

---

---

# **Deepsum Software Architecture Document**

**Version <1.0>**

## Revision History

Date	Version	Description	Author
09/Dec/20	0.1	Initial draft	Nam, Seunggho
22/Dec/20	1.0	Finalize	Nam, Seunggho

# Table of Contents

1. 개요	4
2. 프로젝트 배경	4
3. 아키텍처 드라이버	5
4. 정적 뷰	5
4.1 개요	5
4.2 Entity responsibility catalog for static view	5
4.3 Relationship responsibility catalog	7
4.4 Design rationale	7
5. 동적 뷰	8
5.1 개요	8
5.2 Entity responsibility catalog for dynamic view	9
5.3 Relationship responsibility catalog	9
5.4 Design rationale	10
5.5 시퀀스 다이어그램	10
6. 물리적 뷰	11
6.1 개요	11
6.2 Entity responsibility catalog for physical view	11
6.3 Relationship responsibility catalog	12
6.4 Design rationale	12
7. 분석	12
7.1 보안	12
7.2 성능	12

# Software Architecture Document

## 1. 개요

이 문서는 딥섬의 전체적인 아키텍처 개요를 설명합니다. 딥섬은 아이펠의 교과과정의 일부인 해커톤 프로젝트입니다. 여기서는 여러개의 아키텍처 뷰를 통해 시스템을 다양한 측면에서 묘사하며, 중요한 아키텍처 결정 사항이 어떻게 결정되었는지 제시합니다.

이 문서에서는 프로젝트 배경을 먼저 설명하고 시스템의 세가지 뷰 (정적 뷰, 동적 뷰, 물리적 뷰)를 제시합니다. 시스템의 아키텍처는 향후 세부 디자인에서 사용될 수 있습니다. 이 문서의 나머지 부분은 주로 시스템 뷰 설명과 디자인 근거를 제시하며 아키텍처 평가와 분석으로 마칩니다.

## 2. 프로젝트 배경

이 프로젝트의 목적은 사용자에게 원하는 임의의 한글 문서를 무료로 요약할 수 있게 해주는 웹페이지를 제공하는 것입니다. 현재 뉴스를 요약하는 기능이 일부 뉴스 사이트에 있지만, 만족스러운 결과를 제공하고 있지 못하며, 제약없이 임의의 글 (한글 뉴스)을 요약해 주는 서비스는 없습니다.

우리의 제품은 웹사이트 형태가 될 것이며, 사용자들은 웹 브라우저를 통해 웹사이트에 접근하여 원하는 임의의 텍스트를 텍스트 박스에 입력하고, 요약 요청을 하면 딥섬 서버에서 요약한 결과를 돌려받을 수 있게 됩니다. 아래 그림 1 은 사용자가 어떻게 딥섬 시스템과 상호작용하는지 보여줍니다.



Figure 1 Project Context Diagram

### 3. 아키텍처 드라이버

이 프로젝트의 아키텍처 드라이버는 소프트웨어 요구사항 명세서에 기술되어 있습니다. 이 문서 마지막에 있는 링크에서 다운로드 받을 수 있습니다.

