**Requirement Specification**

**PICKET**

|  |  |  |  |
| --- | --- | --- | --- |
| **제출일** | 2019. 10. 27 | **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**

[**1.** **Preface** 7](#_Toc23095459)

[**1.1 Objective** 7](#_Toc23095460)

[**1.2 Readership** 7](#_Toc23095461)

[**A. User Requirements Readership** 7](#_Toc23095462)

[**B. System Requirements Readership** 7](#_Toc23095463)

[**1.3 Document Structure** 7](#_Toc23095464)

[**A. Preface** 7](#_Toc23095465)

[**B. Introduction** 8](#_Toc23095466)

[**C. Glossary** 8](#_Toc23095467)

[**D. User Requirements Definition** 8](#_Toc23095468)

[**E. System Architecture** 8](#_Toc23095469)

[**F. System Requirements Specification** 8](#_Toc23095470)

[**G. System Models** 9](#_Toc23095471)

[**H. System Evolution** 9](#_Toc23095472)

[**I. Appendices** 9](#_Toc23095473)

[**J. Index** 9](#_Toc23095474)

[**1.4 Version History** 9](#_Toc23095475)

[**2.** **Introduction** 10](#_Toc23095476)

[**2.1 Objective** 10](#_Toc23095477)

[**2.2 Needs** 10](#_Toc23095478)

[**A. Increase in Online Shopping** 10](#_Toc23095479)

[**B. Increase dependency on reviews** 12](#_Toc23095480)

[**2.3 PICKET** 14](#_Toc23095481)

[**A. Integrated Web Shopping Cart** 14](#_Toc23095482)

[**B. Review Classification** 16](#_Toc23095483)

[**2.4 Expected Effects** 17](#_Toc23095484)

[**A. Improved Access to Product** 17](#_Toc23095485)

[**B. Easy Access to Information** 17](#_Toc23095486)

[**3.** **Glossary** 18](#_Toc23095487)

[**3.1 Objective** 18](#_Toc23095488)

[**3.2 Term Definition** 18](#_Toc23095489)

[**4.** **User Requirements Definitions** 20](#_Toc23095490)

[**4.1 Objective** 20](#_Toc23095491)

[**4.2 Functional Requirements** 20](#_Toc23095492)

[**A. Add to Cart** 20](#_Toc23095493)

[**B. Edit Information** 21](#_Toc23095494)

[**C. Delete Item** 21](#_Toc23095495)

[**D. Total Price** 21](#_Toc23095496)

[**E. Group by Web Site** 21](#_Toc23095497)

[**F. Review Classification** 22](#_Toc23095498)

[**G. User Request** 22](#_Toc23095499)

[**4.3 Non-Functional Requirements** 22](#_Toc23095500)

[**A. Product Requirements** 23](#_Toc23095501)

[**B. Organizational Requirements** 23](#_Toc23095502)

[**C. External Requirements** 24](#_Toc23095503)

[**5.** **System Architecture** 25](#_Toc23095504)

[**5.1 Objective** 25](#_Toc23095505)

[**5.2 Overall System Architecture** 25](#_Toc23095506)

[**5.3 Review Classify System** 26](#_Toc23095507)

[**5.4 Item Storing System** 27](#_Toc23095508)

[**5.5 Item Modify System** 28](#_Toc23095509)

[**5.6 Item Deletion System** 29](#_Toc23095510)

[**5.7 Price Checking System** 30](#_Toc23095511)

[**5.8 Item Classify System** 31](#_Toc23095512)

[**5.9 User Request System** 32](#_Toc23095513)

[**6.** **System Requirements Specifications** 33](#_Toc23095514)

[**6.1 Objective** 33](#_Toc23095515)

[**6.2 Functional Requirements** 33](#_Toc23095516)

[**A. Add to Cart Function** 33](#_Toc23095517)

[**B. Edit Information Function** 33](#_Toc23095518)

[**C. Delete Item Function** 34](#_Toc23095519)

[**D. Total Price Function** 34](#_Toc23095520)

[**E. Group by Web Site Function** 35](#_Toc23095521)

[**F. Review Classification Function** 35](#_Toc23095522)

[**G. User Request Function** 36](#_Toc23095523)

[**6.3 Non-Functional Requirements** 36](#_Toc23095524)

[**A. Product Requirements** 36](#_Toc23095525)

[**B. Organizational Requirements** 37](#_Toc23095526)

[**C. External Requirements** 38](#_Toc23095527)

[**6.4 Scenarios** 38](#_Toc23095528)

[**A. Adding Item into Cart Scenario** 38](#_Toc23095529)

[**B. Edit and Delete Item Scenario** 39](#_Toc23095530)

[**C. Grouping Scenarios** 40](#_Toc23095531)

[**D. Review Classification Scenarios** 41](#_Toc23095532)

[**E. User Request Scenarios** 41](#_Toc23095533)

[**7.** **System Models** 43](#_Toc23095534)

[**7.1 Objective** 43](#_Toc23095535)

[**7.2 Context Models** 43](#_Toc23095536)

[**A. Context Diagram** 43](#_Toc23095537)

[**B. Process Diagram** 43](#_Toc23095538)

[**7.3 Interaction Models** 44](#_Toc23095539)

[**A. Use-Case Diagram** 44](#_Toc23095540)

[**B. Tabular description for each use case** 44](#_Toc23095541)

[**C. Sequence Diagram** 47](#_Toc23095542)

[**7.4 Structural Models** 48](#_Toc23095543)

[**A. Class Diagram** 48](#_Toc23095544)

[**7.5 Behavioral Models** 49](#_Toc23095545)

[**A. Activity Diagram** 49](#_Toc23095546)

[**B. State Diagram** 51](#_Toc23095547)

[**8.** **System Evolution** 55](#_Toc23095548)

[**8.1 Objective** 55](#_Toc23095549)

[**8.2 Limitation** 55](#_Toc23095550)

[**A. Chrome Extension** 55](#_Toc23095551)

[**B. Shopping Mall Sites** 55](#_Toc23095552)

[**C. Payment** 55](#_Toc23095553)

[**8.3 Evolution** 55](#_Toc23095554)

[**A. Chrome Extension** 55](#_Toc23095555)

[**B. Shopping Mall Sites** 55](#_Toc23095556)

[**C. Payment** 56](#_Toc23095557)

[**9.** **Appendices** 57](#_Toc23095558)

[**9.1 Objective** 57](#_Toc23095559)

[**9.2 Database Requirements** 57](#_Toc23095560)

[**9.3 Hardware Requirements** 57](#_Toc23095561)

[**10.** **Index** 58](#_Toc23095562)

[**10.1 Objective** 58](#_Toc23095563)

[**10.2 Table Index** 58](#_Toc23095564)

[**10.3 Figure Index** 58](#_Toc23095565)

[**10.4 Diagram Index** 59](#_Toc23095566)

1. **Preface**
   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. **Readership**

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

* + 1. **User Requirements Readership**

User requirements는 본 시스템을 사용할 고객을 예상 독자로 간주한다. 해당 부분은 전문성이 없는 사용자 입장에서 이해하기 쉽게 하기 위해 전문용어의 사용을 지양하고, 자연어와 다이어그램(Diagram) 등의 시각자료를 통해 본 시스템이 제공하는 서비스 및 제약사항에 대하여 서술한다.

* + 1. **System Requirements Readership**

System Requirements는 시스템 개발자와 시스템 설계자 등과 같은 개발진을 주요 독자로 간주한다. 해당 부분은 구조적 언어를 이용하여 본 시스템이 제공하는 기능 및 제약사항을 보다 구체적으로 서술한다. 시스템의 기능에 대한 부분은 고객과 계약자 사이의 계약에 포함될 수 있다.

* 1. **Document Structure**

본 문서는 총 10개의 장으로 구성되어 있다. 10개의 장은 Preface, Introduction, Glossary, User Requirements Definition, System Architecture, System Requirements Specification, System Models, System Evolution, Appendices, Index로 구성되어 있으며, 각 장에서 서술되는 내용 및 역할은 다음과 같다.

* + 1. **Preface**

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

* + 1. **Introduction**

Introduction에서는 본 서비스의 필요성과 본 서비스가 고객의 어떤 요구사항을 반영하고 있는지에 대해 서술한다. 또한, 본 시스템이 제공하는 기능 및 본 시스템이 다른 시스템들과 어떻게 상호작용을 하는지에 대하여 설명한다. 마지막으로 본 서비스를 도입함으로써 기대하고 있는 효과에 대하여 서술한다.

* + 1. **Glossary**

Glossary에서는 본 문서에 등장하는 기술적 용어들에 대해 정의한다. 전문 지식이 없는 독자가 본 문서를 읽더라도 이해할 수 있도록 가능한 자세하게 용어에 대해 설명한다.

* + 1. **User Requirements Definition**

User Requirement Definition에서는 사용자에게 어떠한 서비스를 제공하는지에 대해 서술한다. 기능적 요구사항과 비기능적 요구사항으로 나눠 시스템의 요구사항을 서술한다. 자연어와 함께, 다이어그램과 같은 시각적인 자료를 동원해 서술하여 사용자의 이해를 돕는다.

* + 1. **System Architecture**

System Architecture에서는 본 시스템에 대한 높은 수준의 시스템 구조를 보여준다. 본 시스템의 기능들이 각 서브 시스템에 어떻게 할당되어 분포하는지 및 시스템들의 관계와 구조에 대해 추상적으로 서술한다.

* + 1. **System Requirements Specification**

System Requirement Specification에서는 기능적 요구사항과 비기능적 요구사항을 구체적으로 서술한다. 이때 본 파트는 시스템 설계 단계와 구현 단계를 포함한 개발 프로세스 전반에서 사용됨으로 도표 등을 이용하여 체계적으로 설명한다.

* + 1. **System Models**

System Models에서는 본 시스템과 그 컴포넌트(Component), 동작 환경 사이의 관계들을 시각적인 자료로 나타낸다. Context Diagram, Process Diagram, Use-Case Diagram 등 다양한 다이어그램을 사용한다.

* + 1. **System Evolution**

System Evolution에서는 본 시스템이 취하고 있는 기본적인 가정들을 서술한다. 또한, 향후 본 시스템에 발생할 수 있는 예상 가능한 모든 변경사항과 그에 대한 대응 방안에 대해서 서술한다. 변경사항에는 하드웨어(Hardware)의 발전, 사용자 요구사항의 변화 등이 포함된다.

* + 1. **Appendices**

Appendices에서는 본 문서에서 생략된 하드웨어와 데이터베이스(Database) 요구사항에 대해 서술한다.

* + 1. **Index**

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

* 1. **Version History**

|  |  |  |
| --- | --- | --- |
| Version | Modified Date | Explanation |
| 1.0 | 2019. 10. 15 | 요구사항 명세서 목차 작성 |
| 1.1 | 2019. 10. 16 | 요구사항 명세서 형식 작성 |
| 2.0 | 2019. 10. 17 | Preface 작성 완료 |
| 2.1 | 2019. 10. 25 | 유저 및 시스템 요구사항 작성 완료 |
| 2.2 | 2019. 10. 26 | 나머지 항목 작성 완료 |
| 3.0 | 2019. 10. 27 | 통합 후 검토 및 수정 완료 |

[Table 1] Version History

1. **Introduction**
   1. **Objective**

Introduction에서는 본 서비스인 Picket의 서비스 제안 배경에 대해 밝히고 서비스의 필요성에 대해 논한다. 그러한 필요성에 의해 도출된 Picket의 기능을 상세하게 설명한다. 마지막으로 본 서비스의 도입으로 인한 기대효과를 기술한다.

* 1. **Needs**
     1. **Increase in Online Shopping**

오프라인 매장에서 실제 물건을 확인한 후 온라인에서 좀 더 저렴한 가격에 구매하는 쇼루밍 현상의 비율이 높아지면서 상품을 구매하는 최종 공간은 오프라인 매장이 아니라 온라인이 된 것은 아주 당연한 일이 되었다.

특정 상품 군에 대해서는 온라인을 통한 구매 비율이 오프라인을 뛰어넘는 상품군도 존재하고 있을 정도로 온라인 쇼핑은 현대인의 생활에서 높은 비중을 차지하고 있다. 특히, 온라인 구매의 특징인 간편한 결제 시스템, 빠른 배송, 사용자 맞춤형 상품추천 등은 꾸준히 온라인쇼핑의 상승세에 크게 기여를 하고 있다.



[Figure 1] 2019년 1월 온라인쇼핑의 동향

그림 1에서 통계청에서 발표한 “2019년 1월 온라인쇼핑 동향” 보고서에 따르면 온라인쇼핑 거래 액은 10조 7,034억원으로 전년동월대비 17.9% 증가한 것을 알 수 있다.

온라인 쇼핑의 증가 추세에 따라 상품을 구매할 수 있는 온라인 쇼핑몰은 셀 수 없을 정도로 많아졌다. 통합적인 상품 군을 제공하는 소셜커머스, 대형 오프라인 몰을 기반으로 한 대기업들의 온라인 쇼핑몰의 운영이 대표적인 종합쇼핑몰의 예시이고, 뿐만 아니라 제품 군 별로 다양한 형태의 웹 쇼핑몰이 존재한다. 특히 패션의류나 액세서리 부분에서는 패션의 트랜드나 컨셉에 따라 다양한 타겟층을 분류해서 맞춤형 쇼핑몰들이 존재한다. 그림 2와 그림 3은 소셜커머스와 자체 의류 쇼핑몰의 예시이다.



[Figure 2] 소셜커머스 쇼핑몰



[Figure 3] 자체 의류 쇼핑몰

이렇게 많은 사이트들이 존재하는 것에 따라 소비자의 구매형태도 매우 다양하다. 이 중 크게 두 가지 유형의 소비자 케이스를 통해 웹 쇼핑의 불편한 점과 그로 인한 PICKET 통합 장바구니 기능의 필요성을 설명하고자 한다.

