SW Engineering CSC 648/848 Summer 2022

# YourSports

Dynamic Debugger (Team 02)

—


## Team & Backend Lead

Kshitiz Sareen: ksareen@sfsu.edu

## Frontend

Lead Kevin Islas

Developer Shamar Ireland

## Functional

Lead Sabur Saigani

Assistant Jonathan Ip

## Database Administrator

Wenye Guo

## Github Master

Mathew O Abiola


## Milestone 4

| Milestone | Date |
|---|---|
| **M4V1** | July 27th, 2022 |
| **M3V1** | July 18th, 2022 |
| **M2V2** | July 14th, 2022 |
| **M2V1** | July 7th, 2022 |
| **M1V2** | June 30th, 2022 |
| **M1V1** | June 9th, 2022 |

# Table of Contents

# Product Summary

## Name

**YourSports**

## Major Committed Functions

### Regular user

1. Regular users shall be able to create an account.
2. Regular users shall be able to sign up as a creator account.
3. Regular users shall be able to access the homepage.
4. Regular users shall be able to view the live scores of a game.
5. Regular users shall be able to view the date of a game.
6. Regular users shall be able to view updated statistics of players.
7. Regular users shall be able to search for articles through a search bar.
8. Regular users shall be able to filter articles they would like to read by different types of sports.
9. Regular users shall be able to search articles by heading of the article.
10. Regular users shall be able to view the article heading in the search results for articles.
11. Regular users shall be able to view the article author in the search results for articles.
12. Regular users shall be able to view the article postdate in the search results for articles.
13. Regular users shall be able to view the article image as a thumbnail in the search results for articles.
14. Regular users shall be able to search games by the teams playing in the search bar.
15. Regular users shall be able to filter games by the sport type.
16. Regular users shall be able to filter games by their date.
17. Regular users shall be able to view the game score in the search results for games.
18. Regular users shall be able to view the game date in the search results for games.
19. Regular users shall be able to view the game location in the search results for games.

20. Regular users shall be able to search for player statistics according to the player name in the search bar.
21. Regular users shall be able to filter player statistics by the sport type.

### Registered user

22. Registered users shall have the same privileges as regular users.
23. Registered users shall be able to log in.
24. Registered users shall be able to log out.
25. Registered users shall be able to reset their password.
26. Registered users shall be able to delete their account.
27. Registered users shall be able to post in the discussion forum of the game.
28. Registered users shall be able to delete their posts in the discussion forum.

### Basic Account

29. Accounts shall be able to be deleted by users.
30. Accounts shall contain the name of the users.
31. Accounts shall contain the username of the users.
32. Accounts shall contain the email of the users.
33. Accounts shall contain the password of the users.

### Creator Account

34. Creators shall have the same privileges as registered users.
35. Creators shall be able to upload articles to the website.
36. Creators shall be able to view Dashboard.

### Admin Account

37. Admin shall have all the privileges.
38. Admin shall be able to create discussion forums for upcoming games.

### Articles

39. Articles shall be posted by creator users.
40. Articles shall be able to be filtered by sports type.
41. Articles shall be able to be filtered by keywords specified by the user.
42. Articles shall contain a heading.
43. Articles shall contain a subheading.
44. Articles shall contain an introduction.
45. Articles shall contain contents.
46. Articles shall contain a conclusion.

47. Articles shall contain an image.

### Games

48. Games shall be filtered by date.
49. Games shall be filtered by sport type.
50. Games shall be filtered by keywords specified by users in the search bar.
51. Games shall have an individual discussion forum.
52. Games shall be able to show live scores of corresponding teams.
53. Games shall be able to show the date of the game.
54. Games shall be able to show the location of the game.

### Discussion Forum

55. Discussion forum shall be able to be viewed by all users.
56. Discussion forum shall be able to show posts from registered users.

### Posts

57. Posts in the discussion forum shall be posted by registered users.
58. Posts in the discussion forum shall be able to be deleted by the user who has created the post.

### Dashboard

59. Dashboard shall be able to show the number of views of the article.
60. Dashboard shall be able to show the heading of the article.
61. Dashboard shall be able to show the posted date of the article.

### Backend Service

62. The backend service shall be able to validate the email for duplicates when users try to sign up.
63. The backend service shall be able to check for duplicate accounts when users try to sign up.
64. The backend service shall be able to create an account in the database.
65. The backend service shall be able to store posts by a user in the discussion forums.
66. The backend service shall be able to update player statistics in the database.
67. The backend service shall be able to filter articles specified by the user.
68. The backend service shall be able to filter games specified by the user.
69. The backend service shall be able to filter player statistics specified by the user.

### Website

70. The website shall be able to check if the name is not empty when users try to sign up.
71. The website shall be able to check if the username is not empty when users try to sign up.
72. The website shall be able to check if the email is not empty when users try to sign up.
73. The website shall be able to check if the password is at least 8 characters before the users try to sign up.
74. The website shall be able to check if the password is a maximum of 20 characters before the users try to sign up.
75. The website shall be able to check if the terms of service have been accepted when users try to sign up.

