

Milestone 4: Beta Launch and Final Product Reviews

Course: CEN 4010 - Principles of Software Engineering

Semester: Summer 2020

Team Name: FunkyTech

Team Number: 7

Project/Application Name: Gather+

Team Members

- **Michael Resendes:** mresendes2017@fau.edu | Product Owner
- **Brent Werne:** bwerne2018@fau.edu | Back-End Developer
- **Betsy Kuruvila:** bkuruvil@fau.edu | Front-End Developer
- **Michael Simon:** msimon2015@fau.edu | Scrum Master

Date: 7/28/20

Document Version: 3.0 (7/28/20)

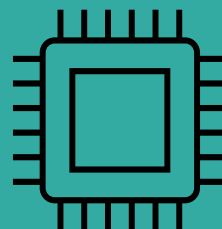


Table of Contents

Peer Evaluation.....	3
Product Summary.....	4
Usability Test Plan.....	5
QA Test Plan	7
Code Review	10
Self-Check: Best Practices for Security	13
Self-Check: Adherence to Original Non-Functional Specs.....	14
Beta Launch Demonstration Video Information	15

Peer Evaluation

We, the members of group seven, agree to the following percentage distribution for our involvement in the Milestone 4 assignment.

Michael Simon (25%) provided the scripts that were used to populate the Games Hub.

Brent Werne (25%) wrote the original draft for the Milestone 4 document, was involved in writing/debugging the back-end aspects of the Gather+ application and populated the Games Hub.

Betsy Kuruvila (25%) created and oversaw front-end design of the Gather+ application and helped populate the MySQL tables in the database.

Michael Resendes (25%) wrote the revised draft of the Milestone 4 document, was involved in writing/debugging the back-end aspects of the Gather+ application, managed the database on the LAMP server, and created the demonstration video.

Product Summary

FunkyTech

The name of our product is called Gather+. It is a web-application centered around the idea of coming together in a time where physical distancing is required due to COVID-19, and social interactions have become more important than ever. Gather+ is a social media website that brings people together based on common interests. The content of Gather+ will be divided between its four main hubs: Books, Podcasts, News and Games. These hubs will serve as centers for a variety of topics that are intended to help build camaraderie among Gather+ users who share a similar passion for certain topics.

Gather+ will allow its users to:

- Create a Gather+ account that they will be able to manage, edit, or delete
- Join a variety of communities centered on topics found within each of the four hubs and take part in a community discussion centered on that topic.
- Browse different book titles in the Books hub and read them via an e-reader.
- Listen to a variety of ongoing podcasts hosted on the Podcasts hub.
- Read current news articles found on the News hub.
- Play several popular online games listed on the Games hub.

Whereas many other social media sites focus solely on user-submitted content, Gather+ will instead focus on fostering relationships between its users and encouraging all communities to enjoy the many interests that we all share during this unique and unprecedented time.

Gather+ URL: https://lamp.cse.fau.edu/~cen4010s2020_g07/GatherPLUS

Usability Test Plan

FunkyTech

Community Message Board Functionality Test

Test Objectives

In this usability test, we will be testing the Community messageboard feature found in placards on Gather+. Community messageboards are intended to be places of discussion that allow members of a placard Community to converse with other Community members on the topic of that placard. The goal of this test is to ensure that the user trying to submit a message to the board is signed in and a part of the Community before posting a message. In order to properly test this feature, we will use three separate set-up cases. The first case being a user is not signed-in to a Gather+ account, the second being a user is signed-in but not a member of the Community they are trying to post in, and the third set-up being the user is signed in and the user is a member of the community. Both the first and second set-up cases should not allow a user to post a message in the Community messageboard, whereas the third set-up should allow the user to successfully post their message.

Test Plan

In order to properly test the Community messageboard, we will need to test the messageboards found on placards in all four hubs on Gather+: Books, Podcasts, News, and Games. In each hub, we would like to test one Community messageboard with all three setups. This ensures that the code for the messageboard is being implemented properly on all four hubs. The different hubs are each slightly different from each other and all access different databases. The test should be performed as if the tester is a user who has never used Gather+ before. The tester should use a standard computer/laptop with internet connectivity and one of the following three internet browsers: Google Chrome, Microsoft EDGE, or Mozilla Firefox.

