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)

This chapter defines the expected readership of the document, and briefly introduces the content of each chapter. This chapter also describes version history including a rationale for the creation of a new version and a summary of the changes made in each version.

## Readership

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

### User Requirements Readership

User requirements의 예상 독자는 본 시스템의 사용자이다. 해당 챕터는 사용자의 관점에서 요구사항을 알기 쉽게 설명하기 위해 전문 용어의 사용을 자제하고, 도표 등의 시각자료를 동반하여 자연어로 서술한다.

### System Requirements Readership

System requirements는 시스템의 기능과 제약 사항을 구조적 언어로 표현한 것으로서, 실제 개발 과정에서 참고 자료로 사용하거나, 고객과의 계약서에서 사용되어질 수 있는 수준으로 체계적으로 서술한다.

## Document Structure

1. Introduction  
   개발하고자 하는 시스템을 둘러싼 니즈를 설명하고, 시스템의 개략적인 구조와 기능에 대해 설명한다. 또한 시스템을 개발함으로써 얻을 수 있는 목표 효과에 대해서 설명한다.
2. Glossary  
   문서 전반에서 사용되고 있는 기술적 용어들에 대해 정의하고, 해당 용어가 어떤 맥락에서 사용되는지 기술한다. 본 문서는 시스템 개발자뿐 아니라 사용자, 이해당사자 등의 독자 또한 대상으로 포함하고 있으므로, 가능한 자세하게 용어에 대해 설명한다.
3. User Requirements Definition  
   시스템의 기능적, 비 기능적 요구사항을 사용자의 입장에서 설명한다. 간략한 도식과 함께 시스템의 사용자가 문서를 이해하기 쉽도록 자연어를 사용하여 서술한다.
4. System Architecture  
   시스템의 구조를 개괄적으로 기술하고, 시스템의 기능이 각 서브시스템에 어떻게 할당되어 분포하는지를 설명한다.
5. System Requirements Specification  
   User Requirements Definition에 간략히 서술되어 있는 요구사항을 기반으로, 기능적 요구사항과 비 기능적 요구사항, 기타 요구사항을 자세히 서술한다. 이때 본 챕터는 시스템 설계 단계와 구현 단계를 포함한 개발 프로세스 전반에서 사용되어야 하므로 도표 등을 사용해 체계적으로 표현한다.
6. System Models  
   시스템의 각 컴포넌트간의 관계, 시스템을 둘러싼 외부 환경과의 관계 등을 다이어그램으로 표현한다.
7. System Evolution  
   시스템의 개발상 한계에 대해 서술하고, 시스템의 운영 과정에서 발생할 수 있는 환경의 변화를 예상하고 그에 대한 대응 방안을 서술한다.
8. Appendices  
   본문에서 생략된 참고자료 등을 기술한다. 하드웨어, 데이터베이스 요구사항, 개발 환경 요구사항 등이 포함되어 있다.
9. Index  
   본 문서에서 사용된 그림, 표, 다이어그램 등의 색인을 기술한다.

# Introduction(우리 팀의 프로젝트 배경, 솔루션, 기대효과 제시 proposal참고)

이 챕터에서는 본 시스템을 둘러싼 니즈를 설명하고, 시스템의 기능을 비롯해 본 시스템이 다른 시스템과 어떻게 상호작용하는지를 간략히 서술한다.

## 2.1. Needs

온라인 쇼핑 시장의 규모는 계속해서 성장하고 있다. G마켓, 11번가, 옥션 등 기존의 전통적인 오픈마켓 빅3를 비롯해 최근에는 마켓컬리, 쿠팡 등 소셜커머스계 신흥 업체들이 떠오르며 마켓쉐어를 두고 치열한 경쟁을 벌이고 있다. 이에 따라 업체들은 기존에는 온라인 쇼핑의 영역으로 인식되지 않았던 식료품 분야까지 진출하는 등 다양한 방법으로 차별화를 시도하고 있다.

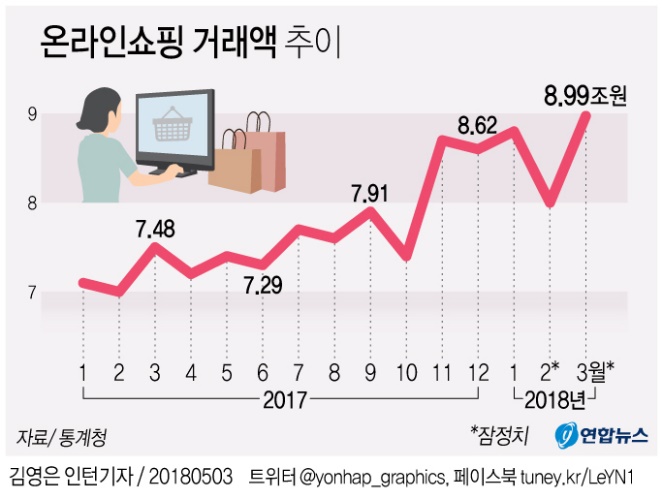


Figure : 온라인쇼핑 거래액 추이

점점 커지는 시장 규모에도 불구하고 온라인 쇼핑이 가진 가장 큰 단점은 상품을 직접 눈으로 확인하고 구매할 수 없다는 데 있다. 따라서 소비자들은 상품의 품질을 확인하는 데 있어 간접적인 정보를 활용할 수밖에 없다. 오랜 기간 쌓인 신뢰 자원이 존재하는 일부 대기업 브랜드를 제외한 제품들은 업체의 신뢰도만으로 제품의 품질을 보장할 수 없으므로 인터넷 검색이나 입소문 등 다른 정보를 활용해 구매를 결정해야 한다. 쇼핑몰에서 제공하는 소비자 리뷰 페이지는 구매에 영향을 미치는 중요한 정보 중 하나인데, 많은 소비자들은 다른 소비자들이 제품에 남긴 리뷰와 평점을 기준으로 상품을 선택하고 있다.

