

FooDa: Designing and Planning

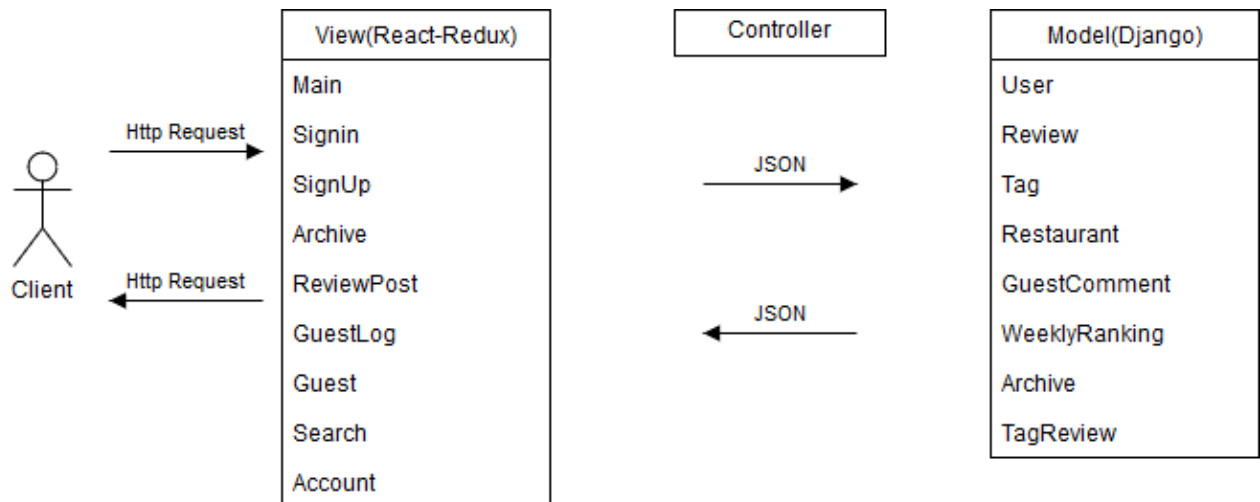
6조: 신수웅, 이민후, 한상현, 한재훈
20190520 - Third revision

Revision Bullet List

- 2019/05/06
 - E-R diagram - "User", "Tag", "TaggedItem" models updated
 - Relationship text를 관계 선 위에 추가하였다.
 - Relation schema diagram - "User", "Tag", "TaggedItem", "Review", "GuestComment", "WeeklyRanking" models updated
 - Backend design - RESTful API is updated
 - ID로 로그인하는 것에서, Email로 로그인하는 것으로 변경하였다.
 - 회원가입할 때 ID를 입력하지 않도록 변경하였다.
 - archive page로의 route를 'id/archive'에서 'nickname/archive'로 변경하였다.
 - Button atom의 설명을 수정하였다.
 - 상기한 내용에 맞게 Molecules와 Relations Diagram을 수정하였다.
- 2019/05/20
 - Backend design - Restful API is updated : review 관련 url을 수정하였다.
 - Implementation Plan을 변경하였다.
 - Frontend에서 PostPage 관련 링크를 변경하였다.

System Architecture

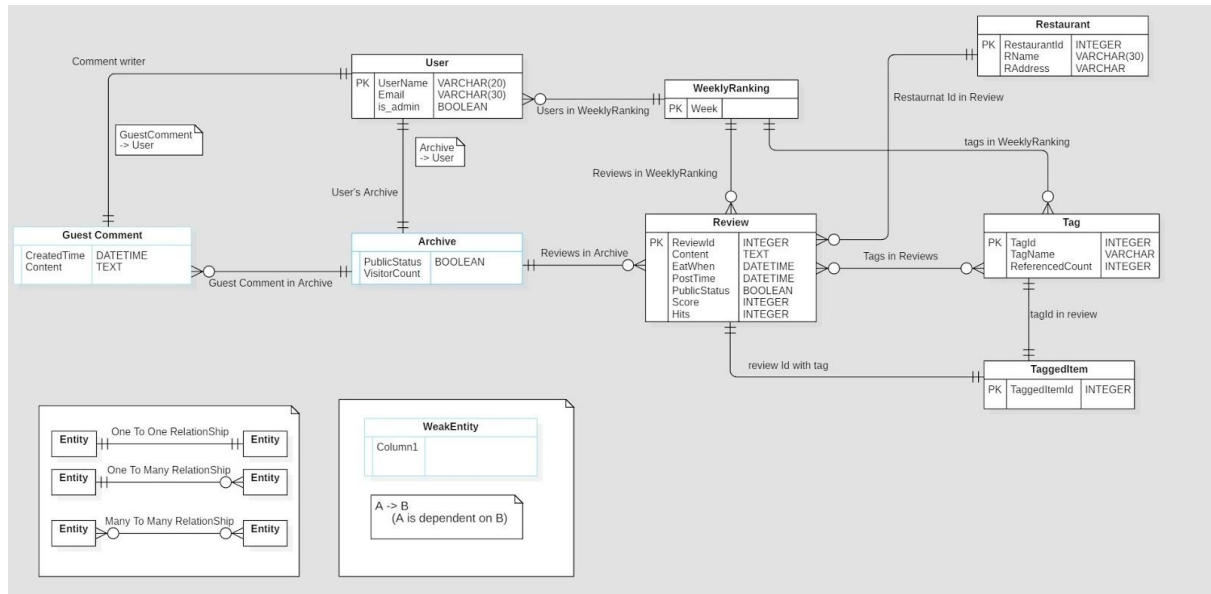
MVC



뷰 9개 모델 8개로 구성되어있다.

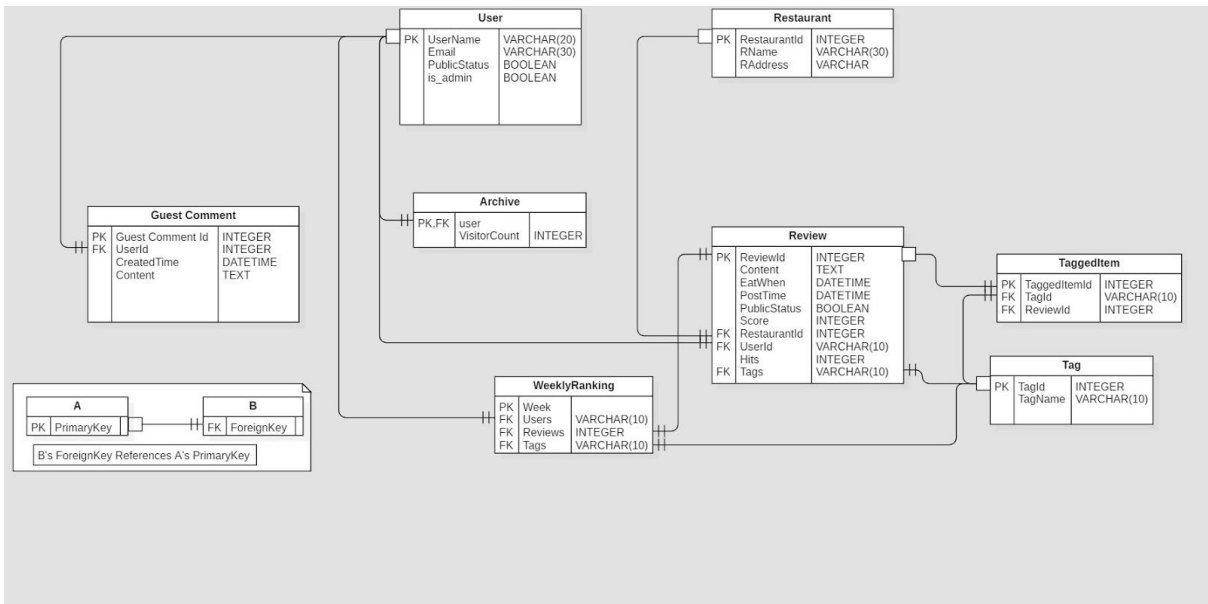
Model

아래의 그림은 모델 디자인을 위한 E-R (Entity Relationship) diagram이다.



