PORTFOLIO

끊임없이 성장하는 개발자

손 현 석

Tel: 010-9144-2948

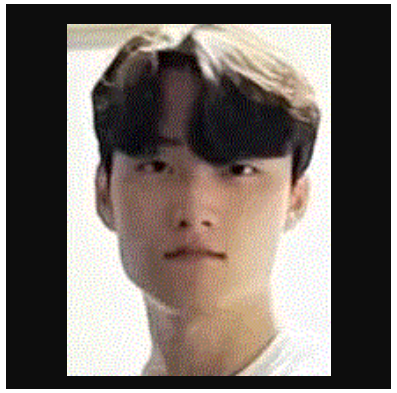
Email: snrnakat@naver.com

Github: hyeonseok0912

Blog: https://blog.naver.com/snrnakat

CONTENTS

자기소개



Son Hyeon Seok

안녕하세요. 끊임없이 성장하는 개발자 손현석 입니다.

다양한 인턴 및 프로젝트 경험

제조 및 항공 분야 회사에서의 2회의 인턴 경험을 통해 사회성과 효율성의 중요성에 대해 느끼고 체득했습니다. 또한 앱 개발 프로젝트를 시작으로 위성 통신 프로그램을 거쳐 웹 개발 및 API 연동 프로젝트까지 이어지는 다양한 경험을 통해 다양한 파트에 대한 관심과 그를 기반으로 한 역량을 지니고 있습니다.

교육 기관 이수를 통한 스텝업

대학 전공때 했던 프로젝트 경험만으로는 다양한 기술 스택을 활용해보지 못해서 아쉽다 판단했습니다. 그래서 개인 역량도 함양시키고 프로젝트도 더 해보고자 싶어서 독학과 동시에 교육기관을 알아보았습니다. 마침내 국비지원 JAVA 기반 풀스택 개발자 과정을 통해 각종 언어와 프레임워크, 기술 스택 등에 대해 익히면서 한층 더 개발자가 되기 위해 나아가는 중입니다.

항상 배우고 성장하려는 의지

처음부터 코딩, 나아가 프로그래밍에 대해 자신이 있었던 것은 아니었습니다. 오히려 생소한 언어들과 프로그래밍은 그를 주저하게 만들었습니다. 하지만 효율성을 추구하고 새로운 것에 흥미를 많이 느끼는 저였기에 프로젝트를 하면서 마주했던 문제들은 되려 저를 끊임없이 노력하게 만들었고, 그 결과 눈부신 성장을 이룰 수 있었습니다.

경험

손 현 석

EDUCATION

- 2016. 03 경상대학교 항공우주소프트웨어공학과 전공
- 2023. 11 ~ 2024. 05 국비지원 자바(JAVA)기반 풀스택 개발자 취업과정 이수

LICENSE

- 정보처리기사 (2023. 06 취득)

CAREER

- 2017. 04 ~ 2019. 03 트리엔㈜ 근무
- 2021. 01 ~ 2021. 01 한국항공우주연구원 인턴
- 2021. 06 ~ 2021. 08 한국표면처리㈜ 인턴

EXPERIENCE

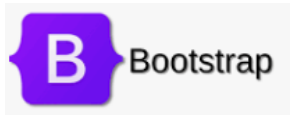
- 교내 프로젝트 참여(앱 개발)
- 한국항공우주연구원 위성 통신 프로그램 참여
- 교내 프로젝트 종합설계 참여
- 지리 정보 시스템 프로젝트I 참여
- 대학 상담 서비스 개발 프로젝트 참여

Skill Set

- **Language** JAVA, JavaScript, Kotlin

- **OS** Ubuntu 22.04 LTS

- **Framework + Library** JQuery, Ajax, Spring, MyBatis, Bootstrap



- **Database** Oracle, MariaDB, PostgreSQL, MySQL

- **Development Tool** eGovFrameDev, Visual Studio, Docker


- **ETC** JSP, Thymeleaf, Git, Apache Tomcat, AWS

참여 프로젝트 목록

프로젝트명	프로젝트 기간	설명	담당역할
영수증을 부탁해	2021. 09 ~ 2021. 12	소비 기한을 사용하여 식품을 관리하는 서비스를 제공, 불필요한 지출 및 식품 폐기물 감소	품목 추출 및 OCR 전처리 담당
SDR 통신 실습	2022. 01 ~ 2022. 01	위성 신호를 수신하여 해당 데이터 값을 이미지 파일로 변환	YAGI 안테나 제작 및 신호해석 툴인 GNURadio를 이용하여 GNURadio Components 로직 제작
버드스트라이크 방지 드론 제작	2022. 03 ~ 2022. 06	비행체의 비행 시의 위험 감지 및 후속 상황에 대한 예방 기술 개발	드론 제작 참여 및 조류 인식 프로그램 구현
지리 정보 시스템 프로젝트 (학원 실무 프로젝트1)	2024. 03 ~ 2024. 04	도시 및 지역의 공간 단위에서 전력 사용량을 파악함으로써 탄력적인 에너지 효율화를 실현하기 위함이다	선택 지역의 전력 사용량 표출 및 범례 구현, 사용자의 데이터 DB에 삽입, 시/도 별 전력 사용량 차트화
대학 상담 서비스 (학원 실무 프로젝트2)	2024. 04 ~ 2024. 05	코로나 19 팬데믹 이후 학생들의 정신 건강을 위한 상담 서비스 개발	집단 상담 및 메인화면 담당 AWS S3를 이용하여 이미지 업로드 기능 구현 프로젝트 성과 발표 담당

7. 주요 포트폴리오

1. 대학 상담 서비스

   (https://github.com/hyeonseok0912/TeamD_Project_Wizian.git)

2. 지리 정보 시스템 프로젝트

   (https://github.com/hyeonseok0912/project_SD.git)

3. 영수증을 부탁해

   (https://github.com/hyeonseok0912/SW_development.git)

(1) 대학 상담 서비스

CHOONGANG UNIVERSITY EST. 2024

- 프로젝트 기본정보

프로젝트명	대학 상담 서비스
프로젝트기간	2024. 04. 22 ~ 2024. 05. 28
프로젝트인원	5명
설명	코로나 19 팬데믹 이후 상담 수요 증가에 따른 학생들의 정신건강을 위한 대학 상담 서비스 개발
담당역할	그룹상담 및 메인페이지 풀스택과 AWS를 활용한 이미지 업로드 기능 개발

[그림1. 상담 페이지 메인화면]

- 개발 주요사항
  - MariaDB RDBMS로 프로젝트 서비스 구현을 위한 데이터베이스 구축
  - MyBatis를 사용하여 Mapping
  - Java를 사용하여 DB에서 요청에 따라 필요한 데이터를 담음
  - AJAX를 사용하여 각 상담 별 하위 메뉴들의 내용들을 비동기적으로 url 이동 없이 빠르게 이동하여 불필요한 UX 제거
  - Datepicker 및 Timepicker API를 활용하여 날짜 및 시간 입력
  - AWS와 연동하여 ImageUpload 기능을 통해 AWS S3에 업로드 후 표시
  - 배포를 위한 EC2 서버에 프로젝트 업로드
  - Git과 Gitfork를 통해 팀원 간의 협력 및 코드 리뷰

[그림2. Datepicker 활용 화면]

(2) 대학 상담 서비스

- 기술 스택

OS	Windows 11
Develop Tool	STS4
Data Base	MySQL 8.4.0 / MongoDB 7.0.9
Language	Java, JavaScript
Java Version	Java 17.0.10
Framework & Library	Spring boot version (3.2.5) MyBatis3 / JPA / Thymeleaf json / Lombok (1.18.24) / jquery
Server	AWS EC2, S3, RDS
Collaborative Software	Github, Gitfork, Google Sheets(Excel)

- 설명

각종 상담에 대한 시연 영상입니다.

