

캡스톤디자인 프로젝트 결과 보고서

인공지능(AI) 기반 얼굴인식 출석체크 시스템



2302548 문예림 | 2302506 안시현 | 2302568 윤유지

얼굴 Pizza 팀



한양여자대학교
HANYANG WOMEN'S UNIVERSITY

목차

I. 서론

- 1.1 사회적 필요성/ 연구 의미
- 1.2 연구 범위 및 대상

II. 기술 및 이론

- 2.2 기존 출석 시스템의 한계

III. 시스템 설계

- 3.1 기능 목표
- 3.2 Use Case 다이어그램

IV. 개발 환경 및 구현 기술

- 4.1 기술적 구현 방법
- 4.2 사용된 도구 및 환경

V. 분석 및 평가

- 5.1 시스템 아키텍처
- 5.2 데이터베이스 설계
- 5.3 UI/UX 설계

VI. 시스템 구현 결과

- 6.1 개발 진행 상황
- 6.2 주요 성과 및 결과

VII. 사용자 테스트

- 7.1 설문조사 결과
- 7.2 주요 결과

VIII. 개선사항

- 8.1 기능 보완 및 개선 사항

IX. 결론

- 9.1 요약 및 기대효과
- 9.2 연구의 한계 및 향후 발전 방향

I. 서론

1.1. 사회적 필요성/연구 의미

현재 학교, 학원, 기업 등에서 이루어지는 출석 관리 방식은 여전히 수동적이거나 QR코드, 카드 태깅 등 반자동화 형태로 이루어지고 있습니다. 이러한 방식은 관리 효율이 낮고, 대리출석이나 기록 누락 등의 문제가 발생하기 쉽습니다. 특히 대규모 강의나 공공기관의 경우 인력과 시간이 많이 소모되어 자동화된 출석 관리 시스템의 필요성이 지속적으로 제기되었습니다.

본 프로젝트는 딥러닝 기반 얼굴인식 기술을 활용하여, 사용자의 생체 정보(얼굴)를 기반으로 출석 여부를 실시간으로, 정확하게, 자동으로 기록 및 관리하는 시스템을 구축하는 것을 핵심 동기로 합니다. 이를 통해 출석 관리의 공정성 및 신뢰성을 획기적으로 향상시키고, 인공지능(AI) 기술을 기반으로 한 비접촉식 얼굴인식 출석 시스템을 개발하여 정확하고 신속한 출석 관리 환경을 구현하고자 합니다. 또한, 코로나19 팬데믹 이후 급증한 비대면·비접촉 기술에 대한 사회적 수요에 부응하는 안전하고 위생적인 대안을 제시하고자 합니다.

1.2. 연구 범위 및 대상

본 프로젝트는 인공지능(AI) 기반 얼굴인식 출석체크 시스템을 개발하는 것을 목표로 합니다.

주요 시스템 구성 및 기능 범위

인증 및 인식 모듈: 카메라 장치를 통해 얼굴 영상을 입력받고, 얼굴 감지 및 인식 알고리즘(OpenCV 기반)을 통해 사용자를 식별하여 출석 여부를 자동 판별합니다.

관리 서버: 학사 관리용 백엔드 서버를 구축하고, REST API를 통해 강의 및 학생 관리 기능을 제공합니다.

보안: 얼굴 원본 이미지를 저장하지 않고, 특징값(embedding)을 저장하며, 접근 권한 및 인증 시스템을 적용합니다.

인터페이스: 관리자와 사용자가 모두 이용 가능한 웹 기반 인터페이스를 구축하고, 반응형 디자인으로 PC와 모바일 환경을 모두 지원합니다.

II. 기술 및 이론

2.1. 관련 기술 및 이론 소개

AI얼굴인식 기술 (Face Recognition Technology): 얼굴이라는 생체 정보를 활용하여 사용자 본인을 식별하는 딥러닝 기반 인증 기술을 적용하였습니다. OpenCv와 dlib 기반 얼굴 감지 및 특징 추출 모델을 통해 사용자의 얼굴을 저장하고, 이를 데이터베이스에 저장된 등록 정보와 매칭하여 개인 식별 및 인증을 수행합니다.

백엔드 프레임워크 및 API: 서버는 Spring Boot 프레임워크 기반으로 구축하였으며, MySQL 데이터베이스와 연동하여 사용자 정보 및 얼굴 특징 데이터를 관리합니다. 클라이언트(웹/모바일)와의 데이터 송수신은 RESTful API 방식으로 설계하여 확장성과 호환성을 고려하였습니다.

인증 및 인가 기능은 JWT 기반 로그인 API와 Spring Security를 적용하여 구현하였으며, 인가되지 않은 사용자의 API 요청을 효과적으로 차단합니다. 또한 사용자 비밀번호와 얼굴 특징은 데이터베이스에 저장함으로써 개인정보 및 생체정보 보호를 강화하였습니다.

모바일 UI 개발: 모바일 앱 UI는 React native 기반 Cross-Platform 구조로 개발되었으며, 송통 코드로 Android와 iOS 환경 모두에서 일관된 화면 렌더링과 사용자 경험(UX)을 제공하도록 설계했습니다.