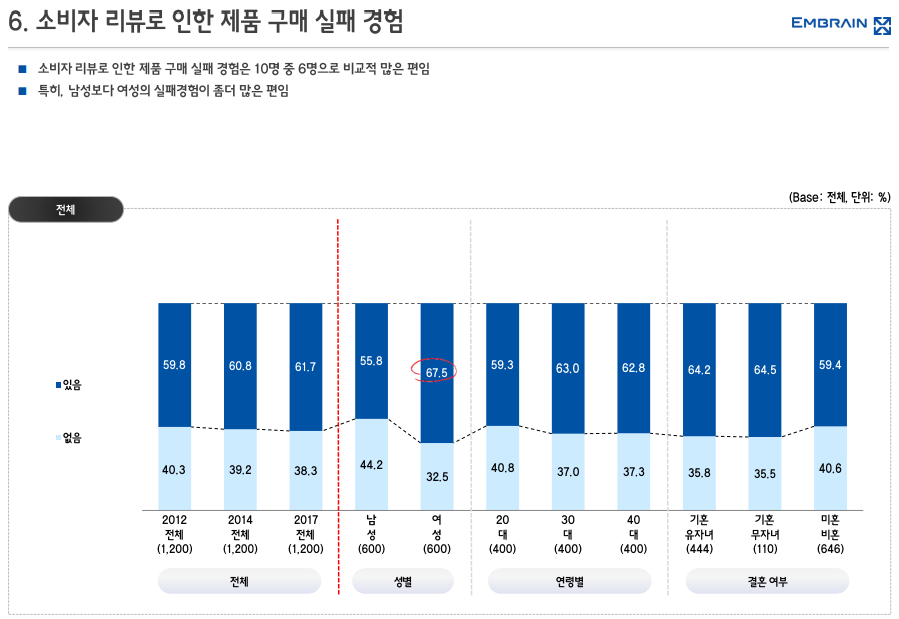


Figure : 소비자 리뷰로 인한 제품 구매 실패 경험

그러나 이러한 소비자 리뷰에 기반한 상품 구매는 리스크를 안고 있는데, 업체의 홍보성 리뷰 등에 의해 평가가 왜곡되어 제품에 대한 부정확한 정보를 얻을 수 있기 때문이다. ㈜마크로밀엠브레인에서 조사한 2017 소비자 리뷰 영향력 조사 보고서를 보면, 소비자 리뷰를 믿고 제품을 구매했다가 실패한 경험이 있다는 응답이 전체 조사대상의 약 60%를 웃도는 것으로 나타났다. 이러한 실패 경험은 시간이 지날수록 늘어나고 있다. (김윤미, 채선애, 송으뜸, 윤덕환, 2017)

