

1. 프로젝트 소개

Calendargrams는 개인 Calendar기능과 자신의 Schedule을 공유할 수 있는 것이 특징인 SNS 서비스이다. 기존 카카오톡의 경우 일정을 채팅상 서로 합의 할 수 있으나, 이를 외부 캘린더로 바로 저장할 수 있는 기능이 부재하다. 또한 페이스북의 경우 요일 별 스케줄 리스트가 제공되나 생일 알림에만 쓰이는 등 실제 사용이 많지 않다. 이와 다르게 Calendargrams를 사용한다면 추가된 친구 간에 버튼 한번으로 자신의 달력에 일정을 추가할 수 있는 직관적이고 쉬운 API를 통해 친구간 손쉬운 스케줄 공유를 할 수 있다. 각자의 스케줄은 자신의 캘린더에 저장되는 형태이므로, 노출의 염려가 없이 원하는 스케줄만 공유를 하여 만들 수 있다.

이와 함께, 프로필 사진 업로드시 선정적인 이미지를 검열하는 기능을 추가하였다. 또한 그룹간 채팅기능을 강화하여 이미지 업로드 및 채팅방 자체를 나가거나, 중단하는 기능을 제공한다. 이와 함께 위치정보를 활용하기 위해 HTTPS 보안이 적용된 도메인을 이용하여 보안을 강화하였으며, 구글 MAPS API를 이용해 브라우저의 위도 및 경도 정보를 address 로 변환하여 보여준다. 이외에도 기본 기능에서 세부적인 디테일이나 디자인 측면에서 깔끔하고 직관적인 디자인으로 개선한 부분이 있다.

프로젝트는 프론트엔드는 React, 백엔드는 Django를 이용하여 개발하였다. 프론트엔드에서 React와 함께 state 관리를 용이하게 하기 위해 Redux를 사용하였으며 서버와 통신하기 위해 redux-saga를 함께 사용하였다. 여기에 react 생태계에서 이용 가능한 다양한 컴포넌트를 활용함으로써 제한된 개발 시간 내에 개발 효율을 높였다. 백엔드는 Django-Rest를 이용하여 프론트에서 rest api를 통해 수행되는 요청을 처리하게 하였다. 이와 함께 selenium을 이용한 frontend test, python을 통한 backend test, Jest를 이용한 unit test를 수행하여 버그 발생시 최대한 빠른 수정을 통해 작은 버그가 추후 큰 문제를 일으키지 않게 선제적으로 처리하도록 노력하였다.

2. 프론트엔드 디자인/백엔드 구조 소개

구현한 SNS는 일정 공유 기능에 방점이 있어서 이를 첫 페이지에서 부각하려 했다. 이에 따라 첫 페이지와 로그인, 회원가입 페이지에 달력 이미지를 배경으로 했다. 또한 구현한 SNS의 이름을 calendar와 instagram을 합성하여 calendargram이라고 지었으므로 이 이름을 instagram의 폰트로 만들어 첫 페이지에 보이도록 했다.[1][2] 또한 전체적으로 주제 색을 정하는 것이 보기에 좋을 듯하여 pantone에서 올해의 색으로 선정한 greenery(#88b04b)를 네비게이션 바에 적용했다. 이 네비게이션 바는 첫 페이지를 제외한 모든 페이지에서 노출된다. 또한 이 주제 색에 어울리도록 대부분의 버튼을 초록색으로 했다. 그 외에는 전반적으로 깔끔하고 군더더기 없이 사용자가 올리는 콘텐츠가 부각되도록 하기 위해 회색과 흰색 바탕을 주로 사용했다.[3]

프론트엔드의 개발에서는 매우 많은 컴포넌트를 작성하게 되었다. 이를 하나의 기능을 하는 가장 작은 컴포넌트로 나누어 작성하여 관리가 용이하다. 백엔드는 Django에서 제공하는 models.py, serializers.py, views.py를 이용하여 적절한 페이지를 만들어 프론트엔드의 요청을 처리하며, 여러 데이터 형태를 처리하는 과정에서 다양한 스타일의 코딩을 하였다. 백엔드 API는 APPENDIX [] 과 같다.

