

ONLINE CODING EDUCATION

온라인 교육

개발기간 2020.03.30 ~ 2020.04.24

참여자 윤보미 정성훈
강승현 조한림

교육기관 구공팩토리

CONTENTS

01

기획

- 개요
- 시나리오
- 요구분석
- 개발일정

02

설계

- 아키텍처
- 화면
- 데이터베이스

03

개발

- 전체모식도
- 세부구조
- 이슈및해결

04

결론

- 결과물
- 보완사항



01

기획

ONLINE CODING EDUCATION

“ 온라인 코딩 교육 ”

참여인원

4명(강승현, 윤보미, 정성훈, 조한림)

개발기간

3주

역할분담

윤보미 (Front 회원관리)

정성훈 (Front 강의관리)

강승현 (Back Spring)

조한림 (Back Django)

형상관리

각자 Git 사용

01



CODE
T  GETHER



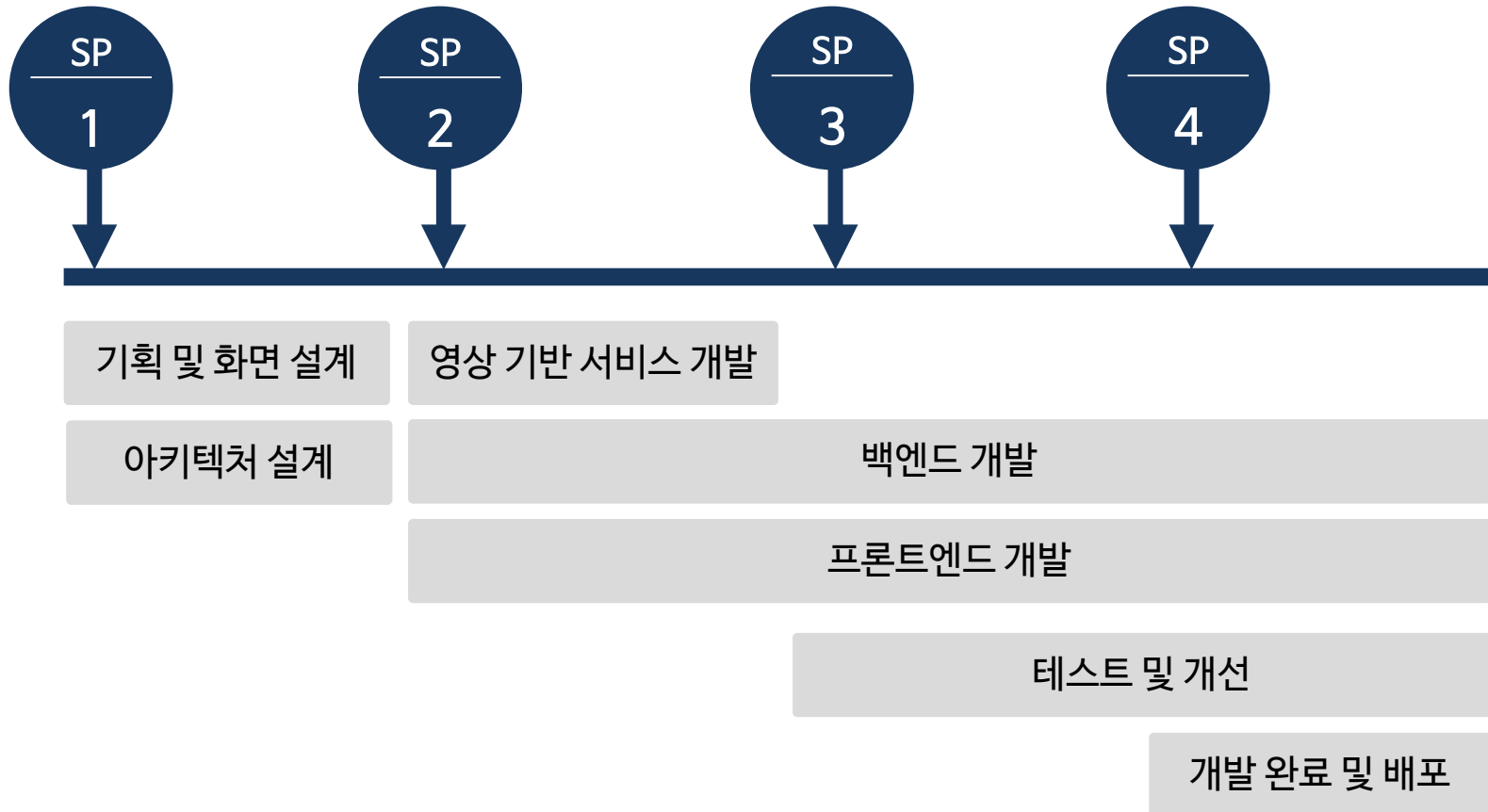
- 코딩을 배우기 위해 강의 사이트를 찾아 학생으로 가입했다.
- 강사들이 올려놓은 강의를 볼 수 있다.
- 시간이 맞는 강의를 선택해 수강 신청을 하였다.
- 강의 결제를 완료했다.
- 정해진 시간에 강의를 수강했다.
- 강의를 듣고 학습 방에서 과제를 다운받았고, 과제를 하는 중에 모르는 게 있어서 질문 방에 올렸다. 이후 답변이 달려서 해결한 후 제출했다.
- 강사가 채점 후 피드백을 주었다.
- 수강을 완료한 후 만족도와 완성도, 후기를 올렸다.