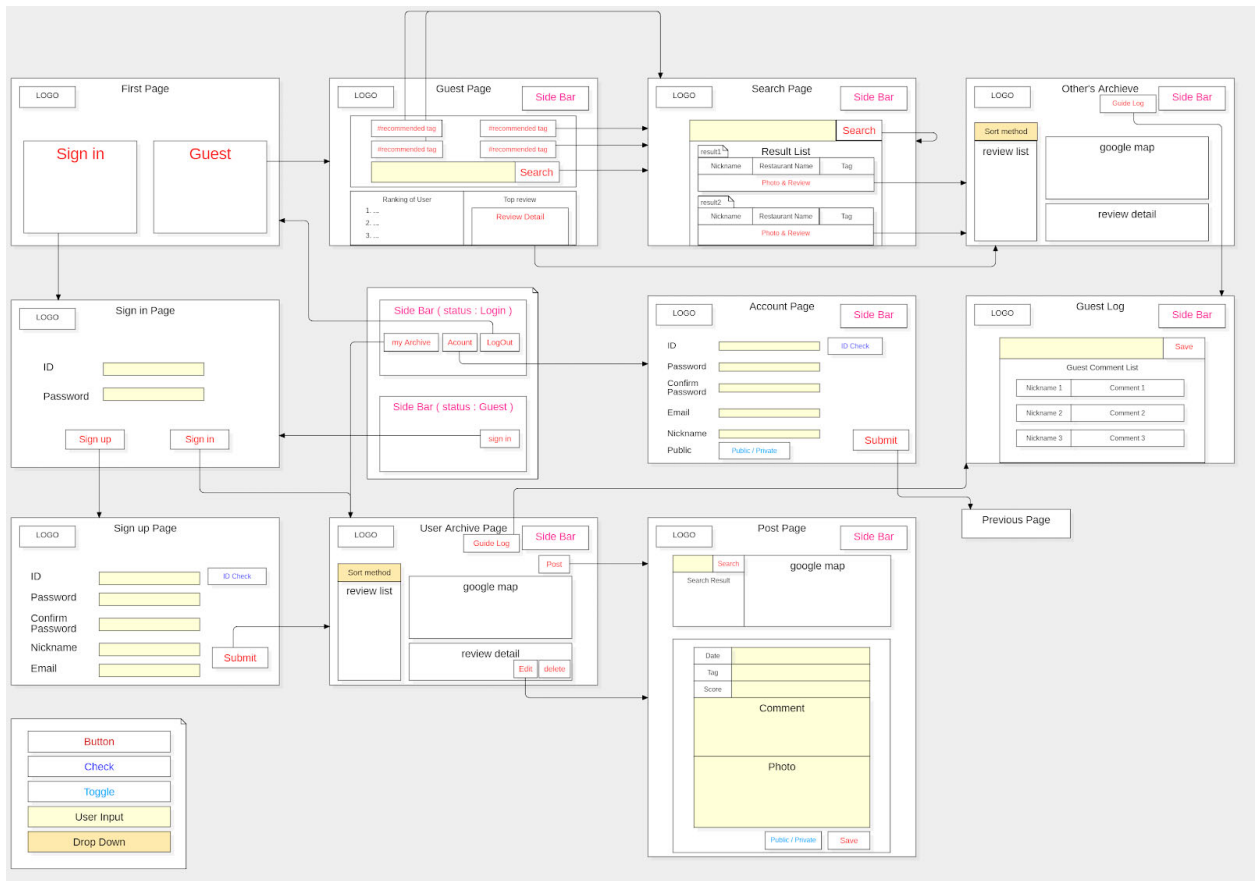
직사각형은 Entity set을 의미하며 Entity들 사이의 관계는 선을 통해 나타내었으며 선의 모양에 따라 One To One, One To Many등의 관계를 나타낸다. 선 위에 있는 text는 Entity 사이의 Relation을 나타낸다. Entity 속성은 직사각형 내부에 나열되어 있으며, 속성 이름 좌측에는 그 속성이 Primary key(PK), Foreign Key(FK) 인지 나타내는 표시가 되어있고 우측에는 속성의 type과 크기가 표시되어있다.

아래의 그림은 E-R diagram을 기반으로 만들어진 Relation schema diagram이다.



직사각형은 Entity set을 의미하며 Entity들간의 관계는 선을 통해 나타내었다. Entity 속성은 직사각형 내부에 나열되어 있으며, 속성 이름 좌측에는 그 속성이 Primary key(PK), Foreign Key(FK) 인지 나타내는 표시가 되어있다. 우측에는 속성의 type과 크기가 표시되어 있다.

