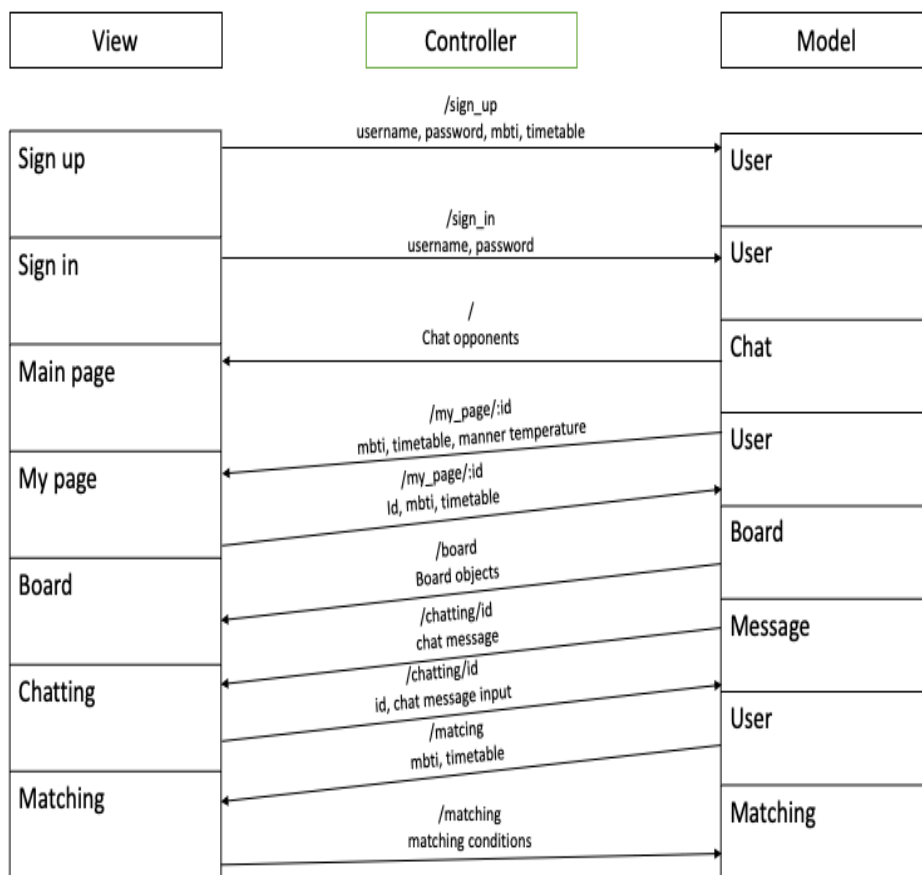


# Design and Planning

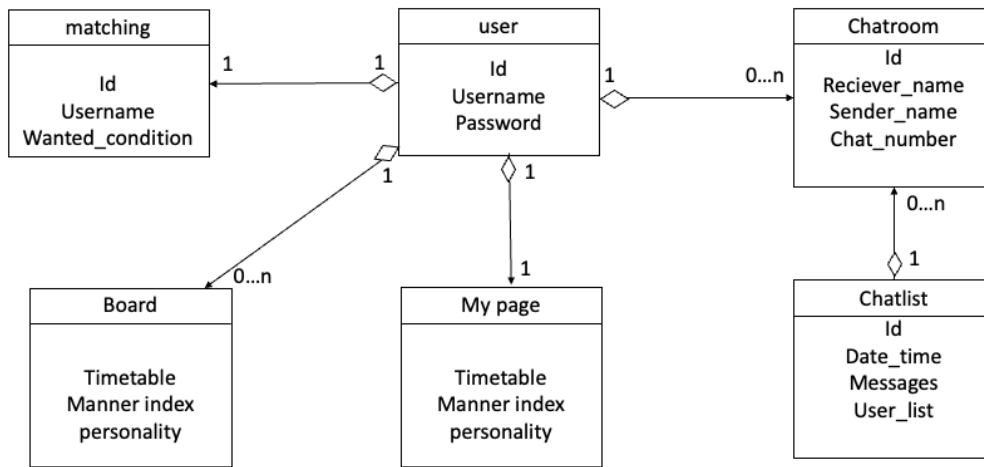
## System Architecture

### Controller

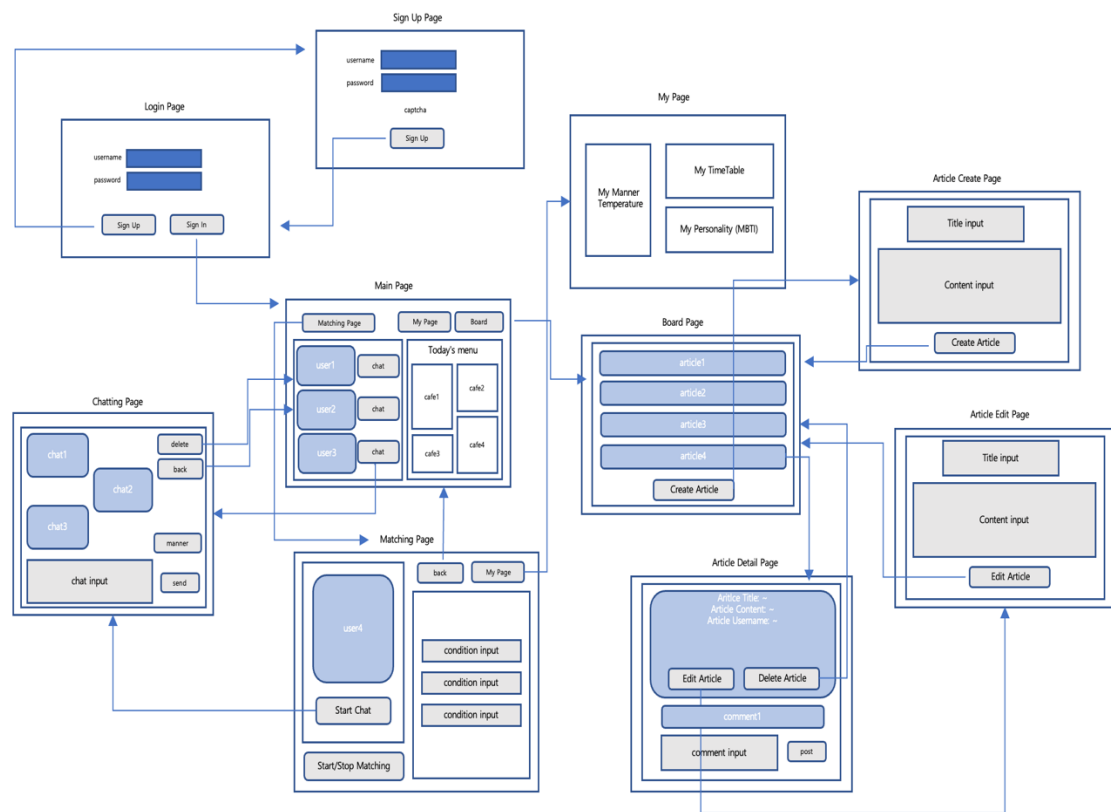
Here is the MVC model.



# Model



# Interface



1. Login page (/sign\_in)
  - Sign in
  - Go to 'sign up page'
2. Sign up page(/sign\_up)
  - Sign up a new user
  - Get 'username', 'password' as inputs
3. Main page (/)
  - Go to 'matching page'
  - Go to 'My page'
  - Go to 'Board page'
  - Show the chat room and if the user clicks 'chat' button, go to the 'chatting page'
  - Show Today's menu each restaurant
4. Chatting page (/chatting/id)
  - Show the last chat messages
  - If user writes the text and clicks the 'send' button, the message is sent

- Show the 'manner temperature'
- If the user clicks 'delete' button, delete the chatting page
- If the user clicks 'back' button, close the 'chatting page' and go to 'main page'

