

# High, Travel!

## Design and Planning

Rev. 1.3 2018-06-11 - fourth version

### Revision Bullet List

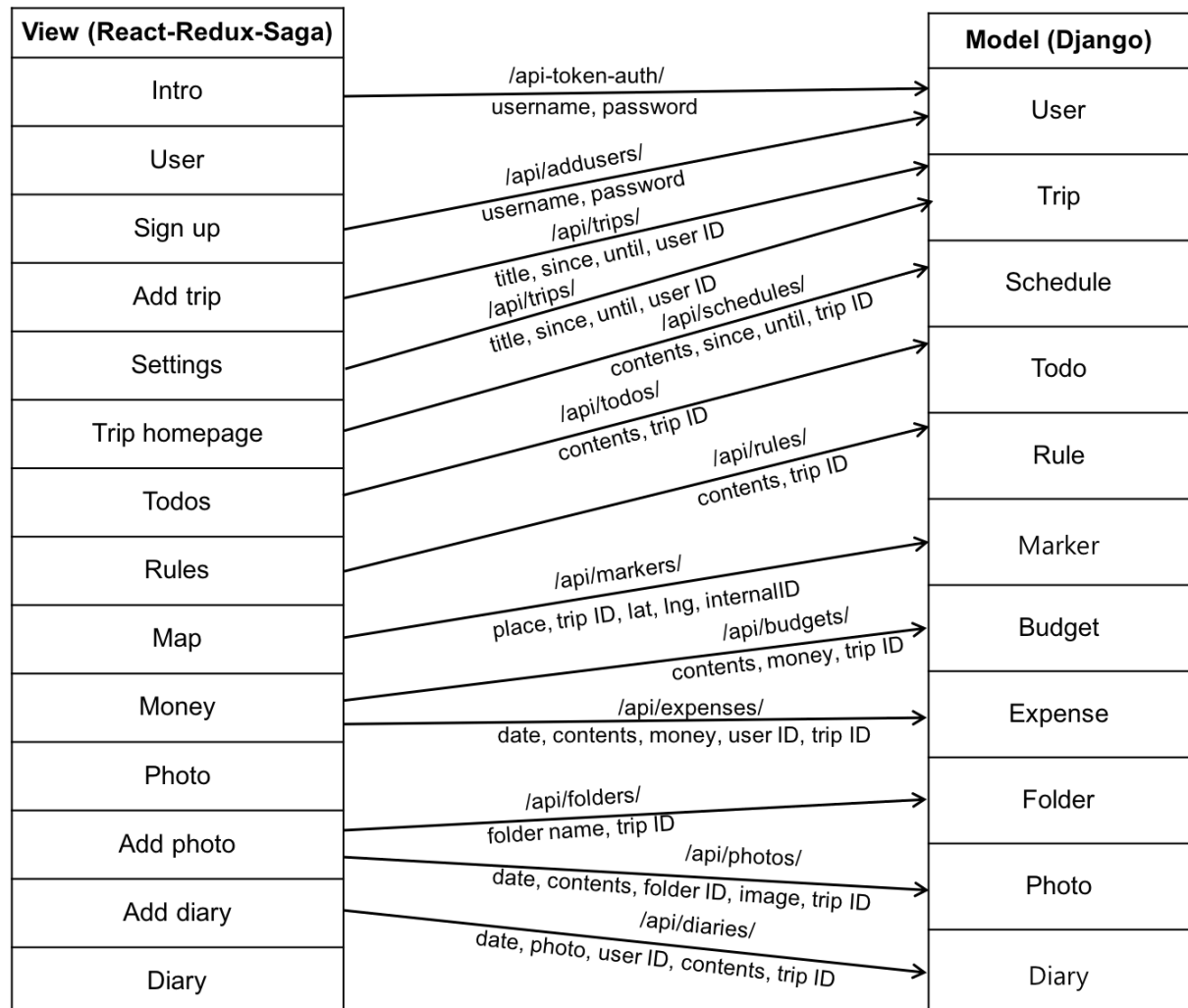
- added testing plan
- RESTful api : login, adduser backend view updated,  
folder model backend view added, date of photo backend view added
- frontend components : todos, settings, headers component updated
- frontend algorithms : add trip form, sidebar component, add diary, add photo updated
- frontend relations : logout button, todo component, add diary, add photo updated
- ER diagram and relations : folder model added
- user interface : add trip form, add diary tab, add photo tab, todos tab updated
- MVC : add diary view, add photo view, folder model, controller updated
- implementation plan : frontend antd design added, add photo implementation person changed,  
settings menu added

