Number of Team members: 2
Names: CJ Syki, Ahmed Salem
Name of team member who uploaded the zip to NYU Classes: CJ Syki
Hash of zip file:
==========
First Extra Feature (for teams of 2 or 3 people)

- Liking a photo

Very Brief summary:

- Users can choose to like/unlike a photo they can view in the Images page

1. The name of the team member who is primarily in charge of implementing this feature

   Ahmed

2. The queries (and any other SQL statements) used in your implementation of the feature

```
    # if user clicked unlike, remove from liked table.
            if option == "unlike":
                query = "DELETE FROM Liked WHERE likerUsername = %s AND\
                        photoID = %s"
                runQuery( query, None, ( username, photoID ) )
            # if user clicked like, add to like table
            elif option == "like":
                query = "INSERT INTO Liked VALUES( %s, %s, %s )"
                runQuery( query, None, ( username, photoID, \
                        time.strftime('%Y-%m-%d %H:%M:%S') ) )
```

3. A clear indication of where to find the application source code for the feature. Include plenty of comments and/or a few sentences explaining the code.

Under the images function, starting at **line 198.** It basically checks which link the user clicked. If the user pressed unlike, we want to remove their like from the Liked table (given their username and photoID); otherwise, if the user did like the photo, we insert their Like into the Like table (timestamp is not necessary).

The most important thing is that we **show the opposite link of whether the user liked the photo or not**. That is, if the user did like the photo, **we show the "Unlike" option**, and vice versa. That way, the user won't accidentally press "Like" twice. (see photos below)

4. One or more screenshots 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. **[Include on next pages or at the end]**
5. For features other than those I suggested (7 -- 11), also include:
   i. What the feature is (in the style of the 6 requirements above)

ii. A sentence or two on why this is a good feature to add to Finstagram

iii. A clear explanation of any changes to the database schema that are needed (including additional tables, additional attributes, or additional constraints)

Before liking the photo (photoID = 29, current user = cjsyki)

Image Gallery

By: cj syki (cjsyki)
Uploaded on: 2019-03-25 13:45:12
my schedule
Like Photo Delete Photo
Enter who you want to tag Submit

Tagged:
- test

Liked by:
- test

Comments:
Enter your comment: Add comment

cjsyki: my comment!

	likerUsername	photoID	timestamp
Edit Copy Delete	test	28	2019-03-31 19:36:13
Edit Copy Delete	sac784	28	2019-03-31 14:23:22
Edit Copy Delete	test	6	2019-03-22 21:02:29
Edit Copy Delete	cjsyki	5	2019-03-21 21:01:16
Edit Copy Delete	sac784	6	2019-03-19 00:00:00
Edit Copy Delete	test	29	2019-03-31 14:18:55
Edit Copy Delete	cjsyki	30	2019-04-04 12:56:33

After liking the photo

Image Gallery

By: cj syki (cjsyki)
Uploaded on: 2019-03-25 13:45:12
my schedule
Unlike Photo Delete Photo
Enter who you want to tag Submit

Tagged:
- test

Liked by:
- test
- cjsyki

Comments:
Enter your comment: Add comment

cjsyki: my comment!

	likerUsername	photoID	timestamp
Edit Copy Delete	test	28	2019-03-31 19:36:13
Edit Copy Delete	sac784	28	2019-03-31 14:23:22
Edit Copy Delete	test	6	2019-03-22 21:02:29
Edit Copy Delete	cjsyki	5	2019-03-21 21:01:16
Edit Copy Delete	sac784	6	2019-03-19 00:00:00
Edit Copy Delete	test	29	2019-03-31 14:18:55
Edit Copy Delete	cjsyki	30	2019-04-04 12:56:33
Edit Copy Delete	cjsyki	29	2019-04-27 20:20:52

Second Extra Feature (for teams of 2 or 3 people)

- Unfollowing someone

Very Brief summary:

- On the Follows page, users (if following someone) can choose to unfollow them.

6. The name of the team member who is primarily in charge of implementing this feature

   Ahmed

7. The queries (and any other SQL statements) used in your implementation of the feature

```
        # if user unfollows someone, delete from follows
        if option == "unfollow":
            query = "DELETE FROM Follow\
                    WHERE followerUsername = %s AND followeeUsername =
%s"
            runQuery( query, None, ( currentUsername, followUsername ) )
```

8. A clear indication of where to find the application source code for the feature. Include plenty of comments and/or a few sentences explaining the code.

   Under the follow( ) function, starting at **line 568**. When the user chooses to unfollow someone, their corresponding row (follower = current user, followee = person current user followed) will be removed. By doing this and instead not setting acceptFollow = 0, we ensure that the user did unfollow, and not that the user did not accept a follow.

