

SW Engineering CSC648-848-05 Spring 2024

Application Title & Name : Teamup

Team : 5

Student Name	SFSU Email	GitHub	Discord	Role
Juan Estrada	jestrada.zuluaga@sfsu.edu	jjestrada2	juan.josee	Team-lead
Areeb Abbasi	aabbasi@sfsu.edu	areeeeb	_xertz	Backend-lead
Cole Chiodo	cchiodo@sfsu.edu	colechiodo	colechiodo	Docs-editor
Jaycee Lorenzo	jlorenzo3@sfsu.edu	jclorenz0	__jaycee	Frontend-lead
Martin Pham	mpham8@sfsu.edu	mar10fam	marnoki	Github-master
Kotaro Iwanaga	kiwanaga@sfsu.edu	iamkotaaa	kotaro8448	Database-admin

Milestone 4

Version #	Submission Date
M4V2	05/22/2024
M4V1	05/16/2024
M3V2	05/15/2024
M3V1	04/25/2024
M2V2	04/18/2024
M2V1	04/04/2024
M1V2	03/21/2024
M1V1	03/01/2024

Content Table

Content Table	2
1) Product Summary	3
• Product: TeamUp	3
• Major Functions:	3
• Unique Features:	4
• Product URL: http://44.220.154.73/	5
2) Usability test plan	6
Test 1: Search Bar for Games	6
Test 2: Joining a game	9
Test 3: Create Game Form	12
Test 4: Give a review to a user	15
Test 5: Users shall be able to detach from a game.	18
3) QA test plan	21
4) Code Review:	26
a) Details of team's coding practices:	26
b) Our Code Reviews	27
c) External Code Review with Team 06:	28
External Code Review with Team 01:	31
5) Self-check on best practices for security	34
Major Assets Being Protected	34
Password Encryption	34
Input Data Validation	35
Search Bar Input Validation	37
6) Self-check: Adherence to original Non-functional specs	40
Reliability:	40
Response Time:	40
Hardware and Networking Requirements:	40
Usability Requirements:	40
Privacy:	41
Compatibility:	41
Media Content:	41
Performance:	41
Licensing and Legal:	41
Training and Support:	42
7. Detailed list of contributions	43

1) Product Summary

- **Product:** TeamUp

- **Major Functions:**

User:

1. Users shall be able to search for games based on sport type.
2. Users shall be able to search for games based on location.
3. Users shall be able to search for games based on time.
4. Users shall be able to join a team in a game.
5. Users shall be able to create new games.
6. Users shall be able to specify the sport for a game.
7. Users shall be able to specify the location of a game.
8. Users shall be able to specify the time of a game.
9. Users shall be able to specify the number of players needed for a game.
10. Users shall be allowed to update their account profile picture.
11. Users shall be allowed to update their account biography description.
12. Users shall have the option to manually log out from their accounts.
13. Users shall have the option to verify their accounts through email.
14. Users shall be able to recover their password through email.
15. Users shall be able to detach from a game.
16. Users shall be able to search all their joined games.
17. Users shall be able to search all their hosted games.

Game listings:

1. Game listings shall be able to provide a facility location map.
2. Game listings shall be able to provide the player's username.
3. Game listings shall be able to provide player's pictures.
4. Game listings shall be able to provide the player's link to their biography.
5. Game listings shall be able to provide facility game rules.
6. Game listings shall be able to provide the name of the organizer.
7. Game listings shall be able to provide the name of the game.
8. Game listings shall be able to provide the day and time of the game.
9. Game listings shall be able to provide the contact information of the organizer.
10. Game listings shall be able to provide a short description of the game.
11. Game listings shall be able to provide the empty slots of the game.

Game Location:

1. Game locations shall be able to show address and facility details.
2. Game locations shall be able to provide parking information.
3. Game locations shall be able to provide users with reviews.

Sport:

1. Users shall be able to choose sports preferences
2. Users shall be able to select their skill level in a specific sport.

Team:

1. Teams shall be able to be formed in a game listing.
2. Teams shall be restricted by a specific number of players in a game.

Profile:

1. Profile preferences shall be able to be edited by the user.
2. Profile pictures shall be able to be uploaded to the user's bio(JPG, 320x320px).
3. Profile email shall be able to be edited by the user.
4. Profile phone number shall be able to be edited by the user.
5. Profile gender shall be able to be edited by the user.
6. Profile birthday shall be able to be edited by the user.
7. Profile password shall be able to be edited by the user.

Review:

1. Users shall be able to rate other players.
2. Users shall be able to write reviews for other players.
3. Users shall be able to rate facilities for game locations.
4. Users shall be able to write reviews for game locations.

• Unique Features:

The unique feature of our app is the ability to use the app to find and join casual sports matches. Our competitors, such as SportsEngine and ZogSports are only for sport tournaments. Our app allows the ability to create and join casual, non-competitive sport matches. Other competitors, like Plei, only focus on a single sport, Soccer. This does not cover casual matches for other sports. And websites like Meetup allow any type of meetup, not just sports. Our app focuses just on sports so it has a better experience for sports.

Another unique feature is the ability to review game locations and users. You can review game locations on aspects such as facility quality or nearby amenities. You can also review other users based on their skill level or attitude. Some of our competitors do offer the ability to give accolades to other users, but not to give a star rating and written review.

- **Product URL:** <http://44.220.154.73/>

2) Usability test plan

Test 1: Search Bar for Games

What is being tested: The search bar's functionality, accuracy, and ease of use.

Why: The search bar is a critical feature that differentiates the app from competitors. It's likely the first function users will interact with, and its performance impacts their initial impression. Additionally, it employs a custom algorithm to match desired games, which needs to be validated for effectiveness in a real-world setting with substantial data.

System Setup:

- The test will be conducted on a personal computer.
- The application URL will be accessed via Google Chrome.

Starting Point:

- Open a new tab in Google Chrome.
- Enter the application URL in the search bar and navigate to the homepage.

Intended Users:

- User 1 (Ivy)
- User 2 (Ishan)
- User 3 (Ralph)
- User 4 (Maynard)

URL : <http://44.220.154.73/>

Objective: To evaluate the usability of the search bar for locating games nearby the user's location.

Test Description: The user must be able to use the search bar function to find a certain game

What is to be Measured:

- Effectiveness: Accuracy of search results in finding relevant games nearby.
- Efficiency: Time taken to locate and select a game from the search results.
- User Satisfaction: Users' subjective satisfaction with the search bar's performance and usability.

