

중간 발표 보고서

삼성SW청년아카데미 대전캠퍼스 7기

대전 공통 1반 B107

정 진 이현태 김연수 이현정 이주희 박범수

목 차

1. 팀 소개 및 담당	3
2. 프로젝트 소개	3
3. 기술 스택	4
4. 협업 내용	5
6. 서비스 아이콘 및 컬러 팔레트	9
7. 시스템 흐름도	10
8. 와이어 프레임	10
9. ERD	12
10. API Docs	16
11. 현재 진행 상황	16
12. 추후 진행 계획	17
13. 피드백	18

1. 팀 소개 및 담당

[표 1] 팀 소개 및 담당

No.	이름	역할	포지션	담당
1	정 진	팀장	프론트엔드	화상 미팅
2	이현태		프론트엔드	커뮤니티
3	김연수		프론트엔드	유저
4	이현정	백팀장	백엔드	화상 미팅
5	이주희		백엔드	유저
6	박범수		백엔드	커뮤니티

2. 프로젝트 소개

1) Idea

- 자체 커뮤니티와 함께 화상 회의가 가능한 교육 플랫폼

2) 제안 배경

- 비대면 수업 만족도가 높아지는 것에 학습 생산성 향상과 저하는 큰 차이가 없음
- 생산성 제고를 위해 가장 필요한 것은 교수·학생 간 소통 채널이라는 응답이 있음

(문유숙, 비대면 수업 만족도 1년만에 21.3% 상승..., 수업 생산성 저하됐다 36%, 2021년 10월 11일)

3) 타겟층

- 학교 및 SSAFY와 같은 교육 기관
- 커뮤니티 이용이 활발한 20 ~ 30대

4) 주요 기능 및 차별점

- 화상 회의와 커뮤니티를 한 서비스 내에서 이용 가능
- 아이콘과 더불어 명확한 서비스 설명 추가
- 소통에 어려움이 없도록 익명 여부를 선택할 수 있음
- 각 사용자 페이지 내 1:1 문의 기능

5) 기대효과

- 화상 회의 종료 후에도 소통을 위한 서비스가 있어 소통 생산성을 높일 수 있음
- 서비스 내 상세 설명으로 편의성을 높임
- 익명 기능의 선택으로 좀 더 자유로운 분위기에서 소통 가능
- 단독으로 사용한 1:1 문의 기능으로 교수자와 학생 간 친밀도 향상 가능

3. 기술 스택

[표 2] 기술 스택 상세 내용

구분	사용 목적	사용 기술	기술 스택	버전
협업	코드 형상 관리	GitLab		
	이슈관리	JIRA		
	커뮤니케이션	Mattermost		
		Notion		
		Webex		
		Google Meet		
		Discord		
BackEnd	개발	DBMS	MySQL	8.0.28
		DB API	JPA(Java Persistence API)	2.7.2
		Java	Zulu	1.8
		Framework	Spring	5.3.22
			Spring Boot	2.7.2
			Spring Security	
		JWT(JSON Web Token)		
		WAS	Tomcat	9.0.60
		Build	Gradle	7.5
		WebRTC	Openvidu	2.22.0
		API Docs	Swagger	
		IDE	IntelliJ	
FrontEnd	개발	HTML		
		CSS		
		JavaScript		
		TypeScript		4.4.2
		Library	React	1.8.0
			Redux	8.0.2
			MUI(Material-UI)	5.9.1
		Style-components		5.3.5
		WebSocket	@stomp/stompjs	
			stompjs	
			sockjs-client	
		IDE	Visual Studio Code	1.69
Server	배포	서버	AWS	
			nginx	
		OS	Ubuntu	
		배포	Docker	
			Jenkins	

4. 협업 내용

1) Scrum

- 사용 목적 : 일일 협업 내용 공유
- 진행 일시 : 매일 조례 후, 10분 뒤
- 내용 : 한 것, 할 것, 전체 전달(요청) 사항, 오류(문제) 상황 공유
- 결과 : 보고서 형태로 정리

B107 Scrum - ④	
일시	22.07.29 09:55
참석자	정진, 이현태, 김연수, 이현정, 박범수, 이주희
전체 전달(요청) 내용	
한 것	<p>[정진(FE)]</p> <p>1. 화상 수업에서 참가자 목록 띄우기</p> <p>2. 화상 수업에서 채팅 주고 받기</p> <p>[이현태(FE)]</p> <p>1. redux (redux-toolkit) 공부</p> <p>[김연수(FE)]</p> <p>1. 회원가입하면 로그인되기 - 로컬에 데이터 저장 이용</p> <p>[이현정(BE)]</p> <p>1. openvidu on premises로 연결 시도</p> <p>[박범수(BE)]</p> <p>1. JPA 학습</p> <p>2. Springboot 학습</p> <p>[이주희(BE)]</p> <p>1. JPA 학습</p>
오류/문제 상황	<p>[정진(FE)]</p> <p>1. 현재 비정상적으로 종료된 경우 비정상적으로 종료된 사람이 나가지지 않고 계속 남아있음</p> <p>[이현태(FE)]</p> <p>1. Post CRUD 중 C를 진행하고 데이터를 받아왔는데 부모-자식 component 관계가 아닌 다른 component에 데이터를 전달하는 것이 안 됨 (현재진행형)됨</p> <p>[이현정(BE)]</p> <p>1. Openvidu 배포 시, 오류 발생</p>
오늘 할 것	<p>[정진(FE)]</p> <p>1. 비정상적으로 종료된 사람 정상적으로 내보내기, 마이크나 카메라 제어 하기</p> <p>[이현태(FE)]</p> <p>1. Post CRUD C 데이터 전달 완료하여 Community page에 데이터 잘 들어오는 지 확인</p> <p>1. redux 활용하여 전달할 예정</p> <p>2. Post Detail page 작성</p> <p>1. layout 생성 (댓글 부분까지)</p> <p>2. Comment CRD 중 C, R</p> <p>[김연수(FE)]</p> <p>1. 아이디 찾기, 비밀번호 찾기 모달 만들기</p> <p>2. 로그인, 회원가입 토큰데이터를 store에 redux-toolkit으로 찍기</p> <p>[이현정(BE)]</p> <p>1. openvidu on premises로 연결</p> <p>[박범수(BE)]</p> <p>1. JPA 학습</p> <p>2. 게시판 CRUD 기능 구현</p> <p>[이주희(BE)]</p> <p>1. JPA 학습 마무리</p> <p>2. JWT</p> <p>3. 로그인, 회원가입 구현</p> <p>4. Social 로그인, 회원가입 구현</p> <p>5. Postman 이나 Swagger 로 확인</p>
추가 회의 내용	
비고	[공통] 매 정시마다 허리/목 스트레칭 진행하기