3. 서비스의 특징 및 특별한 기능

A. 일정 기능

일정 기능의 특징은 유저 개인별 캘린더가 제공되어 private한 calendar로 활용할 수 있는 동시에, 다른 유저에게 쉽게 스케줄 공유 요청이 가능하다. 이를 위해 일정 기능에는 크게 세 가지 컴포넌트가 있는데, 일정이 표현되는 Calendar[1], 개인 또는 친구와의 일정을 추가하는 Schedule, 자신의 스케줄을 현재 이후 임박한 순서로 보여주는 ScheduleList이다.

Calendar는 title, start, end 으로 구성된 object의 리스트로 구성되어 있다. 이러한 object는 외부 google api나 다른 일정관리 앱에서 기본적으로 가지고 있는 데이터 형태이므로, 추후 연결이 용이하다. 여기에 Schedule를 추가할 시에 친구 추가 버튼을 누르면 자신의 캘린더에 일정이 등록됨과 동시에 친구에게 일정 추가 수락 요청이 전달된다.[1] 친구는 이를 추가하거나 삭제할 수 있으며, 이 경우 바로 자신의 달력에 일정이 등록된다.

자신의 일정이나 수락한 일정은 모두 캘린더에 표현이 됨과 동시에 ScheduleList에 표시되는데, List는 임박한 순서대로 표출되며, 24시간 이내 일정이나 1주일 이내 시작되는 일정은 노란색으로 표시를 하여 [] 쉽게 인식할수 있도록 하였다.

B. 프로필 이미지 검열

프로필 이미지는 친구가 아닌 유저에게도 노출된다. 이에 따라 프로필 이미지에 선정적인 사진이 있을 경우 유저들에게 불편함을 줄 수 있다. 기존의 SNS 등은 프로필 이미지에 대한 신고가 들어올 경우 사후 검열의 방식을 취하지만, 선정적인 이미지를 사전에 차단하는 것이 보다 효율적이라고 생각되었다. 따라서 프로필 이미지를 업로드할 때[1] 백엔드에서 CNN기반의 Deep Learning 모델을 사용하여 선정적인지 아닌지 판별하도록 했다. 선정적인 이미지가 발견될 경우 업로드를 거부하고 팝업창을 띄워 이를 알린다.[2] 학습된 모델의 confidence 또한 확인할 수 있다.

CNN 모델을 학습시키기 위한 전체 데이터는 NSFW(not safe for work) 8000여장, SFW(safe for work) 20000여장이었으며 SFW는 Easy와 Difficult의 두 가지 기준으로 분류했다. Easy는 동물, 사물 등 선정적 요소가 전혀 없는 사진들이며 Difficult는 비키니, 상의를 탈의할 남성 등 살색이 많이 포함되었으나 선정적이지 않은 이미지이다.

초기에 VGGNet16을 학습시켜 만든 모델은 테스트 셋에 대해서 94%의 정확도를 보였으나 학습된 가중치 용량이 400MB에 육박했다. 이는 aws instance의 메모리에 올리기에 큰 용량이었다. 따라서 디스크 일부를 메모리로 사용할 수 있도록 swap을 설정했다. 또한 모델의 크기를 줄이기 위해 ResNet50을 Transfer Learning으로 재학습하여 사용했다. 이 모델은 가중치 용량이 300MB 정

도로 기존보다 100MB의 용량 절감 효과가 있었다. 또한 테스트 셋에 대해서 96%의 정확도를 보여 정확도 면에서도 소폭 향상되었다.

구현한 모델은 Deep Learning 모델이 대체로 그러하듯 정확하지는 않다. 모델 성능상의 한계는 여러가지 이유가 있을 수 있는데, 첫째로 공개된 같은 종류의 학습 데이터 셋이 존재하지 않기 때문에 학습 데이터를 처음부터 새로 구성하였고, 이 과정에서 엄밀하지 못한 라벨링이 있었다. 가령 비키니 사진은 SFW에도 포함되고 NSFW에도 일부 포함되어 있었다. 둘째로 이미지를 108*108 사이즈로 resize하여 데이터 셋을 구성했기 때문에 해상도에서 크게 손실이 있었다. 이에 따라 여성의 유두와 같이 작은 부분을 찾아내는 데에는 어려움이 있는 듯하다. 또한 셀카 사진 등을 학습 데이터에 포함 시키지 않은 것도 한계로 작용한다. 하지만 대체로 선정적인 이미지와 선정적이지 않은 이미지를 잘 구별하는 편이며 백엔드에서 검열에 소요되는 시간이 대략 1~2초 정도로 런타임에 실시간으로 검열 하기에 속도 면에서 충분하다고 생각된다. 학습 데이터 셋을 보다 엄밀하게 구성하여 재학습시키면 충분히 상용화가 가능할 것으로 보인다.