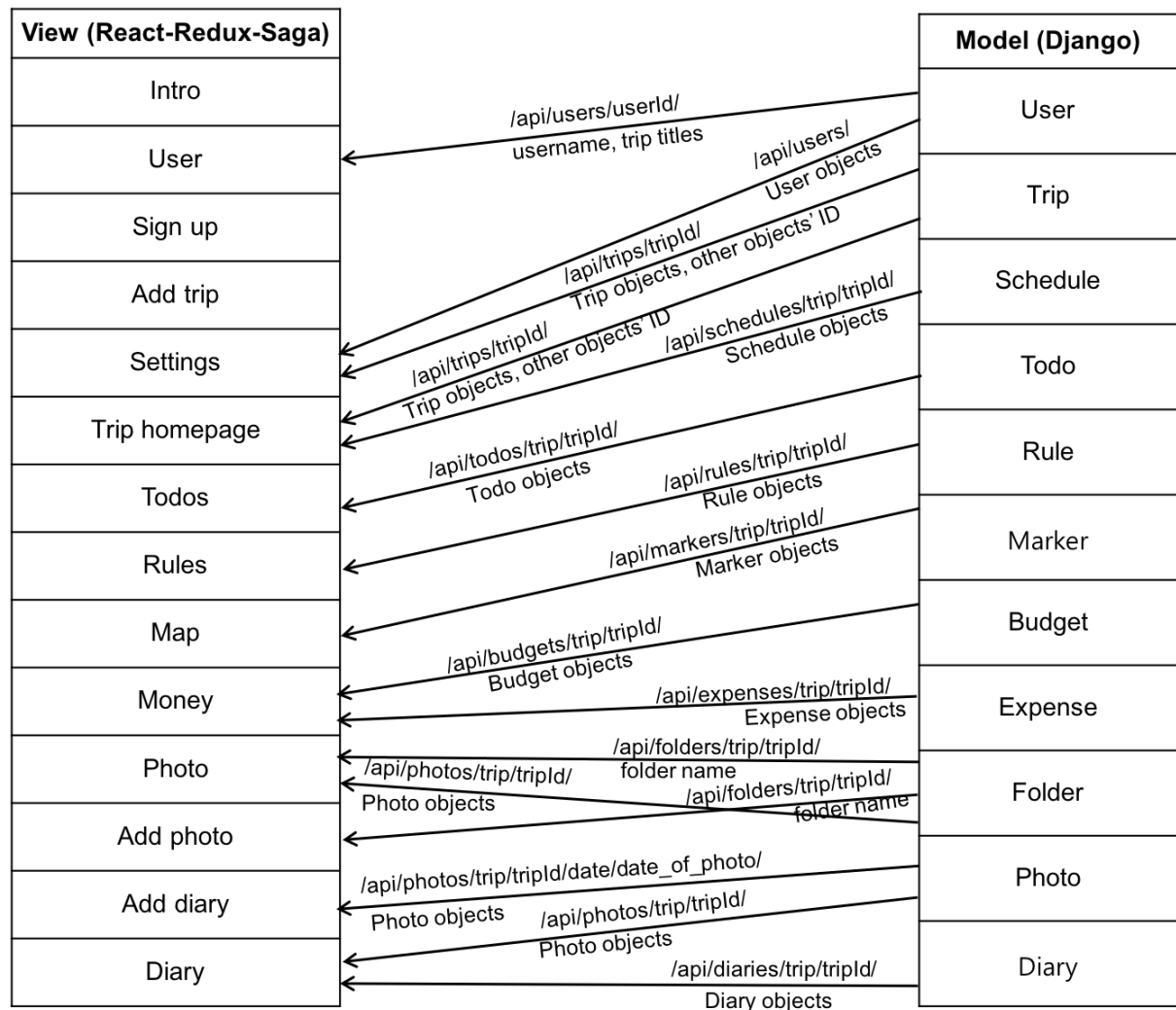
### Members

9 조 양지훈, 박규연, 김지우, 최현지

# System Architecture

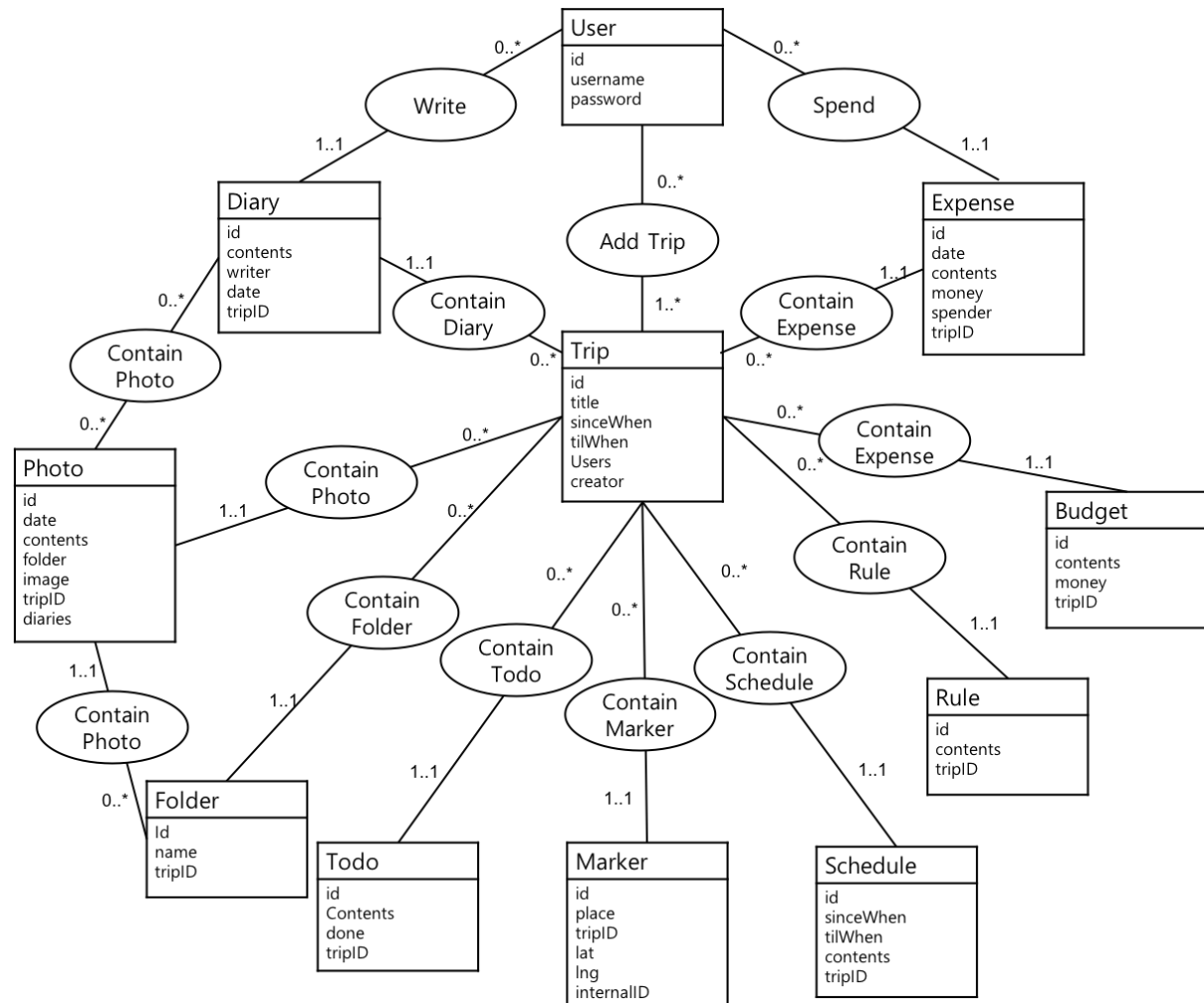
## MVC





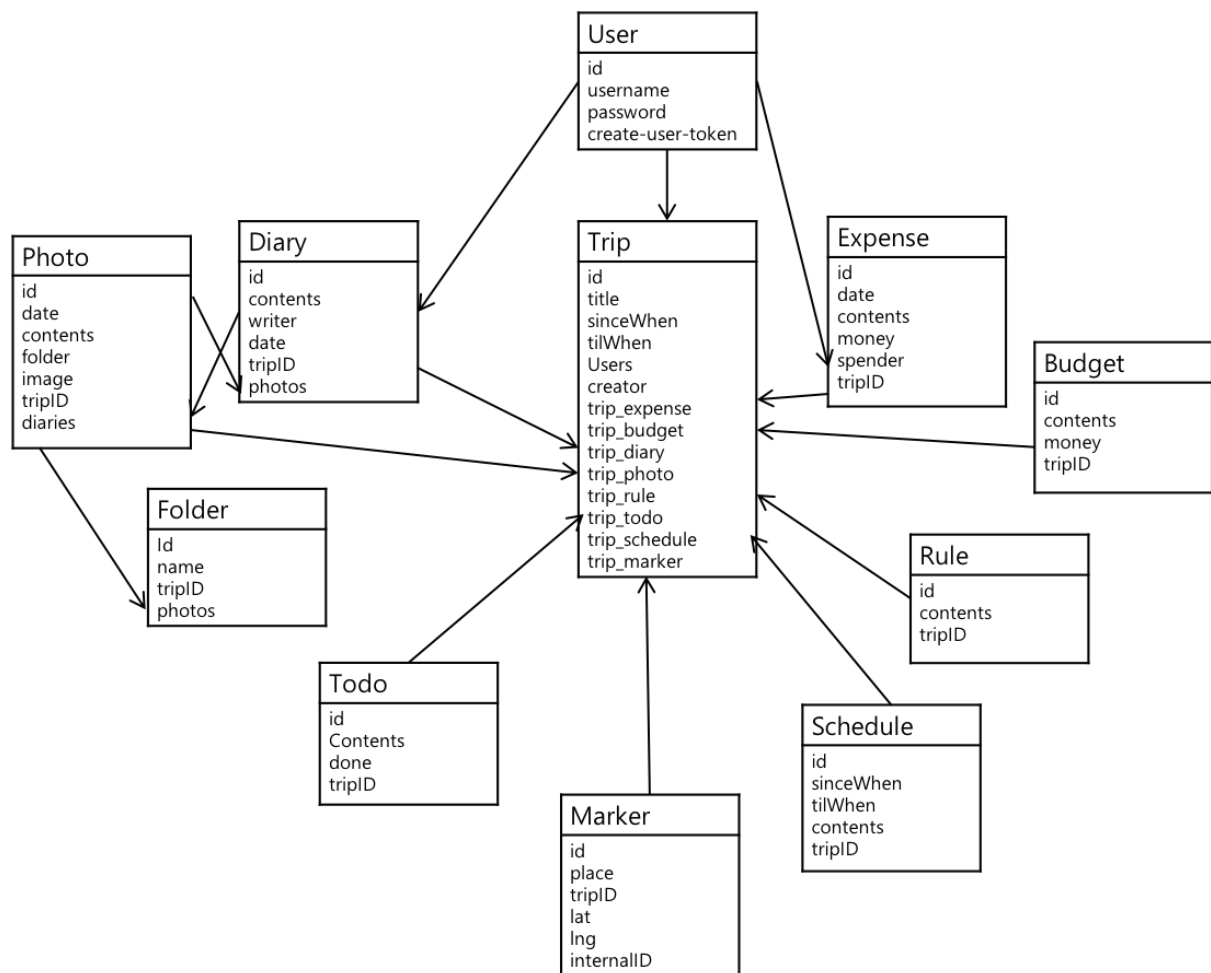
There are 14 views (left) and 11 models (right). For readability, budget tab and expense tab were integrated to money views. Arrows show controllers, left-to-right showing requests with user inputs from the view, and right-to-left showing responses from the models with data. Above the arrows, the backend APIs that frontend views will use to communicate with models are shown.

