

# CEN 4010 Principles of Software Engineering

Fall 2021

## Milestone 1 Project Proposal

### Description:

A communication website for students affected by social distancing

### PROJECT NAME

IQLounge

### TEAM NAME

Group 15

### TEAM NUMBER

15

### Student Names

Bryce Kurek: bkurek2018@fau.edu

Kyle Frudakis: kfrudakis2018@fau.edu

Carlos Castellanos: castellanosc2018@fau.edu

Dunyagozel Durdyeva: ddurdyeva2020@fau.edu

James Goedmakers: jgoedmakers2019@fau.edu

### Date

09/27/2021

### History Table

**TABLE OF CONTENTS**

<b>Executive Summary</b>	<b>1</b>
<b>Use Cases</b>	<b>4-8</b>
<b>Data Definitions</b>	<b>9-10</b>
<b>Initial List of Functional Specifications</b>	<b>11-12</b>
<b>Member Expectations</b>	<b>13-16</b>
<b>High Level System Architecture</b>	<b>17</b>
<b>Competitive Analysis</b>	<b>18-19</b>

## Executive Summary

Due to the Covid-19 pandemic, Society has been forced into a situation where social distancing is mandatory. While this was effective in reducing the spread of the virus, it has resulted in a loss of social interaction between individuals. Humans by nature are social creatures who require a community of others to thrive. Feeling connected socially is important for the overall health and wellbeing of an individual, especially during this time of fear and uncertainty. Luckily, during this technological age, we are able to combat the negatives of social distancing by connecting virtually.

During the crossover of being in this technological era and quarantine, many companies have provided online tools and platforms in order to aid in the need for social interaction. Families, jobs, and especially schools all used these new upcoming chat platforms to aid in the need for life to continue moving along, in a safe way. One of the main populations of people that used these online platforms were students. Students around the world transitioned to fully online classes, using video chats for class sessions. Although it got the job done, it disregarded the need for students to connect and socialize with one another. Students went online for classes, logged off and that was all. The idea was monotonous and did not encourage student interaction that was fruitful nor engaging. Allowing students all around the world to form relationships with fellow peers at their own school or even in different schools lets their social skills and friendships flourish. This is where IQ Lounge comes into play.

The IQ Lounge website exhibits an environment where students can come together to socialize and relax. The name IQ Lounge is a representation of the "Intelligence Quotient" test which measures intellectual potential and "lounge" portrays a relaxed state for students. The website is intended to provide students with tools and accessibility in order to encourage active interaction in an entertaining fashion.

IQ Lounge provides students with the opportunity to converse in one main general chat with other students around the world. Students will be able to login and have the ability to create smaller private chat rooms revolving around particular topics or social groups. Users can also use the search feature in order to find specific private chat rooms based on their own personal interests. Along with chats, users will also have access to many resourceful links recommended by students that could promote either student success, ways to improve mental health during quarantine and other means of staying connected during these hard times.

IQ Lounge has various features which are designed with the intention of giving students a means of social interaction. The multitude of features including a main chat and abundant amount of private chat rooms for all interests allows students to establish connections while staying safe and practicing social distancing. IQ Lounge - student connections in the safety of your own home.

## Use Cases

### Use Case: Create Account

User comes to the landing page and wants to create an account. User selects 'log in', then 'create account', user enters username and password in fields. System checks fields before creating an account.

1. Description
  - 1.1. Use case describes how the user will interact with the system to create an account
2. Actors
  - 2.1. User
  - 2.2. System
3. Preconditions
  - 3.1. User has an active internet connection
  - 3.2. system is available
  - 3.3. User does not have an active account
4. Primary Flow of Events
  - 4.1. User arrives on landing page
  - 4.2. User selects log in
  - 4.3. System displays log in form
  - 4.4. User selects create account
  - 4.5. System displays create account form
  - 4.6. User enters information
  - 4.7. System checks fields
  - 4.8. System displays account created
  - 4.9. Terminate Use Case: Create Account
5. Alternative Flows
  - 5.1. System field check failed (non-unique email, illegal characters, empty fields)
    - 5.1.1. notifies user of invalid input
    - 5.1.2. repeat from step 4.3

### Use Case: View Room

User arrives at the landing page and wants to view rooms created by users. Selects 'view Room'. The system displays posted pages and may search for specific Room. User selects a Room. System then displays room information.