[그림 1] JIRA 보고서 예시

2) JIRA

- 사용 목적 : 이슈 관리, 일일 및 주간 협업 내용 공유
- 진행 일시 : 매주 월요일 스크럼 회의 후, 스프린트 생성
- 내용 : 매주 프로젝트 개발 계획을 토대로 Epic, Story 등 생성
- 결과 : JIRA 규칙 및 스토리 포인트 분배

[표 3] JIRA 규칙

구분	규칙
Epic	회의와 개발에 관련된 것으로 생성
	개발과 관련된 Epic 생성 시, 페이지 별 생성
Story	개당 최대 4포인트 할당 가능
	주당 최소 40시간의 스토리 포인트 할당
	4포인트 이상을 할당해야 할 필요가 있는 경우, 2개로 나누어 생성
	본인의 포지션에 따라서 [BE], [FE] 표기
Task	배포와 관련된 것으로 생성

Workload by assignee - 3rd Sprint		
Assignee	Issues	Story Points
Unassigned	0	0
 박범수	21	50
 이주희	21	49
 이현정	20	46
 이현태	22	45
 김연수	21	41.5
 정진	21	41
Total:	126	272.5

[그림 2] 3주차 스토리 포인트 내역

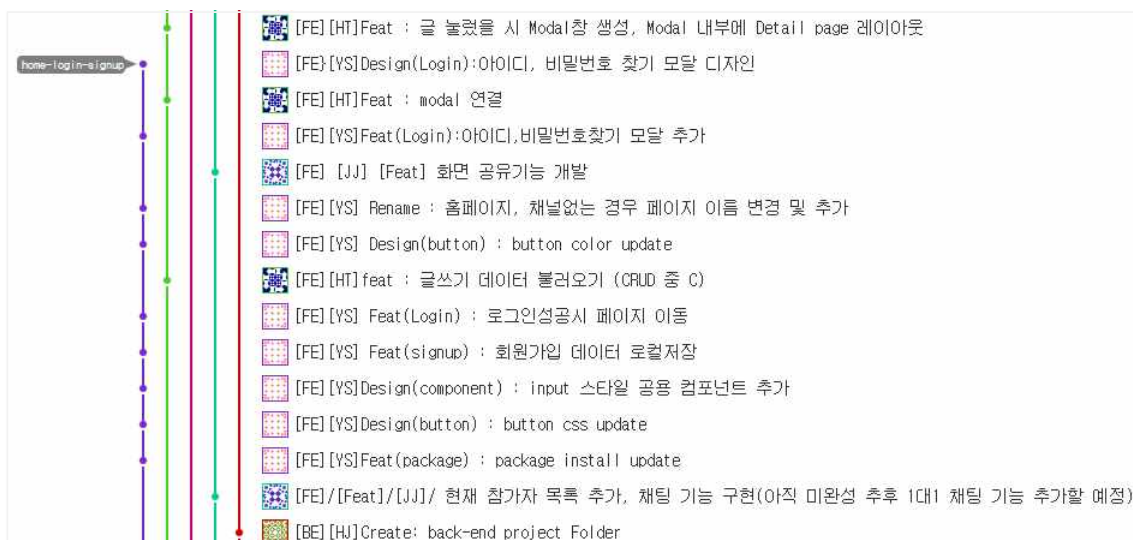
3) GitLab

- 사용 목적 : 코드 형상 관리 및 개발 진행 상황 공유
- 진행 일시 : 각 기능별 개발 완료 시, git push 진행
- 내용 : 포지션별, 기능별 브랜치 관리
- 결과 : git convention 및 git branch 관리

[표 4] Git Convention

No.	태그 이름	분류	내용
1	[FE]/[BE]	포지션	첫번째로 표기하는 태그로 포지션을 구분
2	[HH]	담당자	두 번째로 표기하는 태그로 이니셜로 담당자를 구분
3	Feat	기능	새로운 기능 추가
4	Fix	기능	버그 수정
5	Design	기능	사용자 UI 디자인 변경
6	!Breaking change	기능	급한 치명적인 버그 수정
7	Refactor	개선	프로덕션 코드 리팩토링
8	Style	개선	코드 포맷 변경, 세미콜론 누락 등의 기타 사항 개선
9	Comment	개선	주석 추가 및 변경
10	Docs	기타	문서 수정
11	Chore	기타	빌드 테스트 업데이터 및 패키지 매니저 설정
12	Test	기타	테스트 추가 및 리팩토링
13	Rename	기타	파일명 혹은 폴더명 수정
14	Remove	기타	사용하지 않는 파일 혹은 폴더 수정
15	Create	기타	파일 추가

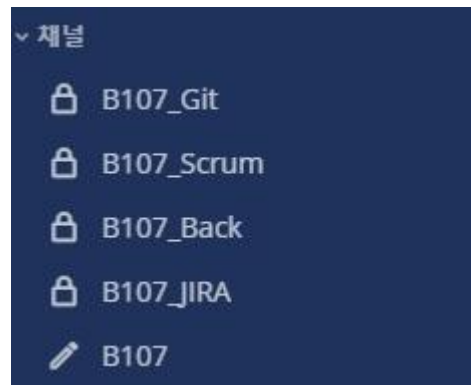
출처 : “Udacity Git Commit Message Style Guide”, Udacity, 2022년 07월 31일 접속



