**University Community based**

**Group purchasing platform**

Software Engineering Team 2

**Design Specification**

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

Contents

[1. Preface 5](#_Toc24560214)

[1.1 Readership 6](#_Toc24560215)

[2. Introduction 6](#_Toc24560216)

[2.1 objective 6](#_Toc24560217)

[2.2 Applied Diagram 6](#_Toc24560218)

[A. UML 6](#_Toc24560219)

[B. Package Diagram 7](#_Toc24560220)

[C. Deployment Diagram 7](#_Toc24560221)

[D. class diagram 7](#_Toc24560222)

[E. State diagram 8](#_Toc24560223)

[F. Sequence diagram 8](#_Toc24560224)

[G. ER diagram 8](#_Toc24560225)

[2.2 Applied Tool 9](#_Toc24560226)

[2.3 Project Scope 9](#_Toc24560227)

[3. System Architecture 11](#_Toc24560228)

[3.1 Objective 11](#_Toc24560229)

[3.2 Subcomponents 11](#_Toc24560230)

[A. Product Management System 11](#_Toc24560231)

[B. Search System 16](#_Toc24560232)

[C. 포인트 충전 시스템 19](#_Toc24560233)

[D. 구매확정/취소 알림 시스템 21](#_Toc24560234)

[E. User Management System 25](#_Toc24560235)

[F. Consumer Page System 29](#_Toc24560236)

[G. Seller Page System 31](#_Toc24560237)

[H. Group Purchasing System 33](#_Toc24560238)

[4. Protocol Design 38](#_Toc24560239)

[4.1. Protocol Description 38](#_Toc24560240)

[A. overview 38](#_Toc24560241)

[B. Login protocol – 구매자 38](#_Toc24560242)

[C. Login protocol – 판매자 38](#_Toc24560243)

[D. registration protocol – 구매자 39](#_Toc24560244)

[E. registration protocol – 판매자 39](#_Toc24560245)

[F. Product post protocol (판매자 제품 등록) 40](#_Toc24560246)

[G. Product view protocol (상품 상세 정보) 40](#_Toc24560247)

[H. product search protocol (제품 검색) 41](#_Toc24560248)

[I. sharing protocol (공유) 42](#_Toc24560249)

[J. buying protocol (구매) 42](#_Toc24560250)

[K. point charge protocol (포인트 충전) 42](#_Toc24560251)

[5. Database Design 43](#_Toc24560252)

[5.1 Objectives 43](#_Toc24560253)

[5.2 ER Diagram 43](#_Toc24560254)

[A. Entities 45](#_Toc24560255)

[5.3 SQL DDL 47](#_Toc24560256)

[A. Consumer 47](#_Toc24560257)

[B. Purchase 48](#_Toc24560258)

[C. Item 48](#_Toc24560259)

[D. sale\_info 48](#_Toc24560260)

[E. seller 49](#_Toc24560261)

[6. Test Plan 49](#_Toc24560262)

[6.1 Objective 49](#_Toc24560263)

[6.2 Testing policy 50](#_Toc24560264)

[6.3 Test case 50](#_Toc24560265)

[A. 회원가입 50](#_Toc24560266)

[B. 로그인 51](#_Toc24560267)

[C. 검색 51](#_Toc24560268)

[D. 포인트충전 52](#_Toc24560269)

[E. 구매 알림 52](#_Toc24560270)

[F. 상품 등록 52](#_Toc24560271)

[G. 상품 구매 53](#_Toc24560272)

[7. Development Environment 53](#_Toc24560273)

[7.1 Objective 53](#_Toc24560274)

[7.2 Github 53](#_Toc24560275)

[7.3 MySQL 54](#_Toc24560276)

[7.4 Firebase 54](#_Toc24560277)

[9. Index 55](#_Toc24560278)

[10.1 Figure 55](#_Toc24560279)

[10.2 Table 55](#_Toc24560280)

[10.3 Diagram 56](#_Toc24560281)

[11. References 56](#_Toc24560282)

# Preface

## Readership

본 문서는 다양한 독자에게 읽힐 것을 상정하고 있다. 따라서 각 부분을 서술하는 데 있어 어떠한 독자층을 상정하고 있는지를 설명한다.

# Introduction

## objective

System design specification에서 사용된 다양한 다이어그램들을 설명하고 시스템의 개발 범위를 설명한다.

## Applied Diagram

### UML

“그림은 천 단어의 가치가 있다” 라는 말이 있다. 다시 말하자면, 다이어그램으로 시각화된 시스템 구조가 자연어로 설명된 시스템 구조보다 이해하기 쉽고 직관적이다. UML은 Unified Modeling Language의 줄임말로, 통합된 다이어그램들로 구성된 표준화된 모델링 언어이다. 시스템과 소프트웨어 개발자들이 소프트웨어 artifacts들을 구체화, 시각화, 구성화 그리고 문서화하는데 효과적인 도구들이다. 모델링은 코드 작성 전 software application design을 의미한다. 모델링은 software의 청사진 같은 역학을 하므로 구체적인 모델링을 통해 소프트웨어 프로젝트의 전체적인 튼튼한 기초를 다질 수 있다. UML은 모델링을 위한 도구로 시스템의 design을 시각적으로 표현한다.

이번 system design specification에서 시스템의 구조를 설명하기 위해서 UML을 중점적으로 사용할 것이다. UML의 종류는 다음과 같다.

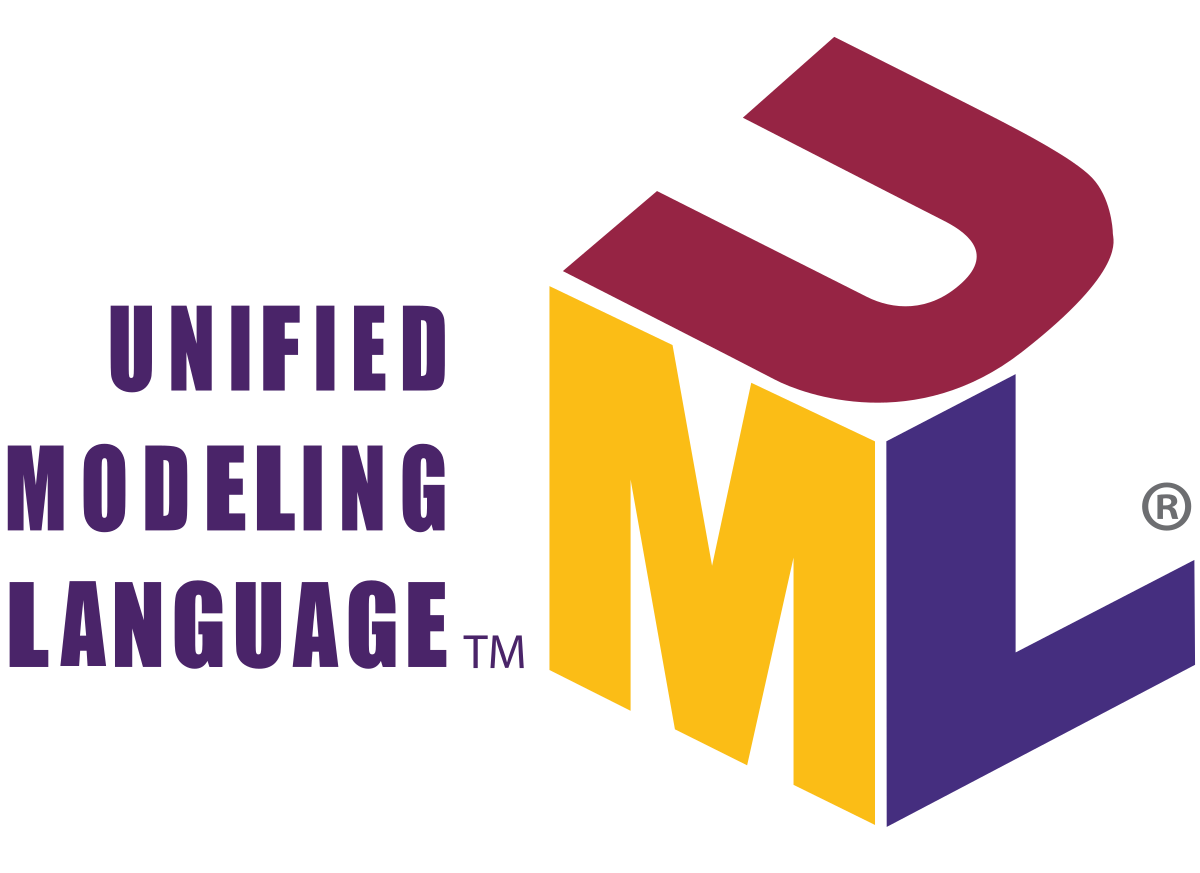


그림 1: UML logo

### Package Diagram

Package Diagram은 model 요소의 arrangement과 organization을 보여준다. Package Diagram은 하위 시스템 또는 모듈 사이의 구조와 의존성을 모두 보여줄 수 있다. 보통 높은 수준의 시스템 요소를 구성하는 데 사용된다. 복잡한 클래스 다이어그램을 단순화하기 위해 사용될 수 있으며, 클래스를 패키지로 그룹화할 수 있다. 보통 package diagram은 중대형의 소프트웨어 프로젝트에 사용되기 때문에 이번 design specification에서는 사용되지 않을 예정이다.

