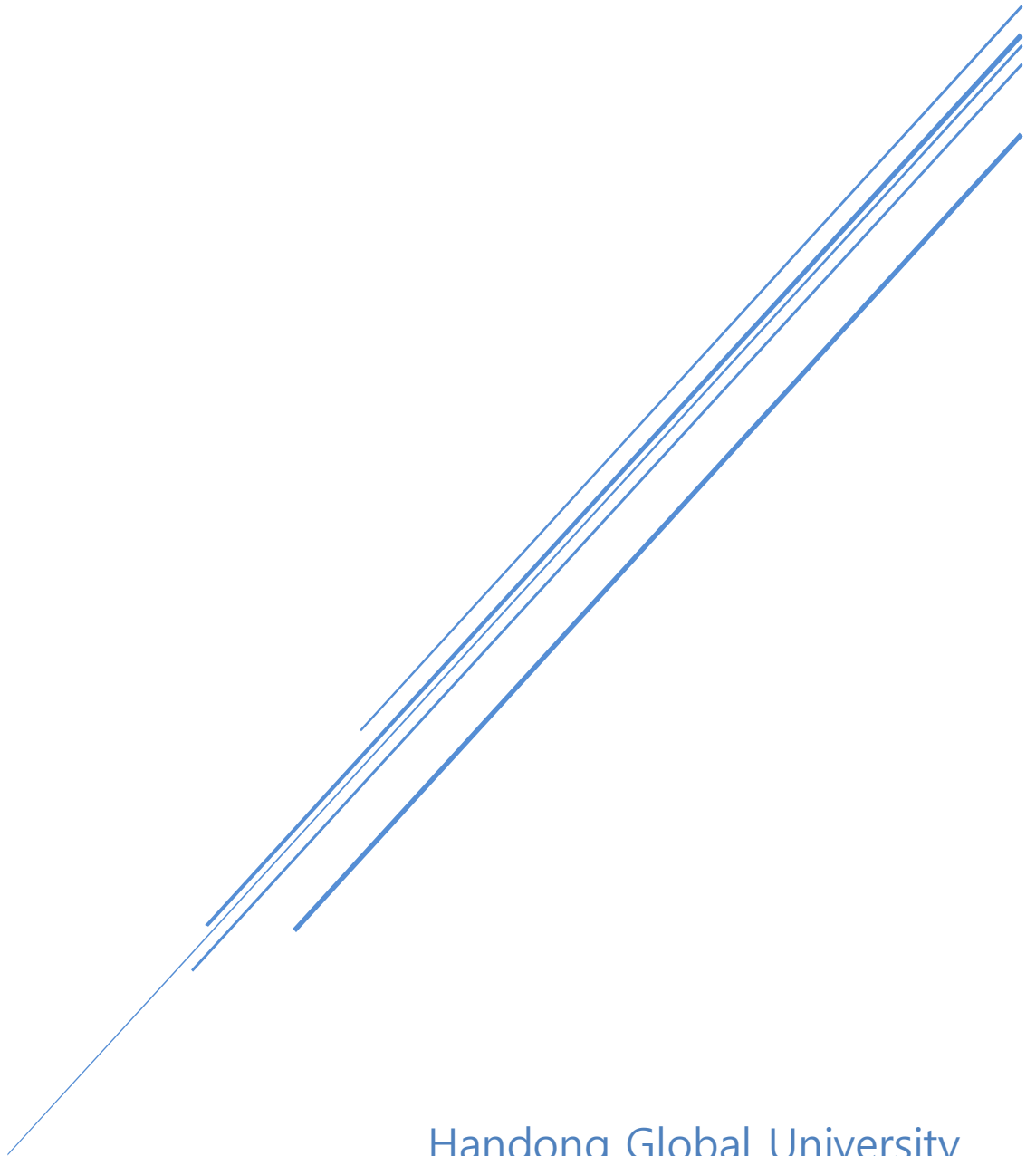


HISTUDY DOCUMENTATION

Group 6



Handong Global University
Software Engineering

Contents

1. Project Set-up
2. Dependency Information
 - 1) Server requirement
 - 2) Local requirement
3. System architecture and design
 - 1) System Model
 - 2) System Architecture
4. Structure of Database
 - 1) Photos
 - 2) Account & Authentication
 - 3) Social Account
 - 4) Django
5. Group matching algorithm
6. System user Manual
 - 1) Administrator
 - 2) User

1. Project Set-up

- original link : <https://github.com/dodoyoon/Histudy/blob/master/guide.md>

서버는 우분투 18.04 환경이다.

Step 1. 가상환경 설정

GCP, NCP 등등을 통해서 ubuntu 18.04 서버를 만든다.

서버를 업데이트하고 파이썬과 pip를 다운받는다.

```
sudo apt update
```

```
sudo apt install python-pip
```

```
sudo apt install python3-pip
```

아래 명령어를 통해서 버전을 확인한다.

```
python3 -V
```

```
# ex) 3.6.9
```

```
# 3.6버전이 아니라면
```

```
sudo add-apt-repository ppa:deadsnakes/ppa ₩
```

```
&& sudo apt update
```

밑에 있는 블로그를 참고해서 python3의 버전을 3.6로 변경하자.