- https://www.youtube.com/watch?v=B7ktlxAldrc

AWS EC2를 이용해 실제 배포한 웹 주소입니다.

- http://3.37.218.174/

[그림3. 담당 파트 메인화면]

(2) 대학 상담 서비스

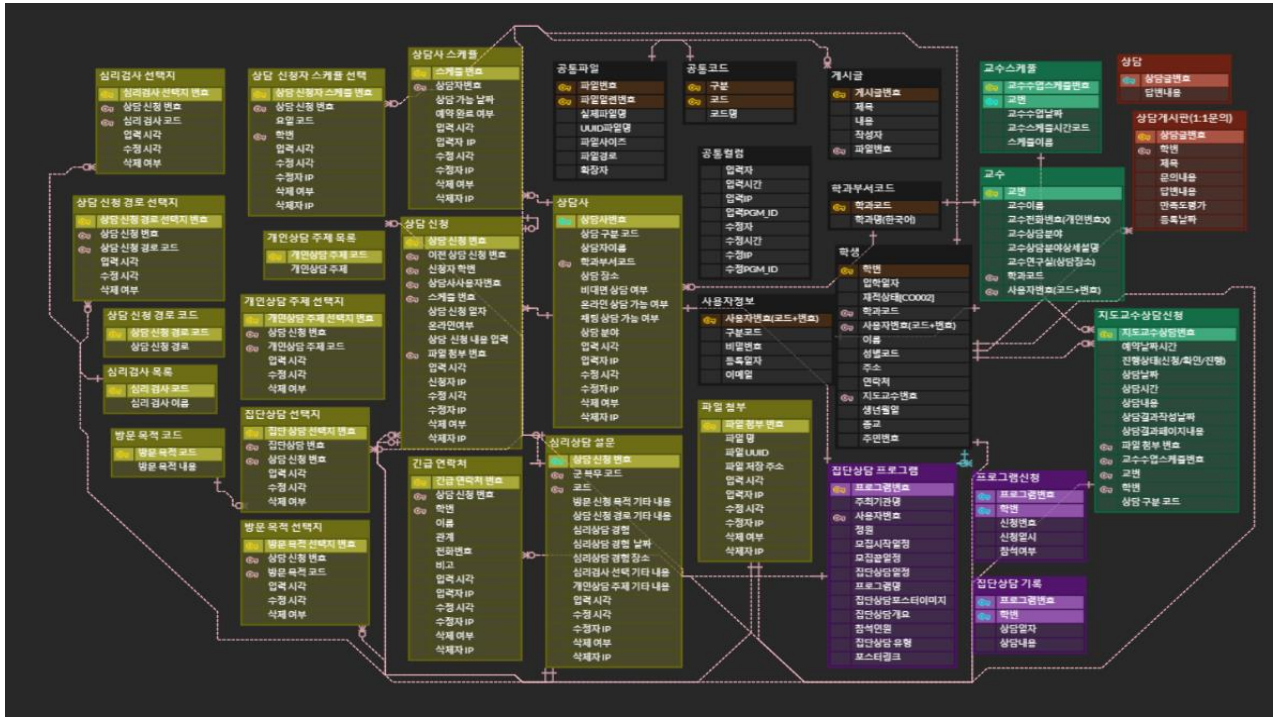
CHOONGANG UNIVERSITY
EST. 2024

- 요구사항 정의서

요구사항 ID	요구사항명	기능 ID	기능명	상세 설명	제약 사항	비고
MEM01	로그인	MEM01_LOGIN01	로그인	모든 기능에 로그인 필요.	실제 DB에 존재 해야하며 등록된 유저번호와 패스워드가 일치해야 한다	아이디가 존재하지 않으면 '올바르지 않은 정보입니다.'를 표시한다. 학생, 교수, 상담사 전부 같은 곳에서 로그인한다.
MEM01	로그아웃	MEM01_LOGOUT01	로그아웃	모든 기능에 로그아웃 필요	미 로그인 시 로그아웃 버튼은 표시 되지 않는다	
MEM02	학생 기본 정보	MEM02_INFO01	학생 정보 조회	이름, 학번, 생년월일, 나이, 성별, 소속(학과 학부), 연락처(휴대폰 번호), 이메일, 재학종류(재학, 휴학, 졸업, 수료), 휴학기간을 확인할 수 있다.	연락처와 이메일만 수정 가능하며 나머지 정보는 학생정보처를 방문하도록 한다	
		MEM02_INFO02	비밀번호 변경	비밀번호를 변경한다.	내 정보를 수정하려면 비밀번호 재입력이 필요하다. 새 비밀번호 변경 시 4자리 이상이어야 하며 공백은 안된다. 새 비밀번호 입력란 2개가 서로 일치해야한다	
MEM04	교수/상담자 정보	MEM03_INFO01	교수/상담자 정보 출력	이름, 소속(상담자는 소속기관 교수는 학부/학과) 연락처(휴대폰 번호), 이메일 이 나와야 한다.		
		MEM03_INFO02	교수/상담자의 상담 내역 조회	자신이 진행한 상담 내역을 조회한다.	숫자가 많으면 페이징이 필요하다. 페이징 개수를 10개를 선택 가능할 수 있어야 한다.	MEM5를 호출한다.
		MEM03_INFO03	비밀번호 변경	비밀번호를 변경한다.		
MEM05	교수/상담자 상담 정보	MEM05_INFO01	교수/상담자 상담 목록 표시	교수/상담자의 지금까지의 모든 상담 목록을 리스트업한다.		본인의 상담 정보만 열람 가능
		MEM05_INFO02	상담 예약 신청서 조회	학생의 상담 신청서를 조회한다.		
		MEM05_INFO03	상담 수락	교수/상담자는 수락해서 상담을 확정한다.		
		MEM05_INFO03	상담 취소	교수/상담자가 상담을 취소한다.	일단 상담이 수락되면 교수/상담자만 취소가 가능하다. 취소시 취소 사유를 기입해야 한다.	내담자 쪽에서 예약 확정이 되었을 시 취소를 하고싶으면 상담자 프로필에서 연락처를 보고 따로 연락해서 상담자쪽에서
MNG01	관리자	MNG01_MNG01	관리자 권한 조회 기능	모든 상담 기록 목록을 조회 가능하다. 이름, 학번, 소속(학과 학부)로 상담 기록 목록을 검색 가능하다.		
		MNG01_MNG02	관리자 권한 상담 현황 변경	상담 기록의 상담 신청 중/ 예약 대기/ 상담 취소 를 변경 가능하다.		

