

Features and Functionality to Test:

1. Registration
2. Login
3. Chat Room
4. adding a new email/user

SW Engineering Fall 2021
[FAU: CEN 4010]
10/25/2021

Co - Lief

*Milestone 3 More Detailed Requirements, Architecture and a Vertical Software
Prototype*

Group: 12

Oasis Husband
ohusband2017@fau.edu
(Team Lead)
Team member

Steven Fernandez
stevenfernan2020@fau.edu
(Team Lead)
Team member

Christian Jimenez
jimenezc2019@fau.edu
(Team Lead)
Team member

Avigail Martinez
avigailmarti2017@fau.edu
(Team Lead)
Team member

Eric Gonzalez
gonzalezeric2016@fau.edu
(Team Lead)
Team member

Milestone 3
10/25/2021
Revision History

Date of revision	Who revised?	Summary of revision
2021-10-19	TA	Made changes according to TA feedback
2021-10-23	Group 12	Added Diagrams

TABLE OF CONTENTS

Executive Summary.....	5
Competitive Analysis	7
Data Definition	9
Overview, Scenarios and Use Cases	12
High-Level Functional Requirements	18
List of Non-Functional Requirements	23
High-Level System Architecture and Database Organization	27
High-Level UML Diagrams	27
Identify actual key risks for your project at this time	28
Vertical Demo	29

Executive Summary

A year ago, our lives changed forever. Covid-19 rampaged through our country, changing not only the way we went about our day but how we felt about ourselves. Productivity is a tricky thing - before, we had specific spaces where we could go to a building with cold-crisp air and maybe even a separate computer where you could hang out with other people working on the same project as you. As a result, many individuals have lost a sense of organization, and many more feel less at peace with themselves. Usually, when one feels unhappy with themselves, they will turn to the conversation—finding community in their struggles.

There are many other private room chats sites, but they each share a problem - being data-rich but custom experience poor. This means that although the amount of data may be significant, its use is not personalized enough to create a unique experience for each user. This pandemic allowed us to see that people are all different; the emotional impact on some was not the same as others. There is a need for a more personal and profound experience that will take into account the user's needs while at the same time providing the best techniques and guidance to overcome this difficult period. Co-lief has been developed for just that purpose.

The Co-lief is the brainchild of a team whose ambition was to foster safe spaces where people can find help and encouragement via private chat rooms amid the uncertain times this recent global pandemic created. Co-lief fosters this positive vision by creating an inviting and responsive user experience, which makes finding a listening ear less intimidating and less office-like. The goal of Co-lief is always to connect those who have a passion for listening to those who need a friend. Co-lief will enable users to flag themselves as either a "friend (aka consultant) or a "chatter." The secret sauce of Co-lief is the breakout rooms that enable one-on-one, intimate conversations where people can find the help they need after dealing with this terrible pandemic.

Co-lief is genuinely revolutionary because it allows for a safe space with its strict moderation system. In public chats, a dedicated team of mods watches with an eagle's eye for anyone actively looking to hurt someone else. In private chats, the block button is readily available for anyone being harassed. Aside from all that, there is also a five-strike moderation system if a personal mod is not readily available. Group chats are separated into positive-minded groups, more vent-style groups, and of course, various general chat rooms. This will attract those seeking a comforting and encouraging community experience.

In addition to the features mentioned above, Co-lief reduces the stress of searching for the proper guidance and support needed depending on each case. The

way data is stored and processed allows one to have a fully customized experience with the correct specialist. This results in a private room chat that is more comfortable, easy, friendly, and stress-free.

Co-lief - A community-building website for those hoping to make their lives a little bit happier.

Competitive analysis

The analysis of competitors' websites will focus on six main features (homepage, design, navigation, search, content, usability). The competitive analysis will utilize a numerical scale (1=bad, 2=poor, 3=fair, 4=good, 5=outstanding) and consists of five websites chosen for their focus on health care support, chatting with a therapist, and use of rating/review systems.

