

와플 Waple : 우리만의 플레이스

한개의 와플 빵 위에 각자의 취향에 맞는 토핑을 올려 먹듯,
하나의 지도에 모임별 기록을 담아내는 와플 서비스!!!

제작 : Team Toppings

박혜진(팀장) : 사과잼

강재구 : 허니잼

최윤주 : 루비잼

오인호 : 노잼

김보훈 : 꿀잼

특징

- 각자 모임별로 만드는 맞춤형 지도
- 그룹 내 멤버들끼리 지도 공유
- 쉽게 약속 잡기
- 방문했던 모든 장소에 대한 리뷰 남기기
- 직접 지정한 테마 별 지도 보기

기능적 요구사항

주요 기능	세부 기능	설명
회원 관리	회원 가입	<ul style="list-style-type: none"> - 카카오 간편 가입 - 초기 닉네임은 카카오톡에 설정되어 있는 것 가져옴
	회원 정보 조회	<ul style="list-style-type: none"> - 그룹 리스트 - 닉네임 - 유저 캘린더
	회원 정보 수정	<ul style="list-style-type: none"> - 닉네임만 수정 가능함
	로그인	<ul style="list-style-type: none"> - 카카오 로그인
	로그아웃	
	회원 탈퇴	<ul style="list-style-type: none"> - 가입되어있는 그룹이 있으면 모두 탈퇴 처리
그룹 관리	그룹 생성	<ul style="list-style-type: none"> - 그룹 이름 설정 필요 - 회원가입 시 자동으로 개인 그룹 생성
	그룹 이름 변경	
	그룹에 멤버 추가	<ul style="list-style-type: none"> - 초대 링크 보내기 -> 해당 링크를 연 사용자에게 로그인 요청 - 그룹장만 가능
	그룹 탈퇴	<ul style="list-style-type: none"> - 리뷰/북마크(사용자 본인이 등록한 것)는 일괄 삭제할지 말지 선택 - 본인이 그룹장일 경우, 그룹장을 누구에게 양도할지 선택 필요
	그룹 삭제	<ul style="list-style-type: none"> - 그룹장 이외 가입되어있는 멤버가 없어야 함
	그룹장 양도	<ul style="list-style-type: none"> - 그룹 내 멤버로 그룹장을 변경
	알림	<ul style="list-style-type: none"> - 그룹에 멤버 / 리뷰 / 댓글 등 추가 시 - 카카오톡 메시지로 보내기 - 페이지 내 자체 알림 창
테마	테마 추가	<ul style="list-style-type: none"> - 사용자가 원하는 아이콘 및 이름으로 추가

관리(그룹 내 테마)	테마 조회	- 테마 조회 (그룹별)
	테마 수정	- 테마명 - 테마 아이콘
	테마 삭제	
	테마 보이기	- 체크박스를 이용해서 한 지도에 보고 싶은 테마만 보이게 함
북마크 관리	북마크 추가	- 해당 그룹 멤버가 원하는 장소 추가 (테마별로) - 멤버 중 누가 추가한 것인지 표시
	북마크 조회	- 그룹별 조회 - 테마별 조회
	북마크 삭제	
약속 관리	약속 추가	- 일시 - 그룹 - 약속명
	약속 조회	- 일시 - 그룹 - 약속명
	약속 수정	- 일시 - 약속명
	약속 삭제	
	약속 잡을 장소 조회	- 본인이 속해있는 그룹 내 약속 잡을 모든 장소 조회
	약속 장소 투표	
	현 위치에서 약속장소로 길찾기	- 카카오 API를 사용해 그룹 멤버별 현 위치에서 길찾기 제공(카카오지도로 리다이렉트)
리뷰 관리	리뷰 생성(남기기)	- 해당 그룹이 갔던 장소 후기 및 리뷰 - 영수증 또는 그룹장이 수동으로 지출 내역 업로드 - 모임 사진 및 영상 업로드(용량 제한)
	리뷰 조회	- 그룹 내 모임원이 남긴 리뷰 열람 기능
	리뷰 수정	- 내용 수정

