

# Design Specification

---

## PICKET

---



제출일	2019. 11. 13	그룹	Group 4
과목	소프트웨어공학개론	교수	이은석 교수님
이름	김우경	학번	2017312637
이름	문재완	학번	2017312554
이름	주혜원	학번	2015314089
이름	성창민	학번	2017310339

# Contents

<b>1. Preface .....</b>	<b>6</b>
<b>1.1 Objective .....</b>	<b>6</b>
<b>1.2 Readership .....</b>	<b>6</b>
<b>1.3 document Structure.....</b>	<b>6</b>
<b>A. Preface .....</b>	<b>6</b>
<b>B. Introduction .....</b>	<b>6</b>
<b>C. System Architecture.....</b>	<b>6</b>
<b>D. Frontend System .....</b>	<b>7</b>
<b>E. Backend System.....</b>	<b>7</b>
<b>F. Protocol Design .....</b>	<b>7</b>
<b>G. Database Design .....</b>	<b>7</b>
<b>H. Testing Plan.....</b>	<b>7</b>
<b>I. Development Plan .....</b>	<b>8</b>
<b>J. Index .....</b>	<b>8</b>
<b>1.4 Version History .....</b>	<b>8</b>
<b>2. Introduction .....</b>	<b>8</b>
<b>2.1 Objective .....</b>	<b>8</b>
<b>2.2 Applied Diagram .....</b>	<b>8</b>
<b>A. Unified Modeling Language (UML).....</b>	<b>9</b>
<b>B. Package Diagram.....</b>	<b>11</b>
<b>C. Deployment Diagram.....</b>	<b>12</b>
<b>D. Class Diagram .....</b>	<b>12</b>
<b>E. State Diagram .....</b>	<b>13</b>
<b>F. Sequence Diagram .....</b>	<b>14</b>

G. ER Diagram.....	15
2.3 Applied Tool.....	15
A. PowerPoint.....	15
2.4 Project Scope.....	16
3. System Architecture .....	17
3.1 Objective .....	17
3.2 System Organization .....	17
A. Frontend System .....	18
B. Backend System .....	18
4. Frontend System .....	19
4.1 Objective .....	19
4.2 Overall Architecture.....	20
4.3 Sub-Systems.....	21
A. User Management System.....	21
B. Shopping Cart System.....	23
C. Review Classify System .....	27
D. User Request System .....	29
5. Backend System.....	31
5.1 Objective .....	31
5.2 Overall Architecture.....	31
5.3 Sub-Systems.....	32
A. Authentication System.....	32
B. Item Configure System .....	37
C. Review Analysis System.....	41
D. User Requirement System.....	44

<b>6. Protocol Design</b>	47
<b>6.1 Objective</b>	47
<b>6.2 JSON</b>	47
<b>6.3 Protocol Description</b>	48
<b>A. Register Protocol</b>	48
<b>B. Login Protocol</b>	49
<b>C. Save Cart Protocol</b>	49
<b>D. Item Configure Protocol</b>	49
<b>E. Review Analysis Protocol</b>	50
<b>F. User Domain Request Protocol</b>	50
<b>G. User Else Request Protocol</b>	51
<b>7. Database Design</b>	51
<b>7.1 Objective</b>	51
<b>7.2 ER Diagram</b>	51
<b>A. Entities</b>	52
<b>B. Relations</b>	53
<b>7.3 Relational Schema</b>	54
<b>7.4 SQL DDL</b>	55
<b>A. Domain</b>	55
<b>B. User</b>	55
<b>C. Items</b>	56
<b>8. Testing Plan</b>	56
<b>8.1 Objective</b>	56
<b>8.2 Testing Policy</b>	56
<b>A. Development Testing</b>	56

B. Release Testing .....	58
C. User Testing.....	59
8.3 Testing Case .....	59
A. Authentication System.....	59
B. Item Configure System .....	61
C. Review Analysis System.....	61
D. User Requirement System.....	62
9. Development Plan .....	62
9.1 Objective .....	62
9.2 Frontend Environment .....	63
A. Google Chrome Extension .....	63
9.3 Backend Environment.....	63
A. Python.....	63
B. MySQL .....	64
9.4 Schedule.....	64
A. Gantt Chart.....	64
10. Index.....	65
10.1 Objective .....	65
10.2 Table Index.....	65
10.3 Figure Index .....	65
10.4 Diagram Index .....	66

# 1. Preface

## 1.1 Objective

Preface에서는 본 요구사항 명세서의 예상 독자를 정의하고 문서의 전반적인 구조와 각 부분의 역할을 기술한다. 또한, 문서의 버전과 변경사항들을 그에 대한 근거와 함께 서술한다.

## 1.2 Readership

본 문서는 다양한 독자층에게 읽힐 것을 상정하고 있다. 본 시스템을 직접 개발하는 소프트웨어 엔지니어, 본 시스템의 구조를 설계하는 시스템 아키텍처, 본 시스템의 유지보수에 관여하는 유지보수 엔지니어, 그 외 본 시스템의 개발에 참여하는 모든 구성원을 본 문서의 독자로 간주한다.

## 1.3 document Structure

본 문서는 총 10개의 장으로 구성되어 있다. 10개의 장은 Preface, Introduction, System Architecture, Frontend System, Backend System, Protocol Design, Database Design, Testing Plan, Development Plan, Index로 구성되어 있다. 각 장에서 서술되는 내용 및 역할은 다음과 같다.

### A. Preface

Preface에서는 본 요구사항 명세서의 예상 독자를 정의하고 문서의 전반적인 구조와 각 부분의 역할을 기술한다. 또한, 문서의 버전과 변경사항들을 그에 대한 근거와 함께 서술한다.

### B. Introduction

Introduction에서는 본 문서에서 사용되는 모든 종류의 다이어그램과 표현 도구들을 소개 및 서술한다. 또한, 본 프로젝트가 다루는 시스템의 범위에 대해 서술한다.

### C. System Architecture

System Architecture에서는 본 시스템에 대한 고수준의 추상화를 제시한다. 시스템의 전체적인 구조를 보여주고, 시스템의 기능들이 각 서브시스템과 하드웨어에 어떻게 할당되는지 개괄적으로 설명한다.

## **D. Frontend System**

Frontend System에서는 본 시스템에서 Frontend에 구현이 되는 기능들을 설명한다. 세부적으로 5개의 sub-system로 구성된다. 4개의 sub-system은 User Management System, Shopping Cart System, Review Classify System, User Request System으로 구성되어 있다. Class diagram, sequence diagram, state diagram을 사용해 frontend system의 구조를 표현한다.

## **E. Backend System**

Backend System에서는 본 시스템에서 Backend에 구현이 되는 기능들을 설명한다. 세부적으로 4개의 sub-system로 구성된다. 4개의 sub-system은 Authentication System, Item Configure System, Review Analysis System, User Requirement System으로 구성되어 있다. Class diagram, sequence diagram, state diagram을 사용해 backend system의 구조를 표현한다.

## **F. Protocol Design**

Protocol Design에서는 Frontend와 Backend 시스템간의 통신 프로토콜에 대하여 기술한다. 통신 과정에서 사용되는 기술 및 메시지(Message)의 형식과 용도, 의미 등을 상세하게 기술한다.

## **G. Database Design**

Database Design에서는 요구사항 명세서에서 기술한 데이터베이스 요구사항을 바탕으로 더 상세하게 각 속성(attribute)간의 관계 등을 기술한다. ER diagram, Relation Schema, SQL DDL을 사용하여 설명한다.

## **H. Testing Plan**

Testing Plan에서는 Test Policy와 Test Case에 대해 설명한다. 본 장의 목적은 본 시스템이 의도한대로 실행이 되는지 검증하기 위함이다. Test Policy에서는 test의 단계

적 접근을 설명하고, Test Case에서는 test에 사용할 입력(input) 및 기대되는 출력(output)을 서술한다.

## I. Development Plan

Development Plan에서는 본 시스템을 개발하는데 사용되는 프로그래밍 언어, 소프트웨어 프레임워크, 라이브러리 등 개발 환경에 대해서 설명한다. 또한, Gantt Chart를 이용하여 개발 계획과 실제 개발 흐름에 대해 서술한다.

## J. Index

Index에서는 본 문서에서 사용된 그림, 다이어그램 및 테이블의 색인을 제공한다.

## 1.4 Version History

Version	Modified Date	Explanation
1.0	2019. 11. 03	디자인 명세서 목차 작성
1.1	2019. 11. 04	디자인 명세서 형식 작성
2.0	2019. 11. 05	디자인 명세서 Preface 작성
2.1	2019. 11. 06	Overall Architecture, Backend System 작성
2.2	2019. 11. 09	Frontend System 작성
2.3	2019. 11. 10	Protocol Design, Database Design, Testing 작성
3.0	2019. 11. 12	나머지 파트 작성 및 통합
3.1	2019. 11. 13	마지막 검토 및 수정 완료

[Table 1] Version History

# 2. Introduction

## 2.1 Objective

Introduction에서는 본 문서에서 사용되는 모든 종류의 다이어그램과 표현 도구들을 소개 및 서술한다. 또한, 본 프로젝트가 다루는 시스템의 범위에 대해 서술한다.

## 2.2 Applied Diagram



## A. Unified Modeling Language (UML)

통합 모델링 언어(Unified Modeling Language, 이하 UML)는 객체지향 소프트웨어 설계에 사용하는 표준설계 언어로 하나의 시스템을 표현하기 위해 만들어졌다. UML의 역사를 살펴 보면 “쓰리 아미고 (Three Amigos)”라고 불리는 세 사람, 그래디 부치(Grady Booch), 제임스 럼버(James Rumbaugh), 이바 야콥슨(Ivar Jacobson)에 의해 80년대 전반부터 90년대 초반까지 각자의 연구 영역에서 연구되어 왔다. 그 후, 객체 관리 그룹 (OMG, Object Management Group)에서 객체 모델링 기술 (OMT, Object Modeling Technique), 객체 지향 소프트웨어 공학 (OOSE, Object Oriented Software Engineering) 방법론 등을 통합하여 UML을 발표하였다.



[Figure 1] UML Logo

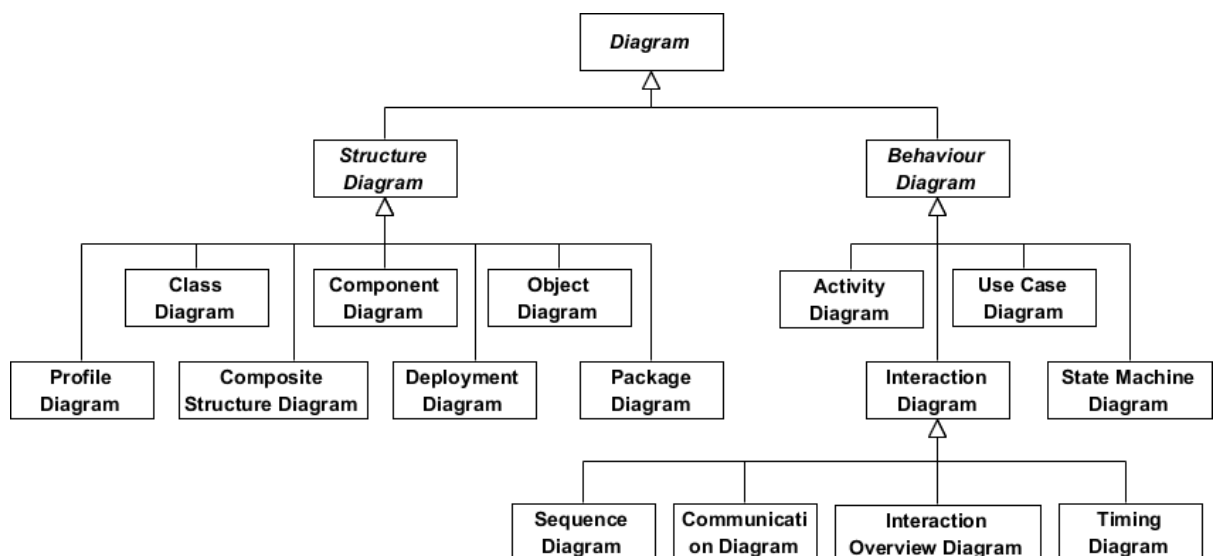
많은 회사에서 소프트웨어의 전략적 가치가 증가함에 따라, 소프트웨어 생산의 자동화, 품질 개선, 비용 및 출시 시간을 단축하기 위한 기술을 찾고 있다. 또한, 시스템의 범위와 규모가 증가함에 따라 시스템의 복잡성을 관리하기 위한 기법 및 반복적인 아키텍처 문제를 해결해야 할 필요가 있어졌다. UML은 이러한 요구에 대응하도록 설

계되어 있다.

UML은 사용자에게 즉시 사용가능하고 표현력이 강하며 직관적인 시각적 모델링 언어를 사용한다는 특징이 있다. 따라서, 시스템을 구현하기 앞서 UML을 통해 표준화 되고 이해하기 쉬운 방식으로 설계가 가능하며, 시스템 개발에 관련된 모든 사람들과의 상호 소통을 가능하게 하는 효율적인 매개체 역할을 한다.

UML은 총 13가지의 다이어그램이 존재한다. 다양한 다이어그램이 많은 이유는 여러 가지 다른 관점에서 시스템을 보는 것이 가능하기 때문이다. 소프트웨어 개발은 시스템 디자이너, 시스템 개발자, 시스템 테스터, 고객 등 다양한 이해 당사자들의 개입이 있고 각각의 사람들은 시스템의 다른 측면에 관심을 가지고 있다. 그렇기 때문에 각각의 세부사항들은 서로 다른 수준을 요하게 된다. 예를 들어 시스템 개발자는 시스템 설계를 이해하고 낮은 수준의 코드로 설계를 변환 할 수 있어야 하는 반면, 시스템 디자이너는 시스템의 전체적인 행동에 관심이 있으며, 제품이 어떻게 기능하는지를 이해할 필요가 있다. 따라서, UML은 모든 이해 당사자들이 적어도 하나의 UML 다이어그램으로부터 이익을 얻을 수 있도록 표현 가능한 언어를 제공하려고 시도한다.

13개의 UML 다이어그램은 Class Diagram, Component Diagram, Deployment Diagram, Object Diagram, Package Diagram, Composite Structure Diagram, Profile Diagram, Use Case Diagram, Sequence Diagram, Communication Diagram, Interaction Overview Diagram, Timing Diagram으로 구성된다.



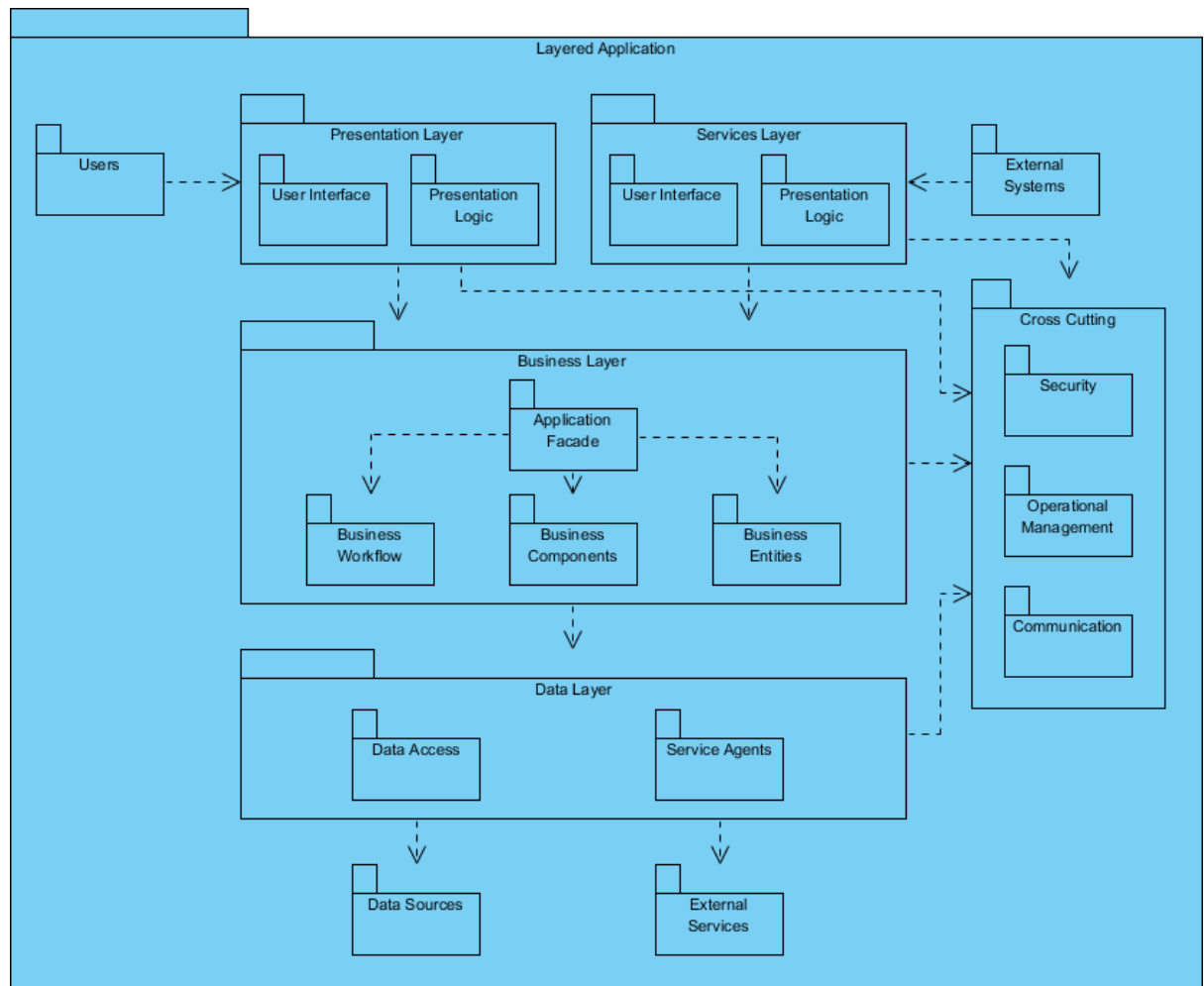
[Figure 2] UML Diagrams

## B. Package Diagram

Package Diagram은 패키지 간의 종속성 및 구조를 보여주는 UML 다이어그램이다. Package Diagram은 복잡한 시스템을 이해하기 위해 높은 수준의 시스템 요소를 서술한다.

