

사람을 좋아하는 개발자 조원형입니다.

Email jwh.dev0103@gmail.com

Mobile 010-3649-3276

Github <https://github.com/JoWonHyeung>

Blog <https://meshjo12.tistory.com>

01 PROFILE

02 강점

03 프로젝트

04 입사 후 계획

[ABOUT ME]

안녕하세요. 사람을 좋아하는 개발자
조원형입니다.

2022년에는 인공지능과 데이터 분야에
관심을 갖게 되어, 다양한 공모전과
프로젝트에 참여하였습니다. 그 중에서
멀티캠퍼스의 융복합 프로젝트에서 '장애인
불법 주차 방지 시스템 구축'이라는 주제로
최우수상을 수상할 수 있었습니다.

하지만, 좀 더 다양한 분야에 대한 궁금증으로
KB IT's your life에 입과하여 백엔드 과정을
수료하는 과정에서 백엔드 개발에 더욱
흥미를 느끼기 시작했습니다. 이를 바탕으로
최근에는 Java, Spring, MyBatis를 기반으로
'클라우드 펀딩 플랫폼'이라는 주제로 최종
프로젝트를 마무리 지었습니다.

다음달에는 프로젝트를 같이 진행했던
팀원들과 함께 카페 정보 제공 웹 서비스를
사이드 프로젝트로 진행할 생각입니다.

다양한 경험이 저의 역량을 넓혔고, 계속해서
새롭고 도전적인 일들을 추구해 나가고자
합니다.



[EDUCATION]

- 삼성 SDS 멀티캠퍼스 '융복합 프로젝트형
빅데이터 분석 서비스 개발' 과정에서 데이터 및
인공지능 분야 역량 강화
- KB IT아카데미를 통해 Java, Spring, MyBatis를
활용한 웹 서버 개발 역량 강화
- 과학기술정보통신부 주관 오픈소스 컨트리뷰션
아카데미에서 'Apache Zeppelin' 프로젝트에
기여를 시작으로 오픈소스에 꾸준한 관심을
보이고 있음

+

[프로그래밍스킬]

- ◆ Java를 활용한 객체 지향 프로그래밍 구현 경험
- ◆ Spring 프레임워크를 활용한 웹 서버 개발 경험
- ◆ MyBatis를 활용하여 프로그램 코드와 SQL을
적극적으로 분리하고자 함
- ◆ AI 및 데이터 분야 학습 경험으로 Python의 여러
라이브러리 및 프레임워크 사용 경험

2. 강점

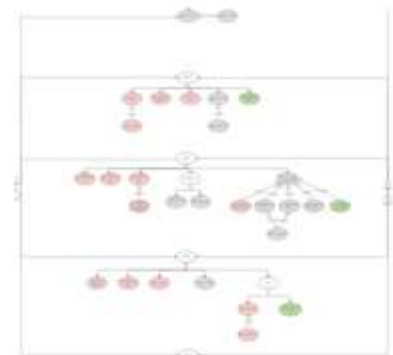
설계 경험으로 인한 문서 작성 역량



ERD : 정규화 과정 및 다:다 관계 해소를 통해 총 13개의 테이블이 작성되었습니다.



Front UI : miro.com 에서 UI설계를 진행했습니다.



Use Case Diagram : 초기 개발시에 사용자 관점에서 어떠한 기능이 필요한지 작성했습니다.



API 명세서 : 노선을 통해 작성하고 있으며, 포스팅 기능을 통해 팀원들과 지속적으로 소통하며 작업하고 있습니다.



23 Use Case Diagram



그림 24 클래스 다이어그램

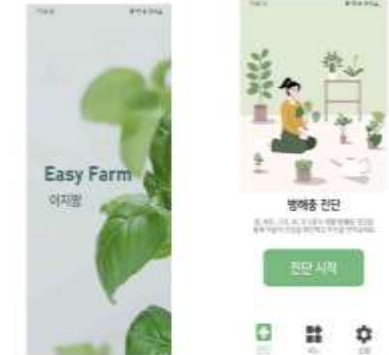
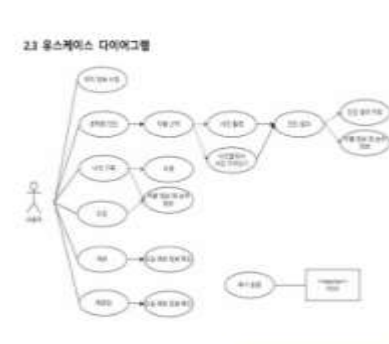


그림 61 Splash 화면



그림 62 메인의 진단 화면

✓ 개발자는 자신이 개발했던 내용을 문서로 작성하는 능력이 중요하다고 생각합니다. 저는 많은 팀 프로젝트에서 팀장 역할을 수행하면서 주도적으로 설계를 이끌어갔고, 이에 따라 많은 문서들을 만들어 왔습니다.