### Deployment Diagram

Deployment라는 용어 자체에서 설명되듯이 Deployment diagram은 소프트웨어 component가 deploy되는 하드웨어 구성요소를 설명하는데 사용된다. 이렇기 때문에 Component diagram과 Deployment diagram은 밀접하게 관련되어 있다.

Component diagram은 component와 각 component간의 관계를 설명하는 데 사용되며 deployment diagram은 이러한 component들이 하드웨어에 구현되는 방법을 보여준다

### class diagram

class diagram은 시스템의 클래스, 그 속성, 운영(또는 방법), 객체 간의 관계를 보여줌으로써 시스템의 구조를 설명하는 정적 구조 다이어그램의 유형이다. 관계라 함은 클래스들 간의 종속성이나 계급관계를 포함한다.

class diagram에서 클래스는 사각형으로 세 부분으로 나눠진다. 가장 상위에는 클래스 이름은, 그 아래는 attributes를, 가장 하단에는 methods를 작성한다. 화살표를 통해 class들 간의 관계를 나타낸다.

### State diagram

State diagram은 시스템의 behaviour를 추상화해 표현하는 도구이다. 특정 event가 발생했을 때 object의 모든 가능한 상태를 고려한 시스템의 행동을 기술하기 위해 사용하는 diagram이다. 이러한 행동은 하나 이상의 가능한 state에서 발생하는 일련의 사건에서 표현되고 분석된다.

직선 화살을 이용해 한 state에서 다른 state로의 전환을 나타내며, 둥근 사각형으로 state와 행해야 할 조건들을 표현한다.

### Sequence diagram

Sequence diagram은 단순히 actor와 object들 사이의 상호작용을 순차적 순서로 묘사한다. 이 도표는 requirement specification에서 유저의 use case를 설명할 때 사용되었다. 이 보고서에서 sequence diagram은 특정 시스템에서 actor와 object가 서로 상호작용하는 과정을 순차적으로 표현한다. 점선들은 object를 표현하며 화살표의 방향은 어느 object가 다른 object에게 상호작용을 요청하였음을 보여준다.

### ER diagram

ER은 높은 수준의 개념 데이터 모델 다이어그램이다. Entity-relationship model은 실제 entities들과 이들 사이의 관계에 기초한다.

ER 모델링은 잘 설계된 데이터베이스를 만들기 위해 데이터 요구사항을 체계적으로 분석하는 데 도움이 된다. 따라서 데이터베이스를 구현하기 전에 ER 모델링을 완료하는 것이 가장 좋은 관행으로 간주된다.

직사각형은 데이터베이스 entity를, 타원은 속성을, 다이아몬드는 entity간의 관계를 나타낸다.

## Applied Tool

이 부분에서는 실제 Diagram을 그리기 위해 사용된 도구를 설명한다.

1. Draw.io

시스템 구조를 설명하는데 가장 많이 사용되는 오픈소스 다이어그램 소프트웨어이다. [Draw.io](http://draw.io)는 기본적인 다이어그램 툴을 지원할 뿐만 아니라 UML과 ER 다이어그램 툴 또한 지원해 시스템 디자인을 시각화 하는데 유용한 도구이다. 조작이 쉽고 간결해 빠르게 다이어그램 생성이 가능하며 Google Drive와 통합되어 팀원들과 빠른 공유가 가능하다.

그림 2 : Draw.io logo

## Project Scope

대공구는 생필품과 같이 매일 같이 사용되는 물품들을 싼 가격에 구입하는 공동구매의 장점은 살리고 공동 구매의 문제점인 배송 문제를 해결하고자 하였다. 대공구는 대학생 커뮤니티를 기반으로 구매자는 본인의 대학교 학생들과 공동 구매가 가능하며 배송은 대학 내 특정 지역으로 배송된다. 공동 구매가 개별 배송에 어려움을 겪고 있다는 점을 해결하고자 함과 동시에 배송을 단순화해 대량 구매로 물품을 훨씬 값싼 가격에 구매할 수 있다. 물품은 최소 주문 수량을 넘으면 공동구매가 성립되며 판매자가 지정한 특정 수량을 넘을 때 마다 할인율이 증가하는 형식이다. 본 시스템의 크게 user management system과 구매 시스템, 물품 등록 시스템으로 구성 되어있다.

먼저 user management system은 기본적인 사용자 등록인 sign-in과 사용자 이용을 위한 log-in과 같은 user management subsystem이 존재하며 구매자와 판매자가 개인의 활동을 관리할 수 있는 user my page system 또한 포함된다.

구매 시스템이 본 프로젝트의 핵심이 되는 시스템으로서 사용자가 물품을 구매하는 전반적인 과정을 담당한다. 구매 시스템은 총 4개의 sub-system으로 구성되어 있는데 먼저 공동 구매 시스템은 구매자가 공동구매 물품을 선택하면 user의 구매리스트 database에 추가되며 공동 구매가 성사되어 판매자에게 주문이 들어가기까지 세세한 시스템 동작에 관여한다. 포인트 관리 시스템은 유저가 물품을 구매하기 위한 포인트 충전과 확인을 담당한다. 구매 알림 시스템은 유저가 공동 구매를 신청하였을 때 주문이 성사되면 구매가 성립되었다 알림을 보내는 서비스를 담당한다. 마지막으로 검색 시스템은 유저가 원하는 물품을 검색하고자 하였을 때 DB에 요청해 검색 조건에 일치하는 물품 리스트를 구매자에게 보여주는 시스템이다.

마지막으로 물품 등록 시스템은 판매자가 팔고자 하는 물품을 DB에 등록해 구매자의 페이지에 나타내는 시스템이다.

자세한 시스템 구조는 후에 설명한다.

# 3. System Architecture

## 3.1 Objective

이번 챕터에서는 본 시스템의 전체적인 구조를 설명한다. 시스템 전체의 구조와 각 서브시스템의 개략적인 구조, 서브시스템 간의 관계를 서술한다.

## 3.2 Subcomponents

### A. Product Management System

1. Class Diagram
2. Product Management System
3. Attributes

+ Seller: List<seller> - 판매자 리스트

+ item: List<item> - 물품 리스트

1. Methods

+ register\_product(List(item)): void – 상품 추가

+ delete\_product(name: string[]): void – 상품 제거

+ update\_product(name: string[]): void – 상품 업데이트

1. Seller
2. Attributes

+ name: string[] – 판매자 이름

+ companyID: int – 회사 고유 번호

1. Item
2. Attributes

+ itemID: int – 물품 고유 번호  
+ Period: timestamp – 공동구매 진행 기간

+ description: string[] – 물품 설명

+ cur\_price: int – 현재 판매가격

+ sale\_info: List<sale\_info> - 할인율 정보

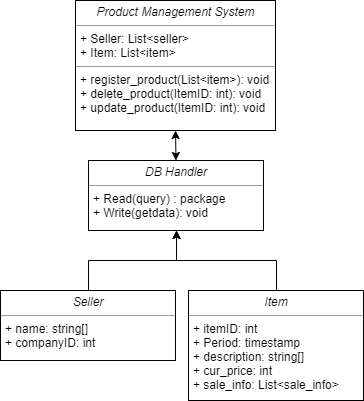


Diagram 1 : Class Diagram – Product Management System

1. Sequence Diagram

* 상품 등록

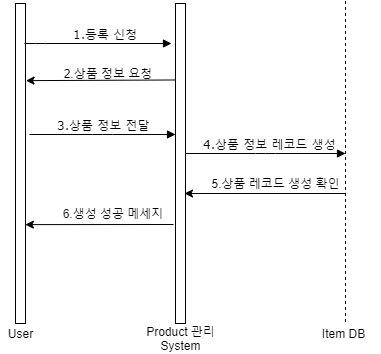


Diagram 2: Sequence Diagram – product management system(상품등록)

* 상품 삭제

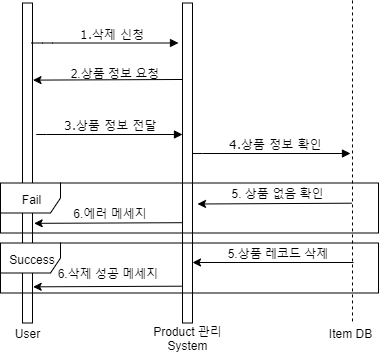


Diagram 3: Sequence Diagram – product management system(상품삭제)

