# Vertical Prototype Features & Functionalities

Prototype shall be linked via our Team site.

In the search box you may enter any team members first name or last name, to receive info about them. Output will be in the format shown below. If no result is found for search term, a message stating so will be displayed.

Examples:

Search term – Output

campbell - Potential Match Found!  
 Username: vCampbell  
 Name: Van Campbell  
 Email: VCAMPBELL2019@FAU.EDU  
 Color: Red  
 ZIP: 33431

Van – Potential Match Found!  
 Username: vCampbell  
 Name: Van Campbell  
 Email: VCAMPBELL2019@FAU.EDU  
 Color: Red  
 ZIP: 33431

Felix - Potential Match Found!  
 Username: fAchelin  
 Name: Felix Achelin  
 Email: felixachelin@gmail.com  
 Color: Blue  
 ZIP: 33431

CEN 4010 Principles of Software Engineering, Fall 2021

Covid Connections

*By Team Onux*

Project Team 8

**Team**

Van Campbell – vcampbell2019@fau.edu

Samuel Adkins - [adkinss2020@fau.edu](mailto:adkinss2020@fau.edu)

Achelin Felix - [felixa2017@fau.edu](mailto:felixa2017@fau.edu)

Gustavo Rodriguez - [Grodriguez2017@fau.edu](mailto:Grodriguez2017@fau.edu)

Samuel Yambo - [syambo2020@fau.edu](mailto:syambo2020@fau.edu)

Frecks Bertrand II - [fbertrandii2017@fau.edu](mailto:fbertrandii2017@fau.edu)

Michael Merlo - [mmerlo2015@fau.edu](mailto:mmerlo2015@fau.edu)

Milestone 3 Updated Project Proposal / Vertical Software Prototype

10/11/2021

Revision History

#1 – 9/28/21

#2 – 10/11/21

Table of Contents

[Vertical Prototype Features & Functionalities 1](#_Toc85993951)

[Executive Summary 4](#_Toc85993952)

[Use Cases 5](#_Toc85993953)

[Data definition 17](#_Toc85993954)

[List of High-level Functional Requirements 20](#_Toc85993955)

[Non-Member Expectations 20](#_Toc85993956)

[Member Expectations 20](#_Toc85993957)

[List of Non-Functional Specs 24](#_Toc85993958)

[Competitive Analysis 25](#_Toc85993959)

[Facebook 26](#_Toc85993960)

[Twitter 26](#_Toc85993961)

[Reddit 26](#_Toc85993962)

[Planned advantages 26](#_Toc85993963)

[Key Risks 27](#_Toc85993964)

[1. Scheduling 27](#_Toc85993965)

[2. Communication 27](#_Toc85993966)

[3. Time Restrictions 27](#_Toc85993967)

[4. Technical Risks 27](#_Toc85993968)

[High-level system architecture 28](#_Toc85993969)

[Database Organization 29](#_Toc85993970)

[High-Level UML Database Class Diagram 30](#_Toc85993971)

[High-level UML diagrams 30](#_Toc85993972)

[High-level UML Class Diagram 31](#_Toc85993973)

[Component Diagram 32](#_Toc85993974)

[Deployment Diagram 33](#_Toc85993975)

[Team Roles 34](#_Toc85993976)

# Executive Summary

Covid-19 first reached the United States in the last few weeks in January 2020, and since then we have all been hoping for a quick return to normalcy. Now being in September of 2021 normalcy has still not returned and seems farther away than ever. But what people want remains unchanged. People still want to connect with one another and take their mind off the day’s stresses. There are many sites that address these sorts of issues, but they are often broad, and meant for national use. While people can still connect and share on these sites, they often do not have that local connection that you get meeting someone who lives in the same area.

