Software Engineering Final Project Team D

**Clothes Shopping Mall using Deep Learning-Requirements Specification**

아래와 같은 수정사항이 있었습니다.

수정일: 2016년 12월 7일

수정 내용:

1. 기존에 제공하고자 했던 Collection 기능 본 문서에서 삭제.
2. 클릭 수 분석을 통한 고객맞춤형 상품 추천 functional requirements에서 삭제 후, system evolution-environment evolution 부분에 추가됨.
3. 위 1, 2번 기능 삭제에 따른 Introduction - 배경 및 필요성 수정.
4. 1, 2번 수정에 따른, 문서 이음새 수정.
5. Class Diagram에 Image Analysis Class Diagram 추가.
6. Appendices 2 에 Deep Learning 관련 내용 추가.
7. Clothes shopping mall 에서 모든 shopping mall solution으로 확장함.

수정본 제출일 : 2016년 12월 7일

Version 7.0.0

**Shopping Mall Solution**

**using Deep Learning**

**Requirements Specification**

Software Engineering(ICE3041(41))

D - 임동욱, 신재민, 송종훈, 장정수

Contents

[**1** **Preface** 8](#_Toc468756964)

[1.1 Objectives 8](#_Toc468756965)

[1.2 Readership 8](#_Toc468756966)

[A. User Requirements 8](#_Toc468756967)

[B. System Requirements 8](#_Toc468756968)

[1.3 Document Structure 8](#_Toc468756969)

[A. Preface 8](#_Toc468756970)

[B. Introduction 8](#_Toc468756971)

[C. Glossary 9](#_Toc468756972)

[D. User Requirements Definition 9](#_Toc468756973)

[E. System Architecture 9](#_Toc468756974)

[F. System Requirements Specification 9](#_Toc468756975)

[G. System Models 9](#_Toc468756976)

[H. System Evolution 9](#_Toc468756977)

[I. Appendices 9](#_Toc468756978)

[J. Index 9](#_Toc468756979)

[1.4 Version of the Documents 10](#_Toc468756980)

[A. Version Format 10](#_Toc468756981)

[B. Version Management Policy 10](#_Toc468756982)

[C. Version Update History 10](#_Toc468756983)

[**2** **Introduction** 11](#_Toc468756984)

[2.1 Objectives 11](#_Toc468756985)

[2.2 배경 및 필요성 11](#_Toc468756986)

[2.3 기대효과 12](#_Toc468756987)

[**3** **Glossary** 12](#_Toc468756988)

[3.1 Objectives 12](#_Toc468756989)

[3.2 Term Definition 13](#_Toc468756990)

[**4** **User Requirements Definition** 15](#_Toc468756991)

[4.1 Objectives 15](#_Toc468756992)

[4.2 Functional Requirements 15](#_Toc468756993)

[A. 이미지 분석을 통한 태깅 및 카테고리화 15](#_Toc468756994)

[B. 관련 상품 추천 15](#_Toc468756995)

[C. 데이터 기록 및 저장 16](#_Toc468756996)

[4.3 Non-Functional Requirements 16](#_Toc468756997)

[A. Product Requirements 16](#_Toc468756998)

[B. Organizational Requirements 17](#_Toc468756999)

[C. External Requirement 17](#_Toc468757000)

[**5** **System Architecture** 18](#_Toc468757001)

[5.1 Objectives 18](#_Toc468757002)

[5.2 Overview 18](#_Toc468757003)

[5.3 상세 시스템 구조 19](#_Toc468757004)

[**6** **System Requirements Specification** 20](#_Toc468757005)

[6.1 Objectives 20](#_Toc468757006)

[6.2 Functional Requirements 20](#_Toc468757007)

[A. 이미지 분석을 통한 태깅 및 카테고리화 20](#_Toc468757008)

[B. 관련 상품 추천 20](#_Toc468757009)

[C. 데이터의 기록 및 저장 21](#_Toc468757010)

[6.3 Non-functional Requirements 22](#_Toc468757011)

[A. Product Requirements 22](#_Toc468757012)

[B. Organizational Requirements 23](#_Toc468757013)

[C. External Requirement 23](#_Toc468757014)

[6.4 Scenario 23](#_Toc468757015)

[A. 판매자 Scenario (Scenario A) 23](#_Toc468757016)

[B. 구매자 Scenario (Scenario B) 25](#_Toc468757017)

[**7** **System Models** 26](#_Toc468757018)

[7.1 Objectives 26](#_Toc468757019)

[7.2 Class Diagram 28](#_Toc468757020)

[A. Overview 28](#_Toc468757021)

[B. Sign\_in Diagram 30](#_Toc468757022)

[C. Search Diagram 32](#_Toc468757023)

[D. Board Diagram 35](#_Toc468757024)

[E. ADM Diagram 37](#_Toc468757025)

[F. Image Analysis Diagram. 39](#_Toc468757026)

[7.3 Sequence Diagram 40](#_Toc468757027)

[A. Sign\_up Diagram 40](#_Toc468757028)

[B. Sign\_in Diagram 40](#_Toc468757029)

[C. Search Diagram 41](#_Toc468757030)

[D. Cart Diagram 42](#_Toc468757031)

[E. Purchasing Cart Diagram 42](#_Toc468757032)

[F. Purchasing Item Diagram 43](#_Toc468757033)

[G. Purchasing and Payment Diagarm 43](#_Toc468757034)

[H. Board Diagram 44](#_Toc468757035)

[I. Mypage Diagram 45](#_Toc468757036)

[J. ADM Diagram 45](#_Toc468757037)

[K. Item Add and Correction Diagram 46](#_Toc468757038)

[L. User Management Diagram 46](#_Toc468757039)

[M. Order Management Diagram 47](#_Toc468757040)

[7.4 State Diagram 48](#_Toc468757041)

[A. Overview Diagram 48](#_Toc468757042)

[B. Sign\_up Diagram 48](#_Toc468757043)

[C. Sign\_in Diagram 49](#_Toc468757044)

[D. Search Diagram 49](#_Toc468757045)

[E. Choose\_Item Diagram 50](#_Toc468757046)

[F. Cart Diagram 50](#_Toc468757047)

[G. Purchasing Diagram 51](#_Toc468757048)

[H. Purchasing Cart Diagram 51](#_Toc468757049)

[I. Payment Diagram 52](#_Toc468757050)

[J. Shipment Diagram 52](#_Toc468757051)

[K. Mypage Diagram 53](#_Toc468757052)

[L. Board Diagram 53](#_Toc468757053)

[M. ADM Diagram 54](#_Toc468757054)

[i. Overall Diagram 54](#_Toc468757055)

[ii. Item Management 54](#_Toc468757056)

[iii. User Management Diagram 55](#_Toc468757057)

[**8** **System Evolution** 55](#_Toc468757058)

[8.1 Objectives 55](#_Toc468757059)

[8.2 Evolution of Hardware 55](#_Toc468757060)

[A. 서버 55](#_Toc468757061)

[8.3 Evolution of User Requirement 56](#_Toc468757062)

[A. 상세화 된 태그 분류에 대한 요구 56](#_Toc468757063)

[B. 보안에 대한 요구 56](#_Toc468757064)

[8.4 Evolution of Environment 57](#_Toc468757065)

[A. 개인화 서비스의 필수화 57](#_Toc468757066)

[B. 딥 러닝 기술의 보편화 및 상용화 58](#_Toc468757067)

[**9** **Appendices** 58](#_Toc468757068)

[9.1 Objectives 58](#_Toc468757069)

[9.2 데이터베이스 요구사항 58](#_Toc468757070)

[9.3 사용자 시스템 요구사항 59](#_Toc468757071)

[A. 최소 시스템 사양 59](#_Toc468757072)

[9.4 적용 개발 프로세스 60](#_Toc468757073)

[9.5 딥 러닝(Deep Learning) 61](#_Toc468757074)

[**10** **Index** 63](#_Toc468757075)

[10.1 Objectives 63](#_Toc468757076)

[10.2 Alphabet Index 63](#_Toc468757077)

[10.3 Figure Index 64](#_Toc468757078)

[10.4 Diagram Index 65](#_Toc468757079)

[**11** **Reference** 67](#_Toc468757080)

1. **Preface**
   1. Objectives

Preface에서는 본 문서의 독자를 밝히고, 문서의 구조와 파트 별 역할에 대해 소개하며, 문서의 작성 및 수정 과정에 대해 서술한다.

* 1. Readership

본 문서는 크게 User requirements와 System requirements로 나누어져 있다. User requirements는 본 문서가 설명하는 시스템(System)을 사용하는 사용자의 관점에서 요구사항과 제약사항을 간략히 서술한 것이며, System requirements는 system developer의 관점에서 요구사항과 제약사항을 상세히 서술한 것이다.

* + 1. User Requirements

주 독자로는 Client와 Contractor Manager가 있으며, 시스템에 대한 전문지식이 없는 이들을 위해 자연어(Natural Language)와 그림(Diagram)을 이용해 이해를 돕는다.

* + 1. System Requirements

시스템의 기능, 서비스, 구조, 제약사항을 상세히 서술한 것으로, 주 독자로는 Software Developer와 System Architecture가 있으며, 상황에 따라 Contractor Manager, Client에게도 제공되어 계약서 작성 시 도움이 되도록 한다.

* 1. Document Structure

본 문서는 총 10개의 파트로 구성되어 있다-Preface, Introduction, Glossary, User Requirements Definition, System Architecture, System Requirements Specification, System Models, System Evolution, Appendices, Index- 각 10개의 파트는 다음과 같은 역할을 한다.

* + 1. Preface

본 문서의 독자를 밝히고, 본 문서가 어떤 형식으로 구성되어 있는지, 각 파트가 어떤 역할을 하는지 소개한다. 또한 문서의 제작 및 수정 과정에 대해 서술한다.

* + 1. Introduction

목표 시스템을 만들게 된 배경과 필요성, 목표에 대해 서술하며, 해당 시스템의 기능에 대해 간략히 설명한다. 또한, 시스템이 앞으로 Business와 Organization그리고 주위 시스템에 미칠 기대 효과에 대해 서술한다.

* + 1. Glossary

본 문서에서 다루고 있는 용어에 대해 설명함으로써, 본 문서에서 사용하는 용어의 뜻을 명확히 하고, 독자의 이해를 돕는다.

* + 1. User Requirements Definition

사용자가 요구한 요구사항과 제약 사항에 대해 서술한다. 이 때, 서술어는 자연어(Natural Language)를 사용하며, 그림과 표(Diagram), 그리고 다양한 개념(Notation)을 이용해 고객의 이해를 돕는다. 제품이나 프로세스(Process)가 반드시 따라야 하는 표준 또한 서술하며,본 문서에서 만들고자 하는 목표 시스템과 process는 모두 이 요구사항을 기반으로 한다.

* + 1. System Architecture

High-level에서 목표 시스템의 개요 즉, 간략한 구조, function과 그 분포에 대해 서술하며, 재사용이 되는 구성 요소(Component)는 강조한다.

* + 1. System Requirements Specification

목표 시스템의 기능적인 요구사항(Functional requirements)과 비기능적인 요구사항(Non-functional requirements)에 대해 상세히 서술한다. 이 때, 필요에 따라 비기능적인 요구사항에 대한 추가 서술이 들어가며, 목표 시스템과 상호작용을 하는 다른 시스템에 대한 인터페이스(Interface)를 정의한다.

* + 1. System Models

시스템의 구성요소(System Component)와 시스템, 그리고 시스템의 작동 환경(Environment)를 다양한 Graphical System Models를 이용해 시각적으로 표현한다. 이 때 이용되는 모델로는 Object models, Data-flow models, Semantic data models이 있다.

* + 1. System Evolution

사회의 발전, 고객의 요구사항의 변화에 대해 구현된 시스템과 현재 사회적 배경을 기반으로 하여 예상하고, 어떻게 발전시킬 것인가에 대해 그 방향을 제시한다. 목표 시스템에서 추측할 수 있는 모든 변경 사항 및 가정들에 대해 서술한다.

* + 1. Appendices

본 문서에 등장하는 개념들과 목표 시스템과 관련된 구체적인 정보를 제공한다.

* + 1. Index

본 문서에 등장하는 단어, 그림(Figure, Diagram), 표(Chart) 등을 인덱스(Index)를 정리한다.

* 1. Version of the Documents
     1. Version Format

버전 번호는 [major.minor.maintenance]로 표현되며, 본 문서는 0.1.0부터 시작한다.

* + 1. Version Management Policy

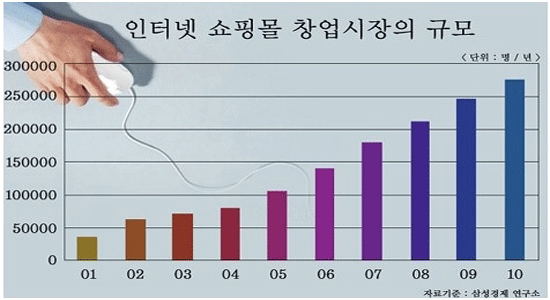
본 문서가 수정될 때 마다, 버전을 업데이트(update)한다. 단, 1시간 이내로 수정을 할 경우 버전 번호를 업데이트를 하지 않고, 하나의 업데이트로 간주한다. 완성된 파트를 수정할 때에는 minor number을 이용하며, 새로운 부분을 추가하거나, 이전 버전에 비해 괄목할 만한 변화가 있는 경우 major number을 이용한다.

* + 1. Version Update History

|  |  |  |
| --- | --- | --- |
| Version | Modified Date | Explanation |
| 0.1.0 | 2016.11.10 | Requirement Specification 초안 작성 |
| 0.1.1 | 2016.11.15 | Page Layout Design |
| 1.0.0 | 2016.11.15 | User requirements 상세화 |
| 2.0.0 | 2016.11.16 | System requirements상세화 및 Scenario 상세화 |
| 2.0.1 | 2016.11.16 | 문서 오탈자 수정 |
| 2.1.1 | 2016.11.16 | 문서 제목 수정 |
| 3.0.0 | 2016.11.18 | Appendices상세화 |
| 3.1.0 | 2016.11.20 | 그림(Figure) 추가 |
| 3.1.1 | 2016.11.20 | References 추가 |
| 4.0.0 | 2016.11.21 | Class diagram 수정 및 추가 |
| 5.0.0 | 2016.11.21 | State diagram 수정 및 추가 |
| 6.0.0 | 2016.11.21 | Sequence diagram 수정 및 추가 |
| 6.1.0 | 2016.11.22 | Index및 목차 정리 |
| 6.1.1 | 2016.11.22 | Appendices 2 첨부 |
| 6.2.0 | 2016.12.05 | Collection 관련 기능 삭제 |
| 7.0.0 | 2016.12.07 | Clothes Shopping Mall using Deep Learning-> Shopping Mall Solution using Deep Learning |

1. **Introduction**
   1. Objectives

Introduction에서는 시스템이 만들어지게 된 배경, 필요성 그리고 시스템의 기대효과에 대해 서술하며, 본 시스템에 포함된 서브시스템에 대해 간략히 소개하고 그 사이의 상호작용에 대해 설명한다.

* 1. 배경 및 필요성

인터넷 쇼핑몰(Internet Shoppingmall)이란, 가상공간에 존재하는 가상 상점을 말한다. 인터넷 쇼핑몰은 시간적, 공간적 제약이 없기 때문에, 소비자들이 언제 어디서든 상품을 원할 때 구매할 수 있다는 매력으로 사람들을 끌었고, 2000년대 초, 인터넷 쇼핑몰 붐을 일으켰다. 그리고 그 붐은 현재까지도 이어지고 있으며, 인터넷 쇼핑몰은 이제 포화상태에 이르렀다. 인터넷 쇼핑몰들이 우후죽순 생겨나면서, 인터넷 쇼핑몰 창업에 도움을 주는 쇼핑몰 사이트 개발 솔루션 또한 많이 생겼다. 같은 판매 아이템(item), 시스템(system) 속에서, 인터넷 쇼핑몰은 이제 다른 사이트들과는 차별화 전략을 써야만 살아남을 수 있는 시대가 되었다.

Figure 1 인터넷 쇼핑몰 창업시장의 규모

우리는 이러한 차별화 전략 중에서 판매자와 구매자 모두에게 ‘간편함’을 제공해 줄 수 있는 쇼핑몰 솔루션을 만들고자 한다. 판매자에게 ‘간단함’, ‘편안함’을 제공하기 위해서, 딥러닝(Deep Learning)과 이미지 분석을 이용한 태그 및 카테고리화 기능을 제공할 것이며, 구매자에게 열람 중인 상품 혹은 구매하고자 하는 상품과 관련된 ‘관련 상품 추천 기능’을 제공할 것이다.

본 문서에서는, ‘의류 쇼핑몰’을 예시로, 만들고자 하는 시스템에 대해 설명할 것이다.

* 1. 기대효과

현재까지의 쇼핑몰들은 판매자가 직접 사진에 대한 태깅(Tagging)을 해야만 했다. 하지만 본 시스템을 사용함으로써, 판매자가 직접 태깅을 하는 수고를 덜 수 있고, 소비자에게는 태그(Tag)를 통해 원하는 상품을 쉽게 찾을 수 있도록 도울 수 있다. 또한, 소비자가 관심 있어 하는 상품과 관련된 상품을 추천함으로써, 소비자가 비슷한 상품을 찾는 번거로움을 줄여줄 수 있다.

