

# 영상 내 불건전 콘텐츠 자동 필터링 서비스



## Requirement Specification

### GROUP 1

2011314730 김용성

2013313184 서원빈

2013314600 이영우

2014311876 조승진

2013314750 허만갑

# Table of Contents

<b>1. Preface</b>	<b>2</b>
1.1. Objective	2
1.2. Readers	2
1.3. Document Structure	2
<b>2. Introduction</b>	<b>5</b>
2.1. Objective	5
2.2. Needs	5
2.3. Contents Filtering Service	6
2.4. Expected Effect	7
<b>3. User requirement definition</b>	<b>8</b>
3.1. Objective	8
3.2. Functional Requirement	8
3.3. Non-Functional Requirement	9
<b>4. System Architecture</b>	<b>11</b>
4.1. Objective	11
4.2. System Architecture	11
<b>5. System Requirement Specification</b>	<b>12</b>
5.1. Objective	12
5.2. Functional Requirement	12
5.3. Non-Functional Requirement	18
<b>6. System Models</b>	<b>20</b>
6.1. Objective	20
6.2. Context Models	21
6.3. Interaction Models	23
6.4. Structural Models Requirement	30
6.5. Behavioral Models	31
<b>7. System Evolution</b>	<b>32</b>
7.1. Objective	32
7.2. Limitation and Assumption	32
7.3. Evolution of User Requirement	32
7.4. Evolution of Environment	33
<b>8. Appendices</b>	<b>34</b>
8.1. Objective	34
8.2. Database Requirements	34
8.3. User-System Requirements	36

# 1. Preface

## 1.1. Objective

Preface는 본 문서의 예상되는 독자들과 전반적인 구조, 그리고 각 부분의 역할에 대해 설명한다. 또한, 각 버전 관리 정책, 버전 변경 기록, 그리고 문서의 변경사항들을 설명한다.

、

## 1.2. Readers

본 요구사항 명세서는 독자에 따라 User Requirement와 System Requirement로 구성되어 있다.

### A. User Requirement Readers

사용자의 관점에서 사용자에게 제공되는 서비스에 대하여 설명한다. Non-functional system requirements도 설명한다. 자연어, 다이어그램 등보다 사용자가 이해하기 쉽도록 명세한다.

### B. System Requirement Readers

Software developer의 관점에서 본 프로젝트에서 개발해야 하는 시스템의 특정 기능이 어떻게 구현되어야 하는지 설명한다. Non-functional system requirements도 설명한다. 주요 독자인 software developer외에도 client engineer, system architects, 경우에 따라서는 system end-user까지도 독자가 될 수 있다.

## 1.3. Document Structure

본 요구사항 명세서는 총 9개의 부분으로 구성된다. Preface, Introduction, User Requirements Definition, System Architecture, System Requirements Specification, System Models, System Evolution, Revenue Model, Appendices, Index로 구성된다. 각 장의 역할과 전반적인 서술 내용은 다음과 같다.

## **A. Preface**

본 문서의 예상되는 독자들과 전반적인 구조, 그리고 각 부분의 역할에 대해 설명한다. 또한, 각 버전 관리 정책, 버전 변경 기록, 그리고 문서의 변경사항들을 설명한다.

## **B. Introduction**

본 시스템의 필요성과 해당 시스템이 반영하고 있는 needs를 설명한다. 또한, 주요 기능들과 이 기능들이 각 target층에게 어떠한 기대효과를 줄 수 있는지 설명한다.

## **C. User Requirements Definition**

본 요구사항 명세서에서 제시한 사용자에게 제공되는 서비스에 대하여 설명한다. Non-functional system requirements도 설명한다. 자연어와 다이어그램과 같은 방법들로 설명하여 보다 더 독자의 이해를 돕는다. Product나 process가 반드시 따라야 하는 표준이 있다면 반드시 서술되어야 한다.

## **D. System Architecture**

본 요구사항 명세서에 제시한 목표 시스템의 architecture에 대한 개요를 보여준다. 시스템의 architecture를 각 기능별로 설명한다. 목표시스템은 5가지의 고유한 기능하도록 구성된다. 각 기능은 기능별로 user-interface system과 server system으로 구성되며, 각 시스템은 다시 기능을 제공하기 위한 sub system으로 구성된다.

## **E. System Requirements Specification**

Functional requirements와 non-functional requirements를 설명한다. 또한, 각 기능에 따른 scenario를 설명하여 각 기능의 흐름을 간략적으로 확인할 수 있다.

## **F. System Models**

System component, system, 그리고 system environment사이의 관계를 설명한다. 가능한 그래픽 모델로는 object models, data-flow models, semantic data models등이 있다.

## **G. System Evolution**

시스템이 세우고 있는 주요한 가정들에 대해 설명한다. 또한, 시스템에 일어날 수 있는 예상되는 변화들, 가령, 하드웨어의 발전, 사용자의 needs변화 등에 대해 설명한다. 이 장은 차후에 발생할 시스템의 설계 변경을 피할 수 있도록 도와주기 때문에 시스템 설계자에게 도움이 될 것이다. 이를 통해, 변화에 잘 적응하는 시스템을 개발할 수 있다.

## **H. Appendices**

개발되는 시스템에 대한 더 자세하고 구체적인 정보들을 제공한다. Hardware, database 등을 설명한다. 시스템 사용에 적합한 혹은 최소의 설정, database 상에서 결정되는 데이터의 관계 혹은 데이터의 구조에 대해서 설명한다.

## **I. Index**

문서들의 인덱스들이 포함된다. 테이블, 다이어그램, 그림 각각의 인덱스가 포함된다.

## 2. Introduction

### 2.1 Objective

본 시스템의 필요성과 해당 시스템이 반영하고 있는 needs를 설명한다. 또한, 주요 기능들과 이 기능들이 각 target층에게 어떠한 기대효과를 줄 수 있는지 설명한다.

### 2.2 Needs

바야흐로 동영상의 시대가 도래했다. 최근 5년 동안 동영상 시청으로 인한 데이터 트래픽이 급증하였고 전체 온라인의 70%에 육박하는 엄청난 수치를 기록하고 있다. 모바일 동영상 어플 중에서 독보적인 지위를 확보한 유튜브의 경우 기존의 검색 엔진의 역할을 대체하고 있고 사용 시간에서 포털 사이트를 압도하는 결과를 보이고 있다. 유튜브의 공식 발표에 따르면, 전 세계 시청자들이 하루에 시청하는 시간이 약 10억 시간에 달한다고 한다. 이러한 추세에 힘입어 한국에서도 유튜브의 성장은 매년 가파르게 상승하고 있고 앱 사용시간 측면에서 타 서비스들을 압도하고 있다. 이렇듯 동영상 콘텐츠는 일반 사용자들의 일상에 깊게 녹아 들고 있으며 사용자들의 감정과 가치관 형성에 영향을 미치고 있다.

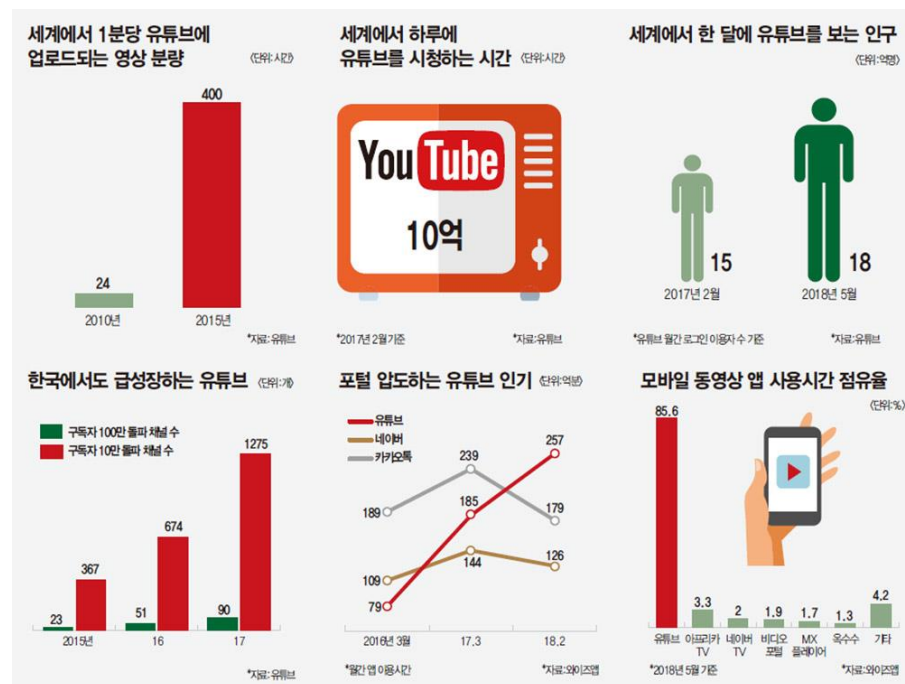


Figure 1 유튜브 사용량 추이

하지만 무수한 동영상의 범람하면서 이에 따른 부작용도 속출하고 있다. 동영상의 조회수가 광고 수익으로 직결되기 때문에, 자극적인 소재를 미끼로 시청자들을 유입시키는 경우가 빈번히 발생한다. 조회수에 집착한 나머지 유해한 영상들이 무분별하게 퍼지고 있으며 이용자들이 무심코 이러한 영상을 클릭하게 되면서 불쾌함을 겪고 심하게는 정신적 충격에 이르기기도 한다. 따라서 유해한 영상들이 온라인 상에 버젓이 자리잡게 되면 가치 혼란이 야기될 수 있고 이러한 현상이 지속되면 사회적 비용을 초래하게 된다.