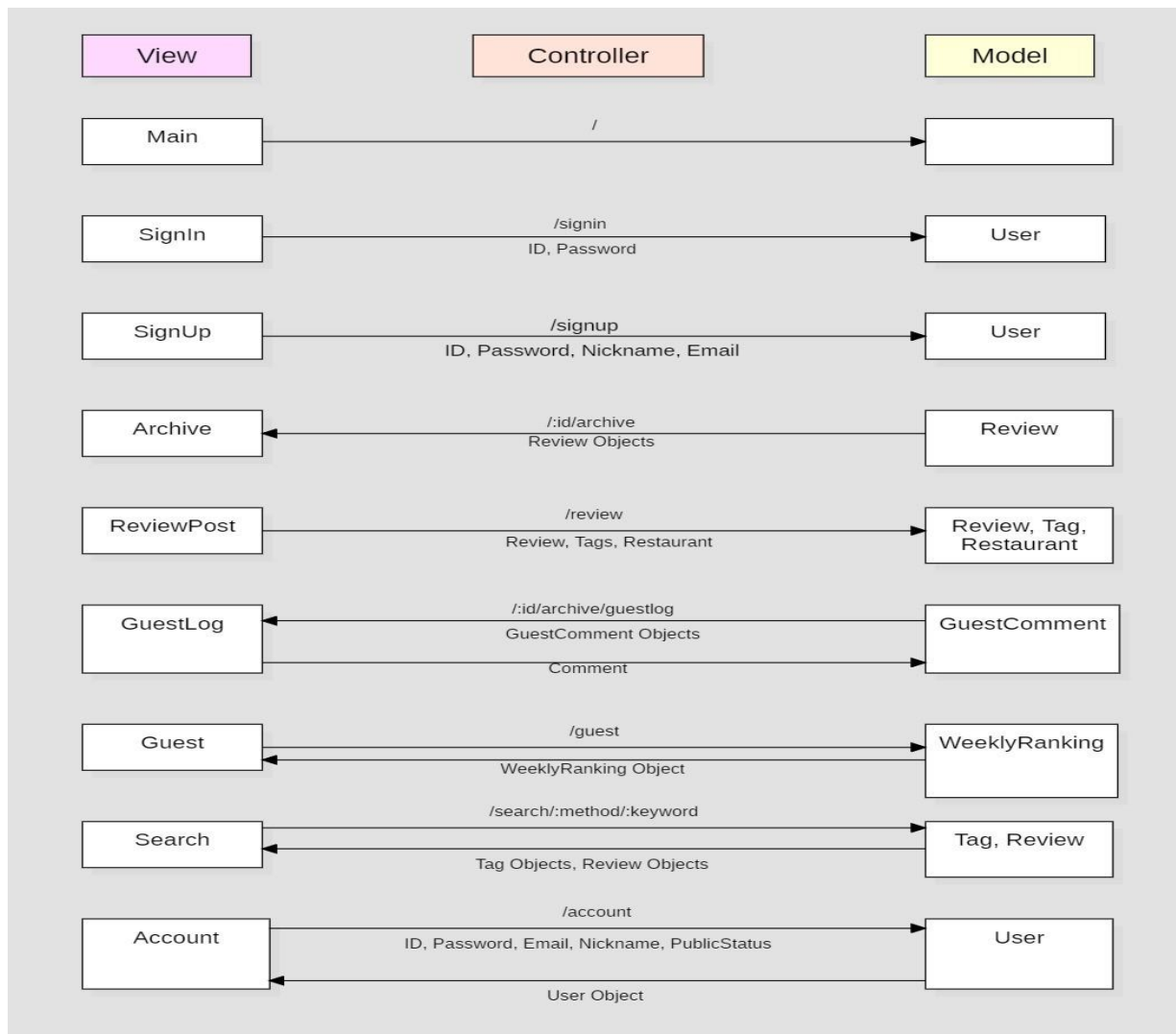
Views



- 메인 페이지('/')
 - 로그인 버튼을 누르면 로그인 화면으로 넘어간다.
 - 게스트 버튼을 누르면 게스트 화면으로 넘어간다.
- 로그인 페이지('/signin')
 - 'email'와 'password'를 user input으로 받는다.
 - 회원가입 버튼을 누르면 회원가입 화면으로 넘어간다.
 - 로그인 버튼을 누르면 이메일과 비밀번호가 맞다면 그 이메일에 해당되는 계정으로 로그인하고, 아니면 로그인이 실패했다고 알린다.
- 회원가입 페이지('/signup')
 - 'email', 'password', 'confirm password', 'nickname'을 user input으로 받는다.
 - id check를 누르면 중복 id가 있는지 확인해서 결과를 화면에 텍스트로 출력한다.
- 아카이브 페이지('/:nickname/archive/')
 - 방명록 버튼을 누르면 방명록 화면('/:nickname/archive/guestlog')으로 넘어간다.
 - '정렬 방식'에는 시간/지역/태그의 세 탭이 있고, 누른 탭에 따라 리뷰 리스트를 정렬한다.
 - 가운데 지도에서 리뷰에 해당하는 식당에 꽂힌 핀을 확인할 수 있다.
 - 리스트에서 리뷰를 클릭하면 리뷰 상세 내용이 아래 나온다.
 - (본인 아카이브일시) 리뷰가 선택되어 있는 상태에서 Edit 버튼을 누르면 리뷰 작성/수정 화면으로 넘어간다.

- f. (본인 아카이브일시) 리뷰가 선택되어 있는 상태에서 Delete 버튼을 누르면 지도/리스트에서 해당 리뷰를 삭제한다.
 - g. (본인 아카이브일시) Post 버튼을 누르면 리뷰 작성/수정 화면으로 넘어간다.
- 5. 리뷰 작성/수정 페이지('/post/:reviewId' or '/post/default')
 - a. 만약 '/post/default'라면 새로운 리뷰를 작성하게 되고, '/post/:reviewId'라면 reviewId에 해당되는 자신의 리뷰 내용을 수정하게 된다.
 - b. 식당 이름등을 input으로 넣고 Search 버튼을 누르면 Google Map과 연동해서 식당 위치/정보를 제공한다.
 - c. Date, Tag, Score, Comment, Photo를 input으로 받는다.
 - d. Public/Private 버튼을 누르면 공개 여부를 설정한다.
 - e. Save를 누르면 리스트에 변화가 반영된 상태로 본인 아카이브 화면으로 돌아간다.
- 6. 방명록('/:nickname/archive/guestlog')
 - a. 글을 input으로 넣고 Save를 누르면 방명록 리스트에 새 항목이 추가된다.
- 7. 게스트 페이지('/guest')
 - a. 추천 태그를 누르면 그 태그의 검색결과 화면으로 넘어간다.
 - b. Search를 누르면 검색어와 검색 기준을 input으로 받아서 검색결과 화면으로 넘어간다.
 - c. 유저 랭킹에서 유저 이름을 클릭하면 오른쪽에 그 유저의 대표 리뷰가 뜬다.
 - d. 대표 리뷰를 클릭하면 그 리뷰가 선택된 상태의 유저 아카이브로 넘어간다.
- 8. 검색 결과('/search/:method/:keyword')
 - a. 결과는 닉네임/식당이름/태그/리뷰 순으로 나온다.
 - b. 닉네임을 클릭하면 아무 리뷰가 선택되지 않은 상태의 유저 아카이브로 넘어간다.
 - c. 리뷰를 클릭하면 그 리뷰가 선택된 상태의 유저 아카이브로 넘어간다.
- 9. 계정 설정('/account')
 - a. password, email, nickname, public여부를 input으로 받아 submit을 누르면 계정설정이 즉시 변경되고 이전 페이지로 되돌아간다.
 - b. Public toggle: 본인 아카이브가 타 유저에게 검색이 가능한지에 대한 여부를 설정한다.

Controller



왼쪽 부분은 view part (frontend)이고 오른쪽 부분은 model part(backend)이다. 왼쪽에서 오른쪽으로 향하는 화살표는 view에서 유저의 input 전송하는 http request를 나타낸다. 그리고 오른쪽에서 왼쪽으로 향하는 화살표는 model에서 data를 전송하는 http request를 나타낸다.

화살표 아래의 JSON data를 전송하는데 사용하는 API는 화살표 위에 표시하였다.

