

240716(화) 코치 세션 - 좋은 기획이란 무엇인가?

1. 좋은 서비스란 무엇인가

1) (앞부분 누락)

(앞부분 누락)

2) 좋은 서비스는 사람을 끌어당긴다

2. 좋은 서비스는 사람을 끌어당긴다

사람을 불러오는



“

제품이 있는 것처럼
이메일 신청 페이지 오픈

3일 동안 1000통의 이메일

잘 될 제품은, 예쁘지 않아도,
노출이 적어도 잘 뒀을 깨달았다

- 토스팀 이승건 리더

”

2. 좋은 서비스는 사람을 끌어당긴다

사람을 불러오는



수요 검증하기



루틴 관리 2주 체험
2일만에 63명 지원!

검증 비용 0원

개인 채널 클릭 대비 전환 8%
다섯 채널 클릭 대비 전환 20%

“

이들만에
엑셀 시트로 MVP 구현...

MVP 임에도 많은 수요...

App 출시 당시 유저 7000명
(마케팅비 총 68,065원)

- 마이루틴 옥민송 대표

”

사람을 불러오는

“

“왜”를 답할 수 있는 서비스는

화려하지 않아도

사람을 불러온다

”

3) 좋은 서비스는 문제를 해결한다

3. 좋은 서비스는 문제를 해결한다



“

신기한 매장, 그 이상의 가치를 주지 못한 아마존 고

기술의 참신함 외에 식료품점이 갖춰야 할
강점이 보이지 않았다

식료품점을 방문하는 고객들에게 정말
중요한 가치를 제대로 파악하지 못한 결과...

”

3. 좋은 서비스는 문제를 해결한다



“

개발 함정

실제로 존재하지 않는 문제점을
존재하는 것처럼 해결하려고 하는 것

”

3. 좋은 서비스는 문제를 해결한다

문제를 해결하는 '책'이 아닌

'실질적인' 문제를 해결할 수 있어야 한다

문제를 해결하기 위해

오히려 더 불편해지고 있지는 않은가?

서비스의 가치 = 고통의 크기 X 사람의 수

IN SSAFY

IN SSAFY 

6주 뒤에 "완성된" 서비스

팀원 모두가 자신이 맡은 "기술"이 아닌

"완성될" 서비스를 바라보고 달려가야 한다

IN SSAFY 

우리는 문제를 해결하는 사람들이다

도전적인 서비스를 겁내지 말자

2. 기획은 어떻게 해야 하는가

II 기획은 어떻게 해야 하는가?



아이디어 구상



와이어프레임



기능 구체화

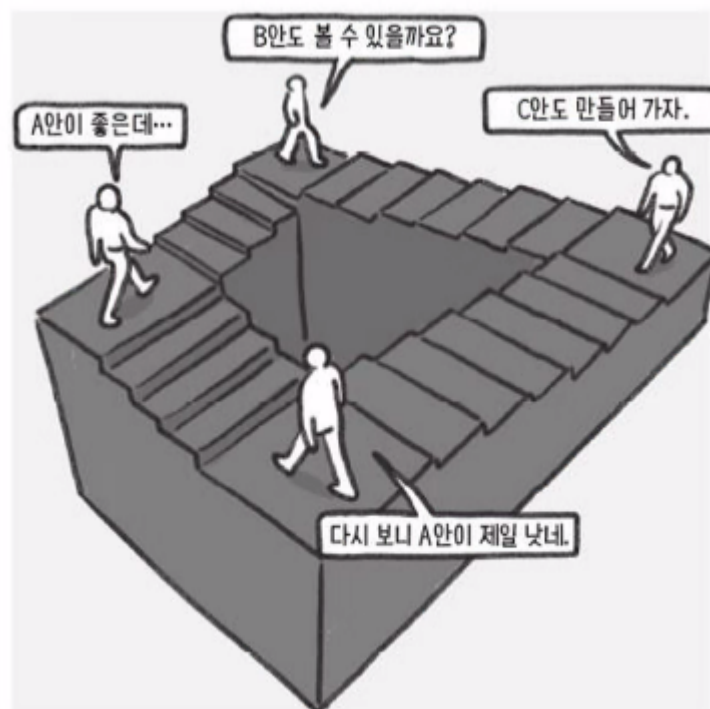


타당성 검토



기술 스택 선정

1) 아이디어 구상



브레인 스토밍

대단한 생각 하나가 아닌

그저 그런 아이디어 100개를 내는 것을 목표해보자

아이디어의 질은 양에 비례한다는 생각에 기초

이후 아이디어를 조합해보자

다른 아이디어에 편승해 조립해보자!

