REVIEW REVOLUTION

requirement specification



|  |  |
| --- | --- |
| Student Number | Name |
| 2017314380 | 윤성경 |
| 2014311058 | 김진태 |
| 2014314650 | 유종현 |
| 2013310608 | 정창호 |

Contents(마지막에 변경예정)

[1. Preface 7](#_Toc10827262)

[1.1. Readership 7](#_Toc10827263)

[A. User Requirements Readership 7](#_Toc10827264)

[B. System Requirements Readership 7](#_Toc10827265)

[1.2. Document Structure 8](#_Toc10827266)

[A. Introduction 8](#_Toc10827267)

[B. Glossary 8](#_Toc10827268)

[C. User Requirements Definition 8](#_Toc10827269)

[D. System Architecture 8](#_Toc10827270)

[E. System Requirements Specification 8](#_Toc10827271)

[F. System Models 8](#_Toc10827272)

[G. System Evolution 9](#_Toc10827273)

[H. Appendices 9](#_Toc10827274)

[I. Index 9](#_Toc10827275)

[2. Introduction 10](#_Toc10827276)

[2.1. Needs 10](#_Toc10827277)

[2.2. System Overview 13](#_Toc10827278)

[2.3. Expected Effects 15](#_Toc10827279)

[3. Glossary 16](#_Toc10827280)

[4. User Requirements Definition 17](#_Toc10827281)

[4.1. Functional Requirements 17](#_Toc10827282)

[A. Sign up/Login 17](#_Toc10827283)

[B. Ranking 18](#_Toc10827284)

[C. item detail 19](#_Toc10827285)

[D. Recommendation 20](#_Toc10827286)

[E. Mypage 20](#_Toc10827287)

[F. Search 20](#_Toc10827288)

[4.2. Non-functional Requirements 21](#_Toc10827289)

[A. Product requirements 21](#_Toc10827290)

[B. Organizational requirements 22](#_Toc10827291)

[C. External requirements 23](#_Toc10827292)

[5. System Architecture 24](#_Toc10827293)

[5.1. Frontend Architecture 24](#_Toc10827294)

[5.2. Backend Architecture 25](#_Toc10827295)

[5.3. Review Analysis System 26](#_Toc10827296)

[5.4. Recommendation System 27](#_Toc10827297)

[5.5. Ranking System 28](#_Toc10827298)

[6. System Requirements Specification 29](#_Toc10827299)

[6.1. Functional Requirements – Frontend 29](#_Toc10827300)

[A. Ranking 29](#_Toc10827301)

[B. Show Item Detail 30](#_Toc10827302)

[C. Add Item to Bookmark 30](#_Toc10827303)

[D. Show Recommendation List 31](#_Toc10827304)

[E. Add Recommendation category 32](#_Toc10827305)

[Add Recommendation category Function 32](#_Toc10827306)

[F. Mypage 33](#_Toc10827307)

[6.2. Functional Requirements – Backend 34](#_Toc10827308)

[A. Login 34](#_Toc10827309)

[B. Review Collecting System 35](#_Toc10827310)

[C. Review Analysis System 35](#_Toc10827311)

[D. Review Database Updater 36](#_Toc10827312)

[6.3. Non-functional Requirements 37](#_Toc10827313)

[A. Product requirements 37](#_Toc10827314)

[B. Organization requirements 38](#_Toc10827315)

[C. External requirements 39](#_Toc10827316)

[6.4. Scenario Examples 40](#_Toc10827317)

[A. Login Scenario 40](#_Toc10827318)

[B. Recommendation Scenario 41](#_Toc10827319)

[7. System Models 42](#_Toc10827320)

[7.1. Context models 42](#_Toc10827321)

[A. Context Diagram 42](#_Toc10827322)

[B. Process Diagram 43](#_Toc10827323)

[7.2. Interaction models 43](#_Toc10827324)

[A. Use case Diagram 43](#_Toc10827325)

[B. Tabular Description of Use case Diagram 44](#_Toc10827326)

[7.3. Behavioral models 48](#_Toc10827327)

[A. Ranking System DFD 48](#_Toc10827328)

[B. Review Analysis DFD 48](#_Toc10827329)

[C. MyPage Sequence Diagram. 49](#_Toc10827330)

[8. System Evolution 50](#_Toc10827331)

[8.1. Compare Feature 50](#_Toc10827332)

[8.2. Voice Analysis 50](#_Toc10827333)

[8.3. Sophisticated Sentiment Search Technique 51](#_Toc10827334)

[8.4. OAuth implementation 51](#_Toc10827335)

[9. Appendices 52](#_Toc10827336)

[9.1. Hardware requirements 52](#_Toc10827337)

[9.2. Database requirements 53](#_Toc10827338)

[A. User 53](#_Toc10827339)

[B. Item 53](#_Toc10827340)

[C. Review 53](#_Toc10827341)

[D. Authority 53](#_Toc10827342)

[E. RecommendCategory 53](#_Toc10827343)

[F. Bookmark 53](#_Toc10827344)

[G. Reference 53](#_Toc10827345)

[H. Keyword 53](#_Toc10827346)

[10. Index 54](#_Toc10827347)

[10.1. Tables 54](#_Toc10827348)

[10.2. Figures 54](#_Toc10827349)

[10.3. Diagrams 55](#_Toc10827350)

[11. References 56](#_Toc10827351)

# Preface(우리 팀의 정의로 대체)

이 장에서는 예상 독자를 정의하고, 각 장의 내용을 요약한다. 또한 현재 version과 이전 version의 차이점에 대해 설명한다. 하지만 본 문서는 초기 버전이기 때문에 이 부분을 생략한다.

## Readership

본 문서는 다양한 독자에게 읽힐 것이라고 상정하고 있다. 이 파트에서는 우리가 서술하는 user requirements와 system requirements가 어떤 독자층을 상정하고 있는지를 설명한다.

### User Requirements Readership

User requirements의 예상 독자는 이 시스템의 사용자, 즉 end user이다. 해당 부분은 기술에 대해 잘 모르는 사용자의 관점에서 보아도 이해하기 쉽게 서술되어야 한다. 이를 위해 전문 용어의 사용을 최대한 자제하고, 이해하기 쉬운 자연어로 서술하고, diagram 등의 시각 자료를 많이 활용해 서술한다.

### System Requirements Readership

System requirements의 예상 독자는 시스템 개발자, contractor, client 등 전문성을 가지고 있는 독자들이다. 이를 위해 구조화된 언어로 표현해 더 명확하고 confusion이 없게 표현해야 한다. 또한 user requirements보다 더 자세하고 기술적으로 서술되어야 한다.

## Document Structure

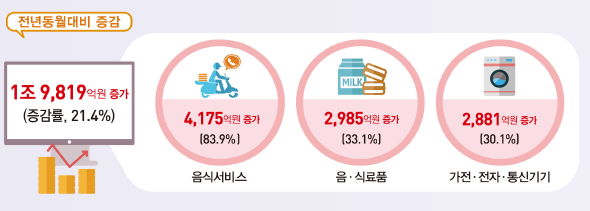
1. Introduction  
   개발하고자 하는 시스템이 왜 필요한지에 대한 배경과 필요성에 대해 설명한다. 또한 시스템 기능에 대한 간략한 설명과, 해당 시스템을 개발함으로써 얻을 수 있는 기대 효과에 대해서 설명한다.
2. Glossary  
   문서 전반에 걸쳐 사용되고 있는 전문 용어에 대한 설명을 하고, 해당 용어가 어떤 맥락에서 어떤 의미로 사용되었는지에 대해 설명한다. 본 문서는 전문성을 가지고 있는 독자들뿐만 아니라 사용자, 이해당사자 등의 비전문가인 독자 또한 대상으로 하기 때문에 가능한 자세하게 전문 용어에 대해 설명한다.
3. User Requirements Definition  
   기본적으로 사용자를 위해 제공되는 시스템의 기능적, 비기능적 요구사항을 사용자의 입장에서 설명한다. 따라서 비전문가도 이해할 수 있도록 자연어를 사용해야 하고, diagram 등의 시각 자료를 통해 이해를 도울 수 있도록 서술해야 한다.
4. System Architecture  
   시스템 구조에 대해 high-level overview가 제시된다. 이때 각 subsystem이 어떤 관계를 가지고 있는지를 같이 설명해야 하며, 만약 재사용되는 components가 있다면 강조해 주어야 한다.
5. System Requirements Specification  
   User Requirements Definition에 간략히 서술되어 있는 요구사항을 기반으로, 기능적 요구사항, 비기능적 요구사항, 기타 요구사항을 더 detail하게 설명한다. 필요하다면 user requirements에 없는 요구사항을 추가할 수 있다. 이때 본 장은 개발 프로세스 전반에 사용되는 장이므로 structured natural language, graphical notations등을 활용해 체계적이고 명확하게 서술한다.
6. System Models  
   graphical하게 system model에 대해 설명한다. 이때 system model에 대한 설명에는 components들 간의 관계, 시스템을 둘러싼 외부 환경과의 관계 등을 전부 포함한다.
7. System Evolution  
   시스템의 기반이 되는 fundamental한 가정에 대해 서술하고, 시스템 운영 과정에서 발생할 수 있는 하드웨어의 변화, client의 요구사항 변화 등에 대해 예측하고 그에 대한 해결 방안을 설명한다.
8. Appendices  
   이 장에는 하드웨어 요구사항, 데이터베이스 요구사항, 개발환경 요구사항 등 개발 중인 시스템과 관련된 항목의 상세하고 구체적인 정보가 서술된다.
9. Index  
   본 문서에서 사용된 그림, 표, 다이어그램 등의 색인을 기술한다.

