**Design Specification**

**Clothes\_hanger**

**SWEngineering Team #1**

이동건

박주형

김범준

최예은

유창휘

목차

1. Preface 6

1.1 Objective 6

1.2 Readership 6

1.3 Document Structure 6

A. Preface 6

B. Introduction 6

C. System Architecture 6

D. User Perspective Process 7

E. 회원가입 7

F. 로그인 7

G. 판매자의 파일 업로드 7

H. 딥러닝 이미지 분류 7

I. Protocol Design 8

J. Database Design 8

K. Testing Plan 8

L. Develop Environment 8

1.4 Version of Document 8

A. Version Format 8

B. Version Management Policy 8

C. Version History 8

2. Introduction 10

2.1 Overview 10

2.2 Applied Diagram & Tools 10

3. System Architecture 13

3.1 Objective 13

3.2 System Organization 13

A. User Management System 14

B. Upload Management System 15

C. Deep Learning Classification 16

*3.3 Deployment diagram 17*

4. User Perspective Process 18

4.1 Objective 18

4.2 User Perspective Process Flowchart 18

4.3 User Interface 19

4.3.1. 공통(MD, 판매자) 19

4.3.2. MD 20

4.3.3. 판매자 22

5. 회원가입(Sign up) 26

5.1 Objective 26

5.2 Class Diagram

5.2.1 MD 회원가입 정보 26

5.2.2 판매자 회원가입 정보 26

5.2.3 Company DB handler 27

5.2.4 MD user DB handler / 판매자 user DB handler 27

5.3 Sequence Diagram 28

*5.4 Features 29*

6. 로그인(Login) 30

6.1 Objective 30

6.2 Class Diagram 30

6.2.1 로그인 30

6.2.2 DB handler 30

6.3 Sequence Diagram 31

*6.4 Features 31*

7. 판매자의 파일 업로드(Multiple file upload) 32

7.1 Objective 32

7.2 Class Diagram 32

7.2.1 판매자 32

7.2.2 Clothes Hanger 32

7.3 Sequence Diagram 33

7.4 State Diagram 33

7.5 Features 34

8. Deep Learning Image Classification 35

8.1 Objective 35

8.2 Image Model Overview 35

8.3 How to collect data 37

A. Web Scraping 37

8..4 Training and validation 38

9. Protocol Design 41

9.1 Objective 41

9.2 JSON 41

9.3 HTTP 41

9.4 Protocol Description 42

A. 회원가입

B. 로그인 42

C. 판매자의 파일 업로드 43

D. 딥러닝 이미지 분류 44

10. Database Design 46

10.1 MySQL 46

11. Testing Plan 47

11.1 Objectives 47

11.2 Testing Policy 47

11.3 Test Case 48

11.3.1 회원가입 및 로그인 48

11.3.2 판매자의 파일 업로드 50

11.3.3 딥러닝 이미지 분류 51

12. Development Environment 52

12.1 Objectives 52

12.2 Programming Language & IDE 52

A. Programming Language 52

B. IDE 53

12.3 개발규정 54

**1. Preface**

1.1 Objective

Preface에서는 본 문서의 대상이 되는 독자를 명확히 하고 문서의 전체 구조와 그에 따른 세부적인 역할을 소개한다. 또한 Version History로 본 문서의 버전 변경 기록과 이유, 변경사항을 기술한다.

1.2 Readership

본 문서는 우리 시스템의 설계, 개발 및 유지보수에 관여하는 모든 사람들을 독자로 간주한다. 서브시스템(Subsystem)과 시스템을 설계하는 시스템 아키텍처(System architecture)를 비롯하여 시스템 개발에 관여하는 소프트웨어 개발자 및 관련업체뿐만 아니 라 시스템 유지보수를 담당하는 모든 외주 업체들이 그 대상이 될 수 있다. 경우에 따라 System End-user나Client engineer 역시 독자로 상정할 수 있다.

1.3 Document Structure

본 문서는 총 12개 장으로 이루어져 있다. Preface, Introduction, System Architecture, SubSystems, Protocol design, Database design, Testing Plan, Development Environment 순으로 진행된다.

1. Preface  
   본 문서의 대상이 되는 독자를 명확히 하고, 문서의 전체 구조 및 세부 내용을 간략히 소개한다. Version History로 본 문서의 버전 변경 기록과 사유, 변경사항을 기술한다.
2. Introduction  
   본 문서에서 시스템을 설계할 때 사용되는 다이어그램의 종류와 UML Tool을 서술한다.
3. System Architecture  
   System Architecture 에서는 전체 시스템에 대한 개요를 기술한다. Block diagram을 통해 시스템 전체 구조를 설명하고, 각 하위 시스템들의 구성과 관계를 package diagram을 통해 설명한다. 하위 시스템에 대한 세부적인 설명은 아래의 각 장을 통해 기술한다.
4. User Perspective Process  
   개발자가 이해하는 시스템의 구성과 사용자가 이해하는 시스템의 구성은 다르다. 개발자는 시스템을 본인이 개발한 모듈 또는 컴포넌트 단위로 이해하는 반면, 사용자는 어떤 페이지가 나타나고, 어떤 버튼을 언제 어디서 클릭해야 하는지를 통해 전체 프로세스를 파악한다. 사용자가 Clothes\_Hanger라는 이 웹서비스를 이용하여 목적을 성취하기 위해서는 어떤 User Interface의 흐름을 따라가게 되는지 도식화하고, 사용자가 직접적으로 접하게 되는 각 페이지 별로 어떤 기능들이 구현되어 있고, 어떻게 접근 가능한지 기술한다.
5. 회원가입

Clothes\_Hanger 시스템을 사용하기 위해서 사용자에게 회원가입 서비스를 제공한다. 회원 가입은 시스템의 목적 상 MD와 상품판매자로 나뉘어져 있으며 각각 요구하는 정보를 입력 받아 회원가입을 진행하는 시스템에 대해 세부적으로 설명한다. Class diagram, Sequence diagram, State diagram을 이용하여 기술한다.

1. 로그인

Clothes\_Hanger 시스템을 사용하기 위해 사용자에게 로그인 서비스를 제공하는 시스템에 대해 기술한다. 회원가입과 마찬가지로 MD와 상품판매자의 처리 경로를 달리 구성하고 이에 대한 세부적인 설명이 첨부된다. Class diagram, Sequence diagram, State diagram 등을 이용하여 기술한다.

1. 판매자의 파일 업로드(Multiple file upload)

판매자가 등록하고자 하는 파일을 판매자의 id를 폴더명으로 하는 폴더에 등록하고자 하는 기능을 설명한다. 해당 기능은 Class Diagram, Sequence Diagram, State Diagram으로 구조화되고, 해당 기능의 component간의 관계를 살펴볼 수 있다.

1. 딥러닝 이미지 분류  
   판매자에게서 받은 여성의류 이미지의 종류를 분류하는 기능을 설명한다. 어떤 모델과 어떤 기법을 사용했고, 데이터 수집과 학습을 어떻게 진행했는지 약술한다.
2. Protocol Design  
   여러 Subsystem이 interact하기 위한 프로토콜을 서술한다. 기본 형식은 JSON을 사용한다.
3. Database Design  
   시스템에 필요한 데이터 구조와 그 관계를 상술한다. Requirement Specification에서 기술한 데이터 관련 요구사항을 구체화한다.
4. Testing Plan

Testing plan에서는 ‘Clothes Hanger’ 시스템 기능이 전달되는 input에 대해 적절한 output을 출력하는지 확인하고, 시스템 내부 결함을 찾기 위해 진행되는 test에 대한 테스트 정책(Test Policy)과 테스트 케이스(Test Case)에 대해 기술한다.

1. Develop environment  
   개발을 위한 적절한 개발언어, 개발환경, IDE 및 버전 관리 도구를 기술한다. 시스템 개발을 위한 개발 규칙 역시 기술한다.

1.4 Version Document

A. Version Format

버전 번호는 major.minor[.maintenance]로 구성되며, 문서의 버전은 0.1부터 시작한다.

B. Version Management Policy

명세서를 수정할 때마다 버전을 업데이트한다. 다만 버전의 변경 간격이 1시간 이내일 경우 버전 번호를 업데이트하지 않고 하나의 업데이트로 간주한다. 이미 완성된 파트를 변경할 때에는 Minor Number를 변경하며, 새로운 부분을 추가하거나 문서의 구성이 달라질 경우 Major Number를 변경한다.

C. Version History

|  |  |  |
| --- | --- | --- |
| 버전 번호 | 수정 날짜 | 수정 내용 |
| 0.1 | 2019.05.14 | 문서 개요, 목차, Introduction 작성 |
| 1.0 | 2019.05.15 | Database Design, Development Environment 작성 |
| 2.0 | 2019.05.15 | Deep Learning image classification, Preface 작성 |
| 3.0 | 2019.05.16 | 판매자의 파일 업로드 기능 작성 |
| 4.0 | 2019.05.17 | 회원가입, 로그인 기능 작성 및 목차 수정 |
| 5.0 | 2019.05.17 | User Interface 항목 작성 |
| 6.0 | 2019.05.18 | System Architecture 작성 |
| 7.0 | 2019.05.19 | Testing Plan 작성 |

**2. Introduction**

2.1 Objectives

- 이 장에서는 본 문서에서 시스템의 설계할 때 사용되는 종류의 다이어그램과 UML(Unified Modeling Language) 툴에 대해 서술한다.

2.2 Applied Diagram / Tool

A. Draw.io

- 본 문서가 포함하는 Class diagram, State diagram, Sequence diagram 그리고 ER diagram까지 대부분은 Draw.io 온라인 UML툴을 이용하여 제작되었다. UML(Unified Modeling Language)을 지원하는 소프트웨어 모델링 플랫폼이며 웹 브라우저 환경에서 작업할 수 있어 편의성이 높다.

