**설계목적/문제의식**

* 교내 팀플을 진행하면서 도서관 회의실 이용 / 다우기술 인턴 기간동안 다우오피스의 회의실 예약 기능을 사용했는데 두 기능 모두 회의실을 실시간으로 예약 기능 외에 내가 찾고자 하는 딱 맞는 회의실을 바로 찾아주는 기능은 없음.
* 내가 원하는 시간대에, 원하는 규모의, 원하는 비품이 구비되어 있는 회의실을 찾기 위해서는 일일히 층병, 회의실별 정보를 확인해야 하는 불편함 존재.
* 기존의 기능을 강화하여 불편함을 해소하고자 ‘회의실 자동 검색’ 기능을 개발 결심

**차별성/독창성**

* 완전히 세상에 없던 서비스나 기능을 개발한 것은 아니지만 주어진 시간에 다룰 수 있는 기술의 범위 내에서 재학중 경험과 또 다우기술 인턴 경험을 살려 기존 기능의 불편함을 파악하고 강화하는 방식을 택했다는 것이 우리만의 차별성
* 간단하고 기본적인 기능이라고 생각할 수도 있음에도 불구하고 아직 두 시스템에는 없다는 점을 캐치

**디장고를 선택한 이유**

* 둘 다 프론트엔드를 주로 개발해왔기 때문에 스프링, node js 과 같은 기술은 경험은 없었고 대신 2명 다 파이썬을 사용할 줄 알고 디장고를 조금이나마 접해봤기 때문에 파이썬 기반 웹 프레임워크인 디장고를 선택하여 파이참에서 셋팅해 환경 구축
* 서버를 구축하지 않았고 로컬에서만 구동
* Mysql 과 연동이 쉽다는 점도 디장고를 선택한 이유

**설계 및 구현**

* 사용자가 예약일, 인원, 시간대 등을 선택할 수 있지만 이런 선택사항이 모두 필수 선택사항이 되면 오히려 검색 결과를 너무 제한할 수 있다는 생각이 들었기 때문에 예약일, 인원 정도만 필수 사항으로 입력하고 결과에 따라 비품여부나 시간대를 확인 한 후 유연하게 판단해 최종 선택을 거칠 수 있도록 함. 기존의 예약 기능까지 개발하도록 설계
* 기능에 사용될 사원 테이블, 회의실 테이블, 얘약정보 테이블로 구성된 간단한 DB 설계. 디장고의 MVT 패턴을 사용해 구현. Model에서 mysql 과의 연동을 통해 데이터베이스를 정의하고, Template 에서는 html.css,js 을 사용해 프론트엔드를 개발하였다. Html 에서 form 태그를 이용해 사용자가 입력한 데이터와 url 을 담아 요청을 보내면 view 에서 해당 url 에 대한 요청을 받았을 때 작업을 처리하기로 정의해둔 메소드가 db 접근하는 등 필요한 작업을 수행하고 그 결과를 요청 받은 template 페이지에 반영

**보완점**

* 핵심 기능은 모두 구현했지만 더 디테일한 부분들에서 발생할 수 있는 버그 해결(??)
* commit 이나 rollback 같은 트랜잭션을 사용해서 데이터베이스에 예약 정보를 Insert 할 때 오류를 방지하고 무결성을 보장할 필요가 있을 것 같음

https://velog.io/@kho5420/Django-%EC%9E%A5%EA%B3%A0-%ED%8A%B8%EB%9E%9C%EC%9E%AD%EC%85%98-%ED%99%9C%EC%9A%A9%ED%95%98%EA%B8%B0

* Employee Table 설계를 하면서 User Table과 혼동을 했던 것 같다. 그래서 굳이 Employee 테이블에는 필요 없을 것 같은 ID/Password 같은 필드는 삭제를 하고 제 3정규형을 만족시킬 수 있도록 해야 할 것 같다.
* URL 및 뷰 매핑 관계에 대해 더 자세히 공부하고 이해하는 과정을 거친 후 코드를 재검토하고 리팩토링 과정을 거쳐야 할 것 같다. 지금은 view에서 하나의 메소드로 프론트엔드에서 접근하는 모든 데이터를 처리하고 있기 때문에 이 부분에서 모듈화가 이뤄져야 할 것 같다. (시간 없으면 빼기)
* ~~이번 아이디어를 업그레이드 시켜 기업 내의 회의실을 찾아주고 예약하는 기능 뿐만 아니라 클라이언트 내외부 미팅 장소를 추천해주고 예약을 편리하게 도와주는 기능이나 서비스를 개발하면 좋을 것 같다.~~