# Introduction

이 장에서는 본 시스템이 등장하게 된 배경과 필요성에 대해 설명하고, 시스템의 대략적인 기능에 대해 설명한다. 또한 해당 시스템을 개발함으로써 기대되는 효과에 대해 설명한다

## 2.1. Background & Needs

온라인 쇼핑은 총 거래 규모와 거래의 증가 폭으로 볼 때 월 단위로 11조원 가량, 전년도 동월 대비 20% 가량 늘어난 추세를 보여주며(2019년 8월 기준) 매우 큰 시장 규모와 시장의 가능성을 보여주고 있다. 또한 쿠팡, 11번가, 위메프, G마켓, 인터파크를 포함한 여러 인터넷 쇼핑몰들이 활발하게 경쟁하고 있고 이런 상황이 장기전으로 넘어가는 모습을 통해 쇼핑몰 입장에서는 차별화된 전략을 통해서 소비자들이 만족할 수 있는 서비스를 제공해야 한다는 필요성이 더욱 커지고 있다.



**그림 1 온라인쇼핑 거래액 전년동월대비 증감률**

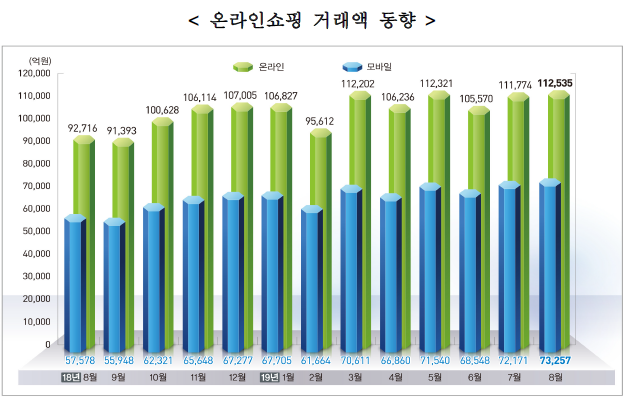


그림 2 온라인쇼핑 거래액 동향

한편 소비자 입장에서는 이와 같은 큰 규모의 온라인 시장에서 다양한 상품들을 비교하고 선택해야 하는 부담이 따르게 된다. 선택의 다양성이 너무 많기 때문에 쉽게 올바른 선택을 내리기가 쉽지 않다. 여기서 소비자들이 상품을 파악할 때 활용할 수 있는 정보는 판매자가 올린 상품 정보와 고객들이 남겨둔 리뷰 정보이다. 실제로 사용해보고 장단점을 파악할 수 없는 온라인 쇼핑의 특성상 상품의 가치를 파악하는데 가장 큰 영향을 미치는 건 믿을 수 있는 다른 소비자의 리뷰이다.

왜 소비자에게 상품 정보보다 리뷰 정보가 더 큰 영향을 미칠까? 물론 상품 정보는 구체적이고 세세한 정보까지 담고 있다는 장점이 있지만, 여러 상품들을 비교하는 경우 한번에 주어지는 정보의 양이 지나치게 많아져서 소비자에게 혼란을 줄 수 있다. 이런 경우 많은 선택에 혼란스러워진 소비자가 군중심리에 기대서 많은 사람들이 좋은 평가를 내린 상품에 자연스럽게 더 호감을 주는 간편한 선택을 하려는 현상도 충분히 발생할 수 있다. 또한 상품 정보는 회사에서 더 만들어서 제공하는 것이기 때문에 100% 신뢰할 수 없지만 리뷰는 실제 사용자들의 경험들로 이루어져 있기 때문에 더 신뢰할 수 있는 정보이다. 실제로 약 90%의 소비자들은 상품을 구매하기 전에 리뷰를 읽어본다는 통계가 있을 뿐 아니라 소비자들은 긍정적인 리뷰가 많은 상품에 대해 31프로까지 추가 지출 의사가 있다는 통계도 존재한다.

이처럼 리뷰는 소비자 관점에서나 판매자 관점에서나 판매에 큰 영향을 미치는 요소이다. 이에 따라 인터넷 쇼핑몰들은 소비자들이 리뷰를 통해 쉽게 정보를 얻을 수 있도록 여러 편의 장치들을 해 두었다. 키워드 검색, 별점 평가 제도, 리뷰 추천 제도, 긍정 부정 표시 기능 등의 편리한 기능들을 제공하고 이를 통해서 소비자들은 리뷰 정보를 좀 더 쉽게 파악할 수 있었다. 하지만 인터넷 쇼핑몰들이 빠뜨린 아주 중요한 기능이 있다. 현재까지 쿠팡, 위메프, 인터파크, 아마존, 네이버 쇼핑, 배달의 민족, 스팀, 네이버 웹툰, 여기어때, 다나와 등의 많은 인터넷 쇼핑몰들에서 한번도 등장하지 않은 기능이지만 (2019년 10월 기준) 소비자들이 상품을 구매하는 중에 대부분 불편을 무릅쓰고 하는 일이다.

평소 우리가 온라인쇼핑을 할 때를 생각해보면 어떤 제품을 사기 전에, 그 하나의 제품만 보는 것이 아니라 여러 경쟁사의 제품이나 비슷한 제품들을 살펴보며 제품 정보나 리뷰를 비교해 구매를 결정하게 된다. 하지만 지금까지의 쇼핑몰 리뷰는 이러한 점에서 불편한 점이 존재한다. 바로 한 번에 한 가지 상품의 리뷰만 확인할 수 있다는 점이다. 방금 설명한 것처럼 소비자들은 쇼핑을 할 때 다양한 상품을 살펴보며 비교한다. 하지만 지금의 쇼핑몰은 한번에 한 가지 상품의 리뷰만 확인할 수 있기 때문에 여러 개의 구매 후보 상품의 리뷰를 비교하기 위해서는 여러 창이나 탭을 열어놓고 소비자가 직접 클릭하며 비교해야 하고, 이 과정에서 오는 피로감은 생각보다 크고, 한 눈에 비교를 할 수 없기 때문에 하나의 상품을 구매할 때 오랜 시간이 소요되게 된다.

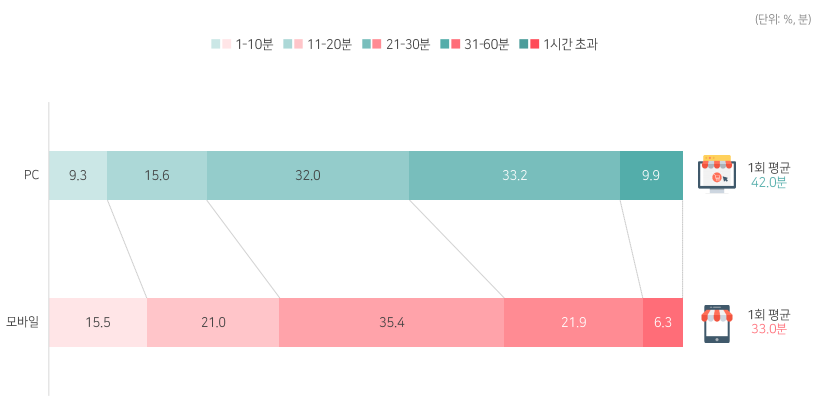


그림 3 1회 평균 쇼핑 시간

이러한 과정에서 지친 고객이 상품 구매를 포기할 수도 있고, 다른 상품의 리뷰를 찾아서 돌아다니다 보면 고객이 다른 웹 쇼핑몰로 빠져나갈 수도 있는 일이다, 만약 그렇게 된다면 쇼핑몰 입장에서는 매출에 직접적인 타격이 생기게 될 것이다.

이러한 문제를 해결하기 위해서는 고객이 편하고 쉽게 제품들의 리뷰를 비교하고, 원하는 정보를 빠르게 얻을 수 있도록 도와주어야 한다. 따라서 소비자의 편리한 쇼핑을 돕기 위해 한 눈에 제품 리뷰를 비교할 수 있는 서비스의 개발이 필요하다고 생각했고, 본 서비스를 개발하기로 했다.

## 2.2. System Overview

Review Revolution 은 기본적으로 각 상품의 카테고리화가 전부 진행되어 있는 기존 웹 쇼핑몰 안에서 리뷰를 관리하고, 비교 분석 하는 기능을 담당하는 모듈로 설계되어 있다.

고객이 미리 선택한 관심 키워드에 맞춰 상품 검색 시점에서 상품들을 보여주는 기능을 제공하고, 이에 따라 고객이 관심 있는 하나의 상품을 선택하게 되면 카테고리와 가격 또는 사용자들의 검색 기록 등을 기준으로 경쟁사 제품이나 비슷한 상품 등 예상 후보제품이 선택된다. 선택된 상품을 후보제품들과 함께 한 화면에 상품의 키워드와 그 키워드가 긍정적인지 부정적인지 여부, 전체 리뷰의 평점과 전체적 긍정도 그리고 best 리뷰를 같이 한 화면에 출력해준다. 이 기능을 통해 우리는 한 눈에 리뷰를 통해서 제품을 비교할 수 있는 서비스를 제공할 것이다.

