MARCH 9, 2014

# ASSIGNMENT 3 REPORT

IAT 352

CASSANDRA DE GIT
301166220
cdegit@sfu.ca

## Project Overview

{ Tutor } is a tool to help connect people who want to learn programming and those who want to teach it. Contributors post Lessons; blog post style content designed to teach some programming Topic. Learners are able to view these Lessons, and follow Contributors or Topics. By grouping content into specific Topics that users can search, { Tutor } makes it easier for Learners to find content they are interested in.

Unlike the suggested project, most of { Tutor }'s focus is on the posts themselves rather than the users creating them, as well as the organizing system for the posts, called Topics. This shaped a great deal of the functionality and design of the system.

## Database Design

The database design required significant changes since the last iteration. 4 new tables have been added: tweets, tweet_topics, flickr_photos and user_settings.

The tweets table stores the cached tweets for users. Tweets are treated very similarly to posts, but there are a few differences. Rather than using a simple auto-incrementing id, the id returned by Twitter itself is used. This makes it far easier to check if a tweet is already present in the database. Additionally, the authorTwitter field contains the user's Twitter account name, rather than their Tutor username. Their Tutor username can be easily obtained, however, by comparing tweets.authorTwitter with users.twitter. This table also contains a Boolean field called tutorTweet. Tweets with this attribute set to 0 are only displayed on that author's profile page, while tweets with the attribute set to 1 are displayed throughout the site. For tutorTweet to be set to 1, the tweet's content must contain #tutortweet. This gives the user some extra control in how their content is displayed on { Tutor }.

The tweet_topics table employs the same technique as the post_topics table. A tweet's id is stored, as well as the name of a topic it contains. If there are multiple topics within a tweet, a new entry is added for each topic. This allows us to have tweets with multiple topics while still maintaining First Normal Form. A tweet will be assigned a topic is the tweet's content contains #<topicName>. If a tweet has tutorTweet = 1 and 1 or more topics associated with it, it will be displayed on the pages of those topics as well as the home page.

The flickr_photos table is almost identical to the tweets table. The id used is the one returned by Flickr, rather than an auto-incrementing id, making it easy to check if a photo exists the in the database. In place of content, flickr_photos contains the photo's url, which is used to display the photo on its author's profile. The author_flickr attribute is set to the user's Flickr account name. Flickr photos are only every displayed on a user's profile, so no flickr_topics table or tutorFlickr attribute is needed.

The user_settings table allows users to configure the display of tweets and Flickr photos on the site. With _ and _, users are able to choose if they want to see other user's tweets and photos, respectively. If they are set to 1, which is the default, tweets or photos are displayed. If set to 0, they are hidden. The other two settings, _ and _, apply specifically to Contributors. Contributors are able to choose whether or not they want their tweets or photos to appear on the site. If set to 0, which is the default, their tweets or photos are hidden. If set to 1, tweets and photos are displayed.

