Principles of Software Engineering Fall 2021

*[FAU: CEN 4010]*

**Project Name: Peace of Mind**

Team Name: *Group 03*

Team Lead: Juan Idrovo

**Team Members:**

Ryan Barath - rbharath2018@fau.edu

Mauricio Retana Rodriguez - mretanarodri2018@fau.edu

Juan Idrovo - jidrovo2019@fau.edu

Ryan Milrad - rmilrad2016@fau.edu

Armaghan Ali - alia2019@fau.edu

Joseph Brown - josephbrown2020@fau.edu

Milestone 3

October 11, 2021

Revision History

| Change # | Date changed | Summary |
| --- | --- | --- |
| 00 | 09-25-2021 | Creation of the document |
| 01 | 10-20-2021 | High Level Functional Requirements - Removed Entertainment Ratings  Data Definition - Removed Entertainment features |
| 02 | 10-21-2021 | High Level Functional Requirements - Added more to the Administrator section, made changes to the Browse Website section  Non Functional Requirements - Minor change to the performance attribute |

## Vertical Prototype Testing Functionality Notes:

* **Project URL (Vertical Demo):** <https://peaceofmind-cen4010.xyz/>
* **List of features to be tested**:
  + Guest users should be able to register in the blog and an account will be created with their information in the database
  + After the user becomes a member he/she should be able to log into the Peace of Mind blog.
  + If a user forgets his/her password, they should be able to use recover password functionality which sends them an email (to their email they have saved in the database) with a unique token which allows them to modify their password.
  + Once the user becomes a member and has logged into Peace of Mind he/she should be able to look at the posts categories and blog posts. These categories and posts are retrieved from the database and rendered in the member’s browser.
  + Users should be able to look at posts for the chosen category. Users should also be able to see all the blog posts that are published.
  + Administrators should be able to login and have access to the admin dashboard.
    - Please note that you need to login with an Administrator account. For this purpose you can use our testing account which is:
      * Username: johndoe
      * Password: test
  + Admins should be able to create a new blog post from the Admin Dashboard. Please note that all the fields are required to be filled in order to test this functionality.
  + All users should be able to see the newly created post.
  + Users should be able to look at their profile information and they should also be able to modify their public name, last name, contact email, biography, and profile picture. Users should be able to change one or multiple fields from their profile information.
  + Users should be able to see the profiles of other member users on the blog. For now the only way to access other profiles is by knowing their username. One way of doing this is by pressing the other person’s name on a comment.

**TABLE OF CONTENTS**

[Vertical Prototype Testing Functionality Notes:](#_jy5x4ox0sznj) 2

[**1 Executive Summary**](#_ioq9sn3amgmr) **4**

[Our mission](#_aeatkwjuwfn0) 4

[The company and management](#_svvsr6wu30o4) 4

[Our products](#_isk9h8z4093h) 4

[The market](#_5ae4pd6nblbj) 4

[Our competitive advantages](#_brq4uyq5oz1q) 4

[**2 Competitive analysis**](#_36sglriw7ubg) **5**

[Quora](#_7txl8j6c1xox) 5

[Twitter](#_4vdcilxk3rv5) 5

[Reddit](#_wj4hp2jqhryk) 6

[Instagram](#_yaku2l91lth8) 6

[**3 Data definition**](#_8c4pzs4f43b5) **6**

[**4 Overview, scenarios and use cases**](#_orwadpeam91l) **8**

[**5 High-Level Functional Requirements**](#_ylt9o0fqyvzf) **9**

[**6 List of Non-Functional Requirements**](#_ohjrqcwlr4bz) **13**

[**7 High-level system architecture and database organisation**](#_uz2tncupaijz) **15**

[**8 High-level UML diagrams**](#_1rdhxckjhj1c) **17**

[**9 Identify actual key risks for your project at this time**](#_fvrv4d5gaqcr) **17**

[**10 Team**](#_k27udqqulxqj) **17**

[**11 Checklist**](#_3almqiocrugg) **17**

[**12 Tasks before submission**](#_8u8m1nc82tz9) **18**

[**13 Submission**](#_hoquis14nxy9) **18**

# 1 Executive Summary

### **Our mission**

*Peace of Mind* is a one-stop program that aids users through wellness activities that have arisen during the COVID-19 pandemic. Individuals around the world have suffered not only physical ailments but mental as well. *Peace of Mind* aims to assist individuals regain control of their sound state of mind to facilitate the transition through the unsure times.