B. UML

- Unified Modeling Language(이하 UML)는 소프트웨어 공학에서 사용되는 표준화된 범용 모델링 언어이다. UML은 소프트웨어 집약 시스템의 시각적 모델을 만들기 위한 도안 표기법을 포함한다. 이는 프로그램 개발이라는 행위에 대해 전문가와 비전문가가 서로 대화할 수 있는 도구로서 기능한다. UML은 객체 지향 소프트웨어 집약 시스템을 개발할 때 산출물을 명세화, 시각화, 문서화할 때 사용한다. 일반적인 프로그램 언어와는 달리 도형으로 표현이 가능하기 때문에 프로그램에 대한 지식이 없는 사람도 프로그램이 제공해야할 기능과 구조를 표현하고 이해하는 것이 가능하다.

C. Class Diagram

- UML (Unified Modeling Language)의 Class diagram은 시스템의 클래스, 속성, 작업 (또는 메소드) 및 객체 간의 관계를 보여줌으로써 시스템의 구조를 설명하는 정적 구조 다이어그램 유형이다. 클래스 다이어그램의 클래스는 주요 요소, 응용 프로그램의 상호 작용 및 프로그래밍 할 클래스를 모두 나타낸다. 클래스 다이어그램의 경우 시스템 내부에 존재하는 클래스들을 선별하여 나타내고 각 클래스들의 속성(Attribute)과 행위(Behavior)를 기입한다. 여기서 클래스들 사이에 여러 가지 관계를 가질 수 있다. 이외에 상속관계(Generalization), 의존관계(Dependency)가 나타날 수 있으며 클래스 다이어그램을 그리고자 할 때 항상 추상화 단계를 고려하여서 그린다. 이를 통해 클래스 간 인터페이스를 빠른 단계에서 명확하게 할 수 있다.

D. Sequence Diagram

- Sequence Diagram은 시간 순서대로 정렬된 객체 상호 작용을 보여준다. 시나리오에 포함 된 객체 및 클래스와 시나리오 기능을 수행하는 데 필요한 객체 간에 교환되는 메시지 시퀀스를 묘사한다. Sequence diagram은 시스템의 동적인 면을 나타내는 대표적인 다이어그램이다. 시스템이 실행 시 생성되고 소멸되는 객체를 표기하고 객체들 사이에 주고받는 메시지를 나타내게 된다. Sequence diagram만의 특징으로는 메시지와 시간의 흐름을 나타내어 메시지의 순서에 중점을 두고 있다는 것이다.

E. State Diagram

- State diagram은 한 객체의 상태 변화를 다이어그램으로 나타낸 것이다. 시스템의 실행 시 객체의 상태는 메시지를 주고받음으로써 또한 어떠한 이벤트를 받음으로써 많은 변화가있을수있다. 실제시스템에서실행시많은객체가생성되고소멸된다.결국상태 다이어그램은 특별히 관심을 가져야 할 객체에 관하여 그리고 특정 조건에 만족하는 기간 동안의 상태를 표시해야 한다.

F. Use case Diagram

- Use case Diagram은 사용자와 사용자 간의 상호 작용을 표현한 것으로 가장 간단한 형식을 취하는 다이어그램 중 하나이며, 사용자와 관련된 다양한 유스 케이스와의 관계를 보여준다. Use case Diagram은 시스템의 여러 유형의 사용자와 다양한 유스 케이스를 식별할 수 있으며 종종 다른 유형의 다이어그램도 수반된다. 유스케이스란 말 그대로 컴퓨터 시스템과 사용자가 상호작용을 하는 하나의 경우이다. 이러한 Use case Diagram은 사용자와의 대화수단으로 그리고 앞으로 구축해 나갈 때의 밑바탕이 된다.

G. Entity-Relationship(ER) Diagram

- Entity-Relationship(ER) Diagram은 데이터베이스에서 각 개체들 간의 관계를 표현하기 위한 다이어그램으로 원칙적으로 UML을 구성하는 다이어그램은 아니다. 데이터베이스라 불리는 데이터들의 저장 공간에 데이터의 구조와 제약조건들을 표현한 것이며 개체(Entity)는 현실 세계의 객체이며 관계(Relation)은 두 개 이상의 객체 사이의 연관성 및 상관관계를 설명한 것이다.

**3. System Architecture**

3.1 Objectives

System Architecture 에서는 개발하고자 하는 시스템에 대한 총괄적인 설명을 기술한다. 시스템의 기본적인 골격을 Block diagram 으로 나타내며 세부적인 Sub-System의 관계와 실제 사용되는 구조는 Deployment diagram을 통해 설명한다. 세부적인 modular들에 대한 자세한 기술은 이후의 4~8 장에서 다루도록 한다.

3.2 System Organization

Clothes\_Hanger 시스템은 크게 3가지의 하위 Sub-System으로 구성되어 있으며 전체적인 User Interface를 통해 각 시스템이 상호 작용하게 되는 구조이다. Sub-System으로는 회원가입과 로그인을 담당하는 User Management System, 판매자의 입장에서 파일을 업로드할 수 있는 Upload Management System, 업로드된 파일을 통해서 자동 분류하는 Deep Learning Classification 으로 구성되어 있다.

30

1. User Management System

User Management System의 경우, 사용자가 시스템을 이용하기 위해서 계정을 등록하고 로그인 하는 절차 까지를 포함한다. MD와 판매자의 회원가입 및 로그인 기능이 서로 나누어져 있다.

70

71

1. Upload Management System

Upload Management System에서는 판매자가 업로드할 파일을 저장하는 역할을 담당한다. 이미지 형식의 파일과 엑셀 형식의 파일 업로드 두 부분으로 구성되어 있다.

72

1. Deep Learning Classification



3.3 Deployment Diagram

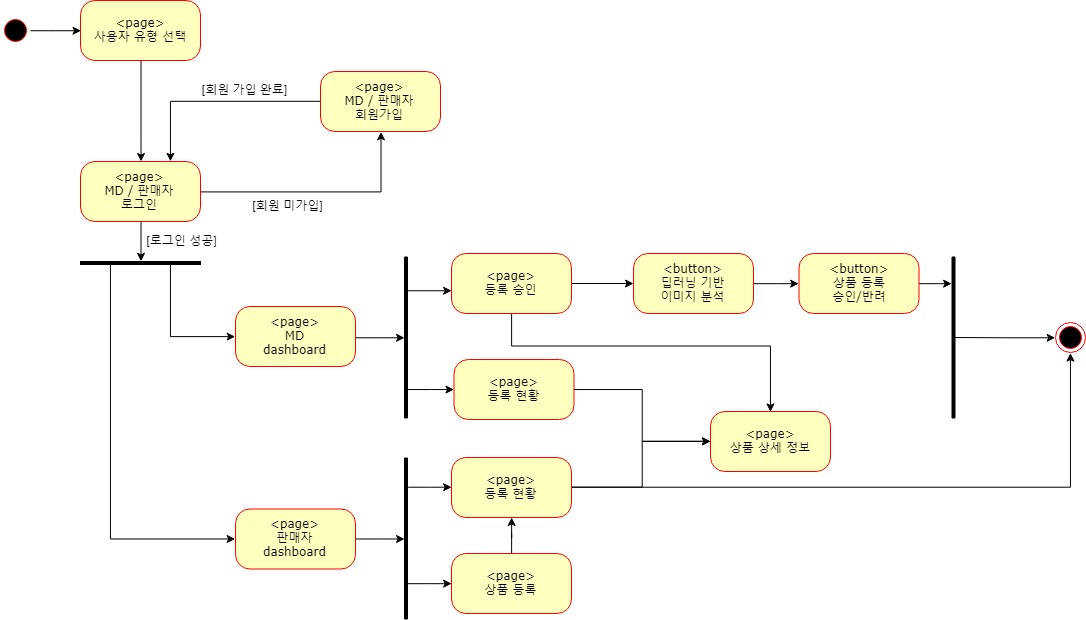
Untitled Diagram

**4. User Perspective Process**

4.1 Objectives

개발자가 개발하는 각 기능들은 사용자가 별다른 어려움 없이 접근할 수 있어야 한다. 사용자 입장에서 각 기능들이 어떤 page에서 어떻게 작동하고 편리하게 사용될 수 있는지 user interface를 디자인하는 것이 필요하다. 따라서 이 장에서는 사용자가 자신의 목적을 성취하기 위해 거쳐야 하는 일련의 process를 나타내고, 사용자 유형별로 각 페이지에서 어떤 기능들이 어떻게 제공되는지 살펴보고자 한다.

4.2 User Perspective Process Flowchart



<User Perspective Process Flowchart>

4.3 User Interface

4.3.1. 공통(MD, 판매자)

A. 사용자 유형 선택

모든 사용자는 자신이 이용하고자 하는 서비스에 따라 사용자 유형을 선택하여야 한다. 사용자 유형에는 [MD, 판매자]가 있으며, 어떤 버튼을 선택하는지에 따라 MD 로그인 페이지 또는 판매자 로그인 페이지로 넘어가게 된다.

B. 상품 상세정보 페이지

상품 상세정보 페이지는 해당 물품을 클릭하였을 때 확인할 수 있는 창으로, 등록 날짜, 등록 상품 명, 수량, 제조자, 제조국가, 소재, 색상, 치수, 제조연월, 품질보증기간이 공통적인 항목으로 표시되며 각 상품의 품목별로 상품의 특징을 확인할 수 있다. 상의의 경우 추가적으로 사이즈, 어깨 넓이, 가슴 둘레, 소매길이, 소매 끝 단면, 겨드랑이 단면, 총장을 표시하고, 하의의 경우 허리 단면, 허벅지 단면, 엉덩이 단면, 밑위, 밑단면, 총장을 추가적으로 표시한다. 신발의 경우 사이즈(cm), 발볼, 깔창, 굽 높이, 앞굽, 총길이를 추가적으로 표시한다.

위 정보는 테이블의 형태로 표시된다.

C. 모든 페이지(회원가입, 로그인 페이지 제외)

로그인 이후 모든 페이지에서 상단의 메뉴 바와 좌측의 사이드 바는 고정되어 있는 상태이다. MD와 판매자에게 메뉴 바는 동일하게 표시된다. 메뉴 바 좌측의 CLOTHESHANGER 로고 배너를 클릭하면 메인 페이지인 dashboard 페이지로 이동하게 된다. 우측 상단에는 알림 아이콘이 있어 어떤 알림이 언제 왔는지 확인할 수 있다.