**A.1 여러 사이트에서 상품을 구매하는 소비자**

특히 패션의류의 부분에서는 종합쇼핑몰을 이용하는 것이 아니라 소비자마다 애용하는 사이트들이 존재한다. 이러한 경우에는 각각의 하나의 홈페이지는 그 사이트만의 장바구니를 제공하지만 타회사의 사이트에 있는 상품을 함께 관리해주는 장바구니는 이용할 수 없다.

즉, 각각의 사이트의 장바구니에 원하는 상품을 담아 놓고 사이트를 옮겨 가며 결제를 하거나 그런 것이 싫은 경우 그 중 일부 사이트의 구매는 포기하기도 한다.

우리는 이러한 문제를 인터넷 쇼핑의 가장 큰 불편함으로 생각했고, 인터넷에 존재하는 모든 온라인 쇼핑몰의 각각의 상품을 하나의 장바구니에 담을 수 있는 PICKET이라는 확장 앱 서비스를 제안하고자 한다.

**A.2 심사숙고를 하며 여러 날에 걸쳐 구매를 하는 소비자**

가격대가 좀 큰 물건을 구매하거나 결정을 하는데 시간이 오래 걸리는 구매 행태를 가지는 소비자의 경우에는 하나의 물건을 구매하는데 하루 이상이 걸리기도 한다. 이 때는, 인터넷 쇼핑을 하면서 마음에 드는 상품이 있는 사이트 주소를 ‘즐겨 찾기’기능을 이용하여 북 마크에 저장을 해놓거나 링크를 별도로 저장하기도 한다.

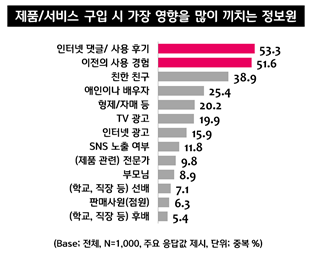
링크를 저장하지 않고 브라우저를 종료한 사용자의 경우에는 다시 기억을 더듬어 자신이 접속한 홈페이지에 들어가 원하는 상품 군을 키워드나 상품명으로 검색해 찾기도 한다. 이 경우에는 지난 날에 봤던 상품과 일치하는 상품을 찾을 수 있다는 보장을 하기는 어렵다.

이러한 소비의 경우도 우리는 온라인 쇼핑에서 큰 불편함 중 하나라고 판단했고, 통합 웹 장바구니인 PICKET을 이용해 원하는 상품을 담아 놓은 후 브라우저를 종료하고 다시 접속했을 때도 그때 그 상품이 장바구니에 그대로 남아있는 쇼핑 서비스를 제공하고자 한다.

이렇게 온라인 시장이 증가하면서 온라인 쇼핑몰의 종류와 개수도 함께 증가하게 되었고 그로 인해 사용자들은 각 사이트에서 각각의 상품을 구매하는 것에 대한 불편함이 존재한다. 따라서 웹 기반의 온라인 쇼핑몰을 대상으로 하는 통합 장바구니라는 기능의 필요성을 도출할 수 있었고, 해당 기능을 포함한 브라우저 확장 프로그램이라는 형태로 PICKET이라는 서비스를 제안하고자 한다.

* + 1. **Increase dependency on reviews**

온라인 시장이 증가하면서 함께 스마트한 소비가 늘어나며 제품에 대한 후기가 다른 소비자들에게 미치는 영향은 매우 크다.



[Figure 4] 구매에 영향을 끼치는 정보원 조사

그림 4에서는 시장조사전문기업 엠브레인 트렌드모니터가 전국 만 19세에서 만59세 성인 남녀 1,000명을 대상으로 소비생활에서의 영향력 설문조사를 실시한 결과 평소 제품과 서비스를 구매 및 이용할 때 소비자들에게 가장 많은 영향을 끼치는 정보원은 53.3%로 인터넷 댓글, 사용후기가 매우 중요한 구매 과정 중 하나인 것을 확인할 수 있다.

이것은 많은 사람들이 서비스를 구매하거나 이용할 때, 전문가들의 의견보다도 실제 사용경험이 있는 소비자의 의견이나 주변 사람들의 의견을 훨씬 중요하게 고려하여 제품을 선택한다는 것을 보여준다.

이렇게 소비자 후기를 확인하는 것이 소비 과정 중 일부로 자리잡은 만큼, 구매자들은 물건을 온라인으로 구매하기 전에 여러 사이트를 찾아 다니며 같은 제품에 대한 다양한 리뷰를 보고 스스로 평가하기를 원한다. 긍정적인 리뷰와 부정적인 리뷰를 모두 고려하여 구매하고자 하는 제품이 좋다면 어떠한 이유로 긍정적인 리뷰를 남겼는지, 또는 부정적이라면 그 이유가 무엇인지가 구매를 결정하는데 결정적인 요인이 된다.

아래의 그림 5와 같은 리뷰 예시를 보면 실제로 구매했던 사람의 리뷰가 상품의 품질 때문에 낮은 평점이 달린 것을 확인할 수 있다. 하지만 직접 리뷰의 글을 읽어 보기 전까지는 왜 낮은 평점인지 알 수 없다. 또, 많은 리뷰들 사이에 묻혀 소비자가 원하는 리뷰를 확인하지 못할 가능성도 매우 높다.

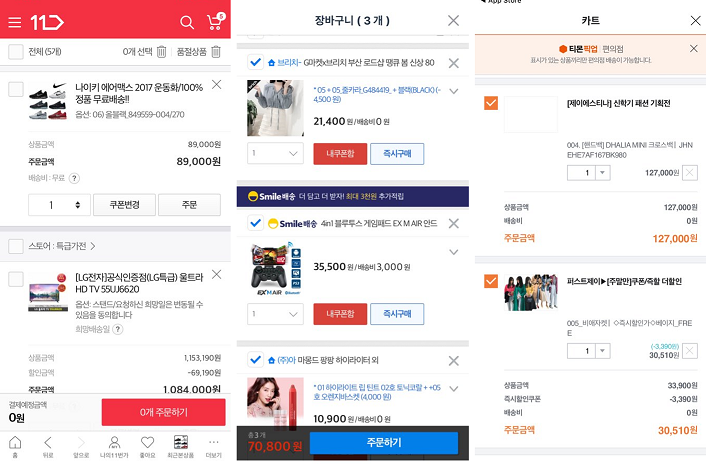


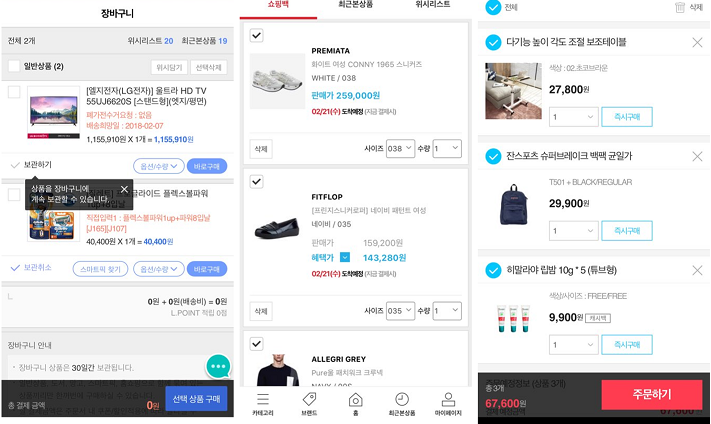
[Figure 5] 상품의 품질 불만에 대한 리뷰 예시

또, 온라인으로 제품을 구매하거나 이용하는 소비자들 중에서 가장 중요하게 생각하는 부분이 다를 수 있다. 예를 들어, 빠른 배송을 가장 우선순위로 하는 구매자가 있을 수 있고, 상품의 품질이 가장 중요한 구매자이거나, 다른 사람들의 구매 횟수가 판단하는 기준이 되는 소비자 등 다양한 우선순위를 가진 구매자들이 존재할 수 있다. 이러한 소비자들은 각각의 우선순위 항목에 맞는 리뷰를 찾아서 보고 싶어할 것이다.

따라서, 다양한 리뷰에 대한 수집과 분류에 대한 필요성에 의해 해당 기능을 포함한 PICKET을 제안하고자 한다.

* 1. **PICKET**
     1. **Integrated Web Shopping Cart**



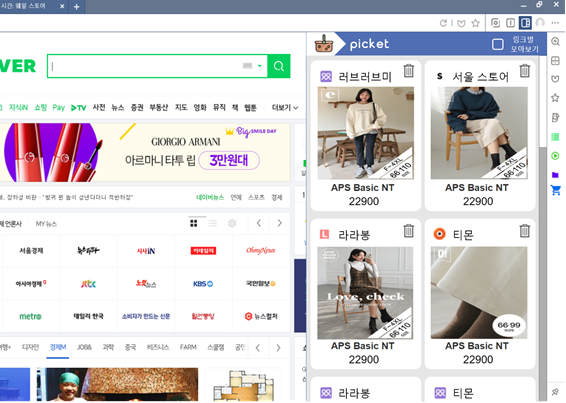


[Figure 6] 다양한 웹 사이트 장바구니 예시

온라인 시장이 증가하면서 온라인 쇼핑몰의 종류와 개수도 함께 증가하게 되었고 그로 인해 사용자들은 각 사이트에서 각각의 상품을 구매하는 것에 대한 불편함이 존재한다. 그림 6과 같이 각각의 쇼핑몰 별로 장바구니 기능을 제공 하고 있다.

하지만 여러 사이트를 같은 상품이라도 여러 사이트를 비교해가며 구매를 하는 소비자의 행동 패턴은 각 사이트에 존재하는 장바구니 기능은 오히려 불편함만 가중 시킨다. 특히 특정 웹 쇼핑몰 사이트에서는 로그인을 해야만 장바구니 기능을 이용할 수 있어 번거로움을 더한다. 여기에 회원가입이 되어 있지 않는 웹 쇼핑몰 사이트라면 회원 가입이라는 절차가 더해진다.

이렇게 최악의 경우에 소비자는 단순히 가격을 비교하기 위해서 상품을 장바구니에 담기 위해서 회원가입, 로그인, 장바구니 이렇게 총 3단계를 거쳐야 한다.

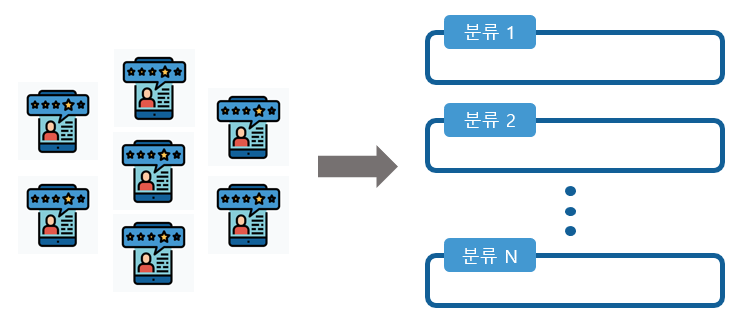


[Figure 7] PICEKT 장바구니 화면

따라서 이러한 불편함을 해소하기 위해서 가능한 모든 웹 사이트를 대상으로 하는 통 합 웹 장바구니 기능이 필수적이다. 따라서 PICKET의 가장 주요한 기능으로는 웹 통합 장바구니 기능이라 할 수 있다.

그림 7과 같이 모든 웹 사이트를 대상으로 상품 하나하나를 브라우저의 확장 프로그램인 PICKET에 Basket 버튼을 한 번 누름으로써 상품이 장바구니에 담기게 된다. 이러한 점에서 상품을 여러 사이트에서 비교하는데 사용자에게 편리함을 줄 수 있다.

* + 1. **Review Classification**



[Figure 8] 리뷰 분류 기능 모식도

원래 해당 상품에 대한 리뷰를 사이트의 리뷰 화면에 가서 직접 하나하나 별점을 확인하고 직접 읽어보면서 얻고 싶은 정보를 얻는 데에 시간이 오래 걸린 불편함이 있었다.

이 것을 해결하기 위하여 리뷰 분류기능을 이용하여 해당 제품의 리뷰를 배송, 품질, 가격, 별점 순 등 대분류로 나누어 각 카테고리와 관련된 리뷰를 확인할 수 있다.

* 1. **Expected Effects**
     1. **Improved Access to Product**

온라인 쇼핑몰 사이트를 돌아다니며 가격 비교를 하거나 상품을 비교할 때, 여러 탭을 띄워 놓고 상품을 구매하는 불편함이 있었다면 PICKET을 사용함으로써 통합 웹 장바구니로 사용할 수 있기 때문에 상품 비교에 용이하다.

또, 제품을 구매하는데 하루 이상이 걸리는 제품들의 경우 해당 제품의 링크를 별도로 저장하는 과정이 필요한데, PICKET을 이용하면 그러한 불편함이 사라져 상품에 대한 접근 성이 개선될 수 있다.

* + 1. **Easy Access to Information**

해당 제품을 구매하기 전 원하는 정보를 얻기 위해 리뷰를 보는 과정에서 모든 리뷰를 하나하나 확인해가며 읽어보는 것에 불편함에 있었다면, PICKET을 이용하여 리뷰의 종류를 선택하여 편하게 정보를 얻을 수 있다는 점에서 소비자입장에서 정보의 접근이 용이하다고 할 수 있다.

1. **Glossary**
   1. **Objective**

Glossary에서는 본 문서에 등장하는 기술적 용어들에 대해 정의한다. 전문 지식이 없는 독자가 본 문서를 읽더라도 이해할 수 있도록 가능한 자세하게 용어에 대해 설명한다.