User	Task Completion (Y/N)	Errors made	Accuracy (1-5)	Time (Sec.)	Steps to Completion	Effectiveness (1-5)	Efficiency	Satisfaction
Ivy	Y	0	5	10.86s	3	5	5	4
Ralph	Y	0	5	15.45s	3	5	5	4
Ishan	Y	0	5	9.20s	3	5	5	3
Maynard	Y	0	5	8.69s	3	5	5	4

Feedback

Likert Scale Questionnaire:

- The search bar was easy to use.
 Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Strongly Agree
Ishan	Strongly Agree
Maynard	Strongly Agree

- The search results were accurate and relevant.
 Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Strongly Agree
Ishan	Strongly Agree
Maynard	Strongly Agree

- I am satisfied with the overall performance of the search bar.

Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Strongly Agree
Ishan	Strongly Agree
Maynard	Agree

Errors users had:

- No result popped up when searching 'pickleball' (See below for comments).

Comments:

- Developer Comments:
 - Search results solely depend on the user base as they are the source of the creation of games.
 - Also depends on supported sports by our application. In this test case, users were searching for pickleball games but could not find any.
 - Reason 1: No pickleball games posted.
 - Reason 2: Pickleball, as a sport selection, is not supported yet.
-

Test 2: Joining a game

What is being tested: The functionality of the "Join Game" button.

Why: This feature is crucial because after users find a game they are interested in, the ability to join it is the next critical step. If this functionality is broken or difficult to use, users are likely to leave the app, which would go against the app's promise on the landing page: "DISCOVER, JOIN, and HOST games for your favorite sports."

System Setup:

- The test will be conducted on a personal computer.
- The application URL will be accessed via Google Chrome.

Starting Point:

- Open a new tab in Google Chrome.
- Enter the application URL in the search bar and navigate to the homepage.
- Perform a search to find a game using the search bar and select a specific game.

Intended Users:

- User 1 (Ivy)
- User 2 (Ishan)
- User 3 (Ralph)
- User 4 (Maynard)

URL : <http://44.220.154.73/home>

Objective: To evaluate the usability of the button to join a team in a specific game.

Test Description: The user must be able to join a specific team in a game in an intuitive way.

What is to be Measured:

- Effectiveness: Success rate of users joining a game using the button.
- Efficiency: Time taken to complete the process of joining a game.
- User Satisfaction: Users' subjective satisfaction with the ease of joining a game.

User	Task Completion (Y/N)	Errors made	Accuracy (1-5)	Time (Sec.)	Steps to Completion	Effectiveness (1-5)	Efficiency	Satisfaction
Ivy	Y	0	4	12.5s	2	5	4	4
Ralph	Y	0	5	5.06s	2	5	3	3
Ishan	Y	0	5	11.05s	4	5	4	4
Maynard	Y	0	5	3.21s	2	5	5	5

Feedback

Likert Scale Questionnaire:

- The process of joining a game was easy and straightforward
 Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Agree
Ralph	Strongly Agree
Ishan	Strongly Agree
Maynard	Strongly Agree

- I am satisfied with the overall functionality of the "Join Game" button.
 Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Strongly Agree
Ishan	Strongly Agree
Maynard	Strongly Agree

- I was able to easily find, view, and manage the games that I have joined.

User	Answer

Ivy	Strongly Agree
Ralph	Agree
Ishan	Strongly Agree
Maynard	Strongly Agree

Errors users had:

- N/A

Comments:

- Ivy's Comments:
 - 'Join Game' button was not very obvious.
-

Test 3: Create Game Form

What is being tested: The functionality and user experience of the form used to create a new game.

Why: This function is crucial because, to maintain user engagement, we need users to take the initiative to create games. Ensuring that this process is smooth and user-friendly will help increase the number of games on the platform, thereby enhancing the app's popularity and competitive edge in the market.

System Setup:

- The test will be conducted on a personal computer.
- The application URL will be accessed via Google Chrome.

Starting Point:

- Open a new tab in Google Chrome.
- Enter the application URL in the search bar and navigate to the homepage.
- Navigate to the "Create Game" page from the homepage.

Intended Users:

- User 1 (Ivy)
- User 2 (Ishan)
- User 3 (Ralph)
- User 4 (Maynard)

URL : <http://44.220.154.73/home>

Objective: To evaluate the usability of the "Create Game" form.

Test Description: The user must be able to input title , sport, description date, time, location and number of players and then be redirected to the page in which they can look at their joined and hosted games.

What is to be Measured:

- Effectiveness: Success rate of users completing the form and creating a game.
- Efficiency: Time taken to complete the form and create a game.
- User Satisfaction: Users' subjective satisfaction with the ease of creating a game and being redirected to the page where they can view their joined and hosted games.

User	Task Completion (Y/N)	Errors made	Accuracy (1-5)	Time (Sec.)	Steps to Completion	Effectiveness (1-5)	Efficiency	Satisfaction
Ivy	Y	0	5	46.20s	9	5	4	5
Ralph	Y	1	4	44.93s	10	4	4	4
Ishan	Y	0	5	41.06s	9	5	4	4
Maynard	Y	3	4	1:33.75s	11	3	3	3

Feedback

Likert Scale Questionnaire:

- The form to create a game was easy to complete.
Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Strongly Agree
Ishan	Strongly Agree
Maynard	Neutral

- The information fields in the form were clear and easy to understand.
Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Agree
Ishan	Strongly Agree
Maynard	Neutral

- I am satisfied with the overall experience of creating a game and being redirected to the detailed game page.

Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Agree
Ishan	Strongly Agree
Maynard	Agree

Errors users had:

- No results for the desired sport, pickleball, available when choosing sport for the game.
- Not all location results were showing when typing in desired locations.
- Time component was difficult to use.

Comments:

- Ivy's Comments:
 - Make it more obvious that sports and locations can be searched by changing labels.
 - Ralph's Comments:
 - Need more sports and locations for games.
 - Make it more obvious that the location dropdown is scrollable.
 - Ishan's Comments:
 - Change time selection/input component to make it more usable.
 - Developer Comments:
 - Users had some trouble using the time component.
 - Need to support more locations.
 - Possibly implement a way for users to add a location if the desired location is not present.
-

Test 4: Give a review to a user

What is being tested: The functionality that allows users to give a review and rate other users.

Why: This function is important because it fosters interaction and engagement on the platform, which can lead to increased user retention and growth. Allowing users to give reviews and rate others enhances the app's credibility and creates a sense of active, genuine user participation, rather than relying on fake data to simulate activity.

System Setup:

- The test will be conducted on a personal computer.
- The application URL will be accessed via Google Chrome.

Starting Point:

- Open a new tab in Google Chrome.
- Enter the application URL in the search bar and navigate to the homepage.
- Navigate to a Game page from the homepage.
- Navigate to a user's profile page where the review and rating can be submitted.