What Floridians need now more than ever, is a site for connecting local people. That is where *Covid Connections* comes in. With *Covid Connections* we wish to build a platform with a variety of services and activities for the people of Florida. First and foremost, we want to be a place where people can come and communicate. Which is why our site shall include forums for our users to discuss and interact with each other. But we also want to build a platform for people to learn. The resources section of our site will hold a plethora of links to different Covid-19 and Mental Health resources for users to discover. Feeling stressed or overwhelmed, go to the games section, where a variety of entertaining games can be played to relax and destress. Though the best way to connect is by seeing each other. Which is why we provide a gallery where our members can post photos and comment on others. We will also have a login system so users can sign up and become members to access some of the services above, But also a special member only homepage.

  The advent of Covid-19, is truly a testament to the unprecedented times we live in. But it is also a good reminder of what is truly important in life. Now that people are feeling more isolated and afraid, they seek what they cannot get. What people need now more than any time in recent history, is that local connection you cannot get just anywhere. That is why the overall goal of building *Covid Connections*, is to create a platform for Floridians to communicate, learn, relax, and most importantly connect during covid.

# Use Cases

Use Case – Homepage

Members will come to the homepage where they will be able to access certain API features (like the weather for their location). Members will be able to access all other services from this page as well.

1. **Description**
   1. Use Case describe how the user and member will select a service on the Home Page
2. **Actors:**
   1. User
   2. System
   3. Member
3. **Precondition:**
   1. User has active internet connection
   2. System is available
   3. User has an active account
   4. User is logged into System
4. **Primary Flow of Events**
   1. Member arrives on the home page
   2. Web page displays services
   3. Member clicks a service’s tab on navbar
   4. Member brought to service they selected
5. **Alternate Flow**
   1. **If user is a member**
      1. Web page displays Home Page
   2. **If user is not a member**
      1. The user will be brought to the Front Page

Use Case – Front Page

Users come to the page upon visiting the site. The user will see the available services but cannot use most of them without an account. The user will be able to sign in if they have an active account and create an account if the user does not have an active one.

1. **Description**
   1. Use Case describe how the user will create an account on the Front Page
2. **Actors**
   1. User
   2. System
   3. Member
3. **Preconditions**
   1. User has an active internet connection
   2. System is available
4. **Primary Event Flows**
   1. User arrives at the front page
   2. User can create an account by selecting Create Account
   3. User fills out necessary information and submits
   4. System will create an active account for new Member
   5. Member will be brought to their Home Page
   6. Use case terminates
5. **Alternate Flows**
   1. **If User is already a member, but not signed in**
      1. User selects login in
      2. User enters credentials and clicks login
      3. Member is brought to Home Page

Use Case – Forum

A Member can select the Forum service from their Home Page, or a User can go to Forums from the Front Page. The Member will use the forum to exchange ideas they have with each other, to access other discussions posted by members, and comment on those posts. There will be a section where Members can find some commonalties between interest they share, called a club.

1. **Description:**
   1. Use case describe how users will utilize the forum to exchange ideas.
2. **Actors:**
   1. Users
   2. System
3. **Preconditions**
   1. User has an active internet connection
   2. User has an active account
   3. User is logged into System
   4. System is available
4. **Primary Flow of Events:**
5. Member arrives at their Home Page
6. Member selects to use Forum
7. The system will display the discussion of the week, the clubs to be joined, and a place where they can comment and post discussions
8. Member posts ideas they want to exchange or clubs they want to join
9. Terminate Use Case: Forum
10. **Alternate Flows**
    1. **User Not signed in**
       1. Attempts to post
       2. The system will prompt the user to sign in or create a new account
       3. User enters their credentials
       4. Return to step 1.
    2. **Club Does not Exist**
       1. If club does not exist, user will have the option to create such club
       2. The system displays a message asking the user to create club
       3. User creates club
       4. Return to step 4

Use Case: Games

A Member comes to the pages and will be able to play the games that are provided. Members can play a variety of JavaScript games to destress and relax.