- 강사로 가입을 하고, 스케줄에 맞게 강의 등록과 커리큘럼을 입력했다.
- 매칭 요청이 들어오거나 수강 신청이 들어와서 승인했다.
- 개인 강의실이 생겨 정해진 시간에 강의를 할 수 있다.
- 강의를 진행하며 과제를 내주고 질문을 받아 해결하고 피드백을 줬다.
- 수강 후 학생이 평점을 주었다.

01

아키텍처	아키텍처 설계
	아키텍처 검토
화면설계	화면 설계
	화면 검토
데이터베이스 설계	데이터베이스 설계
	데이터베이스 검토
사용자 로그인	사용자는 e-mail 주소와 비밀번호를 입력해 로그인할 수 있다
	사용자는 소셜 계정을 이용해 로그인할 수 있다
회원관리	일반가입 사용자는 필수 입력사항 및 기타 개인 정보를 입력해 회원 가입을 할 수 있다.
	사용자는 개인 페이지를 통해서 사용자 정보 및 비밀번호 수정이 가능하다
	사용자는 회원탈퇴를 할 수 있다
강의관리	사용자는 강의를 등록할 수 있다.
	해당 강의를 생성한 사용자는 강의를 수정할 수 있다.
	해당 강의를 생성한 사용자는 강의를 삭제할 수 있다.
강의검색	사용자는 강의를 검색할 수 있다.
	사용자는 강의를 조회할 수 있다.
수강화면	결제를 한 학생들에 한하여 온라인 강의를 볼 수 있다.