[그림4. 초기 설계시 정의서]

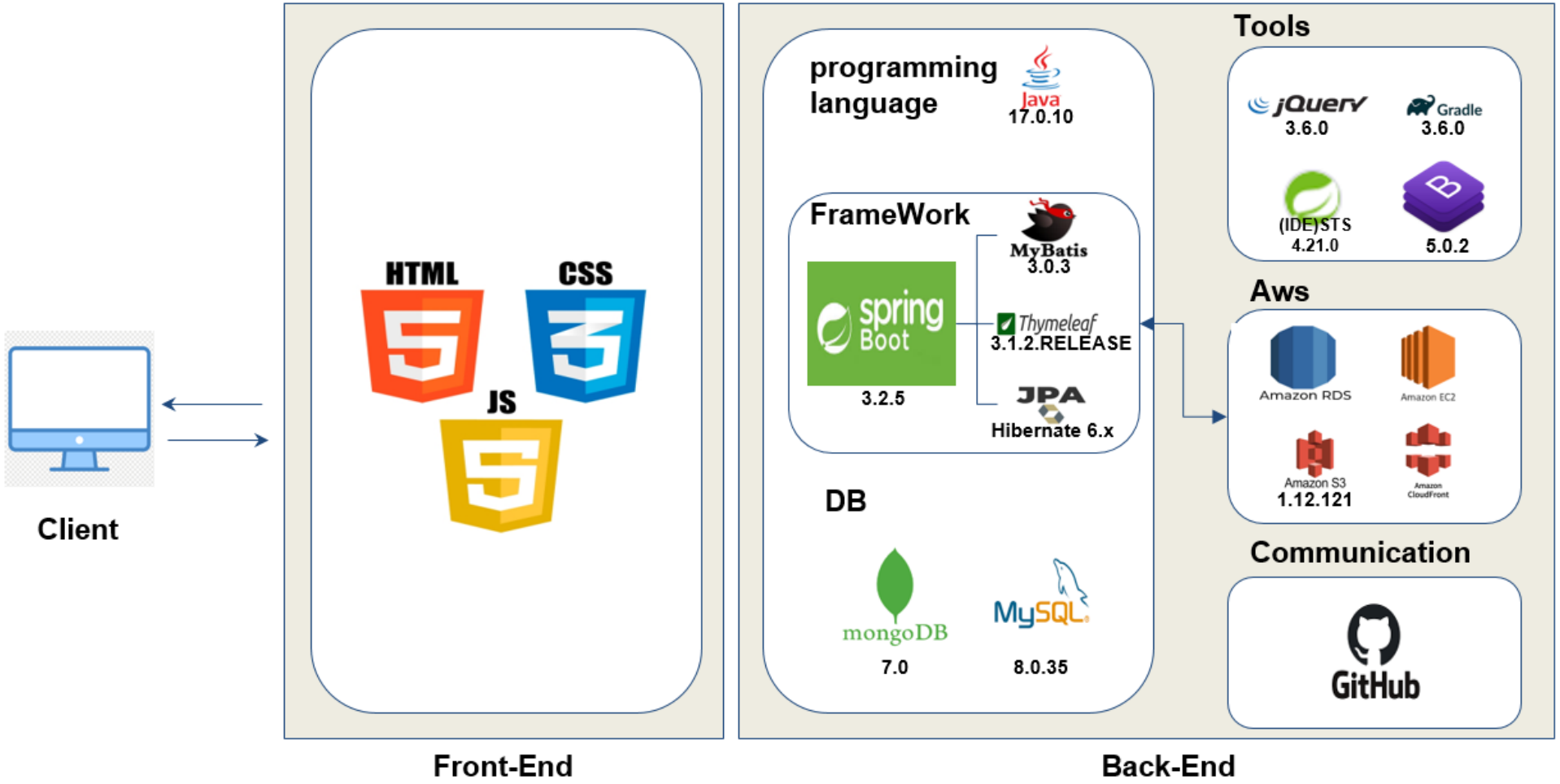
- ERD (Entity Relationship Diagram)



[그림5. DB 설계를 위한 ERD]

(2) 대학 상담 서비스

- 개발환경 및 개발언어



(2) 대학 상담 서비스

- 트러블슈팅

다음은 개발 시 어려움을 겪었던 부분입니다.

이미지 파일을 첨부 후 글을 등록했을 때 Swal.fire 문을 통해 정상적으로 파일이 등록되는 것을 볼 수 있었고, 실제 AWS S3 스토리지에도 이미지가 등록되었습니다. 하지만 콘솔 창에서는 service endpoint 라는 에러가 발생하였습니다.

gradle 추가 및 applicaton.properties에도 구글링에서 찾은 각종 해법들을 넣어 보았지만 제대로 적용되지 않았습니다. 하지만 공식 사이트에서 이것은 버전 문제로 현재 제가 사용하고 있는 버전은 2.2.6 버전으로 최선 버전이었는데, 이러한 error코드가 모두 다 표시되는 버전임을 확인했습니다. 그래서 applicaton.properties에 아래와 같은 문구를 추가해 콘솔창에 에러문구를 지워주었고, 이후 아래와 같이 정상적으로 실행하는 것을 볼 수 있었습니다.

또한 대용량 데이터를 처리하기 위하여 AWS S3 스토리지를 활용하면서 사소한 어려움이 많았는데, 바로 권한 설정을 통한 액세스 허용과 차단, 그리고 지역 설정을 해주어야 한다는 것이었습니다. 여기서 region 이라고 적힌 곳에 따라 스토리지에서 이미지를 가져와 웹에서 띄워줄 때, 지역 변수명 또한 적어줘야 한다는 점은 사소하지만 좋은 배움이 되었습니다.

[그림6. endpoint 에러 해결 과정]

[그림7. 이미지 url 찾는 과정]

(2) 대학 상담 서비스

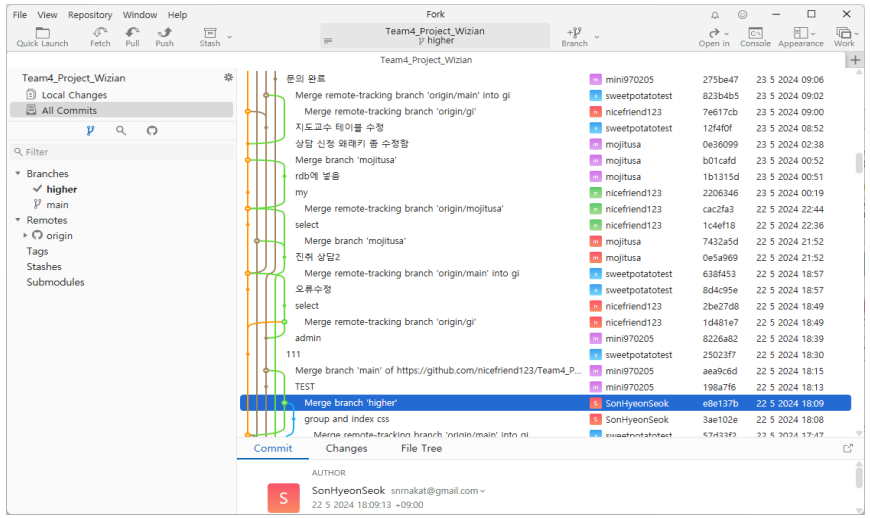
CHOONGANG UNIVERSITY EST. 2024

- 느낀 점(회고)