1. **Description:**
   1. Use case describe the process of how a member selects a game to play
2. **Actors:**
   1. User
   2. System
   3. Member
3. **Preconditions:**
   1. User has active internet connection
   2. User has an active account
   3. User is logged into System
   4. System is available
4. **Primary Flows of Events:**

* User arrives on the frontpage of site
* User signs in with their respective credential
* Member is brought to their Home Page
* Member selects the Games service
* Web page displays the games available to be played
* Member selects the game they want to play
* System loads game
* Terminate Use Case: Arcade Game

1. **Alternate Flows of Event:**
   1. **Member suggests game**
      1. Member selects Suggest Game
      2. Member enters game suggestion in pop-up
      3. Member clicks submit

Use Case – Resources

A Member will come to the resource section to learn more about different Covid-19 & Mental health services in their area. User must be a member to utilize this service. The system will display important information links to them like covid information, mental health resources and many more.

1. **Description:**

Use case describe the process of how users will select a resource to view

1. **Actors:**
   1. Users
   2. System
   3. Member
2. **Preconditions:**
   1. User has an active internet connection
   2. User has an active account
   3. User is logged into System
   4. System is available.

**4. Primary Flow of Events:**

1. User arrives on the front page
2. User logs in using their credentials
3. Member selects the resources section of their Home Page
4. Web page will display covid related information and mental health resources in the area.
5. Member selects the information they wish to view
6. System sends Member to resource
7. Terminate Use Case: Local Florida Resources
8. **Alternate Flows:**
   1. **Member suggests a resource**
      1. Member selects Suggest Resource
      2. Member enters resource link in pop-up
      3. Member clicks submit

Use Case: Gallery

Members will be able to share pictures in a gallery and comment on other members photos that have been posted. A User must have an active account, and login to view Member photos and post their own.

**1. Description:**

Use case describes the process of how a member can post a photo to their Gallery

1. **Actors:**
   1. User
   2. System
   3. Member
2. **Preconditions:**
   1. User has active internet connect
   2. User has an active account
   3. User is logged into System
   4. System is available
3. **Primary Flow of Events:**
   1. User arrives on the frontpage
   2. User logs in using their credentials
   3. Member is brought to home page and selects gallery
   4. Member selects post photo
   5. System asks for file
   6. Member selects photo and clicks submit
   7. System adds photo to Member’s Gallery
   8. Terminate Use Case: Gallery
4. **Alternative Flows:**
   1. **User is not logged in**
      1. User selects Gallery on front page
      2. User is asked to login to view gallery
      3. Member logs in
   2. **Post a photo, but gallery is full**
      1. Web page will display a message telling the user that the gallery is full
      2. User deletes certain pictures in their gallery

Use Case: Sign Up

Members will use the sign page to create an account if they do not possess an active account. This will be the place where they create their account.

**1. Description:**

Use case describes the process of how a member will create an account to become an active user of Covid Connection.

1. **Actors:**
   1. User
   2. System
   3. Member
2. **Preconditions:**
   1. User has active internet connection
   2. User is not a member
   3. System is available
3. **Primary Event Flow:**
   1. User arrives on the front page
   2. User selects sign up on the webpage
   3. System asks for the required information
   4. User inputs their corresponding information
   5. System saves information and user becomes a member
4. **Alternate Flow:**
   1. User inputs Invalid email address
      1. User inputs invalid email address
      2. System will ask to input a valid email
      3. User inputs a valid email
      4. Return to step 4.4
   2. User Inputs an invalid location
      1. User inputs an invalid location
      2. System will ask for a valid location
      3. User inputs a valid location
      4. Return to step 4.4

Use Case: Edit Profile

Member decides that their current profile is not up to date. Member decides they want to edit their current profile.

**1. Description:**

Use case Edit Profile describes the steps a member may follow when updating their profile.

