



4월 3주차 회의록 @2023년 4월 21일

📌 팀	MAC
👥 참석자	① 한결 민 ② 상연 안 ③ 최필환
📍 장소	원흥관 3층 아리수
🕒 시간	14:00-18:00

주요 안건

- 일정 수립
- 브랜치 설정
- 진행 사항
- 회의 결과

회의 내용


1. 일정 수립

- 5월 2주차 까지의 일정 수립

	4월 3주차	4월 4주차	5월 1주차	5월 2주차
Front	템플릿 레이아웃/색상 변경	템플릿 레이아웃/색상 변경	팀 구성 프로세스 구현	공지사항 프로세스 구현
Back	로그인 / AWS 세팅	로그인 / AWS 세팅	팀 구성 / 파일 스토리지	공지사항 / 파일 스토리지

1. **프론트엔드**: 기능 별로 html 페이지 만들어 구현, thymeleaf 사용하여 백엔드 연동
 - 4월 3~4주차: 로그인, 과목 선택, 팀 활동 영역에 맞춰 템플릿 레이아웃 및 색상 변경하기
 - 5월 1주차: 팀 구성 (+버튼 → 새 창 → 학생 선택 → 완료 버튼 → 팀 활동 페이지)
 - 5월 2주차: 공지사항 (+버튼 → 새 창 → 입력 → 팀 활동 페이지 / 삭제 버튼 → 자동 새로고침)
2. **백엔드**
 - 4월 3~4주차: 오픈소스 분석 및 필요 부분 선정/ AWS 세팅/ 로그인 기능 구현
 - 5월 1주차: 팀구성 구현/ 파일 스토리지 업로드 다운로드 구현
 - 5월 2주차: 공지사항 구현/ 커밋 메세지 및 업데이트 구현

2. 브랜치 설정

Default branch		
main	Updated 2 days ago by thisishwan2	Default
 Your main branch isn't protected Protect this branch from force pushing or deletion, or require status checks before merging. Learn more		
Your branches		
Hanyku1	Updated 5 minutes ago by roberniro	0 6
Active branches		
Hanyku1	Updated 5 minutes ago by roberniro	0 6
Sangyeon	Updated yesterday by ahnup	0 8
hwan	Updated 2 days ago by thisishwan2	4 10

- 개인별 브랜치 생성
- 개발 사항 개별 브랜치의 커밋/푸시 후 코드 리뷰

3. 진행 사항

1. 현재 프로젝트 진행사항
 - a. 로그인, 과목 선택 인터페이스 - 안상연
 - a-1. 로그인 인터페이스

오픈소스 코드 수정

- 로그인 화면 좌측에 동국대학교 이미지 배치

```
<body class="bg-gradient-primary">
  <div class="container">
    <!-- Outer Row -->
    <div class="row justify-content-center">
      <div class="col-xl-10 col-lg-12 col-md-9">
        <div class="card o-hidden border-0 shadow-lg my-5">
          <div class="card-body p-0">
            <!-- Nested Row within Card Body -->
            <div class="row">
              <!-- 로그인 화면 좌측에 동국대학교 이미지 배치 -->
              <div class="col-lg-6 d-none d-lg-block bg-login-image"></div>
```

- 로그인 화면 우측에 로고 이미지, 텍스트, 입력창, 버튼 배치

```

<!--로그인 화면 우측에 동국대 로고, 텍스트, 학번/패스워드 입력창, 로그인 버튼 배치-->
<div class="col-lg-6">
  <div class="p-5">
    <div>
      <!--동국대학교 이미지(명진관) 배치-->
      
    </div>
    <!--이클래스임을 설명하는 텍스트 배치-->
    <h1 class="h5 text-gray-900 mb-4">동국대학교 이클래스</h1>
    <form class="user">
      <!--학번 입력창-->
      <div class="form-group">
        <input
          type="email"
          class="form-control form-control-user"
          id="exampleInputEmail"
          aria-describedby="emailHelp"
          placeholder="student ID"
        />
      </div>
      <!--비밀번호 입력창-->
      <div class="form-group">
        <input
          type="password"
          class="form-control form-control-user"
          id="exampleInputPassword"
          placeholder="Password"
        />
      </div>
      <!--저장 여부 체크-->
      <div class="form-group">
        <div class="custom-control custom-checkbox small">
          <input
            type="checkbox"
            class="custom-control-input"
            id="customCheck"
          />
          <label class="custom-control-label" for="customCheck">Remember Me</label>
        </div>
      </div>
      <!--로그인 버튼-->
      <a
        href="index.html"
        class="btn btn-primary btn-user btn-block"
      >
        Login
      </a>
    </form>
  </div>
</div>
<hr />