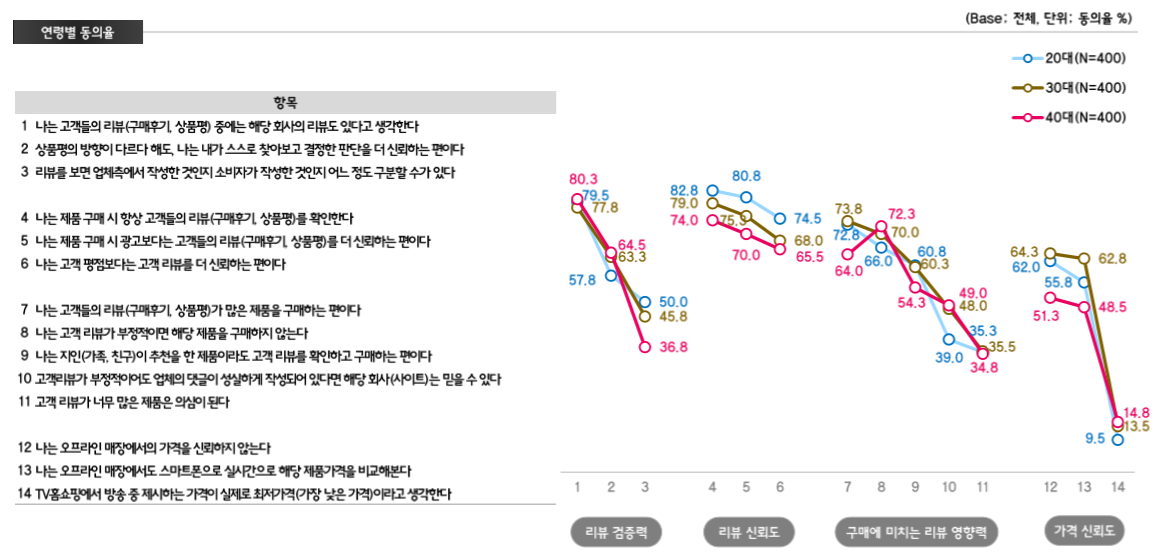


Figure : 소비자 리뷰 종합 설문조사

업체의 홍보성 리뷰에 의해 소비자의 전반적인 리뷰 신뢰도가 낮아지고 있는 상황에서 소비자가 리뷰의 신뢰성을 판별할 수 있는 능력은 성공적인 제품 구매에 있어 매우 중요하다. 그러나 이러한 능력을 모든 소비자가 가지고 있지는 않다. 마찬가지로 ㈜마크로밀엠브레인의 보고서에 따르면, 소비자들은 제품 구매 시에 대부분 소비자 리뷰를 확인하지만, 리뷰를 검증하는 능력이 전 연령대에서 50% 이하로 나타났고, 특히 고연령대로 올라갈수록 리뷰 검증 능력이 떨어져 40대에서는 36.8%로 나타났다.[[1]](#footnote-1) (김윤미, 채선애, 송으뜸, 윤덕환, 2017)

오픈 마켓 리뷰 페이지에서 유용한 리뷰를 가려내 주는 시스템은 상품 구매뿐 아니라 쇼핑몰의 신뢰성에도 영향을 미치는 중요한 요소이다. 그러나 대부분의 오픈 마켓 리뷰 페이지는 소비자 친화적이지 않다. 제대로 된 필터링 시스템 없이 단순한 추천 시스템이나, 사진 첨부 여부만을 통해 리뷰를 정렬하므로 소비자들은 수많은 리뷰를 직접 읽어 가면서 상품을 판단하는 실정이다. 이 와중에, 상품의 품질과 무관한 배송 과정에서의 불만을 상품 리뷰에 올리거나, 스팸성 리뷰가 페이지를 도배하는 등 정보값 없는 리뷰의 범람으로 소비자들의 판단을 힘들게 하고 있다.

따라서 소비자들의 판단을 돕기 위해 유용하지 않은 정보를 걸러내고 신뢰성 있는 평점 시스템을 제공하는 시스템의 개발이 필요해졌다.

## System Overview

Dealistic은 오픈 마켓에 존재하는 리뷰 정보를 활용해 자연어 처리 기술을 이용하여 특정 상품에 대해 사용자들이 주로 언급한 키워드를 추출, 사용자들이 어떤 뉘앙스로 리뷰를 남겼는지를 분석하고 추천 정도를 계량화하여 보다 정확한 평점을 매기는 서드 파티 리뷰 시스템이다. 또한 자체 평점에 기반해 상품의 순위를 매기고, 사용자가 자신의 취향을 입력하면 적합한 상품을 추천하는 기능 또한 갖추고 있다. 이를 통해 사용자는 모든 리뷰를 읽고 검증할 필요 없이 일목요연하게 정리된 리뷰와 평점을 보고 보다 정확한 판단을 할 수 있다.

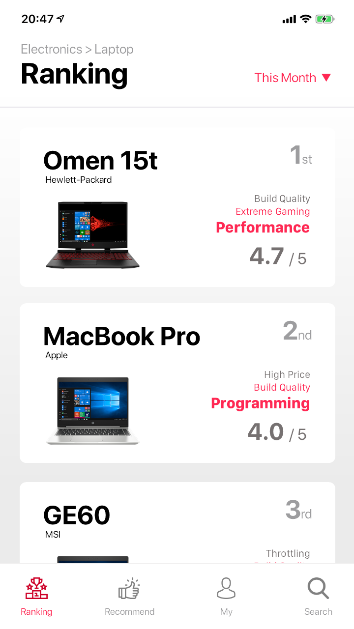
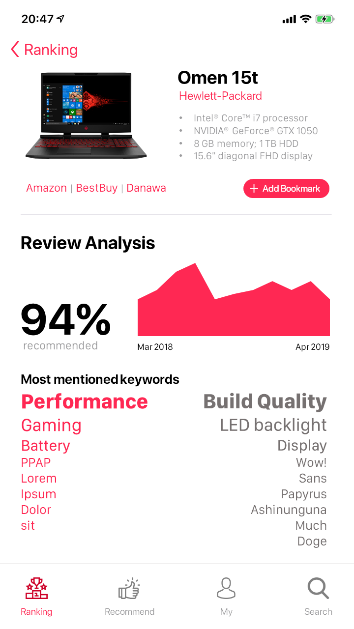


Figure : 시스템 목업 – Ranking

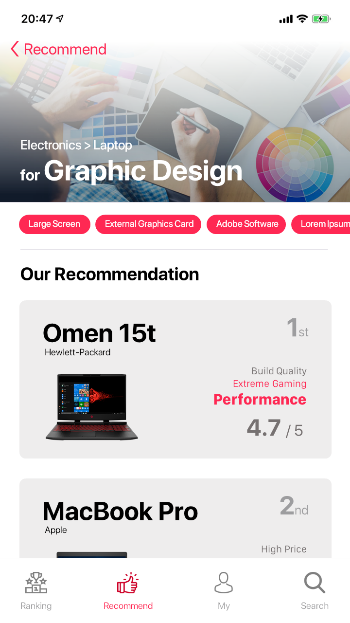
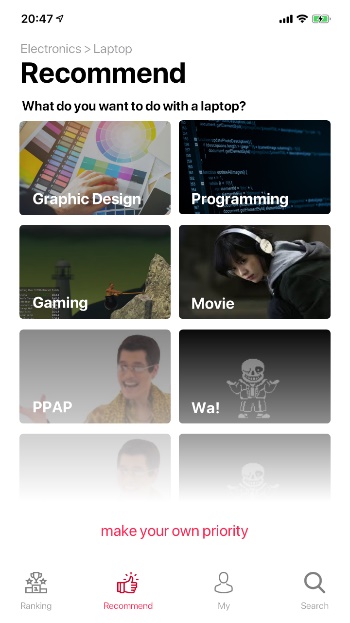