C. 위치정보/HTTPS 세팅

D. 친구 기능/유저 검색 기능

소셜서비스의 핵심이라 할 수 있는 친구 기능은 친구 요청 보내기, 요청 수락 혹은 거절 그리고 친구 목록 관리로 구성된다. 친구 요청을 보내기 위해서는 먼저 네비게이션 바에서 친구 탭길 원하는 유저의 아이디를 검색하고, 목록 가운데 검색한 유저를 클릭해 유저의 담벼락으로 이동한다[1]. 이동한 담벼락에서 친구 상단에 위치한 버튼을 눌러 요청을 보내거나, 보낸 이후에는 다시 이 버튼을 클릭해 요청을 취소할 수 있다[2]. 또 요청을 받은 유저는 본인의 담벼락에서 요청 목록 보기 버튼을 눌러, 본인이 받은 요청을 수락 혹은 거절하거나 보낸 요청을 취소할 수 있다[3]. 또 친구 목록은 본인의 담벼락에서 친구 목록 보기 버튼을 눌러 볼 수 있다. 목록 가운데 친구 삭제 버튼을 누르면 간편하게 친구 끊기를 할 수 있다[4].

친구 기능에서 유저의 편의를 위해 추가적으로 구현한 기능은 크게 두 가지 이다. 첫째는 목록에서 유저 아이디를 누를 시에 해당 유저의 담벼락으로 이동가능 하게 한 것이고 둘째는 목록의 가독성을 고려해 친구 요청 거절 또는 친구 끊기 버튼을 누를시에, 취소선을 그어 목록을 재표시한 것이다[5].

E. 채팅, 그룹채팅 기능

채팅 기능의 특징은 일대일 채팅과 그룹채팅으로 구분, 채팅방에 이미지 업로드 가능, 그룹 채팅에서 중복 유저 채팅방 구현 가능, 그룹 채팅 도중에 유저를 추가하거나, 채팅방 자체를 나가거나 들어오는 기능이다.

그룹 채팅의 경우 채팅 방을 만들 수 있으며 친구 여럿을 선택하여 채팅 방을 생성할 수 있다.[2] 같은 인원이라든가 여러 개의 그룹 채팅 방을 만들 수 있고 생성 시간으로 구분된다. 그룹 채팅 방은 만들어진 이후에도 친구를 초대할 수 있으며[3] 채팅 방을 나가는 것도 가능하다[4].

채팅 메시지는 내가 보낸 메시지를 오른 쪽에, 채팅 상대가 보낸 메시지를 왼쪽에 정렬하도록 하여 한눈에 메시지를 구분할 수 있게 했다. 또한 사진을 메시지로 보낼 수 있다.[5] 채팅 메시지

가 여러 개 쌓이면 채팅창이 무한히 길어지는 것을 막기 위해 채팅창 안에 스크롤을 두었다. 이 스크롤은 채팅 방에 입장하거나 새로운 메시지가 있을 경우 자동으로 맨 밑으로 가도록 했다. 채팅은 실시간으로 가능하며 0.5초마다 새로운 메시지가 있는지 확인하여 업데이트 한다.

일대일 채팅과 그룹 채팅은 나누어서 독립적으로 구현했는데, 이는 구현 상의 편의를 위해서이기도 하지만 유저 간의 일대일 채팅을 보존하기 위해서이기도 하다. 그룹 채팅의 경우 참여 인원이 가변적이어서 누군가를 초대하거나 채팅방을 나가면 기존에 유저 간에 일대일로 채팅하던 기록이 손실될 우려가 있기 때문이다.

F. 포스트/코멘트, 좋아요 기능

프로젝트1에서 진행했던 일반적인 글쓰기 기능을 확장하여 일반 텍스트에 사진첨부 기능과 혹은 투표(poll)를 생성기능을 추가하였다. 또, 획득한 위치를 포스팅시 표시할 수 있도록 했다.