약점을 지적하며 “왜 안될지” 만을 이야기 할 때,

금세 무력해지는 경향이 있다

- 에이드리언 슬라이워츠키 (디맨드 저자)



PIXAR의 피드백 기본 룰

“플러싱 (Plussing)”

상대의 아이디어를 개선시킬 수 있는 건설적인

피드백을 더해 줄 수 있을 때, 비판도 할 수 있다

비판만 하는 것이 아닌, 발전 시킬 수 있어야 한다(Plus).

상대의 아이디어를 “일단 긍정” 하고

발전시킬 수 있는 방법을 생각해보기!



트리즈스토밍

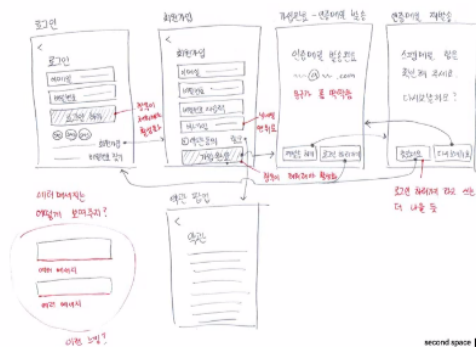
SCAMPER 각 말의 의미와 활용 예시	
SCAMPER 각 말의 의미	
Substitute 대체하기	바다의 토끼를 해마로 [다시, 토끼를 해마로 대체하면?]
Combine 결합하기	가게와 호텔을 결합하면?
Adapt 적용하기	말라가면 죽는 토끼를 만든 동물 전복의 유혹하기
Modify 수정하기	PC방에서 게임을 하는 방법을 동물로 바꾸면?
Put to other uses 다르게 사용하기	계통을 바꾸어 새로운 종류의 토끼를 만든 동물로 만들면?
Eliminate 제거하기	바다에서 토끼를 없애면?
Reverse 반전하기	말라가면 죽을, 전체 토끼를 죽일 수 있는 토끼를 만들면?

스캠퍼 발상법

2) 와이어 프레임

와이어프레임(Wireframe)