## 4. 정적 뷰

### 4.1 개요

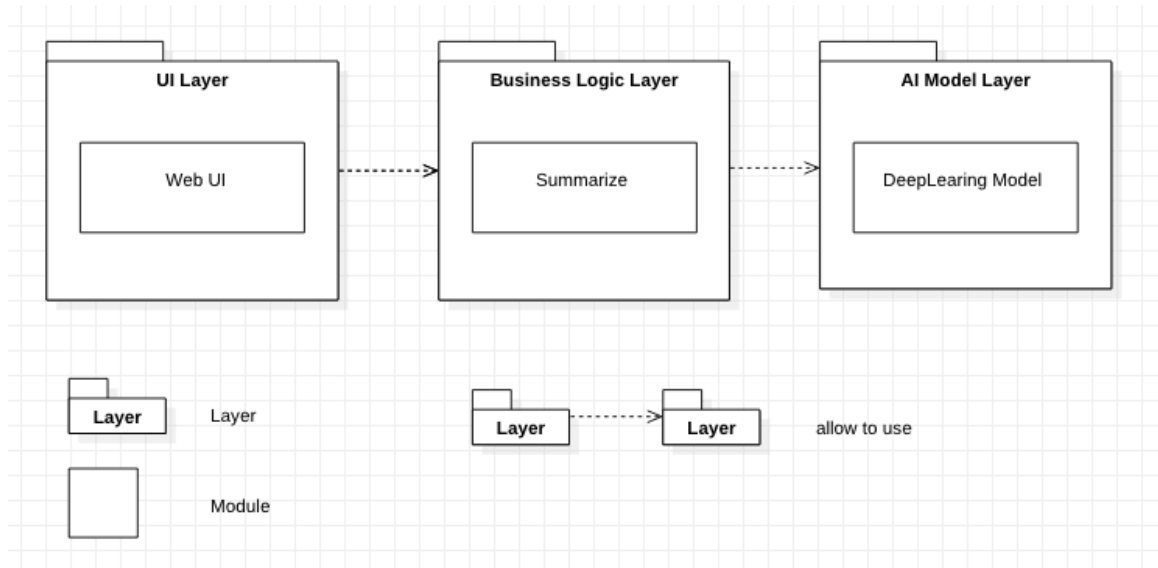


Figure 2 Static View Diagram


이 다이어그램은 딥섬 시스템의 정적인 뷰를 보여줍니다. UI Layer 는 웹 유저 인터페이스 화면 모듈을 포함하고 있습니다. Business Logic Layer 는 시스템에서 비즈니스 로직은 처리하는 모듈을 가지고 있습니다. AI Model Layer 는 딥러닝 모델 모듈을 가지고 있습니다. Web UI 모듈은 웹 사용자 인터페이스를 위해 필요한 페이지와 클래스 파일들을 가지고 있습니다. 이 모듈은 Business Logic Layer 의 모듈을 사용합니다. Summarize 모듈은 사용자가 입력한 텍스트를 요약하는데 필요한 클래스들을 가지고 있습니다. DeepLearning Model 모듈은 텍스트 요약 데이터를 입력 받아 전처리를 하고 딥러닝을 수행하고 텍스트 요약을 수행하는 클래스들을 가지고 있습니다. Business Logic Layer 의 Summarize 모듈은 DeepLearning Model 모듈을 사용합니다.

### 4.2 Entity responsibility catalog for static view

Static view diagram	Perspective: static
Element	Responsibility
UI Layer	<ul style="list-style-type: none"> <li>이 레이어는 사용자 인터페이스 표현과 관계된 모듈을 가지고 있는 파티션입니다.</li> </ul>
Business	<ul style="list-style-type: none"> <li>이 레이어는 비즈니스 로직에 관한 모듈을 가지고 있는</li> </ul>

Logic Layer	파티션입니다.
AI Model Layer	<ul style="list-style-type: none"> <li>이 레이어는 딥러닝 모델을 구현하는 모듈을 가지고 있는 파티션입니다.</li> </ul>
Web UI Module	<ul style="list-style-type: none"> <li>이 모듈은 웹 유저 인터페이스와 연관된 웹 페이지와 클래스 파일을 가지고 있습니다.</li> <li>이 모듈의 웹 페이지들은 Business Logic Layer 의 클래스들을 사용할 것입니다.</li> </ul>
Summarize Module	<ul style="list-style-type: none"> <li>이 모듈은 텍스트 요약과 관련된 클래스들을 가지고 있습니다.</li> <li>이 모듈의 클래스들은 DeepLearning Model 모듈의 클래스들을 사용할 것입니다.</li> </ul>
DeepLearning Model Module	<ul style="list-style-type: none"> <li>이 모듈은 데이터 전처리 및 학습을 통해 요약을 수행하는 딥러닝 모델에 관련된 클래스를 가지고 있습니다.</li> </ul>