	Co-Lief	7 Cups	FAU Mental Health	BetterHelp	Healthful Chat
Homepage:	4	3	5	5	3
Design:	4	4	3	5	3
Navigation:	5	4	5	4	2
Search:	3	3	4	3	2
Content:	5	5	5	4	3
Usability:	4	5	4	5	2
Mean:	4.16	4.00	4.33	4.33	2.50

Co-Lief (4.16)

The Co-lief homepage is simple with artistic designs made for the comfort of its users. The navigation is very simple and easy to navigate to reduce stress and frustration. The content is focused on the main objective of the webpage which is reducing and providing relief for those affected mentally and emotionally from the covid situation.

7Cups (4.00) [Free Online Therapist & Counseling | 7 Cups](#)

7Cups has important information presented on the homepage in regards to chatting with a therapist. The page looks a little unorganized because of everything that is being provided within that page. The text is clean and clear. Navigating through the website is simple and easy to find what you need. The homepage provides information about the website. The website provides emotional support by being able to chat with a therapist. 7Cups offers an app therefore, you won't ever miss a message from the

therapist. It alerts you on the phone. Many people that are going through something and need counseling, can find this website useful.

FAU Mental Health (4.33) [FAU - Mental Health Topics](#)

This website has large images that can confuse and frustrate the user easily. The amount of tabs and links inside the page is considerably large and too confusing. Quick easy phone numbers banner is visible and easy to understand and read. Topics are separate and organized in a way people can easily find them. Colors may be inadequate for a type of website like this. The response time is adequate and very effective for situations like suicide prevention and mental health monitoring.

BetterHelp (4.33) [betterhelp.com](#)

The main page is relatively simple with no eye-catching images. Navigation is relatively easy and simple with quick questions about age, gender, and ethnicity. The usability is the average based on the search and design. It does not have a “chat with a specialist now” feature on the main webpage but it has a phone app. Main web page displays pictures of current members which may discourage some users from signing up for the service. Web page displays the total number of clients which creates confidence and a sense of safety towards the service.

Helpful Chat (2.50) [Health Chat Rooms - Mental Health Chat Rooms - Health Forums - Peer Support Social Networking - HealthfulChat](#)

The homepage for Health chat rooms looks outdated and plain. It shows barely any pictures and information. The design is simple with minimum features. Out of all the websites, this looks like the worst one. Navigating through the website is easy to find what you need. It has different chat rooms you can join depending on the help you need. The website states how many users are in each chat.

Data Definition

Name	Meaning	Usage	Comment
Name	actor	Use Case scenarios,	Name the user can assign to his/her profile
Specialist	actor	Use Case scenarios	Professional in the area of mental health
User	actor	Use Case scenarios	General definition for patient
Member	actor	Use Case Scenarios	A user who is registered with the system.
Non-Member	actor	Use Case Scenarios	A user who is not registered with the system.
Room	data	Place	Represents the rooms where conversations are held
Rating	data	Activity Type	One of the services we let users add and rate
Survey	data	Activity Type	One of the services we let users add and rate
Block	data	Activity Type	Allow users to block other users

Account	data	Use Case scenarios	Store users information
VIP	data	Use Case scenarios	Allow users to get priority

Data Definition

Name	Meaning	Usage	Comment
Newset	data	Use Case scenarios	Store the newest adding activities
Rating	service	Site user service	Allows patients to rate different services
Comments	service	Site user service	Allows users to leave their opinions
Navigator	service	Site user service	Helps patients to navigate web page easier
Documents	service	Site user service	Allows users to upload documents
Search	service	Site user service	Allow user find doctors/rooms
Favorites	service	Site user service	Allows users to add their favorites doctors
Filter	service	Site user service	help user search more clearly by giving specific options

log in	service	Site user service	Allow user have ability to use comment, rating, etc
Web Site	User Interface	User interface	Front end display for user interaction
Home page	User Interface	User interface	The first page that a user go

Data Definition

Name	Meaning	Usage	Comment
Information Page	User interface	User interface	Page that is displayed when an activity is clicked.
System	platform hardware and services	Use-case scenarios	The mySQL database, all code, front end design and back end supporting services.
Co-lief	Domain Name	Use Case scenarios	A real usable name that represent all web pages and the web site
lamp.cse.fau.edu	production server	Use Case Scenarios	It's a server that store all of our data

Overview, Scenarios and Use Cases

Use Case - Navigation bar

The user comes to the page and wants to find a professional or casual friend they can private chat with in a safe space through the general forums.