1. **Glossary**
   1. Objectives

Glossary에서는 본 문서에서 사용하는 용어들에 설명함으로써, 본 문서에 대한 독자의 이해를 돕는다

* 1. Term Definition

|  |  |
| --- | --- |
| **Terms** | **Definition** |
| 데이터  (Data) | 관찰이나 실험, 조사로 얻은 사실이나 정보. '자료'로 순화.  <컴퓨터> 컴퓨터가 처리할 수 있는 문자, 숫자, 소리, 그림 따위의 형태로 된 정보.  본 문서에서는 조사로 얻은 정보를 '컴퓨터'가 처리할 수 있는 그림 따위의 형태로 변환된 정보를 뜻함. |
| 딥러닝  (Deep Learning) | 컴퓨터가 사람처럼 생각하고 배울 수 있도록 하는 인공지능(AI) 기술. |
| 로그인  (Login) | 사용자가 호스트 컴퓨터나 네트워크에 자신의 아이디(ID)와 암호를 입력해서 자신을 알리고 등록하여 호스트 컴퓨터의 사용 권한을 받아 접속하는 작업. |
| 샘플  (Sample) | 전체 물건의 품질이나 상태 따위를 알아볼 수 있도록 그 일부를 뽑아 놓거나, 미리 선보이는 물건. '본보기', '표본'으로 순화.  본 문서에서는 통계 조사에서의 '표본'을 뜻함. |
| 서버  (Server) | <컴퓨터> 주된 정보의 제공이나 작업을 수행하는 컴퓨터 시스템.  서버는 클라이언트 시스템이 요청한 작업이나 정보의 수행 결과를 돌려준다. |
| 서브시스템  (Subsystem) | <컴퓨터> 하나의 시스템을 구성하고 있는 부분이면서, 그 자체로도 시스템을 이루고 있는 것. |
| 소스 코드  (Source Code) | <컴퓨터> 소스코드 또는 원시코드란 컴퓨터 프로그램을(사람이 읽을 수 있는) 프로그래밍 언어로 기술한 글. |
| 소프트웨어  (Software) | <컴퓨터> 컴퓨터 프로그램 및 그와 관련된 문서들을 통틀어 이르는 말.  컴퓨터를 관리하는 시스템 프로그램과, 문제 해결에 이용되는 다양한 형태의 응용 프로그램으로 나뉨. |
| 쇼핑몰  (Shopping mall) | 한 군데에서 여러 가지 물건을 살 수 있도록 상점들이 모여 있는 곳. 같은 말로는 쇼핑센터가 있다.  본 문서에서는 인터넷에서 운영되는 쇼핑몰(Internet Shopping mall)만을 의미한다. |
| 시스템(System) | <컴퓨터> 필요한 기능을 실현하기 위하여 관련 요소를 어떤 법칙에 따라 조합한 집합체 |
| 아이디(ID) | 컴퓨터를 사용할 때 사용자(또는 등록자)를 식별하기 위한 식별기호. |
| 오픈소스  (Open Source) | <컴퓨터> 소프트웨어의 소스 프로그램이 공개되어 있는 것 |
| 웹(Web) | <컴퓨터> 동영상이나, 음성 따위의 각종 멀티미디어를 이용하는 인터넷을 이르는 말.  월드 와이드 웹(World Wide Web) |
| 웹사이트  (Web site) | 인터넷에서 사용자들이 정보가 필요할 때 언제든지 그것을 제공할 수 있도록 웹 서버에 정보를 저장해 놓은 집합체. |
| 이미지(Image) | 거울, 카메라, 화면 등에 나타난 모습(상) |
| 인터넷  (Internet) | 전 세계의 컴퓨터가 서로 연결되어 정보를 교환할 수 있는, 하나의 거대한 컴퓨터 통신망 |
| 인터페이스  (Interface) | <컴퓨터> 서로 다른 두 시스템, 장치, 소프트웨어 따위를 서로 이어주는 부분. 또는 그런 장치. |
| 태그(Tag) | 가격 따위를 표시한 꼬리표. |
| 태깅(Tagging) | 태그(Tag)하기. |
| 폭포수 모델  (Waterfall Model) | 소프트웨어 개발 기법의 하나.  소프트웨어 생명주기(SDLC; Software Development Life Cycle)에 기반하고 있는 소프트웨어 개발 기법으로, 워터폴 모델, 폭포수 모형, 선형순차 모형, 단계적 생명주기라고도 함. 한번 떨어지면 거슬러 올라갈 수 없는 폭포수와 같이 소프트웨어 개발도 각 단계를 확실히 매듭짓고 다음 단계로 넘어간다는 의미에서 붙여진 명칭 |
| 하드웨어  (Hardware) | <컴퓨터> 컴퓨터를 구성하는 기계 장치의 몸체를 통틀어 이르는 말. |

1. **User Requirements Definition**
   1. Objectives

User Requirements Definition에서는 사용자가 요구한 요구사항과 제약사항에 대해 설명한다. 서술어는 자연어를 기반으로 하며, 그림을 통해 이해를 돕고자 한다.

* 1. Functional Requirements

시스템은 다음과 같은 기능을 제공해야 한다.

* + 1. 이미지 분석을 통한 태깅 및 카테고리화

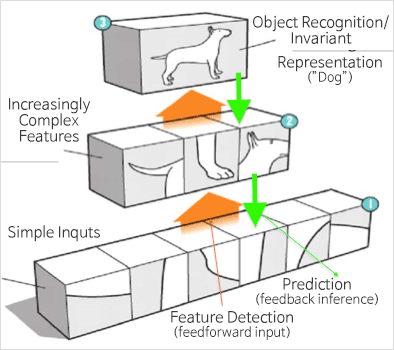
판매자가 올린 사진을 기반으로 하여, 사진 속 물건이 무엇인지 구분하고 태그하는 기능이다. 게시하고자 하는 사진을 찍어서 이미지를 홈페이지에 업로드 하면, 어떤 물건이고(바지 혹은 티셔츠), 어떤 특성을 지니고 있는지 사진 속 물건을 분석하여 상품을 카테고리화 하고(Categorization), 태그를 달아준다.

Figure 2 딥 러닝 기반의 이미지 분석

* + 1. 관련 상품 추천

구매자가 선택한 상품과 비슷한 상품, 함께 구매하면 좋은 상품을 추천해 화면에 띄워준다. 구매자가 선택한 상품에 태그 된 속성을 가지고 있으며, 다른 카테고리(Category)에 속한 물건을 추천해주기도 하며, 같은 카테고리 내의 비슷한 상품을 추천하도록 한다.

Figure 3 상품 추천 기능

* + 1. 데이터 기록 및 저장

구매자가 여태까지 구매한 상품들의 정보들 및 구매자의 개인 정보와 구매자가 클릭(Click)한 광고에 대한 정보를 서버에 저장하고, 이 데이터들을 가공하여 광고 및 추천 시스템에 적용할 수 있도록 한다.

* 1. Non-Functional Requirements

시스템이 제공해야 하는 서비스의 제약사항과 시스템이 만족해야 하는 System Properties(safety, performance 등)에 대해 서술한다.

* + 1. Product Requirements
       1. Usability Requirements

본 시스템의 목적은 판매자에게는 상품 업로드 시의 번거로움을 줄여주며, 구매자에게는 구매자가 관심 있어 하는 상품 정보를 제공하는 것이다. 따라서 판매자가 상품 업로드 시 입력해야 하는 정보를 최소화 할 수 있어야 하며, 직관적인 인터페이스(user interface)를 제공해야 한다. 또한, 다양한 연령대의 구매자가 있으므로, 구매자가 보는 인터페이스 또한 직관적이어야 한다.

* + - 1. Performance Requirements

자동으로 태깅 및 카테고리화 되는 기능을 구현할 때, 이미지 분석에 오류가 없어야 하며, 구매자에게 추천되는 관련상품 및 추천 상품이 구매자의 흥미를 끌 수 있도록, 정확한 상품 분석이 요구된다.

* + - 1. Space Requirements

웹(Web)기반의 쇼핑몰이고, 시스템을 운영하는데 필요한 것은 시스템을 관리 및 데이터를 저장하는 서버(server)뿐이므로, space requirement는 다른 requirements보다 중요도가 떨어진다.

* + - 1. Dependability Requirements

이미지에 대한 정확한 분석, 그리고 구매자의 구매정보 및 선호 상품에 대한 정확한 분석이 없으면, system 사용자의 흥미가 떨어지고, 본 시스템을 외면하게 될 것이다. 높은 만족감을 위해서는 정확한 데이터 분석을 요구한다.

* + - 1. Security Requirements

구매자의 신상정보, 구매 정보는 관리자 권한을 가진 사람만 열람 할 수 있도록 만들며, 판매자가 관리자 권한으로 시스템에 들어가야만 열람할 수 있도록 한다.

* + 1. Organizational Requirements
       1. Environmental Requirements

태깅, 카테고리화, 맞춤형 상품 추천을 위해서는 정확한 데이터 분석이 필요하고, 이러한 정확한 분석을 위해서는 많은 기준 데이터를 필요로 한다. 많은 데이터를 가져오기 위해서는 회사 내에 네트워크 환경이 늘 갖춰져 있어야 하며, 고객의 정보를 다루고 있으므로 고객의 정보를 다루는 서버가 꺼지지 않고, 늘 보안 상태를 유지할 수 있어야 한다.

* + - 1. Operational Requirements

판매자와 구매자가 본 시스템을 이용함으로써, 기존의 시스템보다 사용하기 쉽고, 시간이 절약되는 등의 장점을 느낄 수 있도록 해야 한다.

* + 1. External Requirement

구매자의 신상정보를 다루며, 구매정보를 수집하고 있으므로, 본 시스템에서 이러한 정보를 수집한다는 사실을 미리 구매자에게 알려야 하며, 사전 동의를 거친 후 데이터를 수집해야 한다.

1. **System Architecture**
   1. Objectives

High-Level에서의 목표 시스템의 개요와 간략한 구조, 그리고 각 Function들에 대해 설명함으로써, 시스템에 대한 독자의 이해를 돕고자 한다.

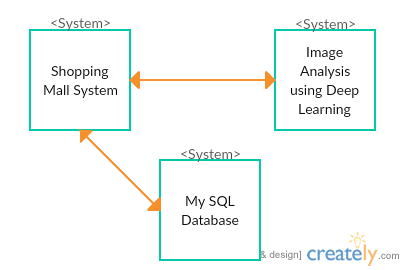
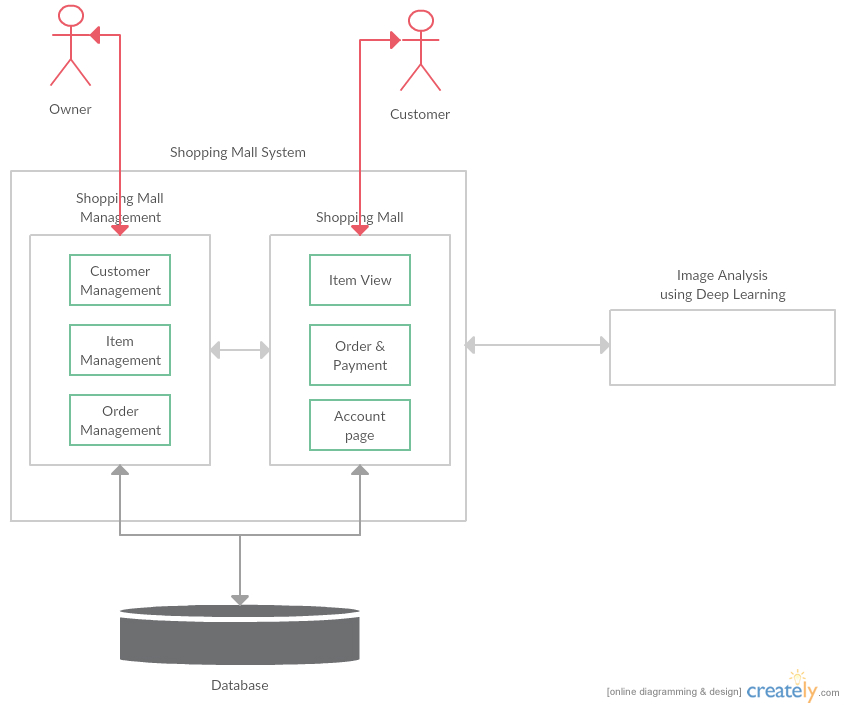
* 1. Overview

Diagram 1 System Architecture Overview

* 1. 상세 시스템 구조

목표 시스템은 크게 Item판매자가 접근하는 Shopping Mall Management level과 구매자가 접근하는 Shopping Mall level이 있다.

Diagram 2 Detail System Architecture

Shopping Mall Management level에서, 판매자는 고객관리, 상품 관리, 주문 내역 관리를 할 수 있으며, 이러한 관리를 위해서는 Admin 관리자 권한을 가지고 있는지 확인이 필요하다. Admin 권한의 확인 여부는Login ID와 데이터베이스 내의 관리자 ID의 일치 여부를 통해 확인한다. 또한 각종 관리를 위한 고객 정보와 아이템 정보, 주문 정보에 관한 것은 관리자 권한으로 데이터베이스에서 가져오도록 한다.

Shopping Mall level에서, 고객은 아이템을 보고, 주문을 하고, 본인의 계정을 관리할 수 있다.본인의 계정관리 시에는 본인 권한이 필요하며,이는 Login ID와 데이터베이스 내의 ID와의 일치 여부를 통해 확인한다. 아이템에 대한 정보와 계정에 대한 정보는 데이터베이스에서 가져오며, 새로운 주문 내역은 직접 작성하고, 이를 데이터베이스에 저장하기로 한다.

1. **System Requirements Specification**
   1. Objectives

System Requirements Specification에서는 시스템의 기능적 요구사항(Functional Requirements)와 비기능적 요구사항(Non-Functional Requirements)에 대해 상세히 서술하고, 본 시스템과 상호작용하는 시스템의 인터페이스(interface)에 대해 설명한다.

* 1. Functional Requirements
     1. 이미지 분석을 통한 태깅 및 카테고리화

|  |  |
| --- | --- |
| **기능** | 이미지 분석을 통한 태깅 및 카테고리화 |
| **설명** | 판매자가 올린 사진을 기반으로 하여, 사진 속 물건이 무엇인지 구분하고 분류하며, 물건의 특성을 태그로 달아준다. |
| **입력** | 사용자가 업로드 하고자 하는 이미지 |
| **출력** | 분류 항목(카테고리), 태그 |
| **처리** | 이미지가 등록되면, 이를 기존의 사진들과 비교해 분류한다. |
| **조건** | 인터넷에 연결되어 있어야 하며, 기존의 이미지들이 저장된 서버에 접속이 가능해야 한다. |

* + 1. 관련 상품 추천

|  |  |
| --- | --- |
| **기능** | 관련 상품 추천 |
| **설명** | 구매자가 구매했던 상품, 클릭했던 상품들을 분석해, 앞으로 구매자가 구매할 것 같은 상품들, 관심 있을 것 같은 상품들을 추천해 준다. |
| **입력** | 구매자가 구매했던 상품들의 태그(특성), 클릭했던 상품들의 태그(특성) |
| **출력** | 같은 특성을 가진 다른 카테고리 내의 상품 및 같은 카테고리 내의 상품 |
| **처리** | 구매했던 상품들의 태그, 클릭했던 상품들의 태그 중 가장 등장 횟수가 많은 태그를 선정한다. 그리고 그 태그를 가진 또 다른 상품을 찾아낸다. |
| **조건** | 모든 상품에 태그가 달려있어야 한다. 구매자의 구매 기록에 접근이 가능해야 한다. 구매자의 구매 기록 및 클릭 기록을 가지고 있어야 한다. |

* + 1. 데이터의 기록 및 저장

|  |  |
| --- | --- |
| **기능** | 데이터 기록 및 저장 |
| **설명** | 구매자의 개인정보, 구매 기록, 광고에 대한 정보를 서버에 저장하고 가공할 수 있도록 한다. |
| **입력** | 구매자의 개인정보, 구매 기록, 광고에 대한 클릭 횟수 |
| **출력** | 데이터의 저장 |
| **처리** | 구매자의 개인정보, 구매 기록, 광고에 대한 클릭 횟수를 데이터베이스에 저장한다. |
| **조건** | 데이터베이스에 접근이 가능해야 한다. 충분한 용량 확보가 필요하다. |

* 1. Non-functional Requirements
     1. Product Requirements
        1. Usability Requirements

판매자가 처음 이용하더라도 어려움 없이 시스템을 이용할 수 있어야 한다. 이미지 분석 과정에서 판매자에게 별도로 요구하는 시스템에 대한 전문적인 지식이 필요 없어야 하며, 한번에 알아볼 수 있는 직관적인 interface를 제공해야 한다. 또한, 구매자 연령층이 다양하므로, 구매자가 보는 interface또한 직관적이고 사용이 쉬워야 한다.

* + - 1. Performance Requirements

자동으로 태깅 및 카테고리화 되는 기능을 구현할 때, 이미지 분석에 오류가 없어야 한다. 이를 위해서는 최대한 많은 샘플 데이터(sample data)를 가지고 있어야 하며, 이 후 판매자가 올리는 상품 이미지 데이터들 또한, 모두 서버에 저장함으로 써, 분석의 기반이 되는 자료들을 많이 모아야만 한다.

* + - 1. Space Requirements

웹(Web)기반의 쇼핑몰이고, 시스템을 운영하는데 필요한 것은 시스템을 관리 및 데이터를 저장하는 서버(server)이다. 직접 서버를 구매해도 되나, 사무실 혹은 회사의 규모가 작으면, 타사에서 제공하는 서버를 이용하도록 한다.

* + - 1. Dependability Requirements

이미지에 대한 정확한 분석, 그리고 구매자의 구매정보 및 선호 상품에 대한 정확한 분석이 없으면, system 사용자의 흥미가 떨어지고, 시스템에 대한 흥미가 떨어질 것이다. 이를 방지하기 위해서는 지속적으로 보유 데이터를 관리하고, 기계(machine)가 계속 학습할 수 있도록 꾸준한 업데이트(update) 및 관리를 해주어 사용자에게 정확한 분석정보를 제공할 수 있어야 한다.

* + - 1. Security Requirements

본 시스템에서 다루는 데이터 중 구매자에 대한 정보는 개인 신상정보이기 때문에 엄격히 보호되어야 한다. 이를 위해서는 구매자의 비밀번호 및 구매자의 정보를 열람할 수 있는 판매자의 비밀번호를 충분히 복잡하게 만들도록 해야 한다. 또한, 이러한 정보를 저장하고 있는 데이터베이스의 접근도 제한해야 하며, 외부에서 접근할 수 없도록 철저한 보안과 관리가 요구된다.

* + 1. Organizational Requirements
       1. Environmental Requirements

태깅, 카테고리화, 맞춤형 상품 추천을 위해서는 정확한 데이터 분석이 필요하고, 이러한 정확한 분석을 위해서는 많은 기준 데이터를 필요로 한다. 많은 데이터를 가져오기 위해서는 회사 내에 네트워크 환경이 갖춰져 있어야 하며, 고객의 정보를 다루고 있으므로 고객의 정보를 다루는 서버가 늘 보안 상태를 유지할 수 있어야 한다.

* + - 1. Operational Requirements

판매자와 구매자가 본 시스템을 이용함으로써, 기존의 시스템보다 사용하기 쉽고, 시간이 절약되는 등의 장점을 느낄 수 있도록 해야 한다.

* + 1. External Requirement

구매자의 신상정보를 다루며, 구매정보를 수집하고 있으므로, 본 시스템에서 이러한 정보를 수집한다는 사실을 미리 구매자에게 알려야 하며, 사전 동의를 거친 후 데이터를 수집해야 한다.

* 1. Scenario
     1. 판매자Scenario (Scenario A)
        1. Initial Assumption

판매자는 웹사이트에서 상품을 등록/수정/삭제 할 수 있고, 구매자의 정보를 열람할 수 있다.