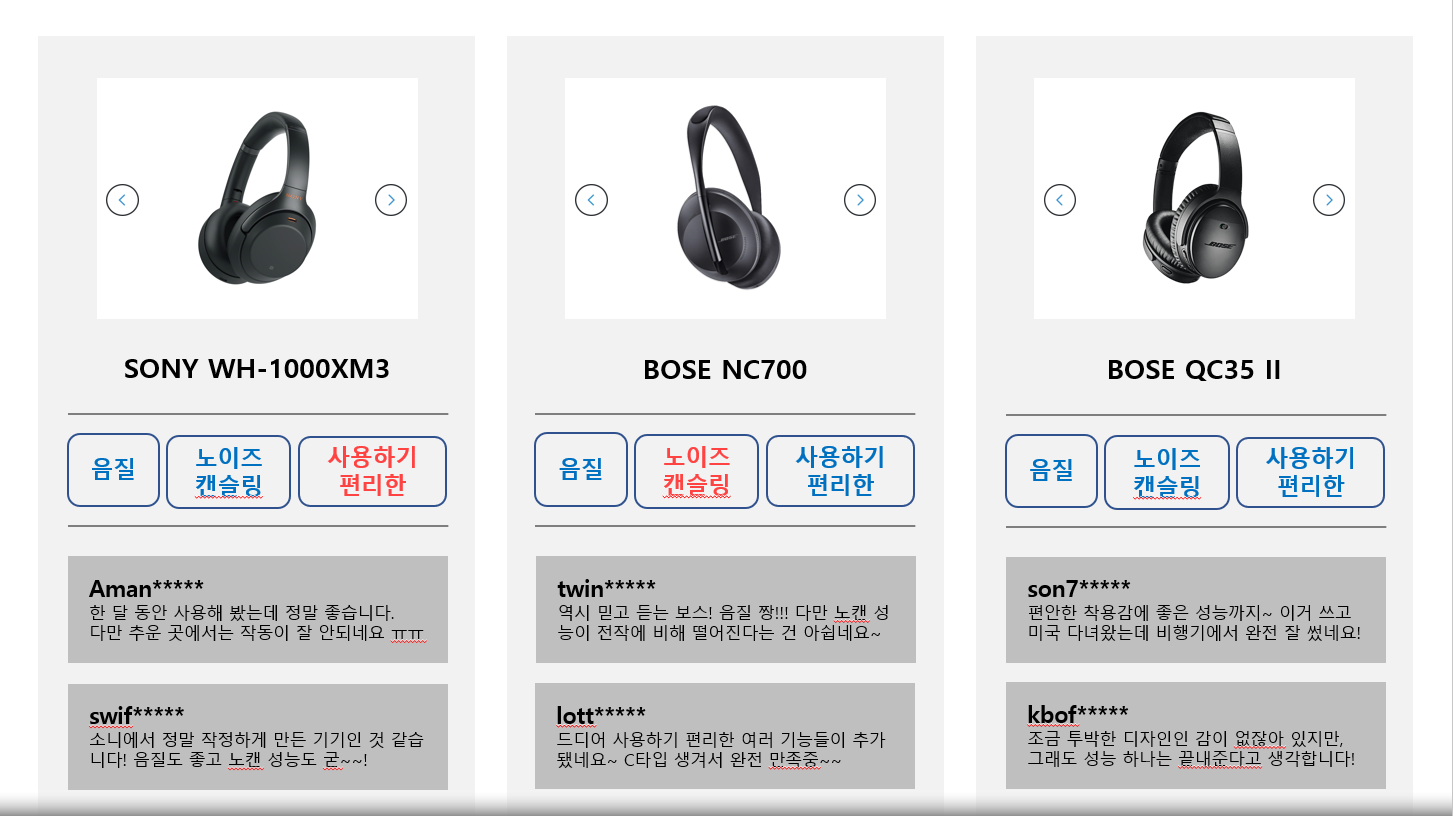


그림 4 Review Revolution - 모아 보기 기능

상품의 키워드는 리뷰에서 추출할 것이며, 이를 위해 자연어 처리 기술을 사용할 예정이다. 상품의 키워드를 리뷰에서 추출하는 이유는 이를 구매한 고객들의 실제 평가를 반영하기 위해서이다. 또한 각 리뷰 별로도 키워드를 추출해 리뷰를 한 눈에 파악하기 쉽게 구성할 예정이고, 키워드 별 리뷰 모아보기 기능을 제공해서 고객이 원하는 리뷰를 더 빠르게 찾을 수 있게 도울 것이다.



그림 5 Review Revolution – 상품 정보 화면

## Expected Effects

1. 쉽고 편리한 쇼핑 제공

모아 보기 기능을 통해 한 눈에 제품을 비교할 수 있게 만들어 기존의 서비스보다 더 쉽고 빠른 쇼핑을 할 수 있게 한다. 또한 키워드 검색 기능을 통해 소비자가 원하는 제품을 더 빠르게 찾을 수 있게 만들었고, 리뷰 분석을 동해 어떤 리뷰가 긍정적인지, 키워드는 무엇인지 한 눈에 파악할 수 있게 만들었다. 이러한 기능들을 통해 소비자가 원하는 상품들을 비교 분석하고 결정을 내리는 과정이 간편하고 짧아져서 소비자가 추가적인 노력을 들이지 않고도 쉽고 편리하게 쇼핑을 할 수 있을 것이다.

1. 고객 유치 및 수익 증대

위의 기대효과가 발생되면 앞선 배경에서 설명한 여러 상품의 리뷰를 비교하기 위해 번거롭게 창을 옮겨가면서 살펴봐야 하는 것을 한 눈에 볼 수 있기 때문에, 기존의 방식에 지쳐 있던 고객들이 이에 매력을 느껴 우리 서비스로 많이 유입이 될 것이고, 구매 결정까지의 시간이 단축되어서 고민하다 생각이 바뀌어 상품 구매를 포기하는 비율이 줄어들 것이다. 또 우리 서비스의 리뷰만으로 결정을 내릴 수 있는 충분한 정보를 얻을 수 있기 때문에 다른 사이트로 후보 상품을 검색하거나 정보를 얻으러 가는 도중 유출되는 소비자의 수가 줄어들 것이다. 이런 요인들로 인해 직접, 간접적으로 고객이 증가하고, 수익이 증대될 것이다.

# Glossary(마지막에 변경예정)

This chapter specifies the terms used in this documentation.

|  |  |
| --- | --- |
| Term | Description |
| Login | A function that let the user enter the application with a private customized information and setting |
| Search | A function that let the user to toggle through the products and find the most appropriate ones |
| Ranking | A page that lists the products in order of popularity or specified field of interest |
| Review | A string representing the user experience of a specific product |
| Rating | A numerical representation of the user experience of a specific product |
| Review Analysis | A systematic way to find some useful information from the review string |
| Review Crawler | A module that helps collecting reviews from different websites |
| Database | A backend term which represents a large collection of data (in this case Reviews) |
| API | A set of higher-class functions and interfaces developed by advanced programmers that can be imported and use easily in different projects |
| NLP | Natural Language Processing |

Table 1: Glossary

# User Requirements Definition

이 장에서는 사용자를 위해 제공되는 시스템의 기능적, 비기능적 요구사항을 사용자의 입장에서 설명한다. 따라서 비전문가도 이해할 수 있도록 자연어를 사용해야 하고, diagram 등의 시각 자료를 통해 이해를 도울 수 있도록 서술해야 한다..

## 4.1. Functional Requirements(sign up/sign in, customized search, item detail, review comparison, review analysis)

### Sign up/Login

해당 요구사항은 사용자가 ID와 Password를 통해 각자의 계정으로 우리의 서비스에 접근할 수 있도록 하는 것이다. 사용자는 서비스 사용에 앞서 서비스 제공에 필요한 정보를 입력하는 회원 가입 절차를 통해 각자의 ID와 Password를 서버에 등록한다. 이 때, ID는 고유한 값이어야 한다. 이 등록된 정보와 ID, Password는 우리의 Database에 보관한다. 사용자가 서비스 사용을 위해 ID와 Password를 입력하게 되면 DB에 해당 ID가 존재하는 경우 입력된 Password 와 Database 에 등록된 Password가 같은지 확인하여 서비스에 접근할 수 있게 한다. 만약 해당 ID가 존재하지 않거나 Password가 같지 않다면 서비스를 제공하지 않는다.

### Customized Search

해당 요구사항은 사용자가 검색 시 사전에 등록한 관심사나 검색 후 설정하는 키워드에 맞춰 상품 또는 키워드 순서를 결정하는 기능이다. 사용자는 회원가입 시 또는 계정 설정 시에 관심사를 사전에 등록할 수 있으며, 변경은 계정 설정을 통해 가능하다. 설정된 관심사는 검색된 상품의 순서를 관심사와 연관된 상품이 위에 보이게 한다. 관심 키워드 설정은 별도로 제공되는 영역을 통해 언제든지 설정 또는 변경이 가능하다. 설정된 키워드는 키워드에 대해 긍정적인 상품이 우선적으로 나올 수 있게 하며, 상품 키워드 표시 시에 설정된 키워드를 우선적으로 보이게 한다.

### item detail

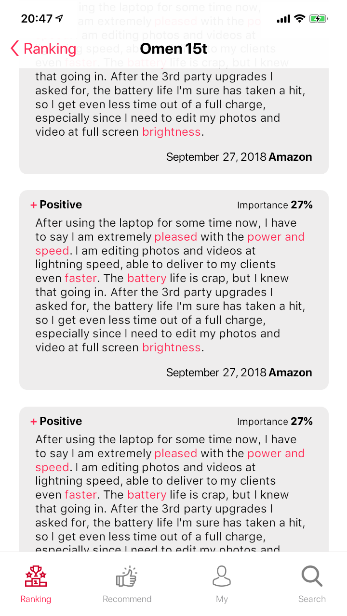
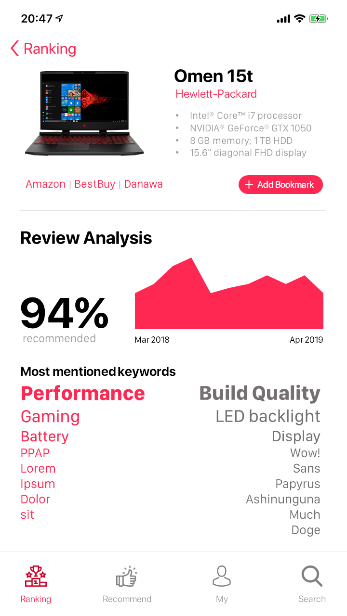


Figure 7: User requirement - Item Detail

Ranking 기능, Recommendation 기능에서 목록으로 나열된 각 상품의 세부 정보와 쇼핑몰의 구매 링크를 표시하고, 해당 상품의 전반적인 긍정 평가 정도, 리뷰에서 주로 언급된 키워드의 목록 등을 알기 쉽게 표시한다. 또한 밑으로 스크롤할 경우 실제 평가 집계에 사용된 리뷰의 목록을 확인할 수 있으며, 각 리뷰는 텍스트와 함께 리뷰의 긍정/부정 평가 여부, 중요도, 리뷰의 중요 키워드 등을 함께 표시해 사용자가 확인할 수 있다. 또한 해당 상품을 북마크에 추가해 다시 시스템에 접속했을 때 재검색할 필요 없이 Mypage에서 바로 확인할 수 있다.