1. Description
  - 1.1. Use case describes how User will navigate to select a viewable post
2. Actors
  - 2.1. User
  - 2.2. System
3. Preconditions
  - 3.1. User has an active internet connection
  - 3.2. System is available
  - 3.3. room exists
    - 3.3.1. user has access to room
4. Primary Flow of Events
  - 4.1. User arrives on landing page
  - 4.2. User selects view rooms
    - 4.2.1. user may enter search fields
  - 4.3. System retrieves room data
  - 4.4. System displays rooms based of search criteria
  - 4.5. User selects the desired room
  - 4.6. System displays full room
  - 4.7. Terminate Use Case: View Room
5. Alternative Flows
  - 5.1. if there is an error displaying the full page
    - 5.1.1. System notifies user that there was an error retrieving page information
    - 5.1.2. return to step 4.4

## Use Case: Create Room

User arrives at the landing page. User logs into account. User wishes to create a room, selects create room. enters content, then the system adds room to the system.

1. Description
  - 1.1. Use case describes the process of a user interacting with the system to create a custom page added to the system
2. Actors
  - 2.1. User
  - 2.2. System
3. Preconditions
  - 3.1. User has an active internet connection
  - 3.2. User is logged into the system
  - 3.3. System is available
4. Primary Flow of Events
  - 4.1. User arrives on landing page
  - 4.2. User selects create room
  - 4.3. System Displays post chat form
  - 4.4. user enters data in all content fields
    - 4.4.1. adds user names from other users for a private room
  - 4.5. System creates chat
  - 4.6. Terminate Use Case: Create Room
5. Alternative Flows
  - 5.1. Invalid fields: no content entered, invalid characters
    - 5.1.1. notify user of error
    - 5.1.2. return to step 4.4
  - 5.2. User is not logged in
    - 5.2.1. System prompts user to log in
    - 5.2.2. System displays log in form
      - 5.2.2.1. User does not have account
      - 5.2.2.2. User carries out Use case: Create Account
    - 5.2.3. user enters login information
    - 5.2.4. return to step 4.4
  - 5.3. if user does not add other users to room
    - 5.3.1. system sets post as public
    - 5.3.2. continue from step 4.5

### Use Case: Post

User arrives on the landing page, user logs into account, navigates to post, selects to comment on post.  
User enters content in all fields and returns, the system adds comment to room.

1. Description
  - 1.1. Use case describes how user can comment on a selected post
2. Actors
  - 2.1. User
  - 2.2. System
3. Preconditions
  - 3.1. User has an active internet connection
  - 3.2. User is logged in to account
  - 3.3. System is active
4. Primary Flow of Events
  - 4.1. User arrives at landing page
  - 4.2. User carries out Use Case: View Room
  - 4.3. User selects post
  - 4.4. System displays post form
  - 4.5. User fills in post fields
  - 4.6. System adds post to room
  - 4.7. Terminate Use Case: Post
5. Alternative Flows
  - 5.1. Invalid fields: no content entered, invalid characters
    - 5.1.1. notify user of error
    - 5.1.2. return to step 4.4
  - 5.2. User is not logged in
    - 5.2.1. System prompts user to log in
    - 5.2.2. System displays log in form
      - 5.2.2.1. User does not have account
      - 5.2.2.2. User carries out Use case: Create Account
    - 5.2.3. user enters login information
    - 5.2.4. return to step 4.4

## Use Case: Edit Account

user arrives on landing page and logs in to their account, and wishes to edit their account, users selects accounts, then edit account, then edits their information