Intended Users:

- User 1 (Ivy)
- User 2 (Ishan)
- User 3 (Ralph)
- User 4 (Maynard)

URL : <http://44.220.154.73/home>

Objective: To evaluate the usability of the review and rating feature for users.

Test Description: The user must be able to input the number of stars and a short comment and post the comment to the user's profile.

What is to be Measured:

- Effectiveness: Success rate of users submitting a review and rating.
- Efficiency: Time taken to complete and submit the review and rating.
- User Satisfaction: Users' subjective satisfaction with the ease of submitting a review and rating.

User	Task Completion (Y/N)	Errors made	Accuracy (1-5)	Time (Sec.)	Steps to Completion	Effectiveness (1-5)	Efficiency	Satisfaction
Ivy	Y	0	5	32.75s	4	5	4	5
Ralph	Y	1	4	56.43s	5	5	4	4
Ishan	Y	0	5	43.05s	4	5	4	4
Maynard	Y	0	5	51.86s	4	5	4	4

Feedback

Likert Scale Questionnaire:

- The process of giving a review and rating another user was easy and straightforward.
 Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Agree
Ralph	Strongly Agree
Ishan	Agree
Maynard	Agree

- The review and rating feature was clearly visible and accessible on the user profile page.
 Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Agree
Ishan	Agree
Maynard	Agree

- I am satisfied with the overall experience of giving a review and rating another user.

Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Strongly Agree
Ishan	Agree
Maynard	Strongly Agree

Errors users had:

- Did not select the number of stars before submitting the review.

Comments:

- Ralph's Comments:
 - Make the 'stars' component of creating a review more visible/obvious. It is hard to see because of the color combination of the stars and background.
 - Ishan's Comments:
 - Star rating component was not very obvious. Realized/noticed it while writing the review description.
-

Test 5: Users shall be able to detach from a game.

What is being tested: The functionality that allows users to give a review and rate game location.

Why: This function is important because it can facilitate future partnerships with businesses and provide a potential revenue stream by recommending certain game locations. User-generated reviews also enhance the credibility and usefulness of the app, encouraging more users to rely on it for finding suitable game locations.

System Setup:

- The test will be conducted on a personal computer.
- The application URL will be accessed via Google Chrome.

Starting Point:

- Open a new tab in Google Chrome.
- Enter the application URL in the search bar and navigate to the homepage.
- Navigate to a Game page from the homepage.
- Navigate to a game location profile page where the review and rating can be submitted.

Intended Users:

- User 1 (Ivy)
- User 2 (Ishan)
- User 3 (Ralph)
- User 4 (Maynard)

URL : <http://44.220.154.73/home>

Objective: To evaluate the usability of the review and rating feature for game locations.

Test Description: The user must be able to input number of stars and a short comment and post the command to game location.

What is to be Measured:

- Effectiveness: Success rate of users submitting a review and rating for game locations.
- Efficiency: Time taken to complete and submit the review and rating for game locations.
- User Satisfaction: Users' subjective satisfaction with the ease of submitting a review and rating for game locations.

User	Task Completion (Y/N)	Errors made	Accuracy (1-5)	Time (Sec.)	Steps to Completion	Effectiveness (1-5)	Efficiency	Satisfaction
Ivy	Y	0	5	8.55s	3	5	4	5
Ralph	Y	0	5	10.03s	3	5	4	5
Ishan	Y	0	5	9.92s	3	5	4	4
Maynard	N	1	-	12.43s	-	1	4	3

Feedback

Likert Scale Questionnaire:

- The process of leaving a joined game was easy and straightforward.
 Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Strongly Agree
Ishan	Strongly Agree
Maynard	Disagree

- I am satisfied with the location and visibility of the 'Leave Game' button.
 Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Strongly Agree
Ralph	Strongly Agree
Ishan	Strongly Agree
Maynard	-

- The option to leave a joined game was made obvious to me.
 Strongly Agree / Agree / Neutral / Disagree / Strongly Disagree

User	Answer
Ivy	Agree
Ralph	Strongly Agree
Ishan	Strongly Agree
Maynard	-

Errors users had:

- Critical error found:
 - When attempting to visit a game that the user has created/hosted, the server would crash.

Comments:

- Developer Comments:
 - Fix API that retrieves the games that the user has created/hosted.

3) QA test plan

Test objectives: To verify the system responds to user interactions within 2 seconds for browsing games.

HW and SW setup:

HW setup:

CPU speed: 2.42 Hz

Memory RAM capacity: 16 GB

Hard Drive Storage: 475 GB

The size of the screen: 14 inches

SWsetup:

Operating System: Windows 11 Pro

Browser: Chrome, Firefox, and Edge

Page: Home

Page:edit-preferences

Page:create-game

Feature to be tested: Response Time

QA test plan: please see the table below

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS/FAIL)
1	Clicking on a game in a home page	Click on a panel of a game, and a page of details need to appear within 2 seconds	Click on a game. It can be any game.	Page appears within 2 seconds	Edge:PASS Chrome: PASS Firefox: PASS
2	Choosing preferences	Edit your preferences and submit your change. It should submit the data within 2 seconds.	Edit your preference and click on Done.	It should submit the change within 2 seconds.	Edge: PASS Chrome: PASS Firefox: PASS
3	Create a new game	Create a new game, and click on create game. It should make a new game within 2 seconds.	Go to the create-game page and create a new game.	It should make a game within 2 seconds.	Edge:PASS Chrome:PASS Firefox:PASS

Test objectives: To verify that responsible design works properly in each browser
HW and SW setup:

HW setup:

CPU speed: 2.42 Hz

Memory RAM capacity: 16 GB

Hard Drive Storage: 475 GB

The size of the screen: 14 inches

SW setup:

Operating System: Windows 11 Pro

Browser: Chrome, Firefox, and Edge

Page: login

Page: signup

Page: create-game

Feature to be tested: Responsive design of the website (Compatibility)

QA test plan: please see the table below

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS/FAIL)
1	Testing the responsible design of the signup page	In each browser, set the screen size to the given test input.	Width: 375 pixels Height: 812 pixels	All input fields need to be in the form.	Edge: PASS Chrome: PASS Firefox: PASS
2	Testing the responsible design of the login page	In each browser, set the screen size to the given test input.	Width: 768 pixels Height: 1024 pixels	All input fields need to be in the form.	Edge: PASS Chrome: PASS Firefox: PASS
3	Testing the responsible design of the create-game page	In each browser, set the screen size to the given test input.	Width: 375 pixels Height: 812 pixels	All input fields need to be in the form.	Edge: PASS Chrome: PASS Firefox: PASS

Test objectives: To verify the system is available 99.9% of the time during peak usage hours.

