

Group 6 - mysocial
Aidan Bjelke - 10160269 (T01)
Steven Canon-Almagro - 10155792 (T01)
Carl Lyss - 30025962 (T02)

Final Report

Abstract:

We were asked to research, design, and implement a database that could become a real world application. This was done by learning about database management techniques in class and through individual research. The database design was selected from several possible applications using EER, relational, UML and Sequential diagrams. The database was created using Python, SQL, PHP, and HTML languages. XAMPP was picked in order to work with MySQL, PHP, Python and Apache software. The database was designed to fulfill the needs of a social media website. Specifically to store the minimal amount of information of our users, their posts, their threads, and the relationships between them. While the website turned out functional, it did not end up with all the promised functionalities from our proposal due to time constraints. A negative of the design is that once an anonymous user creates a post, they cannot delete it after a session. Also, the website has some bugs like for instance it allows you to follow yourself, so these are some simple checks that should be discussed and kept in mind for future iterations. Additionally, the design of the website could use some work to look more modern and visually appealing. The system itself has several interesting properties, posts are identified by their primary key, so no duplicates of posts can be made. This makes every post unique and very powerful.

Introduction:

Social media is a hub that promotes the creation and sharing of information to anyone in your circle and beyond. Lately social media has been monopolized by two big players, Facebook, Instagram (owned by Facebook), and Twitter. This has given these companies huge power. Information can be saved and manipulated to their liking without the fear of retaliation. The solution is to make our own social media platform, raising competition in the market while giving more power to the user. The motivation of this is to allow the user to feel at ease when using social media.

Proposed solution:

Our project aims to be a good alternative for other social media apps that provides a flexible experience and functionality. Our hope is it will increase awareness surrounding this issue of privacy and provide more competition in the social media space

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

Project Design:

The database was built to support 4 types of users: anonymous, logged in, admins, and advertisers.

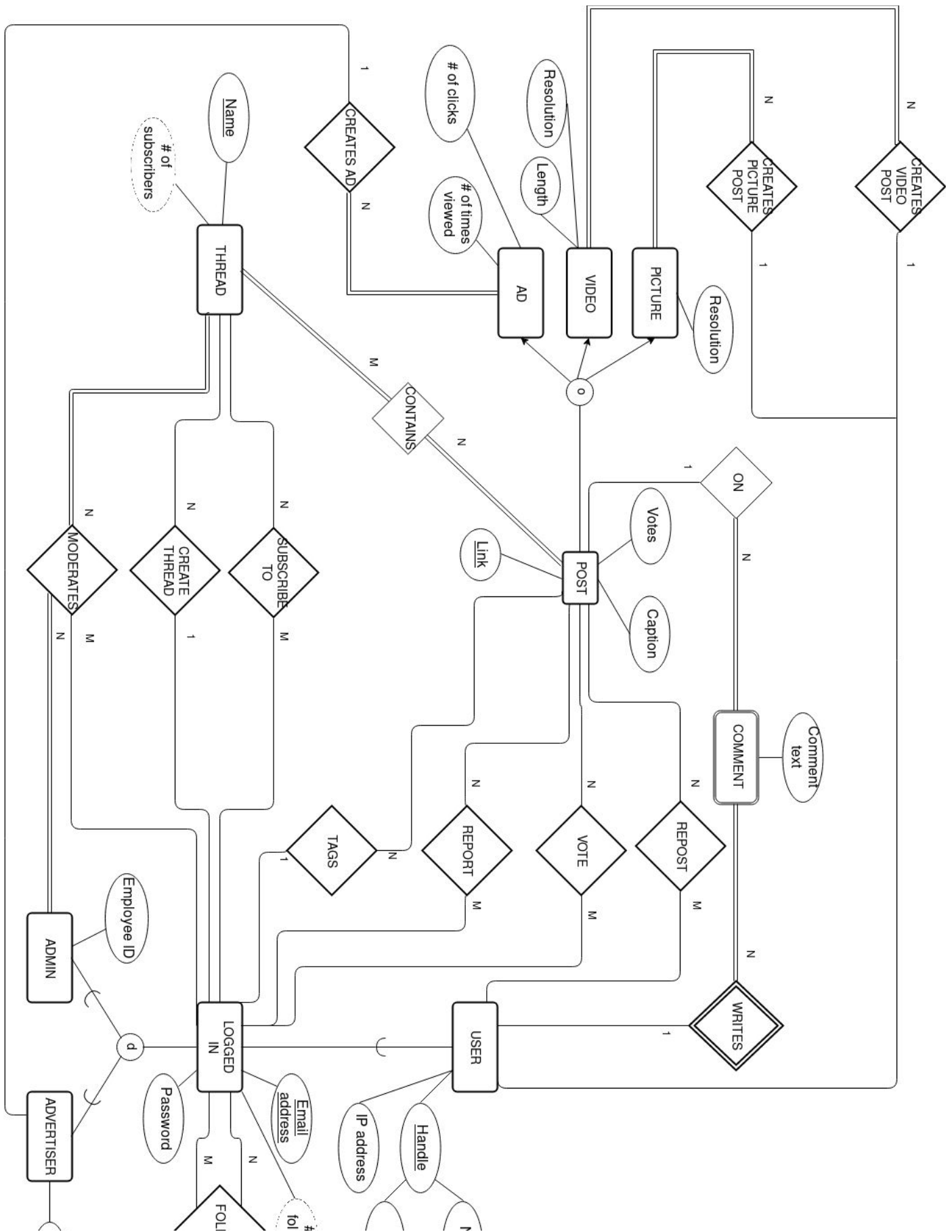
- Anonymous users are able to view posts on threads, create posts on threads, and repost content to other threads.
- Logged in users have all the abilities anonymous users have as well as several more features. They can follow other logged in users and to subscribe to threads, allowing them to see the content posted by these groups. Logged in users have personal threads which is where the content of who their following and their subscribed threads appear. Logged in users can also vote on posts, comment on the posts, report inappropriate posts, and tag posts to more easily find old posts. These users are also able to create new threads and moderate them.
- In the modern online environment advertisers want to seem less corporate. Our design allows for them to have all the abilities of a logged in user, allowing them to interact more closely with consumers. On top of the abilities normal logged in users, advertisers can create ad posts which will have additional information such as the number of times the ad was clicked and the total number of times viewed. These ads were designed so the website would be able to have these ads appear regularly on threads.
- Admins are employees of the website. They are automatically moderators of every thread, including the ability to ban users from threads and delete posts. They can also see the number of flags a post has, and see the most flagged posts.