### Review Comparison

사용자가 상품을 클릭해 상세페이지에 들어갔을 때 클릭한 상품, 유사한 상품 그리고 상품들의 키워드, 대표적인 리뷰를 보여준다.

유사한 상품을 선정할 때, 사용자가 클릭한 상품에 관심도가 높다고 가정하고, 같은 카테고리에 다른 상품의 예상 관심도를 계산한다. 예상관심도가 높은 두 개의 상품을 유사한 상품으로 정하고 클릭한 상품과 같이 표시한다. 예상 관심도를 계산할 때에는 디자인, 가격 등 상품의 특성을 고려한다.

각 상품 밑에는 상품의 대표 리뷰를 표시한다. 사용자는 리뷰가 도움이 되었다고 추천할 수 있고, 추천 수에 따라 3~5개의 리뷰를 대표 리뷰로 정한다. 사용자는 한 화면에서 여러 상품의 키워드를 비교하고 리뷰를 확인해 원하는 제품을 구매할 수 있다.

### Review Analysis

상품들은 상품의 키워드와 함께 표시된다. 각 키워드는 초록색부터 빨간색까지 색깔로 표시되며, 이를 통해 키워드가 상품의 장점 혹은 단점인지 알 수 있다. 예를 들면 옷이 가질 수 있는 키워드에는 재질, 가성비 등이 있다. 재질이 초록색, 가성비가 빨간색으로 표시되면 사용자는 상품의 재질이 좋지만 가성비는 좋지 않다고 직관적으로 알 수 있다.

이를 위해 google nlp api를 이용해 사용자의 리뷰를 분석한다. 리뷰에서 키워드를 추출하고 긍정적 혹은 부정적인 의미로 사용했는지 확인한다. 사용자의 구매 여부에 영향을 끼치는 중요한 키워드는 상품과 함께 같이 표시해 사용자가 제품을 파악하는데 도움을 준다.

## Non-functional Requirements(우리 프로젝트 상황에 맞는 설명으로 대체)

### Product requirements

1. Usability

This is one of the key requirements for the success of any E-commerce application or website. The system should provide a very intuitive yet simple user interface (UI). Moreover, DEAListic implements that through clear and few buttons and tabs which help the user to navigate quickly between different sections without any inconvenience.

또한 본 시스템은 다양한 기기에서의 접속을 전제하고 있으므로, 모바일, PC 등 여러가지 플랫폼에서의 원활한 접속이 가능해야 한다. 또한 사용자의 접속 환경에 맞는 레이아웃을 표시함으로써 사용자 경험을 해치지 않도록 하고, 모바일 사용자의 경우 시스템으로 바로 접속할 수 있는 방법이 존재해야 한다.

1. Performance

Word processing should handle a lot of exceptions to make reliable grade from reviews. So, it should cover a lot of variables and exceptions, and to give maximum satisfaction to user, grading algorithm should divide grade scale in detail for specific grouping. And ranking lists should be updated regularly, because there can be fast update in the product of dealistic web page.

1. Dependability

Our project achieves maximized dependability through making sure all actions are atomic. In other words, if the system crashes in the middle of committing a purchase or writing a review the connection with the server will be lost thus no asynchronous modifications will be reflected in the system. Moreover, if the application crashes randomly because of a hardware failure as soon as the user reconnects with the server everything will go back to its original state. Finally, the overall results provided by our review text analysis system will be extremely accurate and dependable as it is based on a highly developed API (Google Natural Language API).

1. Security

Review must be reliable and must guarantee the anonymity of review writer. And each personal information must be protected from outer access and there for we need security. And also, Id and passwords should be protected by saving it securely.

### Organizational requirements

1. Operational  
   현재 지속적으로 운영할 수 있는 하드웨어 자원이 없으므로, 상용 클라우드 서버에 배포할 것을 전제로 한다. 따라서 다양한 클라우드 서버에 배포할 수 있도록 hardware-dependent한 구현을 자제하고, 서버의 주소가 바뀌어도 대응할 수 있게 시스템을 설계해야 한다.
2. Development  
   보다 빠른 기간에 시스템 개발이 가능하도록 Parallel Development를 도입한다. 이를 위해 개발 프로세스를 plan-driven으로 설계하고, 개발 단위를 프론트엔드와 백엔드로 나누며, 프론트엔드와 백엔드는 서로 독립적으로 구동할 수 있게 설계해야 한다.

### External requirements

1. External System Policy  
   본 시스템은 다른 시스템의 리뷰 목록을 재가공한 정보를 제공하는 시스템이므로, 데이터의 무단 전재를 금지하는 시스템의 경우 데이터 수집 방식에 따라 운영 정책을 위반할 소지가 있다. 따라서 정보 수집 대상 시스템의 운영 정책을 면밀히 확인하고, 해당 정책을 위반하지 않는 선에서 데이터를 수집해야 한다.
2. Personal Information Protection  
   본 시스템은 북마크 기능 및 추천 기능을 제공하기 위해 회원제로 운영한다. 따라서 사용자의 개인정보를 수집하며, 개인 정보가 제대로 보호되지 않을 경우 민형사상의 책임을 질 가능성이 있다. 따라서 가입 시 사용자 식별에 필요한 최소한의 정보만을 제공받고, 개인정보가 유출되지 않도록 충분한 노력을 기울여야 한다.

# System Architecture

이번 챕터에서는 시스템의 개괄적인 Architecture를 기술한다. 시스템의 전체적인 구조와 각 서브시스템의 구성, 서브시스템간 관계를 대략적으로 설명하며, 각 구조는 다이어그램을 첨부하여 이해를 돕는다.

## Frontend Architecture

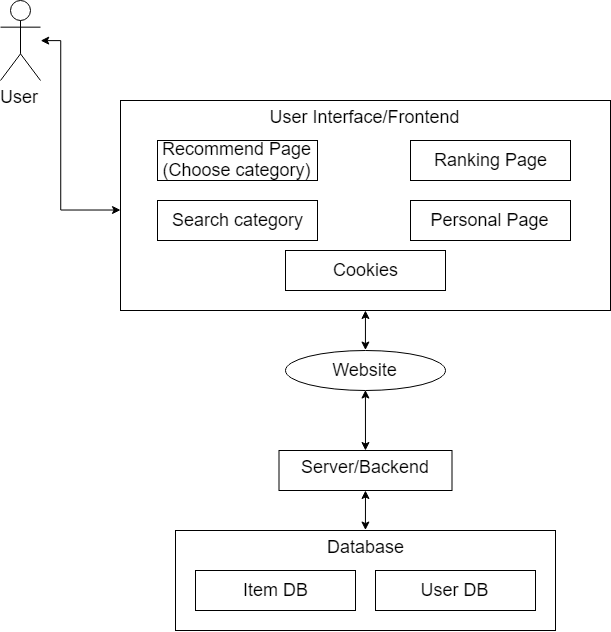


Diagram 3: Frontend Architecture

The frontend architecture is equivalent to the user interface that users interact with, make their requests to or retrieve some data from. In Dealistic, the main interface is made of four main tabs; the recommendation page, Ranking list, User’s personal information page and a search tab. Since the focus of the diagram is on the frontend, hence the details of the server/backend are not described.