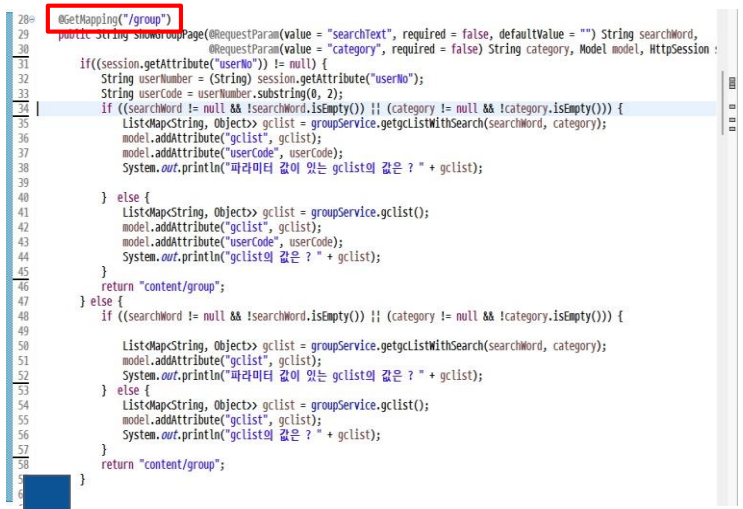
이번 프로젝트에서는 여러 가지 중요한 점이 많았습니다. 먼저 요구사항 정의부터 ERD에 걸쳐 테스트에 이르기까지 시스템의 중요성을 알 수 있었습니다.

또한, 저는 팀원 간의 코드 리뷰가 매우 인상적이었는데, 코드 리뷰를 통해 효율적으로 코드를 작성할 수 있구나를 느낄 수 있었습니다.

마지막으로, 대용량 데이터를 처리할 때 DB의 경량화 및 효율화를 위한 고민 후 AWS S3 스토리지를 개인적으로 공부하여 활용한 것은 스스로 학습할 수 있는 역량을 쌓는데 중요한 발판이 되었습니다.



[그림8. 프로젝트를 위해 쓰인 Fork 기록]



```
@GetMapping("/group")
public String showGroupPage(@RequestParam(value = "searchText", required = false, defaultValue = "") String searchWord,
                            @RequestParam(value = "category", required = false) String category, Model model, HttpSession
    if((session.getAttribute("userNo")) != null) {
        String userNumber = (String) session.getAttribute("userNo");
        String userCode = userNumber.substring(0, 2);
        if ((searchWord != null && !searchWord.isEmpty()) || (category != null && !category.isEmpty())) {
            List<Map<String, Object>> gclist = groupService.getgcListWithSearch(searchWord, category);
            model.addAttribute("gclist", gclist);
            model.addAttribute("userCode", userCode);
            System.out.println("파라미터 값이 있는 gclist의 값은 ? " + gclist);

        } else {
            List<Map<String, Object>> gclist = groupService.gclist();
            model.addAttribute("gclist", gclist);
            model.addAttribute("userCode", userCode);
            System.out.println("gclist의 값은 ? " + gclist);
        }
        return "content/group";
    } else {
        if ((searchWord != null && !searchWord.isEmpty()) || (category != null && !category.isEmpty())) {

            List<Map<String, Object>> gclist = groupService.getgcListWithSearch(searchWord, category);
            model.addAttribute("gclist", gclist);
            System.out.println("파라미터 값이 있는 gclist의 값은 ? " + gclist);
        } else {
            List<Map<String, Object>> gclist = groupService.gclist();
            model.addAttribute("gclist", gclist);
            System.out.println("gclist의 값은 ? " + gclist);
        }
        return "content/group";
    }
}
```

```
@GetMapping("/group")
public String showGroupPage(@RequestParam(value = "searchText", required = false, defaultValue = "") String searchWord,
                            @RequestParam(value = "category", required = false) String category, Model model, HttpSession

    if((session.getAttribute("userNo")) != null) {
        String userNumber = (String) session.getAttribute("userNo");
        String userCode = userNumber.substring(0, 2);
        model.addAttribute("userCode", userCode);
    }

    List<Map<String, Object>> gclist = null;
    if ((searchWord != null && !searchWord.isEmpty()) || (category != null && !category.isEmpty())) {
        gclist = groupService.getgcListWithSearch(searchWord, category);
    } else {
        gclist = groupService.gclist();
    }
    model.addAttribute("gclist", gclist);

    return "content/group";
}
```

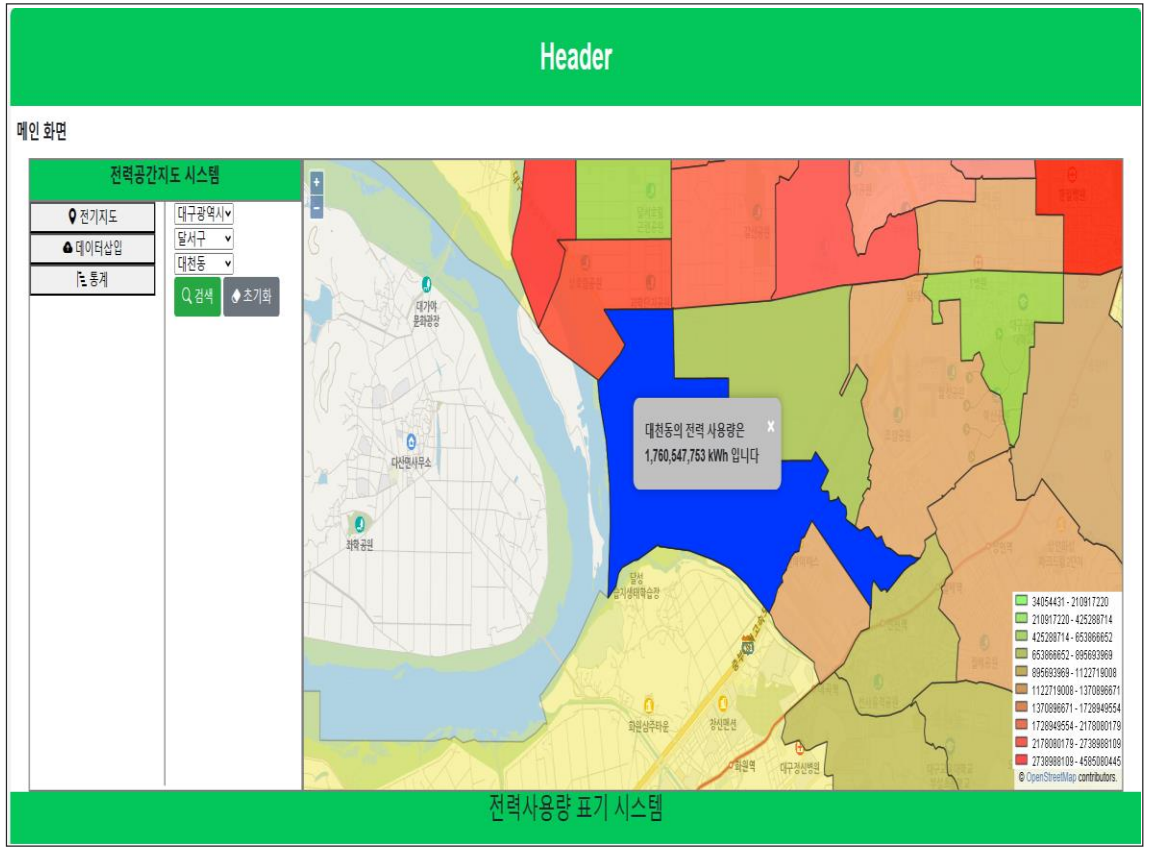
[그림9.코드 리뷰를 통한 코드량 간소화]