* + - 1. Normal
         1. 판매자가 쇼핑몰 웹사이트(Web site)에 관리자ID로 로그인하여 접속한다.

(로그인 실패 시) “아이디와 패스워드를 확인해주세요”라는 창이 뜨고, 다시 로그인을 시도할 수 있도록 한다.

* + - * 1. 판매자가 “상품 등록”버튼을 클릭한다.
        2. 상품 등록 화면에서 “이미지 등록하기” 버튼을 이용해 이미지를 등록한다.
        3. 상품 등록 화면의 “태그” 부분에 판매자가 올린 이미지를 분석하여 이미지와 관련된 태그가 등록된다. (ex. 상품의 색-빨간색, 상품의 종류-하의, 상품의 모양-바지 등) 추가로 원하는 태그가 있을 시, “태그” 부분에 직접 입력하도록 한다.

(이미지 분석 실패 시) “이미지를 인식할 수 없습니다. 태그를 직접 달아주세요.”라는 창이 뜨고, “태그”부분에 사용자가 직접 태그를 입력 할 수 있도록 한다.

* + - * 1. 상품 등록 화면에 이미지 분석을 기반으로 하여 본 이미지의 분류가 “카테고리”부분에 나타난다. (ex. 분류)

(이미지 분석 실패 시) “이미지를 인식할 수 없습니다. 카테고리를 직접 선택해 주세요.”라는 창이 뜨고, “카테고리”부분에 이미지 분류 항목을 직접 선택할 수 있도록 한다.

* + - * 1. 상품 등록 화면에서 나머지 필드(fields) (ex. 상품의 이름, 가격, 상세정보)를 판매자가 작성한 후, “저장하기” 버튼을 누른다.

(상품등록 실패 시)상품 등록에 필요한 정보들이 모두 입력되지 않을 경우(상품의 이름, 가격, 분류, 태그가 없는 경우), “상품 등록에 실패했습니다. 상품 정보를 다시 확인해주세요”라는 창이 뜨고, 상품 등록 화면으로 돌아간다.

* + - * 1. 등록하고자 하는 상품이 웹사이트에 게시된다.
        2. 상품 수정 시, 수정하고자 하는 상품을 클릭하고, 상품 정보가 나타나있는 화면에서 “상품 수정” 버튼을 클릭한다.
        3. “상품 등록”과 같은 화면에 기존의 “상품 정보”들이 나타나며, 정보 수정이 가능해 진다.
        4. 상품 수정을 완료하면, “저장하기” 버튼을 누른다.

(상품 수정 실패 시)상품 등록에 필요한 정보들이 모두 입력되지 않을 경우(상품의 이름, 가격, 분류, 태그가 없는 경우) “상품 업데이트에 실패했습니다. 상품 정보를 다시 확인해주세요”라는 창이 뜨고, 상품 수정 화면으로 돌아간다.

* + - * 1. 판매자의 관리자 페이지 목록에서 “주문 목록”버튼을 클릭한다.
        2. 구매자들이 주문한 주문 상품과 구매자 ID, 배송 주소, 수령인 이름을 확인한다.
        3. 게시된 상품의 수량이 구매자들이 구매할 때 마다 감소된다. 상품의 수량이 0이 된 상품은 “품절”이라는 표시가 상품 이름 옆에 뜨게 된다.
        4. “품절” 표시가 된 상품을 클릭해, 상품 정보가 나타나 있는 화면에서 “상품 삭제” 버튼을 누른다.

(삭제를 원하지 않는 경우) “상품 수정” 버튼을 눌러 상품의 수량을 고친 후, 수정 내용을 “저장하기” 버튼을 눌러 저장한다.

* + - 1. What can go wrong and Concurrent Activities in Scenario A

수정된 상품 정보가 저장이 되지 않을 경우, 인터넷 연결 상태와 로그인 상태를 확인한다. 그래도 상품 정보가 업데이트 되지 않는다면, 시스템 관리자에게 연락해 판매자 권한을 확인 한 뒤, 다시 시도해 보도록 한다.

* + - 1. System State on Completion in Scenario A

판매자는 판매자의 아이디로 로그인 할 경우, 상품을 등록/수정/삭제할 수 있으며, 고객의 정보를 확인할 수 있다.

* + 1. 구매자Scenario (Scenario B)
       1. Initial Assumption

구매자는 구매자 아이디로 로그인 시, 상품 정보를 열람 및 구매할 수 있다.

* + - 1. Normal
         1. 구매자는 쇼핑몰 회원가입을 한다. 이 때, 본 쇼핑몰에서 구매자의 개인 정보 및 구매 정보를 이용한다는 동의서에 동의 여부를 확인하기로 한다.

(개인 정보 열람 동의 여부에 미 동의 시)회원가입이 되지 않으며, “개인정보 이용 동의서에 동의해주세요”라는 창이 뜨고, 다시 회원가입 창으로 넘어가게 된다.

* + - * 1. 카테고리 별로 나눠져 있는 메뉴 바(Menu bar)를 통해, 원하는 상품을 찾거나, 원하는 상품의 특징(색상, 모양 등)을 상단 태그 바에서 선택해서 원하는 상품을 찾는다.
        2. 자세한 정보를 원하는 상품의 이미지를 클릭한다.
        3. 제품 정보의 하단에서 선택한 제품과 유사한 제품 및 추천 제품을 확인할 수 있다. 이 때, 추천 상품의 정보를 확인하고 싶으면, 추천 제품의 이미지를 클릭하여 해당 제품의 제품 정보 창으로 넘어가도록 한다.
        4. 구매하고자 하는 제품의 수량과 색상, 사이즈를 선택한 후 “장바구니” 버튼을 누른다.

(장바구니 버튼이 클릭이 안될 경우) “장바구니에 담기에 실패했습니다. 장바구니에 넣고자 하는 상품의 정보(수량, 색상, 사이즈)를 확인해주세요”라는 창이 뜨고, 제품 정보 화면으로 돌아간다.

* + - * 1. 구매를 원하는 상품을 모두 “장바구니”에 담는다.
        2. 장바구니 화면에서 “구매하기” 버튼을 누른다.
        3. 총 결제금액을 확인한 후, 배송을 원하는 배송지와 수령인 이름을 입력한다. 그리고 “결제하기” 버튼을 누른다.

(결제하기 실패 시) “결제하기에 실패했습니다. 배송지와 수령인 이름을 모두 입력해주세요”라는 창이 뜨고, 구매하기 화면으로 돌아간다.

* + - * 1. 결제금액이 결제 시스템으로 넘어간다.
      1. What can go wrong and Concurrent activities in Scenario B

장바구니에 담기가 안될 경우, 로그인을 하였는지 확인하고, 모든 버튼이 안 눌릴 시, 인터넷 연결 상태를 확인한다. 인터넷과 로그인 상태가 모두 정상이고, 모든 상세 정보를 입력하였는데도 구매 및 결제가 되지 않는다면, 판매자에게 문의하여 조치를 취할 수 있도록 한다.

* + - 1. System State on Completion in Scenario B

구매자는 태그 기능과 카테고리화 기능을 이용해 상품을 편리하게 검색할 수 있고, 선택 상품과 관련된 상품들, 구매자 별로 맞춤 상품을 추천 받을 수 있다.

1. **System Models**
   1. Objectives

시스템의 구성요소와 시스템의 작동 환경을 다양한 모델들을 이용해 시각적으로 표현해, 시스템이 어떻게 구성되어 있고, 어떻게 작동이 되는지에 대해 설명한다.

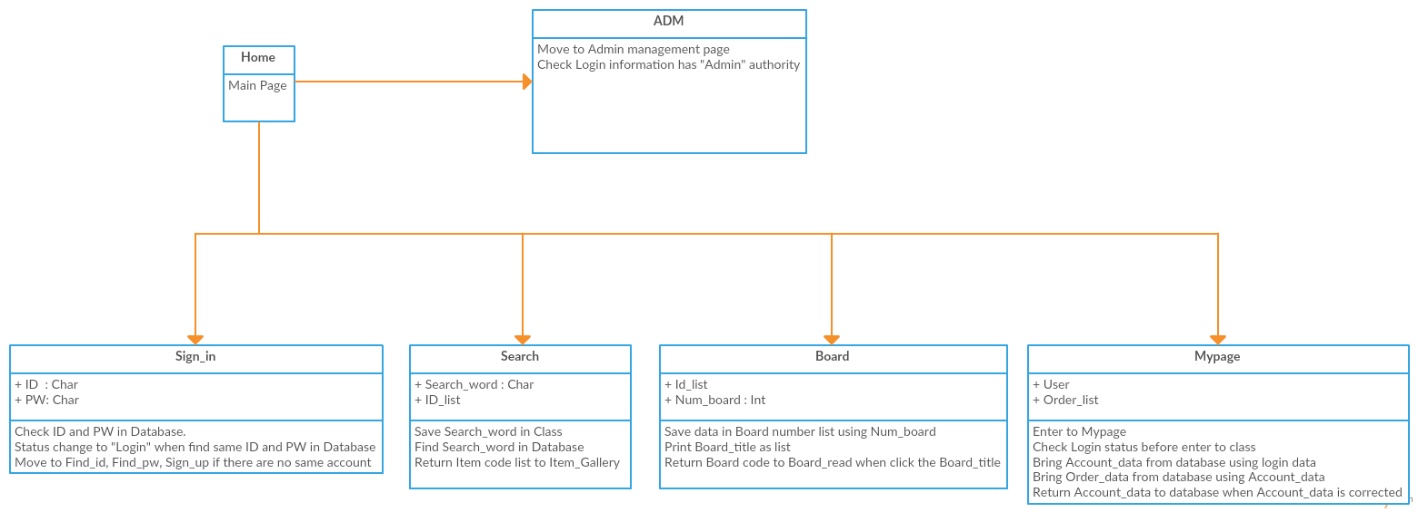
* 1. Class Diagram

Diagram 3 System Overview Class Diagram

* + 1. Overview
       1. Home : 쇼핑몰을 켰을 때 맨 처음 보이는 메인 페이지
       2. Sign\_in
          1. Attributes

ID-입력 받은 아이디

PW-입력 받은 비밀번호

* + - * 1. Description

입력 받은 아이디, 비밀번호와 일치하는 계정이 있는지 데이터베이스에서 확인 후, 일치 하는 것이 있으면 Login 상태로 변경.

Find\_id, Find\_pw, Sign\_up페이지로 이동 가능

* + - 1. Search
         1. Attributes

Search\_word: 입력 받은 검색 단어

Id\_list: 검색된 상품 코드 리스트

* + - * 1. Description

입력받은 검색문을 클래스에 저장

저장된 search\_word를 통해 데이터베이스에서 상품 검색 실행

검색된 상품 코드 리스트를 Item\_gallery에 넘겨줌

* + - 1. Board
         1. Attributes

Id\_list: 게시글 코드 번호 리스트

Num\_board: 게시글 목차 번호

* + - * 1. Description

게시글 목차 번호를 이용하여 최신부터 일부 데이터를 가져와 게시글 번호 리스트에 저장

게시글 제목을 리스트 형태로 출력

* + - 1. Mypage
         1. Attributes

User: 계정 정보

Order\_list: 주문 정보 리스트

* + - * 1. Description

계정 Mypage로 들어가는 클래스

클래스로 들어가기 전 로그인 상태 확인

현 로그인 정보로 데이터베이스에서 해당 계정 데이터를 받아옴.

해당 계정 데이터에 부합하는 주문 정보들을 데이터베이스에서 가져옴.

계정 정보 수정 시 해당 수정 내용을 데이터베이스에 반환.

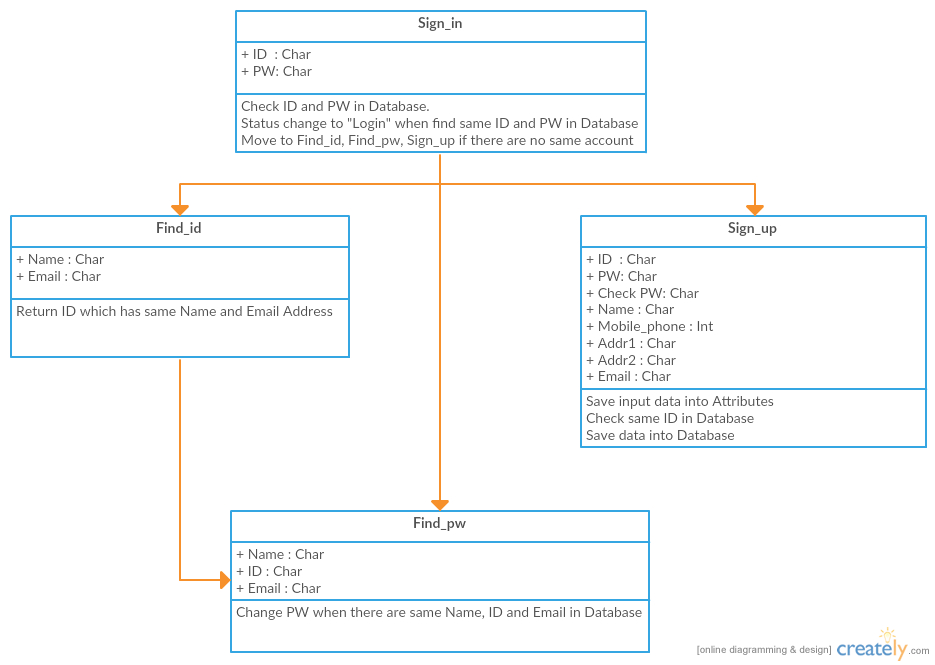
* + - 1. ADM: admin 관리 페이지로 이동하는 클래스로 해당 로그인 계정이 admin인지 아닌지 확인.
    1. Sign\_in Diagram

Diagram 4 Sign\_in Class Diagram

* + - 1. Sign\_in
         1. Attributes

ID: 입력 받은 ID

PW: 입력 받은 비밀번호

* + - * 1. Description

입력받은 아이디, 비밀번호와 일치하는 계정이 있는지 데이터베이스에서 확인 후 일치 하는 것이 있으면 로그인 상태로 변경

Find\_id, Find\_pw, Sign\_up페이지로 이동 가능

* + - 1. Sign\_up
         1. Attributes

ID: 아이디

PW: 비밀번호, Check PW: 비밀번호 확인

Name: 이름, Mobile\_phone: 휴대폰 번호, Phone: 일반 전화

Addr1: 주소, Addr2: 상세주소, Email: 이메일 주소

* + - * 1. Description

회원가입 데이터를 받아서 각 Attribute에 저장

중복되는 ID가 있는지 데이터베이스에서 확인

데이터에 하자가 없으면(누락된 데이터 없음. 중복되는 정보 없음) 이를 데이터베이스에 입력

* + - 1. Find\_id
         1. Attributes

Name: 계정 주 이름, Email: 이메일 주소

* + - * 1. Description

계정 주의 이름과 이메일 주소를 받아서 일치하는 아이디를 반환

* + - 1. Find\_pw
         1. Attributes

Name: 계정 주 이름, ID: 아이디, Email: 이메일 주소

* + - * 1. Description

계정 주의 이름과 아이디, 이메일 주소를 받아서 모두 일치하는 계정이 있으면 패스워드 변경 창을 띄워줌.

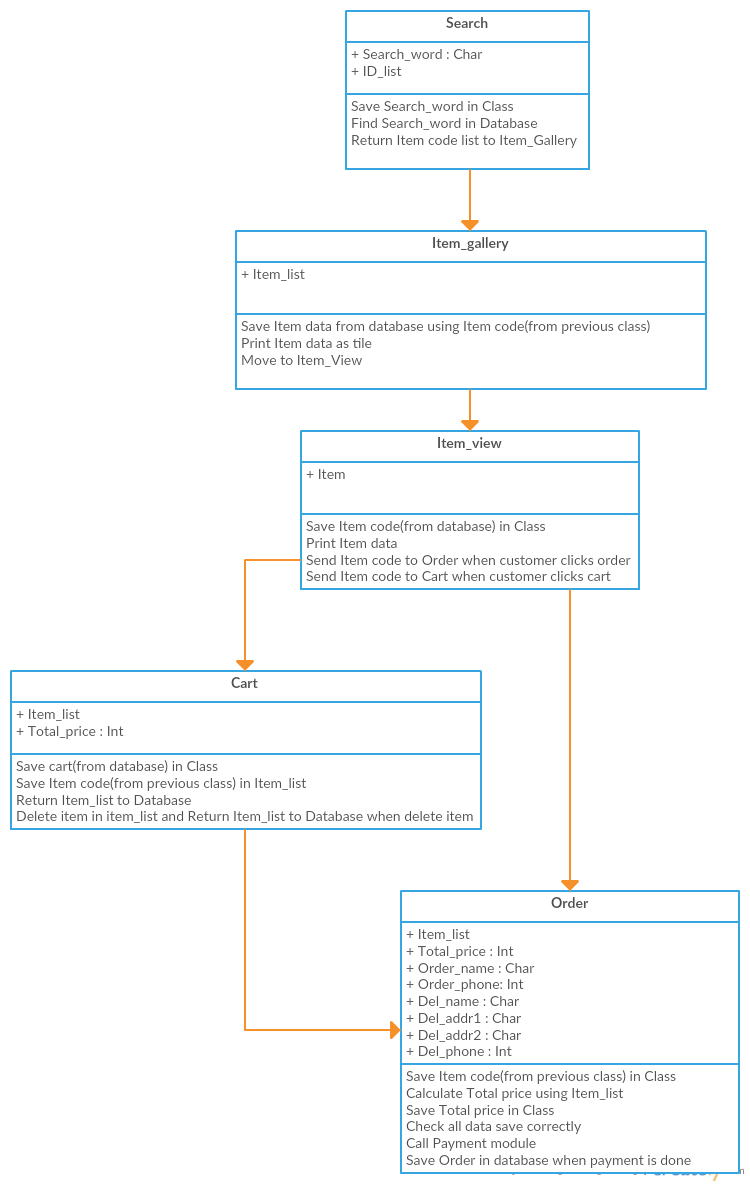
* + 1. Search Diagram

Diagram 5 Search Class Diagram

* + - 1. Search
         1. Attributes

Search\_word: 검색하고자 하는 단어

ID\_list: 검색된 상품 코드 리스트

* + - * 1. Description

입력받은검색문을 클래스에 저장

저장된 search\_word를 통해 데이터베이스 내에서 상품 검색 실행

검색된 상품 코드 리스트를 Item\_gallery에 넘겨줌

* + - 1. Item\_Gallery
         1. Attributes

Item\_list: 표시할 상품 코드 리스트

* + - * 1. Description

이전 클래스에서 넘겨 받은 상품 코드 리스트를 이용해 데이터베이스에서 상품 데이터를 가져와 클래스에 저장.