## Backend Architecture

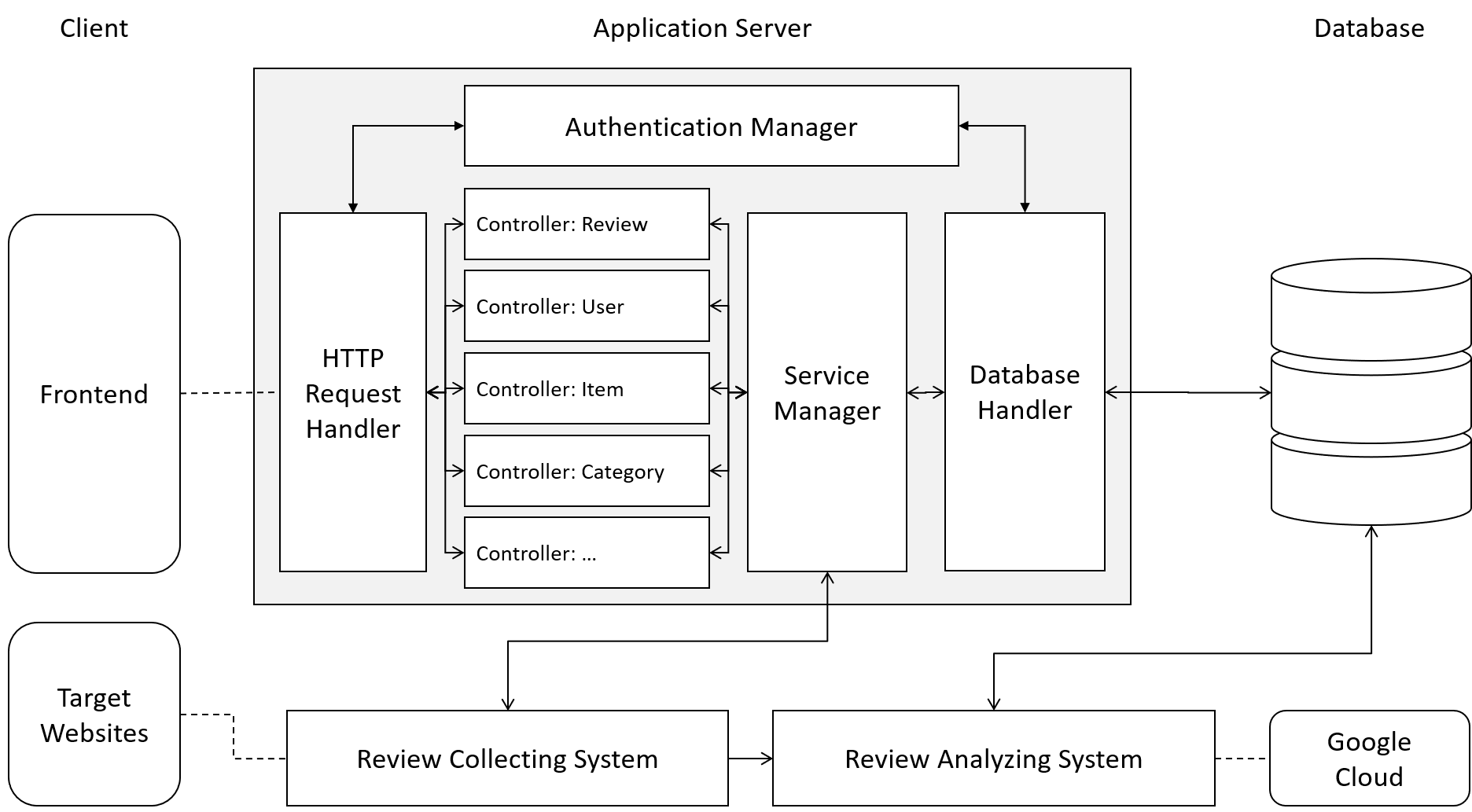


Diagram 4: Backend Architecture

백엔드 시스템은 위와 같이 구성되어 있는데, 크게 나누어 Application Server와 Review Collection System, Review Analyzing System이 있다.

## Review Analysis System

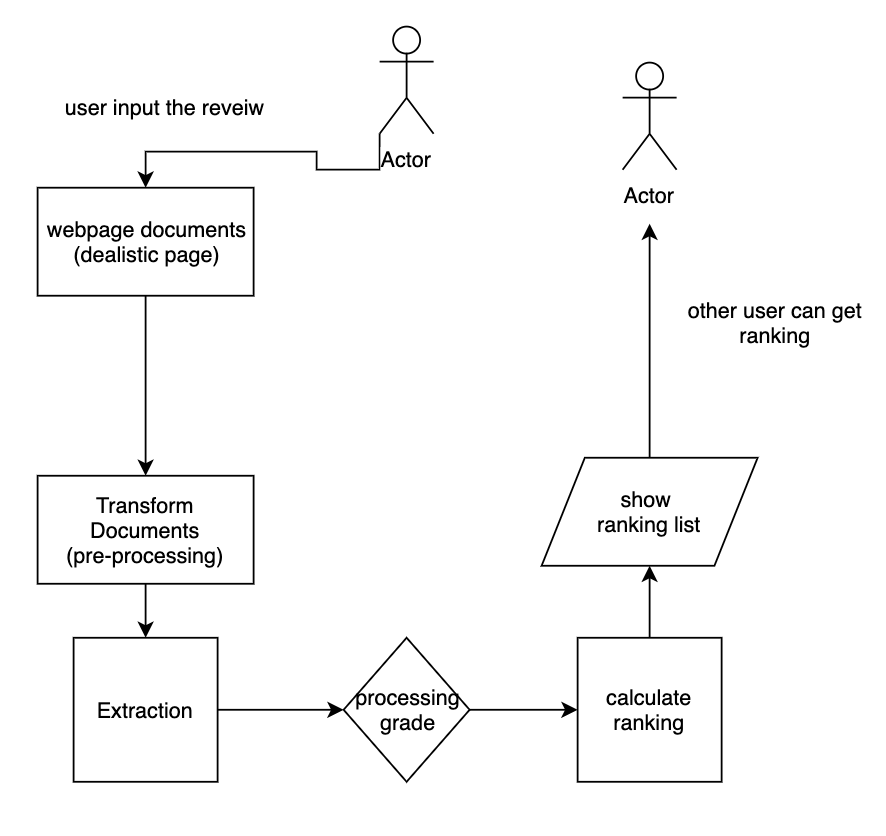


Diagram 5: Review Analysis System Architecture

Review analysis system takes written reviews as arguments, applies tokenization, does sentiment analysis and finally marks the most important keywords and generates a new more accurate rating based on the original rating.

## Recommendation System

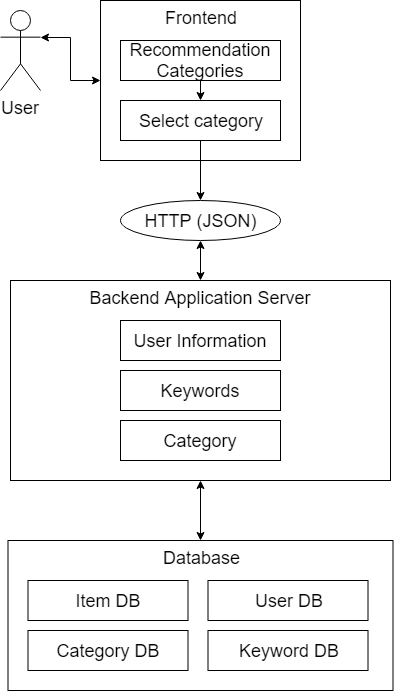


Figure 9: Recommendation System Architecture

Recommendation system shows recommended items based on Search history, User DB and Item DB. Server decides item categories from search history from user DB and choose items of chosen category. Server list image and name of chosen items. User can choose or search category. Search history are saved in User DB. User can access item details from recommendation list.

## Ranking System

Ranking 시스템은 데이터베이스에 저장되어 있는 구매 내역에 대한 정보와 상품에 대한 정보를 사용하여 각 상품평 또는 상품의 특성에서 추출한 keyword별로 누적 구매 수가 높은 상품 순으로 고객에게 정보를 제공해주는 시스템이다.

Ranking 시스템은 두가지 방식으로 사용자에게 제공될 수 있다. 첫째는 사용자가 관심있는 keyword와 관련된 ranking system을 먼저 자동적인 keyword에 관한 ranking을 제공해주는 것이고 둘째는 사용자가 원하는 keyword에 대해 검색함으로써 수동적으로 ranking system을 사용자에게 제공해주는 것이다.

고객은 제공된 ranking system을 참고하여 다른 사용자들의 관심 상품과 다른 사용자들이 많이 구매한 상품을 참고함으로써 더 편리한 쇼핑을 진행할 수 있다.

# System Requirements Specification

이 챕터에서는 유저 요구사항 명세에서 정의된 요구사항을 바탕으로 세부적인 Functional requirements와 Non-functional requirements를 명시한다. 각 requirements는 Name, description, inputs, source, outputs, destination, action, requirements, pre-condition, post-condition, side effects를 포함해 최대한 상세히 기술한다.

## Functional Requirements – Frontend

### Ranking

|  |  |  |
| --- | --- | --- |
| Name | Ranking Function | |
| Description | Show ranking page in ascending order from top to down. | |
| Inputs | Ranking\_Duration | string representing the ranking duration (month, week… etc) |
| Source (Input) | User input | |
| Outputs | Updated sorted ranking list | |
| Destination (Output) | User terminal (web browser) | |
| Action | When the user clicks on the ranking tab the chosen ranking duration is sent to the backend which processes the request and gets it from the data base. Finally, the sorted ranking list will be handed over to the web browser. | |
| Requirements | Ranking duration should be specified | |
| Pre-condition | - | |
| Post-condition | - | |
| Side effects | Frontend will be changed into the chosen ranking list view. | |

Table 2: Ranking Function

### Show Item Detail

|  |  |  |
| --- | --- | --- |
| Name | Show Item Detail Function | |
| Description | Show the inner structure of each individual object in the ranking list | |
| Inputs | Product\_Name | String representing the name of the item |
| Source (Input) | User input | |
| Outputs | An object containing all details associated with one specific item. | |
| Destination (Output) | User terminal (web browser) | |
| Action | When the user clicks on a specific item from the ranking list or after doing a search. The chosen item’s name is sent to the backend which in turn retrieves all the information associated with this item such as rating, reviews… etc. Finally, the item’s details will be shown on the UI. | |
| Requirements | - | |

Table 3: Show Item Detail

### Add Item to Bookmark

|  |  |  |
| --- | --- | --- |
| Name | Add Item to Bookmark | |
| Description | 특정한 상품을 현재 접속한 사용자의 북마크 목록에 저장한다. | |
| Inputs | User ID | 사용자의 아이디 |
| Item ID | 상품 아이디 |
| Source (Input) | 사용자의 버튼 클릭으로 발생한 프론트엔드 요청 | |
| Outputs | 성공/실패 메시지 | |
| Destination (Output) | 사용자 단말 (웹브라우저) | |
| Action | User ID와 Item ID를 이용하여 각각 유저 정보와 상품 정보를 데이터베이스에서 얻고, 데이터베이스에 해당 유저가 이미 상품을 북마크에 추가했는지 확인한다. 만약 유저가 이미 북마크를 가지고 있다면, 실패 메시지를 반환한다. 안약 유저에게 북마크가 없을 경우, 데이터베이스에 기록 후 성공 메시지를 반환한다. | |
| Requirements | User ID, Item ID가 모두 필요하다. | |
| Pre-condition | 상품이 접속한 사용자의 북마크에 존재하지 않는다. | |
| Post-condition | 상품이 접속한 사용자의 북마크에 추가되고, Add to Bookmarks 버튼이 Remove from Bookmarks 버튼으로 바뀐다. | |

Table 4: Add Item to Bookmark

### Show Recommendation List

|  |  |  |
| --- | --- | --- |
| Name | Show Recommendation List | |
| Description | Show recommendation list based on user’s information | |
| Inputs | Category ID | Recommendation Category ID |
| Source (Input) | User input | |
| Outputs | List of item which satisfy all keywords | |
| Destination (Output) | User terminal (web browser) | |
| Action | When user clicks the specific recommendation, category id is passed to backend application. From Recommendation Database, keywords contained in specific recommendation category is gathered, and using these keyword, search items that satisfy all keywords. | |
| Requirements | Category ID must exist | |
| Pre-condition | - | |
| Post-condition | - | |
| Side effects | - | |

