CEN 4010 Principles of Software Engineering

Summer 2021

06/15/21

# Milestone 3 Project Proposal
# &
# High Level Description

Agile Development Team Group 3

ADevTG3

Joshua Gardner - jgardn11@fau.edu

Ivan Duben - Iduben2020@fau.edu

Holly Schlichting - hschlichting2015@fau.edu

Justin Lawson - Lawsonj2016@fau.edu

Steven Solo – ssolo2019@fau.edu

# History Table

| Date | Revision History | Revision Class | Comments |
|---|---|---|---|
| 06/15/2021 | 1.0.0 | Original | |
| 06/15/2021 | 1.0.1 | Add/ Remove | Added Key Risks and High Level UML sections<br>Modified High Level System Architecture, Database Organization, Competitive Analysis, Executive Summary, High Level Functional Requirements, and Non-Functional Requirements. |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Executive Summary

OwlPost is the new-age social media network we're building that is intended to connect college students. It's an iPhone and Android software that allows you to share photos, videos, comment, posts, create unique usernames and passwords in the creation of your personal encrypted private or public account and is user friendly with the most current commonly used web browsers. Students are able to post photos or videos to our website they want to share with their followers or small groups in addition to unlimited numbers of friends. They can also view, comment on, and like Instagram posts that their friends have made.

The app promotes communication with others by allowing you to take images; it expands your friend list; your professional network in a beneficial way for everyone; and it allows you to share information with friends and family in a unique way. You can take a photograph and post it for all fellow students to see; or take a picture of your achievements on OwlPost for all your friends to see.

“Community first, promote creativity, and simplicity matters” are three of OwlPost’s key beliefs. These values are crucial to the firm because they help fellow college students stay focused on its goal and vision statement.

# Competitive Analysis

We plan on making a social media application that allows students to connect to each other through posts that are made from user profiles mainly through pictures that will have a thread attached to allow students to comment on them. The applications main competitors will be similar featured sites that are already well known, established, with extensive brand recognition such as: Facebook and Instagram due to their extensive currently existing outreach globally. Two of the competitive advantages will be our plan to primarily keep the app local within the student body and eliminating the nerve-racking pop-up ads trying to sell you a pitch as you scroll through post and/ or feeds.

### User engagement by creating polls in groups and events

Having the capability to spontaneously start a poll for constructive criticism or just a simple in the moment need for assistance on a topic that may be troubling the user adds another layer of usefulness to the app.

### Keyword usage to search for old posts

Searching for old posts with the ease of keywords in order to find posts on topics users are interested in allows information to be accessed with efficiency.

### Receive notifications regarding unfollowed alerts

Being able to see who unfollowed our users is a convenient way for them to pin point root causes and evolve their usage approach on the app.

### Linkable to profiles from other commonly used apps

The ability to attach other profiles to user accounts on our app results in a user friendly atmosphere for everyone to access various accounts they wish to maintain.

Below is a chart comparing our features against our competitors. These six key features are what we view as highlights in our app, because they are key features that are found in one place within OwlPost's application. Each feature has not been seen on a single app together. A few have been used on secondary apps that are not affiliated with Facebook. This has put users in a position with a higher likelihood involving potential security breach since Facebook is not liable if users information gets lost or stolen using other secondary apps. Having all these features accessible in one secure location will assist in avoiding scammers and pirates on the web.

# Competitive analysis Chart

| Competition | Facebook | Instagram | OwlPost’s |
|---|---|---|---|
| Catering primarily to FAU student body | | | X |
| User engagement by creating polls in groups and events | X | | X |
| Keyword usage to search for old posts | X | | X |
| Receive notifications regarding unfollowed alerts | | | X |
| **Linkable to profiles from other commonly used apps** | X | X | X |
| No pop-up Ads | | | X |
| Total | 33% | 33% | 100% |

# Data definitions

· **Users -** There will only be one user level, “user.” Will have access to view other users’ profiles, create posts, upload images and interact with other users. Users will have access to manage their own data. IE: Delete images, edit profile and posts.

- Required information will be a username , first name, last name, email, and password.

· **Profile** – Users’ page, will contain personal information about the user.

- May provide birthday, location, education, and occupation if the user desires to add it.

· **Images** – Users’ will be able to post images in jpg, jpeg or png file formats.

- Max file size for photos will be (Placeholder)

· **Posts** – Users will be able to create a “post” which will be added to the profile. Can contain a photo.