### **The company and management**

*Peace of Mind* is run by a collective group of scholars from Florida Atlantic University that are seeking degrees within the engineering field.

### **Our products**

*Peace of Mind* is a web-based catalog service focused around an interactive blog. The blog is meant for user interaction to assist other users with comparing, offering, and requesting activities that help global individuals connect and regain control of a healthy state of mind. Categorized by content, the blog aims to identify personal needs in topics such as outdoor, cooking, reading, local vaccination stations, and more. *Peace of Mind* is built to accommodate individuals around the world and connect either anonymously or with great presence.

### **The market**

Currently, we have not been able to identify a blog-based service that is focused around the struggles that COVID-19 has presented. Blog sites, such as Facebook and Reddit, have some opportunities for individuals to connect but fail to highlight communication regarding the well-being of individuals directly affected by COVID-19. *Peace of Mind* strives to display the acceptance of mental illness and the many methods for combating some symptoms of isolation.

### **Our competitive advantages**

In a world where COVID-19 is one of the most prevalent topics, there is a large market to fill that offers aid to individuals struggling with mental illness brought upon by COVID-19. As social media grows in complexity, pressing issues can become lost in the mix and lose steam as a worthy cause. *Peace of Mind* is focusing on issues pertaining to isolation, specifically side effects of life during COVID-19.

# **2** **Competitive analysis**

|  | Peace of Mind | Quora | Twitter | Reddit | Instagram |
| --- | --- | --- | --- | --- | --- |
| Search engine | yes | yes | yes | yes | yes |
| Community | yes | yes | yes | yes | yes |
| Wellness space | yes | no | no | yes | no |
| Ease of use | yes | no | yes | no | yes |
| Responsive | yes | yes | yes | yes | yes |
| Thought  Provoking | yes | no | no | yes | no |
| Specific About Covid | yes | no | no | no | no |

### **Quora**

It is a question and answer based website. Users come with the queries and they can either post it in there or search it in the search bar at top right. It has a virtual community and it is not easy to use especially for beginners because sometimes things are not straightforward. Users are responsive and it is not specific about covid instead it covers a wide range of topics. The posts in there are not thought provoking but just an answer to a question.

### **Twitter**

Almost like Reddit is a space where users make posts whether it be discussion based or a status update or even just random thoughts. For the full-time users it is an easy to use application there isn't much to it when it comes to making tweets which to the average user may not be that much of a thought provoking tweet. With that being said it is also a space where it almost anything can be send so it doesn’t cater to being a censor friendly which can be good or bad to ones mental health, however it does cater to the widespread of communities. By using the search bar one is able to type in any specific thing and there might be a tweet about it. The origin of this platform also traces back further than covid so it is clear to say that it is not specified around that area.

### **Reddit**

This platform is probably the most used site out of all them and serves many different communities and ideologies. With a click of a button ones ideas are brought to a widespread of subreddits dealing with the similar topic making it easy for anyone to use it. Since this is less of a social media platform it is more common for users to feel safe to post subreddits without fear of criticism from others. Users are responsive and with the discussion boards open users can ask any type of thought provoking questions according to the main topic and expect to get answered. Much like the previous and next website this one was created way before covid so it wasn’t made specifically for covid.

### **Instagram**

With the concepts of tags this platform almost acts like a search engine with by typing one word statements or phrases a user gets a widespread amount of pictures or videos dealing with that concept. It houses a widespread of communities and by even looking at ones followers or followings anyone can understand which user they could follow. By no means however is this platform safe from negativity from the world, especially with the introduction of lives and the ability to include one another in yours. This platform is easy to use however may come with minor complications and users on here are responsive. This is not a thought provoking platform and it was made before covid and doesn’t have anything specifically made for the topic.