1. **Actors:**
   1. User
   2. System
2. **Preconditions:**
   1. User has active internet connection
   2. User is a member
   3. System is available
3. **Primary Event Flow:**
   1. User arrives on the homepage
   2. User selects Bio
   3. System show bio
   4. User inputs their desire changes
   5. System saves changes
4. **Alternate Flow:**
   1. User edits Personal banner
      1. User arrives on the homepage
      2. User selects personal banner
      3. System displays personal banner
      4. User input new banner
      5. System saves banner
   2. User edits notification setting
      1. User arrives on the homepage
      2. User selects notification setting
      3. System displays notification setting
      4. User input their preference
      5. System saves notification changes

Use Case: Weather API

System will use weather APIs to display weather to users to their corresponding regions.

**1. Description:**

Use Case Weather API describes how the API will be used to display the weather to certain location.

1. **Actors:**
   1. System
2. **Preconditions:**
   1. User has active internet connection
   2. System is available
3. **Primary Event Flow:**
   1. User input location for weather
   2. System search
   3. Weather API provides available data
   4. System retrieve data
   5. System displays relevant information
   6. Terminate Use Case: Weather API
4. **Alternate Flow:**
   1. Weather API cannot find related information
      1. System will display to user that their related information was not valid
      2. System will ask User to input valid information
      3. Weather API will search related information
      4. System displays weather

# Data definition

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Meaning** | **Usage** | **Comment** |
| Member | Actor | Use Case scenario | This person has an account on the website. |
| Prospector | Actor | Use Case scenario | This person is on the website but does not own an account. |
| User | Actor | Use Case scenario | Any person currently on the website. |
| Log in | Service | Site user service | Personal identification verification for users to have member permissions. |
| Create Account | Service | Site user service | Method for prospectors to sign up on the website |
| Navbar | Service | Site user service | Web tool to give users easy navigation within the website |
| Account | Data | Use case scenario | Top level storage of user information. |
| User Id | Data | Site user service | Each account has a user identification. |
| Password | Data | Site user service | Each account is protected with a password. |
| Profile | Data | Site user service | Low level storage of user information |
| Location | Data | Use case scenario | Store the user’s location information.  Format: Address |
| Weather | Sub-Service | Site user services | Let users know the weather of their current location. |
| Weather\_status | data | Site user Service | Data for holding information on current weather |
| Games | Service | Site user service | Members can play simple browser games within the website. |
| highscore | data | Site user service | Stores the high score of the user |
| Developer’s Choice | Data | Site user service | Stores the developer’s choice game. Which is the favorite game among the developers |
| Recently Played | Data | Site user Service | Stores the last game the user played |
| Resources | Service | Site user service | Users can find Florida based local facilities for Covid issues. |
| Resource Items | Data | Site user service | Format: String name, URL  Resource items are links to local resources in area |
| Forum | Service | Site user service | Members post and reply to discuss topics in this organized a place |
| Clubs | Sub-Service | Site user service | Users have a list of local clubs that they can organize to their preference. |
| Club\_desc | Data | Site user service | For holding a description of a club |
| Comment | Sub-Service | Site user service | Members can leave comments in on photos in the gallery |
| Discussions | Sub-Service | Site user service | Members can create discussions for multiple people to discuss commonalities |
| discussion\_id | data | Site user service | Unique identifier for each discussion |
| Gallery | service | Site user service | Members post and view photos. |
| PhotoID | Data | Site user service | Format: Jpeg |
| Like/dislike | Data | Site user service | Will hold likes and dislikes for a photo |
| Front page | User interface | User interface | Front end landing page for non-members. |
| Home page | User interface | User interface | Front end landing page for member interaction. |

# List of High-level Functional Requirements

## Non-Member Expectations

1. **Create An Account**
   1. The website should allow a non-member user to create an account. The expected inputs are username, password, and ZIP code.  
      The website will not allow the user to be created if the username already exists within the database. There will also be a password re-entry field. The website will prevent user creation if both password fields do not match. The website will provide error feedback if the ZIP code is not a valid ZIP.
   2. **Stimulus/Response Sequence – Account Creation** 
      1. User enters desired UserID
      2. User enters password
      3. User re-enters password for validation
      4. User enters their default ZIP code
      5. Website will check that UserID is unique, passwords match, and ZIP is valid
      6. Website will store username, encrypted password, and ZIP in the database layer
      7. Website will inform user that account was created
      8. Website will redirect to homepage as a logged in user
   3. **Functional Requirement Label**
      1. REQ 1.1 Create an Account – Priority: 1