1. Description
  - 1.1. Use case describes how the user will edit their account information
2. Actors
  - 2.1. User
  - 2.2. System
3. Preconditions
  - 3.1. User has an active internet connection
  - 3.2. System is available
  - 3.3. User is logged into the system
4. Primary Flow of Events
  - 4.1. User arrives on landing page
  - 4.2. User logs into the system
  - 4.3. User selects account
  - 4.4. System displays account information
  - 4.5. User selects edit account
  - 4.6. System displays edit account form
  - 4.7. User enters data into fields
  - 4.8. System confirms changes
  - 4.9. Terminate Use Case: Edit Account
5. Alternative Flows
  - 5.1. User is not logged in
    - 5.1.1. System prompts user to log in
    - 5.1.2. System displays log in form
      - 5.1.2.1. User does not have account
      - 5.1.2.2. User carries out Use case: Create Account
    - 5.1.3. user enters login information
    - 5.1.4. return to step 4.3
  - 5.2. Invalid fields: no content entered, invalid characters
    - 5.2.1. notify user of error
    - 5.2.2. return to step 4.6



## Data definitions

Name	Meaning	Usage	Comment
User	actor	use case scenarios	someone that interacts with the system
Member	actor	use case scenarios	a user that has an account registered with the system
Non-Member	actor	use case scenarios	a user that does not have an account registered with the system
Account	data	use case scenarios	stores member information
Room	data	use case scenarios	stores room information between users
Topic	data	use case scenarios	the title of a post
Post	service	site user service	allows member user to add to room data
Log In	service	site user service	allows user to use comment and post functions
View room	service	site user service	allows a user to view rooms the user has access to
Search	service	site user service	allows user to narrow down selection pool for selecting a room
Web Site	user interface	user interface	front end display for user interaction
Page	user interface	user interface	a web page that displays content created by developers

			or users
Landing Page	user interface	user interface	page that any user first arrives at
Room page	user interface	user interface	displays room page when a room is selected by a user, room pages are created by users
Form	user interface	user interface	prompts user to input data to be added to the system
Field	user interface	user interface	facilitates specific data transfer between user and system located within a form. (username/password/content)
lamp.cse.fau.edu	host server	use case scenarios	the server which will house the system
System	platform hardware and service software	use case scenarios	SQL database, html, javascript, css, python for back and front end services

## Initial List of functional Specifications

## **Non-Member Expectations**

### **1. Creating Account**

- 1.1.** The user will be able to create an account with info such as an email, username, password, first name, and last name. If the user already has an account linked to their email, they will not be allowed to make another account with that same email. The fields states above are required for making an account

#### **1.2. Stimulus/Response Sequence**

- 1.2.1. user enters email
- 1.2.2. user enters username
- 1.2.3. user enters a password
- 1.2.4. user re-enters password to make sure it is correct
- 1.2.5. user enters first name and last name
- 1.2.6. system will check if email is unique in database
- 1.2.7. system will check if username is unique in database

#### **1.3. Function requirement label**

- 1.3.1. REQ 1.1 Creating account

### **2. Browse Rooms**

- 2.1.** Users and non users will be able to browse the existing rooms that they can read about but they will not be able to contribute to a room if they do not have a user account

#### **2.2. Stimulus/Response Sequence - Browsing mechanics**

- 2.2.1. User can scroll through main page and look at the various rooms to enter
- 2.2.2. User can select a room to see the post made in that room

#### **2.3. Stimulus/Response Sequence - Searching**

- 2.3.1. Users can use the search bar to search for a specific room. User cannot create a room without an account

#### **2.4. Function requirement label**

- 2.4.1. REQ 2.1 Browse Rooms
- 2.4.2. REQ 2.2 Browsing Mechanics
- 2.4.3. REQ 2.3 Searching