패키지(Package)는 논리적으로 연관된 UML 요소의 집합이다. 패키지를 구성할 때에는 여러 사람이 동의할 수 있는 형태로 구성되어야 하며, 패키지가 여러 개 중첩된 계층 구조를 따른다.

패키지는 상단에 작은 탭이 있는 직사각형으로 나타내지며, 패키지 이름은 탭 또는 사각형 내부에 쓰인다. 점 선을 이뤄진 화살표는 의존성을 의미한다.



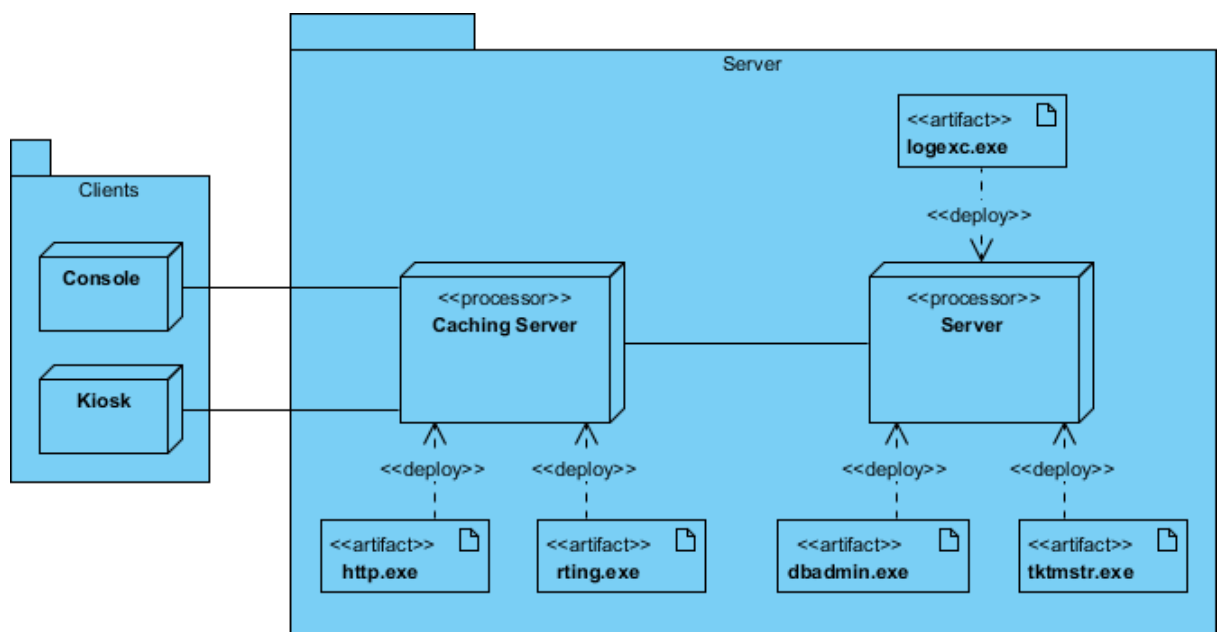
[Figure 3] Package Diagram Example

## C. Deployment Diagram

Deployment Diagram은 시스템을 구성하는 하드웨어 자원 간의 연결 관계를 표현하고, 하드웨어 자원에 대한 소프트웨어 컴포넌트의 배치 상태를 표현한 다이어그램이다. 컴포넌트를 분산된 하드웨어 자원에 적절히 배치하여 원하는 성능과 효율을 내는 것에 목표를 둔다.

Deployment Diagram은 시스템 설계의 마지막 단계에 작성이 되는데, 이는 모든 설계가 마무리 되어야 소프트웨어의 컴포넌트가 정의가 되고 시스템의 하드웨어 자원의 사양도 확정이 되기 때문이다.

노드(node)는 3-D 박스 형태로 표현이 되며 소프트웨어나 하드웨어를 나타낸다. 하드웨어 노드는 <<stereotype>>로 나타낼 수 있으며, 노드 간의 관계는 선으로 표현한다.



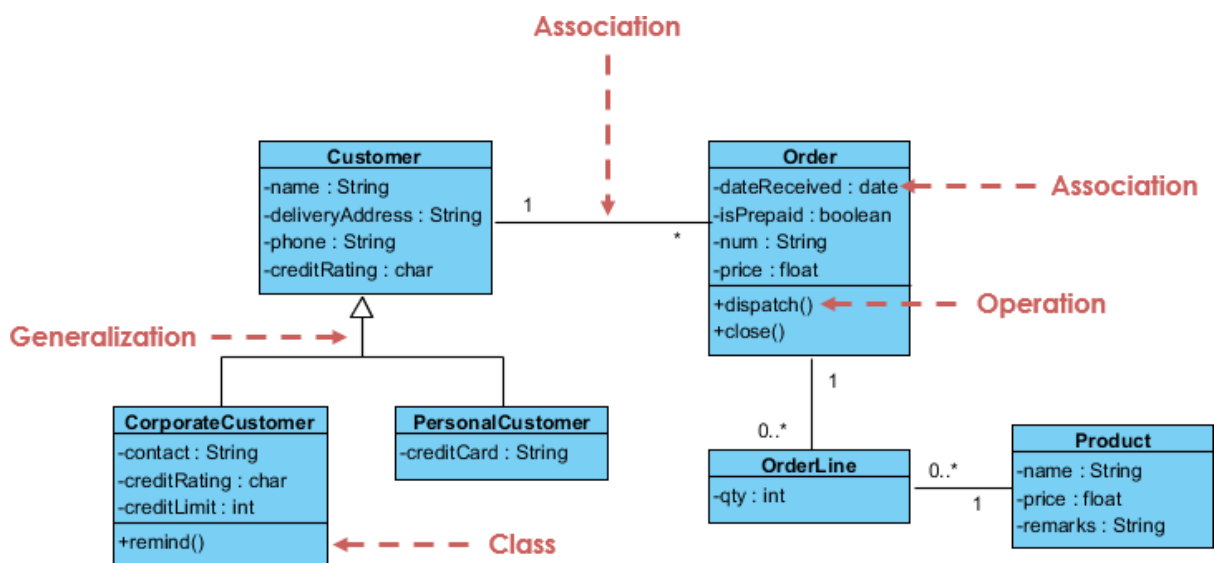
[Figure 4] Deployment Diagram Example

## D. Class Diagram

Class Diagram은 시스템의 클래스(Class), 그 속성(Attribute) 및 메소드(Method), 객체 간의 관계를 보여줌으로써 시스템의 구조를 설명하는 정적 구조 다이어그램의 일종이다. Class Diagram은 객체 지향에서 가장 중요한 요소로, 코드 생성에 있어서 집적적인 원인이 되기 때문에 프로그래밍 개념에서 사용되는 표현을 통해 도식화한다.

클래스란 시스템에서 유사한 역할을 가진 개체 그룹에 대한 설명으로, 클래스의 구조적 또는 정적 특성에 대한 설명인 속성과 클래스의 행동적 또는 동적 특성을 설명하는 메소드로 구성된다.

클래스는 직사각형으로 표현이 되고, 세 부분으로 구성된다. 첫 번째 파티션에는 클래스 이름, 두 번째 파티션에는 속성, 세 번째 파티션에는 메소드가 표시가 된다. 클래스는 하나 이상의 다른 클래스와의 관계에 포함될 수 있으며, 그 관계는 유형에 따라 직선에 화살표와 같은 도형을 붙여서 나타낸다.



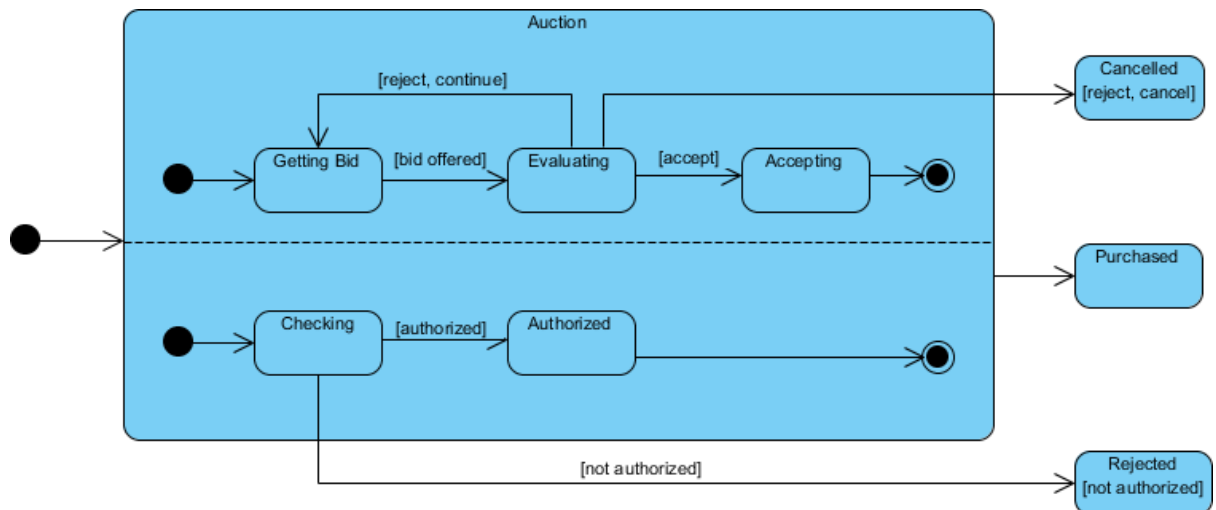
[Figure 5] Class Diagram Example

## E. State Diagram

State Diagram은 사건(Event)에 의한 하나의 상태에서 다른 상태로의 변화, 상태의 변화에 의한 동작을 나타낸다. 하나의 오브젝트(Object)가 생명주기 동안 통과하는 상태들을 발생하는 순서대로 명시한 행동이며 오브젝트는 사건에 도달하면 반응하고 응답을 하게 된다.

상태(State)란 제임스 럼버의 말에 따르면 객체의 속성 값과 링크를 추상화한 것이. 즉, 오브젝트가 생명주기 동안 가질 수 있는 조건이며 특정 조건이 만족된 상태에서 활동을 수행하고 사건을 기다리게 된다.

상태는 가장자리가 둥근 직사각형으로 표시가 되며, 상태 이름과 그 상태와 관련이 있는 활동(Activity)을 그 밑에 표시한다. 상태의 전이는 화살표로 나타낸다.

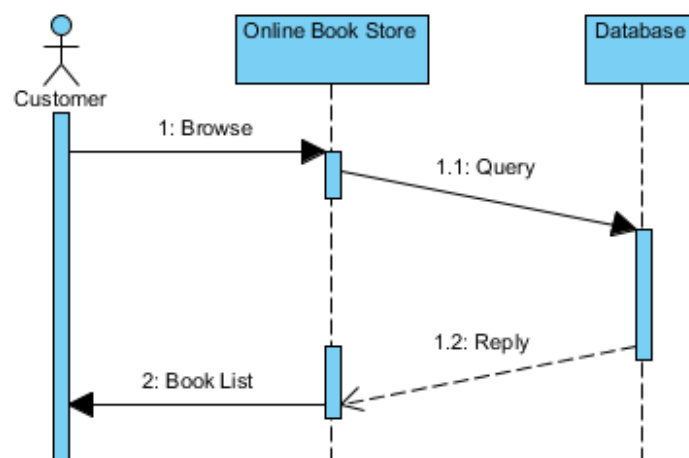


[Figure 6] State Diagram Example

## F. Sequence Diagram

Sequence Diagram은 문제 해결을 위한 객체를 정의하고 객체간의 상호작용 메시지를 시간의 흐름에 따라 나타내는 다이어그램이다. Sequence Diagram은 각 컴포넌트들 간의 상호작용이 명확하게 보이기 때문에 컴포넌트들 사이의 행동들을 더욱 명확하게 할 수 있다.

Sequence Diagram에서 생명선(Life Line)은 상호작용에 참여하는 오브젝트를 의미하고 점선으로 표시한다. 메시지(Message)는 서로 다른 객체간의 상호작용 또는 의사소통을 정의하는 요소로 하나의 객체에서 다른 하나의 객체 생명선까지 화살표로 표시되며 메시지는 그 화살표 위에 표시한다. 활성 박스(Activation Box)는 객체의 생명선 위에 그려지는 박스로 이 박스 위에서의 객체의 호출이 이루어진다.

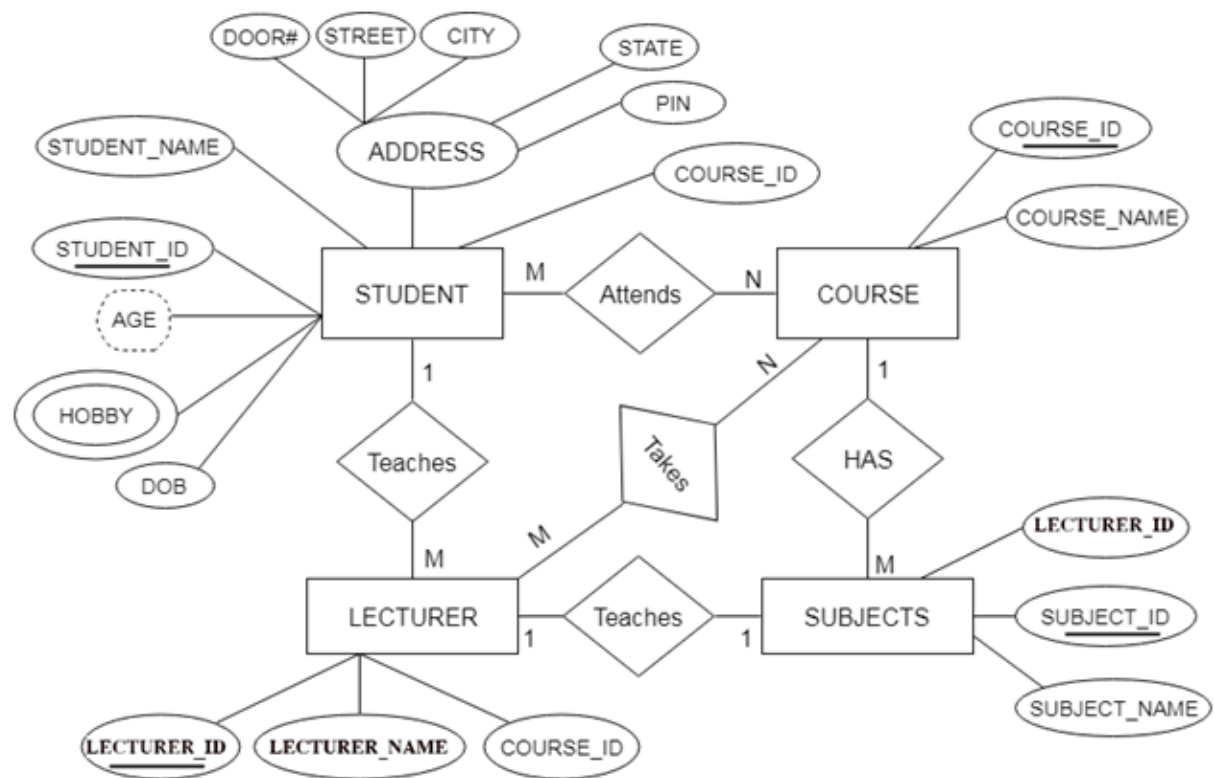


[Figure 7] Sequence Diagram Example

## G. ER Diagram

ER Diagram은 계략적으로 데이터 및 데이터들의 관계를 표현한 다이어그램으로, UML에 포함되는 다이어그램은 아니다. 데이터가 저장되는 데이터베이스의 구조 및 그에 수반한 제약 조건들을 정의하는 하나의 방식인 개체-관리 모델링(Entity-Relationship Modeling)의 산출물이다.

개체(Entity)란 현실 세계에 존재하는 실체를 의미하며 서로 구별이 될 수 있는 것이어야 한다. 개체 간의 관계는 관계로 표현이 되며 관계에도 특성이나 상태를 기술하는 속성을 가진다.



[Figure 8] ER Diagram Example

## 2.3 Applied Tool

### A. PowerPoint

본 명세서에서 사용되는 다이어그램들은 Microsoft PowerPoint를 이용하여 작성되었다. Microsoft PowerPoint는 마이크로소프트사의 마이크로소프트 오피스 시스템에서 개발된 프레젠테이션 소프트웨어이다. 본래의 용도는 발표용이지만, 내장된 도형

작성 기능이 매우 잘 구현이 되어 있기 때문에 복잡한 다이어그램을 작성하기 위해서 활용하였다.



[Figure 9] Microsoft PowerPoint Logo

## 2.4 Project Scope

본 시스템(PICKET)은 분사된 인터넷 쇼핑몰 사이트를 하나의 플랫폼으로 통합하는 시스템으로, 다양한 인터넷 쇼핑몰을 이용할 때 발생하는 불편한 점을 해소하기 위한 목적을 가진다.

본 시스템의 핵심 기능은 여러 쇼핑몰 사이트에 존재하는 상품들을 본 시스템의 장바구니에 장바구니에 담는 기능이다. 또한, 리뷰 서비스 기능 및 상품 분류기능을 제공하여 사용자가 상품을 선택하는 것을 도와준다.

본 명세서에서는 본 시스템을 크게 Frontend System과 Backend System으로 나누어 설명을 한다. Frontend System은 사용자에게 보여지고 사용자의 입력을 받는 부분으로, User Management System, Shopping Cart System, Review Classify System, User Request System 총 4가지 부분으로 나뉜다. Backend System은 사용자의 입력을 받아드려 리뷰 분석, 파싱 등 서버에서의 작업을 수행하는 부분으로, Authentication System, Item Configure System, Review Analysis System, User Requirement System, 총 4가지 부분으로 나뉜다.