좌측의 사이드 바는 MD과 판매자에 따라 나타나는 탭이 다르지만 가장 하단에는 공통적으로 로그아웃 탭이 있다. MD의 경우 dashboard, 등록 승인 탭, 등록 현황 탭, 로그아웃 탭이 있고, 판매자의 경우 dashboard, 상품 등록 탭, 등록 현황 탭, 로그아웃 탭이 있다. 각 탭을 클릭하면 해당 페이지로 연결된다.

4.3.2. MD

A. Registration

MD가 이 웹서비스를 이용하기 위해서 가장 먼저 수행해야 하는 과정이 회원가입이다. 회원가입이 이미 되어있는 경우, 굳이 다시 회원가입을 해야 할 필요는 없다. 로그인 창에서 ‘회원가입’ 링크를 클릭하여 들어오게 된다.

회원가입 페이지는 회원가입 시 필요한 정보 양식을 제출하는 형태이다. MD는 이름, 사번, 비밀번호를 입력하는데, 각 필드에 대한 제한사항이 함께 기술되어 있어 그에 맞춰서 값을 기술해주면 된다.

입력한 사번이 회사 데이터베이스에 없는 경우 ‘MD 권한없음’ 메시지가 뜨게 되고 회원가입이 완료되지 않는다.

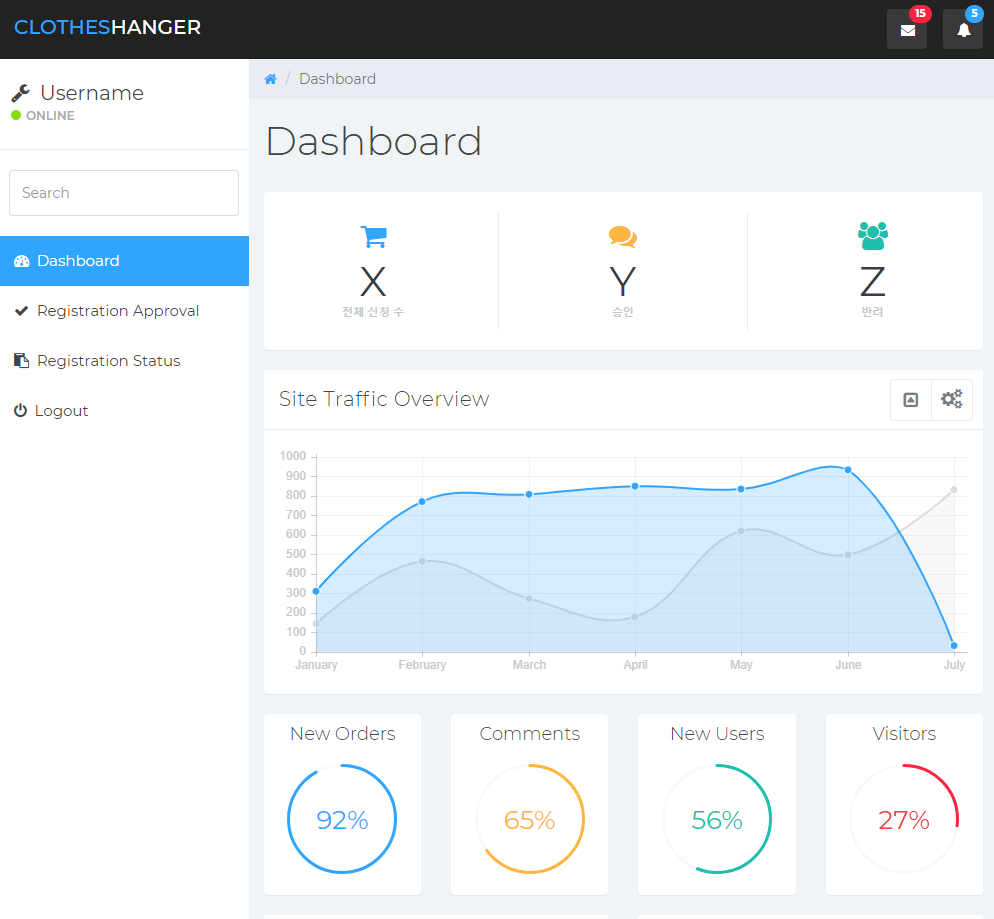
입력이 완료되면 하단의 ‘제출’ 버튼을 클릭한다. 제출에 문제가 없으면 회원가입이 완료되고 MD 로그인 페이지로 넘어간다.

B. Log in

MD 로그인 페이지에서 MD는 자신의 사번과 비밀번호를 입력한다. 사번과 비밀번호가 일치하면 MD dashboard로 넘어간다. 일치하지 않으면 ‘해당 사번이 존재하지 않거나 비밀번호가 일치하지 않음’ 메시지를 출력하며 필드가 초기화된다.

C. Dashboard

MD 는 로그인을 성공적으로 완료하면 MD dashboard 페이지로 접속하게 된다. 페이지의 중앙에는 전체 신청 수, 승인 수, 반려 수 등의 통계를 확인할 수 있다.



<예상 MD dashboard 초안>

D. 등록 승인 페이지

판매자가 신청한 상품들을 최신순으로 확인할 수 있다. 리스트 테이블의 형태로 나열되게 되며 각 상품들은 한 줄에 하나씩 기입되게 된다. 아무 작업도 거치지 않은 상태에서는 각 상품 레코드 옆에 ‘대기’ 알림이 출력되어 있다. 그 우측에는 ‘승인’과 ‘반려’ 버튼이 나란히 있지만 회색으로 비활성화 되어있는 상태이다. 사용자는 스크롤을 통해 테이블의 후반부를 열람할 수 있다.

메뉴 바 아래, 페이지의 우측 상단에는 ‘일괄 처리’ 버튼이 있어 이를 클릭하면 여성 의류 이미지 딥러닝 분류 함수가 작동하게 된다. 이에 따라 판매자가 올린 상품의 이미지와 등록한 카테고리가 일치하는지 여부를 자동적으로 확인할 수 있다. ‘대기’로 표시되어 있던 알림은 ‘일치’ 또는 ‘불일치’ 알림으로 전환되며 옆에 있던 ‘승인’과 ‘반려’ 버튼은 각각 초록색과 빨간색으로 바뀌어 활성화된다. 또한 각 상품명을 클릭하면 상품 상세정보를 확인할 수 있는데 이는 새로운 탭에서 열리게 된다. MD는 이를 참고하여 알맞게 승인/반려 여부를 선택하면 된다.

E. 등록 현황 페이지

MD 가 좌측 사이드 바에서 등록 현황 페이지를 클릭하였을 때 접속된다. 페이지 중앙에 해당 MD가 관여하는 판매자들이 등록한 상품들의 리스트와, 그 상품들의 오른쪽에 등록 결과(승인, 반려, 대기)가 표시된다. 각 상품들을 클릭하면, 해당 상품들의 상품 상세정보 페이지가 새로운 탭에서 열린다.

4.3.3. 판매자

A. Registration

MD가 이 웹서비스를 이용하기 위해서 가장 먼저 수행해야 하는 과정이 회원가입이다. 회원가입이 이미 되어있는 경우, 굳이 다시 회원가입을 해야 할 필요는 없다. 로그인 창에서 ‘회원가입’ 링크를 클릭하여 들어오게 된다.

회원가입 페이지는 회원가입 시 필요한 정보 양식을 제출하는 형태이다. 판매자는 ID, 비밀번호, 성별, 휴대전화 번호, 사업장 주소, 사업자 이메일 주소를 입력하게 된다. 각 필드에 대한 제한사항이 함께 기술되어 있어 그에 맞춰서 값을 기술해주면 된다. 입력한 ID가 중복될 경우, ‘이미 등록된 ID입니다’ 메시지가 뜨게 되고, 새로운 ID를 입력해야 한다.

사업장 주소의 경우 ‘주소 찾기’ 버튼을 클릭하여 도로명 주소를 입력하고 해당 주소를 찾아 클릭한 후 세부 주소를 추가적으로 입력하면 된다.

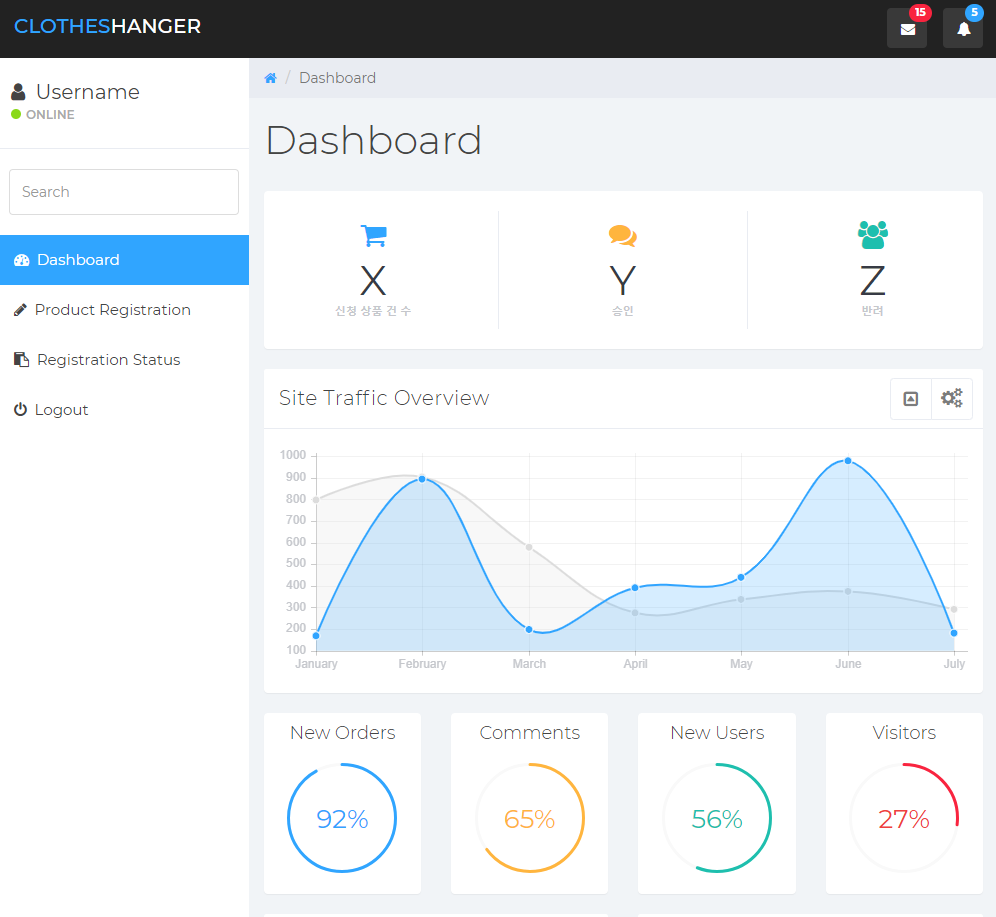
입력이 완료되면 하단의 ‘제출’ 버튼을 클릭한다. 제출에 문제가 없으면 회원가입이 완료되고 판매자 로그인 페이지로 넘어간다.