[\[Django\] Python 3.5에서 Python 3.7로 업그레이드 하기\(mod_wsgi업그레이드\) \(Ubuntu\)](#)

가상환경을 만들기 위해 필요한 virtualenvwrapper 를 받는다.

```
sudo pip3 install virtualenvwrapper
```

.bashrc 파일을 sudo vi ~/.bashrc로 열고, 아래 내용을 파일 제일 밑에 추가한다.

```
export WORKON_HOME=$HOME/.virtualenvs
```

```
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
```

```
export PROJECT_HOME=$HOME/Devel
```

```
source /usr/local/bin/virtualenvwrapper.sh
```

그리고 새로운 가상환경을 만들기 전에 아래의 명령어를 실행해서 bash 파일을 실행한다.

```
source ~/.bashrc
```

원하는 이름으로 가상환경을 만들고 Histudy를 git clone 한다. 그리고 필요한 Python Module를 가상환경에 다운받는다.

```
mkvirtualenv histudy
```

```
# histutor 파이썬 가상환경으로 activate되어있는 상태
```

```
git clone https://github.com/dodoyoon/Histudy.git
```

```
chmod -R 777 ~/Histudy
```

```
cd ~/Histudy
```

```
pip3 install -r server_requirements.txt
```

- 참고 : 가상환경 activate / deactivate 하는 법
 - activate : source ~/.virtualenvs/histudy/bin/activate
 - deactivate : deactivate

Step 2. HisSecret 설정

HisSecret은 Django에 사용되는 비밀번호 같은 정보가 github로 노출되지 않게 비밀번호를 저장해놓은 github private repository다. 이 HisSecret은 개개인이 만들기를 권장한다. 형식은 다음과 같다.

django secret key는 이 사이트에서 생성할 수 있다.

Djecrety

아래는 우리가 사용한 HisSecret에 있는 secret.json 파일이다.

```
{  
  
    "DJANGO_SECRET_KEY": "장고시크릿키",  
  
    "DB_PASSWORD": "디비비번"  
}
```

이제 Histudy/pystagram/settings.py에서 SECRET_BASE 부분을 아래와 같이 절대경로로 바꿔준다.

```
SECRET_BASE = '/home/g21300109/HisSecret'
```

이런 방식이 싫다면 서버의 Histudy/pystagram/settings.py에서 하드코딩해도 된다.

Step 3. Mysql 설치 및 연동

MySQL을 서버에 설치한다.

```
sudo apt install mysql-server
```

- 참고: MySQL 설치 후 최초 비밀번호를 요구한다면 아래와 같이
~/Histudy/pystagram/settings.py에 저장되어 있는 DB 비밀번호와 같이 입력한다.

기본값으로 설정되어 있는 root의 비밀번호를 바꾸기 위해서 mysql을 실행시키고 아래의 명령어를 입력한다.

```
sudo mysql
```

```
alter user 'root'@'localhost' identified with mysql_native_password by 'password';
```

설정한 password는 ~/Histudy/pystagram/settings.py에 있는 DB 비밀번호에 입력한다.

MySQL이 한글로 된 데이터를 저장할 수 있도록 default character set을 변경해야 한다

sudo vi /etc/mysql/my.cnf 로 파일을 열고, 파일 끝에 아래 내용을 추가한다.

```
[client]
```

```
default-character-set=utf8
```

```
[mysql]
```

```
default-character-set=utf8
```

```
[mysqld]
```

```
collation-server = utf8_unicode_ci
```

```
init-connect='SET NAMES utf8'
```

```
character-set-server = utf8
```

sudo service mysql restart로 mysql을 재시작한다.

MySQL을 다시 실행한 뒤 utf가 제대로 변경되었는지 확인한다.

```
mysql -u root -p
```

```
status
```

'study'라는 이름의 DB 생성한다. 꼭 character set을 변경한 후에 생성해야지 데이터베이스의 character set도 utf로 설정된다.

```
CREATE DATABASE study;
```

에러 수정

<https://dodormitory.tistory.com/8> 링크로 가서 4 - 에러 수정 부분을 참고하여 고치자

테이블 생성하기

[manage.py](#) 파일이 있는 디렉토리로 이동하고, migrate 명령어로 테이블을 생성한다.

```
cd ~/Histudy
```

```
python3 manage.py makemigrations
```

```
python3 manage.py migrate
```

static file을 .static_root 디렉토리에 모으는 명령어

```
python3 manage.py collectstatic
```

mysql에서 study의 table을 보면 정상적으로 table들이 생성된 것을 볼 수 있다.

Step 4. Apache 설치 및 연동

deactivate 로 가상환경에서 빠져나온다.

apache와 wsgi 모듈인 libapache2-mod-wsgi, 파이썬 연결 모듈 libapache2-mod-wsgi-py3를 설치한다.

```
sudo apt-get install apache2 # apache2 설치
```

```
sudo apt-get install libapache2-mod-wsgi # wsgi 모듈
```

```
sudo apt-get install libapache2-mod-wsgi-py3
```

sudo vim /etc/apache2/sites-available/000-default.conf 를 통해서 파일을 열고 아래처럼 설정한다.

- 설정 파일 가이드라인

```
<VirtualHost *:80>
```

```
ServerAdmin webmaster@localhost
```

```
DocumentRoot /var/www/html
```

```
ErrorLog ${APACHE_LOG_DIR}/error.log
```

```
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

<Directory {wsgi.py가 있는 디렉토리 주소}>

<Files wsgi.py>

Require all granted

</Files>

</Directory>

Alias {settings.py에 STATIC_URL 변수 값} {settings.py에 STATIC_ROOT 디렉토리의 절대주소}

<Directory {settings.py에 STATIC_ROOT 디렉토리의 절대주소}>

Require all granted

</Directory>

WSGIDaemonProcess tutor python-path={manage.py가 있는 디렉토리의 절대주소} python-home={이 프로젝트를 돌릴 때에 사용하는 virtual environment의 절대주소}

WSGIProcessGroup {프로젝트이름}

WSGIScriptAlias / {wsgi.py가 있는 디렉토리의 주소/wsgi.py}

</VirtualHost>

히즈튜터에 맞는 설정

<VirtualHost *:80>

The ServerName directive sets the request scheme, hostname and port that

the server uses to identify itself. This is used when creating

redirection URLs. In the context of virtual hosts, the ServerName

specifies what hostname must appear in the request's Host: header to

match this virtual host. For the default virtual host (this file) this

value is not decisive as it is used as a last resort host regardless.

However, you must set it for any further virtual host explicitly.

#ServerName www.example.com

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

Available loglevels: trace8, ..., trace1, debug, info, notice, warn,

error, crit, alert, emerg.

It is also possible to configure the loglevel for particular

modules, e.g.

#LogLevel info ssl:warn

ErrorLog \${APACHE_LOG_DIR}/error.log

CustomLog \${APACHE_LOG_DIR}/access.log combined

<Directory /home/g21300109/Histudy/pystagram>

<Files wsgi.py>

Require all granted

</Files>

</Directory>

Static file(js, css 등등)이 들어있는 폴더에 Apache가 접근하게 함

Alias /static /home/g21300109/Histudy/staticfiles

<Directory /home/g21300109/Histudy/staticfiles>

Require all granted

</Directory>

WSGIDaemonProcess histudy python-path=/home/g21300109/Histudy python-home=/home/g21300109/.virtualenvs/histudy

WSGIProcessGroup histudy

WSGIScriptAlias / /home/g21300109/Histudy/pystagram/wsgi.py

</VirtualHost>

이후 가상환경을 다시 작동한다.

source ~/.virtualenvs/{가상환경이름}/bin/activate

파이썬 모듈인 uwsgi를 설치한다.

pip install uwsgi

uwsgi 가 설치되지 않는다면 아래 블로그를 참고하자

[pip3 install uwsgi 설치 에러 Failed building wheel for uwsgi](#)

이제 django 사용 포트를 열어야 한다.

먼저 ufw로 방화벽에서 해당 포트를 개방한다. iptables의 해당 포트를 개방한다. 마지막으로 서버를 실행시켜서 해당 포트가 열린 것을 확인한다.

sudo ufw allow 8000