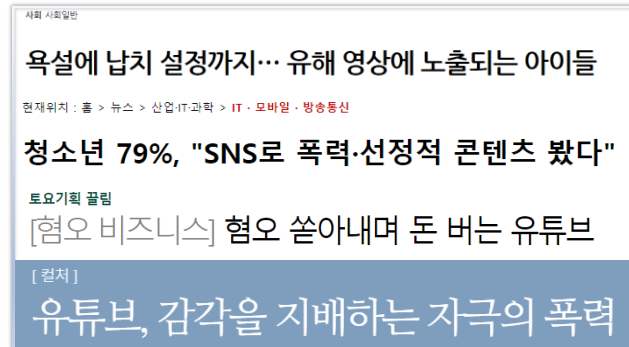


Figure 2 선정적 영상의 문제점을 다루는 기사

## 2.3 Contents filtering service

수많은 동영상들을 하나하나 확인하고 업로드 여부를 결정하는 것은 인적 자원이나 소요 시간 측면에서 불가능에 가깝다. 따라서 동영상 내 유해 콘텐츠를 자동으로 체크하고 필터링할 수 있는 서비스의 필요성을 인지하여 머신러닝 기반의 동영상 변환 사이트를 구상하였다.

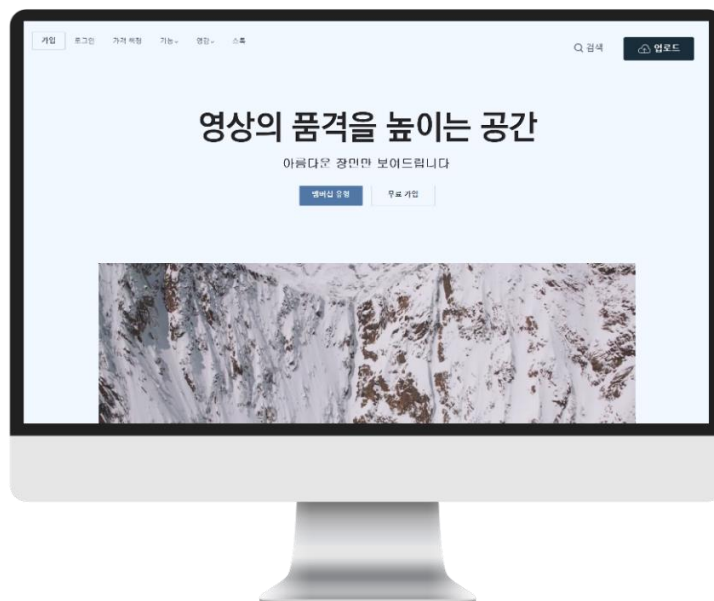


Figure 3 콘텐츠 필터링 웹사이트 컨셉

서비스 이용 방법은 굉장히 간단하다. 이용자가 로그인을 한 후 필터링을 원하는 영상을 웹사이트에 업로드하면 유해한 장면이나 음성을 인지하여 모자이크를 하거나 빼처리 한 뒤 동영상을 변환하는 것이다. 동영상 변환 모드는 알고리즘에 의해 자동으로 장면들을 변환시켜주는 원클릭 모드와 사용자에게 감지된 장면 별로 변환 여부를 묻는 매뉴얼 모드가 있다.

예를 들어, 영상에 피가 낭자하거나, 거친 폭력이 행해지거나, 칼과 같은 흉기가 등장하는 등 학습된 결과에 의해 불건전 장면으로 판단되면, 사물이나 사람, 장면까지도 모자이크로 변환한다. 또한 영상 내 음성을 텍스트로 변환한 뒤, 욕설이나 비하 발언 등 부적절한 표현이 발견된 지점을 기록하여, 해당 시간대에 대체 표현이나 빼-처리를 해주는 방식으로 변환을 진행한다.

## 2.4 Expected effect

### A. 플랫폼 제공자

유튜브, 페이스북 등의 영상 플랫폼 제공자가 유저들의 이탈을 방지하기 위해서는, 유해 콘텐츠들을 거르기 위한 적극적인 장치가 필요하다. 사용자들이 피로감을 느끼게 되면 금방 다른 서비스를 찾을 수 있기 때문에 양질의 콘텐츠 제공이 중요한 것이다. 현재는 동영상 자체를 차단시키기 때문에 완벽하다고 볼 수 없다. 따라서 이러한 영상 변환 서비스가 일정 부분 활용될 수 있는 여지가 있다.

### B. 영상 편집자

개인 방송이 활발해짐에 따라 전문 영상 편집 수요 또한 증가하고 있다. 100 만 구독자 이상의 거대한 크리에이터들의 경우 하루에도 2,3 개 이상의 영상들을 업로드하게 되는데, 편집자들이 매번 유해 장면들을 거르고 모자이크 처리하기에는 상당한 시간이 소요된다. 따라서 장면 필터링 서비스를 통해 업무량을 줄일 수 있고 영상의 재미를 살리는 편집에 집중할 수 있다.

### C. 일반 유저

자극적 콘텐츠들로 인한 시청자들의 피로감이 누적된 상태이다. 이에 따라 건전한 동영상 문화를 향한 자성의 목소리가 존재하고, 이러한 취지에 부합하는 영상 콘텐츠 배급에 서비스가 기여할 수 있다.



## 3. User Requirement Definition

### 3.1 Objective

해당 장에서는 본 시스템이 필수적으로 제공해야 하는 서비스에 대하여 기술한다. 또한 제품이나 시스템 프로세스가 따라야 할 제약 사항에 대해서도 서술한다. Functional Requirements 를 포함한 Non-Functional Requirements 에 대해서 상세히 기술한다.

### 3.2 Functional Requirements

#### A. Sign up

사용자의 정보를 받아 저장하는 기능이다. 업로드한 동영상 히스토리를 볼 수 있거나 다른 사람이 영상을 접근하지 못하도록 사용자의 정보를 기록한다.

#### B. Sign in

sign up 에 입력한 정보를 기반으로 사용자를 인증한다. 사용자가 업로드한 동영상 히스토리를 보거나 다른 사람의 영상에 접근하지 못하도록 한다.

#### C. Upload Video

자동으로 선정적인 내용을 필터링 혹은 블러링 하기 위한 영상을 올린다. 영상을 한 번 업로드하면 임시 저장하여 다음에 들어와도 확인할 수 있게 한다. 영상의 길이가 10 분 이상이면 경고창을 띄워 막는다.

#### D. Detection

업로드한 동영상의 선정적인 부분을 탐색한다. 여성의 가슴, 남성의 성기 부분 등이 나오는 부분을 탐색한다. 음성으로는 욕설, 성적인 단어 등이 나오는 부분을 탐색한다. 탐색하여 나오는 장면들을 사용자가 볼 수 있게 어떤 이유로 선정적이다 라고 판단했는지 보여준다. 영상의 썸네일, 탐색한 부분의 시간, 선정적으로 판단한 이유를 사용자에게 리스트로 보여준다.

#### E. Blurring

업로드 한 동영상에서 선정적으로 탐색된 부분을 모자이크한다. 또한 욕설이 들어간 음성에는 뻘처리를 한다.

## **F. Select Scenes**

선정적으로 판단한 부분들을 사용자에게 보여주면 사용자가 블러링을 선택할 수 있게 한다. 자동으로 선정적으로 판단된 부분을 사용자는 확인하여 선택적으로 포함한다. 사용자는 뺄처리가 필요 없다면 영상 블러링만 할 수도 있다. 또한 음성 뺄처리만을 원한다면 음성만 처리한다.

## **G. Download Video**

사용자가 선택한 부분만 처리한 영상을 다운로드 한다. 선정적으로 판단한 부분 중 사용자가 선택한 부분만을 처리하여 영상의 링크를 보여준다. 사용자는 링크를 클릭하여 해당 영상을 다운받을 수 있다.

# **3.3 Non-Functional Requirements**

## **A. Product Requirement**

### **a. Performance Requirement**

해당 시스템은 사용자가 필터링이 필요한 영상을 업로드하고, 해당 영상에서 불건전한 콘텐츠가 필터링 되는 과정 및 결과물을 다운로드 하는 과정에 문제가 없어야 한다. 또한, 필터링 과정에서 불건전한 콘텐츠를 처리하는 프로세스가 확실히 이루어 져야 하며, 소요되는 시간을 최소화 하여 높은 수준의 Performance 를 보장하여야 한다.