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)



Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

Above is an updated ER diagram that more closely follows the databases current design. Below are changes were made since the creation of the original entity relationship diagram:

- Extra attributes were added such as 'Company name' for ADVERTISER, 'Employee ID' for ADMIN, and '# of followers' as a derived attribute for LOGGED IN.
- The profile functionality was scrapped because our goal was to not have any unnecessary information on the user and we feel the profile contradicted that goal.

Implementation

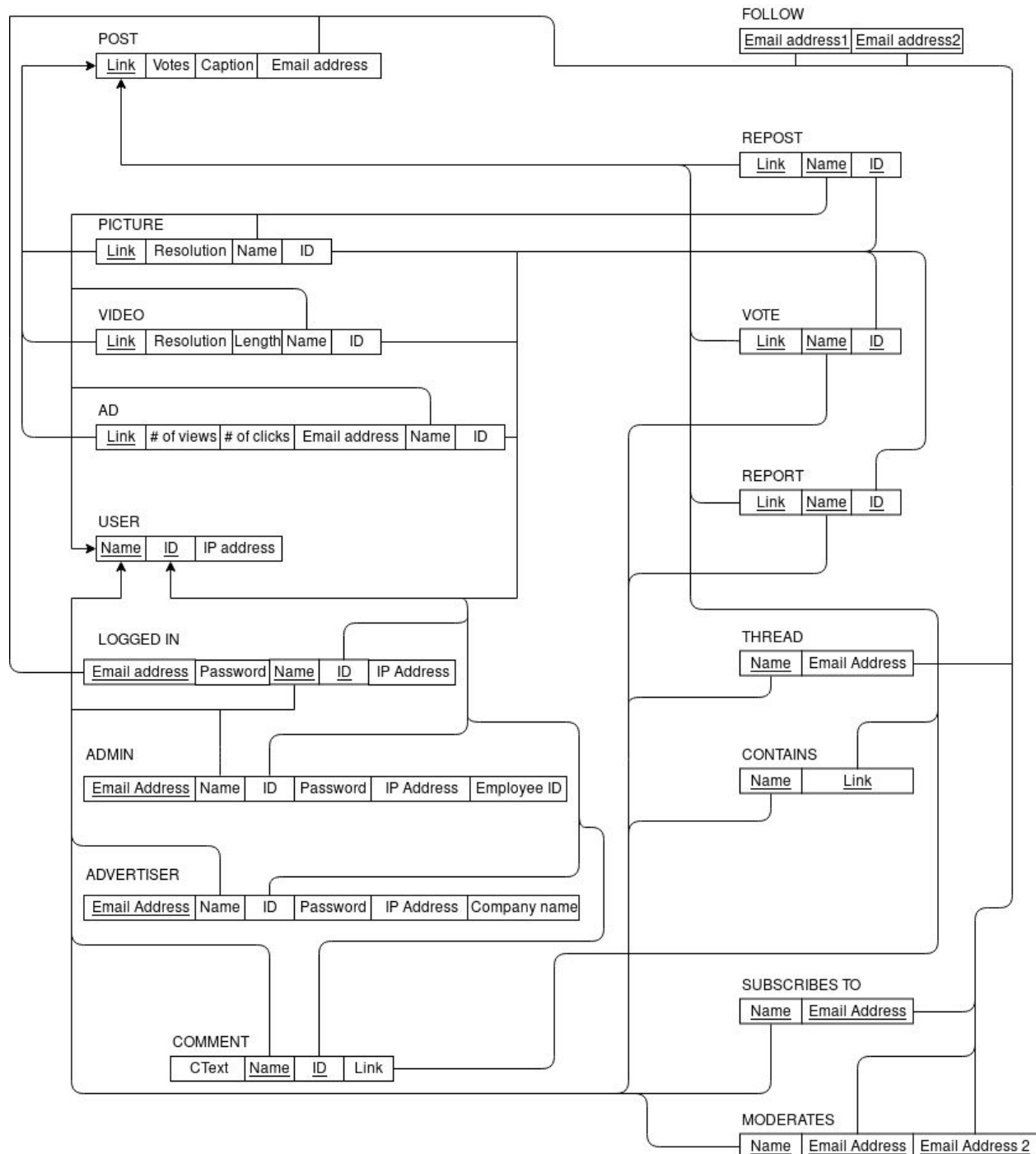
Starting the implementation we converted our ER diagram into a relational diagram to create a more accurate design of the database. The following is the relational diagram based on the above ER diagram.