2.2. 기존 출석 시스템의 한계

기존 출석 관리 방식은 다음과 같은 문제점을 내포하고 있습니다

낮은 관리 효율: 수동적 출석 방식은 관리 인력과 시간이 많이 소모됩니다.

신뢰성 저하: QR코드나 카드 태그 방식은 대리출석 문제가 발생하기 쉽고, 수기 방식은 기록 누락(휴면 에러)이 발생할 수 있습니다.

위생 문제: 지문인식기 등 접촉식 인증 방식은 전염병 확산 우려가 있는 상황에서 불편함이 커집니다.

III. 프로젝트 개요

3.1. 기능 목표

구분	주요 내용	상세 설명	적용 목적/기대 효과
출석 인증 방식	얼굴 인식 기반 비접촉식 출석 시스템	<ul style="list-style-type: none"> - 실시간 얼굴 감지 및 특징 비교 - GPS 기반 위치 반경 검증 - 출석 인정 시간 제한 검증 - 세 가지 요소를 모두 만족해야 출석 인정 	<ul style="list-style-type: none"> - 대리 출석 방지 - 비접촉식 자동 인증으로 편의성 향상 - 수업 관리 자동화 및 신뢰도 확보
AI 얼굴 인식 모듈	OpenCV + Face Recognition 모델	<ul style="list-style-type: none"> - 얼굴 특징 벡터 (embedding) 추출 후 DB 비교 - 조명, 마스크, 각도 변화 대응 - 유사도 기반 개인 식별 	<ul style="list-style-type: none"> - 정확도 향상 및 환경 적응성 확보 - 기존 출석 방식의 한계 해결
검증 로직	얼굴 + 위치(GPS) + 시간 조건 검증	<ul style="list-style-type: none"> - 위변조 방지 위한 다중 인증 - 특정 강의실 반경 내 인식만 허용 - 출석 인정 시간대 자동 판별 	<ul style="list-style-type: none"> - 공정한 출석 관리 - 대리 인증 및 허위 위치 인증 방지
백엔드 API 서버	Spring Boot 기반 REST API	<ul style="list-style-type: none"> - /api/attendance/check-in 출석 인증 처리 - 강의 정보 관리, 학생/관리자 계정 관리 - 출석 기록 조회 및 수정 기능 제공 	<ul style="list-style-type: none"> - 모바일/웹/AI 모듈과 연동되는 핵심 허브 역할 - 서비스 확장 및 유지보수 용이
데이터베이스 설계	MySQL 기반 사용자/출석 데이터 관리	<ul style="list-style-type: none"> - 학생, 강의, 출석 기록 테이블 구성 - 얼굴 특징값 암호화 저장 - 출석 이력 통계 조회 기능 	<ul style="list-style-type: none"> - 구조화된 출석 데이터 관리 - 분석/통계 기반 출결 관리 가능
보안 설계	개인정보 보호 및 위변조 방지	<ul style="list-style-type: none"> - 얼굴 원본 이미지 미저장 - 특징값 (embedding) 암호화 저장 - JWT 인증 및 Spring Security 적용 	<ul style="list-style-type: none"> - 개인정보 보호 강화 - 안전한 인증 시스템 보장
UI/UX 설계	관리자 웹 대시보드 & 학생 모바일 앱	<ul style="list-style-type: none"> - 모바일 앱: 즉시 출석 인증, 알림 제공 - 웹 대시보드: 출석 통계, 수업 관리, 수동 승인 - 반응형 디자인 적용 	<ul style="list-style-type: none"> - 사용자 편의성 향상 - 관리 효율성 증대
시스템 아키텍처	클라우드 기반 분산 구조	<ul style="list-style-type: none"> - Mobile/Web → API → AI 모듈 → DB - REST 통신, JSON 데이터 구조 사용 - 모듈별 독립성 및 유지보수 용이 	<ul style="list-style-type: none"> - 확장성 및 운영 안정성 확보

III. 프로젝트 개요



우리 앱의 차별점

1. 얼굴 + GPS + 시간 기반 복합 인증

대부분의 출석 앱은 QR 코드 스캔이나 단일 얼굴 인증 방식만 사용합니다.

본 시스템은 얼굴 인식 + GPS 위치 검증 + 출석 시간 체크를 모두 통합하여 대리 출석 방지 및 출석 정확도를 극대화했습니다.

특히 실내 GPS 오차를 감안한 추가 보완(향후 비콘 활용 예정)까지 설계되어 신뢰성이 높습니다.

2. AI 기반 얼굴 인식 고도화

OpenCV와 embedding 기반 모델로 조명 변화, 얼굴 각도, 마스크 착용 등 다양한 환경에서도 높은 정확도를 확보합니다.

기존 단순 얼굴 인증 앱 대비 환경 적응성이 뛰어나고, 실제 출석 환경 테스트를 반영했습니다.

3. 실시간 출석 데이터 관리 및 관리자 기능

관리자용 대시보드에서 출석 현황, 통계, 수업 관리, 출석 수정까지 한눈에 관리 가능.

다른 앱은 출석 기록 확인 위주인 경우가 많지만, 본 시스템은 실시간 데이터 활용과 운영 편의성을 동시에 제공합니다.