Table 5: Show Recommendation List

### Add Recommendation category

|  |  |  |
| --- | --- | --- |
| Name | Add Recommendation category Function | |
| Description | 해당 서비스의 부가적인 기능으로, 사용자가 조회했던 item들과 유사한 키워드를 포함한 후기를 가진 상품을 추천해준다. 하나의 상품만을 추천해주는 것이 아니라 여러가지 상품들을 나열하여 추천해주기 때문에 category형태로 나타나 진다. | |
| Inputs | Keyword, data | Keyword, data about user’s interests. |
| Source (Input) | Item viewed by user | |
| Outputs | recommended product | |
| Destination (Output) | User terminal (web browser) | |
| Action | It recommends products with a later period, including keywords similar to items viewed by users. | |
| Requirements | None. If a user is logged in, previous search records can show more recommended products. | |
| Pre-condition | None. | |
| Post-condition | None. | |
| Side effects | None. | |

Table 6: Add Recommendation Category

### Mypage

|  |  |  |
| --- | --- | --- |
| Name | Mypage Function | |
| Description | 사용자의 고유한 정보를 표시하는 페이지 | |
| Inputs | ID | Id string of user |
| Password | Password string of user |
| Source (Input) | User authentication and my page button click | |
| Outputs | Authenticated user information object | |
| Destination (Output) | User terminal (web browser) | |
| Action | 사용자별로 개인화된 페이지이다. 이 페이지를 통해서 사용자의 주문/배송조회, 구매내역, 개인 정보 관리, 자신이 작성한 상품평 관리 등의 기능이 처리될 수 있다. 그러므로 이 페이지는 로그인한 사용자만 접근 가능하다. | |
| Requirements | Both Id and password must be specified. | |
| Pre-condition | non-authorized user cannot access. | |
| Post-condition | authorized user only can access Mypage. | |
| Side effects | Frontend will be changed into authorized version including specific user’s information. | |

Table 7: Mypage

## Functional Requirements – Backend

### Login

|  |  |  |
| --- | --- | --- |
| Name | Login Function | |
| Description | Authenticate and authorize the user | |
| Inputs | ID | Id string of user |
| Password | Password string of user |
| Source (Input) | User input | |
| Outputs | Authenticated user information object | |
| Destination (Output) | User terminal (web browser) | |
| Action | Search user information stored in the database by using the passed ID. After hash the passed password, compare it with the hash stored in the database. If correct, create a user object containing other user information (nickname, authority, …) and session ID, and return the object to the web browser. | |
| Requirements | Both Id and password must be specified. | |
| Pre-condition | User can access only non-authorized area. | |
| Post-condition | User can access all authorized area. | |
| Side effects | Frontend will be changed into authorized version including specific user’s information. | |

Table 8: Login Function

### Review Collecting System

|  |  |  |
| --- | --- | --- |
| Inputs | Target Site List | 크롤링할 목표 사이트의 목록 |
| Source (Input) | Backend Application Server, Cron Scheduler | |
| Outputs | Parsed Item Review List | |
| Destination (Output) | Review Analysis System | |
| Action | 타겟 사이트의 리스트를 순회하며, 목표 상품들을 검색해 나오는 상품 페이지의 리뷰 리스트를 파싱한다. 제목과 작성자, 리뷰 내용, 사진을 분류해 목록화한다. | |
| Requirements | - | |
| Pre-condition | - | |
| Post-condition | - | |
| Side effects | - | |

Table 9: Review Collecting System

### Review Analysis System

|  |  |  |
| --- | --- | --- |
| Inputs | Raw Reviews | 크롤링된 상품 리뷰 목록 |
| Source (Input) | Review Collection System, Backend Application Server | |
| Outputs | Analyzed Review List | |
| Destination (Output) | Review Database Updater | |
| Action | 크롤링된 상품 리뷰 목록의 리뷰 내용을 구글 클라우드 자연어 처리 API에 쿼리한다. 쿼리 결과에서 키워드 목록, 긍정 평가 정도를 따로 분류해 저장한다. 긍정 평가 정도와 상품 사진 게시 여부, 리뷰 길이, 키워드 종류를 평가해 리뷰의 중요도를 매긴 다음 긍정 평가 정도와 곱해 리뷰 점수에 반영한다. | |
| Requirements | - | |
| Pre-condition | 분석되지 않은 상품 리뷰 목록 | |
| Post-condition | 분석된 상품 리뷰 목록 | |
| Side effects | - | |

Table 10: Review Analysis System

### Review Database Updater

|  |  |  |
| --- | --- | --- |
| Inputs | Review List | 분석 완료된 리뷰 리스트 |
| Source (Input) | Review Analysis System, Backend Application Server | |
| Outputs | - | |
| Destination (Output) | - | |
| Action | 리뷰 분석 시스템에서 분석을 완료한 리뷰의 리스트를 저장한다. 리뷰 제목, 작성자, 출처 사이트, 내용, 키워드, 긍정 평가 정도, 중요도를 포함해 데이터베이스에 기록한다. | |
| Requirements | - | |
| Pre-condition | - | |
| Post-condition | - | |
| Side effects | 리뷰 데이터가 업데이트될 경우 현재 표시되고 있는 상품 정보에 변동이 발생한다. 따라서 프론트엔드에서의 다음 요청시 업데이트된 상품 정보를 리턴해야 한다. | |

Table 11: Review Database Updater

## Non-functional Requirements

### Product requirements

1. Usability

This is one of the key requirements for the success of any E-commerce application or website. The system should provide a very intuitive yet simple user interface (UI). Moreover, DEAListic implements that through clear and few buttons and tabs which help the user to navigate quickly between different sections without any inconvenience.

또한 본 시스템은 다양한 기기에서의 접속을 전제하고 있으므로, 모바일, PC 등 여러가지 플랫폼에서의 원활한 접속이 가능해야 한다. 또한 사용자의 접속 환경에 맞는 레이아웃을 표시함으로써 사용자 경험을 해치지 않도록 하고, 모바일 사용자의 경우 시스템으로 바로 접속할 수 있는 방법이 존재해야 한다.

1. Performance

Word processing should handle a lot of exceptions to make reliable grade from reviews. So it should cover a lot of variables and exceptions, and to give maximum satisfaction to user, grading algorithm should divide grade scale in detail for specific grouping. And ranking lists should be updated regularly, because there can be fast update in the product of dealistic web page.

1. Dependability

Our project achieves maximized dependability through making sure all actions are atomic. In other words, if the system crashes in the middle of committing a purchase or writing a review the connection with the server will be lost thus no asynchronous modifications will be reflected in the system. Moreover, if the application crashes randomly because of a hardware failure as soon as the user reconnects with the server everything will go back to its original state. Finally, the overall results provided by our review text analysis system will be extremely accurate and dependable as it is based on a highly developed API (Google Natural Language API).

1. Security

Review must be reliable and must guarantee the anonymity of review writer. And each personal information must be protected from outer access and there for we need security. And also, Id and passwords should be protected by saving it securely.

### Organization requirements

1. Operational  
   현재 지속적으로 운영할 수 있는 하드웨어 자원이 없으므로, 상용 클라우드 서버에 배포할 것을 전제로 한다. 따라서 다양한 클라우드 서버에 배포할 수 있도록 hardware-dependent한 구현을 자제하고, 서버의 주소가 바뀌어도 대응할 수 있게 시스템을 설계해야 한다.
2. Development  
   보다 빠른 기간에 시스템 개발이 가능하도록 Parallel Development를 도입한다. 이를 위해 개발 프로세스를 plan-driven으로 설계하고, 개발 단위를 프론트엔드와 백엔드로 나누며, 프론트엔드와 백엔드는 서로 독립적으로 구동할 수 있게 설계해야 한다.

### External requirements

1. External System Policy  
   본 시스템은 다른 시스템의 리뷰 목록을 재가공한 정보를 제공하는 시스템이므로, 데이터의 무단 전재를 금지하는 시스템의 경우 데이터 수집 방식에 따라 운영 정책을 위반할 소지가 있다. 따라서 정보 수집 대상 시스템의 운영 정책을 면밀히 확인하고, 해당 정책을 위반하지 않는 선에서 데이터를 수집해야 한다.
2. Personal Information Protection  
   본 시스템은 북마크 기능 및 추천 기능을 제공하기 위해 회원제로 운영한다. 따라서 사용자의 개인정보를 수집하며, 개인 정보가 제대로 보호되지 않을 경우 민형사상의 책임을 질 가능성이 있다. 따라서 가입 시 사용자 식별에 필요한 최소한의 정보만을 제공받고, 개인정보가 유출되지 않도록 충분한 노력을 기울여야 한다.

## Scenario Examples

### Login Scenario

1. Initial Assumption

The user has already created an account earlier and wants to login with the same information (Email and ID) he used for registration.

1. Normal flow of events

The user inputs his information and hits the login button. In the meantime, the database checks for matching information and finally the user can enter the platform with a customized interface according to his personal usage.

1. What can go wrong

One of the very obvious unprecedented scenarios will be if the user is not registered and the database. This thing will obviously generate an error msg.

1. System state on completion

The user is welcomed with his customized interface where he can see his information and filters of his preference.

### Recommendation Scenario

1. Initial Assumption

사용자는 특정 카테고리의 상품을 구매하기 위해 본 시스템에 접속하였다. 사용자는 어떤 종류의 상품을 구매해야 하는지, 어떠한 특성을 가진 상품을 원하는지는 알고 있지만 특정 상품의 이름을 알고 있지는 않다.

사용자는 본 시스템에 이미 등록한 사용자이다.