해당 상품 데이터를 바둑판식으로 출력

Item\_view로 이동

* + - 1. Item\_view
         1. Attributes

Item: 상품 관련 데이터

* + - * 1. Description

이전 페이지에서 관련 상품 코드를 가져와 데이터베이스에서 찾아 클래스에 저장

저장된 상품 데이터를 화면에 출력

상품 주문 시 해당 상품 코드를 order에 전달

장바구니에 넣기 시 해당 상품 코드를 cart에 전달

* + - 1. Cart
         1. Attributes

Item\_list: 장바구니 상품 코드 리스트

Total\_price: 총 가격 합계

* + - * 1. Description

해당 계정의 기존 장바구니 리스트를 데이터베이스에서 가져와 저장

이전 클래스에서 가져온 상품 코드를 item\_list에 저장 후 데이터베이스에 반환

장바구니 목록에서 상품 삭제 시 item\_list에서 해당 데이터를 삭제하고 이를 데이터베이스에 반환

* + - 1. Order
         1. Attributes

Item\_list: 주문 상품 정보 리스트

Total Price: 최종가격

Order\_name: 주문자 성함,Order\_phone: 주문자 전화번호

Del\_name: 수령자 이름

Del\_addr1: 배송지 주소, Del\_addr2: 배송지상세주소

Del\_phone: 수령자 전화번호

* + - * 1. Description

주문 상품들 코드 리스트를 이전 클래스에서 받아서 해당 클래스에 저장

Item\_list를 이용해 최종 가격을 산출해 Total\_price에 저장

주문자 데이터,수령자 데이터를 입력 받아 클래스에 저장

해당 데이터들이 전부 맞게 입력되었는지 확인

데이터가 확인되면 외부 주문 결제 모듈을 호출

결제가 완료되면 주문 정보를 데이터베이스에 저장

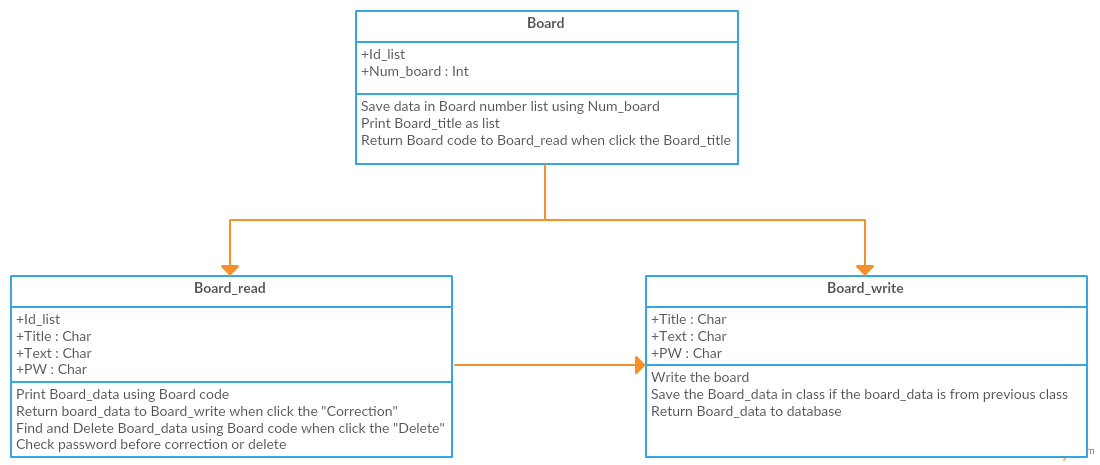
* + 1. Board Diagram

Diagram 6 Board Class Diagram

* + - 1. Board
         1. Attributes

ID\_list: 게시글 코드 번호 리스트

Num\_board: 게시글 목차 번호

* + - * 1. Description

게시글 목차 번호를 이용하여 최신부터 일부 데이터를 가져와 게시글 번호 리스트에 저장

게시글 제목을 리스트 형태로 출력

* + - 1. Board\_read
         1. Attributes

ID:게시글 코드

Title: 게시글 제목, Text: 게시글 내용

PW: 게시글 비밀번호

* + - * 1. Description

넘겨받은 게시글 코드번호를 이용해 해당 게시글 데이터를 데이터베이스로 가져와 이를 출력

수정 시 해당 게시글 데이터를 Board\_write에 넘겨줌

삭제 시 해당 게시글 코드를 이용해 데이터베이스에서 색인 후 게시글 데이터 삭제

게시글 수정 혹은 삭제 전, 게시글 비밀번호와 입력 받은 비밀번호의 일치 여부 확인

* + - 1. Board\_write
         1. Attributes

Title: 게시글 제목, Text: 게시글 내용

PW: 게시글 비밀번호

* + - * 1. Description

게시글 작성 클래스

이전 클래스에서 게시글 데이터를 받아 왔다면, 해당 게시글 데이터를 클래스에 저장.

작성 완료 시 해당 게시글 데이터를 데이터베이스에 반환.

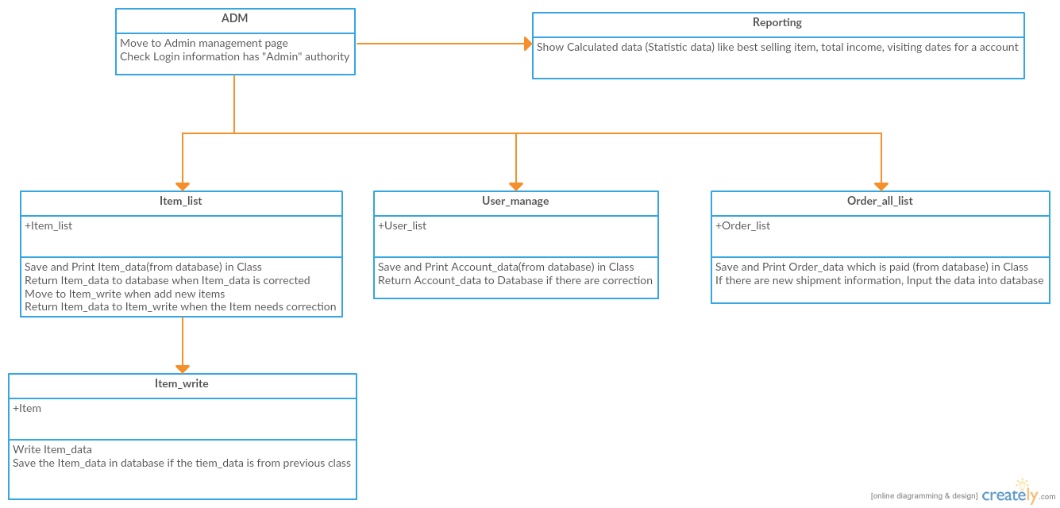
* + 1. ADM Diagram

Diagram 7 ADM Class Diagram

* + - 1. ADM: admin관리 페이지로 이동하는 클래스, 해당 로그인 계정이 admin인지 아닌지 확인함.
      2. Item\_list
         1. Attributes

Item\_list: 상품 데이터 리스트

* + - * 1. Description

상품 데이터들을 데이터베이스에서 가져와 클래스에 저장 및 출력

특정 상품 데이터의 간단한 수정 혹은 삭제 시,이를 데이터베이스에 반환

새 상품 등록 시,Item\_write로 이동함

특성 상품 수정 시,Item\_write에 해당 상품 데이터를 넘겨줌

* + - 1. User\_manage
         1. Attributes

User\_list: 회원 계정 데이터 리스트

* + - * 1. Description

회원 계정 데이터를 데이터베이스에서 가져와 클래스에 저장 및 출력

계정 수정, 삭제 시 이를 데이터베이스에 반환

* + - 1. Order\_all\_list
         1. Attributes

Order\_list: 주문 리스트

* + - * 1. Description

결제 완료되어 저장된 주문들을 데이터베이스에서 가져와 클래스에 저장 및 출력

배송정보를 주문에 입력 혹은 수정 시 이를 데이터베이스에 반영

* + - 1. Reporting: 회원의 방문, item 판매, 매출 등의 통계자료를 보여줌.
    1. Image Analysis Diagram.

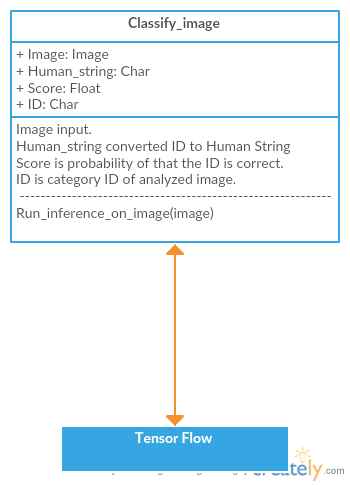


Diagram 8 Image Analysis Class Diagram

* + - 1. Classify\_image
         1. Attributes

Image: 입력 받은 이미지 파일

Human\_string: 해당 ID를 사람의 언어로 변환한 값.