### **b. Security Requirement**

해당 시스템은 사용자의 개인 정보(ID, 비밀번호 등)를 저장하고 이전에 업로드한 영상과 결과물에 대한 기록을 제공하기 때문에, 유출되었을 경우 개인에 대한 여러 정보가 퍼지는 불상사가 있을 수 있다. 따라서 보안성이 높은 시스템을 설계해야 한다.

### **c. Usability Requirement**

해당 시스템은 사용자가 편리하게 서비스를 이용할 수 있도록 선택지를 제공한다. 기본적으로 콘텐츠를 필터링하는 과정에서, 커스터마이징이 가능하도록 하여 불건전한 콘텐츠로 판단된 영상의

프레임 또는 구간에 대한 목록을 제공하고 해당 부분을 필터링할지 하지 않을지를 선택할 수 있는 커스터마이즈 모드를 제공하고, 자동적으로 변환을 원할 경우 자동 모드로 서비스를 이용할 수 있는 선택지를 제공한다.

## **B. Organization Requirement**

### **a. Environmental Requirement**

해당 시스템은 영상을 실시간으로 업로드하고 필터링하는 서비스를 제공하기 때문에, 업로드 및 다운로드의 용이성 및 편리한 접근을 위해 Web 을 기반으로 서비스를 제공한다.

### **b. Operational Requirement**

해당 시스템은 이용자에게 영상 업로드, 필터링 모드 선택, 필터링, 다운로드를 하는 일련의 과정이 유연하게 이어져야 하고, 해당 기능을 주력으로 제공한다.

## **C. External Requirement**

### **a. Regulatory Requirement**

해당 시스템은 사용자의 개인정보를 입력 받는 회원가입을 전제로 하는 시스템이다. 따라서 사용자의 동의 사항에 명시된 대상이 아닌 경우에는 사용자 개인 정보에 접근할 수 없도록 한다. 또한, 개인의 영상 결과물 기록을 제공하기 때문에, 해당 기록 역시 공개에 동의하지 않는다면 다른 사용자가 접근할 수 없도록 해야한다.

### **b. Safety/Security Requirement**

해당 시스템은 외부나 허가 받지 않은 내부의 접근으로부터 안전해야 한다. 사용자의 개인정보가 유출 또는 조작되지 않게 여러 수단을 통해 보안성을 안정성을 강화해야 한다. 또한 관리자는 비정상적인 접근이 감지될 경우 대처가 가능해야 하며, 보안성에 대한 대응 방안, 책임 사항 등을 확고히 해야 한다.

## 4. System Architecture

### 4.1 Objective

해당 장에서는 목표 시스템의 Architecture 에 대한 추상화를 보여준다. 시스템의 기능을 수행하는 단위군으로 묶어 설명한다. 다른 서비스의 사용도 포함한다.

### 4.2 System Architecture

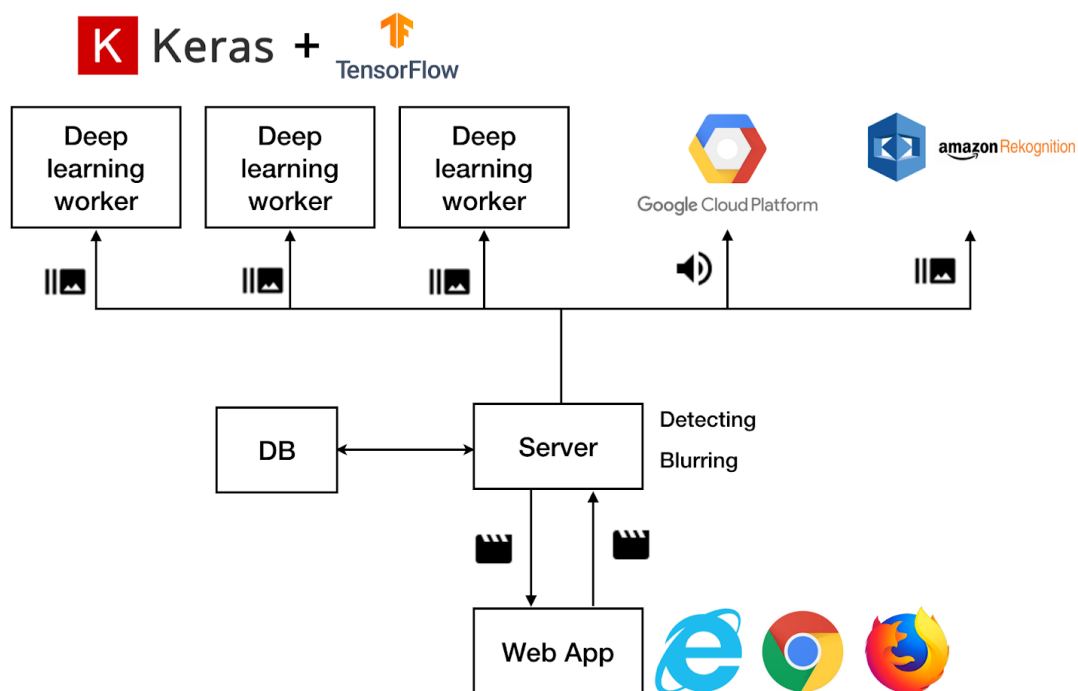


Figure 4 콘텐츠 필터링 시스템 구조

프로젝트에서 Deep learning 을 이용한 선정적인 화면 감지가 가장 중요하다. 동영상은 여러 이미지와 소리의 결합이다. 동영상의 이미지들을 Deep learning 모델의 입력으로 받아 선정적인지를 판단한다. 빠르고 가벼운 Deep learning 모델로 동영상 전체를 한번 탐색한다. 그 다음 복잡한 Deep learning 모델로 이전 모델이 확실하게 선정적이거나 선정적이지 않다고 판단한 부분을

제외한 나머지 부분들을 대상으로 다시 탐색한다. 추가로 복잡한 모델이 확실하게 선정적이거나 선정적이지 않다고 판단한 부분을 제외한 나머지 부분들을 Amazon Rekognition 서비스를 이용하여 탐색의 효율을 높인다. 욕설탐색은 먼저 Google Cloud Platform 을 이용하여 소리를 텍스트로 변환한다. 변환한 이후 나온 텍스트들에서 미리 정의한 욕설에 포함되는 단어가 있는지를 확인하여 선정성 여부를 판단한다. 이렇게 수집한 결과를 서버에서 모아 사용자에게 보여준다. 사용자는 이 중 몇몇 화면들을 선택하여 블러링 여부를 선택하면 서버에서 해당 화면들을 블러링하여 다시 동영상으로 만든다. 사용자는 다운로드 링크를 통해 블러링된 영상을 다운받는다.

## 5. System Requirement Specification

### 5.1 Objective

System Requirement Specification 에서는 Functional requirements 또는 Non-functional requirements 에 대해 더욱 자세히 설명한다. 다른 시스템에 대한 인터페이스 역시 이 장에서 정의된다.

### 5.2. Functional Requirements

#### A. Sign Up Function

---

##### Functional Requirement

기능	Sign Up Function
설명	사용자에게 회원 가입을 위해 필요로 되는 정보를 요청하는 기능이다. 사용자는 해당 서비스에서 제공하는 기능을 이용하기 위해서는 회원가입을 해야 하며, 개인의 영상 필터링 기록을 남기기 위해 필요한 기능이다. 이용약관에 동의해야 회원가입이 가능하다.

---

<b>입력</b>	아이디, 비밀번호, 이메일, 성명, 성별 등의 개인정보를 입력하며, 이메일 인증을 통해 본인 인증을 진행한다. 해당 정보는 필수 정보이므로, 누락된 정보가 있다면 가입이 진행되지 않는다.
<b>출력</b>	사용자의 서비스 가입이 완료 되었다는 팝업 메시지를 출력 하며, 이메일 인증이 되지 않았다면 오류 메시지를 출력한다.
<b>처리</b>	사용자가 기입한 정보를 데이터베이스에 저장한다.
<b>조건</b>	회원가입한 USER 정보 보관 및 관리를 위한 데이터베이스에 정보를 저장한다.

## B. Sign In Function

### Functional Requirement

<b>기능</b>	Sign In Function
<b>설명</b>	사용자가 회원가입을 통해 생성된 아이디를 이용하여 시스템에 로그인하는 기능이다. 사용자는 가입된 아이디를 이용해 시스템에 로그인하여 자신의 데이터베이스에 접근할 수 있으며, 시스템에서 제공되는 서비스를 이용할 수 있다.
<b>입력</b>	사용자에게 회원가입 시 생성된 아이디와 비밀번호를 입력 받는다.
<b>출력</b>	로그인 정보가 일치하는 경우, 로그인에 성공하였음을 알리는 팝업창을 출력한다. 로그인 정보가 일치하지 않는 경우, 로그인에 실패하였음을 알리는 팝업창을 출력한다

<b>처리</b>	데이터베이스 검색을 통해 사용자가 입력한 정보와 일치하는 계정이 있는지 확인 후 로그인 작업을 진행한다.
<b>조건</b>	회원가입한 USER 정보 보관 및 관리를 위한 데이터베이스에 정보를 검색한다.