# 3 Data definition

| **Name** | **Meaning** | **Usage** | **Comment** |
| --- | --- | --- | --- |
| Non-member / guest user | actor | Use case scenarios | Type of user that has not registered to the blog. Only has access to Covid-19 news section and information. |
| Member / registered user | actor | Use case scenarios | Type of user that is registered in the blog. Has access to all different categories of the blog and is able to look at all the posts existing in the blog. Members will also be able to comment on posts and see other members' profile information. |
| Administrator | actor | Use case scenarios | Administrator of the blog. Has capabilities to add/remove blog posts and comments to all categories. Administrators will also be able to edit and publish posts. |
| Registration | service | Site user service | Allows guest users to register to the website. |
| Log in | service | Site user service | Allows registered users to have access to the different categories of the blog. |
| Forgotten password | service | Site user service | Allows members to recover their password through a token sent to their email. |
| User profile | service | Site user service | Members shall be able to see their own profile page as well as other members’ profile pages. Members will be able to edit their profile information such as their biography and user image. This will allow member users to further connect with other members by having access to members’ emails. |
| Blog categories | service | Site user service | The blog will consist of different categories. |
| Search Bar | service | Site user service / User interface | Blog categories will have a search bar that will allow users to look for blog posts by title. |
| Home page | User interface | User interface | Contains a banner image and will contain an overview of our project |
| Information/news page/wellness space | User interface | User interface | Webpage that will show news/information related to Covid-19 |
| Navigation bar | User interface | User interface | Navigation bar that contains the links to all the different pages including all the blog categories, user registration, user login, and user profile. |
| Blog posts | service | Site user service | Administrators will be able to post useful information about Covid-19 related categories.  Members shall have access to commenting/liking blog posts. |
| Comments | service | Site user service | Members and administrators shall be able to comment on any blog post. |
| Likes/dislikes | service | Site user service | Members shall be able to like or dislike blog posts. |
| Account | data | Use case scenarios | Stores information about the users such as name, last name, email, about, profile picture etc. |
| Live Chat | service | Site user service | Users shall be able to communicate with each other through a live chat that shows their username and other users shall be able to access that user’s profile information |

# 4 Overview, scenarios and use cases

The purpose of Peace of Mind, a web-based blog, is to provide a service to the people that were affected by the COVID-19 pandemic. We, as a community, are going to provide tools to people that were affected and help them clear their minds and therefore improve their overall wellness. Since the blog is categorized by content, people that are willing to help are going to be able to comment on their knowledge and recommendations about certain topics and the people that need help about certain situations are going to be able to find the help that they are seeking thanks to the organization of the blog.

During the pandemic, a user locked himself in due to fear of the virus. He didn't get out of his house, did not socialize and didn't get any natural sunlight. A few days passed and he was feeling normal, like any other day. Then, weeks started to pass, he had nothing to do at home since he had lost his job. Like people say, "A lazy mind is an anxious mind". This user started to feel low, stopped eating healthy, and as a result, he lost a lot of weight. He was very unhealthy. One day, while using his computer, he saw a wellness blog that people that were having problems with the pandemic wrote there and brought support between each other via commenting. He made himself an account and started to read comments about the subjects that he was interested in, he used the search engine to locate those important topics to him. He read that there were hundreds of persons that also did the same as him and that were feeling bad, the same way as him. After reading that, he found that people were writing comments about how their lives had changed because of the advice that some people had been writing there. He read that by getting some minutes of sunlight a day, we would obtain vitamin D and also a better mood. He also found very interesting and delicious cooking recipes and tried those at his place, he gained weight and recovered his healthy life that he had. By communicating and feeling throughout this community, this user was able to come to life again. He got very good ideas for improving his life and wellness and also, he got social interactions via the comments. This combination of factors saved the user's life because it gave him a purpose. If they weren't there, who knows what would have happened to him.

If a guest enters the blog, he would only be able to look around. If he/she wants to comment something he/she would need to sign up. In order to create an account, the user needs an email, secure password, name, and username. To have better communication with other users in the blog, the blog encourages you to add more information to your profile. If the user already has an account, he/she only needs to log in with the username and password. An user with an account is a user with full functionality of the blog, he/she would be able to comment, read, search, etc. in the blog.

Case for searching. This is the case where a user wants to search inside the page. The search could be about another user, a post, or a specific comment. The search button is located on the top of the home page, next to other buttons. When the user clicks the search button a text box appears where the user can see what he is writing. Case for commenting. This case is when the user wants to share a comment to his community. The user will need to click on the comment button and then a box entry will appear, where the user will be able to write his thoughts or share interesting things.

# 5 High-Level Functional Requirements

**Non-Member**

1. **Creating an account (priority 1)**

**1.1** The webpage shall allow the user to create an account to access the features of the website. The parameters for creating an account are: First name, last name, username, email, password, confirm password. The system shall not grant users the right to create an account if the username matches with another user, if the password is too short, or if the password is too weak. The system shall not permit users to create an account if one or more parameters are missing from the field. If a user has an account, there is a link to the login page. If the user forgot their password, there is a link to reset password.