The user arrives at the Home Page and can use the navigation bar on the side to interact with the other chatters, be it new or well-known chatters. The website will provide access to all previous chats that can be accessed regularly through the navigation bar as well as options for opening new private chats, flagging for help when in need of help or joining a public chat.

1. Description:

Use cases for the navigation bar may include finding old chats, opening new chats or reporting a bad chatter experience

2. Actors:

- 2.1 User
- 2.2 Other chatters
- 2.3 Mods
- 2.4 System

3. Preconditions:

- 3.1 The user has an active internet connection
- 3.2 System is available
- 3.3 Seeking help or willing to listen and/or chat

4. Primary Flow of Events:

- 1. The user arrives on the web page.
- 2. The user sees navigation menu on the left
- 3. The web page displays relevant use cases.
- 4. User selects Location/Activity on the navigation menu.
- 5. The user is directed to their desired activity
- 6. Navigation remains ready for use

5. Alternate Flows

5.1 User misclicks activity on the navigation menu

- 1. If the user clicks the wrong activity on the navigation bar they will be brought to an unintended area and will need to use the navigation bar again

5.2 User misclicks and early terminates an activity

2. If a user prematurely or accidentally ends one of their chats they can use the navigation bars to bring up the terminated chat

Use Case - Flag for help

The User wants to add a public notification to the general community; they can do so from the homepage, or during their private chats and or public chats. Flags can be used to request a “chatter” or mod’s help and can be private(specifying who should receive the flag) or public flags which are broadcast to the community.

1. Description:

The flag system is designed to enable quick and discrete notifications of a need someone within the community may be experiencing. The flag system also potentially allows users to understand the types of chatting an individual is desiring. .

2. Actors:

- 2.1 User
- 2.2 chatters
- 2.3 mods
- 2.4 System

3. Preconditions:

- 3.1 User has an active Internet Connection
- 3.2 System is available
- 3.3 The user has an active account
- 3.4 The user is logged into the system

4. Primary Flow of Events:

- 4.1 User has a desire to communicate discreetly or publicly the level or the type of chat they desire
- 4.2 Chat type is chosen
- 4.3 Flag is issued
- 4.4 problem might be experienced
- 4.5 flag status may be changed or logged to collect issues experienced

5. Alternative Flows:

- 5.1 Wrong flag might be chosen and chat posted
 - 5.1.1 Update chat type and the flag type will be updated
- 5.2 The user does not have an account (from Alternative Flow 5.1.1.1)
 - 5.2.1 System requests the user create an account
 - 5.2.1.1 If the user does not want to create an account, End
Alternate Flow 5.2, return to Primary Flow 4.6
 - 5.2.2 User clicks CREATE ACCOUNT button
 - 5.2.3 System displays Account Creation Form
 - 5.2.4 The user completes a form, and presses submit
 - 5.2.5 The system creates a new member account in the member database
 - 5.2.6 End Alternative Flow - return to Primary Flow, step 4.3

Use Case - Open a public chat

The user comes to the page and wants to open a public chat or contribute to someone else's public chat; they can do so by utilizing the navigation bar. Public chat's are public spaces where people can find like-minded individuals, share in a communal setting and make new friends.

1. Description:

When a chatter logs into their account they will instantly have the ability to create their own public chat or join someone else's. Public chats are member only chats which will allow for them to be safe spaces that moderators can monitor. If someone doesn't need anonymity but would like to find help or express themselves or lend an open ear they are free to do so in this use case.

2. Actors:

- 2.1 User
- 2.2 chatters
- 2.3 mods
- 2.4 System

3. Preconditions:

- 3.1 The user has an active internet connection
- 3.2 System is available

- 3.3 The user has an active account
- 3.4 User is logged into system
- 3.5 User has rated the Location/Activity

4. Primary Flow of Events:

1. User arrives at Home Page.
2. Users log into their account
3. The user decides if they want to create chat or join chat
4. User decides if they want a public or private chat to join or create
5. Public chat is chosen
6. Chat is created and flagged as a public chat
7. Moderators are invited inside the chat as well as other chatters

5. Alternate Flows:

5.1 User Isn't Logged In

If user does not perform step 2 and they reach step 4

1. Web page requests user to either log in or create an account.
2. User logs in or user creates an account
3. Return to step 4

5.2 User Tries to join a private chat instead of a public chat

1. Web page tells the user they must request and be given permission before entering a private chat.

Use Case - Open a private chat

_____ The user comes to the page and wants to open a private chat or contribute to someone else's private chat; they can do so by utilizing the navigation bar or chatter spaces feature. Private chat's are spaces where people can find like-minded individuals, share in a communal setting and make new friends but with more anonymity and exclusivity.

