CEN 4010 Principles of Software Engineering

Fall 2021

Milestone 3 Detailed Project Proposal and Vertical Prototype

LikU:

Group 7

Gianni Difede

Peterling Etienne

Long Ho

Dukens Louis

Luke Schuknecht

10/25/21

|  |  |
| --- | --- |
| Revision # | Date |
| 1 | 9/28/21 |
| 2 | 10/25/21 |

Vertical Prototype Notes / Features

Since this is a vertical prototype and the team members lacked prior knowledge, some features are not implemented exactly like specifications or GUI mockup. This will be fixed for future versions.

Major features implemented:

* Log in, Register, and logout
* Main hub feed
* Posting
* Visiting other people’s profile
* Visiting your profile
* Editing your profile

Testing major features:

* Login in – user name: Demo, password: demo123
* Create account
* Visit other accounts
* Visit your account
* Edit your profile
* Add a post
* Logout

Major features not implemented yet:

* Feed priority
* Like functionality
* User posts showing up on profile page

**Table of Contents**

[**Executive Summary** 5](#_Toc86093174)

[**Competitive Analysis** 6](#_Toc86093175)

[**Analysis Table** 6](#_Toc86093176)

[**Data Definition** 8](#_Toc86093177)

[**High-Level functional requirements** 13](#_Toc86093178)

[**Non-User expectation** 13](#_Toc86093179)

[**User expectation** 14](#_Toc86093180)

[**Non-functional Requirements** 17](#_Toc86093181)

[**Performance Requirements:** 17](#_Toc86093182)

[**Ease of Use:** 17](#_Toc86093183)

[**Interoperability Requirements:** 17](#_Toc86093184)

[**Security Requirements:** 17](#_Toc86093185)

[**Portability Requirements:** 18](#_Toc86093186)

[**Supportability Requirements:** 18](#_Toc86093187)

[**Storage Requirements:** 18](#_Toc86093188)

[**Survivability:** 18](#_Toc86093189)

[**Availability Requirements:** 18](#_Toc86093190)

[**High-level System Architecture** 19](#_Toc86093191)

[**System Software/Tools** 19](#_Toc86093192)

[**Database Schema** 20](#_Toc86093193)

[**Media Storage:** 21](#_Toc86093194)

[**Search and Algorithm implementation:** 21](#_Toc86093195)

[**UML Diagrams** 22](#_Toc86093196)

[**Risks**: 24](#_Toc86093197)

[**Skill risks** 24](#_Toc86093198)

[**Schedule risks:** 24](#_Toc86093199)

[**Technical risks:** 24](#_Toc86093200)

[**Teamwork risks:** 24](#_Toc86093201)

[**Legal/content risks:** 25](#_Toc86093202)

[**Team Roles** 26](#_Toc86093203)

[**Check List** 27](#_Toc86093204)

# **Executive Summary**

Social media is currently ingrained in our society - one could even make the argument that it surpasses television as the highest form of consumed media. This is because of the access people have through their mobile devices, computers, and tablets. It allows people to engage one another directly. There are many forms of social media examples being YouTube, Facebook, Twitter, Pinterest, and Tumblr. People would often produce content and post it on social media, whether it be photos or comments. Users of these platforms would voice their opinions something that has drawbacks to it. If someone were to spreading misinformation and the user is not fully informed, they could end liking that content without necessarily understanding what they are Liking. This upvotes that content in the Home Page Activity causing it to trend. The Like would tie that users to that opinion or belief. This could lead to the user losing place of employment. likU.com has developed the solution to these problems.

likU.com would be first website that allows the user to gauge the level in which they Like content. This would be 3-Like-System that would allow users to either be anonymous or transparent on whether they like something or not. The First Like would allow the user to promote the content anonymously. The Second Like would allow the user to promote the content anonymously, but also allow the content creator to feature frequently on the user’s main hub or trend on the website. The Third Like would not only promote the content but would publicly display the post for the users’ friends and on their Profile Page. The most important part of this feature is that if a user were to Like and comment anonymously, the content creator cannot see it. The content creator would only be able to engage with users who chose to publicly like their content.

This puts more control in the users’ hands. It would also push the focus on a sense of community. Rather then engage in disagreements that would cause users frustrations, potentially driving them off them off the website, the user can control what content they want to see as well as what content their friends would see on their Profile Page.

The purpose of this is create a social media website that would promote heathier social media engagement. This would also limit the spread of misinformation. It would allow people to give an unbiased opinion on content without other judging their opinion. It would also minimize fake accounts or dummy users who create fake accounts to comment on content anonymously. Ultimately, our goal is to maximize user engagement, and get as many people to be on the website as possible, while filtering out the content that would leave negative impact on site.