Figure : 시스템 목업 – Recommendation

Dealistic의 주요 기능은 두 가지인데, 특정 카테고리에 대해 사용자 리뷰에 기반한 순위를 보여주고 각 상품들이 어떤 평가를 받는지를 보여주는 ‘Ranking’ 기능과, 사용자의 관심사를 입력하면 각 관심사를 기반으로 어떤 상품이 사용자에게 가장 좋은 평가를 받고 있는지 추천해 주는 ‘Recommendation’ 기능이다.

## Expected Effects

1. 소비자 정보 접근 불균형 해소

기존의 단일 기준 정렬에 기반해 소비자가 일일이 리뷰를 확인해 가면서 유용성을 판단하는 프로세스는 매우 비효율적이고 소비자의 경험과 기반 지식에 따라 획득할 수 있는 정보의 질과 양에 차이가 매우 심각하다. 그러나 본 시스템을 사용할 경우 사용자들이 남긴 리뷰를 자동으로 분석해 소비자의 관심사과 중요도에 따라 주요 키워드를 나열해 주고, 그에 기반한 상품 점수 표시를 통해 쇼핑 경험에 무관하게 객관적이고 정확한 상품 리뷰를 확인할 수 있게 되었다.

1. 판매자 광고 비용 절감

오픈마켓은 우선적으로 자사와 제휴해 광고료를 지불하는 업체의 상품을 우선적으로 노출하기 때문에 판매자 입장에서는 상품 판매 실적을 높이기 위해서 광고료를 지불할 수밖에 없고, 이는 수익성 악화로 이어진다. 그러나 본 시스템은 사용자 리뷰를 기반으로 높은 점수를 얻은 순서대로 상품을 노출하므로 판매자가 광고 비용을 지불할 필요가 없다.

# 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 : Glossary

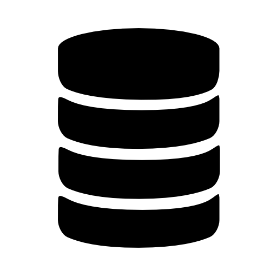
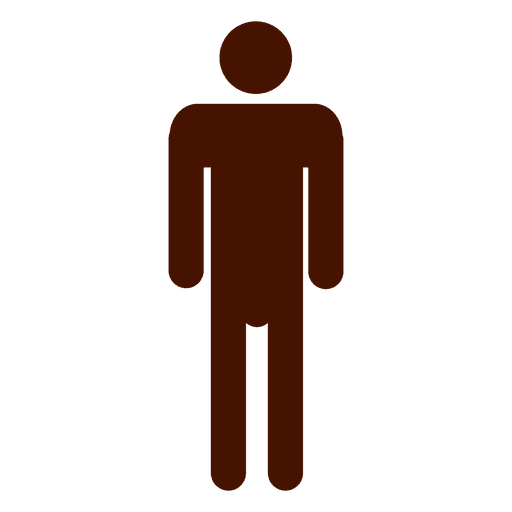
# User Requirements Definition(우리 팀의 내용으로 대체 10/30일까지)

본 챕터에서는 시스템이 사용자에게 제공하는 서비스에 대해 서술하고, 사용자가 시스템을 사용하는 데 있어 중요한 비기능적 요구사항에 대해 설명한다. 이번 챕터의 독자는 사용자이므로 전문지식이 없는 독자도 이해할 수 있도록 기술용어의 사용을 자제하고, 자연어와 시각자료를 활용해 각 요구사항에 대해 설명한다.

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

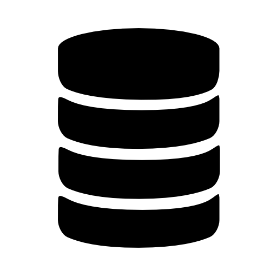
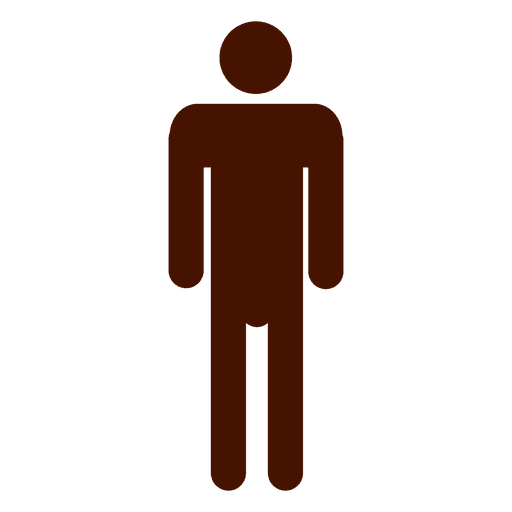
### Sign up/Login

This requirement is one the most fundamental requirements where users can get on our platform using a specific ID and Password. Before joining the platform and starting using DEAListic, users have register for a membership using a unique ID. This information along with some other personal information will be stored in our data base. Later on, when the users want to login the system will authenticate the given (ID, PW) pair and try match it with a one from the data base.



