

National College of Ireland

<Academic Year 2022/2023>

<Paolo Jr. Costarelli><20235305>

[x20235305@student.ncirl.ie](mailto:x20235305@student.ncirl.ie)

<Paddy O’Toole><21105120><x21105120@student.ncirl.ie >

< Muhanned Gssis><21743719>

< x21743719@student.ncirl.ie >

<New Foodie Social-Media>

Technical Report

Contents

[Executive Summary 1](#_Toc129699625)

[1.0 Introduction 2](#_Toc129699626)

[1.1. Background 2](#_Toc129699627)

[1.2. Aims 2](#_Toc129699628)

[1.3. Technology 2](#_Toc129699629)

[1.4. Structure 2](#_Toc129699630)

[2.0 System 2](#_Toc129699631)

[2.1. Requirements 2](#_Toc129699632)

[2.1.1. Functional Requirements 2](#_Toc129699633)

[2.1.1.1. Use Case Diagram 2](#_Toc129699634)

[2.1.1.2. Requirement 1 <Name of requirement in a few words> 2](#_Toc129699635)

[2.1.1.3. Description & Priority 2](#_Toc129699636)

[2.1.1.4. Use Case 2](#_Toc129699637)

[2.1.2. Data Requirements 3](#_Toc129699638)

[2.1.3. User Requirements 4](#_Toc129699639)

[2.1.4. Environmental Requirements 4](#_Toc129699640)

[2.1.5. Usability Requirements 4](#_Toc129699641)

[2.2. Design & Architecture 4](#_Toc129699642)

[2.3. Implementation 4](#_Toc129699643)

[2.4. Graphical User Interface (GUI) 4](#_Toc129699644)

[2.5. Testing 4](#_Toc129699645)

[3.0 Conclusions 4](#_Toc129699646)

[4.0 Further Development 4](#_Toc129699647)

[5.0 References 4](#_Toc129699648)

[6.0 Appendices 4](#_Toc129699649)

[6.1. Project Plan 4](#_Toc129699650)

[6.1. Collaboration Summary 4](#_Toc129699651)

# Executive Summary

This technical report was written to detail the results of our team project, to talk about the technologies and software that was used, the challenges we faced and how we overcame them, and our closing thoughts on how the project went. Our team decided to make a social media website for foodies. We divided the work between the three of us, Paddy took the front end, and Muhanned and Paolo took the back end functionality. We all worked from the same files via a OneDrive folder. Our main resource was YouTube for figuring out how to make our website features functional. We really struggled with some functions such as the chat feature, and in the end, we just had to do without it because we couldn’t get it working. Overall, we are happy with the parts of the website that we were successfully able to make functional, but we definitely would have liked to have implemented more functionality such as a fully functioning notification page and chat function. It was only halfway through the project when we all realized how big of an undertaking it is to create a fully functional social media website.

# Introduction

## Background

We undertook this project because all three of us use social media all the time to keep in touch with friends and family, and we all have a passion for food, so we felt that making a social media website for foodies would be the perfect project idea for us.

## Aims

We aimed to achieve a fully functional social media website for foodies that would allow users to connect with other users and share their passion for food. We wanted to allow chefs to be able to use our website to find work and for foodies to connect with other foodies.

## Technology

We used HTML, CSS, JavaScript, PHP, PhpMyAdmin, and MySQL to create our website.

## Structure

This document was written to talk about the details of our project, including the technologies that were used to create the website, the reason why we created this website, and our future plans for this website. The information provided in each section is self-explanatory based on the title of each section point.

# System

## Requirements

All requirements should be verifiable. For example, experienced controllers shall be able to use all the system functions after a total of two hours training. After this training, the average number of errors made by experienced users shall not exceed two per day.

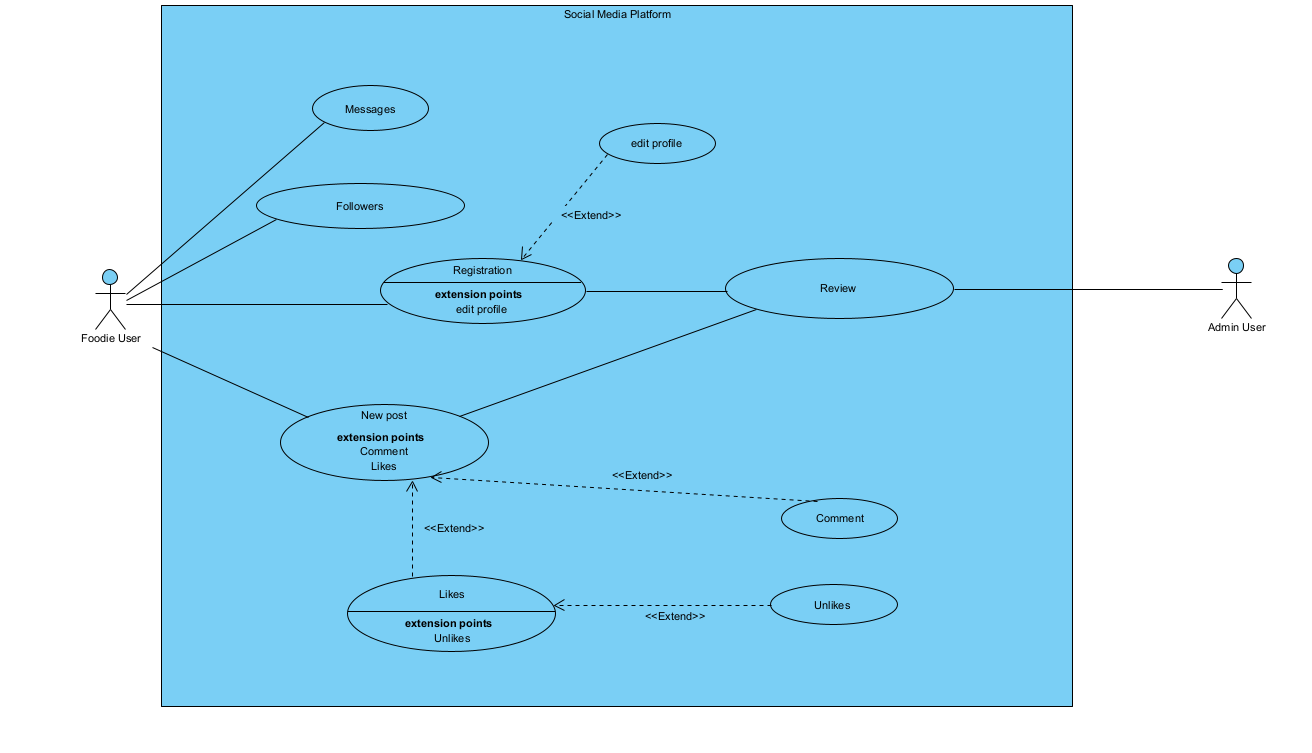
## Functional Requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