## Member Expectations

1. **Use Forums** 
   1. Registered users can use the forums to view and post new comments and threads. The system will not allow unregistered users to view the forums. Registered and authenticated users can use the forums. Members will be able to view and participate in different discussions, including a discussion of the week. Which shall be chosen based on which discussion has the most replies by the end of the week. Members will also be able to create and join different clubs, to communicate to other members with similar interests.
   2. **Stimulus/Response Sequence - View Forums**
      1. User navigates to forums
      2. Website will display various subforums and threads
   3. **Stimulus/Response Sequence – Search Forums**
      1. User can search for posts with keywords – this may be out of scope for the final product.
   4. **Stimulus/Response Sequence – Create New Post/Reply**
      1. User can either reply to an existing thread or create a new one
   5. **Stimulus/Response Sequence – View/Reply in Discussion**
      1. Members can find a discussion by searching for title.
      2. Member can join a discussion by replying to discussion posts.
   6. **Stimulus/Response Sequence – Discussion of the Week**
      1. Members can find the Discussion of the Week by selecting the forum page
      2. The discussion of the week is the discussion with the most replies in the past week.
   7. **Stimulus/Response Sequence – Create/Join Club**
      1. Members can either create their own club to invite other Members too. Or can join other clubs via an invite.
   8. **Functional Requirement Label**
      1. REQ 2.2 View Forums – Priority: 1
      2. REQ 2.3 Search Forums – Priority: 1
      3. REQ 2.4 Post to Forums – Priority: 1
      4. REQ 2.5 View/Reply in Discussion – Priority: 2
      5. REQ 2.6 Create/Join Club – Priority: 2
      6. REQ 2.7 Discussion of the Week – Priority: 3
2. **Play Games**
   1. Registered users can play games. Unregistered users cannot access the games page. Users can choose from various games. User progress/scores can be tracked in the database layer if development time allows. Members will be able to select from a favorite game among the developers, and the game they played most recently. Members will also be able to suggest a game for the developers to add.
   2. **Stimulus/Response Sequence – Play Games**
      1. User navigates to games tab
      2. User can choose from games to play
   3. **Stimulus/Response Sequence – Play Developer Choice Game**
      1. User navigates to game tab, and chooses Developer’s Choice Game
      2. User can choose the game we (the developers) liked the most.
   4. **Stimulus/Response Sequence – Suggest Game**
      1. User navigates to game tab and clicks “Suggest A Game”
      2. System shows pop-up for user to enter suggestion.
   5. **Stimulus/Response Sequence – Play Recently Played Game**
      1. User navigates to game tab and chooses Most recently played.
      2. Users can choose the game they most recently played. Tracked by system.
   6. **Functional Requirement Label**
      1. REQ 3.2 Play Games – Priority: 1
      2. REQ 3.3 Play Developers Choice Game – Priority: 1
      3. REQ 3.4 Suggest Game – Priority: 2
      4. REQ 3.5 Play Recently Played Game – Priority: 2
3. **Use Image Sharing** 
   1. Registered users can view and post images to share with other registered users. Comment may be made to the images found here. Members will be able to see trending images, sorted by most comments. They will also be able to like and dislike shared images, which will be kept track of by the system.
   2. **Stimulus/Response Sequence - View Images**
      1. User navigates to image sharing tab