```
sudo iptables -I INPUT -p tcp --dport 8000 -j ACCEPT
```

```
python manage.py runserver 0.0.0.0:8000
```

Server_IP_Address:8000으로 접속하면 Histudy가 떠야 정상이다.

sudo vi /etc/apache2/ports.conf 로 /etc/apache2/ports.conf 파일을 열고, 위에서 열거된
포트를 추가한다. Listen 80 밑에 Listen 8000을 추가하면 된다.

```
#Listen 추가포트
```

```
Listen 80
```

```
Listen 8000
```

sudo service apache2 restart 로 Apache를 재시작한다.

이제 Server_Ip_Address:8000으로 접속하면 histutor가 성공적으로 보이는 것을 볼 수 있다.

Step 5. Google Login을 위한 Social App 등록하기

[\[Django\] Google 계정으로 로그인하기 \(로컬 서버 + 실제 서버\)](#)

위 블로그 포스트를 참고하면 된다. Google OAuth를 사용해서 도메인 주소를 등록하고 Django에서
Google Social Application을 활성화한다.

Step 6. Let's Encrypt로 https 설정하기

[Let's Encrypt를 사용하여 HTTPS 설정하기](#)

위 블로그 포스트를 참고하자.

https설정을 마친 후 Step4의 블로그 포스트를 참고하여 https로 시작하는 도메인도 추가해줘야
한다.

Step 7. 현재 연도와 학기 설정하기

Histudy를 사용하기 위해선 현재 연도와 학기를 설정해주어야 한다. 이를 위해선 관리자로 로그인할
필요가 있는데, 관리자를 만드는 방법은 다음과 같다.