### 4.3 Relationship responsibility catalog

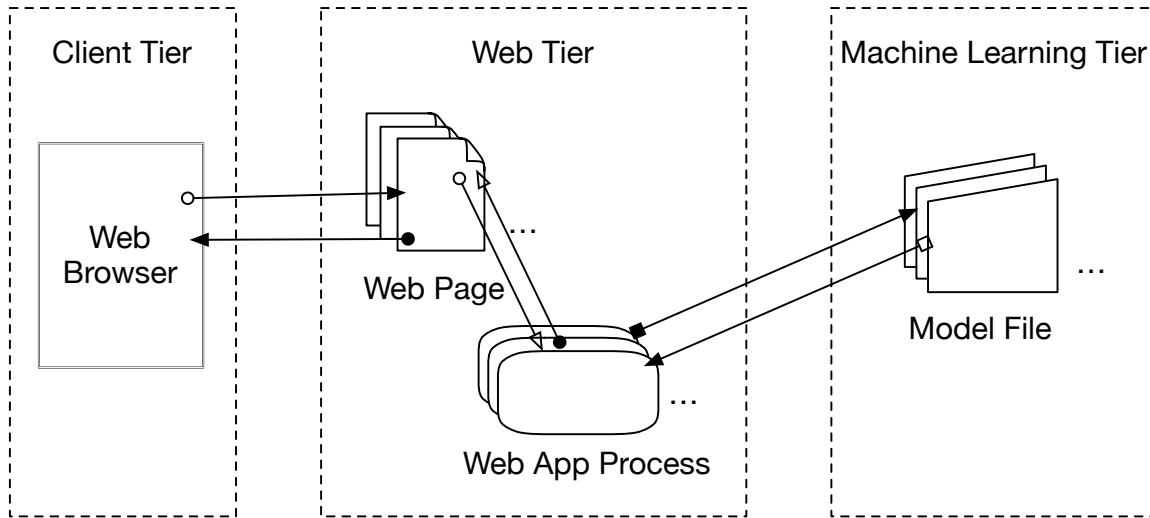
Static view diagram	Perspective: Static
Relationship	Responsibility
A  B	<ul style="list-style-type: none"> <li>이 관계는 레이어 사이에서 사용됩니다.</li> <li>이 관계는 A 레이어의 모듈이 B 레이어의 모듈을 사용할 수 있다는 것을 의미합니다.</li> </ul>

### 4.4 Design rationale

Static view diagram	Perspective: Static
<p>이 시스템은 웹 어플리케이션 형태를 가지고 있으므로, 사용자 인터페이스, 비즈니스 로직, 그리고 인공지능 모델 레이어로 구분하고 있다. 보통의 웹 어플리케이션이 이런 세단계의 레이어 (Model-View-Controller)로 구별하고 있으므로, 어색하지 않은 디자인이라고 생각한다. 이 시스템은 매우 단순하고 직관적인 기능을 수행하는 어플리케이션이 될 것이므로, 레이어마다 하나의 모듈을 가지는 것으로 충분할 것으로 생각된다. 세부 클래스는 여기서는 생략한다.</p>	

## 5. 동적 뷰

### 5.1 개요



#### Legend

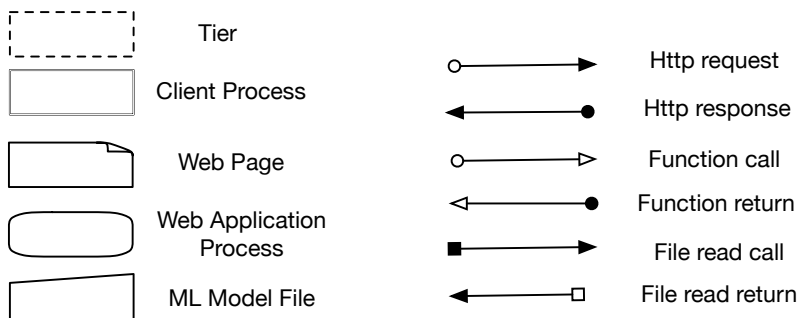


Figure 3 Dynamic View Diagram

이 다이어그램은 시스템의 동적뷰를 보여주며, 각각의 티어에 있는 컴포넌트들이 어떻게 서로 상호작용하고 있는지를 보여줍니다. 클라이언트 티어는 클라이언트 어플리케이션이 실행되는 영역을 나타냅니다. 웹 티어는 웹 어플리케이션 프로세스가 수행되는 영역을 나타냅니다. 머신러닝 티어는 머신러닝 모델 파일이 위치하는 영역을 나타냅니다. 웹 브라우저 어플리케이션은 웹 티어의 웹페이지와 **http request** 와 **http response** 프로토콜을 통해 상호작용합니다. 웹 어플리케이션 프로세스는 웹페이지에서 들어오는 **function call** 을 처리하고 결과값을 **return** 합니다. 그리고 비즈니스 로직을 구현하기 위해 **Machine Learning Tier** 의 모델 파일을 호출하고 결과값을 받아 처리합니다.