ID, PW

Diagram : Login process



ID, Password,  
Personal Information

Diagram : Signup process

### Ranking

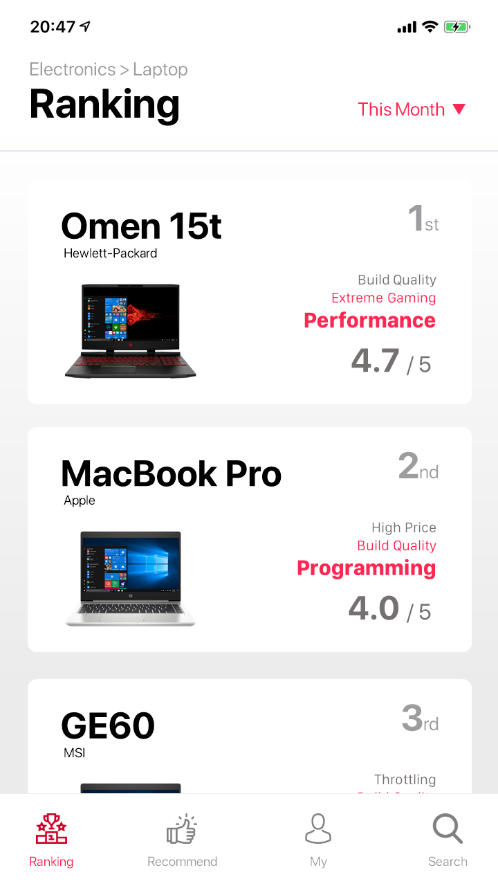


Figure 6: User Requirement - Ranking

This requirement is aimed for making ranking lists from review of products in Dealistic. By this feature, customer can easily check good quality of product and it can be motivation to sellers to improve their products by competing ranking. We will make specific parts of ranking by each category and represent its ranking on the lists. First, get the data from word processing of review and then save it to database of ranking list. And show the ranking by descending order on the page.

### item detail

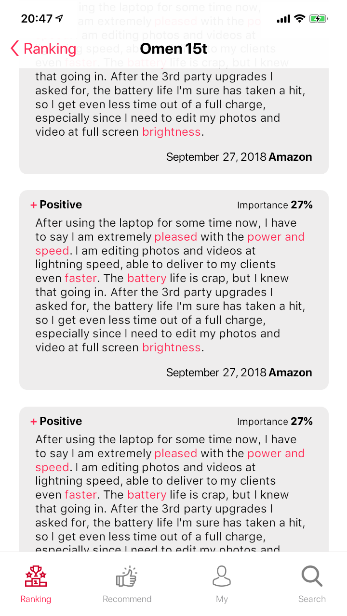
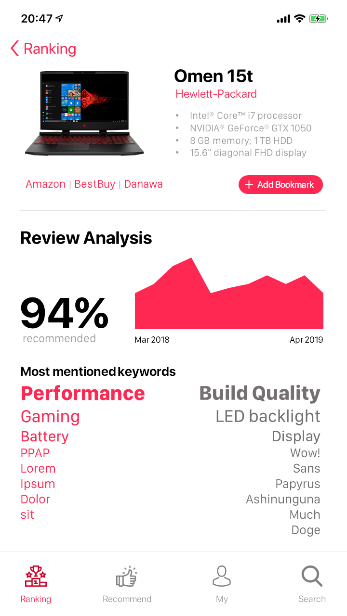


Figure : User requirement - Item Detail

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.

### Recommendation

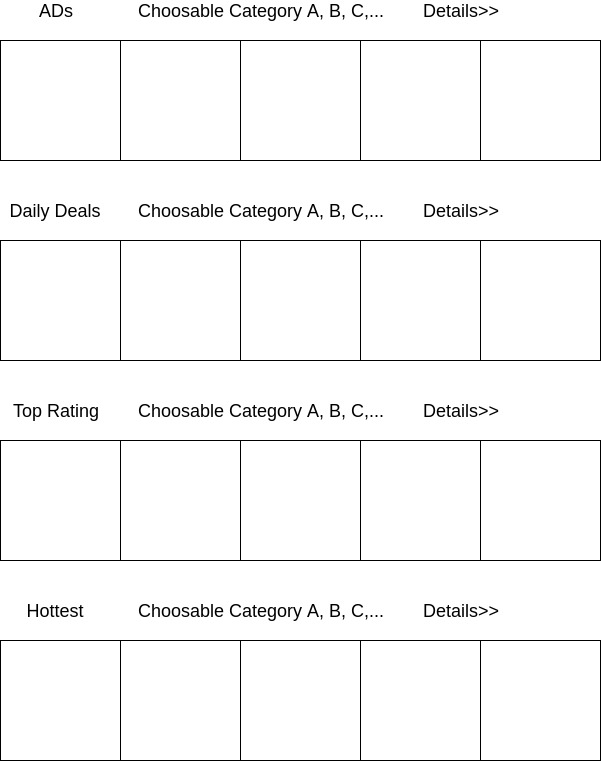


Figure 8: User requirement – Recommendation

### Mypage

개인화된 사용자별 페이지이다. 이 페이지를 통해서 사용자의 개인 정보 관리, 자신이 작성한 상품평 관리, 북마크 등의 기능이 처리될 수 있다. 따라서 이 페이지는 로그인한 사용자만 접근 가능하다.