		<ul style="list-style-type: none"> - 제목 수정 - 미디어 수정
	리뷰 삭제	
지도 조회 기능	그룹으로 지도 조회	- 그룹에 속한 장소 정보 지도에 표시
	테마로 지도 조회	- 그룹 테마 내 속한 장소 정보 지도에 표시
관리자 기능	회원 목록 조회	- 이메일, 최근 방문 일자 조회
	휴면 회원 처리	- 방문한지 3년 초과 시
	휴면 처리 안내	- 휴면 처리 일자로부터 3개월, 1개월 전 안내
문의 기능	문의하기	<ul style="list-style-type: none"> - 문의 작성하여 이메일로 보낼 수 있는 기능 - mailto:waple@gmail.com 같은 느낌으로! - 이메일 템플릿 제공 (사용 디바이스, 문제 발생 일시, 문제 내용 등)
캘린더	달력	- 자신이 속한 그룹 내 모든 일정을 캘린더 형태로 제공

기능 요구사항 분석서

현위치에서 길찾기	- https로 접속했을때만 가능(Kakao API에 명시되어 있음)
약속 잡을 장소 조회	- 투표 중인 장소만 조회 가능
북마크 조회	- 전체 /갔던 곳 / 가고 싶은 곳은 탭으로 구분
그룹장 변경	- 그룹에 가입되어있는 멤버로만 그룹장을 변경할 수 있다.

비기능적 요구사항

허용되지 않은 접근 차단	<ul style="list-style-type: none">- 로그인하지 않은 브라우저에서 로그인 한 유저 전용 페이지에 접근- 해당 그룹의 멤버가 아닌 사용자가 그룹 멤버 전용 페이지에 접근- 그룹의 관리자가 아닌 사람이 그룹 관리자 전용 페이지에 접근- 상기 모든 경우 접근을 차단한다. (error page로 redirect)
언제 어디서나 접근 가능	<ul style="list-style-type: none">- 컴퓨터, 모바일 등 다에서 웹에 접근이 가능해야 한다.- 둘의 사용성에 큰 차이가 없어야 한다.
무결성, 완전성	<ul style="list-style-type: none">- 모든 작업은 무결성과 완전성을 보장해야 한다.- ex) DB transaction

기술 스택

Frontend	Vue.js (vue-router, vuex, vuetify)
Backend	Spring boot (java), REST API
DB	Maria DB
기타	gitlab, jira, docker

Convention

Frontend

Javascript -> Airbnb esLint

[참고](#)

<https://github.com/airbnb/javascript>

[번역](#)

Backend

Java -> <https://naver.github.io/hackday-conventions-java/>

스웨거 ApiResponse

```
@ApiResponse(code = 200, message = "[메서드] 성공"),  
@ApiResponse(code = 401, message = "로그인 후 이용해 주세요"),  
@ApiResponse(code = 403, message = "권한이 없습니다"),  
@ApiResponse(code = 404, message = "[메서드] 실패")
```

Jira Issue

<https://taes-k.github.io/2019/12/07/sw-jira-scrum/> 참고자료

Epic - Story - SubTask 3단계로 구성

Task 하나 당 commit 하나

Git

Commit

([참고 1](#), [참고 2](#))

[예약어] 지라 이슈 번호 : 제목(한 일)

제목의 길이는 최대 40글자까지 한글로 간단 명료하게 작성

제목을 작성하고 반드시 빈 줄 한 줄을 만들어야 함

제목에 .(마침표) 금지

예약어 리스트

feature : 새로운 기능
bug : 버그 수정
update : 비즈니스 로직 변경
docs : 문서 (문서 추가, 수정, 삭제)
test : 테스트
etc : 기타 변경사항

세부 내용 기술

내용의 길이는 한 줄당 60글자 내외에서 줄 바꿈. 한글로 간단 명료하게 작성
어떻게 보다는 무엇을, 왜 변경했는지를 작성할 것 (필수)

Merge Request

(참고: [깃랩에서 merge request template 만드는 방법](#), [convention](#))

[제목]

- 한 커밋일 경우, 커밋 메시지
- 아닐 경우, 커밋들을 summarise한 문장

[내용] 기능은 Feature.md, 버그는 Bug.md

Git flow

(참고 : [우아한 형제들 기술 블로그](#))

항상 유지되는 메인 브랜치들(master, develop)과 일정 기간 동안만 유지되는 보조 브랜치들(feature, release, hotfix)이 있습니다.

- master : 제품으로 출시될 수 있는 브랜치
- develop : 다음 출시 버전을 개발하는 브랜치
- feature : 기능을 개발하는 브랜치
- release : 이번 출시 버전을 준비하는 브랜치

Branch name

master, release, develop은 고정

feature_기능명

(기능명은 기능명세서 참조)

Tag (version manage)

추후 tag 사용할 일이 생길 때 정하기