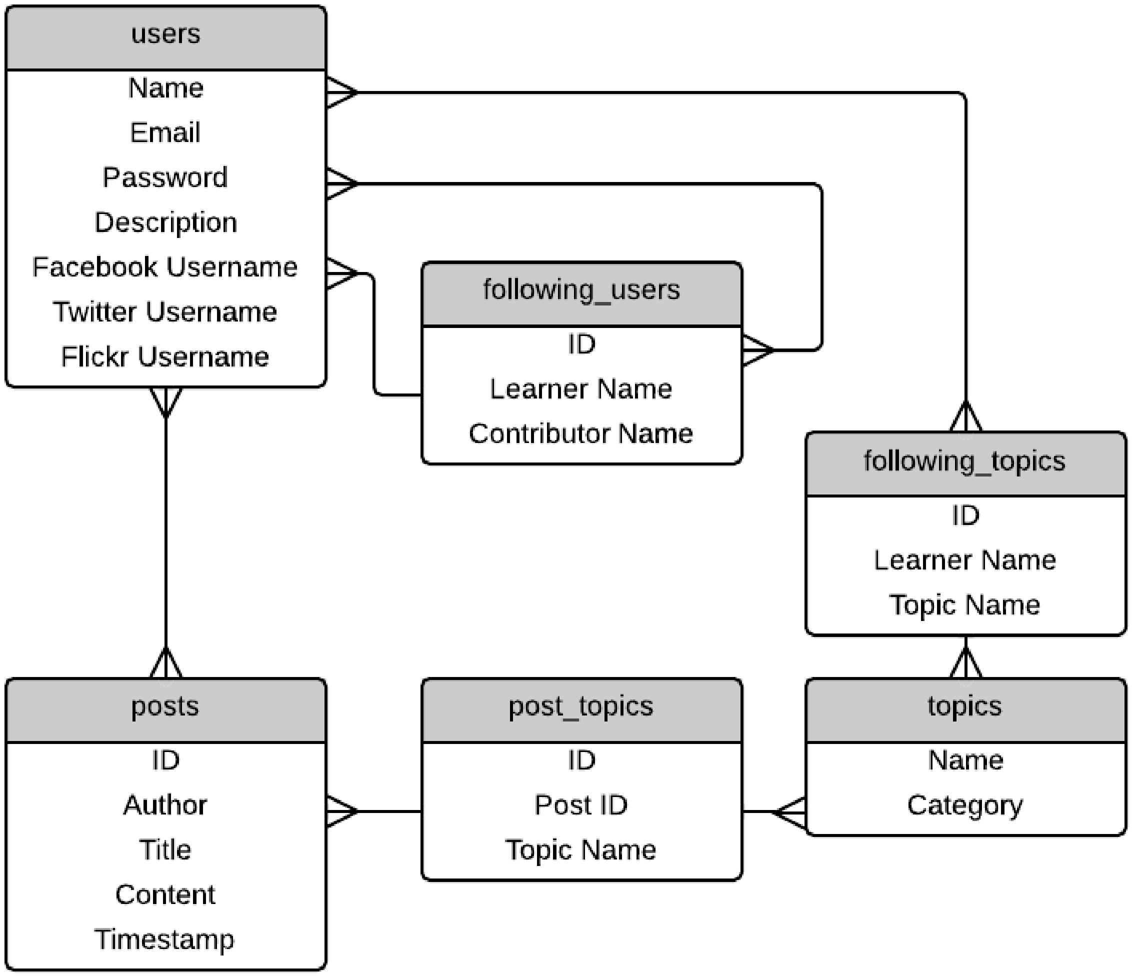

*Figure 1: Database from the last iteration*

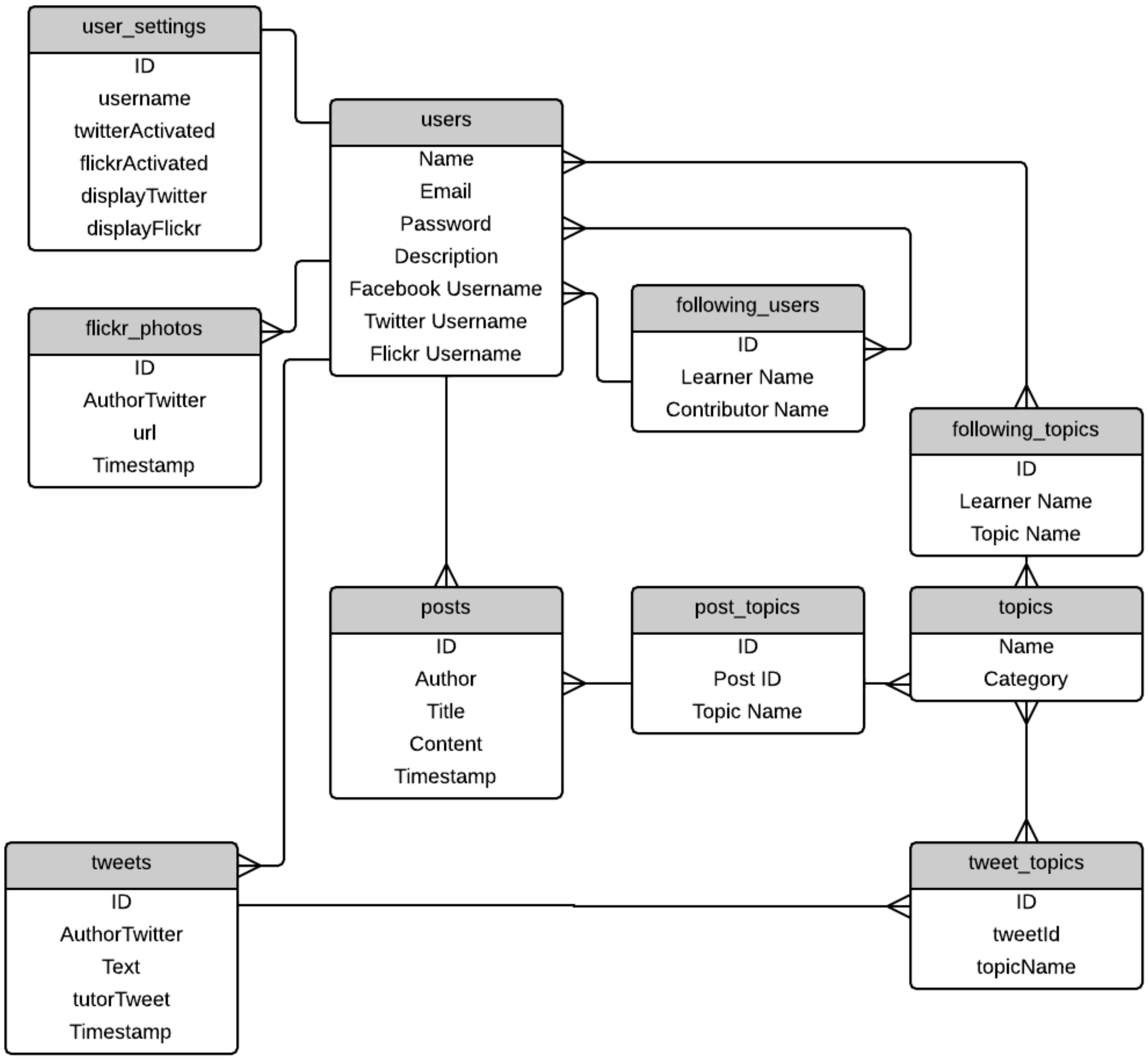

*Figure 2: New database with tweets, tweet_topics, flickr_photos and user_settings tables*

# Implementation

## Getting Tweets

Tweets are fetched from Twitter using the Codebird library, following the code provided in lab. The one difference is that tweets are only fetched from Twitter if the cache has expired. To determine if this has occurred, I look at the timestamp on the user's most recent tweet. Timestamps are based on the time the tweet is added to { Tutor }, not the tweet's Twitter timestamp, so you can determine the last time that new tweets were fetched from Twitter. This timestamp is compared with the current time, and if the difference is greater than a certain threshold, new Tweets are fetched. Currently, this threshold is set to 1 minute, as the Twitter API is quite fast and Twitter tends to be an application that users use very frequently.

## Caching Tweets

Once tweets have been fetched, we add them to the database. The first step is to check, for each tweet, if it already exists in the database. If it does not, we move on to checking if the tweet contains "#tutorTweet". If it does, we add it to the database with the tutortweet attribute set to 1. If not, we add the tweet with the attribute set to 0. This attribute will allow the tweet to be displayed around the site. Next, we go through all topics and check if the tweet contains "#<topicName>". For each topic found, an entry in tweet_topics is added. Tweets that have both tutorTweet = 1 and 1 or more topics set will display on the pages of those topics.