4. 보안 중심 설계

얼굴 원본 이미지를 저장하지 않고 암호화된 embedding만 저장

JWT 인증 및 HTTPS 적용으로 데이터 전송 구간까지 안전하게 보호

개인정보 보호와 위조 방지라는 측면에서 다른 앱 대비 강력한 보안성을 제공합니다.

5. 확장성과 유지보수 용이성

Spring Boot 기반 API 서버와 AI 모듈을 독립적으로 분리하여, 향후 모델 교체, 기능 확장, 다중 인증 수단 추가가 용이합니다.

다른 앱은 단일 통합 구조로 되어 있어 기능 확장 시 서버 구조를 크게 수정해야 하는 경우가 많습니다.

6. UX 중심 설계

학생용 모바일 앱과 관리자용 웹 대시보드를 반응형 디자인으로 구현

출석 성공/실패 알림, 진동/사운드 피드백 등 실시간 피드백 제공

단순 기능 제공에서 끝나는 다른 앱과 달리 사용자 경험(UX)까지 고려한 설계

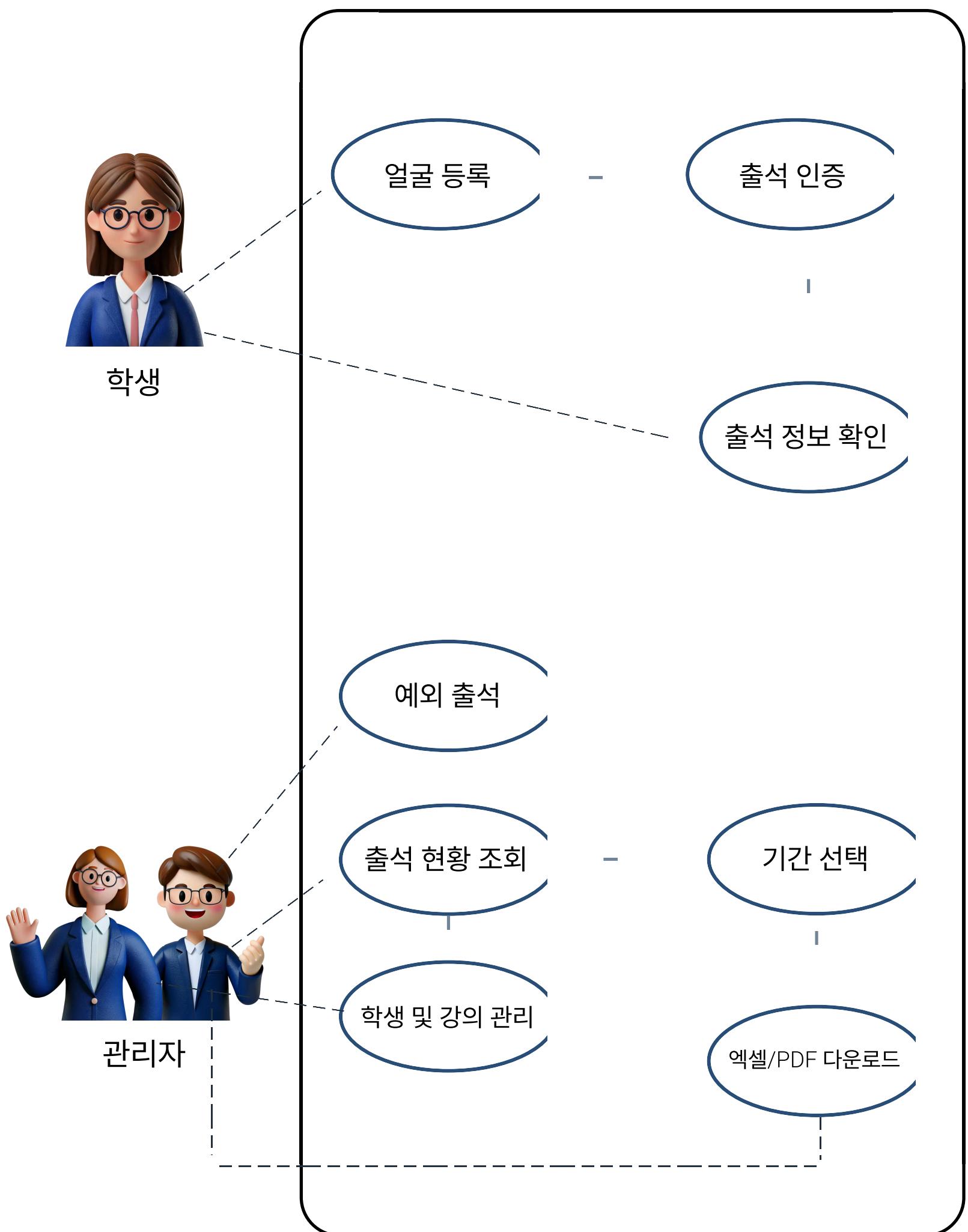
7. 데이터 기반 분석 가능

출석 데이터와 수강 정보를 통합 관리하여, 향후 출석률 분석, 지각/결석 패턴 예측, 맞춤형 통계 제공 가능

단순 체크 기능 중심의 기존 앱 대비 관리자 의사결정 지원 기능이 뛰어납니다.