Here is an Entity-Relationship diagram for model design.



Each rectangle represents an entity with attributes listed inside. Each oval represents the relationship between entities, numbers next to the line showing cardinality constraints (min..max). In this relationship, cardinality constraints combination of 1..1 and 0..\* represents one to many relationship. On the other hand, 0..\* and 0..\* combination(photo and diary relationship), 0..\* and 1..\* combination(user and trip relationship) represents many to many relationship.

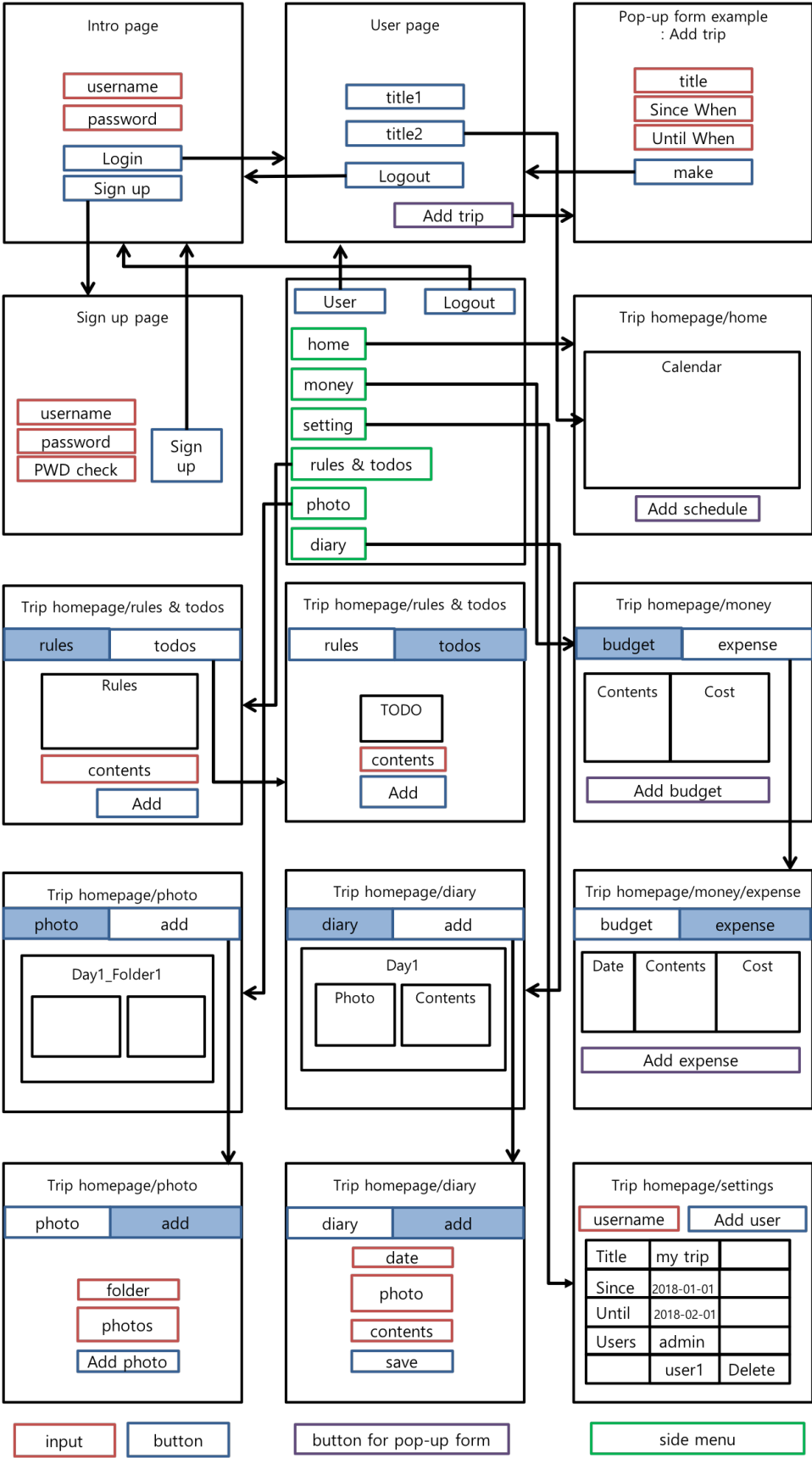
Relation schema diagram based on this E-R diagram is as below.



Rectangle stands for relation schema. Schema attributes are listed inside relation rectangle. In User model case, model has create-user-token method inside the rectangle. Trip model has expense, budget, diary, photo, rule, todo, schedule, marker model as foreign key, and user model as many-to-many field. Expense and diary model has user model as foreign key. Photo and diary models are in many-to-many relation. Finally, folder model has photo model as foreign key.

# View Details

Here is an user interface for view design.



Details of these view pages are as below.