Gather+ Test URL: https://lamp.cse.fau.edu/~cen4010s2020_g07/GatherPLUS/

First Test Case:

1. The tester will start from the home page of Gather+ and should not be signed-in to a Gather+ account. The tester will click on the link for the Books hub and will be taken to the hub. He will then select one of the book placards listed on the hub and will be taken to that placard's Community board.
2. The tester will attempt to post a message on that Community's messageboard and should subsequently receive the following message: "Please Sign-In To Join The Community". The tester should also observe that the message was not posted on the Community messageboard.
3. The tester should then return to the home page of Gather+ and repeat this sequence for the Podcasts, News, and Games hubs.

Second Test Case:

1. The tester will begin on the home page of Gather+ and should start by signing-in to his Gather+ account (or creating an account if they have not done so already). After successfully signing-in, the tester will be taken to his profile page.
2. He will then click on the link for the Books hub and will be taken to the hub. Subsequently, he will select one of the book placards listed on the hub and will be directed to that placard's Community Message Board.
3. The tester will attempt to post a message on that Community's Message Board and should subsequently receive the following alert on the Community board: "Please Join Community Before Posting". The tester should also observe that the message was not posted on the Community Message Board.
4. The tester should then return to the home page of Gather+, remain signed-in to his Gather+ account, and repeat sequences 2-3 for the Podcasts, News, and Games hubs.

Third Test Case:

1. The tester will begin on the home page of Gather+ and should start by signing-in to his Gather+ account (if he has not done so already). After successfully signing-in, the tester will be taken to his profile page.
2. He will then click on the link for the Books hub and will be directed to the hub. He will then select one of the book placards listed on the hub and will be taken to that placard's Community board.
3. The tester should first try and post a message on the Message Board before joining the Community to confirm that he is still unable to post. The tester should then click the "Join the Community!" button on the placard description panel and should subsequently see the following message: "Successfully Joined Community". After this, the tester should click the "Join the Community!" button again and should subsequently see the following alert banner: "You are already a member of this community".
4. Finally, the tester should try and post a message on the Community's Message Board. The tester should notice that the message is posted.
5. The tester should then return to the home page of Gather+, remain signed-in to his Gather+ account, and repeat sequences 2-4 for the Podcasts, News, and Games hubs.

Questionnaire Form

After completing the three test cases for each hub, the tester should be asked to fill out the following questionnaire. The tester should be honest and take the questionnaire as if the tester was a user that didn't know anything about the site before executing the test cases.

Questionnaire Link

https://docs.google.com/forms/d/e/1FAIpQLSfJHl8DpPZ7H49WEBNFkAYqd1zuuj0q94TOyMvw5OjqYDMGgg/viewform?usp=sf_link

QA Test Plan

FunkyTech

Community Messageboard Functionality Test

QA Test Objectives

In this QA test, we will be testing the Community messageboard feature found in placards on Gather+. There will be three objectives for this test...

1. The **First** objective will be to ensure that a Community messageboard requires a Gather+ user to be signed-in to their account and be a member of the Community before they are able to post a message on the Community messageboard.
2. The **Second** objective will be to ensure that a Community messageboard can store posts with a variety of numbers, characters, and symbols in their message.
3. The **Third** objective will be to ensure that a Community messageboard can store many different posted messages from its Community. For the purpose of this QA Test, we will attempt to store 20 messages on the messageboard.

Hardware/Software Setup

The test should be performed as if the tester is a user who has never used Gather+ before. The tester should use a standard computer/laptop with internet connectivity and one of the following three internet browsers: Google Chrome, Microsoft EDGE, and Mozilla Firefox.

Feature to Be Tested

This QA test will be testing the Community messageboard feature found in placards on Gather+. Community messageboards are intended to be places of discussion that allow members of a placard Community to converse with other Community members on the topic of that placard. Thus, it is important to ensure that the messageboards hosted by these communities are able to properly handle management of who can post, what can be posted, and how many messages can be stored in each database table. Each Community messageboard will need to be able to access SQL tables that contain a user's account information, a list of the communities a user has joined, and a list of the messages posted on the messageboard itself.