**1.2 Stimulus/Response Sequence**

1. Users enter their email address and password
2. Users enter their first and last name
3. Error message should be prompted if a user exists already with the same id.
4. User shall re-enter password for confirmation.
5. System shall validate the password
6. System shall store the user name, username, password and email in a database
7. System shall redirect new user to the main page

**1.3 Function requirement label**

i. REQ 1.3 Creating Account

1. **About Us (priority 1)**

**2.1** Non members shall have access to the about us page. This about us page describes to the user what the website is about and what features the website uses. Any features listed are prohibited to the non-members of the website.

**2.2 Stimulus/Response Sequence**

1. Users use navigation bar and clicks “About us”
2. Users redirected to About us page
3. After reading the information, user shall have the option of going back to the main page

**2.3 Function requirement label**

i. REQ 2.3 AboutUs

**Member**

1. **Edit Profile (priority 1)**

**1.1** Users shall have the right to edit their profile by adding an image of themselves, adding a section that describes them, adding their location, hobbies/interests.

**1.2 Stimulus/Response Sequence**

1. Users click on the drop down menu and it displays the option for edit profile.
2. User click on the edit button
3. User is taken to a new page where they can edit their information
4. User should be able to change name, email, and photo
5. After changes are made, user shall be able to confirm changes
6. User is directed back to their profile page

**1.3 Function requirement label**

i. REQ 1.3 Edit Profile

1. **Browse Website (priority 1)**

**2.1** The user shall be able to access most parts of the website (Some functions are restricted to only administrators). The user will be able to view blog posts made by administrators and are able to comment on them too.

**2.2 Stimulus/Response Sequence**

1. Users enter the search criteria into the search bar
2. Hit enter and it displays the specific search related results.
3. User can also use navigation bar to navigate to different pages of the website
4. User clicks on the button that will redirect to a different page (EX: User clicks on profile which will redirect them to their profile page)

**2.3 Function requirement label**

i. REQ 2.3 Browse by Search

1. **Comments (priority 1)**

**3.1** Users shall be able to add comments to a blog post provided that their accounts are valid. Users shall also have the ability to reply to other comments posted by other users.

**3.2 Stimulus/Response Sequence**

1. Users should be able to locate the comment section and post their comments.
2. User click on button called “add comment”
3. User is shown text box and uses keyboard to type in the comment
4. User hits “submit” to submit the comment
5. User is given the option to edit comment
6. User hits “edit comment”
7. User changes comment
8. User hits “submit” to submit the edited comment

**3.3 Function requirement label**

i. REQ 3.3 Post Comment

1. **Blog (priority 1)**

**4.1** Users shall be able to create, comment, and view blogs on this website. Blogs that are created shall be available to the public. Members of the website will be able to comment and like posts made by that person.

**4.2 Stimulus/Response Sequence**

1. Users should navigate to create a blog or view blog bar and create their own blog or view others blogs.
2. User clicks on button “add post”
3. User is shown text box
4. User uses keyboard to add something
5. User shall have option to link something to the blog post
6. User clicks on link icon to add a link
7. Link is generated on the post
8. User hits “submit” to post their blog
9. User can comment on other blog posts
10. User hits “add comment”
11. User is shown text box
12. User uses keyboard to enter comment]
13. User hits “submit” to submit the comment
14. User has the option to edit blog post or comment
15. User hits the button “edit”
16. User edits comment and hits “submit” to submit changes to the comment or post

**4.3 Function requirement label**

i. REQ 4.3 Create Blog

1. **Contact Information (priority 2)**

**5.1** User shall be able to find the contact information for the developers via email.

**5.2 Stimulus/Responsive Sequence**

1. Users shall navigate to the Contact Us bar and see the developers email.
2. User fills out name
3. User fills out email
4. User enters comment in text box
5. User submits query
6. System send the comment as an email to the administrator/owner of the website

**5.3 Functional requirement label**

i. REQ 5.3 Contact Us

**6. Live Chat (priority 3)**

**6.1** The user shall have the ability to chat with other users in a live chat room. Non-members are restricted to this feature.

