**This revision date: 9/25 revision time: 7:00 pm**

CEN4010 – Principles of Software Engineering. Fall 2021

Milestone 1 – Project Proposal.

Group 10 – Emanon.

Maxon Corvil [mcorvil2018@fau.edu](mailto:mcorvil2018@fau.edu),

Ashley Davis [davisashley2020@fau.edu](mailto:davisashley2020@fau.edu), Patrick Messina [pmessina2017@fau.edu](mailto:pmessina2017@fau.edu),

Michael Niebauer [mniebaue@fau.edu](mailto:mniebaue@fau.edu), Andrew Sexton [asexton2020@fau.edu](mailto:asexton2020@fau.edu)

September 28, 2021

Possible project names: Chitter-Chatter, Pigeon Post, Relay, Emanon

Initial project overview/summary

This project is a social media site that provides high-quality connections between its users. Intended as a method of keeping in touch and maintaining a human connection between friends and family when physical contact is not possible.

**Some Details:** Users are able to foster a more in-depth connection than other social media sites through live chatrooms with their friends that allow them to have real-time conversations whenever they feel the need to talk. If users simply want to share something with people they know, they can make posts about whatever they need to share. Users should also feel comfortable using the site with privacy levels that let users specify who is able to see their posts.

**Chatrooms**– text chat minimum, voice chat/calling as a stretch goal. Should be private, and support a few people in one room.

**Posts** – short “tweet” style posts as they default, optional checkbox or some other method to expand the post length. Expanded posts have additional Teaser field where users can specify a preview to appear on other user’s feeds/pages. If not specified, longer post is truncated.

**Privacy levels** – A minimum of 3 privacy levels can be specified for a post:

* public, where anyone can see a post,
* private/friends – where only other users that are at the friend level or higher can see the post,
* extra-private/best friends – where only best friends/privileged friends can see this post.

2 users become friends after one user send the other a friend request, and the recipient accepts it. Each user is able to set the “friendship level” of each of their friends to either:

* normal / friend – default friendship level for 2 users that have become friends through an accepted friend request
* privileged / best friend – higher friendship level that can see all posts.

Once 2 users are friends, each one can independently set the friendship level for the other. For example, users A and B just accepted a friend request, by default, user A can see user B’s public and private posts, but not extra private posts (same for user B seeing user A’s posts). If user A raises B’s friendship level to privileged, user B can now see all of user A’s posts, but user A can still only see user B’s public/private posts. User B does not receive any indication that user A has raised their privacy level (or lowered it), and user B has no indication when looking at one of A’s posts whether it is a public, private, or extra-private posts. Thus, user B (or any user) has no way of knowing what user A has set for their privacy level.

1. **Executive Summary -Andrew**
2. **Competitive Analysis -Andrew**
3. **Data Definitions -Ashley**
4. **Overview, Scenarios, and Use Cases -Maxon / Patrick / Ashley**

**Use Case – Homepage:**

The user goes to the home page and wants to see global posts and has not logged on or created an account.

1. **Description:** Use case describes the first thing that the user sees when going to the website.

2. **Actors**:

2.1 existing User or new user

2.2 System

3. **Preconditions**:

3.1 User has an active internet connection

4. **Primary Flow of Events**:

1. User arrives on web page.

2. User has access to public / community posts

4. user can either create an account or login from this page

5. **Alternate Flows**

5.1 **If the user attempts to interact with posts**

If user tries to interact with posts, they will be forced to either login or create an account.

1. Web site notifies user that they need an account to proceed

2. Return user back to previous post

**Use Case – sign Up:**

When a user goes to the homepage, they will be shown public posts but they will need to login or create an account if they want to interact with post or create a post. If they don’t have an account they can click on sign up to create a new account.

1. **Description:** Use case describes the process of the user creating an account.

2. **Actors**:

2.1 new user

2.2 System

3. **Preconditions**:

3.1 User has an active internet connection.

3.2 user has an email not connected to another account.

4. **Primary Flow of Events**:

4.1 User arrives on web page.

4.2 User clicks on create account

4.3 user enters an email, password, and username

5. **Alternate Flows**

5.1 **If the user attempts to use an email that is already used**

If user tries to use an existing email

1. Web site notifies user that the email Is already being used

2. gives user the option to change email or login.