➤ 입사후에도 이러한 문서 작성 역량을 활용하여 팀원들 간의 원활한 소통을 위해 노력하겠습니다.

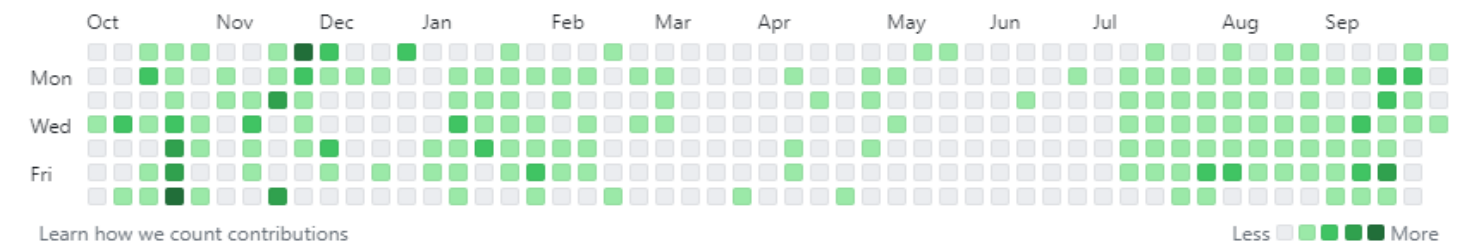
성장을 위해 꾸준히 노력하는 성실함



[2016/03 ~ 2021/03] 한국산업기술대학교(現 한국공학대) 게임공학부 자퇴
[2021/03 ~ 2023/08] 세종대학교 컴퓨터 공학과 졸업
[2022/01 ~ 2022/01] 삼성SDS 알고리즘 특강 수료
[2022/01 ~ 2022/06] 삼성SDS 멀티캠퍼스 '프로젝트형 빅데이터 분석 서비스 개발'과정 수료
[2022/03 ~ 2022/08] 세종대학교 중앙동아리 SAI(Sejong Artificial Intelligence) 5기 활동
[2022/07 ~ 2022/09] 과학기술정보통신부 주관 오픈소스 컨트리뷰션 아카데미 'Apache Zeppelin' 프로젝트 참여
[2023/07 ~ 2023/10] KB국민은행 IT아카데미 IT's your life 4기 과정 수료

659 contributions in the last year

Contribution settings ▾



✓ 개발자는 끊임없이 배우려고 하는 자세가 중요하다고 생각하여 다양한 IT 활동에 참여하며 성장해왔습니다. 또한, 커밋을 꾸준히 하려고 노력하고 있습니다.

➤ 입사후에도 성실함을 바탕으로 지속적인 배움을 실천하겠습니다.