9. One or more screenshots 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. [Include on next pages or at the end]

10. For features other than those I suggested (7 -- 11), also include:
    i. What the feature is (in the style of the 6 requirements above)

    ii. A sentence or two on why this is a good feature to add to Finstagram

iii. A clear explanation of any changes to the database schema that are needed (including additional tables, additional attributes, or additional constraints)

Before unfollowing (follower = cjsyki, followee = sac784)

Follow someone:

Enter username: Send request

current user = cjsyki

Follow requests:

Following:

sac784 - Unfollow

Followed by:

test
sac784

Go back

Follow someone:

Enter username: Send request

current user = sac784

Follow requests:

Following:

cjsyki - Unfollow
test - Unfollow

Followed by:

cjsyki

Go back

+ Options

	followerUsername	followeeUsername	acceptedfollow
Edit Copy Delete	sac784	test	1
Edit Copy Delete	test	cjsyki	1
Edit Copy Delete	cjsyki	sac784	1
Edit Copy Delete	sac784	cjsyki	1
Edit Copy Delete	CC	AA	0
Edit Copy Delete	DD	AA	1

Check all *With selected:* Edit Copy Delete Export

After unfollowing

Follow someone:

Enter username: Send request

current user = cjsyki

Follow requests:

Following:

Followed by:

test
sac784

Go back

Follow someone:

Enter username: Send request

current user = sac784

Follow requests:

Following:

cjsyki - Unfollow
test - Unfollow

Followed by:

Go back

	followerUsername	followeeUsername	acceptedfollow
Edit Copy Delete	sac784	test	1
Edit Copy Delete	test	cjsyki	1
Edit Copy Delete	sac784	cjsyki	1
Edit Copy Delete	CC	AA	0
Edit Copy Delete	DD	AA	1

Check all *With selected:* Edit Copy Delete Export

Third Extra Feature (for teams of 2 or 3 people)

- Commenting on photos

Very Brief summary:

- Users can choose to add comments to visible photos

11. The name of the team member who is primarily in charge of implementing this feature

    CJ

12. The queries (and any other SQL statements) used in your implementation of the feature

```
# grab current user, comment, and photoID
        username = session[ "username" ]
        comment = request.form[ "comment" ]
        photoID =  request.args.get( "photoID" )

        # if user wrote no comment, return an error
        if len( comment ) == 0:
            return redirect( url_for( "images", error = "you must
enter a comment" ) )

        # add comment to the comments table
        query = "INSERT INTO Comment VALUES( %s, %s, %s, %s )"
        runQuery( query, None, ( username, photoID, comment, \
                        time.strftime('%Y-%m-%d %H:%M:%S') ) )
```

13. A clear indication of where to find the application source code for the feature. Include plenty of comments and/or a few sentences explaining the code.

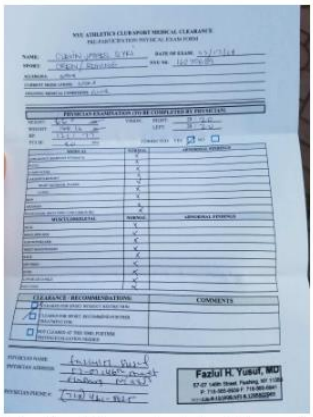
    Under the addComment( ) function starting on **line 724**. We grabbed the comment the user wrote (and their username and corresponding photoID) and inserted it into the Comment table (along with timestamp). We included the timestamp because we want to sort by timestamp (that way, oldest comments appear at the top)

14. One or more screenshots 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. [Include on next pages or at the end]
15. For features other than those I suggested (7 -- 11), also include:
    i. What the feature is (in the style of the 6 requirements above)

    ii. A sentence or two on why this is a good feature to add to Finstagram

iii. A clear explanation of any changes to the database schema that are needed (including additional tables, additional attributes, or additional constraints)

Before adding a comment (photoID = 28, current user = sac784)



By: firstName_test lastName_test (test)
Uploaded on: 2019-03-24 14:38:26
my physical lmao (for my followers)
Unlike Photo
Enter who you want to tag | Submit

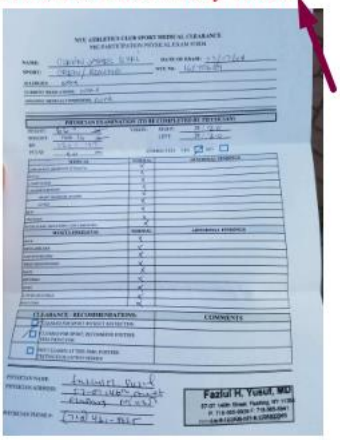
Tagged:
Liked by:
- sac784
- test

Comments:
Enter your comment: | Add comment

username	photoID	commentText	timestamp

After adding a comment

comment successfully added



By: firstName_test lastName_test (test)
Uploaded on: 2019-03-24 14:38:26
my physical lmao (for my followers)
Unlike Photo
Enter who you want to tag | Submit

