GWAJAMS

과잠쓰

과잠 디자인 웹 서비스

TEAM 9

신승아, 이세리, 정현우, 한수현

https://github.com/swapp201901-team9

Abstract

Every start of new semester, most of school clubs or majors design specific uniforms to represent their group called ‘과잠’. These uniforms help the group members to feel a sense of belonging and improve teamwork. ‘과잠’ could be any kind of clothing that a certain group customized, but mostly it refers to certain type of jacket that has great similarity to baseball jacket. Most of students in high college in South Korea probably have had the experience of customizing and ordering these ‘과잠’. However, because this specific clothing is used to represent the whole group, a lot of conflicts arise when customizing ‘과잠’, especially when it is related to design. Group members want their ‘과잠’ to be original and unique, but it is very hard to agree on one specific design due to the difficulty of predicting the output of their design. Therefore, we propose a web service to simulate ‘과잠’ designs. Our web service will also provide voting systems related to the design in order for each group to reach on an agreement about the design.

Motivation & Problem

To make things more clear, we can summarize our web service into 3 main parts:

1) Simulate 과잠 design: Users can simulate the design of their 과잠. This process will be done very thoroughly, from choosing the sleeve color to inserting their school/club logo. Users can save and put a vote to this simulation but they will have to log in and make/choose their community.

2) Make/Join Community: Users can make their own community and let the other group members join the community by logging in and choosing their community. In this community, they can manage their designs that any of the group members created, or they can put their designs to a vote.

3) Vote and agree on the best 과잠 design: Group members in a community can vote on a design they like. Every time a group member votes or suggests an opinion, group page will be updated and shown to all the members in the group.

People who belong to certain kind of organization or community are our potential customers. Since we expect a lot of customers from college or school clubs, we expect an increase of use at the start of every semester or every quarter. However, because a lot of organizations or communities are made at any time of the year and they also require some kind of uniform, we expect an average-looking frequency of access in general.

Related Work

A lot of companies provide design services for their 과잠 selling website. But our web service holds three major differences and therefore has greater competitive edge over these web services.

Websites that provide 과잠 design services are owned by certain manufacturing factories or companies. Therefore, if you want to use the design that you created in this website, you will probably have to use this particular kind of manufacturing company. However, our web service is an independent design simulation service and provides 과잠 design simulation regardless of any kind of factory or company. Simulation, saving, creating communities and voting services are provided without profit nor non-profit conditions.

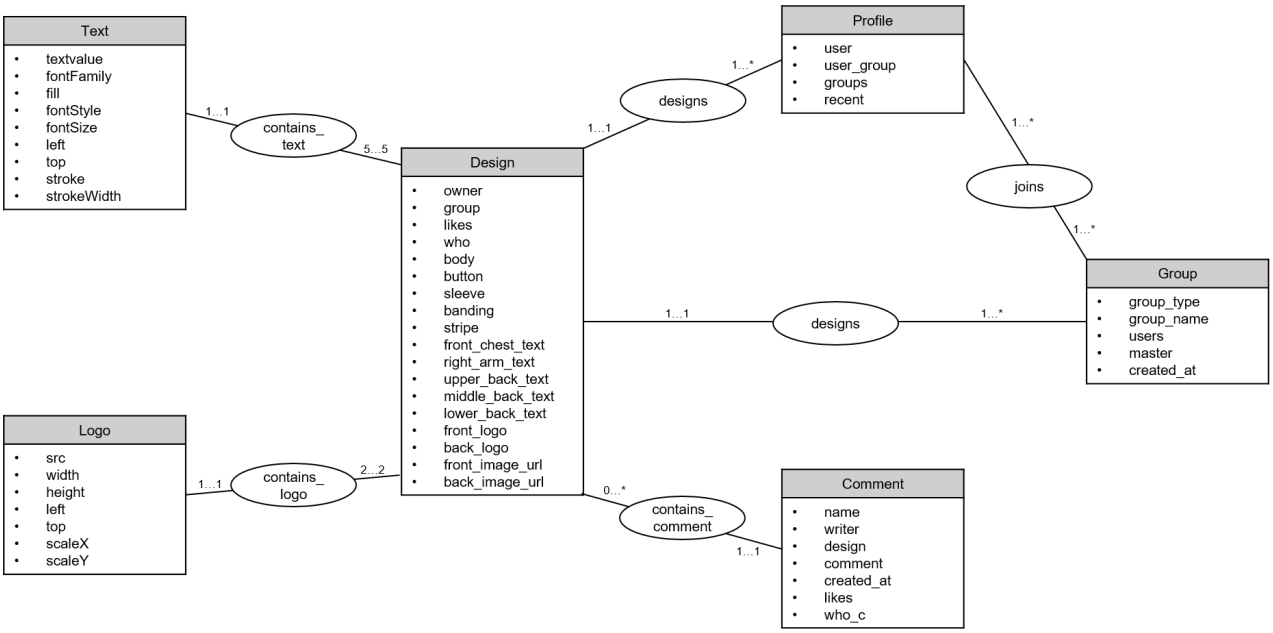
Most design simulation web services provide very basic simulations that produce only rough sketches or prototypes of the design. It is very hard to predict how the actual output or clothing is going to turn out based on these rough sketches. On the other hand, our web service provides design systems that has actual fabric color and school logos as database. Our prototype of the design will be much more sophisticated than the competing websites and will resemble the actual output clothing the most.

As mentioned before, the competing websites provide design services in order to connect users to their online selling store and therefore sell their product. So, precisely speaking, there is no website service providing 과잠 design services only for the purpose of helping the group members agree upon a specific design. Our website also provides community pages and voting systems to help users manage and choose their design within their group. Users can act as an independent individual or they can act within their attached group.