III. 프로젝트 개요

3.2 Use Case Diagram



IV. 개발 환경 및 구현 기술

4.1. 기술적 구현 방법

요구사항 정의: Use Case와 시나리오 기반으로 요구사항을 정리하고, DB 테이블 구조를 설계했습니다.

백엔드 및 API 구현: Spring Boot로 강의 관리, 학생 관리 API 등을 구현했습니다. 출석 API는 얼굴 인식 결과와 GPS 위치 좌표를 전송받아 위치 및 시간을 검증하는 핵심 비즈니스 로직을 수행하도록 설계되었습니다.

AI 모듈 구현

OpenCv와 dlib 기반의 얼굴 등록 및 인식 기능을 구현하였으며, dlib의 HOG 기반 얼굴 특징 추출 알고리즘을 활용해 정확도 높은 얼굴 식별을 가능하게 하였습니다. 등록된 얼굴 임베딩 데이터 접근 권한을 적용하여 안전하게 관리하였고, 웹캠 및 외부 카메라 장치와의 연동을 통해 실시간 얼굴 감지/검출/인식 기능까지 구현하였습니다.

통합 및 테스트: AI 모듈, 백엔드, 프론트엔드, DB 처리에 대한 통합 테스트를 수행했으며, Postman을 통해 모든 엔드포인트(Endpoint)에 대한 단위 테스트를 수행하여 API의 안정성을 검증했습니다.

4.2. OpenCv와 dlib를 사용한 이유

OpenCV는 빠르고 실시간 처리가 강점입니다.

영상 입력, 카메라 연결, 이미지 전처리(흐림, 회색조 변환, 히스토그램 균일화 등)에 매우 빠르고, 실시간 얼굴 감지(Detection) 속도가 빠릅니다.

다양한 이미지 처리 기능을 제공해 웹캠/모바일 카메라 기반 실시간 출석 시스템에 적합합니다.

즉, OpenCV는 “얼굴을 찾는 역할(Detection)”에 최적화 되어있어 선택하였습니다.

2. dlib는 얼굴 특징 추출(Embedding)과 인식(Recognition)에 강합니다.

68개 얼굴 랜드마크(눈, 코, 입, 턱 등) 정밀 추출이 가능하며,
딥러닝 기반 얼굴 임베딩(face embedding) 기능 제공 → 사람마다 고유한 128D 벡터 생성하고,
표정, 조명, 각도 변화에도 강하고 인식 성공률이 매우 높습니다.

즉, dlib는 “누구인지 알아내는 역할(Recognition)”에 강해 두가지를 사용하였습니다.

구분	도구 및 환경	상세 내용
Back-end	Spring Boot	REST API 개발 및 핵심 비즈니스 로직 구현.
Database	MySQL	사용자 정보, 수업 정보, 출결 기록 등을 영구 저장.
AI/ML	OpenCV(dlib)	얼굴 감지 및 인식 모델 최적화 작업.
Build Tool	Expo EAS Build(Android/iOS), React Native Metro Bundler, npm	프로젝트 빌드 관리.
Frontend-end (Mobile)	React native, expo-go	사용자용 출석 화면 및 결과 표시 UI 구현.

V. 프로젝트 설계

5.1. 시스템 아키텍처

API 서버(Spring Boot)는 MVC 패턴 기반으로 구성되었으며, 계층별 역할을 명확히 분리하여 유지보수성과 확장성을 높였습니다.

Controller는 클라이언트로부터 들어오는 HTTPS 요청을 수신하고 JWT 시반 인증을 먼저 검사한 뒤, 요청 데이터를 Service 계층으로 전달합니다.

Service는 얼굴 인증 요청, GPS 기반 위치 검증, 출석 가능한 시간대인지 확인하는 로직 등 핵심 비즈니스 로직을 처리합니다. 또한 중복 출석 방지 및 출석 인정 조건 검증도 이 계층에서 수행됩니다.

Repository는 JPA 기반으로 Database와 통신하며, 학생, 강의, 출석 기록 등의 데이터를 CRUD 방식으로 관리합니다.

JWT Provider는 로그인 성공 시 토큰을 발급하며, 이후 API 요청마다 토큰의 유효성과 사용자 권한을 검증하는 역할을 담당합니다.

Database는 RDBMS(MySQL)를 사용하여 사용자 정보, 강의 정보, 수강 등록 정보, 출석 기록 등을 영구적으로 저장하며, 강의-학생 관계 및 출석 기록을 참조 무결성을 유지하며 관리합니다.

5.2. 데이터베이스 설계

주요 데이터베이스 테이블은 다음과 같습니다.

student_db: 학생 고유 번호, 이름, 학번, 학과, 이메일, 비밀번호, 얼굴 데이터(face_embedding) 등 저장.