      2. Website will display previously posted images
   3. **Stimulus/Response Sequence – Post Images**
      1. User navigates to image sharing tab
      2. User can upload their own image
   4. **Stimulus/Response Sequence – Comment on Images**
      1. User navigates to image sharing tab
      2. User selects an image to comment on
      3. User posts a comment
   5. **Stimulus/Response Sequence – Trending Images**
      1. User navigates to image sharing tab
      2. Trending images will be sorted by most comments.
   6. **Stimulus/Response Sequence – Like/Dislike Images**
      1. User navigates to image sharing tab
      2. User selects like/dislike on a shared image
      3. System records like/dislike on photo.
   7. **Functional Requirement Label**
      1. REQ 5.2 View Images – Priority: 1
      2. REQ 5.3 Post Images – Priority: 1
      3. REQ 5.4 Comment on Image – Priority: 1
      4. REQ 5.5 Trending Images – Priority: 2
      5. REQ 5.6 Like/Dislike Images – Priority: 3
4. **Access COVID Resources**
   1. Registered users can access various COVID related resources. Members will also be able to suggest a resource to be added by submitting a link to the developers.
   2. **Stimulus/Response Sequence – View Resources**
      1. User selects Resources tab. Website will populate with various external COVID resources.
   3. **Stimulus/Response Sequence – Suggest Resource**
      1. User selects Resources tab.
      2. User clicks “Suggest A Resource”.
      3. System shows pop-up for user to enter suggestion with link.
   4. **Functional requirement Label**
      1. REQ 4.2 Resources – Priority: 1
      2. REQ 4.3 Suggest Resource – Priority: 2
5. **Use Home Page**
   1. On their home page members will be able to do a variety of actions. This includes editing a profile, by entering their name, date of birth, and a short biography. This information will be validated and stored when the user selects save. Members will also be able to view basic weather information based off the address they entered on account creation. Forum notifications will be viewable from the home page. This includes replies to your posts, and any posts from clubs the member is a part of. Latest Gallery Notifications shall show on the home page, this shall correspond to the last photo commented on. Finally, the member’s favorite resource item will be shown on their home page.
   2. **Stimulus/Response Sequence – Display Weather**
      1. Widget will display weather on home page.
      2. No input required. Widget will be based on the users ZIP code.
   3. **Stimulus/Response Sequence – Edit Profile**
      1. Member will navigate to their home page.
      2. Member will select “Edit Profile” button.
      3. System will display pop-up for user to enter name, date of birth, and bio.
      4. User enters name, date of birth, and bio.
      5. User clicks save.
      6. System stores and validates input.
      7. System refreshes home page with new profile information.
   4. **Stimulus/Response Sequence – Forum Notifications**
      1. Member will navigate to their home page.
      2. System will load forum notifications on home page
   5. **Stimulus/Response Sequence – Gallery Notifications**
      1. Member will navigate to their home page.
      2. System will load the last photo commented on.
   6. **Stimulus/Response Sequence – Favorite Resource**
      1. Member will navigate to their home page.
      2. System will load the resource item most used by member
   7. **Functional Requirement label**
      1. REQ 7.1 Display Weather – Priority: 1
      2. REQ 7.2 Edit Profile – Priority: 1
      3. REQ 7.3 Forum Notifications – Priority: 2
      4. REQ 7.4 Gallery Notifications – Priority: 2
      5. REQ 7.5 Favorite Resource – Priority: 2