Tagged:
Liked by:
- sac784
- test

Comments:
Enter your comment: | Add comment

sac784: my first comment

	username	photoID	commentText	timestamp
Edit Copy Delete	sac784	28	my first comment	2019-04-27 21:43:47

Check all *With selected:* Edit Copy Delete Export

Fourth Extra Feature (for teams of 2 or 3 people)

- Deleting a photo

Very Brief summary:

- Users can delete photos they own

16. The name of the team member who is primarily in charge of implementing this feature

    CJ

17. The queries (and any other SQL statements) used in your implementation of the feature

```
    # if user clicked delete photo, delete the photo (w photoID)
            # from photo, tag, liked, caption table
            elif option == "delete":
                query = "DELETE FROM Photo WHERE photoID = %s"
                runQuery( query, None, photoID )
                query = "DELETE FROM Share WHERE photoID = %s"
                runQuery( query, None, photoID )
                query = "DELETE FROM Liked WHERE photoID = %s"
                runQuery( query, None, photoID )
                query = "DELETE FROM Tag WHERE photoID = %s"
                runQuery( query, None, photoID )
                query = "DELETE FROM Comment WHERE photoID = %s"
                runQuery( query, None, photoID )
```

18. A clear indication of where to find the application source code for the feature. Include plenty of comments and/or a few sentences explaining the code.

    Under the images( ) function starting on **line 208**. The option to delete a photo will only appear if the current user is the owner of the photo. When the user clicks it, not only will it be removed from the Photo table, but all corresponding entries in Share, Liked, Tag, and Comment tables will be deleted.

19. One or more screenshots 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. [Include on next pages or at the end]
20. For features other than those I suggested (7 -- 11), also include:
    i. What the feature is (in the style of the 6 requirements above)

- The feature is the ability to delete a photo. The option to delete a photo will only appear to the photo owners.

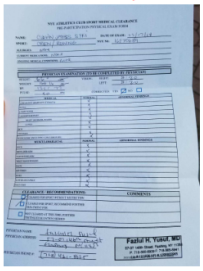
    ii. A sentence or two on why this is a good feature to add to Finstagram

- This is a good feature because in Instagram, you are given the option to delete photos. If the photo owner made a mistake as well (shared to wrong people, uploaded wrong photo, mistyped caption, etc.), the owner can choose to delete and try again.

iii. A clear explanation of any changes to the database schema that are needed (including additional tables, additional attributes, or additional constraints)

- No additional changes were made to the schema. We just need to delete rows in several tables.

<u>Before deleting the photo (current user = sac784, photoID = 30)</u>



By: firstName_test lastName_test (test)
Uploaded on: 2019-03-24 14:38:26
my physical lmao (for my followers)
Unlike Photo
Enter who you want to tag Submit

Tagged:
Liked by:
- sac784
- test

Comments:
Enter your comment: Add comment

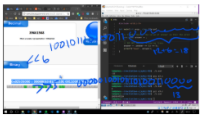
sac784: my first comment

By: cj syki (cjsyki)
Uploaded on: 2019-03-25 13:45:12
my schedule
Like Photo
Enter who you want to tag Submit

Tagged:
- test

Liked by:
- test

Comments:
Enter your comment: Add comment

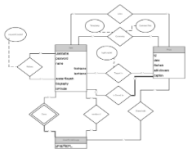


By: susell contreras (sac784)
Uploaded on: 2019-03-31 14:23:54
for my group
Like Photo Delete Photo
Enter who you want to tag Submit

Tagged:
- sac784 Delete Tag

Liked by:
- cjsyki

Comments:
Enter your comment: Add comment



By: susell contreras (sac784)
Uploaded on: 2019-04-27 14:16:00
for my followers again
Like Photo Delete Photo
Enter who you want to tag Submit

Tagged:
Liked by:
- None

Comments:
Enter your comment: Add comment

+ Options

PHOTO TABLE

	photoID	photoOwner	timestamp	filePath	caption	allFollowers
Edit Copy Delete	28	test	2019-03-24 14:38:26	NYU external physical.jpg	my physical lmao (for my followers)	1
Edit Copy Delete	29	cjsyki	2019-03-25 13:45:12	Clevin_Syki - Fall 2018.png	my schedule	1
Edit Copy Delete	30	sac784	2019-03-31 14:23:54	Sketch (1).png	for my group	0
Edit Copy Delete	1	AA	2018-01-19 03:14:07	image1.jpeg	*NULL*	1
Edit Copy Delete	2	AA	2018-01-19 03:14:08	image2.jpeg	*NULL*	0
Edit Copy Delete	33	sac784	2019-04-27 14:16:00	draft01 (2).png	for my followers again	1