## Unique Features

Our product will offer countless services to users who are interested in sports information and are sports fans. Users will be able to share their opinions with each other through news articles. Other users can read news articles, and they can also react to news articles through comments, likes, and dislikes from other users. Users who have uploaded their articles can also view different statistics like the number of likes, number of dislikes, and number of views that their article has received. These statistics will give the Author of the article, how their article is being perceived by other users of the product.

Our product also allows users to interact with other users who are watching the same sports game through the discussion forums provided for each game. Users can share their opinions on the game, and interact with other users who have the same interests. This will serve as a great way to build a community of users with similar interests through sports.

Finally, our product keeps up to date with player statistics, so that users can access the latest information on their favorite players at their fingertips.

## URL

http://34.136.124.189:3000

# Usability Test Plan

## Test Plan

### Purpose

The purpose of these tests is to test the usability of the superior feature of our product, where regular users can sign up using a creator account and upload their own articles, and how other users interact with those articles. A creator account has the privilege of uploading its own articles that contain sports information. Other users can search these articles, and users can read the articles, and based on their perception of the information given in the articles, they can leave comments, likes, and dislikes. The author who has signed up as a creator can view the comments the article has received, as well as check the number of views, likes, and dislikes their article has received. These statistics will give the author an idea of how other users are reacting to the information given in the article. The comments that other users leave will give detailed information to the author.

### Problem Statement and Objectives

The problem is to find out how users find the usability of uploading articles, reading, and reacting to articles. The objective to test the task list mentioned above is as follows:

1. How did regular users find the usability of signing up as a creator.
2. If a registered user who has a creator account wants to upload their article, how smooth was their experience.
3. How satisfied was the user when they searched for articles.
4. How satisfied was the user with how the information given in the article was presented and were they able to express their opinion through the controls given by the website.
5. Were registered users who have creator accounts, easily able to get detailed and easy-to-understand statistics on how other users are reacting to their articles.

### Task List

1. Regular users can sign up as a creator.
2. Registered users who have a creator account can upload their own articles.
3. Users can search articles.

4. Users can read, like, or dislike and add comments to the article.
5. Registered users who have a creator account, can view the different statistics of their article, like number of likes, number of dislikes, and number of views.

### User Profile

The profile of our user is as follows:

1. The user does not use technology regularly. They only use it when required, or when they access information.
2. The user uses a computer to stream through different websites that are commonly used like Google, Facebook, or any other social media website.
3. The user does not understand scroll through any other website, because they don't like to be overwhelmed with information.

All the users will start from the homepage of the website.

### Method (Test Design)

Each of the five functions in the task list will be tested for effectiveness, efficiency, and user satisfaction.

### Test Environment

All the users will test the product in the comfort of their own homes. Once they are done testing, they will send their feedback via email.

### Test Monitor Role

We won't have a test monitor however the team lead will regularly check with the intended users, on how much they have tested their app. The team leader expects the intended users to finish their testing and receive feedback within 48 hours of asking users to volunteer as intended users of the product.

### Evaluation Measures

Each of the five functions in the task list will be tested for effectiveness, efficiency, and user satisfaction.

The effectiveness will be measured with the following metrics:

1. How useful is the feature.
2. How easy is it to navigate.
3. Does it serve any purpose to the target audience.

The efficiency will be measured with the following metrics:

1. How much time did it take to complete the task.

2. How many pages did they have to go through to complete the task.
3. How helpful was the website when they tried to complete the task.

The satisfaction will be measured with the following metrics:

1. How the user felt after completing the task.
2. Did the user have any more expectations from the product.

## Task Description

Task 1

| Task | Description |
|---|---|
| Regular Users will sign up as a creator | A Registered user will try to sign up as a creator. |
| System Setup | Users will access the website on a laptop. |
| Starting Point | Users will be on the homepage |
| Intended Users | The intended users are regular users who have not signed up to the YourSports website. |
| The URL of the system that will be tested is | http://34.136.124.189:3000/SignUp |
| What is to be measured | 1) Did the user find errors while trying to sign up as a creator.<br>2) How much time did it take, and did they find the information asked during the signup useful.<br>3) How much time did it take to complete the entire task. |

Task 2

| Task | Description |
|---|---|
| Registered Creators can upload their own articles | Users will upload an article to the YourSports Website. |
| System Setup | Users will access the website on a laptop. |
| Starting Point | Users will be on the homepage |
| Intended Users | The intended users are registered users who have signed up as a creator. |
| The URL of the system that will be tested is | http://34.136.124.189:3000/UploadArticle |
| What is to be measured | 1) Did the user find errors while trying to upload an article. |

| | |
|---|---|
| | 2) Were they able to fill in all the information they wanted to convey, and were they able to upload the image of their choice.<br>3) How much time did it take to complete the entire task. |

Task 3

| Task | Description |
|---|---|
| Users can search articles | Users will search for an article. |
| System Setup | Users will access the website on a laptop. |
| Starting Point | Users will be on the homepage |
| Intended Users | The intended users are regular users who want to access sports information. |
| The URL of the system that will be tested is | http://34.136.124.189:3000/Home |
| What is to be measured | 1) Was the user able to perform the search in the right search bar.<br>2) Was the user able to use the filters provided easily.<br>3) Did the user know what to do if they got 0 search results.<br>4) Were the users able to access information to their liking and did they face any errors when performing the search.<br>5) How much time did it take to complete the entire task. |