토글기능이 있는 그룹버튼을 이용해 각 기능을 버튼과 연동시켰다. 일반, 사진, 투표, 위치 기능이 한 군데 있어 사용자가 원하는 내용을 손쉽게 포스트에 포함시켜 글을 쓸 수 있도록 하였다[1]. 투표 기능의 경우, 사용자가 투표 항목을 고정시키지 않고 동적으로 추가시킬 수 있도록 하였다[2]. 또 투표가 끝난 항목에 대해서는 단순히 투표 수만을 표시하는 것이 아닌 react chart npm 패키지를 이용해, 투표 상세 내용을 한눈에 들어오도록 도식화시켰다[3].

좋아요 기능의 경우, facebook처럼 사용자가 다양한 이모티콘을 선택하도록 인터페이스를 구성하였다[4]. 🍌 버튼은 사용자가 글 혹은 코멘트에 손쉽게 간편하게 반응할 수 있다는 장점이 있다. 한편 이모티콘 선택 방식을 택함으로써, 단순함과 간편함에 더불어 사용자의 반응 자유도 또한 고려하였다. 코멘트 기능의 경우에도 포스트에 달린 모든 댓글을 표시하는 것이 아닌, 사용자가 원하는 글의 댓글만을 볼수 있도록 코멘트 접기, 펴기 기능을 추가하였다[5].