* 상품 정보 변경

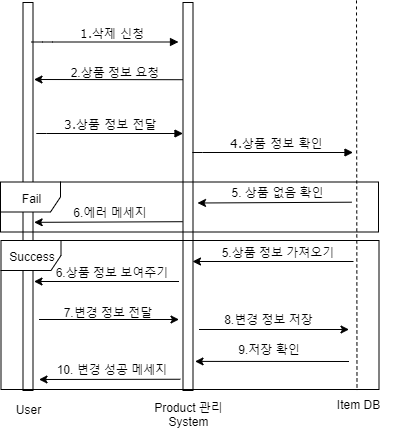


Diagram 4: Sequence Diagram – product management system(상품정보 변경)

1. State Diagram

* 상품 등록

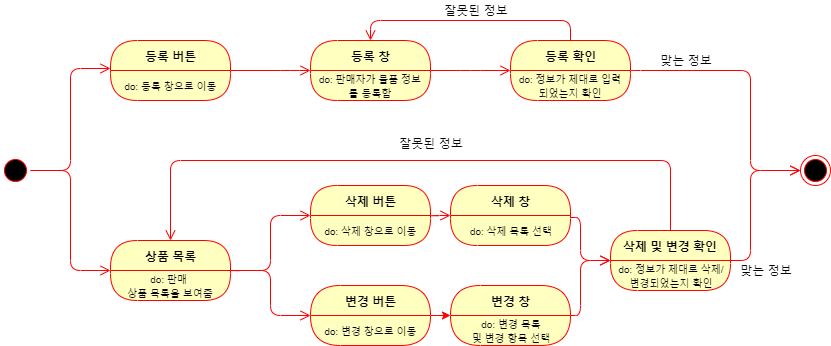


Diagram 5 : StateDiagram – product management system(상품등록)

### Search System

1. Objectives

유저, 그 중 구매자가 원하는 물건을 찾고자 했을 때 검색창에 물품의 이름을 검색하면 search system에서 상품들에 대한 정보를 DB에서 읽은 후 구매자에게 전송하는 시스템이다.

1. Class diagram

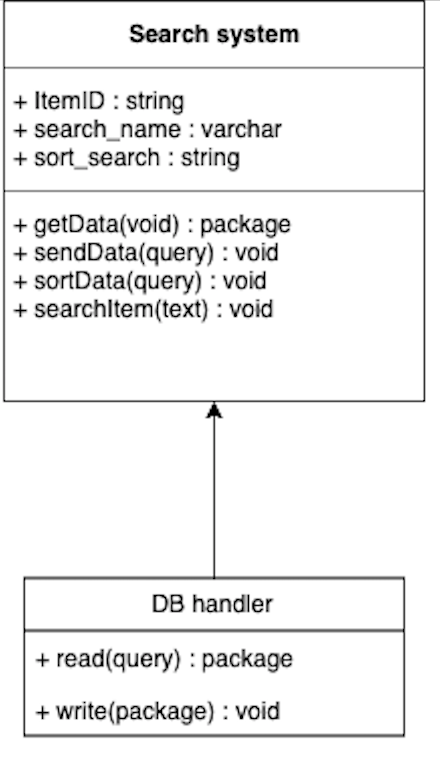


Diagram 6: Class Diagram – search system

1. DB hanlder
   1. Attribute

None

* 1. Methods

+ read(query): 해당 DB에서 원하는 데이터를 읽어온다

+ write(package): 해당 DB에 데이터를 입력한다

1. Search System
   1. Attributes

+ ItemID: string - 판매 상품

+ search: string - 검색어

* 1. Methods

+ getData(): 해당 DB에서 원하는 데이터를 읽어온다

+ sendData(query): 해당 DB에 데이터를 전송한다

+ sortData(query): DB에 정렬을 요청한다

+ searchItem(search): 검색어를 입력 받는다.

.

1. Sequence Diagram

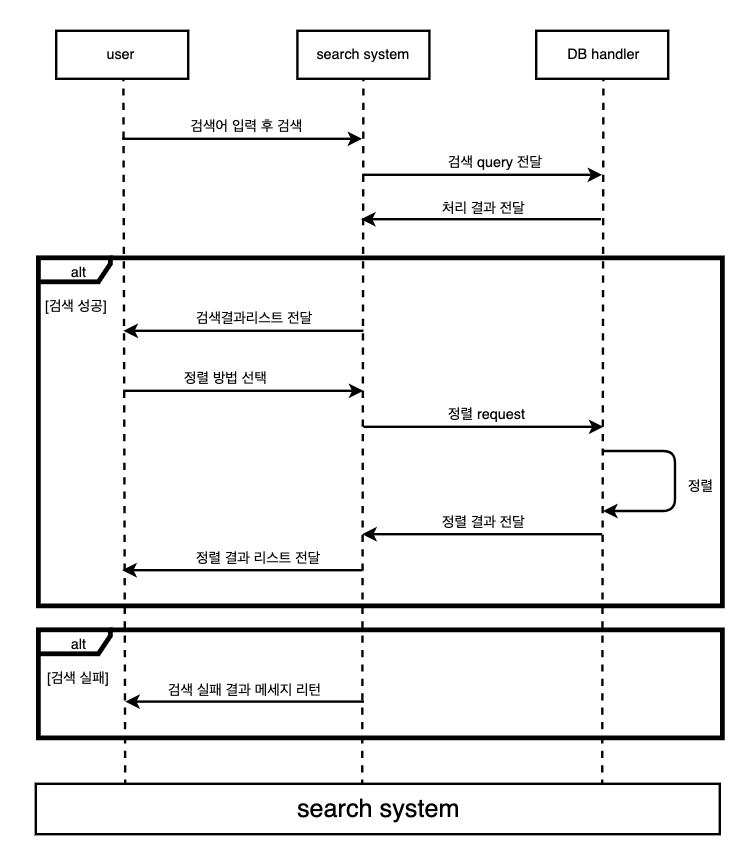


Diagram 7: Sequence Diagram – search system

1. state diagram

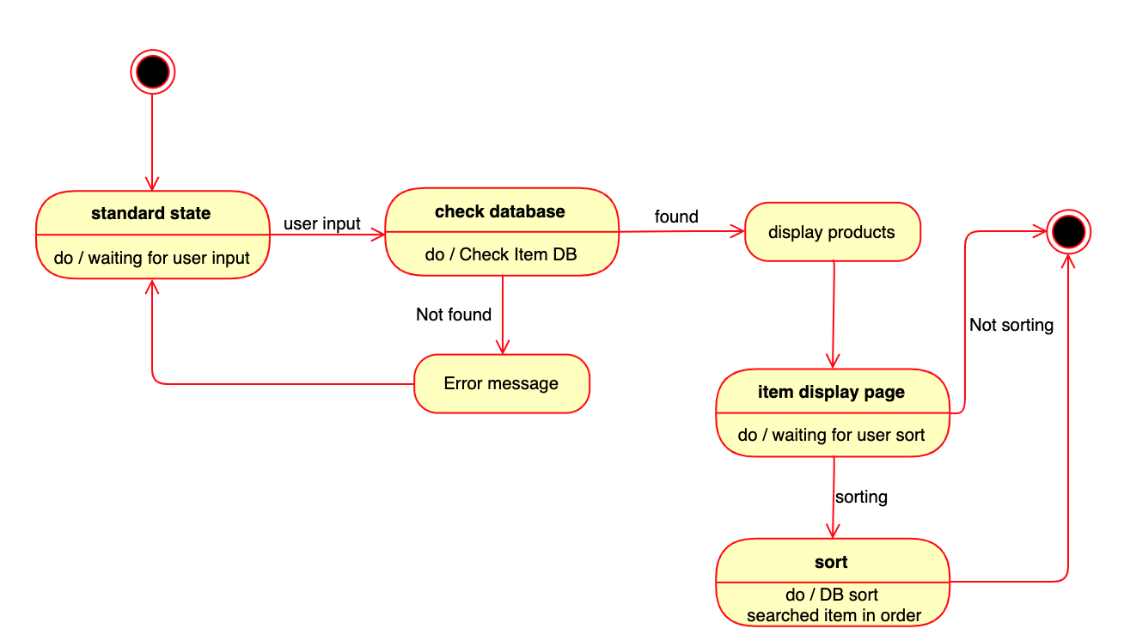


Diagram 8: State Diagram – search system

### 포인트 충전 시스템

1. Class diagram

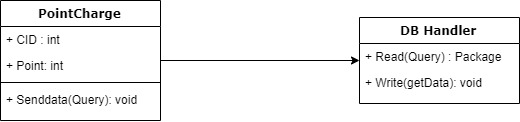


Diagram 9: Class diagram – 포인트 충전 시스템

A. DB Handler

A.1. Attributes

None

A.2. Methods

+read(Query) : 해당 DB에서 원하는 data를 읽어온다.