3. PROJECTS

01. 클라우드 펀딩 플랫폼

소스코드: <https://github.com/orgs/kb-artizen/repositories>

프로젝트 개요

후원 및 펀딩이 가능한 예술 중심의 클라우드 펀딩 플랫폼

기술 스택

Language : Java, Javascript

Backend : Spring boot, MyBatis, AWS S3

Frontend : React

나의 역할

리뷰 및 티켓 관련 Service Layer, Persistence Layer 의 코드 작성

Spring boot – AWS S3 연동

React를 활용하여 마이페이지 및 관리자 페이지 구축

AWS S3를 활용해 대용량의 이미지 데이터 저장 가능

RESTful API 아키텍처 도입으로 클라이언트와 서버를 분리

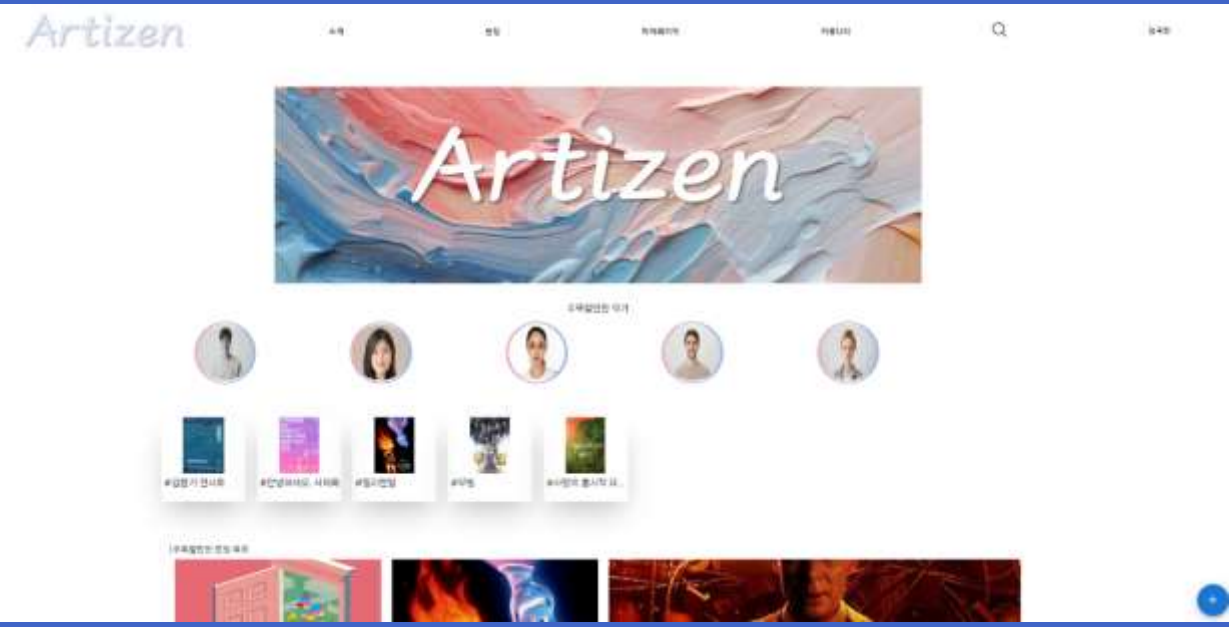
React의 사용으로 높은 반응성과 그에 따른 사용자적 서비스를 구축

프로젝트 기획부터 ERD 설계, 백엔드 및 프론트 개발, 단위테스트까지 경험함으로써 전체적인 진행흐름을 알 수 있었음

Spring boot의 사용법, 구조, 흐름 등 프레임 워크 사용 역량 강화

Java를 활용함으로써 객체 지향 프로그래밍(OOP) 역량 강화

MyBatis의 활용으로 SQL 매핑 프레임워크 사용 역량 강화



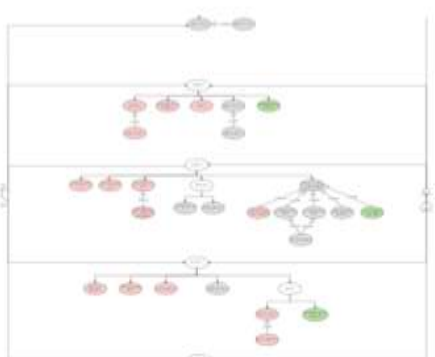
메인 화면



ERD : 정규화 과정 및 다:다 관계 해소를 통해 총 13개의 테이블이 작성되었습니다.



Front UI : miro.com 에서 UI설계를 진행했습니다.



Use Case Diagram : 초기 개발시에 사용자 관점에서 어떠한 기능이 필요한지 작성했습니다.



API 명세서 : 노선을 통해 작성하고 있으며, 포스팅 기능을 통해 팀원들과 지속적으로 소통하며 작업하고 있습니다.

