CS3083 Database
**Project Part 4**
*5/3/2020*

Claire Cheng, Michael (Zixi) Zhao
yc3093 zz1818
Prof. Frankl

**1.** GitHub repo: `https://github.com/claireCheng98/Finstagram.git`

**2.** In this project, Claire Cheng was responsible for Extra Feature **9** (search by tag), and Feature **10** (search by post), Michael Zhao was responsible for Extra Feature **11** (add a friend), and Feature **8** (unfollow).

For feature 8, the unfollow function needs to handle not only between followers and followees, and also tags. Photos posted by followee that the follower was tagged in are no more visible to the follower, thus the tags shall be removed. However, there's an exception that if these two people are in the same FriendGroup, and the photo was shared in that FriendGroup, then the tags could be kept since the photo is still visible after unfollowing. The SQL query first finds if the follower was tagged in any followee's photo in `Follow` table, then checks if that photo has been shared to a FriendGroup that both follower and followee are in. If there exists such photo, then the tag of follower in the photo shall be deleted. (NOTE: This paragraph is to fulfill the requirement of summary of deletion handling, full implementation was coded in the python file.)

CONCEPTS for FEATURE 9 SEARCH BY TAG:
Like Feature 10, there will be a search page, where the user will need to log in, and will be asked to provide the username of whom being tagged. All the results will be displayed with the query:

```
query = 'SELECT DISTINCT username FROM Person WHERE
    username LIKE (%s)'
cursor.execute(query,input)
```

Then by clicking the href the user will land in the search page `/show_posts_of_tag/<tagged_name>`. And the result will be presented with the following query:

```
view = 'CREATE VIEW visiblePhoto AS(SELECT postingDate,
    pID, filePath FROM Photo WHERE poster = %s OR poster IN
    (SELECT followee FROM Follow WHERE follower = %s AND
    followStatus = 1) OR pID IN (SELECT pID FROM BelongTo
    JOIN SharedWith USING (groupName, groupCreator) WHERE
    username = %s) ORDER BY postingDate DESC'
query='SELECT pID FROM visiblePhoto JOIN Tag USING pID
    WHERE username=%s AND tagStatus=1'
cursor.execute(view,(user,user,user))
cursor.execute(query, tagged_name)
```

Finstagram will return the pID of the photos that the tagged user is being tagged in.

**3.** We used python/Flask.

**4.** Some additional comments:

   (a) We inserted original passwords in hashed format into the Database

   (b) We stored the pictures (in filePath) using LONGBLOB datatype, thus the pictures themselves can be retrieved online at anytime, but the insert statement with only file name will not work. We manually updated filePath on MAMP database stored as LONGBLOB.