## Use Case Diagram

**Use Case Diagram**

***Name***

Requirement #01: ‘Registration’

***Scope***

The scope of this use case is to look at how new users will register on our website.

***Description***

New users are required to register themselves as either a foodie or a chef by entering a valid username, password, and email address. They will have to confirm the email address. Once the user is registered, they will have full access to the website.

***Flow Description***

* + *Precondition*

The System is initializing*.*

* + *Activation*

When the user opens our website and is brought to the main page, they will be given an option to either log in or to create a new account it they are not already a registered user.

***Main Flow***

* The system prompts the user with a text area to enter username and password.
* The user enters their details.
* The system sends the entered details to the database to verify that it exists.
* The user is now logged in and has access to the system.

***Alternative Flow***

* The system gives the user details to input their credentials to create a new account.
* The user enters details, registers new account and submits
* The System does a check on the credentials entered to make sure it is up to standards.
* The user account is successfully created.

***Exceptional flow***

* The System prompts the user with a textbox to enter in their email and password
* The user enters their credentials.
* The System sends the credentials enter by the user to the database to check if it exists.
* The records do not exist in the database, the user has failed to login. The user is then asked to try logging in again

***Termination***

The system displays the index page of the website.

***Post Condition***

The system goes into a wait state until the user logs out of it.

***Name***

Requirement #02: ‘checking private messages’

***Scope***

The purpose of this use case is to verify that each registered user is able to log into his or her mailbox and check the messages in it.

***Description***

This use case describes the path a user must follow in order to enter his personal mailbox and see if there are any new messages.

***Flow Description***

* + *Precondition*

The system is active, and the user has already logged into his personal account.

* + *Activation*

This use case starts when the user clicks on the icon to access the personal mailbox.

***Main Flow***

* The user opens the social media page.
* The user logs in by entering username and password.
* The user, from the homepage of the platform, clicks on the icon to access the personal mailbox
* The user will be able to view the messages within his or her mailbox.

***Alternative Flow***

* The user opens the social media page.
* The user logs in by entering username and password.
* The user enters on his personal homepage.
* The user, from the personal homepage of the platform, clicks on the icon to access the personal mailbox.
* The user will be able to view the messages within his or her mailbox.

***Exceptional flow***

* The user opens the social media page
* The user logs in by entering username and password
* The user, from the homepage of the platform, clicks on the icon to access the personal mailbox.
* The user cannot access his mailbox because an error occurs with the server.
* The system displays a message indicating a problem with the server connection.

***Termination***

This requirement will be complete when the user can check whether there are any messages in the personal mailbox.

***Post Condition***

The user will be in his/her mailbox.

***Name***

Requirement #03: ‘follow another users profile’

***Scope***

The purpose of this use case is to give any registered user the possibility of following another user within the platform, so that they can more easily view the content published by the chosen user.

***Description***

A registered user may decide to follow another registered user.

***Flow Description***

* + *Precondition*

The system is active, and the user has already logged into his personal account.

* + *Activation*

This use case starts when the user clicks on the 'follow' button within another user's page.

***Main Flow***

* The user opens the social media page.
* The user logs in by entering username and password.
* The user, after looking at content created by another user on the general page, will click on his username to enter the selected user's homepage.
* The user will be able to view the follow ‘button’.
* The user will click on the button to start following the chosen user.

***Alternative Flow***

* The user opens the social media page.
* The user logs in by entering username and password.
* The user enters on a selected foodie page (ex. Pizzas)
* The user, after looking at content created by another user on the general page, will click on his username to enter the selected user's homepage.
* The user will click on the button to start following the chosen user.

***Exceptional flow***

* The user opens the social media page
* The user logs in by entering username and password
* The user, after looking at content created by another user on the general page, will click on his username to enter the selected user's homepage.
* The user will be able to view the follow ‘button’.
* The user will click on the button, but the system doesn’t allow the user to complete the operation.
* The system will push back then user to the homepage.
* The system displays a message indicating a problem with the website.