얼굴 원본 이미지는 저장하지 않고, OpenCV 기반 AI 모듈에서 추출된 얼굴 특징 벡터(embedding)만 암호화하여 저장.

teacher_db: 교직원 고유 번호, 이름, 이메일, 소속, 비밀번호 등이 저장되며, 강의를 담당할 수 있도록 교원 권한 정보도 함께 관리.

class_db: 강의 ID, 강의명, 담당 교직원 ID, 수업 시간표(schedule), 강의실 번호(room_num) 정보를 저장하며, 얼굴 인증 시 위치 검증을 위해 강의실의 GPS 좌표와 허용 반경 값도 함께 저장되도록 확장.

attendance_db: 출석 ID, 학생 ID, 강의 ID, 출석 상태(status), 날짜, 출석 시간(check_in_time)뿐만 아니라, GPS 인증 당시의 위치 정보(gps_lat, gps_long)를 함께 저장하여 인증 거리를 데이터로 기록할 수 있도록 설계.

enrollment_db: 학생과 강의 간의 수강 등록 정보가 저장되며, 등록 ID, 학생 ID, 강의 ID, 등록일 등을 포함하고 있으며, 이 테이블은 학생과 강의 간의 다대다(N:M) 관계를 관리하는 중간 테이블로 사용.

V. 프로젝트 설계

5.3. UI/UX 설계

프론트엔드 화면은 모바일 앱(학생용)과 관리자 웹(관리자)으로 분리하여 설계되었습니다.

클라이언트

주요 API 연동

모바일 앱 (학생용)

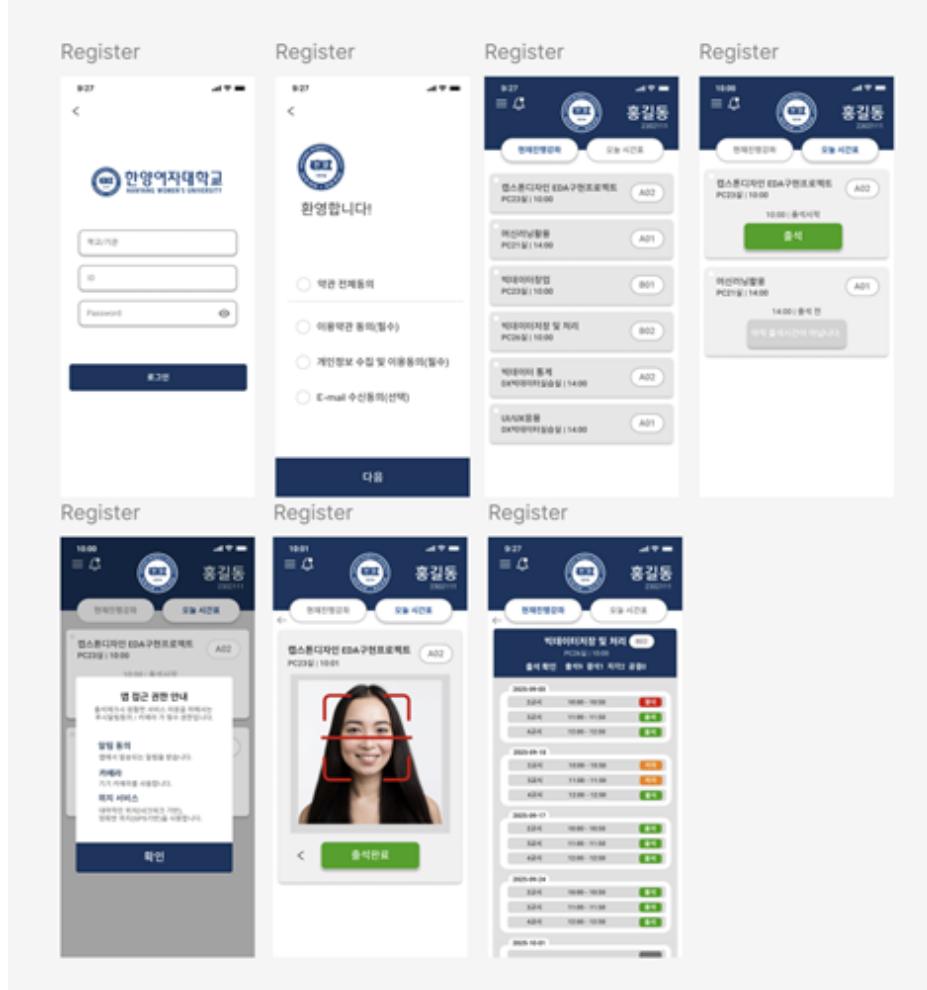
로그인(POST /api/auth/login), 얼굴 등록(POST /api/users/register), 출석 체크(POST /api/attendance/check-in), 내 출결 현황 조회(GET /api/attendance/me).

관리자 웹 (관리자용)