**6.2 Stimulus/Response Sequence**

1. Users can scroll to the live chat box and get in touch with available users.
2. User uses a keyboard to chat with other users
3. User clicks the “enter” button or hit “enter” on their keyboard to submit the comment that they wrote
4. Comment is displayed on other users’ end

**6.3 Function requirement label**

i. REQ 4.3 Live Chat

**Administrator**

1. **Administrative Rights (priority 1)**

**1.1** Administrators shall have the right to add or remove posts that are deemed to be too offensive.

**1.2** Administrators shall have the ability to edit posts

**1.3** Administrators shall have the ability to grant admin status to any user if they are deemed fit for the position

1. **Maintenance of the website (priority 1)**

**2.1** Administrators shall have the ability to perform maintenance on the website if necessary.

# 6 List of Non-Functional Requirements

*Performance* - Web pages must be optimized to be loaded in less than 300ms but no more than 1 second. Of course, internet speed may vary. A slower connection will take a long time to load the page. This product should be a light application that is able to produce a light load to the system. The minimum requirements are Windows 8 Operating System with 4GB RAM and the Recommended requirements is Windows 10 with 8GB RAM. A video card is recommended with at least 2GB VRAM to make use of all the applications of the system. The response time between users for the live chat should take less than 200ms but no more than 500ms.

*Portability* - Webpage shall be designed to be accessed from any web browser supported on any operating system. For example, all users should be able to access the webpage from Windows 10, Mac OS, Linux, etc. and is accessed from any web browser such as Internet Explorer, Google, Bing, Firefox, etc. Any plugins used for the website should be compatible with every operating system and with every web browser available to the user.

*Usability* - Webpage shall be designed to have easy functionality of the website. The interface of the website shall be easy to use and easy to navigate through. High contrast colors should be use to allow the user to see properly.

*Accessibility* - Webpage shall be created to accommodate those with disability. Webpage shall have options for those who are hearing impaired, sight impaired or for any other disability. High color contrast shall be used for those that are sight impaired. Text-to-speech technology should also be implemented for those who are sight impaired.

*System load* - Webpage shall be optimized to induce minimal load to a system. Webpage functionality shall support up to 1000s of people using the service. Of course this depends on the location. A smaller population will see less load on the webpage while a larger population will see more load to the webpage. Webpage shall be optimized to ensure that services are not significantly slowed down or completely stopped.

*Security Requirements* -

* Regular users shall not have access to other user’s passwords. Users shall not have access to administrative rights and they shall not have access to client information except for themselves.
* Administrators shall have access to the client and server side of the product. Administrators shall monitor web pages and make sure that there are no vulnerabilities in the website and monitor that the content posted by users is appropriate for the blog’s theme.
* Passwords shall be encrypted to make it more difficult for the attacker to access it. The blog shall protect itself from SQL injection by using prepared statements for all the text input by the user.
* Webpage shall include a password limit where if the user tries to enter their password more than 3 times, the webpage will lock them out and they will have to wait 24 hours to try again. This is to prevent brute-force attacks on the system.
* Website shall not save anything on a text file.
* Website shall save the data onto a database.
* Users shall have to verify their email address before creating an account.

*Storage* - All data must be stored in a database and vital information must be encrypted.

*Availability* - Webpage must be available 24/7 for 365 days (or all year long).

*Maintainability* - A webpage that is running until the end of time shall be subjected to maintenance, therefore it is imperative that downtime shall be enforced to update or to fix issues regarding the webpage. If it is for maintenance, downtime shall be less than 30 minutes but no more than 1 hour. If there are any other issues, downtime shall be less than 6 hours but no more than a day, depending on how serious the issue is.

*Fault tolerance* - In the event of webpage failure (such data breach, etc.) webpage shall be temporarily taken down to resolve issues and to further prevent attacks to the system. Administrators shall implement policies and procedures in time of a cyber attack. Administrators/Owners of the website shall notify all users if data has been breached to be in compliance with GDPR laws and regulations of other countries or states.

*Privacy* -

* User information shall be hidden and protected from outside sources and their information shall not be sold to third party sources.
* Each user is responsible for the protection of their personal information.
* Users shall have the right to request and download data gathered from the webpage. Users shall also have the right to request termination of their account.
* Users shall have the right to make their information either private or available to the public.

*Quality* - Webpage shall be designed to be up to standards in terms of web application development. Webpage shall provide a user interface that is easy to use and easy to learn. Webpage shall not be bloated with unnecessary features. Webpage shall be scaled to fit devices. For example, a web page should be scaled to fit 1920 x 1080 resolution, 2k resolution, 4k resolution, etc. Ideally, any resolution shall not detract the aesthetic quality of the website, no matter what device it is being accessed from.

