

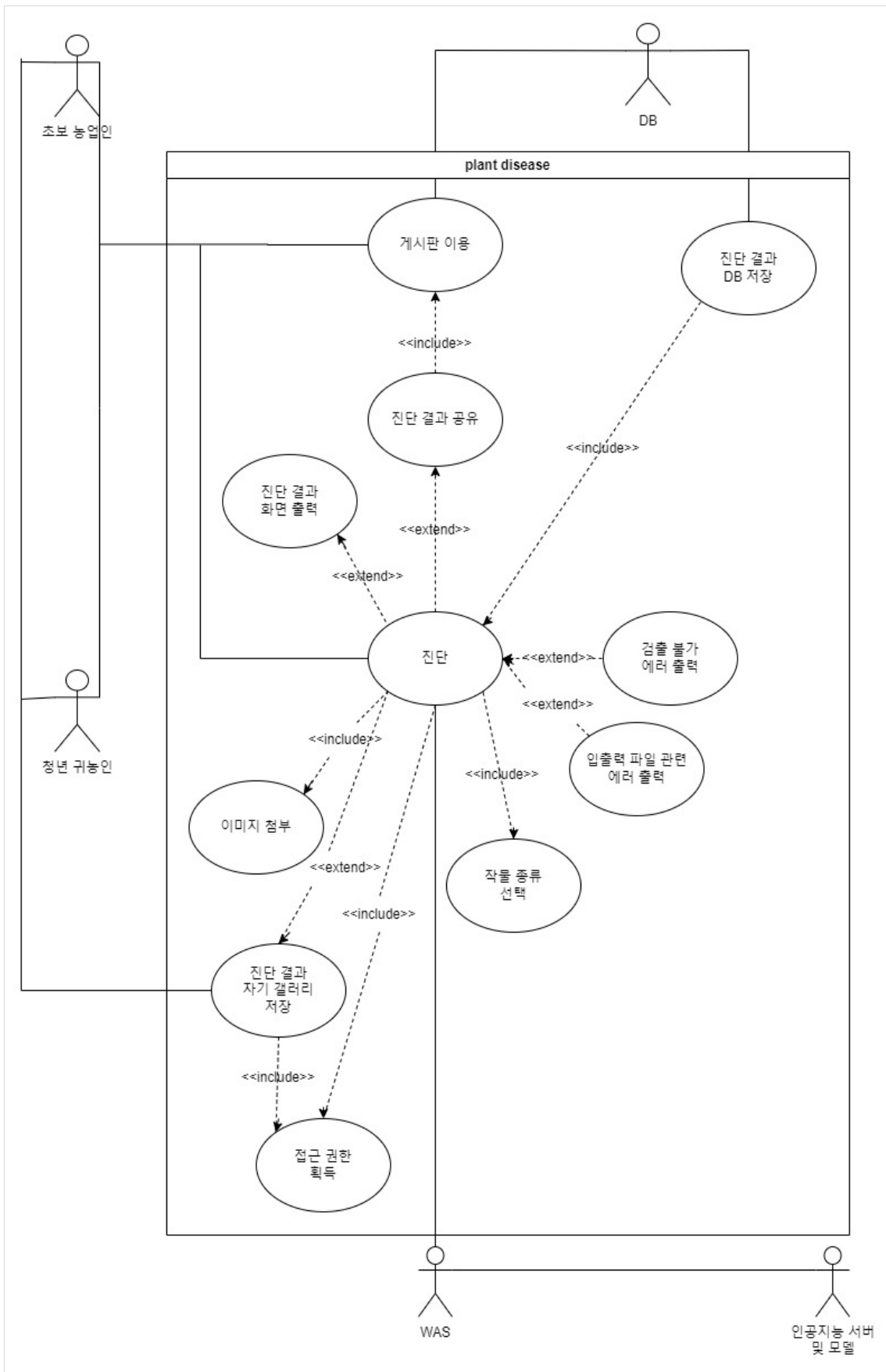
1. 시스템 개요

본 시스템 ‘시설 작물 병해 검출 및 분류 플랫폼’을 구축하기 위한 프로젝트이다. 작물에 대한 지식이 깊지 않거나 정보를 얻기 힘든 직장인, 부부, 귀농인을 위한 인공지능 모델 활용 병해 진단 플랫폼이다. 전문가가 개입되지 않기 때문에 시간, 비용면에서 정확성, 신속성, 효율성 면에서 크게 개선될 것으로 기대한다. 전체 시스템에서 인공지능을 활용한 작물의 병해 진단, 농업인 커뮤니티, API 활용 병해 연관 정보 제공 등의 서비스를 이용할 수 있으며, 일부 기능의 경우 접근 권한과 회원 조건을 만족해야만 이용 가능한 기능이 존재한다. 해당 서비스는 안드로이드 앱(Play Store)을 통해 이용가능하다.