Gather+ Test URL: https://lamp.cse.fau.edu/~cen4010s2020_g07/GatherPLUS/

Test Cases

First Test Case (Sign-In/Member Requirements):

1. The tester will start from the home page of Gather+ and should not be signed-in to a Gather+ account. The tester will click on the link for one of the hubs and will be taken to that hub. They will then select one of the placards listed on the hub and will be taken to that placard's Community board.
2. The tester will attempt to post a message on that Community's messageboard and should subsequently receive the following message: "Please Sign-In To Join The Community". The tester should also observe that the message was not posted on the Community messageboard.
3. The tester should then return to the home page of Gather+ and sign-in to their Gather+ account (or create one if they have yet to do so).
4. The tester should then return to the placard page they were previously on and attempt to post a message on that Community's messageboard, after which they should see the following alert on the Community board: "Please Join Community Before Posting".
5. The tester should then click the "Join the Community!" button on the placard description panel and should subsequently see the following message: "Successfully Joined Community". After this, the tester should click the "Join the Community!" button again and should subsequently see the following message: "You are already a member of this community".
6. Lastly, the tester should try and post a message on the Community's messageboard. The tester should then notice that the message is posted.

Second Test Case (Message Content):

1. After completing the first test case, the tester should then remain on the placard Community page and submit several different messages containing a variety of characters, symbols, and numbers.
2. The tester should be able to successfully submit at least 3 new messages with all these aspects present in their message.

Third Test Case (Message Board Reliability):

1. After completing test cases 1-2, the tester should then continue to submit messages to the Community messageboard until there are 20 different messages listed on the board.
2. The tester should be able to view all 20 messages posted on the board and should submit one final 21st message to ensure that more posts could be added to the board.

Test Results

The following table contains the results of implementing the following QA test plan on three separate internet browsers: Google Chrome, Microsoft EDGE, and Mozilla Firefox.

Test Case	Chrome	EDGE	Firefox
Case 1 (Sign-In/Member Requirements)	PASS	PASS	PASS
Case 2 (Message Content)	PASS	PASS	PASS
Case 3 (Message Board Reliability)	PASS	PASS	PASS

Code Review

FunkyTech

In this section, we will be reviewing the code for the Message Board. Our group's coding style is functional coding. When accessing each feature of the website we try to ask ourselves how we can make this feature's code as independent as possible from other feature's code in order to prevent errors occurring when a different feature is added. We found that this approach was useful when collaborating with a group and code reviews were made easier because of it.

The code that was sent for review is listed below with added comments:

```
<?php

// Error Handler
set_error_handler("errorHandler");

function errorHandler()
{
    echo " ";
}

// Connects to Gather+ SQL Database
$db = new mysqli("localhost", "cen4010s2020_g07", "faueng2020", "cen4010s2020_g07");

if ($_POST["text"] == "")
{
}

//BETSY: IT WOULD BE BETTER IF ALL OF THE H4 ALERTS ARE CHANGED TO H5 TO KEEP IT VISIBLE BUT NOT VISUALLY OVERWHELMING
else if (!isset($_SESSION["username"]))
{
    echo "<h4 class=\"alert alert-danger text-center\">please Sign-in To Use Board</h4>";
}

else
{
    // Retrieves userID from user_accounts
    $sql = "SELECT userID FROM user_accounts WHERE username='$username'";
    $result = $db->query($sql);
    $row = $result->fetch_assoc();

    $userID = $row["userID"];

    $placardNoSpace = str_replace(" ", "", $placardName);

//BETSY: HOW HAVE WE TIED THE $PLACARDNOSPACE VALUE TO THE MESSAGEBOARDS IN SQL? IS IT IN THE STRUCTURE?
    // Retrieves Messageboards from communityMembers
    $sql = "SELECT Messageboard FROM communityMembers WHERE userID='$userID' AND Messageboard='$placardNoSpace'";
    $result = $db->query($sql);

    // Error Message if username Not Found in Database
    if ($result->num_rows == 0)
    {
        echo "<h4 class=\"alert alert-warning text-center\">Please Join Community Before Posting</h4>";
    }

    else
    {