## 5.2 Entity responsibility catalog for dynamic view

Dynamic view diagram	Perspective: Dynamic
Element	Responsibility
Client Tier	<ul style="list-style-type: none"> <li>이 티어는 클라이언트 어플리케이션들이 모여있는 그룹입니다.</li> </ul>
Web Tier	<ul style="list-style-type: none"> <li>이 티어는 웹 페이지와 웹 어플리케이션이 모여 있는 그룹입니다.</li> </ul>
Machine Learning Tier	<ul style="list-style-type: none"> <li>이 그룹은 딥러닝 모델 파일이 모여있는 그룹입니다.</li> </ul>
Web Browser	<ul style="list-style-type: none"> <li>이 프로세스는 사용자와 웹 티어의 웹페이지나 웹 어플리케이션 프로세스와 상호작용하는 클라이언트 어플리케이션입니다.</li> <li>웹 페이지와 http request 와 http response 로 커뮤니케이션합니다.</li> </ul>
Web Pages	<ul style="list-style-type: none"> <li>웹 표현을 만들어내는 웹 페이지 파일입니다.</li> <li>이 파일은 웹 티어의 웹 어플리케이션과 function call / return 으로 커뮤니케이션합니다.</li> </ul>
Web Application Process	<ul style="list-style-type: none"> <li>이 컴포넌트는 웹페이지의 요청을 처리하여 결과값을 반환합니다. Machine Learning 티어의 모델 파일과 File Read Call / Return 으로 커뮤니케이션합니다.</li> </ul>
Model File	<ul style="list-style-type: none"> <li>이 파일들은 Web Application 의 요청을 처리하기 위한 머신 러닝 모델 데이터를 포함하고 있습니다.</li> </ul>

## 5.3 Relationship responsibility catalog

Dynamic view diagram	Perspective: Dynamic
Relationship	Responsibility
A ○ → B	<ul style="list-style-type: none"> <li>이 관계는 http request 를 나타냅니다.</li> <li>이 관계는 클라이언트 웹 브라우저 어플리케이션 프로세스(A)가 웹 티어의 웹 페이지 파일 (B)에 http request 를 보내는 것을 나타냅니다.</li> </ul>
A ● → B	<ul style="list-style-type: none"> <li>이 관계는 http response 를 나타냅니다.</li> <li>이 관계는 웹 페이지 파일 (A)가 웹 브라우저 프로세스 (B)로 http response 를 보내는 관계를 나타냅니다.</li> </ul>
A ○ → B	<ul style="list-style-type: none"> <li>이 관계는 함수 호출 관계를 나타냅니다.</li> <li>이 관계는 웹 페이지 (A)가 웹 어플리케이션 프로세스 (B)의 함수를 호출하는 관계를 나타냅니다.</li> </ul>
A ● → B	<ul style="list-style-type: none"> <li>이 관계는 함수 호출 리턴 관계를 나타냅니다.</li> <li>이 관계는 웹 어플리케이션 프로세스 (A)가 웹페이지(B)에게 함수 결과를 반환하는 관계를 나타냅니다.</li> </ul>

	의미합니다.
A ■ → B	<ul style="list-style-type: none"> <li>이 관계는 파일 읽기 관계를 의미합니다.</li> <li>이 관계에서 웹 어플리케이션 프로세스(A)는 모델 파일(B) 읽기 호출 합니다.</li> </ul>
A □ → B	<ul style="list-style-type: none"> <li>이 관계는 파일 읽기 호출 반환 관계를 의미합니다.</li> <li>이 관계에서 모델 파일 (A)은 웹 어플리케이션 프로세스 (B)에게 파일 읽기 호출 결과를 반환합니다.</li> </ul>

#### 5.4 Design rationale

Dynamic view diagram	Perspective: Dynamic
<ul style="list-style-type: none"> <li>딥섬 시스템은 비교적 간단한 웹 시스템이므로, Flask 프레임워크를 이용하여 구현될 예정입니다.</li> <li>클라이언트 tier는 웹 브라우저 어플리케이션들이 위치하는 그룹입니다. 웹 tier는 웹 어플리케이션과 관련된 컴포넌트가 위치하는 그룹입니다. 간단한 시스템이므로 UI 와 Business Logic 을 별도의 tier로 구분하지 않았습니다.</li> <li>머신러닝 tier는 딥러닝 모델 파일이 위치하는 그룹입니다.</li> </ul>	