In summation, LikU.com will leave the users and content creators with a positive experience. The like system would give the user a since of protection, and it would also allow people to enjoy content they like without being judged by friends. The content creator would benefit by having more user engagement on their content, while reducing misinformation spread by influencers.

# **Competitive Analysis**

This will be simple ratting system of 0-5, 5 being great and 0 being poor. “Webpage” describes the design of the user interface. “Content” refers to what is seen on the website, i.e. videos, pictures, movies and comments. “Anonymity” refers to the user’s ability to be completely anonymous on the site. “Market Share” indicates the current market share on each platform. In Market Share, the 1-5 five rating will be changed to reflect the market of scheme on the market, 5 being the most and 0 being the least none. The information is taken from https://gs.statcounter.com/social-media-stats/desktop/worldwide. “Accessibility” reflects the ease of use for the site by the user. “Like” refers to the variety like system offers.

## **Analysis Table**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Comparison** | **LikeU** | **Twitter** | **Facebook** | **Pinterest** | **YouTube** | **Tumblr** |
| **Webpage** | 5 | 4 | 3 | 3 | 5 | 3 |
| **Content** | 4 | 2 | 3 | 2 | 5 | 4 |
| **Anonymity** | 5 | 1 | 1 | 1 | 1 | 1 |
| **Market Share** | 0 | 3 | 5 | 4 | 1 | 2 |
| **Accessibility** | 5 | 5 | 4 | 3 | 3 | 2 |
| **Like** | 5 | 2 | 2 | 5 | 4 | 2 |
| **Average** | 4 | 2.83 | 3 | 3 | 3.16 | 2.33 |

**Twitter Rating (2.83) https://twitter.com/**

Twitter has great webpage for its targeted audience. It poses the smallest threat because we believe that we would be able to produce more types of content. For example, we would investigate having longer videos and larger number of characters in the text box.

**Facebook Rating (3) https://www.facebook.com/**

Facebook poses the biggest threat because it has the largest market share on desktop. It would be difficult to compete with its marketplace because it is already saturated. Our solution is to attract the demographic between ages 20-35. The like system is one of the weakest because it does not have any variety.

**Pinterest Rating (3) https://www.pinterest.com/**

Pinterest also does not directly pose a threat, because it primarily caters to artist and photographers. It does have the second biggest marketplace on desktop, but because we after a different audience this should not be issue when competing. Pinterest does have an amazing like system. It has 5 tiers that vary from “good idea” to “ha-ha”. This system is close to our system, but it does not provide anonymity.

**YouTube Rating (3.16) https://www.youtube.com/**

YouTube does pose threat the LikU because of the content it offers. The only content we do not offer is the ability to purchase movies. This would be the only competitive edge that YouTube offers, but it has the weakest market share on desktop. The YouTube Like system does not offer variety on how it displays the like to dislike ratio on each video.

**Tumblr Rating (2.33) https://www.tumblr.com/**

The only threat tumbler offers is its current place in the market share. Beyond that it is much more difficult website to use. We rated its accessibility the weakest among the competition. Its like system does not offer much variety.

**LikU Rating (3.8)**

LikU will offer more engaging content. It would be setting the trend when it comes to our custom like system. It would also be the only website that allows users to remain anonymous.

# **Data Definition**

The data definition table contains the name, meaning, and usage for all terms that hold significant value in the context of LikU. The name of a term is how the team refers to the term. The meaning is what the term means or does. Each term can represent a main term definition, a data element, or an application service. The usage column represents which one of these three groups a term is.