1. Normal flow of events

사용자가 본 시스템에 접속해 ID/PW를 입력하고 로그인하면, 4개의 상위 메뉴가 표시된다. 사용자가 Recommendation 메뉴를 누르면, 여러 가지 상품 카테고리가 나열된 하위 메뉴가 표시되며, 그 중 자신이 원하는 하위 메뉴를 선택하면, 해당 카테고리의 상품 중 일부 특성을 만족하는 상품을 분류한 추천 카테고리들이 나열된다. 사용자가 추천 카테고리를 선택하면, 해당 카테고리의 하위 키워드를 만족하는 상품들을 데이터베이스에서 검색한다. 검색한 결과를 Text Analysis system, Ranking System에 의해 이미 분석되어 있는 상품 점수에 따라 정렬하고, 제일 높은 점수를 기록한 상품부터 표시한다. 사용자는 가장 위에 있는 상품을 선택하고 상품을 분석한 데이터를 참고한다. 사용자가 상품을 구매하기 위해 쇼핑몰로 연결되는 버튼을 누르면, 데이터베이스에 저장되어 있는 쇼핑몰 구매 페이지 링크로 이동한다.

1. What can go wrong

특정 추천 카테고리의 키워드를 모두 만족하는 상품이 없을 가능성이 있다. 이 경우 일부 키워드를 만족하는 상품을 보여주되, 결과 상단에 정확한 결과가 없으며 일부 조건을 만족하는 상품을 출력하고 있다는 메시지를 보여준다.

1. System state on completion

사용자가 상품을 구매한 경우 사용자 정보에 구매 기록을 저장하며, 이후 사용자가 다른 물건을 구매하기 위해 다시 시스템에 접속했을 때 해당 기록을 사용하게 된다.

# System Models

이번 챕터에서는 각 시스템 컴포넌트와 전체 시스템, 그리고 시스템을 둘러싼 환경 사이의 관계를 여러 가지 Diagram을 통해 기술한다.

## 7.1. Context models

### A. Context Diagram

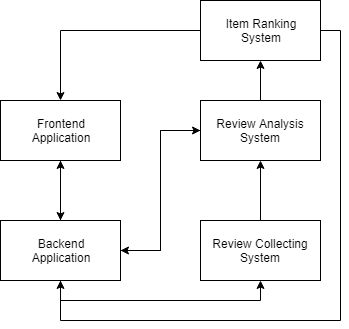


Diagram 6: Overall context diagram

### B. Process Diagram

스크린샷이(가) 표시된 사진

자동 생성된 설명

Diagram 7: Overall process diagram

## 7.2. Interaction models

### Use case Diagram

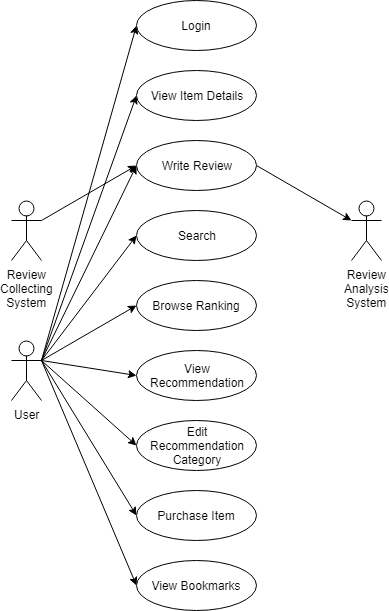


Diagram 8: Use case diagram

### Tabular Description of Use case Diagram

1. Login

|  |  |
| --- | --- |
| Use case | Login |
| Actor | User, Database(User) |
| Description | 접속한 사용자가 시스템에 등록되어 있는 유저 정보와 일치하는지 판단하는 과정이다. |
| Trigger | 사용자가 ID와 PW를 입력한다. |
| Success Response | 사용자 ID가 데이터베이스에 존재하고, PW의 Hash가 데이터베이스의 값과 일치하면 사용자 정보와 접속 권한을 포함한 객체를 돌려준다. |
| Failure Response | 사용자 ID가 데이터베이스에 없을 경우 회원 가입 페이지로 이동하며, PW가 일치하지 않을 경우 오류 메시지를 발생시킨다. |

1. View Item Details

|  |  |
| --- | --- |
| Use case | View Item Details |
| Actor | User, Database(Item) |
| Description | 추천 목록, 랭킹, 검색 등에서 나열된 상품의 세부 정보를 확인한다. |
| Trigger | 사용자가 특정한 상품 목록 엔트리를 클릭한다. |
| Success Response | 상품 ID를 이용해 데이터베이스에서 해당 ID를 가진 상품 정보를 불러온다. |
| Failure Response | - |

1. Write Review

|  |  |
| --- | --- |
| Use case | Write Review |
| Actor | User, Database(Review), Review Analysis System |
| Description | 사용자가 특정한 상품에 대한 리뷰를 작성한다. |
| Trigger | 사용자가 특정 상품에 대해 리뷰 작성 버튼을 클릭하고, 리뷰 제목과 내용을 입력한다. |
| Success Response | 리뷰 데이터베이스에 사용자의 새로운 리뷰를 등록하고, Review Analysis System에 해당 리뷰에 대한 분석을 실행시킨다. |
| Failure Response | 리뷰 내용이 지나치게 짧거나, 비속어를 사용할 경우 리뷰를 등록하지 않고 오류 메시지를 발생시킨다. |

1. Search

|  |  |
| --- | --- |
| Use case | Search |
| Actor | User, Database(Item) |
| Description | 사용자가 이름, 카테고리, 제품 설명 등을 이용해 상품을 검색한다. |
| Trigger | 사용자가 검색 조건을 입력하고 검색 버튼을 누른다. |
| Success Response | 검색 조건을 데이터베이스 SQL 질의문으로 변형하고, 해당 질의문을 실행시킨 후 일치하는 상품 목록을 가공해 돌려준다. |
| Failure Response | 검색 조건에 일치하는 상품이 없을 경우 빈 목록을 돌려주고, 해당하는 상품이 없다는 오류 메시지를 발생시킨다. |

1. Browse Ranking

|  |  |
| --- | --- |
| Use case | Browse Ranking |
| Actor | User, Database(Item), Item Ranking System |
| Description | 사용자가 특정 상품 카테고리의 상품 순위를 조회한다. |
| Trigger | 사용자가 Ranking 탭을 클릭하고, 상품 카테고리를 선택한다. |
| Success Response | 상품 카테고리 ID를 이용해 Item Ranking System에 해당 카테고리의 상품 순위를 조회한다. Item Ranking System은 상품 데이터베이스에서 해당 카테고리의 상품을 모두 조회한 후, 점수 순으로 정렬한다. 정렬한 상품 목록을 돌려준다. |
| Failure Response | - |

1. View Recommendation

|  |  |
| --- | --- |
| Use case | View Recommendation |
| Actor | User, Database(Item, Recommendation, Keyword) |
| Description | 사용자가 특정한 목적에 맞는 추천 카테고리를 조회한다. |
| Trigger | 사용자가 추천 카테고리를 누른다. |
| Success Response | 추천 카테고리에 포함된 키워드를 모두 가지고 있는 상품을 상품 데이터베이스에서 조회한 후, 해당 목록을 상품이 가지고 있는 점수에 따라 정렬해 보여준다. |
| Failure Response | 키워드를 모두 만족시키는 상품이 없을 경우 오류 메시지를 출력 후 일부 키워드를 만족시키는 연관 상품을 출력한다. |

1. Edit Recommendation Category

|  |  |
| --- | --- |
| Use case | Edit Recommendation Category |
| Actor | User, Database(Item, Recommendation, Keyword) |
| Description | 특정한 목적에 맞는 추천 카테고리가 없을 경우 사용자가 직접 추천 카테고리를 추가한다. |
| Trigger | 사용자가 추천 카테고리 추가 버튼을 누르고, 해당 카테고리에 대한 설명과 키워드를 추가한다. |
| Success Response | 성공 메시지와 함께 화면을 새로고침해 추가한 추천 카테고리가 반영된다. |
| Failure Response | 카테고리 설명에 비속어가 들어가 있을 경우 오류 메시지를 출력한다. |

1. Purchase Item

|  |  |
| --- | --- |
| Use case | Purchase Item |
| Actor | User, Database(Item) |
| Description | 사용자가 물건을 구매하기 위해 쇼핑몰 링크로 이동한다. |
| Trigger | 사용자가 쇼핑몰 링크 목록 중 하나를 클릭한다. |
| Success Response | 해당하는 쇼핑몰 링크로 이동한다. |
| Failure Response | 해당하는 쇼핑몰 링크가 삭제되었거나 저장된 조건과 다를 경우, 접속을 중단하고 오류 메시지를 출력한다. |

1. View Bookmarks

|  |  |
| --- | --- |
| Use case | Purchase Item |
| Actor | User, Database(User, Item, Bookmark) |
| Description | 사용자가 자신이 추가한 북마크 목록을 조회한다. |
| Trigger | 사용자가 마이페이지에서 북마크 목록을 조회한다. |
| Success Response | 사용자 id를 이용해 북마크 데이터베이스에서 해당 사용자가 저장한 북마크의 목록을 불러온다. |
| Failure Response | 저장한 북마크가 없을 경우 빈 목록을 리턴한다. |

## 7.3. Behavioral models

이 장에서는 본 시스템에서 가장 중점으로 두는 하위 시스템인 ‘Ranking system’ 과 ‘Text analysis system’에 초점을 두고 behavioral models을 작성하였다.