작업 문서

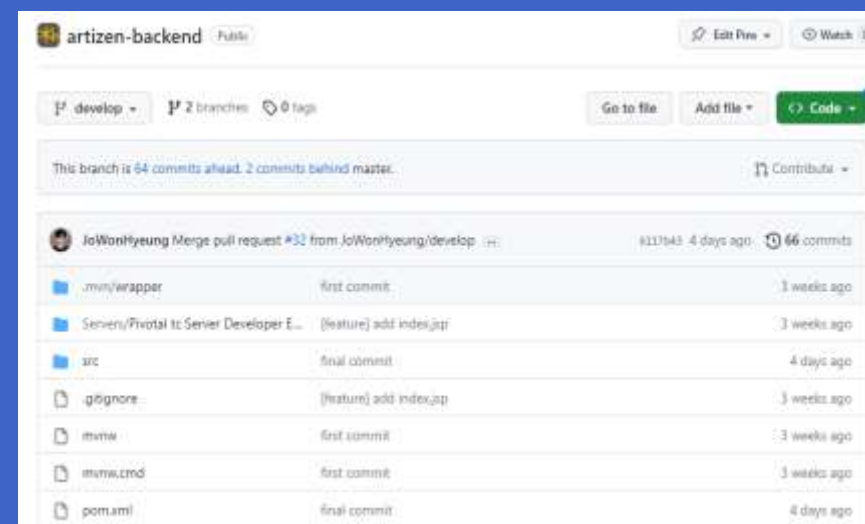
3. PROJECTS

백엔드



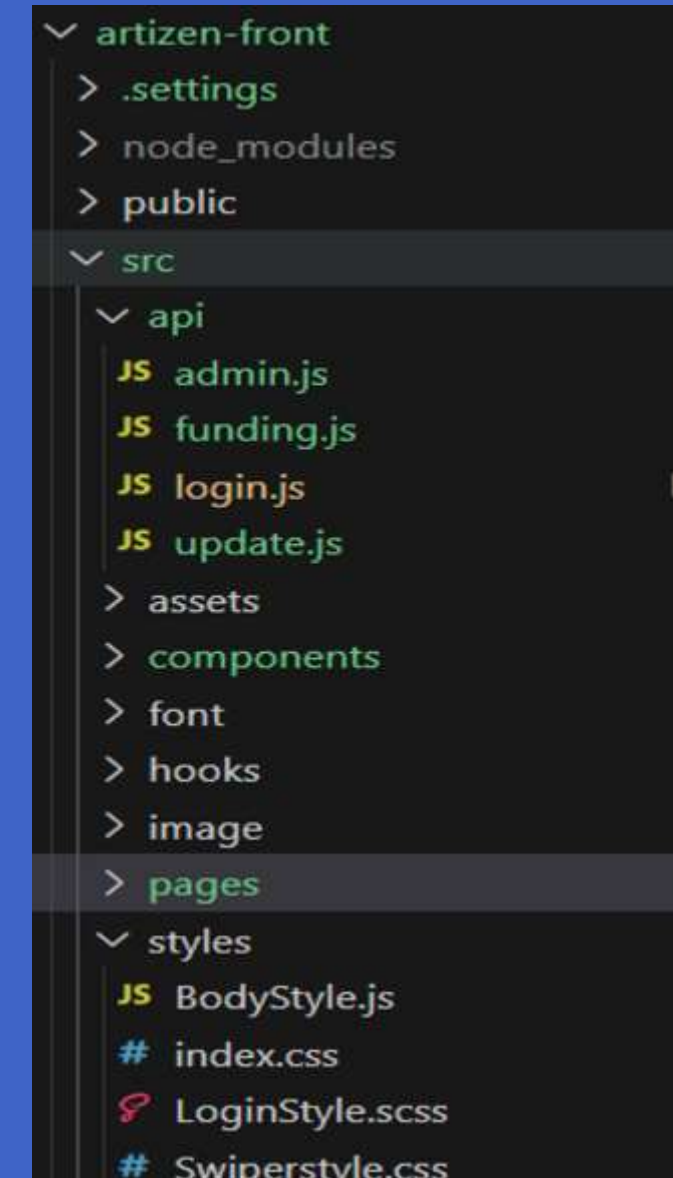
파일 구조

- ◆ Service Layer, Persistence Layer으로 나누어 관심사를 분리
- ◆ 반복적인 DB 접근 코드를 피하기 위해 MyBatis를 사용하여 DB에 접근
- ◆ React를 사용하였기 때문에 Restful 기반 서비스를 개발
- ◆ 또한, Layer 계층 별로 테스트 코드를 항상 작성하려고 합니다.



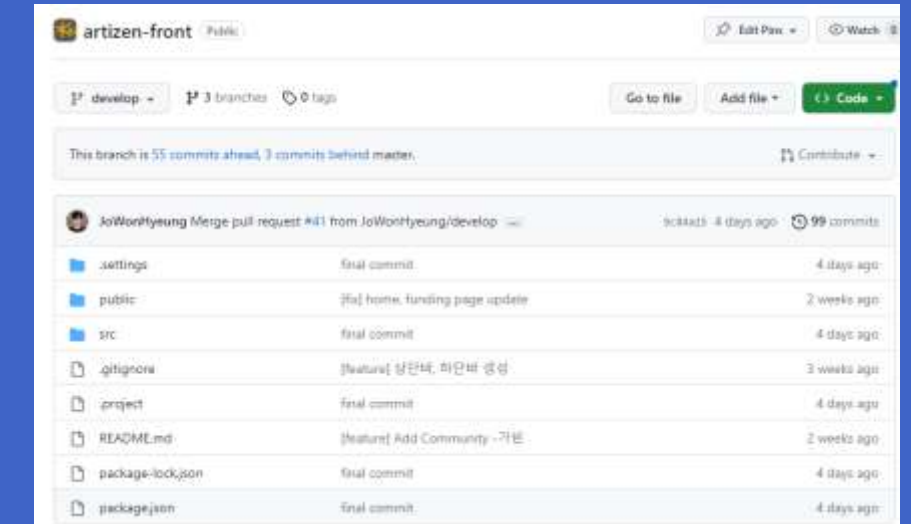
Backend 코드 관리를 위한 Github

프론트



파일 구조

- ◆ 반복적으로 작성되는 코드를 줄이기 위해 팀원들과 함께 Component 제작해 공유하여 사용
- ◆ Spring과의 통신을 위해 axios 및 fetch 라이브러리를 사용



Frontend 코드 관리를 위한 Github

3. PROJECTS

02. 카드 실적 분석 시스템

소스코드: https://github.com/KB-MiniProject/Card_Service

프로젝트 개요

사용자가 다양한 회사의 각 카드로 결제를 하면 해당 지출 내역을 기반으로 간단한 분석을 수행

기술 스택

Java, Oracle

나의 역할

조회 기능 개발(회사, 카테고리, 기간별 카드 사용내역 등)