Task 4

| Task | Description |
|---|---|
| Users can read, like, or dislike and add comments to the article. | The user will add a comment and like the article. |
| System Setup | Users will access the website on a laptop. |
| Starting Point | Users will be on the homepage |
| Intended Users | The intended users are registered users who want to read the information given in the article. |
| The URL of the system that will be tested is | http://34.136.124.189:3000/ArticleView |
| What is to be measured | 1) Was the user able to open an article from the search results. |

| | |
|---|---|
| | 2) Was the information given in the article is visually appealing.<br>3) Was the user able to find the like/dislike button easily.<br>4) Was the user able to find the comments section easily.<br>5) Was the user able to post a comment or like an article if they were not logged in.<br>6) Did the user face any errors when trying to like/dislike the article or add a comment.<br>7) Was the user satisfied with the visual structure of the information given in the article, the like and dislike button, and the comments section.<br>8) How much time did it take to complete the entire task. |

Task 5

| Task | Description |
|---|---|
| Registered Creators can view all the statistics of their article; such as, number of likes, dislikes, and views | Registered users will be able to see the statistics on the articles they have uploaded. |
| System Setup | Users will access the website on a laptop. |
| Starting Point | Users will be on the homepage |
| Intended Users | The intended users are registered users who have uploaded articles to the website and want to check the statistics. |
| The URL of the system that will be tested is | http://34.136.124.189:3000/Dashboard |
| What is to be measured | 1) Was the user able to find the Statistics of their uploaded articles easily.<br>2) Were the statistics visually appealing. And if the statistics are useful.<br>3) Did they face any errors when checking the statistics.<br>4) How much time did it take to complete the entire task. |

## Usability testing for effectiveness and efficiency

| Test/ Use case | % Completed | Errors | Comments | No. of Pages | Time taken to complete the task |
|---|---|---|---|---|---|
| sign up | 100 | Found no errors | I don't know what's the benefit of registering as a creator. I didn't choose that part. | 2 | 2 minutes |
| upload articles | 70 | Cannot change status to creator after signing up | Not allowed to change to a creator after I signed up as a regular user, I need to sign up a new account to upload articles. | 2 | 7 minutes |
| search articles | 90 | Articles cannot found by author | Doesn't support search articles by author name | 1 | 30 seconds |
| View/Review articles | 90 | Cannot edit posted comment | It is not easy to find that I can post comment until I scroll down and get to the bottom of the view article page | 2 | 2 minutes |
| View self article statistics | 100 | Found no errors | It is easy to get the statistics of my published articles | 2 | 1 minute |

## Usability testing for Satisfaction

### Questionnaire

Place a tick in the box, for every question in the questionnaire. For the column of Any extra feedback, add your feedback using text.

| Question | Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree | Any extra feedback |
|---|---|---|---|---|---|---|
| I found it easy to find the sign up page. | | | | | | |
| I found the information required for the sign-up was required. | | | | | | |
| I found it easy to select the option that requires me to sign up as a creator. | | | | | | |
| I found it easy to find the Upload Article Page. | | | | | | |
| I found it easy to fill in all the information. | | | | | | |
| I was able to select the image and it looks good with the whole article. | | | | | | |
| I was able to read the text typed in the search bar easily. | | | | | | |
| I found the filters convenient and useful when searching for relevant sports information. | | | | | | |
| I was able to navigate through the search results easily. | | | | | | |
| I think the like and dislike buttons were clearly visible. | | | | | | |
| I think the comments were clearly visible at first glance. | | | | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| I was able to read through the article easily, without straining my eyes. | | | | | | |
| I was easily able to find the statistics page. | | | | | | |
| I was able to see the statistics of the articles clearly, and they were presented in an organized manner and descriptive manner. | | | | | | |
| I found the statistics useful. | | | | | | |

# QA Test Plan

1. Search Results should be displayed within 5 seconds.

## Test Objectives

The objective of this test is to test how fast search results are retrieved from the database and displayed when a user performs a search on the website.

## Hardware and Software Setup

The hardware setup required for this test is a basic computer where the website will be run and the tests will be performed. In our case, we will be using a MacBook Pro 13-inch to run our website.

The Software setup will be the website with the base URL http://34.136.124.189:3000.

The URL where we will perform our tests is http://34.136.124.189:3000/Home.

## Feature to be tested

The feature to be tested is the efficiency and speed of our search results.

## QA Test Plan

| No. | Description | Test Input | Expected Output | Pass/Fail |
|---|---|---|---|---|
| 1 | Testing User will type “Steph Curry” in the search bar that is present on the homepage. | “Steph Curry” (type: String) | 3 search results, where each article heading contains “Steph Curry”. The results should be displayed within 5 seconds. | Pass |
| 2 | Testing User will type “Harden” in the search bar that is present on the homepage. | “Harden” (type: String) | 18 search results, where each article heading contains “Harden”. The results should be displayed within 5 seconds. | Pass |
| 3 | Testing User will type “ ” in the search bar that is present on the homepage. | “ ” (type: String) | 37 search results, where all articles are returned from the database, with the message saying that, since the search provided was empty, all articles were returned. The results should be displayed within 5 seconds. | Pass |

2. Text inputs should have a minimum of 1 character and a maximum of 250 characters, and the input should be trimmed before processing.