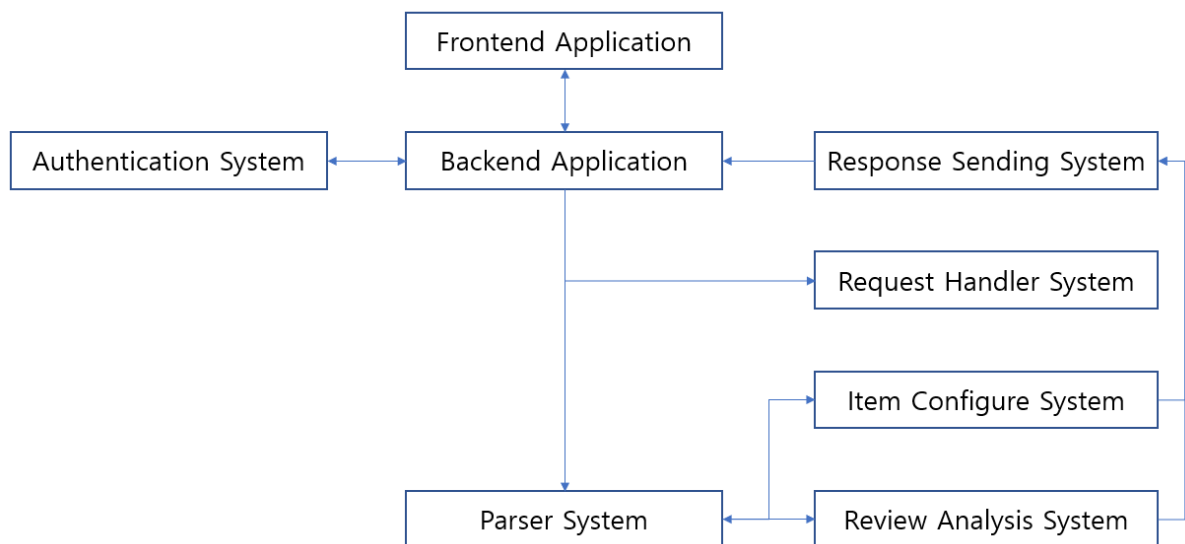


## 3. System Architecture

### 3.1 Objective

System Architecture에서는 본 시스템에 대한 고수준의 추상화를 제시한다. 시스템의 전체적인 구조를 보여주고, 시스템의 기능들이 각 서브시스템과 하드웨어에 어떻게 할당되는지 개괄적으로 설명한다.

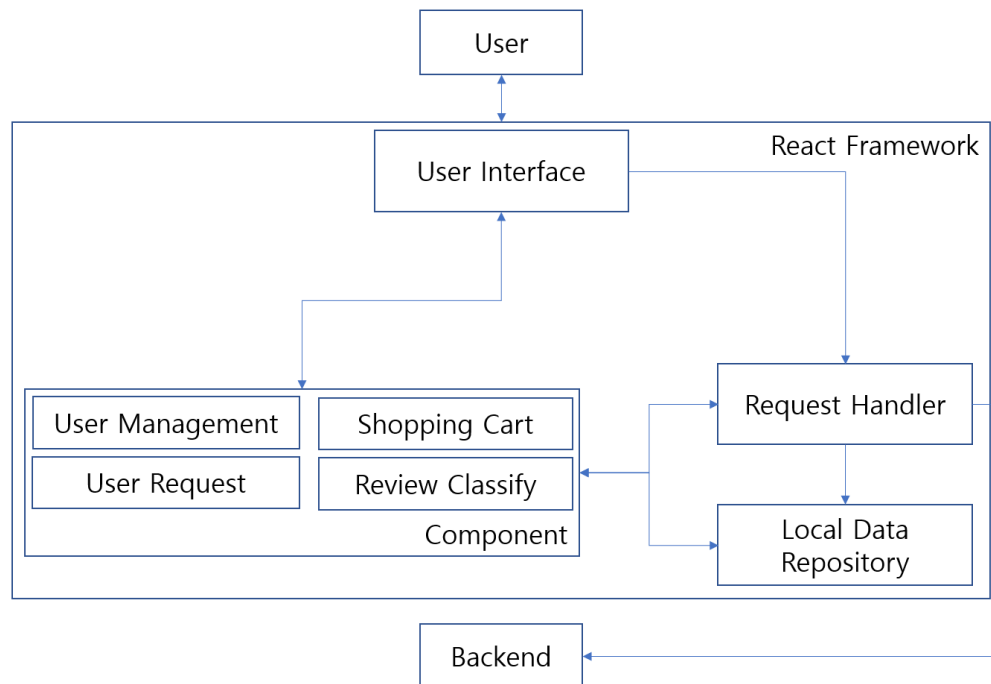
### 3.2 System Organization



**[Diagram 1] System Organization Block Diagram**

본 시스템은 클라이언트-서버 모델을 적용해 디자인되었다. Frontend 시스템과 은 로그인, 회원가입 등을 지원하는 User Management System, 본 시스템의 핵심 기능인 Shopping Cart System, 새로운 도메인을 요청하는 Domain Request System 등을 통해 고객이 Backend System과 상호작용 할 수 있도록 한다. Frontend와 Backend System은 JSON을 기반으로 한 http 통신을 통해 정보를 교환한다. Backend는 Frontend로부터 발생한 고객의 요청을 종류에 따라 적절한 서브 시스템으로 전달하고, 각각의 서브 시스템에서는 전달받은 정보를 가공하여 Response Sending System을 통해 Backend System에게 정보를 전달한다.

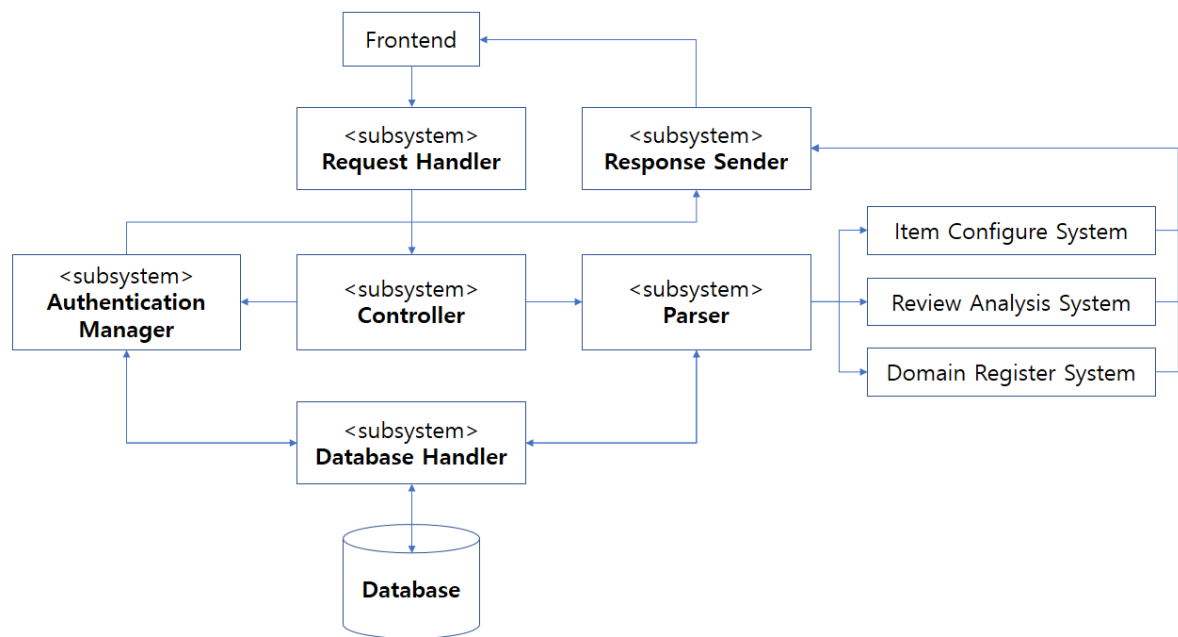
## A. Frontend System



[Diagram 2] Frontend System Block Diagram

Frontend System은 유저와의 상호작용을 담당하며, React Framework를 통해 각각의 컴포넌트를 관리한다. 컴포넌트로는 로그인, 회원가입을 담당하는 User Management, PICKET의 핵심 기능을 담당하는 Shopping Cart, 사용자의 요구사항을 전달하는 User Request, 리뷰를 분류하는 Review Classify의 4가지 Component가 존재한다. 위의 컴포넌트에 사용되는 데이터는 Local Data Repository에 저장될 수 있다. 컴포넌트의 정보를 바탕으로 User Interface subsystem에서 사용자에게 직접적으로 나타나는 UI를 모델링 하게 된다. 또한, 유저의 요청이 발생할 경우, Request handler에 우선적으로 요청을 보내며, 요청의 종류에 따라 필요한 정보를 컴포넌트나 Local Data Repository에서 불러와 Backend System과 통신할 수 있는 프로토콜을 생성한다.

## B. Backend System



[Diagram 3] Backend System Block Diagram

Frontend로부터 넘어온 요청은 가장 처음으로 Request Handler에 의해 분석이 된다. Request handler는 요청이 요구하는 바가 무엇인지를 인식하여 이를 Controller가 받아들일 수 있는 형태로 가공하여 Controller에게 전달한다. Controller는 입력을 바탕으로 (Authentication manager, Parser) 중 하나에게 수행할 작업에 처리해야 하는 정보를 전달한다. Authentication manager는 로그인 정보가 유효한지를 확인하고, 유효한 경우, Database Handler에게 유저의 stateful information을 제공해달라는 요청을 보낸다. Parser는 유저가 요청한 행동에 대응하는 웹 페이지를 파싱하여 sub-system에 (Item Configure System, Review analysis system, Domain Register system)에게 전달한다. 이들의 처리 결과물은 Response Sender에게 전달되어 응답 메시지를 만들어 이를 요청한 frontend에 전달하는 방식으로 작동한다.

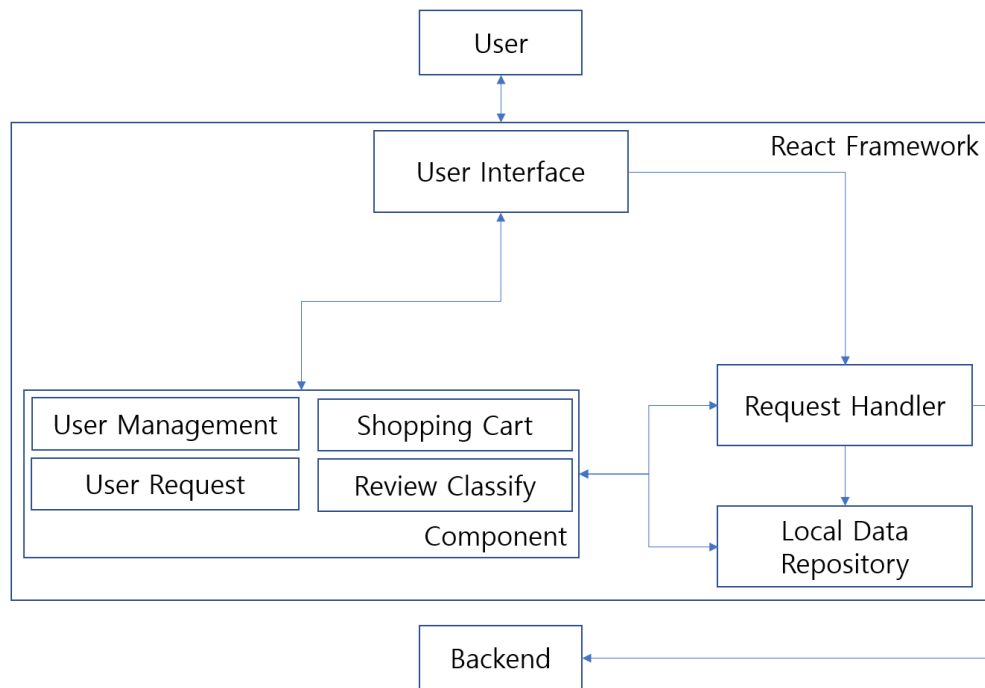
## 4. Frontend System

### 4.1 Objective

Frontend System에서는 본 시스템에서 Frontend에 구현이 되는 기능들을 설명한다. 세부적으로 5개의 클래스로 나뉘어진다. 5개의 클래스는 User Management, Shopping Cart, User Requirement, UI, Socket으로 구성되어 있다. Class diagram, sequence diagram,

state diagram을 사용해 frontend system의 구조를 표현한다.

## 4.2 Overall Architecture



**[Diagram 4] Overall Frontend System Architecture**

Frontend 시스템의 전반적인 구조를 다이어그램으로 나타낸 것이다. 먼저 사용자로부터 User Interface를 통해 사용자가 사용할 기능을 선택한다. 사용자가 사용할 수 있는 컴포넌트를 기능별로 나누었을 때, 크게 4가지로 나뉜다. User Management, Shopping Cart, Review Classify, User Request 이렇게 4가지로 나뉜다.

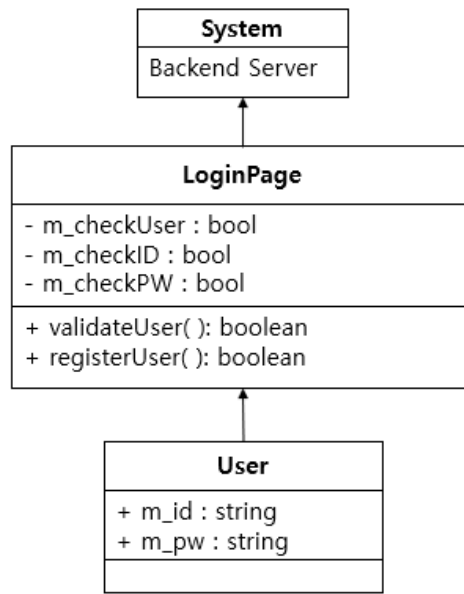
각각의 컴포넌트에 해당하는 기능을 사용하려고 사용자가 접근하면 그 기능이 어떠한 특정한 기능인지에 대한 요청을 처리하는 것부터 시작된다. 해당 요청이 Request Handler를 통해 전달되고 Request Handler는 그 요청이 어떠한 요청인지 식별한 후, Frontend에 저장되어 있는 Local Repository로 해결할 수 있는 요청이면 Backend의 도움 없이 자체적으로 해결을 한다. 하지만 Backend와의 통신이 필요한 요청 서비스라면 Backend에 해당 요청을 전달하여 그에 상응하는 데이터를 받는다.

로컬에서 해결한 요청이든 서버와 통신 후 얻어진 데이터이든 모두 다시 컴포넌트로 데이터가 전달되어 사용자에게 사용자가 요청한 것과 일치하는 결과를 보여준다.

## 4.3 Sub-Systems

### A. User Management System

#### A.1. Class Diagram



[Diagram 5] User Management System Class Diagram

#### 1) LoginPage - 로그인 페이지 객체

User가 id 와 password를 Login Page에 입력하면 그 정보를 Backend로 보내고 회원 정보가 일치하는지 확인한다.

##### a. Attributes

- m\_chekcUser : 유저의 회원 확인 필드
- m\_checkID : 유저 아이디 확인 필드
- m\_checkPW : 유저 비밀번호 확인 필드

##### b. Methods

- + validUser() : 유저의 회원가입 여부를 확인하는 메소드이다.
- + registerUser() : 새 유저의 회원가입을 처리하는 메소드이다.

## 2) User - 사용자 객체

사용자 정보를 담고 있는 객체이다. 사용자가 로그인을 하는데 필요한 아이디와 비밀번호를 가지고 있다.

### a. Attributes

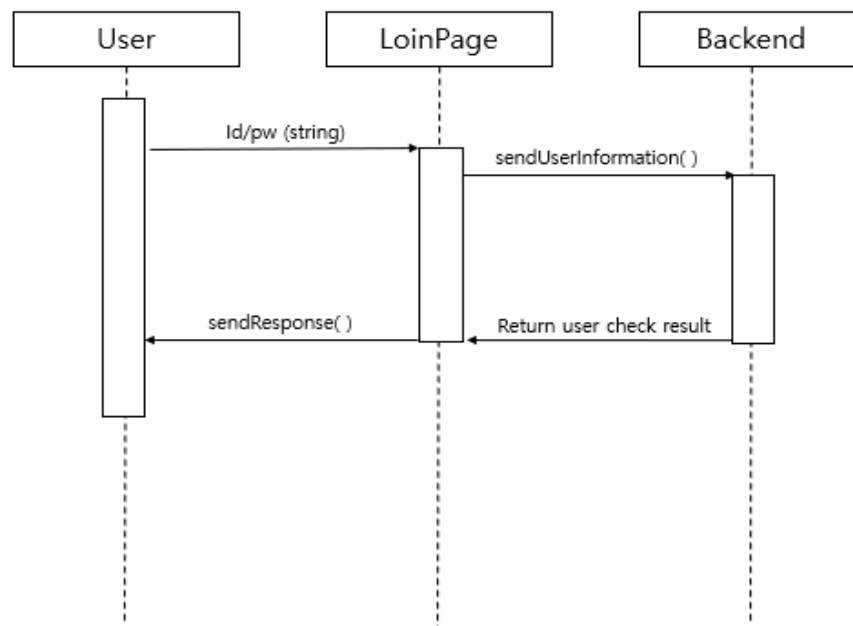
- m\_id : 유저 id 필드
- m\_pw : 유저 password 필드

### b. Methods

없음.