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)



We made some of the following decisions when creating this diagram:

- We decided to get rid of the second email address in MODERATES because it is redundant
 - Instead, admins capabilities should be implemented in the application logic
- TAGS was eliminated completely due to it being too vague and not having a clear idea on how to implement the function.

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

Database management system

MySQL was used with xampp to create the database management system. Below are queries we used within the system.

SQL Queries

The following are examples of the queries designed for the database:

```
CREATE TABLE USER (
```

```
    Name          CHAR(16)    NOT NULL,
```

```
    ID            INT         NOT NULL,
```

```
    IP_Address    CHAR(45)    NOT NULL,
```

```
    PRIMARY KEY(Name, ID)
```

```
);
```

```
CREATE TABLE LOGGED_IN (
```

```
    Email_address CHAR(320)    NOT NULL,
```

```
    Password      VARCHAR(128) NOT NULL,
```

```
    Name          CHAR(16)    NOT NULL,
```

```
    ID            INT         NOT NULL,
```

```
    IP_Address    CHAR(45)    NOT NULL,
```

```
    PRIMARY KEY(Email_address),
```

```
    UNIQUE(Name, ID),
```

```
    FOREIGN KEY(Name) REFERENCES USER(Name)
```

```
        ON DELETE CASCADE    ON UPDATE CASCADE;
```

```
    FOREIGN KEY(ID) REFERENCES USER(ID)
```

```
        ON DELETE CASCADE    ON UPDATE CASCADE;
```

```
);
```

```
CREATE TABLE ADMIN (
```

```
    Email_address CHAR(320)    NOT NULL,
```

```
    Password      VARCHAR(128) NOT NULL,
```

```
    Name          CHAR(16)    NOT NULL,
```

```
    ID            INT         NOT NULL,
```

```
    IP_Address    CHAR(45)    NOT NULL,
```