|  |  |  |
| --- | --- | --- |
| Name | Meaning | Usage |
| LikU | Name of project that resents the entire web application and the associated web pages. | Main term |
| Web application | Reference to both the back end system and front end visualization for actor interaction | Main term |
| Website | Reference to front end visualization for actor interaction | Main term |
| System | All the data inside the database and the backend of the application | Main term |
| Actor | Any induvial that is using LikU | Main term |
| Guest | An actor that is using the application but is not registered in the system | Main term |
| User | An actor that is using the application and is registered in the system | Main term |
| Account | Represents a single user’s collection of data and posts. | Main term |
| UserID | A displayed representation of a user for both other actors and the system | Data element |
| Password | A string of characters that allows the system to recognize a user | Data element |
| Register | The act of an actor inputting their data into the system for the first time | Application service |
| Log in | The act of an actor successfully inputting their password into the system. Allows user to post, comment, and like | Application service |
| 3-Like-System | Reference to a user’s ability to either first like, second like, and third like any given post | Main term |
| First Like | An action a user can take to indicate that a user enjoyed another users post | Application service |
| Second Like | An action a user can take to follow another user | Application service |
| Third Like | An Action a user can take in order to re-post another user’s post | Application service |
| Upvote | Pushing a post towards the top of the Home Page Activity Hub feed | Application service |
| Post | Refers to both the photo and caption that a user allows their followers, users, or other guests see | Application service |
| Photo | Refers specifically to the image that a user uploads to LikU | Data element |
| Caption | Refers specifically to the text that a user can add to a photo | Data element |
| Comment | The opinions other users leave on a post | Application service |
| Discussion | Refers to the conversation that is happening in the comments on a specific user’s post | Main term |
| Trends | A certain type of post that many users are currently enjoying and are liking (first, second, or third) | Main term |
| Search | The act of an actor looking for a specific user or post | Application service |
| Home Page Activity Hub | A user specific webpage that contains a feed of all the recent posts from their second liked users | Main term |
| Profile Page | The webpage that contains the bio and posts of a user | Main term |
| About Us Page | The webpage that contains information on how to use application features | Main term |
| Follow | An action that allows a user to see another users posts on their feed more frequently | Application service |
| Followers | Refers to the users that have second liked a user before | Main term |
| Database | All the data that the system stores and needs in order to properly function | Data element |

**Use Cases and Project Overview**

“LikU” has many unique features to it that are quite simple to understand and can be easily understood by following this broad overview of scenarios below. Some scenarios that will explain the features of “LikU” include: All of the 3 Like-System applications, and the Home Page Activity Hub.

**3-Like-System**

**1st-like:**

When the user sees a post on their Home Page Activity Hub, they are able to give that post a “1st” like which will upvote or push the post upwards on the Hub. A common scenario is a user seeing a post that they favor; the user could then click and add a “1st” like to the post. The user has therefore given the post superiority and will allow the post to be seen first before others posts.  With the addition of more and more “1st” likes the post moves it way up the Hub’s feed on those that have the ability to see the post. The “1st” like system can be given for an endless number of reasons but not limited to, importance, controversy, trends, or even great discussion.

2nd-like:

Similarly, when the user sees a post on their Home Page Activity Hub, they are also able to give that post a “2nd” like, which is similar to a follow. A common scenario is a user seeing several pictures or discussion posts that they have seen from a user and would like to continue to see content from this user, the user could then click and add a “2nd” like to the post. Once the user gives the post a “2nd” like, the user will now see all future content on their Home Page Activity Hub by that person that has posted that content. This will then increase the number of posts that will pop up on users’ feeds in response to how many user posts they “2nd” like.

3rd-like:

Likewise, when the user sees a post on their Home Page Activity Hub, they are again able to give that post a “3rd” like, which is similar to a re-post. A common scenario of a user wanting to share a post with others on “LikU,” they would click “3rd” like and the post is now shared on the Home Page Activity Hub of the users that have followed you via a “2nd” like. The combination of the “2nd + 3rd” like allow for great growth in the amount of post the user will be able to see and allowing for a increase in others also. This is important in the communication aspect of “LikU.”

**Home Page Activity Hub**

After opening up “LikU” on your browser, users are immediately brought to their Home Page Activity Hub. The Home Page is where a user is able to witness all the current/recent posts from the users they have used a “2nd” like on in the past and recommended posts. The posts the user will see can be either pictures or a discussion-type post where many users can discuss within that post. The order in which the user will see the homepage is dependent on how its number of upvotes and whether the post is trending. The user can then continue to scroll down the activity page and if they would like to give a post to either one of the three like’s they may do so by clicking on 1, 2, or 3. The Home Page Activity Hub will continue to allow the user to navigate in a downward scrolling motion until all the past posts within a set time period have been made known to the user.

**Posting and commenting**

To go into further detail, the content that you will see on your Home Page Activity Hub will be posts. Users once on the Hub will be able to click the bottom on the top of the screen to then add a new post. Once adding a new post, the user will be able to either add a picture or add a discussion to be posted. Once adding their content or either starting a discussion, and then posting it, other users that view it will be given the ability to comment or use one of our 3-like systems on the post. When commenting all you would need to do is press the comment button underneath the post, similar to a caption, the user can type what you would like, and then submit their comment to be added to the post.