***Termination***

This requirement will be complete when the user can check whether there are any messages in the personal mailbox.

***Post Condition***

The user will be the page of the user he started to follow.

***Name***

Requirement #04: ‘post creation’

***Scope***

The scope of this use case is to verify that any registered user can share a post. ***Description***

A registered user can create and share new posts within the website. They can be viewed by anyone who has registered for social media.

***Flow Description***

* + *Precondition*

The system is active, and the user has already logged into his personal account.

* + *Activation*

This use case starts when the user enters his personal page and clicks on 'create new post'.

***Main Flow***

* The user opens the social media page.
* The user logs in by entering username and password.
* The user enters their personal page.
* The user clicks on 'create new post'.
* The user chooses whether to enter a text, a photo, or a video.
* The user clicks on 'share post'.
* The new post will be visible within social media.

***Alternative Flow***

* The user opens the social media page.
* The user logs in by entering username and password.
* The user enters on the homepage of the social media.
* The user clicks on 'create new post'.
* The user chooses whether to enter a text, a photo, or a video.
* The user clicks on 'share post'.
* The new post will be visible within social media.

***Exceptional flow***

* The user opens the social media page.
* The user logs in by entering username and password.
* The system shows a message saying that the password or the username is wrong.
* Users cannot share a new post until they enter the correct username and password.

***Termination***

This requirement will be complete when the new post is correctly entered into the platform, and other users are able to view and comment on the post.

***Post Condition***

The system will remain on the same page where the user has expressed their liking.