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

```
Employee_ID      INT      NOT NULL,  
PRIMARY KEY(Email_address),  
UNIQUE(Name, ID, Employee_ID),  
FOREIGN KEY(Name) REFERENCES USER(Name)  
ON DELETE CASCADE ON UPDATE CASCADE;  
FOREIGN KEY(ID) REFERENCES USER(ID)  
ON DELETE CASCADE ON UPDATE CASCADE;  
FOREIGN KEY(Email_address) REFERENCES Logged_in(Email_address)  
ON DELETE CASCADE ON UPDATE CASCADE;  
);
```

```
CREATE TABLE ADVERTISER (  
Email_address      CHAR(320) NOT NULL,  
Password           VARCHAR(128) NOT NULL,  
Name               CHAR(16) NOT NULL,  
ID                 INT NOT NULL,  
IP_Address         CHAR(45) NOT NULL,  
Company_name       CHAR(48) NOT NULL,  
PRIMARY KEY(Email_address),  
UNIQUE(Name, ID, Company_name),  
FOREIGN KEY(Name) REFERENCES USER(Name)  
ON DELETE CASCADE ON UPDATE CASCADE;  
FOREIGN KEY(ID) REFERENCES USER(ID)  
ON DELETE CASCADE ON UPDATE CASCADE;  
FOREIGN KEY(Email_address) REFERENCES Logged_in(Email_address)  
ON DELETE CASCADE ON UPDATE CASCADE;  
);
```

```
CREATE TABLE POST (  
Link               CHAR(255) NOT NULL,  
Caption            VARCHAR(140),  
Email_address      CHAR(320),  
PRIMARY KEY(Link),  
FOREIGN KEY(Email_address) REFERENCES Logged_in(Email_address)  
ON DELETE CASCADE ON UPDATE CASCADE;  
);
```

```
CREATE TABLE PICTURE (  
Link               CHAR(255) NOT NULL,  
Caption            VARCHAR(140),
```

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

```
Resolution      CHAR(16)    NOT NULL,
Name            CHAR(16)    NOT NULL,
ID             INT      NOT NULL,
PRIMARY KEY(Link),
FOREIGN KEY(Link) REFERENCES POST(Link)
    ON DELETE CASCADE    ON UPDATE CASCADE;
FOREIGN KEY(Caption) REFERENCES POST(Caption)
    ON DELETE CASCADE    ON UPDATE CASCADE;
FOREIGN KEY(Name) REFERENCES USER(Name)
    ON DELETE CASCADE    ON UPDATE CASCADE;
FOREIGN KEY(ID) REFERENCES USER(ID)
    ON DELETE CASCADE    ON UPDATE CASCADE;
);
```

```
CREATE TABLE VIDEO (
    Link          CHAR(255)    NOT NULL,
    Caption       VARCHAR(140),
    Resolution    CHAR(16)    NOT NULL,
    Length        INT      NOT NULL,
    Name          CHAR(16)    NOT NULL,
    ID           INT      NOT NULL,
    PRIMARY KEY(Link),
    FOREIGN KEY(Link) REFERENCES POST(Link)
        ON DELETE CASCADE    ON UPDATE CASCADE;
    FOREIGN KEY(Caption) REFERENCES POST(Caption)
        ON DELETE CASCADE    ON UPDATE CASCADE;
    FOREIGN KEY(Name) REFERENCES USER(Name)
        ON DELETE CASCADE    ON UPDATE CASCADE;
    FOREIGN KEY(ID) REFERENCES USER(ID)
        ON DELETE CASCADE    ON UPDATE CASCADE;
);
```

```
CREATE TABLE AD (
    Link          CHAR(255)    NOT NULL,
    Caption       VARCHAR(140),
    Num_views     INT,
    Num_clicks    INT,
    Email_address CHAR(320)    NOT NULL,
    Name          CHAR(16)    NOT NULL,
    ID           INT      NOT NULL,
```


Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

```
PRIMARY KEY(Link),
FOREIGN KEY(Link) REFERENCES POST(Link)
    ON DELETE CASCADE    ON UPDATE CASCADE;
FOREIGN KEY(Caption) REFERENCES POST(Caption)
    ON DELETE CASCADE    ON UPDATE CASCADE;
FOREIGN KEY(Email_address) REFERENCES Logged_in(Email_address)
    ON DELETE CASCADE    ON UPDATE CASCADE;
FOREIGN KEY(Name) REFERENCES USER(Name)
    ON DELETE CASCADE    ON UPDATE CASCADE;
FOREIGN KEY(ID) REFERENCES USER(ID)
    ON DELETE CASCADE    ON UPDATE CASCADE;
);
```

```
CREATE TABLE THREAD (
    Name          CHAR(16)    NOT NULL,
    Email_address CHAR(320)   NOT NULL,
    PRIMARY KEY(Name),
    FOREIGN KEY(Email_address) REFERENCES Logged_in(Email_address)
        ON DELETE CASCADE    ON UPDATE CASCADE;
);
```

```
CREATE TABLE COMMENT (
    CText          VARCHAR(140),
    Name           CHAR(16)    NOT NULL,
    ID             INT         NOT NULL,
    Link           CHAR(255)   NOT NULL,
    PRIMARY KEY(Name, ID),
    UNIQUE(Link),
    FOREIGN KEY(Link) REFERENCES POST(Link)
        ON DELETE CASCADE    ON UPDATE CASCADE;
    FOREIGN KEY(Name) REFERENCES USER(Name)
        ON DELETE CASCADE    ON UPDATE CASCADE;
    FOREIGN KEY(ID) REFERENCES USER(ID)
        ON DELETE CASCADE    ON UPDATE CASCADE;
);
```

```
CREATE TABLE CONTAINS (
    Name          CHAR(16)    NOT NULL,
    Link           CHAR(255)   NOT NULL,
    PRIMARY KEY(Name, Link),
```