## A.2 Sequence Diagram

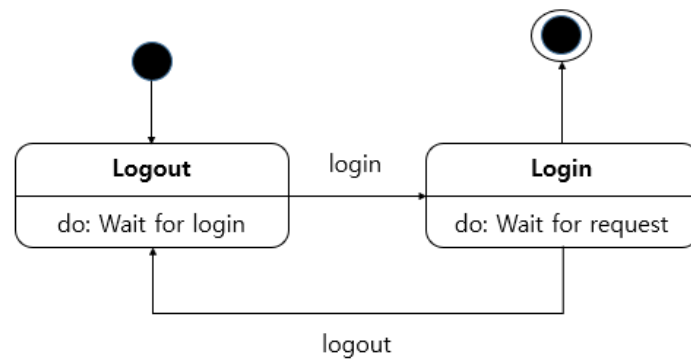
### 1) User Management Sequence Diagram



[Diagram 6] User Management Sequence Diagram

## A.3. State Diagram

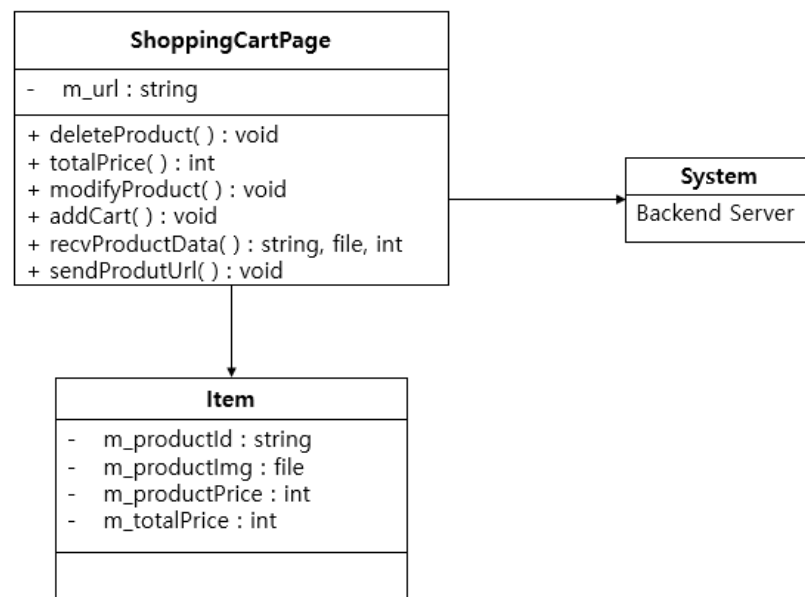
### 1) User Management State Diagram



[Diagram 7] User Management State Diagram

## B. Shopping Cart System

### B.1. Class Diagram



[Diagram 8] Shopping Cart System Class Diagram

#### 1) ShoppingCartPage - 장바구니 페이지 객체

쇼핑몰에서 장바구니에 담고 싶은 상품을 담을 때 사용하는 기능들에 대한 System이다. Sub system 으로는 장바구니에 상품 담기, 상품 삭제, 상품 정보 수정, 총 가격 계산이 있다.

### a. Attributes

- m\_url : 사용자가 보고 있는 쇼핑물의 특정 상품 url 정보를 가지고 있는 필드

### b. Methods

- + addCart() : url을 통해 전달받은 상품 정보를 장바구니에 추가하는 메소드이다.
- + deleteProduct() : 장바구니에 있는 상품을 삭제하는 메소드이다.
- + modifyProduct() : 장바구니에 있는 상품 정보를 수정하는 메소드이다.
- + totalPrice() : 장바구니에 있는 전체 상품의 가격을 구하는 메소드이다.
- + recvProductData() : Backend로부터 url을 통한 상품의 정보의 데이터를 받는 메소드이다.
- + sendProductUrl() : Backend로 상품정보가 담긴 url을 전달하는 메소드이다.

## 2) Item - 상품 객체

장바구니에 담은 상품의 정보를 담고 있는 객체이다.

### a. Attributes

- m\_productId : 상품의 이름 필드
- m\_productImg : 상품의 이미지 필드
- m\_productPrice : 상품의 가격 필드
- m\_totalPrice : 총 상품의 가격 필드

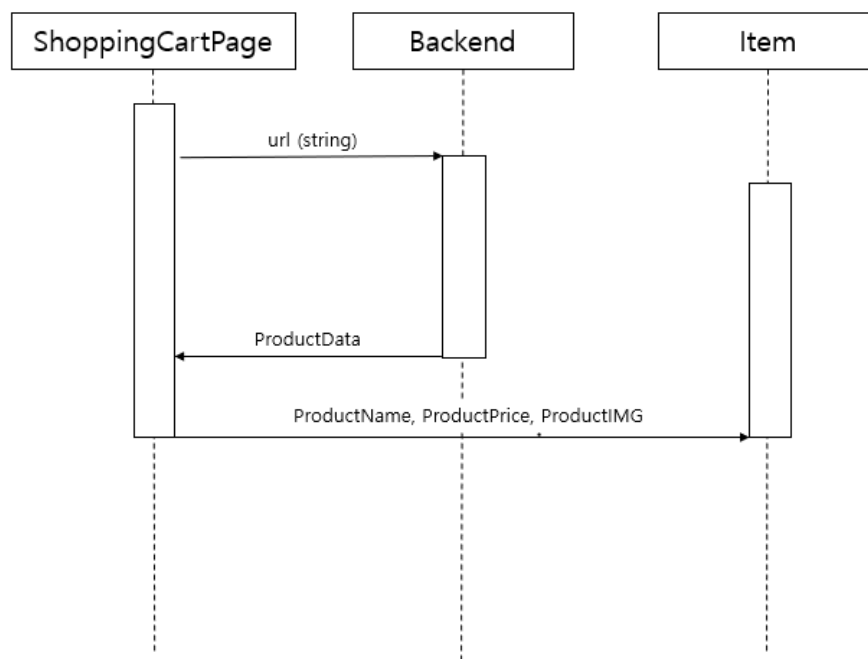
### b. Methods

없음.

## B.2 Sequence Diagram

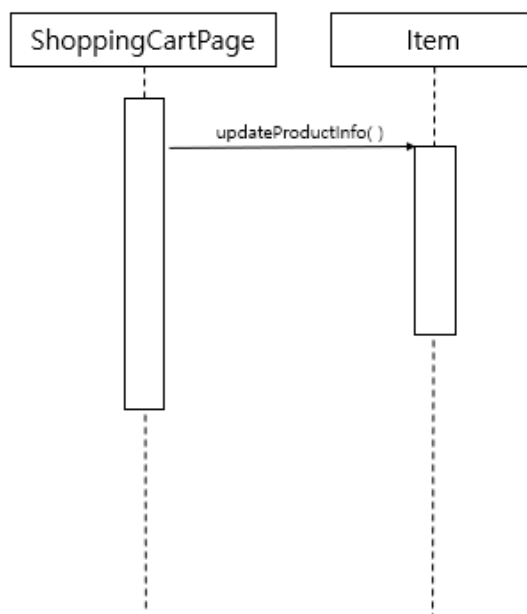
### 1) Add Cart Sequence Diagram





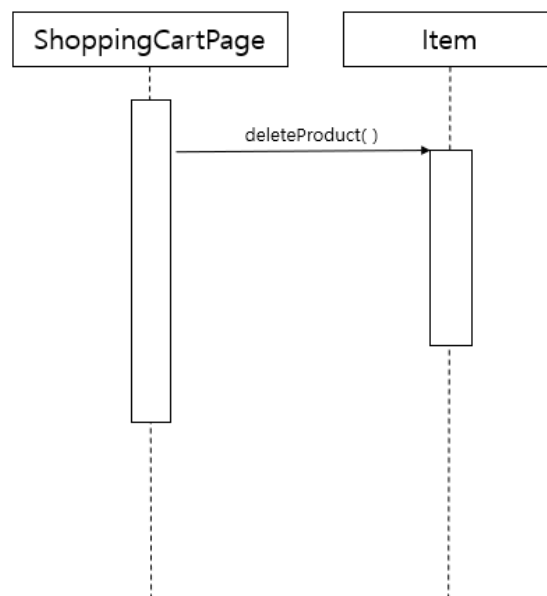
[Diagram 9] Add Cart Sequence Diagram

## 2) Modify Product Sequence Diagram



[Diagram 10] Modify Product Sequence Diagram

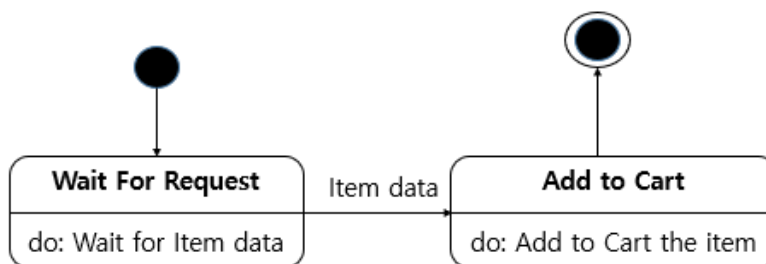
## 3) Delete Product Sequence Diagram



[Diagram 11] Delete Product Sequence Diagram

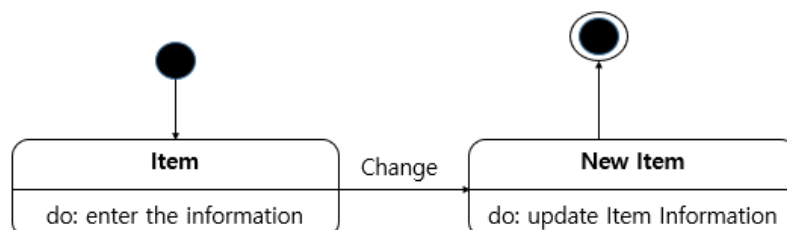
### B.3. State Diagram

#### 1) Add Cart State Diagram



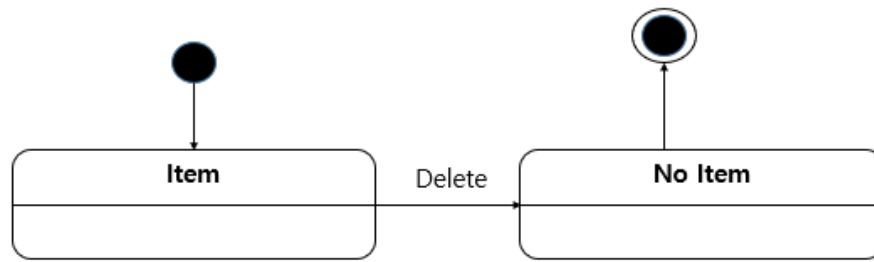
[Diagram 12] Add Cart State Diagram

#### 2) Modify Product State Diagram



[Diagram 13] Modify Product State Diagram

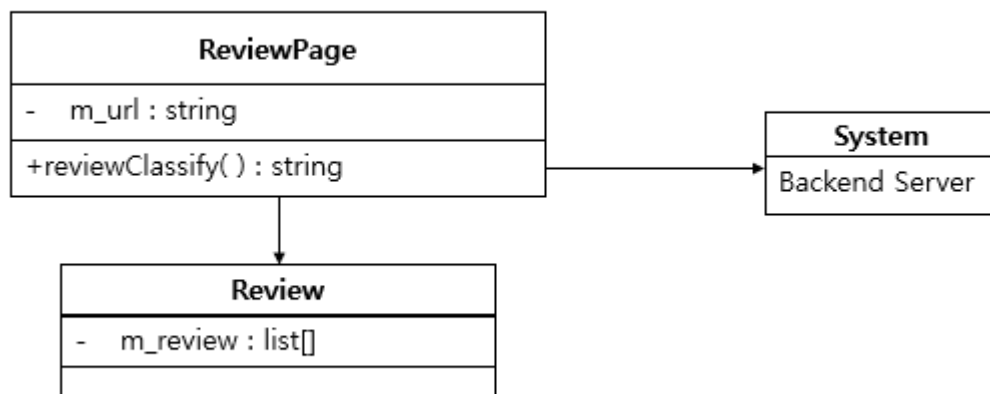
### 3) Delete Product State Diagram



[Diagram 14] Delete Product State Diagram

## C. Review Classify System

### C.1. Class Diagram



[Diagram 15] Review Classify System Class Diagram

#### 1) Review Page - 리뷰 페이지 객체

Backend에 url을 보내고 해당 리뷰 사이트의 리뷰를 분류해서 사용자가 원하는 카테고리  
고리의 리뷰를 보여준다.

##### a. Attributes

- m\_url : 리뷰 사이트 url 필드

##### b. Methods

+ ReviewClassify() : 리뷰를 분류하는 메소드이다.

## 2) Review - 리뷰 객체

장바구니에 담은 상품의 리뷰 정보를 담고 있는 객체이다.

### a. Attributes

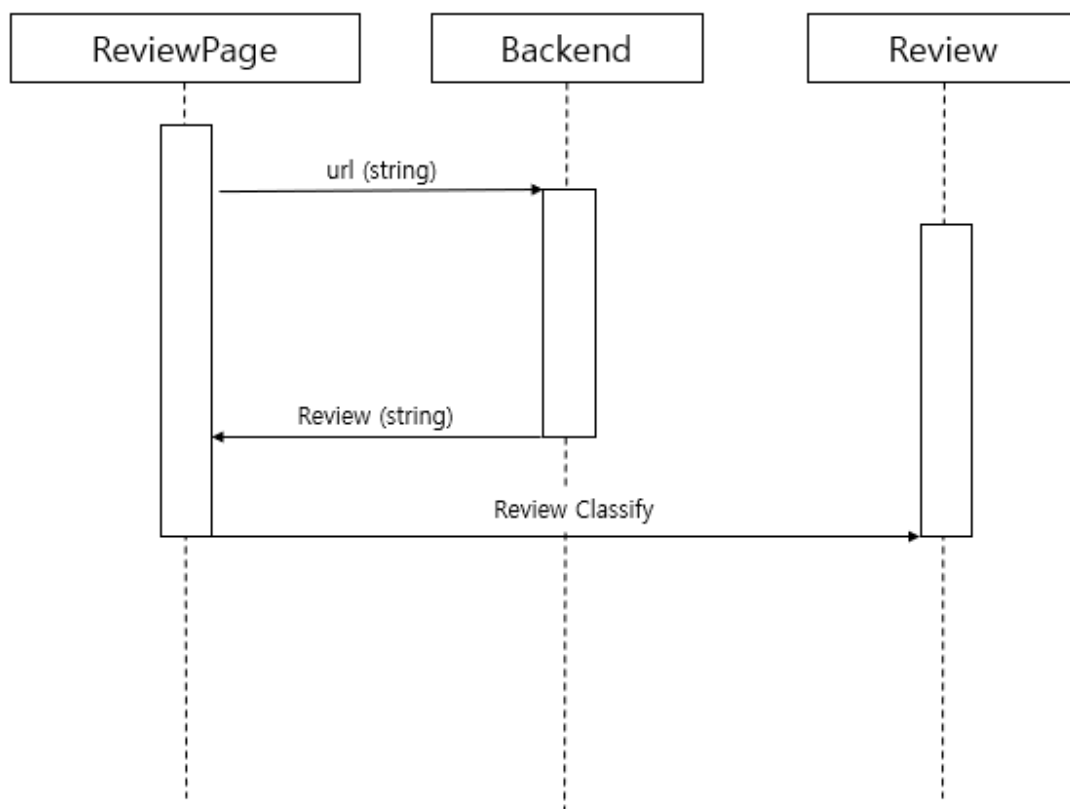
- m\_review : 사용자에게 보여주는 리뷰 필드

### b. Methods

없음.

## C.2 Sequence Diagram

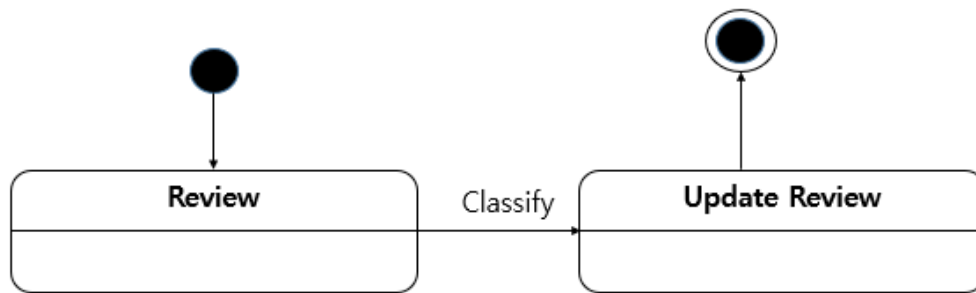
### 1) Review Classify Sequence Diagram



[Diagram 16] Review Classify Sequence Diagram

## C.3. State Diagram

### 1) Review Classify State Diagram

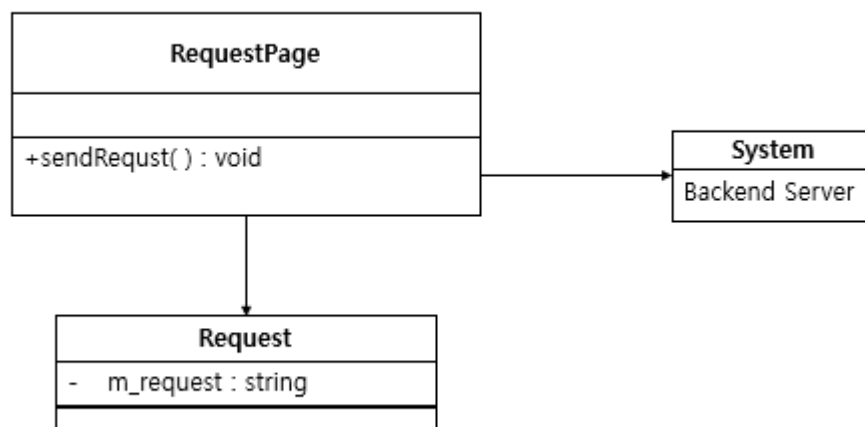


[Diagram 17] Review Classify State Diagram

## D. User Request System

### D.1. Class Diagram

#### 1) Review Page - 리뷰 페이지 객체



[Diagram 18] User Request System Class Diagram

Backend에 url을 보내고 해당 리뷰 사이트의 리뷰를 분류해서 사용자가 원하는 카테고리  
의 리뷰를 보여준다.

#### 1) RequestPage - 요청 페이지 객체

Backend에 사용자의 요청을 보낸다.

##### a. Attributes

없음.

### b. Methods

+ sendRequest() : 요청을 backend로 전달하는 메소드이다.

## 2) Request- 요청 객체

사용자의 요청을 담는 객체이다.