### C. Upload Video Function

#### Functional Requirement

<b>기능</b>	Upload Video Function
<b>설명</b>	사용자가 필터링을 원하는 영상을 서비스에 업로드하여, 해당 영상에 대한 필터링을 진행하는 작업을 위해 Video 를 웹서비스에 업로드하는 기능을 제공한다.
<b>입력</b>	Video(mp4, avi, mpeg4 등 표준 영상 확장자)
<b>출력</b>	Video 업로드가 완료되면, 완료되었다는 팝업을 띄운 뒤, 해당 영상을 스트리밍 할 수 있는 기능을 제공한다. 실패 시에는 업로드가 되지 않았다는 팝업창을 띄운다.
<b>처리</b>	사용자의 로컬에서 영상을 업로드하고, 임시적으로 해당 영상 정보를 기록한다.
<b>조건</b>	영상이 표준 영상 확장자와 일치해야 하며, 영상의 재생시간은 10 분 이하(임시)로 제한한다.

## D. Detection Function

### Functional Requirement

기능	Detection Function
설명	영상 내의 불건전한 콘텐츠에 대한 감지를 하는 기능으로, 사용자가 영상 업로드 후 변환에 대한 요구를 하면, 해당 영상을 탐색하여 선정적인 요소(혐오장면, 성적인 요소, 욕설)등을 탐색하는 기능이다.
입력	업로드된 Video 내의 각 Frame, 음성 정보
출력	선정적인 요소가 포함된 Frame의 연속적인 구간을 썸네일 방식으로 각각 제공하며, 음성의 경우 해당 구간을 리스트로 출력하여 사용자가 확인할 수 있도록 한다.
처리	영상의 경우 기본적으로 Keras 및 TensorFlow 를 통해 Python 을 이용해 Deeplearning 기반의 Detection 작업을 수행하고, 해당 작업을 통해 영상의 어떤 구간에 선정적 요소가 있는지를 출력 후 php 를 이용해 해당 출력 정보를 response 받은 뒤, 웹서비스에 출력시킨다.
조건	일정 이상의 정확도가 나온 경우에 해당 결과를 반환하며, 결과에 애매한 부분이 있다면 추가적으로 AWS Rekognition 을 이용해 불건전 콘텐츠를 걸러낸다.



## E. Select Scenes Function

### Functional Requirement

기능	Select Scenes Function
설명	Detection Function 이후 사용자는 선정적 요소에 대한 영상 구간, 음성 구간 리스트를 확인할 수 있다. 해당 리스트를 사용자가 확인 후, 필터링 해도 되지 않는다고 판단이 되면 해당 구간을 필터링 대상에서 제거할 수 있고, 필터링이 필요한 구간을 선택적으로 취하고 확인하는 작업을 진행하는 기능이다.
입력	사용자의 선택
출력	최종적으로 선택된 필터링 대상 구간에 대한 정보를 사용자가 계속해서 확인할 수 있도록 한다. 또한 제거된 구간의 썸네일은 Gray 처리한다.
처리	선택된 구간의 정보를 Javascript 를 통해 저장한다. 제거된 경우 해당 Select 를 unable 한다.
조건	선택과 제거를 HTML 을 통해 사용자가 진행 후, Javascript 를 통해 해당 정보를 주고받는 작업에 누락이 없도록 작동해야 한다.

## F. Blurring Function

### Functional Requirement

기능	Blurring Function
설명	Detection Function, Select Scenes Function 을 거친 후 최종적으로 영상 내에서 필터링을 해야 하는 구간이 확정이 된 이후, 변환작업을 하는 과정이다. 이 과정에서 영상의 경우 Blur 처리를 통해 해당 구간을 모자이크 하며, 음성의 경우 (빠—)하는 Beep 음으로 변환하는 작업이다.
입력	Video 내에서 필터링을 해야 하는 구간에 대한 정보. Select Scenes Function 으로부터 해당 정보를 얻는다.
출력	Blurring 작업이 완료된 이후, 변환된 동영상을 사용자에게 보여주고 확인할 수 있도록 한다.
처리	구간에 대한 정보를 php 를 통해 영상에 대한 처리를 하는 python 파일에 넘긴다. 그 후 변환작업으로 반환된 영상을 서버에 저장 후, 영상을 사용자에게 재업로드 해준다.
조건	변환 과정 이후 영상을 서버에 저장해야 하고, 해당 경로를 데이터베이스에 저장하여 이후 사용자가 확인할 수 있도록 만들어야 한다.

## G. Download Video Function

### Functional Requirement

기능	Download Video Function
설명	최종적으로 선정적인 요소에 대한 필터링을 진행한 영상을 사용자가 다시 다운로드 받을 수 있도록 하는 기능이다.
입력	사용자가 다운로드 받기 원하는 영상
출력	영상의 다운로드가 완료되면, 팝업창을 띄운다
처리	다운로드를 원하는 영상을 사용자의 local 저장소에 저장한다.
조건	다운로드가 수행되지 않은 경우, 사용자의 History 페이지에서 재 다운로드를 받을 수 있다.

## 5.3 Non-Functional Requirements

### A. Product Requirement

#### a. Performance Requirement

해당 시스템은 사용자가 필터링이 필요한 영상을 업로드하고, 해당 영상에서 불건전한 콘텐츠가 필터링 되는 과정 및 결과물을 다운로드 하는 과정에 문제가 없어야 한다. 또한, 필터링

과정에서 불건전한 콘텐츠를 처리하는 프로세스가 확실히 이루어져야 하며, 소요되는 시간을 최소화 하여 높은 수준의 Performance 를 보장하여야 한다.

**b. Security Requirement**

해당 시스템은 사용자의 개인 정보(ID, 비밀번호 등)를 저장하고 이전에 업로드한 영상과 결과물에 대한 기록을 제공하기 때문에, 유출되었을 경우 개인에 대한 여러 정보가 퍼지는 불상사가 있을 수 있다. 따라서 보안성이 높은 시스템을 설계해야 한다.

**c. Usability Requirement**

해당 시스템은 사용자가 편리하게 서비스를 이용할 수 있도록 선택지를 제공한다. 기본적으로 콘텐츠를 필터링하는 과정에서, 커스터마이징이 가능하도록 하여 불건전한 콘텐츠로 판단된 영상의 프레임 또는 구간에 대한 목록을 제공하고 해당 부분을 필터링할지 하지 않을지를 선택할 수 있는 커스터마이징 모드를 제공하고, 자동적으로 변환을 원할 경우 자동 모드로 서비스를 이용할 수 있는 선택지를 제공한다.

**B. Organization Requirement**

**a. Environmental Requirement**

해당 시스템은 영상을 실시간으로 업로드하고 필터링하는 서비스를 제공하기 때문에, 업로드 및 다운로드의 용이성 및 편리한 접근을 위해 Web 을 기반으로 서비스를 제공한다.

**b. Operational Requirement**

해당 시스템은 이용자에게 영상 업로드, 필터링 모드 선택, 필터링, 다운로드를 하는 일련의 과정이 유연하게 이어져야 하고, 해당 기능을 주력으로 제공한다.

## C. External Requirement

### a. Regulatory Requirement

해당 시스템은 사용자의 개인정보를 입력 받는 회원가입을 전제로 하는 시스템이다. 따라서 사용자의 동의 사항에 명시된 대상이 아닌 경우에는 사용자 개인 정보에 접근할 수 없도록 한다. 또한, 개인의 영상 결과물 기록을 제공하기 때문에, 해당 기록 역시 공개에 동의하지 않는다면 다른 사용자가 접근할 수 없도록 해야한다.

### b. Safety/Security Requirement

해당 시스템은 외부나 허가 받지 않은 내부의 접근으로부터 안전해야 한다. 사용자의 개인정보가 유출 또는 조작되지 않게 여러 수단을 통해 보안성을 안정을 강화해야 한다. 또한 관리자는 비정상적인 접근이 감지될 경우 대처가 가능해야 하며, 보안성에 대한 대응 방안, 책임 사항 등을 확고히 해야 한다.

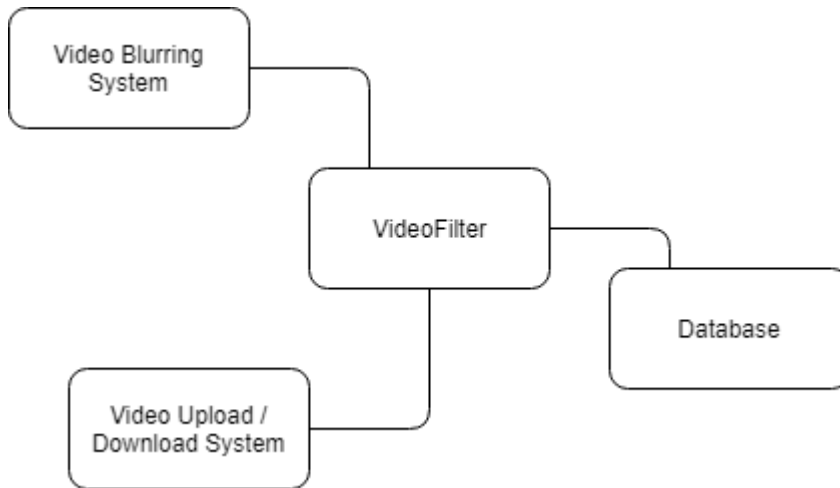
## 6. System Models

### 6.1 Objective

System Models에서는 특정 기능을 사용할 때 어떻게 시스템 컴포넌트, 시스템이 상호작용을 하는지 관계에 대해 기술한다. 이해를 돕기 위해 UML을 사용한 다양한 모델을 사용하여 해당 시스템을 표현하였다.