+write(getData) : 해당 DB에 data를 저장한다.

B. PointCharge

B.1. Attributes

+CID : 구매자의 고유번호

+Point : 구매자의 현재 Point

B.2. Methods

+senddata(query) : DB로 data를 보낸다.

1. Sequence diagram

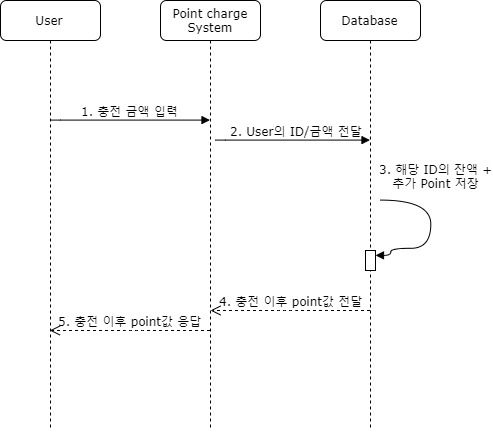


Diagram 10: Sequence diagram – 포인트 충전 시스템

1. State diagram

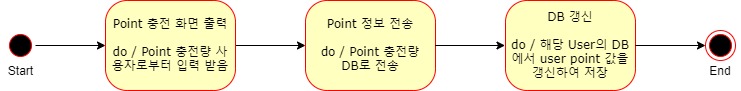


Diagram 11: State diagram – 포인트 충전 시스템

### 구매확정/취소 알림 시스템

1.Class diagram

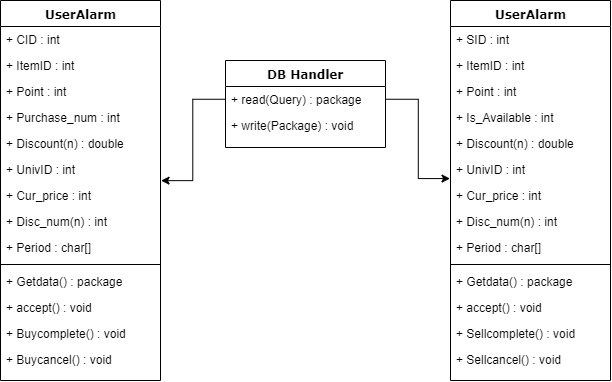


Diagram 12: Class diagram – 구매확정/취소 알림 시스템

A. DB Handler

A.1. Attributes

None

A.2. Methods

+read(Query) : 해당 DB에서 원하는 data를 읽어온다.

+write(getData) : 해당 DB에 data를 저장한다.

B. UserAlarm

B.1. Attributes

+CID : 구매자의 고유번호

+ItemID : 제품의 고유번호

+Point : 구매자의 현재 Point

+Purchase\_num : 구매자가 구매한 제품의 개수

+Discount(n) : 제품 판매 개수에 따른 제품의 할인율

+UnivID : 학교의 고유번호

+Cur\_price : 제품의 초기 개당 가격

+Disc\_num(n) : 제품 할인율에 따른 제품의 판매 개수

+Period : 제품 판매 만료일

B.2. Methods

+Getdata() : DB에서 data를 가져온다.

+accept() : 거래가 승낙되었는지 확인한다.

+Buycomplete() : 거래 성공시 실행되는 메소드

+Buycancel() : 거래 실패시 실행되는 메소드

1. SellerAlarm

C.1. Attributes

+SID : 판매자의 고유번호

+ItemID : 제품의 고유번호

+Point : 판매자의 Point

+Is\_available : 판매되고 있는 물품의 남은 개수

+Discount(n) : 제품 판매 개수에 따른 제품의 할인율

+UnivID : 학교의 고유번호

+Cur\_price : 제품의 초기 개당 가격

+Disc\_num(n) : 제품 할인율에 따른 제품의 판매 개수

+Period : 제품 판매 만료일

C.2. Methods

+Getdata() : DB에서 data를 가져온다.

+accept() : 거래가 승낙되었는지 확인한다.

+Sellcomplete() : 거래 성공시 실행되는 메소드

+Sellcancel() : 거래 실패시 실행되는 메소드

1. Sequence diagram

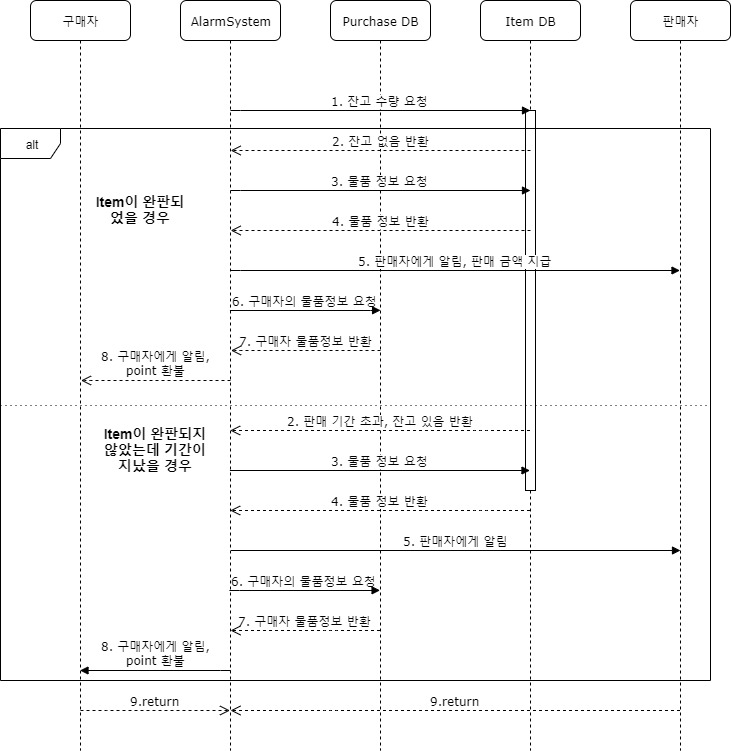


Diagram 13: Sequence diagram – 구매확정/취소 알림 시스템

1. State diagram

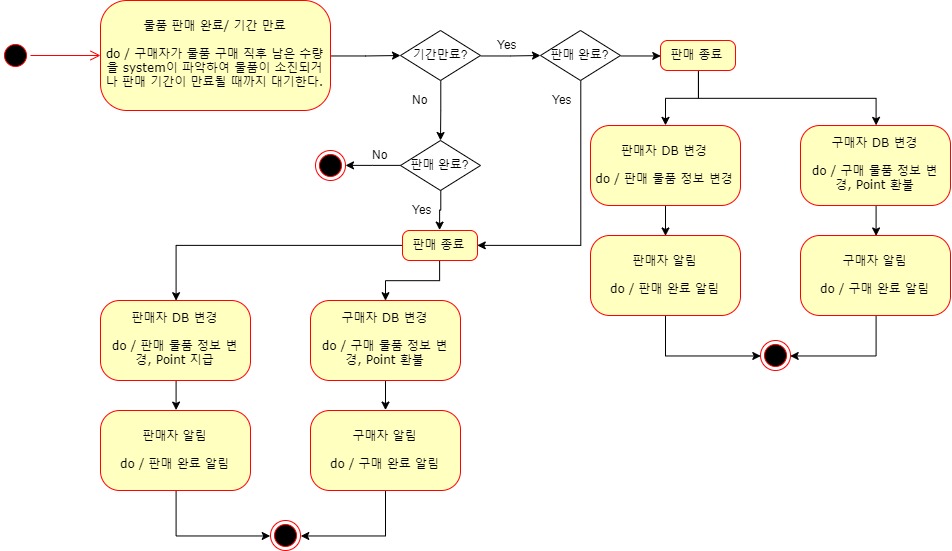


Diagram 14: State diagram – 구매확정/취소 알림 시스템

### User Management System

1. Objective

Explain the role of the user management system, the process of system processing when registering members and using the login function. Use the Class Diagram, Sequence Diagram, State Diagram and other diagrams to represent and describe the structure of the User Management System.

2. Class Diagram

图片包含 屏幕截图

描述已自动生成

Diagram 15: Class Diagram - User management System

A. DB Handle

a. Attributes

없음

b. Methods

+ package read(query): 해당 데이터베이스에서 원하는 데이터를 읽어온다.

+ void write(package): 해당 데이터베이스에 데이터를 저장한다.

B. Account

a. Attributes

+ username : 해당 계정의 ID

+ password: 해당 계정의 password

+ admin\_type : 해당 계정의 타입 (구매자/판매자)

+ UnivID: 이 계정 구매자의 학교

+ school\_email: 구매자의 학교 이메일, 인증 학교에 사용

+ companyID: 판매자의 회사 번호

b. Methods

+ package getData(): 해당 데이터베이스에서 원하는 데이터를 얻어온다.