1. Sign up page (/signup)
  - Sign up new user
  - Get username and password as inputs
2. Intro page (/intro)
  - Sign in
  - Get username and password as inputs
  - Go to sign up page
3. User page (/user)
  - Show user's trip titles
  - Log out
  - Go to trip homepage
  - create new trip by getting title, since when, until when as inputs
4. Trip homepage (/)
  - ① home menu
    - Show calendar
    - Get since, until, contents as inputs and add to calendar
    - Show schedule list
    - Delete schedule
  - ② rules and todos menu
    - Rules tab(default) show rules
    - Rules tab get contents as input and add to rules list
    - Rules tab allow rule deletion
    - Todos tab show todolist
    - Todos tab get contents as input and add to todolist
    - Todos tab allow to mark done
  - ③ money menu
    - Budget tab(default) gets contents and cost as inputs and add to budget
    - Budget tab shows total amount of budget
    - Budget tab allow modification and deletion of budget
    - Expense tab gets date, contents and cost as inputs and add to user's expense
    - Expense tab shows total amount of expense for each users

- Expense tab allow modification and deletion of expense

④ map menu

- Show Google map API
- Right click map and add markers
- Add place name to markers
- Delete markers
- Modify ID of markers

⑤ photo menu

- Photo tab(default) show photos under folders
- Photo tab allow deletion of folder and photo
- Add tab get date and folder name as inputs and create new folder
- Add tab get folder and photos as inputs and post photos

⑥ diary menu

- Diary tab(default) show diaries
- Diary tab allow deletion of diary
- Diary tab allow modification of diary contents
- Add tab get date as input and show photos that are available
- Add tab get date, contents, photos(optional) as inputs and create diary

⑦ settings menu

- Show trip information
- Show user list
- Get username as input and add user to trip
- Allow deletion of user only for admin user