### Test Objectives

The objective of this test is to test if the input data provided by a user is validated before processing from the backend.

### Hardware and Software Setup

The hardware setup required for this test is a basic computer where the website will be run and the tests will be performed. In our case, we will be using a MacBook Pro 13-inch to run our website.

The Software setup will be the website with the base URL http://34.136.124.189:3000.

The URL where we will perform our tests is http://34.136.124.189:3000/ArticleView.

### Feature to be tested

The feature to be tested is the security and validation of data before processing.

### QA Test Plan

| No. | Description | Test Input | Expected Output | Pass/Fail |
|---|---|---|---|---|
| 1 | Testing User will type “Warriors win again” in the input text box of the comments section that is present on the article view page of an article. | “Warriors win again” (type: String) | A new comment will appear in the Comments section that has the content “Warriors win again”. | Pass |
| 2 | Testing User will type “ ” in the input text box of the comments section that is present in the article view page of an article. | “ ” (type: String) | No comment will be added to the Comments section. | Pass |
| 3 | Testing User will type “Warriors” 250 times with a space between each word in the input text box of the comments section that is present on the article view page of an article. | “Warriors” (type: String). The input will be typed 250 times, with a space in between each word. | No comment will be added to the Comments section.<br><br>An error message will pop up stating “Max number of characters is 250”. | Fail |

3. Articles uploaded by a creator user shall need to follow community guidelines provided by the website.

### Test Objectives

The objective of this test is to test if the input data provided by a user is validated and checked by the backend, to ensure compliance with community guidelines.

### Hardware and Software Setup

The hardware setup required for this test is a basic computer where the website will be run and the tests will be performed. In our case, we will be using a MacBook Pro 13-inch to run our website.

The Software setup will be the website with the base URL http://34.136.124.189:3000.

The URL where we will perform our tests is http://34.136.124.189:3000/UploadArticle.

### Feature to be tested

The feature to be tested is the security, validation, safety, and compliance of input data provided by a user.

### QA Test Plan

| No. | Description | Test Input | Expected Output | Pass/Fail |
|---|---|---|---|---|
| 1 | Testing User will type Sports Information in the Heading, subHeading, Introduction, Content, and Conclusion section. | Heading: “Test Heading” (type: String).<br><br>SubHeading: “Test SubHeading” (type: String).<br><br>Introduction: “Test Introduction” (type: String).<br><br>Content: “Test Content” (type: String).<br><br>Conclusion: “Test Conclusion” (type: String). | A new article will be uploaded. | Pass |
| 2 | Testing User will use profane language in the Heading, | Heading: “**** *** Heading” (type: String). | A new article will not be uploaded. | Fail |

| | | | | |
|---|---|---|---|---|
| | subHeading, Introduction, Content, and Conclusion section. | SubHeading: "**** *** SubHeading" (type: String).<br><br>Introduction: "**** *** Introduction" (type: String).<br><br>Content: "**** *** Content" (type: String).<br><br>Conclusion: "**** *** Conclusion" (type: String).<br><br>"**** ***" is a profane word. | A message will pop up, stating the information given in the article does not follow community guidelines. | |
| 3 | Testing User will use an NSFW image as a thumbnail for the article. | The input image is an NSFW image. | A new article will not be uploaded.<br><br>A message will pop up, stating that the information given in the article does not follow community guidelines. | Fail |

4. The website shall be resizable and compatible across different screen sizes and operating systems.

### Test Objectives

The objective of this test is to test if the website is resizable and compatible across different devices with different screen sizes.

### Hardware and Software Setup

The hardware setup required for this test is a basic computer or phone where the website will be run and the tests will be performed. We will be using a MacBook Pro 13-inch, an Iphone12 Pro, and an Alienware x14.

The Software setup will be the website with the base URL http://34.136.124.189:3000.

The URL where we will perform our tests is http://34.136.124.189:3000/Home.

### Feature to be tested

The feature to be tested is the compatibility and resizability of the website.

### QA Test Plan

| No. | Description | Test Input | Expected Output | Pass/Fail |
|---|---|---|---|---|
| 1 | Testing User will open the homepage, with some search results.<br><br>The page will be tested on a MacBookPro 13-inch screen that has a macOS operating system. | Screen: 13-inch Macbook Pro<br><br>Operating System: macOS Sierra<br><br>Page URL: http://34.136.124.189:3000/Home<br><br>Search Bar Input: “Steph Curry”<br><br>Number of Search Results: 3 | 3 search results are displayed which contain the heading, author, article post date, and the thumbnail of the article. | Pass |
| 2 | Testing User will open the homepage, with some search results.<br><br>The page will be tested on a 34- screen that has a Windows operating system. | Screen: 34-inch screen<br><br>Operating System: Windows 11<br><br>Page URL: http://34.136.124.189:3000/Home<br><br>Search Bar Input: “Steph Curry”<br><br>Number of Search Results: 3 | 3 search results are displayed which contain the heading, author, article post date, and the thumbnail of the article. | Pass |

| | | | | |
|---|---|---|---|---|
| 3 | Testing User will open the homepage, with some search results.<br><br>The page will be tested on a phone screen that has an IOS operating system. | Screen: iPhone 12 Pro, 2.82x5.78 inches.<br><br>Operating System: IOS 15.5<br><br>Page URL: http://34.136.124.189:3000/Home<br><br>Search Bar Input: “Steph Curry”<br><br>Number of Search Results: 3 | 3 search results are displayed which contain the heading, author, article post date, and the thumbnail of the article. | Fail |

5. The YourSports API shall not accept more than 10 requests per 10 seconds from the same IP address.

### Test Objectives

The objective of this test is to test if the YourSports API can protect itself from overloading of network traffic, and prevent starvation of the resources provided by the virtual machine from a single user.

