

# 시스템 요구명세서

---

지역 공동 구매 플랫폼

Team		
팀장		
팀원		

# 내용

1. 개요.....	4
1.1. 시스템의 개요 .....	4
1.2. 시스템의 목적 .....	4
1.3. 정의, 약어 .....	4
2. 기능적 요구 .....	6
2.1. 외부 인터페이스 요구 .....	6
2.1.1. 사용자 인터페이스 .....	6
2.1.2. 하드웨어 인터페이스.....	8
2.1.3. 소프트웨어 및 통신 인터페이스.....	8
2.2. 기능 요구 .....	9
2.2.1. 기능 #1(회원가입).....	9
2.2.2. 기능 #2(로그인).....	10
2.2.3. 기능 #3(회원 검색).....	11
2.2.4. 기능 #4(카테고리 등록).....	12
2.2.5. 기능 #5(상품 등록).....	13
2.2.6. 기능 #6(상품 목록 조회) .....	14
2.2.7. 기능 #7(상품 검색).....	15
2.2.8. 기능 #8(상품 재고 수정) .....	16
2.2.9. 기능 #9(주문 조회).....	16
2.2.10. 기능 #10(주문 취소).....	17
2.2.11. 기능 #11(위치 설정).....	18

2.2.12.	기능 #12(딥 생성).....	18
2.2.13.	기능 #13(딥 참여).....	19
2.2.14.	기능 #14(결제).....	20
3.	기타 요구 및 제약 사항 .....	21
3.1.	성능 요구 .....	21
3.2.	H/W 요구 .....	21
3.3.	예외 조건 및 이의 처리.....	22
3.4.	자원, 인력에 대한 제약 조건 .....	22
4.	인수 조건.....	22
4.1.	기능 시험 및 성능 시험.....	22
5.	참고 자료.....	23

# 1. 개요

## 1.1. 시스템의 개요

최근 1인 가구가 증가하고 있고, 특히 학교 주변에서 자취를 하는 학생들은 매우 많다. 이러한 자취생들은 끼니를 해결하는 것에 부담을 느낀다. 매 끼니마다 외식을 할 경우 식비에 부담을 느껴 집에서 음식을 해먹으려 하지만 이것도 역시 부담스럽다. 그 이유는 요리를 하기 위한 재료들을 구매하는 과정에서 재료를 소분해서 팔지 않아 대량 구매를 해야 하기 때문이다. 이는 쓰레기 배출량의 증가로 쓰레기를 처리해야하는 부담도 증가시킨다. 이러한 문제를 해결하기 위해 '딤(DDIP)' 지역 공동 구매 시스템에서는 내가 속해 있는 지역 안에서 상품을 구매할 수 있는 마트를 알려주는 기능을 제공하고, 그 안에서 다량의 상품을 여러 사람과 함께 구매할 수 있는 공동 구매의 공간을 만들어준다.

## 1.2. 시스템의 목적

- 첫째, 집에서 요리를 해먹고 싶은 자취생들에게 소량의 재료를 구매할 수 있는 기회를 제공한다.
- 둘째, 재료를 구입하고 버려지는 쓰레기 배출량이 감소한다.
- 셋째, 재료를 구입할 때 드는 비용과 생필품이나 다양한 상품을 구입할 때 불필요하게 드는 비용이 줄어들어 생활비를 절약한다.
- 넷째, 앱을 통해 미리 상품을 선택하고 결제를 함으로써, 마트에 직접 방문했을 때 소요되는 시간을 최소화한다.
- 다섯째, 앱을 통해 미리 상품을 구매함으로써, 마트 방문 시의 장보기 절차를 간소화한다.

## 1.3. 정의, 약어

- DDIP - 우리가 개발할 어플리케이션 이름. 지역 공동 구매 플랫폼. 어플 내에서는 공동 구매 방을 일컫는 말이다.
- 공동으로 구매할 수 있는 공간 - 판매자가 공동 구매하고자 하는 상품을 선택하여 DDIP을 올리면 동일한 상품을 구매하고자 하는 구매자들이 DDIP에 참여할 수 있다.
- 지역 마트 활성화 - 처음에는 학교 주변 마트들을 상대로 시작. 마트들과 협력하여 우리 어플리케이션에 마트의 상품들이 등록되고 사용자들이 구매할 수 있도록 하여 마트 상품의 구매를 촉진한다.