B. Log in

판매자 로그인 페이지에서 판매자는 자신의 ID와 비밀번호를 입력한다. ID와 비밀번호가 일치하면 판매자 dashboard로 넘어간다. 일치하지 않으면 ‘해당 ID가 존재하지 않거나 비밀번호가 일치하지 않음’ 메시지를 출력하며 필드가 초기화된다.

C. Dashboard

판매자는 로그인을 성공적으로 완료하면 판매자 dashboard 페이지로 접속하게 된다. 페이지의 중앙에는 본인이 신청한 상품 건수, 승인 수, 반려 수 등의 통계를 확인할 수 있다.



<예상 판매자dashboard 초안>

D. 상품 등록 페이지

판매자가 좌측 사이드바에서 상품 등록 페이지를 클릭하였을 때 접속되며, 페이지 중앙의 윗부분에 상품의 세부정보가 담긴 Excel 파일의 등록 안내사항이 있고, 등록 안내사항 바로 아래 파일을 업로드할 수 있도록 하는 ‘파일 추가’ 버튼, 등록된 파일을 제거하는 ‘파일 삭제’ 버튼, 파일을 최종적으로 제출하는 ‘등록’ 버튼이 있다. ‘파일 추가’ 버튼을 클릭하여 탐색기 창을 활용해 엑셀 파일을 추가하고 ‘등록’ 버튼을 통해 웹사이트에 등록하게 되면, 웹페이지에서 해당 Excel 파일의 각 행을 읽어오는 동시에 페이지가 동적으로 전환되며, 페이지 중앙의 윗부분은 이미지 업로드를 위한 안내사항이 출력된다. 그 아래에는 Excel 파일에서 읽어온 각 상품을 표시하는 행들이 나타나게 되며, 최하단에 ‘이미지 등록’ 버튼이 있어, 해당 상품들의 이미지를 한꺼번에 업로드 할 수 있다.

E. 등록 현황 페이지

판매자가 좌측 사이드 바에서 등록 현황 페이지를 클릭하였을 때 접속된다. 페이지 중앙에 본인이 등록한 상품들의 리스트와, 그 상품들의 오른쪽에 등록 결과(승인, 반려, 대기)가 표시된다. 각 상품명을 클릭하면, 해당 상품들의 상품 상세정보 페이지가 새로운 탭에서 열린다.

**5. 회원가입(Sign up)**

5.1 Objectives

- 시스템을 사용하기 위해서 가장 먼저 선행되어야 하는 기본적인 기능으로, 회원가입 이후에 로그인 절차를 밟음으로 시스템의 기능들을 이용할 수 있다. 해당 장에서는 로그인과 관련된 항목들을 설명한다.

5.2 Class diagram

C:\Users\user\Desktop\1.png

5.2.1 MD 회원가입 정보

A. id: MD가 가지고 있는 고유 회사의 사원 번호 6자리

B. pw: 가입자의 비밀번호

C. name: 가입자의 이름

5.2.2 판매자 회원가입 정보

A. id: 가입자의 아이디

B. pw: 가입자의 비밀번호

C. name: 가입자의 이름

D. sex: 가입자의 성별

E. phone\_num: 가입자의 휴대전화 번호

F. location: 가입자의 사업장 주소

G. email: 가입자의 이메일 주소

5.2.3 company DB handler

A. idConfirm(int): 실제 회사의 DB에 저장된 사번과 입력된 id를 비교해서 존재하는 여부를 확인. 저장되어 있지 않은 경우에는 회원가입 진행 불가.

5.2.4 MD user DB handler / 판매자 user DB handler

A. idChecker(int): 아이디의 중복성 검사

B.pwChecker(String): 비밀번호 유효성 검사

5.3 Sequence diagram

5.3.1 MD 회원가입

2

5.3.2 판매자 회원가입

3

5.4 features

- 판매자의 회원가입 정보가 잘못 입력되거나, 허위 사실을 기입하는 경우에는 서비스 이용 시 불이익의 책임이 서비스 이용자에게 있음을 명시한다. 또한 이를 방지하기 위해 정확한 정보를 입력해야 한다는 것을 회원가입 양식에 추가한다.

**6. 로그인 (Login)**

6.1 Objectives

회원가입을 통해 계정을 생성하였다면, 로그인을 통해서 서비스를 본격적으로 사용할 수 있다. 로그인은 MD와 판매자의 경우 모두가 시도 및 과정이 거의 유사하다고 볼 수 있다.

6.2 Class Diagram

6

6.2.1 로그인

A. id: 가입자의 아이디 (단, MD의 경우 id는 int변수값을 취한다.)

B. pw: 가입자의 비밀번호

6.2.2 DB handler

A. Check: 아이디와 비밀번호 일치 여부를 판독.

6.3 Sequence diagram

7

6.4 features

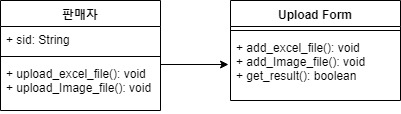
- 로그인 이후에 MD의 경우에는 MD http 페이지가, 판매자의 경우에는 판매자 http 페이지로 연결되는 차이점이 존재한다.

**7. 판매자의 파일 업로드(Multiple file upload)**

7.1 Objectives

판매자가 등록하고자 하는 파일을 판매자의 id를 폴더명으로 하는 폴더에 등록하고자 하는 기능을 담당한다. 해당 폴더에는 상품의 정보를 양식에 맞게 기입한 하나의 엑셀파일과 상품의 여러 이미지 파일이 등록된다. 해당 기능은 Class Diagram, Sequence Diagram, State Diagram으로 구조화되고, 해당 기능의 component간의 관계를 살펴볼 수 있다.

7.2 Class Diagram



7.2.1. 판매자

A. sid: 판매자 id

B. upload\_excel\_file(): 엑셀 파일 등록

C. upload\_Image\_file(): 이미지 파일 등록

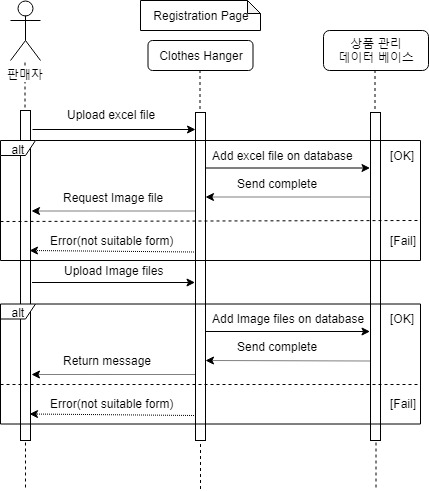
7.2.2. Clothes Hanger

A. add\_excel\_file(): 상품정보파일을 판매자 id를 폴더명으로 하는 폴더 아래에 등록

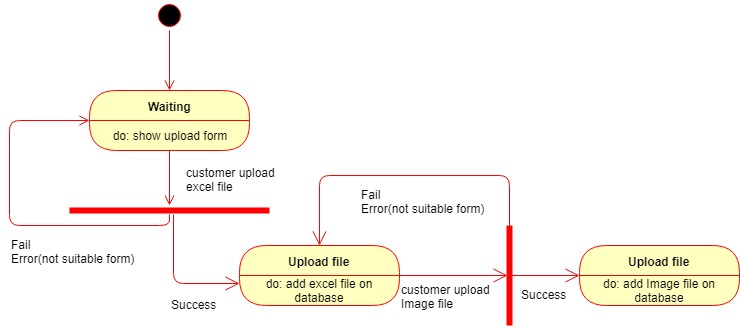
B. add\_Image\_file(): 상품이미지파일을 판매자 id를 폴더명으로 하는 폴더 아래에 등록

C. get\_result(): Boolean. 상품정보파일과 상품이미지파일 등록의 성공여부를 출력

7.3 Sequence Diagram



7.4 State Diagram



7.5 Features

- 판매자는 지정된 양식과 일치하는 상품 정보파일을 올려야 한다. 그렇지 않은 경우 상품 처리가 반려될 수 있음을 공지해야 한다.

- 해당 기능은 엑셀양식 등록 후 이미지 파일을 등록하기 위해서는 엑셀 양식에 등록된 상품명과 일치하는 이미지 파일명을 명시해야 함을 공지해야 한다.