Score: 해당 ID일 확률

ID: 분석한 이미지의 카테고리 ID

* 1. Sequence Diagram
     1. Sign\_up Diagram

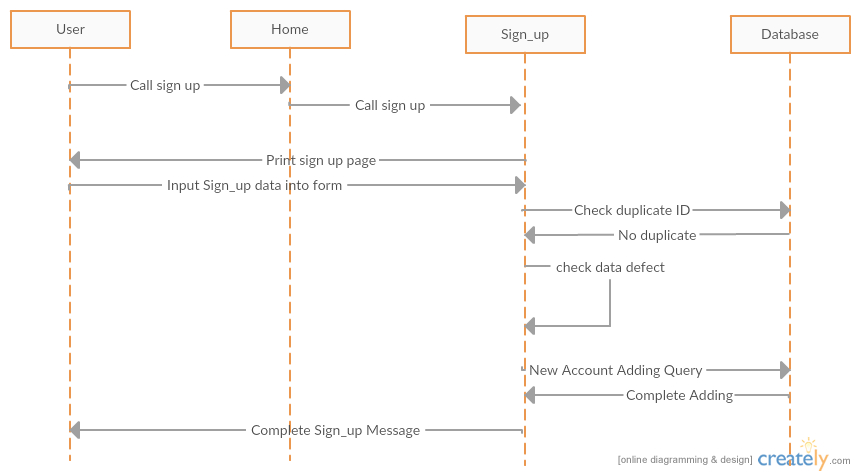


Diagram 9 Sign\_up Sequence Diagram

* + 1. Sign\_in Diagram

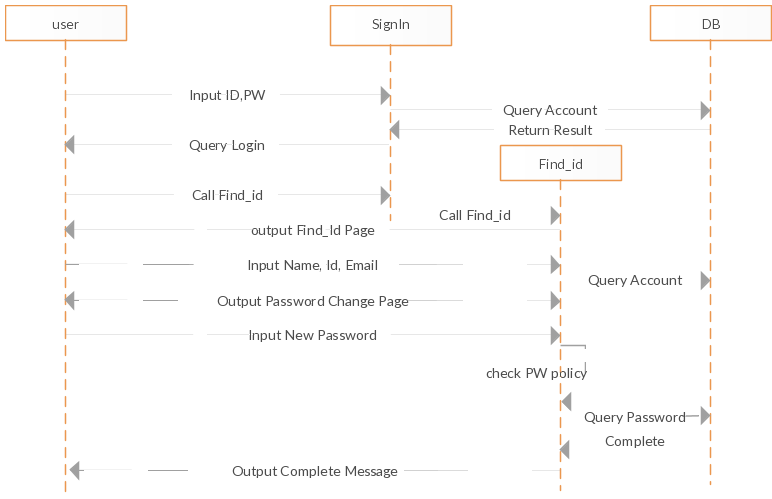


Diagram 10 Sign\_in Sequence Diagram

* + 1. Search Diagram

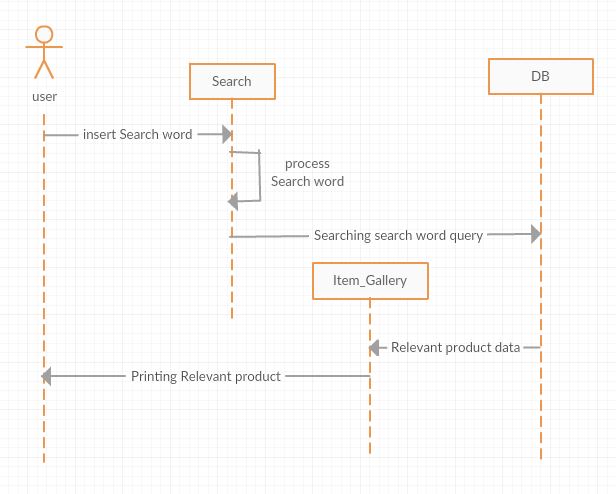


Diagram 11 Search Sequence Diagram

* + 1. Cart Diagram

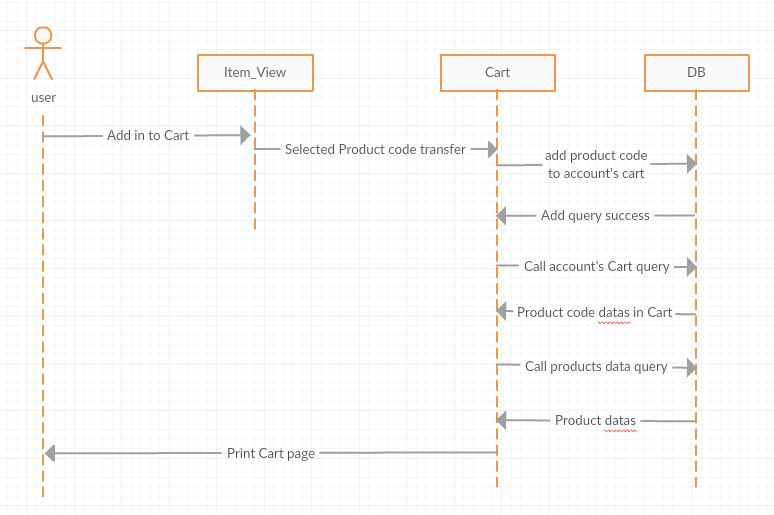


Diagram 12 Cart Sequence Diagram

* + 1. Purchasing Cart Diagram

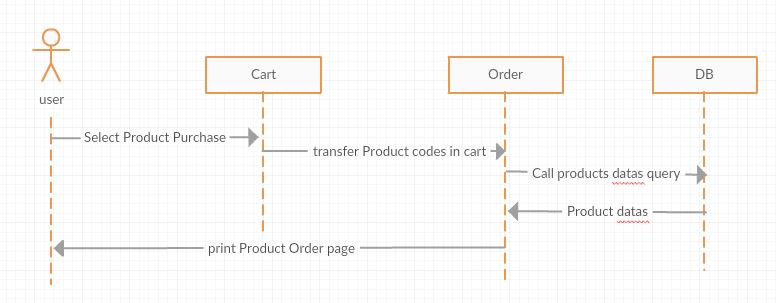


Diagram 13 Purchasing Cart Sequence Diagram

* + 1. Purchasing Item Diagram

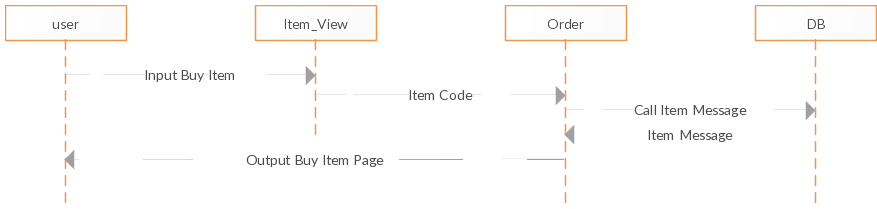


Diagram 14 Purchasing item Sequence Diagram

* + 1. Purchasing and Payment Diagm

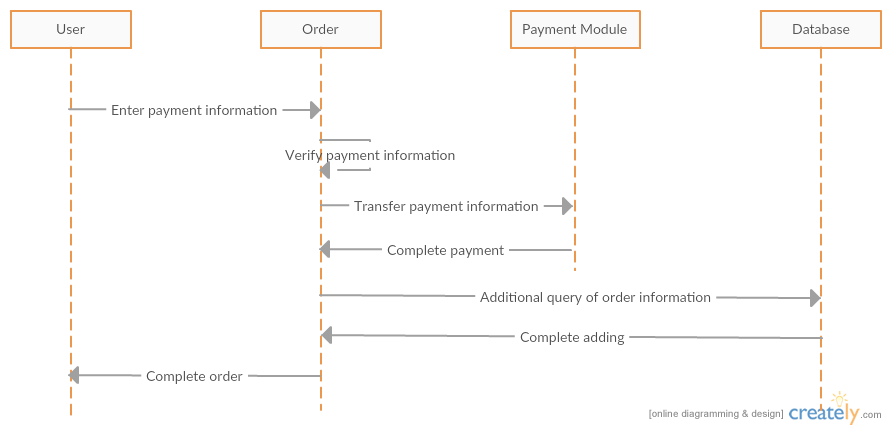


Diagram 15 Purchasing and Payment Sequence Diagram

* + 1. Board Diagram

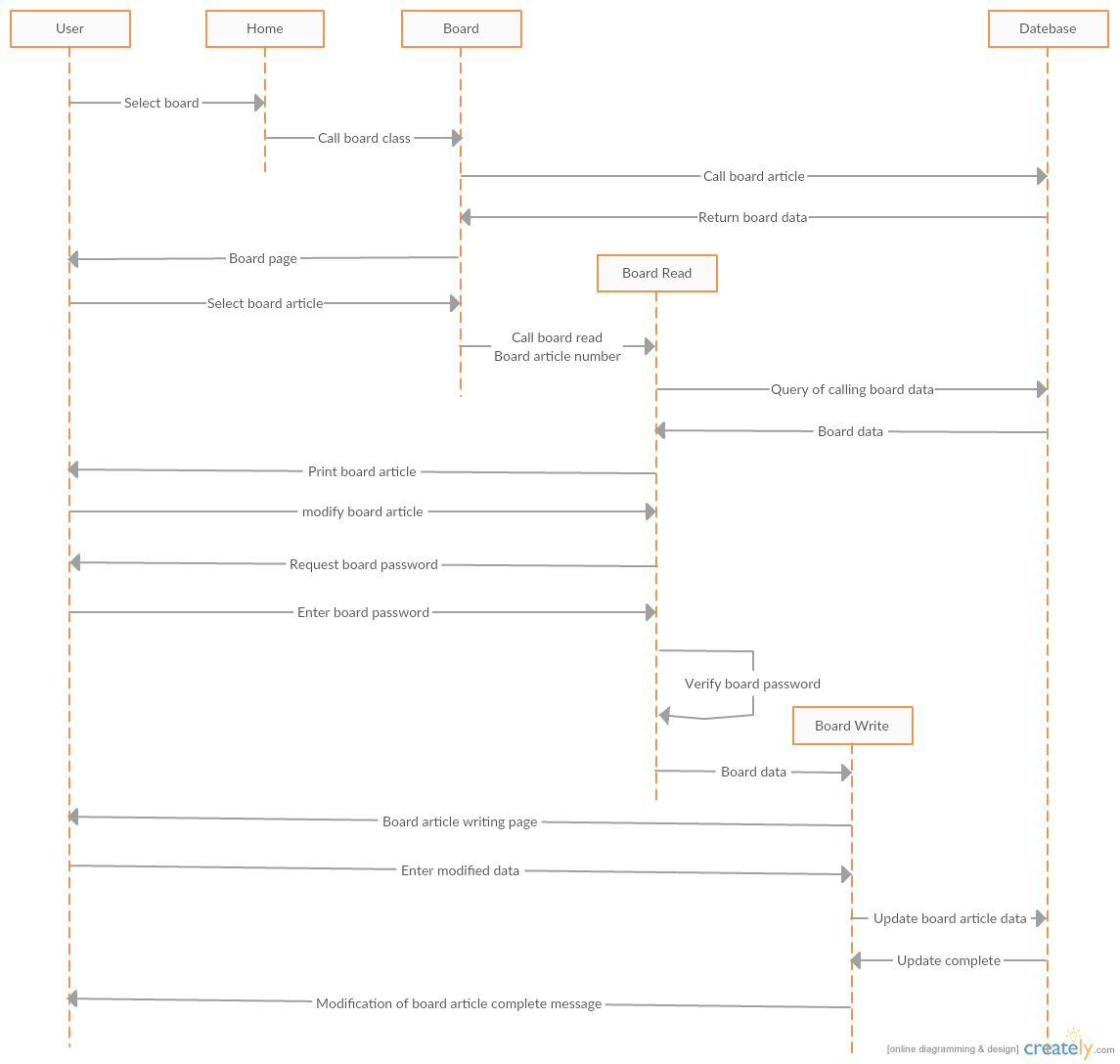


Diagram 16 Board Sequence Diagram

* + 1. Mypage Diagram

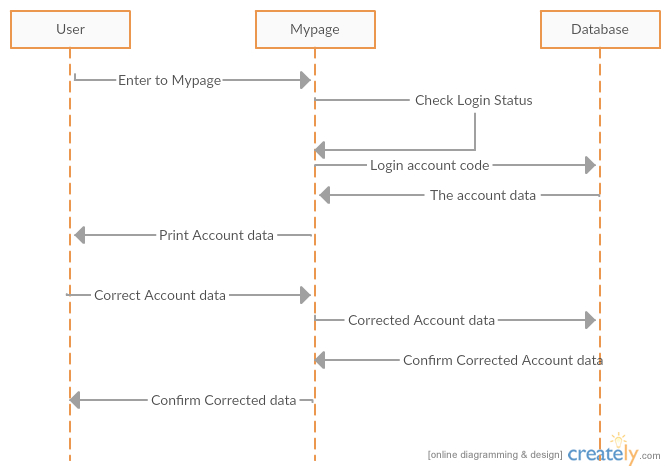


Diagram 17 Mypage Sequence Diagram

* + 1. ADM Diagram

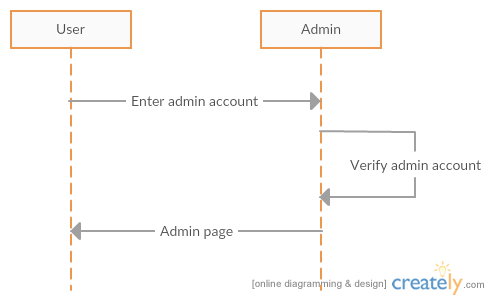


Diagram 18 ADM Sequence Diagram

* + 1. Item Add and Correction Diagram

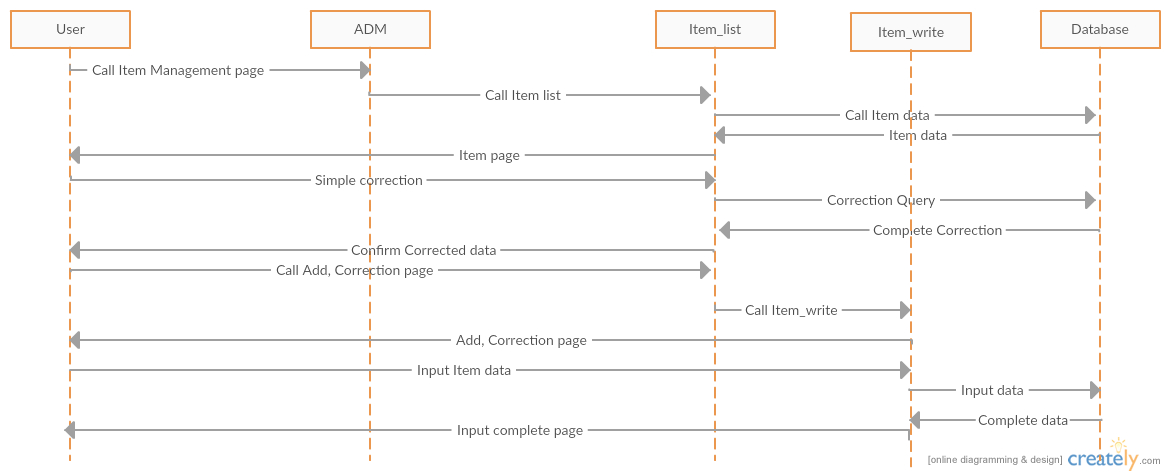


Diagram 19 Item Add and Correction Sequence Diagram

* + 1. User Management Diagram

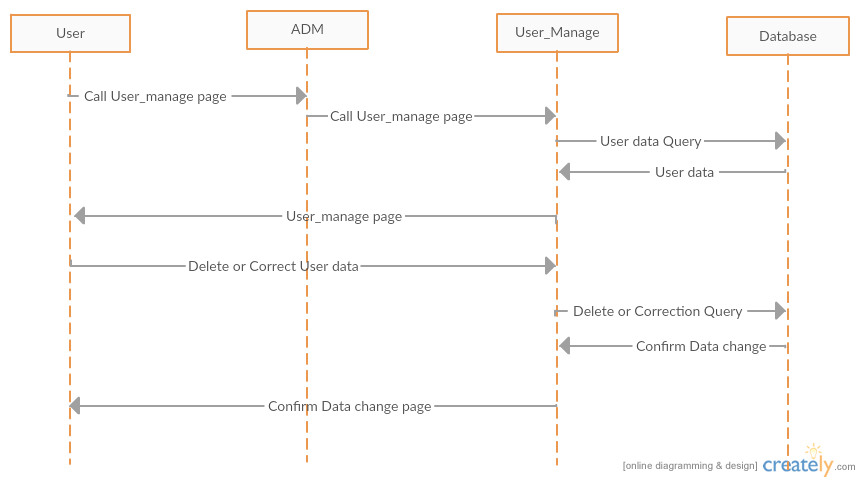


Diagram 20 User Management Sequence Diagram

* + 1. Order Management Diagram

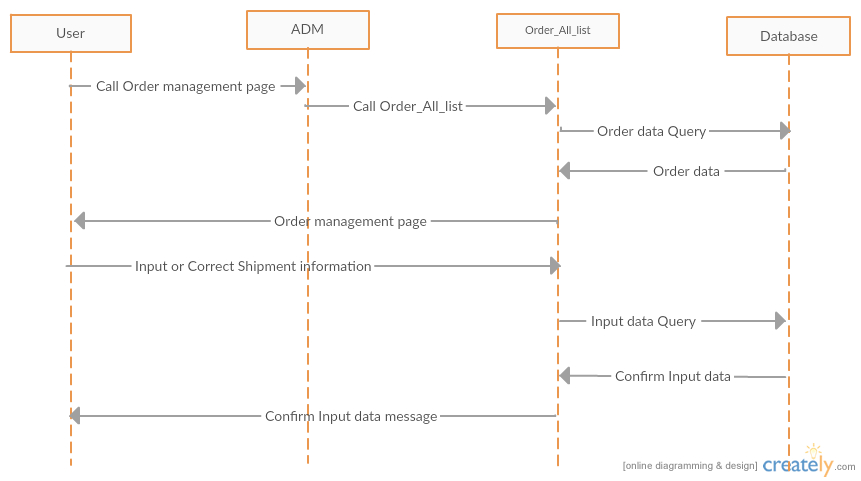


Diagram 21 Order Management Sequence Diagram

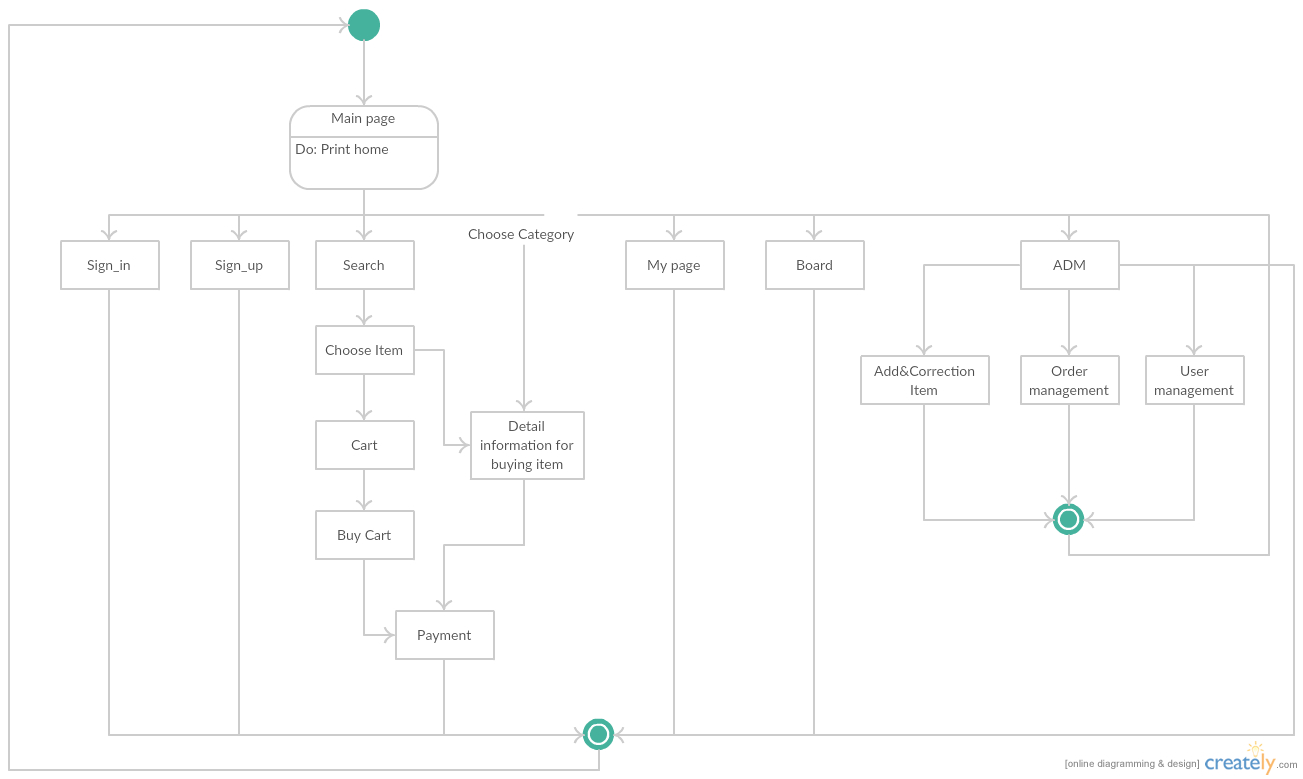
* 1. State Diagram
     1. Overview Diagram