### a. Attributes

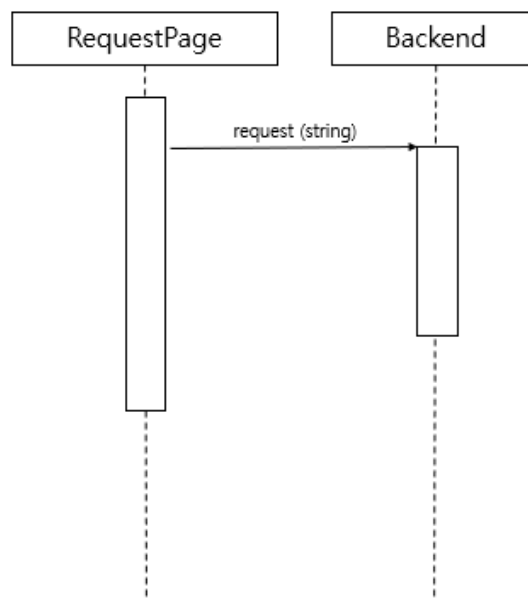
- m\_request : 리뷰 사이트 url 필드

### b. Methods

없음.

## D.2 Sequence Diagram

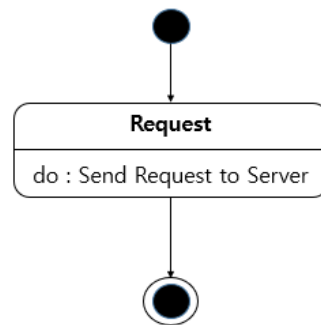
### 1) User Request Sequence Diagram



[Diagram 19] User Request Sequence Diagram

## D.3. State Diagram

### 1) User Request State Diagram



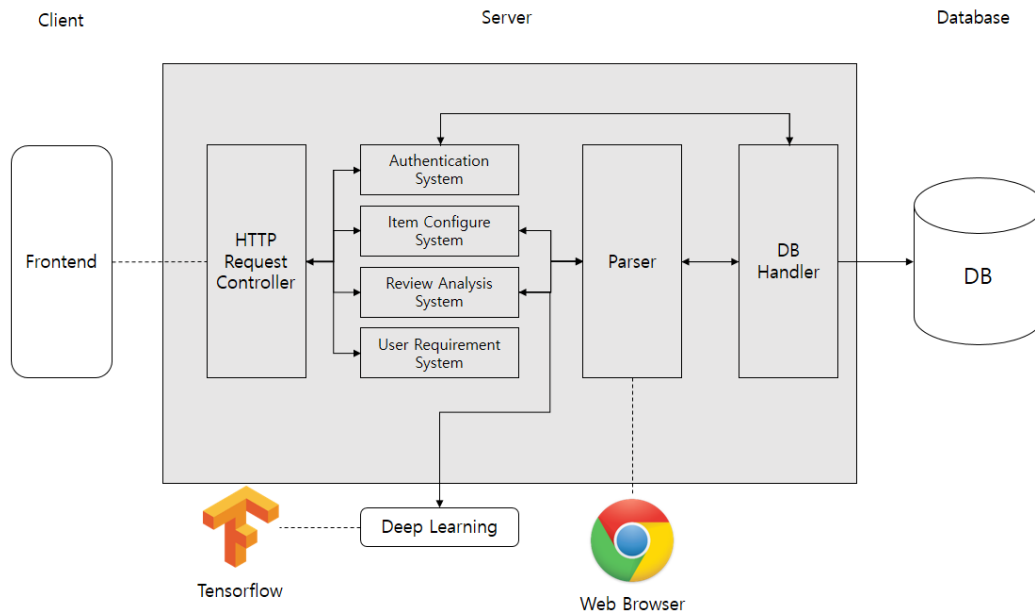
[Diagram 20] User Request State Diagram

## 5. Backend System

### 5.1 Objective

Backend System에서는 본 시스템에서 Backend에 구현이 되는 기능들을 설명한다. 세부적으로 6개의 sub-system로 구성된다. 6개의 sub-system은 Authentication System, Parser System, Review Analysis System, Request Handler System, Item Configure System, Respond Sending System으로 구성되어 있다. Class diagram, sequence diagram, state diagram을 사용해 backend system의 구조를 표현한다.

### 5.2 Overall Architecture



**[Diagram 21] Overall Backend System Architecture**

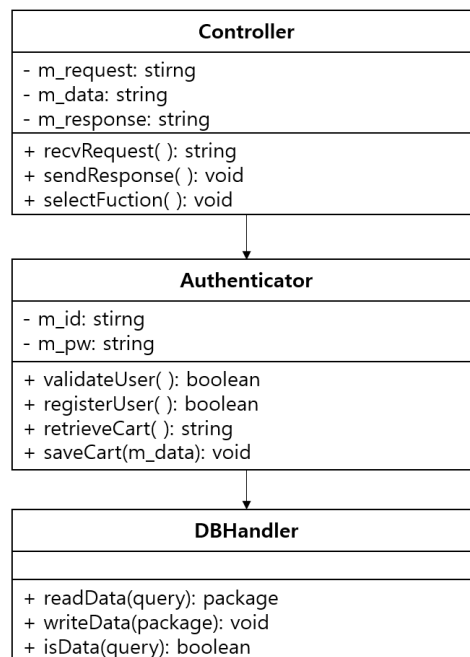
Backend 시스템의 전반적인 구조를 다이어그램으로 나타낸 것이다. 먼저, Frontend를 통해 유저로부터 받은 시스템 요청을 HTTP Request Controller를 통해 요구 시스템을 거쳐 분류한다. 분류된 요구사항은 각각 Authentication System, Item Configure System, Review Analysis System, User Requirement System을 통해 수행된다. Authentication System에서는 유저의 회원가입, 로그인을 담당하고 로그인 시 유저의 데이터를 불러온다. Item Configure System에서는 Parser를 통해 유저가 관심있는 상품 정보를 가져온다. Review Analysis System에서는 Parser를 통해 리뷰 정보를 불러오고, 딥러닝 기법으로 리뷰들을 정해진 분류 기준으로 분류한다. User Requirement System에서는 유저가 서비스를 원하는 사이트, 또는 개발자에게 요구하는 건의사항을 입력 받아 개발자가 그 정보를 확인할 수 있도록 한다. 위의 4가지 시스템은 DB와 통신하면서 역할을 수행하게 되고 처리된 시스템 요구는 다시 HTTP Request Controller를 통해 최종적으로 Frontend로 전달되게 된다.

## 5.3 Sub-Systems

### A. Authentication System

#### A.1. Class Diagram





[Diagram 22] Authentication System Class Diagram

### 1) Controller - 요청 분류 객체

Frontend에서 받은 시스템 요청을 분류해 해당 System으로 각각 보내주고 처리된 시스템 응답을 다시 Frontend로 보내준다. Authentication System에서는 로그인, 회원가입, 장바구니 저장과 관련된 요청을 분류한다.

#### a. Attributes

- m\_request : 시스템 요청 종류 필드
- m\_data : 시스템 요청 내용 필드
- m\_response : 요청 응답 내용 필드

#### b. Methods

- + recvRequest( ) : Frontend로부터 시스템 요청을 받아오는 메소드이다.
- + sendResponse( ) : Frontend로 처리된 시스템 응답을 보내는 메소드이다.
- + selectFunction( ) : 받아온 시스템 요청을 m\_request 값에 따라 지정된 system을 m\_data값 넘겨주어 실행하도록 하는 메소드이다.

## 2) Authentication - 유저 인증 처리 객체

유저의 회원가입, 로그인을 관리하고 로그인 시 유저의 데이터를 불러오거나 저장한다.

### a. Attributes

- m\_id : 유저 id 필드
- m\_pw : 유저 password 필드

### b. Methods

- + validateUser( ) : 유저의 회원가입 여부를 확인하는 메소드이다.
- + registerUser( ) : 새 유저의 회원가입을 처리하는 메소드이다.
- + retrieveCart( ) : 유저의 장바구니 사용정보를 DB에서 받아오는 메소드이다.
- + saveCart(m\_data): 유저의 장바구니 정보를 DB에 저장하는 메소드이다.

## 3) DBHandler - DB 요청 처리 객체

DB에서 해당 데이터를 가져오거나 저장한다. Authentication System에서는 유저의 등록 정보 및 장바구니 정보를 저장 및 가져오는 역할을 수행한다

### a. Attributes

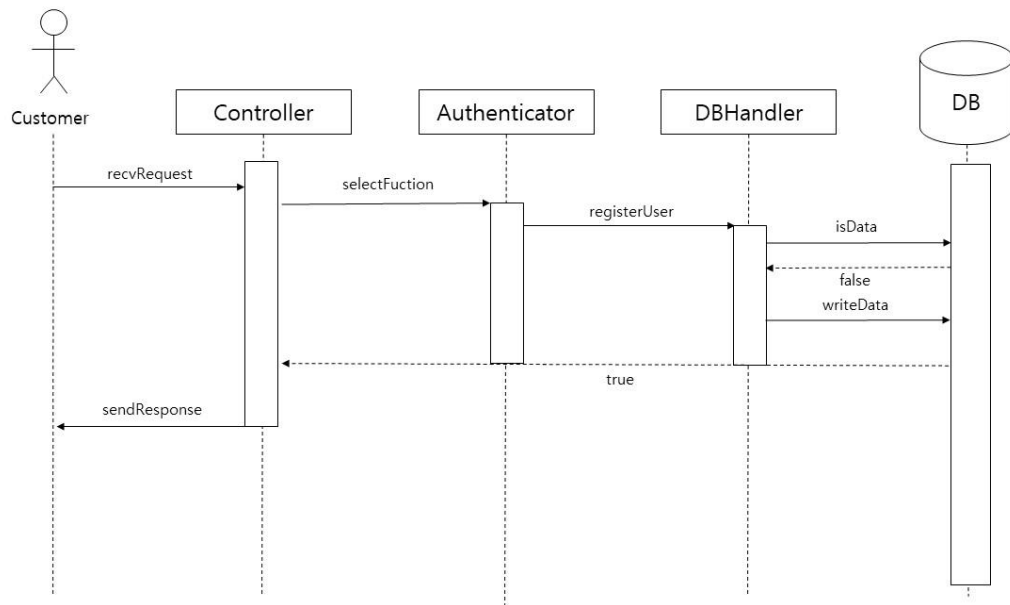
없음.

### b. Methods

- + readData(query) : query를 작성하여 DB의 데이터를 읽어오는 메소드이다.
- + writeData(package): package를 작성하여 DB에 데이터를 쓰는 메소드이다.
- + isData(query): query를 작성하여 해당 데이터가 DB에 존재하는지 확인하는 메소드이다.

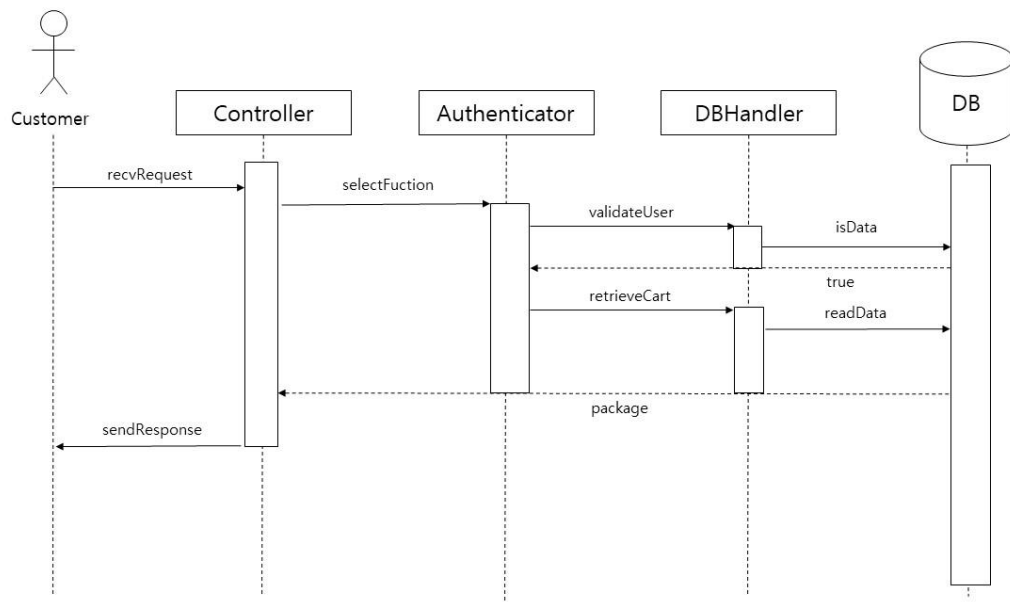
## A.2 Sequence Diagram

### 1) Register Sequence Diagram



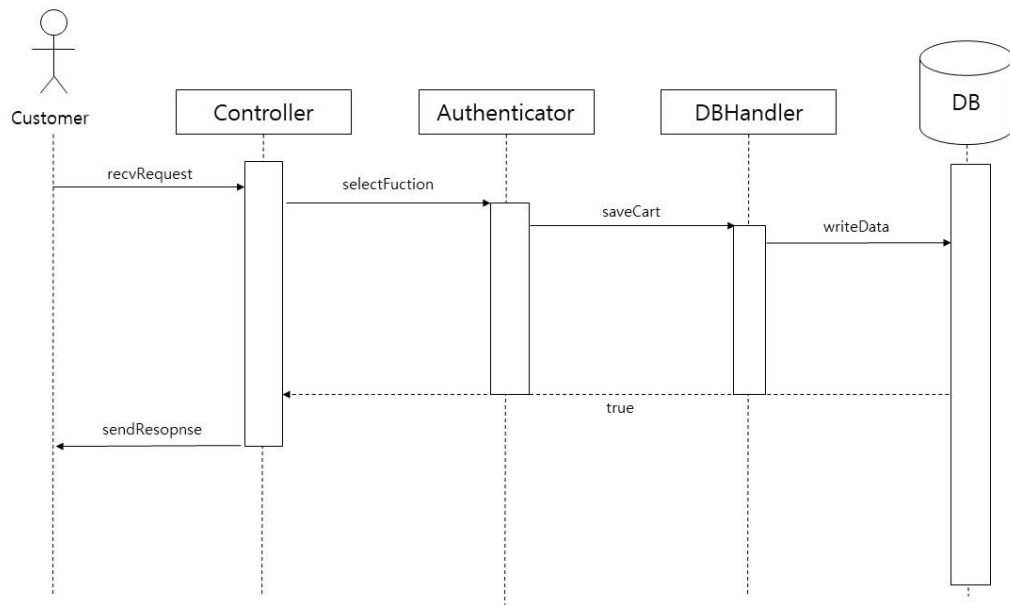
[Diagram 23] Register Sequence Diagram

## 2) Login Sequence Diagram



[Diagram 24] Login Sequence Diagram

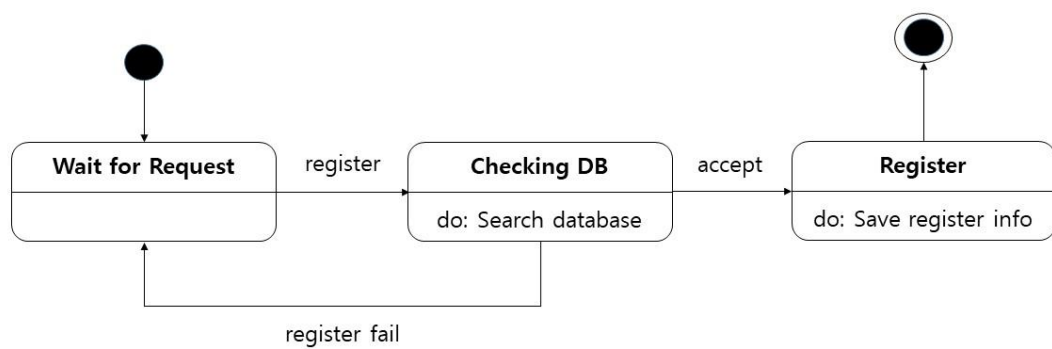
## 3) Register Sequence Diagram



[Diagram 25] Save Cart Sequence Diagram

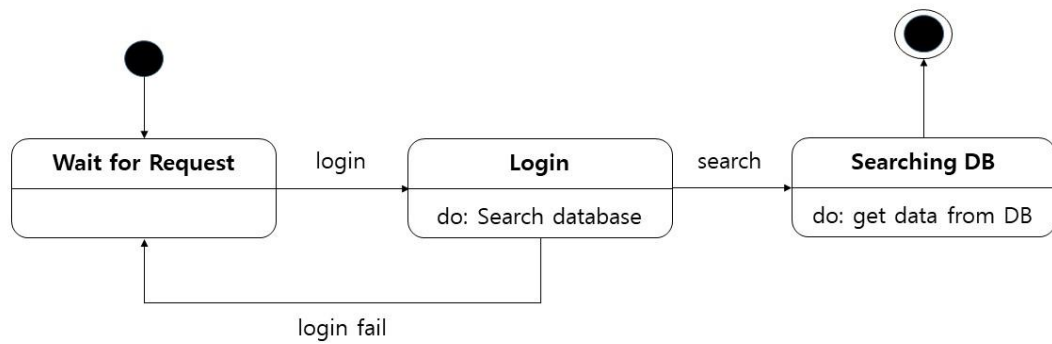
### A.3. State Diagram

#### 1) Register State Diagram



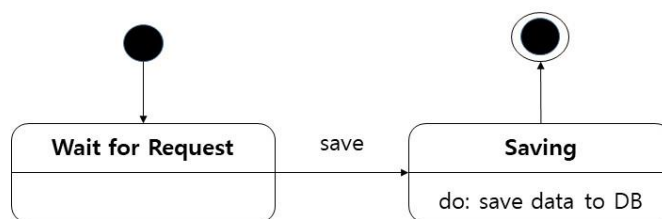
[Diagram 26] Register State Diagram

#### 2) Login State Diagram



[Diagram 27] Login State Diagram

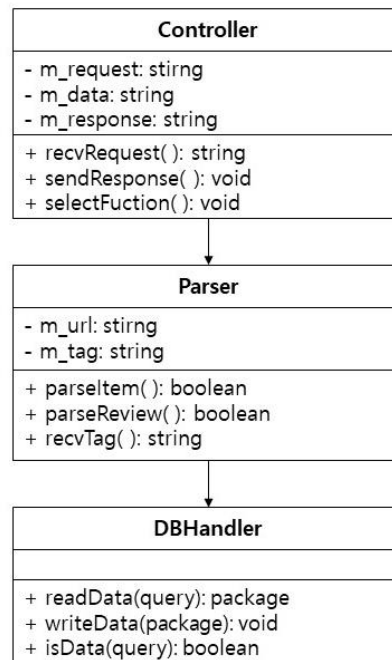
### 3) Save Cart State Diagram



[Diagram 28] Save Cart State Diagram

## B. Item Configure System

### B.1. Class Diagram



[Diagram 29] Item Configure System Class Diagram

### 1) Controller - 요청 분류 객체

Frontend에서 받은 시스템 요청을 분류해 해당 System으로 각각 보내주고 처리된 시스템 응답을 다시 Frontend로 보내준다. Item Configure System에서는 상품 담기와 관련된 요청을 분류한다.