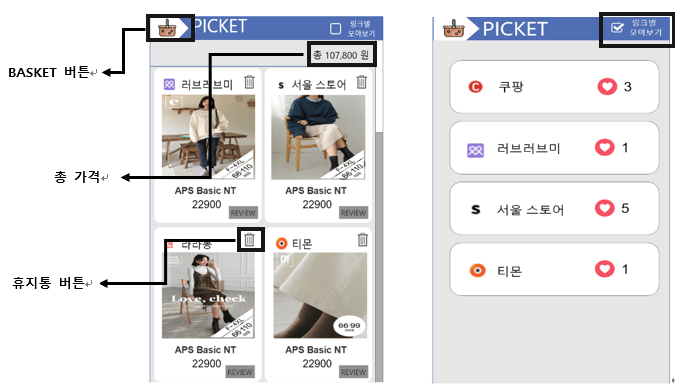
* 1. **Term Definition**

|  |  |
| --- | --- |
| Term | Definition |
| 하드웨어(hardware) | 중앙 처리 장치, 모니터, 자판, 컴퓨터 기억 장치, 그래픽 카드, 사운드 카드, 메인보드와 같은 컴퓨터의 물리적 부품을 지칭한다. |
| 데이터베이스(Database) | 여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합이다.[1] 작성된 목록으로써 여러 응용 시스템들의 통합된 정보들을 저장하여 운영할 수 있는 공용 데이터들의 묶음이다. |
| 파싱(Parsing) | 컴퓨터 과학에서 파싱은 일련의 문자열을 의미 있는 토큰(token)으로 분해하고 이들로 이루어진 파스 트리(parse tree)를 만드는 과정을 말한다. |
| 머신러닝(Machine Learning) | 기계 학습 또는 머신 러닝)은 인공 지능의 한 분야로, 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 말한다. |
| 딥러닝(Deep Learning) | 딥 러닝 또는 심층학습은 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화를 시도하는 기계학습 알고리즘의 집합으로 정의되며, 큰 틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야라고 이야기할 수 있다. |
| 구글 크롬 확장프로그램(Google Chrome Extension) | 구글 크롬 확장 프로그램은 구글 크롬 브라우저를 수정하는 브라우저 확장 프로그램 이다. 이러한 확장은 HTML, JavaScript 및 CSS 와 같은 웹 기술을 기반으로 사용하여 작성된다. |

1. **User Requirements Definitions**
   1. **Objective**

User Requirement Definition에서는 사용자에게 어떠한 서비스를 제공하는지에 대해 서술한다. 기능적 요구사항과 비기능적 요구사항으로 나눠 시스템의 요구사항을 서술한다. 자연어와 함께, 다이어그램과 같은 시각적인 자료를 동원해 서술하여 사용자의 이해를 돕는다.

* 1. **Functional Requirements**



[Figure 9] 화면 구성 1

* + 1. **Add to Cart**

한 사이트의 장바구니와 같은 기능으로 웹 기반의 온라인 쇼핑몰 페이지에서 어떠한 상품이든 PICKET 확장 앱을 열어 왼쪽 상단의 Basket 버튼을 누르는 형태로 장바구니에 상품을 담을 수 있다.

상품을 담은 이후의 화면은 그림 9의 왼쪽 화면처럼 사이트의 이름과 상품사진, 상품이름과 상품가격정보가 간단하게 적힌 사각형 카드 형태로 (이하 제품 카드) 브라우저의 확장 프로그램인 PICKET에 저장된다.

* + 1. **Edit Information**

제품 카드에 적힌 정보가 잘못 입력되거나 아예 입력이 안 되는 경우, 사용자가 직접 사이트 명, 상품명, 상품 가격 등의 정보를 수정할 수 있다.

이 때, 자동으로 입력되는 제품 카드의 정보보다 사용자가 직접 수정한 정보를 가장 우선으로 하여 반영한다.

* + 1. **Delete Item**

제품 카드마다 오른쪽 상단에 휴지통 버튼을 클릭하면 제품카드를 삭제할 수 있어 최종적으로 구매하지 않겠다고 결정한 경우 쉽게 PICKET에서 물건을 삭제할 수 있다.

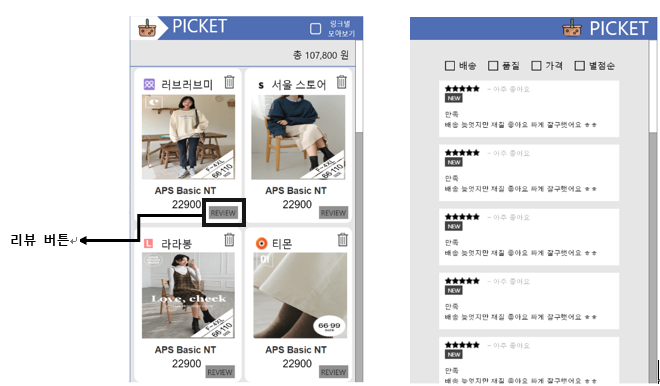
삭제 시 제품 카드가 PICKET에서 삭제되고 총 상품 가격에서 해당 제품의 가격이 빠진 가격으로 다시 변경된다.

* + 1. **Total Price**

PICKET의 오른쪽 상단에 현재 PICKET에 담은 제품들의 총 가격의 합을 보여주는 기능도 있다. 이 기능을 통해 소비자들이 각자의 소비 예산과 비교해 가며 쇼핑을 할 수 있다.

* + 1. **Group by Web Site**

확장 앱을 열고 오른쪽 상단의 링크 별 모아보기의 체크박스 (그림 9의 오른쪽 화면 구성 참조)를 체크하면 사이트 별로 picket에 담아 놓은 제품의 수를 확인할 수 있다. 그 중에서 한 사이트를 선택한다면 그 사이트에서만 선택하여 장바구니에 담은 제품을 확인할 수 있다.



[Figure 10] 화면 구성 2

* + 1. **Review Classification**

그림 10의 왼쪽 화면에서 제품 카드에서 오른쪽 하단에 있는 REVIEW 버튼을 누르면 그림 10의 오른쪽 화면과 같은 해당 제품의 리뷰 화면으로 넘어가게 된다.

이 때, 해당 제품의 리뷰를 배송, 품질, 가격, 별점 순 등으로 대분류로 나누어 각 카테고리와 관련된 리뷰를 확인할 수 있다. 예를 들어, 리뷰 화면에서 배송 버튼을 클릭하면 리뷰들 중에서 배송과 관련된 리뷰들만 확인할 수 있다.

* + 1. **User Request**

자동으로 PICKET에 제품 카드로 만들어 지지 않는 웹 쇼핑몰 사이트에 대해서는 사용자가 직접 요청하여 PICKET의 서비스 대상이 되게 할 수 있다.

서비스 대상 웹 사이트 요청뿐만 아니라 사용자들이 사용하면서 불편한 점이나 개선 사항에 대한 의견을 받을 수 있는 기능을 추가한다.

* 1. **Non-Functional Requirements**

Non-Functional Requirements는 Product Requirements, Organizational Requirements, External Requirements로 나누어진다. 각 비기능적 요구사항들은 더 세부적으로 분류될 수 있는데, 그 중 본 시스템에 필요한 것들을 서술한다.

* + 1. **Product Requirements**

**A.1 Usability Requirements**

본 시스템은 온라인 쇼핑의 이용에 도움을 주는 기능을 제공한다. 본 시스템에서 가장 중요하게 여겨지는 기능은 여러 분산된 쇼핑몰 사이트에 있는 제품들을 하나의 장바구니에 담는 기능이다. 따라서 본 시스템이 제공하는 UI는 사용자가 제품을 장바구니에 담기 편리해야 하며, 제품의 사이트 별 분류, 총 가격 확인, 리뷰 분석 등이 직관적으로 확인 가능해야 한다.

**A.2 Performance Requirements**

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.

**A.3 Dependability Requirements**

본 시스템은 제품에 대한 리뷰분석을 서버와 통신을 통해 가져오게 된다. 이 과정에 있어서 안전성이 보장되어야 한다. 또한, 리뷰 분석이 제대로 이뤄지지 않을 경우 사용자의 신뢰를 잃을 수 있으므로 최대한 많은 리뷰 데이터를 모아 정확한 분석을 할 수 있어야 한다.

**A.4 Security Requirements**

본 시스템은 사용자의 개인정보를 필요로 하지 않는다. 사용자가 담는 제품들은 사용자의 개인 브라우저 상에서 저장이 되고 관리된다. 따라서, 본 시스템에 있어서 사용자의 개인정보 유출은 고려하지 않아도 된다.

* + 1. **Organizational Requirements**

**B.1 Environmental Requirements**

본 시스템은 기본적으로 PC 구동에 초점을 시스템이다. Chrome 브라우저를 기본 브라우저로 두고 있음으로, 사용자가 언제 어디서든 크롬 브라우저에 로그인을 하면 사용할 수 있어야 한다.

**B.2 Operational Requirements**

리뷰 분석기능이 돌아가는 서버는 상시 구동이 되어야 한다. 나머지 기능들은 사용자의 브라우저 내에서 작동이 가능해야 한다.

**B.3 Development Requirements**

보다 정확한 리뷰 분석이 가능해야 하며, 이를 달성하기 위해 최신 분석 기술을 가져다 쓴다. 또한, 웹 브라우저에서 작동을 해야 하므로 해당 환경에서 작동하는 웹 언어를 사용한다.

* + 1. **External Requirements**

**C.1 Regulatory Requirements**

본 시스템은 Chrome 브라우저 기반으로 작성이 되기 때문에 Chrome에서 제공하는 규약을 따라야 한다.

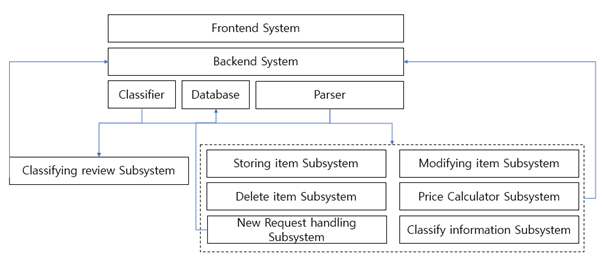
**C.2 Safety/Security Requirements**

본 시스템은 등록이 되지 않은 쇼핑몰 사이트에 대해 고객이 등록 요청을 할 수 있다. 이 과정에서 외부자의 접근을 막을 수 있어야 한다.

1. **System Architecture**
   1. **Objective**

System Architecture에서는 System module 간의 functions의 distribution과 더불어, anticipated system architecture의 overview에 대해 명시한다. 또한, 추가적으로 Subsystem에 대한 간략한 Architecture도 함께 소개한다. 본 System architecture는 크게 Overall architecture과 Subsystem architecture로 구성된다.

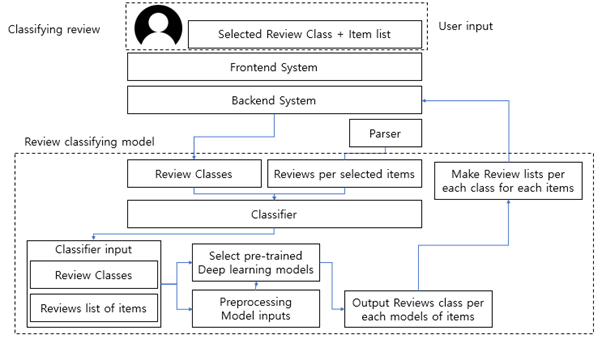
* 1. **Overall System Architecture**



[Diagram 1] Overall System Architecture

System Overall Architecture이다. Top-down direction으로는 Frontend System에서 User input을 받아 목적에 따라 Backend System의 여러 Component로 이를 전달한다. 전달된 정보는 Subsystem에서 Processing되는데, 크게 Classifier의 Subsystem인 Classifying review Subsystem, Parser의 Subsystem인 Storing item Subsystem, Modifying item Subsystem, Delete item Subsystem, Price Calculator Subsystem, New Request handling Subsystem, Classify information Subsystem으로 구성된다. Processed data는 다시 Down-top direction으로 Backend System에서의 Post processing을 거쳐 Frontend System에서 UI를 통해 표현된다.

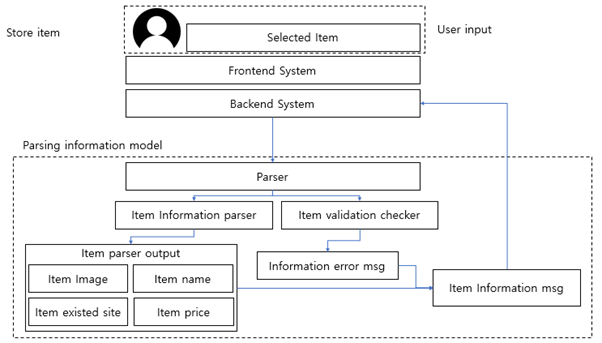
* 1. **Review Classify System**



[Diagram 2] Review Classify System Architecture

Classifying review Subsystem에 대한 Architecture이다. User가 선택한 Review Class와 Item list를 input으로 받는다. 이는 Backend System에서 처리가 완료되는데, 이를 처리하는 모듈을 Review classifying model이라고 정의하였다. Parser에 의해 User가 Review classifying을 원하는 Item에 identifier를 부여하며, 이는 Deep-learning model을 활용한 Pre-training된 Classifier에서 각각의 Item의 Review에 대한 각각의 Review class의 일치 여부를 Review 개수를 depth로 가지는 k-hot vector 형식의 Output을 제공한다. 제공된 k-hot vector는 backend system에서 Post processed되어 User의 UI에 제공된다.

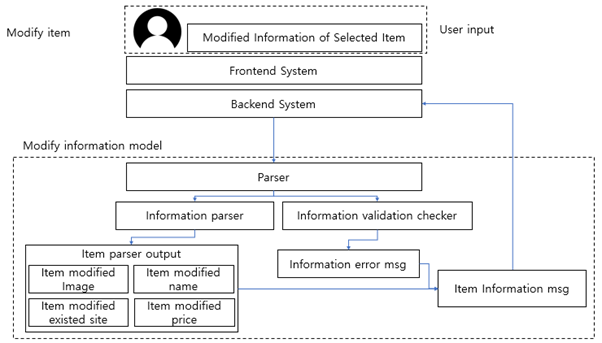
* 1. **Item Storing System**



[Diagram 3] Item Storing System Architecture

Storing item Subsystem에 대한 Architecture이다. User가 선택한 Item을 input으로 받는다. 이는 Parsing information model에서 처리되는데, Parser에 의해 1. Item information을 parsing, 2. Item information의 validation이 확인된다. 1.에 의해 UI에 표현하기 위한 Item image, item name, item exited site, item price의 정보를 얻어내고, 2.에 의해 Item information에 대한 error message를 생성한다. 이 두 정보는 합쳐져 Item information message를 형성해 Backend System에 제공한다.

