

FaceFit 시스템 아키텍처

최종 프로젝트 2 Team

SK Family Networks 7기

2025-03-31

목차

1. 개요.....	4
1.1 문서 목적.....	4
1.2 대상 독자.....	4
1.3 시스템 개요.....	4
1.4 주요 기능.....	4
2. 아키텍처 개요.....	5
2.1 시스템 구성도.....	5
2.2 기술 스택.....	6
2.2.1 Web Server.....	6
2.2.2 WAS.....	6
4. 시스템 아키텍처 상세 설계.....	8
4.1 클라이언트-서버 인터랙션.....	8
4.2 API 설계 개요.....	8
4.3 데이터 흐름 및 처리 과정.....	10
4.4 데이터베이스 설계.....	11
5. 네트워크 및 배포 구조.....	12
5.1 서버 인프라 개요.....	12
5.2 네트워크 토폴로지.....	12
5.3 배포 및 운영 방식.....	12
6. 보안 및 인증.....	13
6.1 사용자 인증 및 권한 관리.....	13
6.2 API 보안 정책.....	13
6.3 데이터 보호 및 개인정보 보안.....	13
7. 부록.....	14
7.1 용어 정의.....	14

7.2 참고 자료.....	14
----------------	----

1. 개요

1.1 문서 목적

본 문서는 FaceFit 서비스의 시스템 아키텍처를 정의하고, 전체적인 구성 요소와 데이터 흐름을 설명합니다. 개발자, 운영자 및 이해 관계자들이 시스템의 구조를 이해하고 유지 보수할 수 있도록 합니다.

1.2 대상 독자

- 개발자
- 시스템 설계자
- DevOps 엔지니어
- 프로젝트 관리자

1.3 시스템 개요

FaceFit은 사용자의 얼굴 사진과 선택한 안경 모델을 결합하여 이미지를 합성하고 제공하는 서비스입니다. FastAPI를 기반으로 하는 RESTful API로 구성되어 있습니다.

1.4 주요 기능

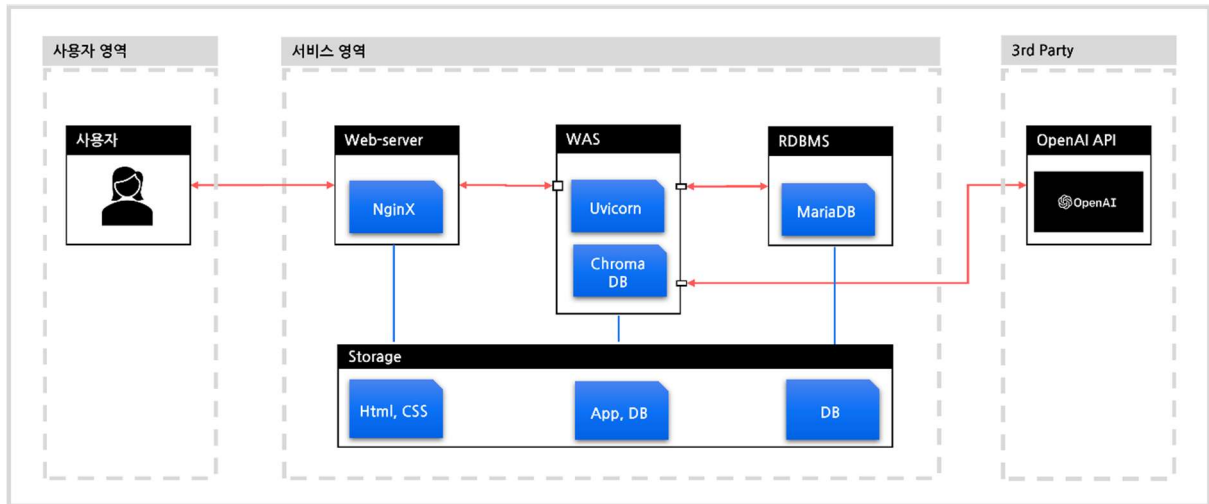
- 사용자 UI 제공
- 챗봇을 통한 안경 추천
- 사용자 얼굴형 분석
- 안경 착용 이미지 생성
- 파일 관리 및 저장

2. 아키텍처 개요

2.1 시스템 구성도

FaceFit 시스템은 다음과 같은 주요 요소로 구성됩니다.