#### 5. Matching page (/matching)

- The user can set the condition
- If the user clicks 'start/stop matching' button, Start the matching or stop the matching
- If the user clicks 'my page' button, go to 'my page'
- End of the matching, show the other user profile and create the 'chat' button
- If the user clicks 'chat' button, go to 'chatting page'
- If the user clicks 'back' button, go to 'main page'

#### 6. My page (/my\_page/:id)

- Show the user's manner temperature
- Show the user's timetable
- show the user's personality

#### 7. Board (/board)

- Show the article(post)
- If the user clicks the articles, go to the 'Article Detail page'
- If the user clicks 'create article' button, go to the 'Article create page'

#### 8. Article Detail page (/board)

- Show the title, contents, username
- If the user clicks 'Edit article' button, go to the 'Article Edit page'
- If the user clicks 'Delete article' button, delete the page and go to 'Board page'
- If the user writes the comments and clicks the 'post' button, create the comment

#### 9. Article Edit page (/board)

- The user can edit title and contents and then clicks the 'Edit article' button, article is rewritten and go to 'Board page'

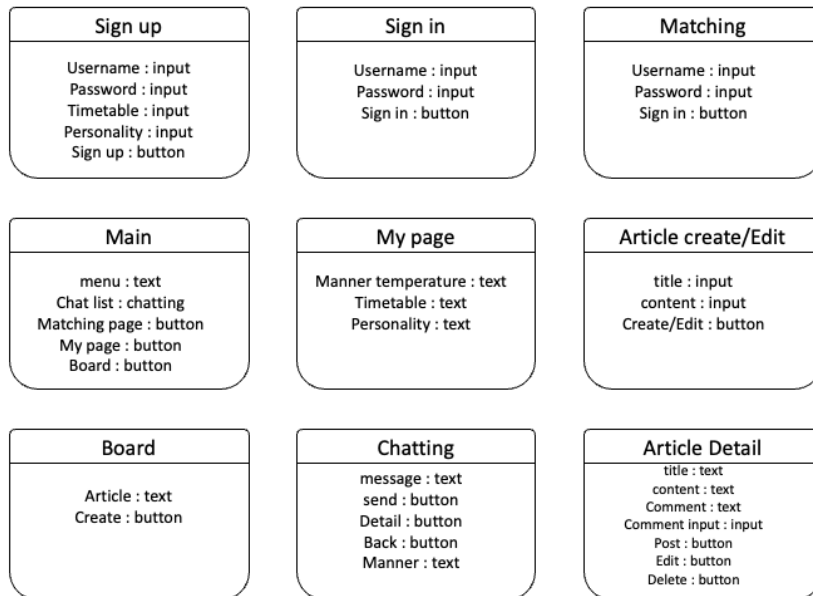
#### 10. Article Create page (/board)

- The user writes title and contents and then clicks the 'Create article' button, article is created

# Frontend Design

## Component

component



### 1. sign up

- onClickSignUpButton : If sign-up is accepted, redirect to sign in page.

### 2. sign in

- onClickSignInButton : If the user input valid username & password, redirect to main page.

- onClickSignUpButton : make the new user. redirect to the sign up page.

### 3. main

- onClickMatchingButton : redirect to Matching page.

- onClickChattingButton : redirect to each chatting page.

- onClickMypageButton : redirect to My page.

- onClickBoardButton : redirect to Board page.

### 4. my page

- onClickTimetableButton : edit timetable..

### 5. chatting

- messageInput : input the chat message as string.

- onClickSendButton : send POST to chat API.

- onClickDeleteButton : delete the chatting page and redirect to main page.

- onClickBackButton : redirect to main page.

## 6. matching

- onClickMatchingButton : start the matching, using the user's setting condition. Click the button while matching , stop the matching.
- onClickConditionButton : select the condition.
- onClickChatButton : redirect to the chatting page.
- onClickBackButton : redirect to the main page.
- onClickMypageButton : redirect to the my page.
- 

## 7. Board

- onClickArticleButton : redirect to the Article detail page.
- onClickCreateButton : redirect to the Article create page.