**Use Case – Login:**

When a user goes to the homepage, they will be shown public posts, but they will need to login or create an account if they want to interact with post or create a post. If they have an account, they can just sign on with their credentials.

1. **Description:** Use case describes the process of the user logging into an account.

2. **Actors**:

2.1 existing user

2.2 System

3. **Preconditions**:

3.1 User has an active internet connection.

3.2 user has an existing email connected to an account.

3.3 User isn’t already logged in

4. **Primary Flow of Events**:

4.1 User arrives on web page.

4.2 User clicks on sign in

4.3 user enters an email and password

4.4 user is taken to their personal feed

5. **Alternate Flows**

5.1 **if the entered credentials aren’t found on the database**

In this situation, the user entered credentials that aren’t found on the database.

1. Web site notifies user that one or both of the fields need to be changed

2. gives user the option to reenter the credentials or to create an account.

**Use Case – Create short Post:**

After a user has already verified their account, they get the ability to create posts, when creating a post, the user can select post type (long or short if under 200 characters). Additionally, user can add images or videos. Finally, user gets the option of who can view the post so. It can be either public, friends only, or selected individuals/close friends.

1. **Description:** Use case describes the process of the user creating a post, customizing it and selecting who can view it.

2. **Actors**:

2.1 existing user

2.2 System

3. **Preconditions**:

3.1 User has an active internet connection.

3.2 user has already logged in

4. **Primary Flow of Events**:

4.1 user is already logged on

4.2 User clicks on create new post

4.3 User customizes post and selects the privacy settings.

5. **Alternate Flows**

5.1 **if the post is over 200 characters**

In this situation, the user enters over 200 characters. Website notifies user that the post has become a long post and they can either send the post as is, delete it or change it into a long post

1. If this is a long post, the user has to option to create a snippet/ teaser for the post and there is no cap for the length of the post.
2. If the user wants to post as is then the post will be sent as a short post
3. If user deletes the post then nothing will be saved or posted

**Use Case – Create Long Post:**

After a user has already verified their account, they get the ability to create posts, when creating a post, the user can select post type (long or short if under 200 characters). Here, the user selects “Long Post”. Additionally, user can add images or videos. Finally, user gets the option of who can view the post so. It can be either public, friends only, or selected individuals/close friends.

1. **Description:** Use case describes the process of the user creating a Long post, customizing it and selecting who can view it.

2. **Actors**:

2.1 existing user

2.2 System

3. **Preconditions**:

3.1 User has an active internet connection.

3.2 user has already logged in

3.3 User has surpassed 200 characters or user has already opted for a long post

4. **Primary Flow of Events**:

4.1 user is already logged on

4.2 User clicks on create a new long post or surpasses 200 characters

4.3 User is given the option to create a personal snippet and does.

4.3 the post will appear as a short post with the option to be expanded and if expanded it will take you to the full post.

5. **Alternate Flows**

5.1 **if the user opts not to create a custom snippet.**

In this situation, the user opts to not create a snippet for their long post. Website notifies user that the snippet will be automatically the first 200 characters either accept or change the snippet.

1. If user accepts then the first 200 characters, then will be the snippet
2. If the user declines, then they can create their own teaser

5.2 **if the user cancels the post**

In this situation, the user opts to not create a post at all. Website asks if they are sure and they can delete the post permanently or send it to drafts.

1. If user deletes the post, then they will be returned to dashboard
2. If the user saves to drafts , then it will be saved in the data base but not viewable by anyone.

**Use Case – search:**

After a user has already logged into their account, they get the ability to search. The search is one search and will return users and public posts with the query inside of the post.

1. **Description:** The search function allows the user to find other users based on username and posts. One search will return both if these results on the same page and the user can interact with the results.