1. Description:

When a chatter logs into their account they will instantly have the ability to create their own private chat or join someone else's. Private chats are member only chats which will allow for them to be safe spaces that moderators can monitor. If someone wants anonymity or to talk exclusively while finding help or expressing themselves or lending an open ear they are free to do so in these use cases.

2. Actors:

- 2.1 User
- 2.2 chatters
- 2.3 mods
- 2.4 System

3. **Preconditions:**

- 3.1 User has an active internet connection
- 3.2 System is available
- 3.3 User has an active account
- 3.4 User is logged into system

4. **Primary flow of events:**

- 4.1 User arrives at Home Page.
- 4.2 Users log into their account
- 4.3 The user decides if they want to create chat or join chat
- 4.4 User decides if they want a public or private chat to join or create
- 4.5 Private chat is chosen
- 4.6 Chat is created and flagged as a private chat
- 4.7 Moderators are invited inside the chat as well as other chatters

5. **Alternative Flows:**

5.1 **User Isn't Logged In**

If user does not perform step 2 and they reach step 4

- 4. Web page requests user to either log in or create an account.
- 5. User logs in or user creates an account
- 6. Return to step 4

5.2 **User Tries to join a private chat instead of a public chat**

- 5.3 1. Web page tells the user they must request and be given permission before entering a private chat. .

Use Case - Access prior chats

The user comes to the page and wants to quickly access a prior chat or return to a chat that was closed accidentally.

1. **Description:**

Users can upon logging in or at any other time while logged in access prior chats by making use of the navigation bar: use cases for this could be when a user had a great conversation and wants to reflect on the details shared, or a user accidentally exits out of a chat but wants to rejoin or re-invite another user

2. Actors:

- 2.1 User
- 2.2 chatter
- 2.3 mods
- 2.4 System

3. Preconditions:

- 3.1 User has an active internet connection
- 3.2 System is available
- 3.3 User has an active account
- 3.4 User is logged into system

4. Primary flow of events:

- 1. User arrives on the web page.
- 2. User accesses the navigation bar
- 3. User can view all prior chats
- 4. User can re invite the chatter directly from the reopened chat
- 5. User can request a public or private chat with the old chatter

5. Alternative Flows:

5.1 User isn't logged in

In step 4, the user attempts to rate and isn't logged in.

- 1. Web page displays a prompt asking the user to log in or create an account.
- 2. User logs in or creates a new account.
- 3. Return to step 5.

5.2 User didn't want to look at older chats but open a new one

- 1. **The user can use the navigation tool to go home and begin a new chat or join a different chat**

Non-Member expectation

1. Creating Account

- **1.1** The system shall allow the user to create an account by storing

UserID, Password, Date of Birth, and answer to security questions.

The system will not allow the User to Create an account if the UserID chosen by the User already exists in the System's Database. The system will prevent the user from creating an account if the User's chosen password does not match the re-enter password field. Also The system will also prevent the user from creating an account if the date of birth does not conclude that the user is 13 years or older. System shall prevent the creation of the user's account if the following fields are not filled. Fields that have to be filled are UserID, Password, Re-enter Password, Security Answer Security Question, and Date of Birth.

- **1.2 Stimulus/Response Sequence**

- i. User enters a UserID
- ii. User enters a Password
- iii. User re-enters Password for confirmation
- iv. User shall enter their date of birth
- v. User shall provide to an answer to given Security Question
- vi. System shall check if UserID is available
- vii. System shall validate Password
- viii. System shall validate Age
- ix. System shall store userID, date of birth, and answer to selected security question
- x. System shall confirm that the account was created to the User
- xi. System will have a button to redirect the user back to the home page at will

- **1.3 Function requirement label**

- i. REQ 1.1 Creating Account

2. About Us

○ **2.1** System provides information to the user about the use that

can benefit the user with information that the user might find useful and links that will direct the user to specific topics. The system shall not allow the user to erase any information about the system and how the system functions for the user.