#### 5.5 시퀀스 다이어그램

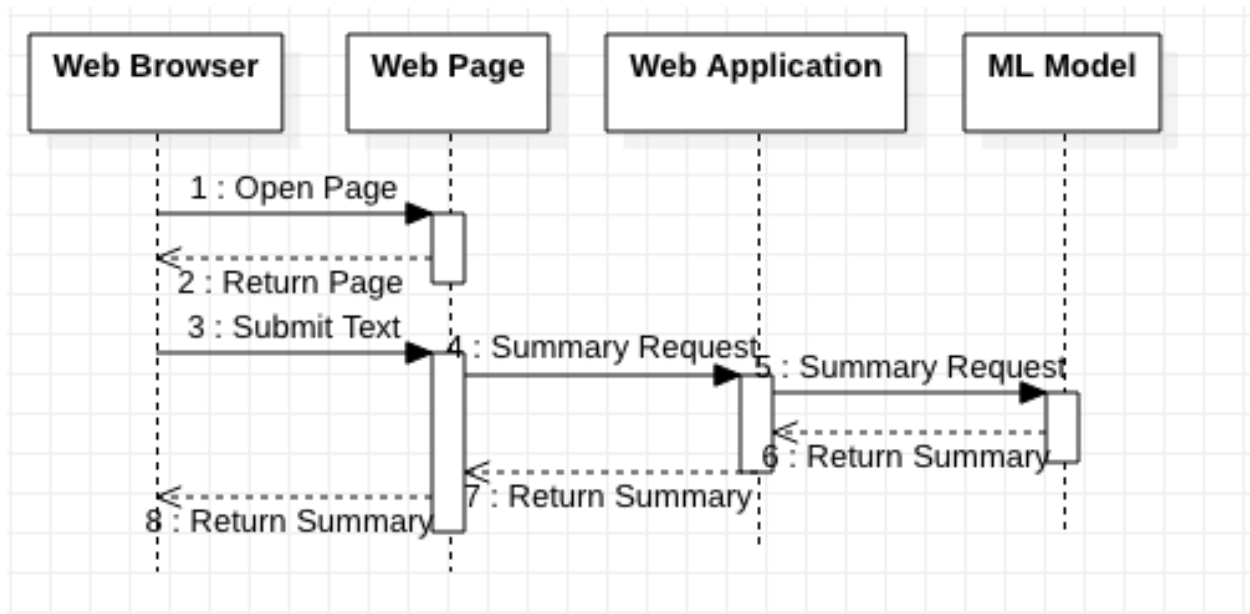


Figure 4 Sequence Diagram – Summary Request

딥섬 사용자는 웹 브라우저를 통해 딥섬 웹 페이지에 접근하여 웹페이지를 확인하고, 자신이 요약하고자 하는 텍스트를 입력한다. 그러면, 웹페이지는 연결된 웹 어플리케이션의 요약 요청 함수로 텍스트를 전달한다. 웹 어플리케이션은 다시 머신러닝 모델 파일에 정의된 요약 함수로 텍스트를 전달하고 요약 결과를 반환받는다. 받은 결과를 다시 웹 페이지로 표시하도록 돌려준다.