2. **Actors**:

2.1 existing user

2.2 System

3. **Preconditions**:

3.1 User has an active internet connection.

3.2 user has already logged in

3.3 user has a query in mind

4. **Primary Flow of Events**:

4.1 user is already logged on

4.2 User enters a search and sends it in

4.3 the search returns the results and user can interact with them

5. **Alternate Flows**

5.1 **if the user search returns no results**

In this situation, the users search returned no results. So there won’t be anything for them to interact with.

1. The User is shown a “no results” screen
2. The user can either search again or return to their personal feed.

**Use Case – Add friend/ follow:**

After a user has already logged into their account, they get the ability to follow/ add other users. This will populate a user’s personal feed with the post of users they follow/add and give them the ability to interact with the other person.

1. **Description:** Adding a user can be done from either the search screen or from the public post screen and it adds the users who you are following’s posts to your personal feed.

2. **Actors**:

2.1 existing user

2.2 Another existing user

2.3 System

3. **Preconditions**:

3.1 User has an active internet connection.

3.2 user has already logged in

3.3 user has found another user

4. **Primary Flow of Events**:

4.1 user is already logged on

4.2 User finds another user they want to add/ follow

4.3 the followed users posts get added to the users personal feed.

5. **Alternate Flows**

5.1 **if the user already added/followed**

In this situation, the two users are already friends. If this happens then the system alerts the user of this and they can either return or remove that person as a friend.

1. The User is alerted by the system to tell them that they are already friends

**Use Case – Like/ comment on a post:**

When a post is on a feed the user has the option to like. Or add a comment. This requires that the user is already logged into their account. When they like the count will go up with an animation and if they comment then their comment will appear under the post.

1. **Description:** This feature will create the feeling of community as it gives people the ability to interact with others posts. They will also have the option to react with preselected emojis.

2. **Actors**:

2.1 existing user

2.2 Another existing users post

2.3 System

3. **Preconditions**:

3.1 User has an active internet connection.

3.2 user has already logged in

3.3 user found a post they want to like or comment on

4. **Primary Flow of Events**:

4.1 user is already logged on

4.2 User finds another users post

4.3 the likes on the post go up by 1 if they like it and the comment appears as the newest comment if they comment.

5. **Alternate Flows**

5.1 **if the user deletes or unlike the post**

In this situation, the user either deletes their comment or unlike the post.

1. The User is alerted by the system question if they are sure
2. If they proceed then the total likes will go down by 1 or the comment under the post is removed

**Use Case – Chat :**

Users have the ability to chat with other users. This requires that the user is already logged into an existing account. And all chats are saved on the database

1. **Description:** This feature will create the feeling of community as it gives people the ability to message each other. Chats will be delivered in real time but for users security only friends can message each other.

2. **Actors**:

2.1 existing user

2.2 Another existing user

2.3 System

3. **Preconditions**:

3.1 User has an active internet connection.

3.2 user has already logged in

3.3 user is friends with the user they want to message

4. **Primary Flow of Events**:

4.1 user is already logged on

4.2 User is already friends with the other user

4.3 The sent message appears on the screen and sent messages will appear as a stack with newest being closest to the bottom.

5. **Alternate Flows**

5.1 **if the user tries to message someone who is not their friend**

In this situation, the user attempts to message someone who is not on their friend list.

1. The User is alerted by the system with a message that states they need to be friends first.
2. The message also gives them the ability to send a friend request right there
3. Once the friend request is accepted the user can now message them.
4. **Initial List of High-level Functional Requirements -Everyone**
5. **List of Non-functional Requirements -Everyone**
6. **High-level System Architecture -Michael**
7. **Team Roles**
   1. **Team lead, Github Manager, Scrum Master, Front & Back End support**Andrew Sexton
   2. **Front-end developers**

Maxon Corvil

Patrick Messina

* 1. **Back-end developers**

**Back-end lead:** Michael Niebauer

Ashley Davis

1. **Checklist**

|  |  |
| --- | --- |
| Team decided on basic means of communication | **X** |
| Team found a time slot to meet outside of the class | **X** |
| Front and back end team leads chosen |  |
| Github master chosen | **X** |
| Team ready and able to use the chosen back and front end frameworks |  |
| Skills of each team member defined and known to all | **X** |
| Team lead ensured that all team members read the final M1 and agree/understand it before submission |  |