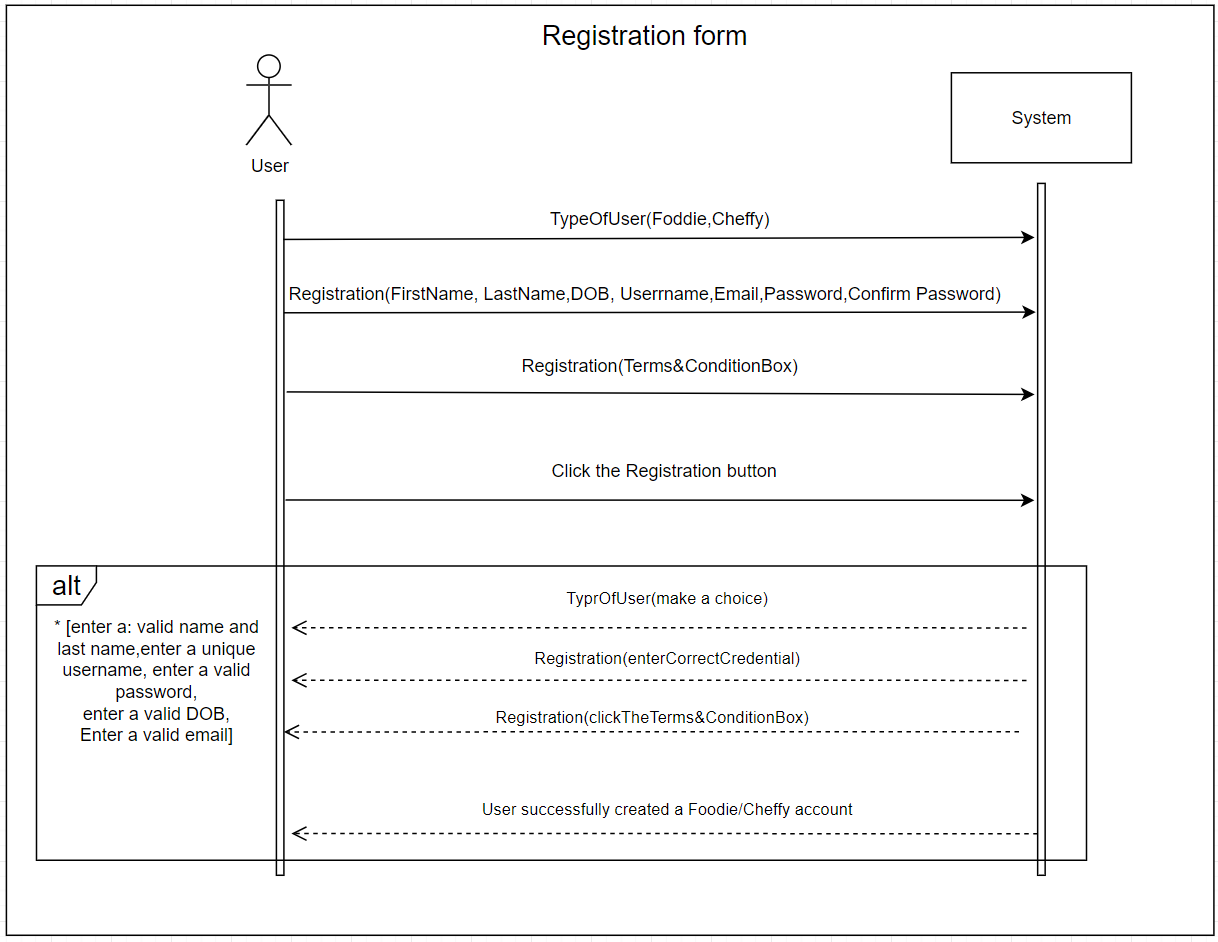
## Non Functional-Requirement

* ***6.1* *Usability:*** The app will be very user-friendly, and the design will be simple with clear functions. The web app will encourage users to engage more with each other.
* ***6.2* *Availability:*** The app will always be available to anyone with a smartphone, computer/laptop or tablet and connected to the internet.
* ***6.3 Reliability:*** Reliability is a very important requirement as it shows how long can the web app run for without any failures.
* ***6.5* *Recoverability:*** Besides having all data shared in the cloud, all the files will be stored in an external hard drive.
* ***6.6* *Scalability:*** The web app will be using MySQL which can handle a huge number of data nodes, relationships, and properties. Meaning that the app will still function properly even after large changes.
* ***6.7* *Performance:*** We will optimize our web app’s performance by reducing the size of our stored files and minimizing the HTTP requests.
* ***6.8* *Supportability:*** The app will have remote support. Will provide the users an e-mail to send any queries or complaint.
* ***6.9* *Security:*** The app will have a login system, which will have a password and a unique e-mail to each user. The passwords stored in the database will be encrypted for the user’s security.
* ***6.10* *Capacity:*** To enhance the capacity of our website we will make the inflow average equal to the outflow average. We will also make use of the Stress/Load tests to check the capacity of our web app.
* ***6.11* *Portability:*** The app will be developed using HTML, PHP, phpMyAdmin and MySQL, meaning that it will be deployable on all web hosts.

## Design & Architecture

This social media app is designed to help users create and share content with others, connect with friends and followers, and participate in discussions. The app can be accessed from any mobile device or her web browser and includes features such as news feeds, messages, notifications, and search.

System architecture:

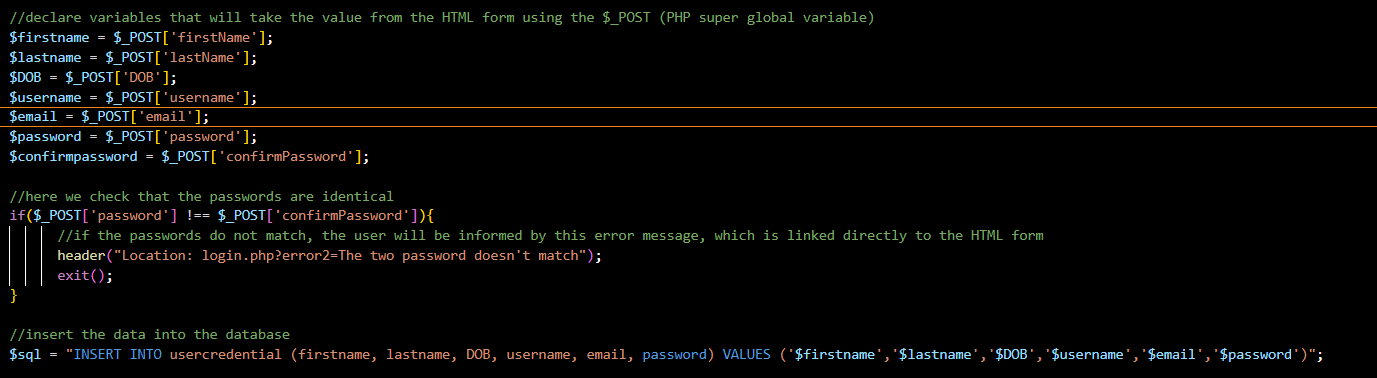


## Implementation

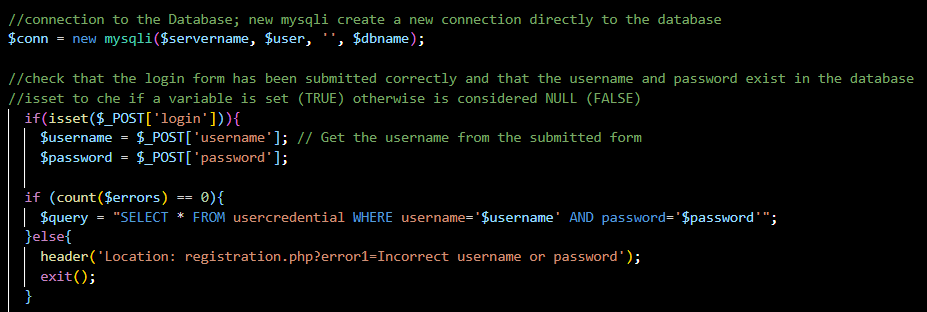
1. Connection with the database



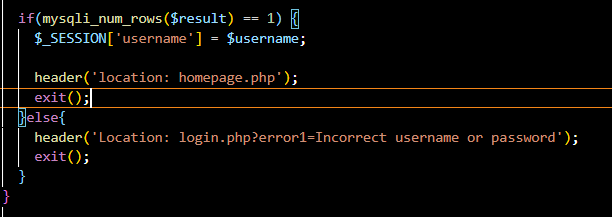
1. Declaration of variable that will store the value from the HTML form, and then sent to the chosen table in the database.



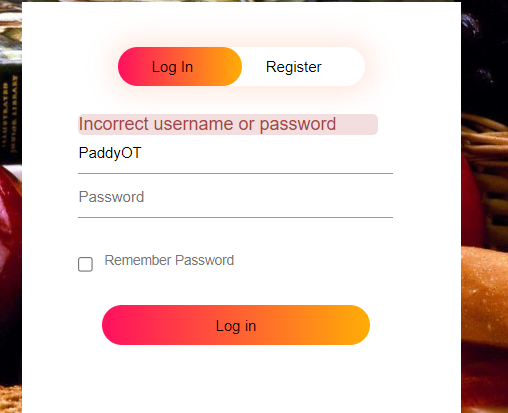
**C)** Check whether the user has entered the correct username and password.