시스템 구성도



시스템 이름	시스템 구성 요소 설명	기 타
Web Server	서비스에 필요한 HTML, JS, CSS 및 PNG 등등 정적인 요소를 사용자에게 제공한다.	NginX
WAS	API Server 기능을 수행한다. 다음과 같은 업무를 수행한다. <ul style="list-style-type: none">• 사용자의 메시지를 분석하여 어떤 명령을 요청하고 있는지 파악하고 적절한 응답을 작성한다.• 사용자의 메시지를 분석하기 위하여 1차로 VectorDB의 유사도 검색을 수행하고, 필요한 경우 2차로 이전 대화 내용과 현재 요청을 OpenAI API를 이용하여 어떤 업무를 요청하는지 판별한다.• 사용자가 업로드한 사진을 분석하여 얼굴형을 판별한다.	Uvicorn, ChromaDB, Python Application

	<ul style="list-style-type: none"> • 얼굴형별 안경목록을 RDBMS에서 검색하여 제공한다. • 선택한 안경을 얼굴 사진에 합성한다. 	
RDBMS	안경 정보, 얼굴형 정보, 업로드한 사진 정보 등을 제공한다	MariaDB
STORAGE	<p>Web Server에서 사용하는 HTML, JS, CSS, Image 파일 등 정보를 저장한다.</p> <p>WAS에서 사용자가 업로드한 Image, 합성한 사진 등을 저장한다.</p> <p>Vector DB를 와 RDMBS 의 DB 파일을 저장한다.</p>	<p>Amazone S3,</p> <p>Docker</p> <p>Volume(개발)</p>
OpenAI API	<ul style="list-style-type: none"> • 사용자 대화에서 어떤 서비스를 요구하는지 판별한다. • 분석된 얼굴형 정보와 데이터를 이용하여 자연스러운 대화를 생성한다. 	

2.2 기술 스택

2.2.1 Web Server

- **프로그래밍 언어:** HTML5, CSS3, Javascript
- Javascript Library: JQuery, Face Mesh
- **웹 서버:** NginX
- **운영체제:** Rocky Linux

2.2.2 WAS

- **개발언어:** Python 3.X
- 프레임워크: FastAPI

- 데이터 처리: NumPy, Pandas
- 이미지 처리: OpenCV, Pillow
- 얼굴 인식 및 랜드마크 추출: Mediapipe
- 머신 러닝/딥러닝: TensorFlow, scikit-learn
- 추론 서버: TensorFlow
- 데이터베이스: MariaDB, ChromaDB
- ORM: Tortoise
- 웹서버: Uvicorn