HW and SW setup:

HW setup:

CPU speed: 2.42 Hz

Memory RAM capacity: 16 GB

Hard Drive Storage: 475 GB

The size of the screen: 14 inches

SW setup:

Operating System: Windows 11 Pro

Browser: Chrome, Firefox, and Edge

Page:

URL:

Feature to be tested: Reliability

QA test plan: please see the table below

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS/FAIL)
1	Clicking on the link of the website with other people	Click on the link of the website with other people. It should take you to the home page without any problems.	Click on the URL	Taking users to home page	Edge:PASS Chrome:PASS Firefox:PASS
2	Making many games	Making many games with other users to test it doesn't break the system	Try to make many games with other users at the same time	It should make games without problems	Edge:PASS Chrome:PASS Firefox:PASS
3	Joining a team and detach many times	Do cycles of joining a team and detaching from that team to make sure it doesn't break the system	Find a game you want to join, and try to join and detach many times with other users	Users should be able to join and leave the team without any problems. Usernames should be shown under teams.	Edge:PASS Chrome:PASS Firefox:PASS

Test objectives: To verify that user interactions and workflows shall be consistent across different sections of the application.

HW and SW setup:

HW setup:

CPU speed: 2.42 Hz

Memory RAM capacity: 16 GB

Hard Drive Storage: 475 GB

The size of the screen: 14 inches

SW setup:

Operating System: Windows 11 Pro

Browser: Chrome, Firefox, and Edge

Page: create-game

Feature to be tested: Usability Requirements

QA test plan: Please see the table below

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS/FAIL)
1	Invalid date for game creation	When you create a game, try to set a invalid date. It should give an error message.	Invalid date	Enter a valid date	Edge:PASS Chrome:PASS Firefox:PASS
2	Enter a invalid number of players for game creation	When you create a game, try to enter a invalid number of players such as minus	Minus number	Enter a valid number of players	Edge:PASS Chrome:PASS Firefox:PASS
3	Click on the survey	Try to click on a survey button at the bottom of the website. It should take you to a survey page	Clicking on the survey button	Taking you to a survey page	Edge:FAIL Chrome:FAIL Firefox:FAIL

Test objectives: To verify application is capable of handling concurrent user interactions without significant degradation in performance.

HW and SW setup:

HW setup:

CPU speed: 2.42 HZ

Memory RAM capacity: 16 GB

Hard Drive Storage: 475 GB

The size of the screen: 14 inches

SW setup:

Operating System: Windows 11 Pro

Browser: Chrome, Firefox, and Edge

Page:game

Page: create-game

Page:review

Feature to be tested: Performance

QA test plan: please see the table below

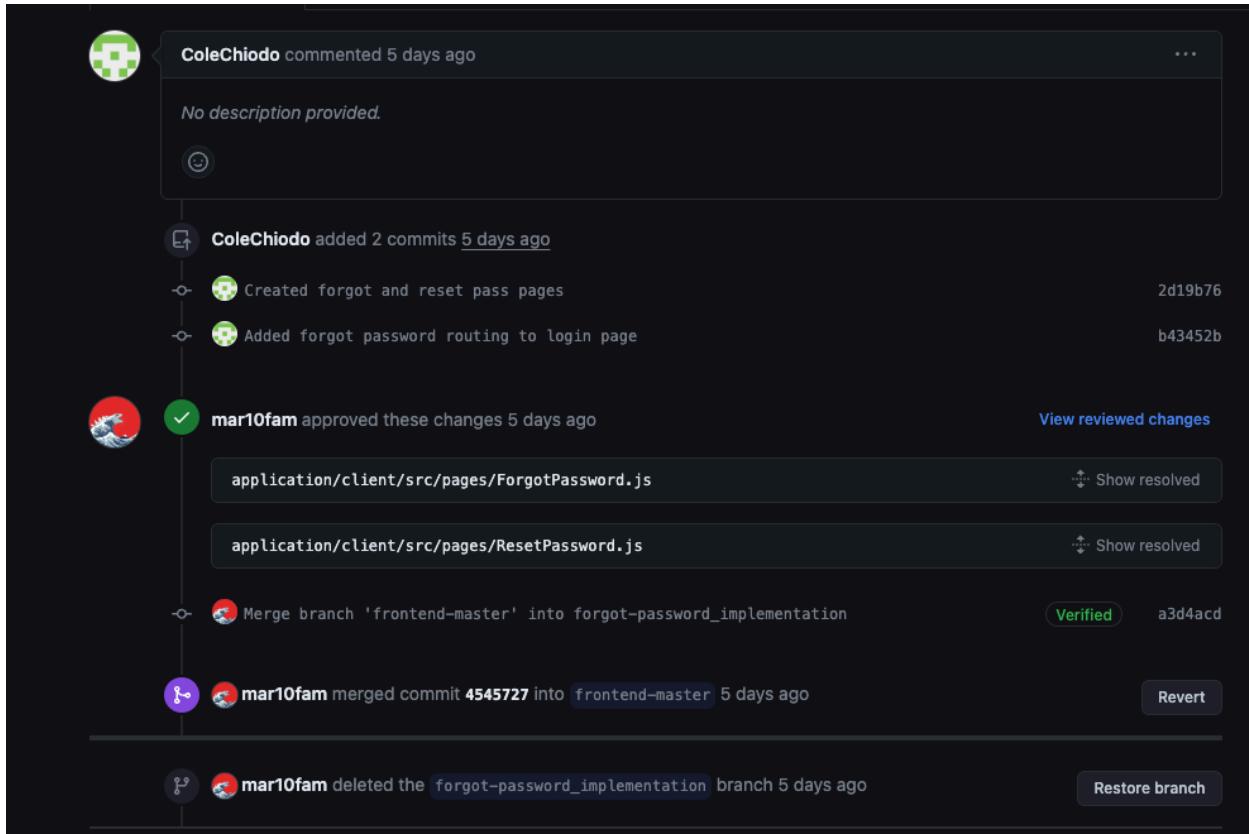
Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS/FAIL)
1	Joining a team with other users	Click on a game, and you can see two teams. Try to join one of them and do it concurrently with other users.	Try to join a team with other users	They should be able to join a game without any problems	Edge:PASS Chrome:PASS Firefox:PASS
2	Creating a game with other users	Create a game with other users at the same time.	Try to create a random game with other users.	It should create the games without any problems.	Edge:PASS Chrome:PASS Firefox:PASS
3	Writing a review with other users	Write a review of other users with other users at the same time.	Try to write a review of a user with other people	They should be able to write reviews without problems	Edge:PASS Chrome:PASS Firefox:PASS