```

- 배경 색상 #f59c00, 버튼 색상 #ef9457으로 변경

```

.bg-gradient-primary {
  background-color: #f59c00;
  background-image: linear-gradient(180deg, #f59c00 10%, #f59c00 100%);
  background-size: cover;
}

```

```

.btn-primary {
  color: #fff;
  background-color: #ef9457;
  border-color: #ef9457;
}

```

로그인 페이지 결과

a-2. 과목 선택 인터페이스



오픈 소스 코드 수정

- 과목 선택 화면 좌측에 로고, 텍스트, 버튼 배치

```

<body class="bg-gradient-primary">
  <div class="container">
    <!-- Outer Row -->
    <div class="row justify-content-center">
      <div class="col-xl-10 col-lg-12 col-md-9">
        <div
          class="card o-hidden border-0 shadow-lg my-5"
          style="height: 478.2px"
        >
          <div class="card-body p-0">
            <!-- Nested Row within Card Body -->
            <div class="row" style="height: 100%; margin-right: 0">
              <!--과목 선택 화면 좌측에 로고, 사용자 정보, 마이페이지 버튼, 로그아웃 버튼 배치-->
              <div class="col-lg-6">
                <div class="p-5 py-5">
                  <!--동국대 로고 배치-->
                  
                  <!--이클래스임을 설명하는 텍스트 배치-->
                  <h1 class="h5 text-gray-900 mb-4">동국대학교 이클래스</h1>
                  <div class="p-1">
                    <!--사용자 이름, 학번 정보-->
                    <span
                      class="mr-2 d-none d-lg-inline text-gray-800 small border-right-0"
                      >김동국(2023000001)님</span>
                    <!--마이페이지 버튼-->
                    <a
                      href="index.html"
                      class="btn btn-primary btn-user"
                      style="width: 55%"
                    >
                      My Page
                    </a>
                    <!--로그아웃 버튼-->
                  </div>
                  <div class="p-1 py-3">
                    <a
                      href="index.html"
                      class="btn btn-primary btn-user btn-block"
                    >
                      Logout
                    </a>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

- 과목 선택 화면 우측에 텍스트, 과목 선택 버튼 배치

```

<!--과목 선택 화면 우측에 과목 선택 섹션 배치-->
<div class="p-3 col-lg-6" style="height: 100%">
  <div
    class="p-1 border border-secondary"
    style="height: 100%; border-radius: 10%"
  >
    <div class="p-5">
      <!--과목 선택 섹션임을 알려주는 텍스트 배치-->
      <h1 class="h5 text-gray-900 mb-4">내 강의실</h1>
      <!--과목 선택 버튼-->
      <form class="user">
        <div class="form-group">
          <a
            href="index.html"
            class="btn btn-primary btn-user btn-block"
          >
            오픈소스소프트웨어실습
          </a>
        </div>
        <div class="form-group">
          <a
            href="index.html"
            class="btn btn-primary btn-user btn-block"
          >
            오픈소스소프트웨어프로젝트
          </a>
        </div>
        <div class="form-group">
          <div class="custom-control custom-checkbox small">
            <input
              type="checkbox"
              class="custom-control-input"
              id="customCheck"
            />
          </div>
        </div>
      </form>
    </div>
  </div>
</div>