2. Use Case

가. Actor 목록

Actor	구분	설명
초보 농업인	시스템사용자	주말농장이나 집 옥상에서 작은 스마트팜을 운영 중인 소규모 농업인
청년 귀농인	시스템사용자	1년 전 귀농 후 농사에 어려움이 많은 젊은 청년
DB	DB	진단 결과를 저장하는 DataBase
WAS	WAS	진단할 데이터를 AI 서버로 전달, 진단 결과를 클라이언트에 전달
인공지능 서버 및 모델	인공지능 서버 및 모델	WAS로부터 받은 이미지 데이터를 분류기에 넣어 분류한 뒤 진단 결과를 WAS에게 전달



나. 새 프로젝트 시작

(1) 개요

‘AI 병해충 진단하기’ 기능을 이용하기 위해 사용자는 카메라와 갤러리 접근 권한을 반드시 허용해야만 한다. 단순 진단에는 로그인 과정을 요구하지 않지만 진단 결과를 커뮤니티에 공유하기 위해서는 로그인 및 회원가입 과정이 필요하다. 진단 가능한 작물로는 딸기, 상추, 고추, 토마토 총 4가지이고, 각 작물마다 2가지 병해로 구분한다.

(2) 사건 흐름

- 기본 흐름

1. 사용자는 본인의 작물에 발생한 병해 정보를 얻기 위해 해당 앱을 설치한다.
2. 작물의 종류(딸기, 상추, 고추, 토마토 中 택1)를 선택하고 병해가 발생한 작물의 이미지(촬영/앨범)를 첨부한다. 이미지를 첨부하는 과정에서 사용자의 권한 승인이 필요한데 ‘checkPermission()’ 함수를 이용해 WRITE_EXTERNAL_STORAGE, READ_EXTERNAL_STORAGE, CAMERA(기기 접근 권한) 각각 갤러리, 앨범 접근 권한을 얻는다. 권한이 모두 허용되면, ‘dispatchTakePictureIntent()’ 함수를 호출하여 카메라 앱을 실행하고, 사용자가 1:1 비율로 crop한 이미지를 얻습니다. 해당 이미지는 앱의 UI에 표시된다.
3. ‘진단하기’ 터치 시 ‘inspectPlants()’ 메소드를 호출해 JSON 형식의 데이터(ex. {"crop_sort" : "딸기"})와 진단할 작물 이미지(crop_image)를 WAS 서버로 전송한다. WAS 서버에서는 받은 작물 이미지 파일명을 '작물명-시간-랜덤문자열' 형식으로 변경한다. 예를 들어, 'strawberry_202304020819_dkp1swqllasp'와 같은 형태가 될 수 있다. 변경된 파일명과 함께, WAS는 받은 JSON 형식의 작물 종류 정보와 작물 이미지를 AI 서버로 전송한다.
4. AI 서버에서 WAS 서버로부터 받은 선택 작물과 이미지를 분석해 질병을 분류한다.

Result	Description
A(정상)	input작물과 같은 작물임, 결과의 정확도가 높음.
B(오류)	input작물과 같은 작물임, 결과의 정확도가 낮음.
C(오류)	input작물과 다른 작물임, 결과의 정확도가 높음.
D(오류)	input작물과 다른 작물임, 결과의 정확도가 낮음.

```
// result.json
{
  "result": "1",    // 경우는 1,2,3,4
  "input_crop": "tomato",
  "output_crop": "tomato",
  "diagnosis": {
    "label1": "Disease A",
    "confidence1": 0.75,
    "label2": "Disease B",
    "confidence2": 0.2,
    "label3": "Normal",
    "confidence3": 0.05
  }
}
```

Description	Return
작물-질병 일치 O → confidence ↑	A(정상)
작물-질병 일치 O → confidence ↓	B(오류)
작물-질병 일치 X → confidence ↑	C(오류)
작물-질병 일치 X → confidence ↓	D(오류)

선택된 작물에 속하는 질병 여부, confidence 수치를 바탕으로 JSON 형태의 분석 결과를 다시 WAS 서버로 반환한다.

5. 사용자는 진단 결과를 확인하고, 결과를 스크린샷으로 저장하거나 커뮤니티에서 다른 사용자와 공유한다.