웹 앱의 골격이나 사용자 인터페이스(UI) 및 핵심 기능을 보여주기 위해 선과 도형으로 만든 간단한 형태 또는 프레임



서비스 시나리오 작성

유저 플로우(User Flow) 작성

누락된 기능 확인

3&4) 기능 구체화 & 기술 스택 선정



기능 구체화

유저 플로우를 바탕으로 화면 별 기능 구체화



기술 스택 선정

기능을 구현하기 위한 알맞은 기술 스택 선정

5) 타당성 검토

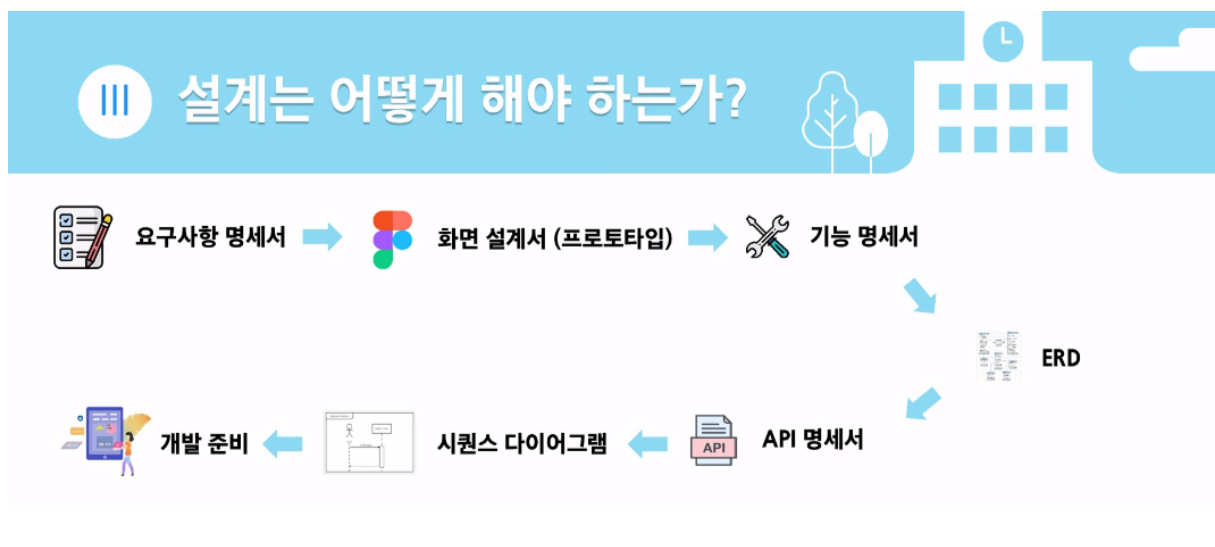
타당성 검토

6주라는 짧은 개발 시간

현재 팀원들의 역량으로

어디까지 개발할 수 있을 것인가?

3. 설계는 어떻게 해야 하는가?



1) 요구사항 명세서

요구사항 명세서

요구사항을 분석하여 명확하고 완전하게 기록하는 것

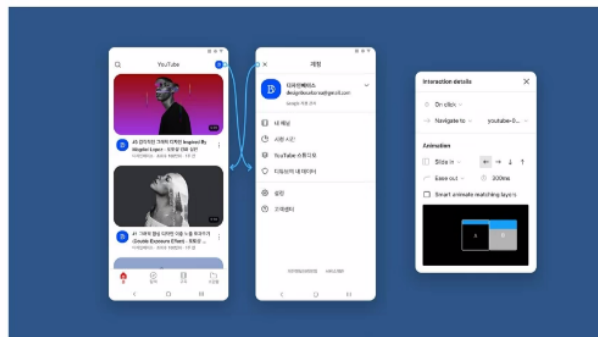
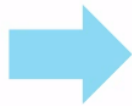
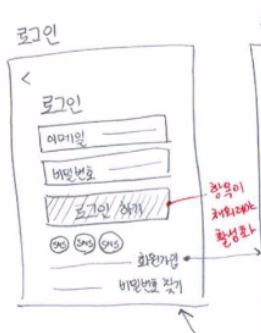
소프트웨어 시스템이 수행해야 할 모든 기능과 구현상의 제약 조건에 대해 개발자와 관련자(클라이언트, 기획자, 경영진 등)가 합의한 스펙

분류	기능명	기능 설명	비고
회원가입/로그인	회원가입	아이디, 비밀번호, 비밀번호 확인, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수
	로그인	아이디, 비밀번호, 로그인 버튼	아이디, 비밀번호 필수
	회원가입/로그인	아이디, 비밀번호, 로그인 버튼	아이디, 비밀번호 필수
관리	회원정보 찾기	회원정보 찾기, 비밀번호 찾기, 이메일 찾기	회원정보 찾기, 비밀번호 찾기, 이메일 찾기
	회원정보 찾기	회원정보 찾기, 비밀번호 찾기, 이메일 찾기	회원정보 찾기, 비밀번호 찾기, 이메일 찾기
	회원정보 찾기	회원정보 찾기, 비밀번호 찾기, 이메일 찾기	회원정보 찾기, 비밀번호 찾기, 이메일 찾기
회원정보	회원정보	회원정보, 비밀번호, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수
	회원정보	회원정보, 비밀번호, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수
	회원정보	회원정보, 비밀번호, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수
회원정보	회원정보	회원정보, 비밀번호, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수
	회원정보	회원정보, 비밀번호, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수
	회원정보	회원정보, 비밀번호, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수
회원정보	회원정보	회원정보, 비밀번호, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수
	회원정보	회원정보, 비밀번호, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수
	회원정보	회원정보, 비밀번호, 이메일, 주소, 생년월일, 핸드폰번호, 지역별, 본인확인필름	비밀번호, 아이디 중복확인, 비밀번호 6자 이상, 주소, 생년월일 필수

2) 화면 설계서 (프로토타입)

화면 설계서 (프로토타입)

팀원이 다같이(백엔드, 프론트엔드 구분 없이) 참여!!!!