### Hardware and Software Setup

The hardware setup required for this test is a basic computer where the website will be run and the tests will be performed. In our case, we will be using a MacBook Pro 13-inch to run our website.

The Software setup will be Postman. Postman will be used to test the API.

The URL where we will perform our tests is http://34.136.124.189:8080/api/searchnews/search.

### Feature to be tested

The feature to be tested is the security, availability, and rate-limiting feature of the YourSports API.

### QA Test Plan

| No. | Description | Test Input | Expected Output | Pass/Fail |
|---|---|---|---|---|
| 1 | Testing User will make 5 API calls to the YourSports API within 10 seconds from one device. | Testing Application: Postman<br><br>API URL: http://34.136.124.189:8080/api/searchnews/search<br><br>Request Type: POST<br><br>Number of calls: 5<br><br>Duration: 10 seconds<br><br>Number of devices: 1 | All API calls will succeed, and the requested results will be returned. | Pass |
| 2 | Testing User will make 14 API calls to the YourSports API within 10 seconds from two different devices. | Testing Application: Postman,<br><br>API URL: http://34.136.124.189:8080/api/searchnews/search | All API calls from device one will return the requested results. | Pass |

| | | | | |
|---|---|---|---|---|
| | Device one will make 7 API calls within 10 seconds.<br><br>Device two will make 7 API calls within 10 seconds. | Request Type: POST<br><br>Number of calls: 14<br><br>Duration: 10 seconds<br><br>Number of devices: 2 | All API calls from device two will return the requested results. | |
| 3 | Testing User will make 15 API calls to the YourSports API within 10 seconds from one device | Testing Application: Postman,<br><br>API URL: http://34.136.124.189:8080/api/searchnews/search<br><br>Request Type: POST<br><br>Number of calls: 15<br><br>Duration: 10 seconds<br><br>Number of devices: 1 | The API will return the requested results until the 10th API call.<br><br>On the 11th call, the API will return an error message stating “Too much traffic from the same IP address, please try again later”. | Fail |