*Support* - Webpage shall have support running 24/7 for 365 days (or all year long). It is the administrator/Owners responsibility to acknowledge and try to alleviate the issue to the best of their ability. Users shall have the ability to submit any tickets

# 7 High-level system architecture and database organisation

Front-end Architecture:

Languages:

* HTML: Our application will be a Web application, so in order to display the application’s structure in the users’ web browser we will need a markup language such as HTML.
* CSS: We will add style to the frontend of our application with CSS.
* Bootstrap 5: We will use the Bootstrap 5 framework to support our front-end tasks of the project. This framework utilizes CSS and jQuery to display pre-decorated HTML elements in our website.
* JavaScript: For the information section of our web application, we will display information about Covid through the use of APIs. This information requested from these APIs will be displayed asynchronously by using JavaScript methods such as fetch.
* jQuery: This JavaScript framework comes as a dependency of Bootstrap 5, however, it will also be used to support our client side functionality that is not so neat to implement with “vanilla” JavaScript.
* APIs: Our website will include an Information Page that will display to the user data related to Covid-19 that is originally contained in the database of another domain. As mentioned previously, these APIs will be consumed on the users-end using JavaScript or Jquery. Some of the APIs that we will include are the following:
  + <https://covid19api.com/>
  + <https://rapidapi.com/vaccovidlive-vaccovidlive-default/api/vaccovid-coronavirus-vaccine-and-treatment-tracker/>
  + <https://apidocs.covidactnow.org/#data-for-individual-locations>

Browser Support:

* Supported browsers: Our web application will be compatible with the modern major browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Opera. However, in order to provide the most recent functionality with APIs we will use functions compliant with the JavaScript ES6 version, which might not be compatible with older versions of Internet Explorer.

Backend Architecture

Languages:

* PhP: We will use PhP as our back-end scripting language. This language is compatible with MySQL, which at the same time is compatible with any LAMP server. PhP language will be in charge of managing user input from HTML forms, so it will connect the input from the front-end and store, modify/update, or delete the users’ information in the database.
  + Due to PhP’s capabilities of displaying HTML elements, PhP will also be in charge of displaying the information that is retrieved from the database into the end users’ browser.
* MySQL: MySQL will be our database management system that we will be using to store and organize user information. This DBMS tool is natively compatible with so it makes it optimal to use it. Inside MySQL, we will be containing all users and the history of information to display the content on frontend work.
* jQuery: With jQuery we can use AJAX requests which communicate with the server by sending a request and receiving a response which then can be displayed into the user’s browser. This allows us to render information into the user’s browser without having to reload the webpage. This form of communication with the server combines both the functionality of backend and frontend.

Other Dependencies and Tools

Integrated development environment (IDE):

* VisualStudio Code: this will be our text editor/IDE used by the team to develop the project. This IDE has a complete integration with Github which facilitates working in groups.

Communication tools / Project Management tools / Version Control tools:

* Github: This tool will be used to maintain version control and for the team to be able to work in parallel to develop different aspects of the project in a concurrent manner.
* Gmail: We will create a Google Gmail account to send emails to users who have forgotten their password.
* WhatsApp Messenger and Discord: These two applications will be our medium of communication for our group. All live meetings will be done through Discord calls and any other type of messages will be sent through WhatsApp.
* Atlassian Jira: Jira will be our tool to plan our project’s tasks in accordance with Agile methodologies. All team members need to report their progression in this application.

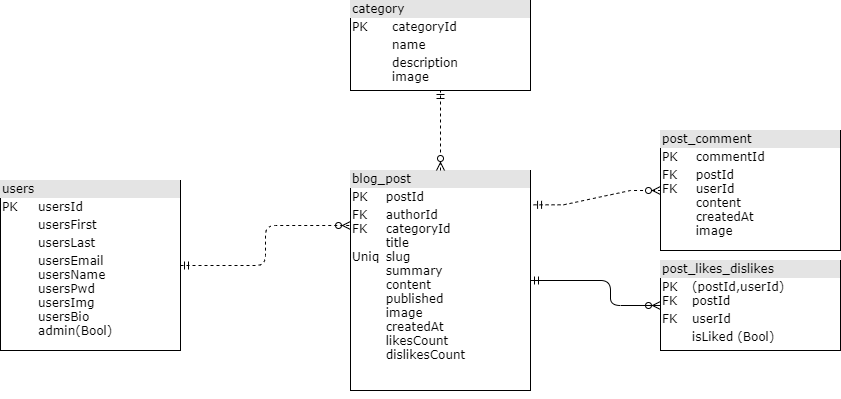
Project Host:

* NameCheap: Due to incompatibility issues with FAU’s LAMP server, our team needed to find another host to our project. We opted for NameCheap which sells domain names and a server platform at an affordable price. NameCheap’s Stellar server provides LAMP functionalities such as being compatible with PhP version 7.4 and MySQL version 5.1 which are the two dependencies for our project.