### **3. View a specific Room**

- 3.1.** To view a room, the user just has to click on a box containing the room that will lead to a separate page involving all the information about the room. To contribute to a room, the user has to have an account first

#### **3.2. Stimulus/Response Sequence - selecting a room**

- 3.2.1. To select a room, the user has to click on the “box” that contains a brief description of the room

#### **3.3. Function requirement label**

- 3.3.1. REQ 3.1 View a specific room
- 3.3.2. REQ 3.2 Selecting a room

### **4. About Us**

- 4.1.** The About Us section will be a separate page that can be accessible at the bottom of every page. This page will contain information about the developers of the project. There will also be contact information for the website if help is needed. Finally, there

will be a short description about what the website is about and how it came into existence.

**4.2. Stimulus/Response Sequence - Developer Info**

4.2.1. A section of the about us page will have the names of all the contributors as well as how they contributed

4.2.2. about us page will have contact information at the bottom of the page

**4.3. Stimulus/Response Sequence - Description**

4.3.1. A section of the about us page will be a description of what the project is about and how everyone contributed

**4.4. Function requirement label**

4.4.1. REQ 4.1 About Us

4.4.2. REQ 4.2 Developer Info

4.4.3. REQ 4.3 Description

## **Member Expectations**

### **1. Edit Account**

**1.1.** To edit an account, the user has to select the “edit account” option after logging in. The user is then given the option to edit their first name, last name, or password.

#### **1.2. Stimulus/Response Sequence**

- 1.2.1. User selects “edit account”
- 1.2.2. User selects “edit first name”/ “edit last name”/ “edit password”
- 1.2.3. User types new first name/last name/password”
- 1.2.4. User types new first name/last name/password a second time to confirm.
- 1.2.5. User selects “confirm change”

#### **1.3. Functional Requirement Label**

- 1.3.1. REQ 1.1 Edit Account

### **2. Create New Post**

**2.1.** To post new material, the user has to select the “create new post” option after logging in. The user is then prompted with a text box to type their post and/or upload an image, as well as an option to select the desired room to make their post in. The user must then select “confirm post” to post their content.

#### **2.2. Stimulus/Response Sequence**

- 2.2.1. User selects “create new post”
- 2.2.2. User selects the desired room to post in, from a list of rooms
- 2.2.3. User enters desired text or uploads image
- 2.2.4. User selects “confirm post”

#### **2.3. Functional Requirement Label**

- 2.3.1. REQ 2.1 Create New Post

### **3. Create new Room**

**3.1.** To create a new room, the user has to select the “create new room” option after logging in. The user is then prompted to enter the name of the new room and confirm room creation.

#### **3.2. Stimulus/Response Sequence**

- 3.2.1. User selects “create new room”
- 3.2.2. User enters text for the room name
- 3.2.3. User selects “confirm new room”

#### **3.3. Functional Requirement Label**

- 3.3.1. REQ 5.1 Create New Room

### **4. Follow a Room**

**4.1.** To follow a room, the user must first select the “view rooms” option after logging in.

#### **4.2. Stimulus/Response Sequence**

- 4.2.1. User selects “view rooms”
- 4.2.2. User selects a room from the displayed list of rooms
- 4.2.3. User selects “follow room”