## Displaying Tweets

Tweets are displayed in two main places: the user's profile and the lessons pages. To be displayed on the user's profile page, tweets must meet a set of conditions:

```
$user['twitter'] != "" && $user['displayTwitter'] == 1 &&
(!isset($_SESSION['twitter']) || $_SESSION['twitter'] == 1)
```

This means that the user whose profile we are viewing must have their Twitter account set up, as well as their tweets enabled. `(!isset($_SESSION['twitter']) || $_SESSION['twitter'] == 1)` handles the settings for the user viewing the tweets. `!isset($_SESSION['twitter'])` allows non-logged in users to view the tweets if the other conditions are met, and `$_SESSION['twitter'] == 1` applies a logged-in user's settings.

To be displayed on the lessons pages, the tweet must meet all of the conditions to be displayed on the profile plus 1 more: tutorTweet must be set to 1. Tweets are displayed in topics only if an entry for that tweet and topic exists in tweet_topics.

## Deleting Tweets

Once tweets are no longer visible on the site (they are not within the most recent 5 tweets), there is no point in keeping them. Dropping tweets is only done when new tweets are fetched from Twitter. The query itself to handle this was surprisingly complicated, due to MySQL's lack of support for LIMIT within sub queries. I used a slightly modified version of the query provided in this Stack Overflow answer: http://stackoverflow.com/a/578926 .

```
$query = "DELETE FROM tweets WHERE authorTwitter = '" . $username . "' AND id NOT IN
(SELECT id FROM (SELECT id FROM tweets WHERE authorTwitter = '" . $username . "' ORDER
BY id DESC LIMIT 5)sub )";
```

While likely not the most efficient solution and certainly not the most human readable, it works and removes only the correct tweets.

## Getting Photos

Flickr photos are fetched in the same method shown in class. The only difference is that they are fetched only when their cache expires, exactly like tweets. The implementation is also the same, although the Flickr photos' cache expires in 1 day, rather than 1 minute. While this is a fairly long time, the Flickr API is exceptionally slow, and I wanted to avoid using it whenever possible. In addition, users are likely to use Flickr less frequently than Twitter.

## Caching Photos

Photos are cached in much the same way as tweets. Because they are only displayed on a user's profile, I don't check for any tags or have an equivalent tutorTweet field.

## Displaying Photos

Flickr photos are only displayed on the user's profile, as it didn't make sense to integrate them with the rest of the site. There are several conditions that must be met for photos to display.

```
$user['flickr'] != "" && $user['displayFlickr'] == 1 && (!isset($_SESSION['flickr'])
|| $_SESSION['flickr'] == 1)
```

`$user['flickr'] != ""` means that the user whose profile we are viewing must have their Flickr account set up. The second condition, `$user['displayFlickr'] == 1` means that this user must have enabled the displaying of their photos on their profile. This was done so users had more control over their own content and data. `!isset($_SESSION['flickr']` allows non-logged in users to view the photos if the other conditions are met, while `$_SESSION['flickr'] == 1` handles the logged in, viewing user's settings. If `$_SESSION['flickr']` is set to 1, the user has Flickr enabled and will see the photos.

## Deleting Old Photos

Deleting old Flickr photos uses an identical process as deleting old tweets. The query used here was:

```
$query = "DELETE FROM flickr_photos WHERE authorFlickr = '" . $username . "' AND id
NOT IN (SELECT id FROM (SELECT id FROM flickr_photos WHERE authorFlickr = '" .
$username . "' ORDER BY id DESC LIMIT 9)sub )";
```

## Settings

Each user has 4 settings associated with them, all stored in the table user_settings.

twitterActivated: If set to 1, tweets are displayed on the site. If 0, tweets are hidden. Default 1.

flickrActivated: If set to 1, Flickr photos are displayed on the site. If 0, photos are hidden. Default 1.

displayTwitter: A setting for contributors. If set to 1, their tweets will appear on the site. If 0, the tweets are hidden. It is set to 0 by default.

displayFlickr: A setting for contributors. If set to 1, their photos will appear on their profile. If 0, the photos are hidden. It is set to 0 by default.