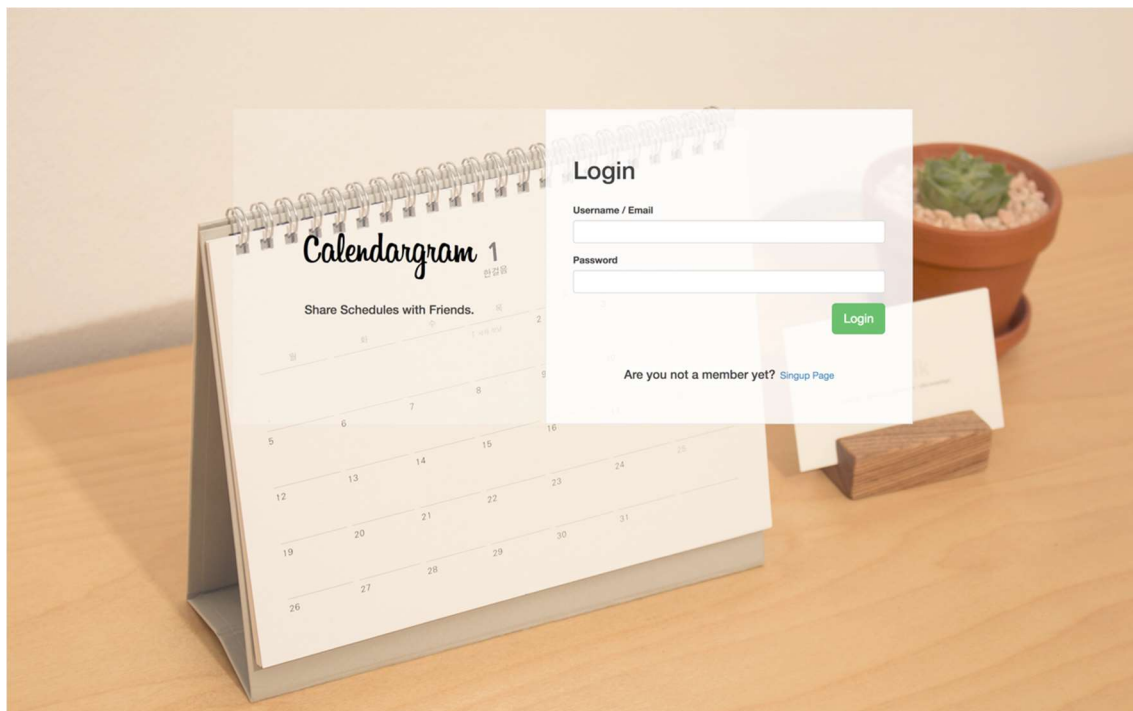
G. 기본 기능

4. 개발 과정 및 결론

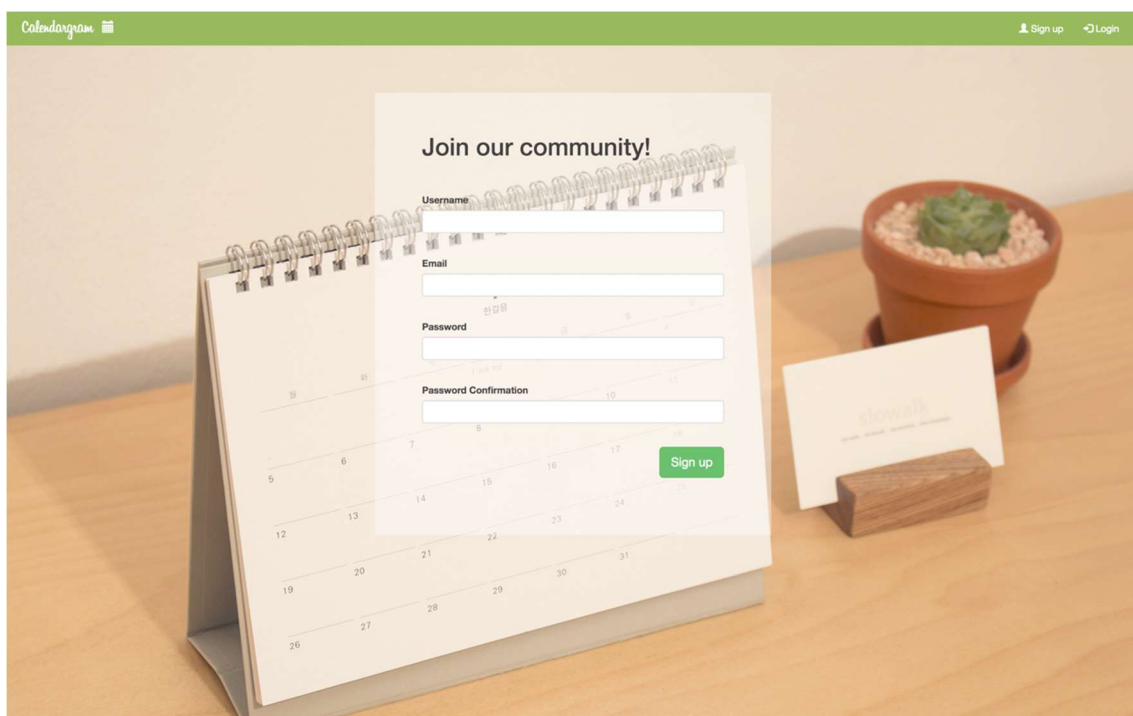
매주 1~2회 직접 만나 서로의 작업에 대한 평가와 배분, 계획을 세우면서 적절한 양의 작업이 진행될 수 있도록 노력하였다. 이를 통해 제한된 시간에 상당히 많은 기능을 구현하는 것이 가능했다고 생각된다. 또한, 수시로 메신저를 이용하여 피드백을 주고, 빗버킷 이슈와 커밋, 풀리퀘스트를 통해 필요에 따라 소통을 하며 개발이 원활하게 진행될 수 있었다.

현재 Calendargrams는 다양한 기능의 기본적인 부분을 제공한다. 예를 들면, 일정의 경우 기본적인 리스트 형태의 정보를 parsing하여 달력에 구현하고, 위치정보의 경우 Google Maps API를 이용할 수 있는 상태이다. 따라서 추후 이를 통합하여 더 많은 기능을 구현해 나가기 용이하다. 따라서 본 프로젝트 이후 추가 개발을 통해 새로운 소셜 네트워크 서비스로 충분히 런칭 가능할 것으로 생각된다.