**예상 질문**

1. 개발에 있어 가장 신경 썼던 것

* 프론트의 경우 위 발표에서 언급했듯이 사용자가 검색기능을 이용하되 검색 결과로 인해 오히려 선택의 폭이 제한적이게 되거나 시간을 계속 바꿔가며 여러 번 검색해야 하는번거로움이 없도록 하는 것에 주의했고, 백엔드의 경우 사실 직접 이렇게 개발하는 것은 처음이기 때문에 동작할 수 있도록 하는 것에 집중했다. DB 설계를 하면서 이번 학기 배웠던 정규화를 고려해보려고 노력했는데 결국 employee 테이블에는 조금 문제가 있었고, 테이블수가 적어 크게 신경 쓸 내용이 없었던 것 같다.

1. 이번 경험을 통해 배웠던 것

* 디장고에 대해 경험해 볼 수 있어서 좋았다. 백엔드를 다뤄본 것은 처음인데 꽤 간단하고 빠르게 사용할 수 있는 것 같았다. 사실 난 안드로이드 앱개발을 하고 싶은데 혼자 토이프로젝트를 진행할 때 디장고를 이용해보는 것도 좋을 것 같다.

1. URL 맵핑에 대해 지금은 얼마나 공부했는지
2. 정규화에 대해 설명해 봐라

* 설계된 DB가 적절한가를 판단 하는 기준 제공
* attribute 간에 종속관계 분석
* 데이터의 중복으로 인한 데이터의 일관성 훼손 방지
* 기본키가 기본키 외 모든 다른 속성을 가르킴(종속관계)(제2정규화)
* 기본키를 제외한 속성들 간에 종속 관계 허용 X (3 정규화)

1. 피티 자료에 employee 테이블이 user 테이블과 혼돈했고 제2정규형만 만족했다고 했는데 이게 뭔 소린지?

* 보통 서비스를 위한 테이블 설계를 하면 유저 테이블을 먼저 만들곤 한다. 그래서 나도 습관처럼 사원 테이블이라고 해놓고 아이디 패스워드 같은 속성을 만들어버렸다. 로그인 기능도 없는데 말이다. 그래서 이런 속성이 불필요하게 들어갔다고 생각을 했고, 사원 일련번호가 여기서 기본키인데 이 키가 이메일을 가르키고 이메일은 사람마다 유일하기 때문에 다시 사원 이름을 가르킨다. 물론 사원 일련번호는 기본키이므로 사원 이름을 결정한다. 따라서 a->b, b->c 인데 a->c를 결정하면서 제 3 정규형을 만족하지 못한다.
* 사원번호가 두 사람간에 교체되는 경우 이름이 바뀌면 이메일 주소도 바뀌어야 하는데 두 속성을 다 고쳐야 하는 번거롭다. 따라서 테이블을 분리하여 사원번호에 해당하는 이름이 바뀌어도 이메일 테이블을 따로 만들어서 참조만 다르게 해주는 방식을 택해야함

1. 다우기술 인턴 경험이 있었다면 현재 이 기능을 다우오피스에 바로 적용 가능할지?

* 다우기술에서는 디장고는 사용하지 않고 스프링을 사용하고 자바를 많이 사용하는 것으로 알고 있다. 하지만 기능이 단순하고 DB도 이미 다 구축되어 있기 때문에 기획팀과 기능을 좀 더 구체화 한 다음에 개발하면 금방 적용이 가능할 것 같다.

1. Get, post 방식  
   
2. 디장고란 무엇인가?

* 웹사이트를 신속하게 개발하는 하도록 도움을 주는 파이썬 웹 프레임워크이다. 초보자도 명령어 몇번으로 디비 연결을 가능하게 해주고 특별히 환경 설정에 어려움 없이 개발을 진행할 수 있었다.
* 페이지 url 별로 view 함수를 정의하고 그 안에 수행할 내용을 작성하면 해당 페이지가 새로고침 되거나 action이 발생했을 때 내용이 수행되는 것으로 공부하였다.