**8. Deep Learning Image Classification**

8.1 Objective

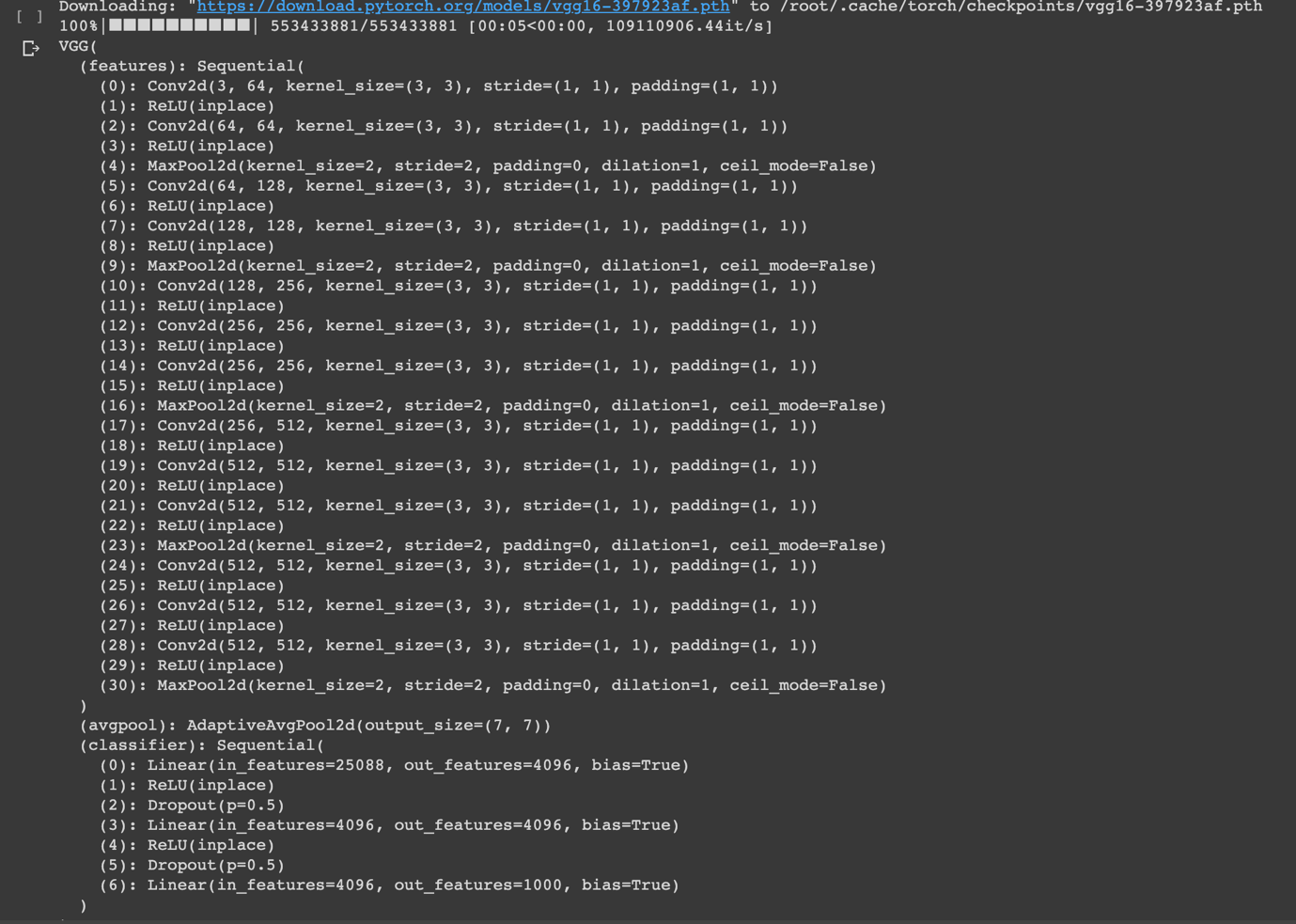
‘이미지 분류 자동화’의 핵심 기능인 딥러닝 이미지 분류 시스템의 개요와 구성, 사용 방법, 특징 등을 서술한다.

8.2 Image Model Overview

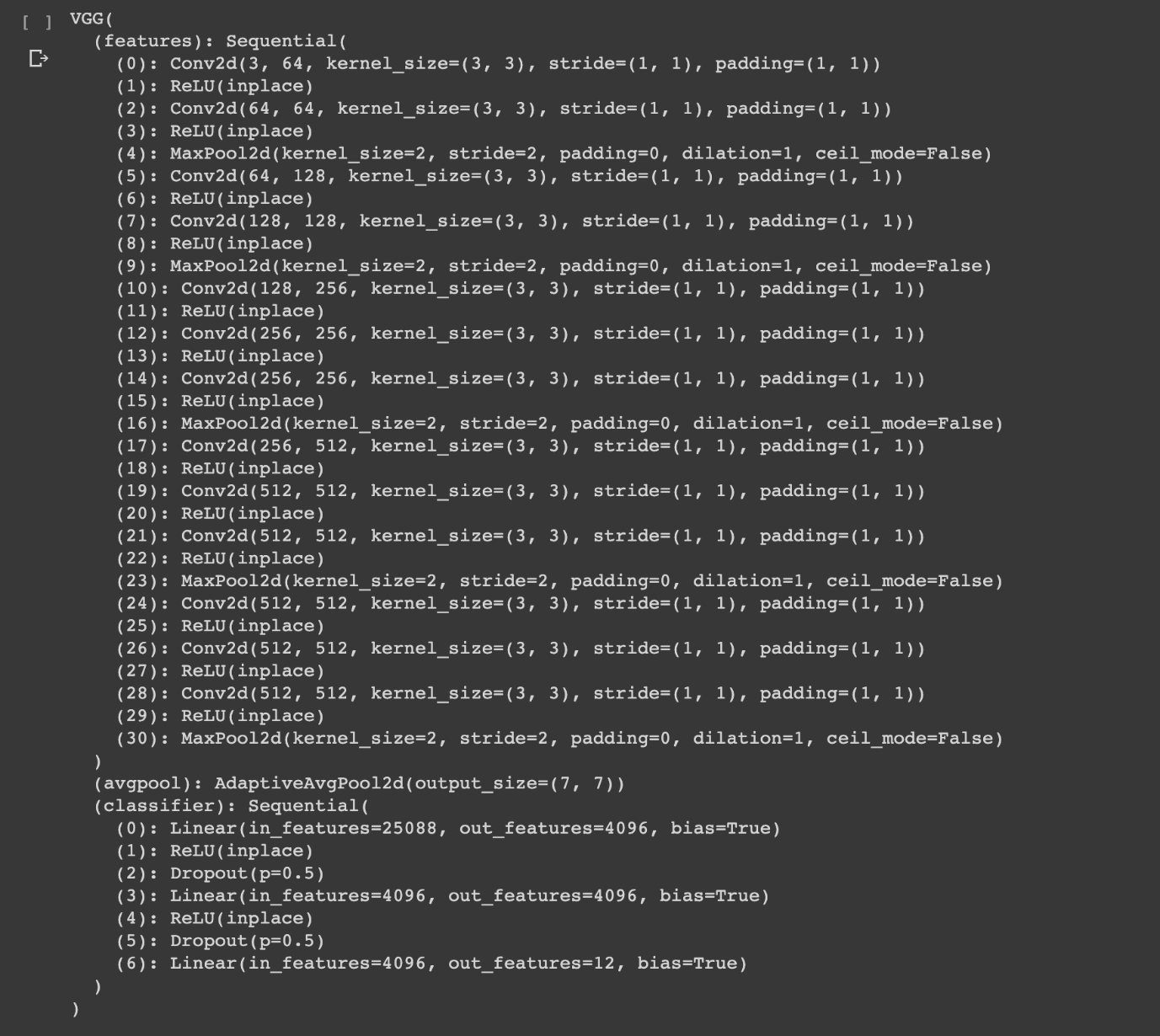
Transfer Learning 기법을 사용하여, Pytorch에서 Pretrained 형태로 제공되는 Image Classification Model 중 vgg16을 사용했다. 연산량이나 정확도에서 최상급 성능을 자랑하는 모델은 아니지만, 가장 모델 구조가 단순하기에 Error Checking이 쉽다는 장점 때문에 vgg16 모델을 채택했다.

Input layer 1개, Convolutional Layer가 12개, Fully Connected layer 3개로 총 16개 Layer로 구성되어 있으며, Image를 input으로 받은 다음 Convolutional Layer를 거치며 이미지의 edge를 파악하는 weight값을 구한다. Convolutional Layer를 모두 통과한 뒤 남은 Feature map을 Vectorize한 후 Fully Connected Layer를 거쳐 최종적인 분류 대상 Class를 예측하는 것이 모델의 구조다.

Pytorch에서 Pretrained 모델로 제공하는 vgg16의 모델 구조는 아래와 같다.



이미지 분류를 시행하려면, classifier의 가장 마지막 Linear의 output 개수를 1000개가 아니라 분류 클래스 개수로 정해야 한다. Requirement Specification 과정에서 이미지 분류 클래스는 12개로 제한했기 때문에, 12개로 변환한 결과는 아래와 같다.



Linear 마지막 out\_feature를 12개로 변환한 다음, 이미지 학습에 필요한 데이터를 수집한다.