Functionality

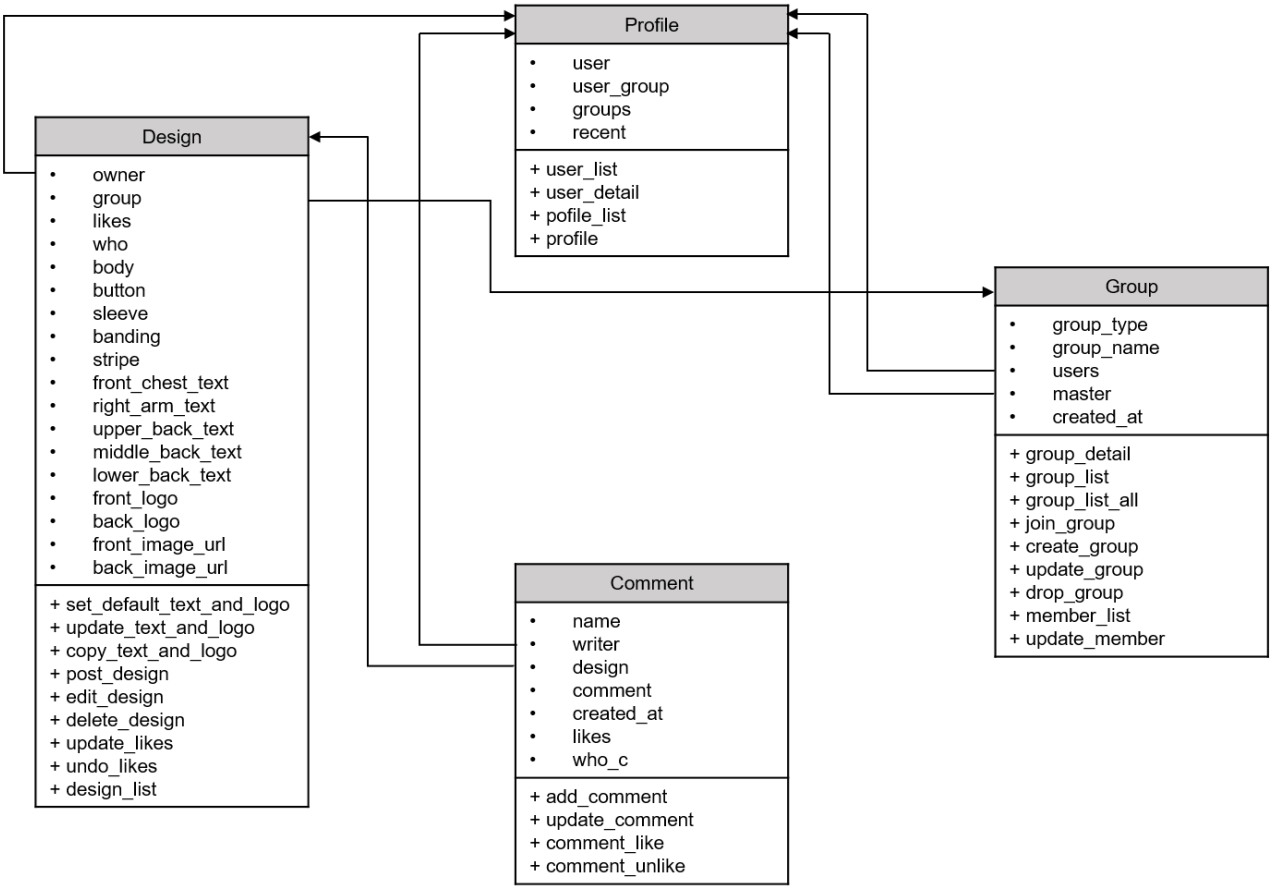
1) Model

Here is E-R (Entity Relationship) diagram for model design.



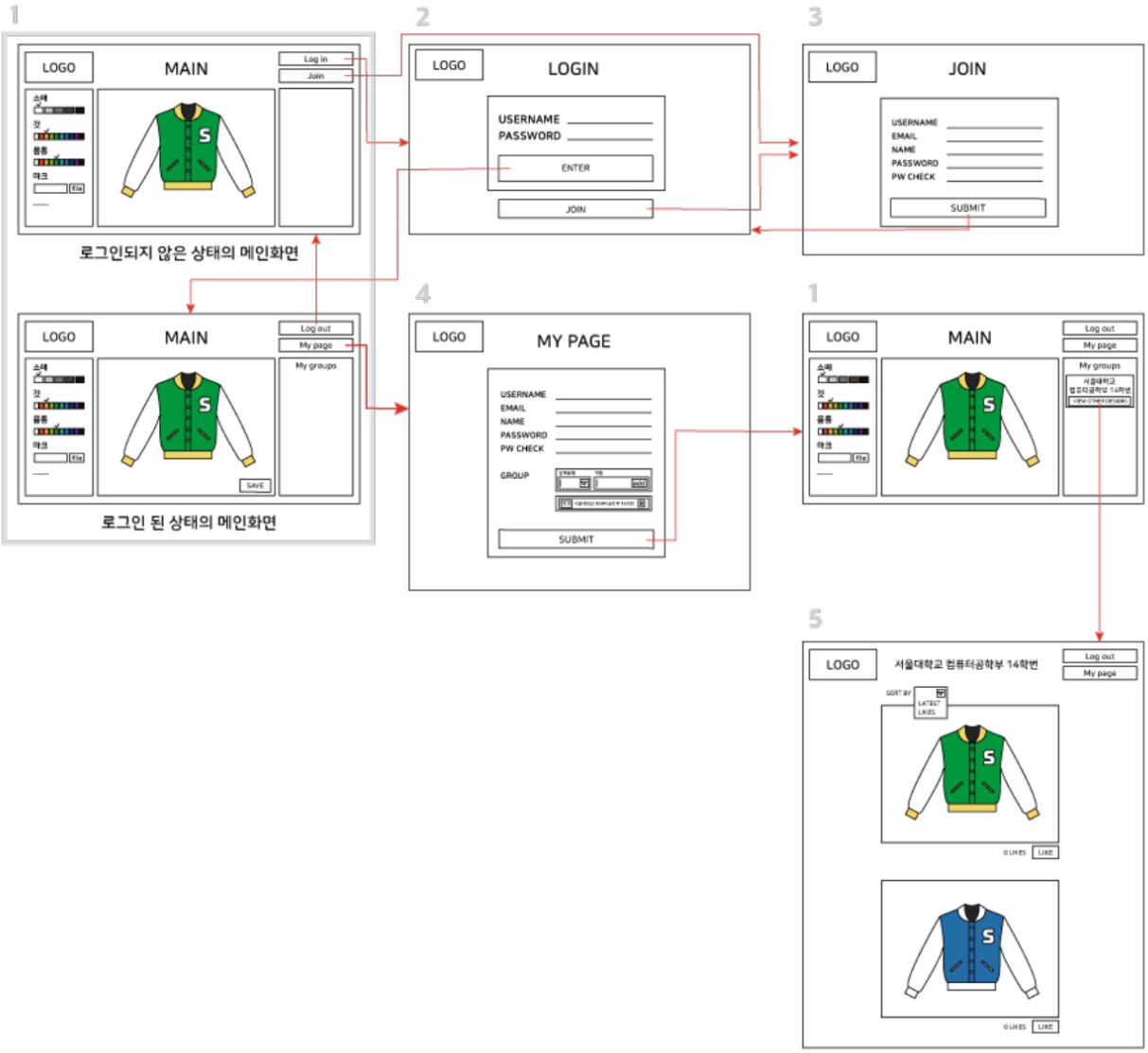
Rectangle stands for entity set, oval stands for relationship set, and numbers next to line means mapping cardinality constraints (min…max). Entity attributes are listed inside entity rectangle.

This is a relation schema diagram based on E-R diagram. This is also the structure of Django models the service is going to use. It also contains the core methods of each model.



Rectangle stands for relation schema. Schema attributes and methods for Django models are listed inside the rectangle. Arrow represents foreign key constraints.