4. 시스템 아키텍처 상세 설계

4.1 클라이언트-서버 인터랙션

- 클라이언트는 UI를 통해 API 서버와 통신
- API 서버는 필요한 데이터를 DB 및 외부 서비스에서 가져와 응답

4.2 API 설계 개요

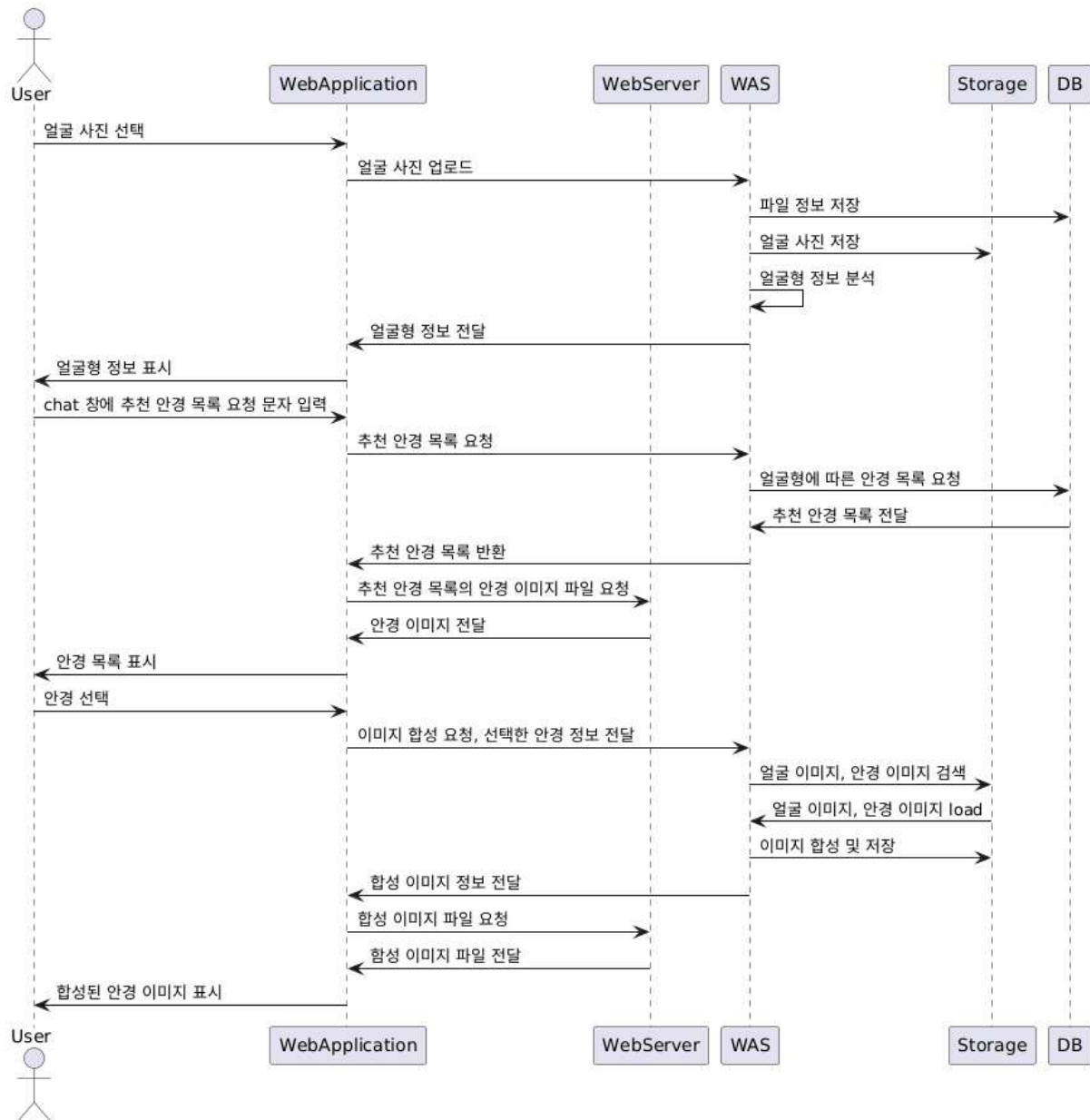
- RESTful API 방식으로 설계
- 주요 API 엔드포인트: 챗봇, 이미지 업로드, 얼굴형 분석, 이미지 생성, 안경목록
- API 목록

uri	method	name	응답	설명
/welcome/	get	초기화	ChatModel:T00	
/good-bye/	get	종료	ChatModel:TX X	
/chat/	post	대화형 명령	ChatModel	
/file/	post	사진 업로드		
/file/{file_id}/	get	사진 상세 정보		
/image/{path}/{stored_file_name}	get	사진 다운로드	binary	
/face/type?file_id={file_id}	get	사진 분석 요청	ChatModel	
/glasses/{glasses_id}	get	안경 정보 요청	ChatModel	
/webcam/state/on/	put	웹캠이 켜졌다	ResponseMessage.OK	웹캠이 켜졌다는 것을 알려준다
/webcam/state/off/	put	웹캠이 꺼졌다	ResponseMessage.OK	