4) Code Review:

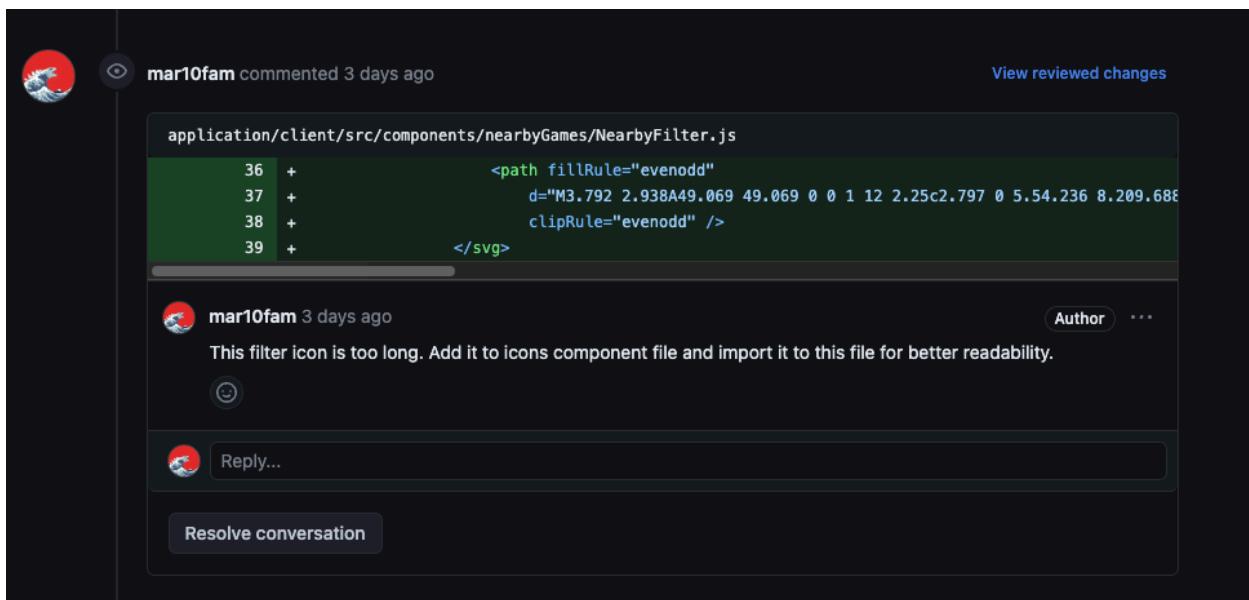
a) Details of team's coding practices:

1. **Use of Components:** Utilize components to modularize and encapsulate functionality. This approach promotes code reusability, readability, and maintainability. By breaking down complex UIs into smaller, manageable components, the code becomes more condensed and easier to understand. Additionally, it encourages a consistent structure across the application.
2. **Headers for Pages and Components:** All pages and components should contain headers that provide essential information such as contributors, a brief description of the purpose and the components used within them. This ensures better understanding of the codebase.
3. **Proper Indentation:** Maintain consistent and proper indentation throughout the codebase for enhanced readability. Clear indentation will help us grasp the code's structure and logic, making it easier to navigate and modify.
4. **Limit CSS Usage, Prefer Tailwind CSS:** Minimize the direct use of CSS and instead leverage Tailwind CSS for styling. Tailwind CSS offers a utility-first approach, allowing us to apply styling directly within HTML markup, resulting in more concise and maintainable code. This approach also reduces the need for writing custom CSS rules and helps achieve a consistent design system.
5. **Line Length Limitation:** Restrict lines of code to a maximum of 150 characters to prevent overly long lines that can negatively impact readability. Breaking lines at appropriate points enhances code clarity and makes it easier to view and understand code in various environments and screen sizes.
6. **Meaningful Function Names:** Choose descriptive and meaningful names for functions that accurately reflect their purpose and behavior. This practice improves code readability and comprehension, making it easier for us to understand the function's role within the application.
7. **Consistent Naming Conventions:** Functions will be written in camelCase (e.g., firstName). This will maintain consistency within the codebase to ensure uniformity and readability.
8. **Exported Pages Naming Convention:** Exported pages should follow PascalCase naming convention (e.g., FirstName) to distinguish them as standalone entities and adhere to standard naming conventions for React components.

b) Our Code Reviews



The screenshot shows a GitHub pull request interface. At the top, ColeChiodo comments "No description provided." Below this, ColeChiodo adds two commits: "Created forgot and reset pass pages" (commit 2d19b76) and "Added forgot password routing to login page" (commit b43452b). mar10fam approves these changes 5 days ago. The pull request includes code reviews for `application/client/src/pages/ForgotPassword.js` and `application/client/src/pages/ResetPassword.js`. mar10fam merges the branch 'frontend-master' into 'forgot-password_implementation' (verified commit a3d4acd). mar10fam then merges commit 4545727 into 'frontend-master' 5 days ago. Finally, mar10fam deletes the 'forgot-password_implementation' branch 5 days ago.



The screenshot shows a GitHub pull request interface. mar10fam comments 3 days ago on the file `application/client/src/components/nearbyGames/NearbyFilter.js`, specifically on lines 36-39. The code block shows:

```
36 +         <path fillRule="evenodd"  
37 +             d="M3.792 2.938A49.069 49.069 0 0 1 12 2.25c2.797 0 5.54.236 8.209.688  
38 +             clipRule="evenodd" />  
39 +         </svg>
```

mar10fam's comment: "This filter icon is too long. Add it to icons component file and import it to this file for better readability."

Buttons at the bottom include "Reply..." and "Resolve conversation".

c) External Code Review with Team 06:

5/13/24, 10:03 AM

Mail - Juan Jose Estrada Zuluaga - Outlook

External Code Review - Payment.jsx

Kirk Ruble <kruble@sfsu.edu>

Sat 2024-05-11 23:05

To:Juan Jose Estrada Zuluaga <jestrada.zuluaga@sfsu.edu>

Cc:Sean Nguyen <snguyen36@sfsu.edu>;Nixxy Dewalt <tdewalt@sfsu.edu>;Vincent Duong <vduong@sfsu.edu>;Gregory Luke Arruiza <garruiza@sfsu.edu>;Prasanna Wagle <pwagle@sfsu.edu>

Greetings Juan Estrada of Team 05,

I believe that you would serve as an effective code reviewer for a recent file we completed named Payment.jsx. Please see the attached zip file below for the code we would like you to review.