- 대안 흐름

1. 작물의 종류와 작물의 이미지 두 가지가 선택된 경우 진단 가능, 그렇지 않은 경우 알림 출력
2. 이미지 파일의 확장자명을 .jpg, .png로 제한하고 있지만, 이외의 확장자명이 진단 이미지로 선택된 경우 오류 문구 출력
3. 선택된 작물과 다른 종류의 작물의 이미지가 첨부된 경우, 모델에서 일정 수치 이상 기록되는지 여부로 판단
4. 진단 결과가 비정상적인 경우(ex. 정상: 33%, A 질병: 33%, B 질병: 33%) ‘한 줄 요약’을 통해 다른 이미지로 재검사 권장

예외 상황	예외 대안
작물의 종류, 작물의 이미지중 하나라도 선택되지 않은 경우	“병해충 진단을 위해서 작물의 종류와 이미지 모두 선택해주세요.”라는 알림 출력
이미지 파일의 확장자명이 “.jpg”, “.png”가 아닌 경우	“jpg, png 형식의 파일만 지원합니다.”라는 에러 메시지 출력
선택한 작물의 종류와 작물 이미지가 일치하고, 진단 결과 값의 신뢰도가 낮은 경우	“진단의 정확도가 낮습니다.”라는 에러 메시지 출력
선택한 작물의 종류와 작물 이미지가 일치하지 않는 경우	“작물 선택이 잘못되었을 가능성이 큼니다. 작물 종류를 올바르게 선택 후 재검사해주세요.”라는 에러 메시지 출력
선택한 작물의 종류와 작물 이미지가 일치하지 않고, 진단 결과 값의 신뢰도가 낮은 경우	“지원되지 않는 작물일 가능성이 큼니다.”라는 에러 메시지 출력