2) View

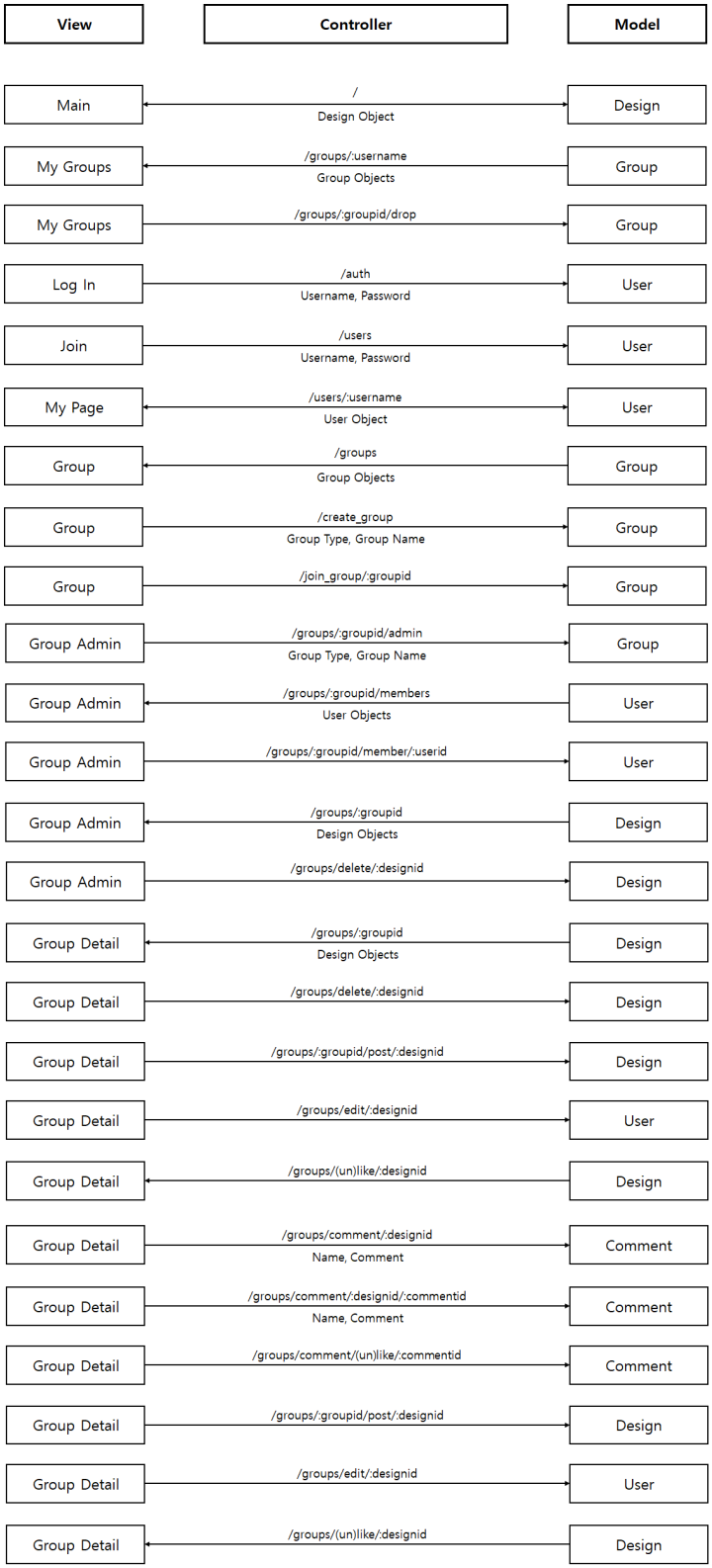


1. MAIN (/)
   - Navigation bar: our logo, buttons
     - log in, join buttons when users are not logged in
     - log out, my page, group buttons after logged in
   - Select style: select various design option of ‘과잠’
     - Pick colors for ‘과잠’ elements
     - Insert texts to ‘과잠’ design
     - Insert logos to ‘과잠’ design
   - My group
     - deactivated when users are not logged in
     - If users are logged in, they will see the group list that they joined

    - Next to each group there is a '탈퇴' button to drop from that group
    - If user is admin of a group there will be a '관리' button alongside with the '탈퇴' button and will be redirected to GROUP ADMIN page.
2. LOGIN(/log_in)
    - 'JOIN' button in case user is not registered yet
    - Get USERNAME and PASSWORD as user inputs and a 'LOGIN' button
3. JOIN(/sign_up)
    - Sign up a new user
    - Get USERNAME, E-MAIL, PASSWORD, PASSWORD CHECK as user inputs
4. MYPAGE(/profile)
    - Users can see their username
    - Users can their change passwords by clicking on the 'CHANGE PASSWORD' button
    - User can withdraw from website
5. GROUP(/groups)
    - Users can create new groups
    - Users can search for existing groups and join them
6. GROUP ADMIN(/admin/:groupid)
    - Admin can change group info (type and name)
    - Admin can delete group
    - Admin can delete members from groups
    - Admin can give admin privilege to normal users
    - Admin can also delete designs from this page
7. GROUP DETAIL – 'User Group' (/group/:groupid)
    - Show list of 과잠 which user designed (and saved)
    - Sort photos by most recent order
    - User can delete a saved design
    - User can edit a saved design
    - User can post designs to a group he/she is registered to
8. GROUP DETAIL – '동아리'/'학과' (/group/:groupid)
    - Show list of 과잠 which group members designed.
    - Sort photos by most liked and see how many users liked each of them
    - '좋아요' button
        - User can like a design
        - Users cannot like a design more than once and like button will be changed to a '좋아요 취소' button once pressed
    - Comment
        - User can post a comment to each design
        - Each comment will be liked just like designs and will be sorted by most liked also
    - Delete

- User can delete a design or a comment if he/she is the owner of it or is an admin of the group

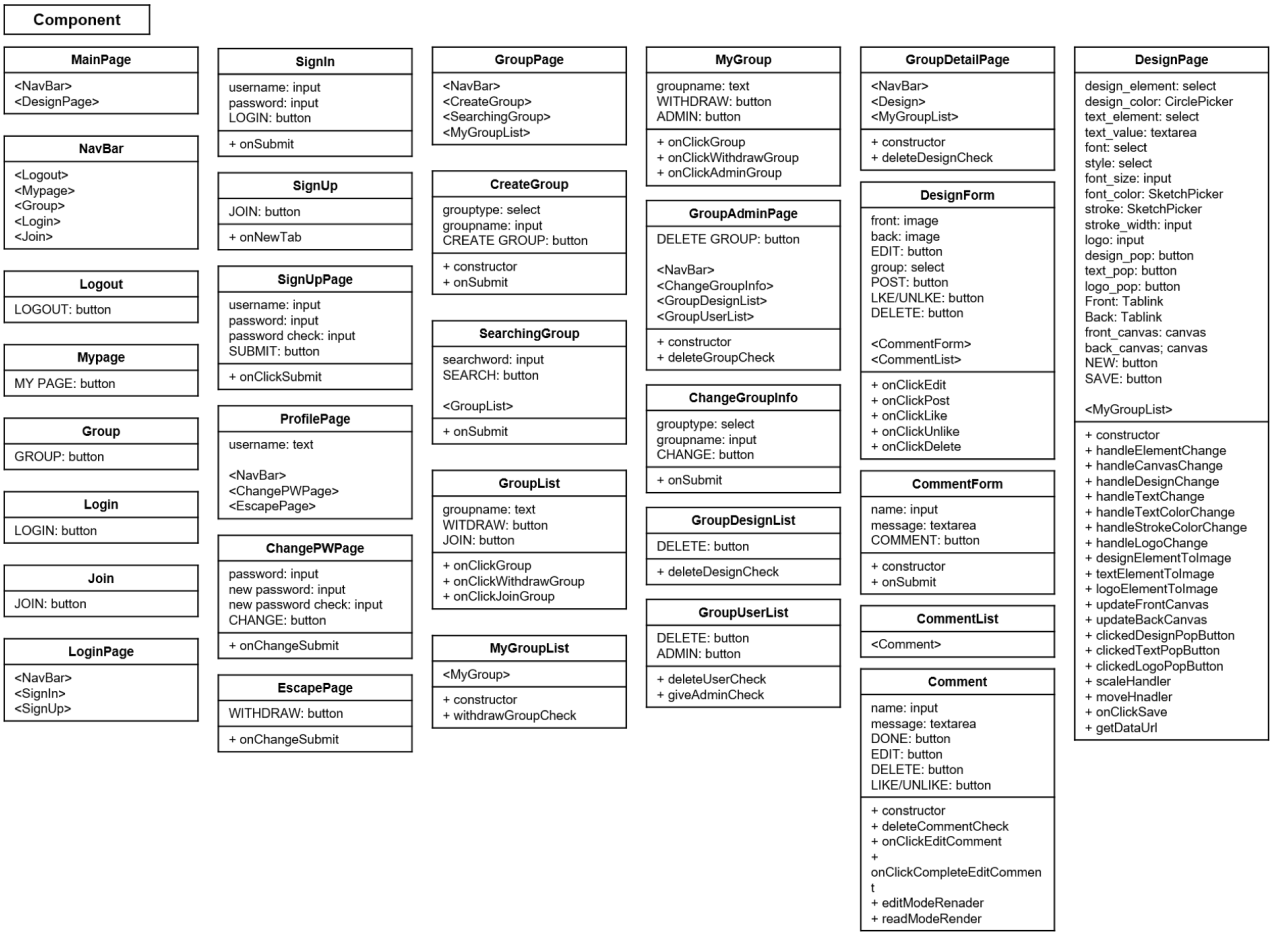
3) Controller



Design & Implementation

1. Frontend Components

Below is frontend components which contain attributes and the methods of each component.