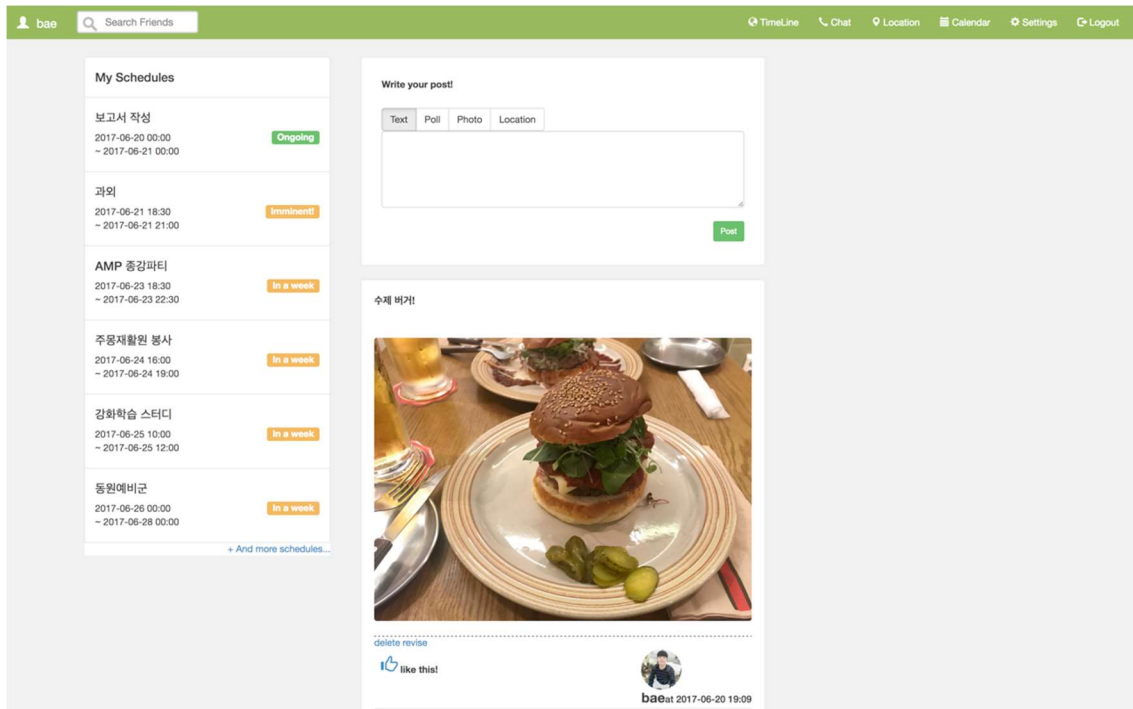
Fig 1.



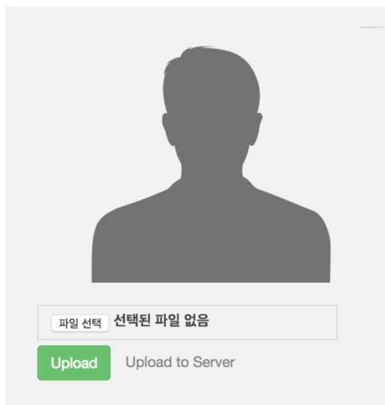
[1] 접속시 첫 페이지



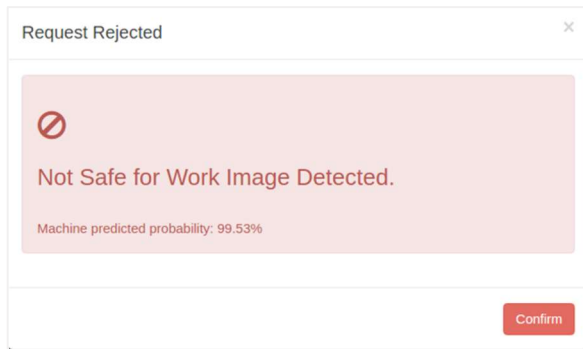
[2] 회원가입 페이지. 로그인 페이지는 회원가입 페이지와 비슷하다.



[3] 타임라인 페이지



- 1.
2. [1] 프로필 이미지를 업로드할 수 있는 settings 페이지의 일부
- 3.

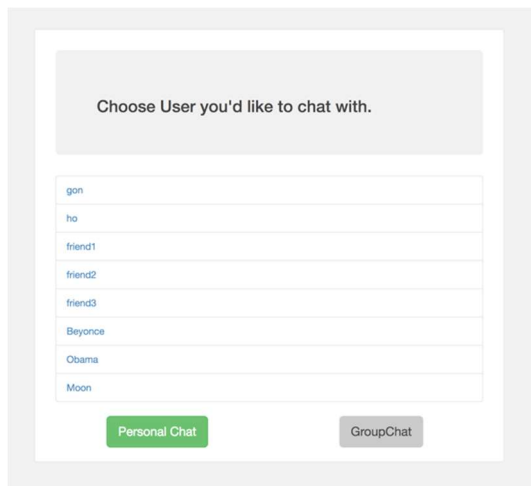


4.