(2) 지리 정보 시스템 프로젝트

- 프로젝트 기본정보

프로젝트명	학원 실무 프로젝트1
프로젝트기간	2024.03 ~ 2024.04
프로젝트인원	5명
설명	배경 지도 표출, 시도/시군구/범례 선택 시 지도 확대 및 레이어 표출 레이어 선택 시 해당 레이어의 전력 사용량 조회 팝업, 파일 업로드 기능, 업로드 상태에 따른 스낵바, 시도/시군구 별 통계 그래프 표출
담당역할	전기 지도(레이어, 팝업, 필터링) 및 통계 부분 담당

- 개발 주요사항
  - MyBatis 활용 및 SQL Query를 실행하여 DB로부터 필요한 값을 Map 형식으로 받아 Json 타입으로 jsp에 넘겨줌
  - Vworld API를 활용한 배경지도 표출
  - GeoServer와 QGIS를 사용하여 지도 위에 레이어 표출
  - PostgreSQL DBMS로 프로젝트 서비스 구현을 위한 데이터베이스 구축
  - Map 객체를 통해 데이터 전송
  - JSP에서 HTML, CSS, JavaScript, jQuery, Bootstrap 등을 활용해 화면 구축
  - Modal, Swal, SweetAlert 등 화면에 필요한 디자인적 요소를 사용해 더욱 편리하고 깔끔한 화면 구성



[그림1. 레이어 및 팝업 표시 화면]

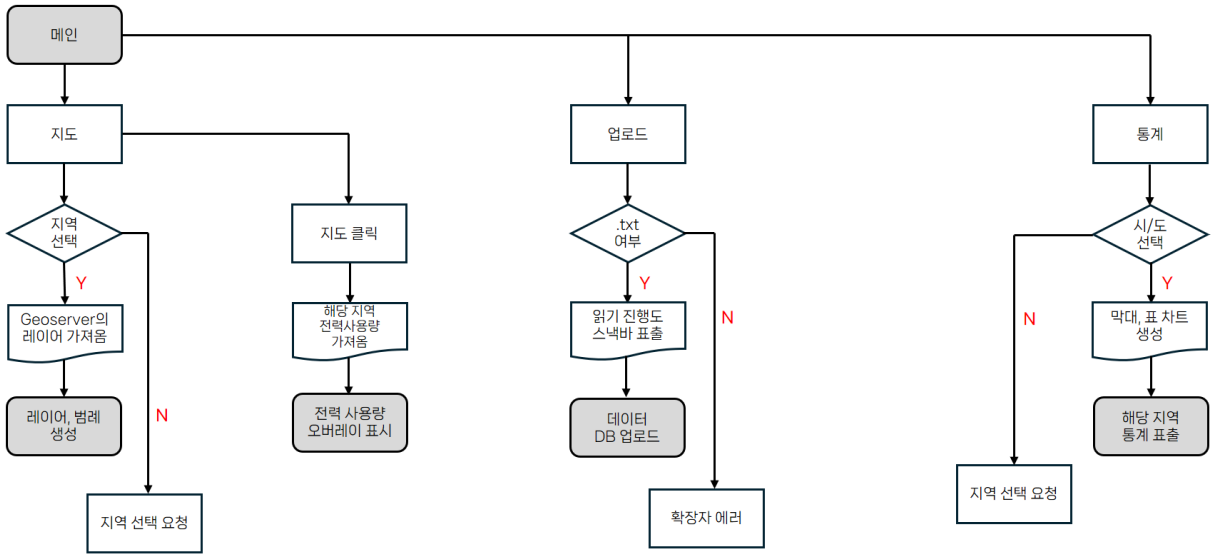
(2) 지리 정보 시스템 프로젝트

- 기술 스택

OS	Windows 11
Develop Tool	GovFrameDev (3.10.0)
Data Base	PostgreSQL(4.1)
Language	Java, JavaScript
Java Version	JavaSE-1.8
Framework & Library	GovFrameDev (3.10.0) / Mybatis (3.5.6) / JSON / Lombok (1.18.39) / jQuery
Web Application Server	Apache Tomcat 9.0.
Collaborative Software	Github, Google Sheets(Excel)

첫 번째는 지역 선택과 범례 기능,
두 번째는 파일 업로드 기능,
세 번째는 통계 기능입니다.

각 기능에 대한 시연 영상입니다.
전기 지도 : https://youtu.be/ga0Gd0-Vmm4
파일 업로드: https://youtu.be/F4KwH3iUoLs
통계 : https://youtu.be/uRxiRkiKIhw



[그림2. 실무 프로젝트 화면 흐름도]

(2) 지리 정보 시스템 프로젝트

- 트러블슈팅(1)

JSON data type에서의 문제

우측 코드는 옵션바에서 시도를 선택하면, 시도값을 AJAX를 통해 Controller로 보내주고 해당 시도 이름값을 가진 시군구 데이터 값을 map 형식으로 jsp에 보내주고, jsp에서는 그 시군구 값을 JSON 형식으로 받아 사용하는 것입니다.