**D)** If there are any errors during login The user will display an error message



And this is how the code looks on the website:



**D**) I was tasked with making the front end for our website. I will talk about some of the parts of my code that I found most interesting.

Post function - The most important piece of functionality for our website is the post function. I wrote the code for the template of the post box, and Muhanned used this to add the backend functionality for it. Here is a snippet of the code:

Text

Description automatically generated

And this is how the code looks on the website:

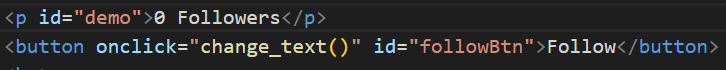
Graphical user interface, application, Teams

Description automatically generated

**E)** I created a div named userPost-box that the post box would go into, I made a description box using the input type=“text”, allowing the user to enter in an image description. I created a post button using the <button> tag that allows the user to post his description and image.

Follow button:

I created a button on the user profile page then when clicked, changes the follow button to unfollow, and above it, it changes the amount of followers from 0 to 1. This is the code for that:



Text

Description automatically generated

And here is how it looks on the website:

Application

Description automatically generated with low confidence Graphical user interface, application

Description automatically generated

I used JavaScript to create this function. In the script tag, I created an if/else statement that would come into action once the follow button was clicked. By using the .textContent property, I can change the original ‘Follow’ text to whatever I want, in this case I changed it to ‘Unfollow’.

**F)** Navigation bar:

This piece of code shows how I made the navigation bar for the website. I wrapped the images for the symbols in buttons that the user can click and be brought to the pages.

Text

Description automatically generated

This is how it looks on the website.

A picture containing screenshot

Description automatically generated

**G)** Delete your account page:

This piece of code shows how I created the box for the account deletion page.

Text

Description automatically generated

This is how it looks on the website:

Graphical user interface, text, application, email

Description automatically generated

**H**) Here we have the **Database** which has 3 methods:

**Connect:** This instantiates the connection to the database

**Read:** Reads data from the MySQL database

**Save:** Sends data from the website to the database

Graphical user interface

Description automatically generated with medium confidence

**I)** Here this file inherits connect.php which has the Database class. This Post class has 2 methods:

**Create\_post:** which makes use of the save method in the database class.

**Get\_posts:** this one make use of the read method in the database class.

Text

Description automatically generated

**L)** Here we have the post template. The first php tag prints the data from the post row in the database. The second php tag checks if the same post has an image or not if it does it will print it out in the post.Text

Description automatically generated

**M)** Here this file inherits post.php and creates a new instance of the Post class and calls the create\_post method

Text

Description automatically generated

**N)** Here two files inherit the post.php and create a new instance of the Post class then they call the get\_posts method, to display any posts that are stored in the database

A picture containing text

Description automatically generated

**O)** In this piece of code, we have a loop. This loop checks if the posts variable is empty or not, and we are storing the whole table in this variable. Then includes the post-d which has the post template. This loop keeps looping until it printed every post in the database.

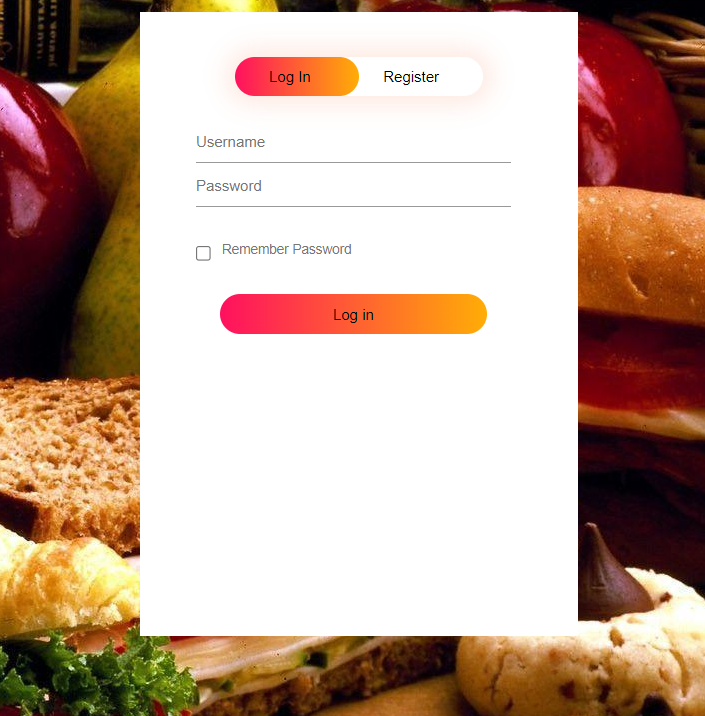
Text

Description automatically generated

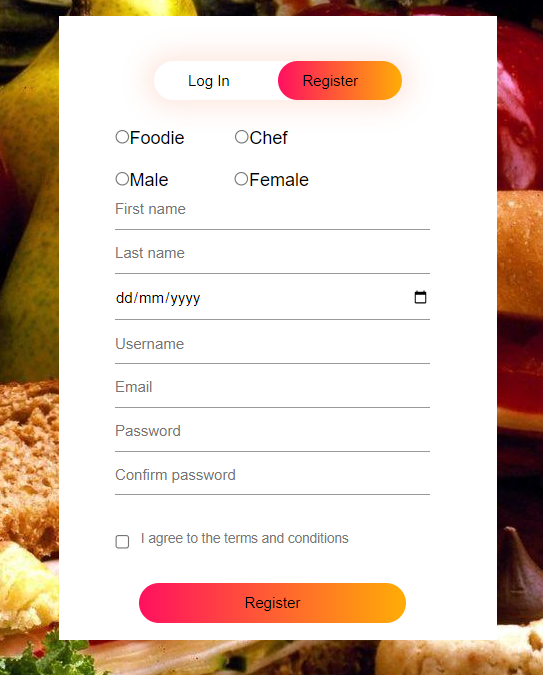
## Graphical User Interface (GUI)