1) MainPage
   - •<NavBar>: contain links to other pages
   - •<DesignPage>: contain design tools, canvases and my group list
2) NavBar
   - •<Logout>: link for logout
   - •<MyPage>: link to the profile page
   - •<Group>: link to the group page
   - •<Login>: link to the login page
   - •<Join>: link to the join page
3) LoginPage
   - •<SignIn>: contain sign-in form

- <SignUp>: link to the join page

4) SignIn
   - onSubmit: submit username and password => if username and password are correct, call backend signin api and redirect to the main page(logged in state)
5) SignUpPage
   - onClickSubmit: submit username, password and password check => if username does not exist already and password and password check is same, call backend sig-nup api and redirect to the main page(logged in state)
6) ProfilePage
   - <ChangePWPage>: contain changing password form
   - <EscapePage>: contain withdraw form
7) ChangePWPage
   - onChangeSubmit: submit current password, new password and new password check => if the current password is correct and the new password and the new password check is same, call backend changing password api and redirect to the main page with changed password(logged in state)
8) EscapePage
   - onChangeSubmit: call backend withdraw api and redirect to the main page(logged-out state)
9) GroupPage
   - <CreateGroup>: contain creating new group form
   - <SearchingGroup>: contain searching group form
   - <MyGroupList>: contain my group list
10) CreateGroup
   - onSubmit: submit grouptype and groupname => if groupname does not exist already, call backend creating group api and redirect to the new group detail page
11) SearchingGroup
   - <GroupList>: contain filtered group list by a searchword
   - onSubmit: filter groups with a searchword and call backend searching api
12) GroupList
   - onClickGroup: call backend redirecting to the group detail page api and redirect to the group detail page
   - onClickWithdrawGroup: call backend withdrawing from group api
   - onClickJoinGroup: call backend joining group api and redirect to the group detail page
13) MyGroupList
   - <MyGroup>: contain my group data form
   - withdrawGroupCheck: after checking whether the user really wants to withdraw, call backend withdrawing group api and redirect to the group page
14) MyGroup
   - onClickGroup: call backend redirecting to the group detail page api and redirect to the group detail page

- onClickWithdrawGroup: call backend withdrawing from group api
- onClickJoinGroup: call backend joining group api and redirect to the group detail page

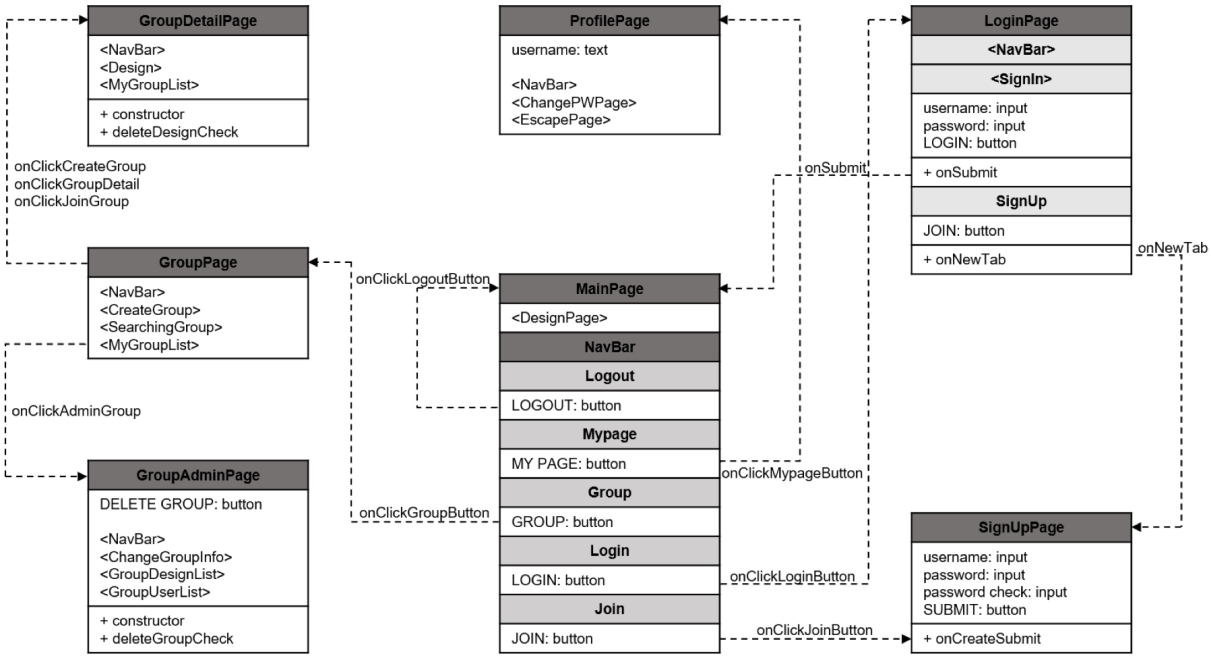
15) GroupAdminPage
    - <ChangeGroupInfo>: contain changing group information form
    - <GroupDesignList>: contain list of designs which are belonging to the group
    - <GroupUserList>: contain list of users which are belonging to the group
    - deleteGroupCheck: after checking whether the user really wants to delete the group, call backend deleting group api and redirect to the group page
16) ChangeGroupInfo
    - onSubmit: submit new grouptype and new groupname => if the new groupname doesn't exist already, call backend changing group information api
17) GroupDesignList
    - deleteDesignCheck: after checking whether the administrator really wants to delete the design, call backend deleting design api
18) GroupUserList
    - deleteUserCheck: after checking whether the administrator really wants to delete the user, call backend deleting user api
    - giveAdminCheck: after checking whether the administrator really wants to give an admin authority to the user, call backend giving admin authority api
19) GroupDetailPage
    - <DesignForm>: contains design images and comment form
    - deleteDesignCheck: after checking whether the user really wants to delete the de-sign, call backend deleting design api
20) DesignForm
    - <CommentForm>: contain adding comment form
    - <CommentList>: contain comment list
    - onClickEdit: call backend editing design api and redirect to the main page with the design
    - onClickPost: call backend posting design api and redirect to the group detail page
    - onClickLike: call backend liking design api
    - onClickUnlike: call backend unliking design api
    - onClickDelete: call backend deleting design api
21) CommentForm
    - onSubmit: submit name and message => call backend adding comment api
22) CommentList
    - <Comment>: contain comment content and buttons
23) Comment
    - deleteCommentCheck: after checking whether the user really wants to delete the comment, call backend deleting comment api
    - onClickEditComment: change to the edit mode

•onClickCompleteEditComment: change to the read mode and call backend editing comment api
•editModeRender: render comment in edit mode
•readModeRender: render comment in read mode

24) DesignPage
•onClickSave: make image url of canvas and call backend saving design api