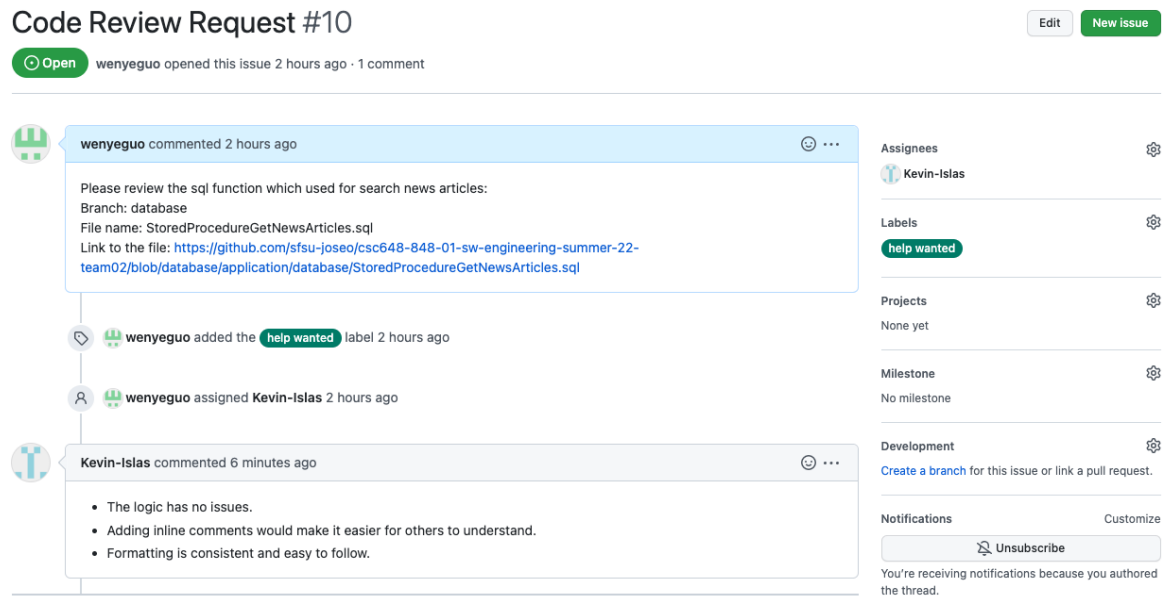
# Code Review

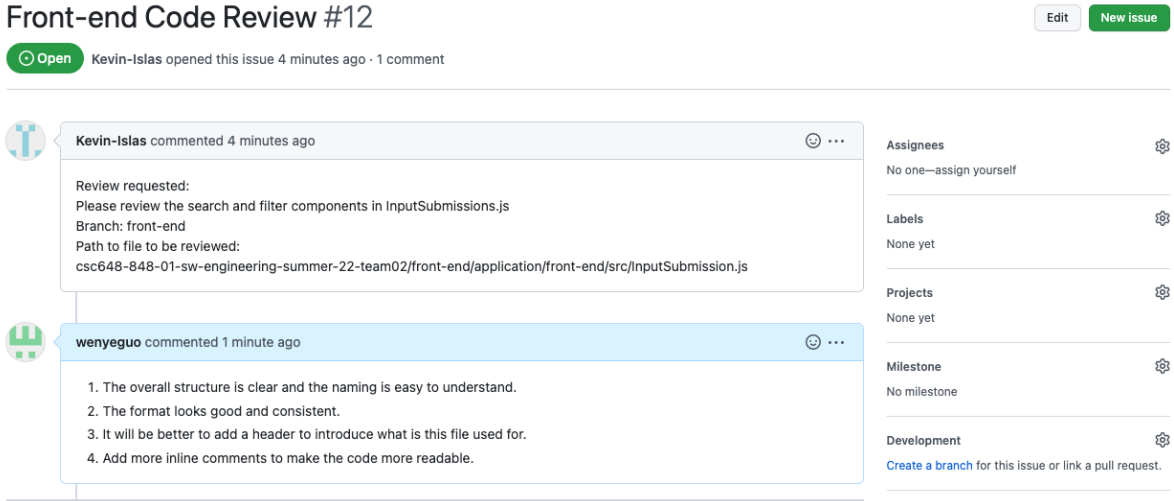
## Coding style

We separate the responsibilities for different functions in different files. We used longer function names to make it much easier to understand as later we debug or enhance the code. Proper alignment helps us easily to find which statements are controlled by which control structures. For the statements, we also group statements together and placing a blank line between groups.

## Code Peer Reviews

### Reviews from the same team



## Reviews from the different team

application > front-end > src > JS InputSubmission.js > ...

```
import React, { useState, useEffect } from "react";
import axios from "axios";
import "./Navbar.css";
import ArticleView from "./ArticleView";
import "./Footer";

/*Function receives user input and calls api to return articles*/
function InputSubmission() {
  const [data, setData] = useState("");
  const [InputText, setInputText] = useState("");
  const [date, setDate] = useState("");
  const [sport, setSport] = useState("");
  const [length, setLength] = useState("");
  const [searchApplied, setSearchApplied] = useState(false);

  function handlesearch() {
    if (InputText === "" && sport === "") {
      setSearchApplied(false);
    } else {
      setSearchApplied(true);
    }
    var config = {
      method: "post",
      url: "http://34.136.124.189:8080/api/searchnews/search",
      data: {
        sport: sport,
        searchText: InputText,
        date: date,
        length: length,
      },
    };
    axios(config)
      .then(function (response) {
        setData(response.data);
        console.log(response.data.sport);
        console.log(response.data.length);
        console.log(response.data);
      })
      .catch(function (error) {
        console.log(error);
      });
  }
```

## Team 06's review:

"At the beginning of the file, the comment "/*Function receives user input and calls api to return articles*/" was extremely helpful as it was a clear and concise comment that explained exactly what the following function does.

Another great comment is "/*When no filters applied all articles will be displayed */", for similar reasons to what was stated above, as it helps give the reader context on what the following code does.

While this is technically not required, our team believes that having a "header" comment with a brief description of the purpose of the whole file itself, will help readers have a better understanding of the following code as it will be put into context.

While our team is familiar with react code, others reading the code that have never used react before may be confused when it comes to react specific lines of code. For example the use-state hooks. Some clarifying comments would be helpful.

Overall indentation, organization and formatting of code is very well done.

The logic of the functions make sense and the way your team is displaying data to the front end checks out. Great job on these aspects of the code."

application > database > StoredProcedureGetNewsArticles.sql

```sql
Use YourSportsDB;

DELIMITER $$
CREATE PROCEDURE `GetNewsArticles`
(IN inPostdate VARCHAR(255),
IN inSport VARCHAR(255),
IN inSearchText VARCHAR(255))

BEGIN

SELECT Image_url,Heading,SubHeading,PostDate,Name as `Author`,Sport FROM
(SELECT Image_url,Heading,SubHeading,fk_Author_ID,PostDate,Sport FROM Article WHERE
(PostDate >= inPostdate OR inPostdate is NULL)
AND
(Sport = inSport OR inSport is NULL )
AND
(Heading LIKE CONCAT('%',inSearchText,'%') OR inSearchText is NULL)
) AS Article JOIN Account ON Account_ID = Article.fk_Author_ID ORDER BY PostDate DESC;

END$$
DELIMITER ;
```

### Team 06's review:

"The first thing our team noticed regarding the code is it does not have any in-line comments, nor an overall description to explain the purpose of this file. We recommended commenting on the code even though it is a .sql file, as the comments would help those who are unfamiliar with SQL code, such as a front-end developer, or someone learning the technology.

The indentation, spacing, formatting of this SQL code is excellent, and makes it easy to follow along for the reader.

We could not find any problems with the logic/process of the SQL code."

### Explanation to our choice of code

This is the most commonly used function. We believe that these codes are the most representative of our code style.