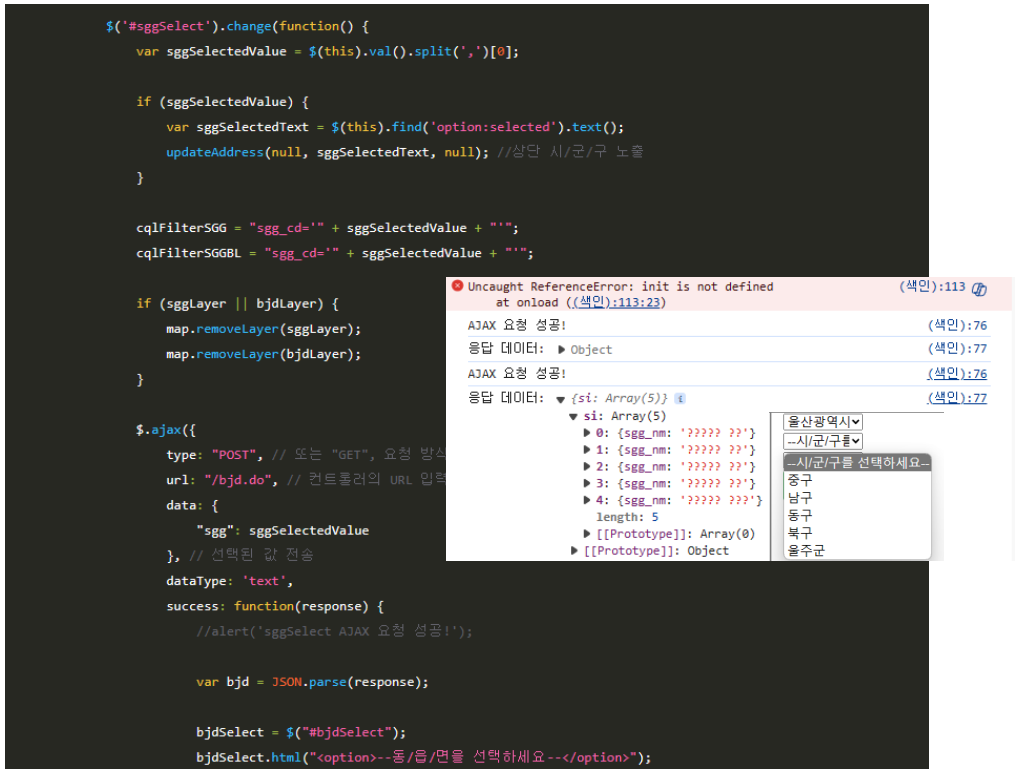
하지만 해당 부분을 개발할 때, 시군구 데이터 값을 map 형식으로 받아내는 것까지는 System.print문을 통해 확인했으나, 계속해서 웹 상에서 우측 이미지와 같이 시군구의 이름이 ???로 뜨는 것이 문제였습니다.

이는 jsp에서 JSON 으로 받을 때 문제가 발생한 것이었고, 이 부분을 해결하기 위해 처음에는 언어가 맞지 않아서 그런가 싶어 JSON, controller 등 맞춰줄 수 있는 모든 부분에 대해 charset을 모두 UTF_8로 맞춰주었습니다. 하지만 문제는 해결되지 않았습니다.

이번에는 controlle에서 map 형식을 다시 JSON 타입으로 변환해서 jsp로 보내주는 Jackson 및 Gson library도 활용했습니다. 하지만 여전히 해당 문제는 해결되지 않았습니다.

이 문제를 해결하기 위해 구글링을 계속 했고, Spring MVC가 @Controller에 요청을 보내기 위해 필요한 HandlerMapping과 HandlerAdapter를 bean으로 등록하고, 이렇게 등록된 bean에 의해 요청 url과 컨트롤러를 매칭할 수 있다는 사실을 발견했습니다. 그래서 pom.xml 파일에 아래 문구를 추가해주고 나니 정상적으로 JSON 데이터를 ???가 아닌 정상적인 값으로 받을 수 있었습니다.

```
<mvc:annotation-driven/>
```



[그림3. 실무 프로젝트 AJAX 통신]

(2) 지리 정보 시스템 프로젝트

- 트러블슈팅(2)

CORS 충돌 문제

다음은 지도의 좌표 클릭 시 해당 부분의 좌표값을 통한 지역 이름과 해당 지역의 전력사용량을 나타내는 팝업 기능을 개발할 때 발생한 문제입니다.

여기서 발생한 문제는 CORS 충돌이었는데, 보안을 강화하기 위해 다른 출처의 정보는 막기 위해 존재하는 스템이 바로 CORS 였습니다.

하지만 저는 기존에 이러한 개념을 알지 못했던 상태였습니다. GeoServer의 port번호는 8080이고, 제 프로젝트의 port번호는 80이었기 때문에 80 포트에서 8080포트의 정보를 가져오려고 할 때 CORS 충돌이 났었습니다.

이를 해결하기 위해 서버측 에서 허용할 출처를 헤더의 Access - Control - Allow - Origin 에 기입해주었습니다. 하지만 이 방법으로 해결되지 않았고, 다른 방법을 시도해보기로 했습니다.

이번에는 proxy 서버를 이용해 모든 출처를 허용해주는 것으로 해결해보려고 했으나, 이 역시 GeoServer 에서 값을 가져오는데 실패했습니다.

제가 CORS 문제를 해결한 방법은 다음과 같습니다.
C:\apache-tomcat-9.0.84\webapps\geoserver\WEB-INF 내에 있는 web.xml 파일에서 CORS에 대해 허용해 주게끔 코드를 변경해 주었고, 이후 CORS 충돌을 해결할 수 있었습니다.

```
//클릭 이벤트 리스너 설정
map.on('singleclick', function(evt) {
    // 클릭한 지점의 좌표를 가져옴
    var coordinate = evt.coordinate;

    // 해당 좌표에서의 지리적 정보를 가져오는 요청을 서버에 보냄
    var featureRequest = new ol.format.WFS().writeGetFeature({
        srsName: 'EPSG:3857',
        featureNS: 'http://localhost:8080/geoserver/cite',
        featurePrefix: 'cite',
        featureTypes: ['shinjinview23'],
        outputFormat: 'application/json',
        geometryName: 'geom',
        filter: new ol.format.filter.Intersects('geom', new ol.geom.Point(coordinate))
    });

    // 서버에 요청 보내기
    fetch('http://localhost:8080/geoserver/cite/wms', {
            method: 'POST',
            body: new XMLSerializer().serializeToString(featureRequest)
        })
        .then(function(response) {
            return response.json();
        })
        .then(function(json) {
            // 가져온 정보에서 단계 구분 값을 추출하여 팝업에 표시
            if (json.features.length > 0) {
                var properties = json.features[0].properties;
                var sgg_pu = properties['usage'];
                var sgg_cd = properties['sgg_cd'];
                var sgg_nm = properties['sgg_nm'];
```

```
// 서버에 요청 보내기
fetch('http://localhost:8080/geoserver/cite/wms',
```

Access to fetch at 'http://localhost:8080/geoserver/cite/wms' from origin 'http://localhost' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource. If an opaque response serves your needs, set the request's mode to 'no-cors' to fetch the resource with CORS disabled.
sido.do:457

```
<!-- Uncomment following filter to enable CORS in Tomcat. Do not forget
<filter>
  <filter-name>cross-origin</filter-name>
  <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
  <init-param>
    <param-name>cors.allowed.origins</param-name>
    <param-value>*</param-value>
  </init-param>
  <init-param>
    <param-name>cors.allowed.methods</param-name>
    <param-value>GET,POST,PUT,DELETE,HEAD,OPTIONS</param-value>
  </init-param>
  <init-param>
    <param-name>cors.allowed.headers</param-name>
    <param-value>*</param-value>
  </init-param>
</filter>
```

[그림5. web.xml 파일 코드]

localhost:8080/geoserver/web/?2

≠

Port Name	Port Number
Tomcat admin port	8006
HTTP/1.1	80

[그림4. 포트 번호 불일치]

(2) 지리 정보 시스템 프로젝트

- 느낀 점(회고)

이번 프로젝트는 게시판을 만들면서 배웠던 것 외에 처음 접하는 라이브러리나 툴, 프로그램들로 진행되면서 다양한 경험을 쌓을 수 있었는데, 배운 점이 많아 만족스러웠습니다.