+ void sendData(Package): 해당 데이터베이스에 데이터를 보낸다.

3. Sequence Diagram

A. 회원가입 및 로그인

图片包含 屏幕截图

描述已自动生成

Diagram 16: Sequence Diagram - User management System

4. State Diagram

A．회원가입

图片包含 文字

描述已自动生成

Diagram 17: State Diagram - User management System(회원가입)

B. 로그인

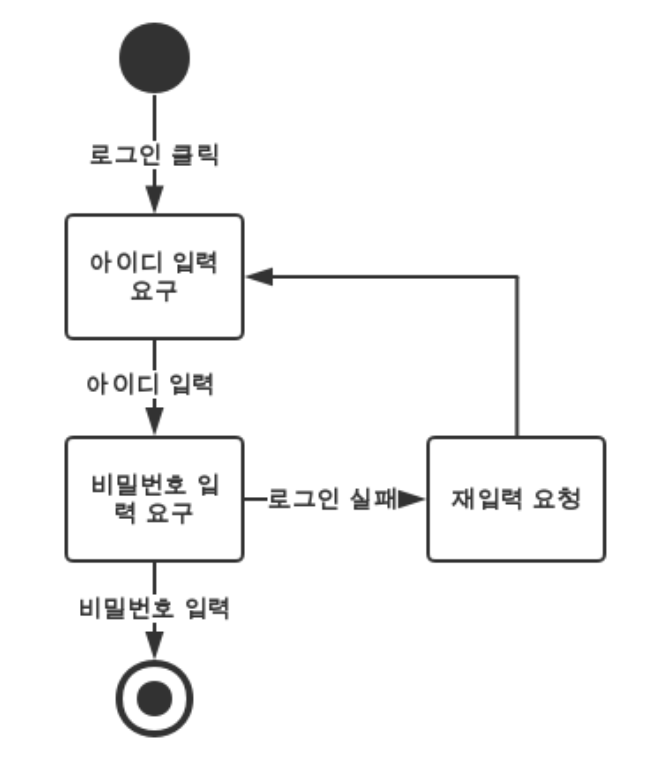


Diagram 18: State Diagram - User management System(로그인)

### Consumer Page System

1. Class Diagram
2. Consumer page

* Attribute

+ Consumer: List<Consumer>: 유저 정보

+ Purchase\_List: List<Item>: 구매 목록

+ UserAlarm: List <UserAlarm>: 실시간 알람

1. Consumer

* Attribute

+ CID: string[] - 아이디

+ name: string[] - 이름

+ UnivID: int – 학교 번호

+ point: int - 포인트

1. Purchase\_List

* Attribute

+ Item\_Name: string[] – 상품명

+ Period: timestamp – 남은 기간

+ Status: int – 현재 상태

+ discount 1,2,3 – 현재 할인율

* Method

none

1. User Alarm

* Attribute

+ CID: string[] – 아이디

+ PID: int – 공동구매 번호

+ point: int - 포인트

+ price: int - 가격

+ purchase\_num: int - 수량

+ discount1, 2, 3: double – 할인율

+ status: int – 상태

* Method

+ getMessage(int PID, int point, int price, double discount, int status): string[] : 메시지 출력

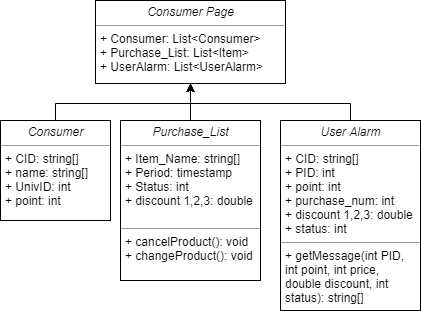


Diagram 19: Class Diagram – Consumer page system

### Seller Page System

1. Class Diagram
2. Mypage

* Attribute

+ Seller: List<seller> - 판매자 정보

+ Selling\_List: List<Item> - 판매 목록

+ UserAlarm: List <UserAlarm> - 실시간 알람

1. Seller

* Attribute

+SID: int - 아이디

+name: char[] - 이름

+point: int - 포인트

1. Selling\_List

* Attribute

+ Item\_Name: string[] – 상품명

+ Period: timestamp – 남은 기간

+ Status: int – 현재 상태

+ cur\_total: int – 판매 수량

+ discount 1,2,3 – 현재 할인률

* Method

+ cancelProduct(): void – 판매 상품 취소(완전 취소)

+ updateProduct(): void –상품 정보 변경(할인률 및 기간 연장)

1. User Alarm

+ SID: string[] – 아이디

+ PID: int – 공동구매 번호

+ point: int - 포인트

+ price: int - 가격

+ Number: int - 수량

+ discount1, 2, 3: double - 할인률

+ status: int – 상태

* Method

+ getMessage(int SID, int point, int price, double discount, int status): string[] : 메시지 출력

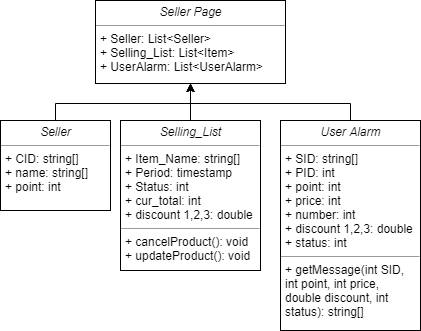


Diagram 20: Class Diagram – Seller page system

### Group Purchasing System

1. Class Diagram

A. Database Handler

A.1 Attributes

없음

A.2 Methods

+package read(query): DB에서 원하는 데이터를 가져온다

+void write(package): DB에 data를 저장한다.

B. Item

B.1 Attributes

+ItemID: int 상품 고유 번호

+name: String 상품 이름

+cur\_total: Number 현재 모인 금액

+Is\_Available: boolean 상품 이용가능 여부

+Period: TIMESTAMP 공동구매 진행할 기간

B.2 Methods

+getData(): package

+sendData(query): void

C. Purchasing state

C.1 Attributes

+ItemID: int 상품 고유 번호

+participants: list[] 현재 공동구매 참여자 목록

+UnivID: int 대학 고유 번호

+current\_price: double 할인율을 반영한 현재 가격

+period: TIME 공동구매 경과 시간

+Status: String 현재 상품의 상태(공구 진행 상태)

C.2 Methods

+getData(): package

+sendData(query) : void

E. User

E.1 Attribute

+UserID: int

+is\_seller: boolean

+name: char[]

+UnivID: int

+point: int

E.2 Methods

+getData(): package

+sendData(query) : void

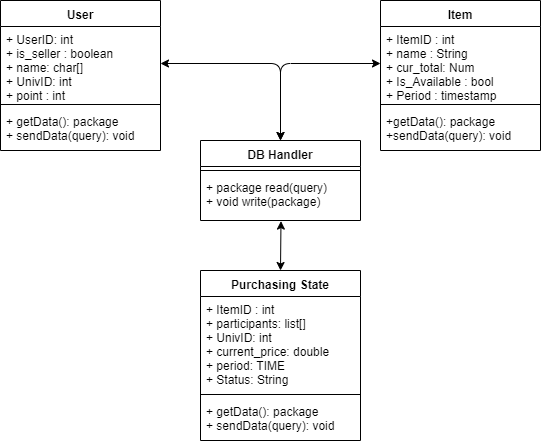


Diagram 21: Class Diagram – Group Purchasing System

1. Sequence Diagram

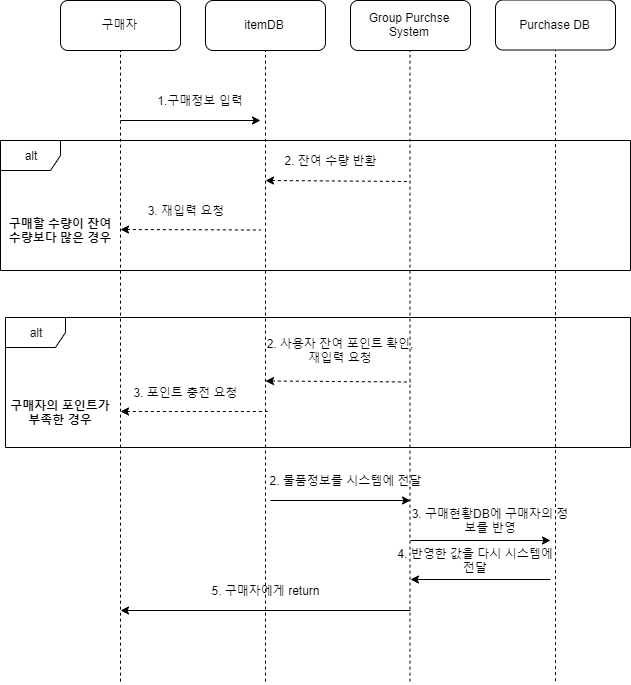


Diagram 22: Sequence Diagram – Group Purchasing System

1. State Diagram

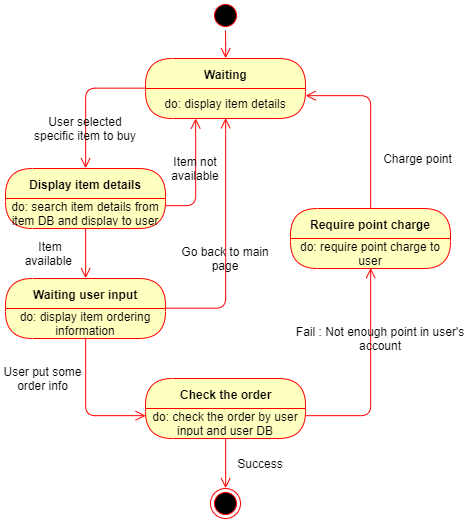


Diagram 23: State Diagram – Group Purchasing System

# 4. Protocol Design

## 4.1. Protocol Description

### A. overview

HTTP 통신에서 client와 server사이에서 전송되는 메시지의 형태를 용도 별로 정의한다. Client 에서의 요청(Request) 메시지와 Server에서의 응답(Response) 메시지로 구분한다. 해당 절의 표는 캡션을 생략한다.