# 

# List of Non-Functional Specs

**Compatibility Requirements:**

1. *Browsers:* The site will be created using bootstrap to ease compatibility between desktop and mobile based browsers. The initial scope of the site covers compatibility with Chrome and Firefox, which should also provide compatibility with other major browsers. No alternative site functionality will be considered if the browser does not have JavaScript installed.
2. *Computer and OS:* The site is being built with no particular operating system in mind. If the OS can run any popular browser, it will run the site.

**Expected Load:**

1. As the site should not be receiving large use traffic simultaneously, we will make all possible accommodations to tolerate multiple users at once. Scripts will be executed to simulate multiple simultaneous users/requests to the server.

**Storage Requirements:**

1. System will be stored on LAMP server. Backup options are not yet determined but may include cloud storage for image and database files.

**Availability Requirements:**

1. System will be stored on LAMP server. As such, it is subject only to LAMP server maintenance constraints.
2. In the event of LAMP downtime, website will be inaccessible.

**Performance Requirements:**

1. *Responsiveness:* The system will be coded using bootstrap which should help accommodate any screen size.
2. *Test Requirements:*  Testing will include all functional requirements as well as simulated multiuser load and response time.
3. *Reliability:* The system should be operational 100% of the time as long as LAMP is up and running. Failures should be addressed prior to deployment.
4. *Bug Count:* No more than 10 bugs during development at any time. No more than 5 bugs when system is deployed.

**Security Requirements:**

1. *Login/Password:* Users will be required to authenticate via username and password. Passwords will be salted and hashed on the database layer to protect user privacy and security as much as possible.
2. *Test Requirements:*  Testing will include all functional requirements as well as simulated multiuser load and response time.
3. *Reliability:* The system should be operational 100% of the time if LAMP is up and running. Failures should be addressed prior to deployment.
4. *Bug Count:* No more than 10 bugs during development at any time. No more than 5 bugs when system is deployed.

# Competitive Analysis

This chart is a for the comparison of different social media sites and their different features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Features | Our website | Facebook | Reddit | Twitter |
| Discussion forums | yes | yes | yes | no |
| Games | yes | yes | no | no |
| Solely for the state of Florida | yes | no | no | no |
| User images and comments | yes | yes | yes | yes |
| Links to mental health resources | yes | yes | no | no |
| Respects user privacy | yes | no | no | no |

## Facebook

Facebook does not directly have any kind of discussion forums, but they do have groups where discussions can be held. Facebook has a variety of games ranging from poker to matching games. Facebook does indeed include a variety of links which are linked to both mental health and information about coronavirus. As to be expected from most social media websites Facebook also allows users to upload pictures and comment on other pictures.

## Twitter

Twitter usually has very good discussions through replying to tweets and using hashtags to make what the tweet is about more specific. When it comes to games twitter doesn’t have any, but they do allow images and videos to be posted like other social media sites. Twitter is a worldwide platform but usually a lot of emphasis is put on celebrities and whatever they say. There isn’t a huge presence of mental health resources unless you specifically look for them, and some apply for covid 19 information.

## Reddit

Reddit’s focus is forum and discussions in small communities called subreddits. There are no games in these subreddits but there are videos and images that are allowed to be posted on these subreddits. Although there are subreddits for groups in Florida reddit is primarily used with intentions of staying semi-anonymous although some people choose to reveal more personal information about themselves. For user privacy there are targeted ads so it’s not very private but more private than some other social media sites.

## Planned advantages

This will be our social platform for connecting others in this difficult, and lonely time. Now there are other websites that do the same thing, but we plan to be different form the rest. Our target audience will be solely people that reside in Florida! Other websites have thousands of users who connect across the world, and oftentimes we tend to forget the people around us such as neighbors or close friends. For this very reason we want people to become more familiar with those around them. Another advantage we will have been the lack of advertisements. Advertisements are usually a source of income for these websites, but not for us. Our sole purpose is bringing people together not making the big bucks. In doing this we really emphasize the user’s privacy.

# 

# Key Risks

#### Scheduling

As all members have different schedules, classes, and lifestyles it is difficult to gather everyone for a meeting at the same time on the same day.

This issue has been partially remedied through the use of polls for members to choose the times best suited for them for meetings. This solution will work splendidly as soon as the next issue is solved.

#### Communication

Members do not always update the group on progress or schedules. Whether this be from being swamped with work or sheer forgetfulness is anyone’s guess, however it needs to be addressed for this project to progress smoothly.

Members should try to make a habit out of checking Slack and or Jira at either the beginning or end of the day at least. Through this there will be no missed polls, and updates will be able to be made and seen by all.

#### Time Restrictions

With the scope of this project being as ambiguous as it is, there are many concepts we want to add to this project but will be hard to implement with the due date set.

As we go down the list of priority requirements, some of the least important ones will most likely have to be dropped due to lack of time.