구매내역을 활용하여 연관 분석 기능 개발

결과

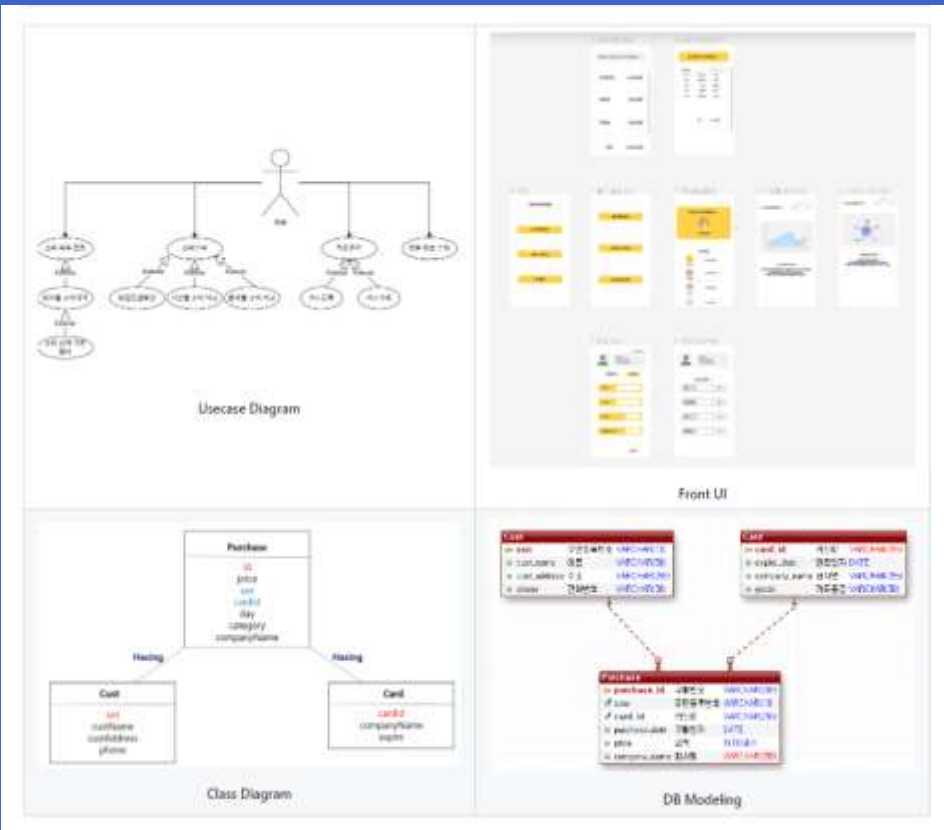
연관 분석 기능을 통해 특정 상품을 구입한 고객이 어떤 상품을 추가로 구매하는지 알아냄

Customer와 Card 간의 다대다 관계를 Purchase 테이블을 사용해서 해결

배운점

다대다 관계를 해소하는 테이블을 자바 클래스 내부에 필드로 선언함으로써 재사용성과 유지보수성이 높은 코드를 작성함. 이를 통해, OOP 역량 강화

변수 및 함수 네이밍 규칙들을 적용하였고, 오버로딩을 적극적으로 활용함. 이를 통해 가독성 높은 코드 작성 역량 강화



작업 문서

```
public interface ServiceDAO {
    void buy(String ssn, String cardId,String companyName, String category, int price, String date) throws SQLException; //결제
    ArrayList<Purchase> getPurchase(String ssn) throws SQLException;
    ArrayList<Purchase> getPurchaseCompanyDetails(String companyName, String ssn) throws SQLException; //세부 회사 내역(모든 이력)
    Map<String, Integer> getPurchaseByCompany(String ssn) throws SQLException; //
    Map<String, Integer> getPurchaseByPeriod(String ssn) throws SQLException; //기간별 분석
    Map<String, Integer> getPurchaseByCategory(String ssn) throws SQLException; //카테고리별 소비 분석
    ArrayList<Purchase> getPurchaseByFiltering(String companyName,String ssn, int criteria, boolean flag) throws SQLException; //필터
    int getTotalAmount(String ssn) throws SQLException; //총합계
    void checkGrade(String companyName, String ssn, String name) throws SQLException;
    ArrayList<Purchase> getPurchase() throws SQLException;
    void printAlgorithm() throws SQLException;
}
```