### B. Login protocol – 구매자

B.1. Request

|  |  |
| --- | --- |
| Attribute | Value |
| CID | 구매자의 ID |
| Password | 사용자의 Password |

표 1: Login protocol – 구매자(Request)

B.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| Login\_success | 로그인 성공 여부 |

표 2: Login protocol – 구매자(Response)

### C. Login protocol – 판매자

C.1. Request

|  |  |
| --- | --- |
| Attribute | Value |
| SID | 판매자의 ID |
| Password | 사용자의 Password |

표 3: Login protocol – 판매자(Request)

C.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| Login\_success | 로그인 성공 여부 |

표 4: Login protocol – 판매자(Response)

### D. registration protocol – 구매자

D.1. Request

|  |  |
| --- | --- |
| Attribute | Value |
| CID | 구매자의 ID |
| Password | 구매자의 Password |
| Name | 구매자의 이름 |
| UnivID | 구매자의 학교 id |
| Email | 구매자의 E-mail |

표 5: Registration protocol – 구매자(Request)

D.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| Reg\_success | 회원가입 성공 여부 |

표 6: Registration protocol – 구매자(Response)

### E. registration protocol – 판매자

E.1. Request

|  |  |
| --- | --- |
| Attribute | Value |
| SID | 판매자의 ID |
| Password | 판매자의 Password |
| Name | 판매자의 이름 |
| E-mail | 판매자의 E-mail |

표 7: Registration protocol – 판매자(Request)

E.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| Reg\_success | 회원가입 성공 여부 |

표 8: Registration protocol – 판매자(Response)

### F. Product post protocol (판매자 제품 등록)

F.1. Request

|  |  |
| --- | --- |
| Attribute | Value |
| ItemID | 제품의 고유번호 |
| Cur\_price | 제품의 초기 가격 |
| Img\_src | 제품 이미지 url |
| description | 제품 설명 |
| SID | 판매자 ID |
| Is\_available | 제품 개수 |
| Period | 공동구매 기간 |
| Name | 제품 이름 |
| UnivID | 제품 판매할 학교 id |
| Discount(n) | 제품 판매 개수에 따른 제품의 할인율 |
| Disc\_num(n) | 제품 할인율에 따른 제품의 판매 개수 |

표 9: Product post protocol(request)

F.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| post\_success | 제품 등록 성공 여부 |

표 10: Product post protocol(response)

### G. Product view protocol (상품 상세 정보)

G.1. Request

|  |  |
| --- | --- |
| Attribute | Value |
| ItemID | 제품의 고유번호 |

표 11: Product view protocol(request)

G.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| Cur\_price | 제품의 초기 가격 |
| Img\_src | 제품 이미지 url |
| description | 제품 설명 |
| sellerID | 판매자 ID |
| Cur\_total | 제품 개수 |
| Period | 공동구매 기간 |
| Name | 제품 이름 |
| Is\_Available | 제품 남은 수량 |
| UnivID | 제품 판매할 학교 id |

표 12: Product view protocol(response)

### H. product search protocol (제품 검색)

H.1. request

|  |  |  |
| --- | --- | --- |
| Attribute | Value | |
| Search\_filter | Search\_word | 검색할 단어 |
| Search\_scope | 검색 범위 (카테고리/전부검색) |
| Search\_Sort | Sort\_time | 남은 기간 순 정렬 |
| Sort\_recent | 최근 등록 순 정렬 |

표 13: product search protocol(request)

H.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| Cur\_price | 제품의 초기 가격 |
| Img\_src | 제품 이미지 url |
| sellerID | 판매자 ID |
| Cur\_total | 제품 개수 |
| Period | 공동구매 기간 |
| Name | 제품 이름 |
| Is\_available | 제품 남은 수량 |
| UnivID | 제품 판매할 학교 id |
| Discount(n) | 할인율 |

표 14: product search protocol(response)

### I. sharing protocol (공유)

I.1. Request

|  |  |
| --- | --- |
| Attribute | Value |
| ItemID | 구매 한 제품의 고유번호 |

표 15: sharing protocol(request)

I.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| Sell\_url | 판매하는 제품의 url 링크 |

표 16: sharing protocol(response)

### J. buying protocol (구매)

J.1. Request

|  |  |
| --- | --- |
| Attribute | Value |
| ItemID | 구매 할 제품의 고유번호 |
| Purchase\_num | 구매 할 제품의 개수 |

표 17: buying protocol(request)

J.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| Buy\_success | 구매 성공/실패 여부 |

표 18: buying protocol(response)

### K. point charge protocol (포인트 충전)

K.1. Request

|  |  |
| --- | --- |
| Attribute | Value |
| CID | 구매자의 고유번호 |
| point\_charge | 충전할 포인트의 수량 |

표 19: point charge protocol(request)

K.2. Response

|  |  |
| --- | --- |
| Attribute | Value |
| charge\_success | 충전 성공/실패 여부 |

표 20: point charge protocol(response)

# 5. Database Design

## 5.1 Objectives

요구사항 명세서에서 작성한 데이터베이스 요구사항 기반으로 세부적 데이터 베이스 설계를 기술한다. ER 다이어그램을 통해 Entity 간의 관계를 기술하고 SQL DDL 명세를 만든다

## 5.2 ER Diagram

본 시스템에는 Consumer, Purchase, Item, sale\_info, seller로 총 5개의 Entity 가 존재한다. 각각의 Entity 는 네모 박스로 표현되고 그들간의 관계는 마름모꼴로 표현된다. 각 Entity가 가지는 Attribute는 타원형으로 표현되며 각 Entity를 식별하는 Primary Key는 라벨에 밑줄을 그어 표시한다.

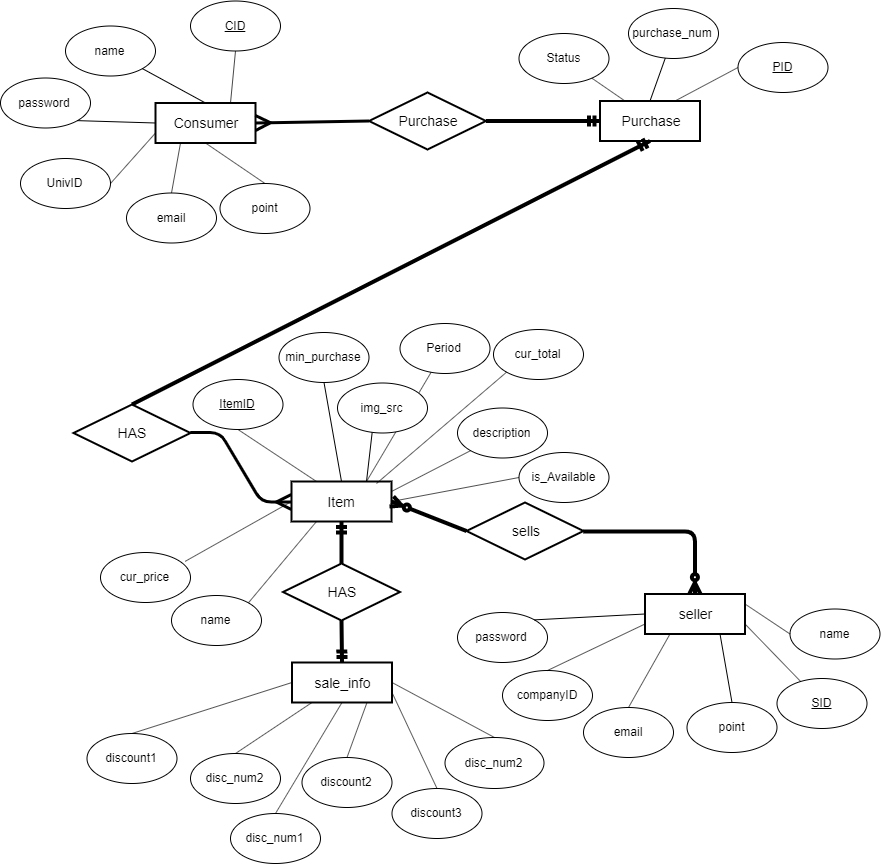


Diagram 24: ER Diagram

### Entities

#### a. Consumer

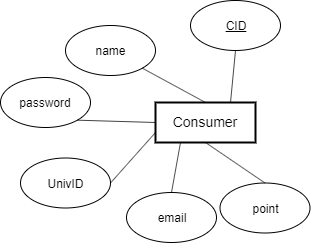


Diagram 25: ERD - consuemr

Consumer Entity는 소비자의 정보를 담고 있으며 primary key는 CID이다.

