User Accounts and User Homepage

Group 6

Numaan Cheema | cheeman1@tcnj.edu | Integration Lead
Jon Bernstein | bernstj6@tcnj.edu
Sam Bishop | bishops5@tcnj.edu
Matt Perino | perinom3@tcnj.edu

Project Github: https://github.com/CSC-315/cab-socialsrhub-group-6

Executive Summary

- **Need**

The main aspect we identified to be addressed in the srhub and civicstory websites that needs to be improved is giving the users a reason to return to the site regularly. By providing some system that can encourage users to interact with the respective sites daily, or even weekly, it is our belief that the goals of the stakeholders and users will be met.

We meet the stakeholders goals through regular interaction with the website. By encouraging users to come back, we are able to expose them to the sustainability content posted on these sites effectively. By showing users the content on these sites, we hope they will be inspired to get involved or look further into the topics presented.

We also provide users with a way to connect with peers through the website. It is our hope that providing users with a customizable and interactive user page, they will be motivated to return often and interact with the content available on these sites.

- **Approach**

To achieve these goals, we have implemented the beginnings of a fully interactable and customizable user page. To begin with, we have a fully functional account creation and login system updated. This allows a user to create an account with an email address and their information. The credentials are stored within our database so that the user can log in to the website with these details. Additionally, we have implemented a user page where the user can post messages for others to see. Our hope is that in the future, users will be able to "follow" other users, giving these pages a more defined purpose. Additionally, in the future, we would like to allow users to post articles on their page. Currently, it is possible to link an article on a user page but we would like to utilize a more integrated approach.

- **Benefits**

    We believe our implementation will raise awareness on sustainability-related issues in the communities throughout New Jersey. Through this awareness, we hope more people will take action into addressing this issue, thus improving the communities they live in for everyone that lives there.

    We also believe our system will help the civicstory and srhub websites by increasing regular traffic to their sites. It is their goal also to raise awareness of sustainability-related news. By encouraging users to return to the site regularly, it is our goal to expose them to this content more regularly, as well as exposing them to a wider range of sustainability news than they may have previously explored.

- **Cost**

    During implementation, it was one of our main goals to make this as cost-effective in terms of implementation as possible. We've designed the system to be able to stand on its own as an extension of the current system. However, with a little integration into the existing website, the system can be brought to its full potential. Integration would allow users to post articles on their page more easily and would enhance the user experience overall.

Project Proposal and Specifications

- **Problem statement**

The main issue we aim to solve involves the growth of the srhub.org and civicstory.org websites. The main issue we've been able to identify in the srhub.org and civicstory.org websites is the lack of community building. For a site so focused on events happening within communities for a good cause, the site lacks the ability to have people connect with one another and share content on topics that they hold an interest in. Another issue we've identified is a lack of reasons for people to keep returning to these websites.

- **Objective of our proposed module**

We aim to tackle the issue of community building by adding additional functionality that we believe will enhance the user experience, overall. By creating user pages, and allowing users to interact with other user pages (by allowing a user to set their pages to public, giving anybody the ability to see them or private, allowing only permitted individuals to see their page), we believe that we can allow users to build communities within this domain.

We believe our proposed module will also address the issue of reasons to return to the website by allowing users to see content that may be outside of their immediate realm of interest. By exposing users to new and interesting content, we believe that all users will have a reason to return to the sites often.

- **Description of the desired end product**

Our vision for the module we aim to develop is similar to that of a user's page on common social media sites, such as Facebook or Twitter. The end product will function similarly to these sites, allowing users to see content that users they "follow" or prominent figures within their communities are reposting or commenting on. We also would aim to recommend content to users, based on previous things they've looked at, potentially furthering their scope of interaction with the site.

As far as the module we will develop for class goes, our main priorities center around the setup for the user page, allowing users to favorite content they find interesting or useful, and potentially recommending additional content to users. Stretch goals include allowing users to see other user pages, and commenting and reposting posts

- **Description of the importance and need for the module and how it addresses the problem**

We believe that development of the proposed module will ultimately increase traffic to srhub.org and civicstory.org, allowing for rapid growth of these websites. The ability for people to share things that they find important has proven useful for the growth of social media sites over the past few years and we think a similar principle can be applied here.