5. [2] 선정적인 이미지를 업로드할 경우의 팝업창

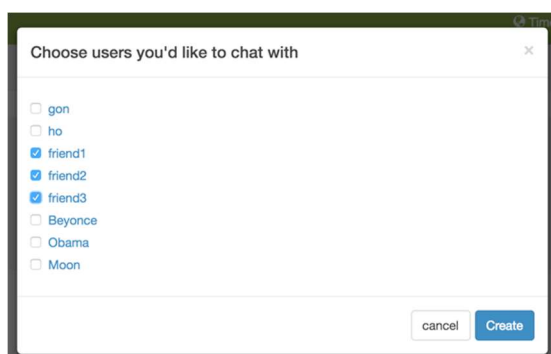
6.

7.



8. [1] users/ 페이지의 일부

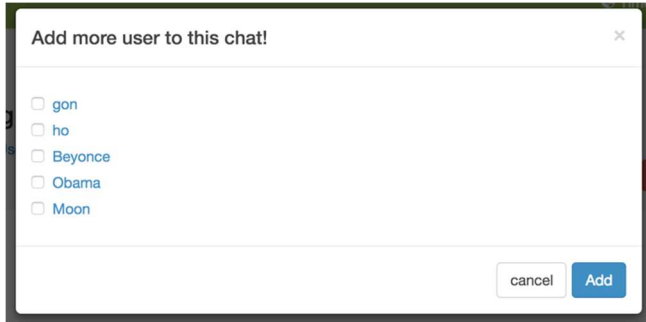
9.



10.

11. [2] 채팅방 생성을 위한 팝업창

12.

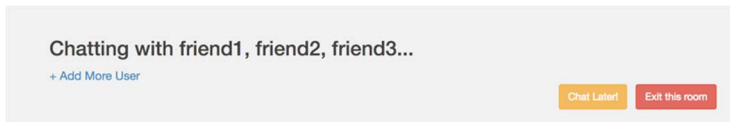


13.

14. [3] 그룹 채팅방 친구 초대를 위한 팝업창

15.

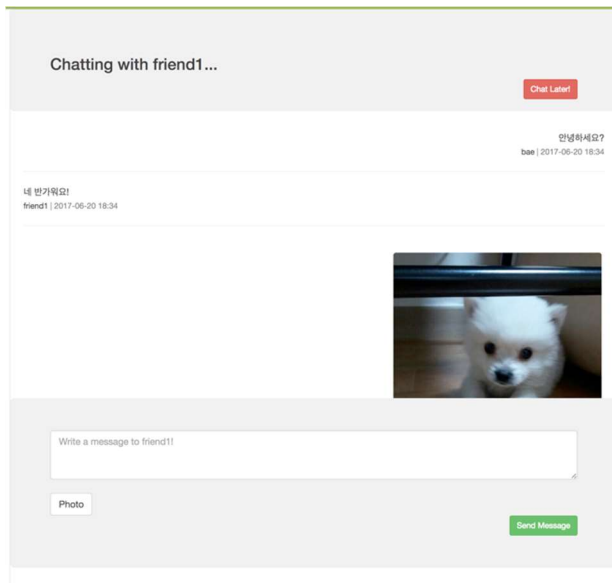
16.



17.

18. [4] 그룹 채팅방의 일부. 오른쪽 빨간 버튼을 통해 채팅방을 나갈 수 있다.

19.



20.

21. [5] 채팅 메시지와 사진 보내기.

22. [1] 글쓰기 그룹 버튼 인터페이스

Write your post!

Text	Poll	Photo	Location
------	------	-------	----------

23.

24. [2] 투표 항목 동적 표시

Write your post!

Text	Poll	Photo	Location
------	------	-------	----------

Create Poll

Topic:

What day do you prefer the most?

Options 1

June 18

Options 2

June 19

Options 3

June 20

Options 4

Enter option 4...