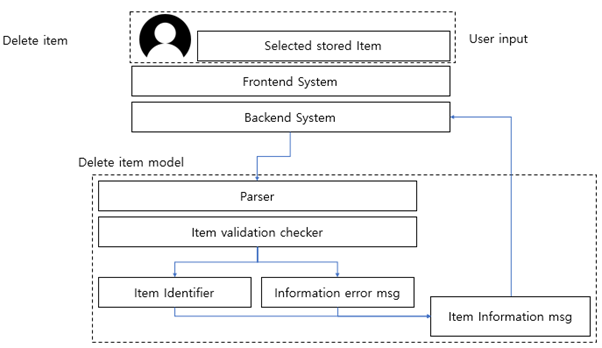
* 1. **Item Modify System**



[Diagram 4] Item Modify System Architecture

Modifying item Subsystem에 대한 Architecture이다. User가 선택한 Selected item의 Modified information을 input으로 받는다. 이는 Modify information model에서 처리되는데, Storing item subsystem과 유사하게 처리된다.

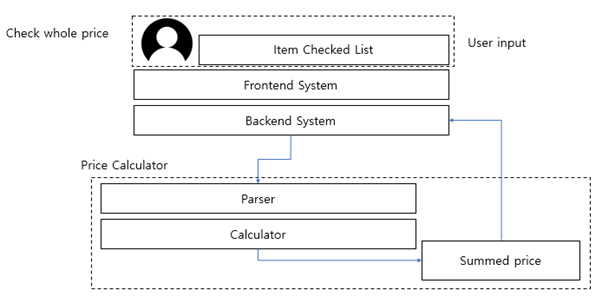
* 1. **Item Deletion System**



[Diagram 5] Item Deletion System Architecture

Delete item Subsystem에 대한 Architecture이다. User가 선택한 저장된 selected item을 input으로 받는다. 이는 Delete item model에서 처리되는데, input을 바탕으로 저장된 item의 list에서 해당되는 item의 identifier을 찾아 삭제할 item을 명시한 information을 backend system에 제공한다.

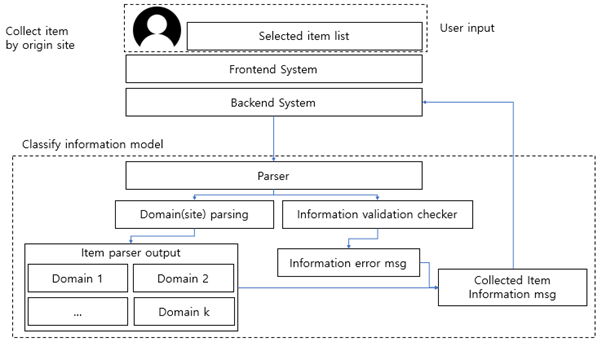
* 1. **Price Checking System**



[Diagram 6] Pricing Checking System Architecture

Check whole price subsystem에 대한 architecture이다. 이는 선택한 Item의 가격의 합을 보여주는 것을 목표로 하며, Item checked list를 input으로 받아 Calculator를 통해 Summed price를 계산해 backend system에 제공한다.

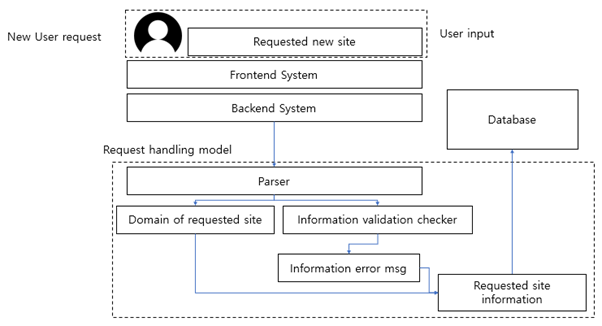
* 1. **Item Classify System**



[Diagram 7] Item Classify System Architecture

Collect item by origin site subsystem에 대한 architecture이다. 이는 선택한 item list에 대해, 각각의 item을 domain에 따라 나누어 UI에 표현하는 것을 목표로 한다. Selected item list를 input으로 활용하여 Domain에 따라 item을 collect한 information을 backend system에 제공한다.

* 1. **User Request System**



[Diagram 8] User Request System Architecture

New user request store subsystem에 대한 architecture이다. 이는 User가 작동하기를 원하는 New site request를 input으로 받아, Parser에 의해 site domain을 얻어 DB에 저장하여 Developer가 Post processing하는 방식으로 이루어진다.

1. **System Requirements Specifications**
   1. **Objective**

System Requirement Specification에서는 기능적 요구사항과 비기능적 요구사항을 구체적으로 서술한다. 이때 본 파트는 시스템 설계 단계와 구현 단계를 포함한 개발 프로세스 전반에서 사용됨으로써 도표 등을 이용하여 체계적으로 설명한다.

* 1. **Functional Requirements**
     1. **Add to Cart Function**

|  |  |
| --- | --- |
| Name | Add to Cart Function |
| Objective | 장바구니에 담고 싶은 제품을 저장 |
| Description | Basket 버튼을 클릭하면 현재 사용자가 보고 있는 웹페이지에서 쇼핑몰의 사이트 이름과, 제품의 사진, 제품의 이름, 제품의 가격을 제품카드로 만들어 PICKET의 장바구니 란에 담을 수 있다. |
| Inputs | 웹 쇼핑몰 사이트에서 파싱 된 데이터 |
| Outputs | Input 데이터가 들어간 제품 카드 |
| Action | 웹 쇼핑몰 사이트별로 개별화 된 파싱 알고리즘에 따라 사이트 이름, 제품 사진, 제품 이름, 제품 가격에 대해서 해당 정보가 들어가 있는 제품 카드를 만들어서 장바구니에 저장한다. |
| Condition | 해당 웹 쇼핑몰 사이트의 쇼핑 품목에 대해서 데이터 파싱이 가능해야 한다. |

[Table 2] Add to Cart Function

* + 1. **Edit Information Function**

|  |  |
| --- | --- |
| Name | Edit Item Information Function |
| Objective | 장바구니에 담은 제품의 정보가 부정확할 때 사용자가 직접 수정 |
| Description | Basket에 담은 제품 카드에서 Edit 버튼을 누르면 해당 제품의 이름, 가격, 사이트 이름, 사진 등의 정보를 수정할 수 있는 입력창이 나타난다. |
| Inputs | 제품의 이름, 가격, 사이트 이름, 사진 |
| Outputs | 업데이트 된 제품의 이름, 가격, 사이트 이름, 사진 |
| Action | 사용자가 입력한 정보를 받아 PICKET에 저장된 제품 정보를 업데이트 한다 |
| Condition | PICKET에 저장 한 제품만 정보 수정이 가능하다 |

[Table 3] Edit Item Information Function

* + 1. **Delete Item Function**

|  |  |
| --- | --- |
| Name | Delete Item Function |
| Objective | 장바구니에 담은 제품을 삭제 |
| Description | 장바구니에 담은 제품카드에서 삭제 버튼을 누르면 장바구니에서 해당 제품을 삭제한다. |
| Inputs | 삭제를 원하는 장바구니 내의 제품 |
| Outputs | 삭제 결과 |
| Action | 사용자가 입력한 정보를 받아 PICKET에 저장된 제품을 삭제한다 |
| Condition | PICKET에 저장 한 제품만 삭제 가능하다 |

[Table 4] Delete Item Function

* + 1. **Total Price Function**

|  |  |
| --- | --- |
| Name | Total Price Function |
| Objective | 장바구니에 담긴 모든 제품들 가격의 합 구하기 |
| Description | 장바구니에 담겨 있는 모든 제품들의 제품 가격들의 총합을 구해서 PICKET의 메인 화면 오른쪽 상단에 총 가격을 표시한다. |
| Inputs | 제품 들의 가격 |
| Outputs | 총 가격 |
| Action | 제품 들의 가격을 모두 합한 값을 총 가격으로 하여 표시한다. |
| Condition | 제품들의 가격이 모두 명시되어 있어야 한다. |

[Table 5] Total Price Function

* + 1. **Group by Web Site Function**

|  |  |
| --- | --- |
| Name | Group by Web Site Function |
| Objective | 장바구니에 담긴 제품을 가져온 웹 사이트 별로 묶어서 보여준다 |
| Description | 장바구니에 담긴 제품을 가져온 웹 사이트 별로 묶고, 해당 웹 사이트에서 가져온 제품을 웹 사이트 하위 목록으로 표시해 준다 |
| Inputs | PICKET에 저장된 모든 제품 |
| Outputs | 웹 사이트 리스트 및 해당 웹 사이트에서 가져온 제품 목록 |
| Action | PICKET에 저장된 모든 제품을 검사하여 같은 웹 사이트 별로 묶어 화면에 표시 해 준다 |
| Condition | PICKET에 저장된 제품이 하나 이상 존재해야 한다 |

[Table 6] Group by Web Site Function

* + 1. **Review Classification Function**

|  |  |
| --- | --- |
| Name | Review Classification Function |
| Objective | 리뷰를 카테고리에 맞게 분류 |
| Description | 해당 제품의 리뷰를 배송, 품질, 가격, 평점 순 등으로 카테고리를 나누어 분류하고, 특정 카테고리의 체크박스에 체크 표시를 하면 해당 카테고리의 내용과 관련된 리뷰가 표시된다. 카테고리의 중복 체크도 가능하다. |
| Inputs | 리뷰 Text, 리뷰의 평점 |
| Outputs | 카테고리 별 리뷰 분류 |
| Action | Text로 된 리뷰와 평점을 input으로 받아서 해당 리뷰를 카테고리로 분류하여 사용자에게 보여준다. |
| Condition | 리뷰가 작성된 언어가 한글 또는 영어여야 한다. |

[Table 7] Review Classification Function

* + 1. **User Request Function**

|  |  |
| --- | --- |
| Name | User Request Function |
| Objective | 사용자가 PICKET에 서비스 요청 |
| Description | 제품 카드가 자동으로 완성되는 기능이 서비스 되지 않는 웹 쇼핑몰 사이트의 경우 PICKET 서비스 제공자에게 직접 해당사이트의 서비스를 요청할 수 있다.  또는, 리뷰를 분류하는 카테고리를 추가하는 요청을 할 수도 있다. |
| Inputs | 사용자의 요청사항 (Text) |
| Outputs | 서비스 대응 |
| Action | 사용자가 PICKET에서 필요한 특정한 기능이나 추가 기능을 Text 형태로 요청할 수 있고 서비스 제공자는 해당 요청에 대응하는 서비스를 제공하도록 노력한다. |
| Condition | PICKET의 사용 목적에 맞는 서비스 요청이어야 한다. |

[Table 8] User Request Function

* 1. **Non-Functional Requirements**

Non-Functional Requirements는 Product Requirements, Organizational Requirements, External Requirements로 나누어진다. 각 비기능적 요구사항들은 더 세부적으로 분류될 수 있는데, 그 중 본 시스템에 필요한 것들을 서술한다.

* + 1. **Product Requirements**

**A.1 Usability Requirements**

본 시스템은 온라인 쇼핑의 이용에 도움을 주는 기능을 제공한다. 본 시스템에서 가장 중요하게 여겨지는 기능은 여러 분산된 쇼핑몰 사이트에 있는 제품들을 하나의 장바구니에 담는 기능이다. 따라서, 클릭을 이용하여 편리하게 장바구니에 제품을 담을 수 있어야 하며, 본 시스템이 제공하는 UI에서 리뷰 분석, 제품의 사이트 별 분류, 총 가격 확인 등이 직관적으로 가능하게 구현이 되어야 한다.

**A.2 Performance Requirements**

본 시스템은 사용자가 임의의 온라인 쇼핑몰 사이트에서 제품을 클릭하여 본 시스템에 담을 때, 제품의 정보가 잘못 담기면 안 된다. 제품의 정보는 HTML tag의 parsing을 통해 이뤄지므로, 여러 쇼핑몰 사이트의 tag들을 hash table 형태로 저장해야 한다. 또한, 본 시스템에 담긴 제품들은 사이트 별로 자동적으로 분류가 되어야 하므로 해당 제품이 어떤 사이트에서 가져왔는지에 관한 정보도 오류 없이 기록이 되어야 한다. 본 시스템은 한국어로 된 모든 쇼핑몰 사이트에서 작동이 가능하게 하는 것이 목표이므로, 만약 등록이 안된 쇼핑몰 사이트가 있는 경우 개발자 측은 해당 사이트에 대한 등록 요청을 사용자로부터 받고 처리를 할 수 있어야 한다.

**A.3 Dependability Requirements**

본 시스템은 제품에 대한 리뷰분석을 서버와 통신을 통해 가져오게 된다. 이 과정에 있어서 해당 제품의 리뷰 전달이 똑같이 이뤄져야 한다. 리뷰 분석이 제대로 이뤄지지 않을 경우 사용자의 신뢰를 잃을 수 있으므로, 구현 단계에 있어서 최대한 많은 리뷰 데이터를 모아 리뷰 분석 시스템 모델의 정확성을 높일 수 있어야 한다.

**A.4 Security Requirements**

본 시스템은 사용자의 개인정보를 필요로 하지 않는다. 사용자가 담는 제품들은 사용자의 개인 브라우저 상에서 저장이 되고 관리된다. 따라서, 본 시스템에 있어서 사용자의 개인정보 유출은 고려하지 않아도 된다.

* + 1. **Organizational Requirements**

**B.1 Environmental Requirements**

본 시스템은 Chrome Extension으로 구현이 되기 때문에 Chrome 브라우저 환경에서만 작동을 한다. 또한, Chrome Extension을 사용함으로써 사용자가 Chrome 브라우저에 로그인을 하면 자동적으로 사용자의 장바구니 정보를 어느 PC에서든지 불러 올 수 가 있다.