○ **2.2 Stimulus/Responsive Sequence**

- i. User uses the navigation bar and clicks “About us”
- ii. User is redirected to About us page
- iii. User sees links about topics at the top of the page
- iv. User clicks links so System will scroll to proposed topics
- v. System has a button that allows the User to return back to Home page

○ **2.3 Function requirement label**

- i. REQ 2.1 About Us

3. Questions?

○ **3.1** System provides a link where the user can be provided the opportunity to ask questions that may be added to a future FAQ section

can benefit the user with the ability to pool possible questions about using the site with other users. System shall prevent the asking of the user’s question if the number of words are less than ten.

○ **3.2 Stimulus/Responsive Sequence**

- i. User uses the navigation bar and clicks “Questions?”
- ii. User is redirected to Questions? page
- iii. User sees an ask box prompting the User to ask a question
- iv. User types a question into the ask box
- v. System counts the amount of words and checks it with proposed minimum
- vi. If the number of words is less than ten, the system prompts the user to write more.

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

- **3.3 Function requirement label**

- i. REQ 3.1 Questions

Members expectations

4. Filling out Customization Quiz

- **4.1** Until the User has part of creating the account, the system shall heavily guide the user to create a customized experience by answering five yes or no questions

Do you like listening to people? Do you like talking to people? Would you prefer a darker website? Do mountains relax you? Do flowers relax you?

The system shall not allow the User to Finish Quiz if any answers have been left blank instead of choosing yes or no.

- **4.2 Stimulus/Response Sequence**

- i. User clicks either yes or no for each question
 - ii. User clicks Finish Quiz
 - iii. System checks to see if all questions have been answered
 - iv. System shall store answers for each question with User username
 - v. System shall edit background and username label to match with answers to question
 - vi. User shall enter their date of birth
 - xii. User shall provide to an answer to given Security Question Or provide their phone number
 - xiii. System shall check if UserID is available
 - ix. System shall validate Password
 - x. System shall store user Name, date of birth, and answer to selected security question/phone number
 - xi. System shall confirm that the account was created to the User
 - xii. System will have a button to redirect the user back to the home page at will

- **4.3 Function requirement label**

- i. REQ 4.1 Quiz Answers

5. Profile

- **5.1** Members will be able to edit their profiles with their name, birth date, gender, and summary about themselves. They will be able to change the information at any time they would like.
- **5.2 Steps to navigate to profile**
 - i. Members will be able to click the drop down menu that states “my profile”
 - ii. Members will be able to click a section named “edit profile”
 - iii. Members will be able to input or change any information like their name, date of birth, gender, and summary about themselves.
 - iv. Member should click “save” after finishing with their profile
 - v. The System shall save and store all the updates made to the members profile
 - vi. The system will show the update on the screen
- **5.3 Function requirement label**
 - i. REQ 5.1 Edit Profile

6. Rate comments posted by users

- **6.1** Users will navigate to a specific Location/Activity they will read the facts and read comments provided by users. Some users will find it useful and up-vote the comment, others the users will find it useful and downvote the comment. System shall keep track of these upvotes and downvotes.

The system prevents the user from deleting upvotes or downvotes. Once an Upvote or Downvote has been past it stays. The system shall prevent the User from giving multiple Upvotes and Downvotes on a specific location.