- UI - User Interface(사용자가 어플리케이션을 사용하는 방법)
- MRF – Modification Request Form
- CR – Change Request
- 공동구매 - 마트의 상품을 여러 사용자들이 함께 구매하고 소분하여 갖는 행위
- 나눔 - 본인이 사용하지 않는 물건을 필요로 하는 사용자들에게 나눠주는 행위.  
어플리케이션 출시 이후 추후 업데이트될 버전에 추가될 기능
- 크로스 플랫폼(cross-platform) – 컴퓨터 프로그램, 운영 체제, 컴퓨터 언어, 프로그래밍 언어, 컴퓨터 소프트웨어 등이 여러 종류의 컴퓨터 플랫폼에서 동작할 수 있다는 것을 뜻함.
- RSA (Rivest Shamir Adleman) - 공개키와 개인키를 세트로 만들어서 암호화와 복호화를 하는 인터넷 암호화 및 인증 시스템
- 스푸핑(Spoofing) - 외부 악의적 네트워크 침입자가 임의로 웹사이트를 구성하고 일반 사용자들의 방문을 유도하여, 해당 웹사이트를 통해 이용자의 정보를 빼가는 해킹 수법

## 2. 기능적 요구

### 2.1. 외부 인터페이스 요구

인터페이스 요구사항 ID	명칭	설명
IR-001	DB 서버 연결	DB 서버와 애플리케이션 서버를 연결한다.
IR-002	어플에서 카메라 접근 권한 허가	QR 코드를 인식하기 위해 어플에서 카메라를 사용할 수 있도록 허용한다.
IR-003	어플에서 위치 정보 접근 권한 허가	현재 위치를 기반으로 주위의 마트를 추천하기 위해 어플에서 위치 정보를 사용할 수 있도록 허용한다.
IR-004	개인정보 활용 동의	물건 공급자-구매자 간 인증을 위해 QR 코드에 들어갈 사용자의 구매 정보 등을 사용할 수 있도록 허가한다.
IR-005	결제	어플에서의 상품 구매를 위해 외부 결제 인터페이스와 통신한다.
IR-006	QR 코드 생성	공동구매가 성사되면, 각 사용자들은 자신의 구매 정보를 포함하는 QR 코드를 발급받는다.
IR-007	QR 코드 인식	물건 공급자는 구매자의 QR 코드를 카메라로 인식하여 구매자의 구매 정보를 확인한다.
IR-008	주소 검색	사용자의 입력에 해당하는 주소지를 검색하여 결과를 보여준다.

#### 2.1.1. 사용자 인터페이스

입력 : 스마트폰 화면 터치
출력 : 화면을 통한 출력

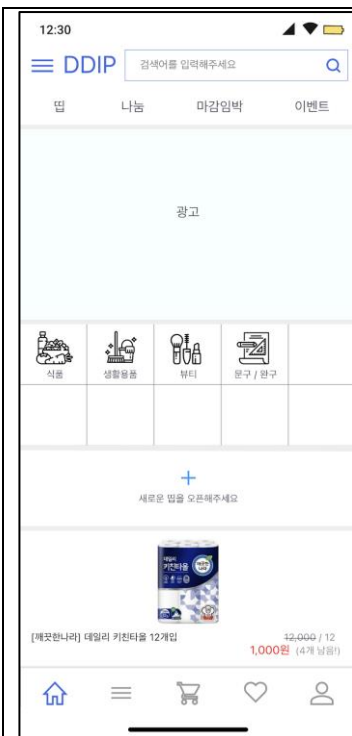


그림 2.1.1-1 어플 실행 시 보이는 기본 화면(홈 화면)