```
cd ~/Histudy/ # manage.py가 있는 디렉토리로 이동
```

python3 manage.py createsuperuser

https://{histudy ip address 또는 domain name}/admin 으로 접속하면 관리자 페이지가 나온다.
관리자로 로그인 후, https://www.histudy.cafe24.com/set_current (이번 년도 & 학기 지정하기)로
가서 현재 년도와 학기를 지정해준다.

개발 팁

1. Super User 생성하기

장고의 관리자 계정을 생성하기 위해서는

1. 먼저 가상환경을 켜고: `source ~/.virtualenvs/histudy/bin/activate`
2. manage.py 파일이 있는 디렉토리로 이동한다. : `cd ~/Histudy`
3. Super User를 생성한다. : `python3 manage.py createsuperuser`

이 과정을 거치면 Super User가 생성되고 장고 관리자 페이지(주소: Histutor_domain/admin)로 접속할 수 있게 된다.

2. 더 자세한 에러메시지 보기

서버에서 문제가 생기면 Server Internal Error만 달랑 떠서 문제의 원인을 정확히 알 수 없다.

그럴 때, ~/Histudy/pystagram/settings.py 파일에서 'DEBUG' 라는 변수를 True로 변경하면 됩니다.

하지만 실제 서비스에서는 보안상의 이유로 DEBUG는 False 이어야 합니다.

따라서 에러를 고친 다음에는 DEBUG를 False로 변경해 주시기 바랍니다.

3. Static File을 수정하거나 추가한 경우

장고에서는 Static File(css, js) 들을 한 곳에 모아두고 사용한다.

그래서 수정되거나 추가된 Static File들은 해당 디렉토리에 추가가 되어야 한다.

이를 추가하기 위한 과정은 다음과 같다.

1. 먼저 가상환경을 켜고: `source ~/.virtualenvs/histudy/bin/activate`
2. manage.py 파일이 있는 디렉토리로 이동한다. : `cd ~/Histudy`
3. collectstatic 명령어를 실행한다. : `python3 manage.py collectstatic`

그러면 'This will overwrite existing files!' 와 같은 경고문이 뜨는데 그냥 yes를 치면 된다.

4. 서버에서 자주 사용하는 명령어

Apache 관련

1. 에러로그파일 위치: /var/log/apache2/error.log

2. 아파치 Config파일 위치: /etc/apache2/sites-available/000-default-le-ssl.conf

3. 아파치 재시작: sudo service apache2 restart

Reference

<https://dodormitory.tistory.com/>

<http://wanochoi.com/?p=3575>

<https://calvinjmkim.tistory.com/23>

2. Dependency Information

1) Server requirement

Name	Version	Name	Version
asgiref	3.3.1	odfpy	1.4.1
async-timeout	3.0.1	openpyxl	3.0.3
attrs	19.3.0	pandas	1.1.2
autobahn	20.12.3	Pillow	8.1.1
Automat	0.8.0	psycpg2-binary	2.8.4
certifi	2019.11.28	pyasn1	0.4.8
cffi	1.13.2	pyasn1-modules	0.2.8
chardet	3.0.4	pycparser	2.19
constantly	15.1.0	pyexiv2	1.3.0
cryptographyp	3.3.2	PyHamcrest	2.0.0
defusedxml	0.6.0	PyJWT	1.7.1
diff-match-patch	20181111	PyMySQL	0.9.3
django-database-url	0.5.0	pyOpenSSL	19.1.0
Django	3.1.8	python-dateutil	2.8.1
django-allauth	0.42.0	python-magic	0.4.18
django-ckeditor	5.9.0	python-memcached	1.59
django-import-export	2.0.1	python3-openid	3.1.0
django-js-asset	1.2.2	pytz	2019.3
django-sslserver	0.22	PyYAML	5.4
django-summernote	0.8.11.6	requests	2.22.0
django-user-agents	0.4.0	requests-oauthlib	1.3.0
et-xmlfile	1.0.1	service-identity	18.1.0
unicorn	20.0.4	six	1.13.0
hiredis	1.0.1	sqlparse	0.3.0
hyperlink	19.0.0	tablib	0.14.0
idna	2.8	txaio	20.1.1
incremental	17.5.0	ua-parser	0.10.0
jdcal	1.4.1	urllib3	1.26.5
MarkupPy	1.14	user-agents	2.1
msgpack	0.6.2	xlrd	1.2.0
numpy	1.19.2	xlwt	1.3.0
oauthlib	3.1.0	zope.interface	4.7.1

2) Local requirement

Name	Version	Name	Version
asgiref	3.3.1	pandas	1.2.0
certifi	2020.12.5	Pillow	8.1.1
cffi	1.14.4	pycparser	2.20