속성으로는 name, password, UnivID, email, point 가 있다

#### b. Purchase

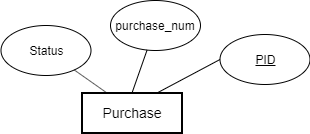


Diagram 26: ERD - purchase

Purchase entity는 구매정보를 담고 있으며 primary key는 PID 이다. 속성으로는 Status, purchase\_num 이 있다.

#### c. Item

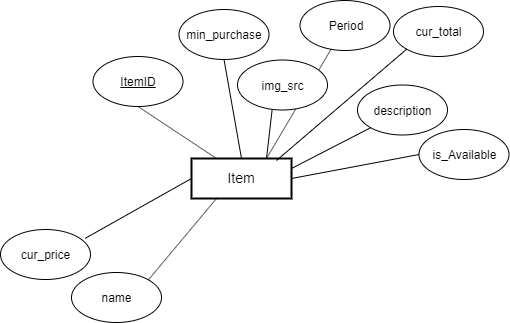


Diagram 27: ERD - item

Item Entity는 각각의 상품에 대한 정보를 담고 있으며 primary key는 ItemID 이다.

속성으로는 min\_purchase, img\_src, Period, cur\_total, description, is\_available, cur\_price, name이 있다.

#### d. sale\_info

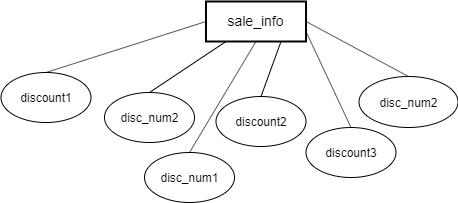


Diagram 28: ERD – sale\_info

Sale\_info entity는 각각의 상품에 대한 할인율 정보를 담고 있다. 속성으로는 discount1~3,

Disc\_num1~3 가 있다.

#### e. seller

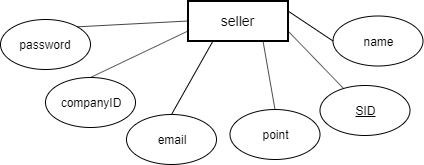


Diagram 29: ERD - seller

Seller entity는 판매자의 정보를 담고 있다. Primary key는 SID이고 속성으로는 password, companyID, email, point, name이 있다.

## 5.3 SQL DDL

### A. Consumer

|  |
| --- |
| CREATE TABLE Consumer  (  CID VARCHAR(30) NOT NULL,  name VARCHAR(10) NOT NULL,  UnivID VARCHAR(30) NOT NULL,  Email VARCHAR(30) NOT NULL,  Point INT NOT NULL  PRIMARY KEY (CID)  ); |

표 21: DDL - consumer

### Purchase

|  |
| --- |
| CREATE TABLE Purchase  (  PID INT NOT NULL,  Status VARCHAR(10) NOT NULL,  Purchase\_num INT NOT NULL,  PRIMARY KEY (PID)  FOREIGN KEY (CID) REFERENCES Consumer(CID),  FOREIGN KEY (ItemID) REFERENCES Item(ItemID)  ); |

표 22: DDL - purchase

### Item

|  |
| --- |
| CREATE TABLE Item  (  ItemID INT NOT NULL,  min\_purchase INT NOT NULL,  img\_src VARCHAR(100),  cur\_total INT NOT NULL,  description VARCHAR(300),  is\_Available VARCHAR(10) NOT NULL,  cur\_price INT NOT NULL,  name VARCHAR(30) NOT NULL,  Period TIMESTAMP NOT NULL,  PRIMARY KEY (ItemID),  FOREIGN KEY (PID) REFERENCES Purchase(PID),  FOREIGN KEY (SID) REFERENCES seller(SID)  ); |

표 23: DDL – Item

### sale\_info

|  |
| --- |
| CREATE TABLE sale\_info  (  Discount1 INT NOT NULL,  Discount2 INT NOT NULL,  Discount3 INT NOT NULL,  Disc\_num1 INT NOT NULL,  Disc\_num2 INT NOT NULL,  Disc\_num3 INT NOT NULL,  FOREIGN KEY (ItemID) REFERENCES Item(ItemID)  ); |

표 24: DDL – sale\_info

### seller

|  |
| --- |
| CREATE TABLE Item  (  name VARCHAR(10) NOT NULL,  SID INT NOT NULL,  point INT NOT NULL,  email VARCHAR(30) NOT NULL,  companyID INT NOT NULL,  password varchar(30) NOT NULL,  PRIMARY KEY (SID),  FOREIGN KEY (ItemID) REFERENCES Item(ItemID),  ); |

표 25: DDL – item

# 6. Test Plan

## 6.1 Objective

시스템이 설계한 의도대로 제대로 작동하고 내부의 결함을 검사하기 위해 설계단계에서 testing plan을 계획한다. 각 testing은 구현 과정에서 실행한다.

## 6.2 Testing policy

본 시스템의 Testing 과정은 development testing, release testing, user testing을 거치며 총 세가지의 testing으로 나눠진다

1. Development testing

본 Testing은 개발 과정에서 수행되는 testing으로 크게 세 가지로 나뉜다.

먼저 시스템을 component 단위로 개발한 후 각 component가 제대로 작동하는지 확인하는 component testing이 있다.

다음 각 component를 점진적으로 또는 부분적으로 합치며 그러한 과정에서 또는 합친 후 시스템이 제대로 작동하는지 확인하는 integrating testing이 있다.

마지막으로 모든 하위 시스템들을 하나로 합쳐 전체적인 시스템 동작을 검사하는 SYSTEM TESTING 이 있다.

1. Release testing

사용자에게 본격적으로 출시하기 전 마지막으로 사용자의 요구사항이 잘 받아들여졌는지 전체적인 시스템은 정상적으로 작동하는지를 검사한다.

1. User testing

사용자 환경에서 시스템이 제대로 동작하는지 검사한다.

## 6.3 Test case

### 회원가입

* User: 로그인 페이지에서 회원 가입 버튼을 누름
* 시스템 동작: 회원 가입 페이지로 이동한다
* User: 판매자/구매자 중 택한다
  + 시스템 동작: 판매자 유저는 판매자 회원가입 양식 페이지로, 구매자 유저는 구매자 회원가입 양식 페이지로 이동한다.
  + User: 필요한 양식을 채운다
  + 시스템 동작: 양식이 올바르게 채워졌는지 검사한다.
  + 구매자 case
    - 시스템 성공: 회원가입에 성공하였다는 메세지 출력
    - 시스템 실패:
      * 인증 번호 미입력/입력 오류 시 “인증번호를 확인하세요” 메세지 출력
      * 아이디 중복 미확인 시 “아이디 중복 미확인” 메세지 출력
  + 판매자 case
    - 시스템 성공 : 회원가입에 성공하였다는 메세지 출력
    - 시스템 실패:
      * 사업자등록번호 조회 실패 시 “입력한 사업자 등록번호를 확인해주세요” 메세지 출력
      * 아이디 중복 미확인 시 “아이디 중복 미확인” 메세지 출력

### 로그인

* User : 로그인 페이지에서 ID와 PW 입력한다.
* 시스템 동작 : DB에 저장된 정보가 user가 입력한 정보와 일치하는지 확인한다.
  + 시스템 성공 : 메인 페이지로 이동
  + 시스템 실패 : “입력하신 ID/PW를 확인해주세요” 메세지 출력

### 검색

* User : 원하는 상품을 검색한다.
* 시스템 동작 : 검색한 상품과 일치하는 물품을 DB에서 출력한다.
  + 성공 : 일치하는 상품을 보여준다
  + 실패: 일치하는 상품이 없다는 메세지 출력

### 포인트충전

* User : 포인트 관리 페이지에서 원하는 충전 금액을 입력한다.
* 시스템 동작 : 유저가 원하는 금액만큼의 포인트를 consumer DB에서 해당 유저의 보유 포인트에 더한다
  + 성공 : 포인트가 성공적으로 충전된다.
  + 실패 : 포인트가 추가되지 않는다.