+ Options

TAG TABLE

	username	photoID	acceptedTag
Edit Copy Delete	sac784	30	1
Edit Copy Delete	test	29	1
Edit Copy Delete	DD	1	0
Edit Copy Delete	DD	2	1

+ Options

SHARE TABLE

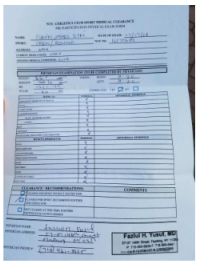
	groupName	groupOwner	photoID
Edit Copy Delete	family	AA	2
Edit Copy Delete	myGroup	sac784	30
Edit Copy Delete	test	test	23

+ Options

LIKED TABLE

	likerUsername	photoID	timestamp
Edit Copy Delete	test	28	2019-03-31 19:36:13
Edit Copy Delete	sac784	28	2019-03-31 14:23:22
Edit Copy Delete	test	6	2019-03-22 21:02:29
Edit Copy Delete	cjsyki	5	2019-03-21 21:01:16
Edit Copy Delete	sac784	6	2019-03-19 00:00:00
Edit Copy Delete	test	29	2019-03-31 14:18:55
Edit Copy Delete	cjsyki	30	2019-04-04 12:56:33

After deleting photo



By: firstName_test lastName_test (test)
Uploaded on: 2019-03-24 14:38:26
my physical lmao (for my followers)
Unlike Photo
Enter who you want to tag Submit

Tagged:
Liked by:
- sac784
- test

Comments:
Enter your comment: Add comment

sac784: my first comment

By: cj syki (cjsyki)
Uploaded on: 2019-03-25 13:45:12
my schedule
Like Photo
Enter who you want to tag Submit

Tagged:
- test

Liked by:
- test

Comments:
Enter your comment: Add comment

By: susell contreras (sac784)
Uploaded on: 2019-04-27 14:16:00
for my followers again
Like Photo Delete Photo
Enter who you want to tag Submit

Tagged:
Liked by:
- None

Comments:
Enter your comment: Add comment

PHOTO TABLE

+ Options

	photoID	photoOwner	timestamp	filePath	caption	allFollowers
Edit Copy Delete	28	test	2019-03-24 14:38:26	NYU external physical.jpg	my physical lmao (for my followers)	1
Edit Copy Delete	29	cjsyki	2019-03-25 13:45:12	Clevin_Syki - Fall 2018.png	my schedule	1
Edit Copy Delete	1	AA	2018-01-19 03:14:07	image1.jpeg	*NULL*	1
Edit Copy Delete	2	AA	2018-01-19 03:14:08	image2.jpeg	*NULL*	0
Edit Copy Delete	33	sac784	2019-04-27 14:16:00	draft01 (2).png	for my followers again	1

SHARE TABLE

+ Options

	groupName	groupOwner	photoID
Edit Copy Delete	family	AA	2
Edit Copy Delete	test	test	23

+ Options

TAG TABLE

	username	photoID	acceptedTag
Edit Copy Delete	test	29	1
Edit Copy Delete	DD	1	0
Edit Copy Delete	DD	2	1

+ Options

LIKED TABLE

	likerUsername	photoID	timestamp
Edit Copy Delete	test	28	2019-03-31 19:36:13
Edit Copy Delete	sac784	28	2019-03-31 14:23:22
Edit Copy Delete	test	6	2019-03-22 21:02:29
Edit Copy Delete	cjsyki	5	2019-03-21 21:01:16
Edit Copy Delete	sac784	6	2019-03-19 00:00:00
Edit Copy Delete	test	29	2019-03-31 14:18:55

Fifth Extra Feature (for teams of 3 people)

Very Brief summary:

21. The name of the team member who is primarily in charge of implementing this feature

22. The queries (and any other SQL statements) used in your implementation of the feature

23. A clear indication of where to find the application source code for the feature. Include plenty of comments and/or a few sentences explaining the code.

24. One or more screenshots 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. [Include on next pages or at the end]

25. For features other than those I suggested (7 -- 11), also include:
    i. What the feature is (in the style of the 6 requirements above)

    ii. A sentence or two on why this is a good feature to add to Finstagram

    iii. A clear explanation of any changes to the database schema that are needed (including additional tables, additional attributes, or additional constraints)

Sixth Extra Feature (for teams of 3 people)

Very Brief summary:

26. The name of the team member who is primarily in charge of implementing this feature

27. The queries (and any other SQL statements) used in your implementation of the feature

28. A clear indication of where to find the application source code for the feature. Include plenty of comments and/or a few sentences explaining the code.

29. One or more screenshots 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. [Include on next pages or at the end]

30. For features other than those I suggested (7 -- 11), also include:
    i. What the feature is (in the style of the 6 requirements above)

    ii. A sentence or two on why this is a good feature to add to Finstagram

    iii. A clear explanation of any changes to the database schema that are needed (including additional tables, additional attributes, or additional constraints)