네이밍 규칙을 적용한 함수

3. PROJECTS

03. 이지팜

소스코드: <https://github.com/JoWonHyeung/easyfarm>

프로젝트 개요

딥러닝 모델을 활용한 노지작물 병해충 플랫폼 구축

기술 스택

Language : Python
Backend : Docker, AWS S3, FastAPI, Docker, Jenkins, GCP

나의 역할

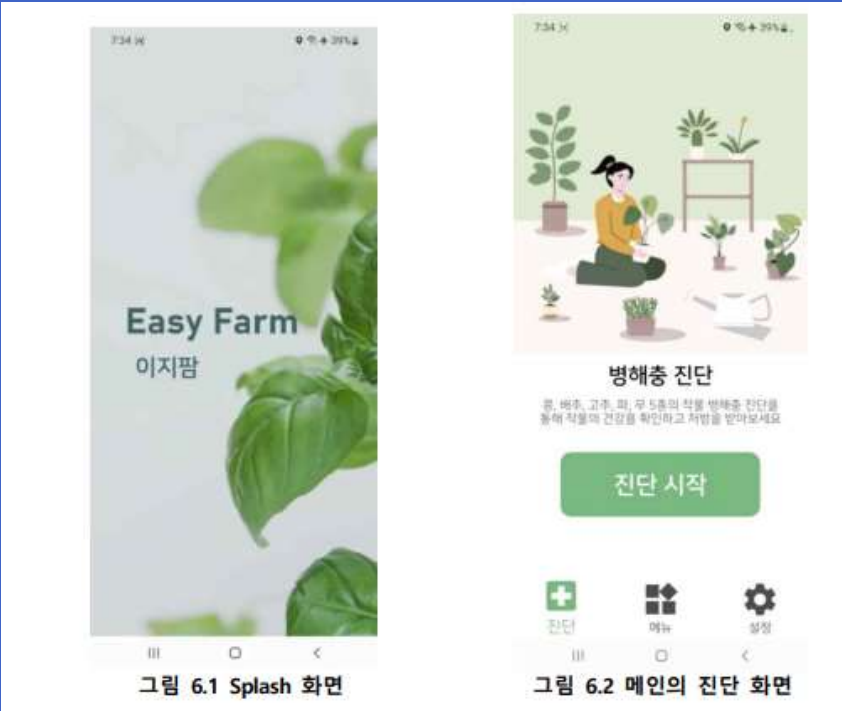
병해충 탐지를 위한 딥러닝 모델 개발 – EfficientNet 모델 사용
모델 서빙 서버 구축
Android – FastAPI 서버간 통신
효율적인 개발을 위해 Jenkins를 사용하여 CI/CD 파이프라인 구축
학습 목적으로 Grafana, Prometheus를 활용하여 모니터링 시스템을 구축

결과

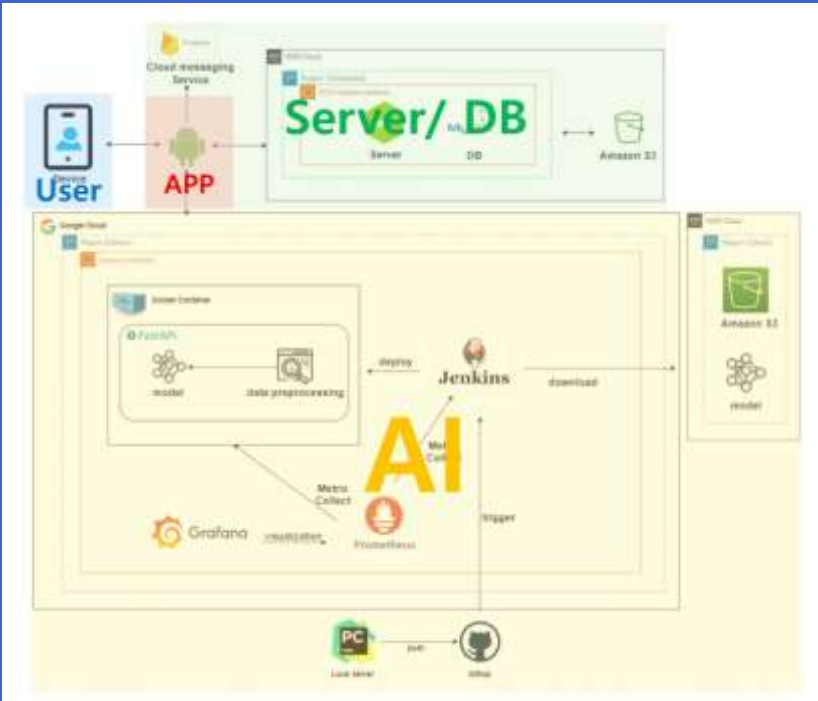
EfficientNet 모델을 Fine tuning 시킴으로써 정확도를 98%까지 달성
Jenkins 파이프라인을 구축하면서 1,500분의 시간을 절약

배운점

UML, ERD, UI 등 개발 전 문서 작업의 중요성을 느낄 수 있었음
오픈소스 및 클라우드 서비스 사용 역량 강화
Python의 다양한 프레임워크 및 라이브러리 사용 역량 강화