Provide screenshots of key screens and explain what can be seen in each one.

1. Log in page.



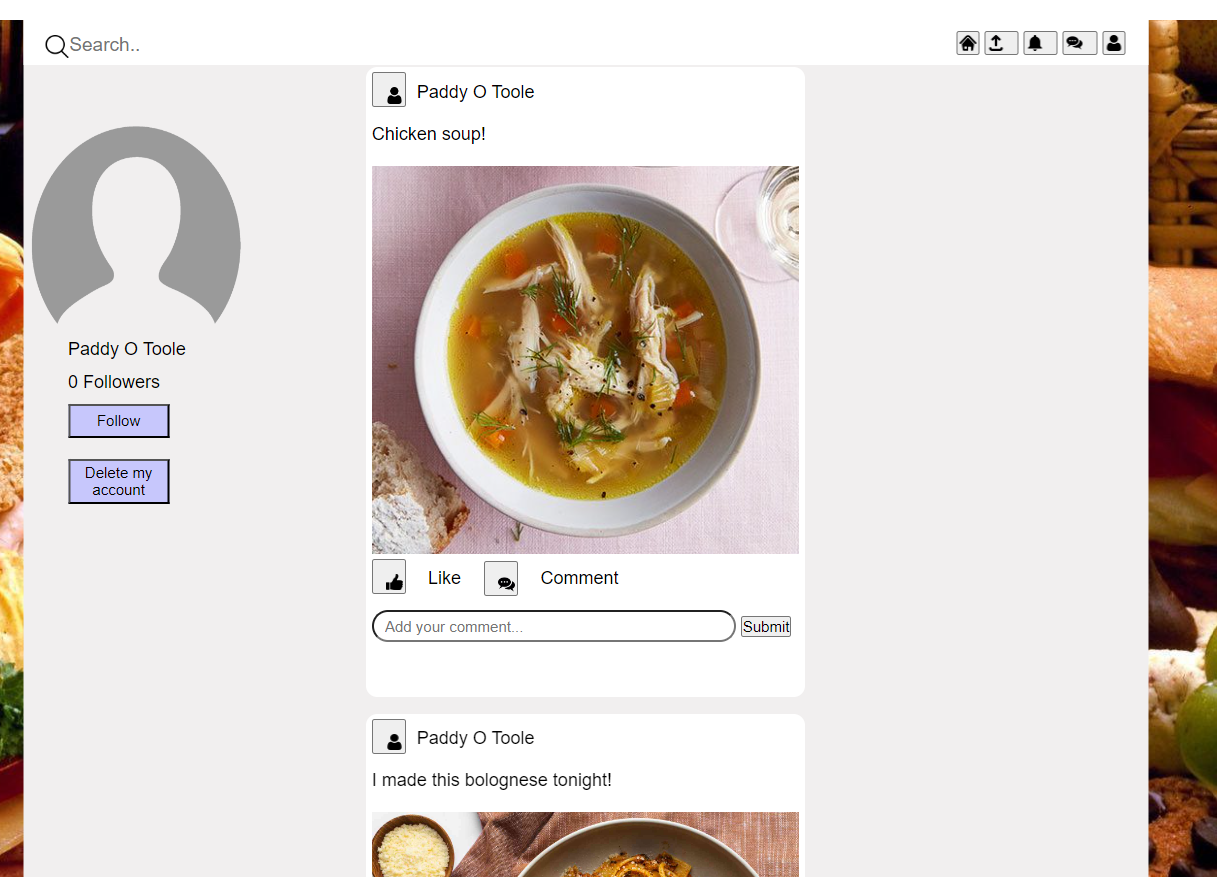
1. Registration page

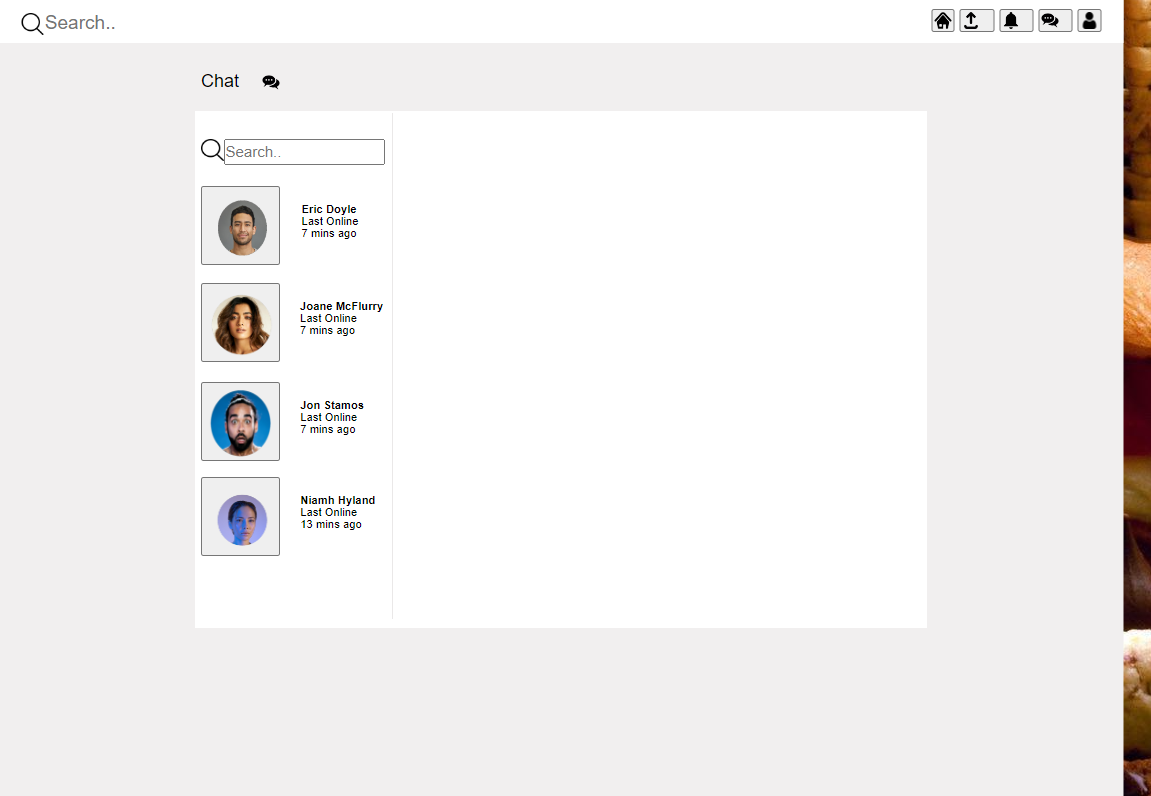


