SCHEMA:

```
CREATE DATABASE IF NOT EXISTS photoshare;
USE photoshare;

CREATE TABLE Users(
        user_id INTEGER AUTO_INCREMENT,
        first_name VARCHAR(100) NOT NULL,
        last_name VARCHAR(100) NOT NULL,
        email VARCHAR(100) UNIQUE NOT NULL,
        birth_date DATE NOT NULL,
        hometown VARCHAR(100),
        gender VARCHAR(100),
   score INTEGER,
        password VARCHAR(100) NOT NULL,
        PRIMARY KEY (user_id)
 );

 INSERT INTO Users (user_id, first_name, last_name, email, birth_date, score, password)
VALUES (-1, "Anon", "Anonymous", "anon@anon.com", 0000-00-00, 0, "anon");

 CREATE TABLE Friends(
        user_id1 INTEGER,
        user_id2 INTEGER,
        PRIMARY KEY (user_id1, user_id2),
        FOREIGN KEY (user_id1)
                REFERENCES Users(user_id)
      ON DELETE CASCADE,
        FOREIGN KEY (user_id2)
                REFERENCES Users(user_id)
      ON DELETE CASCADE,
        CONSTRAINT selfFriend
                CHECK (NOT user_id1 = user_id2)
);

CREATE TABLE Albums(
        albums_id INTEGER AUTO_INCREMENT,
        name VARCHAR(100),
        date DATE,
        user_id INTEGER NOT NULL,
        PRIMARY KEY (albums_id),
        FOREIGN KEY (user_id)
                REFERENCES Users(user_id)
      ON DELETE CASCADE
```

```
);

CREATE TABLE Tags(
        tag_id INTEGER AUTO_INCREMENT,
        name VARCHAR(100),
    num_photos INTEGER,
        PRIMARY KEY (tag_id)
);

INSERT INTO Tags (tag_id, name, num_photos) VALUES (-1, "empty", 0);

CREATE TABLE Photos(
        photo_id INTEGER AUTO_INCREMENT,
        caption VARCHAR(100),
        imgdata LONGTEXT,
        albums_id INTEGER NOT NULL,
        user_id INTEGER NOT NULL,
   num_likes INTEGER NOT NULL,
        PRIMARY KEY (photo_id),
        FOREIGN KEY (albums_id)
                REFERENCES Albums (albums_id)
                ON DELETE CASCADE,
        FOREIGN KEY (user_id)
                REFERENCES Users (user_id)
                ON DELETE CASCADE
);

CREATE TABLE Tagged(
        photo_id INTEGER,
        tag_id INTEGER,
    user_id INTEGER,
        PRIMARY KEY (photo_id, tag_id),
        FOREIGN KEY(photo_id)
                REFERENCES Photos (photo_id)
      ON DELETE CASCADE,
        FOREIGN KEY(tag_id)
                REFERENCES Tags (tag_id)
      ON DELETE CASCADE,
        FOREIGN KEY (user_id)
                REFERENCES Users (user_id)
                ON DELETE CASCADE
);

CREATE TABLE Comments(
```

```
        comment_id INTEGER AUTO_INCREMENT,
        user_id INTEGER NOT NULL,
        photo_id INTEGER NOT NULL,
        text VARCHAR (255),
        date DATE,
        PRIMARY KEY (comment_id),
        FOREIGN KEY (user_id)
                REFERENCES Users (user_id)
    ON DELETE CASCADE,
        FOREIGN KEY (photo_id)
                REFERENCES Photos (photo_id)
    ON DELETE CASCADE
);

CREATE TABLE Likes (
   photo_id INTEGER,
   user_id INTEGER,
   PRIMARY KEY (photo_id , user_id),
   FOREIGN KEY (photo_id)
                REFERENCES Photos (photo_id)
     ON DELETE CASCADE,
   FOREIGN KEY (user_id)
                REFERENCES Users (user_id)
     ON DELETE CASCADE
);
```

Additional Assumptions and Constraints:

- Users cannot upload photos with captions that contain special characters.
- Users cannot create comments with special characters.
- Users cannot create album names with special characters.
- You can upload the same photo
- You must be the owner of a photo to add a tag to it.
- There are no limits to the number of tags on a photo.
- Friendships are one-way
- Different users can create Albums with identical names
- You can’t like a photo if you’re not logged in
- You can search by comment if you’re not logged in
- When commenting on a photo when you’re not logged in, the comment shows up as “Anonymous”
- Comments can’t be longer than 255 characters
- Users’ contribution score is not deducted when photos/albums are deleted