- Max character limit of 250.
- Will be public with the ability to make private.
- Will be no limit on how many posts can be made.

· **Feed** – Website homepage, gives users a scrolling page with the newest posts from connected users.

· **Friends** – Users will be able to create connections with people they know. Accepting a friend will create a link between the profiles and allow users to see your posts and updates.

- No limit on how many friends you can add.

# Overview, Scenarios and use cases

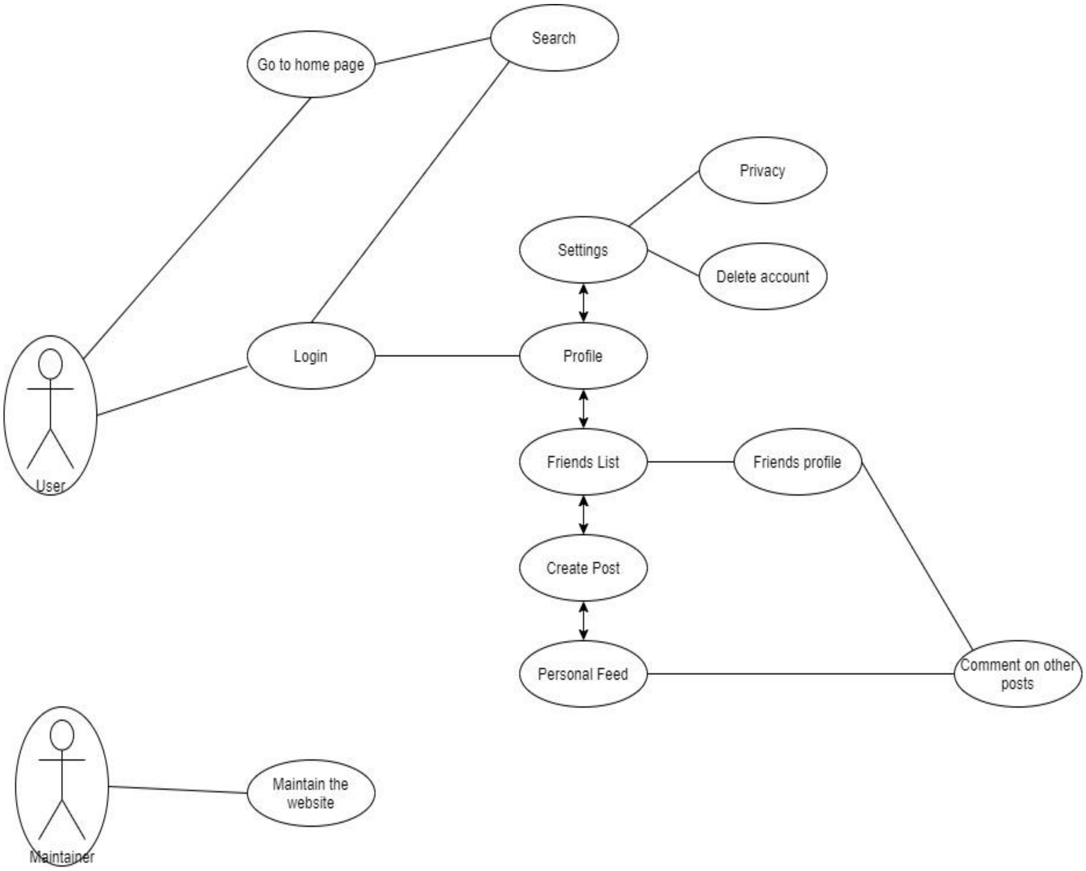
The project will be a social media type of site. This site's primary goal is to connect people via an online platform. Due to the unprecedented issues created by Covid-19, people are more socially isolated than ever seen before in documented history both physically and mentally. College students have been strongly affected by the recent lockdowns imposed over risks involved from the potential spreading of Covid-19 through in person social interaction. Generally, college students consider this time period as an important period of interacting with new people and other cultures which in turn has been disrupted due to the aspects of social distancing preventing them from having the traditionally expected experience.

The event of Covid-19 has paved the way to college students being the ideal target audience for this site. This site assists in coping with the mental part of being socially distanced by allowing others to share parts of their lives interacting with others while still maintaining the required distance during this difficult time. Users will be able to post text, share pictures, and respond to others. The skill level of the users will be a basic idea of how to interact with websites with a potentially higher skill level needed in order to post pictures as it would require users to understand how to navigate their own computers files.