#### 

#### Technical Risks

Most members of this project do not have much experience working on a project of this caliber and thus are observing more than doing in terms of the technical aspects of the project such as programming the database

For now, the best course of action is to give those who know how to properly do these technical aspects the opportunity to do so while others can learn so they can help in the future.

# High-level system architecture

Lists of main software products, tools, languages, and systems to be used, list of core APIs available at this point, supported browsers etc. You also have to decide on which frameworks you will use if any. These provide both user interface, as well as cross-platform and cross browser layout/CSS. All external code you plan to use must be listed along with their license.

High-level system architecture

1. **https://lamp.cse.fau.edu/~cen4010\_fa21\_g08/ Lamp Server:** The FAU provided Lamp Server is the host server for our Fall 2021 Principles of Software Engineering project.
2. **Slack**: The team will communicate interpersonally with Slack, a proprietary business communication platform with chat rooms organized by topics.
3. **MySQL Database:** The MySQL open-source relational database management system will store user information. Users will have the ability to store input and store their information via logging into their profile on the website.
4. **Visual Studio 2019 (IDE):** Visual Studio 2019 is the IDE that the developers will build their code in. The following list of languages will be used in the IDE during development of the website:
5. Hyper Text Markup Language (HTML) - standard language to design web browser.
6. Cascading Style Sheets (CSS) – style sheet language to describe presentation by assisting HTML
7. Personal Home Page (PHP) – scripting language for web development and server-side requests
8. JavaScript (JS) – just-in-time scripting language for web development and assists HTML
9. jQuery – a JavaScript library to assist HTML manipulation and client-side functions
10. **Chrome and Firefox Browser Compatibility:** The system requires full operation with at least two major brows, including Google Chrome, Mozilla Firefox, Safari, Opera, and Internet Explorer. Web based functionality will have full support with Google Chrome and Mozilla Firefox.
11. **GitHub:** GitHub facilitates code collaboration and productivity with online repositories that allow the storing, mering, commenting, organizing, etc. of teams the teams’ code.
12. **Canvas:** This higher education software is the platform that our stakeholder, the instructor, communicates with the development team.
13. **Jira:** Jira is a proprietary issue tracking product software tool that maintains the teams bug tracking, issue management, and organizes the overall Scrum development process.

Media storage

Images will be uploaded by the user and saved with our project code in the “./images” subfolder. Our site will accept JPEG images. This may be expanded to include other formats as time allows. The site will be monitored for storage limitations, but no size restriction will be enforced initially.

Search/filter architecture and implementation

The current search algorithm will be a naïve string search. While this search is not very efficient, the data set is small enough that this should not cause any noticeable impact to the system. The homepage, forums, and image gallery will include search/filter inputs as follows:

* Homepage – users can search for other users. Naïve string search will be performed on:
  + user.user\_name
  + user.user\_first\_name
  + user.user\_last\_name
* Forums – users can search for other posts within the forums. Naïve string search will be performed on:
  + forum.forum\_name
  + post.content
  + post.title
  + reply.content
  + user.user\_name
  + user\_first\_name
  + user\_last\_name
* Image gallery – users can search for images. Naïve string search will be performed on:
  + image.image\_name
  + image.image\_caption
  + user.user\_name
  + user\_first\_name
  + user\_last\_name

# Database Organization

## High-Level UML Database Class Diagram

Diagram

Description automatically generated

# High-level UML diagrams

## High-level UML Class Diagram

Diagram

Description automatically generated

## Component Diagram

Diagram

Description automatically generated

## Deployment Diagram

Diagram

Description automatically generated

# Team Roles

**Team Lead, Scrum Master** - Van Campbell

**Product Owner, Front-end Developer** - Samuel Adkins

**Front-end Lead** - Achelin Felix

**Back-end Lead** - Gustavo Rodriguez

**GitHub Master, Developer -** Samuel Yambo

**Developer –** Michael Merlo

**Developer** – Frecks Bertrand II