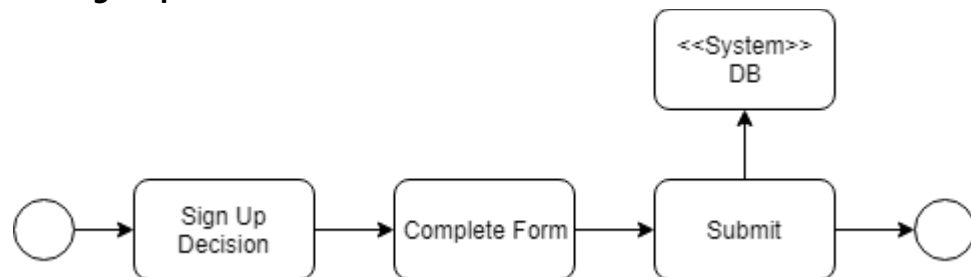
## 6.2 Context Models

### A. Context Model

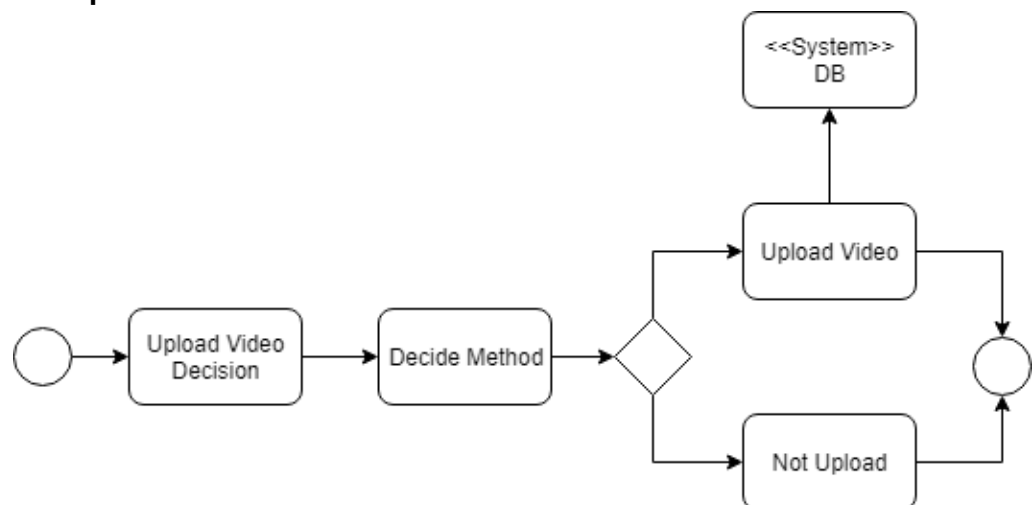


### B. Process Diagram

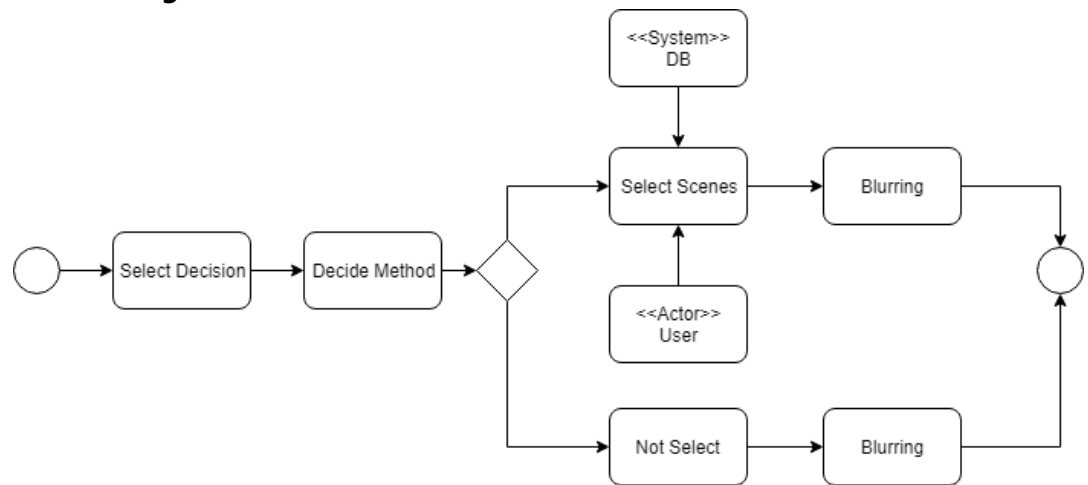
#### B.1 Sign Up Process



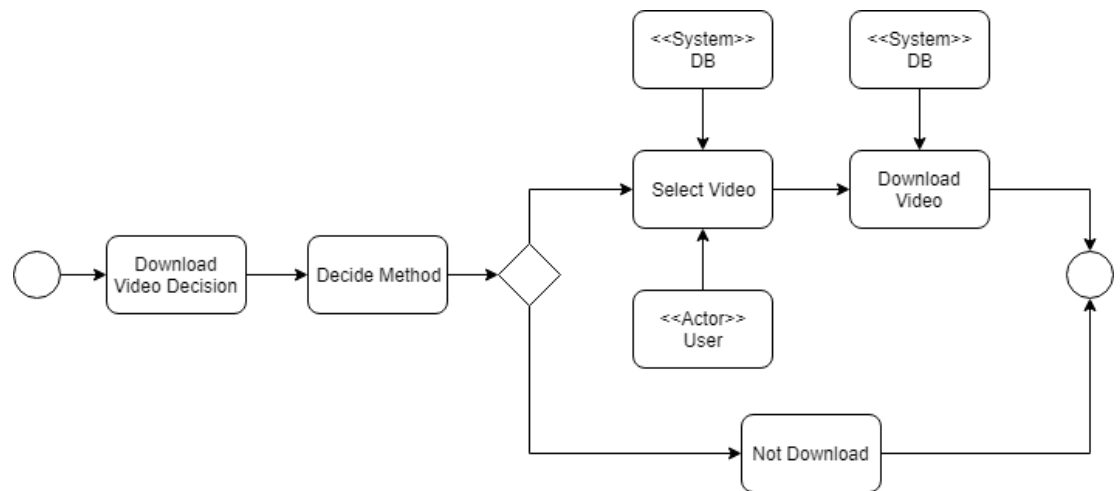
#### B.2 Upload Video Process



### B.3 Blurring Process

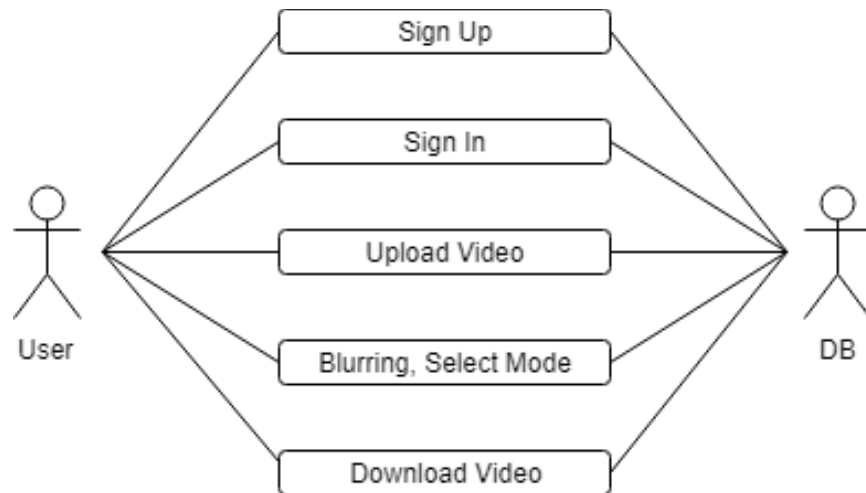


### B.4 Download Video Process



## 6.3 Interaction Models

### A. Use case Model



### B. Tubular Description For each use case

#### a. Sign Up

SUB SYSTEM	LOGIN MANAGER
USE CASE	Sign Up
ACTOR	User, DB
DESCRIPTION	<p>회원가입을 위해 사용자의 개인 정보를 요청하고 이를 데이터베이스에 저장하는 기능이다.</p> <p>User 는 VideoFilter 가 제공하는 서비스를 이용하기 위해 Sign up 을 진행해야 한다.</p>
STIMULUS	<p>User 의 Query 는 정보 입력을 Request 한다.</p> <p>입력이 완료되면 회원 가입을 위해 User 는 Request 를 보낸다.</p>
RESPONSE	입력을 완료한 경우 DB 에 해당 User 의 정보가 갱신되게 된다.