1. Homepage where it showed all the post on the website.

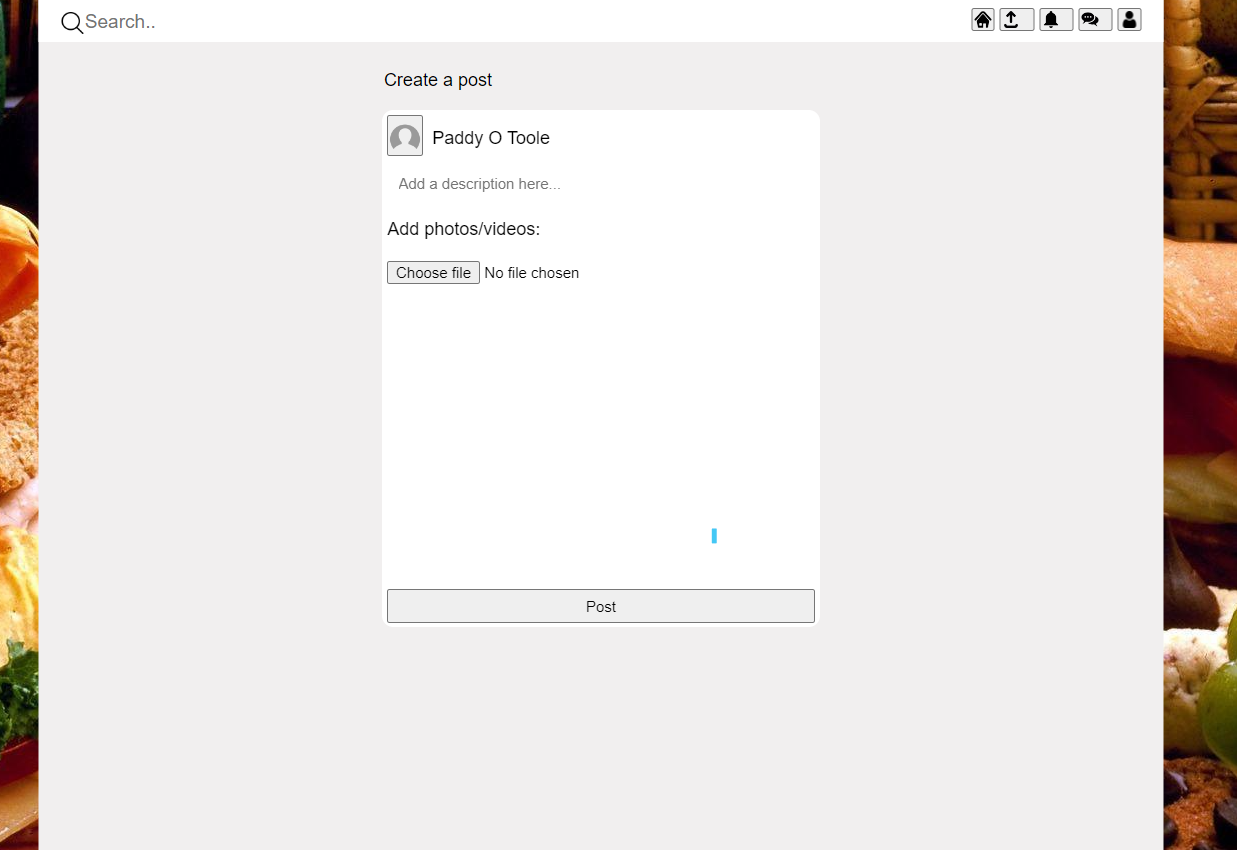


1. User page.



1. Chat/messaging page.

d) Create a post page.