# Self-check on Best Practices for Security

## Major Assets under protection

Passwords - by storing encrypted passwords

Image assets - by setting the Google Cloud Storage bucket to public read-only

User data - by not collecting any data on the client and server side, and making sure the user's password is always encrypted.

## Encrypted Passwords in the DB

We encrypt passwords using the MD5 hash function that is provided in MySQL. An MD5 hash is created by taking a string of any length and encoding it into a 128-bit fingerprint. Encoding the same string using the MD5 algorithm will always result in the same 128-bit hash output. MD5 hashes are commonly used with smaller strings when storing passwords, credit card numbers, or other sensitive data in databases such as the popular MySQL.

Here is a screenshot, where we encrypt passwords provided by the user in the backend itself.

```
query = 'INSERT INTO RegisteredUser (name,username,email,password) VALUES ("'+name+'","'+username+'","'+email+'",MD5("'+password+'"));';
con.query(query, (error, results, fields) => {
    if (error) {
      res.json(error);
    }
    res.json("Account Created Succesfully");
  });
```

## Input Data Validation

Data Validation during registration

Name - It should contain at least 1 character that is not a space.

Username - It should contain at least 1 character that is not a space.

Email - It should contain at least 1 character that is not a space and it should follow the regular expression rules of an email

Password - It should be 8-20 characters long.

Terms of Service - It should be validated and checked if it is ticked, before registration.

We have converted all of our input data to lower case for safety and accurate results, however, that is not required since MySQL automatically handles the upper case and lower case checking when searching for data.

```
let name = req.body.name == null || req.body.name == "" ? null : req.body.name.trim().toLowerCase();
let username = req.body.username == null || req.body.username == "" ? null : req.body.username.trim().toLowerCase();
let email = req.body.email == null || req.body.email == "" ? null : req.body.email.trim().toLowerCase();
let password = req.body.password == null || req.body.password == "" ? null : req.body.password;
let isCreatorAccount = req.body.isCreatorAccount == null || req.body.isCreatorAccount == false ? null : true;
let termsOfServiceAgreed = req.body.termsOfServiceAgreed == null || req.body.termsOfServiceAgreed == false ? null : true;


if (termsOfServiceAgreed==true)
{
if (name!=null && username!=null && email!=null && password!=null)
{
  if (email.match(
    /^(([^<>()[\]\\.,;:\s@"]+(\.[^<>()[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/
  ))
  {
  if(password.length>=8 && password.length<=20)
  {

    query = 'INSERT INTO RegisteredUser (name,username,email,password) VALUES ("'+name+'","'+username+'","'+email+'",MD5("'+password+'"));';
    con.query(query, (error, results, fields) => {
        if (error) {
          res.json(error);
        }
        res.json("Account Created Succesfully");
      });
    }
    else
    {
      res.json("Please adhere to the password requirements")
    }
  }
  else
  {
    res.json("Please enter a valid email");
  }
}
else
{
    res.json("Please fill in all the details");
}
}
else
{
  res.json("Please accept terms of service");
}
```

## Data Validation during search

Input Search Text - It should contain at least 1 character that is not a space. If it does not contain at least 1 character that is not space, it will be set to NULL. If all the inputs provided to the search bar are NULL, all the data from the database will be returned.

We have converted all of our input data to lower case for safety and accurate results, however, that is not required since MySQL automatically handles the upper case and lower case checking when searching for data.

Here is an example

```
let sport = req.body.sport == "" || req.body.sport == null ? 'NULL' : '"'+req.body.sport.trim().toLowerCase()+'"';
let searchText = req.body.searchText == "" || req.body.searchText == null ? 'NULL' : '"'+req.body.searchText.trim().toLowerCase()+'"';
let query = 'CALL getNewsArticles('+date+','+sport+','+searchText+');';
```

# Adherence to Original Non-functional Specs

## System Requirements

1. The system shall be hosted on a Google Cloud Compute Engine instance of 1 cpu core, 1gb RAM and 10GB storage size. (DONE)
2. A MySQL relational database shall be hosted in Google Cloud to store the data. (DONE)
3. NodeJS and express shall be used to build the backend server. (DONE)
4. React shall be used to host the front-end server. (DONE)
5. The front-end service shall run on port 3000. (DONE)
6. The back-end service shall run on port 8080. (DONE)
7. The Google Cloud compute engine shall use an Ubuntu 18.04.6 LTS. (DONE)
8. The code repository shall be hosted in Github. (DONE)
9. The front-end and backend service shall accept traffic from all ip-addresses. (DONE)
10. The Virtual Machine hosted in Google Cloud shall allow both HTTP and HTTPS traffic. (DONE)
11. The website shall be resizable and compatible across different screen sizes and operating systems. (ON TRACK)

## Performance Requirements

12. An unregistered user shall receive an email within 30 seconds of account creation, stating that their account has been created. (ON TRACK)
13. If a registered user enters the wrong password, the system shall respond within 3 seconds of submitting their details for signup. (DONE)
14. Search results in the homepage when searching for articles should happen within 7 seconds. (DONE)
15. Comments in the Article view page should be added within 5 seconds. (ON TRACK)
16. Search results for games in the Games page should happen within 3 seconds. (DONE)
17. Posts in the discussion forum for a game should be added within 3 seconds. (DONE)
18. Search for player statistics should return results within 5 seconds. (DONE)
19. Article Statistics should be displayed on the website within 3 seconds. (DONE)