Database Organization

Database Entity-Relationship Diagram:

* The following is a diagram that describes Peace of Mind’s database and the relation between the different database tables.



* The database for Peace of Mind will be hosted in NameCheap’s Stellar server and will be updated/maintained through a dashboard that will be accessible for Peace of Mind administrators.
* Administrators will also have access to the server’s PhPMyAdmin tool which will give them more control over the information stored in the database.

Users table:

* Users’ information will be stored in a single table called users. This table will contain:
  + userId[Int(11)] (Primary Key)
  + userFirst [VARCHAR(128)] - which is the user’s first name
  + userLast [VARCHAR(128)] - which is the user’s last name
  + userEmail [VARCHAR(128)]
  + userPassword[VARCHAR(60)] - this will be stored as an encrypted hash of the original password input by the user. This hash consists of 60 characters.
  + userImg[VARCHAR(128)] - This will store the file path to the user’s profile picture
  + userBio[TEXT] - This will store a long string containing the biography of the user
  + userAdmin[TinyInt] - This is a boolean value that determines if the user is an admin or a regular member.

Blog Post Table:

* Every blog post will be stored into this table. The table will consist of the following attributes:
  + id [int(11)] (Primary Key)
  + authorId[int(11)](Foreign Key): this foreign key references a user from the users table
  + categoryId[int(11)](Foreign Key): this references the category id of the category table
  + title [VARCHAR(128)]
  + Slug [VARCHAR(128)]: this uniquely identifies a blog post and it serves as a key for searching a specific post
  + Summary [VARCHAR(128)]: this will be shown when the user looks for a post. This serves as a quick reference for what the user is looking.
  + Content [TEXT]
  + Published[TinyInt]: Boolean value that determines if the post has been published (revised by an administrator)
  + Image [VARCHAR(128)]: stores the file location of the post image
  + createdAt[timestamp]: keeps track of the date and time the post was created
  + likesCount[int(11)]: keeps track of the number of likes that the post has received from users
  + dislikesCount[int(11)]
* Category table
  + id [int(11)] (Primary Key)
  + name[varchar(75)]
  + slug[varchar(255)]
  + description[TEXT]
  + image[varchar(255)]
* Post comments table
  + id [int(11)] (Primary Key)
  + postIdid [int(11)] (Foreign Key): references the post were the comment was made
  + userIdid [int(11)] (Foreign Key): references the user that made the comment
  + Content [Text]
  + createdAt[Timestamp]
* Post likes and dislikes
  + postId [int(11)] (Primary Key)
  + userId [int(11)] (Primary Key)
    - Both postId and userId will be composite primary keys so that users can only like of dislike a post one time only
  + isLiked[Bool]: stores if the user liked or disliked the post

Media Storage

* The images used for the blog posts, blog categories, and user profiles will be saved in the local file storage of the Stellar server inside the images folder that is inside the assets folder.

Search/Filter architecture

* Search functions such as searching/filtering posts by categories is done through MySql queries functions that are provided by PhP.
* In order to access every database query function, it is necessary to be logged in into the blog.
* Search bar functionality will allow the user to look at posts by keywords which will be searched inside the post title and post content.

Non-trivial Functionality Implementation

* Like functionality
  + Every like given by an user will be stored into the likes table which will register the id of the user and the id of the post. These two attributes will be used as primary keys so that user can only give 1 like or dislike to the post. This will then be counted using the aggregate function from MySql called count.
  + This count will then be stored into the post table into its corresponding post.
  + Then the count will be displayed in the post when the user looks at that webpage.

# 8 High-level UML diagrams

The class diagram is describing the structure of our Peace of Mind blog, we are showing some of the system’s classes, some of their attributes, some of their methods, and their relationship between them.