# User Stories

1. I as a user need a profile to achieve the goal of others being able to find and interact with me.
2. I as a user need the ability to comment on posts to achieve the goal of interacting with friends.
3. I as a user need the ability to create posts to achieve the goal of being able to share parts of my life or things I like.
4. I as a user need the ability to control who can see my posts to achieve the goal of having privacy.
5. I as a user need to be able to add friends to achieve the goal of easily interacting with the people I know.
6. I as a user need the ability to create an account to achieve the goal of being able to use the website.
7. I as a potential user need the ability to view public posts to achieve the goal of deciding if I want to use the website.
8. I as a user need the ability to delete my account to achieve the goal of being able to control my internet life.

9. I as a user need the ability to delete posts in order to achieve the goal of being able to control my content.



# Initial List of High Level Requirements

The social media platform we are creating is designed for college students to connect. It is a photo and video sharing app that can be downloaded on iPhone and Android. People can use our service to submit photos or videos and share them with their followers or a small group of friends. They can also look at, comment on, and like posts that their friends have published on Instagram. Anyone college student can sign up for an account by entering their email address and choosing a username.

Specific requirements:

1. Mobile app friendly
2. Username/password sign up
3. Access user's camera to post pictures
4. A "member profile" for each user
5. Comments are available for every post

# Non-Functional Requirements

**Security**

1. Login requirements - access levels, CRUD levels

2. Password requirements - length, special characters, expiration, recycling policies

3. Inactivity timeouts – durations, actions

**Audit**

1. Audited elements – what business elements will be audited?

2. Audited fields – which data fields will be audited?

3. Audit file characteristics - before image, after image, user and time stamp, etc

**Performance**

1. Response times - application loading, screen open and refresh times, etc

2. Processing times – functions, calculations, imports, exports

3. Query and Reporting times – initial loads and subsequent loads

**Capacity**

1. Throughput – how many transactions per hour does the system need to be able to handle?

2. Storage – how much data does the system need to be able to store?

3. Year-on-year growth requirements

**Availability**

1. Hours of operation – when is it available? Consider weekends, holidays, maintenance times, etc

2. Locations of operation – where should it be available from, what are the connection requirements?

**Reliability**

1. Mean Time Between Failures – What is the acceptable threshold for down-time? e.g. one a year, 4,000 hours

2. Mean Time To Recovery – if broken, how much time is available to get the system back up again?

**Integrity**

1. Fault trapping (I/O) – how to handle electronic interface failures, etc

2. Bad data trapping - data imports, flag-and-continue or stop the import policies, etc

3. Data integrity – referential integrity in database tables and interfaces

4. Image compression and decompression standards

**Recovery**

1. Recovery process – how do recoveries work, what is the process?

2. Recovery time scales – how quickly should a recovery take to perform?

3. Backup frequencies – how often is the transaction data, set-up data, and system (code) backed-up?

4. Backup generations - what are the requirements for restoring to previous instance(s)?

**Compatibility**

1. Compatibility with shared applications – What other systems does it need to talk to?

2. Compatibility with 3rd party applications – What other systems does it have to live with amicably?

3. Compatibility on different operating systems – What does it have to be able to run on?

4. Compatibility on different platforms – What are the hardware platforms it needs to work on?

**Maintainability**

1. Conformance to architecture standards – What are the standards it needs to conform to or have exclusions from?

2. Conformance to design standards – What design standards must be adhered to or exclusions created?

3. Conformance to coding standards – What coding standards must be adhered to or exclusions created?

**Usability**

1. Look and feel standards - screen element density, layout and flow, colors, UI metaphors, keyboard shortcuts

2. Internationalization / localization requirements – languages, spellings, keyboards, paper sizes, etc

**Documentation**

1. Required documentation items and audiences for each item

# High Level Functional Requirements

1. The System must use "rule based" configuration options that allows the site to design how the software will be deployed.
2. The staff must be able to adjust common variables, such as codes, tables, report parameters, etc., without the services of a professional programmer.
3. The software must have a one-time, single-point of data entry to reduce redundant work.
4. All software programs must seamlessly integrate to maximize operator and system
5. The software must have an easy to use system for preparing various statistical and analytical reports.
6. Totally integrated search capabilities on all fields without the use of a secondary application.
7. The software must provide a look-up capability for frequently entered information; and, once selected the information will automatically populate the corresponding data record.