01



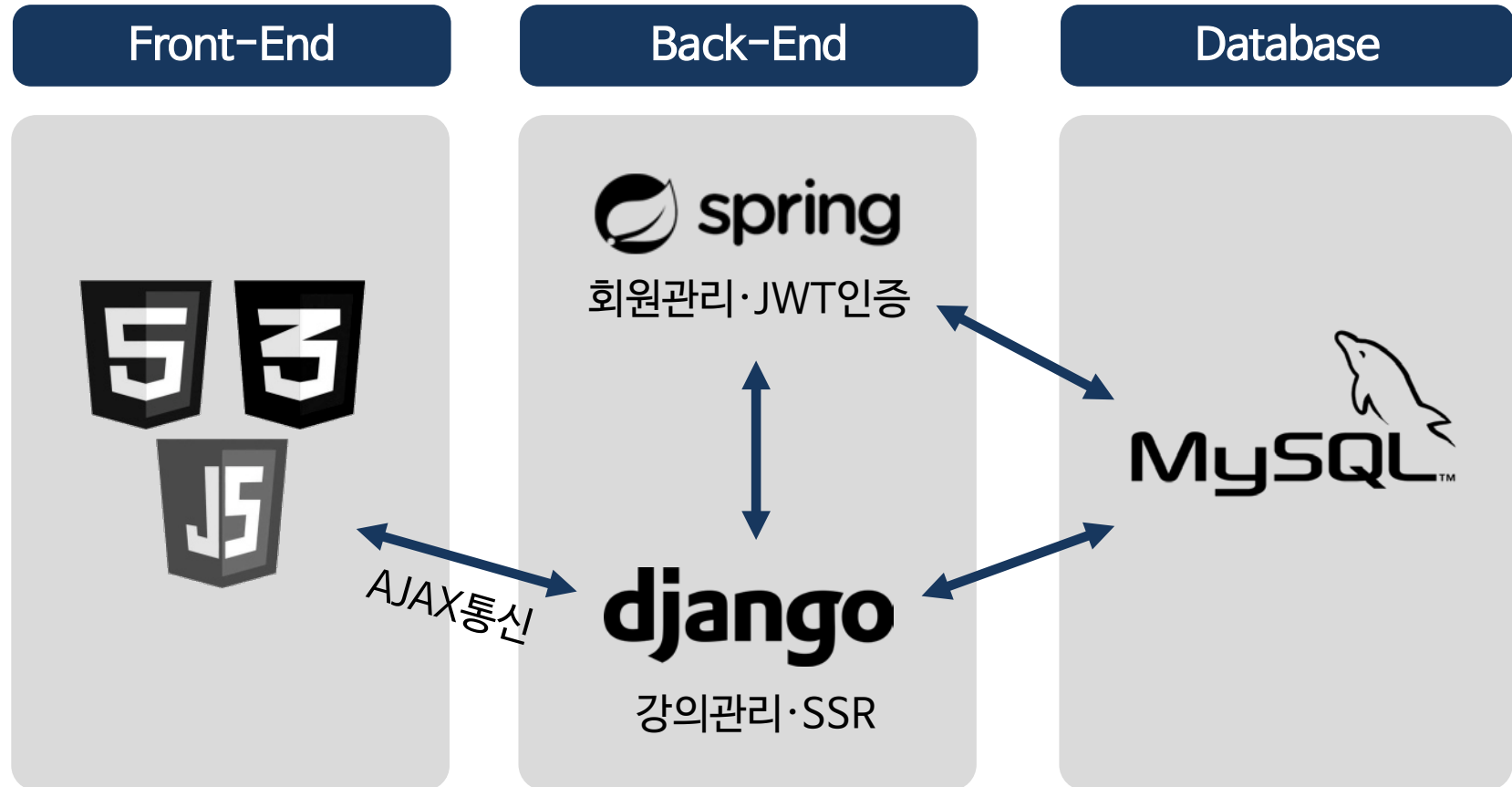


02

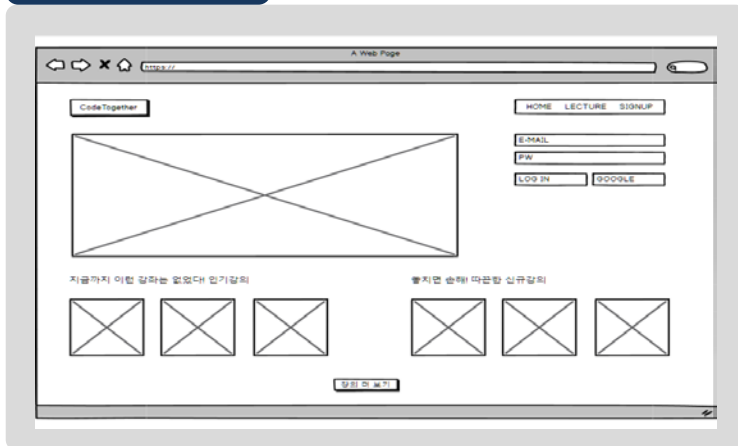
설계

ONLINE CODING EDUCATION

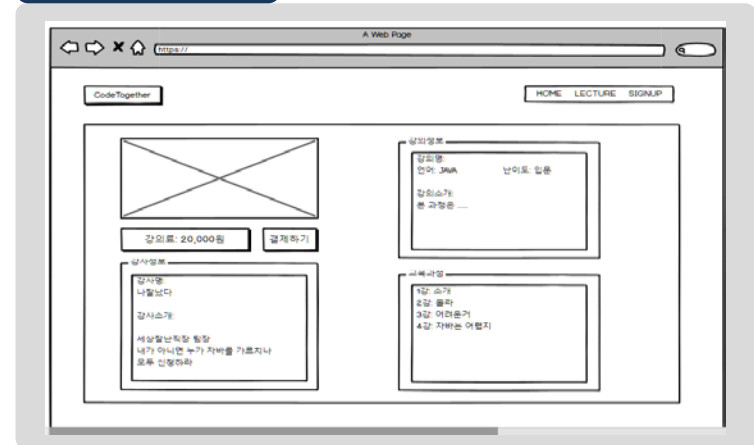
02



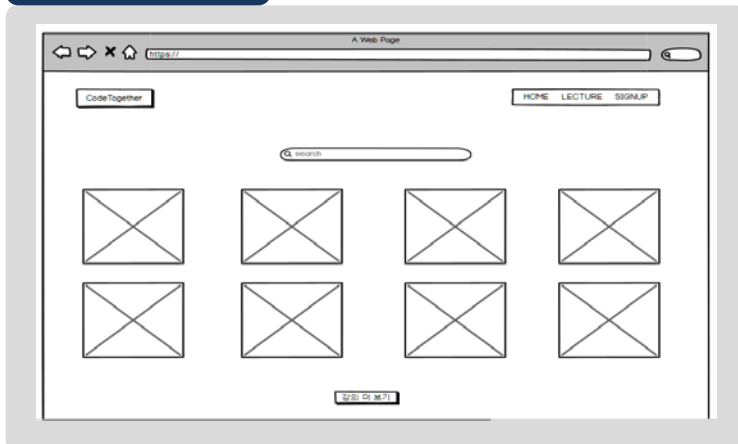