**B.2 Operational Requirements**

리뷰 분석기능이 돌아가는 서버는 상시 구동이 되어야 한다. 하지만 현재 지속적으로 운영할 수 있는 하드웨어 자원이 없으므로, 서버는 개발자의 개인컴퓨터에서 작동하는 것을 전제로 한다. 따라서, 제품 저장 기능 등 대부분의 기능들은 사용자의 Chrome Extension에 구현을 한다.

**B.3 Development Requirements**

보다 정확한 리뷰 분석 기능을 제공하기 위해 머신러닝(Machine Learning) 기술을 도입한다. 또한, Chrome Extension을 기반으로 구현이 되기 때문에, 개발 언어는 주로 JavaScript를 사용한다.

* + 1. **External Requirements**

**C.1 Regulatory Requirements**

본 시스템은 Chrome 브라우저 기반으로 Chrome Extension를 이용해 구현이 되므로, 개발자는 Chrome과 Chrome Extension에서 제약하고 있는 사항을 반드시 지켜야 한다.

**C.2 Safety/Security Requirements**

본 시스템은 등록이 되지 않은 쇼핑몰 사이트에 대해 고객의 등록 요청을 서버에서 받는다. 이 과정에서 외부자의 접근을 막을 수 있어야 한다.

* 1. **Scenarios**

Non-Functional Requirements는 Product Requirements, Organizational Requirements, External Requirements로 나누어진다. 각 비기능적 요구사항들은 더 세부적으로 분류될 수 있는데, 그 중 본 시스템에 필요한 것들을 서술한다.

* + 1. **Adding Item into Cart Scenario**

**A.1 Initial Assumption**

사용자가 PICKET 장바구니에 담고자 하는 제품이 존재하는 쇼핑몰 사이트는 사전에 PICKET에서 이용 가능한 사이트에 등록이 되어 있어야 한다.

사용자는 Google 웹 스토어를 이용해 PICKET 확장 프로그램을 설치를 해야 한다. 설치가 완료 되었다면, Google 브라우저 우측 상단에 있는 PICKET 아이콘을 클릭하여 본 서비스를 실행한다.

**A.2 Normal Flow of Events**

사용자는 구입 의사가 있는 제품을 클릭을 하면 자동적으로 제품의 이름, 가격, 사이트, 사진이 PICKET 장바구니에 담기게 된다.

**A.3 What Can Go Wrong**

이용하고 자 하는 쇼핑몰 사이트가 사전에 등록이 안되어 있을 수 있다. 이 경우, 사용자는 고객 요청 버튼을 통해 개발자 측으로 해당 사이트를 등록해달라는 요구를 보낼 수 있다. 또한, 담긴 제품의 사진, 이름, 가격 등이 잘못 담기는 오류가 발생 할 수 있다. 이 경우,

**A.4 Concurrent Activities**

해당 제품을 PICKET 장바구니에 담는 것과 동시에 제품의 리뷰들은 서버 쪽으로 보내지게 되어 리뷰 분석을 하게 된다. PICKET 장바구니에 담긴 제품들의 총 가격의 합 또한 자동적으로 계산이 되어 표시가 되어야 한다.

**A.5 System State on Completion**

해당 제품이 정상적으로 PICKET 장바구니에 담기면, 사용자는 해당 제품의 사진, 이름, 가격, 사이트들을 장바구니에서 확인할 수 있다.

* + 1. **Edit and Delete Item Scenario**

**B.1 Initial Assumption**

사용자가 PICKET 장바구니 제품을 정상적으로 담고 확인이 가능해야 한다.

**B.2 Normal Flow of Events**

사용자가 장바구니에 담긴 제품의 정보를 수정하기를 원하는 경우, 제품 카드의 Edit 버튼을 통해 제품의 이름, 가격, 사이트, 사진이 수정이 가능해야 한다. 사용자가 장바구니에 담긴 제품의 삭제를 원하는 경우, 제품카드의 삭제 버튼을 통해 PICKET에 저장된 제품을 삭제할 수 있어야 한다.

**B.3 What Can Go Wrong**

PICKET 장바구니에 담은 제품만 수정 및 삭제가 가능하다.

**B.4 Concurrent Activities**

제품의 가격 정보가 수정이 된 경우와 제품이 삭제가 된 경우에 있어서 장바구니에 담긴 제품들의 총 가격의 합은 동시에 업데이트가 되어야 한다.

**B.5 System State on Completion**

제품 수정이 이뤄진 경우 수정이 이루어진 항목은 정상적으로 화면에 표시되어야 하며, 제품 삭제가 이뤄진 경우 해당 제품은 PICKET 장바구니에서 더 이상 확인이 불가능 해야 한다.

* + 1. **Grouping Scenarios**

**C.1 Initial Assumption**

사용자가 여러 쇼핑몰 사이트에서 제품들을 PICKET 장바구니에 정상적으로 담은 후, 링크 별 모아보기 버튼을 클릭한다.

**C.2 Normal Flow of Events**

사용자가 링크 별 모아보기 버튼을 클릭하면, PICKET 장바구니에 담긴 제품들은 사이트 별로 분류가 이루어진다. 각 웹 사이트에서 가져온 제품들은 해당 웹 사이트의 하위 목록으로 표시가 된다.

**C.3 What Can Go Wrong**

PICKET 장바구니에 담긴 제품들만 분류가 된다. 만약 동일한 제품을 다른 쇼핑몰 사이트에서 가져온 경우, 해당 제품을 가져온 쇼핑몰의 목록에 표시가 된다.

**C.4 Concurrent Activities**

웹 사이트의 로고 또한 파싱을 해와서 PICKET 링크 별 모아보기 목록에 표시가 되어야 한다.

**C.5 System State on Completion**

사용자는 링크 별로 분류가 된 제품들의 목록을 확인 가능해야 한다.

* + 1. **Review Classification Scenarios**

**D.1 Initial Assumption**

사용자가 여러 쇼핑몰 사이트에서 제품들을 PICKET 장바구니에 정상적으로 담기고, 해당 제품의 리뷰 또한 정상적으로 파싱이 되어야 한다.

**D.2 Normal Flow of Events**

파싱이 된 제품의 리뷰는 서버로 보내진다. 서버에서는 해당 제품의 리뷰를 배송, 품질, 가격, 평점 카테고리로 나누어 분류를 하게 된다. 서버에서 리뷰 분석이 완료가 되면 다시 고객의 브라우저에 보내진다. 고객은 제품 카드 우측 하단의 리뷰 버튼을 통해 분류가 완료된 리뷰를 확인 가능해야 한다.

**D.3 What Can Go Wrong**

리뷰 분석이 제대로 이뤄지지 않을 수 있다. 리뷰 분석의 신뢰도가 높아지기 위해서는 리뷰 분석 모델이 사전에 학습이 잘 되어 있어야 한다. 따라서, 사전에 충분한 리뷰 데이터를 통해 리뷰 분석 모델의 학습을 잘 시켜야 한다.

**D.4 Concurrent Activities**

사용자의 브라우저에서도 해당 제품의 리뷰들이 배송 품질, 가격, 평점 카테고리로 분류되어야 한다. 이 4개 카테고리마다 하나의 체크박스가 존재하며, 해당 카테고리의 체크박스를 클릭하면 클릭한 카테고리의 리뷰들만 확인이 가능해야 한다.

**D.5 System State on Completion**

사용자는 분류가 된 제품의 리뷰를 리뷰 버튼을 통해 확인이 가능해야 한다.

* + 1. **User Request Scenarios**

**E.1 Initial Assumption**

PICKET이 지원이 불가능한 쇼핑몰 사이트에 대해서 사용자는 개발자 측에 해당 쇼핑몰 사이트를 이용할 수 있도록 사이트 등록 요청을 할 수 있다.

**E.2 Normal Flow of Events**

사용자가 고객 요청버튼을 클릭하면 개발자에게 요청 메시지가 전송이 된다. 개발자는 요청 메시지를 보고 해당 쇼핑몰 사이트가 이용 가능하도록 등록을 한다.

**E.3 What Can Go Wrong**

이미 등록이 되어 있는 쇼핑몰 사이트에 대해 등록 요청이 들어올 경우, 해당 요청은 개발자에 의해 폐기가 된다.

**E.4 Concurrent Activities**

사이트 등록 요청 뿐만 아니라 리뷰 카테고리에 대한 추가 요청도 개발자 측에 보낼 수 있다.

**E.5 System State on Completion**

등록이 완료가 된 쇼핑몰 사이트는 다음 업데이트 때부터 이용이 가능해야 한다.

1. **System Models**
   1. **Objective**

System models에서는 본 시스템과 그 컴포넌트(Component), 동작 환경 사이의 관계들을 시각적인 자료로 나타낸다. Context diagram, Process diagram, Use-case diagram 등 다양한 다이어그램을 사용한다.

* 1. **Context Models**
     1. **Context Diagram**

텍스트, 지도이(가) 표시된 사진

자동 생성된 설명

[Diagram 9] Context Diagram

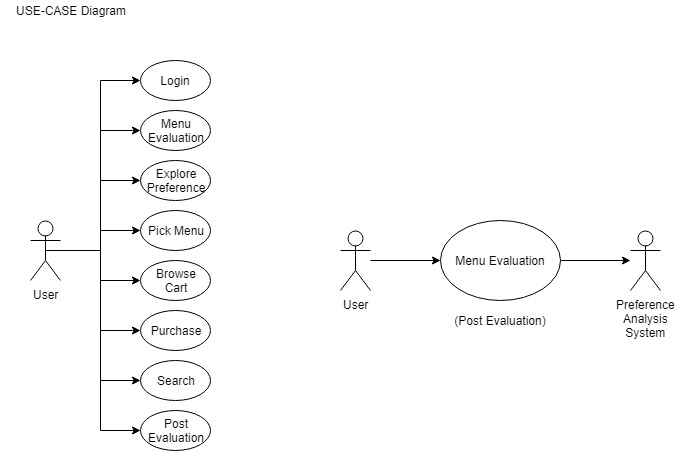
* + 1. **Process Diagram**

스크린샷이(가) 표시된 사진

자동 생성된 설명

[Diagram 10] Process Diagram

* 1. **Interaction Models**
     1. **Use-Case Diagram**



[Diagram 11] Use-case Diagram

* + 1. **Tabular description for each use case**

**B.1 Add to Cart**

|  |  |
| --- | --- |
| actor | User, User Database |
| description | 사용자가 관심 있는 상품을 장바구니에 추가한다. |
| situation | 사용자가 관심 있는 상품을 장바구니에 담는다. |
| system response | 사용자의 데이터베이스에 상품의 이미지 정보와  가격 정보를 사이트로부터 parsing하여 저장한다. |

[Table 9] Add to Cart Use-case

**B.2 Edit Information**

|  |  |
| --- | --- |
| actor | User, User Database |
| description | 사용자가 장바구니에 담은 상품 정보를 수정한다. |
| situation | 상품을 선택하여 사이트 명, 상품명, 상품 가격 등의 정보를 수정할 수 있다. |
| system response | 사용자로부터 특정 상품의 사이트 명, 상품명, 상품 가격 등의 정보를 입력 받을 시 해당 상품의 정보를 그것으로 대체한다. |

[Table 10] Edit Information Use-case

**B.3 Delete Item**

|  |  |
| --- | --- |
| actor | User, User Database |
| description | 사용자가 장바구니에 추가된 상품을 장바구니에서 제거한다. |
| situation | 장바구니의 상품 목록에서 휴지통 버튼을 누른다. |
| system response | 사용자의 데이터베이스에서 해당 상품 정보를 제거한다. |

[Table 11] Delete Item Use-case

**B.4 Total Price**

|  |  |
| --- | --- |
| actor | User, User Database |
| description | 사용자가 장바구니에 등록된 상품들의 총 가격을  볼 수 있다. |
| situation | PICKET의 우측 상단에서 총 가격을 확인한다. |
| system response | 장바구니에 상품이 추가되거나 삭제될 시 상품의 총 가격을 즉각적으로 계산해 우측 상단에 표기한다. |

[Table 12] Total Price Use-case

**B.5 Group by Web Site**

|  |  |
| --- | --- |
| actor | User, User Database |
| description | 장바구니에 담긴 제품들을 사이트 별로 분류한다. |
| situation | 사용자는 사이트 별 분류 버튼을 클릭하면 제품들이 사이트 별로 분류가 된걸 확인 할 수 있다. |
| system response | 분류된 사이트를 클릭하면 해당 사이트로 분류된 제품들의 목록이 나열되어 보여진다. |

[Table 13] Group by Web Site Use-case

**B.6 Review Classification**

|  |  |
| --- | --- |
| actor | User, Developer |
| description | 사용자가 상품에 대한 리뷰 정보를 볼 수 있다. |
| situation | 사용자는 자신이 선택한 상품 목록 중 특정 상품을 선택하면 그에 대해 사용자가 관심 있는 리뷰 정보를 확인할 수 있다. |
| system response | 사용자가 상품 목록에 있는 상품을 선택하면 해당 쇼핑 사이트에서 리뷰를 parsing하여 딥러닝 기법을 이용해 사용자의 관심사를 분석하여 관심 있는 리뷰를 추출하여 사용자에게 보여준다. |

[Table 14] Review Classification Use-case

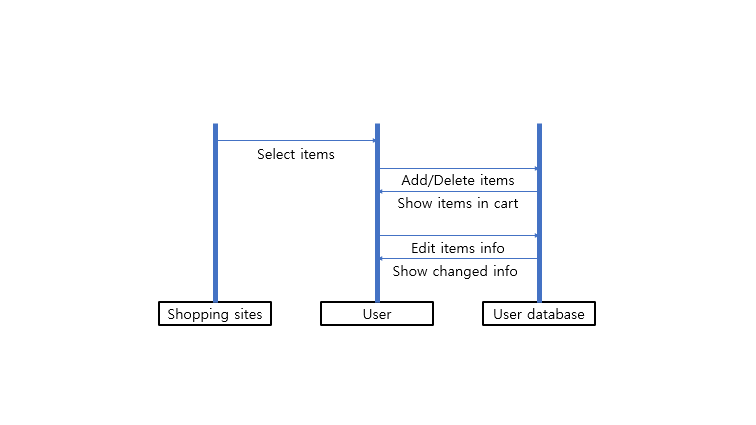
**B.7 User Request**

|  |  |
| --- | --- |
| actor | User, Developer |
| description | 사용자가 PICKET에 원하는 서비스를 요청한다. |
| situation | 사용자가 새로운 사이트에 대한 PICKET서비스를  요청하거나 리뷰 카테고리 분류 등에 대해서  개발자에게 요구사항을 전달한다. |
| system response | 개발자는 사용자의 요구사항을 받아 업데이트를 통해 새로운 사이트에 대한 서비스를 제공하거나 카테고리 분류 업데이트를 한다. |