## Summary of Current Functionality

New additions are bolded

- Users can register on the site as either Contributors or Learners
- Users can log in to the site
- Contributors are able to create posts and tag them with predefined topics
- Contributors are able to edit their own posts
- Contributors can edit their profile (currently just their description)
- Both user types and non-logged in users are about to browse the site
    - All posts can be viewed by selecting the Lessons link
    - All posts from a specific Topic can be viewed
    - The Topic navigation menu on the Lessons page will update whenever new topics or categories are added (this has to be done directly in PHPmyAdmin at the moment)
    - All users can be viewed, either sorted alphabetically or by the topics they write about
        - A user's topics are determined by the topics selected for their posts
    - Individual users' profiles can be viewed, and displays all of their posts
    - Individual posts can be viewed
- Learners are about to follow users and topics
    - Posts either from these users or in these topics will appear on their dashboard
    - Learners are able to unfollow users and topics
- Users are able to log out
- **Contributors are able to connect their Twitter account through the Edit Profile page**
- **Contributors are able to connect their Flickr account through the Edit Profile page**
- **If the user has Twitter enabled, their 5 most recent tweets are displayed**
    - **The Twitter icon provides a link to their Twitter profile**
- **If the user had Flickr enabled, their 9 most recent images are displayed**
    - **A link to their Flickr page is also provided**
- **If a Tweet is tagged #tutortweet, it is integrated with the rest of the site in a post-like manner**
    - **Tweets are displayed on the Lessons page**
    - **Tweets with tags containing topic names, such as #Python, are displayed in that tag**
    - **Tweets that are not tagged will only appear on the user's page**
- **Both contributors and visitors can disable or enable Tweets and Flickr photos**
    - **If these are disabled, Tweets and Flickr photos will not be visible to them while browsing the site**
- **Contributors can disable or enable posting of their own Tweets or photos**
- **Tweets and Flickr photos are cached. Currently, the Twitter cache expires after a minute, and the Flickr cache expires after a day**
    - **This is done because Flickr is quite slow, and users are unlikely to post to Flickr as frequently as they do Twitter**
    - **When the user's cached Flickr photos exceed 9, the max amount shown on their profile, all older photos are deleted from the database**
    - **When the user's cached Tweets exceed 5, all older tweets are deleted from the database**

- **Tweets made by followed users will appear in Learner's dashboards**

## Files Changed

- controller.php
- dashboard.php
- displayTwitter.php
- displayFlickr.php
- displayProfile.php
- displayLessons.php
- header.php
- footer.php
- processLogin.php
- processRegistration.php
- processUpdate.php
- processSettings.php
- editProfile.php
- twitterConfig.php
- userSettings.php
- script.js
- style.css

## Future Changes

Users should be able to add a profile picture, as well as select a photo to display on their posts. These photos could even be obtained from their Flickr photos. Additionally, the template for posts needs to be fixed so that posts without photos will display appropriately.

Currently, when populating a Learner's dashboard, two queries are made: 1 to get the posts from users they follow, and the other to get posts in topics they follow. Both individual queries are ordered by their timestamp. However, when the results are placed into 1 array to display them, the new array is no longer in order – posts from topics will always be behind posts from followed users. To fix this, I need to implement the merge step from merge sort, and combine the two arrays into one with proper order. However, this is not an extremely critical fix, as the exact time posts were created isn't displayed to the user, so they won't be able to detect that the posts are out of order.

Pagination still needs to be implemented.

Renaming of "lessons" to "posts" throughout the site will not be done due to the amount of work required for a change that doesn't affect the end user.