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

```
        FOREIGN KEY(Name) REFERENCES THREAD(Name)
            ON DELETE CASCADE      ON UPDATE CASCADE;
        FOREIGN KEY(Link) REFERENCES POST(Link)
            ON DELETE CASCADE      ON UPDATE CASCADE;
    );

CREATE TABLE SUBSCRIBES_TO (
    Name          CHAR(16)    NOT NULL,
    Email_address CHAR(320)   NOT NULL,
    PRIMARY KEY(Name, Email_address),
    FOREIGN KEY(Name) REFERENCES THREAD(Name)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(Email_address) REFERENCES Logged_in(Email_address)
        ON DELETE CASCADE      ON UPDATE CASCADE;
);

CREATE TABLE MODERATES (
    Name          CHAR(16)    NOT NULL,
    Email_address CHAR(320)   NOT NULL,
    Email_address2 CHAR(320)   NOT NULL,
    PRIMARY KEY(Name, Email_address, Email_address2),
    FOREIGN KEY(Name) REFERENCES THREAD(Name)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(Email_address) REFERENCES Logged_in(Email_address)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(Email_address2) REFERENCES Logged_in(Email_address)
        ON DELETE CASCADE      ON UPDATE CASCADE;
);

CREATE TABLE FOLLOW (
    Email_address1 CHAR(320)   NOT NULL,
    Email_address2 CHAR(320)   NOT NULL,
    PRIMARY KEY(Email_address1, Email_address2),
    FOREIGN KEY(Email_address1) REFERENCES Logged_in(Email_address)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(Email_address2) REFERENCES Logged_in(Email_address)
        ON DELETE CASCADE      ON UPDATE CASCADE;
);

CREATE TABLE REPOST (
```

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

```

    Link            CHAR(255)    NOT NULL,
    Name            CHAR(16)     NOT NULL,
    ID              INT          NOT NULL,
    PRIMARY KEY(Link, Name, ID),
    FOREIGN KEY(Name) REFERENCES USER(Name)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(Link) REFERENCES POST(Link)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(ID) REFERENCES USER(ID)
        ON DELETE CASCADE      ON UPDATE CASCADE;
);
```

```

CREATE TABLE VOTE(
    Link            CHAR(255)    NOT NULL,
    Name            CHAR(16)     NOT NULL,
    ID              INT          NOT NULL,
    PRIMARY KEY(Link, Name, ID),
    FOREIGN KEY(Name) REFERENCES USER(Name)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(Link) REFERENCES POST(Link)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(ID) REFERENCES USER(ID)
        ON DELETE CASCADE      ON UPDATE CASCADE;
);
```

```

CREATE TABLE REPORT (
    Link            CHAR(255)    NOT NULL,
    Name            CHAR(16)     NOT NULL,
    ID              INT          NOT NULL,
    PRIMARY KEY(Link, Name, ID),
    FOREIGN KEY(Name) REFERENCES USER(Name)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(Link) REFERENCES POST(Link)
        ON DELETE CASCADE      ON UPDATE CASCADE;
    FOREIGN KEY(ID) REFERENCES USER(ID)
        ON DELETE CASCADE      ON UPDATE CASCADE;
);
```

```

CREATE TABLE TAG (
    Link            CHAR(255)    NOT NULL,
```

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

```
Tagtext          CHAR(24)    NOT NULL
);
```

Insertions

Create account

```
INSERT INTO      LOGGED IN
VALUES           ('aidan.bjelke@ucalgary.ca', 'hunter2', 'TheLegend27', 4096);
```

Create picture post

```
INSERT INTO      PICTURE
VALUES           ('https://i.kym-cdn.com/entries/icons/mobile/000/029/775/1942143-ajtdeu9r-v4.jpg',
'1280x720', 'TheLegend27', 4096);
```

Create video post

```
INSERT INTO      VIDEO
VALUES           ('https://i.kym-cdn.com/entries/icons/mobile/000/029/775/1942143-ajtdeu9r-v4.jpg',
'1280x720', 123, 'TheLegend27', 4096);
```