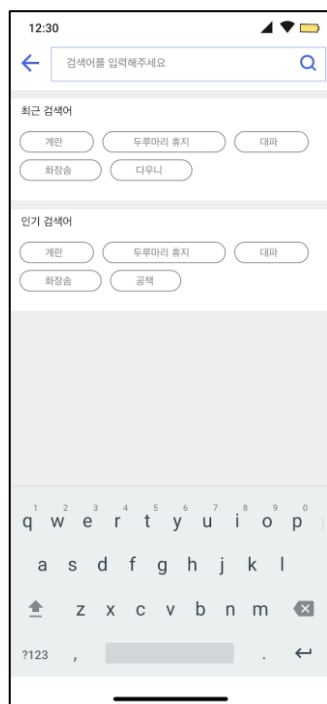


그림 2.1.1-2 상품 검색 중 보이는 화면



그림 2.1.1-3 상품 검색 후 보이는 결과 화면

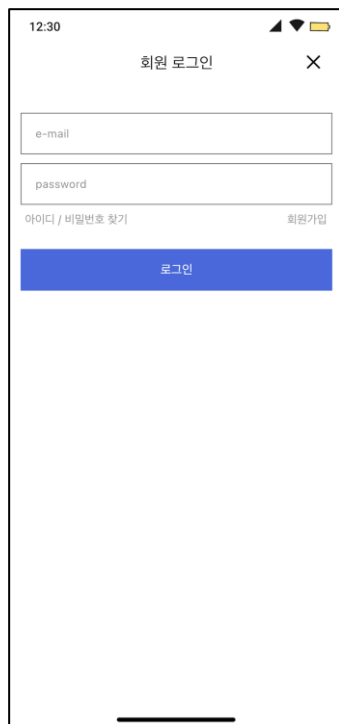


그림 2.1.1-4 회원 로그인 시 보이는 화면

그림 2.1.1-5 회원가입 시 보이는 화면



### 2.1.2. 하드웨어 인터페이스

'딥(DDIP)'은 안드로이드와 iOS를 동시에 지원하는 크로스 플랫폼 기반 애플리케이션으로 iOS 또는 안드로이드 운영체제를 사용하는 스마트폰 기기가 필요하다.

### 2.1.3. 소프트웨어 및 통신 인터페이스

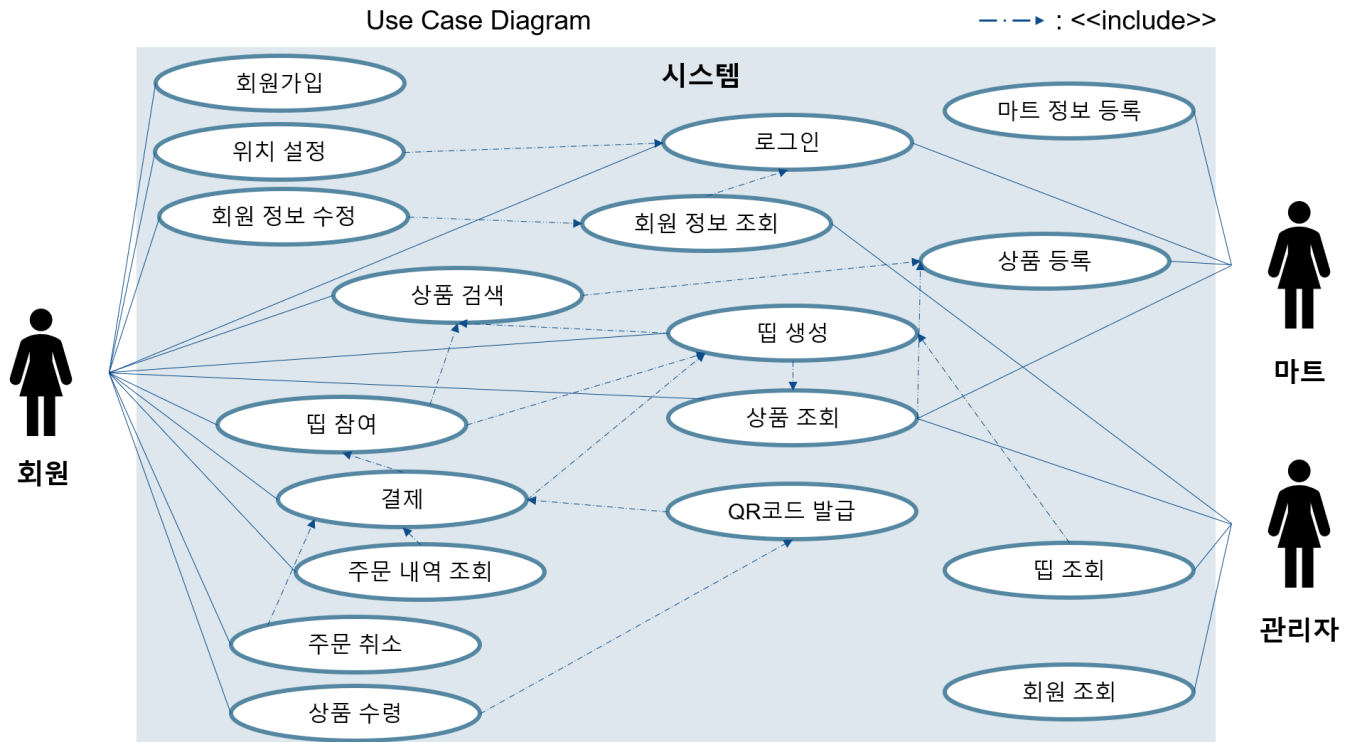
- 소프트웨어 인터페이스
  - OS: 다양한 사용자에게 서비스를 제공하기 위해 크로스 플랫폼을 사용하여 안드로이드, iOS 운영체제를 모두 지원
  - DB: USER, MARKET, PRODUCT, 거래(DDIP) 정보의 저장을 위한 데이터베이스로 MySQL 사용
  - Server: 백엔드 구현을 위해 Javascript 언어가 구동되는 환경으로 Nodejs 사용
  - Client: 프론트엔드 구현을 위한 React JS 사용
  - 기타: 기능 구현을 위한 데이터베이스 툴로 Prisma 사용
- 통신 인터페이스
  - 서버/클라이언트 통신
    1. Server: 실행 상태가 되어, Client 의 접속을 기다림
    2. Client: Server 의 서비스 이용을 위해 request 를 보내고, 서버로부터 response 를 기다림
    3. Server: Client 의 접속을 받아 서비스를 제공할 유효한 대상인지 확인, 보안 확인 후 Client 에게 서비스 제공
    4. Client: Server 의 서비스를 받아 처리
  - 애플리케이션 보안
    1. Google Play: Developer Console 에서 애플리케이션과 연결된 개인 키를 사용하여 서명 생성
    2. Developer Console: 각 애플리케이션에 대한 RSA 키 쌍 생성