8.3 How To Collect Data

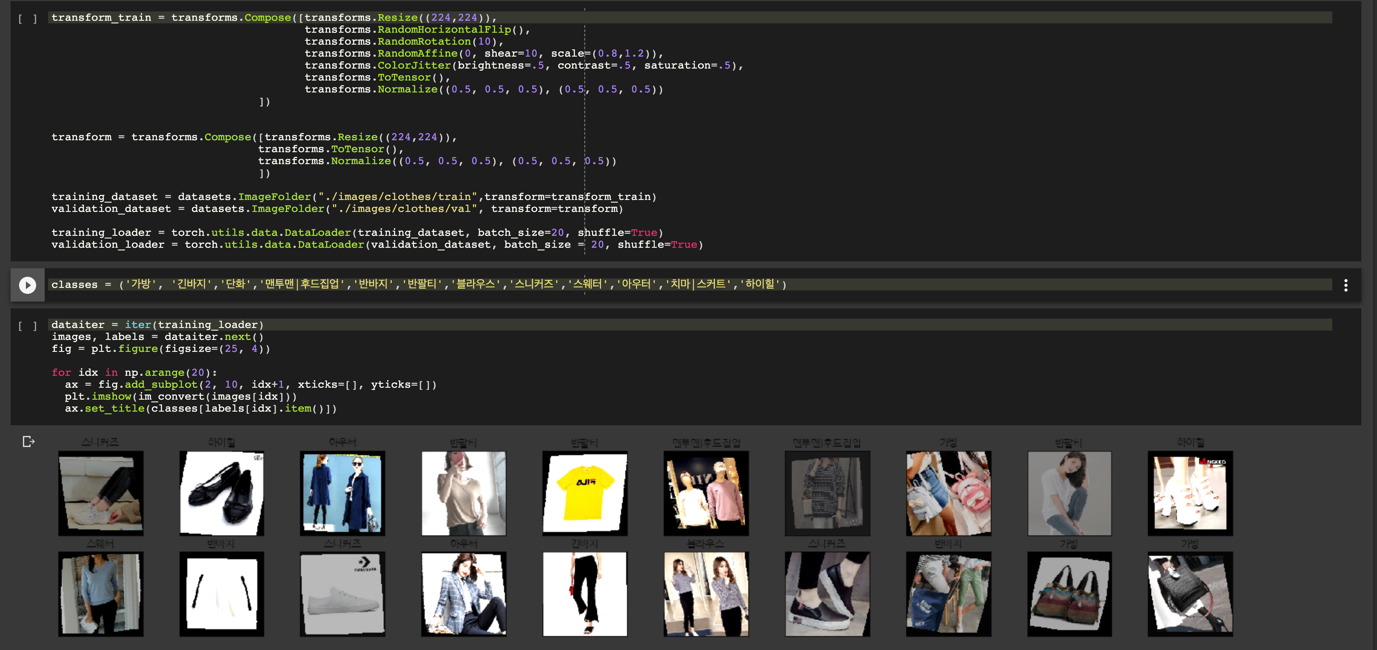
12개 클래스별로 이미 분류된 이미지를 수집해야 하는데, 본 팀은 쿠팡에서 제공하는 이미지 데이터를 웹 크롤링으로 추출하기로 했다.

1. Web scraping

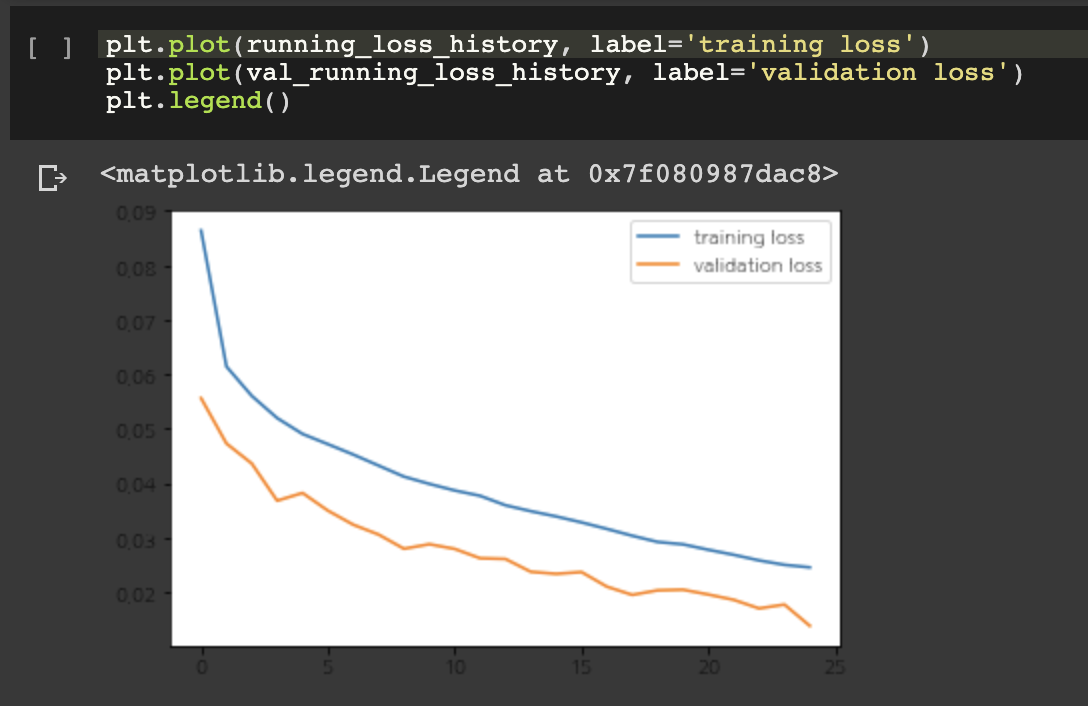
쿠팡 웹페이지의 카테고리별 이미지 데이터를 불러와 저장하는 함수다. 하나의 카테고리당 쿠팡에서 약 1000여 개의 이미지 데이터를 불러올 수 있었으며, 12개 카테고리당 1200여 개의 이미지이므로 총 12000여 개의 데이터를 확보할 수 있었다.

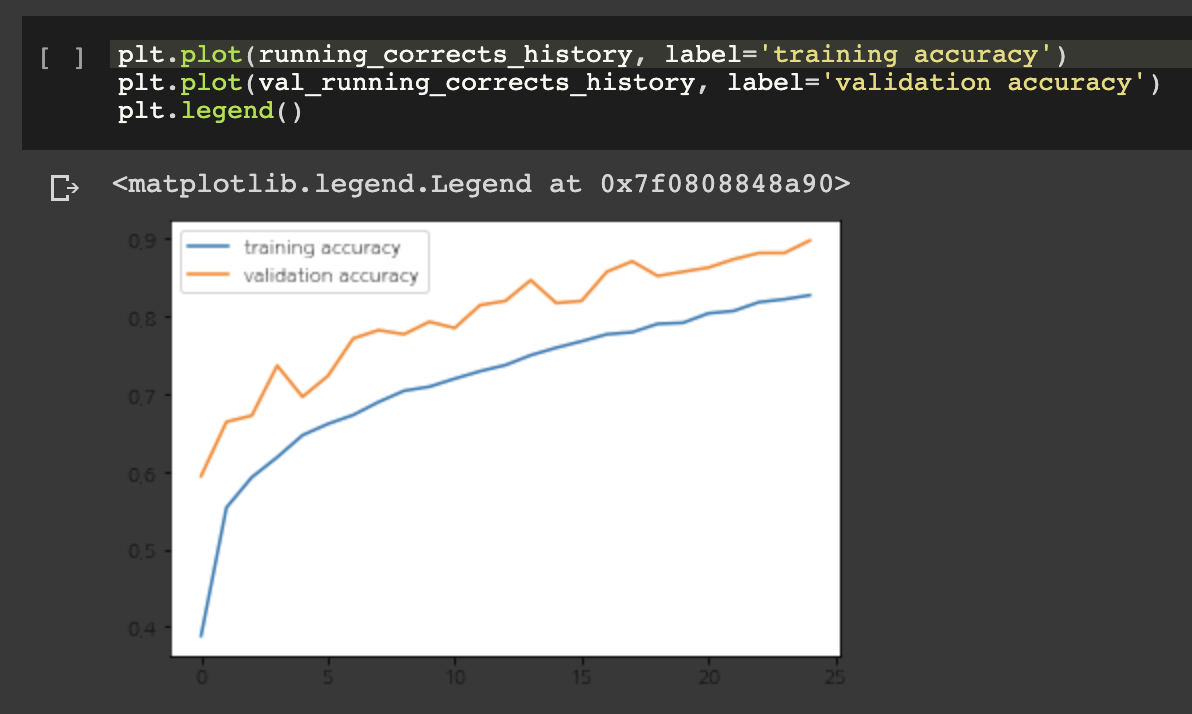
8.4 Training and Validation

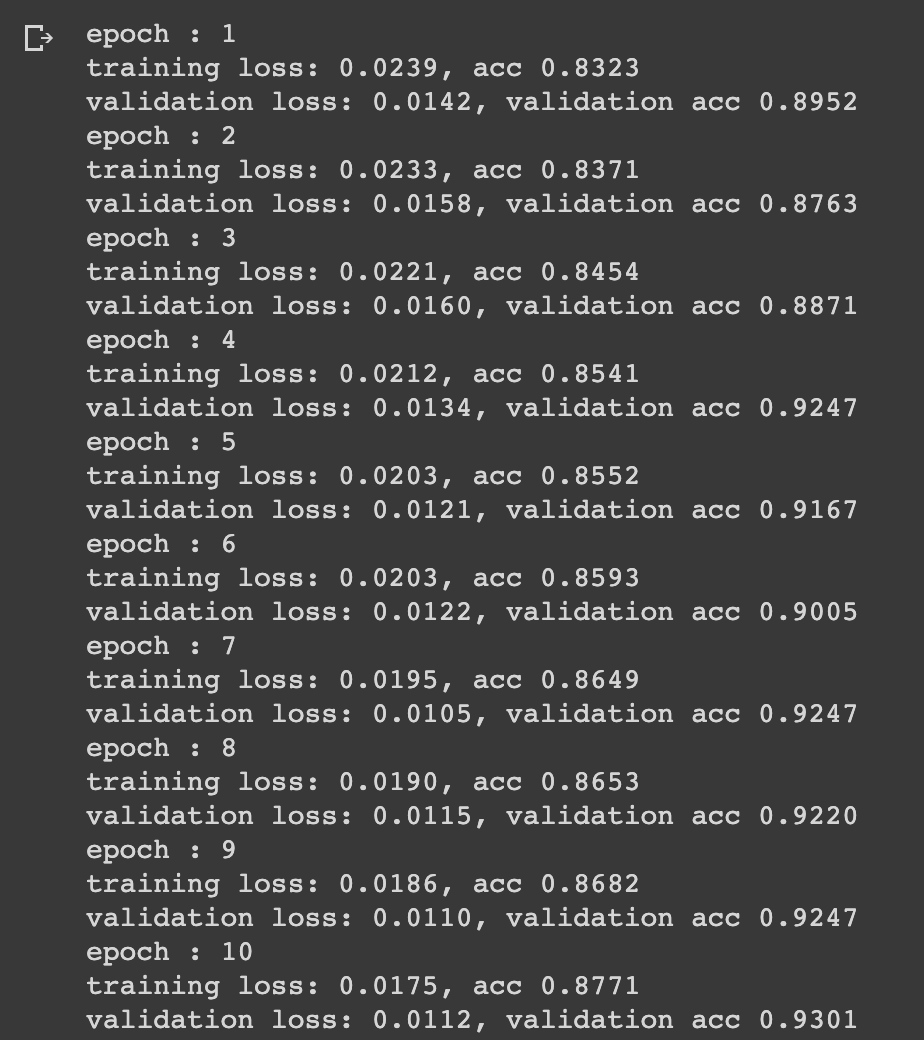
카테고리별로 확보한 1000여 개의 이미지에서 validation을 위한 데이터셋을 따로 분류했다. 카테고리당 50개의 이미지를 validation 데이터로 따로 분리한 후, Pytorch에서 제공하는 Data Augmentation 기법을 활용해 Training dataset의 양을 확보했다. 명도 / 채도 변화, 이미지 기울기 변화, 이미지의 edge 부분 잘라내기 등 다양한 기법과 normalization을 적용했다.