Create advertiser

```
INSERT INTO      ADVERTISER
VALUES           ('advertiser@hotmail.com', 'hunter3', 'TheAdvertiser27', 2048, 'lumenary.');
```

Create ad

```
INSERT INTO      AD
VALUES           ('https://i.kym-cdn.com/entries/icons/mobile/000/029/775/1942143-ajtdeu9r-v4.jpg', 0,
0, 'advertiser@hotmail.com', 'TheAdvertiser27', 2048);
```

Create thread

```
INSERT INTO      THREAD
VALUES           ('databases', 'aidan.bjelke@ucalgary.ca')
```

Add 'decor' tag to a post

```
INSERT INTO      TAG
VALUES           ('https://i.kym-cdn.com/entries/icons/mobile/000/029/775/1942143-ajtdeu9r-v4.jpg', 'decor')
```

Write comment

```
INSERT INTO      COMMENT
VALUES           ('follow for daily minecraft posts', 'TheLegend27', 4096,
https://i.kym-cdn.com/entries/icons/mobile/000/029/775/1942143-ajtdeu9r-v4.jpg);
```

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

Repost

INSERT INTO REPOST

VALUES

('https://i.kym-cdn.com/entries/icons/mobile/000/029/775/1942143-ajtdeu9r-v4.jpg', 'TheLegend27', 4096);

Vote on post

INSERT INTO VOTE

VALUES

('https://i.kym-cdn.com/entries/icons/mobile/000/029/775/1942143-ajtdeu9r-v4.jpg', 'TheLegend27', 4096);

Report post

INSERT INTO REPORT

VALUES

('https://i.kym-cdn.com/entries/icons/mobile/000/029/775/1942143-ajtdeu9r-v4.jpg', 'TheLegend27', 4096);

Queries

Show home feed (subscribed threads)

```
SELECT SUBSCRIBES_TO.Name, POST.Caption, CONTAINS.Link
FROM SUBSCRIBES_TO AS SUB, LOGGED_IN AS LI, POST, CONTAINS, THREAD
WHERE SUB.Email_Address = LI.Email_Address
      AND
      SUB.Name = CONTAINS.Name
      AND
      SUB.Name = THREAD.Name
      AND
      CONTAINS.Link = POST.Link
```

Subscribe

INSERT INTO SUBSCRIBES_TO

VALUES ('databases', 'aidan.bjelke@ucalgary.ca');

Restart (Delete all users posts)

```
DELETE FROM (((POST NATURAL JOIN PICTURE) NATURAL JOIN VIDEO) NATURAL
JOIN AD) AS P
WHERE P.Email_address = 'aidan.bjelke@ucalgary.ca'
```

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

Show followers of 'TheLegend27', id = 4096 (shown previously)

```
SELECT      Name, ID
FROM        LOGGED_IN AS U, FOLLOW AS F
WHERE       F.Email_address1 = U.Email_address
           AND
           F.Email_address2 IN (SELECT Email_address
                                FROM LOGGED_IN AS U1
                                WHERE U1.Name = 'TheLegend27'
                                AND U1.ID = 4096);
```

Show all users that 'TheLegend27' follows

```
SELECT      Name, ID
FROM        LOGGED_IN AS U, FOLLOW AS F
WHERE       F.Email_address2 = U.Email_address
           AND
           F.Email_address1 IN (SELECT Email_address
                                FROM LOGGED_IN AS U1
                                WHERE U1.Name = 'TheLegend27'
                                AND U1.ID = 4096);
```

Get all reported posts on a thread

```
SELECT      POST*
FROM        POST, THREAD, CONTAINS, REPORT
Where       THREAD.name = 'sampleName' AND POST.link = CONTAINS.link
           AND CONTAINS.name = THREAD.name AND P.link = REPORT.link
```

Get all a users posts with a tag

```
SELECT      P*
FROM        (((POST NATURAL JOIN PICTURE) NATURAL JOIN VIDEO) NATURAL JOIN
AD) AS P, TAG AS T
WHERE       T.tagtext = 'sampletag'
           AND P.Email_address = 'aidan.bjelke@ucalgary.ca'
           AND P.Link IN T
```

Get all reported posts with a tag

```
SELECT      R*
FROM        REPORT AS R, TAG AS T, POST AS P
WHERE       T.tagtext = 'sampletag' AND T.link = P.link AND P.link = R.link
```

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

Get users and how many reported posts they have

```
SELECT      Name, ID, COUNT(*)
FROM        (USER NATURAL JOIN REPORT)
GROUP BY    NAME, ID
```