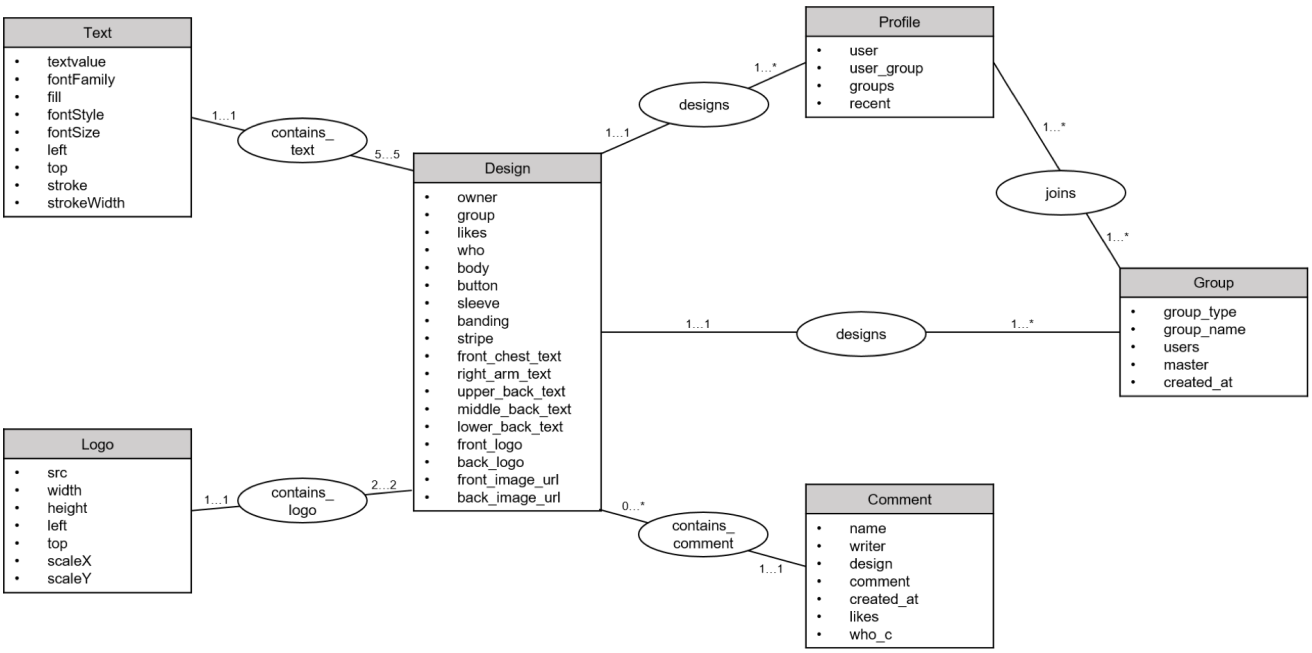
2. Frontend Components Relationship

Here are the relations between frontend components.



3. Backend Model

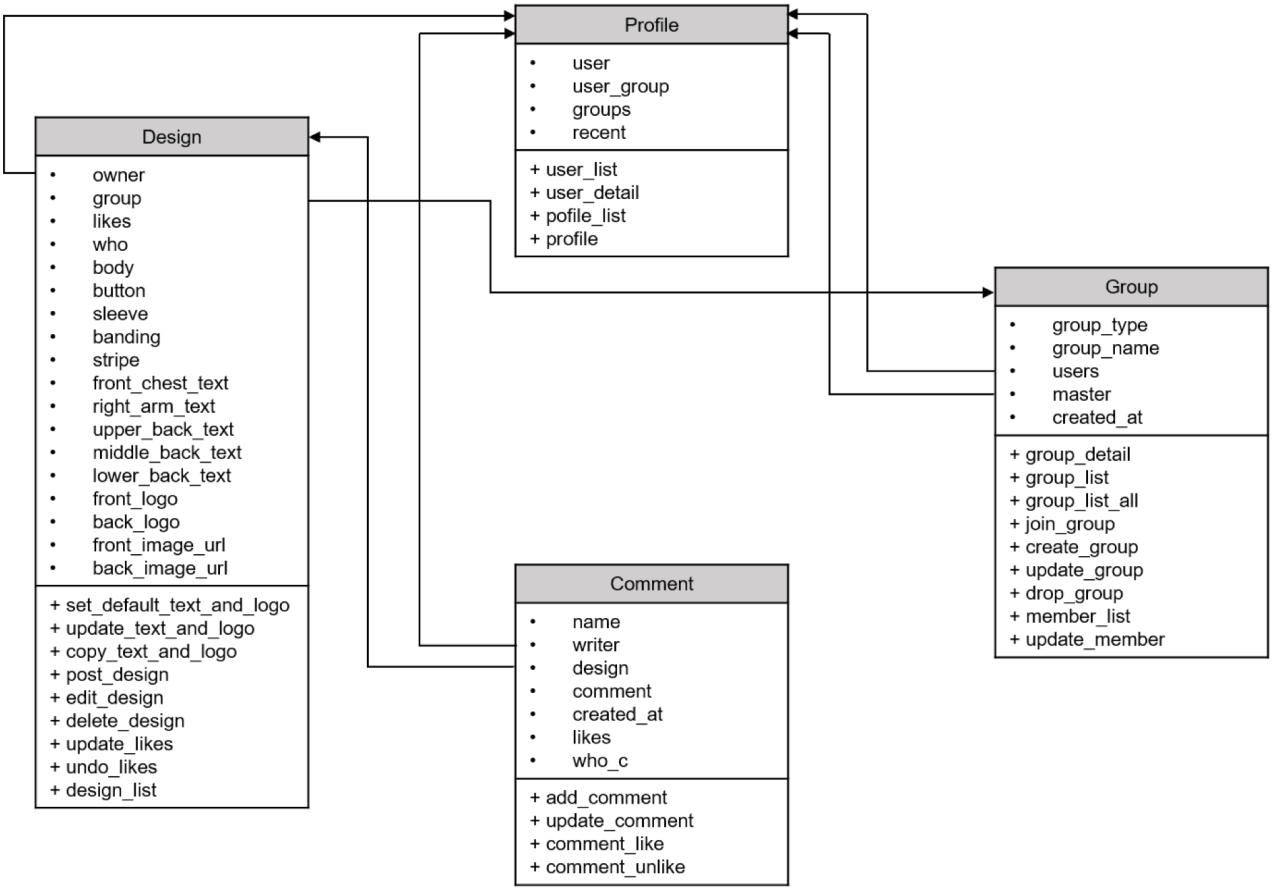
Here is E-R (Entity Relationship) diagram for model design. Rectangle stands for entity set, oval stands for relationship set, and numbers next to line means mapping cardinality constraints (min…max). Entity attributes are listed inside entity rectangle.



4) Backend Model Relationship

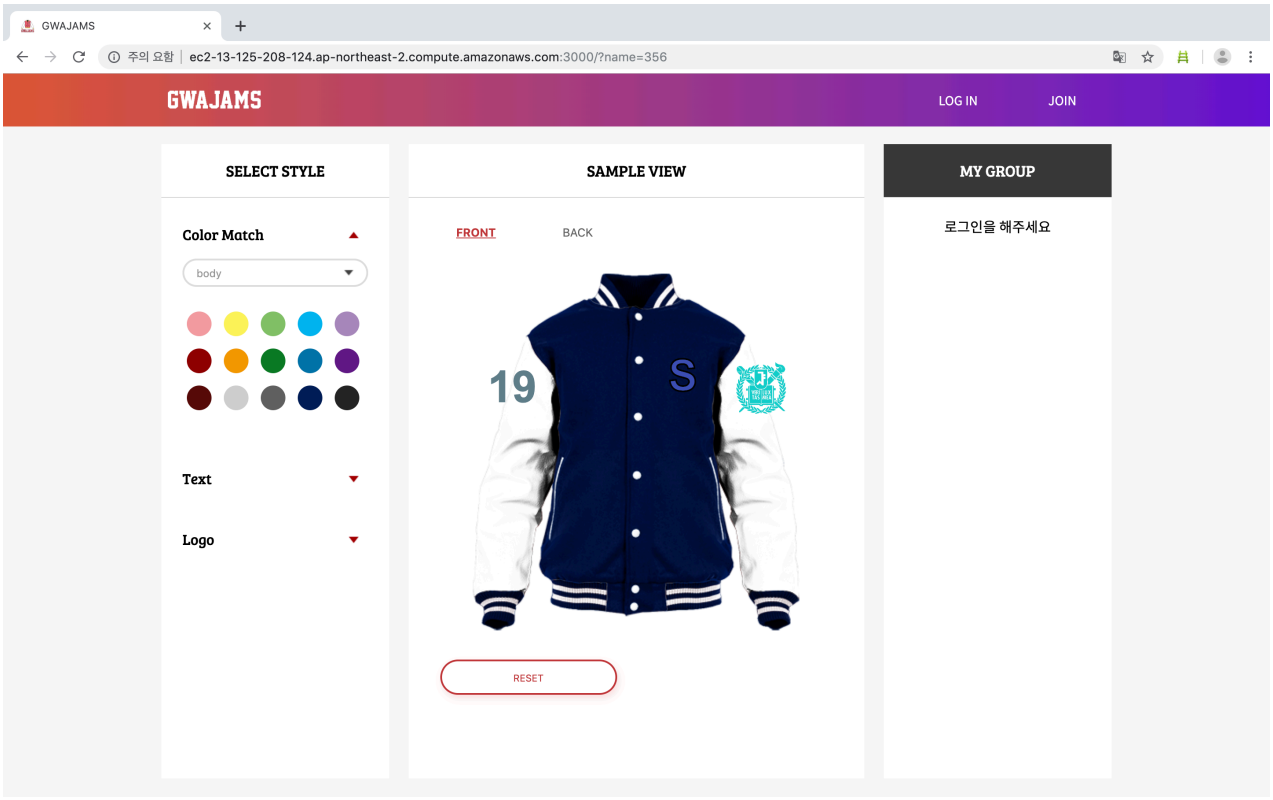
This is a relation schema diagram based on E-R diagram. This is also the structure of Django models the service is going to use. It also contains the core methods of each model.