 [Payment.zip](#)

Thank you again for taking the time to review this code we look forward to hearing from you.

Best,

Kirk of Team 06.

5/13/24, 10:04 AM

Mail - Juan Jose Estrada Zuluaga - Outlook

Re: External Code Review - Payment.jsx

Juan Jose Estrada Zuluaga <jestradazuluaga@sfsu.edu>
Sun 2024-05-12 14:05
To:Kirk Ruble <kruble@sfsu.edu>

■ 1 attachments (5 KB)
game.controller.ts;

Hi Kirk,

I've reviewed the Payment component you shared. Here are some suggestions:

Functionality and Validation: You've implemented the form inputs and their formatting well. However, ensure all error handling is user-friendly. Specifically, the error alert for the email check could provide guidance on what's wrong instead of just saying the email isn't in the database.

State Management: You're managing state effectively with React hooks. Just a note: consider extracting logic like date calculations and form validations into separate functions or use custom hooks to make the component cleaner and easier to manage.

UX Improvements: When a payment is processed, consider disabling the button or indicating that something is happening to prevent duplicate submissions.

Error Handling: It's good to handle the axios error case, but also consider handling potential errors in your date functions or other logic that might unexpectedly fail.

I'm also attaching a file with some code I need help reviewing. Could you take a look when you have a chance?

Thanks, and looking forward to your thoughts on this!

Best,
Juan Estrada

From: Kirk Ruble <kruble@sfsu.edu>
Sent: Saturday, May 11, 2024 23:05
To: Juan Jose Estrada Zuluaga <jestradazuluaga@sfsu.edu>
Cc: Sean Nguyen <snguyen36@sfsu.edu>; Nixxy Dewalt <tdewalt@sfsu.edu>; Vincent Duong <vduong@sfsu.edu>; Gregory Luke Arruiza <garruiza@sfsu.edu>; Prasanna Wagle <pwagle@sfsu.edu>
Subject: External Code Review - Payment.jsx

Greetings Juan Estrada of Team 05,

I believe that you would serve as an effective code reviewer for a recent file we completed named Payment.jsx. Please see the attached zip file below for the code we would like you to review.

 [Payment.zip](#)

Thank you again for taking the time to review this code we look forward to hearing from you.

<https://outlook.office.com/mail/inbox/id/AAQkAGUyNTdhMDhlLTY3ZmItNGI0NC05NWY4LTdhYTdkMTQ3M2UyYQAQAEXi67YvHsdDI57hFyaxxM0%3D>

1/2

5/13/24, 10:05 AM

Mail - Juan Jose Estrada Zuluaga - Outlook

External Code Review - game.controller.ts

Kirk Ruble <kruble@sfsu.edu>

Sun 2024-05-12 15:15

To:Juan Jose Estrada Zuluaga <jestrada.zuluaga@sfsu.edu>

Hello, Juan

I took the time to review the Game Controller file you shared with me. Here are some observations I made:

Code Structure: Your import statements at the top of the file are minimal and neatly organized. Your functions are consistently spaced with no excess code between each function. These practices effectively made the readability of this file easy to view. One improvement I would add to your code structure would be consistent naming in all of your functions. I noticed in your "createGame" and "createGameWithTeams" that you are defining your variables with "snake_case", but throughout the whole file you are using "camelCase".

Code Comments: You have comments in a few of your functions that help with understanding the logic of certain lines of code within them, but there is a lack of consistency throughout the file with providing comments. I would suggest placing comments above each function header in this file to provide insight on how these functions are implemented throughout your project, seeing as I don't know or have access to how these functions interact with your web pages.

Error Handling: It is great that you are handing errors and returning status numbers to your web components. However, make sure that the error messages you are sending back are not ambiguous. In your "joinTeamHandler" function, you are returning "An unknown error occurred" as an error message which could create confusion for a developer or a user if they see this message.

Input Validation: Your import of the joi library for handling input validation is good practice to prevent unexpected behavior and security vulnerabilities. Just make sure that you put it to use as it is declared but never read within this file.

Best Regards,
Kirk of Team 06

External Code Review with Team 01:

5/13/24, 10:00 AM

Mail - Juan Jose Estrada Zuluaga - Outlook

Code Review

Lennart Richter <lrichter@sfsu.edu>

Sat 2024-05-11 8:36

To:Juan Jose Estrada Zuluaga <jestrada.zuluaga@sfsu.edu>

📎 1 attachments (2 KB)

likeController.ts;

Hi Juan,

Here is the file for our code review. Please provide any feedback you may have. I look forward to receiving your email.

Thanks,

Lennart (On behalf of Team 1)

5/13/24, 10:01 AM

Mail - Juan Jose Estrada Zuluaga - Outlook

Re: Code Review

Juan Jose Estrada Zuluaga <jestradazuluaga@sfsu.edu>

Sun 2024-05-12 13:54

To:Lennart Richter <lrichter@sfsu.edu>

 1 attachments (3 KB)

user.controller.ts;

Hi Lenni,

I checked out the LikeController code you sent over. Here are a few pointers:

Constructor: You don't need to assign the services inside the constructor since TypeScript does this automatically when you list them as private in the constructor itself.

Error Handling: The way you check if userId2 is a number is good. For errors, the message you send back could be clearer or more specific based on what went wrong.

Status Codes: Make sure you use CREATED only when something new is actually made. For other errors, 500 Internal Server Error might be more suitable than 422 Unprocessable Entity.

Business Logic: The part where you check if both users liked each other could perhaps go into the LikeService or MatchService. This would keep your controller simpler and just focused on handling the request and response.

Clean Up: If there's any code you're not using anymore, like // const userId1 = 1; it's a good idea to remove it to keep things neat.

I'm also attaching a file with some code I need help reviewing. Could you take a look when you have a chance?

Let me know if you have any questions or need more help with this!

Cheers,

Juan

From: Lennart Richter <lrichter@sfsu.edu>
Sent: Saturday, May 11, 2024 8:36
To: Juan Jose Estrada Zuluaga <jestradazuluaga@sfsu.edu>
Subject: Code Review

Hi Juan,

Here is the file for our code review. Please provide any feedback you may have. I look forward to receiving your email.

Thanks,
Lennart (On behalf of Team 1)

5/16/24, 8:39 AM

Mail - Juan Jose Estrada Zuluaga - Outlook

Code Review

Lennart Richter <lrichter@sfsu.edu>

Tue 2024-05-14 21:30

To:Juan Jose Estrada Zuluaga <jestrada.zuluaga@sfsu.edu>

Hi Juan,

We've reviewed your code and had a few pieces of feedback.