3. 애플리케이션: 서명된 응답을 수신

4. 보안 원격 서버: RSA 키 쌍의 공개 키 부분을 사용하여 서명 확인 → 무단 변경 또는 스푸핑된 응답 탐지 가능

## 2.2. 기능 요구



### 2.2.1. 기능 #1(회원가입)

구분	내용
요구사항명	회원가입
액터명	구매자
개요 및 설명	구매자는 자신의 정보를 등록하여 회원가입한다.
사전 조건	회원가입되어 있지 않은 상태여야 한다.
기본 흐름	1. (시스템) 기존 가입 회원인지 이름, 전화번호를 검색하여 확인한다.

	2. (액터) 회원가입을 요청한다. 3. (시스템) 회원약관을 보여준다. 4. (액터) 회원약관에 동의한다. 5. (시스템) 회원정보 입력항목을 보여준다. 6. (액터) 회원정보 항목을 입력하고 등록 요청을 한다. 7. (시스템) 입력된 정보를 확인한다. 8. (시스템) 신규 회원정보를 저장, 등록을 완료한다.
예외 흐름	E1. 입력란을 비우고 등록 버튼을 클릭할 경우 오류 메시지를 띄우고 재입력을 요청한다. E2. 기존에 가입되어 있는 회원인 경우 메시지를 보여주고, 아이디 찾기 메뉴를 추천한다. E3. 회원약관에 동의하지 않을 경우 약관 동의 하에 회원가입이 가능하다는 오류 메시지를 보여주고, 동의를 재요청한다. E4. 입력 형식이 틀렸을 경우, 메시지를 보여주고 재입력을 요청한다.
대안 흐름	A1. 취소 버튼을 클릭할 경우 로그인 화면을 보여준다.
사후 조건	신규 사용자 정보가 저장되고, 로그인 창이 보여진다.

### 2.2.2. 기능 #2(로그인)

구분	내용
요구사항명	로그인
액터명	구매자, 판매자, 관리자
개요 및 설명	액터는 자신의 ID로 로그인한다.

사전 조건	회원가입된 상태여야 한다.
기본 흐름	1. (시스템) ID, 비밀번호 입력창을 보여준다. 2. (액터) ID, 비밀번호를 입력하고 로그인 요청을 한다. 3. (시스템) 입력된 정보를 확인한다. 4. (시스템) 사용자 로그인을 완료한다.
예외 흐름	E1. 입력한 ID가 존재하지 않는 경우 오류 메시지를 띄우고, 회원가입/아이디 찾기를 추천한다. E2. 비밀번호가 틀린 경우 오류 메시지를 띄우고, 로그인 창을 다시 보여준다. E3. 동일한 ID에 대해 비밀번호를 5회 이상 틀린 경우 5분 간 해당 액터의 로그인 시도를 차단한다.
대안 흐름	-
사후 조건	로그인 된 상태로 메인 화면을 보여준다.