메인



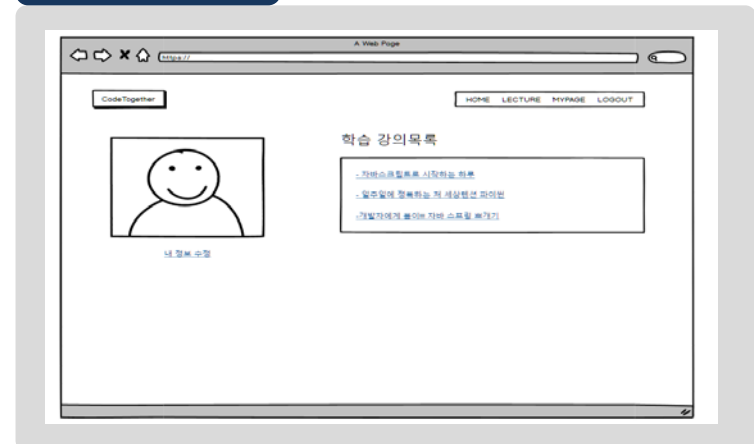
강의상세



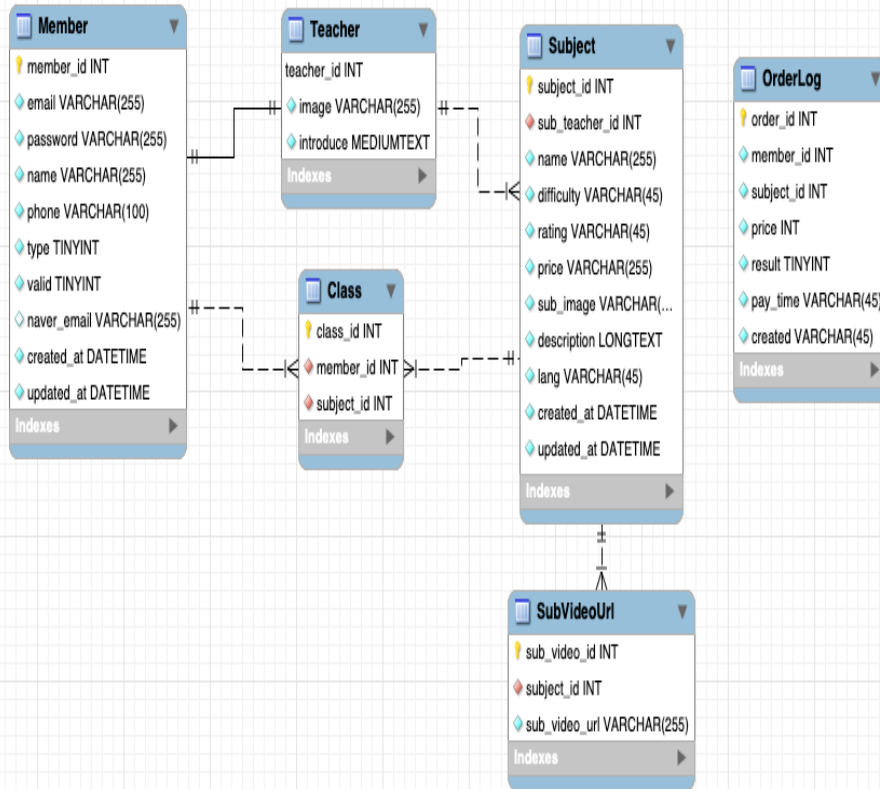
강의검색



개인계정



테이블 구성



Member

- 모든 회원정보를 저장하는 테이블이며 email은 unique값이다.