## Testing

For our Project we are going to use black-box testing together with some of its methods, such as:

* Equivalence Partitioning
* Boundary Value Analysis
* Decision Table Testing

We are also going to implement with-box testing using Control-flow based techniques.

We plan to test the main functionalities that only Log-in, Registration and make a post.

# Conclusions

Overall, as a group, we really wanted to have more working features in our website. There were many aspects of the website that we weren’t able to pull off such as making our website responsive, being able to follow another user and the chat function.

**Advantages:**

Our login/registration feature is fully functional, allowing the user to create a new account and sign in. The post function is also working as we intended it to. We are happy with these features that we got working.

**Disadvantages:**

There are a few back end parts of our website that don’t work. We really struggled with many aspects of this project and the end product isn’t what we had initially hoped for. The main difference that separated our app from our competitor Pepper, was that Pepper doesn’t have a messaging function, which we wanted our website to have. Unfortunately, we were unable to get the chat function working. Some other important features such as the notification page and the chat function also do not work.

**Strengths:**

Our group came together well to figure out how to make the login/register page fully functional, and we worked well as a team to get the Post function working.

**Limitations**:

Our group struggled to work together as a team to complete each step of the project. We didn’t realize at the start just how much work is involved in creating a social media platform, and the difficulties and frustrations that would await us going forward. We struggled with most of the important back end functionality including setting up a user profile, chat function, following another user, and the notification page.

# Further Development

With some more resources and time, our website could improve in a few ways. We can add a functional chat system for users to interact with each other via messages private one-on-one or group chats. We can make the notification feature functional. We can also make the feed page more interactive by adding the categories feature which will allow the users to choose the type of content they would like to see on their home page. Create the following system to allow users to follow each other and keep up with their followee’s new content.

# References

(@QuickProgramming, 2021)

(@ProgrammingKnowledge, 2019)

(W3SCHOOL, 1998)

(geeksforgeeks, s.d.)

(@TechTips, 2020)

# Appendices

This section should contain information that is supplementary to the main body of the report.

## Project Plan

**The Idea Phase:**

           The first phase was the Idea phase. In this phase, we took the first week of the week to come up with an idea for our project. We had a few ideas, such as making a game (Puzzle, platformer, or FPS game), or any application using Java (since we had the most experience with this programming language), or a website. Then after a lot of brainstorming and consulting with a few lecturers, and for the sake of simplicity, we decided to make an almost unique social media platform for foodies.

**The Planning Phase:**

            After producing an idea for our project, we started our planning phase. For this phase, we wanted to come up with a plan for the project, including what programming languages will we be using, and what kind of technologies and tools we’ll make use of. We also wanted to divide the workload between us, in other words, who will be doing what. After doing our research and what we needed to do, we made a project plan which took a few days to finish.

**The Design Phase:**

           For the third week after finishing our planning phase. We started our designing phase. For this phase we wanted to produce a template for our website. How do we want our website to look like, and what kind of features do we want the user to enjoy. What kind of visuals do we want the website to have, and what colour scheme will we use for the website. After discussing it for a while, and all members sharing their thoughts, we have concluded our design phase, by producing our website template with our desired design and colour scheme.

**The Implementing Phase:**

           After the planning phase, the most important phase starts. The implementation phase. In this phase, we put all of our phases together. Here we started coding our website from scratch. This phase took the longest, as we chose to use a new language that we have never used before **PHP**. It was stressful, and it took a lot of time, but we managed to learn a good bit about the language. Each one of us had their own part of the website to code, but we kept working together. We helped each other when we could. Every now and then we’d one member would explain his code to the other members, we did this so we could maximize our learning from this project and for us to understand our own code in a better way.

**The Debugging Phase:**

            Now we have the debugging phase in a way this phase started a little bit after the implementing phase, as we decided to debug as we go. Instead of finishing the project and then we started debugging. This phase ended last, as even when we finished coding our website, we did have a few bugs here and there and we managed to fix them. We tried to debug our code individually, and together, we researched the internet to see if other people had the same bugs as us, and of course, we asked our lecturers for guidance when we had the chance.

**The Submission Phase:**

           This is the last phase of our project planning. We wrapped up our project. Started to put all the files together and lastly, we started preparing our final report and our final presentation.

## Collaboration Summary

We were very well organized for this group project. We divided the work between us equally everyone played their part in the project and delivered their share of the work. We did face a few obstacles along the way, of course, but we stayed organized and stayed on top of things. We had a Teams group where we shared the files of the project between us and had call meetings every week or so. We tried to meet up once a week in college to see where everyone is at. We also stayed after class to work on the project together, a few days every week. We used Monday.com to keep track of our progress in the project. We also made use of GitHub to share the code between us and try to debug together when necessary. We have created a What’s up group to be able to reach each other when we don’t have access to our personal laptops or the college’s computers. Here we will provide some screenshots of the groups and the websites we used to collaborate our work:

1. Graphical user interface, text, application, email

   Description automatically generatedA screenshot of a computer

   Description automatically generatedGraphical user interface, text, application, email

   Description automatically generatedA screenshot of a computer

   Description automatically generatedGraphical user interface, text, application

   Description automatically generatedTable

   Description automatically generated