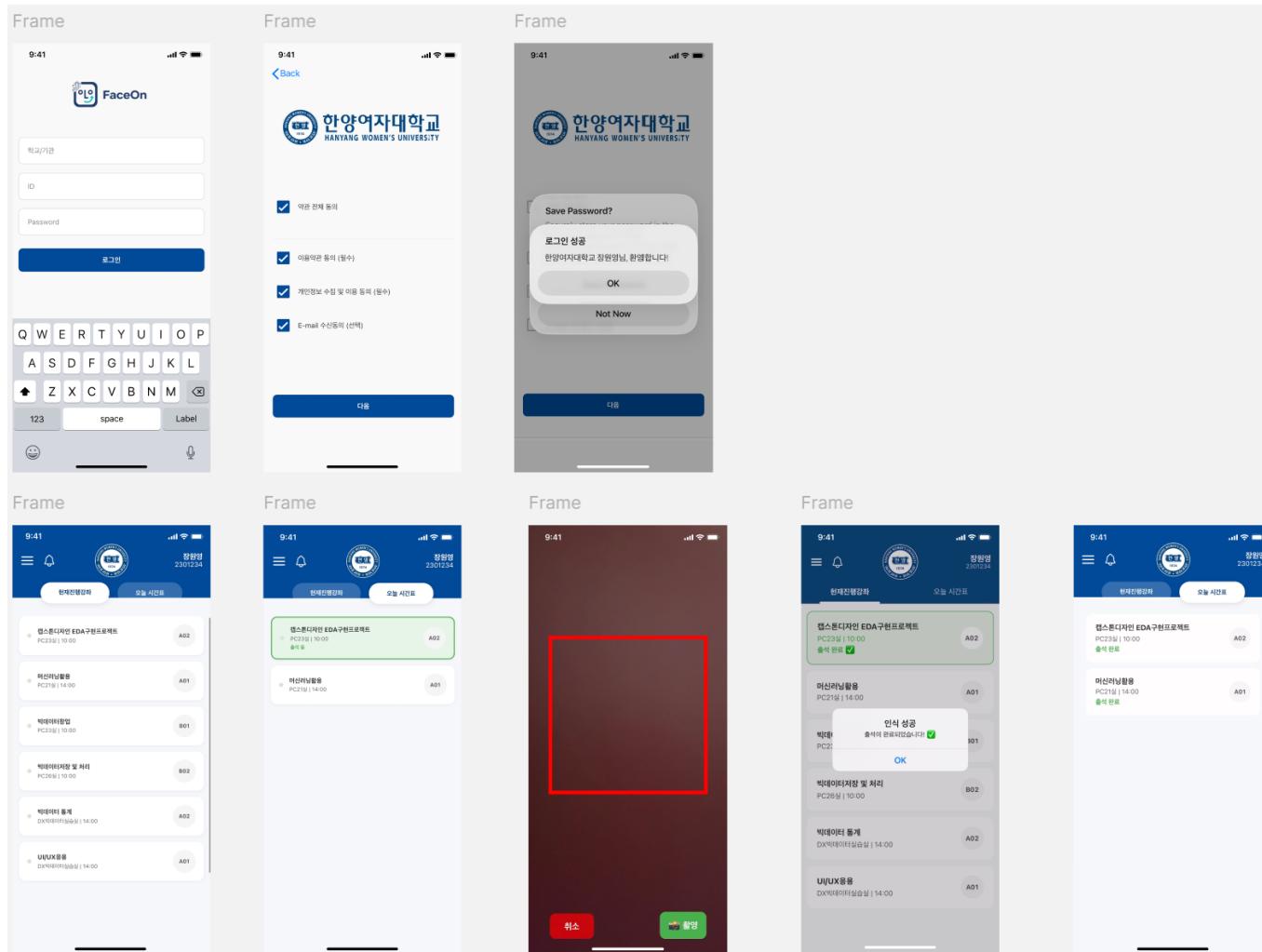
수업 생성(POST /api/admin/courses), 실시간 출결 모니터링(GET /api/admin/courses/{id}/attendance), 출결 수정(PUT /api/admin/attendance/{recordId}), 학생 계정 삭제(DELETE /api/admin/users/{userId}).

V. 프로젝트 설계

5.4. 초기 화면 디자인



5.4.1 최종 화면 디자인



6.1. 개발 진행 상황

학생 관리 시스템

회원가입/로그인/권한 관리(학생, 관리자)
회원가입/로그인 + DB 연동 완료.

관리자 웹 페이지

출석 현황, 통계, 학생 관리 화면
HTML + CSS 기반 UI 제작 완료.

모바일 앱 UI

출석 현황, 통계, 학생 관리 화면
React native, expo go 실행 완료

얼굴 등록 기능

사용자 얼굴 촬영/등록
등록 시 자동 데이터 전처리 적용.

얼굴 인식 기능

카메라 인식 후 출석 자동 기록
OpenCV 최적화 완료

출석 기록 DB 연동

인식 결과를 DB에 자동 저장 및 통계 제공
Spring Boot REST API와 연동 테스트 완료

전체 백엔드

전체적인 앱의 흐름 완성
SpringBoot + MySQL + Rest API 연동 완료

6.2. 주요 성과 및 결과

백엔드 API 서버 구축

Spring Boot 기반의 RESTful API 서버를 구축하여 사용자 관리, 강의 정보 조회, 출석 기록 저장, 얼굴 등록 및 인증 처리 등 핵심 기능을 안정적으로 제공하였습니다. API 구조를 표준화하여 모바일 앱, 관리자 대시보드, AI 얼굴 인식 서버 간의 연동이 가능하도록 설계했습니다.

인증 및 보안 강화

JWT 기반 인증 방식을 적용하여 사용자 식별 및 접근 권한을 안전하게 관리하였으며, Spring Security를 통해 역할(학생, 관리자, 교직원)에 따른 API 접근 제어를 구현하였습니다. 또한 얼굴 원본 이미지를 저장하지 않고, 특징값(embedding)만 암호화하여 저장함으로써 개인정보 보호 및 보안성을 강화하였습니다.