- Chat 명령

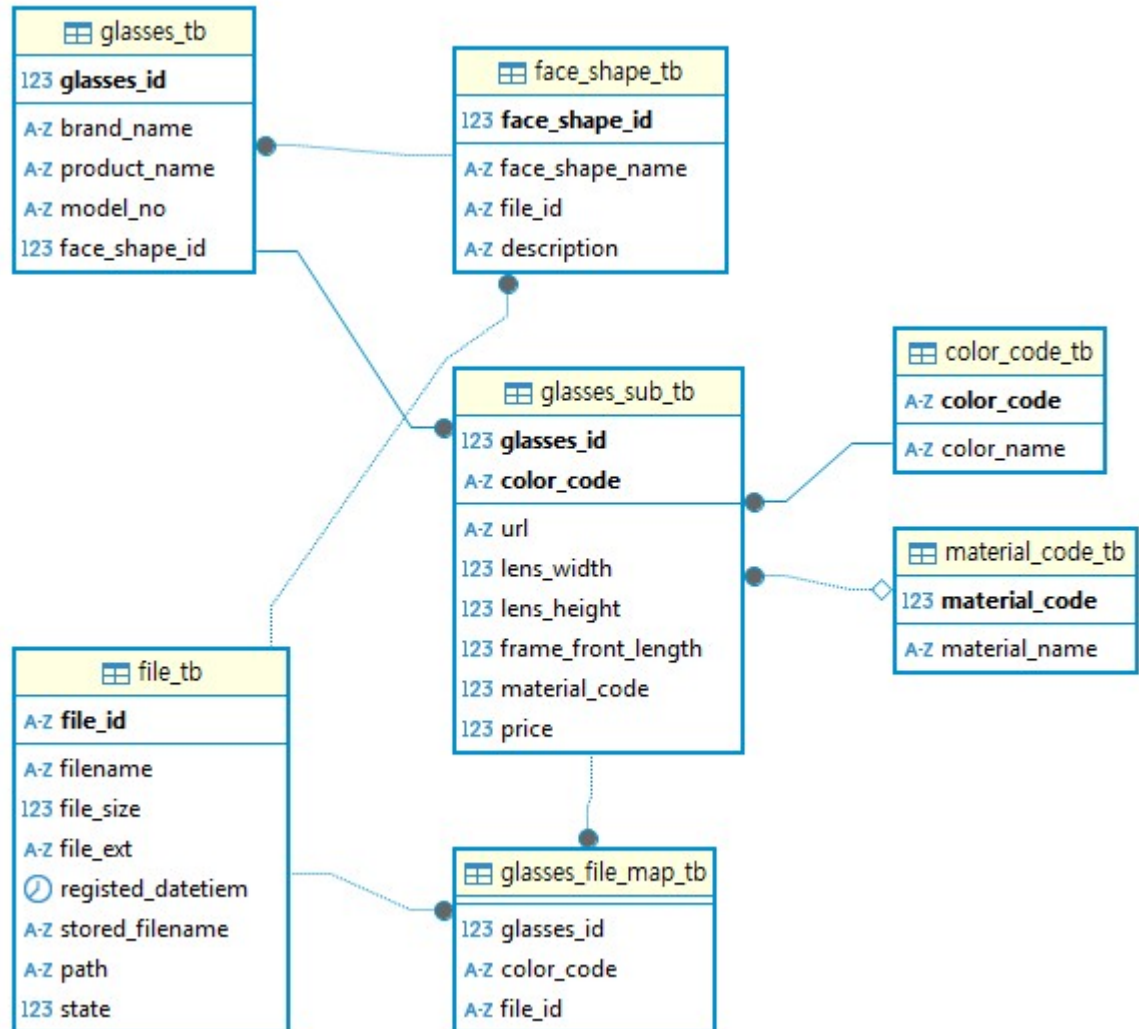
task_type	name	task_group
T00	초기화	초기화
T01	사진 캡처	얼굴분석
T02	사진 업로드	얼굴분석
T03	얼굴형 목록 요청	얼굴분석
T04	얼굴형 분석 요청- 파일	얼굴분석
T05	얼굴형 분석 요청- 대화	얼굴분석
T06	안경 목록 요청	피팅
T07	안경 상세 정보 요청	피팅
T08	피팅 이미지 요청	피팅
T09	웹캠을 켜라	STATE
T10	웹캠을 꺼라	STATE

4.3 데이터 흐름 및 처리 과정



4.4 데이터베이스 설계

- MariaDB: 안경 모델 및 사용자 정보 저장



- ChromaDB: 벡터 기반 검색을 위한 데이터 저장

5. 네트워크 및 배포 구조

5.1 서버 인프라 개요

- 컨테이너 기반 아키텍처 (Docker 활용)
- 주요 서버: Web Server, WAS, DB Server

5.2 네트워크 토폴로지

- 내부 네트워크를 통해 API 서버와 DB 연동
- 외부 접속은 NginX를 통해 관리

5.3 배포 및 운영 방식

- Docker를 이용한 컨테이너 기반 배포
- AWS 클라우드 환경에서 운영 예정

6. 보안 및 인증

6.1 사용자 인증 및 권한 관리

- 공개 서비스로 인증은 업습니다.
- Session을 이용하여 사용자별 상태 데이터 관리

6.2 API 보안 정책

- HTTPS 사용하여 통신 암호화

6.3 데이터 보호 및 개인정보 보안

- 개인을 구별할 수 있는 데이터를 DB에 저장하지 않습니다.
- 개인의 사진은 서비스 이용 후 최대 12 시간 이후에는 자동 삭제됩니다.

7. 부록

7.1 용어 정의

- **FastAPI**: 고성능 비동기 API 프레임워크
- **Uvicorn**: ASGI 서버 구현체
- **MariaDB**: 관계형 데이터베이스 시스템
- **ChromaDB**: 벡터 데이터베이스

7.2 참고 자료

- 프로젝트 GitHub Repository
- 공식 문서 (FastAPI, Docker, MariaDB 등)
- 2Team-기술스택문서 (구글 프로젝트 공유 폴더)
- 2Team-데이터베이스설계서-20250328 (구글 프로젝트 공유 폴더)
- 2Team-시스템_구성도-20250328 (구글 프로젝트 공유 폴더)
- 2Team-코드_컨벤션 (구글 프로젝트 공유 폴더)