### 2.2.3. 기능 #3(회원 검색)

구분	내용
요구사항명	회원 검색
액터명	관리자
개요 및 설명	액터는 쇼핑몰에 가입된 회원을 검색한다.
사전 조건	관리자 ID로 로그인 된 상태이다.
기본 흐름	1. (시스템) 전체 회원 목록을 보여준다. 2. (액터) 조건을 입력한다. 3. (시스템) 조건을 만족하는 회원을 표시한다.

예외 흐름	<p>E1. 검색 입력 내용이 없는 경우</p> <p>내용 입력 요구 메시지를 출력하고, 조건을 입력하도록 한다.</p> <p>E2. 조건을 만족하는 회원이 없는 경우</p> <p>“검색 결과가 없습니다”라는 메시지를 띄운다.</p>
대안 흐름	-
사후 조건	검색된 회원 목록을 보여준다.

2.2.4. 기능 #4(카테고리 등록)

구분	내용
요구사항명	카테고리 등록
참여 액터	관리자
개요 및 설명	상품 분류를 위한 카테고리를 등록한다.
사전 조건	관리자는 관리 권한을 가진 ID로 로그인한다.
기본 흐름	<ol style="list-style-type: none"> <li>1. 시스템이 최상위 카테고리 목록(카테고리명)을 관리자에게 보여준다.</li> <li>2. 관리자가 조회하고자 하는 카테고리를 선택한다.</li> <li>3. 시스템이 선택된 카테고리의 하위 카테고리(카테고리명) 목록을 보여준다.</li> <li>4. 등록할 카테고리의 상위 카테고리가 나타날 때까지 기본 흐름 2-3을 반복한다.</li> <li>5. 상위 카테고리로 지정하고자 하는 카테고리를 선택한다.</li> <li>6. 등록할 카테고리명을 입력하고 저장 기능을 실행한다.</li> <li>7. 카테고리 등록 후 카테고리 목록을 갱신하여 출력한다.</li> </ol>
예외 흐름	<p>6-E1. 입력한 카테고리명이 이미 존재하는 경우</p> <p>6-E1-1. 동일한 이름이 이미 존재한다는 메시지 출력</p> <p>6-E1-2. 중복 카테고리가 어느 위치에 존재하는지 출력</p>

대안 흐름	1-A1. 최상위 카테고리를 등록하려는 경우  1-A1-1. 상위 카테고리를 선정하지 않고, 카테고리명 입력 후 저장 기능 실행
사후 조건	등록한 카테고리가 포함된 전체 카테고리 목록이 보여진다.

## 2.2.5. 기능 #5(상품 등록)

구분	내용
요구사항명	상품 등록
참여 액터	판매자
개요 및 설명	앱에서 판매할 새로운 상품의 정보를 등록한다.
사전 조건	관리자는 관리 권한을 가진 ID로 로그인한다.
기본 흐름	<ol style="list-style-type: none"> <li>1. 시스템이 최상위 카테고리 목록(카테고리명)을 관리자에게 보여준다.</li> <li>2. 관리자가 조회하고자 하는 카테고리를 선택한다.</li> <li>3. 시스템이 선택된 카테고리의 하위 카테고리(카테고리명) 목록을 보여준다.</li> <li>4. 등록할 카테고리의 상위 카테고리가 나타날 때까지 기본 흐름 2-3을 반복한다.</li> <li>5. 상품을 등록할 최하위 카테고리를 선택한다.</li> <li>6. 선택된 카테고리에 속한 상품들의 목록(상품명, 가격, 등록일자)을 보여준다.</li> <li>7. 등록할 상품의 정보(상품명, 가격, 상품사진, 상품설명)를 입력하고 상품을 등록한다.</li> <li>8. 신규 상품의 등록 결과를 보여준다.</li> </ol>
예외 흐름	<p>E1. 상품 정보 입력란 중 빈 칸이 존재하는 경우</p> <p>E1-1. 누락된 값이 존재한다는 메시지 출력</p> <p>E1-2. 값이 누락된 입력란으로 커서 이동</p> <p>E2. 입력한 신규 상품의 상품명에 이미 존재하는 경우</p> <p>E2-1. 동일한 이름이 이미 존재한다는 메시지 출력</p>

	E2-2. 해당 입력란으로 커서 이동
대안 흐름	-
사후 조건	관리자가 입력한 신규 상품 정보가 저장된다.