### b. Sign In

<b>SUB SYSTEM</b>	<b>LOGIN MANAGER</b>
<b>USE CASE</b>	Sign In
<b>ACTOR</b>	User, DB
<b>DESCRIPTION</b>	회원가입을 한 사용자가 서비스를 사용하기 위해 저장된 기존 정보와 입력 정보를 비교하여 사용자 인증을 하는 서비스이다. 가입된 사용자의 정보와 회원가입 시 저장된 데이터를 비교함으로써 서비스에 접근할 수 있다.
<b>STIMULUS</b>	사용자가 회원가입(Sign Up)시 설정한 아이디와 비밀번호를 입력 받는다. 입력이 완료되면 Sign In 기능에 대한 query 를 Request 한다.
<b>RESPONSE</b>	시스템은 사용자가 입력한 정보가 DB 에 존재하는지 확인하고 일치하는 정보가 있을 경우에 로그인 팝업창을, 없는 경우에 로그인 오류 팝업창을 출력한다

### c. Upload video

<b>SUB SYSTEM</b>	<b>VIDEO MANAGER</b>
<b>USE CASE</b>	Upload video
<b>ACTOR</b>	User, DB

<b>DESCRIPTION</b>	User 가 필터링을 원하는 동영상을 필터링하기 위해 진행하는 서비스 이다. 업로드를 마친 후에 동영상은 데이터베이스에 저장된다.
<b>STIMULUS</b>	User 가 형식에 맞추어 동영상을 입력하고 등록 Request 를 보낸다.
<b>RESPONSE</b>	User 가 입력한 동영상이 DB 에 저장되며 해당 User 의 정보 또한 갱신된다. 실패 시에는 업로드가 되지 않았다는 팝업창을 띄운다.

#### d. Select mode

<b>SUB SYSTEM</b>	<b>VIDEO MANAGER</b>
<b>USE CASE</b>	Select mode
<b>ACTOR</b>	User
<b>DESCRIPTION</b>	Select Scenes Function 으로부터 해당 정보를 얻어 server 가 선정적이라고 판단한 부분을 사용자가 확인한 후 블러링할 부분을 선택할 수 있게 하는 customizing mode 와 사용자의 선택없이 server 가 동영상을 블러링하는 mode 중 선택할 수 있는 서비스이다.
<b>STIMULUS</b>	사용자는 select mode 페이지에서 원하는 mode 버튼을 클릭함으로써 request 를 보낸다.
<b>RESPONSE</b>	사용자가 선택한 모드에 대한 블러링 페이지를 출력하고 서버에 해당 블러링 기능에 대한 request 를 보낸다.

e. Detecting

<b>SUB SYSTEM</b>	<b>VIDEO MANAGER</b>
<b>USE CASE</b>	Detecting
<b>ACTOR</b>	DB
<b>DESCRIPTION</b>	영상 내의 불건전한 콘텐츠에 대한 감지를 하는 기능으로, 사용자가 영상 업로드 후 변환에 대한 요구를 하면, 해당 영상을 탐색하여 선정적인 요소(혐오장면, 성적인 요소, 욕설)등을 탐색하는 기능이다.
<b>STIMULUS</b>	사용자가 detection 버튼을 클릭함으로써 request 를 보낸다.
<b>RESPONSE</b>	DB 에서 검색한 업로드된 동영상을 server 에 detection request 를 하고 detection 이 완료되면 선정적인 요소가 포함된 Frame 의 연속적인 구간을 썸네일 방식으로 출력하고 음성의 경우 해당 구간을 리스트로 출력한다.

## f. Blurring

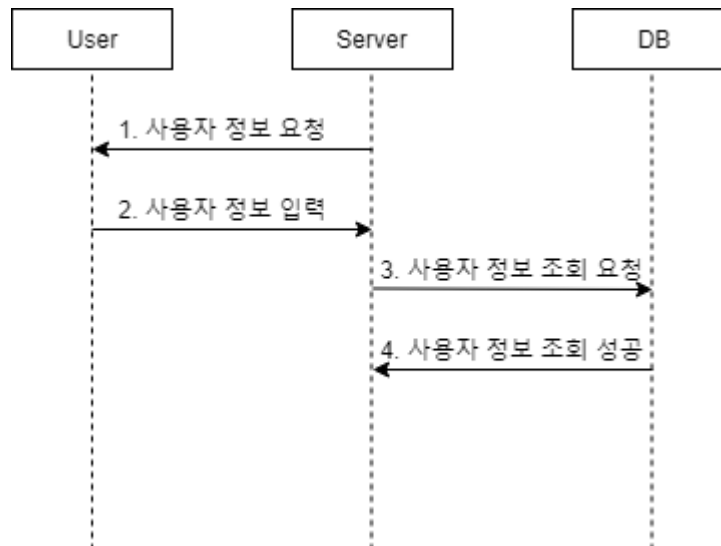
<b>SUB SYSTEM</b>	<b>VIDEO MANAGER</b>
<b>USE CASE</b>	Blurring
<b>ACTOR</b>	User, DB
<b>DESCRIPTION</b>	Detection Function, Select Scenes Function 을 거친 후 최종적으로 영상 내에서 필터링을 해야 하는 구간이 확정이 된 이후, 변환작업을 하는 과정이다. 이 과정에서 영상의 경우 Blur 처리를 통해 해당 구간을 모자이크 하며, 음성의 경우 (삐——)하는 Beep 음으로 변환하는 작업이다.
<b>STIMULUS</b>	User 는 detection 된 구간을 확인, 선택하고 blurring 버튼을 클릭함으로써 request 를 보낸다.
<b>RESPONSE</b>	선택된 구간에 대한 정보를 server 에 전달하고 blurring request 를 보낸다. blurring 작업이 완료된 이후, 변환된 동영상을 사용자에게 보여주고 확인할 수 있도록 출력한다. 확인이 완료되면 변환된 동영상을 DB 에 전달한다.

**g. Download video**

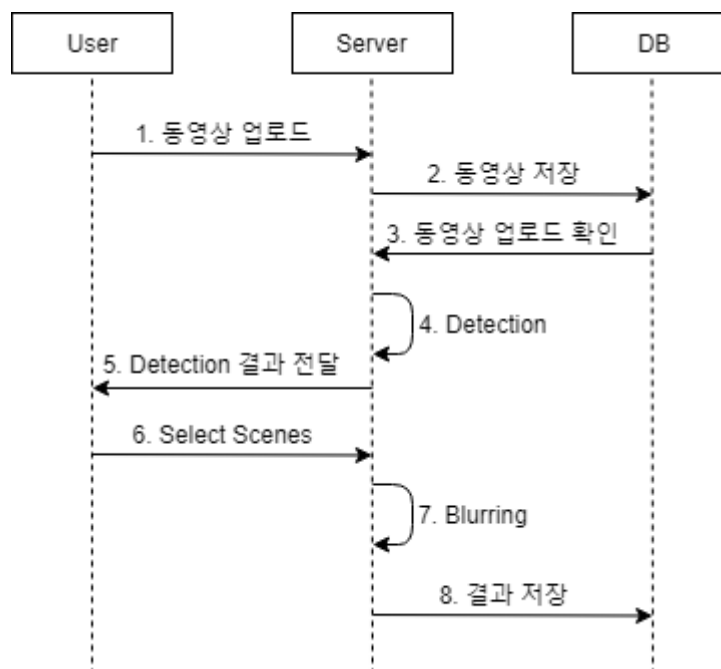
<b>SUB SYSTEM</b>	<b>VIDEO MANAGER</b>
<b>USE CASE</b>	Download video
<b>ACTOR</b>	User, DB
<b>DESCRIPTION</b>	최종적으로 선정적인 요소에 대한 필터링을 진행한 영상을 사용자가 다시 다운로드 받을 수 있도록 하는 기능이다.
<b>STIMULUS</b>	User 는 다운로드를 원하는 동영상을 확인, 선택하고 다운로드 버튼을 클릭함으로써 request 를 보낸다.
<b>RESPONSE</b>	DB 에 User 의 동영상 목록을 요청하고 User 가 다운로드 받고자 하는 동영상을 선택하면 선택된 동영상을 DB 에 요청한다. DB 로부터 전달받은 해당 동영상을 사용자의 local 저장소에 저장한다.