# **High-Level functional requirements**

## **Non-User expectation**

1. **Creating Account** – The system shall allow the user to create an account by storing UserID, Password, Date of Birth, First Name, Last name, Location, and answer to security question/phone number. The system shall not allow the Guest to Create an account if the UserID choose by the Guest already exist in the System’s Database. The system shall prevent the Guest from creating an account if the chosen password does not match the re-enter password field. System shall prevent the creation of the Guest’s account if the following fields are not filled. Required fields are First Name, Last Name, Location, UserID, Password, Re-enter Password, Security Answer Security Question or Phone number, and Date of Birth.

* **Response Sequence –**

Guest enter a UserID

Guest enter a password

Guest re-enter a password for conformation

Guest enter their first and last name

Guest enter their location

Guest enter their date of birth

Guest provide answers to security question or phone number

System checks if UserID is available

System validates password

System confirms first Name, last Name, location, UserID, password, re-enter password, security answer security question or phone number, and date of birth.

System will have button to redirect to home page

* **Function Requirement Label –** Creating Account

1. **Browse Post/Activity** – Actor browsed from a list of post/activities on the activity hub. User will be able to add their own post to the activity hub.

* **Response Sequence** –

Actor enters search criteria (geographic Location or Name of cities) into the search

System shall supply the Actor with a list of activities within the area

System shall have a button that will allow the Actor to return to the home page

* **Function Requirement Label** – Browse by Search

1. **View Specific Activity** – The system has provided a list post and activities within a region. The Actor has the option to click on and view specific activity. The system shall not allow the Actor to edit the information for a specific activity on the information page.

* **Responsive Sequence** –

User clicks on an activity among the list

System shall provide information about the activity

System shall have a button to allow the user to return to the region list

* **Function Requirement Label** – View Selected Item on List

1. **About Us** – System provides information to the Actor about the functionality that can benefit prospective Actor with links that will direct the Actor to specific topics and the button on the bottom to bring the Actor back to the topics. The Actor will also be able to see core values, FAQ, and the team. The system shall not allow the Actor to erase any information about the system and how the system functions for the user.

* **Responsive Sequence** –

Actor uses the navigation bar and clicks “About us”

Actor is redirected to About us page

Actor sees links about topics in the FAQ section of the page

Actor clicks a topic and get redirected to that topic

Actor after reading topic clicks “back to top”

Actor redirected to top of the page with links and repeated until satisfy

System has a button that allows the User to return back to Home page

* **Function Requirement Label** – AboutUs

## **User expectation**

1. **Edit Profile** – User shall be able to edit their profile by providing a name, username, date of birth, and a brief summary about themselves and the System shall store name, date of birth, and summary when the user clicks the save button. The user will also be able update a new password and email by enter an old password and username. The System shall prevent any changes to the User’s profile if any of the fields are left blank. The user has typed the information in a valid format in order for the system to store the information.

* **Responsive Sequence** –

User will navigate to “my account” and a drop-down menu will be activated

User will click on edit profile

User will input their name, date of birth, and write a summary about themselves

User shall click save

System shall store their name, date of birth, and summary

System shall refresh to user profile with updated information as a confirmation

System will have a button to redirect the user back to the home page at will

* **Function Requirement Label** – Edit Profile

1. **Like Posts** – User will navigate to specific post through the browse functions and have the ability to rate a certain post using the 3-Like-System. System shall provide user with option to like post with 3 likes.

* **Responsive Sequence** –

User will use the browse function to navigate to specific post

User will click on post of choice

User will scroll to click like feature

System shall store number of likes. System will prevent user from rating again

System will have a button to redirect the user back to the home page at will

* **Function Requirement Label** – Like Scale

1. **Comment on Posts** – The system shall have a comment box where Users will have the liberty to state their experience of the post. The systems will prevent any Actor from commenting if the Actor has not created an account and has not given the post a first like. Users are able to comment and like anonymously. Users are supposed to be members and have to like the post they wish to comment.

* **Responsive Sequence** –

User shall use the browsing function

User will navigate to desired post

User shall scroll to the comment box at the bottom of the page

User shall type their experience via keyboard

User shall click “Post Comment”

System shall store User’s comment

System shall display User’s Comment at bottom of the screen with a sorting of newest to oldest order as a confirmation

System will have a button to redirect the user back to the home page at will

* **Function Requirement Label** – Post Comment

1. **User Add Post** – User will be allowed to add post to the database if, and only, if post is not found in the database. The system shall store the given information at the moment the User clicks on “add” button. System shall prevent other Users from Deleting posts from the Database.

* **Responsive Sequence** –

User shall navigate the browse function

User shall navigate to desired location/post and realize such activity is not found.