Get number of subscribers to a thread

```
SELECT      COUNT(*)
FROM        SUBSCRIBES_TO AS ST
WHERE       ST.Name = 'databases'
```

Get posts with between 100 and 300 votes

```
SELECT      POST*
FROM        POST, VOTE
WHERE       POST.Link = VOTE.Link
GROUP BY    VOTE.link
HAVING      COUNT(*) > 100 AND COUNT(*) < 300)
```

Revisions from previous relational model (reflected in tables above)

- Removed votes attribute in post
- Removed email_address2 from moderates table (this is redundant, it made sense in the ERD but not in the final version- it pertains to admins which “moderate” every thread, but storing a copy for every other moderator is unnecessary)
- Added tag table, effectively making the relation many to many and adding an attribute for tag text (this change makes it fit better with our original idea of tags which is similar to hashtags on Twitter)

User Manual

Using the System

Home page -

- Here the user can view all posts
 - Clicking on a link opens a new tab and takes you to that page
 - I.e. clicking on a YouTube link will open a new page redirecting you to YouTube
- Vote on posts (only positive feedback)
- Create a post
 - The dropdown menu allows users to associate their post with an existing thread

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

- Search for threads
- Log in
- Sign up
- Follow other users if logged in

mysocial
Search For Threads
[Home](#) [Sign up](#) [Log in](#)
Create a post
Caption: Optional
Link: Required
Thread: Optional

Votes	Link	Caption	Thread(s)	Poster (click to follow)	Comments
0	view	destroy all humans 2 gameplay	gaming,	baugle #461	comments
0	view	excited for smm2			comments
1	view	Haha look at this cat		@igBadAdmin #0	comments
0	view	Look at this bubble		Steven #2	comments
1	view	These fires are out of control		Carl #1	comments
0	view	This was a great victory		TheLegend27 #4	comments
0	view	Look at these words.		Aidan #0	comments
0	view	Who need images when you have text		TheLegend27 #4	comments
0	view	Bubbles.		Steven #2	comments
0	view	This is my second post		Aidan #0	comments

Signup page -

- Email is the primary
- Password and username can be whatever the user wants
- After the user is redirected to the logged in home screen

Sign-up

[Home](#) [Log-in](#)
E-mail:
Password:
Username:

Login page -

- Entering the right user credentials will redirect them to the logged in home screen

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

Log-In

[Home](#) [Sign-up](#)

E-mail:

Password:

Logged in main page -

- A feature that an anonymous user can't perform is creating a thread
 - Shown right above the search bar.

mysocial
[CreateThread](#)
Search For Threads
[Following](#) [Log out](#)

Create a post
Caption: Optional
Link: Required
Thread: Optional

Welcome, bungle

Votes	Link	Caption	Thread(s)	Poster (click to follow)	Comments
0	view	Check out our Canadian merch		Carl #3	comments
0	view	Check out our sweet merch	Not Bubbles,	Carl #3	comments

Create A Thread page -

- Here the user can create a thread
- First they need to provide a name for it then they can make one.

Group 6 - mysocial
Aidan Bjelke - 10160269 (T01)
Steven Canon-Almagro - 10155792 (T01)
Carl Lyss - 30025962 (T02)

Creating A Thread

[Home](#)

Thread Name:

Create

Comment page for a post -

- Here the user can anonymously comment on a post, or sign up/ log in and comment under a name of their choosing

Showing comments for image.com/image111

[Home](#) [Sign-up](#) [Log-in](#)

Leave a comment:

Submit

Comment	Name	ID
haha	anon	403

Appendix

Examples of Sample Data Used to Populate the Database

Note: For testing the submission, see the included '*mysocial.sql*' file which can be imported into a local MySQL database. Additionally the '*mysql_connection.py*' Python script can be used to initialize the empty tables, and '*populate.py*' can be used for generating test data from scratch.

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

POST sample data:

	Link	Caption	Email_address
▶	bigcompany.ca	Check out our Canadian merch	bigCompany@company.com
	bigcompany.com	Check out our sweet merch	bigCompany@company.com
	https://www.youtube.com/watch?v=Fg8Ht8W...	OOOOOOOOOOOOOOOOOOOOOOOOOOOOOO...	test@s
	https://www.youtube.com/watch?v=PJielywHIjY	test link	test@s
	https://www.youtube.com/watch?v=sfal2DbZcjs	hockey	NULL
	image.com/image111	Haha look at this cat	important@outlook.com
	image.com/image555	Look at this bubble	bubbleBlower@hotmail.com
	image.com/image658	These fires are out of control	fireman64@gmail.com
	image.com/image789	This was a great victory	someTwo@gmail.com
	page.text1	Look at these words.	someOne@gmail.com
	page.text2	Who need images when you have text	someTwo@gmail.com
	page.text3	Bubbles.	bubbleBlower@hotmail.com
	page.text4	This is my second post	someOne@gmail.com
	page.text5	Don't do anything I wouldn't	important@outlook.com
	smallcorperation.com	Support your local businesses	smallCorp@company.com
	youtube.com/watch/7fhg7g6	So many BUBBLES	someTwo@gmail.com
	youtube.com/watch/asfdsaf	Spoilers for Avengers Endgame	fireman64@gmail.com
	youtube.com/watch/d4was564d	More high quality bubbles	bubbleBlower@hotmail.com
	youtube.com/watch/ds45ad46		important@outlook.com
	youtube.com/watch/r41ew354r	Buy gold!!	fireman64@gmail.com
✱	NULL	NULL	NULL

USER sample data:

	Name	ID	IP_Address
▶	Aidan	0	314
	anon	119	::1
	anon	135	::1
	anon	281	::1
	anon	415	::1
	anon	715	::1
	anon	808	::1
	BigBadAdmin	6	333
	bungle	909	::1
	Carl	1	111
	Carl	3	222
	s	697	::1
	Steven	2	510
	t	127	::1
	TheLegend27	4	333
	TheLegend28	5	333
✱	NULL	NULL	NULL

THREAD sample data:

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

	Name	Email_address
▶	Bubble Blowers	bubbleBlower@hotmail.com
	Not Bubbles	bubbleBlower@hotmail.com
	Fire Stoppers	fireman64@gmail.com
	Empty Thread	someTwo@gmail.com
	Text Thread	someTwo@gmail.com
	hello	test@s
	yellow	test@s
*	NULL	NULL

COMMENT sample data:

	CText	Name	ID	Link
▶	Wow text posts needing links is kinda weird.	Aidan	0	page.text5
	s	anon	119	https://www.youtube.com/watch?v=Fg8Ht8W...
	hope this works	anon	808	https://www.youtube.com/watch?v=Fg8Ht8W...
	my name is jeff	bungle	909	https://www.youtube.com/watch?v=Fg8Ht8W...
	Who cares about bubbles when there are fires?	Carl	1	youtube.com/watch/7fhg7g6
	That really was a bad fire!	Carl	3	image.com/image658
	we are in the endgame now	s	697	https://www.youtube.com/watch?v=Fg8Ht8W...
	yes	s	697	https://www.youtube.com/watch?v=Fg8Ht8W...
	Wow, what cool products	Steven	2	smallcorporation.com
*	NULL	NULL	NULL	NULL

CONTAINS sample data:

	Name	Link
▶	Bubble Blowers	image.com/image555
	Bubble Blowers	youtube.com/watch/7fhg7g6
	Bubble Blowers	youtube.com/watch/d4was564d
	Fire Stoppers	image.com/image658
	Fire Stoppers	youtube.com/watch/r41ew354r
	Not Bubbles	bigcompany.com
	Not Bubbles	image.com/image111
	Not Bubbles	page.text1
	Not Bubbles	youtube.com/watch/asfdsaf
	Text Thread	page.text1
	Text Thread	page.text2
	Text Thread	page.text3
	Text Thread	page.text4
	Text Thread	page.text5
	yellow	https://www.youtube.com/wat...
*	NULL	NULL

VOTE sample data:

Group 6 - mysocial

Aidan Bjelke - 10160269 (T01)

Steven Canon-Almagro - 10155792 (T01)

Carl Lyss - 30025962 (T02)

	Link	Name	ID
▶	https://www.youtube.com/watch?v=Fg8Ht8W...	anon	281
	https://www.youtube.com/watch?v=Fg8Ht8W...	anon	415
	https://www.youtube.com/watch?v=Fg8Ht8W...	anon	715
	image.com/image658	Carl	3
	page.text2	anon	135
	youtube.com/watch/d4was564d	Aidan	0
	youtube.com/watch/d4was564d	Steven	2
•	NULL	NULL	NULL

FOLLOW sample data:

	Email_address1	Email_address2
▶	bubbleBlower@hotmail.com	someOne@gmail.com
	bubbleBlower@hotmail.com	someTwo@gmail.com
	someOne@gmail.com	bubbleBlower@hotmail.com
	someOne@gmail.com	fireman64@gmail.com
	someOne@gmail.com	someTwo@gmail.com
	someTwo@gmail.com	bubbleBlower@hotmail.com
	test@s	test@s
•	NULL	NULL