# Design Details

## Frontend Design

### Frontend Components

<table><tr><th>intro</th></tr><tr><td>username : input password : input login : button sign_up : button</td></tr><tr><td>+onClickLoginButton +onClickSignUpButton</td></tr></table>	intro	username : input password : input login : button sign_up : button	+onClickLoginButton +onClickSignUpButton	<table><tr><th>rules</th></tr><tr><td>rules : tab button todos : tab button rules : text contents_rule : input add_rule : button delete_rule : button</td></tr><tr><td>+onClickAddRuleButton +onClickTodosTab +onClickDeleteRuleButton</td></tr></table>	rules	rules : tab button todos : tab button rules : text contents_rule : input add_rule : button delete_rule : button	+onClickAddRuleButton +onClickTodosTab +onClickDeleteRuleButton	<table><tr><th>trip_home</th></tr><tr><td>calendar : calendar contents_schedule : input since : input until : input add_schedule : button sinceWhen : text untilWhen : text contents_schedule_list : text delete_schedule : button</td></tr><tr><td>+onClickAddScheduleButton +onClickDeleteScheduleButton</td></tr></table>	trip_home	calendar : calendar contents_schedule : input since : input until : input add_schedule : button sinceWhen : text untilWhen : text contents_schedule_list : text delete_schedule : button	+onClickAddScheduleButton +onClickDeleteScheduleButton
intro											
username : input password : input login : button sign_up : button											
+onClickLoginButton +onClickSignUpButton											
rules											
rules : tab button todos : tab button rules : text contents_rule : input add_rule : button delete_rule : button											
+onClickAddRuleButton +onClickTodosTab +onClickDeleteRuleButton											
trip_home											
calendar : calendar contents_schedule : input since : input until : input add_schedule : button sinceWhen : text untilWhen : text contents_schedule_list : text delete_schedule : button											
+onClickAddScheduleButton +onClickDeleteScheduleButton											
<table><tr><th>user</th></tr><tr><td>title : button add_trip : button logout : button</td></tr><tr><td>+onClickTitleButton +onClickAddTripButton +onClickLogOutButton</td></tr></table>	user	title : button add_trip : button logout : button	+onClickTitleButton +onClickAddTripButton +onClickLogOutButton	<table><tr><th>Photo</th></tr><tr><td>day_date : text folder_name : text photo : image add_photo : tab button</td></tr><tr><td>+onClickAddPhotoTab</td></tr></table>	Photo	day_date : text folder_name : text photo : image add_photo : tab button	+onClickAddPhotoTab	<table><tr><th>money_budget</th></tr><tr><td>budget : tab button expense : tab button contents : text cost : text contents_budget : input cost_input : input add_budget : button delete_budget : button</td></tr><tr><td>+onClickAddBudgetButton +onClickExpenseTab +onClickDeleteBudgetButton</td></tr></table>	money_budget	budget : tab button expense : tab button contents : text cost : text contents_budget : input cost_input : input add_budget : button delete_budget : button	+onClickAddBudgetButton +onClickExpenseTab +onClickDeleteBudgetButton
user											
title : button add_trip : button logout : button											
+onClickTitleButton +onClickAddTripButton +onClickLogOutButton											
Photo											
day_date : text folder_name : text photo : image add_photo : tab button											
+onClickAddPhotoTab											
money_budget											
budget : tab button expense : tab button contents : text cost : text contents_budget : input cost_input : input add_budget : button delete_budget : button											
+onClickAddBudgetButton +onClickExpenseTab +onClickDeleteBudgetButton											
<table><tr><th>add_trip</th></tr><tr><td>title : input sinceWhen : input untilWhen : input make : button</td></tr><tr><td>+onClickMakeButton</td></tr></table>	add_trip	title : input sinceWhen : input untilWhen : input make : button	+onClickMakeButton	<table><tr><th>photo_add</th></tr><tr><td>date : input folder : input photos : image add_folder : button add_photo : button photo : tab button</td></tr><tr><td>+onClickAddFolderButton +onClickAddPhotoButton +onClickPhotoTab</td></tr></table>	photo_add	date : input folder : input photos : image add_folder : button add_photo : button photo : tab button	+onClickAddFolderButton +onClickAddPhotoButton +onClickPhotoTab	<table><tr><th>money_expense</th></tr><tr><td>budget : tab button expense : tab button date : text contents : text cost : text contents_expense : input cost_input : input date_input : input add_expense : button delete_expense : button</td></tr><tr><td>+onClickAddExpenseButton +onClickBudgetTab +onClickDeleteExpenseButton</td></tr></table>	money_expense	budget : tab button expense : tab button date : text contents : text cost : text contents_expense : input cost_input : input date_input : input add_expense : button delete_expense : button	+onClickAddExpenseButton +onClickBudgetTab +onClickDeleteExpenseButton
add_trip											
title : input sinceWhen : input untilWhen : input make : button											
+onClickMakeButton											
photo_add											
date : input folder : input photos : image add_folder : button add_photo : button photo : tab button											
+onClickAddFolderButton +onClickAddPhotoButton +onClickPhotoTab											
money_expense											
budget : tab button expense : tab button date : text contents : text cost : text contents_expense : input cost_input : input date_input : input add_expense : button delete_expense : button											
+onClickAddExpenseButton +onClickBudgetTab +onClickDeleteExpenseButton											
<table><tr><th>sign_up</th></tr><tr><td>username : input password : input pwd_check : input sign_up : button</td></tr><tr><td>+onClickSignupButton</td></tr></table>	sign_up	username : input password : input pwd_check : input sign_up : button	+onClickSignupButton	<table><tr><th>diary_add</th></tr><tr><td>date : input photo : image contents : input save : button diary : tab button</td></tr><tr><td>+onClickSaveButton +onClickDiaryTab</td></tr></table>	diary_add	date : input photo : image contents : input save : button diary : tab button	+onClickSaveButton +onClickDiaryTab	<table><tr><th>todos</th></tr><tr><td>rules : tab button todos : tab button todo : text contents_todo : input add_todo : button</td></tr><tr><td>+onClickAddTodoButton +onClickRulesTab</td></tr></table>	todos	rules : tab button todos : tab button todo : text contents_todo : input add_todo : button	+onClickAddTodoButton +onClickRulesTab
sign_up											
username : input password : input pwd_check : input sign_up : button											
+onClickSignupButton											
diary_add											
date : input photo : image contents : input save : button diary : tab button											
+onClickSaveButton +onClickDiaryTab											
todos											
rules : tab button todos : tab button todo : text contents_todo : input add_todo : button											
+onClickAddTodoButton +onClickRulesTab											
<table><tr><th>diary</th></tr><tr><td>day_date : text photo : image contents : text add_diary : tab button</td></tr><tr><td>+onClickAddDiaryTab</td></tr></table>	diary	day_date : text photo : image contents : text add_diary : tab button	+onClickAddDiaryTab	<table><tr><th>Header</th></tr><tr><td>user : button logout : button trip_title : text</td></tr><tr><td>+onClickUserButton +onClickLogoutButton</td></tr></table>	Header	user : button logout : button trip_title : text	+onClickUserButton +onClickLogoutButton				
diary											
day_date : text photo : image contents : text add_diary : tab button											
+onClickAddDiaryTab											
Header											
user : button logout : button trip_title : text											
+onClickUserButton +onClickLogoutButton											
<table><tr><th>sidebar</th></tr><tr><td>home : button money : button rules_and_todos : button photo : button diary : button settings : button</td></tr><tr><td>+onClickHomeButton +onClickMoneyButton +onClickRulesandTodosButton +onClickPhotoButton +onClickDiaryButton +onClickSettingsButton</td></tr></table>	sidebar	home : button money : button rules_and_todos : button photo : button diary : button settings : button	+onClickHomeButton +onClickMoneyButton +onClickRulesandTodosButton +onClickPhotoButton +onClickDiaryButton +onClickSettingsButton	<table><tr><th>Settings</th></tr><tr><td>username : input add_user : button</td></tr><tr><td>+onClickAddUserButton</td></tr></table>	Settings	username : input add_user : button	+onClickAddUserButton				
sidebar											
home : button money : button rules_and_todos : button photo : button diary : button settings : button											
+onClickHomeButton +onClickMoneyButton +onClickRulesandTodosButton +onClickPhotoButton +onClickDiaryButton +onClickSettingsButton											
Settings											
username : input add_user : button											
+onClickAddUserButton											

## Frontend Algorithms

### 1. intro

Check the user is logged in by calling backend api, and redirect to intro page if user is not signed

-onClickLoginButton(username: string, password: string) : Call backend Signin api. If sign-in is accepted, redirect to User page . If sign-in is rejected, alert the user.