## 2.2.6. 기능 #6(상품 목록 조회)

구분	내용
요구사항명	상품 목록 조회
참여 액터	구매자, 판매자, 관리자
개요 및 설명	특정 카테고리를 선택하여 카테고리 별 상품 목록을 조회한다.
사전 조건	-
기본 흐름	<ol style="list-style-type: none"> <li>1. 시스템이 최상위 카테고리 목록(카테고리명)을 관리자에게 보여준다.</li> <li>2. 관리자가 조회하고자 하는 카테고리를 선택한다.</li> <li>3. 시스템이 선택된 카테고리의 하위 카테고리(카테고리명) 목록을 보여준다.</li> <li>4. 등록할 카테고리의 상위 카테고리가 나타날 때까지 기본 흐름 2-3을 반복한다.</li> <li>5. 상품을 등록할 최하위 카테고리를 선택한다.</li> <li>6. 선택된 카테고리에 속한 상품들의 목록(상품명, 가격, 등록일자)을 보여준다.</li> <li>7. 등록할 상품의 정보(상품명, 가격, 상품사진, 상품설명)를 입력하고 상품을 등록한다.</li> <li>8. 신규 상품의 등록 결과를 보여준다.</li> </ol>
예외 흐름	<p>E1. 입력한 신규 상품 정보 중 누락된 값이 존재하는 경우</p> <p>E1-1. 누락된 값이 존재한다는 메시지 출력</p> <p>E1-2. 값이 누락된 텍스트 박스로 커서 이동</p>

	<p>E2. 입력한 신규 상품의 상품명에 이미 존재하는 경우</p> <p>E2-1. 동일한 이름이 이미 존재한다는 메시지 출력</p> <p>E2-2. 상품명 텍스트 박스로 커서 이동</p>
대안 흐름	-
사후 조건	관리자가 입력한 신규 상품 정보가 저장된다.

### 2.2.7. 기능 #7(상품 검색)

구분	내용
요구사항명	상품 검색
액터명	구매자, 판매자, 관리자
개요 및 설명	상품명을 입력하여 해당하는 상품을 검색한다.
사전 조건	로그인 한 상태이다.
기본 흐름	<p>1. (시스템) 검색어 입력창을 보여준다.</p> <p>2. (액터) 상품 검색 조건을 입력한다.</p> <p>3. (시스템) 검색된 상품 목록을 표시한다.</p>
예외 흐름	<p>E1. 검색 입력 내용이 없는 경우</p> <p>내용 입력 요구 메시지를 출력하고, 조건을 입력하도록 한다.</p> <p>E2. 조건을 만족하는 상품이 없는 경우</p> <p>“검색 결과가 없습니다”라는 메시지를 띄운다.</p>
대안 흐름	A1. 검색어를 일부 입력했을 때, 해당 입력 단어로 시작하는 검색어를 추천한다.
사후 조건	검색된 상품 목록을 보여준다.

### 2.2.8. 기능 #8(상품 재고 수정)

구분	내용
요구사항명	상품 재고 수정
액터명	판매자, 관리자
개요 및 설명	등록된 상품의 재고를 수정한다.
사전 조건	판매자, 관리자 ID로 로그인한 상태이다.
기본 흐름	<ol style="list-style-type: none"> <li>1. (시스템) 해당 액터에게 등록된 상품을 보여준다.</li> <li>2. (액터) 재고를 수정할 상품을 선택한다.</li> <li>3. (시스템) 선택한 상품의 재고와 입력창을 띄운다.</li> <li>4. (액터) 입력된 상품의 수정 후 재고를 입력한다.</li> <li>5. (시스템) 입력된 상품의 재고를 수정한다.</li> </ol>
예외 흐름	<p>E1. 재고 내용이 없는 경우</p> <p>내용 입력 요구 메시지를 출력하고, 조건을 입력하도록 한다.</p> <p>E2. 재고가 범위에서 벗어난 경우</p> <p>“올바른 값을 입력해주세요”라는 메시지를 띄운다.</p>
대안 흐름	<p>A1. 재고가 0인 경우</p> <p>해당 상품 정보에 “품절” 표시를 한다.</p>
사후 조건	선택한 상품의 재고 수정 정보를 보여준다.