Rectangle stands for relation schema. Schema attributes and methods for Django models are listed inside the rectangle. Arrow represents foreign key constraints.

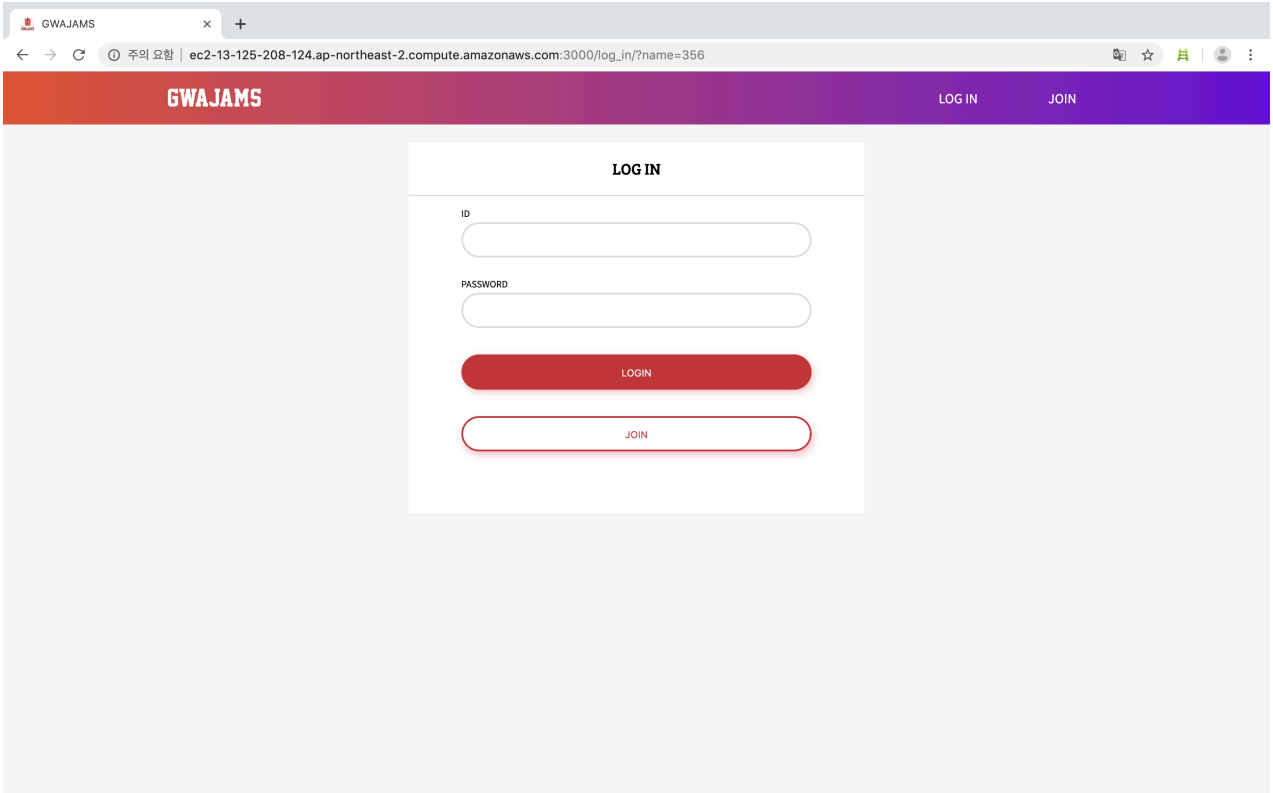


UI/UX

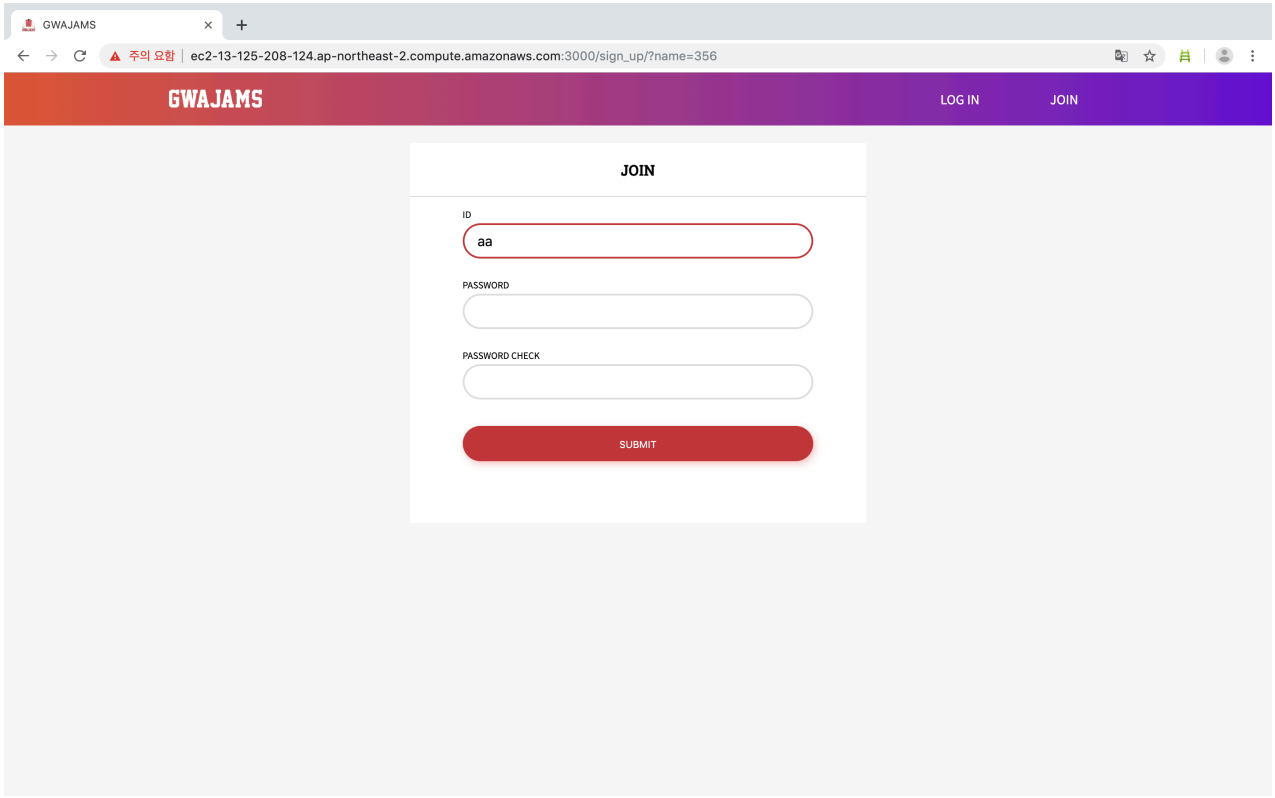
MAIN (/)