[그림 3] Git Branch

4) Mattermost

- 사용 목적 : 자료 공유 및 JIRA와 GitLab 알림봇 이용, 전체 공지
- 진행 일시 : 상시 사용
- 내용 : 용도 별 채널 관리
- 결과 : 용도 별 채널 관리



[그림 4] Mattermost 채널 목록

5) Webex, Google Meet

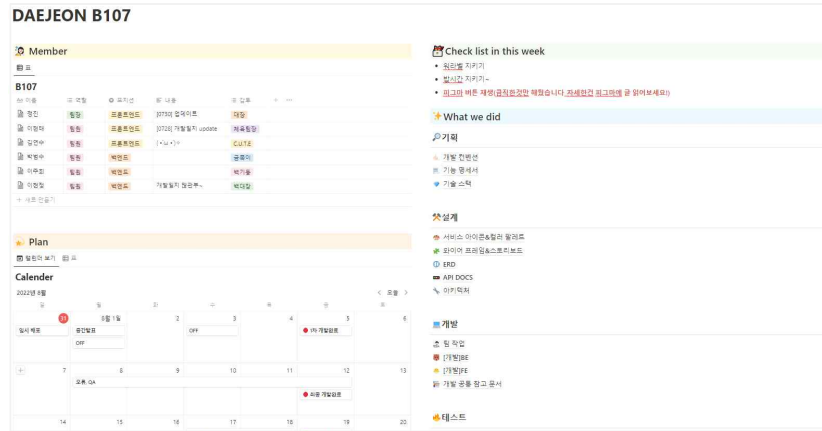
- 사용 목적 : 영상 회의
- 진행 일시 : 상시 사용
- 내용 : 영상 회의 시 사용

6) Discord

- 사용 목적 : 음성 회의
- 진행 일시 : 상시 사용
- 내용 : 음성 회의 시 사용

7) Notion : 프로젝트 전체 자료 공유

- 사용 목적 : 프로젝트 전체 자료 및 개발 일지 공유
- 진행 일시 : 상시 사용
- 내용 : 프로젝트 일정 관리 및 용도 별 워크스페이스 분류
- 결과 : 자료 공유 및 용도 별 워크스페이스 관리



[그림 5] Notion 페이지



[그림 6] 공유된 개발일지

5. 기능 명세서

- 1) 페이지 : 페이지 구분
- 2) 모달 : 모달 상세 기능
- 3) 권한 : 페이지별 접근 권한
- 4) 우선순위 : 개발 우선 순위
- 5) 기능 : 기본 기능 구분
- 6) 기능 상세 내용 : 기능 상세 내용 및 동작 과정

No.	페이지	모달	관한	우선순위	기능	기능 상세 내용
1	접속 페이지		모든 사용자	1	페이지 이동	로그인 버튼으로 로그인 페이지로 이동
2	접속 페이지		모든 사용자	1	페이지 이동	회원가입 버튼으로 회원가입 페이지로 이동
3	로그인 페이지		모든 사용자	1	일반 로그인	일반 로그인
4	로그인 페이지		모든 사용자	1	소셜 로그인	카카오, 네이버를 통한 소셜 로그인 : 만약 회원가입이 되어 있지 않다면 소셜 회원가입 페이지로 이동
5	로그인 페이지		모든 사용자	1	자동 로그인 기능	자동 로그인 기능
6	로그인 페이지		모든 사용자	1	최근 로그인 서비스 확인	최근 로그인한 서비스 확인
7	로그인 페이지		모든 사용자	1	아이디/비밀번호 찾기	아이디/비밀번호 찾기 버튼 클릭 시, 아이디/비밀번호 찾기 모달 오픈
8	로그인 페이지		모든 사용자	1	비밀번호 시각화	비밀번호 시각화 선택
9	로그인 페이지		모든 사용자	1	로그인 실패 안내	로그인 실패 시, 실패 내용을 안내
10	로그인 페이지		모든 사용자	1	페이지 이동(뒤로가기)	뒤로가기 아이콘을 클릭하여 이전 페이지(접속 페이지)로 이동
11	로그인 페이지		모든 사용자	1	로그인 성공 시, 페이지 이동	로그인 성공 시, 커뮤니티 페이지로 이동
12	로그인 페이지	아이디/비밀번호 찾기	모든 사용자	1	아이디/비밀번호 찾기 선택	아이디와 비밀번호 중 찾을 내용을 선택
13	로그인 페이지	아이디/비밀번호 찾기	모든 사용자	1	아이디 찾기	이름, 이메일, 성년월일을 입력 받아, 아이디를 안내
14	로그인 페이지	아이디/비밀번호 찾기	모든 사용자	1	비밀번호 찾기	아이디를 입력받아, 이메일로 임시 비밀번호를 보냈다는 문구 안내
15	로그인 페이지	아이디/비밀번호 찾기	모든 사용자	1	모달 종료	닫기 버튼으로 모달 종료
16	회원가입 페이지		비회원	1	일반 회원가입	일반 회원가입
17	회원가입 페이지		비회원	1	소셜 회원가입	카카오, 네이버 소셜 회원 가입
18	회원가입 페이지		비회원	1	휴대폰 실명 인증	휴대폰 실명 인증
19	회원가입 페이지		비회원	1	만 14세 이상 구분	만 14세 이상 구분
20	회원가입 페이지		비회원	1	아이디 중복 체크	중복 여부를 아이콘으로 표시
21	회원가입 페이지		비회원	1	이메일 인증	회원가입하면 자동으로 인증 이메일이 전송되고, 전송된 링크를 클릭하여 자동 인증
22	회원가입 페이지		비회원	1	비밀번호 확인 기능	비밀번호 확인 기능
23	회원가입 페이지		비회원	1	개인 정보 처리 방침	개인정보 처리 방침 동의 체크 박스
24	회원가입 페이지		비회원	1	회원가입 실패 안내	회원가입 실패 안내
25	회원가입 페이지		비회원	1	페이지 이동(뒤로가기)	뒤로가기 아이콘을 클릭하여 이전 페이지(접속 페이지)로 이동
26	회원가입 페이지		비회원	1	회원가입 성공 시, 페이지 이동	회원가입 성공 시, 성공 모달이 띄워지고 로그인 페이지로 이동
27	커뮤니티 페이지		공동 사용자	1	로그인 후 첫 페이지 : 채널이 없는 경우	채널 생성 버튼과 채널 코드를 입력할 수 있는 페이지
28	커뮤니티 페이지		공동 사용자	1	채널 생성 버튼	채널 생성 모달 오픈
29	커뮤니티 페이지	채널 생성	공동 사용자	1	채널 생성	채널명, 해시태그를 입력 받음
30	커뮤니티 페이지	채널 생성	공동 사용자	1	해시태그 생성 및 선택 기능	최대 10개 입력이 가능하고 추가 가능
31	커뮤니티 페이지	채널 생성	공동 사용자	1	생성 버튼 클릭 후, 페이지 이동	해당 채널의 공지사항 페이지로 이동
32	커뮤니티 페이지		공동 사용자	1	채널 코드 입력 및 채널 입장	해당 채널의 공지사항 페이지로 이동