System shall prompt user, "Do you wish to add post”

User shall hit yes

System shall redirect User to add activity

User shall enter required information and upload a picture

System shall store activity once user clicks Add

System shall thank the user and redirect User to home page

* **Function Requirement Label** – User add

1. **User Contact Us** – User will be allowed to contact the developers via email. User shall type in their name, email, and their query. System shall store these fields and submit it once the user clicks the send button. System shall prevent the User from contacting developers directly.

* **Responsive Sequence** –

User shall navigate to Contact Us among the navigation bar

User shall fill in Name, Email, and type up their Query

User shall then click “Send” button under neither the Query box

System shall store the information and submit

System shall state that the information was sent and thank the user

System shall a button to redirect the user to Home page

* **Function Requirement Label** – Contact Us

1. **User Log out** – User shall be able to log out of their account by clicking on the logout button. System shall stop storing all user related fields upon logout.

* **Responsive Sequence** –

User shall navigate to the log out button on the settings page

User shall click the log out button

System shall stop storing all session data about user

* **Function Requirement Label** – Log out

## **Priority Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| High-Level  Function Name: | Description: | Functions: | Activity: | Requirement Type: | Priority: |
| Sign-in | User will be able to sign into their account with either an accompany username/password which will then check to see if valid or invalid. | * Username * Password | * User enters in Username * User enters in Password * System checks database to see if this exists and correctly matches * Signs into database if correct | User/System | 1 |
| Account Creation | Guest will be able to create an account with unique information ONLY and will be stored on database for future use of logging in. | * User ID * First Name * Last Name * Bio * Username * Password * DOB | * User enters First name * User enters Last Name * User enters Username * User enters password * User enters DOB * User enters in Phone Number * User clicks submit and System checks if username is unique and will create using information if username is unique. | User/System | 1 |
| Main Activity Hub | The System will load up a home page hub where activities will be already posts will be loaded, and for viewing by scrolling down the page. Also allowing the actor to be able to post/caption/comment | User id  Content  Time  User id  Caption  Image  Like 1  Like 2  Like 3  Like total  Comment | * User is taken to main hub after login is successful or via navigation bar under Main Activity Hub * Actor may scroll down hub to view all content * Actor can use one of the 3 like system to the page | System/User | 1 |
| Upload posts and captioning | User will be able to upload a picture to the main hub along with a caption if they would like. | Post id  Content  User id  Caption | * User will upload image via main activity hub * User can also choose to caption post * User clicks submit post * System will upload if post is unique and not found in database | User | 1 |
| Commenting | User can upload a comment underneath posts to add to the richness of the post | Post id  Comment  Content  Time | * User clicks on a post * User will add comment under post * User Clicks ‘Comment’ * System posts comment to post to show on main activity hub | User | 1 |
| 3 - Like System | User will be able to like a post on the main activity hub in 3 separate ways based on their liking | Post id  Like 1  Like 2  Like 3 | * User will use First Like (enjoyed post) * User will use Second Like (follow that User’s post) * User will use Third Like (repost the post) | User | 1 |
| Edit Profile Page | User will be able to edit their own profile page and store on the page their first name, last name, Date of Birth, and bio. | User ID  Username  First Name  Last Name  Date of Birth  Bio | * User clicks on their profile icon * User then clicks the Edit Profile Button * User can change their First Name, Last Name, Date of Birth, or bio if they would like * User clicks the save button * System saves new information under the User ID. * User can click home page button to go back to the main hub | User | 1 |
| About Us | HTML Webpage that will tell the Guest, Actor, or User about how ‘LikeU’ started, the goals of the devolvement team, values, and website specific information |  | * User visits about us Page via navigation bar button * User views information * User can click home page button to go back to the main hub | Business | 2 |
| Contact Us | User will be able to send comments to the web developers | First name  Comment | * User visits contact us page via navigation bar button * User enter in first name * User then types into Query box what they would like to send * User clicks send * System stores First name and comment Sent * User can click home page button to go back to the main hub | Business/User | 2 |
| Profile Page | User can view their own, or another user’s personal webpage | User ID  Username  Fire Name  Last Name  Bio | * User clicks on Username’s Profile Page link or their own via navigation bar button. * User can view information about that user * User can click home page button to go back to the main hub | User | 1 |
| Search for Users | User can use the search utility to find other users that have posted on the main activity hub | Username | * User clicks on search bar at the top of the screen * Types in username of the other user he would like to find * System finds username and brings user down to the mentioning of the username searched | User | 3 |
| Logout | User Is able to logout of ‘LikeU’ when they please to | Session unset/destroy | * User is done using various ‘LikeU’ features * User clicks Logout button on navigation bar * System clears session and destroys it * System brings you back to login screen | User/System | 1 |

# **Non-functional Requirements**

## **Performance Requirements:**

1. **Responsiveness**: The website will be responsive on various monitor sizes, ranging from 10” netbooks to 24” desktop monitors. It will also be responsive with a wide variety of resolutions, from 1024x600 through 1900x1200.