### Search

사용자는 이 기능을 통하여 상품을 검색할 수 있다. 검색 조건은 이름, 카테고리, 스펙 등 다양한 조건이 될 수 있으며, 목록을 점수, 리뷰 많은 순, 가격 등 다양한 조건으로 재정렬이 가능해야 한다. 상품을 누르면 해당 상품의 자세한 정보를 담은 페이지로 연결된다.

## 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

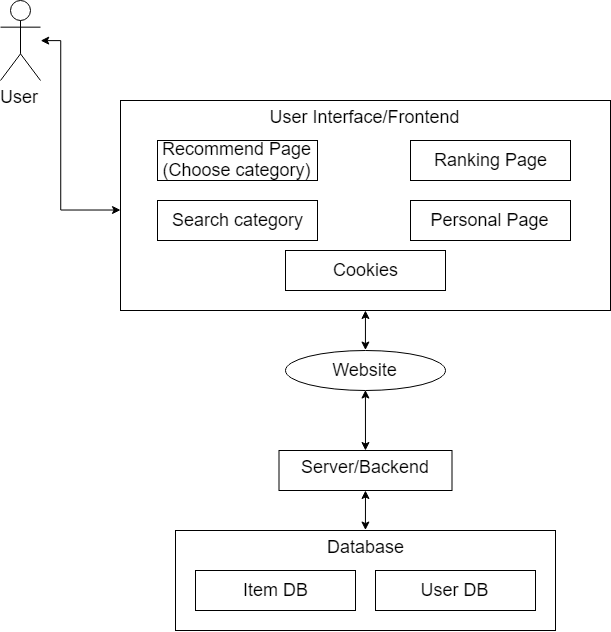


Diagram : Frontend Architecture

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

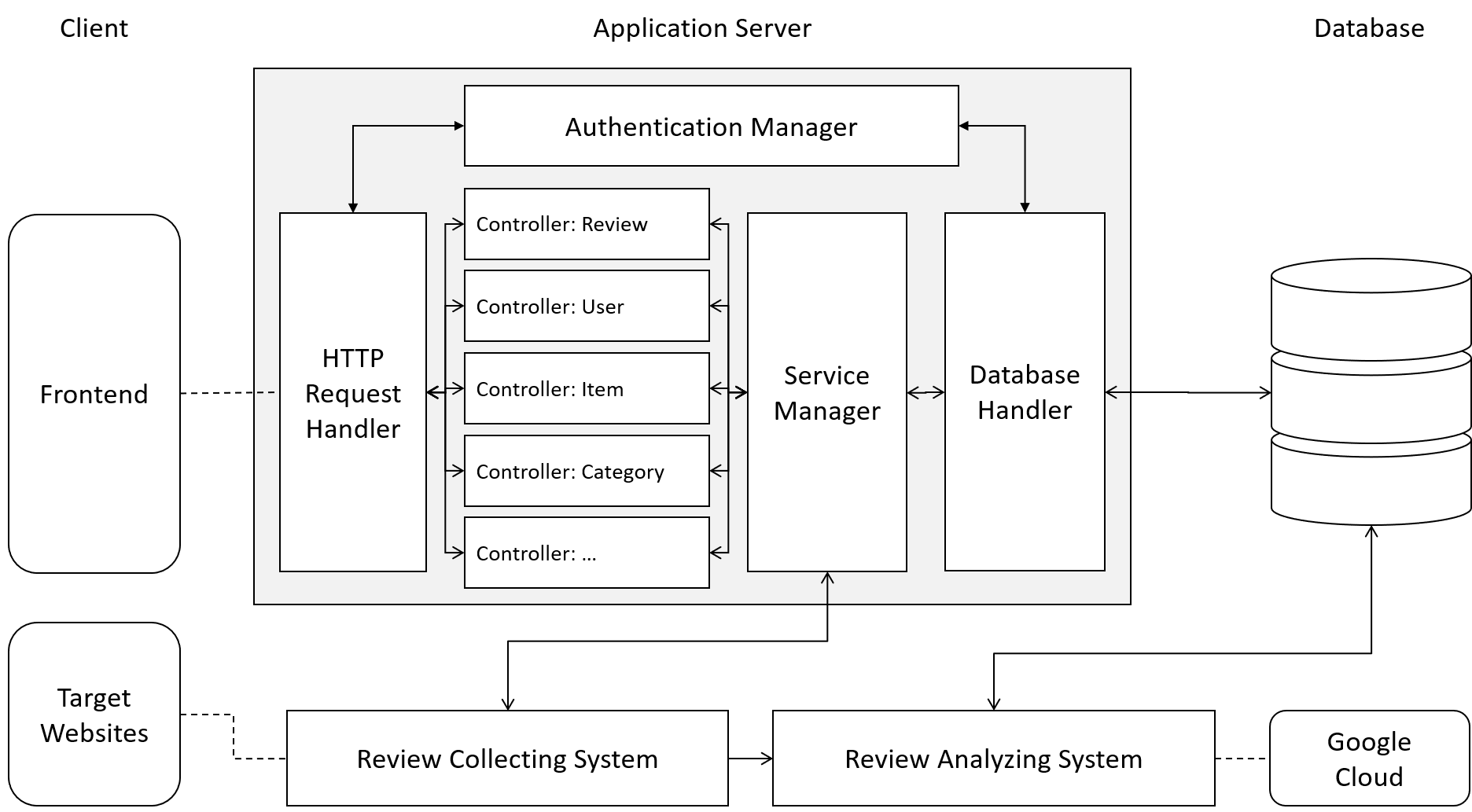


Diagram : Backend Architecture