- **Plan for how you will research the problem domain and obtain the data needed**

We've already performed some research into the problem domain by browsing other sites with similar goals and looking for features we found particularly useful or features that improved the user experience. In order to obtain the data needed, we will very likely use artificial data modeled after the content available on srhub.org and civicstory.org

- **Other similar systems / approaches that exist, and how your module is different or will add value to the existing system**

Obviously, popular social media sites such as Facebook, Twitter, and LinkedIn were our main inspiration for development of our proposed module. Our module is unique in that it focused on the niche srhub.org and civicstory.org, which aims to provide users with resources and information relating to sustainability in New Jersey. We believe our module will add value to the site by allowing users to form connections through the context of these websites.

- **Possible other applications of the system (how it could be modified and reused)**

Our system will provide a template for other small sites to implement similar systems. As previously mentioned, the main goal of our system is to allow users to build communities within the site and we think this can be applied to other sites with similar goals.

- **Performance**

We will aim to maximize performance by performing as few queries as we can while accomplishing the functionality proposed. We believe this will be the main source of performance enhancement within the system. We are also researching alternative ways to maximize performance within this context.

- **Security**

While we believe that the nature of the proposed system is relatively secure, we will provide some features that allow for additional security for the user and the system. For users, we will allow them to specify to what extent they want their page to be accessible by other users. For users who want only specified (or no one) to be able to view the content on their page, we will provide an option to make their page private. This function will allow the user to moderate who is able to see the information on their page.

In terms of system security, we will be parsing all user input to ensure that none of the common mistakes that applications like this have will appear in this system. One of the most common problems with systems that utilize a database is SQL injections, and parsing input along with a whitelist of accepted input will largely eliminate this threat.

- **Backup and recovery**

In determining the most efficient way to ensure our system will remain backed up and recoverable is establishing what we need to be backing up. Each user will have a user profile that contains their personal information. This will be our main focus, preserving user data with the database. We will utilize logical backups, a system used in our DBMS. The server will save each user profile in a file within our VM, ensuring that if the original database fails, we have user information protected. We can automate this process within SQL, making backup and recovery a part of our system from very early on. Since we do not anticipate having tons of users data, along with other data, we will be able to store that data within the VM. If the amount of data we use exceeds that, we can switch over to hard disk data that is automatically updated over a set period of time.

- **Technologies and database concepts the team will need to learn, and a plan for learning these**

The technology and concepts we will need is going to be the hardest task to accomplish. We are going to need to understand back end database development and how to efficiently store this data. This will be done through class along with supplementary work such as a deep dive into SQL and PHP. On the front end, we are going to need to familiarize ourselves with CSS and HTML to keep the user experience clean and easy to understand. In order for users to receive data similar to their favorites and other users they follow, some information retrieval and algorithms work is going to be used as well. The overall technology we are going to have to understand is how social networks work, front end and back end.

•

Social Sustainability

Jon Bernstein, Matt Perino, Numaan Cheema, Sam Bishop

Objective

Build a community that allows users to connect with each other within the sustainability sector in New Jersey

Provide srhub.org and civicstory.org a way to connect with their users in a more technologically advanced way

Develop an easier way for srhub.org and civicstory.org to identify the leading problems in communities and the general populations opinions on sustainability in New Jersey

Building a Community

Approach

Create a community around shrub.org and civicstory.org that allows users to interact with each other by:

1. Create a social media platform that allows users to create a profile and interact with other users
2. Recommend content based on user profiles and activity within the platform
3. Allow users to share articles, ideas, opinions, and videos within the platform
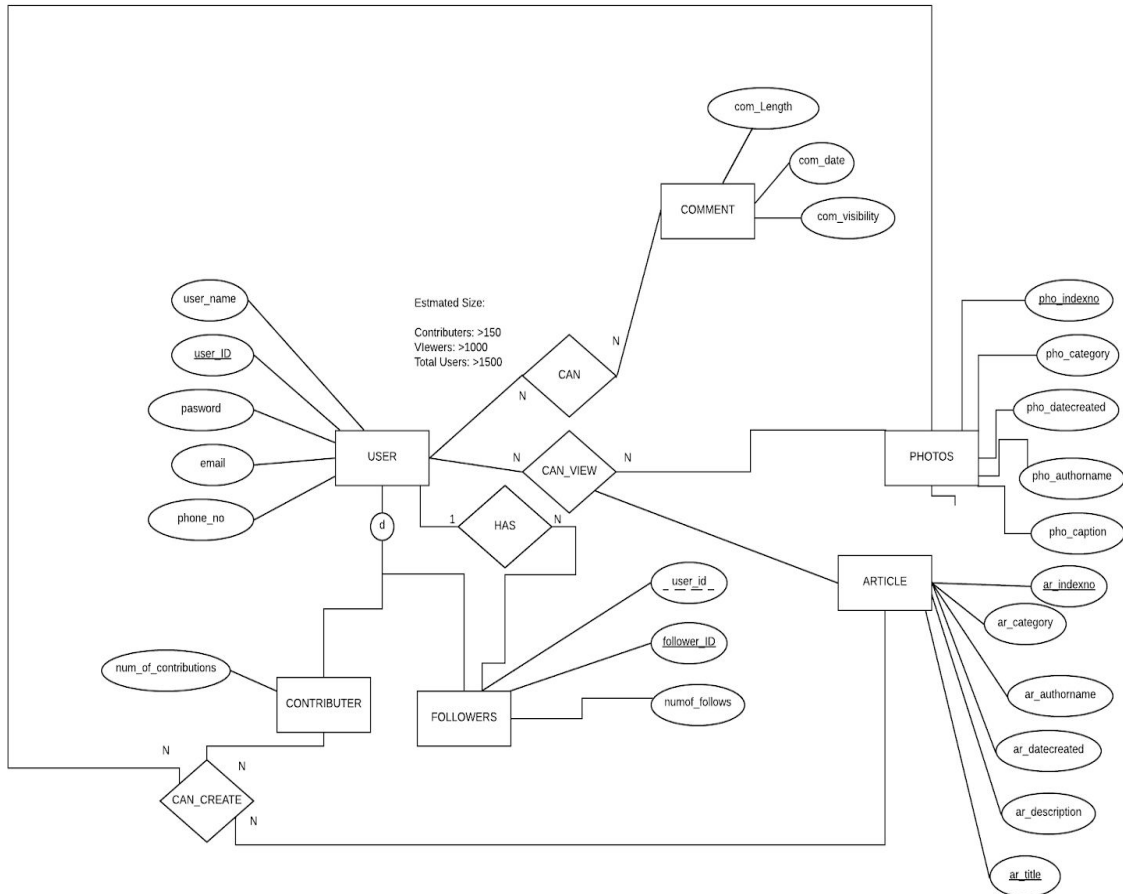
Key Milestones

• Project Specifications	02/24/20
• Database Model	03/09/20
• Overall Design	03/26/20
• Implementation of GUI and DB	04/27/20
• Final System Report	05/04/20

02/12/20

Database Model

- ## ● ER Diagram

Diagrammatic representation of the database architecture.

● **Relational Schema**

USER

user_name	user_ID	password	email	phone_no

ARTICLE

ar_category	ar_datecreated	ar_authorname	ar_description	ar_title	ar_indexno

PHOTO

pho_category	pho_datecreated	pho_authorname	pho_description	pho_title	pho_indexno