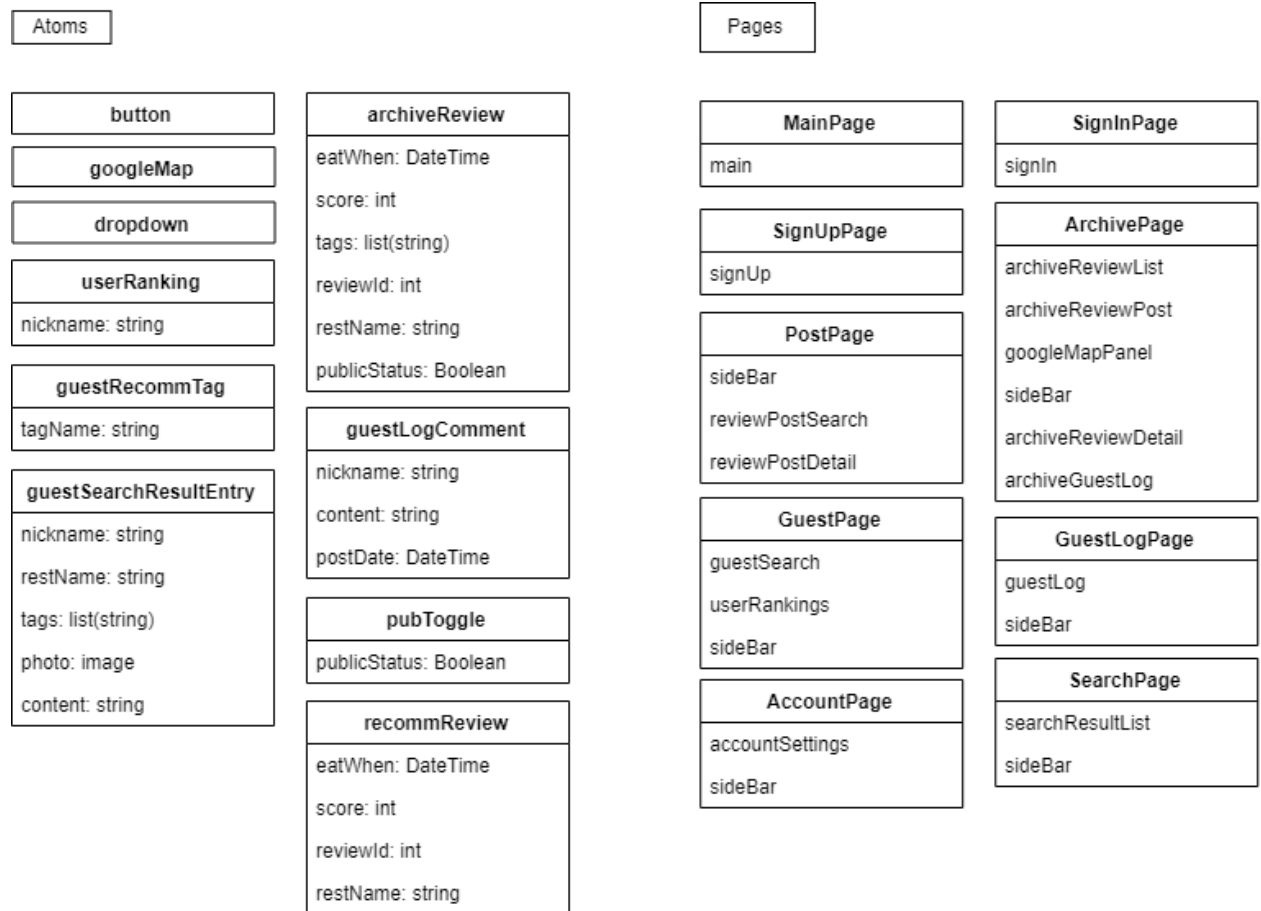
Design Details

Frontend Design

Frontend Components

Molecules

<table><tr><td>main</td></tr><tr><td>signIn: button</td></tr><tr><td>guest: button</td></tr><tr><td>onClickSignIn()</td></tr><tr><td>onClickGuest()</td></tr></table>	main	signIn: button	guest: button	onClickSignIn()	onClickGuest()	<table><tr><td>archiveReviewDetail</td></tr><tr><td>eatWhen: DateTime</td></tr><tr><td>tags: list(string)</td></tr><tr><td>score: int</td></tr><tr><td>comment: string</td></tr><tr><td>photo: image</td></tr><tr><td>delete: button</td></tr><tr><td>edit: button</td></tr><tr><td>onClickDelete()</td></tr><tr><td>onClickEdit()</td></tr></table>	archiveReviewDetail	eatWhen: DateTime	tags: list(string)	score: int	comment: string	photo: image	delete: button	edit: button	onClickDelete()	onClickEdit()	<table><tr><td>signUp</td></tr><tr><td>email: input(string)</td></tr><tr><td>password: input(string)</td></tr><tr><td>confirmPassword: input(string)</td></tr><tr><td>nickname: input(string)</td></tr><tr><td>checkEmail: button</td></tr><tr><td>checkNickname: button</td></tr><tr><td>submit: button</td></tr><tr><td>onClickEmailCheck(email: string)</td></tr><tr><td>onClickNicknameCheck(nickname: string)</td></tr><tr><td>onClickSubmit(email, password, confirmPassword, nickname)</td></tr></table>	signUp	email: input(string)	password: input(string)	confirmPassword: input(string)	nickname: input(string)	checkEmail: button	checkNickname: button	submit: button	onClickEmailCheck(email: string)	onClickNicknameCheck(nickname: string)	onClickSubmit(email, password, confirmPassword, nickname)	<table><tr><td>reviewPostSearch</td></tr><tr><td>map: googleMap</td></tr><tr><td>keyword: input(string)</td></tr><tr><td>search: button</td></tr><tr><td>onClickSearch()</td></tr></table>	reviewPostSearch	map: googleMap	keyword: input(string)	search: button	onClickSearch()
main																																		
signIn: button																																		
guest: button																																		
onClickSignIn()																																		
onClickGuest()																																		
archiveReviewDetail																																		
eatWhen: DateTime																																		
tags: list(string)																																		
score: int																																		
comment: string																																		
photo: image																																		
delete: button																																		
edit: button																																		
onClickDelete()																																		
onClickEdit()																																		
signUp																																		
email: input(string)																																		
password: input(string)																																		
confirmPassword: input(string)																																		
nickname: input(string)																																		
checkEmail: button																																		
checkNickname: button																																		
submit: button																																		
onClickEmailCheck(email: string)																																		
onClickNicknameCheck(nickname: string)																																		
onClickSubmit(email, password, confirmPassword, nickname)																																		
reviewPostSearch																																		
map: googleMap																																		
keyword: input(string)																																		
search: button																																		
onClickSearch()																																		
<table><tr><td>guestSearch</td></tr><tr><td>recTag: guestRecommTag</td></tr><tr><td>method: dropdown</td></tr><tr><td>keyword: input(string)</td></tr><tr><td>search: button</td></tr><tr><td>onClickTag()</td></tr><tr><td>onClickDropdown()</td></tr><tr><td>onClickSearch(method, keyword)</td></tr></table>	guestSearch	recTag: guestRecommTag	method: dropdown	keyword: input(string)	search: button	onClickTag()	onClickDropdown()	onClickSearch(method, keyword)	<table><tr><td>signIn</td></tr><tr><td>email: input(string)</td></tr><tr><td>password: input(string)</td></tr><tr><td>signUp: button</td></tr><tr><td>signIn: button</td></tr><tr><td>onClickSignUp()</td></tr><tr><td>onClickSignIn(email, password)</td></tr></table>	signIn	email: input(string)	password: input(string)	signUp: button	signIn: button	onClickSignUp()	onClickSignIn(email, password)	<table><tr><td>searchResultList</td></tr><tr><td>keyword: input(string)</td></tr><tr><td>search: button</td></tr><tr><td>results: guestSearchResultEntry</td></tr><tr><td>onClickSearch(method, keyword)</td></tr><tr><td>onClickEntry()</td></tr></table>	searchResultList	keyword: input(string)	search: button	results: guestSearchResultEntry	onClickSearch(method, keyword)	onClickEntry()	<table><tr><td>accountSettings</td></tr><tr><td>nickname: input(string)</td></tr><tr><td>checkNickname: button</td></tr><tr><td>password: input(string)</td></tr><tr><td>confirm: input(string)</td></tr><tr><td>email: input(string)</td></tr><tr><td>publicStatus: pubToggle</td></tr><tr><td>save: button</td></tr><tr><td>onTogglePub(publicStatus)</td></tr><tr><td>onClickSave(nickname, password, confirm, email, publicStatus)</td></tr></table>	accountSettings	nickname: input(string)	checkNickname: button	password: input(string)	confirm: input(string)	email: input(string)	publicStatus: pubToggle	save: button	onTogglePub(publicStatus)	onClickSave(nickname, password, confirm, email, publicStatus)
guestSearch																																		
recTag: guestRecommTag																																		
method: dropdown																																		
keyword: input(string)																																		
search: button																																		
onClickTag()																																		
onClickDropdown()																																		
onClickSearch(method, keyword)																																		
signIn																																		
email: input(string)																																		
password: input(string)																																		
signUp: button																																		
signIn: button																																		
onClickSignUp()																																		