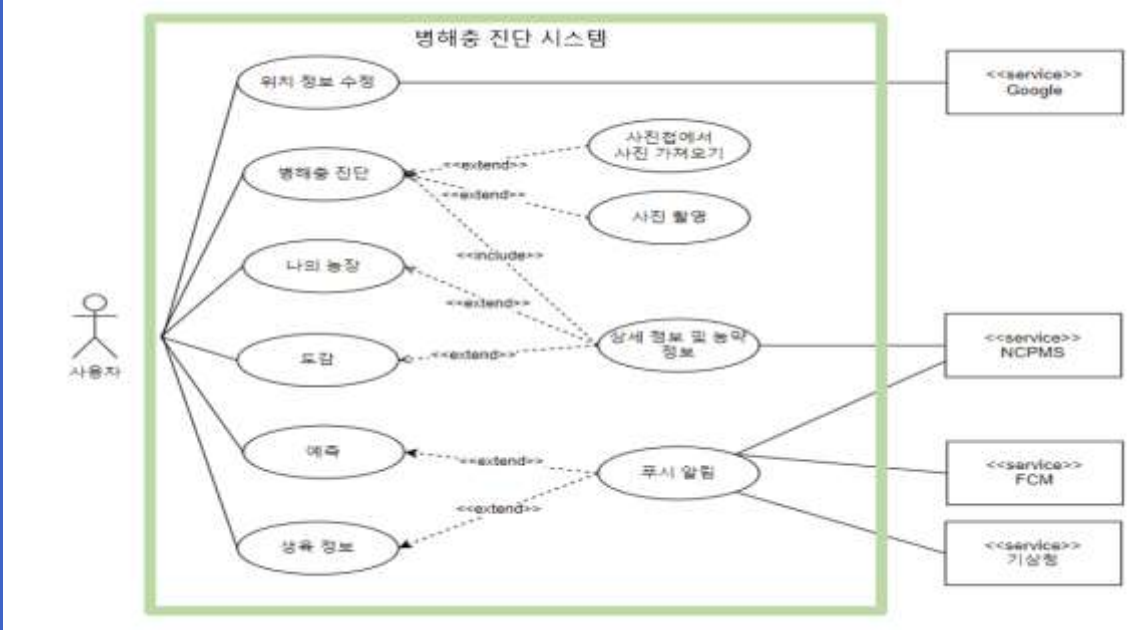


메인 화면

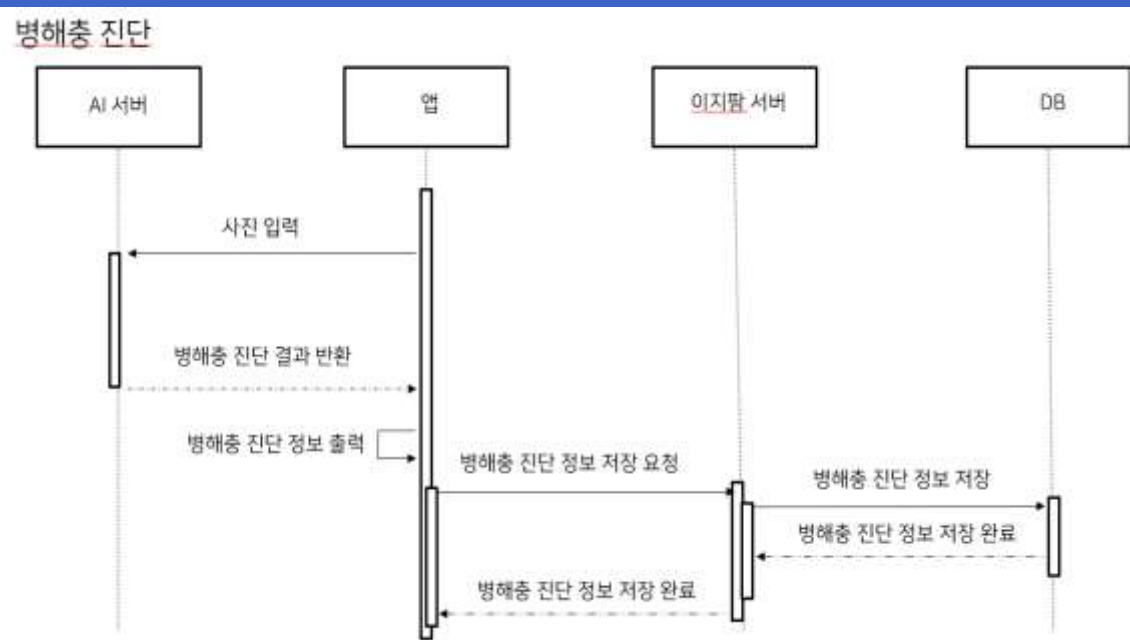


아키텍처

3. PROJECTS



유스케이스 다이어그램



시퀀스 다이어그램

◆ 기존에는 문서화 작업을 전혀 수행하지 않았습니다. 그러나 이로 인해 팀원들 간의 이해도 차이가 발생하여 기능 구현 과정에서 어려움을 겪었습니다.

