**Databases Project Summary**

**Link to GitHub repository:**

https://github.com/Sed458/Finstagram.git

**Date of last commit to repository:**

12/9/2019

**Number of Team members:**

2

**Names and netIDs of Team members (one per line):**

Florence Tong (Ft700)

Sophia Danielsen (Sed458)

1. Name of the feature: Manage Follows
2. The full name of the team member who is primarily in charge of implementing this feature Sophia Danielsen
3. The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don’t have to include them here; just include the SQL statements that do the main work for this feature.)

🡪 username = session[‘username’]

query = SELECT \* FROM follow WHERE followstatus = 0 and username\_followed = ‘%s’

cursor.execute(query, (username))

🡪 username = session[‘username’]

🡪 followerUsername = request.form[‘followerName’]

If accepted:

updateFollowQuery = UPDATE Follow SET followstatus = 1 WHERE username\_followed = ‘%s’ and username\_follower = ‘%s’

cursor.execute(updateFollowQuery, (username, followerUsername))

Elif declined:

updateFollowQuery = DELETE FROM Follow WHERE username\_followed = ‘%s’

cursor.execute(updateFollowQuery, (username))

1. A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

Filename: init1.py

Look for functions: manage() and followStatus()

Html template: manage.html

1. One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that’s in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.

Demo found on Github 🡪 Manage\_Followers\_Demo

1. Name of the feature: Manage Tags
2. The full name of the team member who is primarily in charge of implementing this feature: Sophia Danielsen
3. The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don’t have to include them here; just include the SQL statements that do the main work for this feature.)

username = session[“username”]

photoID = request.form[“photoID”]

query = SELECT \* FROM Tagged NATURAL JOIN Photo WHERE tagstatus = 0 AND username = ‘%s’

cursor.execute(query,(username))

If accepted:

updateQuery = UPDATE Tagged SET tagstatus = 1 WHERE username = %s AND photoID = ‘%s’

Elif declined:

updateQuery = DELETE FROM Tagged WHERE username = ‘%s’ AND photoIS = ‘%s’

cursor.execute(updateQuery, (username, photoID)

1. A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

Filename: init1.py

Look for functions: manage() and tagStatus()

Html template: manage.html

1. One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that’s in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.

Demo found on Github 🡪 Manage\_Tags\_Demo

1. Name of the feature: Search by Poster
2. The full name of the team member who is primarily in charge of implementing this feature: Sophia Danielsen
3. The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don’t have to include them here; just include the SQL statements that do the main work for this feature.)

poster = request.form[‘poster’]

username = session[‘username’]

If poster = username:

SELECT \* FROM Photo NATURAL JOIN Person WHERE photoPoster = username AND photoPoster = %s

cursor.execute(query, (username))

Else:

SELECT \* FROM Photo NATURAL JOIN Person WHERE photoPoster = username AND photoPoster = %s AND (allFollowers = 1 AND photoPoster IN (SELECT username\_followed AS photoPoster FROM Follow WHERE username\_followed = %s AND username\_follower = %s AND followstatus = 1) OR photoID IN (SELECT photoID FROM Photo NATURAL JOIN SharedWith NATURAL JOIN BelongTo WHERE member\_username = %s)) ORDER BY postingdate DESC

cursor.execute(query, (poster, poster, username, username))

1. A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

Filename: init1.py

Look for function show\_posts()

Html file = home.html, upload.html and show\_post.html

1. One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that’s in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.

Demo found on Github 🡪 Search\_By\_Poster\_Demo

1. Name of the feature: Add Friend Group
2. The full name of the team member who is primarily in charge of implementing this feature: Florence Tong
3. The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don’t have to include them here; just include the SQL statements that do the main work for this feature.)

username = session["username"]

groupName = request.form['groupName']

query = INSERT INTO FriendGroup (groupOwner, groupName) VALUES(%s, %s)

cursor.execute(query, (username, groupName))

belongQuery = INSERT INTO BelongTo (member\_username, owner\_username, groupName) VALUES(%s, %s, %s)

cursor.execute(belongQuery, (username, username, groupName))

1. A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

Filename: init1.py

Look for function createGroup()

Html file: home.html

1. One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that’s in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.

Demo found on Github 🡪 Add\_Friend\_Group\_Demo

1. Name of the feature: Add Friend
2. The full name of the team member who is primarily in charge of implementing this feature: Florence Tong
3. The queries (and any other SQL statements) used in your implementation of the feature. (If there are standard queries used in most or all of your features, you don’t have to include them here; just include the SQL statements that do the main work for this feature.)

username = session["username"]

groupName = request.form["groupName"]

newMember = request.form["newMember"]

query = 'INSERT INTO BelongTo (member\_username, owner\_username, groupName) VALUES(%s, %s, %s)'

cursor.execute(query, (newMember, username, groupName))

1. A clear indication of where to find the application source code for the feature within your GitHub repository (filename and where to look in the file).

Filename: init1.py

Look for function addMember()

Html file: home.html

1. One or more screenshots or a short video demonstrating the feature, showing how it appears in the browser; Also show the relevant data that’s in the database when you execute this demonstration (before and after if the feature changes the data), either as screenshots or as text. You may either include this in your GitHub and provide the link here or add screenshots here or on a separate page. Make it clear where the graders should look for this.

Demo found on Github 🡪 Add\_Friend\_Demo