2. **Response Time**: The average response time is around 3.5 seconds for a website. Google recommends a page to load in under 2 seconds. With this in mind, the LikU website is intended to load at an average website speed with minimal user usage. The website will start lagging as more users continue to use the website. The website will load slower than average as more users continue to use the website

3. **Test Requirements**: performance test will include multiple load tests. Such as Load test on a different platform, load test on different computers with different specs.

4. **Storage**: The website will use all the available storage provided for its usage to maximize the website functionality.

## **Ease of Use:**

1. **Training**: Training should be minimal, if any, as our site will be very user-friendly and can be used by anyone who visits it.

## **Interoperability Requirements:**

1. **Browser Compatibility**: The website will operate on any browser that supports JavaScript, including Google Chrome, Mozilla Firefox, Safari, Opera, and Internet Explorer.

2. **Computer and OS Compatibility**: The website will operate on various operating systems, including Windows, OS X, and Linux. It will work on any type of computer that can run a browser.

## **Security Requirements:**

1. **Login/Password System**: The website will have a login/password system to maintain the profile preferences. This implementation will also require password confirmation upon creation. We will also ask the user for their email address to retrieve a forgotten password

2. **Resource Utilization**: Resources such as the MySQL database on the https://lamp.cse.fau.edu server will be accessed through the PHP code using the usernames and passwords therein. All-access to the https://lamp.cse.fau.edu servers and their resources will be obtained with the usernames and passwords given.

## **Portability Requirements:**

1. **Platform Compatibility**: The website is developed for mobile and tablet.

Mobile and tablet compatibility will be implemented by our team.

## **Supportability Requirements:**

1. **Coding Standards**: the website will be coded in a range of 75-80% of coding standards for HTML5 and CSS3, JavaScript, and PHP. The code will be produced, reviewed, tested, reviewed again, and finalized by yet another developer for efficiency.
2. **Naming Conventions**: HTML classes and id tags will be coded in lowercase except in the situation where there is more than one word in a name, and then it shall use camel case convention.

## **Storage Requirements:**

1. The storage for the website will consist of the phpMyAdmin.net server holding our MySQL databases within an unknown capacity as well as holding our files for the actual site.

## **Survivability:**

1. The website will be stored on <https://lamp.cse.fau.edu> servers, and the significant files and documentation will be backed up on Google Drive to prevent loss in case of a fire or other physical destruction of the servers.

## **Availability Requirements:**

1. **Accessible Times**: the website will be available for use 24 hours a day, seven days a week. It will be up and running if the https://lamp.cse.fau.edu server is available.

2. **Downtime Impact**: The downtime will be minimal, but when necessary, a splash page will be used to identify that the system is in maintenance. Downtime impact is expected to be minimal, and the scheduled downtime will be announced ahead of time.

3. **Rating**: The Website is suitable for everyone

# **High-level System Architecture**

## **System Software/Tools**

1. **Group 7 Lamp Server:** Our given group LAMP Server, lamp.cse.fau.edu/~cen4010\_fa21\_g07, will be used to host our web application. The lamp server will also give the team access to the necessary PHP and MySQL facilities.

1. **VMware Engineering Virtual Environment:** Due to security measures put in place by FAU, the team will use the provided virtual environment in order to connect to and make changes to the group LAMP server.
2. **WinSCP**: WinSCP is a Secure File Transfer Protocol (SFTP) software. The team shall use WinSCP in order to transfer files to the group LAMP server from our virtual environment. WinSCP will allow the team to create and manage our LAMP server.

1. **MySQL Database:** MySQL is a database management system that allows our team to create, update, and delete our database. Our database will contain all the necessary information/data that our application needs to function. Users will be able to add or update their data through our web application.
2. **phpMyAdmin:** phpMyAdmin is an administrative tool to manage our MySQL database. PhpMyAdmin allows the team to manipulate data and manage our database schema in an easier fashion.
3. **GitHub:** The team shall use GitHub in order to allow for better communication and code/file sharing applications. GitHub will allow the team to simultaneously work on the web application. The team will also use GitHub for version control. This means that the team will be able to merge their files with the most updated version of the code or return to earlier version of the web application.
4. **Hyper Text Markup Language (HTML):** HTML is a programing language that allows for web development. The team will use HTML as the building blocks to define webpage structure for the LikU website.
5. **Cascading Style Sheet (CSS):** CSS is a programming language that helps better define how a webpage will look. The team will use CSS to make an aesthetically pleasing webpage that is intuitive to browse.
6. **JavaScript:** JavaScript is a programming language that allows for interactive web pages. The team will use JavaScript to aid in the process of creating an enjoyable user experience.