### 2.2.9. 기능 #9(주문 조회)

구분	내용
요구사항명	주문 조회
액터명	구매자



개요 및 설명	사용자의 주문 내역을 조회한다.
사전 조건	사용자 ID로 로그인한 상태여야 한다.
기본 흐름	1. (시스템) 해당 액터에 해당하는 주문 목록을 띄운다. 2. (액터) 목록에서 원하는 주문을 선택한다. 3. (시스템) 선택한 주문의 상세 정보를 표시한다.
예외 흐름	E1. 주문 내역이 없는 경우  "주문 내역이 없습니다"라는 메시지를 표시한다..
대안 흐름	-
사후 조건	사용자의 주문 정보를 표시한다.

#### 2.2.10.기능 #10(주문 취소)

구분	내용
요구사항명	주문 취소
액터명	관리자
개요 및 설명	요청한 사용자의 주문을 취소한다.
사전 조건	- 관리자 ID로 로그인한 상태여야 한다. - 사용자의 주문 내역이 있어야 한다. - 사용자가 주문 취소를 요구한 경우여야 한다.
기본 흐름	1. (시스템) 사용자의 주문 목록을 보여준다. 2. (액터) 취소 요청된 주문을 선택한다. 3. (시스템) 해당 주문을 취소 처리한다.
예외 흐름	E1. 주문 내역이 없는 경우

	"주문 내역이 없습니다"라는 메시지를 표시한다.
대안 흐름	-
사후 조건	취소 처리된 사용자의 주문을 보여준다.

#### 2.2.11.기능 #11(위치 설정)

구분	내용
요구사항명	위치 설정
액터명	구매자
개요 및 설명	구매자의 위치 정보를 설정한다.
사전 조건	사용자 ID로 로그인한 상태여야 한다.
기본 흐름	1. (시스템) 액터의 핸드폰 GPS 정보를 이용하여 현재 위치를 표시한다. 2. (액터) 지도 상에 표시된 현재 위치를 선택한다. 3. (시스템) 선택된 위치 정보를 액터의 위치로 설정한다.
예외 흐름	E1. GPS 정보를 가져올 수 없는 경우 "GPS 상태를 읽어올 수 없습니다."라는 메시지를 표시한다.
대안 흐름	A1. 위치 설정 방법으로 "주소 직접 입력"을 하는 경우 입력한 주소로 사용자의 위치를 설정한다.
사후 조건	설정된 사용자의 위치 정보를 표시한다.

#### 2.2.12.기능 #12(딴 생성)

구분	내용
----	----

요구사항명	딤 생성
액터명	구매자
개요 및 설명	판매 상품에 대한 새로운 딤을 생성한다.
사전 조건	<ul style="list-style-type: none"> <li>- 사용자 ID로 로그인한 상태여야 한다.</li> <li>- 구매자가 원하는 상품을 선택한 상태이다.</li> </ul>
기본 흐름	<ol style="list-style-type: none"> <li>1. (시스템) 해당 상품 상세 정보를 띄운다.</li> <li>2. (액터) 주문 수량을 선택한다.</li> <li>3. (액터) 상품 주문을 선택한다.</li> <li>4. (시스템) 생성된 딤을 목록에 띄운다.</li> </ol>
예외 흐름	E1. 상품의 재고가 없는 경우  "상품이 품절되었습니다"라는 메시지를 표시하고, 상품 주문 버튼을 비활성화한다.
대안 흐름	-
사후 조건	생성된 딤을 상품 상세 정보의 딤 목록에 표시한다.

### 2.2.13.기능 #13(딤 참여)

구분	내용
요구사항명	딤 참여
액터명	구매자
개요 및 설명	구매자가 원하는 상품의 진행중인 딤에 참여한다.
사전 조건	<ul style="list-style-type: none"> <li>- 사용자 ID로 로그인한 상태여야 한다.</li> <li>- 사용자가 원하는 상품을 선택한 상태이다.</li> </ul>

	- 해당 상품의 진행 중인 뽕이 존재한다.
기본 흐름	1. (시스템) 해당 상품 상세 정보를 띄운다. 2. (액터) 뽕 참여하기를 선택한다. 3. (시스템) 진행 중인 뽕 목록을 보여준다. 4. (액터) 원하는 뽕을 선택한다. 5. (액터) 주문 수량을 선택한다. 6. (액터) 상품 주문을 선택한다. 4. (시스템) 뽕 참여 정보를 업데이트하여 목록에 띄운다.
예외 흐름	E1. 입력한 주문 수량이 선착순 잔여 수량을 초과하는 경우  “최대 구매 수량은 n개 입니다”라는 메시지를 표시하고, 주문 수량을 재선택하도록 한다.
대안 흐름	-
사후 조건	선착순 재고가 수정된 뽕을 상품 정보의 뽕 목록에 표시한다.