CONTRIBUTER

num_of_contributers

FOLLOWERS

user_id	numof_followers	follower_id

COMMENT

com_length	com_date	com_visbility
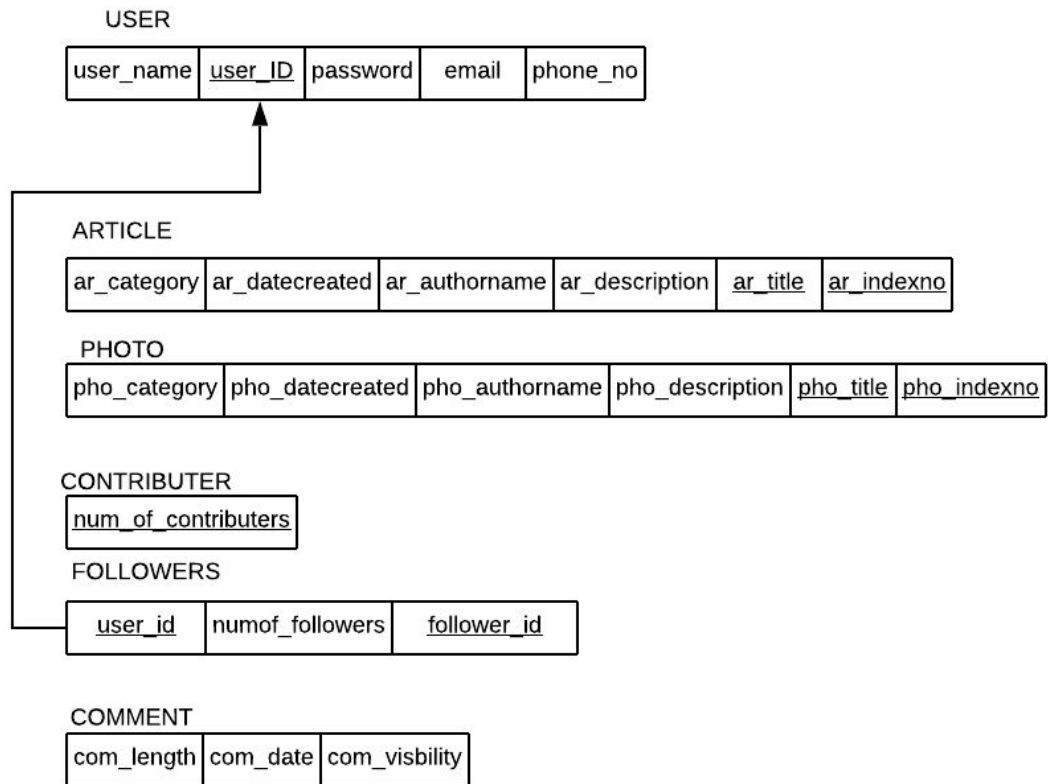
Design

- **Demonstrate that all relations in the relational schema are normalized to Boyce-Codd normal form (BCNF).**
    - For each table, specify whether it is in BCNF or not, and explain why.
    - For each table that is not in BCNF, show the complete process that normalizes it to BCNF.

USER

user_name	user_ID	password	Email	Phone_no

Satisfies the first three normalizations because:

1. All values are atomic, columns are unique, all data in each column is of the same type
2. There are no partial dependencies
3. There are no transitive decencies

It does also satisfy BCNF because:
user_ID → user_name
user_name → {password, email, phone_no}

ARTICLE

ar_category	ar_datecreated	ar_authorname	ar_description	ar_title	ar_indexno

Satisfies the first three normalizations because:

1. All values are atomic, columns are unique, all data in each column is of the same type
2. There are no partial dependencies
3. There are no transitive decencies

It does also satisfy BCNF because:
ar_title → {ar_datecreated, ar_authorname}
ar_indexno → {ar_category, ar_description}

PHOTO

pho_category	pho_datecreated	pho_authorname	pho_description	pho_title	pho_indexno

Satisfies the first three normalizations because:

1. All values are atomic, columns are unique, all data in each column is of the same type
2. There are no partial dependencies
3. There are no transitive decencies

It does also satisfy BCNF because:
pho_title → {pho_datecreated, pho_authorname}
pho_indexno → {pho_category, pho_description}

CONTRIBUTOR

num_of_contributor

Satisfies BCNF because:

1. All values are atomic, columns are unique, all data in each column is of the same type
2. There are no partial dependencies
3. There are no transitive decencies
4. For every functional dependency is dependant on a super key

FOLLOWERS

user_id	numof_followers	follower_id

Satisfies the first three normalizations because:

1. All values are atomic, columns are unique, all data in each column is of the same type
2. There are no partial dependencies
3. There are no transitive decencies

It does also satisfy BCNF because:
user_id → follower_id
follower_id → numof_followers

COMMENT

com_length	com_date	com_visibility	com_index

Satisfies the first three normalizations because:

4. All values are atomic, columns are unique, all data in each column is of the same type
5. There are no partial dependencies
6. There are no transitive decencies

It does also satisfy BCNF because:
com_index → {com_length, com_date, com_visibility}

- **Define the different views required. For each view, list the data and transaction requirements. Give a few examples of queries, in English, to illustrate.**

For our project, we'll only need a few views, with more being required as we move towards our stretch goals for the semester.

For our user page, we'll need a view for the articles a user has liked as well as posts they choose to share on their page.

For the liked articles view, we'll need to ensure that the database keeps track of which articles are "liked" by a user, so that we do not allow the user to like the same article twice. We also need to do this to allow a user to revoke their like for a particular article. Additionally, we'll need

to ensure that these views are user-specific. For example, we don't want any user to be able to influence the liked articles of another user.

The view for shared posts is a little more complicated. We'll need to keep track of the articles themselves, as well as other users' comments on these posts. We'll need to parse input to some extent to ensure there's no abuse of the input features provided to users. We will also consider putting restraints on comments, such as character limits and limiting how many comments a user can post on 1 article.

An example of the logic for the query that handles users liking a post follows: First, we need to make sure that the user has not liked this article previously. We do this by matching the article name, or some primary key for that article. If the article has already been liked, we enter a new flow that will remove that article from the liked articles. If the article has not been liked before, we will then enter the element into the users liked articles.

- **Complete set of queries to satisfy the transaction requirements**

For liking an article:
SELECT ar_title
FROM ARTICLE
where ar_title == selection
INSERT INTO LIKED_ARTICLES
VALUES(selection)

For commenting:
WHEN
      comment_text != blacklisted_words THEN INSERT INTO COMMENT(comment_text)
                                        VALUES(textbox)
      comment_text == blacklisted_word THEN exit
END;

To follow a user:
SELECT user_id
FROM USER
Where user_id == selection
INSERT INTO FOLLOWERS(follower_id)
VALUES(selection)

To like a photo:
SELECT photo_indexno
FROM PHOTO
Where photo_indexno == selection

INSERT INTO LIKED_PHOTOS(photo_indexno)
VALUES(selection)

To create an article:
CREATE article_no
FROM ARTICLE
Where article_title == NEW
INSERT INTO ARTICLES

To post a photo:
SELECT user_id
FROM USER
INSERT INTO PHOTOS(pho_authorname)
VALUES(user_id)

Tables and Queries

Insert commands shortened for the sake of the report.

```
CREATE TABLE P_USER
(user_name VARCHAR(250),
user_ID VARCHAR(250) NOT NULL,
password VARCHAR(250),
email VARCHAR(250),
phone_no VARCHAR(250),
PRIMARY KEY (user_ID)
);
CREATE TABLE ARTICLE
(ar_category VARCHAR(250),
ar_datecreated VARCHAR(250),
ar_authorname VARCHAR(250),
ar_description VARCHAR(250),
ar_title VARCHAR(250) NOT NULL,
ar_indexno VARCHAR(250) NOT NULL,
PRIMARY KEY (ar_title, ar_indexno)
);
CREATE TABLE PHOTO
(pho_category VARCHAR(250),
pho_datecreated VARCHAR(250),
pho_authorname VARCHAR(250),
pho_description VARCHAR(250),
pho_title VARCHAR(250) NOT NULL,
pho_indexno VARCHAR(250) NOT NULL,
PRIMARY KEY (pho_title, pho_indexno)
);
CREATE TABLE CONTRIBUTOR
(num_of_contributor VARCHAR(250) NOT NULL,
PRIMARY KEY (num_of_contributor)
);
CREATE TABLE FOLLOWERS
(user_ID VARCHAR(250) NOT NULL,
numof_followers VARCHAR(500),
follower_id VARCHAR(250) NOT NULL,
PRIMARY KEY (user_id, follower_id)
FOREIGN KEY (user_id) REFERENCES p_user(user_id)
);
CREATE TABLE COMMENT
(com_length VARCHAR(250),
```

```
com_date VARCHAR(250),
com_visibility VARCHAR(250),
com_index VARCHAR(250) NOT NULL,
PRIMARY KEY (com_index)
);
```