1. **Hypertext Preprocessor (PHP):** PHP is a programming language that allows for communication between the website and the database. The team will use PHP to communicate and make changes to the database.

1. **Personal Integrated Development Environment (IDE):** An IDE is a software application that makes development easier. Most IDEs contains an editor for source code, contains suggestions, and a debugger. Every team member will use an IDE of his or her choice.

1. **Bootstrap:** Bootstrap is a framework that contains both CSS and JavaScript that makes front end development easier. The team will use Bootstrap to create better looking webpages easier. License link: https://github.com/twbs/bootstrap/blob/main/LICENSE
2. **Supported Browsers:** The team will take the necessary efforts to ensure that the web application works Mozilla Firefox, Google Chrome, Safari, Opera, and Internet Explorer. The team will ensure that any functionality is supported by these web browsers.

## **Database Schema**

**Tables**

The database will contain three different tables. These tables are a table to store user information, post information, and comment information. The user table will contain all data required to have a log in system and display personalized profile pages. The post table will contain all data required to display all post content and like counts. The comment table will contain all data required to display all comment content and sort comments.

**User Table**

The user table will have the following items in the table:

* Id – primary key
* First – user’s first name
* Last – user’s last name
* Username – user’s display and login name
* Password – user’s password
* Dob – date of birth
* Following – List of users that a user wishes to see more of on their feed

**Post Table**

The post table will have the following items in the table:

* Id – primary key
* user\_id – foreign key to show who the post belongs to
* caption – poster’s caption added to a image
* img – post’s photo
* like1 – total number of 1st likes on a post
* like2 – total number of 2nd likes on a post
* like3 – total number of 3rd likes on a post
* time – the time of posting

**Comments Table**

The comments table will have the following items in the table:

* id – primary key
* post\_id – foreign key to show what post user made comment on
* content – the comment itself
* time – the time of posting

## **Media Storage:**

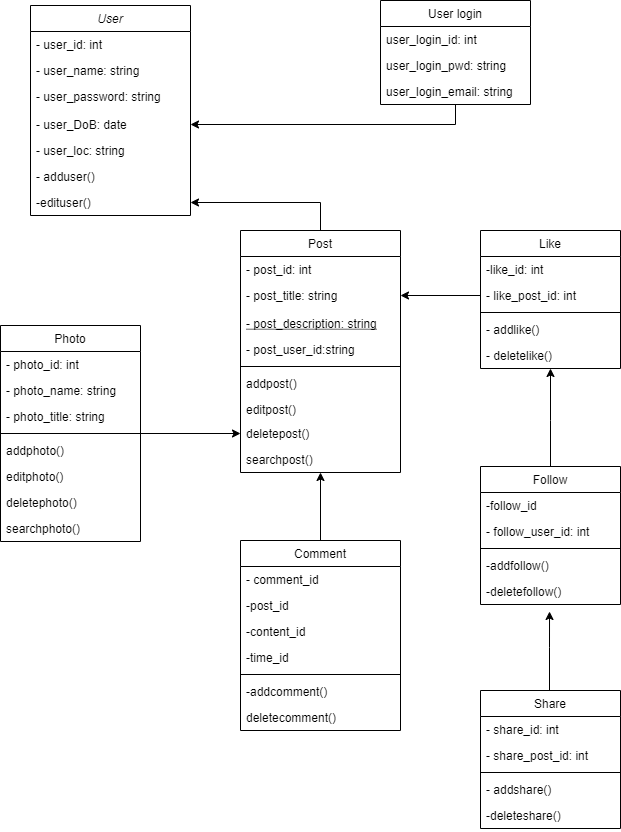
The post photo is the only special data that will be stored. The images will be store within the post table of the database as a BLOB.