### 구매 알림

* 시스템 동작 : 유저가 공동 구매를 신청한 물품이 성공적으로 성사가 되거나 기간이 지나 공동구매에 실패했을 경우 구매자와 판매자에게 알림을 보낸다.
  + 성공 : 해당되는 알림을 유저에게 보낸다
  + 실패 : 구매가 성사되었거나 실패되었음에도 불구하고 어떠한 알림도 보내지 않는다

### 상품 등록

* User : 판매자가 팔고자 하는 상품을 등록하거나 상품을 삭제 또는 변경한다.
* 시스템 동작:
* 1) 상품 등록:
  + User: 상품의 정보를 양식에 맞게 입력한 후 등록한다
  + 시스템 동작: 판매자가 등록한 상품정보를 Item DB에 저장
    - 성공: 성공 등록 메세지 출력
    - 실패: DB에 저장되지 않아 상품 등록이 안된다.
* 2) 상품 삭제 :
  + User : 자신이 등록한 상품 중 원하는 상품을 삭제
  + 시스템 동작 : 판매자가 등록한 상품을 DB에서 삭제
    - 성공 : 상품 레코드가 DB에서 성공적으로 삭제
    - 실패 : 오류 메세지 출력
* 3) 상품 변경 :
  + User : 등록한 상품 중 원하는 상품의 정보를 변경한다
  + 시스템 동작 : 유저가 전달한 변경된 상품 정보를 저장, 업데이트 한다.
    - 성공 : 상품 정보가 변경된다
    - 실패 : 오류 메세지 출력

### 상품 구매

* User : 상품의 상세정보 페이지에서 구매하고자 하는 상품의 수량을 선택 후 공동구매 등록한다.
* 시스템 동작 : 상품의 가격에 맞게 유저의 포인트를 차감한 후 해당 상품을 구매목록 DB에 추가한다
  + 성공 : 주문이 완료되었다는 메세지 출력
  + 실패 : “포인트를 충전해주세요” 알림 메세지 출력

# 7. Development Environment

## 7.1 Objective

Development environment에서 우리는 시스템을 개발할 때 필요한 개발 환경과 기술에 대해 설명하고 개발 일정과 진행 상황을 설명한다.

## 7.2 Github



그림 3: Github logo

GitHub 는 Git을 사용하여 소프트웨어 개발 버전 제어 를위한 호스팅 을 제공하는 미국 회사이다. Git의 모든 분산 버전 제어 및 소스 코드 관리 (SCM) 기능과 자체 기능을 제공한다. 액세스 제어 및 모든 프로젝트에 대한 버그 추적 , 기능 요청 , 작업 관리 및 위키 와 같은 여러 협업 기능을 제공한다. GitHub는 무료, 전문가 및 엔터프라이즈 계정에 대한 계획을 제공한다. 무료 GitHub 계정은 일반적으로 오픈 소스 프로젝트를 호스팅하는 데 사용한다.

## 7.3 MySQL



그림 4: MySQL Logo

MySQL은 관계 데이터베이스 관리 시스템(Relational Database Management System)이며, 관계 데이터베이스는 데이터를 서로 다른 표에 저장함으로써 속도를 늘리고 유연성을 향상시킨다. MySQL에서 사용되는 SQL 언어는 데이터베이스를 방문하기 위해 사용되는 가장 사용되는 언어이다.MySQL 소프트웨어는 커뮤니티 버전과 비즈니스 버전으로 분할되어 있으며, 크기가 작고 속도가 빠르며 비용이 낮기 때문에 일반적으로 중소형 사이트 개발은 사이트 데이터베이스로 MySQL을 선택한다.

## 7.4 Firebase



그림 5: Firebase logo

Firebase는 개발자가 웹 끝과 모바일 끝을 빠르게 작성할 수 있도록 도와주는 실시간 백엔드 데이터베이스 스타트업이다. 그 중에 Firebase Analytics，Firebase Cloud Messaging，Firebase Auth，Firebase Cloud Firestore，Firebase Storage，Firebase Hosting등 서비스를 포함된다.

# 8. Index

## 8.1 그림

[그림 1: UML logo 7](file:///C:\Users\Hyeon%20Subin\Desktop\공부\3-2\소프트웨어공학개론\요구사항%20명세서\Software%20Engineering%20Team%202.docx#_Toc24561809)

[그림 2 : Draw.io logo 9](file:///C:\Users\Hyeon%20Subin\Desktop\공부\3-2\소프트웨어공학개론\요구사항%20명세서\Software%20Engineering%20Team%202.docx#_Toc24561810)

[그림 3: Github logo 54](#_Toc24561811)

[그림 4: MySQL Logo 54](#_Toc24561812)

[그림 5: Firebase logo 55](#_Toc24561813)

## 8.2 표

[표 1: Login protocol – 구매자(Request) 38](#_Toc24561817)

[표 2: Login protocol – 구매자(Response) 38](#_Toc24561818)

[표 3: Login protocol – 판매자(Request) 38](#_Toc24561819)

[표 4: Login protocol – 판매자(Response) 39](#_Toc24561820)

[표 5: Registration protocol – 구매자(Request) 39](#_Toc24561821)

[표 6: Registration protocol – 구매자(Response) 39](#_Toc24561822)

[표 7: Registration protocol – 판매자(Request) 39](#_Toc24561823)

[표 8: Registration protocol – 판매자(Response) 40](#_Toc24561824)

[표 9: Product post protocol(request) 40](#_Toc24561825)

[표 10: Product post protocol(response) 40](#_Toc24561826)

[표 11: Product view protocol(request) 40](#_Toc24561827)

[표 12: Product view protocol(response) 41](#_Toc24561828)

[표 13: product search protocol(request) 41](#_Toc24561829)

[표 14: product search protocol(response) 41](#_Toc24561830)

[표 15: sharing protocol(request) 42](#_Toc24561831)

[표 16: sharing protocol(response) 42](#_Toc24561832)

[표 17: buying protocol(request) 42](#_Toc24561833)

[표 18: buying protocol(response) 42](#_Toc24561834)

[표 19: point charge protocol(request) 42](#_Toc24561835)

[표 20: point charge protocol(response) 42](#_Toc24561836)

[표 21: DDL - consumer 47](#_Toc24561837)

[표 22: DDL - purchase 48](#_Toc24561838)

[표 23: DDL – Item 48](#_Toc24561839)

[표 24: DDL – sale\_info 49](#_Toc24561840)

[표 25: DDL – item 49](#_Toc24561841)

## 8.3 Diagram

[Diagram 1 : Class Diagram – Product Management System 12](#_Toc24560283)

[Diagram 2: Sequence Diagram – product management system(상품등록) 13](#_Toc24560284)

[Diagram 3: Sequence Diagram – product management system(상품삭제) 14](#_Toc24560285)

[Diagram 4: Sequence Diagram – product management system(상품정보 변경) 15](#_Toc24560286)

[Diagram 5 : StateDiagram – product management system(상품등록) 16](#_Toc24560287)

[Diagram 6: Class Diagram – search system 17](#_Toc24560288)

[Diagram 7: Sequence Diagram – search system 18](file:///C:\Users\Hyeon%20Subin\Desktop\공부\3-2\소프트웨어공학개론\요구사항%20명세서\Software%20Engineering%20Team%202.docx#_Toc24560289)

[Diagram 8: State Diagram – search system 19](#_Toc24560290)

[Diagram 9: Class diagram – 포인트 충전 시스템 19](#_Toc24560291)

[Diagram 10: Sequence diagram – 포인트 충전 시스템 20](#_Toc24560292)

[Diagram 11: State diagram – 포인트 충전 시스템 21](#_Toc24560293)

[Diagram 12: Class diagram – 구매확정/취소 알림 시스템 21](#_Toc24560294)

[Diagram 13: Sequence diagram – 구매확정/취소 알림 시스템 24](#_Toc24560295)

[Diagram 14: State diagram – 구매확정/취소 알림 시스템 25](#_Toc24560296)

[Diagram 15: Class Diagram - User management System 26](#_Toc24560297)

[Diagram 16: Sequence Diagram - User management System 27](#_Toc24560298)

[Diagram 17: State Diagram - User management System(회원가입) 28](#_Toc24560299)

[Diagram 18: State Diagram - User management System(로그인) 29](#_Toc24560300)

[Diagram 19: Class Diagram – Consumer page system 31](#_Toc24560301)

[Diagram 20: Class Diagram – Seller page system 33](#_Toc24560302)

[Diagram 21: Class Diagram – Group Purchasing System 35](#_Toc24560303)

[Diagram 22: Sequence Diagram – Group Purchasing System 36](#_Toc24560304)

[Diagram 23: State Diagram – Group Purchasing System 37](#_Toc24560305)