#### a. Attributes

- m\_request : 시스템 요청 종류 필드
- m\_data : 시스템 요청 내용 필드
- m\_response : 요청 응답 내용 필드

#### b. Methods

- + recvRequest( ) : Frontend로부터 시스템 요청을 받아오는 메소드이다.
- + sendResponse( ) : Frontend로 처리된 시스템 응답을 보내는 메소드이다.
- + selectFunction( ) : 받아온 시스템 요청을 m\_request 값에 따라 지정된 system

을 m\_data값 넘겨주어 실행하도록 하는 메소드이다.

## 2) Parser - 웹 페이지 파싱 객체

웹 주소를 가져와 아이템 상품 또는 리뷰를 파싱해 오는 객체이다. Item Configure System에서는 상품을 파싱해 온다.

### a. Attributes

- m\_url : 웹 사이트 주소 필드
- m\_tag : 파싱 태그 필드

### b. Methods

- + parseltem( ) : 사이트로부터 아이템 정보(이름, 가격, 사진, 사이트 주소, 사이트 이름)를 가져오는 메소드이다.
- + parseReview( ) : 사이트로부터 아이템 리뷰를 가져오는 메소드이다.
- + recvTag( ) : 아이템 정보를 파싱할 때 필요한 태그 정보를 DB로부터 가져오는 메소드이다.

## 3) DBHandler - DB 요청 처리 객체

DB에서 해당 데이터를 가져오거나 저장한다. Item Configure System에서는 상품과 관련된 태그를 가져오는 역할을 한다.

### a. Attributes

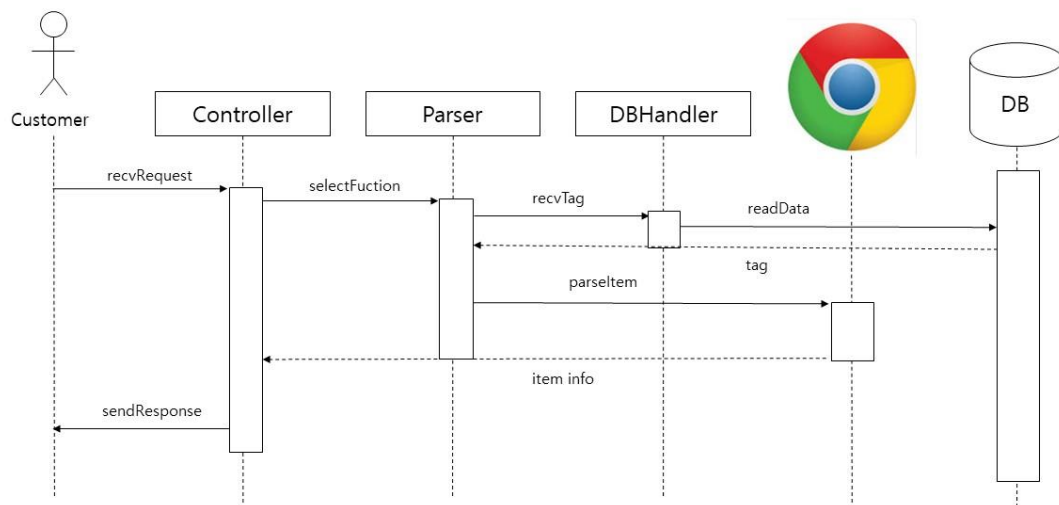
없음.

### b. Methods

- + readData(query) : query를 작성하여 DB의 데이터를 읽어오는 메소드이다.
- + writeData(package): package를 작성하여 DB에 데이터를 쓰는 메소드이다.
- + isData(query): query를 작성하여 해당 데이터가 DB에 존재하는지 확인하는 메소드이다.

## B.2 Sequence Diagram

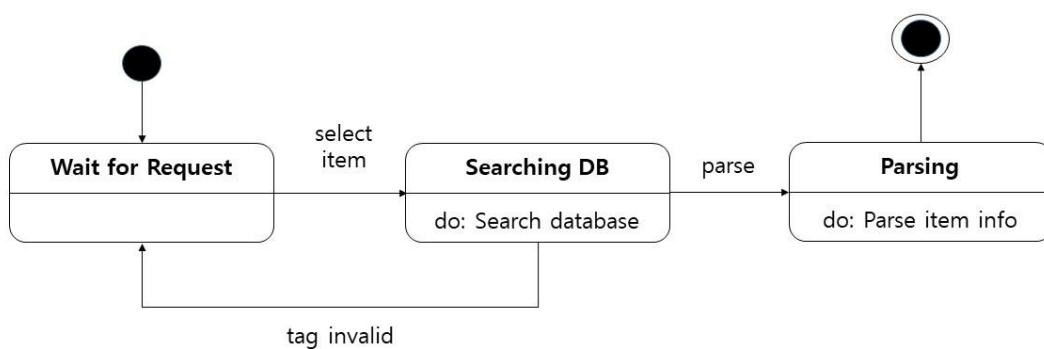
## 1) Item Configure Sequence Diagram



[Diagram 30] Item Configure Sequence Diagram

## B.3 State Diagram

### 1) Item Configure State Diagram

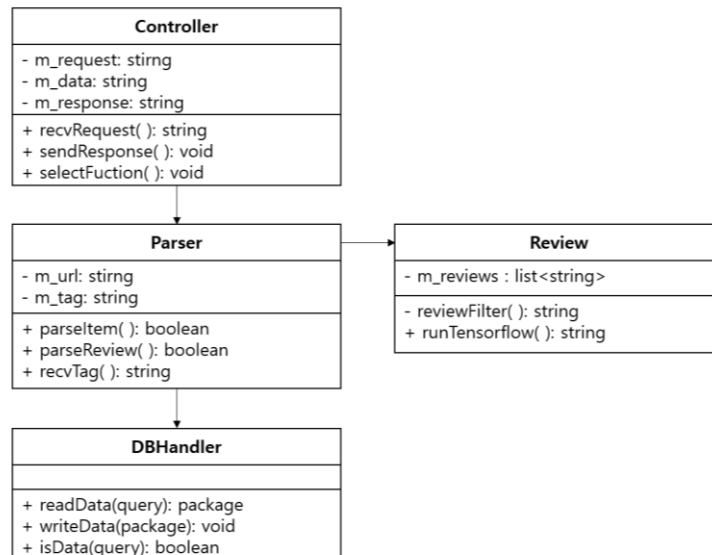


[Diagram 31] Item Configure State Diagram



## C. Review Analysis System

### C.1. Class Diagram



[Diagram 32] Review Analysis System Class Diagram

#### 1) Controller - 요청 분류 객체

Frontend에서 받은 시스템 요청을 분류해 해당 System으로 각각 보내주고 처리된 시스템 응답을 다시 Frontend로 보내준다. Review Analysis System에서는 리뷰 분석과 관련된 요청을 분류한다.

##### a. Attributes

- m\_request : 시스템 요청 종류 필드
- m\_data : 시스템 요청 내용 필드
- m\_response : 요청 응답 내용 필드

##### b. Methods

- + recvRequest( ) : Frontend로부터 시스템 요청을 받아오는 메소드이다.
- + sendResponse( ) : Frontend로 처리된 시스템 응답을 보내는 메소드이다.
- + selectFunction( ) : 받아온 시스템 요청을 m\_request 값에 따라 지정된 system

을 m\_data값 넘겨주어 실행하도록 하는 메소드이다.

## 2) Parser - 웹 페이지 파싱 객체

웹 주소를 가져와 아이템 정보와 리뷰를 파싱해 오는 객체이다. Review Analysis System에서는 리뷰를 파싱해 온다.

### a. Attributes

- m\_url : 웹 사이트 주소 필드
- m\_tag : 파싱 태그 필드

### b. Methods

- + parseItem( ) : 사이트로부터 아이템 정보(이름, 가격, 사진, 사이트 주소, 사이트 이름)를 가져오는 메소드이다.
- + parseReview( ) : 사이트로부터 아이템 리뷰를 가져오는 메소드이다.
- + recvTag( ) : 아이템 정보를 파싱할 때 필요한 태그 정보를 DB로부터 가져오는 메소드이다.

## 3) DBHandler - DB 요청 처리 객체

DB에서 해당 데이터를 가져오거나 저장한다. Review Analysis System에서는 리뷰 태그를 가져오는 역할을 한다.

### a. Attributes

없음.

### b. Methods

- + readData(query) : query를 작성하여 DB의 데이터를 읽어오는 메소드이다.
- + writeData(package): package를 작성하여 DB에 데이터를 쓰는 메소드이다.
- + isData(query): query를 작성하여 해당 데이터가 DB에 존재하는지 확인하는 메소드이다.

## 3) Review - 리뷰 처리 객체

파싱을 해온 리뷰의 전처리를 하여 tensorflow에 input으로 넘겨 준다. 그리고 tensorflow로부터 분류가 된 리뷰를 받는다.

#### a. Attributes

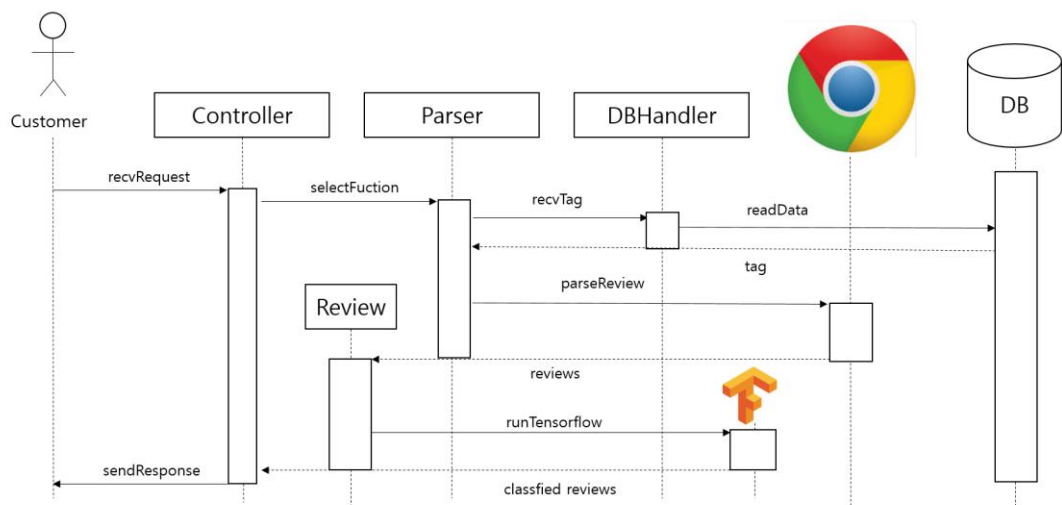
- m\_reviews : 가공된 리뷰 배열 필드

#### b. Methods

- reviewFilter( ) : 리뷰를 텐서플로우의 데이터 형식에 맞게 가공해주는 메소드다.
- + runTensorflow( ) : 가공된 리뷰를 텐서플로우를 거쳐 분류된 리뷰 결과를 받는 메소드이다.

### C.2. Sequence Diagram

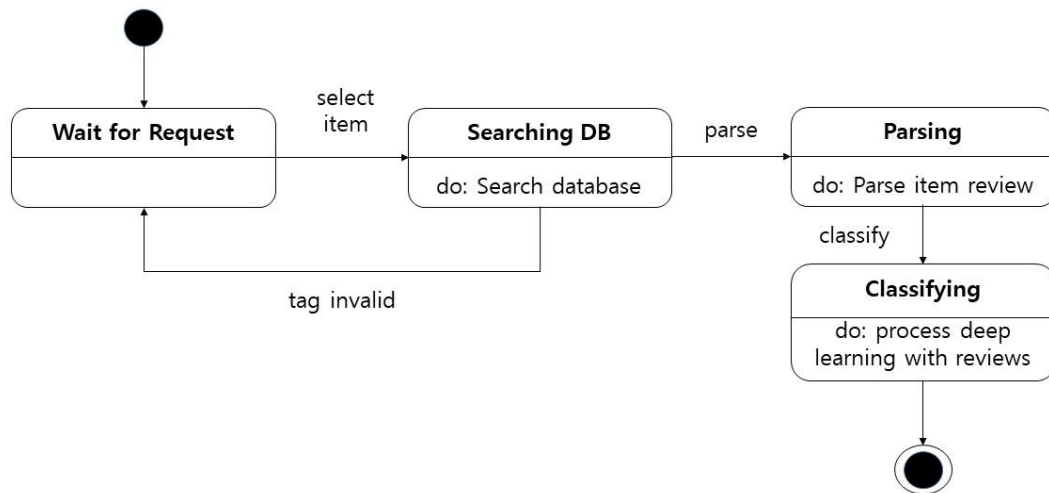
#### 1) Review Analysis Sequence Diagram



[Diagram 33] Review Analysis Sequence Diagram

### C.3. State Diagram

#### 1) Review Analysis State Diagram

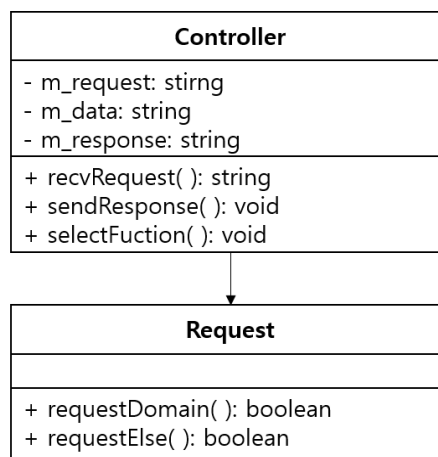


[Diagram 34] Review Analysis State Diagram

## D. User Requirement System

### D.1. Class Diagram

#### 1) Controller - 요청 분류 객체



[Diagram 35] User Requirement System

Frontend에서 받은 시스템 요청을 분류해 해당 System으로 각각 보내주고 처리된 시스템 응답을 다시 Frontend로 보내준다. User Requirement System에서는 사용자의 도

메인 요청 및 그 외 요청을 분류한다.

**a. Attributes**

- m\_request : 시스템 요청 종류 필드
- m\_data : 시스템 요청 내용 필드
- m\_response : 요청 응답 내용 필드

**b. Methods**

- + recvRequest( ) : Frontend로부터 시스템 요청을 받아오는 메소드이다.
- + sendResponse( ) : Frontend로 처리된 시스템 응답을 보내는 메소드이다.
- + selectFunction( ) : 받아온 시스템 요청을 m\_request 값에 따라 지정된 system 을 m\_data값 넘겨주어 실행하도록 하는 메소드이다.

**2) Request - 사용자의 요청 객체**

사용자가 보낸 요청을 개발자에게 보내주는 메소드이다. 요청은 text 파일에 담기게 된다.

**a. Attributes**

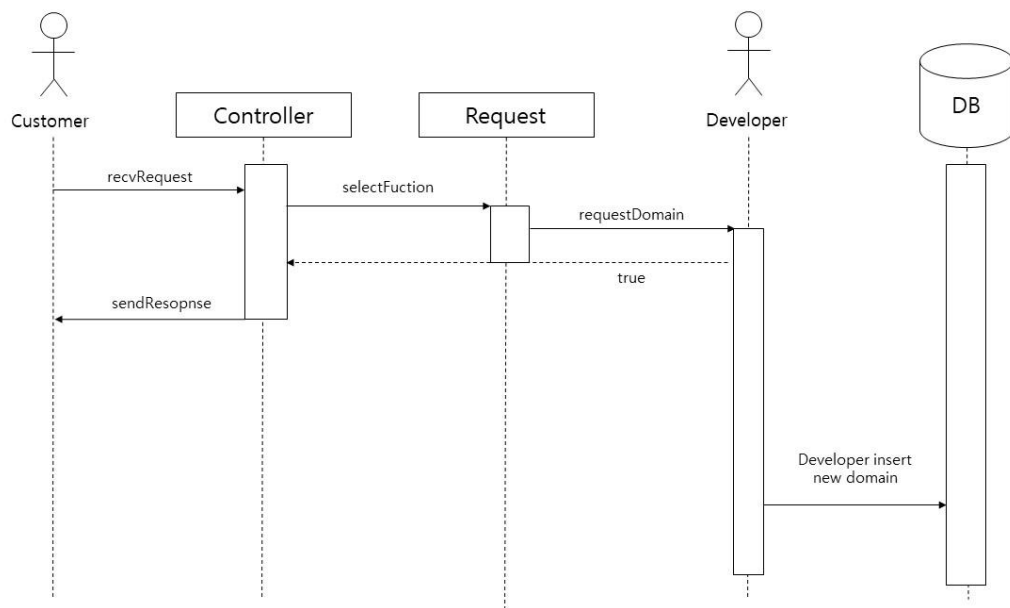
없음.

**b. Methods**

- + requestDomain( ) : 유저가 추가하고 싶은 도메인을 입력 받아 개발자에게 보내는 메소드이다.
- + requestElse( ) : 유저의 요구사항 글을 입력 받아 개발자에게 보내는 메소드이다.