onClickSignIn(email, password)																																		
searchResultList																																		
keyword: input(string)																																		
search: button																																		
results: guestSearchResultEntry																																		
onClickSearch(method, keyword)																																		
onClickEntry()																																		
accountSettings																																		
nickname: input(string)																																		
checkNickname: button																																		
password: input(string)																																		
confirm: input(string)																																		
email: input(string)																																		
publicStatus: pubToggle																																		
save: button																																		
onTogglePub(publicStatus)																																		
onClickSave(nickname, password, confirm, email, publicStatus)																																		
<table><tr><td>archiveReviewList</td></tr><tr><td>review(s): archiveReview(s)</td></tr><tr><td>sortingMethod: dropdown</td></tr><tr><td>onClickReview()</td></tr><tr><td>sortReviews(sortingMethod)</td></tr></table>	archiveReviewList	review(s): archiveReview(s)	sortingMethod: dropdown	onClickReview()	sortReviews(sortingMethod)	<table><tr><td>userRankings</td></tr><tr><td>ranking: userRanking</td></tr><tr><td>review: recommReview</td></tr><tr><td>onURClick()</td></tr><tr><td>onRRClick()</td></tr></table>	userRankings	ranking: userRanking	review: recommReview	onURClick()	onRRClick()																							
archiveReviewList																																		
review(s): archiveReview(s)																																		
sortingMethod: dropdown																																		
onClickReview()																																		
sortReviews(sortingMethod)																																		
userRankings																																		
ranking: userRanking																																		
review: recommReview																																		
onURClick()																																		
onRRClick()																																		
<table><tr><td>archiveReviewPost</td></tr><tr><td>post: button</td></tr><tr><td>onClickPost()</td></tr></table>	archiveReviewPost	post: button	onClickPost()	<table><tr><td>googleMapPanel</td></tr><tr><td>map: googleMap</td></tr></table>	googleMapPanel	map: googleMap	<table><tr><td>guestLog</td></tr><tr><td>log: input(string)</td></tr><tr><td>save: button</td></tr><tr><td>guestlog: guestLogComment</td></tr><tr><td>onClickSave(log)</td></tr></table>	guestLog	log: input(string)	save: button	guestlog: guestLogComment	onClickSave(log)																						
archiveReviewPost																																		
post: button																																		
onClickPost()																																		
googleMapPanel																																		
map: googleMap																																		
guestLog																																		
log: input(string)																																		
save: button																																		
guestlog: guestLogComment																																		
onClickSave(log)																																		
<table><tr><td>reviewPostDetail</td></tr><tr><td>date: input(DateTime)</td></tr><tr><td>tag: input(string)</td></tr><tr><td>score: input(int)</td></tr><tr><td>comment: input(string)</td></tr><tr><td>photo: input(image)</td></tr><tr><td>publicStatus: Boolean</td></tr><tr><td>save: button</td></tr><tr><td>onTogglePub()</td></tr><tr><td>onClickSave(date, tag, score, comment, photo, publicStatus)</td></tr></table>	reviewPostDetail	date: input(DateTime)	tag: input(string)	score: input(int)	comment: input(string)	photo: input(image)	publicStatus: Boolean	save: button	onTogglePub()	onClickSave(date, tag, score, comment, photo, publicStatus)	<table><tr><td>archiveGuestLog</td></tr><tr><td>guestLog: button</td></tr><tr><td>onClickGuestLog()</td></tr></table>	archiveGuestLog	guestLog: button	onClickGuestLog()	<table><tr><td>sideBar</td></tr><tr><td>myArchive: button</td></tr><tr><td>account: button</td></tr><tr><td>logout: button</td></tr><tr><td>signin: button</td></tr><tr><td>onClickMyArchive()</td></tr><tr><td>onClickAccount()</td></tr><tr><td>onClickLogout()</td></tr><tr><td>onClickSignIn()</td></tr></table>	sideBar	myArchive: button	account: button	logout: button	signin: button	onClickMyArchive()	onClickAccount()	onClickLogout()	onClickSignIn()										
reviewPostDetail																																		
date: input(DateTime)																																		
tag: input(string)																																		
score: input(int)																																		
comment: input(string)																																		
photo: input(image)																																		
publicStatus: Boolean																																		
save: button																																		
onTogglePub()																																		
onClickSave(date, tag, score, comment, photo, publicStatus)																																		
archiveGuestLog																																		
guestLog: button																																		
onClickGuestLog()																																		
sideBar																																		
myArchive: button																																		
account: button																																		
logout: button																																		
signin: button																																		
onClickMyArchive()																																		
onClickAccount()																																		
onClickLogout()																																		
onClickSignIn()																																		



Frontend Algorithm

1. Molecules

a. main

- i. "Sign in": button
 1. onClick → onClickSignIn()
- ii. "Guest": button
 1. onClick → onClickGuest()
- iii. onClickSignIn(): SignInPage로 접속한다.
- iv. onClickGuest(): GuestPage로 접속한다.

b. signIn

- i. email, password: string input fields
- ii. "Sign up": button
 1. onClick → onClickSignUp()
- iii. "Sign in": button
 1. onClick → onClickSignIn(email, password)
- iv. onClickSignUp(): SignUpPage로 접속한다.
- v. onClickSignIn(email: string, password: string): 백엔드의 SignIn API를 부른다. 로그인 성공시 로그인한 유저의 ArchivePage로 접속하고, 로그인 실패시 Email이나 비밀번호가 틀리거나, 계정이 존재하지 않는다고 사용자에게 알린다.