#### **4.3. Functional Requirement Label**

- 4.3.1. REQ 4.1 View Rooms
- 4.3.2. REQ 4.2 Follow Rooms

## List of Non-Functional Specifications

### Performance:

1. Requests processed: The site should be able to process at least 10 requests per second under ideal conditions.
2. Response time: Site response time when using high-speed internet should average a few hundred milliseconds, and should not exceed 5 seconds under normal load and use conditions.
3. Refresh time: Site content should be updated at least once every 30 seconds.
4. User Load: Initial rollout of the site should support up to 50 active users.
5. Storage requirements: The storage requirements of the core web app should stay fairly minimal, consisting of at most a few dozen megabytes of web code, text, and some supporting images. The database which stores user information such as posts and profile info will be primarily text based, however image uploads are a desired feature dependent on FAU's Lamp server limitations. Initial implementation of the site will involve a storage limit on the database of around 100 Mb. Although user account data will never be deleted, the database storing user posts may be cleared in some fashion to prevent exceeding storage capacity (i.e. deleting oldest posts first, or just deleting the entire post database when limit is reached).
6. Hardware requirements: The system will be running on FAU's Lamp servers and must not exceed the hardware capabilities of those servers.
7. Network requirements: Only text and some images would be sent to user devices, so network requirements are minimal. The system should be usable on connections greater than 1 Mb/s down for users, and the system should be functional on connections greater than 100 Mb/s down, 25 Mb/s up for the hosting server.

### Accessibility:

1. Training time: Users should require no more than a few minutes, ideally no more than 10, to learn how to use all of the site features.
2. Help pages: The site homepage will provide a brief overview of functionality and usage instructions. The homepage will also contain a link to a dedicated help page for more detailed instructions and FAQs.
3. Support: A support email will be displayed on various parts of the site for users who are seeking additional assistance.
4. Interface: The site should be usable without any interactions exceeding text input and clicking, in order to provide a consistent experience between desktop and mobile web use.

### Reliability (\*FAU Lamp Servers, not really in our control):

1. Average uptime: Although the site should at least maintain an average uptime of around 90% per day.
2. Probability of downtime: The site should experience downtime on average no more than once per day.
3. Average time to failure: The site should have less than one hour of downtime per day.
4. Time to restart after failure: Restart of the site after a failure should take less than one hour.
5. Percentage of events causing failure: Less than 1% of events on the system should cause failure.
6. Probability of data corruption on failure: Data corruption should occur on less than 1% of failures.
7. Data Protection: More critical data such as user account information will be backed up to a google drive account. Less critical data such as user posts will not be backed up due to its expendable nature.
8. Exception Handling: There will be exception handling implemented in all cases where exceptions may occur.

### Compatibility:

1. Supported browsers: The site should function on most Chromium and Firefox based browsers, as well as Safari. Specifically, Google Chrome, Mozilla Firefox, Microsoft Edge, and Apple's Safari browser will be tested and at least 2 should function with the system without issue. Variants of these browsers, such as Brave, should still work but might not maintain full functionality with the site.
2. Supported devices: The site should work on all mobile versions of the above mentioned browsers: Chrome, Firefox, Edge, and Safari.
3. Supported Operating Systems: The site should work on Linux, Windows, and Mac OS computers, as well as Android and iOS mobile devices using one of the supported browsers mentioned above.
4. Target-Dependent Statements: No target-dependent statements will be used in the initial rollout of the system.
5. Coding standards: The system will follow coding standards for HTML 5, CSS3/CSS4, MySQL, Javascript, PHP 8, and Python 3.
6. Naming Conventions: The system will follow the standard naming conventions for each respective programming language.

**Security:**

1. Login/Password System: Users will be able to create an account with a login and password of their choice.
2. Encryption: No encryption will be used in the system. No sensitive data is required by the users to use the web app.
3. Access Control: Front and back end code modification will only be accessible to the development team members. Users should not be able to modify the site code in any way.
4. Spam Protection: Users will be asked to solve a simple math equation in order to create an account.
5. Resource Utilization: A MySQL database will be used to store data pertaining to users, such as user posts and profile information. The database will be hosted by lamp.cse.fau.edu and will be accessed with PHP based logins and passwords.