chardet	4.0.0	PyJWT	2.0.0
cryptography	3.3.2	PyMySQL	0.10.1
defusedxml	0.6.0	python-dateutil	2.8.1
diff-match-patch	20200713	python-magic-bin	0.4.14
Django	3.1.8	python-memcached	1.59
django-allauth	0.44.0	python3-openid	3.2.0
django-import-export	2.4.0	pytz	2020.5
django-summernote	0.8.11.6	PyYAML	5.4
django-user-agents	0.4.0	requests	2.25.1
et-xmlfile	1.0.1	requests-oauthlib	1.3.0
idna	2.10	six	1.15.0
install	1.3.4	sqlparse	0.4.1
jdcal	1.4.1	tablib	3.0.0
libmagic	1.0	ua-parser	0.10.0
MarkupPy	1.14	urllib3	1.26.5
numpy	1.19.4	user-agents	2.2.0
oauthlib	3.1.0	xlrd	2.0.1
odfpy	1.4.1	xlwt	1.3.0
openpyxl	3.0.5		

You can check the dependencies of this project through the links below.

https://github.com/dodoyoon/Histudy/blob/master/server_requirements.txt

https://github.com/dodoyoon/Histudy/blob/master/local_requirements.txt

<https://github.com/dodoyoon/Histudy/blob/master/package-lock.json>

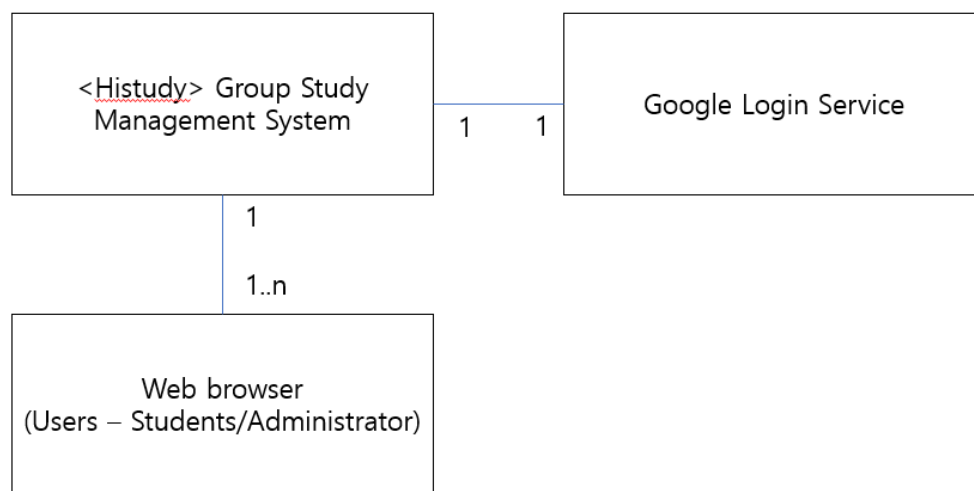
Also, all the dependency information can be checked through the dependency graph provided by GitHub. Here is the link.

<https://github.com/dodoyoon/Histudy/network/dependencies>

3. System architecture and design

1) System Model

- Context Model

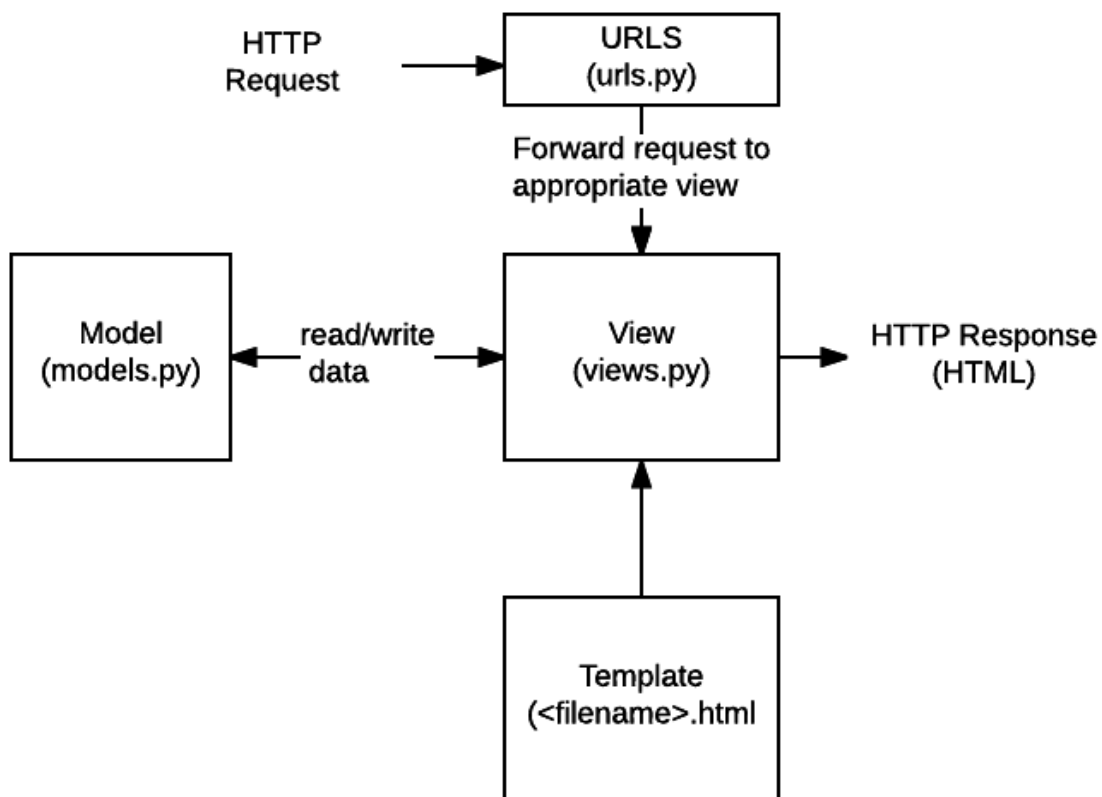


- <Histudy> web-application makes use of Google log-in service api for the external system as a log-in service. <Histudy> web-application acts as a host for recording group study accessed by many users via browser. The users include students who want to register group study or who are already in the study group and also the administrator(professor) who can manage the whole system.

2) System Architecture

- MVC Pattern

- The <Histudy> System is based on Django Framework. Django Framework is based on MVC pattern, where M stands for Model, V for View, C for controller. Model is all about the database input/output which is delivered through View. View, as the word implies does all the work regarding the 'view'. For example, if there is a chart on the web page, the view is responsible to receive data from the model and show that data to the user in a certain way. The specific designs or templates to show a certain view is configured in a separate files inside 'template' folder. Controller is all about the given URL from the user. If the user gives a specific url, then the controller matches that url to a related view. With this mechanism, Model, View, and Controller can do their independent work enhancing the maintainability and functionality.



<Django's MVC pattern Histudy use>

- > node_modules/jquery
- > photos App 1
- > pystagram App 2
- > static
- > templates/registration
 - .gitignore
 - README.md
 - dump.rdb
 - guide.md
 - local_requirements.txt
 - manage.py
 - my.cnf
 - package-lock.json
 - server_requirements.txt