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

## Review Analysis System

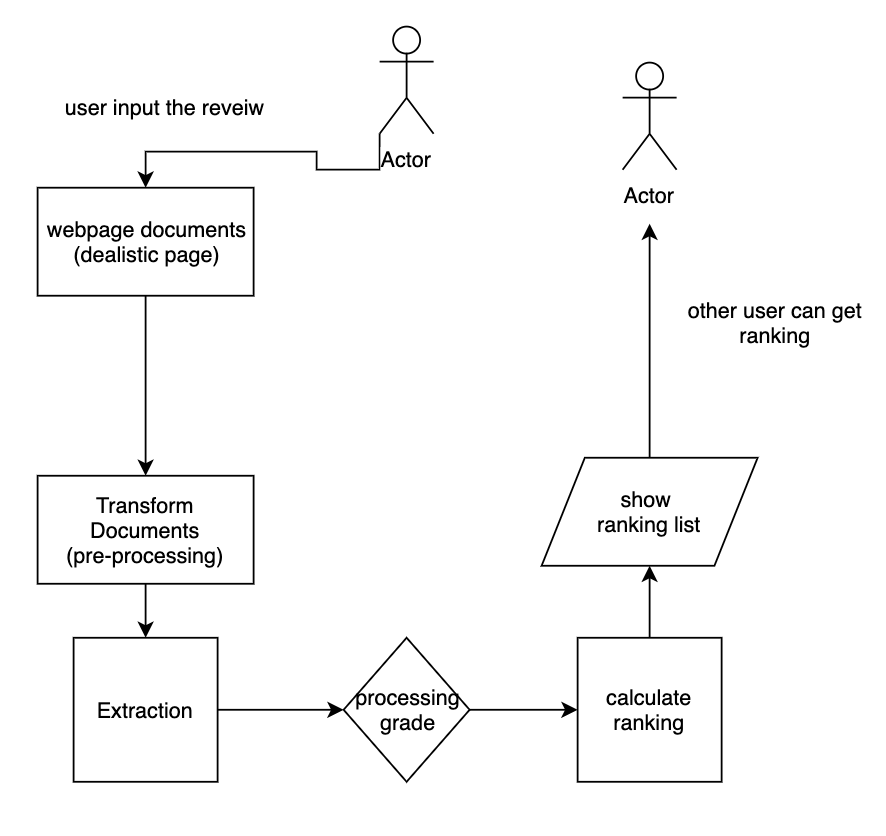


Diagram 5: Review Analysis System Architecture

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

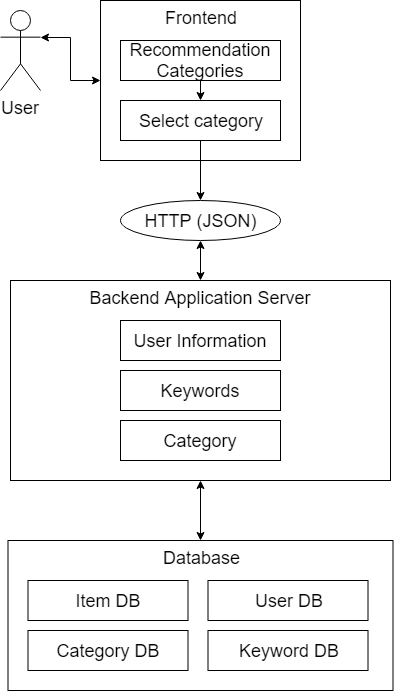


Figure 9: Recommendation System Architecture

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 : 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