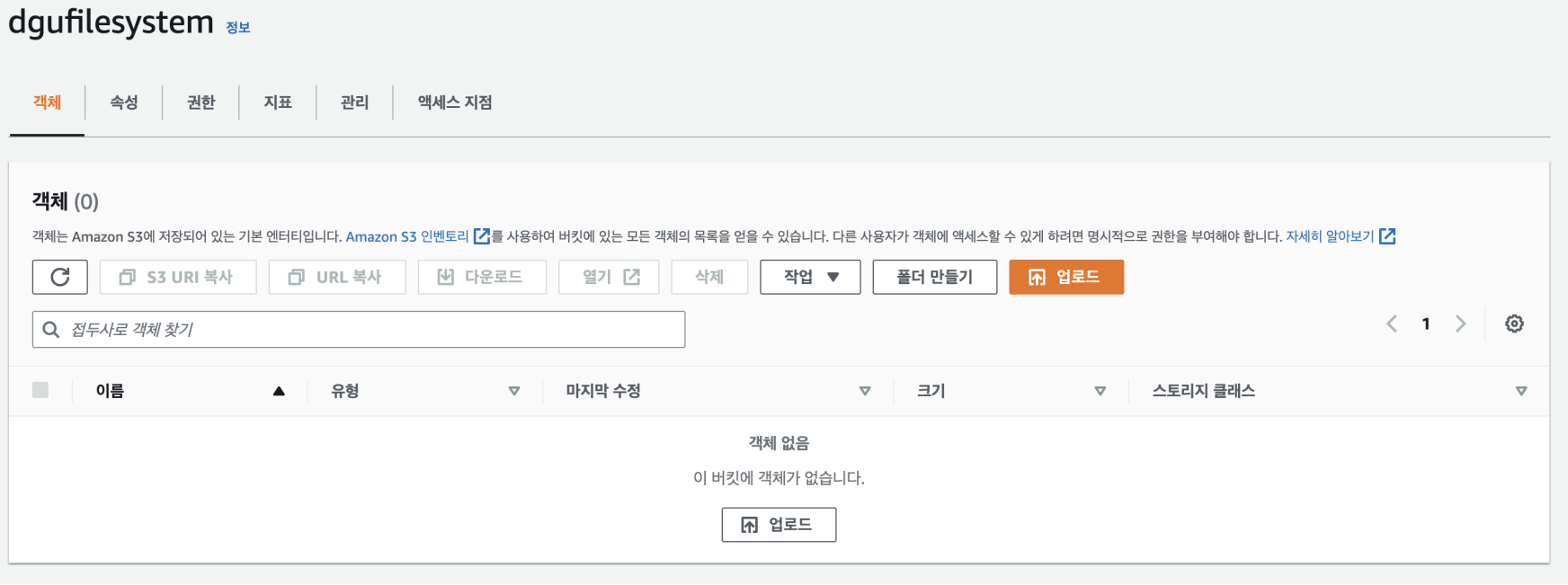
```

과목 선택 페이지 결과



b. AWS 세팅 및 파일 시스템 엔티티 생성 - 최필환

AWS 세팅



- AWS S3 객체 생성.

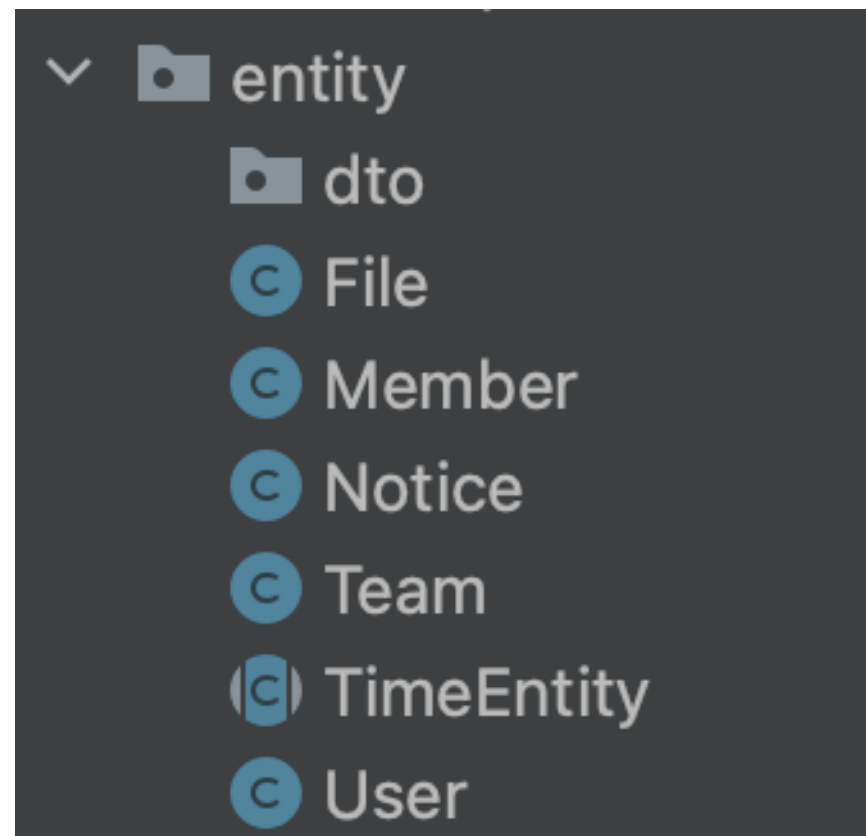
```
# AWS 버킷 정보
cloud:
  aws:
    s3:
      bucket: dgufilesystem
      region:
        static: ap-northeast-2
      stack:
        auto: false