The system will allow for flagging of chatters and the opening of issues to the mods for arbitrarily deciding systemic and detrimental behaviors that might be aimed towards the community

- **6.2 Stimulus/Responsive Sequence**
 - i. User will use the browsing features to view list of Locations/Activities
 - ii. User will click on desired Location/Activity among the list of suggestions

- iii. Users will read facts and then scroll to Users comments
- iv. User will upvote comments that they find useful and downvote comments they find unuseful
- v. System shall record upvote or downvote on specific comment
- vi. System shall prevent the user from upvoting and downvoting again on same comment
- vii. System shall show a confirmation that the user has voted on the comment.
- viii. System will have a button to redirect the user back to the home page at will

- **6.3 Function requirement label**

- i. REQ 6.1 Rate Comments

7. Contact Customer Service

- **7.1** Users will be allowed to contact the developers via email. user shall type in their name, email, and their query. System shall store these fields and submit it once the user clicks the send button. System shall prevent the User from contacting developers directly. Users have to submit a ticket and the ticket shall be redirected to the correct personnel.
- **7.2 Stimulus/Responsive Sequence**
 - i. Users shall navigate to Contact Us among the navigation bar
 - ii. Users shall fill in Name, Email, and type up their Query
 - iii. Users shall then click the “Send” button underneath the Query box
 - iv. System shall store the information and submit
 - v. System shall state that the information was sent and thank the user
 - vi. System shall a button to redirect the user to Home page

- **7.3 Function requirement label**

- i. REQ 7.1 Customer Service

List of Non-Functional Requirements

Performance Requirements:

1. Responsiveness: The system will be quick to respond, operating on different monitor sizes, ranging from 10" portable screens to 24" desktop monitors. It will also be quick to respond with a wide variety of resolutions, from 1024 x 600 through 1900 x 1200.
2. Cycle Time: The cycle time at expected performance will be 1.0. With this in mind, the system will operate between 1.0 - 1.2 with a load of 5-10 concurrent users or a slight amount of lag. The system will operate with a 1.21 - 1.30 with a load of 11-25 concurrent users or a moderate amount of lag. The system will operate with a 1.31-1.50 with a load of 26-45 concurrent users or a heavy amount of lag. Finally the system will operate with a 1.51-1.70 with a load of 46-50 concurrent users or a very heavy amount of lag. Any number of concurrent users over 50 will cause the system's performance to halt briefly until a user finishes.
3. Speed Per Transaction: The speed per transaction will be between 20-100 milliseconds, depending on the cycle time. The system will process 10-50 transactions per second.
4. Test Requirements: the test requirements for performance will include an expected load test as well as testing on all of the functional specs listed and their speed per transaction.
5. Reliability: The mean time between failures is that it must have 1 hour or less of downtime in a total of 3 months. This downtime can be used to perform maintenance and update information. The system should be operational 100% of the time for the first 99.8% of the calendar for the first year of its operation.
6. Minimum Bug Counts:
 - No more than 5 bugs in the system during integration and testing.
 - No more than 3 bugs can remain in the system after delivery.
7. Execution Speed: Execution speed of the initial home page on a high-speed internet connection should load within 100-200 milliseconds, depending on the current cycle time.
8. Storage Utilization: Storage utilization should be within 75 - 90% of the available storage provided at the time as to not get too close to using all storage and having a technical issue if more storage is needed for an emergency situation.
9. Robustness: The time needed to restart after a failure should be under an hour. The percentage of events that cause a failure will be under 0.1%. The probability of the data being corrupted on failure must be below 0.8%.

Ease of Use:

1. Training Time: Training time should be minimal if any at all as our site will be very user-friendly and should be able to be used by anyone who visits it.

Interoperability Requirements:

1. Browser Compatibility: The system will be a web-based web app that operates on at least two of the all of the major browsers, including Google Chrome, Mozilla Firefox, Safari, Opera, and Internet Explorer. It will have functionality in it that will provide alternatives if the browser does not have JavaScript installed on it.

3. Computer and OS Compatibility: The system will operate on various types of operating systems, including Windows, OS X and Linux. It will also operate on any type of computer which can run a browser which is supported.

Expected Load:

1. We will attempt to make allowances for up to 50 users at the same time. Load testing will be done to record performance times during periods of high traffic, both in a continuous and in a spiked pattern.

Security Requirements:

1. Login/Password System: Our system will have a login/password system to maintain the preferences, ratings and reviews of our site's visitors and locals. This implementation will also require a password confirmation upon creation. We will also ask the user for a security question that we will store along with the answer and if the user forgets their password, they will be able to retrieve it by providing the answer.

2. Encryption: The website will not be encrypted as no purchases or exchange of valuable information, such as credit card numbers, will occur.