우선 DB 사용에 있어 많은 어려움을 겪었는데, 지리 정보를 가지고 있는 칼럼에서 좌표를 뽑아내기 위해 SQL 함수 Geom을 다루면서 ST_함수에 대해서 공부해볼 수 있는 계기가 되었으며, DB의 데이터 값들을 받아오는 과정에서 테이블 간 join 기능을 썼을때 오래 걸렸기에,이를 해결하기 위해 materialized view를 쓰면서 select문의 처리 시간을 단축시키기 위해 고민 하는 데 많은 시간을 할애해보면서 프로젝트의 보여지는 기능 뿐 아니라 처리 속도와 데이터의 효율화 또한 중요하다는 것을 느끼게 되었습니다.

그리고 파일 업로드 과정에서 pagesize를 지정해주면서 대용량 데이터를 처리하는데 있어 속도와 안정성을 올리기 위한 고민도 해볼 수 있었습니다.

마지막으로 직접 개발 계획서, 보고서를 만들어보면서 프로그래밍을 하는데 필요한 문서 작업에 대한 경험을 해볼 수 있었습니다.

IT 프로젝트 개발 계획서

(「전력 사용량 지도 구축」)

2024. 04. 16.

훈련과정명 (소속)	프로젝트 기반 자바(JAVA) 응용 SW 개발자 취업과정 (중앙정보기술인재개발원)
팀 명	TEAM E
팀장 성명	손현석
팀원 성명	김지훈
팀원 성명	노재희
팀원 성명	이지은
팀원 성명	정진수
팀원 성명	채영선
지도교사	윤승현

3. 프로젝트 개발일정

작 업 명	담당자	개발 일정							비고
		1 W	2 W	3 W	4 W	5 W	6 W	7 W	
1. 프로젝트 기획	공통								
- 주제 선정	-								03.19 ~ 03.20
- 기존 앱 분석	-								
- 구현기능 구상	-								
- 업무 분장	-								
2. 기획 및 설계	-								03.20 ~ 03.27
- DB 설계	-								
- 개발환경 설정 - 와이어프레임 제작	- -								
- 기획서 작성	-								
- 흐름도 제작	-								
- ERD 제작	-								
3. 개발	-								03.25 ~ 04.05
- 기능구현	-								
- UI 구현/디자인	-								
- 추가 기능 개발	-								
4. 테스트/오류수정	-								04.01 ~ 04.11
- 테스트	-								
- 오류수정	-								

[그림6. 실무 프로젝트 보고서 중]

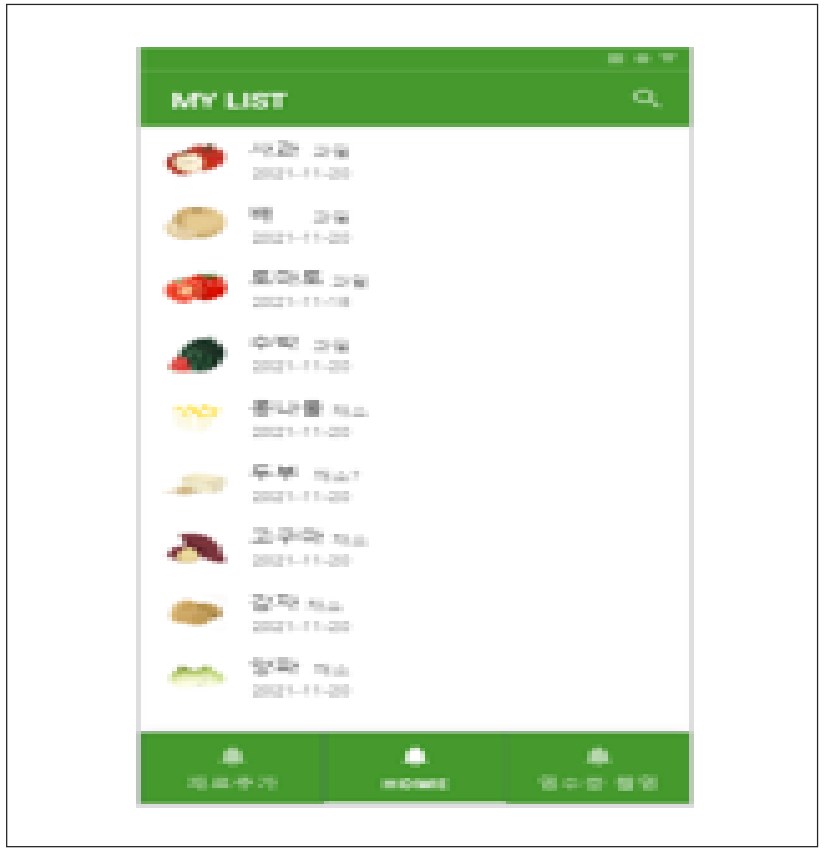
(2) 영수증을 부탁해 영부

- 기술 스택
  - Kotlin
  - Android Studio
  - Data Format(JSON)
  - Open CV
  - OCR(Optical Character Recognition)

- 설명

  국내에서는 2023년부터 유통기한이 아닌 소비기한을 표시합니다. 소비기한은 보관 조건을 지키면서 식품을 소비하면 안전에 이상이 없는 기한을 의미합니다.

  저희는 이런 소비기한을 사용하여 식품을 관리하는 서비스를 제공함으로써 불필요하게 버려지는 식품 폐기물과 음식물 쓰레기를 줄이고, 보관 기간이 길어짐에 따라 불필요한 지출 또한 최소화하는 것을 목표로 개발을 진행하였습니다.



[그림1.영부 앱서비스 III 화면]

(2) 영수증을 부탁해

- 품질 시나리오(시스템의 작업 속도 3초 이내)

```
private fun launchImageCrop(uri: Uri): Bitmap? {
    CropImage.activity(uri)
        .setGuidelines(CropImageView.Guidelines.ON)
        .setCropShape(CropImageView.CropShape.RECTANGLE)
        .start( activity: this)
    return loadBitmap(uri)

}
```

```
CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE -> {
    val result = CropImage.getActivityResult(data)
    if (resultCode== Activity.RESULT_OK) {
        var start = System.currentTimeMillis()
        result.uri?.let{uri ->
            val bitmap = loadBitmap(uri)
            binding.imagePreViewOCR.setImageURI(uri)

            if (bitmap != null) {
                imageExtract(bitmap.copy(Bitmap.Config.ARGB_8888, isMutable: true))
            }
        }
        binding.imagePreViewOCR.setImageURI(result.uri)
        var end = System.currentTimeMillis()
        Log.d( tag: "수행시간", (end-start).toString())
    }
    else if (resultCode == CropImage.CROP_IMAGE_ACTIVITY_RESULT_ERROR_CODE){
        Log.e(TAG,  msg: "Crop error: ${result.error}")
    }
}
```

[그림2. 영수증을 부탁해 코드]

IMAGECROP 기능 추가

속도 측면에서의 요구사항을 만족시키기 위해 IMAGECROP 기능을 추가하였습니다.