**D.2. Sequence Diagram**

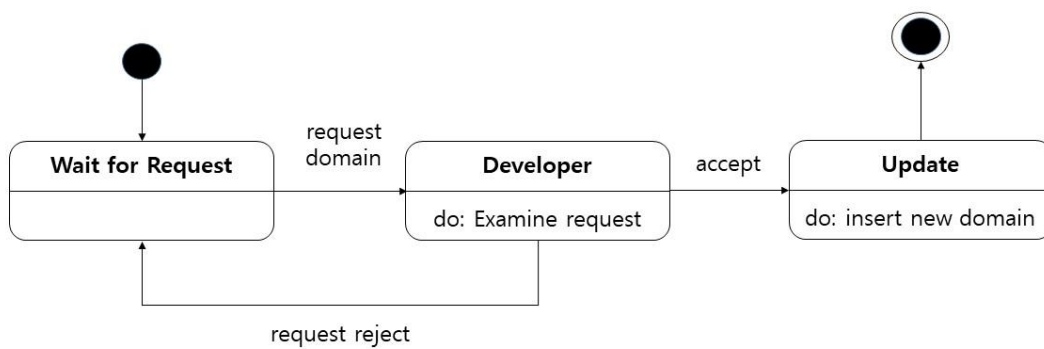
**1) User Requirement Sequence Diagram**



[Diagram 36] User Requirement Sequence Diagram

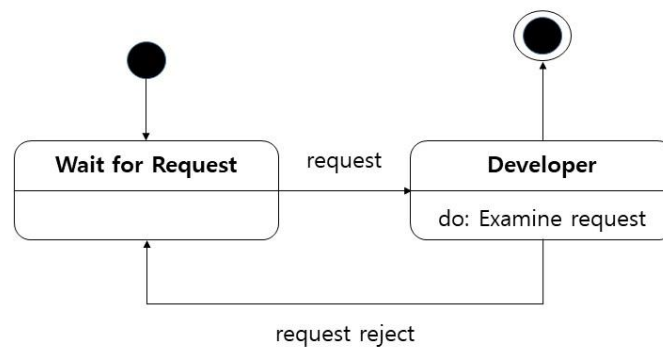
### D.3. State Diagram

#### 1) Domain Requirement State Diagram



[Diagram 37] Domain Requirement State Diagram

#### 2) Else Requirement State Diagram



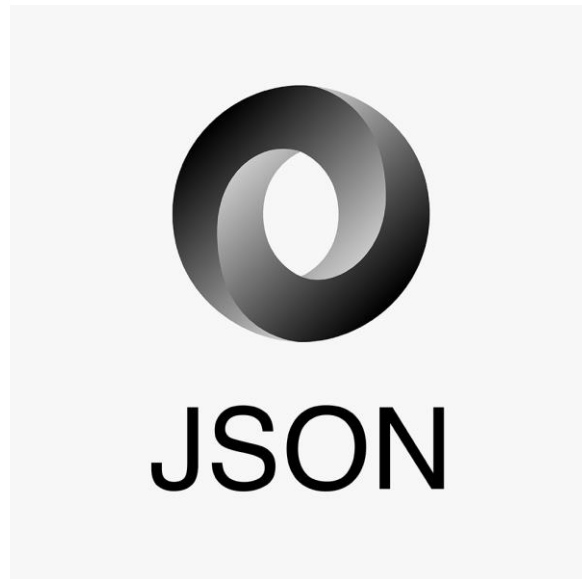
[Diagram 38] Else Requirement State Diagram

## 6. Protocol Design

### 6.1 Objective

Protocol Design에서는 Frontend와 Backend 시스템간의 통신 프로토콜에 대하여 기술한다. 통신 과정에서 사용되는 기술 및 메시지의 형식과 용도, 의미 등을 상세하게 기술한다.

### 6.2 JSON



[Figure 10] JSON Logo

JSON은 JavaScript Object Notation의 약자로서 (속성, 값)의 쌍 또는 (키, 값)의 쌍으로 이루어진 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다. 인터넷에서 자료를 주고받을 때 그 자료를 표현하는 방법 중 하나이다. 본래는 자바스크립트 언어로부터 파생되어 자바스크립트의 구문 형식을 따르지만 언어 독립 형 데이터 포맷으로 발전되었다. 따라서, 프로그래밍언어나 플랫폼에 영향 받지 않고 다양한 프로그래밍 언어에서 사용할 수 있다. 본 시스템에서는 자료 통신을 통해 JSON을 사용한다.

## 6.3 Protocol Description

Sub-System들이 통신하는 과정에서 사용되는 프로토콜에 대해 설명한다. 각 프로토콜은 요청(request), 응답(response), 속성(attribute), 그리고 값(value)로 표현한다.

### A. Register Protocol

#### 1) Request

attribute	value
request	시스템 요청 id
id	유저의 id
pw	유저의 password



## 2) Response

attribute	value
register_success	가입 성공/실패

**B. Login Protocol**

## 1) Request

attribute	value
request	시스템 요청 id
id	유저의 id
pw	유저의 password

## 2) Response

attribute	value
login_success	로그인 성공/실패
cart_info	저장된 유저의 카트 정보

**C. Save Cart Protocol**

## 1) Request

attribute	value
request	시스템 요청 id
id	유저의 id
pw	유저의 password

## 2) Response

attribute	value
save_success	카트 정보 저장 성공/실패

**D. Item Configure Protocol**

## 1) Request

attribute	value
request	시스템 요청 id
url	상품의 사이트 주소

## 2) Response

attribute	value
item_name	상품 이름
item_price	상품 가격
item_image	상품 사진
site_url	사이트 주소
site_name	사이트 이름

**E. Review Analysis Protocol**

## 1) Request

attribute	value
request	시스템 요청 id
url	상품의 사이트 주소

## 2) Response

attribute	value
classified_reviews	분류된 상품 리뷰들

**F. User Domain Request Protocol**

## 1) Request

attribute	value
request	시스템 요청 id
domain	유저가 요청한 도메인 이름

## 2) Response

attribute	value
domain_success	도메인 요청 성공/실패

**G. User Else Request Protocol**

## 1) Request

attribute	value
request	시스템 요청 id
domain	유저가 요청한 도메인 이름

## 2) Response

attribute	value
else_success	요구사항 요청 성공/실패

## 7. Database Design

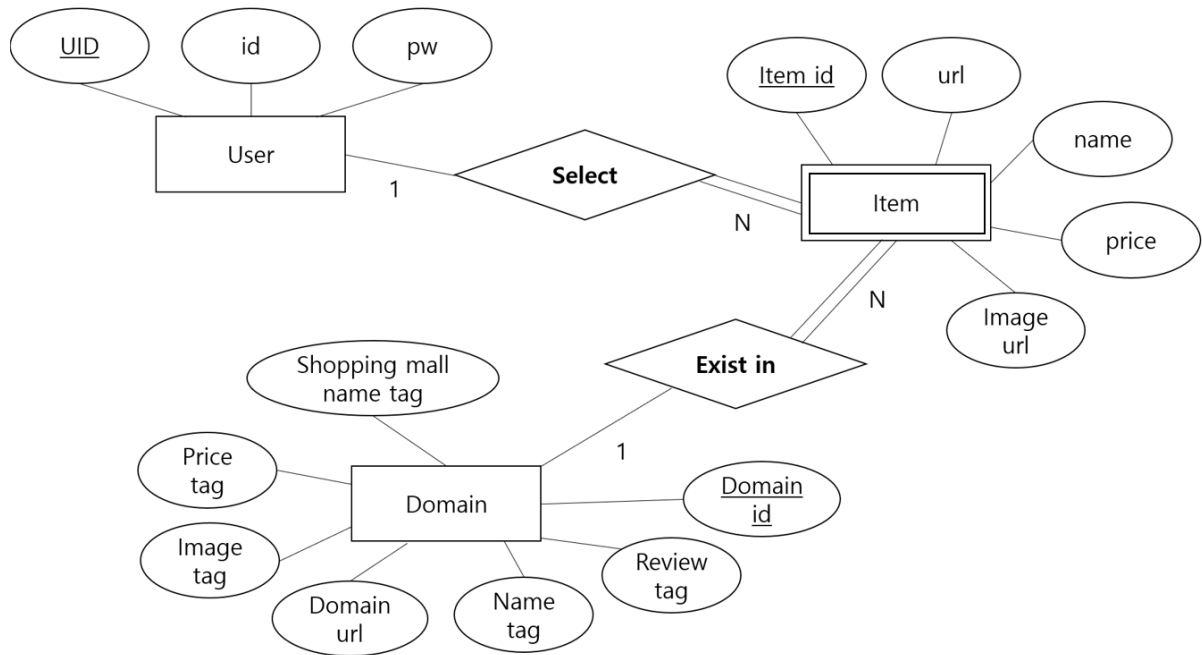
### 7.1 Objective

Database Design에서는 요구사항 명세서에서 기술한 데이터베이스 요구사항을 바탕으로 더 상세하게 각 속성간의 관계 등을 기술한다. ER diagram, Relation Schema, SQL DDL을 사용하여 설명한다.

### 7.2 ER Diagram

본 시스템에는 Domain, User, Item으로 총 3개의 entity가 존재한다. 이외의 Entity는 Database에서 모델링하지 않고, Local repository를 활용하므로 이에 대한 추가는 생략한다. 각각의 Entity는 Rectangle box로 표현되고, Entity 간의 관계은 마름모꼴로 표현된다. 이때 Entity가 가지는 attribute는 타원으로 표현하고, Key attribute는 attribute 아래에 밑줄을 그어 표현한다. 또한, multiple value를 가질 수 있는 attribute는 double 타원으로 표현한다. Cardinality는 One-to-one, One-to-many, Many-to-many로 구별되는데,

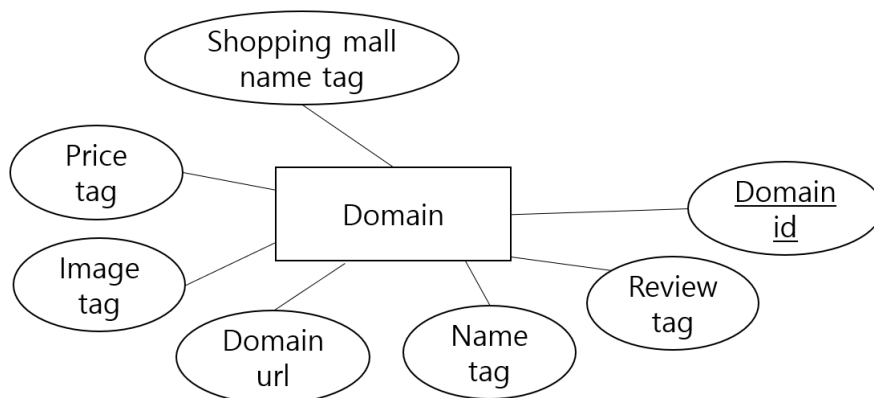
이에 대한 표현은 Relationship과 Entity 사이의 edge위에 숫자를 표시함으로써 표현한다. Participation Constraint는 Total participation, partial participation이 존재하는 데, 이에 대한 표현 방식으로는 Total participation인 경우 double line으로, partial participation인 경우 single line으로 이를 표현한다.



[Diagram 39] Overall ER Diagram

## A. Entities

### A.1 Domain

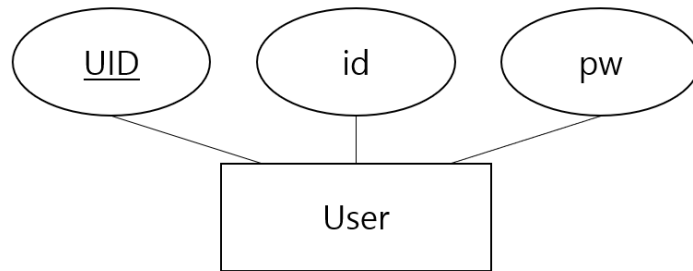


[Diagram 40] Domain ER Diagram

Domain Entity는 Domain으로부터 정보를 얻어내기 위해 parsing의 대상이 되는 tag를 저장한다. 이에 대한 것으로는 상품 가격의 tag인 price tag, 상품 이름의 tag인

name tag, shopping mall의 이름의 tag인 shopping mall name tag, review의 tag인 review tag, Domain의 URL, 그리고 Developer가 정의하는 Domain id를 key attribute로 가진다.

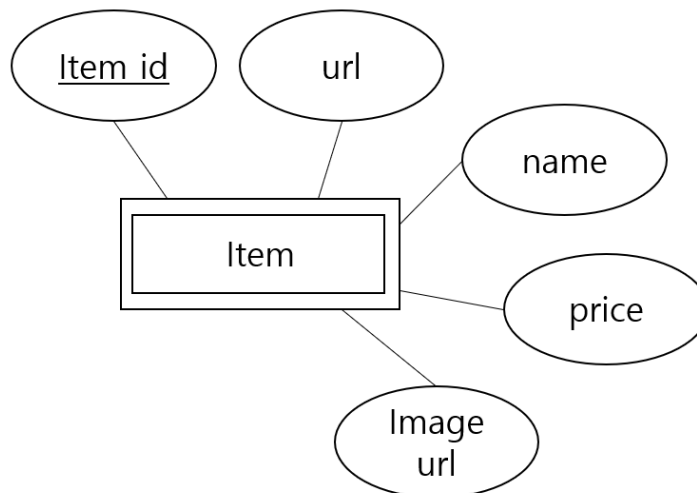
## A.2 User



[Diagram 41] User ER Diagram

User Entity는 User의 정보를 저장한다. UID가 key attribute이고, id, password에 대한 정보를 attribute로 가진다.

## A.3 Item



[Diagram 42] Item ER Diagram

Item entity는 Item을 UI에 표현하거나 다른 system에서 활용하는 정보를 저장한다. 따라서, Item entity는 Weak entity이다. Item이 존재하는 url, Item 이름을 나타내는 name, Item의 가격을 나타내는 price, Item의 이미지의 url 위치인 Image url, key attribute로 Item id를 가진다.

## B. Relations

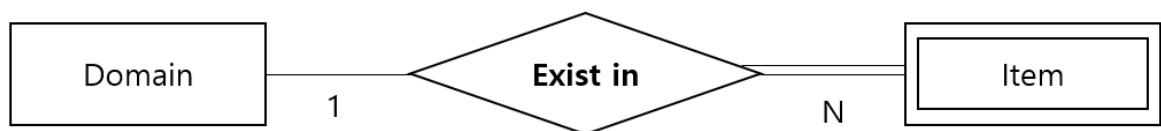
## B.1 Select



[Diagram 43] Select Relationship

User는 Item을 Select하는 Relation을 가진다. 이 때, 한 User는 여러 Item을 선택할 수 있으므로 User과 Item은 one-to-many Cardinality를 가진다. 또한, Item은 Total participation을 하고, User은 partial participation을 하는데, 이는 Item을 저장하는 용도가 User를 위한 개인화된 service인 stateful service이기 때문이다.

## B.2 Exist in

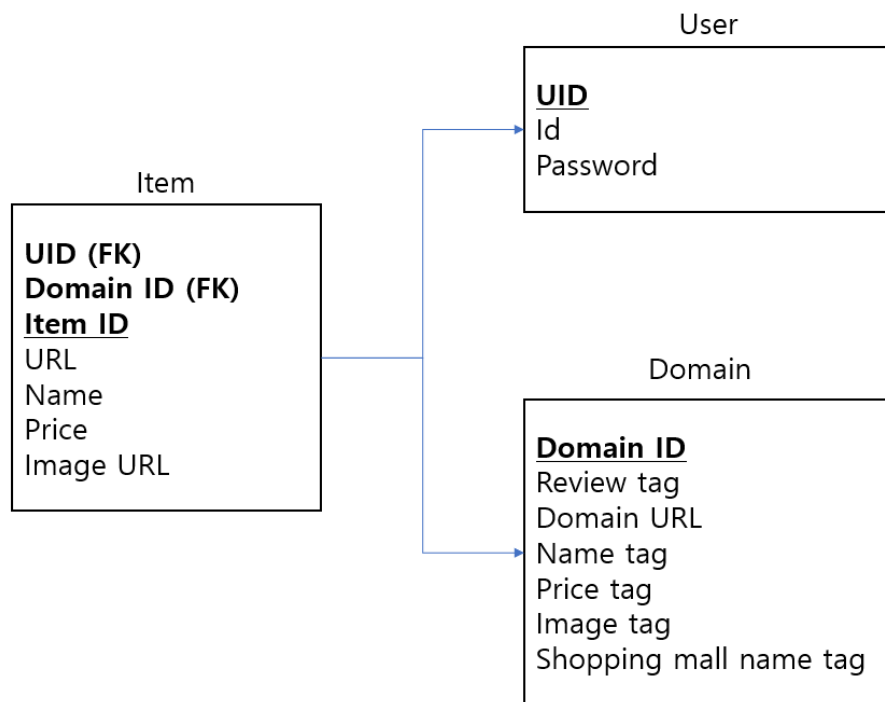


[Diagram 44] Exist in Relationship

Item entity는 Domain entity에 exist in하는 Relation을 가진다. 이 때, 한 Domain에는 여러 개의 Item이 포함되므로, Domain과 Item은 one-to-many Cardinality를 가진다. 또한, Item은 Total participation을 하고, Domain은 partial participation을 하는데, 이는 User가 고른 Item만으로 등록된 모든 Domain을 Cover할 수 없기 때문이다.

## 7.3 Relational Schema

Testing Plan에서는 Test Policy와 Test Case에 대해 설명한다. 본 장의 목적은 본 시스템이 의도한대로 실행이 되는지 검증하기 위함에 있다. Test Policy에서는 test의 단계적 접근을 설명하고, Test Case에서는 test에 사용할 입력 및 기대되는 출력을 서술한다.



[Diagram 45] Relational Schema

## 7.4 SQL DDL

### A. Domain

```

CREATE TABLE Domain
(
    Domain_id int not null,
    Review_tag VARCHAR(50) NOT NULL,
    Name_tag VARCHAR(50) NOT NULL,
    Price_tag VARCHAR(50) NOT NULL,
    Image_tag VARCHAR(50) NOT NULL,
    Shopping_mall_name_tag VARCHAR(50) NOT NULL,
    Domain URL VARCHAR(50) NOT NULL,
    PRIMARY KEY (Domain_id)
);
  
```

### B. User

```
CREATE TABLE User
(
    Uid int not null,
    ID VARCHAR(50) NOT NULL,
    Password VARCHAR(50) NOT NULL,
    PRIMARY KEY (Uid)
);
```

### C. Items

```
CREATE TABLE Item
(
    Domain_id int not null,
    UID int NOT NULL,
    Item_id int not null,
    URL VARCHAR(50) NOT NULL,
    Price VARCHAR(50) NOT NULL,
    Image_URL VARCHAR(50) NOT NULL,
    Name VARCHAR(50) NOT NULL,
    PRIMARY KEY (Domain_id, UID, Item_id),
    FOREIGN KEY (Domain_id) REFERENCES Domain(Domain_id),
    FOREIGN KEY (UID) REFERENCES User(UID)
);
```

## 8. Testing Plan

### 8.1 Objective

Testing Plan에서는 Test Policy와 Test Case에 대해 설명한다. 본 장의 목적은 본 시스템이 의도한대로 실행이 되는지 검증하기 위함에 있다. Test Policy에서는 test의 단계적 접근을 설명하고, Test Case에서는 test에 사용할 입력 및 기대되는 출력을 서술한다.