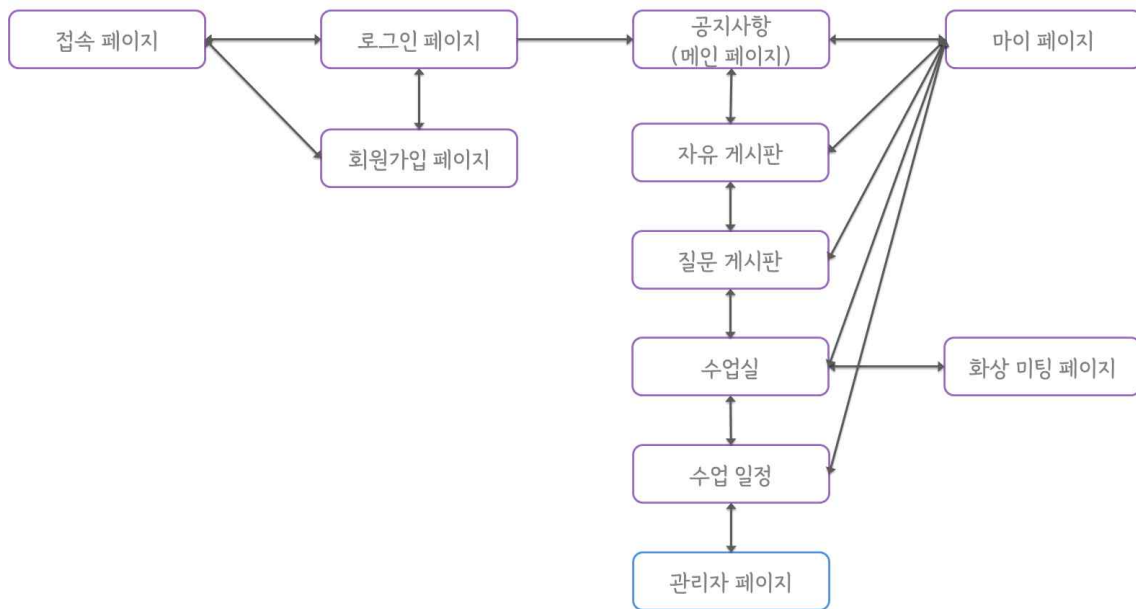
[그림 7] 기능 명세서 일부

6. 서비스 아이콘 및 컬러 팔레트



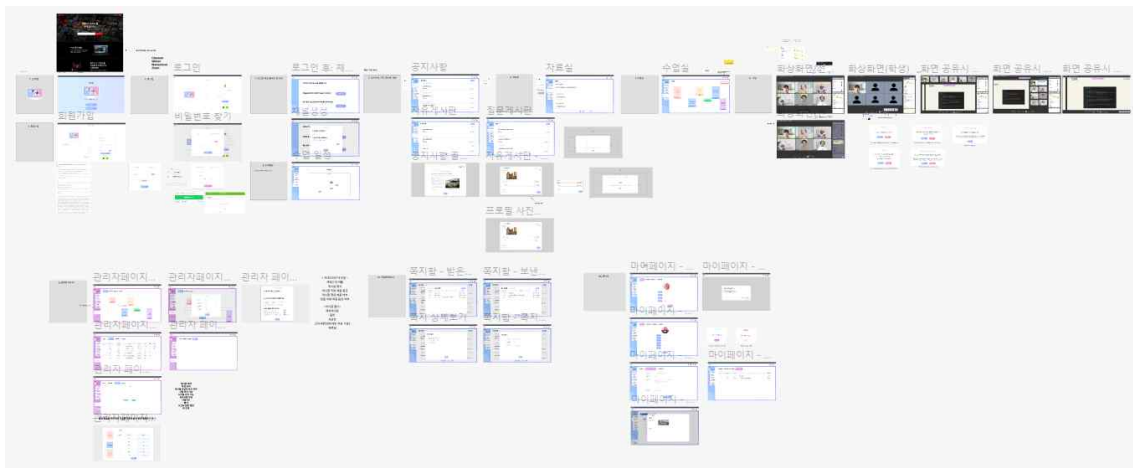
[그림 8] 서비스 아이콘 및 컬러 팔레트

7. 시스템 흐름도

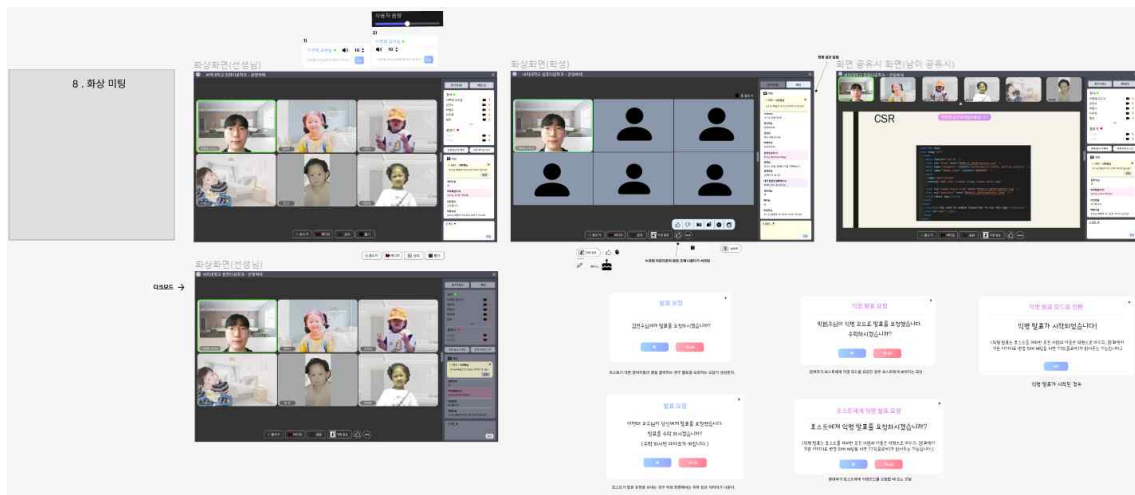


[그림 9] 시스템 흐름도

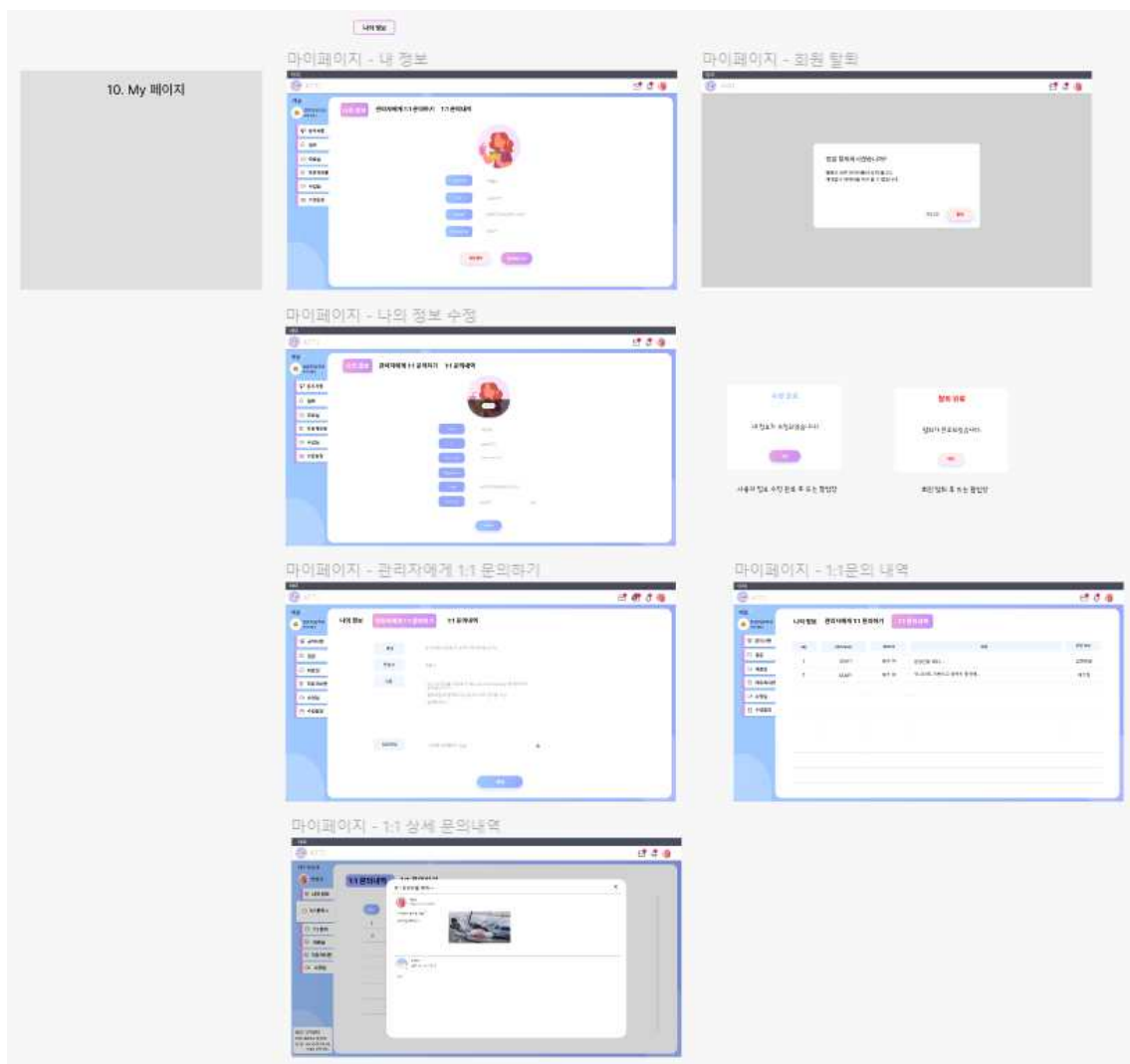
8. 와이어 프레임



[그림 10] 전체 와이어 프레임

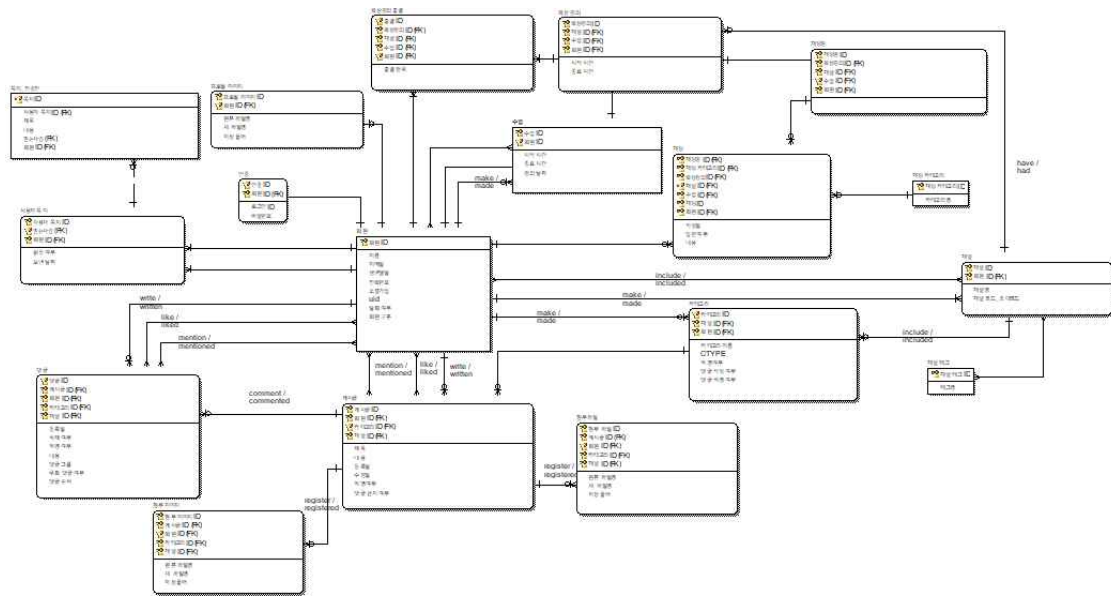


[그림 11] 화상 미팅 페이지 와이어 프레임

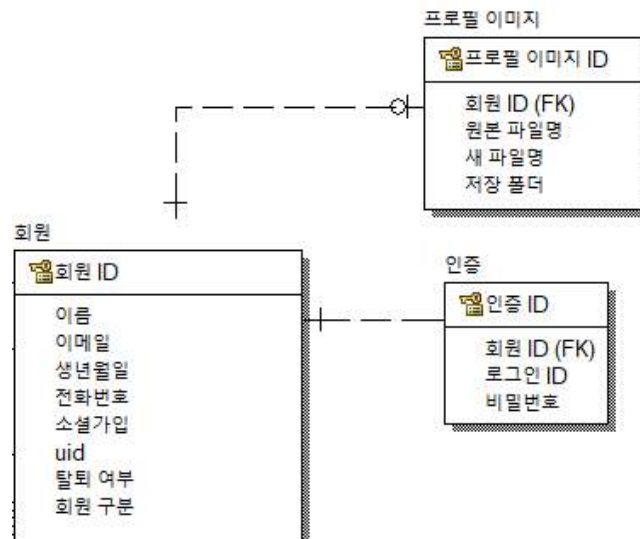


[그림 12] 마이 페이지 와이어 프레임

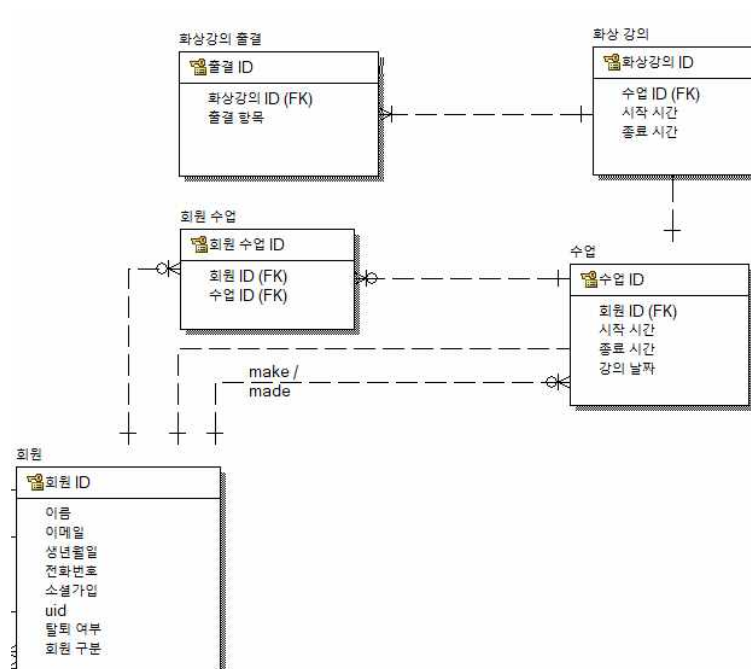
9. ERD



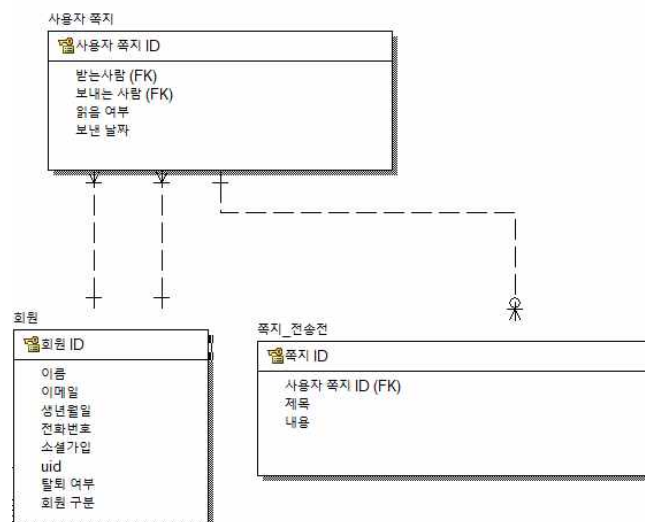
[그림 13] 전체 ERD



[그림 14] 사용자 관련 Entity



[그림 15] 화상 미팅 관련 Entity



[그림 16] 쪽지 관련 Entity

No.	테이블명	PK	FK	logical	physical	type
1	USER	PK		사용자 ID	user_id	VARCHAR(100)
2				사용자명	user_name	VARCHAR(20)
3				이메일	email	VARCHAR(30)
4				생년월일	birth	VARCHAR(20)
5				휴대폰	phone	VARCHAR(20)
6				소셜여부	social	VARCHAR(10)
7				uid	uid	VARCHAR(100)
8				탈퇴여부	user_delete_info	TINYINT
9	AUTH			권한여부	user_role	VARCHAR(20)
10		PK		인증 ID	auth_id	VARCHAR(100)
11		PK	FK	사용자 ID	user_id	VARCHAR(100)
12				비밀번호	pwd	VARCHAR(30)