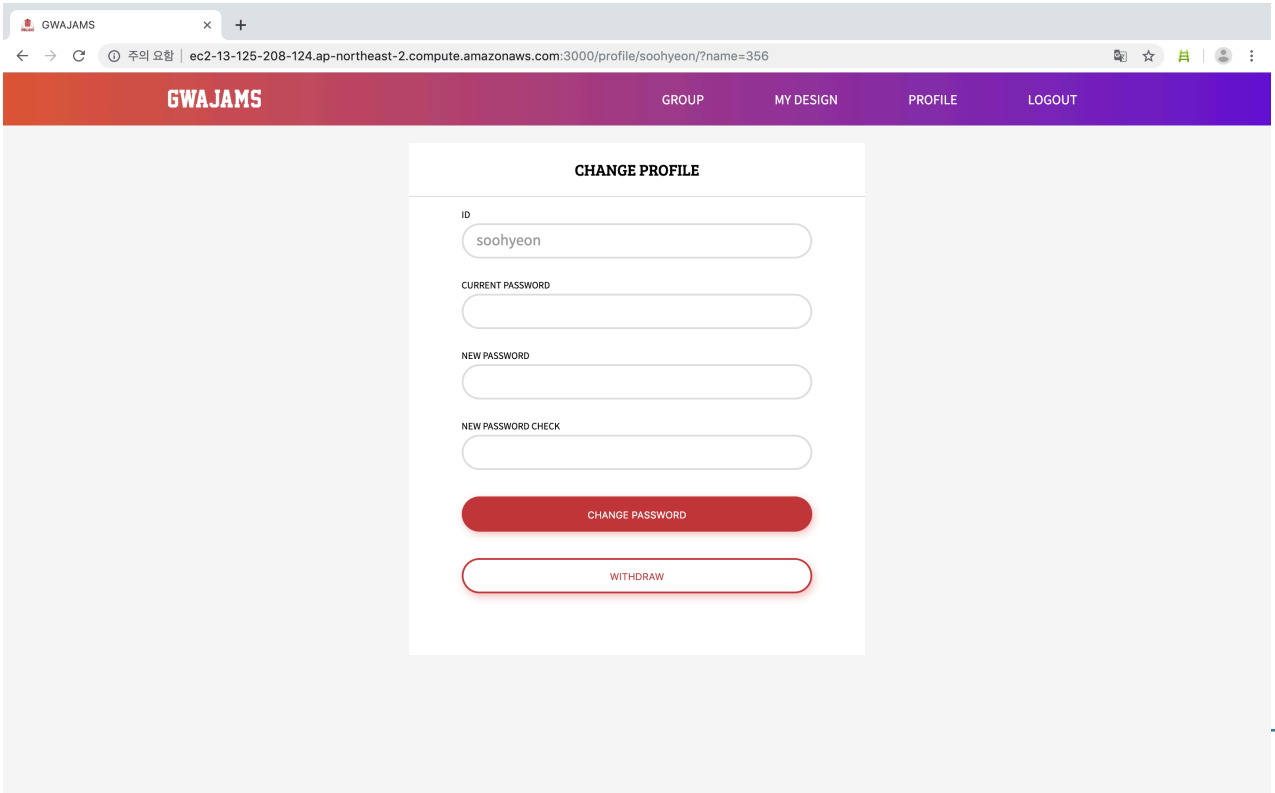
LOGIN(/log_in)



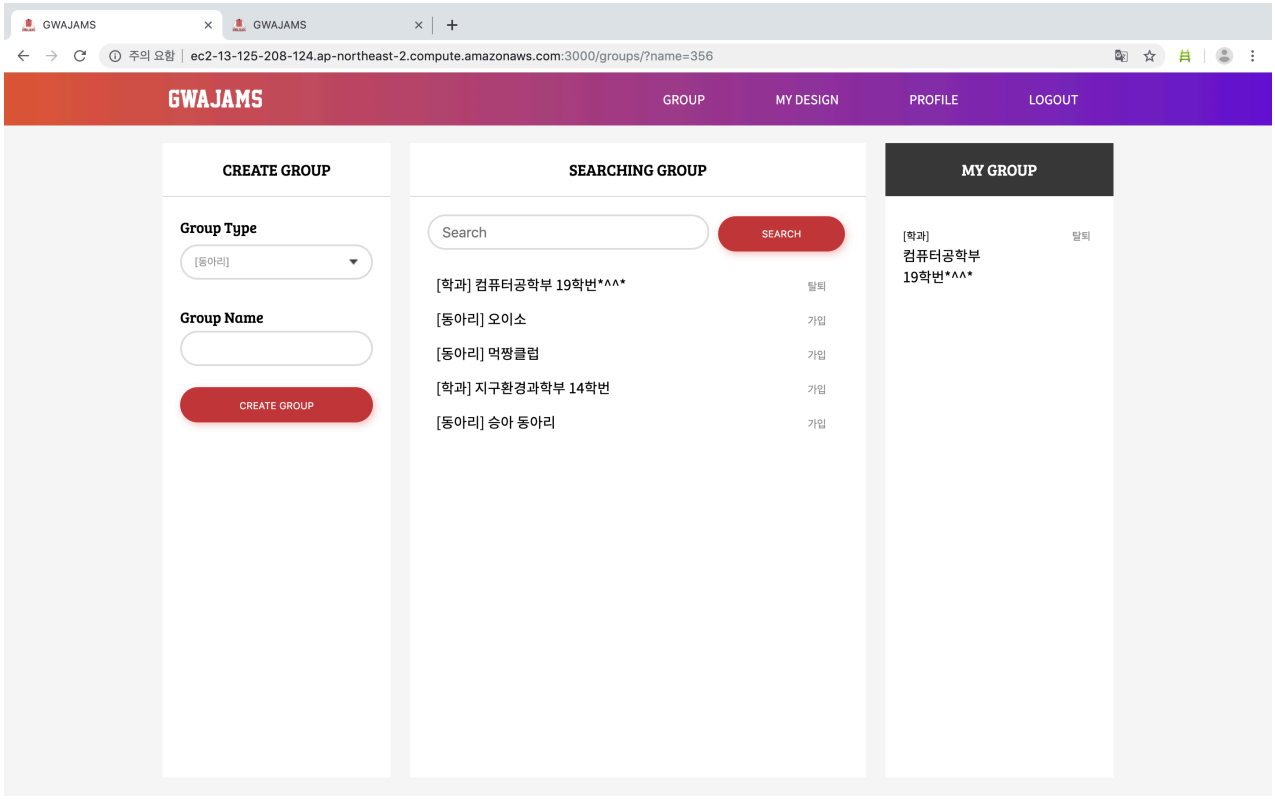
JOIN(/sign_up)