c. signUp

- i. email, password, confirmPassword, nickname: string input fields

- ii. “이메일 중복확인”: button
 - 1. onClick → onClickEmailCheck(email)
 - iii. “닉네임 중복확인”: button
 - 1. onClick → onClickNicknameCheck(nickname)
 - iv. “Submit”: button
 - 1. onClick → onClickSubmit(email, password, confirmPassword, nickname)
 - v. onClickEmailCheck(email: string): 중복되는 Email이 있는지 확인하는 백엔드의 Email Check API를 불러서, 중복 여부를 사용자에게 알린다.
 - vi. onClickNicknameCheck(nickname: string): 중복되는 Nickname이 있는지 확인하는 백엔드의 Nickname Check API를 불러서, 중복 여부를 사용자에게 알린다.
 - vii. onClickSubmit(email: string, password: string, confirmPassword: string, nickname: string): email이 중복되는지 백엔드의 Email Check API를 불러서 확인하고, 만약 중복된다면 사용자에게 알린다. 두 비밀번호 입력이 일치하지 않는다면 사용자에게 또한 알린다. Email이 중복되지 않고 두 비밀번호 입력이 일치한다면 입력된 정보로 새 계정을 만들고, 그 계정으로 로그인하여 그 계정의 ArchivePage로 접속한다.
- d. guestSearch
- i. 추천 태그: guestRecommTag
 - 1. onClick() → onClickTag()
 - ii. 기준: dropdown
 - 1. onClick → onClickDropdown()
 - iii. keyword: string input field
 - iv. “Search”: button
 - 1. onClick → onClickSearch()
 - v. onClickTag(): selected_tag를 선택한 상태의 URL인 ‘/search/tag/:selected_tag’로 접속한다.
 - vi. onClickDropdown(): 작성자/식당 이름/태그 세 가지 기준 중 하나를 선택할 수 있다.
 - vii. onClickSearch(): 기준을 method, 검색어를 keyword로 입력받아 ‘/search/:method/:keyword’로 접속한다.
- e. userRankings
- i. 유저 랭킹: userRanking
 - 1. onClick() → onURClick()
 - ii. 추천 리뷰: recommReview
 - 1. onClick() → onRRClick()
 - iii. onURClick(): 백엔드 API에 콜을 하여 오른쪽에 유저의 추천 리뷰가 뜨게 한다.
 - iv. onRRClick(): 해당 유저의 아카이브에 클릭한 리뷰가 선택되게 변수와 함께 ‘/리뷰작성자nickname/archive’로 접속한다.
- f. searchResultList
- i. keyword: string input field
 - ii. “Search”: button
 - 1. onClick → onClickSearch()

- iii. 검색 결과들: `guestSearchResultEntry`
 - 1. `onClick()` → `onClickEntry()`
- iv. `onClickSearch()`: 기준을 `method`, 검색어를 `keyword`로 입력받아 `'/search/:method/:keyword'`로 접속한다.
- v. `onClickEntry()`: `'entryusername/archive/:entryid'`로 접속한다.
- g. `archiveReviewList`
 - i. 백엔드에서 `archive review list` API를 호출하여 리뷰 목록을 불러온다.
 - ii. 리뷰 목록: `list(archiveReview)`
 - 1. 각 `archiveReview`에 대해 `onClick()` → `onClickReview()`
 - iii. `sortingMethod`: dropdown
 - 1. 항목: “시간”, “태그”, “지역”
 - 2. `onClick` → `sortReviews(sortingMethod)`
 - iv. `onClickReview()`: 누른 리뷰에 대해 리뷰 ID인 `reviewid`를 받아와서 `'/:nickname/archive/reviewid'`로 접속한다.
 - v. `sortReviews(sortingMethod: string)`: `sortingMethod`가 “시간”이라면 리뷰 목록을 최신 순으로 정렬하고, “태그”나 “지역”이라면 태그/지역 카테고리들을 가나다/알파벳 순으로 표시하고, 각 카테고리 안에서는 리뷰를 최신 순으로 정렬한다.
- h. `googleMapPanel`
 - i. 지도: `googleMap`
 - ii. Google Map API를 통해 pin된 식당을 지도 GUI에서 찾아볼 수 있다.
- i. `archiveReviewDetail`
 - i. 현재 선택된 `reviewid`에 해당하는 리뷰 세부 정보와 사진을 백엔드 API를 호출하여 얻고, 이 세부 정보들을 표시한다.
 - ii. “Delete”: button
 - 1. `onClick()` → `onClickDelete()`
 - iii. “Edit”: button
 - 1. `onClick()` → `onClickEdit()`
 - iv. `onClickDelete()`: 백엔드 API를 콜해서 `reviewID`의 리뷰를 삭제한다.
 - v. `onClickEdit()`: `PostPage` 페이지에 `reviewid`를 보내서 접속한다.
- j. `archiveReviewPost`
 - i. “Post”: button
 - 1. `onClick()` → `onClickPost()`
 - ii. `onClickPost`: `'/post'`로 접속한다.
- k. `archiveGuestLog`
 - i. “방명록”: button
 - 1. `onClick()` → `onClickGuestLog()`
 - ii. `onClickGuestLog()`: 현재 아카이브의 방명록 `'/:nickname/archive/guestlog'`로 접속한다.
- l. `accountSettings`
 - i. “Nickname 중복확인”: button
 - 1. `onClick()` → `onClickNicknameCheck(nickname)`
 - ii. `publicStatus`: `pubToggle`
 - 1. `onClick()` → `onTogglePub(publicStatus: Boolean)`
 - iii. “Save”: button

1. onClick() → onClickSave(nickname: string, password: string, confirmPassword: string, email: string, publicStatus: Boolean)
- iv. onTogglePub(publicStatus): !publicStatus
- v. onClickSave(nickname, password, confirmPassword, email, publicStatus): 들어온 parameter를 반영하여 User 정보를 수정한다.
- m. guestLog
 - i. log: string input field
 - ii. Save: button
 1. onClick() → onClickSave(log)
 - iii. 방명록: guestLogComment
 - iv. onClickSave(log: string): log를 input으로 백엔드에 POST, 이후 방명록에 표시한다.
- n. reviewPostSearch
 - i. “지도”: googleMap
 - ii. keyword: string input field
 - iii. “Search”: button
 1. onClick() → onClickSearch(keyword)
 - iv. onClickSearch(keyword: string) → Google API에 콜하여 지도를 keyword로 검색한다.
- o. reviewPostDetail
 - i. date, tag, score, comment: string input fields
 - ii. photo: image input field
 - iii. publicStatus: pubToggle
 1. onClick() → onTogglePub(publicStatus)
 - iv. onTogglePub(publicStatus): !publicStatus
 - v. Save: button
 1. onClick() → onClickSave(date, tag, score, comment, photo, publicStatus)
 - vi. onClickSave(date, tag, score, comment, photo, publicStatus): 주어진 parameter들을 input으로 리뷰를 작성/수정한다.
- p. sideBar
 - i. “My Archive”: button
 1. 이미 로그인 하였을 때만 표시한다.
 2. onClick() → onClickMyArchive()
 - ii. “Account”: button
 1. 이미 로그인 하였을 때만 표시한다.
 2. onClick() → onClickAccount()
 - iii. “Logout”: button
 1. 이미 로그인 하였을 때만 표시한다.
 2. onClick() → onClickLogOut()
 - iv. “Sign In”: button
 1. 로그인을 하지 않은 상태에서만 표시한다.
 2. onClick() → onClickSignIn()
 - v. onClickMyArchive(): 현재 유저의 아카이브인 ‘/selfnickname/archive/’로 접속한다.
 - vi. onClickArchive(): 현재 유저의 계정 설정 페이지 ‘/account’로 접속한다.