For liking an article: SELECT ar_title FROM ARTICLE
where ar_title == selection INSERT INTO LIKED_ARTICLES
VALUES(selection)
For commenting: WHEN
END;

comment_text != blacklisted_words THEN INSERT INTO COMMENT(comment_text)
VALUES(textbox) comment_text == blacklisted_word THEN exit

To follow a user:
SELECT user_id FROM p_USER
Where user_id == selection
INSERT INTO FOLLOWERS(follower_id) VALUES(selection)
To like a photo: SELECT photo_indexno FROM PHOTO

Where photo_indexno == selection
INSERT INTO LIKED_PHOTOS(photo_indexno) VALUES(selection)

To create an article:
CREATE article_no FROM ARTICLE
Where article_title == NEW INSERT INTO ARTICLES

To post a photo:
SELECT user_id FROM p_USER
INSERT INTO PHOTOS(pho_authorname) VALUES(user_id)

INSERT INTO article
VALUES ('Pede Et Risus Ltd','19-07-01','Myles Cole','vulputate, posuere','Tanner, Christen J.',2),
('Natoque Penatibus Associates','19-10-03','Chase Blair','conubia nostra, per inceptos hymenaeos. Mauris ut quam vel sapien imperdiet ornare. In','Parks, Imelda O.',3),
('Euismod Associates','20-11-09','Mohammad Madden','nec,','Dillon, Gwendolyn S.',4);

INSERT INTO comment

VALUES
(16,'20-03-27','Visible',2),(12,'20-10-17','Visible',3),(15,'21-01-19','Visible',4),(22,'20-01-08','Visible',5),(19,'19-11-15','Visible',6),(50,'21-02-03','Visible',7),(36,'19-07-16','Visible',8),
(37,'19-05-09','Visible',9),(40,'21-02-13','Visible',10),(11,'21-02-24','Visible',11),(21,'20-11-05','Visible',12),(42,'19-06-01','Visible',13),(21,'20-08-08','Visible',14),(2,'20-04-27','Visible',15),
(46,'21-02-16','Visible',16),(37,'19-10-14','Visible',17);

INSERT INTO p_user
 VALUES ("Barrett
Jimenez","47816","SXW19BKG8MS","ut.aliquam.iaculis@metusAliquam.net","1-375-282-0232")
,
 ("Ryder Winters","45689","FPT41MOA3DC","eleifend@sitametrisus.co.uk","1-803-469-1678"),
 ("Kennan Lawrence","20597","YVR71QKS4DI","eleifend@Aliquam.net","1-182-710-3313"),
 ("Malcolm
Madden","35398","RDE46FCD6FM","nisl.elementum.purus@Duisgravida.co.uk","1-983-730-19
77"),
 ("Abdul
Petty","97424","BBE43NIO8DO","fringilla.ornare@mifelisadipiscing.edu","1-680-112-3163"),;

 INSERT INTO photo
 VALUES ('Tincidunt Tempus Risus Inc.','19-07-04','Odysseus Durham','bibendum.','Odom,
Lane A.',2),('Lectus Justo Limited','19-08-23','Christopher Wall','malesuada vel, convallis
in,','Mcintyre, Adele W.',3),('Tincidunt Foundation','20-09-28','Ethan Wolfe','dis parturient
montes,','Mccullough, Nehru F.',4),('Natoque Penatibus Et Consulting','19-12-01','Kaseem
Rowe','ipsum','Sykes, Lani T.',5),
 ('Consequat Inc.','19-12-26','Jasper Peck','Nulla tincidunt, neque','Arnold, Hedy O.',6),('Vel
Pede Institute','20-10-19','Stephen Hammond','in','Justice, Kevin Y.',7);