[Table 15] User Request Use-case

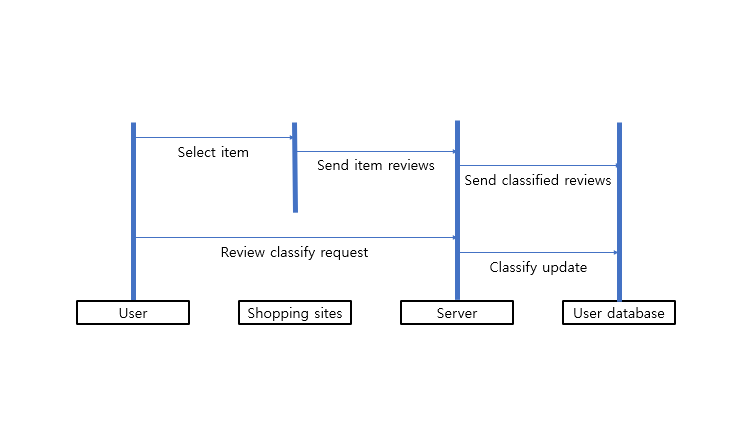
* + 1. **Sequence Diagram**

**C.1 Shopping Cart Sequence Diagram**



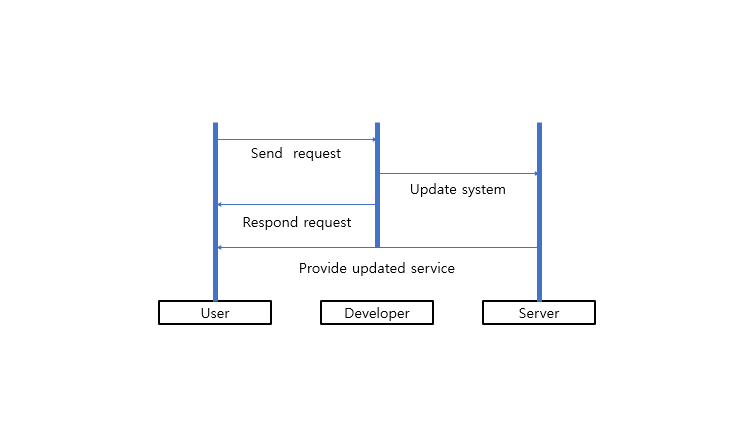
[Diagram 12] Shopping Cart Sequence Diagram

**C.2 Review Sequence Diagram**



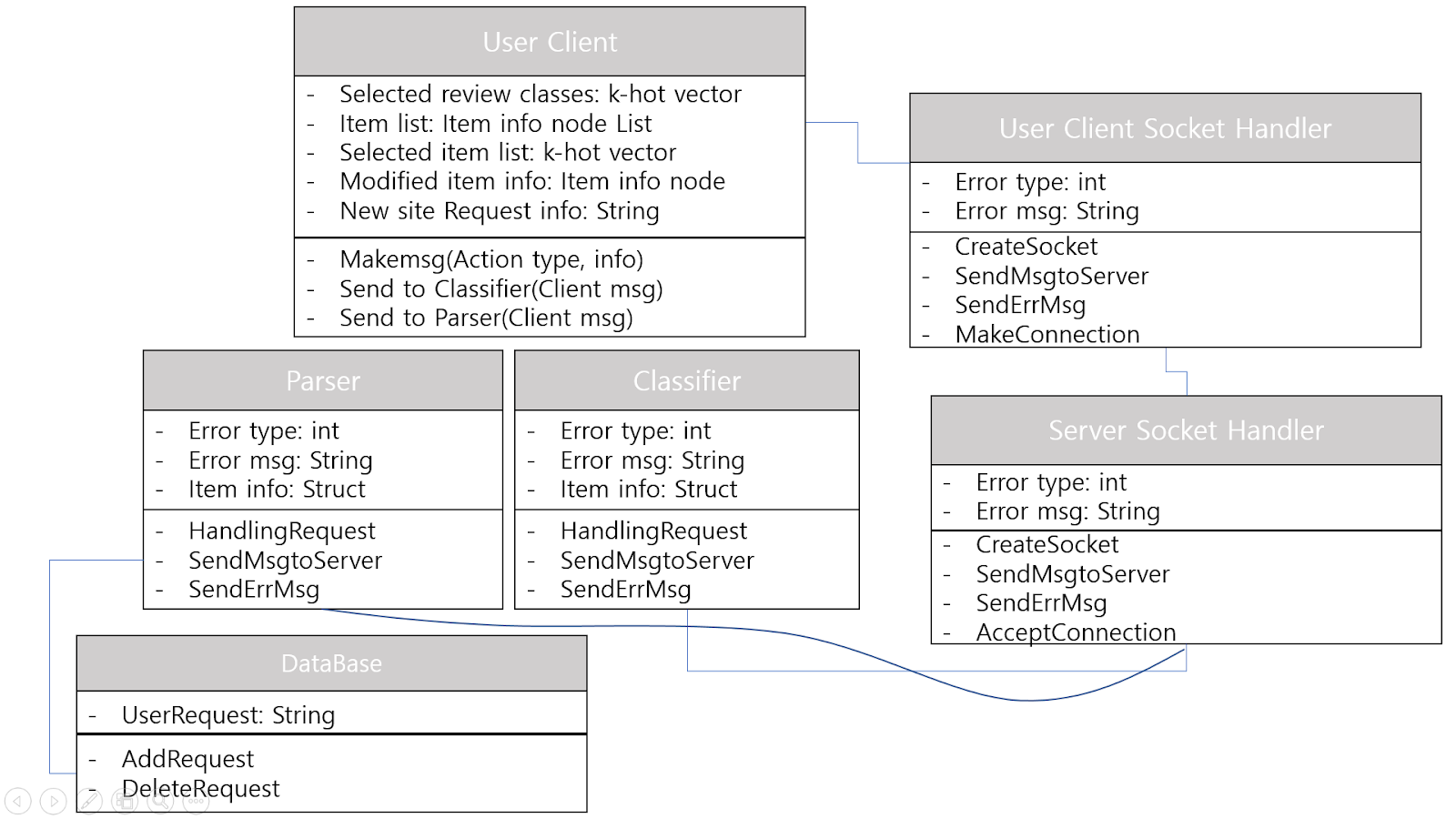
[Diagram 13] Review Sequence Diagram

**C.3 User Request Sequence Diagram**



[Diagram 14] User Request Sequence Diagram

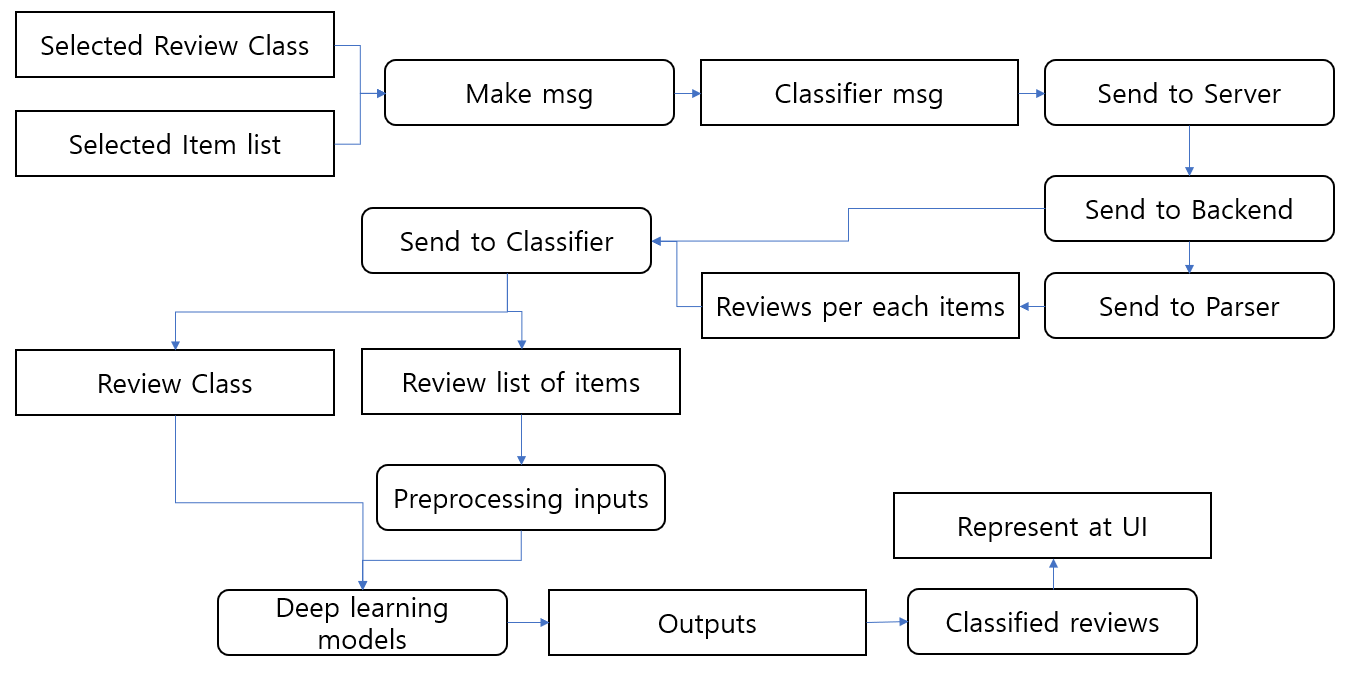
* 1. **Structural Models**
     1. **Class Diagram**



[Diagram 15] Class Diagram

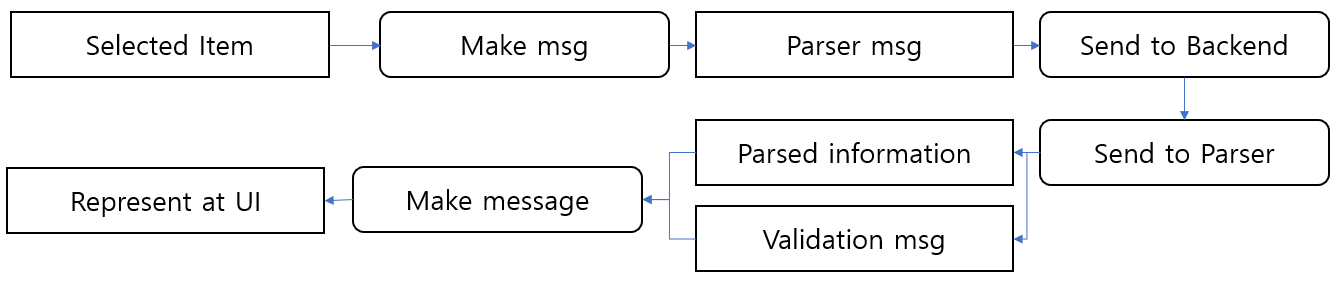
* 1. **Behavioral Models**
     1. **Activity Diagram**

**A.1 Review**



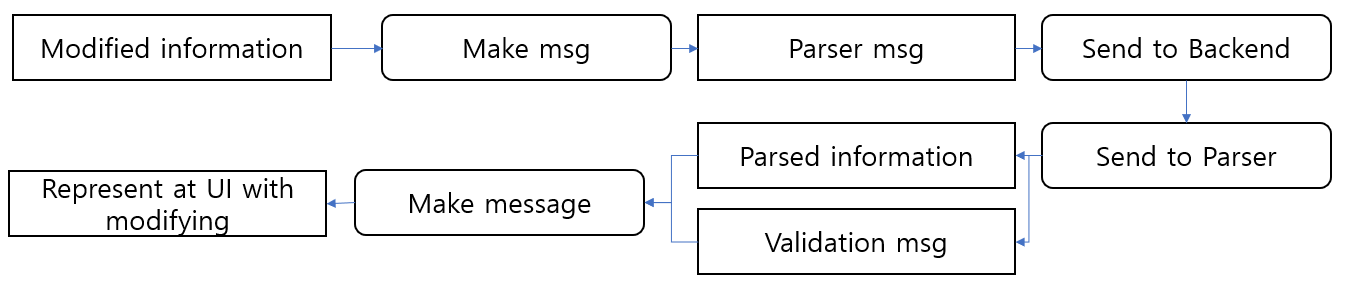
[Diagram 16] Review Activity Diagram

**A.2 Item Storing**



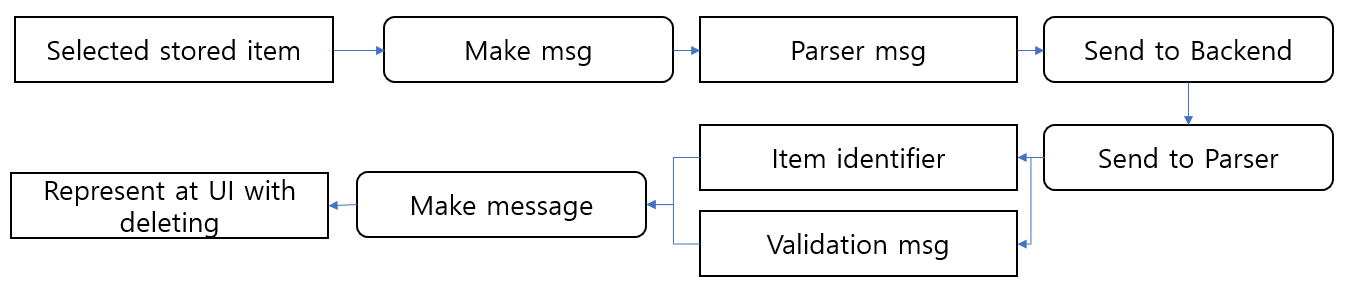
[Diagram 17] Item Storing Activity Diagram