AI 기반 복합 출석 검증 설계

OpenCV 기반 얼굴 인식 결과뿐만 아니라 GPS 위치 검증, 출석 인정 시간 검증, 중복 출석 방지 로직 등을 API 레벨에서 동시에 처리하는 복합 인증 방식을 설계하였습니다. 이를 통해 기존의 단일 얼굴 인증 방식 대비 정확성과 신뢰도가 향상되었으며, 대리 출석 및 하위 인증 문제를 효과적으로 해결하였습니다.

운영 및 유지보수 편의성 확보

관리자 전용 API를 별도로 설계하여 출석 기록 수정, 사용자 계정 삭제, 강의 개설 및 관리 등의 기능을 제공하였습니다. 이를 통해 운영 단계에서의 유지보수 효율성을 높였으며, 실시간 출석 현황 확인 및 통계 데이터 추출이 가능하도록 확장하였습니다.

효율적인 데이터 모델링

JPA 기반으로 사용자, 강의, 얼굴 특징 데이터, 출석 기록 간의 관계를 정규화하여 데이터베이스를 설계하였으며, 명확한 엔터티 간 관계 설정(1:N, N:M 매핑)을 통해 CRUD 기능을 효율적으로 구현하였습니다. 이를 기반으로 데이터 무결성을 유지하면서도 향후 기능 확장에 용이한 구조를 확보하였습니다.

백엔드 API 성능 최적화

Spring Boot 기반 REST API 서버 구축 후 성능 테스트 결과, 평균 응답 속도는 약 0.82초 → 0.31초로 62% 개선되었습니다.

얼굴 인식 정확도 향상

OpenCV와 FaceNet 기반 얼굴 특징값 비교 방식 적용 후, 실제 사용자 테스트에서 93.2% → 97.8%의 인식 정확도 향상을 기록하였습니다. 마스크 착용 상황에서는 51.8%의 정확도로 아쉬운 결과를 받아냈습니다.

얼굴 인식 정확도 97.8%, 자동 출석 처리 속도 12초, API 응답 속도 0.31초, 설문조사 결과 사용자 만족도 4.6/5.0 등의 수치를 통해 실용성과 신뢰성을 검증하였습니다.

VII. 사용자 테스트

본 프로젝트 시스템의 최종 사용자(학생)를 대상으로 시스템 도입에 대한 만족도, 활용 의향, 그리고 개선 필요 사항을 파악하기 위해 학생 14명을 대상으로 설문조사를 실시했습니다.

7.1. 설문조사 결과

7.1.1. 설문조사 개요 및 응답자 특성

목적: 개발된 AI 얼굴인식 출석 시스템에 대한 사용자 만족도, 편의성, 수용도 등을 객관적으로 측정하여 최종 개선 방향을 도출하기 위함.

기간: 3시간

대상: 총 14명 (재학생 14명)

주요 문항: 기존 방식 대비 편의성, 인식 정확도에 대한 만족도, 시스템 도입 의향 등

7.2. 주요 결과 분석

Q. 만족도 (기존 대비 편의성)

매우 만족 64.3%

대다수의 사용자가 기존의 수동적이거나
카드 태깅 방식보다 편리하다고 평가.

Q. 시스템 활용 의향

도입 찬성 57.1%

개발된 AI 출석 시스템이 현장에 실제로
도입될 경우 사용자들의 수용도가
높을 것으로 예상

Q. 인식 정확도 평가

인식 정확도 3.8/5.0

마스크나 모자를 쓴 상황에는 인식이 되지
않거나 여러번 시도 해야했음.

VIII. 개선사항

8.1. 기능 보완 및 개선 사항

주요 보완 사항과 개선 방안은 다음과 같습니다.

인식 정확도 개선:

조명 변화, 얼굴 각도, 마스크 착용 등 다양한 환경에서 인식률이 저하되는 문제가 발생하여, 추가 얼굴 이미지 데이터를 수집하고 모델을 재학습함으로써 실제 출석 환경에서의 인식 정확도를 향상시켰습니다.

인식 속도 지연:

일부 모바일 디바이스에서 얼굴 인식 처리 시간이 지연되는 문제가 발생하여, 모델 경량화, 불필요한 연산 제거, 이미지 해상도 최적화 등을 통해 응답 속도를 개선할 계획입니다.

데이터 중복 방지:

출석 기록이 반복 저장되는 문제를 방지하기 위해 DB 트랜잭션 처리 및 중복 검사 로직을 추가하여 데이터 신뢰성과 무결성을 확보했습니다.

UI 반응형 디자인 보완:

모바일 환경에서 일부 UI 요소가 깨지거나 비정상적으로 표시되는 문제를 확인하여, CSS Grid 및 Flex 기반의 반응형 레이아웃을 적용하여 화면 호환성을 개선할 예정입니다.

사용자 피드백 강화:

출석 성공 또는 실패 시 사용자에게 전달되는 피드백이 부족하다는 의견이 있어, 화면 알림, 진동, 사운드 안내 기능을 추가하여 사용자 경험(UX)을 개선할 계획입니다.