 - uwsgi.ini
 - uwsgi_params

- > node_modules/jquery
- photos
 - migrations
 - static/css Static folder 1
 - templates Template folder 1
 - _init_.py
 - admin.py
 - apps.py
 - forms.py Forms
 - models.py Models
 - tests.py
 - urls.py Urls 1
 - views.py View
- pystagram
 - _init_.py
 - asgi.py
 - settings.py
 - urls.py Urls 2
 - wsgi.py
- recruit
- static Static folder 1
- templates/registration Template folder 2
 - .gitignore
 - README.md
 - dump.rdb
 - guide.md

- photos
 - migrations
 - static/css
 - templates
 - announce.html
 - announce_content.html
 - announce_write.html
 - base_template.html
 - csv_upload.html
 - detail.html
 - edit.html
 - export_all_page.html
 - export_page.html
 - grid.html
 - img_download_page.html
 - inquiry.html
 - list.html
 - main.html
 - messages.html
 - new_userinfo.html
 - popup.html
 - rank.html
 - reset_profile_group.html
 - set_current.html
 - top3.html
 - upload.html
 - userlist.html
 - warn_overwrite.html

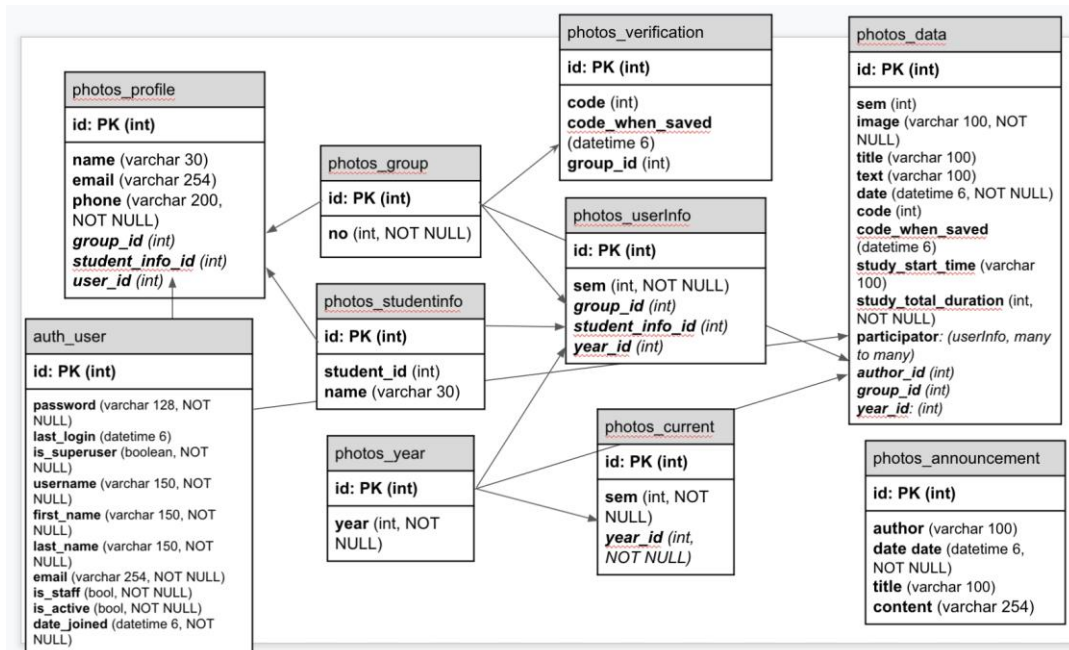
- templates/registration
 - change_password.html
 - create_userinfo.html
 - delete_userinfo.html
 - delete_userinfo_confirm.html
 - group_profile.html
 - histudy_guideline.html
 - login.html
 - member.html
 - no_group_notice.html
 - no_student_id.html
 - profile.html
 - save_profile.html
 - signup.html
 - staff_profile.html

Pages shown
to the user

<Histudy's file architecture>

4. Structure of Database

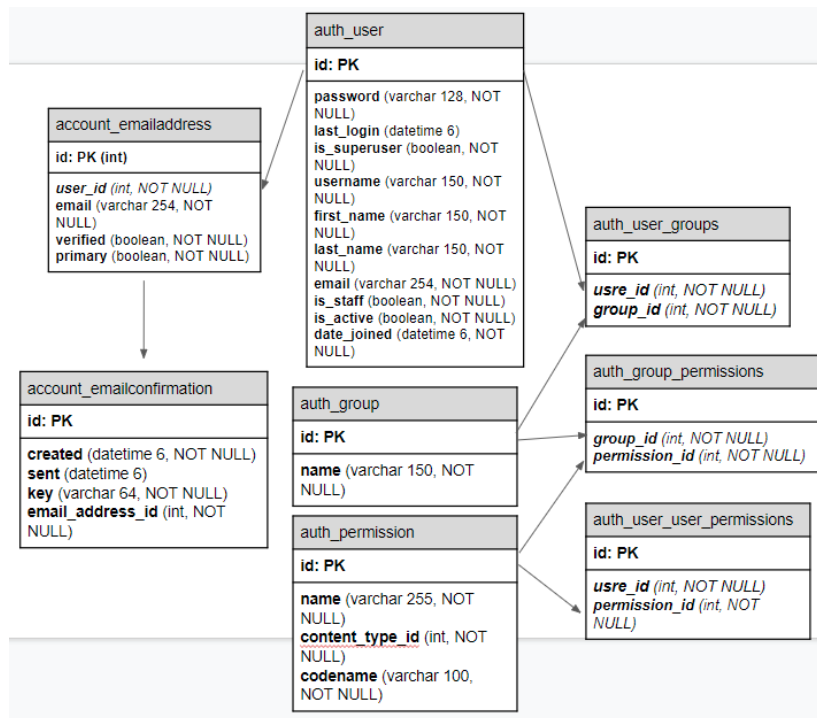
1) Photos



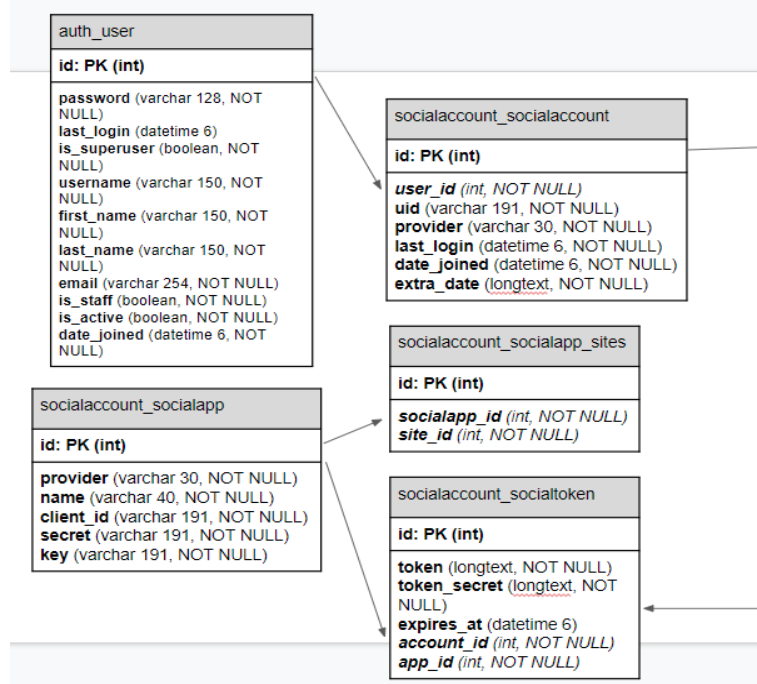
2) Account & Authentication

- (django.contrib.auth, API Reference)

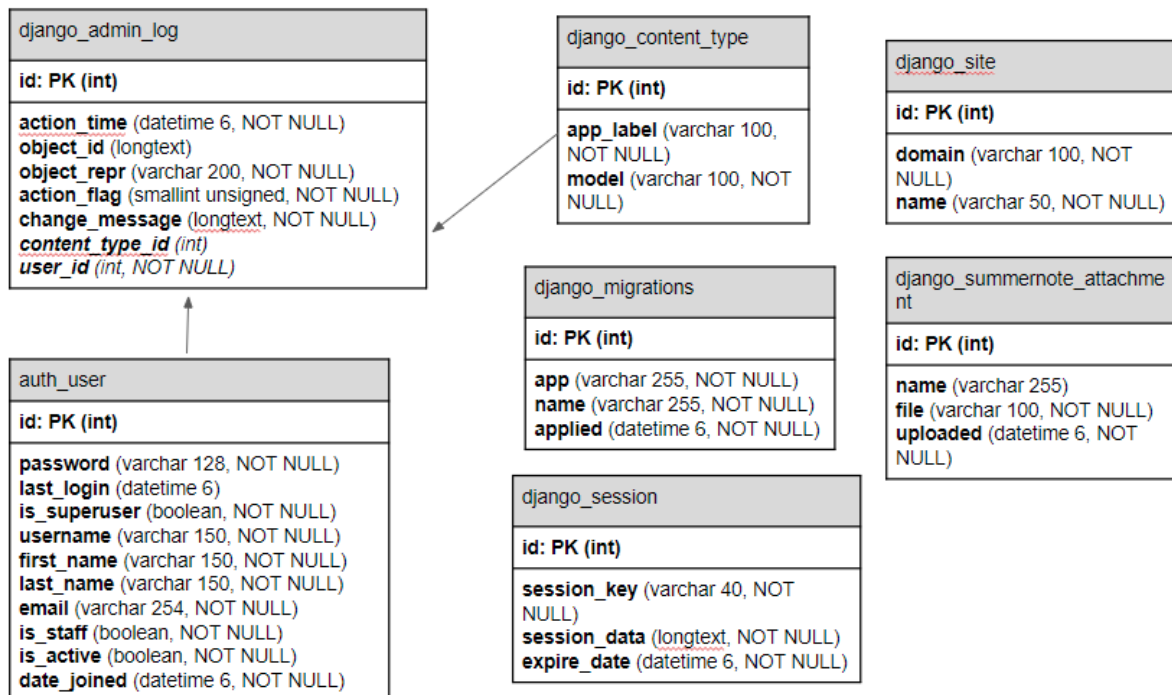
document reference: <https://docs.djangoproject.com/en/3.2/ref/contrib/auth/>



3) Social Account



4) Django



5. Group matching algorithm

Explanation: group number(#)

group # is 0: student information is okay but cannot be grouped

group # is less than 0: there's a problem while processing group matching

- group # is -1: disagreement to provide personal information
- group # is -2: failed to create friend group

group # is more than 0: normal group # (successfully grouped)

Steps of algorithm

1. read user uploaded .csv file

2. data refinement

2.1 remove whitespace, uncorrect data

2.2 assign group # -1 and remove students who do not agree with provision of personal information

3. make groups for whom listed friend name and code

3.1 get friends list data, convert it to set, make large sets by union-ing converted sets of the other studying together

3.2 if the number of friends is within range [3, 5] and there's no problem in

friends list(all members wrote friends list EXCEPT himself that matches with a big

- union-ed - set preprocess in 3-1)

3.2.1 assign group #

3.3 Otherwise(number of friends out of range or friends list mismatch what was preprocessed)