3) 기능 명세서

기능 명세서

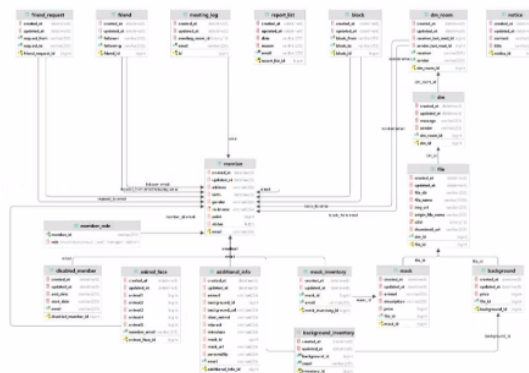
아주 상세하게 작성해야 한다!

기능명세서					
호 구분	주 기능	보조 기능	보조 설명	가능 여부	비고
1 회원 가입	1.1 이메일 가입			가능	
	1.2 SNS 로그인	1.2.1 카카오로그, 네이버, 구글		가능	
2 회원 기록	2.1 개인 회원 내시보드	2.1.1 학교출생 연인 스프링 기록		가능	
		2.1.2 이는 등 회원 기록 / 다 라인 회원 기록 (회원 내 링크)	장 기록	가능	데이터가 없을 경우
		2.1.3 회원 페이지	- 내가 가진 회원ID를 기준으로 페이지 제공	가능	
	2.2 회원 기록 피드 작성	2.2.1 이미지 업로드	- 디바이스 종류에 의존	가능	
		2.2.2 회원 태그 기능	- 비로 할당 (카메라 호출)	가능	
			- 이미지 미리 보여주는 위치와 태그	가능	오류가 될 경우
			- 태그 선택 시 해당 기록 확인 가능		
			- 회원 상세로 이동하는 링크 제공		
		2.2.3 태그된 회원 공개 기록	- 5월 별칭 불가(월)	가능	
			- 텍스트 필드(선택)		
3 회원 기록 피드 공유하기	3.1 인스타그램 공유	3.1.1 인스타그램 공유	작성된 게시글 이미지와 하위 이미지 업로드	확인완료	자녀 앱 인스타그램과 공유 링크
	3.2 카카오톡 공유	3.2.1 카카오톡 공유	작성된 게시글 이미지와 하위 이미지 업로드	확인완료	
3 회원 검색	3.1 회원 이미지 검색	3.1.1 키워드와 키워드로 검색하여 회원 검색	작성된 게시글 이미지와 하위 이미지 업로드	불가능	API 확인

4) ERD

ERD (Entity-Relationship Diagram)

팀원이 다같이(백앤드, 프론트앤드 구분 없이) 참여!!!!



5) API 명세서

API 명세서

API의 기능, 동작, 사용법 등을 나타낸 문서

<input type="checkbox"/> * 확인 중	user	회원 탈퇴	DELETE	주영 이		200 OK 400 Bad Request
* 확인 중	user	회원 정보 조회	GET	JOOHEE		200 OK 400 Bad Request
* 확인 중	user	회원 정보 조회	GET	주영 이		200 OK 400 Bad Request
* 확인 중	product	상품 목록 조회	GET	최승준	{ { name: String, grade: String, imgUrl: String } }	200 OK 400 Bad Request
* 확인 중	product	상품 등록	POST	최승준	{ name: String, grade: String, image: MultiPart }	200 OK 400 Bad Request
* 확인 중	product	상품 구매	POST	최승준	{ name: String, grade: String }	200 OK 400 Bad Request