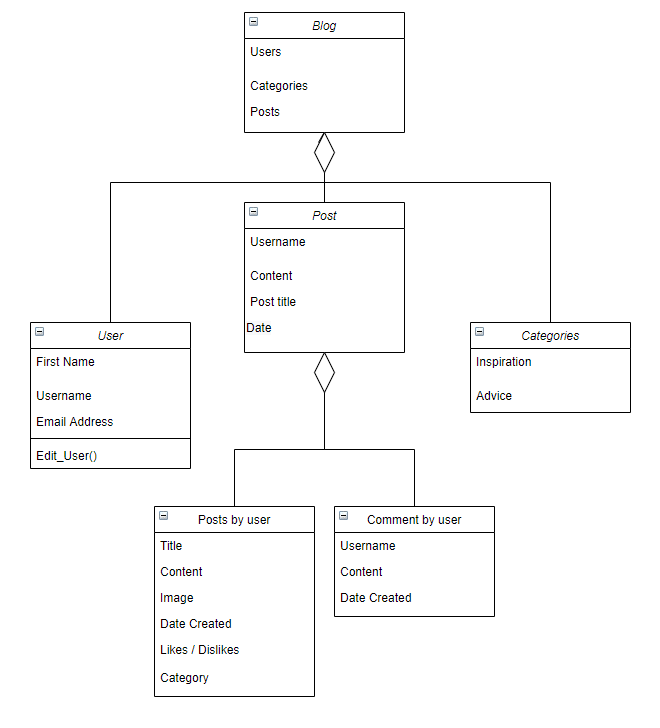


Figure x.x Class diagram for milestone 03

Our use case diagram describes the high-level functionality of our Peace of Mind blog. This diagram basically identifies the interactions between the system and its users.

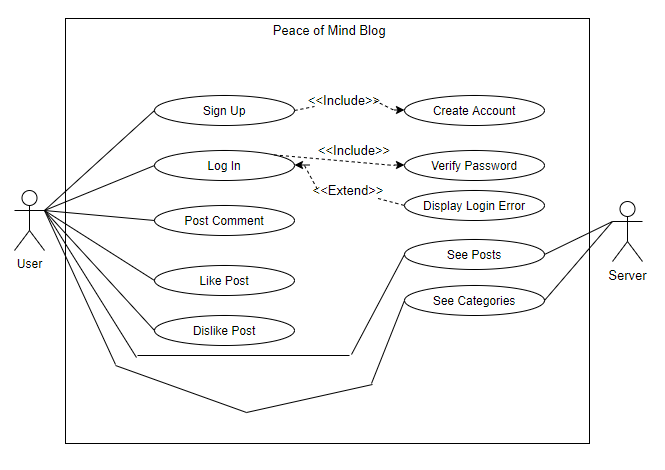


Figure x.x Class diagram for milestone 03

# 9 Identify actual key risks for your project at this time

**Skills Risks**: Deploying the project into the Stellar server poses the skill risk of not having the technical knowledge of how to work with this server.

**Schedule Risks**: Because of the issue mentioned below, there may be some difficulty in adding more content than proposed. Our main priority is hosting the product that is compatible with the server. Best-case scenario, this may take us a couple of weeks but unfavourable circumstances may force us to cut back on some features.

**Technical Risks**:

One major technical risk that has caught our attention was that there was a function in php that is incompatible with the lamp server on FAU. This function is used when the user logs into the website. To address this problem, we will look into using another function that is compatible with the lamp server. Worst case scenario, we will have to host the website on a different server.

**Teamwork Risks**: Teamwork risks include some team members not being available most of the time due to having other responsibilities.

**Legal/content Risks**: Adding images to blog posts poses a risk of copyright infringement. This will be handled by a system of post revisions where each newly created post will be unpublished until an administrator revises the post.

# 10 Team

**Team Roles**

Product Owner - Ryan Milrad

Lead Frontend - Juan Pablo Idrovo

Lead Backend - Mauricio Retana Rodriguez

Scrum master/Github master – Ryan Milrad

Frontend Support – Armaghan Ali, Ryan Bharath, Joseph Brown

Backend Support – Ryan Milrad, Ryan Bharath

# 11 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 back end 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

# 12 Tasks before submission

*Teams must collaborate in creating M1 document by having working M1 document on their team GitHub repository (similar to managing code) so all team members can access it. Added advantage of doing it this way is that it builds teamwork and communication. We recommend having a folder for project documentation on team’s GitHub where milestones and other similar files can be kept.*

Github Repository Link: <https://github.com/CEN4010-F2021-g3/cen4010-group3>

# 13 Submission

a) Course Title and term: CEN 4010 Principles of Software Engineering, Semester and Year

b) Document name: Milestone 3: More Detailed Requirements, Architecture and a Vertical Software Prototype

c) Your team name, and project name (you can use the name you chose for your team)

d) Team number (I will assign you one)

e) Names of students (team lead first) with names and e-mails

f) Documentation Date

g) History table (revisions dates) (Note: you will update this document based on instructors’ feedback so this is important)