```

```

$username = $_SESSION["username"];
$message = $_POST["text"];

$sql = "INSERT INTO $placardNoSpace (username, message) VALUES ('$username', '$message')";

if($database->query($sql))
{
    echo "<h4 class=\"alert alert-success text-center\">Success</h4>";
}
else
{
    echo "<h4 class=\"alert alert-danger text-center\">Could Not Submit Comment</h4>";
}
}

}
$placardNoSpace = str_replace(" ", " ", $placardName);

// Retrieves messages
$sql = "SELECT * FROM $placardNoSpace";
$result = $database->query($sql);

//BETSY: IS THE 'DIE' ALERT LIKE THE OTHER BANNER ALERTS OR IS IT A SIMPLE PHP ALERT?
// Error Message if Site is Unable to Retrieve Information
if (!$result)
{
    die("Error: Unable to Connect to Database");
}

else
{
    // Creates array of message Info
    while ($row = $result->fetch_assoc())
    {
        $name = $row['username'];
        $message = $row['message'];
        $userImage = "";

        $sql = "SELECT userImage FROM user_accounts WHERE username='$name'";
        $result2 = $database->query($sql);

//BETSY: DO WE HAVE A MESSAGE AVAILABLE TO USERS SPECIFYING FILE TYPES AND SIZE LIMITATIONS FOR THE
// USER IMAGES? ARE THERE CONSTRAINTS FOR THE SAME APPLIED BY THE DATABASE?
        if ($result->num_rows != 0)
        {
            $row2 = $result2->fetch_assoc();

            $userImage = $row2["userImage"];
            $userImage = base64_encode($userImage);
        }

        echo "<div class=\"media border p-3\">";

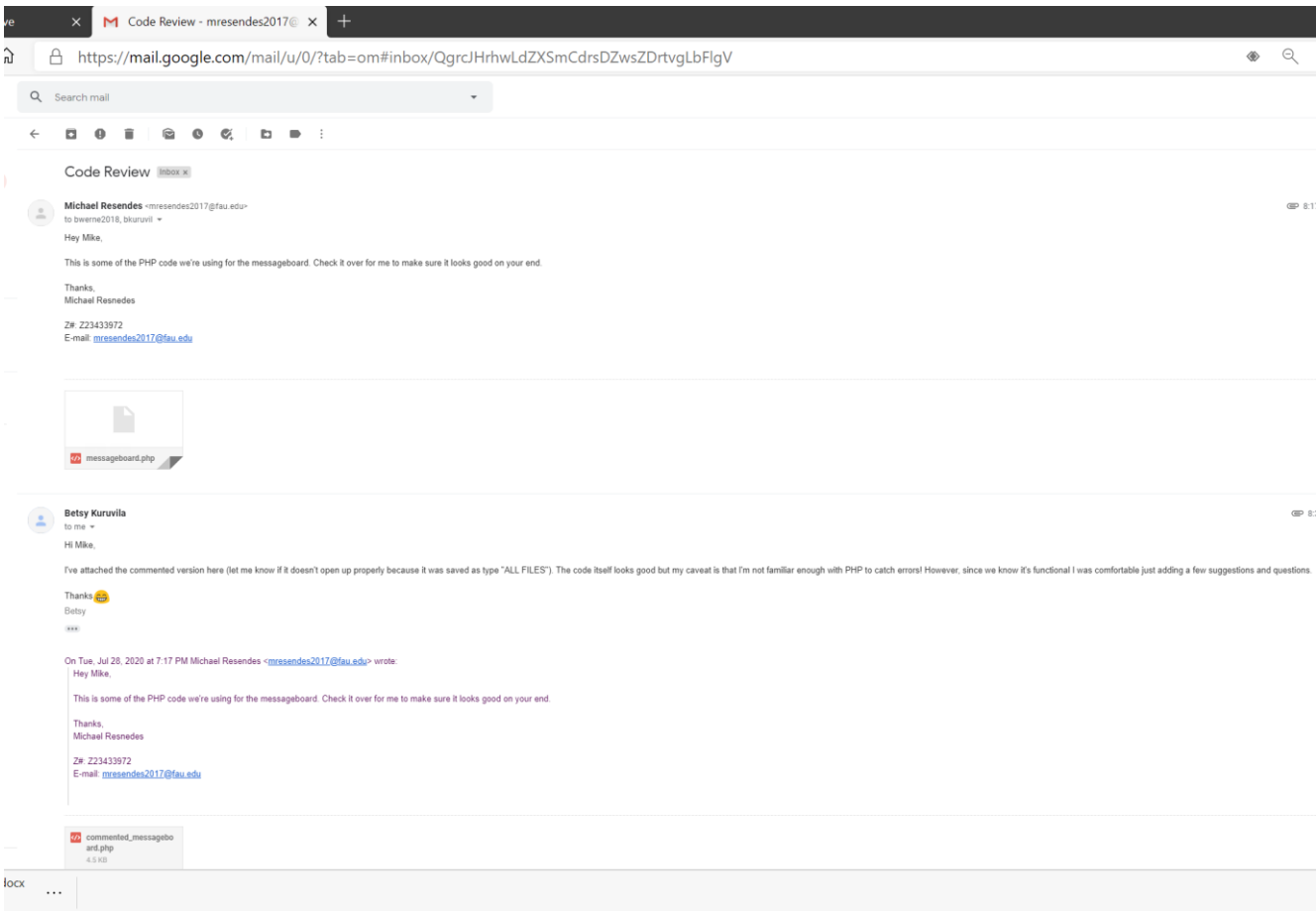
        if ($userImage != "")
        {
            echo "<img src=\"data:image/jpg;charset=utf8; base64, $userImage \" alt=\"\$name\"
class=\"mr-3 mt-3 rounded-circle\" style=\"width:60px; height:60px;\" />";
        }
        else
        {
            echo "<img src=\"images/blank.png\" alt=\"John Doe\" class=\"mr-3 mt-3 rounded-circle\"
style=\"width:60px; height:60px;\">";
        }

        echo "<div class=\"media-body\">";
        echo "<h4>$name</h4>";
        echo "<p>$message</p>";
        echo "</div>";
        echo "</div>";
    }
}

}
?>