Diagram 22 System Overview State Diagram

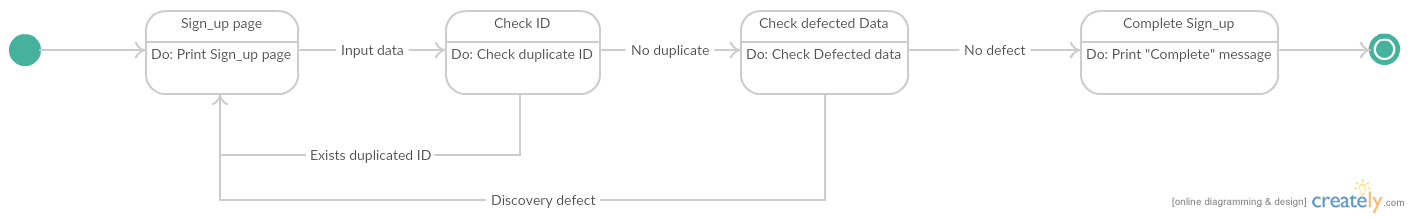
* + 1. Sign\_up Diagram

Diagram 23 Sign\_up State Diagram

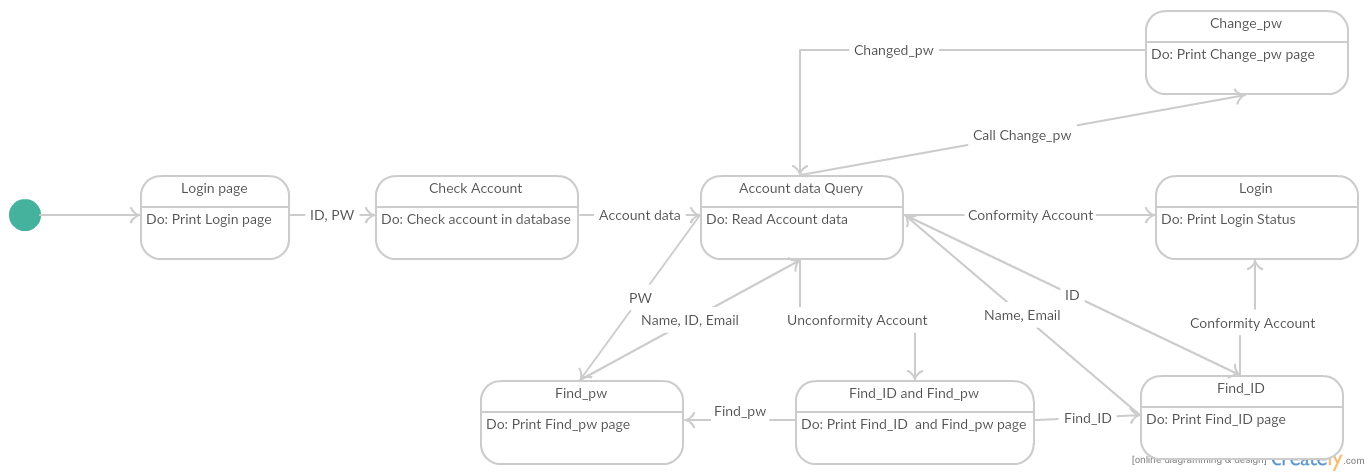
* + 1. Sign\_in Diagram

Diagram 24 Sign\_in State Diagram

* + 1. Search Diagram

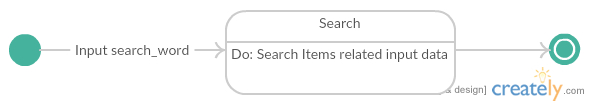


Diagram 25 Search State Diagram

* + 1. Choose\_Item Diagram

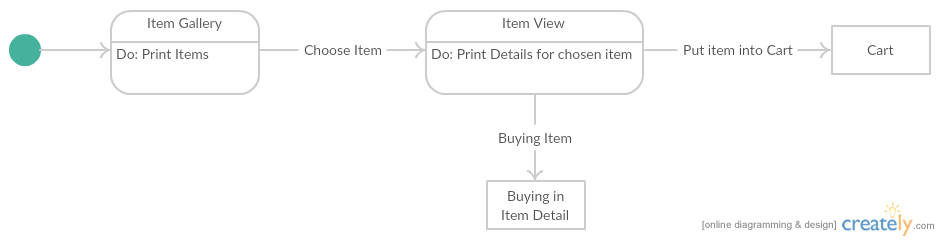


Diagram 26 Choose\_Item State Diagram

* + 1. Cart Diagram

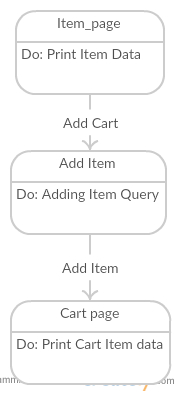


Diagram 27 Cart State Diagram

* + 1. Purchasing Diagram

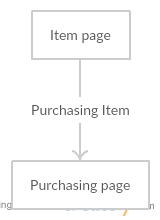


Diagram 28 Purchasing State Diagram

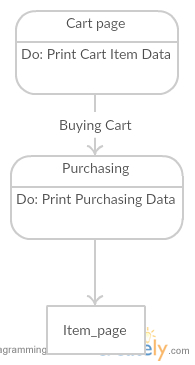
* + 1. Purchasing Cart Diagram

Diagram 29 Purchasing Cart State Diagram

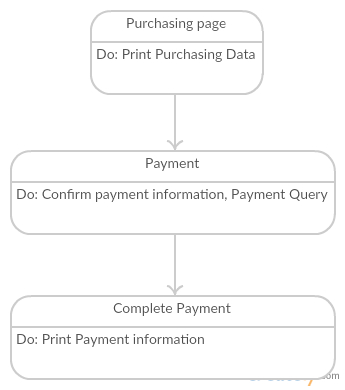
* + 1. Payment Diagram

Diagram 30 Payment State Diagram

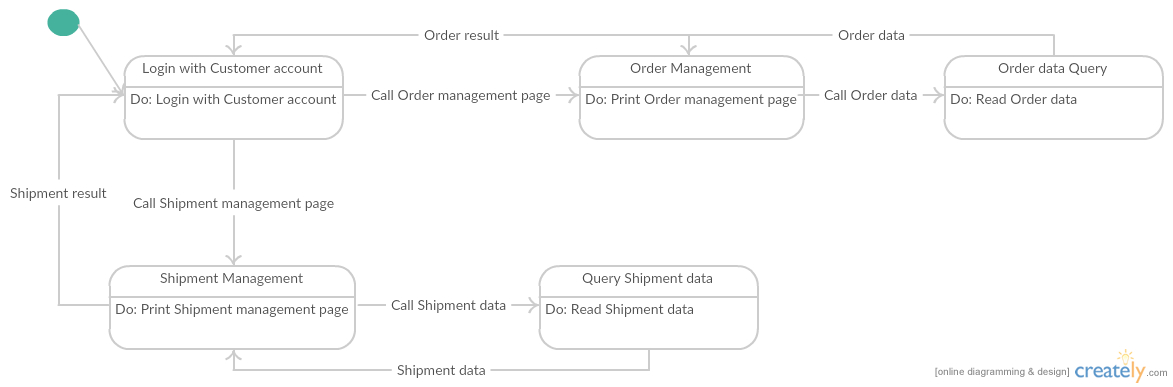
* + 1. Shipment Diagram

Diagram 31 Shipment State Diagram

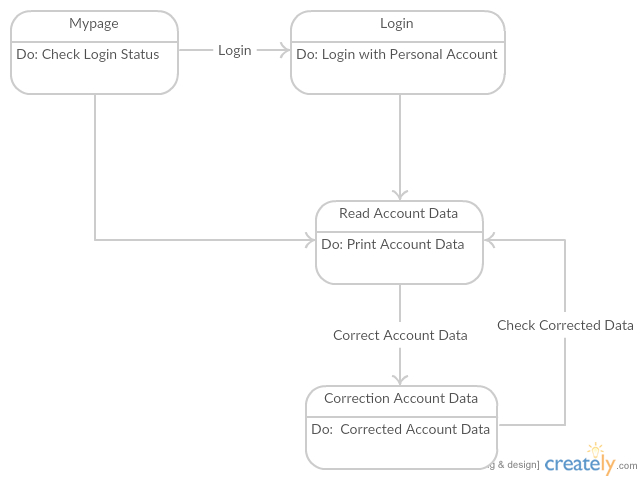
* + 1. Mypage Diagram

Diagram 32 Mypage State Diagram

* + 1. Board Diagram

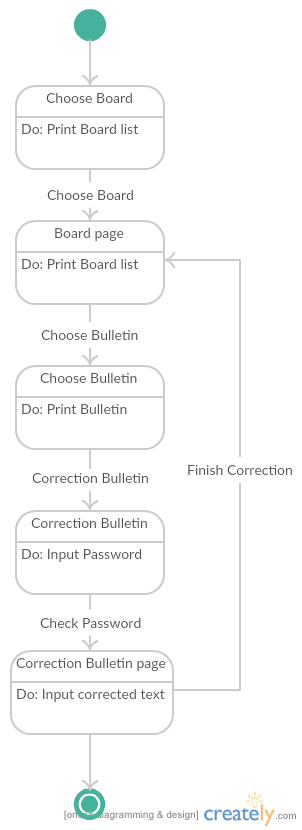


Diagram 33 Board State Diagram

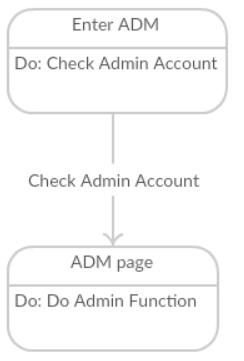
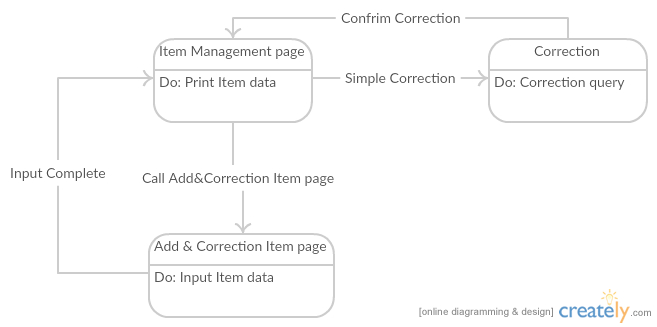
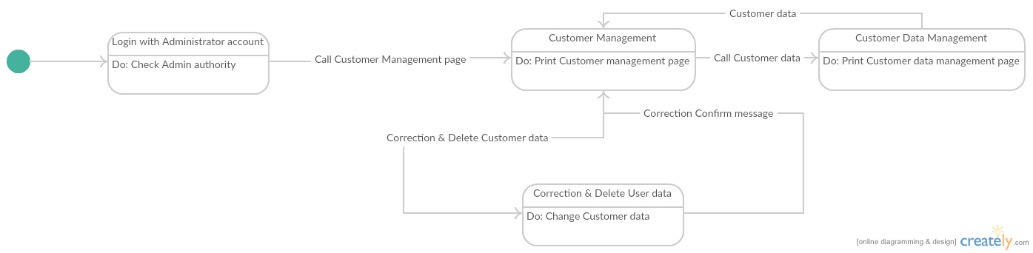
* + 1. ADM Diagram
       1. Overall Diagram
       2. Item Management

Diagram 35 Item Management State Diagram

Diagram 34 ADM overall State Diagram

* + - 1. User Management Diagram

Diagram 36 User Management State Diagram



1. **System Evolution**
   1. Objectives

사용자의 요구사항 변화, 하드웨어의 발전을 예측하고, 이를 본 시스템에 적용하여, 앞으로 본 시스템에 일어날 변화들과, 시스템이 앞으로 어떤 방향으로 발전하면 좋을지 서술한다. 또한, 앞으로 이 시스템을 이용할 사용자 및 시스템을 관리해 나갈 시스템 관리자 및 시스템 디자이너들에게 앞으로 어떠한 시스템 변화를 적용하면 좋을지 제안한다.

* 1. Evolution of Hardware
     1. 서버



Figure 4 서버(Server)

이미지 분석 및 딥 러닝 기술은 수 많은 데이터가 있다면, 더욱 정확해질 수 있다. 정확하지 않은 카테고리화와 태깅 기능은 사용자에게 불편을 증가시키게 되므로, 본 시스템에 있어서 정확도는 필수이다. 또한, 이러한 기능을 적용하는 데에 오랜 시간이 걸린다면, 이 또한 사용자에게 불만을 일으키게 될 것이다. 그렇기에, 대용량 서버를 이용하고, 많은 데이터를 수 많은 데이터를 빠르게 처리할 수 있게 된다면, 카테고리화와 태깅 기능에 있어, 작동 시간도 줄이고, 정확도를 높일 수도 있다. 또한, 구매자의 구매 패턴 분석에 있어서도, 더 빠른 정보처리가 가능할 것이다.

* 1. Evolution of User Requirement
     1. 상세화 된 태그 분류에 대한 요구

현재 시스템은 큰 카테고리 분류(상의, 하의, 신발)정도 밖에 불가능하다. 또한, 뽑아낼 수 있는 태그 또한, 그 수준에 그친다. 하지만 이후 사용자들은 좀 더 세분화된 태그 분류를 원할 것이다. 예를 들면, 쇼핑몰에 있어서 필수적인 색상, 옷의 재질 등을 요구할 것이다. 이러한 부분까지 태그 및 분류가 가능 하도록, 이미지 분석 및 딥 러닝 기능을 더욱 개발하고, 여러 사진을 학습시킬 필요가 있다.

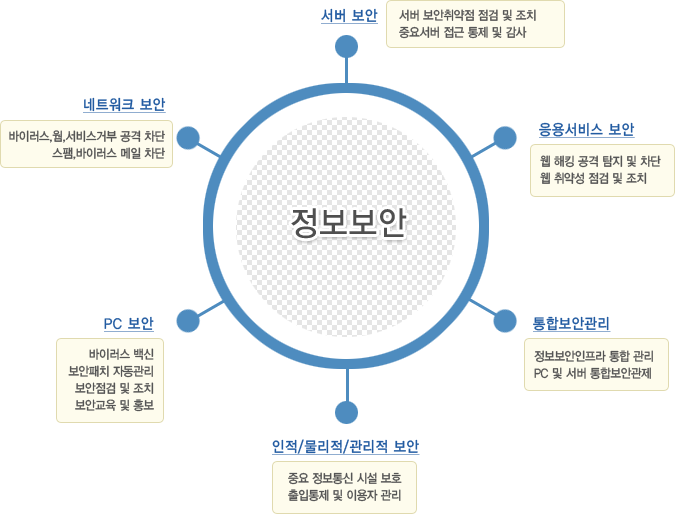
* + 1. 보안에 대한 요구

Figure 5 정보보안

신상 정보와 개인의 구매 정보에 대한 데이터를 다루고 있기 때문에, 외부에서의 불법적인 접근은 큰 위협이 된다. 따라서, 사용자들은 본 시스템에 해킹을 방지하고, 보안을 강화하도록 요구할 것이다.이에 대처하기 위해,외부에서의 접근에 대한 규제를 강화하고,본 시스템 내부에서 또한,각 사용자에 대한 수정 및 접근 권한 부여 절차 및 그 외 보안에 관련한 기능을 강화해야 한다.

* 1. Evolution of Environment
     1. 개인화 서비스의 필수화

개인 맞춤형 서비스는 IT 및 인터넷 서비스에서는 점차 필수화가 되어가고 있다. 소비자는 본인에게 맞춤화 된 서비스를 요구한다. 딥러닝을 이용하여 구매패턴을 분석해, 구매자에게 맞춤화 된 광고를 게시하는 서비스를 제공한다면, 쇼핑몰 수익을 극대화 할 수 있을 것이다.

Figure 6 개인별 맞춤 상품 추천

* + 1. 딥 러닝 기술의 보편화 및 상용화

현재, 딥 러닝 기술은 보편화, 상용화가 많이 되지 않은 기술이다. 하지만 점차, 기술이 발달함에 따라, 적용이 쉽고, 더욱 정확하며 세분화 된 분류가 가능한 기술이 보편화 된다면, 본 시스템 또한, 더욱 세분화 되고, 정확한 분류가 가능해질 것이다. 또한, 본 문서에서 예를 들은, 옷뿐만 아니라, 현재는 구분하기 힘든, 다른 상품(모양에 큰 차이가 없는 상품)에도 딥 러닝을 이용한 이미지 분석 시스템을 적용할 수 있을 것이다.

1. **Appendices**
   1. Objectives

본 시스템의 기능 및 개발 과정에 대한 자세한 정보를 제공한다. 본 시스템을 이용할 때 요구되는 최소 하드웨어 및 데이터베이스 요구사항뿐만 아니라, 본 시스템에서 사용한 핵심 기술에 대한 설명을 포함한다.

* 1. 데이터베이스 요구사항

구매자의 구매 정보, 상품 정보를 저장하기 위해 데이터베이스가 필요하다. 크게 판매자, 구매자, 상품으로 나누어 데이터를 관리하도록 한다. 데이터베이스 관리가 쉽도록 MySQL을 사용하며, PHP에 적용 가능하고, 딥 러닝 기능을 구현 할 수 있도록 설계해야 한다.

* 1. 사용자 시스템 요구사항
     1. 최소 시스템 사양
        1. 판매자 시스템 요구사항

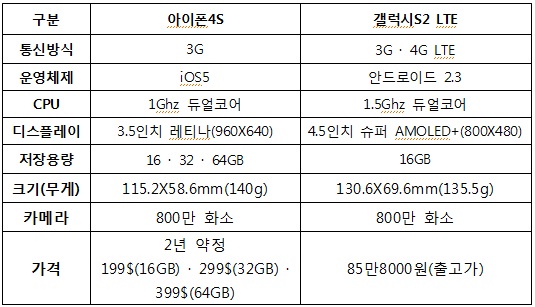
본 시스템의 핵심인 ‘이미지 분석을 이용한 태깅 기능’을 사용하기 위해서는, 이미지 분석 동안의 원활한 인터넷 환경 조성이 필요하며,정확한 ‘이미지’을 위해서는 2012년 발매된 스마트폰 이상의 화소를 가지고 있는 카메라 사용을 권장한다.

Figure 7 2012발매 된 스마트폰 카메라 화소

* + - 1. 구매자 시스템 요구사항

구매자에게는 본 시스템을 이용하기 위한 시스템 요구사항이 별도로 요구되지 않는다. 본 시스템의 원활한 사용을 위해서,인터넷이 끊기지 않는 컴퓨터 만을 요구하며,스마트폰에서의 사용은 지원하지 않는다.

* 1. 적용 개발 프로세스

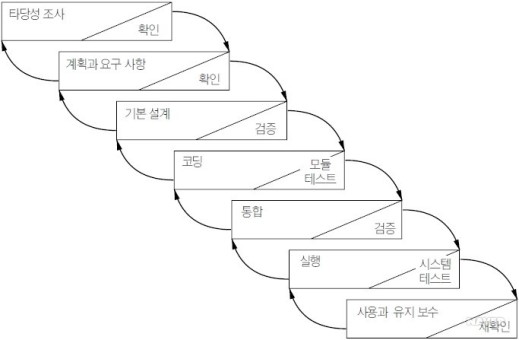
본 시스템의 개발 모델은 Waterfall Model을 따라 진행한다. Waterfall Model은 시스템의 개발을 계획-분석-설계-구현-시험-유지보수의 단계로 구분하고, 각 단계를 순차적으로 접근하며, 계획-분석-설계 단계를 개발 전 자세히 함으로써, Parallel development를 가능하게 한다.

Figure 8폭포수 모델(Waterfall Model)

본 시스템을 개발함에 있어서, 각 기능별로 Parallel Development의 필요성을 느꼈고, 이에 시스템 개발 모델로 Waterfall Model을 선택하였다.

* 1. 딥 러닝(Deep Learning)

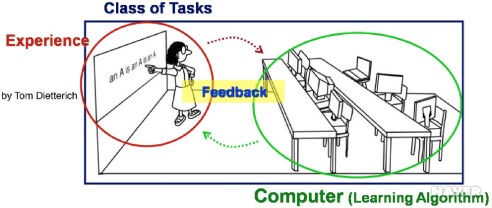
딥러닝은 사물이나 데이터를 군집화하거나 분류하는 데 사용하는 기술로, ‘기계 학습(Machine Learning)이라는 방법을 이용한다. 기계 학습은 수 많은 데이터를 이용하여, 새로운 데이터가 입력될 경우,기존의 데이터를 분석하여 비슷한 것끼리 분류하도록 하는 기술이다.데이터를 어떻게 분류할 것인가를 두고, ‘의사결정나무’, ‘베이지안망’, ‘서포트벡터머신’, ‘인공신경망’ 등 많은 기계학습 알고리즘이 등장했고,이 중 딥러닝은 ‘인공신경망’을 발전시킨 것이다.

Figure 9딥러닝(Deep Learning)

딥러닝은 크게 ‘지도 학습’과 ‘비지도 학습’으로 나눌 수 있다. ‘지도 학습’이란, 컴퓨터에 먼저 정보를 입력해 두는 방법이다. 수 많은 바지 사진을 주고,이와 비슷한 사진은 ‘바지’라고 알려주는 방식이다. ‘비지도 학습’이란, 이렇게 미리 입력하는 과정 없이, 컴퓨터가 스스로 학습하는 과정이다.

본 시스템은 이 두가지 과정 중 ‘지도 학습’을 이용하여 ‘옷’을 미리 지정된 큰 카테고리(티셔츠, 바지, 신발, 모자)로 구분하며, 기술의 발전에 따라, ‘비지도 학습’을 이용하여 점차 세분화 된 구분이 가능하도록 시스템을 발전시킬 것이다.

정리하면,딥러닝(deep learning)은 여러 비선형 변환기 법의 조합을 통해 높은 수준의 추상화(abstractions, 다량의 데이터나 복잡한 자료들 속에서 핵심적인 내용 또는 기능을 요약하는 작업)를 시도하는 기계학습(machine learning) 알고리즘의 집합으로 정의 되며, 큰 틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계 학습의 한 분야라고 이야기 할 수 있다. 어떠한 데이터가 있을 때 이를 컴퓨터가 알아 들을 수 있는 형태로 표현하고 이를 학습에 적용하기 위해 어떻게 하면 더 좋은 표현기법을 만들고 또 어떻게 이것들을 학습할 모델을 만들지에 대한 연구가 진행되고 있으며, 이러한 노력의 결과로 deep neural networks, convolutional deep neural networks, deep belief networks와 같은 다양한 딥 러닝 기법들이 컴퓨터 비전, 음성인식, 자연어 처리, 음성/신호처리 등의 분야에 적용되어 최첨단의 결과들을 보여주고 있다.

딥러닝에 관한 추가 자료는 맨 마지막에 별도로 첨부한다.

1. **Index**
   1. Objectives

본 문서를 읽는 독자를 위해 본 문서에 등장하는 단어를 알파벳(Alphabet) 순으로 정리하며, 그림(Figure, Diagram), 표(Chart) 등을 본 문서의 순서대로 정리한다.