20. The homepage shall open within 10 seconds. (DONE)
21. Search Results should be displayed within 5 seconds. (DONE)
22. The website shall show updates in a game every 5 seconds. (ISSUE: We need to set a listener in the website that fetches live updates every 5 seconds. The team is not familiar with Pub/Sub systems that will be used to implement this requirement)

## Storage, Security and Environmental Requirements

23. The forum shall be deleted after 15 days of the game. (ON TRACK)
24. Registered users shall be allowed to delete their posts within 5 minutes of posting. (ON TRACK)
25. Unregistered Users shall be able to register if their email is unique. (DONE)
26. Registered users shall not be able to send more than 10 messages in 10 seconds in the discussion forum. (ON TRACK)
27. The same user shall not be able to post more than 10 comments in 10 seconds in the view articles page. (DONE)
28. Users shall be able to receive emails after registering. (ON TRACK)
29. Users shall be able to receive messages through their phone after registering. (ISSUE: We may not be able to do this, since we need to use a third party service that will send messages through SMS. Most of these services are not free, and have a very short free trial period)
30. The email of the user shall be a valid email in order to receive emails. (DONE)
31. The phone number of the user shall be a valid phone number in order to receive messages. (ISSUE: We cannot figure out a way to verify phone numbers, since we can only verify through SMS, and the services that provide SMS services are not free)
32. Registered users that have been red flagged more than 3 times, shall not be allowed to post in the discussion forum. (ON TRACK)
33. The statistics of the player will be updated every 24 hours. (ISSUE: We need to set a listener in the website that fetches live updates every 24 hours. The team is not familiar with Pub/Sub systems that will be used to implement this requirement)
34. Registered users shall be only allowed to delete their posts within 5 minutes of posting in the discussion forum. (DONE)
35. Registered users shall be only allowed to delete their comments within 5 minutes of posting in the article view page. (DONE)

36. The YourSports API shall not accept more than 10 requests per 10 seconds from the same IP address. (ON TRACK)

## Marketing and Legal Requirements

37. An unregistered user shall not be able to register if they are less than 13 years old. (ON TRACK)
38. Unregistered Users shall be able to accept the terms of service before signing up. (DONE)
39. Articles uploaded by a creator user, shall need to follow community guidelines provided by the website. (Issue: Very difficult to check if content uploaded by a user follows community guidelines. We will need to implement Natural Language technologies, and high end neural networks that can check if the information uploaded by the user is compliant with the community guidelines)

## Content

40. The password shall be hidden when typed when trying to sign up. (DONE)
41. The password shall be a minimum of 8 characters when trying to sign up. (DONE)
42. The username shall not exceed more than 30 characters, and it should not contain any spaces or special characters when trying to sign up. (ON TRACK)
43. The registered user shall be redirected to the homepage if their account has been created. (DONE)
44. Registered Users shall be prompted to log in if an account with the same email already exists. (ON TRACK)
45. Unregistered users shall have to fill in the first name with at least one character while signing up. (ON TRACK)
46. Unregistered users shall have to fill in the last name with at least one character while signing up. (ON TRACK)
47. Unregistered users shall have to fill in the email with at least one character while signing up. (DONE)
48. Unregistered users shall have to fill in the password with at least one character while signing up. (DONE)
49. If a Registered User has not put any filters in the news articles they are searching for, they shall receive all the news articles in the database. (DONE)
50. The email shall only contain the links to the news articles. (ON TRACK)

51. Articles that have been marked as read, shall not appear on the user's search results when going through articles. (ON TRACK)
52. Posts in the discussion forum shall have a maximum 250 character limit and a minimum 1 character limit. (ON TRACK)
53. The game shall contain links that redirect users to a live broadcast. (ON TRACK)
54. Comments in the Article view shall have at least one character and a maximum of 250 characters. (ON TRACK)
55. The website logo shall be on the top left of every page. (DONE)
56. Text inputs should have a minimum of 1 character and a maximum of 250 characters, and the input should be trimmed before processing. (ON TRACK)

## Privacy

57. The email, username and password shall be collected and only used for authentication. (DONE)
58. Registered users shall have their log in details stored. (DONE)

# List of Team Contributions

### Kshitiz Sareen (Team Lead, Back-end Lead)

Organized team meetings and set team agendas.

### Kevin Islas (Front-end Lead)

Did the code review for team 06, defined the coding style, and selected the files that were to be sent for code review.

### Shamar Ireland (Front-end Developer)

Decided the tasks that used for the usability tests.

### Wenye Guo (Database Administrator)

Decided the questions asked in the questionnaire. Assessed the current status of the non-functional requirements. Performed the internal code review. Added the rules and code screenshots that validated our input data.

### Sabur Saigani (Functional Lead)

Added the unique features in the product summary, and helped in adding the functional requirements.

### Jonathan Ip (Functional Assistant)

Contributed to the finalized list of P1 requirements that were to be implemented. Fully responsible in formatting and organizing the documentation's layout as well as checking the documentation's grammar.

### Mathew O Abiola (Github Master)

Added the assets we would be protecting and all the encryption method used for the passwords.