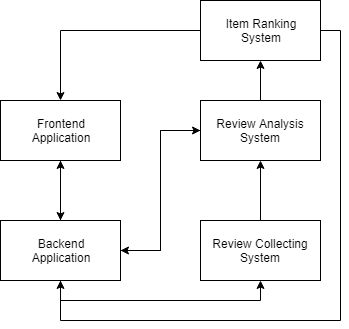


Diagram : Overall context diagram

### B. Process Diagram

스크린샷이(가) 표시된 사진

자동 생성된 설명

Diagram : Overall process diagram

## 7.2. Interaction models

### Use case Diagram

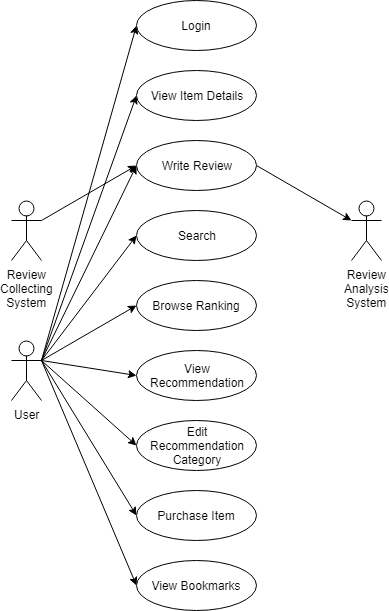


Diagram : Use case diagram

### 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

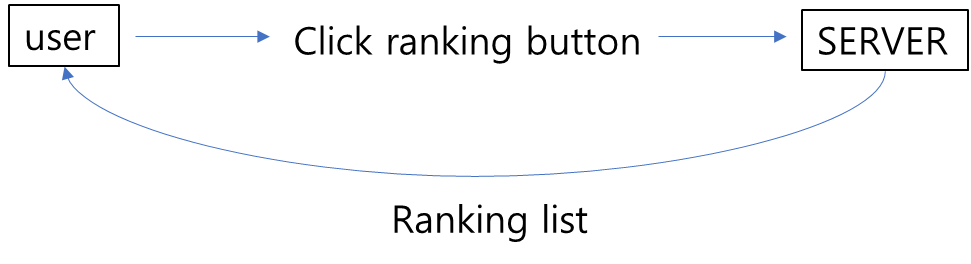


Diagram 9: Data flow diagram of ranking system

### Review Analysis DFD

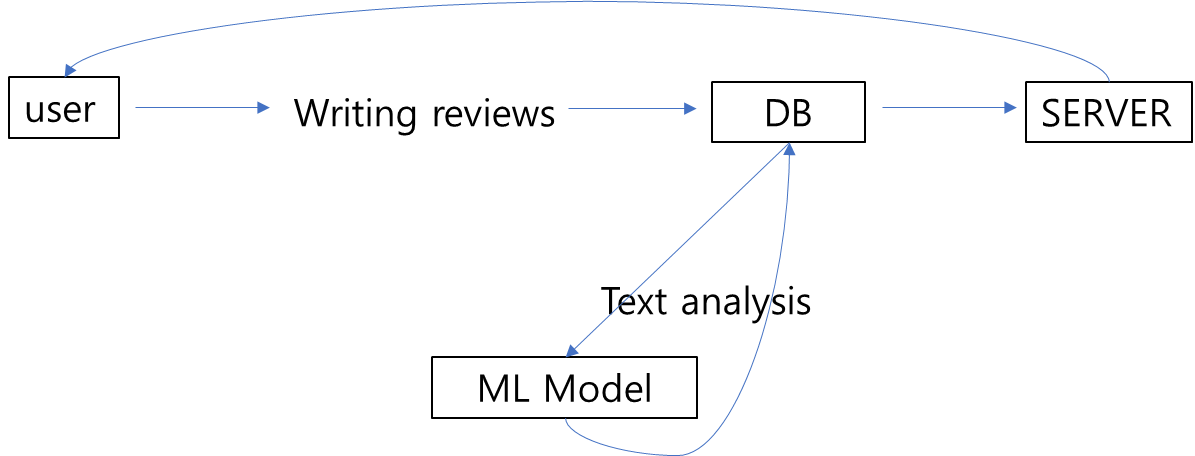


Diagram 10: Data flow of text analysis system

### MyPage Sequence Diagram.

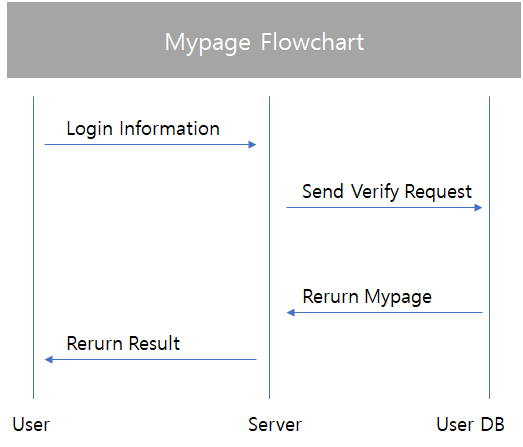


Diagram 11: Sequence Diagram of Mypage Function

# 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”

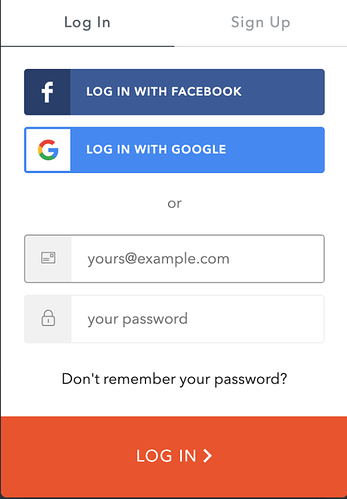
Show results matching the keyword(s)

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>)

1. 또한 흥미로운 점은 소비자가 제품 평점보다 직접 글로 작성된 리뷰를 더 신뢰한다는 점이다. 스팸성 리뷰로 인해 평점이 왜곡되어 있고, 대부분의 소비자가 제품의 장단점을 나열하는 데는 익숙하지만 제품을 정량적으로 평가할 기준이 없다는 것도 제품 평점의 신뢰성을 낮추는 데 일조했을 것으로 예상된다. [↑](#footnote-ref-1)