3. Access Control: The ability to edit the front end code and back end code and databases will be provided to everyone on the development team. The users and visitors will have limited access to using the system based on the user interface.

4. Spam Protection: The site will ask the user to enter a string of characters shown on a picture to create an account, thereby preventing bots from spamming the site and creating bogus accounts.

5. Resource Utilization: Resources such as the mySQL database on the sfsuswe.com server will be accessed through the PHP code using the usernames and passwords therein. All access to the sfsuswe.com servers and their resources will be obtained with the usernames and passwords given. The system will utilize bootstrap

and jQuery as frameworks and will document proper licenses and/or qualifications of each.

Portability Requirements:

1. Platform Compatibility: The system may possibly develop for mobile and tablet, but that is based purely on whether it will have the ability to develop in the given time frame.

It's possible that in a further version, mobile and tablet compatibility will be implemented.

2. Percentage of Target-Dependent Statements: In this version of the system, 0% of statements will be target dependent. In a future version for use on mobile and tablet, 30% of statements will be target dependent.

Supportability Requirements:

1. Coding Standards: Our system will be coded in a range of 75-80% of coding standards for HTML 5 and CSS3. The code will be produced, reviewed, tested, reviewed again and finalized by yet another developer for efficiency.

2. Naming Conventions: HTML classes and id tags will be coded in lowercase except in the situation where there is more than one word in a name, and then it shall use camelcase convention. SQL tables and data names will be agreed on by the team and always use first letter capitalization and lowercase remaining letters.

Storage Requirements:

1. The storage for our system will consist of the sfsuswe.com server holding our MySQL databases within an unknown capacity as well holding our files for the actual site.

Survivability:

1. Our system will be stored on a sfsuswe.com server and the major files and documentation will be backed up on Google Drive to prevent loss in case of a fire or other physical destruction of the servers.

Availability Requirements:

1. Accessible Times: Our system should be available for use 24 hours a day, 7 days a week. It will be up and running as long as the sfsuswe.com server is available.

2. Downtime Impact: The down time will be minimal but when it is necessary, a splash page will be used to identify that the system is in maintenance. Downtime

impact is expected to be minimal and the scheduled downtime will be announced ahead of time.

3. Support: There will be support availability by email that will filter to several assigned developers. They will be responsive within 24 hours.

Fault Tolerance:

1. Exception Handling: There will be exception handling provided in all situations where an exception could occur. This will provide the user with an explanation as to why an exception occurred and give them a chance to either input the correct answer or they will be taken back to the home page.

2. Self-checking software will not be implemented as this system isn't critical enough to warrant it.

Planned advantages:

Co-lief is all about improving an individual's mental health throughout these tough times. As individuals we have been dealing with covid for the past two years. Many people need someone to talk to and grow mentally. Our plan with Co-lief is to be the best mental health chat website. Joining our website is as simple as signing up with an account. Once you're signed up, you can start chatting with others about covid and how it affected you.

High-level system architecture and Database Management

Software/Tools:

- MyPHP will be used as the primary database for the software.
- npm: package management system if needed
- Brackets/Visual Studio Code: Code editor used for HTML, CSS, JavaScript, etc.
- GitHub: Version control software used for source code management.

Programming Languages:

- HTML: Programming language will be used for website appearance and design.
- CSS: Programming language will be used for overall presentation of the website.
- Javascript: Scripting language will be used for programming site functionality.
- mySQL: Programming language will be used for the database.

Frameworks:

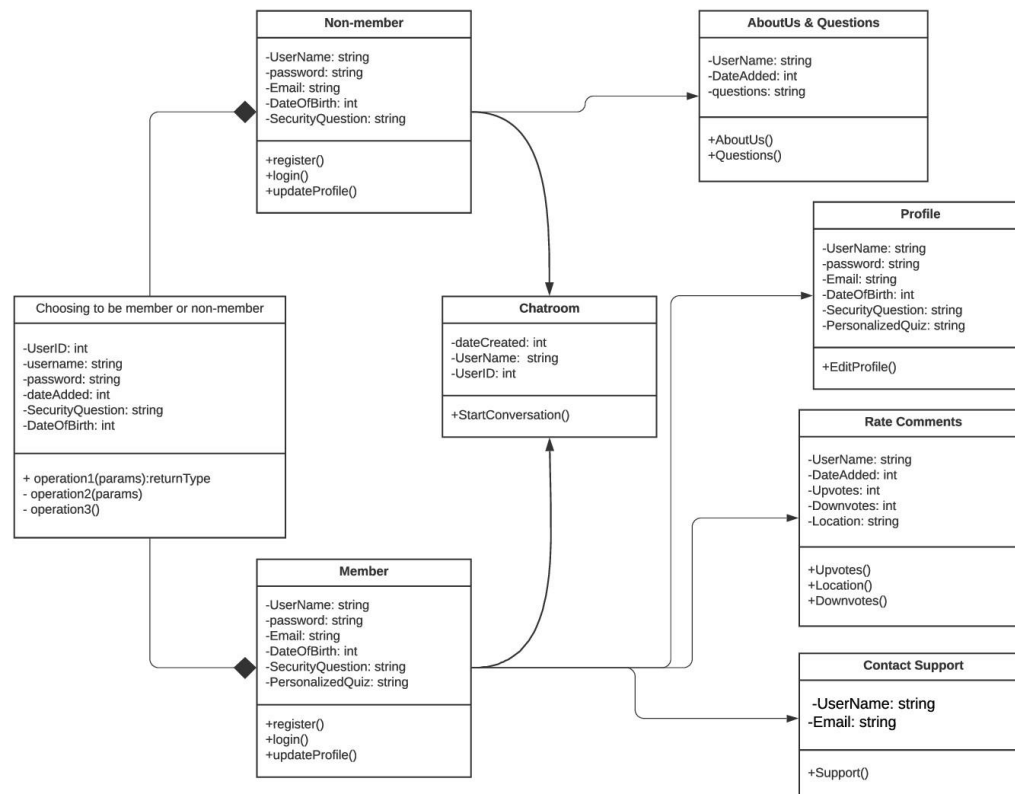
- Bootstrap - Responsive CSS framework used for website presentation.

DB organization:

- Database will consist of a collection for each user and collections for each of the multiple topics.
- Within a user collection will be documents of all said user's posts. Additionally, this collection will store the profile information for the user.
- Within the topic collection will be documents of all the posts within the topic, along with a document on the number of posts in the topic. Within the post documents will be a field of the post ID and a field of the post title and post text.
- Searching will be conducted by looking for matching documents within a collection based on a given user input.
- User profile pictures will be stored on the database using MyPHP in a given user's collection.

High-level System Architecture (UML)

UML diagram



Identify actual key risks for your project at this time

1. Skill risks – The back-end developers have prior experience with PHP. We will have to take time however to research on how to integrate myPHP UTL8 into the Lamp server. The front-end developers have taken courses in web design and so already have the required skills for developing the front-end of the website.
2. Schedule risks – Since the start of the development cycle we had to strip down planned features to make sure we can meet the scheduled release. As of now we are on track to release on time.
3. Technical risks – We can only make an approximate estimation for how much storage will be required to store every session. Throughout the development cycle we will have to monitor the storage requirement of the website as we open it up for beta testing. Another risk we must monitor is bandwidth. If a lot of users are connected to the website at once it will drive up the bandwidth of the website and could cause it to crash.

4. Teamwork risks - Team members naturally have different schedules/agendas, therefore not everyone in the team will be available to participate in meetings and discussions 100% of the time. With that said, communication can be hindered because participation is a key factor when it comes to teamwork. All team members are encouraged to free a specific time of the week in order to update one another about the progress of development. Additionally, due to the pandemic, team members are not able to meet face to face so they are forced to participate in voice chats or zoom meetings only.

5. Legal/content risks - Some of the legal risks involved in our website involve, Data Protection Legislation, Licensing for API's, Defamation, and Copyright. We ask for personal information on our website such as first name and last name and we plan to have some other social media links as well as telephone numbers so that could be a civil claim. On our website we allow authorized users to edit the information on our website so that can cause defamation. The last legal risk we could run into is copyright, we are considering allowing authorized users to upload their own picture but using google to log in they could have a copyrighted photo already.

Vertical Demo

YouTube Link: <https://www.youtube.com/watch?v=M7Wh4-jJ6gY>