8. The software shall provide the ability to input, access, and store a user-defined level of historical data.
9. The software must verify the validity of the data being entered into the database by performing immediate error checking.
10. The software must provide the ability for the system manager to identify any individual who entered, edited, deleted or otherwise changed any transaction and the date and time that the change was made.
11. The software should allow all corrections to be done in the source module.
12. The software must provide the ability for multiple users to be on the system at the same time and multiple users to be in the same programs at the same time.
13. The software must be written using a relational database.
14. The software must provide the ability to direct output of a data search to a printer upon user demand.
15. The system must provide the ability for the user to create PDF, spreadsheet (XLS, ODF), delimited text file format for all reports

16. System should allow different input entry screens based on user’s needs (i.e. a simple

requisition entry screen for casual users, fully featured screens for power users).

17. User experience should be customizable by user. This would include program launch portal, menu's, reports and browse screens.
18. Modules must provide for user defined fields for the master files and input screens.
19. System should provide capability to integrate with a Document Management system
20. System must provide the ability to design site specific output forms (PO's, Checks, etc.)
21. Software should provide a "query" builder to create, save and run common database queries.
22. Software should provide a report auto schedule capability to run reports at designated times in the future.
23. System should utilize "checklist" that the user can create to guide them through a process (such as hiring). Need to indicate if step on checklist is mandatory or informational.
24. System should have a notification system for process that are coming due, such as reapplying for a grant, license expiring, etc.
25. System should have the ability for electronic delivery of forms that include Direct Deposit Advices, Purchase Orders and Bids.
26. System should provide staff ability to clock in and out electronically.
27. System should provide self service modules for employees, vendors, businesses and citizens.
28. Software enhancements provided to other customers should be made available to the client at no charge.

# High-level system architecture and database organization

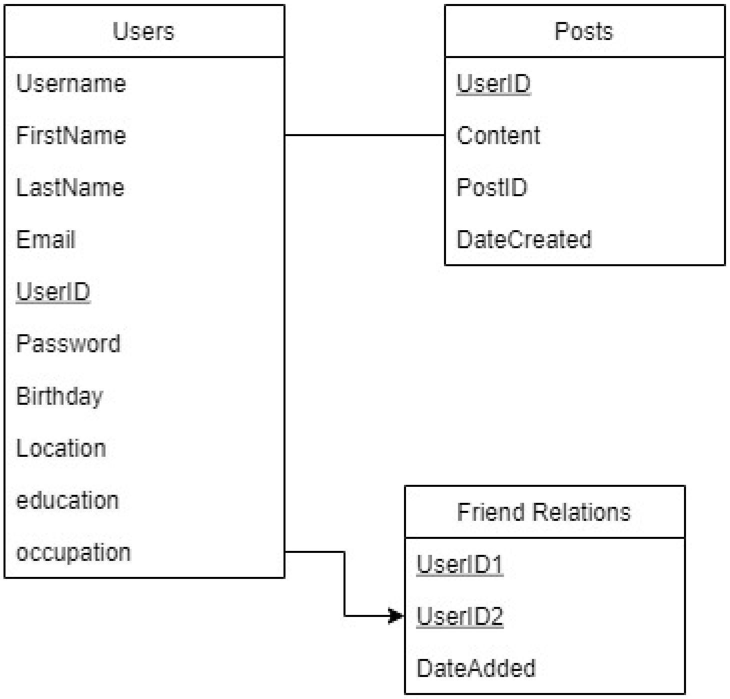
The Database is mostly organized around the “Users” table. The user table includes Username, Firstname, Lastname, Email, Password, birthday, location, education, occupation and UserID. UserID will be the primary key in the table. The “Posts” table has the items UserID, Content, PostID, and DateCreated. UserID will be acting as a foreign key in this table and PostID will be a primary key. The table “FriendRelations” will contain the connections between users. It contains UserID1, UserID2, and DateAdded. Having the primary and foreign keys as above will allow us to easier keep track and search for different users and different posts.

Regarding media storage, images will be kept in file systems. The requirements for images are that they must be in a jpg, jpeg or png file format and will have a maximum size. The high level API will be phpMyAdmin.