13	PROFILE			로그인 ID	login_id	VARCHAR(30)
14		PK		프로필 이미지 ID	profile_id	VARCHAR(100)
15		PK	FK	사용자 ID	user_id	VARCHAR(100)
16				프로필이미지 원 이름	profile_origin	VARCHAR(100)
17				프로필 이미지 새 이름	profile_new	VARCHAR(100)
18	MESSAGE			프로필 저장 폴더	profile_folder	VARCHAR(100)
19		PK		메세지 ID	message_id	VARCHAR(100)
20		PK	FK	사용자 ID	to_user_id	VARCHAR(100)
21		PK	FK	사용자 ID	from_user_id	VARCHAR(100)
22		PK	FK	유저 메세지 ID	user_message_id	VARCHAR(100)
23				제목	message_title	VARCHAR(20)
24				내용	message_content	VARCHAR(100)
25	USER_MESSAGE	PK		사용자 메세지 ID	user_message_id	VARCHAR(100)
26		PK	FK	사용자 ID	to_user_id	VARCHAR(100)
27		PK	FK	사용자 ID	from_user_id	VARCHAR(100)
28				읽음 여부	read_info	TINYINT
29				보낸 날짜	send_date	DATE
30		PK		게시글 ID	post_id	VARCHAR(100)
31			FK	카테고리 ID	category_id	VARCHAR(100)
32			FK	채널 ID	depart_id	VARCHAR(20)

[그림 19] Entity 컬럼명 정리

10. API Docs

No.	페이지	권한	기능	Method	URI	Request Header	Params	Request Body	Success Code	Response Header	Response Body
1	로그인 페이지		일반 로그인	POST	/login/normal	X	X		201		depart_table, category_first_data, access token, refresh token, user table, user_depart table