**A.3 Item Modify**



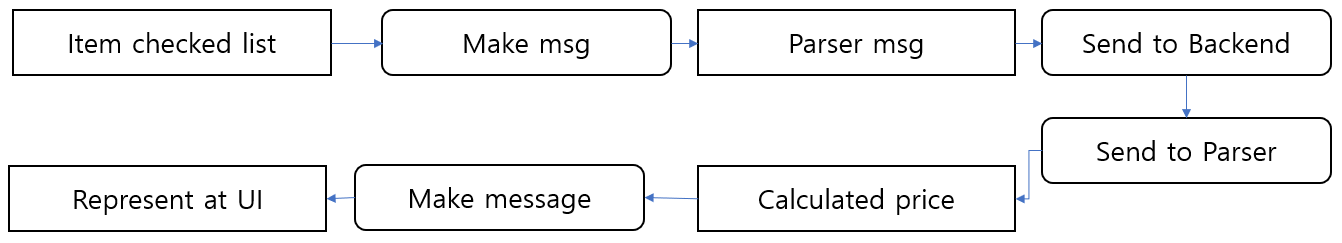
[Diagram 18] Item Modify Activity Diagram

**A.4 Item Deletion**



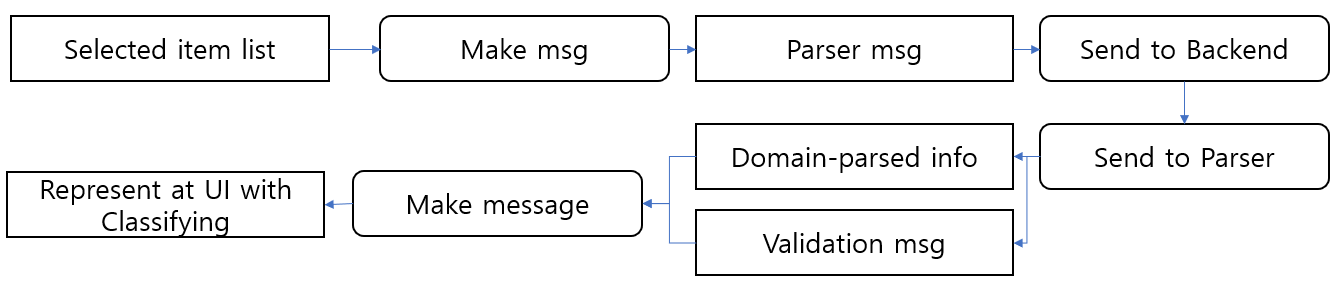
[Diagram 19] Item Deletion Activity Diagram

**A.5 Price Calculate**



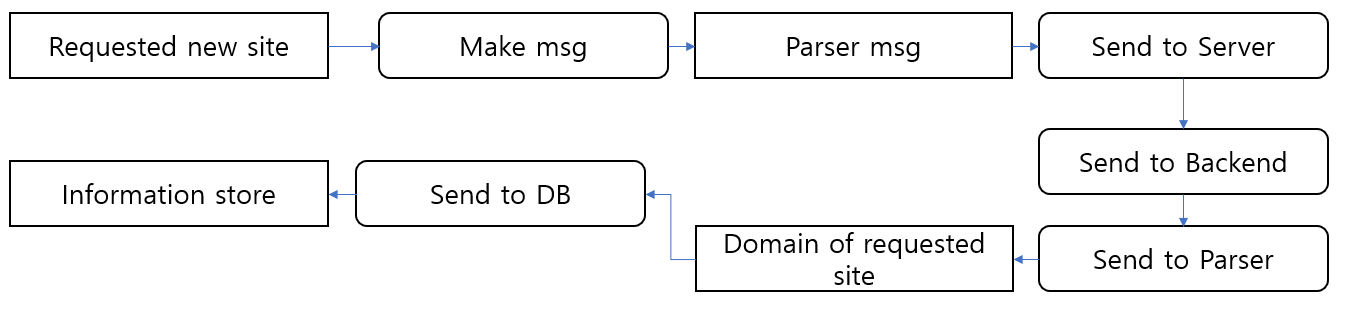
[Diagram 20] Price Calculate Activity Diagram

**A.6 Item Classify**



[Diagram 21] Item Classify Activity Diagram

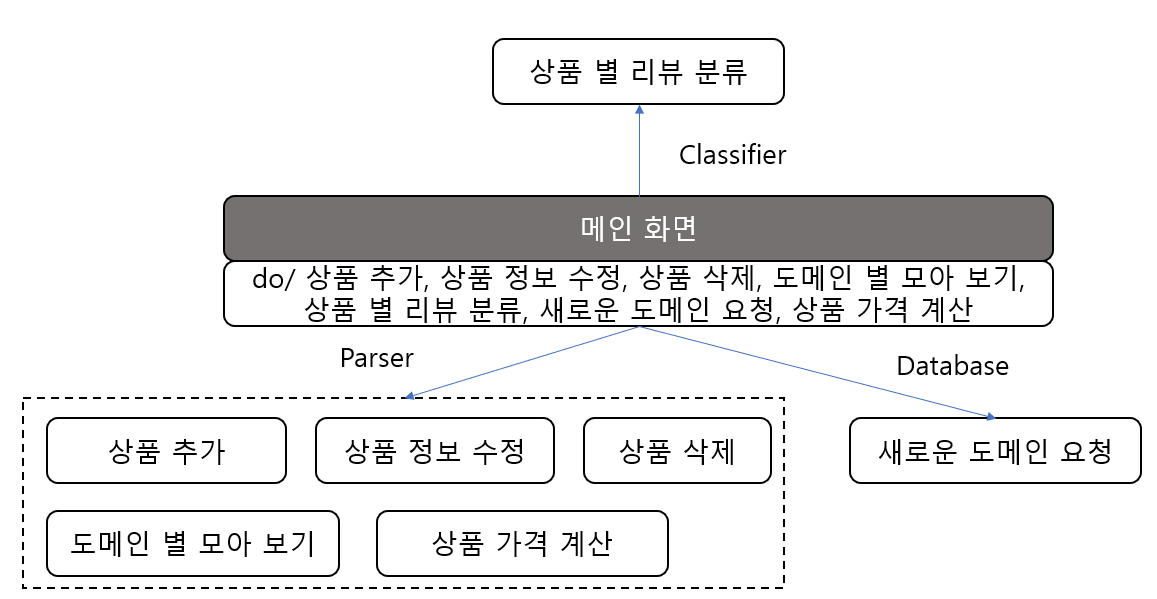
**A.7 Request Handling**



[Diagram 22] Request Handling Activity Diagram

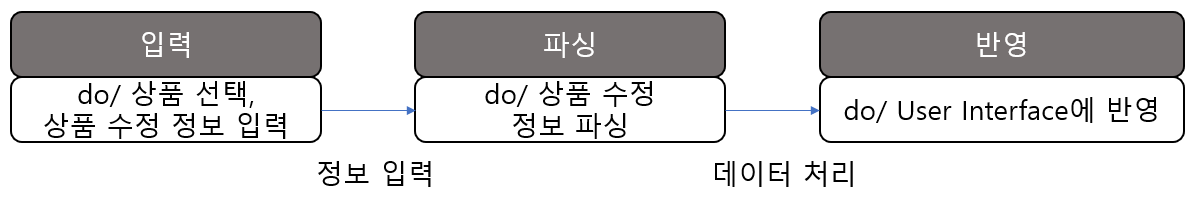
* + 1. **State Diagram**

**B.1 Main Screen**



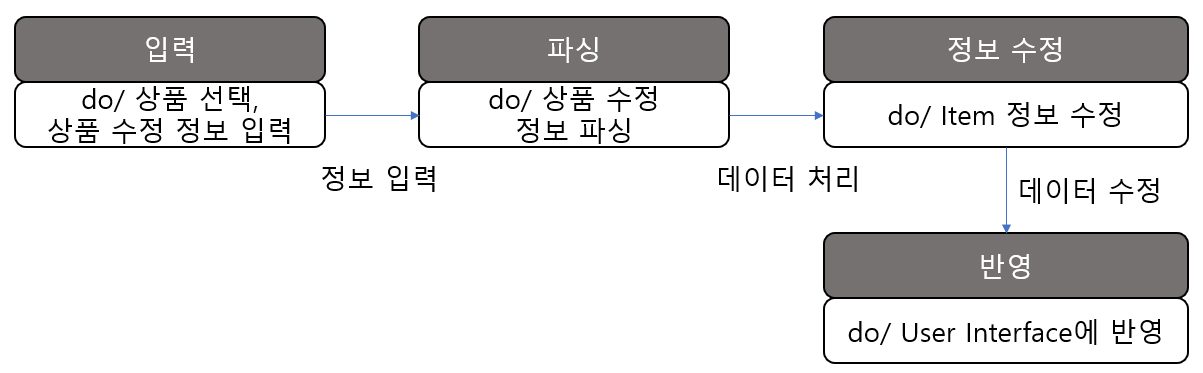
[Diagram 23] Main Screen State Diagram

**B.2 Item Addition**



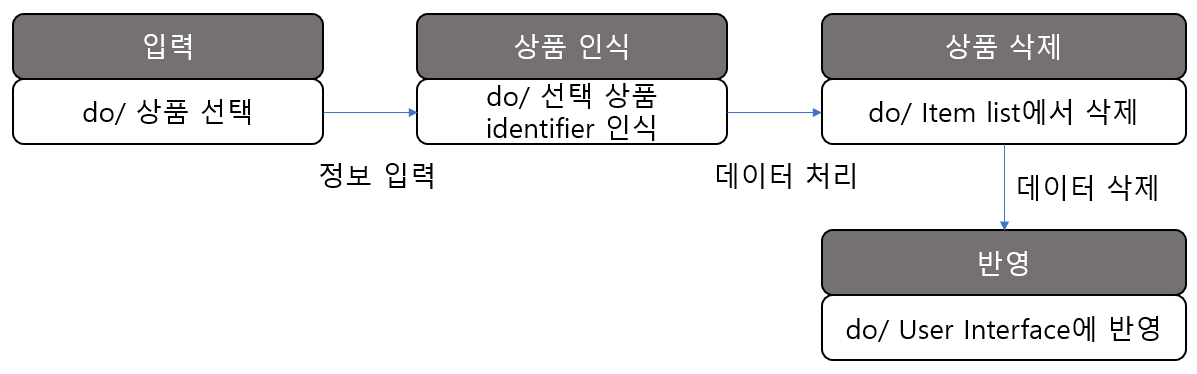
[Diagram 24] Item Addition State Diagram

**B.3 Item Edit**



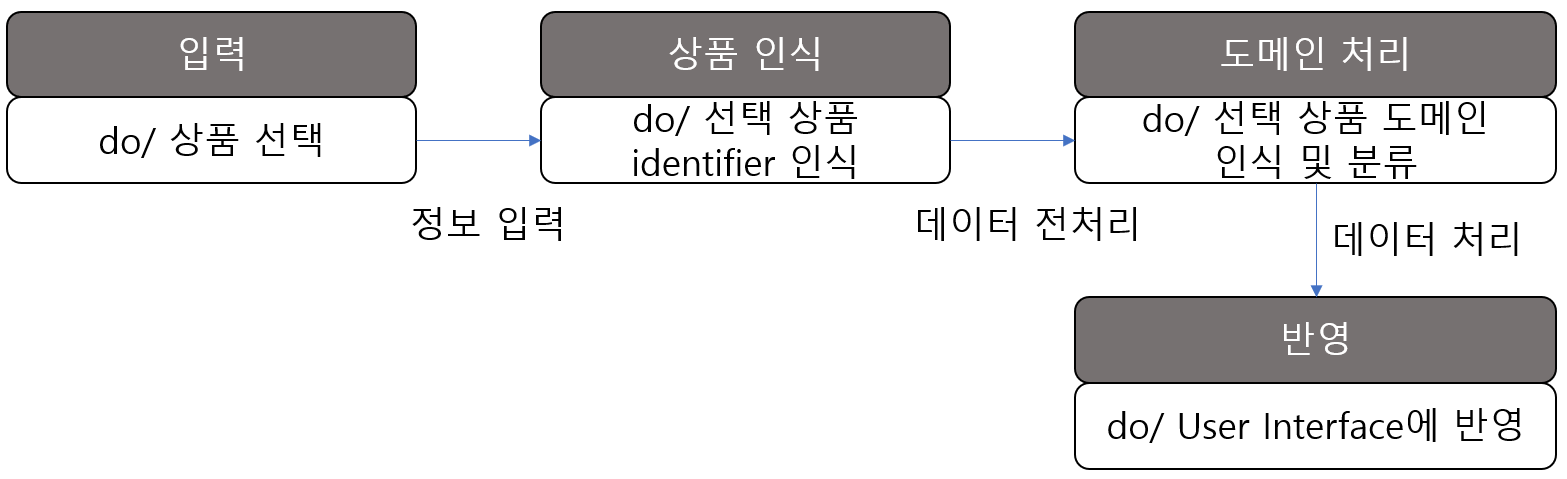
[Diagram 25] Item Edit State Diagram

**B.4 Item Deletion**



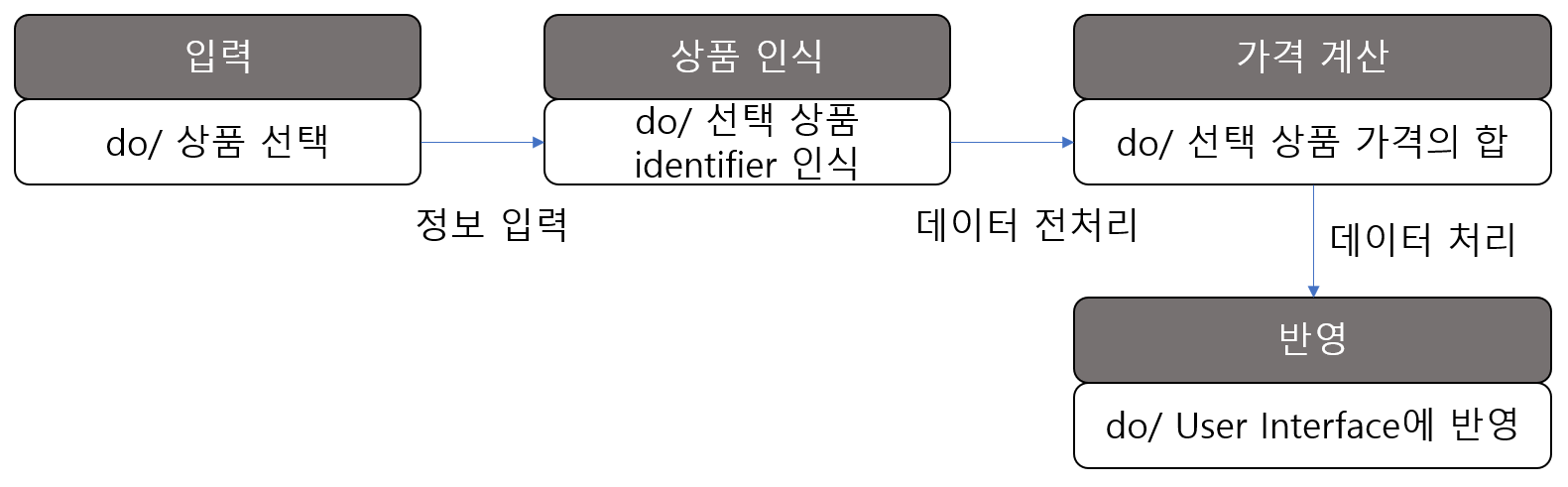
[Diagram 26] Item Deletion State Diagram