Teacher

- 일반회원이 강사등록을 할때 필요한 정보를 저장하는 테이블
- 사진은 한장씩만 저장

Subject

- 강사들이 수업을 등록할때 필요한 정보를 저장하는 테이블

SubVideoUrl

- 수업들이 가지고 있는 차수들의 정보를 저장하는 테이블

Class

- 일반 회원이 수업을 수강하기 위해 결제를 했을때 회원과 강의를 매칭하는 테이블

OrderLog

- 결제를 시도 하였을때 결제 정보, 로그를 저장하는 테이블

테이블 관계

- Member ↔ Teacher 1 : 1
- Member ↔ Class 1 : N
- Teacher ↔ Subject 1 : N
- Subject ↔ Class 1 : N
- Subject ↔ SubVideo 1 : N
- Subject ↔ SubImage 1 : N
- OrderLog

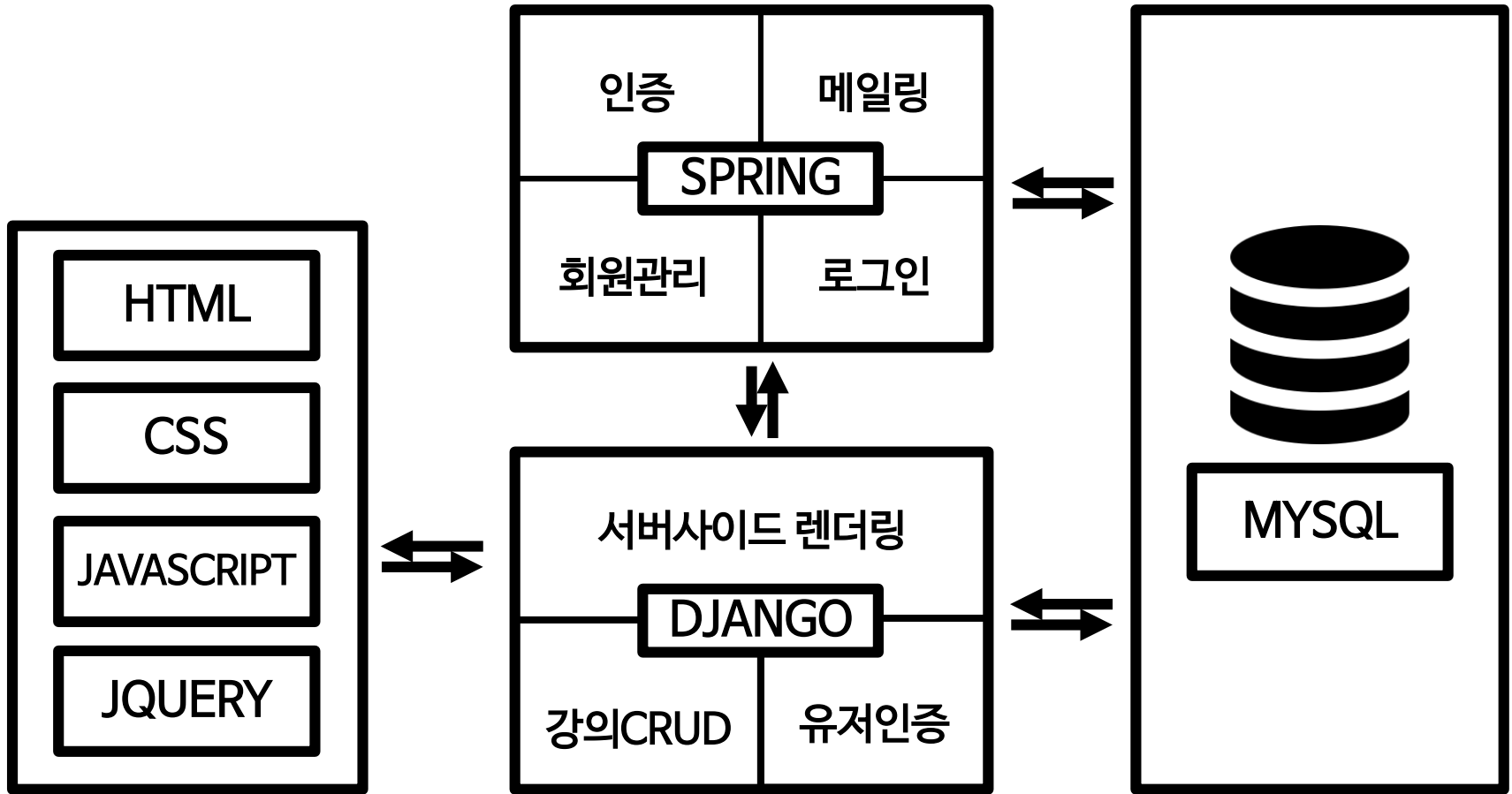


03

개발

ONLINE CODING EDUCATION

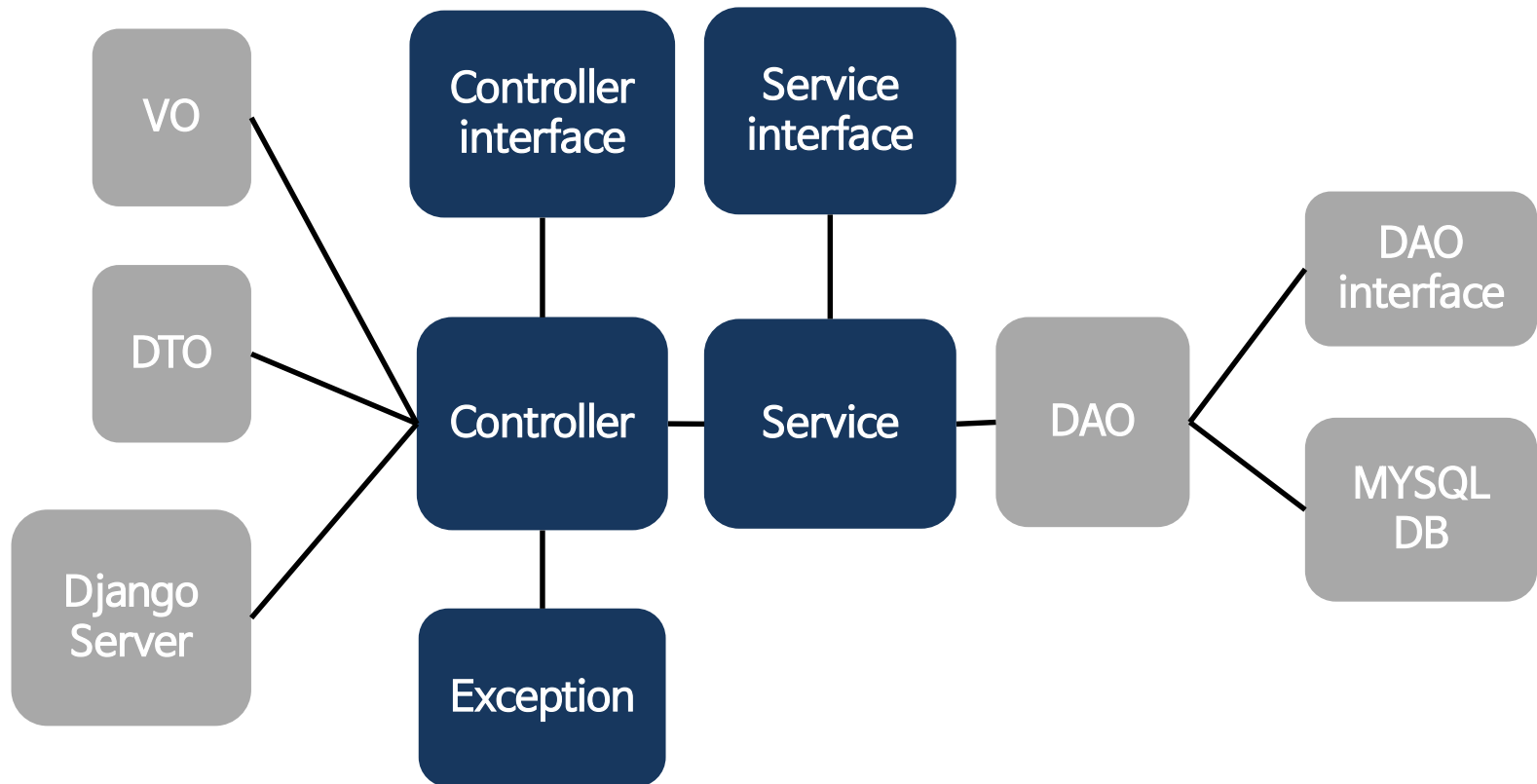
03



03

[Spring 구조]