2	로그인 페이지		소셜 로그인	POST	/login/social	X	X		201		depart_table, category_first_data
3	로그인 페이지		아이디 찾기	GET	/find/id		name String, birth String, email String		200		
4	로그인 페이지		비밀번호 찾기	GET	/find/pw		id String		200		
5	회원가입 페이지		소셜 회원가입	POST	/signup/social			Access Token	201		depart_table, category_first_data, access token, refresh token, user table, user_depart table
6	회원가입 페이지		일반 회원가입	POST	/signup/normal			name, id, pw, email, phone number, 생년월일	201		
7	회원가입 페이지		아이디 중복체크	GET	/check/id		id		200		
8	커뮤니티 페이지		채널이 있는 경우 커뮤니티 페이지	GET	/depart/category	application/json, Authorization	depart_name, category table		201		category_post
9	커뮤니티 페이지		채널 입장	GET	/depart	application/json, Authorization	depart_name				category_first_post_table
10	커뮤니티 페이지		전체 게시물 조회	GET	/post	application/json, Authorization	depart_name, category_name, page_number		201		post_list
11	커뮤니티 페이지 관리자		전체 게시물 일괄 삭제	DELETE	/post/delete	application/json, Authorization	depart_name, category_name, post_number				
12	커뮤니티 페이지		단일 게시물 조회	GET	/post/read	application/json, Authorization	depart_name, category_name, post_number		201		post_name, post_content, post_like, post_comment, post_created_at
13	커뮤니티 페이지		단일 게시물 삭제	DELETE	/post/{post_number}/delete	application/json, Authorization	depart_name, category_name, post_number				
14	커뮤니티 페이지		단일 게시물 작성	POST	/post/write	application/json, Authorization		depart_name, category_name, post_writer, post_content, post_anonymous, post_comment_ban	201		post_id

[그림 20] API Docs 일부

11. 현재 진행 상황

1) 1주차(7월 5일 ~ 7월 10일)

- 아이디어 기획 회의
- 기술 스택 선정
- 프로젝트 관련 지식 학습
- 와이어 프레임 초안 작성
- 포지션 분배

2) 2주차(7월 11일 ~ 7월 17일)

- 아이디어 구체화
- 개발 컨벤션 정하기
- 기능 명세서 작성
- 로고 및 와이어프레임 작성
- ERD 설계
- Git Convention을 포함한 팀 내 규칙 선정

3) 3주차(7월 18일 ~ 7월 24일)

- 프로젝트 이름 선정
- ERD 수정
- 기능 명세서 우선 순위 수정
- API Docs 작성
- 요구사항 정리
- 시스템 흐름도 작성
- JIRA 및 Gitlab 사용 시작
- 프로젝트 활용 기술 학습 공유(WebRTC)
- 개발 파트 분배
- 레포지토리 구조 설계

4) 4주차(7월 25일 ~ 7월 31일)

- 개발 환경 설정
- 개발 일정 구체화
- 파트별 개발 시작
- Openvidu 서버 연결 완료
- 1차 배포 및 연결 테스트

12. 추후 진행 계획

1) 5주차(8월 1일 ~ 8월 7일)