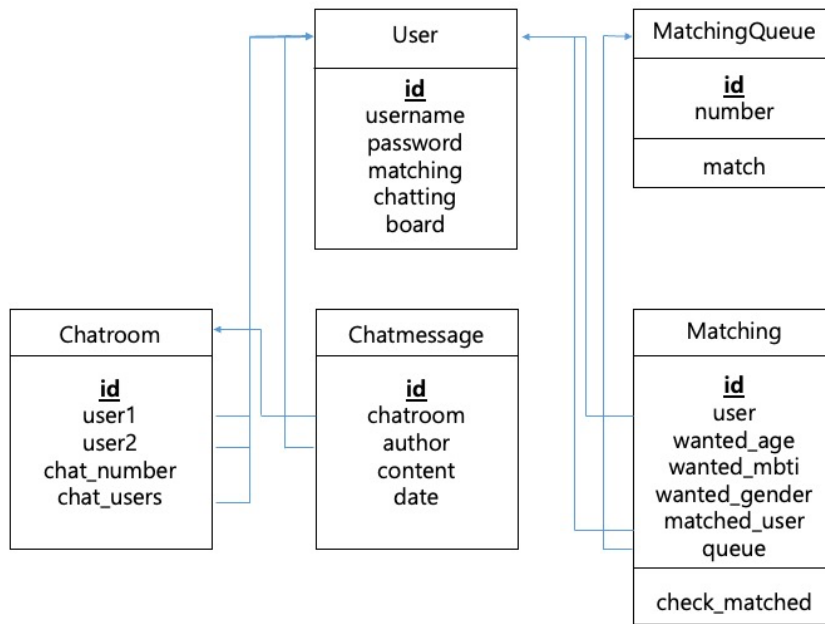
## 8. Article create/Edit

- TitleInput : input the title as a string.
- contentInput : input the content as a string.
- onClickCreateButton(or onClickEditButton) : create(edit) the article with the entered input. And redirect to the Board page.

## 9. Article detail

- commentInput : input the comment.
- onClickPostButton : Post the comment. And show the comment in the comment box.
- onClickEditButton : redirect to the Article Edit page.
- onClickDeleteButton : Delete the Article, and then redirect to the Board page.

# Backend Design



API	url	get	post	put	delete
User	/user/signin/	x	signin	x	x
	/user/signup/	x	signup	x	x
	/user/signout/	logout	x	x	x
	/user/:id/info/	get user info	set user info	x	x
Matching	/matching/start/	x	enqueue match	x	stop matching
	/matching/check/	check match status	x	x	x
Chat	/chat/user/:id/	get user's chatroom	x	x	x
	/chat/chatroom/	x	make chatroom	x	x
	/chat/chatroom/:id	get message	send message	x	remove chatroom
	/chat/user/:id/:id	x	x	x	remove message
Board	/board/	get article list	TBD	x	x

## Implementation Plan

Components	Feature	Importance	Sprint	Person	Challenge
------------	---------	------------	--------	--------	-----------

Matching	algorithm	3	3	김의기, 손민규	Make appropriate algorithm
	function	3	2	김의기, 손민규	
Chatting	Django chat	3	3	강병우, 김광현	
	Socket chat	3	2	강병우, 김광현	Using socket
Main	Redirect to chatting	1	3	김의기	
	Redirect to maching	1	3	김의기	
	Redirect to board	1	3	김의기	
	Redirect to my page	1	3	김의기	
	Show the menu	1	3	김의기	
Login	Sign up	1	3	김광현	
	Sign in	1	3	김광현	
Board	Show article list	2	3	강병우	
	Create article	2	3	강병우	
	Edit article	2	3	강병우	
My page	Show the information	2	3	손민규	
	Edit the information	2	3	손민규	

## Testing Plan

### Unit Testing

Every components and modules should be tested. In each sprint, we would test implemented modules by following frameworks.



- React : React-testing-library, Jest
- Django : Python unit test

### Functional Testing

Every APIs should be tested.

- React : React-testing-library, Jest
- Django : Python unit test

In Sprint3, cover test RESTful API about matching, chat, User. In Sprint4 test API about User model.

### Acceptance & Integration Testing

- Acceptance Testing : Cucumber
- Integration Testing : Travis CI