# AWS에서 instance profile 사용
credentials:
  instance-profile: true

# AWS 접근 키
accessKey:
secretKey:
```

- AWS 버킷 생성 및 연결 yml 작성

파일 스토리지 Entity 생성



- 파일 시스템 엔티티 생성 및 연관관계 매핑


```

package OSSP.demo.entity;

import lombok.AccessLevel;
import lombok.Builder;
import lombok.Getter;
import lombok.NoArgsConstructor;

import javax.persistence.*;
import java.time.LocalDateTime;

2 usages  👤 thisishwan2
@Getter
@Entity
@NoArgsConstructor(access = AccessLevel.PROTECTED)
public class File extends TimeEntity{

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long file_Id;

    1 usage
    private String commit_Message;

    private String S3_Url;

    1 usage
    private String file_Name;

    //단방향
    1 usage
    @ManyToOne
    @JoinColumn(name = "member_Id")
    private Member member;

    👤 thisishwan2
    @Builder
    public File(String commit_Message, String file_Name, Member member){
        this.commit_Message = commit_Message;
        this.file_Name = file_Name;
        this.member = member;
    }
}

```

c. 로그인 기능 - 민한결

User Entity 작성

```
no usages
@Id
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long user_Id;

1 usage
@Column(nullable = false, length = 10, unique = true)
private int student_Id;

1 usage
@Column(nullable = false, length = 20, unique = false)
private String dept;

1 usage
@Column(nullable = false, length = 10, unique = false)
private String name;

1 usage
@Column(length = 50)
private String password;

no usages  👤 Hankyul Min
@Builder
public User(String name, String password, String dept, int student_Id){
    this.name = name;
    this.password = password;
    this.dept = dept;
    this.student_Id = student_Id;
}
```

UserRepository 오픈소스 코드 수정

```

public interface UserRepository extends JpaRepository<User, Long> {
    /* Security */
    1 usage
    Optional<User> findByUsername(String username);

    /* OAuth */
    1 usage
    Optional<User> findByEmail(String email);

    /* user GET */
    2 usages
    User findByNickname(String nickname);

    /* 중복 검사> 중복인 경우 true, 중복되지 않은 경우 false 리턴 */
    1 usage
    boolean existsByUsername(String username);
    1 usage
    boolean existsByNickname(String nickname);
    1 usage
    boolean existsByEmail(String email);
}

```

- 소셜 로그인 관련 코드 삭제
- 변수 User Entity 맞추어 변경

```

no usages  🧑 Hankyul Min
public interface UserRepository extends JpaRepository<User, Long> {
    /* Security */
    no usages  🧑 Hankyul Min
    Optional<User> findByStudent_Id(String student_Id);

    /* 중복 검사> 중복인 경우 true, 중복되지 않은 경우 false 리턴 */
    no usages  🧑 Hankyul Min
    boolean existsByStudent_Id(String student_Id);
}

```

UserDto 오픈소스 코드 수정

- 회원가입시 학번 검증 추가
- 회원가입시 학과 검증 추가

```
/**
 * request, response DTO 클래스를 하나로 묶어 InnerStaticClass로 한 번에 관리
 */
no usages  🧑 Hankyul Min *
public class UserDto {

    /** 회원 Service 요청(Request) DTO 클래스 */
    no usages  🧑 Hankyul Min *
    @Data
    @AllArgsConstructor
    @NoArgsConstructor
    @Builder
    public static class Request {

        1 usage
        @Range(min = 2016000000, max = 2023999999, message = "학번은 2016000000~2023999999 사이의 숫자여야 합니다.")
        private int studentId;

        no usages  new *
        @AssertTrue(message = "학과를 선택해주세요.")
        public boolean isDeptValid() {
            List<String> deptList = Arrays.asList(new String[]{"건설환경공학과", "멀티미디어공학과", "산업시스템공학과"});
            return deptList.contains(dept);
        }
    }
}
```

Pushed 1 commit to origin/Ha

회의 결과

- 이번주 결과
 - ✓ 디렉토리 구조 생성
 - ✓ 오픈소스 분석 및 적용 내용 수정
 - ✓ AWS 세팅 설정
 - ✓ 파일 스토리지 엔티티 생성
 - ✓ 로그인 기능 엔티티, 리포지토리, DTO
 - ✓ 로그인 페이지/과목 선택 화면 구현
- 다음주 안건
 - ☐ 파일 업로드 다운로드 구현/ DB 연결
 - ☐ 팀 활동 페이지 화면 구현
 - ☐ 로그인 기능 구현