(3) 사전/사후조건

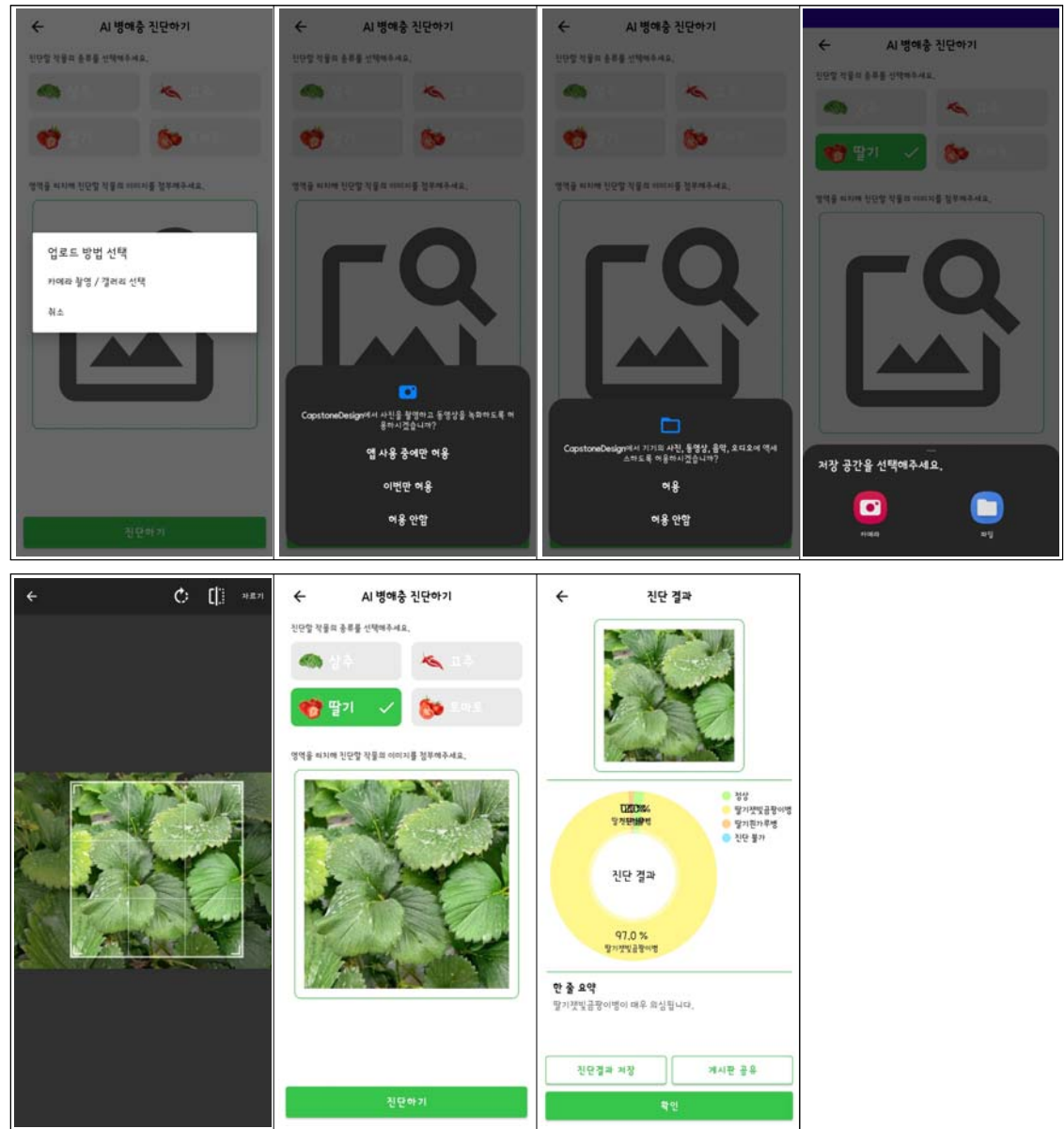
- 사전 조건

1. 사용자는 진단 전 반드시 작물의 종류를 선택하고, 병해가 발생한 작물의 이미지를 첨부하여야 한다.
2. 선택한 작물의 종류와 첨부한 작물 이미지의 종류는 같아야 한다.

- 사후 조건

1. 사용자가 진단 결과를 커뮤니티에 공유하기 위해서는 회원가입/로그인 과정이 필요하다.

(4) 관련 화면



3. Data map

use case 명	작물의 병해충 진단하기		
use case 설명	작물의 종류, 사진을 이용해 작물의 병해충을 진단한다.		
Data Name	자료형	크기	설명
crop_sort	String	6	작물 종류
crop_image	multipartfile	10MB	병해충이 있는 작물 사진

기타 use case에서 필요한 데이터

Member	Data Name	자료형	크기	설명
	memberId	Long	8	회원번호
	password	String	30	비밀번호
	email	String	30	이메일
	nickname	String	30	닉네임
	region	Enum	25	대한민국 시 단위 이름
	createdDate	LocalDateTime	8	생성일자
	updatedAt	LocalDateTime	8	수정일자

Region (Enum)	Data Name	설명
	SEOUL	서울특별시
	INCHEON	인천광역시
	DAEJEON	대전광역시
	GWANGJU	광주광역시
	DAEGU	대구광역시
	ULSAN	울산광역시
	BUSAN	부산광역시
	SEJONG	세종특별자치시
	GYEONGGI_DO	경기도
	GANGWON_DO	강원도
	CHUNGCHAEONGBUK_DO	충청북도
	CHUNGCHAEONGNAM_DO	충청남도
	JEOLLABUK_DO	전라북도
	JEOLLANAM_DO	전라남도
	GYEONGSANGBUK_DO	경상북도
	GYEONGSANGNAM_DO	경상남도
	JEJU	제주특별자치도

diagnosis Result	Data Name	자료형	크기	설명
	diagnosisResultId	Long	8	진단 결과 번호
	memberId	Long	8	해당 진단을 요청한 회원 번호
	diseaseName	Enum	50	진단받은 병해충 명
	createdDate	LocalDateTime	8	생성일자
	updatedAt	LocalDateTiem	8	수정일자

Disease Name (Enum)	Data Name	설명
	PEPPER_MILD_MOTTLE_VIRUS	고추마일드모틀바이러스
	PEPPER_BACTERIAL_SPOT	고추점무늬병
	STRAWBERRY_GRAY_MOLD	딸기잿빛곰팡이병
	STRAWBERRY_POWDERY_MILDEW	딸기흰가루병
	RETTUCE_SCLEROTINIA_ROT	상추균핵병
	RETTUCE_DOWNY_MILDEW	상추노균병
	TOMATO_LEAF_MOLD	토마토잎곰팡이병
	TOMATO_YELLOW_LEAF_CURL_VIRUS	토마토황화잎말이바이러스

Board	Data Name	자료형	크기	설명
	boardId	Long	8	게시글번호
	title	String	45	게시글 제목
	content	String	200	게시글 본문 내용
	tag	Enum	10	knowhow, question 2개로 구분
	image	String	400	게시글의 이미지 파일이 저장된 경로
	isSolved	Boolean	1	tag가 question인 게시글의 경우, 문제 해결 여부
	createdDate	LocalDateTime	8	생성일자
	updatedAt	LocalDateTime	8	수정일자