-onClickSignUpButton() : redirect to signup page

### 2. Signup

-onClickSignUpButton(username: string, password: string, pwd\_check: string) : Make a new user. If username is already used, alert the user to use other username. If password is different with pwd\_check, alert the user to check password to be same with pwd\_check. If sign up is successful, redirect to the intro page

### 3. User

-onClickTitleButton(): Redirect to specific trip page.

-onAddTripButton(): Show add trip pop-up form.

-onClickLogOutButton(): Logout and redirect to intro page.

### 4. Add trip

-onClickMakeButton(title: String, sinceWhen: Date, untilWhen: Date): Make a new trip. If any input is blank, alert to fill it

### 5. Sidebars

-onClickHomeButton(): Show schedule component.

-onClickMoneyButton(): Show budget component.

-onClickRuleButton(): Show rules component.

-onClickPhotoButton(): Show photo component.

-onClickDiaryButton(): Show diary component.

-onClickSettingsButton(): Show settings component.

## 6. Trip homepage

-onClickAddScheduleButton(contents\_schedule: String, since: Date, until:Date): Add a schedule to calendar.  
If any input is blank, alert to fill it.

## 7. Todos

-onClickAddTodoButton(contents\_todo: string): Add a what to do to todo list. If any input is blank, alert to fill it.

-onClickRulesTab() : Switch the current page to Rules page.

## 8. Rules

-onClickAddRuleButton(contents\_rule: string): Add a rule to rule list „If input is blank, alert to fill it.

-onClickDeleteRuleButton(contents\_rule: string) : Delete a rule in rule list.

-onClickTodosTab() : Redirect to todos page.

## 9. money\_budget

-onClickAddBudgetButton(contents\_budget: string, cost\_input: int): Add contents of budget plan and cost to budget list. If any input is blank, alert to fill in.

-onClickExpenseTab() : Switch the current page to expense management tab.

## 10. photo

-onClickAddPhotoTab(): Switch current page to add photo tab.

## 11. diary

-onClickAddDiaryTab(): Switch current page to add diary tab.

## 12. money\_expense

-onClickAddExpenseButton(contents\_expense: string, cost: int, date: date): Add contents of expense details and cost to expense list. If any input is blank, alert to fill in.

-onClickBudgetTab(): Switch the current page to budget management menu.

### 13. photo\_add

-onClickAddFolderButton(date: date, name: string): Add new folder to photo list in a specific day. If any input is blank, alert to fill in.

-onClickAddPhotoButton(date: date, folder: string, photo: image): Add photos to photo list in a corresponding date and folder where user want to add them. If any input is blank, alert to fill in. We decided photo is added in image format by now, but it is alterable.

-onClickPhotoTab: Switch current page to photo tab.

### 14. diary\_add

-onClickSaveButton(date: date, photo: image, contents\_diary: string): Add a diary to diary list in a corresponding date. User can choose a photo that they want to add.

-onClickDiaryTab: Switch current page to diary tab.

### 15. Settings

-onClickAddUserButton(username: string): Invite user corresponding username to specific trip. Send a invitation message to the user. if user accept the invitation, user can access the trip page.

