How to work the database:

It's pretty simple. There are basic functions for the three different classes: User, Tag and Photo

syntax for importing the database

import db #db.py is the python code

from db import (class: User or Photo or whatever)

The variables within class User are:

username TEXT PRIMARY KEY,

password TEXT,

profilepicurl TEXT, # I am assuming that we use the url for photos

firstname TEXT,

lastname TEXT,

email TEXT,

country TEXT,

sex TEXT,

age INT,

The variables within class Photo are:

id INTEGER PRIMARY KEY AUTOINCREMENT,

latitude FLOAT,

longitude FLOAT,

description TEXT,

uploader TEXT,

uploaddate DATETIME,

caption TEXT,

artist TEXT,

url TEXT,

The variables within class Tag are:

tagid INTEGER PRIMARY KEY AUTOINCREMENT,

tagstring TEXT, # text of the tag

id INTEGER, # id relating to the photo that it was tagged with

The functions within User are (other than \_\_init\_\_):

create

getpassword(self)

getprofilepicurl(self)

getfirstname

getlastname

getcountry

getsex

getemail

setpassword

setemail

setcountry

The functions within Photo are (other than \_\_init\_\_):

create

setprofilepicurl

getallprofilepicurl

getlocation

getallphotos

getpics #not available for use yet

The functions within Tag are (other than \_\_init\_\_):

Create

getallphotos

Example uses:

user1 = User.create("username", "password", etc...)

user1.getpassword() # it will return 'password')

user1.setpassword("12345") # it will replace the old password with '12345'

user1.getpassword() # will now return '12345'

# there are things that once set initially when the object is created cannot be changed (username, sex, age/birthday)

photo1 = Photo.create("id", "latitude", "longitude", etc...)

photo1.setprofilepicurl() # changes the profile picture url. don't know how we are implementing photos

photo1.getallphotos() # literally gets all photos

photo1.getpics(criteria, order, limit) # we will implement later