- vii. `onClickLogout()`: 로그아웃 후 메인 페이지 '/'로 접속한다.
- viii. `onClickSignIn()`: 로그인 페이지 '/signin'으로 접속한다.

2. Atoms

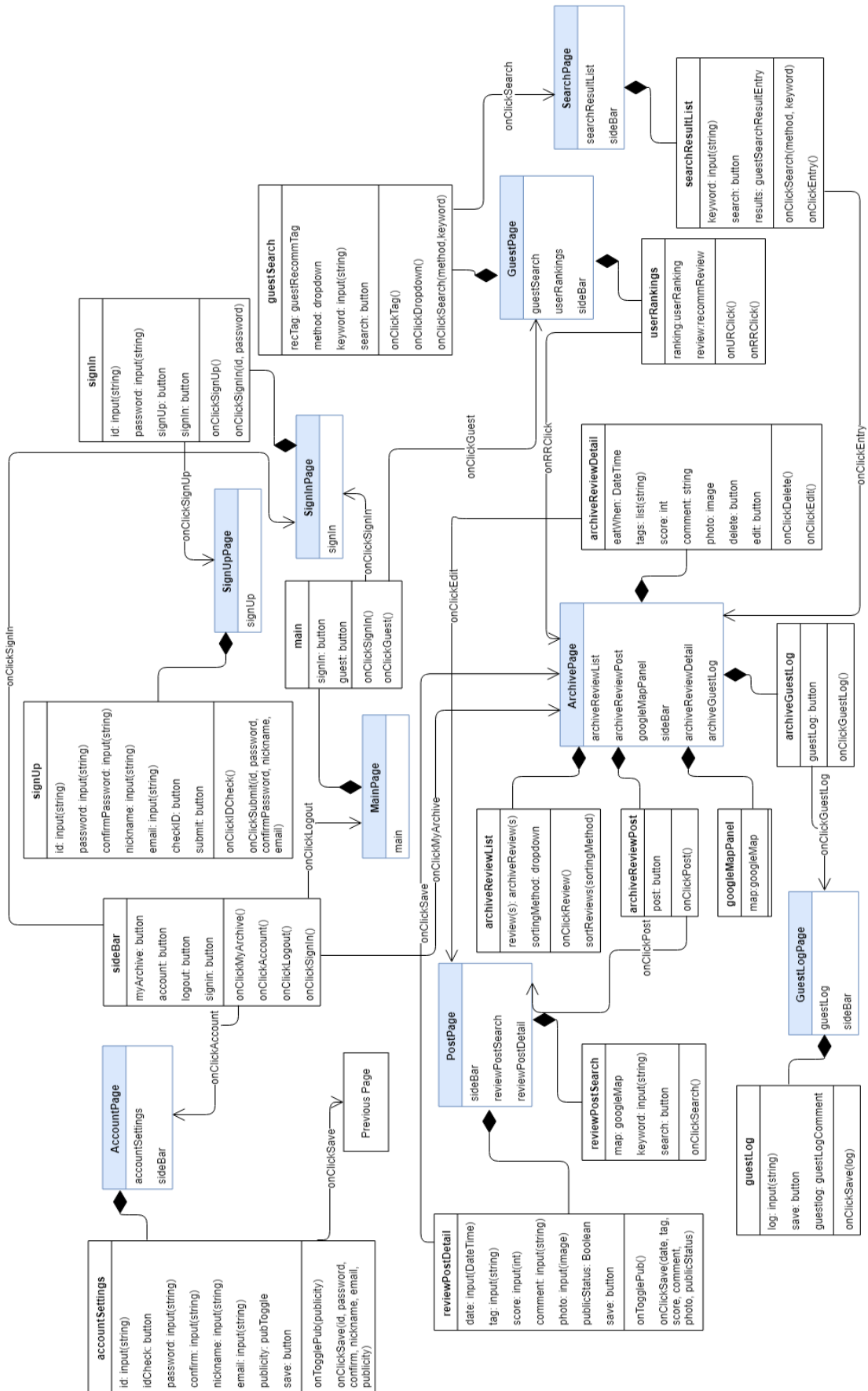
- a. `button`
 - i. page에 따라, 기능에 따라 `button`을 다르게 구현할 것이다. 예를 들어, `main molecule`에서는 `Guest` 버튼과 `Sign In` 버튼을 각각 다른 `atom`으로 구현할 것이다.
- b. `googleMap`
 - i. Google Map API를 호출하여 표시한다.
- c. `dropdown`
 - i. `react-dropdown`을 import하여 사용한다.
- d. `userRanking`
 - i. 랭킹에 해당하는 유저의 닉네임을 표시한다.
- e. `guestRecommTag`
 - i. `tagName`
- f. `guestSearchResultEntry`
 - i. `nickname`, `restName` (식당 이름), `tags`, `photo`, `content` 표시한다.
- g. `archiveReview`
 - i. `eatWhen`, `score`, `tags`, `reviewId`, `restName` (식당 이름), `publicStatus`
 - ii. 이 중에서 `eatWhen`, `restName`, `score`가 표시되지만, `publicStatus`가 `false`이고 현재 아카이브가 로그인한 계정의 것이 아니라면 표시되지 않는다.
- h. `guestLogComment`
 - i. `nickname`, `content`, `postDate`
- i. `pubToggle`
 - i. `publicStatus`
- j. `recommReview`
 - i. `eatWhen`, `score`, `reviewId`, `restName`

3. Pages

- a. `MainPage`
 - i. `main`
- b. `SignInPage`
 - i. `signIn`
- c. `SignUpPage`
 - i. `signUp`
- d. `ArchivePage`
 - i. `archiveReviewList`
 - ii. `archiveReviewPost`
 - iii. `googleMapPanel`
 - iv. `sideBar`
 - v. `archiveReviewDetail`
 - vi. `archiveGuestLog`
- e. `PostPage`
 - i. `sideBar`
 - ii. `reviewPostSearch`
 - iii. `reviewPostDetail`

- f. GuestLoginPage
 - i. guestLog
 - ii. sideBar
- g. GuestPage
 - i. guestSearch
 - ii. userRankings
 - iii. sideBar
- h. SearchPage
 - i. searchResultList
 - ii. sideBar
- i. AccountPage
 - i. accountSettings
 - ii. sideBar

Frontend Relations



Backend Design

백엔드 디자인은 MVC모델을 이용할 것이며 이는 앞에서 MVC section에서 다루었다.