IMAGECROP은 촬영한 이미지를 잘라서 원하는 부분만 OCR의 INPUT으로 사용하는 방법입니다.

실제 테스를 진행한 결과 이미지를 자르지 않고 텍스트를 추출했을 때는 약 3000~4000MS이 소요되었지만, 필요한 부분의 이미지만 잘라서 텍스트를 추출했을 때는 약 2000MS 줄어든 것을 볼 수 있습니다.

```
2021-12-20 15:49:56.111 3591-3591/com.example.receipt D/수행시간: 3
2021-12-20 15:50:20.172 3591-3591/com.example.receipt D/수행시간: 1
2021-12-20 15:51:30.534 14489-14489/com.example.receipt D/수행시간: 3830
2021-12-20 15:51:50.670 14489-14489/com.example.receipt D/수행시간: 1947
2021-12-20 15:54:06.901 14489-14489/com.example.receipt D/수행시간: 3506
2021-12-20 15:54:17.255 14489-14489/com.example.receipt D/수행시간: 1928
2021-12-20 15:54:37.145 14489-14489/com.example.receipt D/수행시간: 3003
2021-12-20 15:54:47.282 14489-14489/com.example.receipt D/수행시간: 1338
```

(2) 영수증을 부탁해

- 트러블슈팅

다음은 개발 시 어려움을 겪었던 부분입니다.

OCR를 활용하여 영수증의 문자를 추출한 결과, 추출된 문자의 정확도가 그리 높지 않았습니다. 하지만 저희의 목표는 정확도 90% 이상이었기 때문에, 해당 요구사항을 만족시키기 위해 노력했습니다.

먼저 OCR 기능의 정확도를 높이기 위해, Jtessboxeditor을 사용하여 직접 학습 데이터를 만들어서 ocr을 학습시켰습니다. 하지만, 기존의 OCR 기능이 최신화가 되어있어 오히려 정확도가 더 낮아지는 것을 볼 수 있었습니다.

이에 저희는 이 외 방법인 Grayscale, 이진화, 기울기 보정 방법 등을 통해 OCR로 문자 추출시 정확도를 15% 가량 올릴 수 있었습니다.

[그림3. 영수증을 부탁해 테스트 결과 이미지]

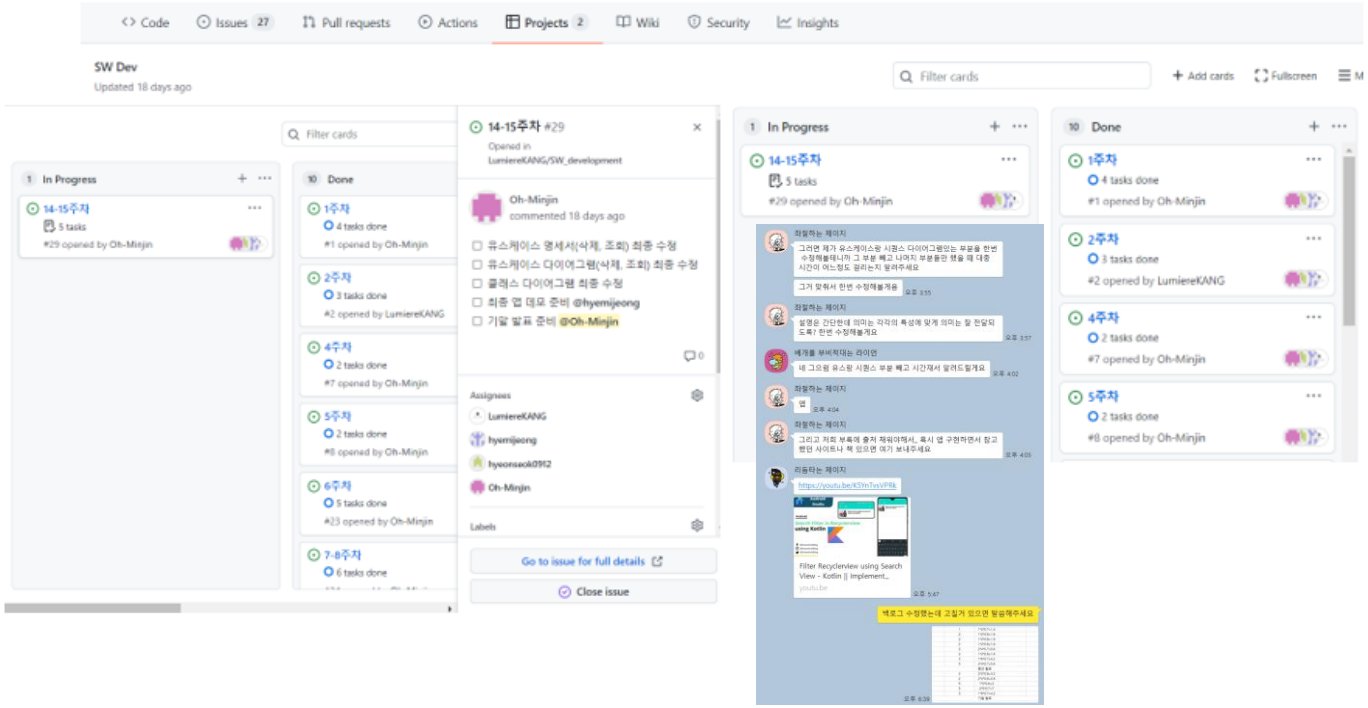
(2) 영수증을 부탁해

- 느낀 점(회고)

저희 조는 매주 수업이 끝난 이후 비대면 DR 대면 회의를 통해 발표 당시 받은 피드백을 반영하여 개발 상황을 조정하였고, 해당 주차의 과제를 수행 및 역할 분담, 각자 맡은 개발 진행 상황 브리핑 등을 수행하였습니다.

또한, 회의록 작성자를 지정하여 매주 회의했던 내용을 정리하고 각자의 진행상황을 한눈에 파악할 수 있도록 깃허브의 칸반 보드를 활용하였고, 원활한 소통을 위해 카카오톡 단체 채팅방을 사용하여 의견 조율을 하였습니다.

칸반 보드를 활용함으로써 서로 맡은 사항이 어디까지 진행되었는지, 각자가 진행해야 할 과제가 무엇인지 잘 파악할 수 있었고 계획대로 개발이 큰 어려움 없이 잘 진행되었습니다.



[그림4. 영수증을 부탁해 프로젝트 단톡방]

END

손 현 석

Tel: 010-9144-2948

Email: snrnakat@naver.com

Github: hyeonseok0912

Blog: https://blog.naver.com/snrnakat