3.3.1 assign group num -2

4. gather people by 1st preferred subject of every student

4.1 get # of students grouped by 'subject code', 'professor' in order of

“대면만”, “비대면만”, “대면/비대면 상관없음” each

4.2 gather people within same subject code and professor and preference in order of

“대면만”, “비대면만”, “대면/비대면 상관없음” each

4.2.1 if the # of students is less than 2

4.2.1.1 add those students to 'cannot_grouped' list

5. gather the ungrouped with their 1st preferred subject

5.1 tie rest of “대면만” and “대면/비대면 상관없음” students

5.1.1 repeat 'step 4'

5.2 tie rest of “비대면만” and “대면/비대면 상관없음” students

5.2.1 repeat 'step 4'

6. gather the still ungrouped people to other gathered sets

6.1 search for group that matches 1st preferred subject code of the student's

6.2 search for group that matches 2nd preferred subject code of the student's

6.3 search for group that matches 3rd preferred subject code of the student's

7. assign group number to every gathered students regarding the number of students in the gathered set

7.1 figure out how many people are in the gathered set

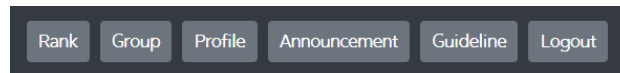
7.2 Assign the same number to certain number of students, the list of which are:

7.2.1 [3], [4], [5], [3, 3], [4, 3], [4, 4], [5, 4], [5, 5] when the number of students in the set are in range of [3, 10] (e.g., if the number of students is 7, assign the same number A to 4 students and the other number B to the other 3, thus expressed as [4, 3].)

7.2.2 $[[4, 4], [4, 4], [5, 4], [5, 5]][n \% 4 - 3] + ([4] * (((n + 1) // 4) - 3)) + ([3]$
if $n \% 4 == 3$ else [4]) otherwise.

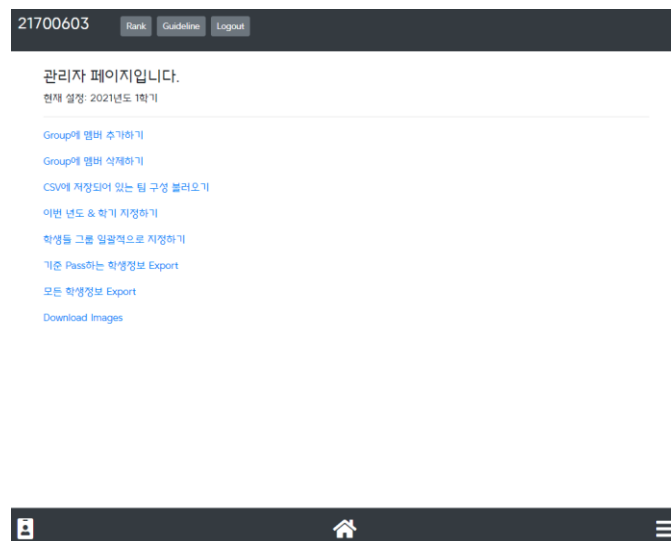
6. System user Manual

1) Administrator

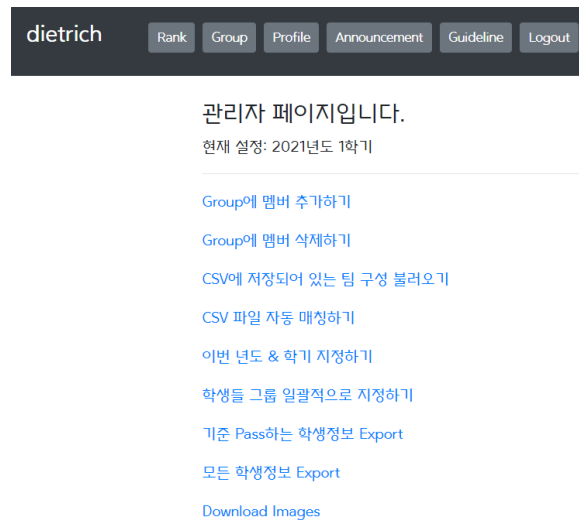


a) Profile

-origin menu list(access by home button which is in center of bottom)



-modification Menu list(access by clicking profile)



Administers can manage the study group composition using the functions shown in the picture.

csv automatic matching function has been added.

Please refer to the following for a detailed function description.

- Add new member to group([Group에 멤버 추가하기](#))

The form is titled '새 UserInfo 추가하기' (Add new UserInfo). It contains the following fields:

- 년도(ex. 2020): A text input field.
- 학기: A dropdown menu with '1' selected.
- Group 번호: A text input field.
- 학번: A text input field.
- 이름(ex. 홍길동): A text input field.

At the bottom of the form is a '제출' (Submit) button.

This function is used to add group members to an already formed group.

- Delete user from group (loading user info, delete user) ([Group에](#)
[멤버 삭제하기](#))

This function is what the administrator uses to remove a member from an already formed group.

그룹에서 유저 삭제하기

연도

2021

학기

1

그룹 번호

303

Load

Click the "Load" button

Group 303 정보