Batch size는 20, Epoch는 25회 시행 후 추가로 10회 더 실시했다. 25회 시행했을 때 training loss와 validation loss는 아래와 같다. Training loss가 더 큰 이유는 Data Augmentation으로 의도적으로 training 데이터를 뒤틀었기 때문이다. Validation data는 Augmentation 없이, 실제로 쓰는 이미지 데이터를 활용했기 때문에 loss가 낮다.



마찬가지 이유로 Accuracy도 Training set에서보다 Validation set이 더 높게 측정된다.

25회 + 10회 Epoch 시행 후, Validation set의 Accuracy는 약 93%였다. Epoch를 더 늘릴 경우 정확도가 올라갈 수는 있으나 overfit문제가 심해질 수 있음을 우려해 더 이상 학습을 진행하지는 않았다.

**9. Protocol Design**

9.1 Objective

Protocol design에서는 하위 시스템 간 통신할 때의 프로토콜을 설명한다. 프로토콜의 형식은 HTTP, JSON을 사용한다.

9.2 JSON

JSON은 자바스크립트 기반의, 데이터 객체를 전송하기 위해 텍스트를 사용하는 데이터 형식이다. 언어나 플랫폼 독립적인 데이터 형식이기에 여러 환경에서 쉽게 이용할 수 있다. 다른 데이터 객체 전송 형식인 xml이 header, tag등 여러 요소들 때문에 가독성이 떨어지고 용량이 큰 것과 달리, 간결하고 통일된 양식을 사용할 수 있어 사용성이 높다. Python의 경우 json이라는 내장 라이브러리를 활용해 json형식과 Python Dictionary 형식을 자유롭게 변환하여 사용할 수 있다.

9.3 HTTP

HTTP란 HyperText Transfer Protocol의 약자로, 하이퍼텍스트를 빠르게 교환하기 위해 클라이언트와 서버 간 메세지를 전송할 때 쓰이는 규칙을 말한다. 최근에는 HTTP의 보안 문제 때문에 HTTPS라는, 보안이 강화된 형태가 많이 쓰인다.

9.4 Protocol Description

A. 회원가입

A-1. MD

1. Request

|  |  |  |
| --- | --- | --- |
| 요청변수 | 값 | 설명 |
| ID | Int | MD의 사원번호 |
| PW | String | User의 PW |
| Name | String | User의 이름 |

1. Response

|  |  |  |
| --- | --- | --- |
| 출력변수 | 값 | 설명 |
| HTTP Response |  | 회원가입 성공 여부 페이지 |

A-2. 판매자

1. Request

|  |  |  |
| --- | --- | --- |
| 요청변수 | 값 | 설명 |
| ID | String | User의 ID |
| PW | String | User의 PW |
| Name | String | User의 이름 |
| Sex | String | User의 성별 |
| Phone\_num | Int | User의 휴대전화 번호 |
| Location | String | User의 사업장 주소 |
| Email | String | User의 이메일 주소 |

1. Response

|  |  |  |
| --- | --- | --- |
| 출력변수 | 값 | 설명 |
| HTTP Response |  | 회원가입 성공 여부 페이지 |

B. 로그인

a. Request

|  |  |  |
| --- | --- | --- |
| 요청변수 | 값 | 설명 |
| ID | String  (단, MD의 경우에은 int 값을 변수로 받는다.) | User의 ID |
| PW | String | User의 PW |

1. Response

|  |  |  |
| --- | --- | --- |
| 출력변수 | 값 | 설명 |
| Http Response |  | MD의 경우 MD 전용의 Http 페이지를 띄우고, 판매자의 경우에도 역시 판매자가 이용 가능한 Http 페이지를 출력한다. |

C. 판매자의 파일 업로드(Multiple file upload)

a. Request

|  |  |  |
| --- | --- | --- |
| 요청변수 | 값 | 설명 |
| 상품정보파일 | Excel file | 판매자가 업로드할 정해진 양식에 맞춘 상품 정보 파일 |
| 상품이미지파일 | Image file(확장자는 jpg) | 판매자가 업로드할 다수의 상품 이미지 파일 |

1. Response

|  |  |  |
| --- | --- | --- |
| 출력변수 | 값 | 설명 |
| Http Response |  | 파일 양식이 맞는 경우 성공적으로 업로드가 되었다고 안내한다.  파일 양식이 맞지 않은 경우 파일 양식을 확인하라고 안내한다. |

D. 딥러닝 이미지 분류

1. Request

|  |  |  |
| --- | --- | --- |
| 요청변수 | 값 | 설명 |
| 상품정보파일 | Excel file | 판매자가 업로드할 정해진 양식에 맞춘 상품 정보 파일 |
| 상품이미지파일 | Image file(확장자는 jpg) | 판매자가 업로드할 다수의 상품 이미지 파일 |

1. Response

|  |  |  |
| --- | --- | --- |
| 출력변수 | 값 | 설명 |
| Http Response |  | 파일 양식이 맞는 경우 분석을 진행하고, 진행 결과로 안내할 html로 redirect한다.  파일 양식이 맞지 않은 경우 파일 양식을 확인하라고 안내한다. |
| result | json | 등록한 이미지 순서대로 분류 작업을 실행한 후 나온 결과를 json 형식으로 전송한다. |

**10. Database Design**

10.1 MySQL

MySQL은 관계형 데이터베이스 관리 시스템(RDBMS)으로, 구조화된 데이터의 집합을 의미한다. 데이터를 하나의 저장 공간에 저장하지 않고 서로 별개의 테이블에 나누어 분산 저장하는 시스템이다. MySQL 소프트웨어는 오픈 소스이며, MySQL 데이터베이스는 GPL을 준수하고 모든 사용자에게 무료로 배포되고 있다. MySQL 서버는 클라이언트/서버 또는 임베디드 시스템에서 사용 가능하며 다중-쓰레드 SQL 서버로 구성되어 있다.

일반적으로 MySQL에 저장된 데이터에 접근하거나 새로운 데이터를 입력하고 수정하고 저장하기 위해서는 별도의 관리프로그램이 필요하지만, 개발에 사용하는Django 웹 프레임워크에서는 MySQL 서버와 연동해 편리하게 사용 가능하다.

**11. Testing plan**

11.1 Objectives

Testing plan에서는 ‘Clothes Hanger’ 시스템의 기능 작동 여부와 그 기능의 성능을 보장하기 위해 진행되는 test에 대한 테스트 정책(Test Policy)과 테스트 케이스(Test Case)에 대해 기술한다.

11.2 Testing Policy

‘Clothes Hanger’ 시스템 개발 중에 진행되는 테스트는 Development test, System test, Acceptance test 총3가지로 나눌 수 있다.

A. 개발 진행 시 시스템의 기능을 테스트하는 Development test는 Unit test, Integration test로 나눠진다. 먼저, Unit test는sub-system내의 하나의 모듈이 개발될 때마다 수행하는 테스트로, 원하는 input값에 적절한 output을 내는지 모듈 단위로 확인하는 테스트이다. Unit test를 통과한 모듈은 설계시 지정한 sub-system으로 통합되어야 하는데, 이때 수행되는 테스트가 Integration test이다. 따라서 해당 테스트 시에는 모듈들의 통합이 설계에 맞춰 잘 이뤄지는지를 확인한다. Integration test를 성공하면 목표로 하는 system의 구성요소인 하나의 sub-system이 성공적으로 작동한다는 점을 확인할 수 있다.

B. System test는, 앞서 성공적으로 작동하는 sub-system들이 목표로 하는 system으로 통합될 때 진행되는 test이다. 따라서 sub-system이 하나의 시스템으로 통합됨에 있어서 발생하는 문제들을 살펴보기 위한 작업이다. 또한 통합 후에 발생하는 문제를 확인하고 sub-system사이의 관계에 대한 인식하지 못했던 오류들을 확인하고 시스템의 성능을 점검하는 test 단계이다. 즉, 사용자에게 배포되기 전에 해당 system이 하나의 시스템으로써 기능이 정상적으로 작동되고 사용자의 요구사항에 맞는 성능을 제공하는 시스템인지 확인하는 test 단계이다.

C. User test는 사용자의 환경에 system을 맞춰, 해당 시스템을 사용자가 사용할 수 있게 사용자 환경에서 적합하게 돌아가는지를 확인하는 test단계이다.

11.3 Test Case

사용자가 해당 시스템의 기능에서 원하는 동작에 대해 살펴보고 성공사례와 실패사례를 예측함으로써 test를 계획한다.

11.3.1 로그인 및 회원가입

A. 판매자 회원가입

- 사용자: 자신의 개인정보를 입력함으로써 시스템을 이용할 권한을 얻는다.

- 시스템 동작: 사용자가 입력한 개인정보를 사용자 정보 서버에 요청해 DB의 내용과 일치하는지 확인한다.

(회원가입 성공)

- 시스템 알림: “회원가입에 성공했습니다.”

- 시스템 기능: 사용자의 id와 password를 해당 시스템 사용자 정보 DB에 등록한다. 판매자에 맞는 권한을 부여한다.

(회원가입 실패)

- 시스템 알림: “회원가입에 실패하였습니다.”

B. MD 회원가입

- 사용자: 자신의 개인정보를 입력함으로써 시스템을 이용할 권한을 얻는다.