: 유지보수 및 확장성을 고려한 MVC패턴 적용



[SpringAPI]

: 시각화를 통한 업무효율 향상 및 RESTful API
구현을 위한 Swagger 사용

home-controller : Home Controller
[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/	home
-----	---	------

jwt-controller : Jwt Controller
[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST	/auth/checkToken	checkToken
------	------------------	------------

POST	/auth/getId	getMemberId
------	-------------	-------------

login-controller : Login Controller
[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST	/login	loginPOST
------	--------	-----------

GET	/snsService/callback	naverCallback
-----	----------------------	---------------

mail-controller : Mail Controller
[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST	/findPassword	findPassword
------	---------------	--------------

GET	/verify	verify
-----	---------	--------

teacher-controller : Teacher Controller
[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST	/teacher/createInfo	createTeacherInfo
------	---------------------	-------------------

POST	/teacher/selectInfo	selectTeacherInfo
------	---------------------	-------------------

POST	/teacher/transFromUser	trans_teacher
------	------------------------	---------------

PUT	/teacher/updateInfo	updateTeacherInfo
-----	---------------------	-------------------

user-controller : User Controller
[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

POST	/user/checkId	checkId
------	---------------	---------

POST	/user/create	create
------	--------------	--------

DELETE	/user/delete	delete
--------	--------------	--------

POST	/user/select	select
------	--------------	--------

PUT	/user/update	update
-----	--------------	--------

03

[Django Rendering]

: GET 요청과 POST 요청 시 로직과 화면을 렌더링



[사용자인증과정]

이슈

- 인증 과정 설계
: 암호화 세션과 쿠키를 통한 사용자 인증
- 문제
: Django와 Spring으로 구성된 이중 서버
- 고려한 해결 방안
 1. 공개키로 암호화된 세션값을 쿠키에 저장 (공개키는 장고와 스프링만 가지고 있다.)
 2. 세션 값을 암호화 하여 DB 또는 Redis를 사용하여 저장
 3. JWT를 사용하여 사용자 정보 저장



해결

- 쿠키에 저장
: 암호화 방식은 개발기간에 영향을 주므로 적용이 어려움
- DB 혹은 Redis에 저장
 1. DB 저장은 공부 목적의 프로젝트에 부적절하다고 판단함
 2. Redis를 사용하는 것은 암호화 방식과 마찬가지로 개발기간에 영향을 주므로 적용이 어려움
- JWT 사용
 1. 라이브러리 사용이 가능하고 개발기간에 가장 영향을 덜 미칠 것으로 판단함
 2. 해당 규모의 서비스에 Access Token 사용이 적당하다고 판단함
 3. 추후 Refresh Token 사용을 대비해 Spring에만 인증 서버를 두기로 결정

04

결론

ONLINE CODING EDUCATION

[회원관리]

회원가입

Code Together

이름 두 글자 이상 입력하세요

강

핸드폰 번호

010-0000-0000

이메일 중복 확인하세요

primavera0331@naver.com

비밀번호

비밀번호 확인 비밀번호가 일치

**

가입

정보수정

Code Together

수정할 내용을 입력하세요

이름

강승현

핸드폰 번호

010-0000-0000

이메일 (이메일 변경 불가)

primavera0331@naver.com

비밀번호

|

이 입력한을 작성하세요

비밀번호 확인

수정하기

“ 등록 및 수정 ”



- 입력된 정보 유효성 검사
- 유효한 정보가 아닐 시 안내 메시지 출력

[회원관리]



“ 조회 ”

- 학생
 - 현재 학습 중인 강의 목록 확인 가능
 - 목록 클릭 시 해당 강의 수강 가능
 - 강사등록, 회원정보 수정 및 탈퇴 신청 가능



- 강사
 - 현재 강의 중인 목록 확인 가능
 - 강의 등록, 회원정보 수정 및 탈퇴 신청 가능

“ ”

탈퇴

- 본인 인증 후 탈퇴 진행

[강의관리]

“ 등록및수정 ”

강사등록

조한필님!
강의를 등록하기 위해서는 아래 정보가 필요해요!

강사사진 | **과목선택** | 선택한 과목 없음

강사소개

강사님을 소개해주세요!