```

Correspondence for the Peer Review



Self-Check: Best Practices for Security

FunkyTech

Major Asset Protection

The following is a list of the important assets that the Gather+ application will store for its user.

- **User's e-mail, first and last name:** These pieces of user information are required only for account creation and are not visible to any other users on Gather+.
- **User's password:** See Password Encryption Below
- **Community messageboard content:** Checks are placed in the messageboard's PHP scripts that allow only members of a Community to use and post on the messageboard hosted on that Community.

Password Encryption

User passwords in Gather+ are protected by a bcrypt hash function utilized by the site's PHP scripts. When a user creates their Gather+ account, they must first correctly enter their desired password into two separate input forms to ensure that the password is correctly stored in the database. If the two passwords match, then the password is hashed through a bcrypt hash function that generates a 60-character text string that is then stored in the user_accounts database. This hash ensures that a user's password is impossible to determine from just the text string stored in the SQL table. For login, a user will enter their username and password, from which the password entered by the user will be run through the hash function. If the function returns a string that matches the one stored in the database, the user will be logged-in to their account.

Confirm Input Validation

Input data validation utilized by Gather+ includes the cross checking of dates entered for date-of-births to ensure that they are valid, ensuring that password are at least 8 characters in length, and that messageboard content can only be submitted by members of that messageboard's Community.

Self-Check: Adherence to Original Non-Functional Specs

FunkyTech

Non-Functional Requirements

- The Gather+ website will allow up to 20 users posting on a single community board at a time.
STATUS: DONE
- The Gather+ website will be extremely easy to use and learning how to navigate the website should take less than 30 minutes on average.
STATUS: DONE
- For security, we will make use of a b-crypt hash function for a user's login information. It will hide users' personal account information from potential threats.
STATUS: DONE
- Expected load time for each page will be under three seconds, even if internet connection is below average.
STATUS: DONE
- All data will be stored in SQL tables hosted by the team's account on the LAMP server.
STATUS: DONE
- The browsers that the Gather+ website will be tested on are Firefox, Chrome, Microsoft EDGE, and Safari.
STATUS: DONE
- The time it will take to refresh the page after a crash will be under five seconds.
STATUS: DONE

Beta Launch Demonstration Video Information

FunkyTech

- YouTube Link: <https://youtu.be/mOiqq3wGQNc>
- Gather+ Application Link:
https://lamp.cse.fau.edu/~cen4010s2020_g07/GatherPLUS
- Gather+ GitHub Repository:
<https://github.com/CEN4010-S2020-G07/CEN4010-S2020-G07>