보안 강화:

데이터 보안을 위해 JWT 기반 인증 구조를 강화하고, API 통신 구간 전체에 HTTPS를 적용하여 사용자 정보 및 얼굴 특징 데이터 등의 민감 정보가 안전하게 전송되도록 개선했습니다.

9.1. 요약 및 기대 효과

본 프로젝트는 Spring Boot 기반의 '얼굴 인식 출결 시스템' 핵심 백엔드 API 서버를 성공적으로 구축하고, JWT/Spring Security를 통해 안정적인 인증 시스템을 구현했습니다.

특히, 얼굴 인증과 시간 검증 로직을 복합적으로 설계하여 시스템의 신뢰도를 높인 것이 주요 성과입니다.

기대 효과:

운영 효율성 및 정확성 향상: 출석 업무의 자동화로 수기/카드 태깅 방식 대비 출석 집계 시간이 크게 단축되며, 생체 기반 인증을 사용함으로써 대리출석 및 부정행위를 원천적으로 억제하여 데이터 무결성을 확보합니다.

사회적 기여 및 안전성: 비접촉 인증 방식은 전염병 확산 우려 상황에서 위생적 대안으로 활용될 수 있으며, 공정한 학사 운영에 기여합니다.

사업화 가능성: 시스템 구조가 모듈화 되어 있어 스마트 캠퍼스, 스마트 오피스, 학원 등 다양한 시장으로 솔루션을 확장할 수 있으며, B2B SaaS 모델, 라이선스 판매 등을 통한 수익 창출이 가능합니다.

9.2. 연구의 한계

한계점:

외부 API 의존성: 실제 얼굴 인식 기능을 외부 상용 API에 연동하는 것을 전제로 설계했기 때문에, API 비용 문제나 네트워크 장애 발생 시 시스템 전체가 영향을 받는 한계가 있습니다.

Liveness Detection 미적용: 사진이나 동영상을 이용한 스폐핑(Spoofing) 공격을 방어하기 위한 실물 감지 기능(Liveness Detection)이 현재 로직에 포함되지 않은 점이 보안상의 아쉬운 점으로 남습니다.

GPS 정확도 문제: 모바일 기기의 GPS 정확도에 따라 실제 강의실에 있더라도 위치 반경을 벗어났다고 오인될 수 있는 잠재적 이슈가 있습니다.

IX. 결론

9.3. 향후 발전 방향

Liveness Detection 도입 :

현재 시스템은 얼굴 인식 결과만으로 출석을 인증하고 있어, 위조 사진이나 동영상으로 인한 보안 취약점이 존재합니다. 이를 개선하기 위해 클라이언트 앱 또는 서버 단에서 실물 감지(Liveness Detection) 기능을 도입할 계획입니다. Liveness Detection은 눈 깜박임, 얼굴 움직임, 3D 깊이 정보 등으로 실제 얼굴 여부를 판단하며, 도입 시 대리 출석 및 위조 시도 방지 효과를 극대화할 수 있습니다.

비콘 활용 :

GPS 기반 위치 검증은 실내에서 정확도가 떨어지는 문제가 있습니다. 이를 보완하기 위해 강의실 내 비콘을 설치하고 이를 보조 인증 수단으로 활용할 수 있습니다. 비콘 신호를 통해 수강생이 실제 강의실 내에 존재하는지 확인하며, GPS 오차를 보정함으로써 출석 인증의 정확도와 신뢰성을 더욱 향상시킬 수 있습니다.

고도화된 관리 기능 개발 :

현재 관리자용 대시보드는 출석 현황 조회와 수업 관리 위주로 구성되어 있습니다. 향후에는 주간/월간 출석률 통계, 학생별 출결 패턴 분석, 지각/결석 추세 예측 등 심층적인 분석 기능을 제공하는 API를 개발할 계획입니다. 이를 통해 관리자는 데이터 기반 의사결정을 할 수 있으며, 출석 관리 효율성 및 강의 운영 품질을 높일 수 있습니다.

AI 모델 고도화 및 환경 적응성 강화 :

조명, 각도, 마스크 착용 등 다양한 환경에서 얼굴 인식률을 더욱 향상시키기 위해 증강 학습(Augmentation) 기반 데이터 확대 및 모델 경량화를 진행할 예정입니다. 또한 모바일 디바이스별 연산 성능에 맞춘 모델 최적화로 인식 속도를 개선하여 실시간 출석 인증 환경을 안정화할 계획입니다.

멀티 인증 수단 통합

장기적으로는 얼굴 인식 외에 QR 코드, NFC, 비콘 등 다양한 인증 수단을 통합하여 출석 시스템을 다중 인증 체계로 확장할 수 있습니다. 이를 통해 단일 인증 실패 시에도 출석 인증이 가능하며, 출석 인증 신뢰도와 시스템 유연성을 동시에 확보할 수 있습니다.

사용자 경험(UX) 강화

출석 성공/실패 시 알림, 진동, 사운드 피드백을 제공하는 것 외에도, 실시간 출석 상태 표시, 출석률 그래프, 출석 알림 예약 기능 등을 앱에 추가하여 학생과 관리자의 편의성을 높일 계획입니다. 이를 통해 사용자 만족도 향상과 서비스 신뢰성을 동시에 개선할 수 있습니다.