RESTful API

RESTful API의 상세한 내용은 다음과 같다 :

Model	API	GET	POST	PUT	DELETE
User /api/rest-auth	/registration	X	Create new User	X	X
	/login	X	Log In	X	X
	/logout	X	Log Out	X	X
User /api/users	/	Get user list	X	X	X
	/detail/:username	Get user detail	X	Update user account Info	Delete user
	/get-nickname/:email	Get user name	X	X	X
	/exists/email/:email	Get email existence info	X	X	X
	/exists/username/:username	Get username existence info	X	X	X
Review /FooDa	/myreviewlist/:username	Get user's Review List	X	X	X
	/myreviewlist/:username/:sortOption	Get sorted Review List	X	X	X
	/reviewlist	X	Post review	X	X
	/reviewdetail/:reviewid	Get Specified Review	X	Update Review	Delete Review
	/reviewhit/:reviewid	X	X	Update Review's hits	X
Archive /FooDa	/archvie list	Get Archvie list	X	X	X

	/archive Detail /:username	Get Specified Archive	X	X	Delete Archive
Restaurant /FooDa	/restaurant list	Get Restaurant list	Create new Restaurant	X	X
	/restaurant detail	Get Specified Restaurant	X	Update Restaurant	Delete Restaurant
Tag /FooDa	/tagging	Get Tag List	X	X	X
	/taggeditem	Get Tagged Item List	X	X	X
	/tagfilter /:tag_id	Get Review list have tag_id	X	X	X
WeeklyRanki ng /FooDa	/weekly rankinglist	Get Ranking list	X	X	X
	/weekly rankingdetail /:id	Get Specified Ranking	X	Update Ranking	Delete Ranking
GuestComm ent /FooDa	/guest commentlist	Get Guest Comment list	Create New Guest Comment	X	X
	/guest comment detail	Get Specified Guest Comment	X	Update Guest Comment	Delete Guest Comment

Implementation Plan

Implementation plan을 통해 각 iteration에서 팀원들이 진행할 역할과 업무를 분배하고자 한다. 각각의 기능은 Requirement and Specification에서 작성한 user story를 page의 형태로 정리하여 나누었다. 개발 순서는 대략적인 골격을 잡아줄 수 있는 메인 페이지에서 시작하여 각 개인의 리뷰 아카이브 페이지를 구현하고, 최종적으로 이러한 아카이브들을 검색할 수 있는 검색 페이지를 구현하는 방향으로 나아갈 예정이다. 실질적으로 각각의 페이지들의 dependency를 고려해보면 다음 순서로 배열하는게 가능하다. 우선 user가 웹에 가입을 해야 서비스 이용이 가능하고, 각각의 아카이브와 리뷰 작성/수정 기능이 완성되어야 이를 열람할 수 있는 게스트 페이지와 검색 기능의 활성화가 가능하다. 그렇게 검색까지 완성되면 각 아카이브에 방명록을 남기고, 계정 정보를 변경하고, 추가적인 기능의 도입이 가능해질 것이다.

Iteration	개발할 기능(Page)
2nd iteration	메인 페이지, 로그인 페이지, 회원가입 페이지
3rd Iteration	아카이브 페이지, 리뷰 작성/수정 페이지, Sidebar
4th Iteration	게스트 페이지, 검색 결과 페이지
5th Iteration	방명록, 계정 설정, 추가 기능

각 업무의 분배는 최대한 업무 난이도를 균등하게 하는 방향으로 설정하기 위해 노력했다. 프론트엔드와 백엔드 개발을 나누어 진행하는 것이 개발 진행 속도에 도움이 될 것으로 판단하여 신수웅, 이민후, 한상현은 프론트엔드 개발에 집중하고, 한재훈이 백엔드 개발에 집중하는 것으로 계획을 세웠다. HTML과 CSS를 이용한 전체적인 웹디자인은 팀원 중 아무도 전문적으로 배운 경험이 없어 동아리에 속해있으며 한 번이라도 간단히 배워본 경험이 있는 한상현이 주로 맡고, 다른 팀원이 보조해주는 방향으로 결정했다.

다음 표는 각 업무에 할당된 인원과 그 난이도, 주요 challenge를 보여준다. 난이도는 어려움의 정도에 따라 1~5로 책정했다. 각 작업에 별도의 예상 시간을 기재하지는 않았으며 앞선 표에 기재했듯 그 작업이 어떤 iteration에서 수행되어야 할지를 명시했다.

Page	Feature (front / backend)	Difficulty (1-5)	Iteration	Person	Challenge
Main page	Signin page 나 Guest page로 이동	1	2	한상현	
Signin page	email 과 password를 입력하는 빈칸과 sign in 버튼 구현 / Sign in 기능 구현	2	2	이민후, 한재훈	User data의 저장 방식
Signup page	회원가입 페이지 + 그 계정으로 로그인 / 계정 생성 기능	3	2	신수웅, 한재훈	User data의 저장 방식
Archive page	아카이브 페이지 전반적인 디자인		3	한상현	Design
Archive page	정렬된 Review List, Detail 화면 표시 기능 구현	2	3	신수웅, 이민후	

Archive page	정렬된 Review List, Review Detail을 Get하는 기능	4	3	한재훈	정렬 알고리즘
Archive page	Google Map API를 이용한 pin 기능 / pin 정보 저장	5	3	한상현, (+알파), 한재훈	pin을 유의미하게 구분할 수 있게 api 사용하는 방법
Sidebar	Guest와 User의 Sidebar 구현(sign in / logout / my archive / my account)	2	3	신수웅	로그인한 user와 하지 않은 유저 사이의 sidebar 차이 구현
Post page	Google Map API를 이용한 식당 검색기능 / 검색된 식당 저장	5	3	한상현, 한재훈	식당만을 지도 상에 표시하고, 검색할 수 있는 기능 구현
Post page	Review Post를 위한 입력 창 구현 / Review Post 기능 구현	3	3	이민후, 한재훈	
Guest page	인기 태그 6개 화면 상단 노출 / 태그 사용 빈도수 계산 기능	4	4	신수웅, 한재훈	Tag 사용 횟수에 따른 sorting

Guest page	검색 기능 구현	2	4	한상현, 한재훈	
Guest page	User ranking 노출/ Archive 방문자수 기능 구현	4	4	이민후, 신수웅, 한재훈	조회수 기반 user sorting
Search page	검색 키워드에 맞는 review들 list의 형태로 노출 / 태그 검색 기능 구현	4	4	이민후, 신수웅, 한재훈	키워드 종류에 따른 검색 결과 제공(기준 확립)
Search page	클릭 시 해당하는 archive의 review page로 이동	2	4	한상현	
Guestlog page	방명록 기능 구현(login 한 user만 이용가능) / 방명록 Post, Get, Put기능 구현	4	5	이민후, 한상현, 한재훈	백엔드 데이터 저장 형태
Account page	Sign up page와 유사하게 계정정보 수정 기능 구현 (+ public, local 여부 결정) / User 정보 update 기능	2	5	신수웅, 한재훈	

추가 기능	미정	?	5	미정	
-------	----	---	---	----	--

Testing Plan

Unit Testing

웹 서비스에서 구동되는 모든 모듈은 test가 필요할 것이다. Code coverage는 80% 이상이 되도록 하는게 최소한의 목표이며 서비스가 완성되는 시점에는 90%가 넘을 수 있도록 노력할 예정이다. 다음 framework들을 이용해 test를 진행할 예정이다.

Frontend(React+Redux+Redux-saga) : Jest

Backend(Django) : Python unit test

Functional Testing

구현한 모든 API가 test의 대상이 될 것이다. 각 Iteration에서 어떤 부분을 Test할지는 구현 상황에 맞추어 결정할 계획이며 사용할 Tool은 다음과 같다.

Frontend(React+Redux+Redux-saga) : Jest

Backend(Django) : Python unit test

Acceptance Testing

작성 시점(19.04.21)에 Acceptance Test를 위한 Tool에 대한 정보가 부족하여 검색을 통해 찾은 Selenium을 Acceptance Test Tool로 결정했다. Integration Test의 경우 Travis CI를 이용할 계획이다.

Acceptance Test : Selenium

Integration Test : Travis CI