강의등록

강의정보:

강의명:

강의분류: 선택한 과목 없음

강의도:

강의도:

강의 소개:

강의 소개:

교육과정:

강의: **등록하기**

강의 추가 목록

제목 * react - 입문하기 * **삭제**

강의 * react - 입문하기 * **등록하기**

제목 * react - 입문하기 * **삭제**

강의 * react - 입문하기 * **등록하기**

- 입력된 정보 유효성 검사
- 유효한 정보가 아닐 시 안내 메시지 출력

“ 강의정보 ”

강의정보

강의명: 이재혁

강사소개: 세상 친절한 리액트 전문가

강의정보

강의명: 모던 리액트 입문

언어: javascript, 난이도: 입문

강의소개: 세상 친절한 리액트 전문가

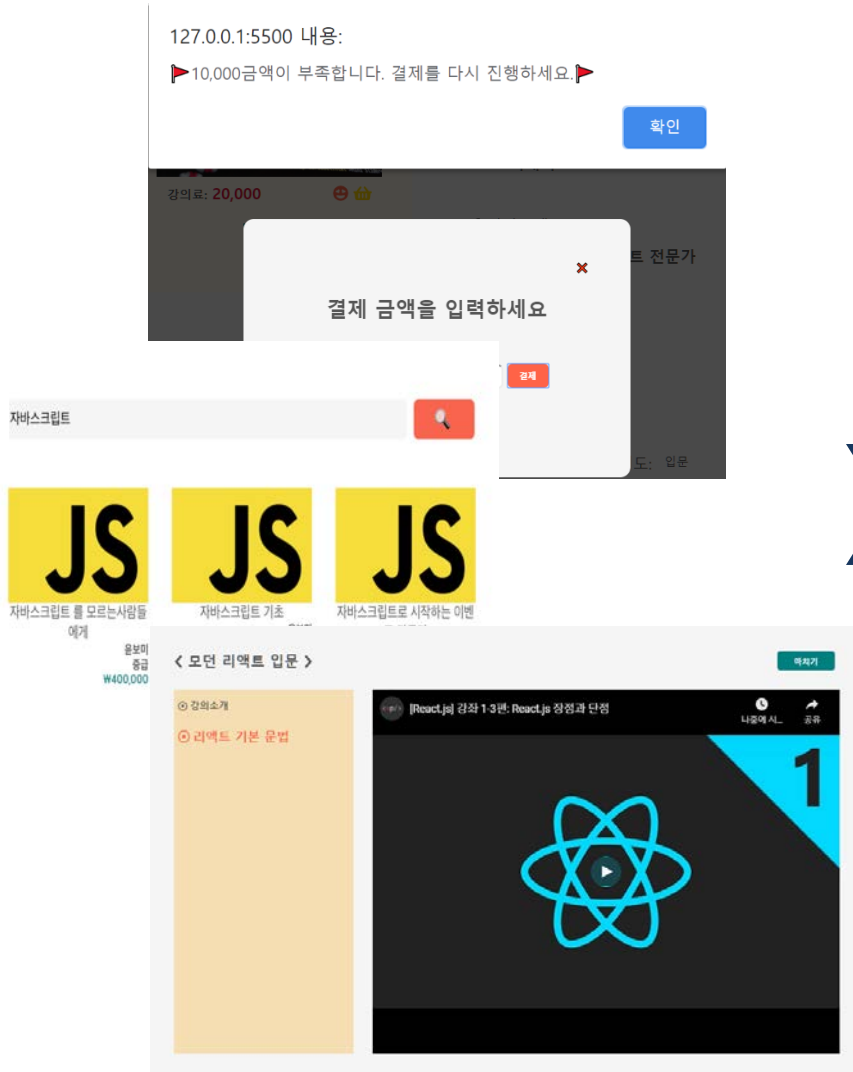
교육과정

1강: 강의소개

2강: 리액트 기본문법

- 회원만 결제 가능

[강의관리]



“ 강의결제 ”

- 결제된 강의는 마이페이지 학습강의 목록에 추가

“ 검색 ”

- 강의 검색 결과 페이지네이션

“ 강의화면 ”

- 강의 목록 클릭 시 해당 영상 확인 가능

04

보완점

- 찜 기능, 리뷰 및 이슈 기능의 부재

이슈 회고

- 사용자 인증 처리에 관한 방법을 학습함
- 스토리 점수를 이용하여 개발기간을 미리 고려함으로써 이슈에 맞춰 유동적 기간 조율이 가능했음
- 위의 유동적 기간 조율로 인해 스프린트도 유연하게 가져갈 수 있었음



THANK YOU

ONLINE CODING EDUCATION