**B.5 Item Classify**



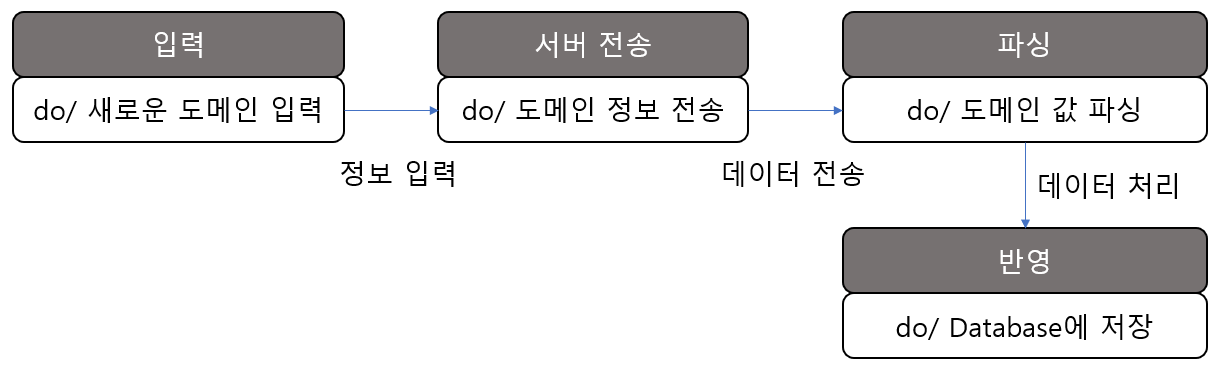
[Diagram 27] Item Classify State Diagram

**B.6 Price Calculation**



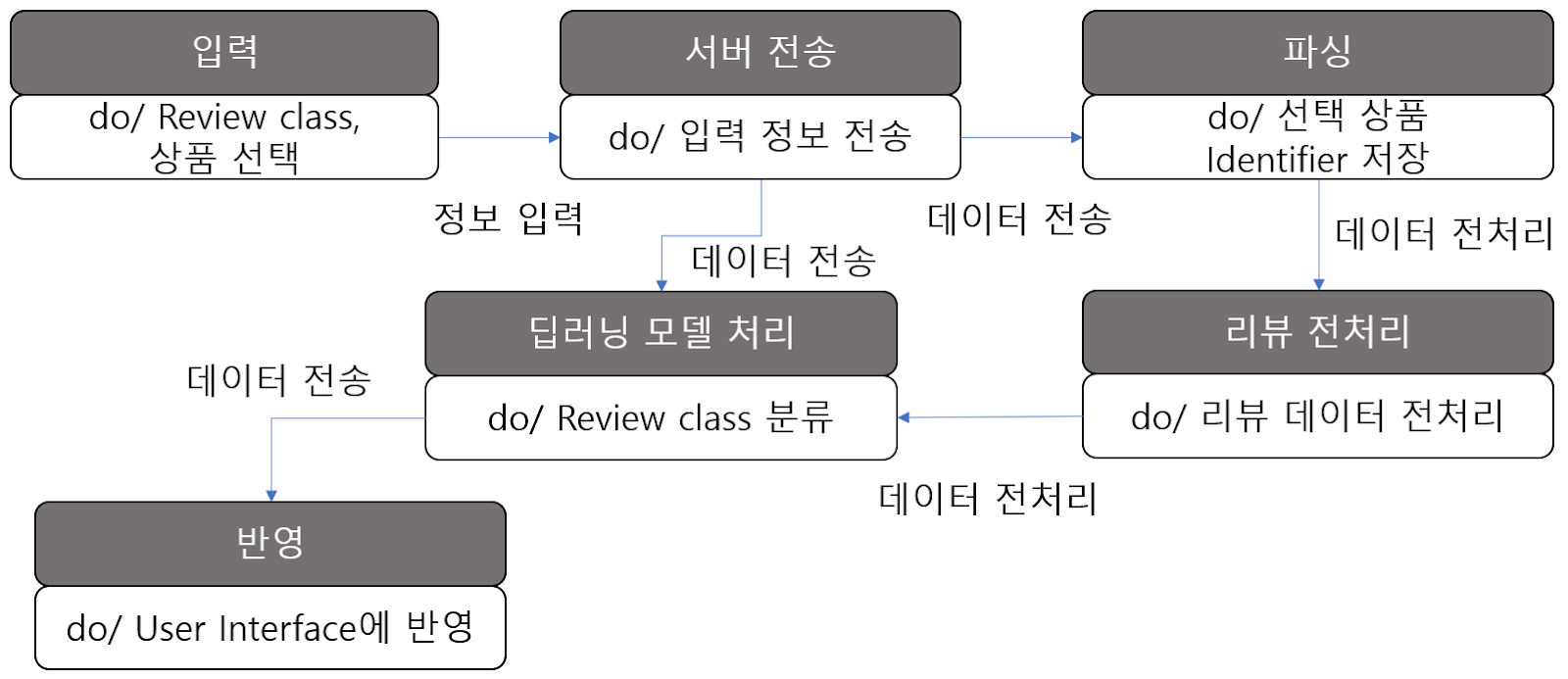
[Diagram 28] Price Calculation State Diagram

**B.7 Domain Addition**



[Diagram 29] Domain Addition State Diagram

**B.8 Review Classify**



[Diagram 30] Review Classify State Diagram

1. **System Evolution**
   1. **Objective**

System evolution에서는 본 시스템이 취하고 있는 기본적인 가정들을 서술한다. 또한, 향후 본 시스템에 발생할 수 있는 예상 가능한 모든 변경사항과 그에 대한 대응 방안에 대해서 서술한다. 변경사항에는 하드웨어의 발전, 사용자 요구사항의 변화 등이 포함된다.

* 1. **Limitation**
     1. **Chrome Extension**

본 시스템은 chrome extension 환경에 기반을 두고 있기 때문에 사용자는 chrome을 반드시 설치해야 하고, chrome extension 사용법을 알아야만 서비스를 이용 가능하다.

* + 1. **Shopping Mall Sites**

본 시스템은 기존 사이트들에 한하여 서비스를 제공하기 때문에 일부 사이트에서 서비스 이용이 제한될 수 있다.

* + 1. **Payment**

본 시스템은 장바구니에 다양한 쇼핑 사이트의 아이템들을 담을 수는 있지만 결제 시 직접 해당 사이트를 방문하여 결제해야 한다.

* 1. **Evolution**
     1. **Chrome Extension**

사용자들은 본 시스템을 사용하기 위해 chrome을 이용해야만 한다. 향후 Internet Explorer나 스마트 폰 등에서도 본 시스템을 이용할 수 있도록 하여 서비스 이용 환경을 확장하도록 한다.

* + 1. **Shopping Mall Sites**

사용자들이 많이 사용하는 사이트나 사용자들이 서비스를 원하는 사이트를 향후 업데이트를 통해 해당 사이트들을 추가한다. 업데이트를 거듭하여 최대한 많은 사이트에서 서비스를 이용할 수 있도록 한다.

* + 1. **Payment**

향후에 장바구니에 담은 쇼핑 아이템들을 통합 결제할 수 있는 서비스를 제공할 수 있도록 한다.

1. **Appendices** 
   1. **Objective**

Appendices에서는 본 문서에서 생략된 하드웨어와 데이터베이스 요구사항에 대해 서술한다.

* 1. **Database Requirements**

사용자가 크롬 창을 종료한 뒤에도 관심 상품 목록을 유지해야 하므로 상품목록을 저장할 수 있는 공간이 필요하다. 또한, 사용자가 관심 있는 상품 리뷰들 또한 사용자의 데이터베이스에 저장되어야 한다.

* 1. **Hardware Requirements**

본 시스템은 사용자의 관심사에 따른 리뷰들을 딥러닝 기법을 통해 분석하는 서비스를 제공한다. 하지만 서버에서 딥러닝을 하기 때문에 사용자의 하드웨어는 요구되지 않는다. 서버에서 정보를 받기 위한 서버와의 연결 정도는 요구된다.

1. **Index** 
   1. **Objective**

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

* 1. **Table Index**

[[Table 1] Version History 9](#_Toc23095444)

[[Table 2] Add to Cart Function 33](#_Toc23095445)

[[Table 3] Edit Item Information Function 34](#_Toc23095446)

[[Table 4] Delete Item Function 34](#_Toc23095447)

[[Table 5] Total Price Function 35](#_Toc23095448)

[[Table 6] Group by Web Site Function 35](#_Toc23095449)

[[Table 7] Review Classification Function 36](#_Toc23095450)

[[Table 8] User Request Function 36](#_Toc23095451)

[[Table 9] Add to Cart Use-case 45](#_Toc23095452)

[[Table 10] Edit Information Use-case 45](#_Toc23095453)

[[Table 11] Delete Item Use-case 45](#_Toc23095454)

[[Table 12] Total Price Use-case 46](#_Toc23095455)

[[Table 13] Group by Web Site Use-case 46](#_Toc23095456)

[[Table 14] Review Classification Use-case 46](#_Toc23095457)

[[Table 15] User Request Use-case 47](#_Toc23095458)

* 1. **Figure Index**

[[Figure 1] 2019년 1월 온라인쇼핑의 동향 10](#_Toc23095434)

[[Figure 2] 소셜커머스 쇼핑몰 11](#_Toc23095435)

[[Figure 3] 자체 의류 쇼핑몰 11](#_Toc23095436)

[[Figure 4] 구매에 영향을 끼치는 정보원 조사 13](#_Toc23095437)

[[Figure 5] 상품의 품질 불만에 대한 리뷰 예시 14](#_Toc23095438)

[[Figure 6] 다양한 웹 사이트 장바구니 예시 15](#_Toc23095439)

[[Figure 7] PICEKT 장바구니 화면 16](#_Toc23095440)

[[Figure 8] 리뷰 분류 기능 모식도 16](#_Toc23095441)

[[Figure 9] 화면 구성 1 20](#_Toc23095442)

[[Figure 10] 화면 구성 2 22](#_Toc23095443)

* 1. **Diagram Index**

[[Diagram 1] Overall System Architecture 25](#_Toc23095404)

[[Diagram 2] Review Classify System Architecture 26](#_Toc23095405)

[[Diagram 3] Item Storing System Architecture 27](#_Toc23095406)

[[Diagram 4] Item Modify System Architecture 28](#_Toc23095407)

[[Diagram 5] Item Deletion System Architecture 29](#_Toc23095408)

[[Diagram 6] Pricing Checking System Architecture 30](#_Toc23095409)

[[Diagram 7] Item Classify System Architecture 31](#_Toc23095410)

[[Diagram 8] User Request System Architecture 32](#_Toc23095411)

[[Diagram 9] Context Diagram 43](#_Toc23095412)

[[Diagram 10] Process Diagram 44](#_Toc23095413)

[[Diagram 11] Use-case Diagram 44](#_Toc23095414)

[[Diagram 12] Shopping Cart Sequence Diagram 47](#_Toc23095415)

[[Diagram 13] Review Sequence Diagram 48](#_Toc23095416)

[[Diagram 14] User Request Sequence Diagram 48](#_Toc23095417)

[[Diagram 15] Class Diagram 49](#_Toc23095418)

[[Diagram 16] Review Activity Diagram 49](#_Toc23095419)

[[Diagram 17] Item Storing Activity Diagram 50](#_Toc23095420)

[[Diagram 18] Item Modify Activity Diagram 50](#_Toc23095421)

[[Diagram 19] Item Deletion Activity Diagram 50](#_Toc23095422)

[[Diagram 20] Price Calculate Activity Diagram 50](#_Toc23095423)

[[Diagram 21] Item Classify Activity Diagram 51](#_Toc23095424)

[[Diagram 22] Request Handling Activity Diagram 51](#_Toc23095425)

[[Diagram 23] Main Screen State Diagram 51](#_Toc23095426)

[[Diagram 24] Item Addition State Diagram 52](#_Toc23095427)

[[Diagram 25] Item Edit State Diagram 52](#_Toc23095428)

[[Diagram 26] Item Deletion State Diagram 52](#_Toc23095429)

[[Diagram 27] Item Classify State Diagram 53](#_Toc23095430)

[[Diagram 28] Price Calculation State Diagram 53](#_Toc23095431)

[[Diagram 29] Domain Addition State Diagram 53](#_Toc23095432)

[[Diagram 30] Review Classify State Diagram 54](#_Toc23095433)