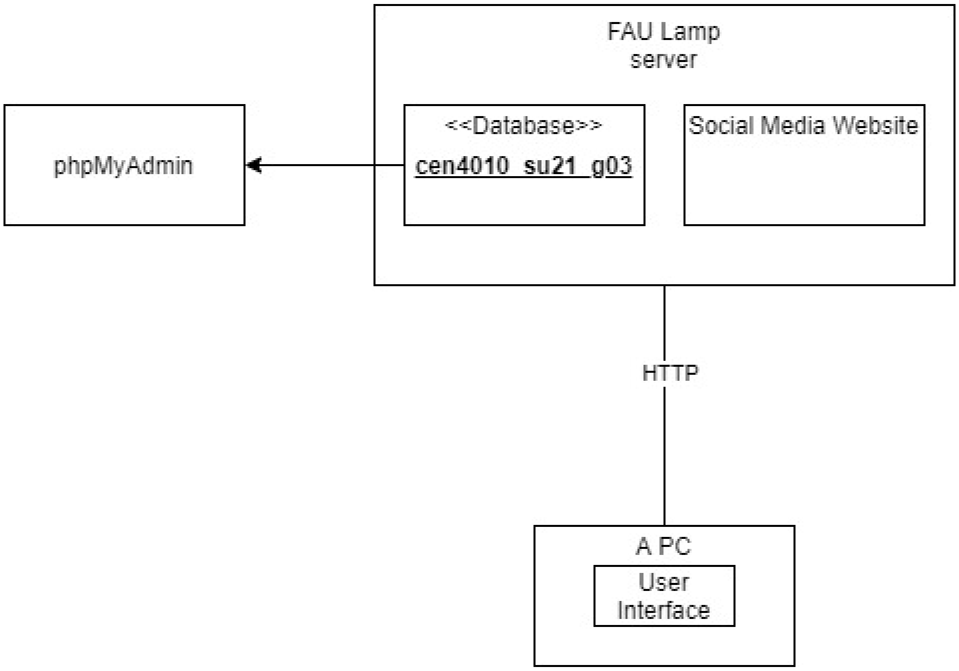
An important algorithm will be involving the FriendRelations table. In order to keep track of who has added who as a friend every time someone adds someone else it will create a new entry into the table containing both userID’s. This will allow us to use an algorithm to see who is whose friends as it will contain both ID’s. Using this method allows us to avoid creating a new table for every user that would hold all their friend entries.

# High-Level UML diagrams

## Class Diagram



# Component and Deployment



# IDENTIFIED ACTUAL KEY RISK FOR OUR PROJECT AT THIS TIME

## Skills Risk

None of our team members have any experience working with a database such as MYSQL on a project. Our team members have limited usage exposure with database software. Creating MYSQL accounts have proven to be difficult and time consuming to resolve. In order to solve these issues, we have reached out to other classmates in the course to determine a successful way to establish our MSQL accounts. Our Scrum master has provided an extensively detailed PDF of how the SQL language operates so that we have a source to reference when working with the SQL language.

## Schedule/Teamwork Risk

Our project has strict deadlines and our team members have limited amounts of time where we can come together as a group to work on various aspects of our project. Due to our conflicting schedules it proves to be difficult to have a perfectly well polished product. What has been determined, is that we will break down each sprint's assignments into three parts with incremental deadlines. This process will enable us to maintain a steady pace of production which will in turn give us the ability to fine tune the elements of our project before our final deadlines.

## Technical Risk

There are many different web browsers, operating systems, and technological devices that are commonly used by the average person. Our product is at risk by uncertainty of which web browsers, operating systems, and technological devices it will allow smooth operation for our users. In order to resolve this risk during the course of our product production we will test OwlPost's regularly on the most commonly used web browsers, operating systems, and technological devices.

## Legal/Content Risk

We were considering including photo editing capabilities within the software of our social media website, but none of our team members are familiar with the process of implementing the code that is required to accomplish a photo editing option. The only other option that we could think of would be to utilize an outsourced photo editing software which probably would require us to obtain permission and/pr licensing otherwise we might run into legal and copyright issues. Currently we have decided not to include the photo editing part of the software in order to avoid the potential risk and/or time required to accomplish this goal.

# Team

Joshua Gardner - Scrum Master – jgardn11@fau.edu

Ivan Duben- Product Owner – Iduben2020@fau.edu

Holly Schlichting – Developer – hschlichting2015@fau.edu

Justin Lawson – Developer – Lawsonj2016@fau.edu

Steven Solo - GitHub Master – ssolo2019@fau.edu

# Checklist

a) Team decided on basic means of communications

- Done

b) Team found a time slot to meet outside of the class

- Done

c) Front and backend team leads chosen

- Done

d) GitHub Master chosen

- Done

e) Team ready and able to use the chosen back and front-end frameworks

- Done

f) Skills of each team member defined and known to all

- Done

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

- Done