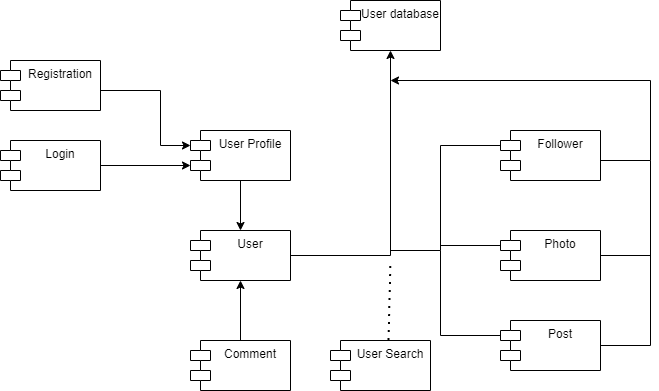
## **Search and Algorithm implementation:**

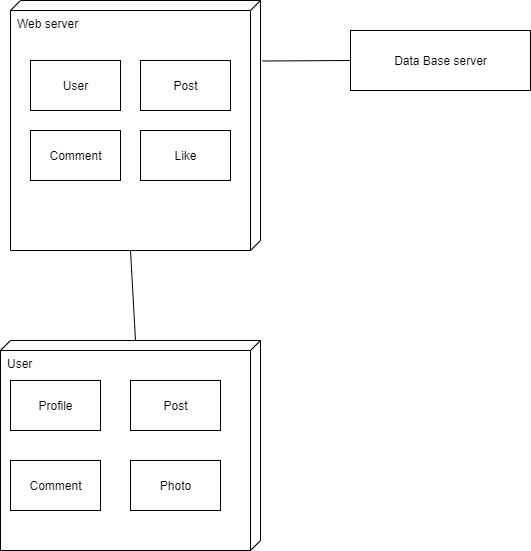
Users will be able to search for other user’s through the use of their usernames. A user will be able to enter a part of another user’s username and the system will display all usernames that contain that phrase. The main hub feed will be sorted/filtered through three main constraints. These three constraints are liked content, trending posts, and time. The system will prioritize posts made by other users that the user has second liked in the past. After that the system will prioritize posts that currently have the most amount of likes. Finally, the system will ensure that the posts shown first are less than 24 hours old.

## **UML Diagrams**

**Class**

****

**Component**

**Deployment**

# **Risks**:

## **Skill risks**

This project requires a high level in database management as well strong comfort with HTML and CSS. As a team, our biggest weakness is the fact that we don’t have many people with strong background in database management. Database management is vital to the project because we would be managing users’ data. This includes things date of birth, name, and email etc. Managing such vital information would require delicate care. Although we have and enthusiastic team that is more than capable of adapting to this risk.

Solution:

The solution is to try and get started on this early and utilize all the resources available. This would all more time to implement what we learn. Thus, leaving us a greater advantage on this project.

## **Schedule risks:**

To ensure the success of this project, commitment is vital to the project. The includes things like scheduling meets to discuss different aspects of this project. The only risks are that everyone on the team has other priorities like jobs, family, and school.

Solution:

The solution is to have a weekly team meeting that we hold on Saturday to discuss the project. We also have a group chat on WhatsApp so that the group is in constant communication.

## **Technical risks:**

A major technical risk is that some group members had trouble connecting their php to the phpMyAdmin database. This makes it difficult to run test when developing php code. Php is the backbone of this project, and it is required because it allow us to access the database from phpMyAdmin so that we can display what is in the database to the website.

Solution:

We still developing the skills to address these issues. This solution requires time, research and communication.

## **Teamwork risks:**

Teamwork risks include would be a subset of schedule risk because we do not meet every day. Due to this, it is difficult to gauge everyone commitment.

Solution:

Jira is also a tool that we that updates everyone on the progress projects. This would allow us to address the problems with teamwork. GitHub also allow us to communicate and see each other progress in the project.

## **Legal/content risks:**

The type of website that we are creating allows the users to upload pictures, files and videos to our server. We are not currently able to efficiently control and monitor what gets posted to the website. We will also be using bootstrap to help with design.

Solution:

The primary solution to this is only using copyright free bootstrap and to set a terms and condition for the website that set expectations for how we would like the users to use your website.

# **Team Roles**

**Team Lead and Back End Developer:**

* Gianni Difede

**Scrum Master:**

* Long Ho

**Product Owner:**

* Dukens Louis

**Back End Lead:**

* Luke Schuknecht

**Front End Lead:**

* Peterling Etienne

# **Check List**

Team decided on basic means of communications - Done

Team found a time slot to meet outside of the class - Done

Front and back end team leads chosen - Done

Github master chosen - Done

Team ready and able to use the chosen back and front-end frameworks – On Track

Skills of each team member defined and known to all – Done

Team lead ensured that all team members read the final M1 and agree/understand it before submission – Done