## C. Sequence Diagram

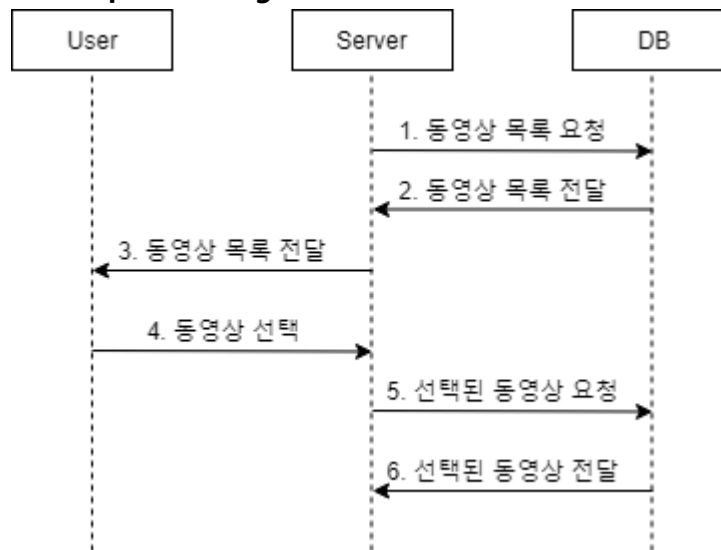
### C.1 Sequence Diagram of Sign In



### C.2 Sequence Diagram of Blurring

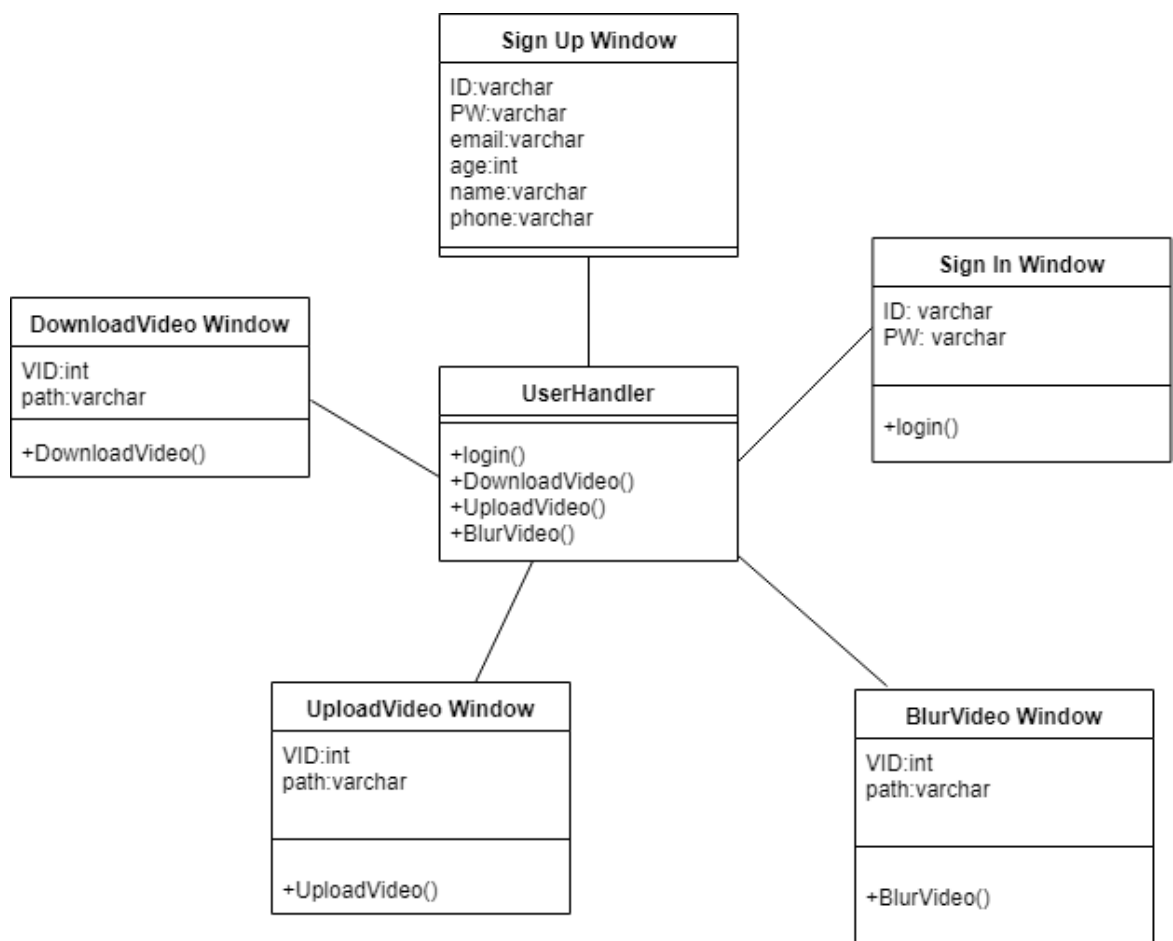


### C.3 Sequence Diagram of Download



## 6.4 Structural Models

### A. User System class Diagram

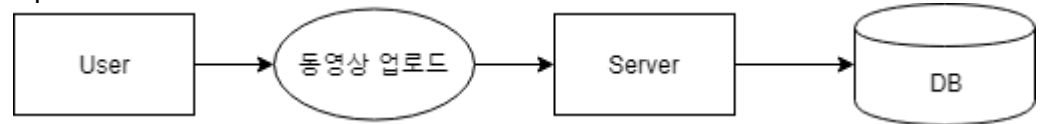


## 6.5 Behavioral Models

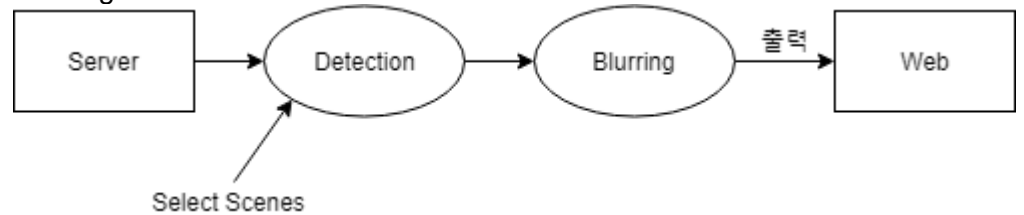
### A. Data-Driven Modeling

#### A.1 User

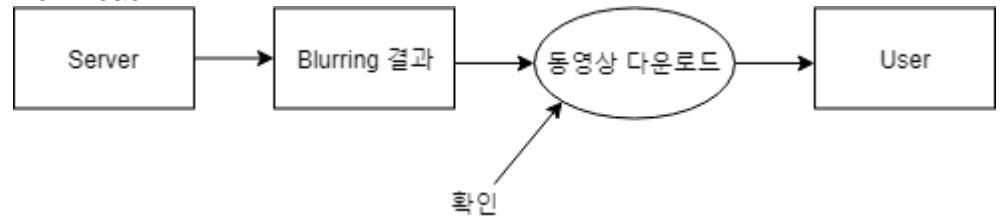
- Upload



- Blurring

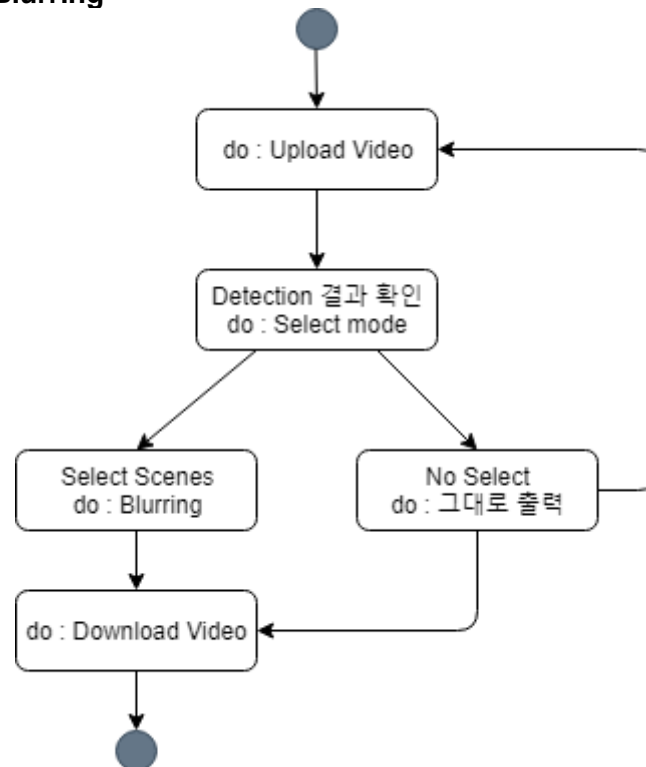


- Download



### B. Event-Driven Modeling

#### B.1 Blurring





## 7. System Evolution

### 7.1 Objective

System Evolution 시스템에 일어날 수 있는 예상되는 변화들, 가령, 하드웨어의 발전, 사용자의 Needs 변화 등에 대해 설명한다. 이 장은 차후에 발생할 시스템의 설계 변경을 피할 수 있도록 하기 때문에 시스템 설계자에게 도움이 될 것이다. 이를 통해, 변화에 잘 적응하는 시스템을 개발할 수 있다.

### 7.2. Limitation and Assumption

웹사이트 상에서 동영상을 변환하는 서비스이기 때문에 동시 접속하는 이용자들이 많아질 경우 실행 속도가 지연될 우려가 있다. 또한 유저들의 이용 로그를 저장해야 하는데, 사용 빈도가 높아져 데이터가 쌓이게 되면, 물리적인 용량에 부하가 걸릴 가능성이 있다.