- there's a typo "const preferencess = await userService.getUserPreferences(req.params.userId);," there's an extra S in "preferences"
- i don't see validation but I'm assuming there is one somewhere else in their code.
- not a big thing but changing the order of parameters in the "try{ const user" to match the order of the "const createUser" or vice versa
- Everything else looks really good

Thanks, Lenni

5) Self-check on best practices for security

Major Assets Being Protected

1. User Data: Personal information such as email, name, date of birth, gender, username, and phone number.
2. Authentication Tokens: Refresh tokens and reset password tokens.
3. User Preferences: User-selected sports and skill levels.

Password Encryption

We confirm that passwords are encrypted before being stored in the database. The process involves using the bcryptjs library to hash passwords.

Code Snippet

```
export const encryptPassword = async (password: string) => {
  const encryptedPassword = await bcrypt.hash(password, 8);
  return encryptedPassword;
};
```

Input Data Validation

We validate various user inputs to ensure data integrity and security. Below is a list of validated inputs and the corresponding code:

User Validation

- Email: Must be a valid email format.
- Password: Custom validation to ensure complexity.
- Name: Required string.
- Role: Must be either USER or ADMIN.
- Date of Birth: Must be a valid date.
- Gender: Required string.
- Username: Required string.
- Phone Number: Required string.

Code Snippet

```
const createUser = {
  body: Joi.object().keys({
    email: Joi.string().required().email(),
    password: Joi.string().required().custom(password),
    name: Joi.string().required(),
    role: Joi.string().required().valid(Role.USER, Role.ADMIN),
    // Include new fields in validation schema
    dob: Joi.date().required(),
    gender: Joi.string().required(),
    username: Joi.string().required(),
    phone_number: Joi.string().required(),
  }),
}
```

Authentication Validation

- Email: Must be a valid email format.
- Password: Custom validation to ensure complexity.
- Refresh Token: Required string.
- Reset Password Token: Required string.

Code Snippet

```
const register = {
  body: Joi.object().keys({
    email: Joi.string().required().email(),
    password: Joi.string().required().custom(password),
    dob: Joi.date().required(),
    gender: Joi.string().required(),
    username: Joi.string().required(),
    name: Joi.string().required(),
    role:Joi.string().optional(),
    phone_number: Joi.string().required(),
  })
};

const login = {
  body: Joi.object().keys({
    email: Joi.string().required(),
    password: Joi.string().required()
  })
};
```

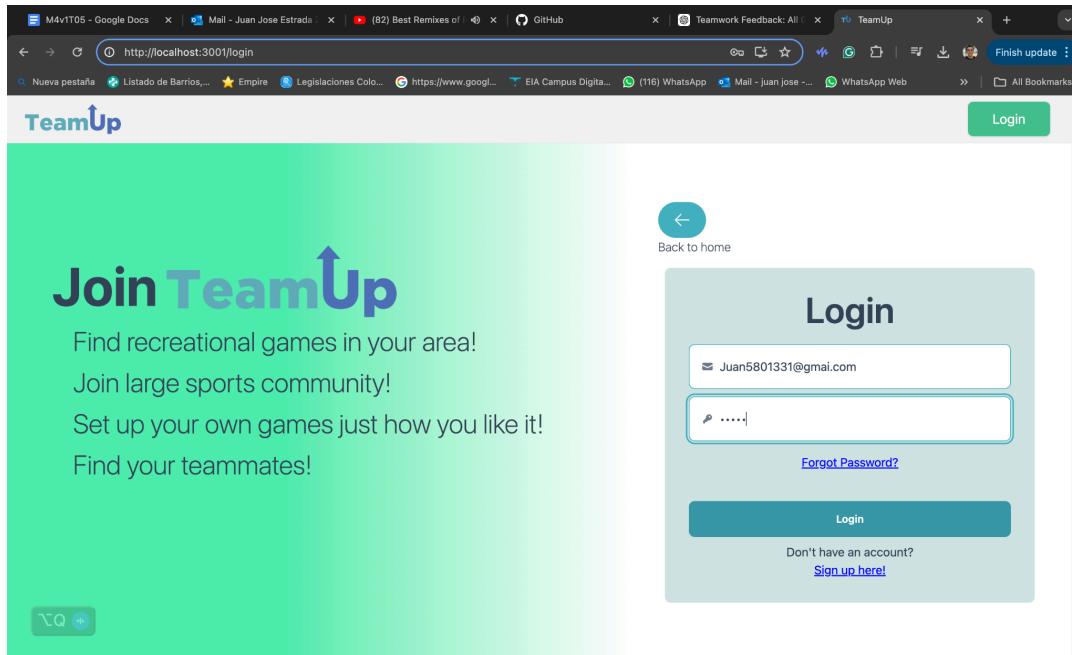
Search Bar Input Validation

We validate the search bar input to ensure it is a string and prevent potential security issues.

Example Code

```
const onSearch = async (searchTerm) => {
  try {
    const searchResults = [];
    for (const sport of selectedSports) {
      const response = await fetch(`#${url}/game/search`, {
        method: 'POST',
        body: JSON.stringify({ sport, gameName: searchTerm }),
        headers: {
          'Content-Type': 'application/json'
        }
      });
      if (!response.ok) {
        throw new Error('Failed to search games');
      }
      const searchData = await response.json();
      searchResults.push(...searchData);
    }
    setGames(searchResults);
  } catch (error) {
    console.error('Error searching games: ', error);
  }
};
```

User logins:



	id	email	name	imageUri	password	role	isEmailVerified	createdAt	updatedAt
1	sample4@gmail.com	John Doe	NULL	\$2a\$08\$UJWHnQVwSS3aHgP811Xv5DDn0zZnheFlAWBxjZegEwY-	ADMIN	0	2024-05-04 03:58:30.411	2024-05-08 04:30:45.080	2024-05-08 04:30:45.080
12	sample5@gmail.com	John Doe	NULL	\$2a\$08\$XbxDU1085DnWtb3DpOyREKan6LpFGNG31HzJFfR5kE	ADMIN	0	2024-05-08 04:30:45.080	2024-05-08 04:30:45.080	2024-05-08 04:30:45.080
21	juan001331@gmail.com	Juan Estrada	NULL	\$2a\$08\$Q52xV46Lc9XKC6SwQ0v6tOyyjeEU0sRk1Zm4ge0SPrQuaq7O	USER	0	2024-05-10 02:30:14.638	2024-05-10 02:30:14.638	2024-05-10 02:30:14.638

The screenshot shows the TeamUp application interface. At the top, there is a navigation bar with various tabs and links. Below the navigation bar, the main title "TeamUp" is displayed. A search bar with the placeholder "Search for games" is centered above a list of five game cards. Each card contains the game name, date, and time, along with fields for location, number of players, and description.