### Ranking System DFD

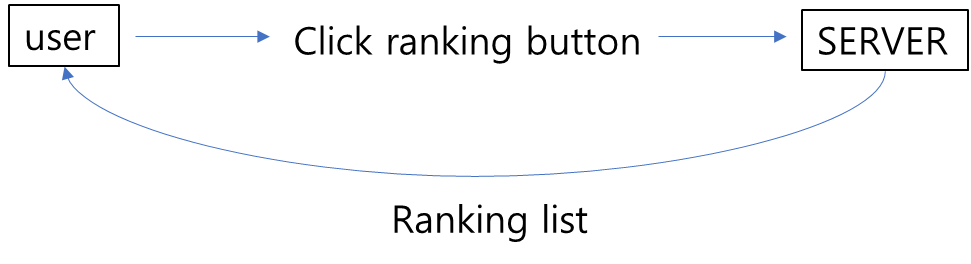


Diagram 9: Data flow diagram of ranking system

### Review Analysis DFD

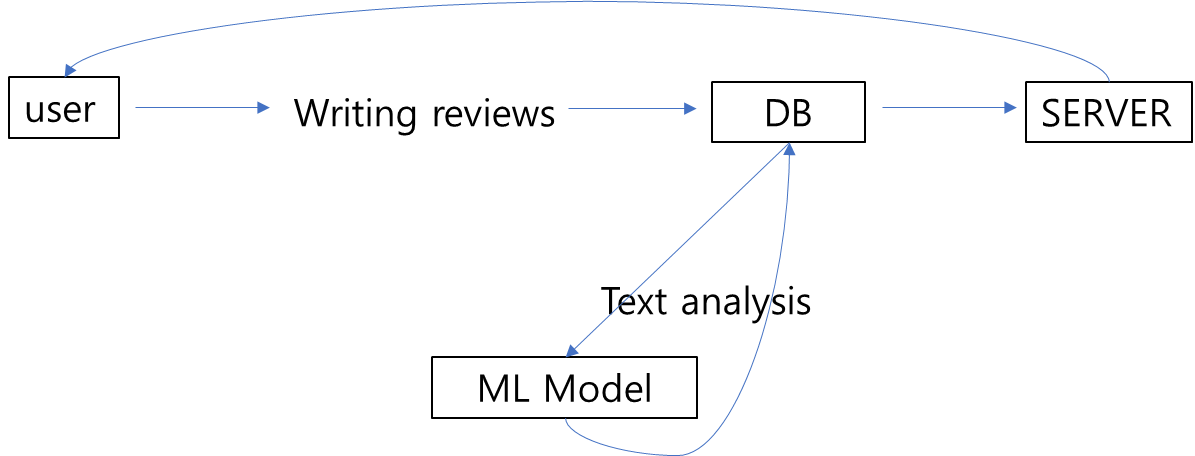


Diagram 10: Data flow of text analysis system

### MyPage Sequence Diagram.

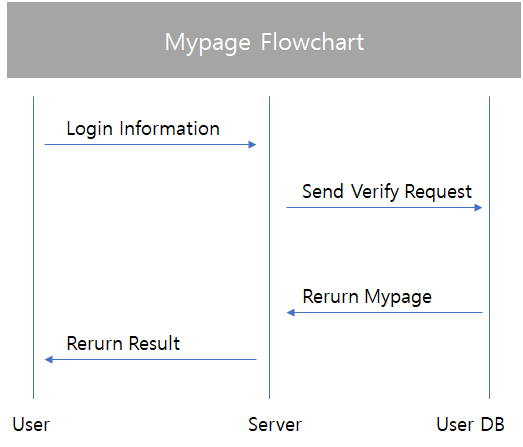


Diagram 11: Sequence Diagram of Mypage Function

# System Evolution

이번 챕터에서는 시스템을 둘러싼 환경과, 시스템을 배포하고 운영하며 발생할 수 있는 시스템을 둘러싼 여러 가지 변화를 예상하고, 예상된 변경사항에 시스템이 어떻게 대응할 수 있을지에 대해 기술한다. 이를 통해 시스템의 추후 수정 시 발생할 수 있는 부수적인 설계 변경의 여지를 없애고 수정 비용을 낮춘다.

### Compare Feature

One of our future plans to secure maintainability and to achieve one of our schemes is to implement a compare feature where users can search for 2 different products and after that a comparison technique can be applied to check which phone is more recommended based on the written reviews.

### Voice Analysis

Since on the main inputs to our smartphones are not only text but through voice also. We can give the users the ability to record their voices as a review which will be further used for text generation and text analysis. The previous process can be expressed by the following diagram:

Voice Review

Text Generation

Review Analysis

### Sophisticated Sentiment Search Technique

Some users might not be able to toggle through different categories. From that perspective, we thought about providing the ability to search for a specific product using a full sentence including their requirements. In simple words, the text analysis system can also be applied to the searched keyword rather than only to reviews.

e.g.

Search: “I need a fast laptop”

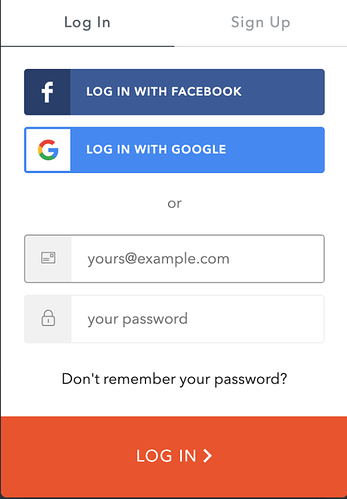
Show results matching the keyword(s)

I need a ***fast*** laptop

Text Analysis

### OAuth implementation

A convenient way of signing in/up could be implemented using Google Identity platform (API) or Facebook login (API).



# Appendices

이번 챕터에서는 앞서 본문에서 다루지 못한 시스템 개발에 관련된 세부 정보를 다룬다. 시스템 운영 환경에 필요한 하드웨어 요구사항, 시스템에 사용되는 데이터베이스의 개략적 구조 등에 대해 기술한다.

## Hardware requirements

본 시스템은 HTML 5, ES 6 등 최신 웹 기술을 사용한 웹 애플리케이션으로 개발되고 있다. 그러므로 본 시스템을 사용하기 위해서는 HTML 5, ES 6 등을 지원하는 최신 웹브라우저를 갖추어야 하며, 백엔드 서버에서 상품 목록과 리뷰 목록, 추천 카테고리 등을 업데이트하므로 항상 네트워크에 연결되어 있어야 한다. 또한 본 시스템은 모바일 접속 환경을 위해 Progressive Web App 기술을 적용해 웹브라우저에 주소를 입력하지 않고도 네이티브 애플리케이션처럼 작동하는 사용자 경험을 제공하는데, 이를 사용하기 위해서는Progressive Web App을 지원하는 최신 운영체제와 브라우저가 필요하다.

## Database requirements

본 시스템은 RDBMS를 사용해 데이터를 저장한다. 데이터베이스를 이루는 각 엔티티는 다음과 같다. 세부적인 데이터베이스 설계는 Design Architecture 문서에서 다루게 된다.

### User

시스템의 사용자 정보를 담고 있는 엔티티

### Item

상품의 정보를 담고 있는 엔티티

### Review

상품 리뷰의 정보를 담고 있는 엔티티

### Authority

사용자의 권한 정보를 담고 있는 엔티티

### RecommendCategory

추천 카테고리에 대한 정보를 담고 있는 엔티티

### Bookmark

사용자가 특정 상품에 대해 추가한 북마크에 대한 정보를 담고 있는 엔티티

### Reference

상품 리뷰의 출처 사이트에 대한 정보를 담고 있는 엔티티

### Keyword

검색, 추천 시스템에 필요한 키워드에 대한 정보를 담고 있는 엔티티

# 10. Index

## 10.1. Tables

[Table 1: Glossary 16](#_Toc9286211)

[Table 2: Ranking Function 29](#_Toc9286212)

[Table 3: Show Item Detail 30](#_Toc9286213)

[Table 4: Add Item to Bookmark 30](#_Toc9286214)

[Table 5: Show Recommendation List 31](#_Toc9286215)

[Table 6: Add Recommendation Category 32](#_Toc9286216)

[Table 7: Mypage 33](#_Toc9286217)

[Table 8: Login Function 34](#_Toc9286218)

[Table 9: Review Collecting System 35](#_Toc9286219)

[Table 10: Review Analysis System 35](#_Toc9286220)

[Table 11: Review Database Updater 36](#_Toc9286221)

## 10.2. Figures

[Figure 1: 온라인쇼핑 거래액 추이 10](#_Toc9286222)

[Figure 2: 소비자 리뷰로 인한 제품 구매 실패 경험 11](#_Toc9286223)

[Figure 3: 소비자 리뷰 종합 설문조사 12](#_Toc9286224)

[Figure 4: 시스템 목업 – Ranking 13](#_Toc9286225)

[Figure 5: 시스템 목업 – Recommendation 14](#_Toc9286226)

[Figure 6: User Requirement - Ranking 18](#_Toc9286227)

[Figure 7: User requirement - Item Detail 19](#_Toc9286228)

[Figure 8: User requirement – Recommendation 20](#_Toc9286229)

[Figure 9: Recommendation System Architecture 27](#_Toc9286230)

## 10.3. Diagrams

[Diagram 1: Login process 17](#_Toc9286231)

[Diagram 2: Signup process 17](#_Toc9286232)

[Diagram 3: Frontend Architecture 24](#_Toc9286233)

[Diagram 4: Backend Architecture 25](#_Toc9286234)

[Diagram 5: Review Analysis System Architecture 26](#_Toc9286235)

[Diagram 6: Overall context diagram 42](#_Toc9286236)

[Diagram 7: Overall process diagram 43](#_Toc9286237)

[Diagram 8: Use case diagram 43](#_Toc9286238)

[Diagram 9: Data flow diagram of ranking system 48](#_Toc9286239)

[Diagram 10: Data flow of text analysis system 48](#_Toc9286240)

[Diagram 11: Sequence Diagram of Mypage Function 49](#_Toc9286241)

# 11. References

(김준영유도영,, 2019)

해당repository의 requirement문서의 형식을 기반으로 작성했습니다. (<https://github.com/skkuse/2019spring_41class_team5/blob/master/docs/requirement.docx>)