#### 2.2.14.기능 #14(결제)

구분	내용
요구사항명	결제
액터명	구매자
개요 및 설명	참여한 뽕 주문에 대해 결제한다.
사전 조건	- 사용자 ID로 로그인한 상태여야 한다. - 사용자의 주문이 완료된 상태여야 한다.
기본 흐름	1. (시스템) 결제 방식을 보여준다. 2. (액터) 결제 방식을 선택한다. 3. (시스템) 결제 방식에 따라 결제 정보 입력을 요청한다.

	4. (액터) 결제 정보를 입력한다. 5. (시스템) 결제를 진행하고, 완료한다.
예외 흐름	E1. 잘못된 결제 정보가 입력된 경우 “잘못된 결제 정보입니다.”라는 메시지를 표시하고, 액터에게 결제 정보를 재요청한다. E2. 카드 결제의 경우, 결제를 실패한 경우 결제 실패 메시지를 표시하고, 다른 결제 정보를 재요청한다.
대안 흐름	-
사후 조건	결제 내역과 해당 주문 내역을 표시한다.

### 3. 기타 요구 및 제약 사항

#### 3.1. 성능 요구

● 반응 시간 – 3초. 사용자의 요청에 대한 최소한의 반응 시간의 기준
● 처리 소요 시간 – 클라이언트 처리시간 – 0.5초 / 서버 처리 시간 – 0.5초 / DB 처리 시간 – 1초
● 처리율 – 3건/시간

#### 3.2. H/W 요구

● 최소 Android 8.0 이상 버전을 사용할 수 있는 모든 스마트폰 ● Android Studio를 위한 Windows system requirements - 2GB RAM minimum, 4GB RAM recommended - 400MB hard disk space plus at least 1GB for Android SDK, emulator system images, and
--

### 3.3. 예외 조건 및 이의 처리

- 서버와의 통신이 끊겼을 때 - 오류 메시지 출력 → 빠른 시일내에 복구
- 검색 결과가 존재하지 않을 때 - 검색 결과를 빠르게 지우고 다시 입력 요청
- 형식에 맞지 않는 자료를 입력하였을 때 - 오류 메시지 출력 → 다시 입력하도록 유도
- 마켓의 아이디와 비밀번호를 임의로 부여 - 보안 철저히 유지
- 사용자 비밀번호 보안 철저히 관리

### 3.4. 자원, 인력에 대한 제약 조건

- 인력: 2명(팀원 전체)
- 서버관리의 문제
- 지역 마트의 협조 문제

## 4. 인수 조건

### 4.1. 기능 시험 및 성능 시험

- 마트의 상품을 토대로 공동구매가 진행되기 때문에 처음에 협력 업체를 구하는 것이 매우 중요하다. 처음 협력 업체에 해당하여 많은 지원을 해주면서 마트의 상품을 데이터베이스에 입력하여 상품 목록 확인, 검색기능, 상품 정보 조회, 공동구매방 생성까지 여러 기능들을 시험해본다.
- 참여자가 없으면 공동구매가 진행될 수 없기 때문에 지인들을 위주로 공동구매를 진행하여 문제가 없는지 시험한다.
- 공동구매방이 생성되고 참여자들이 모여 하나의 거래가 완료될 때까지 어느 정도의 기간과 시간이 걸리는지 시험한다.

- 각각의 기능들이 오류 없이 수행되는지 시험한다.
- 임의로 상품 QR 코드를 생성하여 QR 코드 리더기로 읽었을 때 잘 작동하는지 시험한다.

## 5. 참고 자료

- ① Sommerville, Software Engineering (8), Addison-Wesley
- ② 최은만, 객체지향 소프트웨어공학, 한빛미디어, 2017
- ③ Database System Concepts - Seventh Edition, Avi Silberschatz