### 8.2 Testing Policy

#### A. Development Testing

Development Testing은 소프트웨어를 개발함에 있어서 위험을 줄이기 위해서 넓은 범위에서 결함을 예방하고 전략을 탐지하는 것을 목표로 한다.

따라서 Performance, Dependability, Security 측면에서 PICKET을 Testing하여 평가한다.



이 과정을 통해서 결정적인 결함을 탐지하고 최종적으로 고객에게 서비스가 전달되었을 때 생길 수 있는 치명적인 결함을 예방할 수 있다.

### **A.1. Performance**

본 소프트웨어는 크게 쇼핑몰에서 장바구니로 상품을 담는 쇼핑 카트 기능, 리뷰 기능으로 나뉜다. 이때 각각의 시스템과 그 하위 시스템이 해당 기능이 의도한 대로 사용자가 사용했을 때 올바르게 작동하는 것이 가장 중요하다.

구체적인 상황을 설명하자면 본 시스템은 사용자가 임의의 온라인 쇼핑몰 사이트에서 제품을 클릭하여 본 시스템에 담고, 해당 제품의 리뷰 분석을 받는 데 아무런 문제가 없어야 한다.

또한, 본 시스템에 담긴 제품들은 사이트 별로 자동적으로 분류가 되어야 한다. 한글로 된 어떤 쇼핑몰 사이트에 들어가도 본 시스템을 이용하는데 문제가 없어야 하며, 만약 등록이 안된 쇼핑몰 사이트가 있는 경우 개발자 측에 해당 사이트를 등록요청을 할 수 있도록 해야 한다.

이 Performance Test를 통해서 사용자가 사용하는 기능이 의도하지 않은 비정상적인 작동을 하거나 또는 아예 작동하지 않아서 생기는 문제를 탐지할 수 있다.

또, 각 기능이 해당 기능을 수행하는 데 걸리는 시간 즉 Speed에 관해서 Testing을 진행할 수 있다.

각 기능이 모두 1초 이내로 이루어져야 한다는 것을 정량적인 목표로 두고 Testing을 진행할 것이다. 이때 만약 속도의 측면에서 목표치가 달성되지 않는다면 데이터를 가져오는 방법이나 처리하는 과정에서 최적화를 할 수 있도록 소프트웨어를 수정할 수 있다.

### **A.2. Security**

본 소프트웨어에서 Security가 가장 보장되어야 할 부분은 사용자의 개인정보이다. PICKET을 사용할 때, 일회성으로 장바구니를 사용하지 않을 사용자는 필수적으로 장바구니에 담긴 상품들을 서버로 전달해 저장해야 하기 때문에 로그인 이 필요하다. 이때 아이디와 패스워드를 이용해 로그인 기능을 사용할 수 있는데, 이것은 사용자의 개인정보이기 때문에 Security가 보장되어야 한다.

사용자의 아이디와 패스워드를 저장하고 있는 데이터베이스와 서버의 관리가 중요하기 때문에 해쉬 알고리즘을 사용하는 등 별도의 보안 장치가 필요하다. 또, 서버와의 통신 과정 중에서 스니핑 등의 공격을 통해 개인 정보가 유출되지 않도록 별도의 암호화 프로토콜을 사용하는 등의 노력이 필요하다.

따라서 시스템의 Security Testing 과정에서는 로그인 과정에서 추가될 수 있는 보안 요소들이나 보안 알고리즘 등을 검사하고 평가한다.

### **A.3. Dependability**

본 시스템은 제품에 대한 리뷰를 분석하고 카테고리 별로 분류해 사용자가 원하는 카테고리의 리뷰를 확인할 수 있게 한다.

이때 리뷰를 특정한 카테고리로 분류하는 작업에서 머신러닝을 이용해 분석이 이루어진다. 그래서 많은 데이터를 이용해서 모델을 학습시켜서 모델의 정확도를 높이는 것이 가장 중요하다.

이 Testing에서 머신 러닝 모델의 정확도 수치가 낮아 리뷰의 분류가 제대로 되지 않는다면 사용자는 서비스를 사용하는데 있어서 신뢰성을 잃을 수 있으므로 중요한 Testing 요소가 될 수 있다.

### **B. Release Testing**

Release Testing은 소프트웨어나 애플리케이션의 최신 버전을 빌드하고 테스트하여 문제가 없고 의도한대로 작동하는지 확인하는 프로세스이다. Release이전에 수행되는 테스트로 디자인이나 코딩 이외에도 소프트웨어의 성공을 위한 가장 중요한 부분 중 하나이다. 이 Testing에서 좋은 평가가 이루어진다면 고객이 좀 더 서비스를 안심하고 원활하게 이용하는 것을 보장할 수 있다.



**[Diagram 46] Release Testing Process**

Release Testing 과정은 위의 그림을 따른다. 소프트웨어를 개발한 후 개발자들끼리 진행하는 Pre-alpha test를 진행하고, Alpha test를 통해 사용자를 대표할 사람들을 선정해 개발자들이 보는 앞에서 다같이 테스트를 진행할 수 있다. 또, 추후에 Beta test를 통해서 실제로 사용자들의 피드백을 받으며 요청사항들이나 수정할 부분을 수렴할 수 있다.

이후에 서비스를 Release한 후에도 사용자의 요청사항을 받는 기능이 있기 때문에 해당 기능을 이용하여 사용자들의 피드백을 받을 수 있다.

### **C. User Testing**

User Testing은 사용자의 환경에서 시스템을 테스트하는 것이다.

## **8.3 Testing Case**

소프트웨어 엔지니어링에서 Test Case는 특정 소프트웨어 테스트 목표를 달성하기 위해 실행할 단일 테스트를 정의하는 것이다. 구성 요소로는 입력, 실행 조건, 테스트 절차 및 예상 결과가 될 수 있다.

### **A. Authentication System**

### A.1. Register User

- 1) 사용자: 아이디와 비번을 만들어 회원가입을 시도한다.
- 2) 시스템 동작: Backed DB에 있는 ID와 중복 여부를 체크한다.

#### 2-1) 성공

시스템 동작: Backed DB에 새로 만든 사용자의 아이디 비번을 저장한다.

시스템 알림: "회원가입이 완료되었습니다."

#### 2-2) 실패

시스템 동작: 회원가입 시퀀스를 다시 시작한다.

시스템 알림: "이미 사용중인 ID입니다"

### A.2. Login User

- 1) 사용자: 아이디와 비번을 입력해 로그인을 시도한다.
- 2) 시스템 동작: 입력 받은 아이디와 비번에 Backend DB에 있는지 확인한다.

#### 2-1) 성공

시스템 동작: 사용자의 장바구니 정보를 불러온다.

시스템 알림: "로그인에 성공하셨습니다."

#### 2-2) 실패

시스템 동작: 로그인 시퀀스를 다시 시작한다.

시스템 알림: "존재하지 않는 아이디이거나, 비밀번호가 맞지 않습니다"

### A.3. Save Cart

- 1) 사용자: 장바구니 저장 버튼을 클릭한다.
- 2) 시스템 동작: Backend DB에 장바구니 정보를 저장한다.

#### 2-1) 성공

시스템 알림: "장바구니 저장이 완료되었습니다."

2-2) 실패

시스템 알림: "장바구니 저장이 실패하였습니다."

User Testing은 사용자의 환경에서 시스템을 테스트하는 것이다.

## **B. Item Configure System**

### **B.1. Item Parse**

- 1) 사용자: 장바구니에 담고 싶은 상품이 있을 시, 상품 저장 버튼을 클릭한다.
- 2) 시스템 동작: Backed DB에 있는 html item tag를 불러와 해당 웹 페이지에 있는 상품을 파싱한다.

2-1) 성공

시스템 알림: "장바구니에 저장되었습니다."

2-2) 실패

시스템 알림: "지원이 되지 않는 페이지 입니다."

## **C. Review Analysis System**

### **C.1. Classify Review**

- 1) 사용자: 장바구니에 담긴 상품 중 리뷰를 확인 하고 싶은 경우, 리뷰 보기 버튼을 클릭한다.
- 2) 시스템 동작: Backed DB에 있는 html review tag를 불러와 해당 웹 페이지에 있는 리뷰를 파싱한 후 분류한다.

2-1) 성공

시스템 알림: "리뷰를 불러왔습니다."

2-2) 실패

시스템 알림: "리뷰 기능이 지원 되지 않는 상품입니다."

## D. User Requirement System

### D.1. Domain Requirement

1) 사용자: 새로운 쇼핑몰 사이트를 등록해달라는 요청버튼을 클릭한다.

2) 시스템 동작: text파일에 해당 요청을 기록한다.

2-1) 성공

시스템 알림: "성공적으로 요청되었습니다."

2-2) 실패

시스템 동작: Domain Requirement 시퀀스를 다시 시작한다.

시스템 알림: "요청에 실패하였습니다."

### D.2. Else Requirement

1) 사용자: 요청할 내용을 텍스트로 적은 후 요청버튼을 클릭한다.

2) 시스템 동작: text파일에 해당 요청을 기록한다.

2-1) 성공

시스템 알림: "성공적으로 요청되었습니다."

2-2) 실패

시스템 동작: Else Requirement 시퀀스를 다시 시작한다.

시스템 알림: "요청에 실패하였습니다."

## 9. Development Plan

### 9.1 Objective

Development Plan에서는 본 시스템을 개발하는데 사용되는 프로그래밍 언어, 소프트웨어 프레임워크, 라이브러리 등 개발 환경에 대해서 설명한다. 또한, Gantt Chart를 이용

하여 개발 계획과 실제 개발 흐름에 대해 서술한다.

## 9.2 Frontend Environment

### A. Google Chrome Extension



[Figure 11] Google Chrome Extension Logo

구글 크롬 확장 프로그램(Google Chrome Extension) 구글 크롬 브라우저를 수정하는 브라우저 확장 프로그램이다. HTML, JavaScript, CSS와 같은 웹 기술을 기반으로 작성되며, 확장 프로그램은 크롬 웹 스토어(Chrome Web Store)를 통해 다운로드를 할 수 있다. 구글 계정을 가진 모든 사용자는 구글 크롬 확장 프로그램을 사용하여 개발 및 배포가 가능하다.

본 시스템은 구글 크롬 확장 프로그램을 이용하여 개발 및 배포가 될 예정이다. 해당 프로그램을 사용하는 이유는 브라우저 확장 프로그램을 이용함으로써 사용자가 어떤 쇼핑몰 사이트를 이용함과 동시에 오른쪽에 본 시스템을 이용할 수 있게 함에 있어서이다.

## 9.3 Backend Environment

### A. Python



[Figure 12] Python Logo

파이썬(Python)은 객체지향적 프로그래밍 언어 중 하나로 본 시스템의 Backend부분을 구현할 때 사용하게 될 프로그래밍 언어이다. 소켓 API가 잘 구현되어 있으며, 머신러닝을 돌리는데 있어서 파이썬이 편하기 때문에 이 프로그래밍 언어를 선택하게 되었다.

## B. MySQL



[Figure 13] MySQL Logo

마이에스큐엘(MySQL)은 세계에서 가장 많이 쓰이는 오픈 소스의 관계형 데이터베이스 관리 시스템이다. 다중 스레드, 다중 사용자 형식의 구조질의어 형식의 데이터베이스 관리 시스템으로서 오라클이 관리 및 지원하고 있다. 본 Backend System에서는 MySQL을 사용하여 데이터베이스를 관리 및 지원할 것이다.

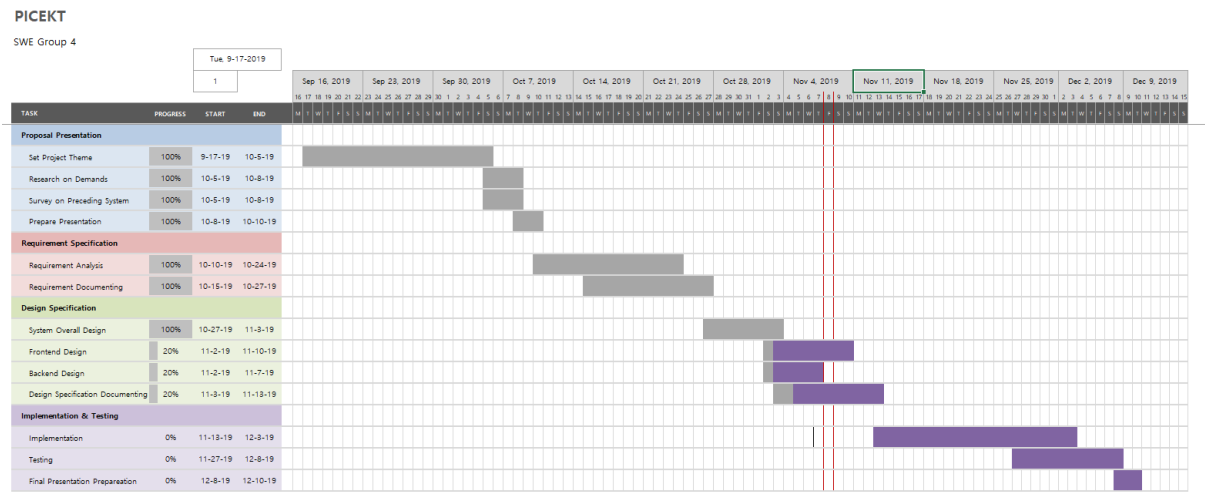
## 9.4 Schedule

### A. Gantt Chart

본 시스템의 현재까지의 개발 사항 및 앞으로의 계획을 Gantt Chart로 나타내었다. 주제 선정과 요구사항 명세서는 각각 4주, 3주 가량 걸렸다. 앞으로 디자인 명세서와 개발 및 테스트는 각각 2주, 4주가 걸릴 예정이다

가장 중요한 개발사항 일정은 Frontend 와 Backend를 동시에 구현을 함으로써 시간을 단축하고, 일정 부분의 구현 후부터 테스트 또한 병행으로 이루어질 계획이다.





[Table 2] Gantt Chart

# 10. Index

## 10.1 Objective

Index에서는 본 문서에서 사용된 그림, 다이어그램 및 테이블의 색인을 제공한다.

## 10.2 Table Index

[Table 1] Version History ..... 8

[Table 2] Gantt Chart..... 65

## 10.3 Figure Index

[Figure 1] UML Logo ..... 9

[Figure 2] UML Diagrams ..... 10

[Figure 3] Package Diagram Example..... 11

[Figure 4] Deployment Diagram Example ..... 12

[Figure 5] Class Diagram Example ..... 13

[Figure 6] State Diagram Example ..... 14

[Figure 7] Sequence Diagram Example ..... 14

[Figure 8] ER Diagram Example ..... 15

[Figure 9] Microsoft PowerPoint Logo .....	16
[Figure 10] JSON Logo .....	48
[Figure 11] Google Chrome Extension Logo.....	63
[Figure 12] Python Logo.....	64
[Figure 13] MySQL Logo.....	64

## 10.4 Diagram Index

[Diagram 1] System Organization Block Diagram.....	17
[Diagram 2] Frontend System Block Diagram .....	18
[Diagram 3] Backend System Block Diagram .....	19
[Diagram 4] Overall Frontend System Architecture .....	20
[Diagram 5] User Management System Class Diagram.....	21
[Diagram 6] User Management Sequence Diagram .....	22
[Diagram 7] User Management State Diagram .....	23
[Diagram 8] Shopping Cart System Class Diagram.....	23
[Diagram 9] Add Cart Sequence Diagram .....	25
[Diagram 10] Modify Product Sequence Diagram.....	25
[Diagram 11] Delete Product Sequence Diagram .....	26
[Diagram 12] Add Cart State Diagram .....	26
[Diagram 13] Modify Product State Diagram.....	26
[Diagram 14] Delete Product State Diagram .....	27
[Diagram 15] Review Classify System Class Diagram .....	27
[Diagram 16] Review Classify Sequence Diagram.....	28
[Diagram 17] Review Classify State Diagram.....	29
[Diagram 18] User Request System Class Diagram .....	29
[Diagram 19] User Request Sequence Diagram.....	30

[Diagram 20] User Request State Diagram.....	31
[Diagram 21] Overall Backend System Architecture .....	32
[Diagram 22] Authentication System Class Diagram .....	33
[Diagram 23] Register Sequence Diagram.....	35
[Diagram 24] Login Sequence Diagram.....	35
[Diagram 25] Save Cart Sequence Diagram .....	36
[Diagram 26] Register State Diagram.....	36
[Diagram 27] Login State Diagram.....	37
[Diagram 28] Save Cart State Diagram .....	37
[Diagram 29] Item Configure System Class Diagram .....	38
[Diagram 30] Item Configure Sequence Diagram .....	40
[Diagram 31] Item Configure State Diagram.....	40
[Diagram 32] Review Analysis System Class Diagram .....	41
[Diagram 33] Review Analysis Sequence Diagram.....	43
[Diagram 34] Review Analysis State Diagram.....	44
[Diagram 35] User Requirement System.....	44
[Diagram 36] User Requirement Sequence Diagram.....	46
[Diagram 37] Domain Requirement State Diagram.....	46
[Diagram 38] Else Requirement State Diagram .....	47
[Diagram 39] Overall ER Diagram .....	52
[Diagram 40] Domain ER Diagram .....	52
[Diagram 41] User ER Diagram .....	53
[Diagram 42] Item ER Diagram .....	53
[Diagram 43] Select Relationship .....	54
[Diagram 44] Exist in Relationship.....	54

[Diagram 45] Relational Schema .....	55
[Diagram 46] Release Testing Process .....	59