이름 및 설명

- 이름

- 설명

기본 정보

- Host 주소

- Endpoint

- HTTP Method

API 명세서

API의 기능, 동작, 사용법 등을 나타낸 문서

<input type="checkbox"/> * 확인 중	user	회원 탈퇴	DELETE	주영 이		200 OK 400 Bad Request
* 확인 중	user	회원 정보 조회	GET	JOOHEE		200 OK 400 Bad Request
* 확인 중	user	회원 정보 조회	GET	주영 이		200 OK 400 Bad Request
* 확인 중	product	상품 목록 조회	GET	최승준	{ { name: String, grade: String, imgUrl: String } }	200 OK 400 Bad Request
* 확인 중	product	상품 등록	POST	최승준	{ name: String, grade: String, image: MultiPart }	200 OK 400 Bad Request
* 확인 중	product	상품 구매	POST	최승준	{ name: String, grade: String }	200 OK 400 Bad Request

Request

- Header

- Body

- Parameter

- Example

Response

-Header

- Example

- Body

API 명세서

API의 기능, 동작, 사용법 등을 나타낸 문서

<input type="checkbox"/> * 확인 중	user	회원 탈퇴	DELETE	주영 이		200 OK 400 Bad Request
* 확인 중	user	회원 정보 조회	GET	JOOHEE		200 OK 400 Bad Request
* 확인 중	user	회원 정보 조회	GET	주영 이		200 OK 400 Bad Request
* 확인 중	product	상품 목록 조회	GET	최승준	{ { name: String, grade: String, imgUrl: String } }	200 OK 400 Bad Request
* 확인 중	product	상품 등록	POST	최승준	{ name: String, grade: String, image: MultiPart }	200 OK 400 Bad Request
* 확인 중	product	상품 구매	POST	최승준	{ name: String, grade: String }	200 OK 400 Bad Request

API 명세서만 보고도

사용할 수 있어야 한다



API 명세서만 보고도 사용할 수 있어야 한다!

API 명세서

협업의 자세 1

API 설계는 먼저 하자!

Ex) 이미 코드 다 짰는데…?

API 명세서

협업의 자세 2

API 설계는 같이 하자!

Ex) 그냥 좀 하면 안돼?

API 명세서

협업의 자세 3

변경사항을 공유하자!

Ex) 혹시 API 뭐 바꿨어?

API 명세서

협업의 자세 4

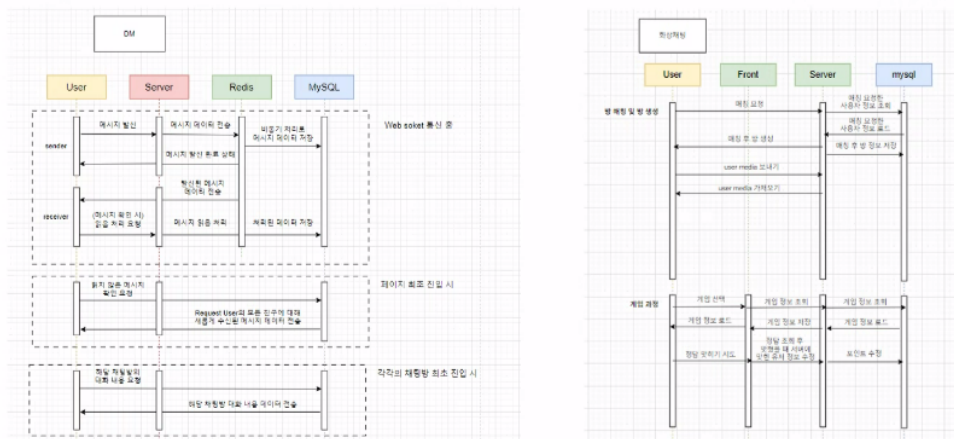
상대방 입장에서 설명하자!

Ex) @RequestParam(String)

6) 시퀀스 다이어그램

시퀀스 다이어그램

시간의 흐름에 따라 객체들이 메시지를 주고받는 상호 작용을 명세히 나타내는 다이어그램



7) 개발 준비

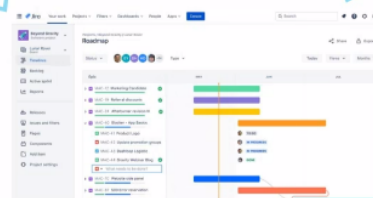
개발 준비



팀 컨벤션

~주차 스프린트에
~까진 되어있어야

출시할 수 있겠군!



일정 관리 (Gantt 차트)

마무리

“

탄탄한 서비스는

탄탄한 기획에서 생긴다

”