* 1. Alphabet Index

ㄱ

개인 정보, 16, 25

결제하기, 26

고객 관리, 19

관리자, 17, 19, 23, 24, 25, 55

구매 정보, 17, 25, 57, 58

구매자, 15, 16, 17, 19, 20, 21, 22, 23, 25, 26, 56, 57, 58, 59

ㄷ

데이터, 13, 16, 17, 21, 22, 23, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 56, 57, 58, 61, 67

데이터베이스, 19, 21, 23, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 58

딥러닝, 12, 13, 61, 67

ㄹ

로그인, 13, 23, 25, 26, 29, 30, 37

ㅅ

상품 관리,, 19

상품 등록, 23, 24, 37

상품 삭제, 25, 34

상품 수정, 24, 25, 37

상품 정보, 16, 24, 25, 34, 58

샘플, 13, 22

서버, 13, 16, 17, 20, 21, 22, 23, 55, 56, 68

서브시스템, 11, 13

서술어, 9, 15

소스 코드, 13

소프트웨어, 13, 14

쇼핑몰, 11, 12, 13, 17, 22, 23, 25, 28, 56, 57, 67

시스템, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 25, 26, 27, 55, 56, 57, 58, 59, 60, 61

ㅇ

아이디, 13, 23, 25, 28, 30, 31

오픈소스, 13

요구사항, 8, 9, 15, 20, 55, 58, 59

웹, 13, 17, 22, 23, 24

이미지, 12, 13, 15, 16, 17, 20, 22, 24, 26, 56, 59

이미지 등록하기, 24

이미지 분석, 12, 15, 16, 20, 22, 24, 56, 59

인터넷, 11, 13, 14, 20, 25, 26, 57, 59, 67

인터넷 쇼핑몰(Internet Shoppingmall), 11

인터페이스, 9, 14, 16, 20

ㅈ

자연어, 8, 9, 15

장바구니, 26, 33, 34

저장하기, 19, 24, 25, 58

제약사항, 8, 15, 16

주문 내역 관리, 19

주문 목록, 24

ㅊ

차별화, 11, 12

ㅋ

카테고리, 12, 15, 16, 17, 20, 22, 23, 24, 25, 26, 56, 61

카테고리화, 12, 15, 16, 17, 20, 22, 23, 26, 56

ㅌ

태그, 12, 14, 15, 16, 20, 24, 25, 26, 56

태깅, 12, 14, 15, 16, 17, 20, 22, 23, 56, 59

ㅍ

판매자, 12, 15, 16, 17, 19, 20, 22, 23, 24, 25, 26, 58, 59

폭포수 모델, 14

ㅎ

하드웨어, 14, 55, 58

* 1. Figure Index

[Figure 1 인터넷 쇼핑몰 창업시장의 규모 11](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877734)

[Figure 2 딥 러닝 기반의 이미지 분석 15](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877735)

[Figure 3 상품 추천 기능 16](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877736)

[Figure 4 서버(Server) 56](#_Toc468877737)

[Figure 5 정보보안 57](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877738)

[Figure 6 개인별 맞춤 상품 추천 58](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877739)

[Figure 7 2012발매 된 스마트폰 카메라 화소 60](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877740)

[Figure 8 폭포수 모델(Waterfall Model) 61](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877741)

[Figure 9 딥러닝(Deep Learning) 62](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877742)

* 1. Diagram Index

[Diagram 1 System Architecture Overview 19](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877743)

[Diagram 2 Detail System Architecture 20](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877744)

[Diagram 3 System Overview Class Diagram 29](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877745)

[Diagram 4 Sign\_in Class Diagram 31](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877746)

[Diagram 5 Search Class Diagram 33](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877747)

[Diagram 6 Board Class Diagram 36](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877748)

[Diagram 7 ADM Class Diagram 38](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877749)

[Diagram 8 Image Analysis Class Diagram 40](#_Toc468877750)

[Diagram 9 Sign\_up Sequence Diagram 41](#_Toc468877751)

[Diagram 10 Sign\_in Sequence Diagram 41](#_Toc468877752)

[Diagram 11 Search Sequence Diagram 42](#_Toc468877753)

[Diagram 12 Cart Sequence Diagram 43](#_Toc468877754)

[Diagram 13 Purchasing Cart Sequence Diagram 43](#_Toc468877755)

[Diagram 14 Purchasing item Sequence Diagram 44](#_Toc468877756)

[Diagram 15 Purchasing and Payment Sequence Diagram 44](#_Toc468877757)

[Diagram 16 Board Sequence Diagram 45](#_Toc468877758)

[Diagram 17 Mypage Sequence Diagram 46](#_Toc468877759)

[Diagram 18 ADM Sequence Diagram 46](#_Toc468877760)

[Diagram 19 Item Add and Correction Sequence Diagram 47](#_Toc468877761)

[Diagram 20 User Management Sequence Diagram 47](#_Toc468877762)

[Diagram 21 Order Management Sequence Diagram 48](#_Toc468877763)

[Diagram 22 System Overview State Diagram 49](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877764)

[Diagram 23 Sign\_up State Diagram 49](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877765)

[Diagram 24 Sign\_in State Diagram 50](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877766)

[Diagram 25 Search State Diagram 50](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877767)

[Diagram 26 Choose\_Item State Diagram 51](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877768)

[Diagram 27 Cart State Diagram 51](#_Toc468877769)

[Diagram 28 Purchasing State Diagram 52](#_Toc468877770)

[Diagram 29 Purchasing Cart State Diagram 52](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877771)

[Diagram 30 Payment State Diagram 53](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877772)

[Diagram 31 Shipment State Diagram 53](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877773)

[Diagram 32 Mypage State Diagram 54](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877774)

[Diagram 33 Board State Diagram 54](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877775)

[Diagram 35 Item Management State Diagram 55](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877776)

[Diagram 34 ADM overall State Diagram 55](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877777)

[Diagram 36 User Management State Diagram 56](file:///C:\Users\JEONGS~1\DOCUME~1\카카오~1\SOFTWA~2.DOC#_Toc468877778)

1. **Reference**
2. 스마트과학관-사물인터넷머신러닝 (<http://terms.naver.com/entry.nhn?docId=3386834&cid=58369&categoryId=58369>)
3. 컴퓨터인터넷IT용어대사전,전산용어사전편찬위원회, 2011.01.20 일진사(<http://terms.naver.com/list.nhn?cid=42344&categoryId=42344>)
4. 두산백과(<http://terms.naver.com/list.nhn?cid=40942&categoryId=40942>)
5. 홈앤쇼핑,맞춤상품 추천 ‘큐레이션 서비스’개시 빅데이터 활용해 알고리즘 분석…”지속적 방문 재구매 기대”, 이상혁, 2016.05.13, Business Report (<http://www.businessreport.kr/news/articleView.html?idxno=3128>)
6. 영화 <채피>로 알아보는 쉬운 딥러닝(Deep learning) 이야기, Semiconductor, 2015.07.07, AMKOR in STORY (<http://amkorinstory.com/797>)
7. 쇼핑몰이 나를 안다!..프리미엄몰‘엘롯데’ 오픈,문정태, 2012.03.26, 중앙일보(<http://news.joins.com/article/7715813>)
8. ‘가격은 아이폰,속도는 갤럭시’, 정병묵, 2011.10.05, 이데일리 (<http://www.edaily.co.kr/news/NewsRead.edy?SCD=JE31&newsid=02266486596409968&DCD=A00503&OutLnkChk=Y>)
9. 스마트폰 카메라 어디까지 진화할까?,양태훈, 2012.04.15, 아이티투데이 (<http://www.ittoday.co.kr/news/articleView.html?idxno=26587>)
10. [2016 서울미래컨퍼런스]딥러닝의 힘… AI 의사, 불치병을 고친다. 인공지능으로 가능한 미래,유용하, 2016.10.05, 서울신문(<http://www.seoul.co.kr/news/newsView.php?id=20161006004003&wlog_tag3=naver>)
11. [미래포럼]개인화 서비스로의 진화, 설원희,2004.09.08, 전자신문 (<http://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=105&oid=030&aid=0000081650>)
12. 정보보안,인하대학교 정보통신처(<https://cic.inha.ac.kr/info_service/serv_030501.aspx>)
13. 모두를 위한 머신러닝/딥러닝 강의,hunkim, Github (<http://hunkim.github.io/ml/>)
14. Classification: Instant Recognition with Caffe, Jupyter (<http://nbviewer.jupyter.org/github/BVLC/caffe/blob/master/examples/00-classification.ipynb>)
15. DIY Deep Learning for Vision: a Hands-On Tutorial with Caffe, Evan Shelhamer, Jeff Donahue, Jon Long, YangqingJia, Ross Girshick (<https://docs.google.com/presentation/d/1UeKXVgRvvxg9OUdh_UiC5G71UMscNPlvArsWER41PsU/edit#slide=id.gc2fcdcce7_216_0>)
16. MS, ‘캐터펄트’ 서버 설계 개선 … FPGA 속도 높여 AI에 최적화,Agam Shah, 2016.08.30, CIOKorea(<http://www.ciokorea.com/news/31045>)
17. ConvNetJS, Deep Learning in your browser(<https://docs.google.com/presentation/d/1UeKXVgRvvxg9OUdh_UiC5G71UMscNPlvArsWER41PsU/edit#slide=id.gc2fcdcce7_216_0>)

**Appendices 2**

**딥 러닝이란?**

딥러닝(deep learning)은 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화(abstractions, 다량의 데이터나 복잡한 자료들 속에서 핵심적인 내용 또는 기능을 요약하는 작업)를 시도하는 기계학습(machine learning) 알고리즘의 집합으로 정의 되며, 큰틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야라고 이야기 할 수 있다. 어떠한 데이터가 있을 때 이를 컴퓨터가 알아 들을 수 있는 형태로 표현하고 이를 학습에 적용하기 위해 어떻게 하면 더 좋은 표현기법을 만들고 또 어떻게 이것들을 학습할 모델을 만들지에 대한 연구가 진행되고 있으며, 이러한 노력의 결과로 deep neural networks, convolutional deep neural networks, deep belief networks와 같은 다양한 딥 러닝 기법들이 컴퓨터비전, 음성인식, 자연어처리, 음성/신호처리 등의 분야에 적용되어 최첨단의 결과들을 보여주고 있다.

출처 : https://ko.wikipedia.org/wiki/%EB%94%A5\_%EB%9F%AC%EB%8B%9D

**프로젝트에 사용될 TensorFlow의 기본 설명**

텐서플로(TensorFlow) 는 구글 제품에 사용되는 머신러닝(기계학습)을 위한 오픈소스 소프트웨어 라이브러리이다. 구글 내 연구와 제품개발을 위한 목적으로 구글 브레인(Google Brain) 팀이 만들었고 2015년 11월 9일 아파치 2.0 오픈소스 라이센스로 공개되었다. 안드로이드와 iOS같은 모바일 환경은 물론 64비트 리눅스, OS X 의 데스크탑이나 서버 시스템의 여러개의 CPU와 GPU에서 (GPU에서 일반 연산을 수행하게 하는 CUDA 확장기능을 사용) 구동될 수 있다. 그래프의 노드(Node)는 수치 연산을 나타내고 엣지(edge)는 노드 사이를 이동하는 다차원 데이터 배열(텐서,tensor)를 나타낸다. 유연한 아키텍처로 구성되어 있어 코드 수정 없이 데스크탑, 서버 혹은 모바일 디바이스에서 CPU나 GPU를 사용하여 연산을 구동시킬 수 있다. 텐서플로우는 원래 머신러닝과 딥 뉴럴 네트워크 연구를 목적으로 구글의 인공지능 연구 조직인 구글 브레인 팀의 연구자와 엔지니어들에 의해 개발되었다. 하지만 이 시스템은 여러 다른 분야에도 충분히 적용될 수 있다.

출처 : https://ko.wikipedia.org/wiki/TensorFlow

**TensorFlow의 장점**

데이터 플로우 그래프를 통한 풍부한 표현력

코드 수정 없이 CPU/GPU 모드로 동작

아이디어 테스트에서 서비스 단계까지 이용 가능

계산 구조와 목표 함수만 정의하면 자동으로 미분 계산을 처리

Python/C++를 지원하며, SWIG를 통해 다양한 언어 지원 가능

**이미지 인식 방식**

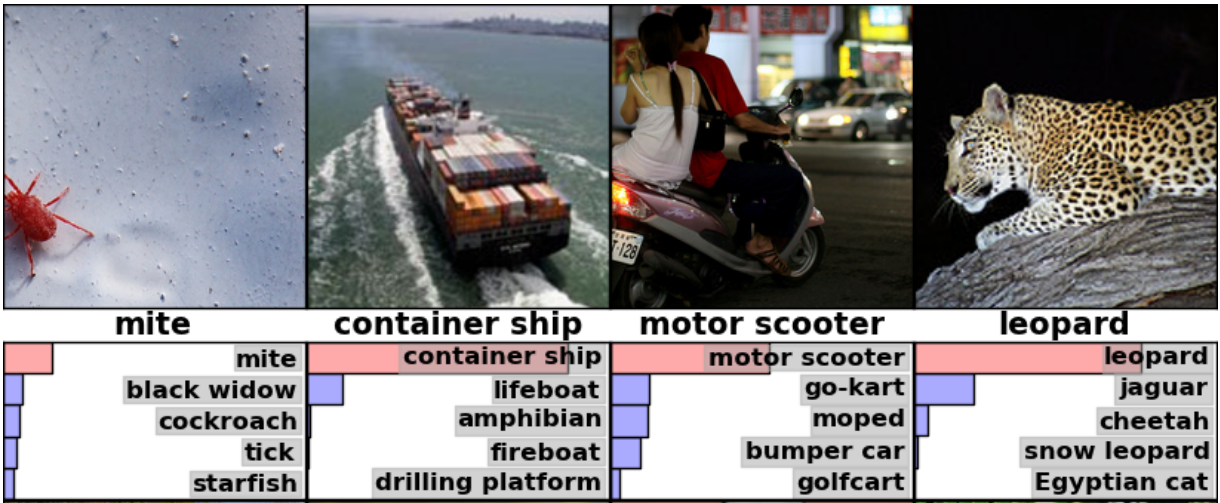
우리의 뇌를 생각하면 시각적으로 인식하는 일은 쉬워 보인다. 보통 사람이라면 사자와 재규어를 구별할 줄 알고 표지판을 읽을 수 있으며 다른 사람의 얼굴 또한 어렵지 않게 인식할 수 있다. 그러나 이는 이미지를 인식하는 뇌의 능력이 놀라울 정도로 뛰어나기 때문에 가능한 것이지 이와 같은 일을 컴퓨터를 통해 해결하는 것은 매우 어려운 문제다.

+

지난 수 년간 기계학습 분야는 이미지 인식에 대해 엄청난 진전을 이루어 냈다. 특히 딥 러닝 기법의 하나인 convolutional neural network를 통해 혁신적인 성과를 거두었는데, 일부 분야에서는 사람의 인식 능력에 버금가거나 더 나은 결과를 보여주기도 했다.

연구자들은 학계에서 시작된 컴퓨터 비전 프로젝트인 ImageNet에서 자신들의 작업을 검증해왔고, 그들의 연구는 QuocNet, AlexNet, Inception (GoogLeNet), BN-Inception-v2와 같은 최신식 모델을 만들어냈다. 구글 내부 연구자와 외부 연구자 모두 이러한 모델을 설명하는 자료를 발표해 왔지만 자료가 널리 배포되고 있지는 않다. 그래서 TensorFlow는 구글이 개발한 이미지 인식의 가장 최신 모델인 Inception-v3를 활용하는 코드를 공개한다.

Inception-v3는 ImageNet의 Large Visual Recognition Challenge에서 2012년 데이터를 사용하여 훈련된 모델이다. 모든 이미지를 "얼룩말", "달마시안", "식기세척기"와 같은 1000 classes로 분류하는 것이 컴퓨터 비전의 표준 작업이다. 다음의 예는 AlexNet이 몇 가지 사진을 분류한 결과이다:



모델의 성능을 비교할 때는 "top-5 error rate"를 측정한다. 이는 모델이 가장 높은 확률로 예측한 5가지 예측이 정답이 아닌 빈도를 검토하는 것이다. 2012년 검증 데이터 세트에서 나타난 각 모델의 top-5 error rate는 AlexNet이 15.3%, BN-Inception-v2이 6.66%였고 Inception-v3는 3.46%를 달성했다.

ImageNet 챌린지에서 사람의 성과는 어떨까? Andrej Karpathy가 blog post에서 밝힌 바에 의하면 그의 top-5 error rate는 5.1%였다고 한다.

출처 : https://tensorflowkorea.gitbooks.io/tensorflow-kr/content/

**이미지 인식에 사용되는 Convolutional Neural Network (CNN)의 기본 특징**

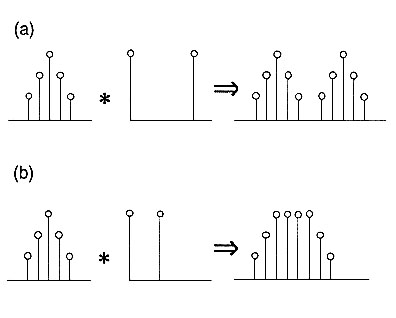
1989년에 제안된 이 모델은 아직까지도 많이 쓰이는 deep network 모델이다. 특히 computer vision에 특화된 이 네트워크는 인간의 시신경 구조를 모방하여 인간이 vision 정보를 처리하는 것을 흉내낸 모형이다. CNN은 overfitting issue를 모델 complexity를 줄이는 것으로 해결한다. CNN은 convolution layer와 pooling layer라는 두 개의 핵심 구조를 가지고 있는데, 이 구조들이 model parameter 개수를 효율적으로 줄여주어 결론적으로 전체 model complexity가 감소하는 효과를 얻게 된다.

**CNN의 Convolution Layer**

Convolution layer를 설명하기 전에 먼저 convolution operation에 대해 알아보자. Convolution이란 signal processing 분야에서 아주 많이 사용하는 operation으로, 다음과 같이 표현된다.

s(t)=(x∗w)(t)=∫x(a)w(t−a)da.

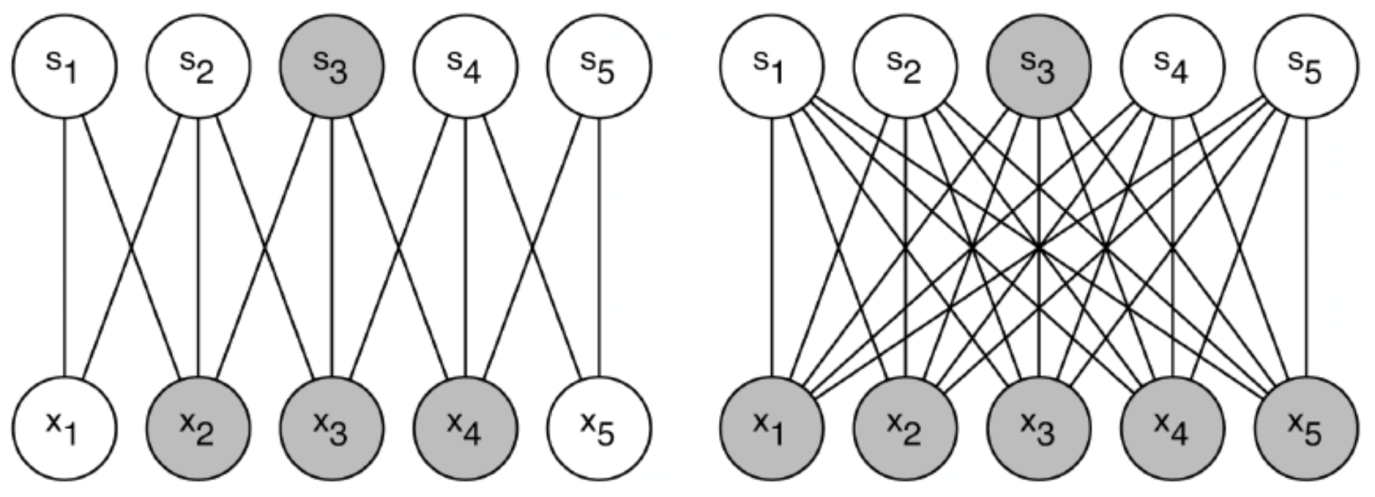
예를 들어 이 operation은 주어진 데이터 $x$에 filter $w$를 사용해 데이터를 처리할 때 사용된다. 이 operation을 적용한 간단한 예를 보자.