Tag (Enum)	Data Name		설명
	KNOWHOW		노하우
	QUESTION		질문

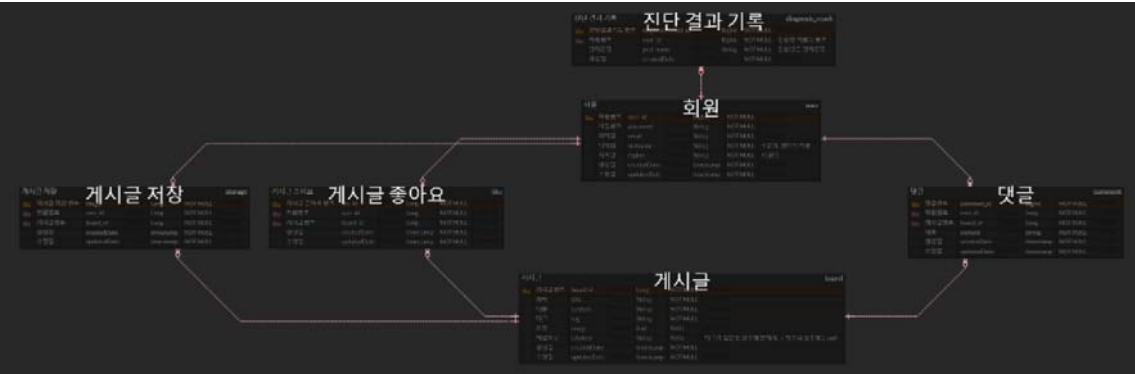
Board Like	Data Name	자료형	크기	설명
	likeId	Long	8	좋아요 고유 번호
	memberId	Long	8	좋아요를 누른 회원 번호
	boardId	Long	8	좋아요가 눌린 게시물 번호
	createdDate	LocalDateTime	8	생성일자
	updatedAt	LocalDateTime	8	수정일자

Board Storage	Data Name	자료형	크기	설명
	storageId	Long	8	저장 고유 번호
	memberId	Long	8	저장을 누른 회원 번호
	boardId	Long	8	저장이 눌린 게시물 번호
	createdDate	LocalDateTime	8	생성일자
	updatedAt	LocalDateTime	8	수정일자

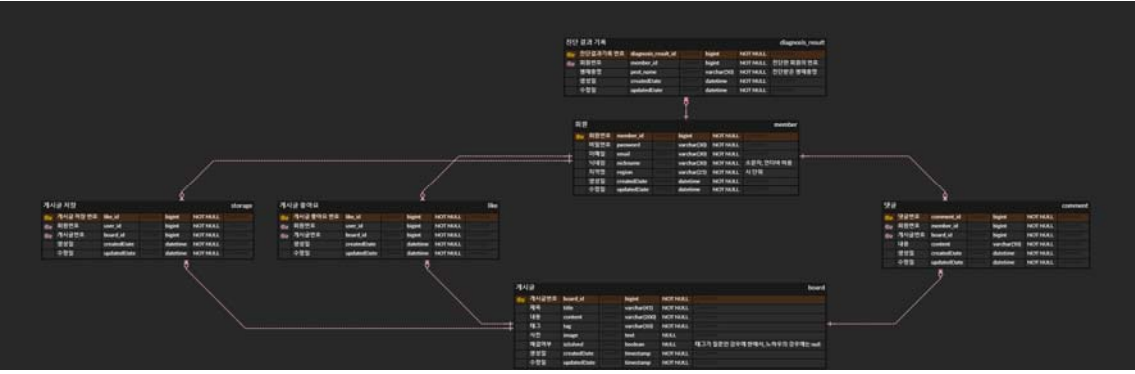
Comment	Data Name	자료형	크기	설명
	commentId	Long	8	댓글 번호
	memberId	Long	8	해당 댓글을 작성한 회원 번호
	boardId	Long	8	해당 댓글이 속한 게시물 번호
	content	String	50	댓글 내용
	createdDate	LocalDateTime	8	생성일자
	updatedAt	LocalDateTime	8	수정일자

4. ER Diagram

ER Diagram - 크게 보기



ER Diagram - 구체적으로



5. 통신상에서 주고받는 데이터 구조도

```
json 형식 사용
{
  "crop_name": "tomato",
  "diagnosis": {
    "label": "Disease A",
    "confidence": 0.95
  }
}
```