## High-Level System Architecture

1. **lamp.cse.fau.edu Lamp Server:** Shall be the host for our 2021 Fall semester web development project
2. **MySQL Database:** MySQL Database will be the database system that will be used to store the data used in this project. Users will be creating a profile, adding to the database as well as deleting and editing.
3. **Visual Studios IDE:** The Visual Studios Integrated Development Environment is going to be the platform that the developers are going to use to handle the coding for the project. The languages to be used are as followed:
  - a. HyperText Mark-Up Language (HTML) - for front end development of the webpage
  - b. Cascading Style Sheet (CSS) - will be used to decorate the web pages
  - c. Bootstrap - will be used to create a better template for our web pages
  - d. JavaScript - for our web pages to be interactive with the user and user interface
  - e. Python - programming language to develop code on back end
4. **Django:** Django will be the framework for the project that will allow us to manipulate the database, html files, and functionality.
5. **Brackets Text Editor:** Brackets will be used specifically for web design. Languages used in this text editor as as follows:
  - a. HTML - for front end development of the webpage
  - b. CSS - for decorating the web page
  - c. JavaScript - for user interface on webpage

## Competitive Analysis

The analysis table consists of five competitor websites to evaluate their features to determine how many features they have relative to IQ Lounge. The columns represent websites, and rows represent features. All cases in the table are coded as 1 or 0 to reflect the presence or absence of a particular feature.

	<b>IQ Lounge</b>	<b>MS Teams</b>	<b>Twitter</b>	<b>Facebook</b>	<b>Discord</b>	<b>Pinterest</b>
<b>Private groups</b>	1	1	1	1	1	1
<b>Search</b>	1	0	1	1	1	1
<b>One-to-one chat</b>	1	1	1	1	1	1
<b>Add posts to private groups</b>	1	1	1	0	1	0
<b>Bookmarks</b>	1	1	1	1	0	1
<b>News feed</b>	1	0	1	1	0	1
<b>Secure login</b>	1	1	1	1	1	1
<b>Total</b>	7	5	7	6	5	6

### Microsoft Teams

Microsoft Teams is the collaboration platform where teams connect through shared notebooks, channels, and group or private chat sessions. It allows all team members to share resources within a team. Users can bookmark any messages, and access them in the flyout menu. Microsoft Teams uses modern authentication to provide a simple and secure sign-in process. However, users cannot access Microsoft Teams without a Microsoft account, search teams of interest, and it doesn't have a homepage news feed.

### Twitter

Twitter is an online social networking and communication platform that allows users to communicate through messages, called "tweets". It has a homepage news feed and can be accessed without an account. Twitter has a feature to select topics of interest. It has an easy and guided sign-in process. Twitter allows private messaging, however, the group message function is limited to up to 20 users. The Twitter community allows users to connect and share what they care about. Bookmarks let users save Tweets for quick access at any time.

## **Facebook**

Facebook is a website which allows users to connect, discover what is going on in the world, and share what matters to them. It has an easy and secure login process, and users can read anything on a Facebook page without an account. News feed is constantly updating on the homepage. Users can create a group chat or one-to-one conversation. Users can also search for facebook pages of interest and save posts they like. However, only a page owner can post on the facebook page.

## **Discord**

Discord is a video, voice, and text messaging platform that originally was built as a way for video game fans to communicate. Over time Discord has developed into a social network where people can search and join certain groups based on their interests. Discord does not begin with a news feed for students to scroll through. Users can contribute to the group they joined, and also have one-to-one conversations. The login process is easy and secure. there's no way to bookmark the messages to reference for later for users.

## **Pinterest**

Pinterest is a tool that lets users search for, upload, and collect images for later review. Users can collect individual posts in collections called boards, and then share their boards publicly or save them for private use. In addition to one-to-one conversations, Pinterest allows messaging for groups of up to 10 people. Login process is easy and secure.

## **Planned Advantages:**

The IQ Lounge provides students with the opportunity to connect with other students. Students will be able to securely login and, in addition to one-to-one conversations, have the ability to create private chat rooms and groups revolving around particular topics or social groups. Users can also use the search feature in order to find specific groups based on their own personal interests. Along with chats, users will also have access to many resourceful links recommended by students, and bookmark the links for the later review. Users will also be able to contribute to groups created by other students, and groups of interests will appear on the home page news feed.