Game	Date	Description
Football Match	3/26/2024 @ 07:00 AM	Location: Number of players: 10
Football	1/1/2024 @ 06:00 AM	Location: Number of players: 9
Lets play football	2/21/2024 @ 06:00 AM	Location: Number of players: 14
Football friday	2/21/2024 @ 06:00 AM	Location: Number of players: 16
Football time	4/2/2024 @ 07:00 AM	Location: Number of players: 1

Below the game cards, there is a section titled "Games Near You" with a search bar and a refresh button.

User is then given token for Session:

The screenshot shows a database management tool interface with a query results grid. The query being run is "SELECT * FROM my_database.Token;". The results grid displays a table with columns: id, token, type, expires, blacklisted, createdAt, and userId. The data in the grid shows multiple rows of tokens, each with a unique ID, token value, type (e.g., ACCESS, REFRESH, RESET_PASSWORD), expiration timestamp, and user ID.

id	token	type	expires	blacklisted	createdAt	userId
1	asodjf234	ACCESS	2024-03-26 08:00:00.000	0	2024-03-26 07:00:00.000	1
2	kjowG44	REFRESH	2024-03-26 09:00:00.000	0	2024-03-26 07:00:00.000	2
3	kdjhg32	RESET_PASSWORD	2024-03-26 10:00:00.000	0	2024-03-26 07:00:00.000	3
4	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-03-26 03:58:20.430	0	2024-04-03 03:56:30.433	4
5	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-03-03 04:39:45.195	0	2024-04-04 04:39:45.197	4
6	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-07 03:28:30.705	0	2024-05-08 03:26:30.711	4
7	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-07 04:30:45.140	0	2024-05-08 04:30:45.144	12
8	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-07 15:33:38.070	0	2024-05-08 15:33:38.076	4
9	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-09 02:12:49.321	0	2024-05-10 02:12:49.329	4
12	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-09 02:12:49.321	0	2024-05-10 02:12:49.329	4
13	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-05-10 02:29:14.916	0	2024-05-10 02:29:14.916	21
15	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-05-10 03:51:17.075	0	2024-05-10 03:51:17.075	21
18	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-09 03:47:27.558	0	2024-06-10 03:47:27.558	4
19	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	VERIFY_EMAIL	2024-05-10 03:58:29.783	0	2024-05-10 03:48:29.785	4
20	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-09 03:49:27.770	0	2024-05-10 03:49:27.791	21
21	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	VERIFY_EMAIL	2024-05-10 04:00:16.305	0	2024-05-10 03:50:16.305	21
22	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-12 20:52:36.113	0	2024-05-13 20:52:36.121	4
26	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-15 15:32:59.901	0	2024-06-15 15:32:59.904	21
27	ey.hhbGoQJuUz1NisInR5cClBikpXVCJ.B ey.Jzd	REFRESH	2024-06-15 15:35:40.168	0	2024-06-15 15:35:40.173	4

At the bottom of the interface, there are buttons for Action Output, Time, Action, Response, Duration / Fetch Time, and a status message "Query Completed".

6) Self-check: Adherence to original Non-functional specs

Reliability:

- The system shall be available 99.9% of the time during peak usage hours. DONE
- The application shall be resilient to server failures and able to recover gracefully without data loss. DONE

Response Time:

- The system shall respond to user interactions within 2 seconds for browsing games and tournaments. DONE
- User actions such as joining a game or registering for a tournament shall complete within 5 seconds. ON TRACK

Hardware and Networking Requirements:

- The application shall be hosted on servers with sufficient processing power and memory to handle concurrent user requests. DONE
- Network bandwidth shall be adequate to support simultaneous interactions from multiple users. DONE

Usability Requirements:

- The user interface shall be intuitive and easy to navigate for users with basic computer literacy. DONE
- User interactions and workflows shall be consistent across different sections of the application. DONE

Privacy:

- User data collected shall include name, email address, and sports preferences. DONE
- Data collected shall only be used for facilitating game and tournament management and shall not be shared with third parties without user consent. DONE

Compatibility:

- The application shall be compatible with major web browsers such as Chrome, Firefox, Safari, and Edge. DONE
- Responsive design principles shall be implemented to ensure optimal viewing and usability across desktop and mobile devices.DONE

Media Content:

- Images and media files uploaded by users shall adhere to specified formats and size limits to ensure efficient storage and retrieval. DONE
- Accepted file formats and size limits shall be communicated to users during the upload process. ON TRACK

Performance:

- The application shall be capable of handling concurrent user interactions without significant degradation in performance. DONE
- Database queries and server-side operations shall be optimized to minimize latency and ensure smooth user experience.DONE

Licensing and Legal:

- The application shall comply with relevant laws and regulations governing data privacy, user consent, and intellectual property rights. DONE
- Proper licensing for third-party libraries and software components used in the development shall be obtained and documented. ON TRACK

Training and Support:

- Users with a high-school diploma, after 1 hour of training, shall be able to navigate and use core features of the application. DONE
- Support documentation and resources shall be provided to assist users in troubleshooting common issues and accessing help when needed.ON TRACK

7. Detailed list of contributions

Juan Estrada	9	<ul style="list-style-type: none">• Self-check: Adherence to original Non-functional specs• API to get user bio, profile image, name, username, reviews and achievements• POST API detached from a team• POST API GameLocation Review• GET review API by username• POST API review user by username• Add bio attribute for user entity• API to post a bio of the user
Kotaro Iwanaga	9.5	<ul style="list-style-type: none">• QA test plan• API to post bio is implemented.• GET API GameLocation list• GET API team1 & team2 by game ID• Create dummy data 30 locations
Cole Chiodo	9	<ul style="list-style-type: none">• Frontend API implementation (Get user by ID profile page)• Product Summary• Frontend API account email verification.• Frontend API implementation (Forgot password & Reset password)

Martin Pham	9	<ul style="list-style-type: none"> • Implement search filter by day and time • Code Review • Implement Search Nearby games • Implement adding, updating, removing, retrieving preferences. • Implement geolocation to track user latitude and longitude
Jaycee Lorenzo	9	<ul style="list-style-type: none"> • Usability test plan • Frontend API implementation Logout • Detach API frontend implementation • Implement UX/UI corrections from the last milestone. • Create a Join Game component and implement it with its API • Create User profile page • Implement POST and GET API user review
Areeb abbasi	9	<ul style="list-style-type: none"> • Implement s3 bucket to store user profile & game location images • Function to send email notification • Document deployment and deploy app in one EC2 instance • Self-check on best practices for security