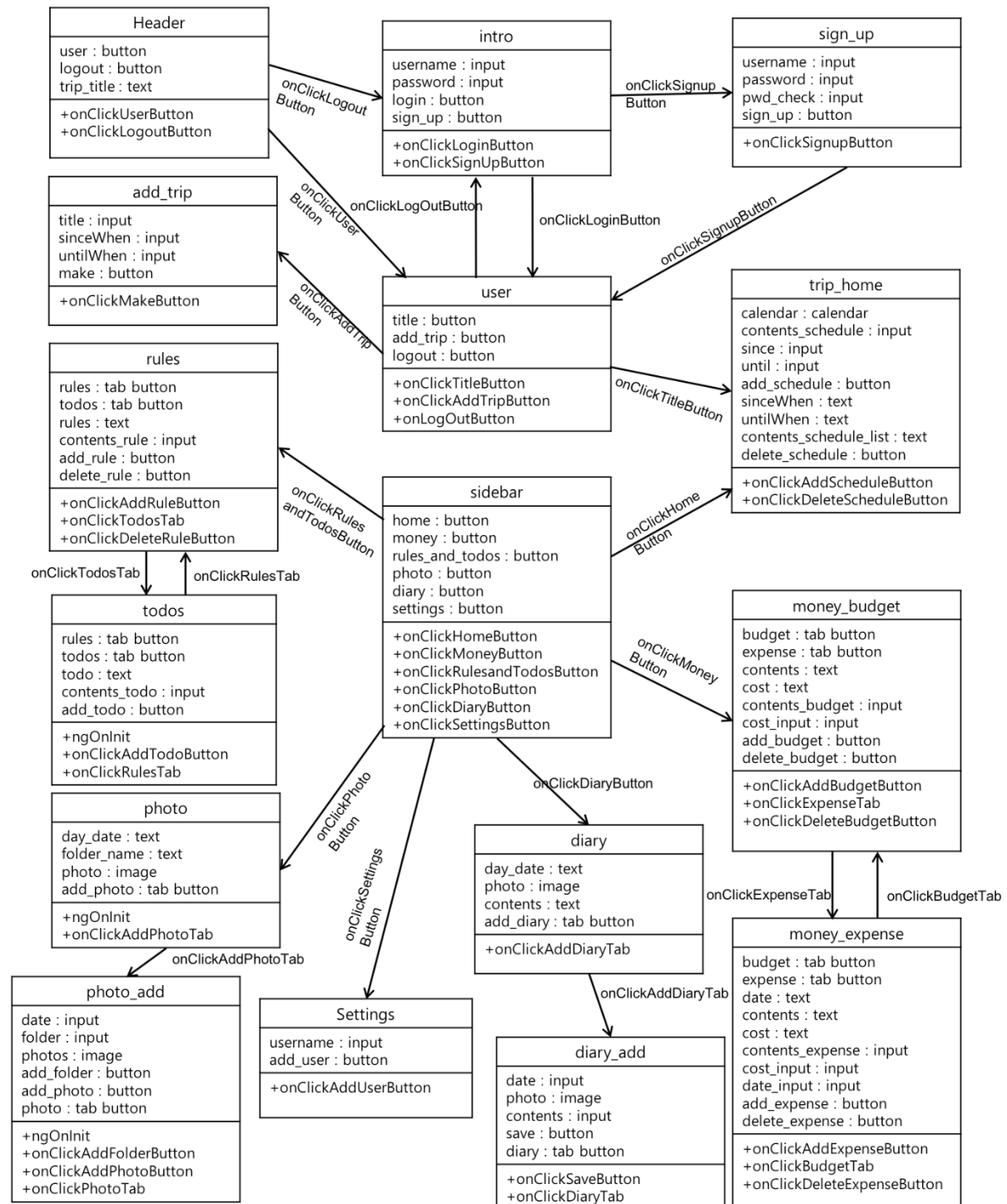
### 16.Headers

-onClickLogoutButton() : Logout from current user account. Redirect to Intro page.

-onClickUserButton() : Redirect to user page.

## Frontend Relations

There are the relations between components.



## Backend Design

### RESTful API

Model	API	GET	POST	PATCH	DELETE
User	/api/users/id/	Get specific user			
	/api/addusers/		Create new user		
	/api-token-auth/		Login		
Trip	/api/trips/	Get trip list	Create new trip		
	/api/trips/id/	Get specific trip		Modify trip	Delete trip
Expense	/api/expenses/	Get expense list	Create new expense		
	/api/expenses/id/	Get specific expense		Modify expense	Delete expense
	/api/expenses/trip/tripId/	Get list of expenses for specific trip			
Budget	/api/budgets/	Get budget list	Create new budget		
	/api/budgets/id/	Get specific budget		Modify budget	Delete budget
	/api/budgets/trip/tripId/	Get list of budgets for specific trip			

Folder	/api/folders/	Get folder list	Create new folder		
	/api/folders/id/	Get specific folder		Modify folder	Delete folder
	/api/folders/trip/tripId/	Get list of folders for specific trip			
Photo	/api/photos/	Get photo list	Create new photo		
	/api/photos/id/	Get specific photo			Delete photo
	/api/photos/trip/tripId/	Get list of photos for specific trip			
	/api/photos/trip/tripId/date/date_of_photo	Get list of photos for specific trip & date			
Diary	/api/diaries/	Get diary list	Create new diary		
	/api/diaries/id/	Get specific diary		Modify diary	Delete diary
	/api/diaries/trip/tripId/user/userId/	Get list of diaries for specific trip & user			
Marker	api/markers/	Get marker list	Create new marker		
	/api/markers/id/	Get specific marker		Modify marker	Delete marker

	/api/markers/ trip/tripId/	Get list of markers for specific trip			
Rule	/api/rules/	Get rule list	Create new rule		
	/api/rules/id/	Get specific rule			Delete rule
	/api/rules/ trip/tripId/	Get list of rules for specific trip			
Schedule	/api/schedules/	Get schedule list	Create new schedule		
	/api/schedules/id/	Get specific schedule			Delete schedule
	/api/schedules/ trip/tripId/	Get list of schedules for specific trip			
Todo	/api/todos/	Get todo list	Create new todo		
	/api/todos/id/	Get specific todo		Modify todo	Delete todo
	/api/todos/ trip/tripId/	Get list of todos for specific trip			



## Implementation Plan

Our implementation plan provides the division of work into sprints and people. There are four sprints in our development process. Features cover from backend structure to frontend design. Estimated difficulties are shown in scale 1-5.

Sprints	Feature	Difficulties	Time(min)	Person
1	backend structure	1	120	최현지
1	backend views	1	90	김지우
2	backend side sign in, sign up, sign out, token auth	2	150	김지우 최현지
2	frontend intro page	1	90	박규연
2	frontend signup page	1	90	양지훈
2	frontend user page	2	120	박규연
2	frontend add trip page	2	150	양지훈
2	frontend rule page	1	90	김지우
2	frontend routing	3	200	최현지
2	saga	2	120	김지우
3	frontend photo page	3	240	양지훈 박규연
3	frontend antd design	3	320	최현지
3	frontend money page with two tabs	3	320	김지우 최현지
3	frontend add photo page	4	400	김지우 최현지
4	frontend diary page	3	240	박규연 양지훈

4	frontend add diary page	3	240	김지우 최현지
4	frontend settings page	3	240	김지우
4	frontend map page (optional)	5		

## Testing Plan

### Backend testing

- For iteration2, shell scripts and python requests - referenced TA's code
- For iteration3,4, Django test

### Frontend testing

- For iteration 2,3,4, jest

### Continuous Integration testing

- For iteration 3,4, travis CI tool