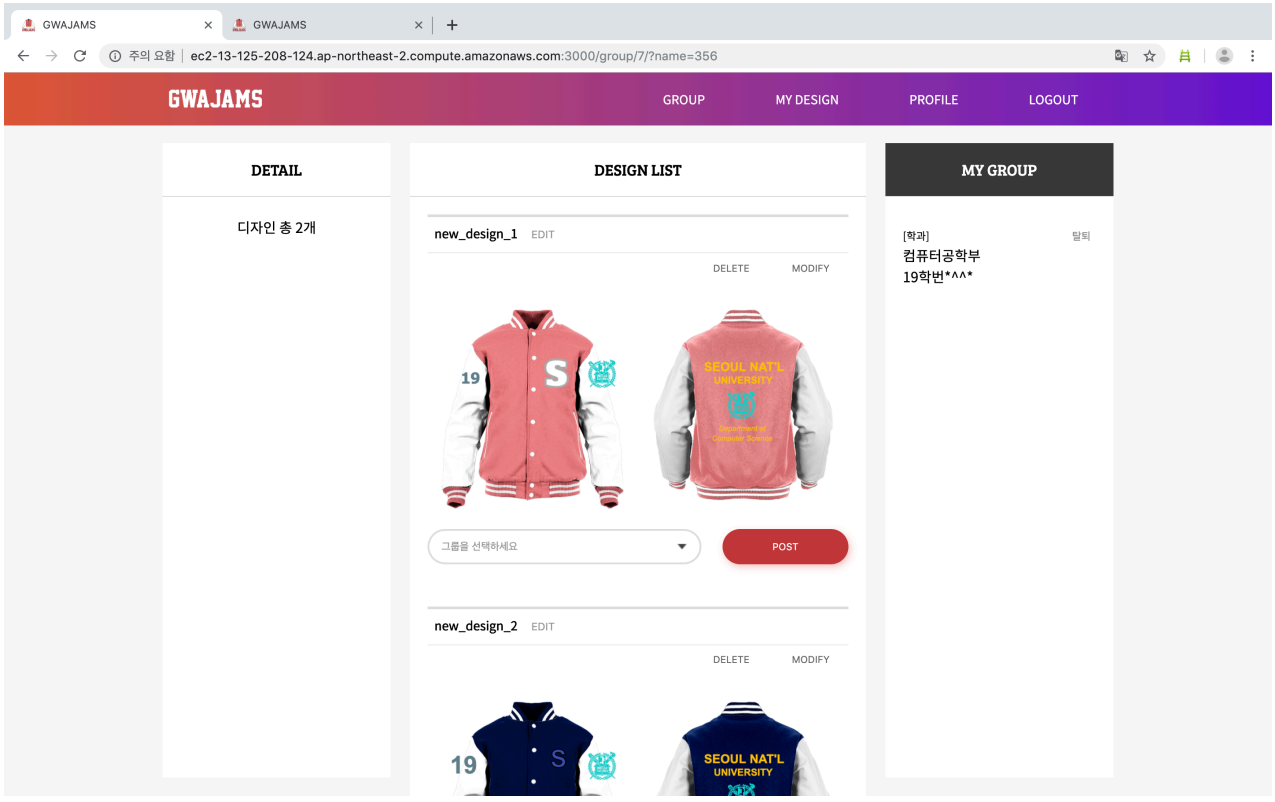
MYPAGE(/profile)



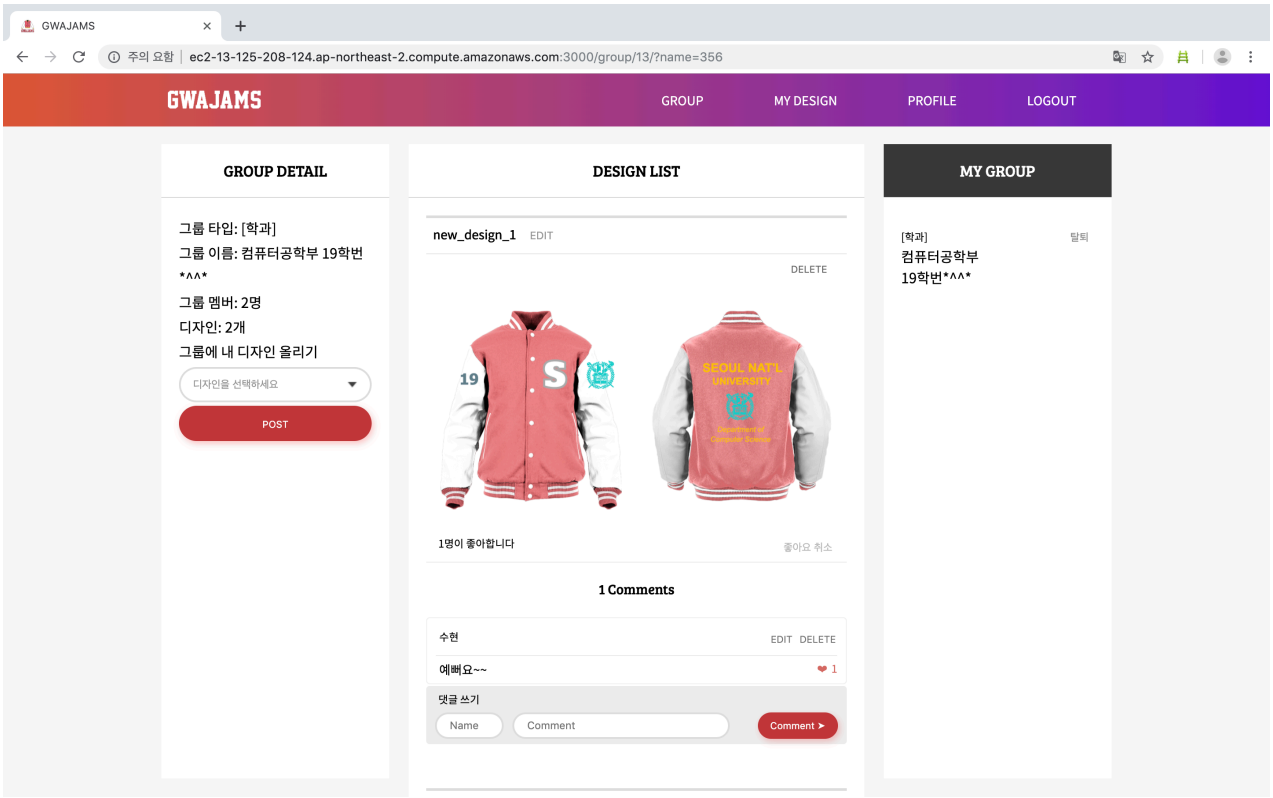
GROUP(/groups)



GROUP DETAIL – ‘User Group’ (/group/:groupid)



GROUP DETAIL – ‘동아리’/’학과’ (/group/:groupid)



Demo & How To Start

Demo: http://ec2-13-125-208-124.ap-northeast-2.compute.amazonaws.com:3000/

Getting Started

[Frontend] (https://github.com/swapp201901-team9/frontend)

0. Install NodeJs
1. Fork Frontend repository on Github
2. Clone your fork to your local machine

```
git clone git@github.com:<yourname>/frontend.git
```

3. Go to the project root directory

```
cd frontend
```

4. erase all unnecessary data

```
rm package-lock.json
rm -r node_modules
```

4. install node dependencies

```
sudo npm install
```

*do not "sudo npm audit fix" 5) start

```
sudo npm start
```

[Backend] (https://github.com/swapp201901-team9/backend)

0)install pip3, python3 *for linux

```
- sudo apt-get update
- sudo apt-get install python3.5
- sudo apt-get -y install python3-pip python-dev
- sudo python3 -m pip install --upgrade pip
- sudo pip3 install -U setuptools
- python3 -V
- pip3 -V
```

1. Fork Backend repository on Github
2. Clone your fork to your local machine

```
git clone git@github.com:<yourname>/backend.git
```

3. Go to the project root directory

```
cd backend
```

4. erase all unnecessary data and migrate

```
rm -r homepage/migrations
python3 manage.py makemigrations homepage
python3 manage.py migrate
```

4. install node dependencies

```
sudo pip3 install -r requirements.txt
```

5. start

```
sudo python3 manage.py runserver
```

Conclusion

It was both exciting and hard to carry out a project all throughout the semester. Not only did we gain skills in handling web frameworks, but also we learned a lot about collaboration in computer science.

We met every tuesdays and thursdays, and when those times weren't enough, we also met on the weekends. The project definitely has room for improvment, but for now, we present you with the final result of our hard work and dedication!

Reference

fabricjs https://github.com/fabricjs/fabric.js/tree/master

react-tabs-redux https://github.com/patrik-piskay/react-tabs-redux

react-color https://github.com/casesandberg/react-color