Create

Post

25.

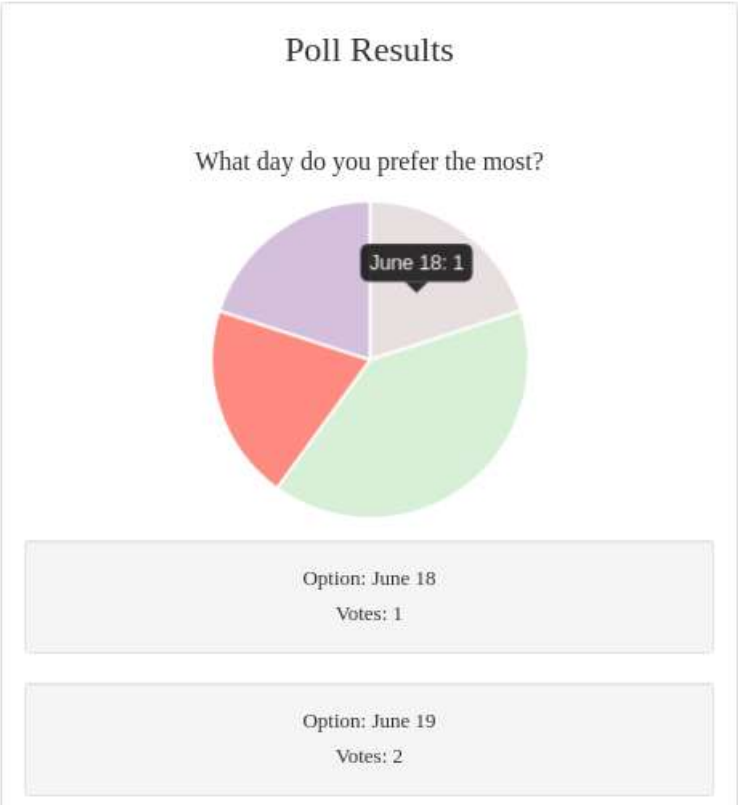
26.

27.

28.

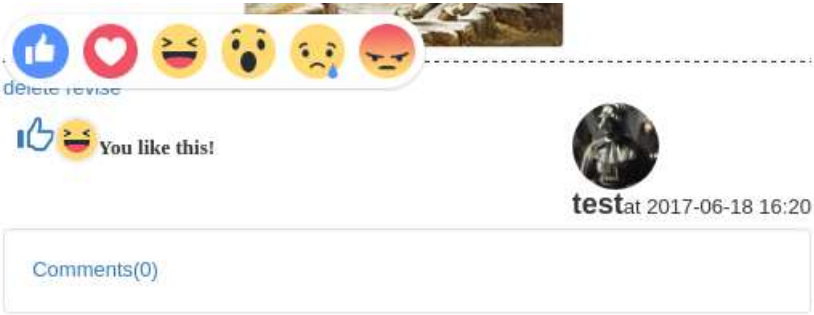
29. [3] 투표 결과 도식화

This is the poll for possible mid term day, the most voted day will be chosen.



30.

31. [4-1] 다양한 이모티콘 선택 (전)



32.

33. [4-2] 다양한 이모티콘 선택 (후)



34.


35.

36.

37. [5-1] 코멘트 여닫기 (닫기)

Comment Test

You like this!



sns_adminat 2017-06-21 21:49

Comments(3)


38.

39.

40. [5-1] 코멘트 여닫기 (열기)

Comment Test

You like this!



sns_adminat 2017-06-21 21:49

Comments(3)

I like obama

user1

2017-06-21 21:50

You like this!

So do I ! :)

user2

2017-06-21 21:51

You like this!

No, Trump is the best ~

user3

2017-06-21 21:52

like this!

Leave Comment

41.

42.

43.

44.

45.

46.

47.

48.

49.

50.

51. [6] 사진 글쓰기

**“All great events hang by a single thread.
The clever man takes advantage of everything,
neglects nothing that may give him some added opportunity;
the less clever man, by neglecting one thing,
sometimes misses everything.”**
— Napoléon Bonaparte



[delete](#) [revise](#)

 [like this!](#)



testat 2017-06-18 16:20

[Comments\(0\)](#)

52.