	학번	이름	
1	21600312	박천성	삭제
2	21500436	유채우	삭제

Click the "삭제" button

유저를 성공적으로 삭제했습니다.



If successful, a corresponding message is displayed.

- Uploading user information (CSV에 저장되어 있는 팀 구성 불러오기)

This is where group matching results are stored. CSV file format must be "Match User information"(CSV 파일 자동 매칭하기) format

- Setting the current year and semester(이번 년도 & 학기 지정하기)

- Setting user group's year and semester(학생들 그룹 일괄적으로 지정하기)

사용자들의 group을 지정할 년도와 학기 선택

해당 연도

해당 학기

지정

- Exporting student information who satisfies study pass condition(기준 Pass하는 학생정보 Export)

Export Student Info

Export할 연도

Export할 학기

스터디 Pass 횟수 입력

Export

- Export all students' information([모든 학생정보 Export](#))

모든 학생들 정보 Export

Export할 연도

Export할 학기 ▼

- Download all study images([Download Images](#))

이미지 다운로드

다운로드 받을 연도

2021

다운로드 받을 학기

1

지정

-Additional menu(CSV 파일 자동 매칭하기)

- Match User information

Match User Information

CSV 파일 선택

파일 선택

선택된 파일 없음

제출

Send information about preferences.

Brings the matched result.

It can be used for "Uploading user information" ([CSV에 저장되어 있는 팀 구성 불러오기](#))

b) Group

-Group page (list of study groups) ([스터디 그룹 목록](#))

<div>스터디 그룹 목록 스터디 사진 모음 TOP 3</div>				
<div>2021 ▾ 1 ▾ Load</div>				
	그룹 번호	보고서 수	최신 보고서 날짜	누적 공부 시간(분)
1	Group 404	1	2021-05-07 22:29:20	75
2	Group 303	0	-	0

-Group page (list of study pictures) ([스터디 사진 모음](#))

[스터디 그룹 목록](#)
[스터디 사진 모음](#)
[TOP 3](#)

2021
1
Load

5174

-Group page (Top 3 groups) ([TOP 3](#))

[스터디 그룹 목록](#)
[스터디 사진 모음](#)
[TOP 3](#)

2021
1
Load

그룹 번호	10번째 스터디 날짜	총 스터디 수
-------	-------------	---------

c) Announcement

-Managing the announcement

Rank
Group
Profile
Announcement
Guideline
Logout

click the "Announcement" button

공지 추가

공지	날짜	
1	공지 테스트 1	2021-05-07 23:16:00
		공지삭제

Click the "공지 테스트 1", then can see the contents. 공지삭제 is for deleting an announcement

날짜: 2021-05-07 23:16:00

공제 내용 작성함

bold 글씨체

공지 삭제

- Writing announcement

Click the "공지추가", then write the contents.

write announcement

제목

B

I

U

X'

X_i

S

Helvetica ▾

14 ▾

A ▾

☰

☷

☶ ▾

T ▾

☐

GO

🖼️

▶■

-

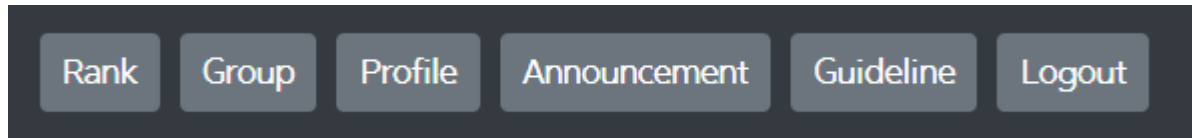
✕

</>

?

2) User

a) Profile(Study group status)



click the profile

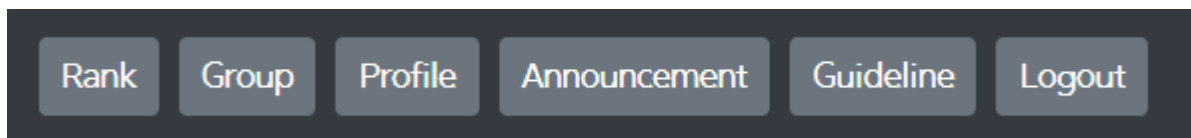
Group 64 멤버정보

학번	이름	출석횟수	누적시간
21700083	김도윤	1	3
21300109	김성민	0	None

Shows how many times members of this group have attended. The names of students who have attended more than a certain number are highlighted.

b) Group(management for report)

-register report



click the "Group" button



click the “보고서 쓰기” for writing report.

공지사항

문의

	공지	날짜
1	공지 테스트 1	2021-05-07 23:16:00
2	공지 test	2021-05-19 18:35:33
3	Announce create in admin page	2021-05-19 18:35:40
4	공지 test	2021-05-19 21:16:08

To view the content of the notice, click the title.