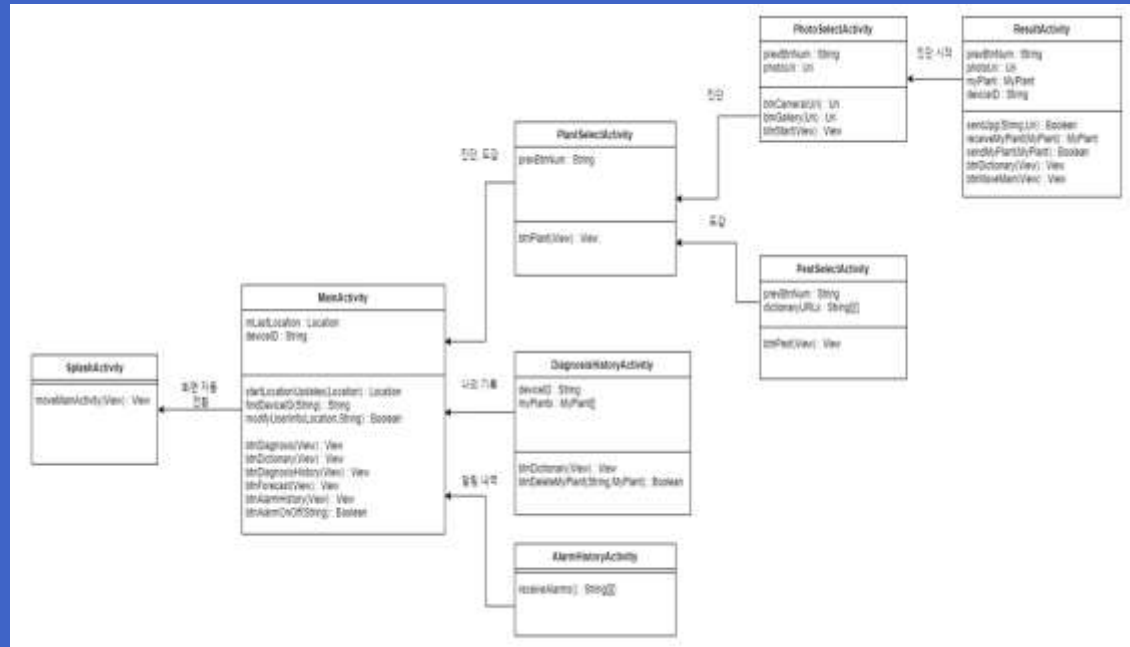
◆ 이러한 부분을 해결하고자 모든 기능 구현 및 회의 내용을 문서화하여 팀원들과 함께 공유하였습니다.

◆ 그 결과, 팀원들 사이의 생각을 일치시킬 수 있었고 작업 중 발생하는 이슈를 크게 줄일 수 있었습니다.

◆ 이러한 깨달음을 통해, 현재 모든 프로젝트에서는 문서화 작업을 필수적으로 진행하고 있습니다.

2. 시나리오 - 병해충 진단	
행위자	사용자
유스케이스 명	병해충 진단
유스케이스 내용	업로드 된 사진을 기반으로 병해충 진단한다.
사전 조건	사진을 업로드 한다.
사후 조건	진단하기 버튼을 누른다.
기본 흐름	1. 진단을 원하는 카테고리 버튼 클릭 2. 사진을 업로드 후 진단하기 버튼 클릭 3. 진단 결과 출력
예외 흐름	E1. 식별 불가능한 사진을 업로드 한 경우
대안 흐름	A1. '진단이 불가능한 사진입니다.' 출력

유스케이스 다이어그램을 기반으로 한 시나리오 작성



클래스 다이어그램

4.입사 후 계획



1. AWS 자격증 취득



많은 프로젝트에서 클라우드 서비스를 사용해본 경험자로서 앞으로 클라우드 시장은 점차 더 커질 것으로 생각합니다.

이에 입사 후에 AWS 자격증을 취득함으로써 클라우드 사용 역량을 높이하고자 합니다.



2. 사이드 프로젝트 진행



사이드 프로젝트를 진행함으로써 새로운 기술, 언어, 라이브러리를 학습하고 실험해보고자 합니다.

이러한 경험은 회사가 지속적으로 최신 트렌드를 쫓아갈 수 있도록 도움을 줄 것입니다.

잘부탁드립니다😊

THANK YOU

.....

jwh.dev0103@gmail.co

m

010-3649-3276