- 시스템 동작: 사용자가 입력한 개인정보를 사용자 정보 서버에 요청해 DB의 내용과 일치하는지 확인한다. MD의 경우, 기업이 지닌 사원 정보 DB내 MD의 사번이 일치하는 지를 확인해야 한다.

(회원가입 성공)

- 시스템 알림: “회원가입에 성공했습니다.”

- 시스템 기능: 사용자의 사번과 password를 해당 시스템 사용자 정보 DB에 등록한다. MD에 맞는 권한을 부여한다.

(회원가입 실패)

- 시스템 알림: “회원가입에 실패하였습니다.”

B. 판매자 로그인

- 사용자: 판매자로서 자신이 지닌 id, password를 입력해 판매자 페이지에 접속한다.

- 시스템 동작: 회원가입시 등록된 사용자 정보 DB에서 입력된 판매자 id, password를 대조한다.

(로그인 성공)

- 판매자 전용 페이지로 진입한다.

(로그인 실패)

- 시스템 알림: “로그인에 실패하였습니다. Id와 password를 확인하세요. ”

C. MD 로그인

- 사용자: MD로서 자신이 지닌 사번, password를 입력해 판매자 페이지에 접속한다.

- 시스템 동작: 회원가입시 등록된 사용자 정보 DB에서 입력된 MD 사번, password를 대조한다.

(로그인 성공)

- MD 전용 페이지로 진입한다.

(로그인 실패)

- 시스템 알림: “로그인에 실패하였습니다. 사번과 password를 확인하세요. ”

11.3.2 판매자 파일 업로드

A. 상품정보파일 업로드

- 사용자: 양식에 맞춘 하나의 상품 정보 파일을 제출한다.

- 시스템 동작: 사용자가 제출한 파일이 엑셀파일이고 하나의 파일인지 확인하고 ‘Clothes Hanger’ 서버에 판매자 id를 폴더명으로 하는 폴더 내에 저장한다.

(상품정보파일 업로드 성공)

- 다수의 이미지파일을 업로드할 수 있는 페이지로 진입한다.

(상품정보파일 업로드 실패)

- 시스템 알림: “업로드에 실패하였습니다. 파일의 양식을 확인하세요.”

B. 상품이미지파일 업로드

- 사용자: 상품 정보에 맞는 다수의 이미지 파일을 제출한다.

- 시스템 동작: 사용자가 제출한 상품정보와 일치하는 이미지 파일인지 안내 메시지를 통해 공지하고 ‘Clothes Hanger’ 서버에 판매자 id를 폴더명으로 하는 폴더 내에 저장한다.

(상품이미지파일 업로드 성공)

- 시스템 알림: ”성공적으로 상품이 등록되었습니다. 상품 승인 현황을 확인하세요.”

- 시스템 기능: 확인 버튼을 누르면 상품 승인 현황 페이지로 이동한다.

(상품이미지파일 업로드 실패)

- 시스템 알림: “업로드에 실패하였습니다. 파일의 양식을 확인하세요.”

11.3.3 딥러닝 이미지 분류

- 사용자: MD는 등록된 상품 정보, 이미지 파일을 확인 후 해당 상품 등록의 승인여부를 결정한다.

- 시스템 동작: 시스템은 이미지 내용과 분류가 일치하는 상품에 대해서는 ‘일치’, 일치하지 않는 상품에 대해서는 ‘일치하지 않음’이라는 라벨과 함께 시스템에 저장한다.

(딥러닝 이미지 분류 성공)

- MD가 상품 승인 버튼을 누름으로써 상품 등록이 승인된다.

(딥러닝 이미지 분류 실패)

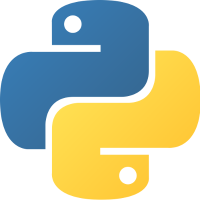
- MD가 직접 상품이미지를 확인하고 내용을 수정한다.

**12. Development Environment**

12.1 Objective

Development Environment에서는 프로그램을 개발하는 개발환경과 개발에 사용하는 IDE를 서술한다.

12.2 Programming Language & IDE

A. Programming Language

기본적으로 사용하는 언어는 Python이다. 웹 개발을 위한 프레임워크 Django를 사용하기 위한 언어이자, 딥러닝 이미지 처리를 위한 라이브러리 Pytorch도 Python 언어로 되어 있다. 웹 백엔드와 딥러닝 이미지분류 프로그램 개발에 사용한 버전은 Python 3.7.3, Django 2.1.2, Pytorch 1.1.0 버전이다.

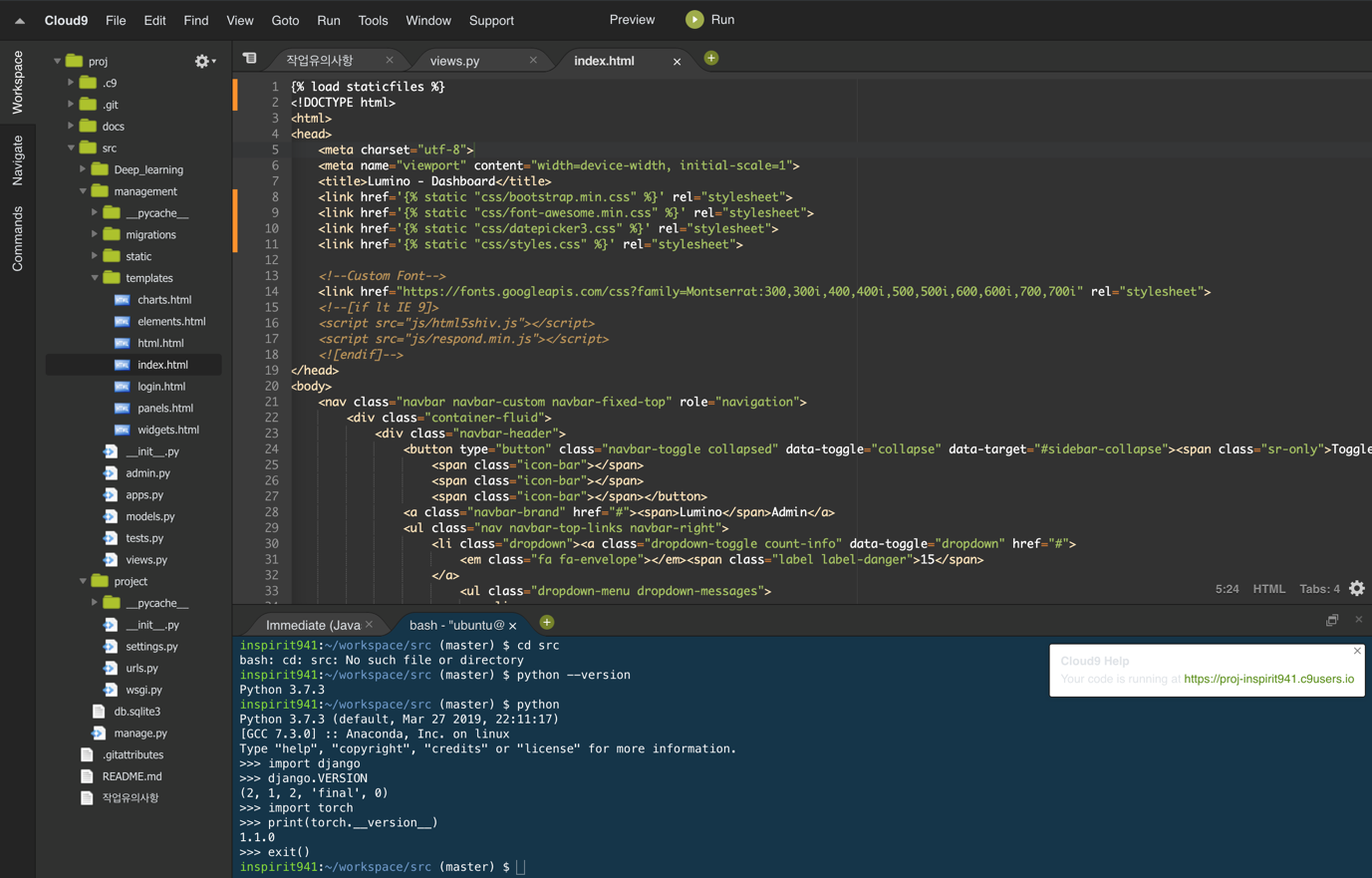
 

웹 프론트에서 사용한 언어는 html과 css, javascript이며, 오픈소스 라이브러리인 Bootstrap을 사용해 웹 프론트 부분의 디자인과 동적 영역을 작업했다.

B. IDE

Cloud9 IDE를 사용했다. Cloud9 IDE는 버전 3.0부터 오픈소스로 출시되고 있는 온라인 통합 개발 환경(IDE)이다. 2016년 아마존에 인수된 뒤로 AWS cloud9으로 유료화되었으나, AWS에 인수되기 전에 계정이 있을 경우2018년 12월까지는 무료로 서비스 이용이 가능하다. Ubuntu 버전 14.04, 최대 5GM RAM과 10GB의 HDD, Python 2.7, Django 1.9 버전을 기본적으로 지원한다. 하지만 Python 2 버전은Pytorch 라이브러리를 지원하지 않기 때문에, Python 패키지 버전 통합성과 라이브러리 설치의 용이성을 위해 Anaconda3을 설치해서 사용했다.



  
Cloud9 IDE 개발환경 예시

12.3 개발규정

Ubuntu Server의 코드가 들어가는 모든 파일과 소스코드는 git repository를 생성하여 관리한다. 동일한 개발환경에서 작업하지만, 작업을 마친 후 git push로 repository에 commit을 수행하는 방법으로 형상관리를 진행한다.

Backend 언어는 Python을 사용하고, Frontend는 javascript, html, css를 사용한다. 이미지 처리를 위한 웹 크롤러는 Python으로 작업하며, 이미지 딥러닝 모델 학습을 위해 google colab의 GPU 지원을 적극 활용한다. 엑셀 파일로 제공되는 상품 데이터는 Python Pandas Library를 활용한 Function으로 Database에 자동 migration을 진행한다.