### 7.3. Evolutions of User Requirement

#### A. Community service

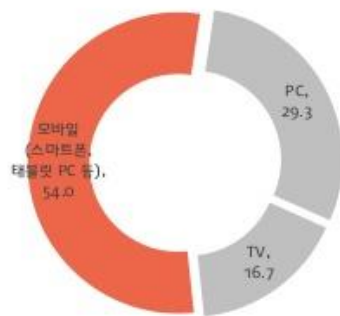
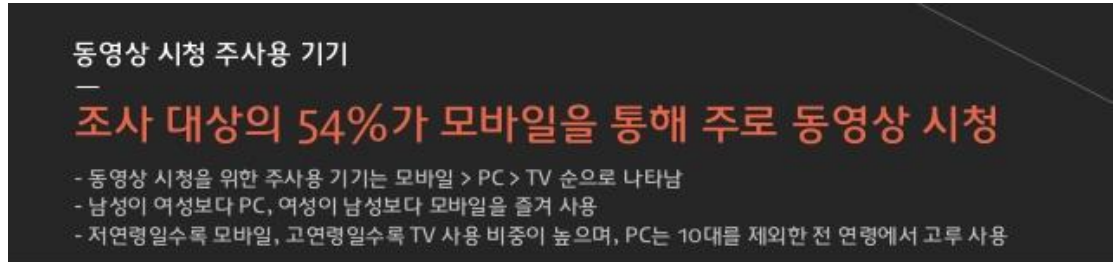
현재 구상한 웹사이트에서는 영상의 업로드와 변환을 주기능으로 한다. 하지만 동영상 변환이 아닌 건전한 영상을 시청하길 원하는 사용자들에게 이 웹사이트가 일종의 휴식처와 같은 역할을 해줄 수 있다. 가령 영상 변환 시 마다 건전 지수를 산출하여 웹사이트의 첫 화면에 건전 지수가 높은 영상들을 나열한다거나, 각 영상들에 대하여 코멘트를 남길 수 있는 서비스를 제공함으로써 유저들 간의 소통을 이룰 수 있다.

#### B. Multi-process

현재 서비스는 단일 영상의 변환을 고려하고 있다. 하지만 플랫폼 제공자들의 경우 한 번에 수십개 이상의 영상 변환이 가능해야 효율성이 보장되기 때문에, 그러한 요구에 맞추어 병행 처리 서비스를 제공해야한다. 이를 일반 사용자들에게 모두 제공하기에는 시스템 부하가 발생할 수 있으므로 구독형 서비스를 통해서 업그레이드된 성능을 제공하여야 한다.

## 7.4 Evolutions of Environment

사용자들의 영상 시청 기기가 pc 에서 모바일로 빠르게 옮겨가고 있고 유튜브의 경우에도 모바일 시청 시간이 압도적으로 높다. 따라서 영상 변환을 위해 웹 플랫폼을 사용하는 것은 다수의 모바일 시청자들의 편의와는 동떨어질 수 있다.



● 동영상 콘텐츠 시청 주사용 기기 (단위: %)

	전체	성별		연령					
		남성	여성	10대	20대	30대	40대	50대	
Base for %	(1000)	(511)	(489)	(102)	(192)	(215)	(251)	(240)	
모바일(스마트폰, 태블릿 PC 등)	54.0	51.1	57.1	72.5	64.1	51.6	50.2	44.2	
PC	29.3	34.1	24.3	18.6	26.0	34.0	30.7	30.8	
TV	16.7	14.9	18.6	8.8	9.9	14.4	19.1	25.0	
계	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	

Figure 5 영상 콘텐츠 주사용 기기 조사 결과

시청자들은 연결된 사용 경험을 중요시하므로 모바일로 영상을 시청하다가 건전한 영상 시청만을 위해서 PC 를 사용하기에는 굉장히 번거로울 수 있다. 따라서 안드로이드나 IOS 플랫폼 상에서도 가동할 수 있도록 어플리케이션 대응을 하여야 한다면 많은 이용자 수를 확보할 수 있고 서비스가 탄력 받을 수 있다.

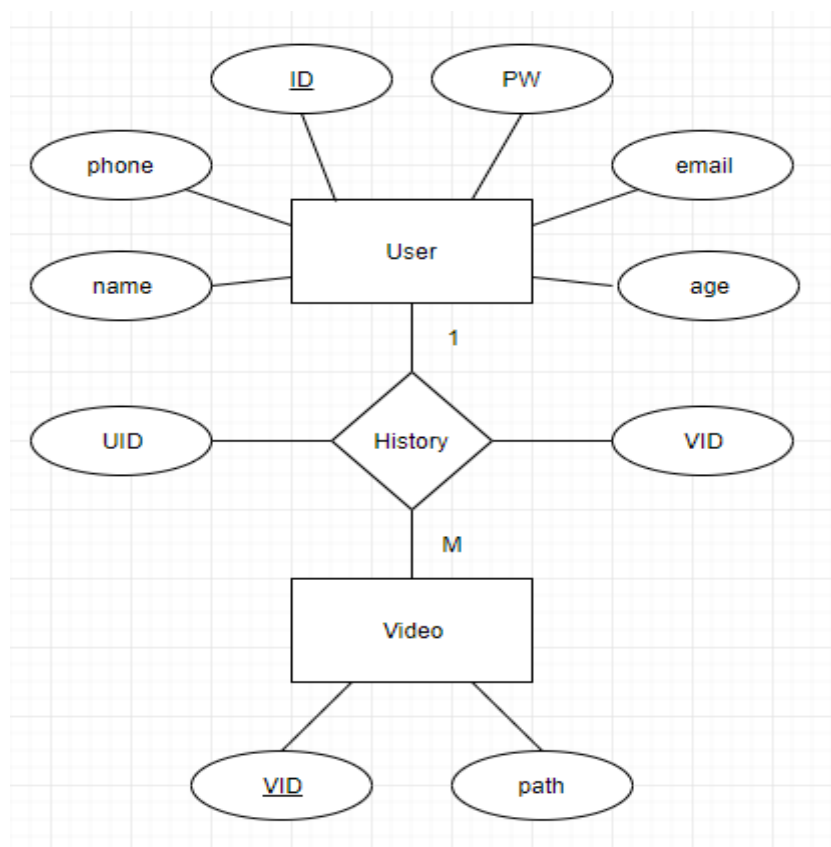
## 8. Appendices

### 8.1 Objective

Appendices 에서는 개발하려는 시스템에 대한 더 자세하고 구체적인 정보들을 제공한다. 하드웨어, 데이터베이스 등에 대한 것은 물론, 시스템 사용에 적합한 혹은 최소의 설정, 데이터베이스 상에서 결정되는 데이터의 관계 혹은 데이터의 구조에 대해서 설명한다.

### 8.2 DATABASE REQUIREMENTS

해당 장에서는 Database 의 필요 조건을 기술한다. 이하 설명하는 내용들은 MySQL 을 이용하여 저장되는 데이터베이스에서 테이블로 입력되어야 한다. 기본적으로 해당 서비스는, 유저의 개인정보를 담고있는 User Entity 와, 각 변환된 영상의 저장 경로를 담고있는 Video Entity, 그리고 각 유저별로 변환 내역을 담고있는 History Relation 으로 구성이 되어있다. 먼저 ER Diagram 을 이용해 표현하면 다음과 같다.



User Entity 의 경우, 해당 유저가 서비스의 Sign up 기능을 이용하여 자신의 ID, PW 를 입력하고 추가적으로 개인정보에 대한 내용을 담고있는 Entity 로, 개인화를 위해 존재한다. ID 에는 중복이 존재하지 않기 때문에 primary key 로 설정하였다.

Video Entity 의 경우, 각 비디오별로 ID(VID)를 부여하고, 이는 Video 가 추가될 때마다 1 씩 증가하는 Auto Increment 번호를 부여한다. 역시 고유한 번호이기 때문에 primary key 로 설정한다. 또한, 변환된 비디오는 서버에 저장되기 때문에, 그 Video 의 저장경로를 테이블에 추가적으로 저장한다.

이 두가지 Entity 는 각 유저의 Video 변환 기록을 보여주기 위해 History 라는 관계(Relation)으로 이어지게 되고, 이 History 의 경우 User Entity 의 ID 를 참조하고, Video Entity 의 VID 를 참조하여 최종적으로 어떤 유저가 어떠한 영상의 기록을 가지고 있는지를 보여주게 된다.

#### A. User Table

	종류	길이/값	NULL
<u>ID</u>	varchar	20	N
PW	varchar	20	N
email	varchar	30	N
age	int	10	N
name	varchar	20	N
phone	varchar	20	N

#### B. Video Table

	종류	길이/값	NULL
<u>VID</u>	int	10	N
path	varchar	50	N

### C. History Table

	종류	길이/값	NULL
UID	varchar	20	N
VID	int	10	N

## 8.3 USER-SYSTEM REQUIREMENTS

해당 서비스는 기본적으로 HTML5 를 기반으로, Mysql 과 PHP 및 Javascript 이용하여 웹서비스를 제작한다. 따라서 사용자는 HTML5 를 지원하는 웹 브라우저를 통해 해당 서비스를 이용할 수 있다.

Browser	
Minimum	Any browser can support HTML5
Recommended	Google Chrome, Mozilla Firefox (Internet Explorer is not recommended)