이렇듯 convolution은 어떤 filter를 사용하여 주어진 image의 적절한 feature를 뽑아내기 위해 사용했던 operation이다. 이때 $s(t)$를 데이터 $x$의 feature map이라고 부른다. Deep learning이 널리 사용되기 이전에는 다른 머신러닝 framework에 이미지를 input으로 넣고 처리하기 위해서는 먼저 filter를 고르고 그 filter로 image를 convolution하는 preprocessing을 거쳐서 적절한 feature map을 얻어낸 이후에 그것을 machine learning framework의 input으로 넣어 돌리는 방식을 사용했었다. 그렇기 때문에 이런 feature engineering이 전체 performance에 큰 영향을 미치는 경우가 많았다. 어떤 filter를 선택할 것이며, 얼마나 많은 filter를 고를 것인지 등의 영역은 feature engineering의 영역이고, 이론적인 영역이 아니기 때문에 machine learning 분야에서는 큰 관심을 두는 분야는 아니었다. 데이터는 잘 처리되었다고 가정하고 그 데이터를 사용해 어떤 좋은 알고리즘을 개발하느냐가 그 동안 머신러닝 framework들의 아이디어였다면, CNN의 핵심 아이디어는 preprocessing이 실제 performance에 크게 영향을 미치니까, 아예 이 preprocessing을 가장 잘해주는, 가장 좋은 feature map을 뽑아주는 convolution filter를 learning하는 모델을 만들어버리자는 것이다.

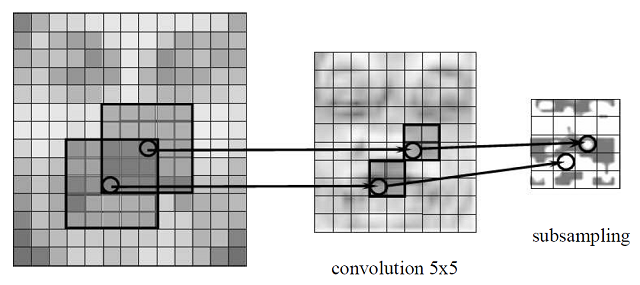
최대한 작은 complexity를 가지면서 우수한 filter를 표현하기 위한 CNN의 핵심 아이디어는 다음 세 가지이다: sparse interactions (혹은 sparse weight라고도 한다), parameter sharing (혹은 tied wieght라고도 한다), equivariant representations. 즉, CNN은 layer와 layer간에 모든 connection을 연결하는 대신 일부만 연결하고 (sparse weight), 그리고 그 weight들을 각각 다른 random variable로 취급하여 따로 update하는 대신 특정 weight group들은 weight 값이 항상 같도록 parameter를 share한다 (parameter sharing). 그리고 앞의 아이디어를 잘 활용하여 shift 등의 transform에 대해서 equivariant한 (자세한 내용은 밑에서 설명한다) representation을 learning하도록 모델을 구성한다.

Sparse weight를 사용하게 되면 모든 가능한 connection을 사용하는 것 보다 훨씬 적은 표현형을 learning하게 된다는 단점이 있지만, 반대로 model의 complexity가 낮아진다는 장점이 존재한다. CNN은 vision과 관련된 task를 수행하도록 design된 network라는 것은 이미 언급한바 있다. 이런 vision 데이터를 처리하는 task를 하게 될 경우에는 주어진 input의 dimension에 비해 실제 필요한 feature의 dimension은 극히 적다는 domain knowledge를 우리는 이미 가지고 있다. 즉 input인 이미지의 경우 픽셀 값이 적으면 몇 백에서 많으면 몇 백만에 이를 정도로 dimension이 엄청나게 높지만, 우리가 필요한 ‘feature’는 그 중에서도 극히 일부 영역, 이를테면 edge detection 등의 그에 비해 훨씬 적은 dimension으로 표현 가능하기 때문에 최대한 parameter를 줄여서 더 효율적인 feature map을 뽑아내기 위하여 weight를 sparse하게 사용한다. 이를 그림으로 표현하면 아래와 같다.



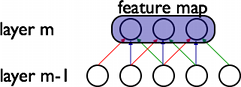
이 그림은 같은 output $s\_3$에 영향을 주는 edge들과 input node를 표현한 그림이다. 왼쪽은 가능한 connection이 전부 있는 것이 아니라 그 일부만 존재하고, $s\_3$에 영향을 주는 input이 $x\_2, x\_3, x\_4$ 뿐이지만 오른쪽은 모든 가능한 connection이 있어서 model parameter의 개수가 크게 차이가 나고 모든 input이 $s\_3$에 영향을 주는 것을 알 수 있다. 실제 image 데이터를 처리하기에는 왼쪽 모델이 조금 더 나은데, 그 이유는 한 feature를 결정하기 위해서 모든 image 정보가 필요한 것이 아니라, image의 일부분만 필요하기 때문이다. 예를 들어 내가 face segmentation, 즉 얼굴 사진에서 눈 코 입 등을 찾아내는 task를 수행한다고 하면, 주어진 사진에서 ‘눈’이 어디인지 표현하기 위해서 모든 이미지가 다 필요한 것이 아니라 눈 주변의 local한 데이터만 필요할 것이라고 유추할 수 있다. 오른쪽 그림은 필요하지 않은 배경까지 모두 고려하여 눈에 대한 정보를 찾는 셈이고, 왼쪽 그림은 local한 정보만을 주고 눈에 대한 정보를 처리하게 하는 것이다. 따라서 vision task를 처리하기에는 적절한 sparse weight가 더 효율적인 모델이라는 것을 알 수 있다. 때문에 CNN의 convolution layer는 hidden node 하나가 image의 local한 patch와 연결되어있는 형태로 되어있다. 예를 들어 한 hidden node 마다 image의 3 by 3 patch만을 연결하는 방식이다. 그림으로 표현하면 아래와 같은 식이다.

출처 : [Code project - Online handwriting recognition using multi convolution neural networks](http://www.codeproject.com/Articles/523074/Online-handwriting-recognition-using-multi-convolu)

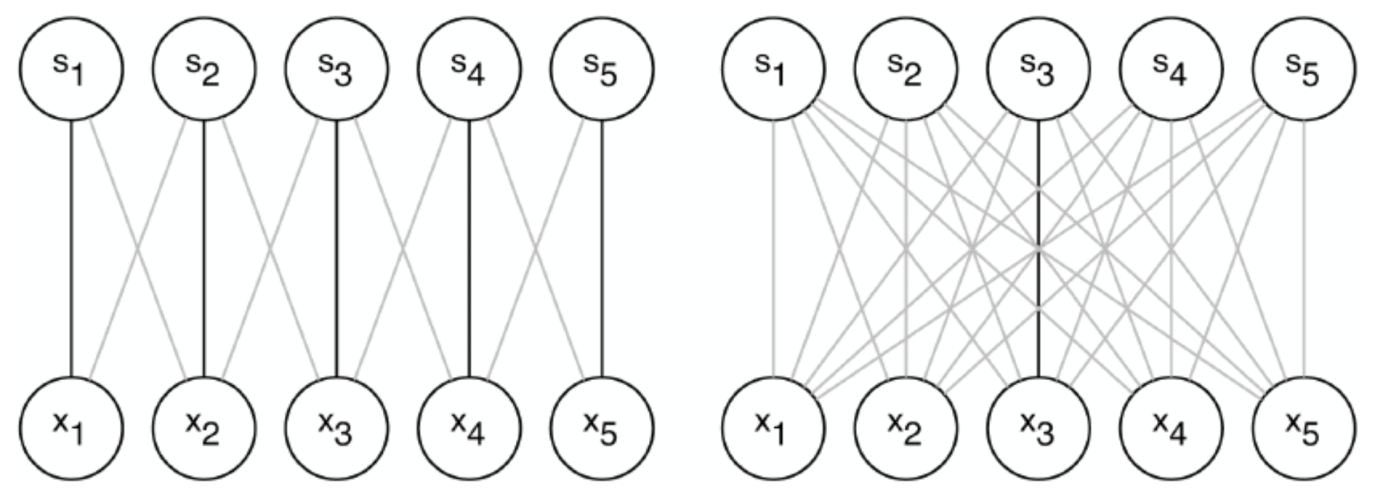


여기에서 subsampling은 일단 나중에 설명하도록 하고 (subsampling part가 pooling layer에 해당한다) 가장 왼쪽의 image data의 일부분에 해당하는 patch만 다음 hidden layer의 한 unit에 연결하는 것이다. 이런 식으로 네트워크를 만들게 되면, patch size에 따라 다음 feature map의 size가 결정될 것이다. 예를 들어 100 by 100 이미지에서 5 by 5 patch를 사용해 convolution layer를 구축할 경우, 이 layer의 feature map은 96 by 96이 될 것이다.

CNN은 이런 sparse weight에 parameter sharing을 또 더하여 vision task에 최적화된 network를 learning하게 된다. Parameter를 share하게 되면 그러지 않는 것과 비교하여 보다 적은 parameter만을 가지게 되므로 model의 complexity가 줄어드는 효과가 있을 뿐 아니라, 각각의 patch마다 따로 필터를 learning하는 대신, 모든 patch에 동일한 필터를 적용하도록 강제하는 효과가 있다. CNN은 아래 그림과 같이 각각의 hidden node들이 같은 location에 대해 같은 weight를 가지도록 설정하여 모든 hidden node들이 각각 다른 patch에 대해 같은 filter를 처리하는 것과 같은 형태로 모델을 디자인한다.



위 그림에서 같은 색으로 칠해진 edge는 서로 같은 weight를 가진다. 위 그림에서 볼 수 있듯, CNN은 fully connected layer를 가지지 않고, 그 sparse한 weight들에서도 서로 weight를 공유하도록 설정되어있다. 그렇지 않은 네트워크와 비교해보면 다음과 같다.

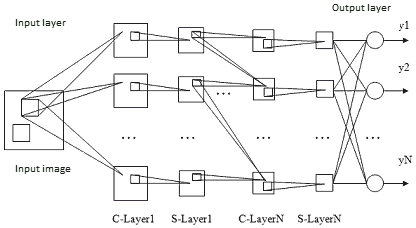


각각의 그림에서 검은색으로 연결된 edge들은 서로 같은 parameter를 가진다. 즉, 왼쪽은 한 번에 5개의 edge가 같은 weight를 가지지만, 오른쪽은 하나의 parameter로 한 개의 edge만 표현할 수 있다. 이렇게 표현하게 되면 convolution layer operation이 간단한 matrix multiplication으로 주어지게 되어 gradient를 계산하기 한 층 더 수월해진다는 장점도 존재한다.

마지막으로 equivalent representations는 위와 같은 sparse weight와 tied weight를 어떤 특정한 형태로 효율적으로 배치하게 되었을 때, 주어진 input의 변화에 대해 output이 변화하는 방식이 equivariant해지는 현상을 의미한다 (equivalent가 아니다). 예를 들어 function $f$가 function $g$와 equivariant하다는 의미는, $f(g(x)) = g(f(x))$인 경우를 말한다. 이미지 처리를 예로 들면 $g$는 임의의 linear transform이라고 할 수 있다. Shift, rotate, scale등의 image에 대한 transform들이 그것인데, 우리는 같은 이미지가 돌아가거나 움직이거나 살짝 scale되더라도 그 이미지가 어떤 이미지인지 잘 판별할 수 있지만, 컴퓨터에게는 그런 transform이 픽셀 값이 완전히 바뀌는 결과를 낳기 때문에 어떤 정보인지 판별하기 어려운 것이다. 그런데 만약 우리가 어떤 transform $g$에 대해 equivariant representation을 만들어내는 network $f$를 만들 수 있다면, input이 shift되거나 rotate되더라도 항상 적절한 representation을 가지도록 할 수 있을 것이다. (실제 CNN은 shift에만 equivariant하다.)

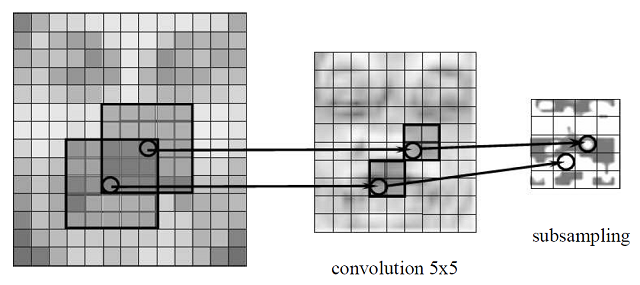
즉, 앞에서 shared parameter가 각각의 patch에 대해 같은 filter를 처리하는 것처럼 설정하였기 때문에, 만약 image가 shift되더라도 feature map의 형태가 크게 뒤틀리는 것이 아니라, feature map도 image와 함께 shift되는 형태를 보이게 될 것이다.

그런데 실제로는 한 image에 한 개의 filter가 아니라 여러 개의 filter가 필요할 수도 있다. 앞에서 설명한 convolution layer는 한 개의 convolution filter를 표현할 뿐이지만, 실제로는 이런 convolution filter가 한 개가 아니라 여러 개 만든 다음 그 값들을 concate하여 feature map을 표현해야할 수도 있다. 그렇기 때문에 실제로 CNN model은 한 개의 convolution layer가 아니라 아래와 같이 여러 개의 convolution layer가 결합된 꼴을 하고 있다. 참고로 공식적으로는 각각의 layer 혹은 filter를 kernel이라 하고, 그 kernel들이 모여있는 것을 한 layer로 부른다.



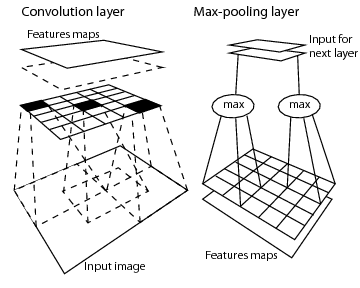
**CNN의 Pooling Layer**

Convolution Layer만 여러 개 연결하여 deep network를 구성하는 것도 가능하지만, 실제로는 더 dimension이 낮은 feature map을 얻기 위하여 subsampling이라는 것을 하게 된다. 앞에서 예로 들었던 것처럼 100 by 100 이미지에 5 by 5 convoltion patch size를 가지는 convolution layer를 연결할 경우 feature map의 size는 96 by 96이 되는데, 사실 이 96 by 96 feature map은 서로 매우 highly correlated 되어있는 값이 것이다. 특히 서로 이웃해있을수록 겹치는 영역이 많기 때문에 거의 비슷한 값을 가질 것이라고 예상할 수 있다. 아래 그림을 보자.



이미 앞에서 나왔던 그림이지만 설명을 위하여 다시 가져왔다. Pooling layer는 convolution layer의 feature map을 조금 더 줄여주는 역할을 한다. 전체 feature map을 그대로 들고가는 대신, 예를 들어 96 by 96 image feature map을 2 by 2 patch들로 쪼개는 것이다. 이렇게 할 경우 총 48 by 48 개의 output이 생기게 될텐데, subsampling이라는 것은 각각의 2 by 2 patch는 max, average 등의 operation을 행하는 것을 의미한다. 보통 max operation을 사용하고, 이 경우 간단하게 max pooling을 사용한다 라고 이야기 한다. 가끔 average pooling을 사용하는 경우도 있지만 보통 classification을 위한 모델들은 max pooling을 사용한다.

CNN은 이렇게 convolution layer와 pooling layer가 결합된 형태로 deep 하게 구성이 된다. 개인적으로 아래 그림이 CNN의 convolution layer와 max pooling layer를 잘 표현하는 그림이라고 생각한다.



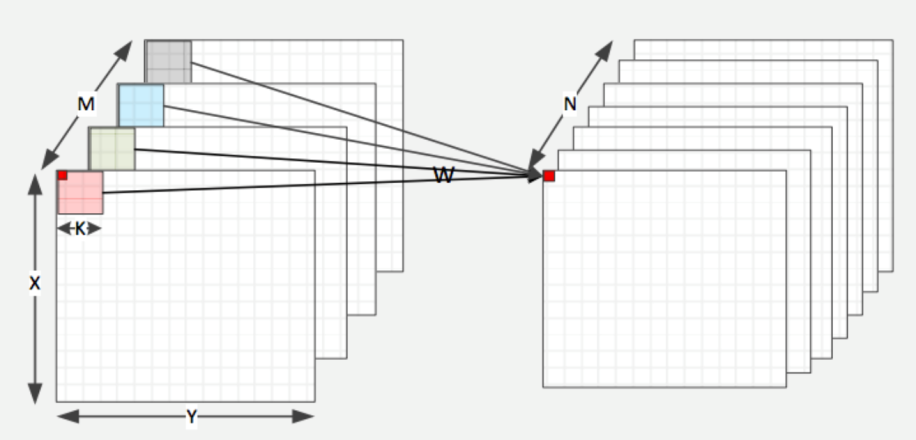
**CNN의 Backpropagation 알고리즘**

CNN의 기본 model은 알았으니 이제 이 network의 parameter를 어떻게 learning해야할지 알아보자. 먼저 간단한 max pooling layer 부터 살펴보자. Pooling layer는 아래 p by q size의 patch 중에서 max 값을 선택하는 layer이다. 때문에 이를 수식으로 표현해보면 다음과 같이 쓸 수 있다. $(x,y)$는 pooling layer feature map의 x,y좌표를 나타내고, ($h\_l$은 l번째 layer의 hidden variable들)

hl+1(x,y)=maxa−p≤a≤a+p,b−q≤b≤b+q(hl(x+a,y+b))

Parameter는 없으므로 $\frac{\partial h\_{l+1} }{\partial h\_{l}}$만 계산하면 된다. 이 경우 주어진 $(x,y)$가 만약 max pooling을 통해 선택된 값이라면 값을 그대로 passing하고, 만약 선택되지 않은 값이라면 0을 할당하면 된다.

Convolution layer는 operation이 꽤 복잡한데, 먼저 아래 그림을 보자.



이때 l+1 번째 layer 중에서 i 번째 convolution filter의 x,y 좌표 값은 아래와 같이 표현된다. 아래 conv layer의 kernel은 m 개, 위 conv layer의 kernel은 n개 라고 해보자.

hl+1(i,x,y)=∑j=1m∑a=1p∑b=1qhl(j,x+a,y+b)∗w(i,n;a,b)

미분 값을 계산해보면, parameter의 gradient는 $\frac{E}{\partial w} = \sum\_x \sum\_y \frac{E}{h\_{l+1} } (x,y) h\_l (x,y)$와 같이 바로 전 layer의 pixel값에 대해 gradient 값을 곱한 것을 전부 더한 형태로 구할 수 있고, $\frac{E}{\partial h\_l} $은 weight w로 이전 layer의 gradient를 convolution한 것들을 전부 더한 것과 같은 결과를 얻게 된다.

CNN의 모든 operation들은 단순 연산이 많고 branch가 없기 때문에 core가 많고, 모든 core가 하나의 operation pointer를 공유하는 GPU를 사용해 효율적으로 parallization하기 좋다.

출처 : http://sanghyukchun.github.io/75/