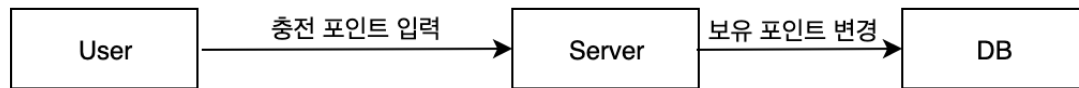


Diagram 12: 포인트 충전 DFD

## 8. System Evolutions

After the completion of this software, we will predict the possible future changes around the environment and system operation of the system. And infer the possible future system evolution. Through the prediction of these evolutions, we will know the future system updates in advance. This will reduce the cost of maintenance and change.

### 8.1. Popular product recommendation

요즘 사람들은 쇼핑을 할 때 명확한 목적이 없기 때문에, 해당 서비스가 사용자의 행동 패턴을 분석하여 적절한 상품을 추천할 수 있고, 최근 1주일 동안 가장 인기 있는 상품을 고객에게 추천할 수 있다.



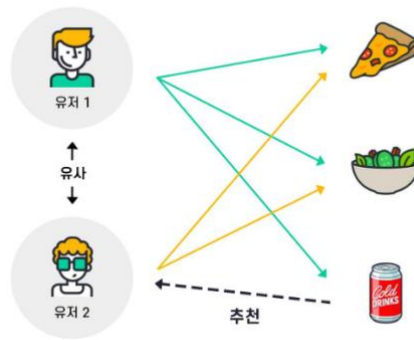


Figure 3: 추천 시스템 예시

## 8.2 Real-time tracking of delivery status

고객이 상품을 구매한 후 그들의 최대 관심사는 언제 구매한 물건을 받을 수 있는지에 대한 것이다. 따라서 제품의 배송에 대한 실시간 추적기능이 추가로 필요하며 이는 배달 업체가 제공하는 서비스를 통하여 실현할 수 있고 이를 통해 소비자들은 구매한 상품의 현재 위치를 파악하는데 도움을 받을 수 있다. 또한 AI, 머신 러닝 기술의 발달로 인하여 통계를 기반으로 특정 날짜에 제품이 도착할 확률을 계산하는 서비스도 이미 존재한다. (네이버 쇼핑 등)

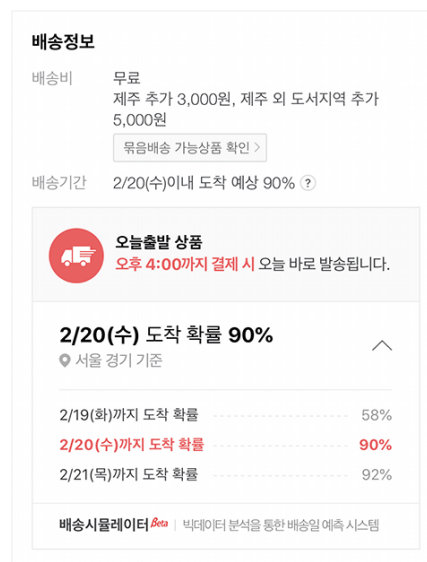


Figure 4: 실시간 배송 예측 시스템

### 8.3 OAuth implementation

고객들은 사이트에 새롭게 회원가입을 하는 것을 귀찮아 하는 경향이 있다. 따라서 카카오, 네이버나 구글 계정 등과 연동하여 간편하게 회원가입 하는 방식이 편의성을 높여 줄 수 있다. Oauth 을 통해 안전하고 손쉬운 접속이 여러 플랫폼에서 거의 필수적인 요소가 되었다.

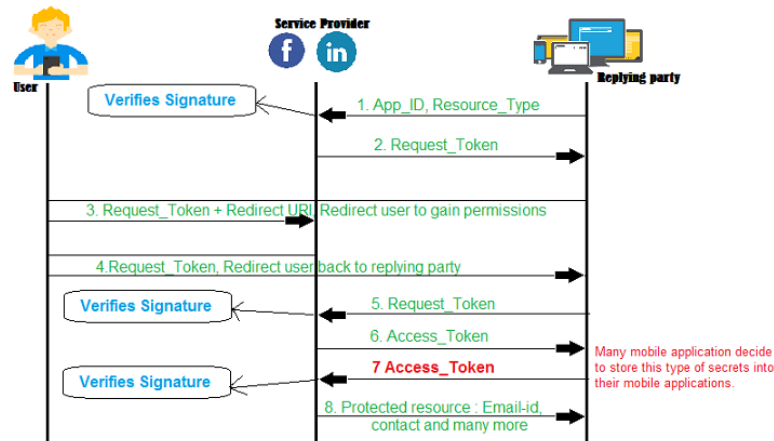


Figure 5: OAuth Sequence Diagram

### 8.4 Sorting and filtering

고객들이 쇼핑을 할 때, 특정 상품이 다양한 가격대인 경우가 있으므로 상품을 효율적으로 고객이 원하는 대로 정렬하고 선별하는 기능은 시스템의 발전에 있어서 중요한 부분이다.

## 9. Appendices

### 9.1. Objectives

appendices에서는 개발되는 시스템에 대해 더 자세하고 구체적인 정보들을 제공한다. 데이터베이스 및 requirements, 시스템 개발 방식에 대해 설명한다.

### 9.2. DB requirements

해당 시스템의 database 요구사항을 설명한다. 아래에서 설명하는 애용들은 적절한 형태로 database 테이블 형태로 저장되어야 한다. 각 요구사항에 Attribute 는 *이탤릭체*, Entity type 는 **Bold 체**, Relation type 는 **Bold 이탤릭체**, Constraints 는 밑줄로 표시하였다.