- 8월 1일 ~ 8월 3일 : API 70% ~ 80% 완성
- 8월 4일 ~ 8월 12일 : 배포 및 연결 테스트

2) 6주차(8월 8일 ~ 8월 14일)

- 8월 4일 ~ 8월 12일 : 배포 및 연결 테스트
- 8월 12일 ~ 8월 16일 : QA

3) 7주차(8월 15일 ~ 8월 18일)

- 8월 12일 ~ 8월 16일 : QA
- 8월 17일 ~ 8월 18일 : 발표 자료 준비

13. 피드백

1) 1차 팀 미팅(7월 12일 화요일)

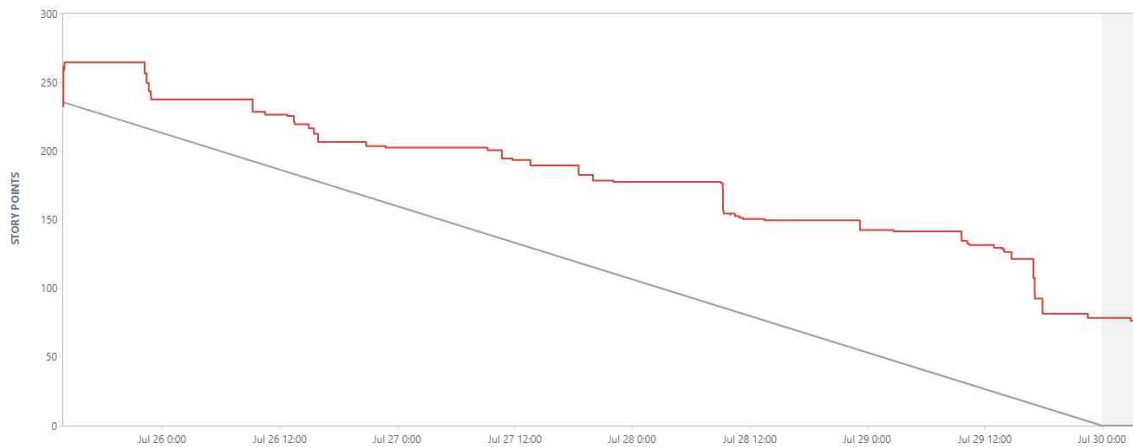
- DB 설계서 : 추후 수정이 될 수 있으므로 메인 기능 위주로 설계 ► 반영
- 기능 명세서 : 메인 기능 위주로 설계 ► 반영 : 우선순위 선정
- WebRTC를 위해 Kurento와 Opnevidu 중 선택
 - 반영 : 빠른 개발을 위해 Opnevidu 선택
- 다음 주까지 기능명세서, 와이어프레임, DB 설계서 작성 ► 반영

2) 2차 팀 미팅(7월 25일 월요일)

- 배포를 고려한 일정 수정 필요 ► 반영 : 금주 수요일부터 배포 작업 시작
- 2주 내 메인 기능의 70% ~ 80% 완성을 목표로 진행 후, 추가 기능 개발 ► 반영 중
- WebRTC를 위해 Kurento와 Opnevidu 최종 선택하기
 - 반영 : socket 이용 경험이 없으므로 Opnevidu로 선택
- Git Convention 및 branch 관리 ► 반영

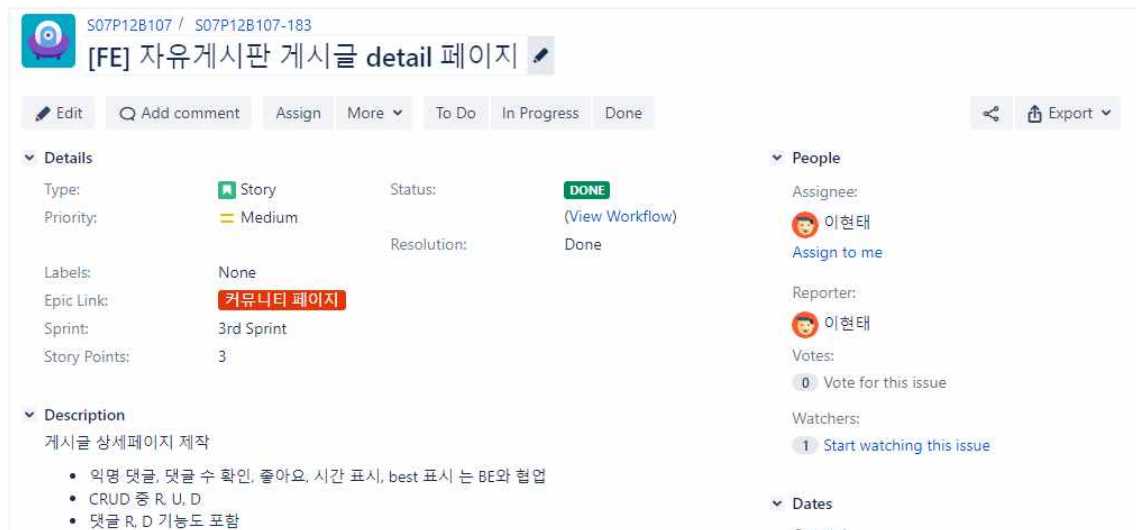
3) 2주차 코치 피드백(7월 22일 금요일)

- 이슈 추가 후, 스프린트를 생성하기 ► 반영



[그림 21] 3주차 Burn Down Chart

- 이슈 생성 시, 설명을 활용하여 세부적으로 한 일 알아볼 수 있게 하기 ▶ 반영



[그림 22] 3주차 이현태님 Story Description

4) 3주차 코치 피드백(7월 29일 금요일)

- 1일 1커밋 실행 ▶ 반영 예정
- smart commit(특정 상태에서 Jira 이슈 상태 변경) ▶ 반영 예정
- MR 템플릿 활용(구체적인 개발 현황 공유) ▶ 반영 예정
- Burn Down Chart 이슈 해결 속도 높이기 ▶ 반영 예정