## 6. 물리적 뷰

### 6.1 개요

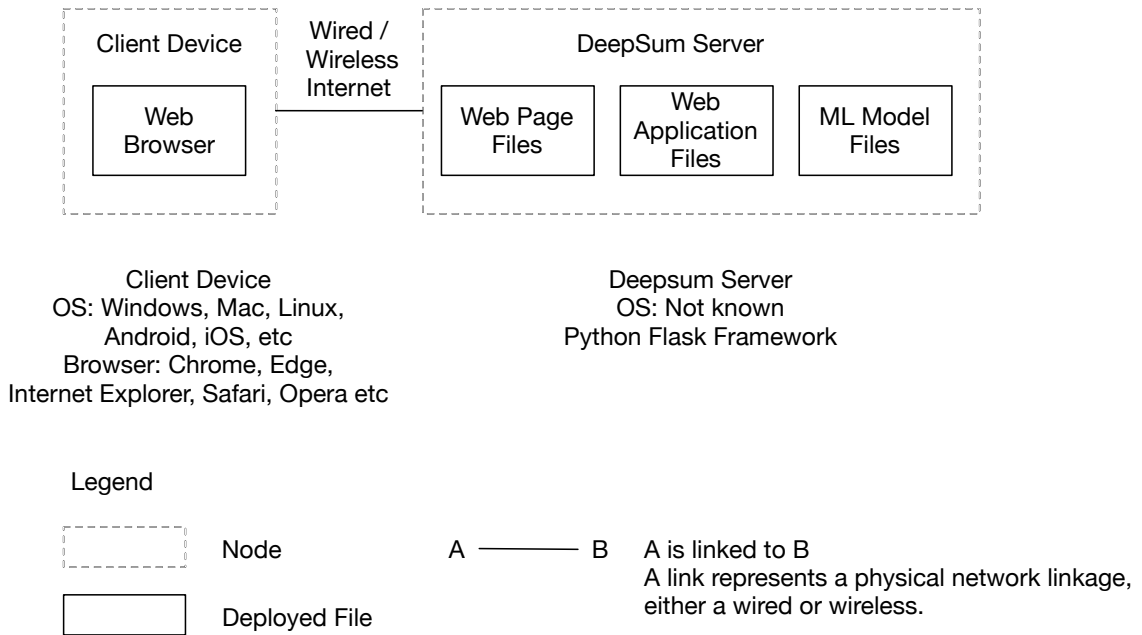


Figure 5 Physical View Diagram

이 다이어그램은 시스템의 물리적 뷰입니다. 클라이언트 기기는 인터넷을 통해 딥섬 서버로 접근합니다. 클라이언트 기기는 웹 브라우저를 가지고 있어야 하며, 크롬, 인터넷 익스플로러, 엣지, 사파리, 오페라 등의 브라우저로 접근할 수 있습니다. 서버는 파이썬으로 어플리케이션을 구동하며, 플라스크를 사용하여 웹 서비스를 수행할 것입니다.

### 6.2 Entity responsibility catalog for physical view

Physical view diagram	Perspective: Physical
Element	Responsibility
Client Device	<ul style="list-style-type: none"> <li>웹 브라우저 실행이 가능한 클라이언트 기기를 의미합니다. 모바일 기기 혹은 랩탑, 데스크탑 기기가 될 수 있으며, 이를 통해 사용자는 딥섬 서버에 접근할 수 있습니다.</li> </ul>
Deepsum Server	<ul style="list-style-type: none"> <li>이 서버는 파이썬과 플라스크 웹 서버를 실행시킵니다.</li> <li>이 서버에는 웹 페이지 파일과, 웹 어플리케이션 파일, 딥러닝 모델 파일이 위치할 것입니다.</li> </ul>

### 6.3 Relationship responsibility catalog

Physical view diagram	Perspective: Physical
Relationship	Responsibility
A — B	<ul style="list-style-type: none"><li>이 관계는 클라이언트와 서버 사이의 연결 관계를 의미합니다. 무선 혹은 유선으로 인터넷에 연결된 관계를 의미합니다.</li></ul>

### 6.4 Design rationale

Physical view diagram	Perspective: Physical view
<ul style="list-style-type: none"><li>딥섬 서버는 간단한 웹 서버 시스템으로 하나의 서버 머신에 배포될 것입니다. 이 서버는 클라우드에 위치할 수 있으며, 향후 사용자 증가추이에 따라 여러 개의 노드로 확장될 수 있습니다.</li></ul>	

## 7. 분석

### 7.1 보안

이 시스템은 시스템 무결성을 유지해야 하며 해킹 등의 공격으로부터 보호할 수 있는 장치를 구현해야 합니다. 현재 어떤 플랫폼에서 서비스를 제공할 지 명확하지 않으므로, 추후 플랫폼 결정이후 구체적인 방안을 확정할 것입니다.

### 7.2 성능

이 시스템은 사용자에게 요약 결과를 10 초이내에 보여줄 것을 목표로 합니다. 이 목표를 이루기 위해 딥러닝 모델 완성 단계부터 성능을 테스트할 것이며, 혹시라도 사용자가 불편함을 느낄 경우를 대비하여 진행 상황을 확인할 수 있는 애니메이션을 추가할 예정입니다.