해당 시스템에서 구매자 엔티티는 구매자에 관한 정보를 가진 데이터베이스이다. 구매자 번호(Buyer\_number)가 키 특성이 되고 ID, 비밀번호, e-mail, point, 소속 학교의 특성을 가진다.

판매자 엔티티는 판매자에 관한 정보를 가진 데이터베이스이다. 판매자 번호(Seller\_number)가 키 특성이 되고 ID, 비밀번호, e-mail, point 의 특성을 가진다.

제품 목록 엔티티는 판매자가 판매하는 물품에 대한 정보를 가진 데이터베이스이다. 제품 ID(product\_ID)가 키 특성이 된다. 판매자의 정보를 참조해야 하기 때문에 판매자 ID 에 대한 정보를 가진다. 또한 학교에 따라 제품 목록이 달라지기 때문에, 학교 속성도 추가되었다. 그 이외에 제품 이름, 개당 가격, 할인률, 최종 목표 개수, 판매 개수, 최소 배송 개수, 공동구매 기간에 관한 속성을 가진다.

구매 목록 엔티티는 구매자가 구매한 물품에 대한 정보를 가진 데이터베이스이다. 구매자 번호(Buyer\_number), 제품 ID(product\_ID)가 키 특성이 된다. 추가로 구매 개수에 대한 정보를 담고 있다. 구매자가 구매한 상품의 개수만큼의 정보를 담고 있어야, 나중에 공동구매가 성사되었을 때 판매자에게 수익금과 구매자에게 할인율만큼 할인된 가격을 다시 지급해 주기 때문에 반드시 필요한 엔티티이다.

### 9.3. User-system requirements

해당 시스템은 HTML5 기반의 Node.js tool 을 이용하여 설계된다. 따라서 사용자는 최신 HTML5 를 지원하는 웹 브라우저가 필요하다. 데스크탑, 모바일 모두 같은 조건을 요구한다.

최소 사양	HTML5 를 제공하는 브라우저
권장 사양	Google Chrome

Table 16: User-system requirements

### 9.4. Development process

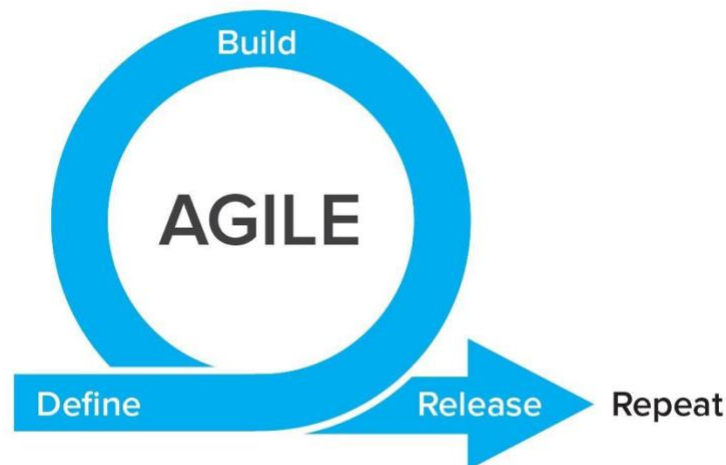


Figure 6: Development process

소비자의 요구사항이 완벽하게 파악된 것이 아니고, 대형 프로젝트가 아니기 때문에 Agile Method 를 선택하였다. 팀원과 역할을 분할하여, 각각의 기능을 빠르게 개발하고 지속적으로 테스트 및 피드백을 받는다. 시간이 부족하기 때문에, 역할을 분할하여 개발한 후 6~7 일마다 합쳐 중간 결과물을 만들고 수시로 회의를 하며 코드를 공유한다. 이 중간 결과물은 수시로 customer 들에게 보여주며 incrementally 하게 전달한 후 feedback 을 적극적으로 수용한다.

## 10. Index

### 10.1 Figure

Figure 1: MAU 규모 증감폭 .....	11
Figure 2: 카카오톡 Talk Deal .....	12
Figure 3: 추천 시스템 예시 .....	49
Figure 4: 실시간 배송 예측 시스템 .....	49
Figure 5: OAuth Sequence Diagram .....	50
Figure 6: Development process .....	52

### 10.2 Table

Table 1: Glossary .....	14
Table 2: Functional Requirement – Sign Up .....	26
Table 3: Functional Requirement – Log in .....	27
Table 4: Functional Requirement – Point Charge .....	28
Table 5: Functional Requirement – Item register .....	29
Table 6: Functional Requirement – Search items .....	30
Table 7: Functional Requirement - Share .....	31
Table 8: Functional Requirement – Item details .....	32
Table 9: Functional Requirement – My page .....	33
Table 10: Functional Requirement - Purchase .....	34
Table 11: Tabular Desc – Sign up .....	41
Table 12: Tabular Desc – Log in .....	42
Table 13: Tabular Desc – Search .....	42
Table 14: Tabular Desc – 구매 .....	42

Table 15: Tabular Desc – 포인트 충전 .....	43
Table 16: User-system requirements .....	52

### 10.3 Diagram

Diagram 1: Overall Architecture .....	20
Diagram 2: Point management System .....	21
Diagram 3: Item search System .....	22
Diagram 4: Group purchase System 1 .....	23
Diagram 5: Group purchase System flow chart .....	24
Diagram 6: Context Model .....	39
Diagram 7: Use case .....	40
Diagram 8: Sequence Diagram – 검색 .....	44
Diagram 9: Sequence Diagram – Log in .....	45
Diagram 10: Sequence Diagram – 구매 .....	46
Diagram 11: Sign up Event-Driven Model .....	47
Diagram 12: 포인트 충전 DFD .....	48

## 11. References

Agile method, <https://medium.com/adalab/what-is-the-agile-methodology-98396e9c7b0e>