

Trivia Tree

Team Mahogany

Daniel Laszczych, Mathew Li-Ah-Kim, Timothy Hsu, Tamzid Chowdhury

August 30th, 2021 - December 13th, 2021

Table of Contents

Table of Contents	2
1. System Analysis	
1.1 Executive Summary	4
1.2 Objectives	4
1.3 Constraints	4
1.4 Actors	4
1.5 Functional Requirements	5
1.6 Non-Functional Requirements	5
1.7 Technology	6
1.8 Training Requirements	6
1.9 Action Plan	6
2. Use Case Model	
2.1 Use Case Listing	7
2.2 Use Case Descriptions	11
3. UI Model	
3.1 User Interface View Listing	96
3.2 User Interface Mockup Diagrams	98
4. Domain Model	
4.1 Similar Problems	111
4.2 Technology Sets	111
4.3 Architecture	112
4.4 Dataflow	116
4.5 Training Verification	117
5. Data Model	
5.1 Logical Design	119
5.2 Data Structures	122
5.3 Local Storage	122
5.4 Data Dictionary	123
6. Software Model	
6.1 Front End Components	130
6.2 Routing	130
6.3 Backend API	131
7. Progress Reviews	
7.1 Deployment Details	132
7.2 Build #1 (Project Setup and Deployment)	133
7.3 Build #2	133
7.4 Build #3	134

7.5 Build #4	134
7.6 Build #5	135
7.7 Build #6	135
8. Post Development and Presentation	
8.1 Post Mortem	136
Appendix A: Meeting Minutes	136
Appendix B: Presentation Slides	138
Appendix C: Final Presentation Slides	138
Appendix D: Team Member Contributions	138

1. System Analysis

1.1 Executive Summary

Trivia Tree provides a service for people to test their knowledge on an endless variety of different subjects through quizzes. Users can also create their own quizzes to test other users on the website, contributing to the growing collection of trivia for all types of topics. More importantly, users can create platforms which are customizable pages used to present quizzes they generate relating to a specific topic.

Trivia Tree differs from other quizzing sites by providing user leaderboards, a reward system for good performance, platform customization, and allowing users to see their relative performance compared to other quizzers. It also offers a simple to use, modernized user interface. Additionally, organizations, influencers, and users that are popular among the community will have the distinction of being verified by the site.

1.2 Objectives

- Allow users to create and customize platforms that are pages focused on singular topics. The platform creators attach quizzes akin to the platform's topic to test players on their knowledge of the topic.
- Allow users to create quizzes for any topics that interest them, and let them share their quizzes with the community through platforms.
- Design a system that allows users to answer user generated quizzes for fun, educational purposes, or just to show off how much they know.
- Provide each user with an account page that will serve as a page to show off their platforms and quizzes.
- Develop a reward system where users are awarded currency for doing well on quizzes. This currency can be used to purchase rewards from a shop page.
- Give users a personalized experience by providing platform and quiz recommendations based on previous quizzes they've taken in the past.
- Create a modern user interface that is fun to interact with while also being easy to use. The style of the UI will fit the educational theme of the site.

1.3 Constraints

- Finite server space for storing user/quiz information. Ensure that this is closely monitored so that the necessary adjustments can be made before hitting capacity.
- Creating accounts can only be done through Single Sign-On. There will be no unique login system created for the site.
- Copyright infringing and inappropriate content cannot be allowed legally or morally and will be removed.

1.4 Actors

- **[Guests]** - Most basic actor that can use the website and does not require any sort of login
 - Cannot create platforms and quizzes
 - Can visit platforms and account pages
 - Can take quizzes
 - Cannot receive currency
 - Scores are not ranked in the leaderboards
- **[Users]** - The standard actor of the website that is required to be logged in through Google SSO

- Can create a platform and quizzes
- Can visit platforms and account pages
- Can receive currency
- Scores are ranked in the leaderboards
- [Administrators] - Only website developers are given administrator privileges
 - Can delete quizzes
 - Can delete platforms
 - Can ban/delete users

1.5 Functional Requirements

- [Account Creation and Login] - Users will be able to log in to the site using their Google account through SSO. This will allow them to keep track of their results on past quizzes, allow them to create platforms, receive currency and purchase rewards, and be included in leaderboards.
- [User Account Page] - Users will have an account page where they will be able to choose quizzes and platforms that they want featured or “pinned”. They will be able to make small cosmetic adjustments to this page and it will mainly serve as a homepage for anyone interested in that user.
- [Create Platforms] - Users will be able to create/edit/delete platforms where they can present a topic and quizzes related to that topic. Platforms will feature a title, banner, icon, quizzes, etc.
- [Create Quizzes] - Users can create quizzes using a quiz-generating tool provided by the application; users can modify their quizzes with a timer, different questions types, etc. Users will also be able to edit or delete their own quizzes; admins will have the ability to delete any quiz they feel breaks user guidelines.
- [Take Quizzes] - Users will be able to take quizzes created by other users and organizations and see their performance compared to previous quiz takers.
- [Rating Quizzes] - Users will be able to rate and review the quizzes they take and provide feedback which will be used to figure out which quizzes will appear on the home page (most popular quizzes, best rated, etc.).
- [Reward System] - Users with accounts will be able to earn rewards based on their performance on quizzes and how well quizzes they create are received. Users will earn currency after every quiz, earning more for better performance. Currency can be used to purchase animated backgrounds, icon effects, and banners from the shop that can be used to customize their platforms. Users whose quizzes consistently receive high praises can become verified.
- [Leaderboards] - Users will be placed onto leaderboards based on their performance on individual quizzes and specific platforms. Users can view the leaderboards to see where they rank compared to other quiz takers.
- [Recommendation System] - After finishing a quiz, users will be recommended quizzes and platforms based on user data as well as the quiz that they just finished.

1.6 Non-Functional Requirements

- [Security] - Google SSO will authenticate users securely via an external Identity Provider, so important information such as passwords will not be at risk.
- [Performance] - Platforms/Quizzes will respond to users in a timely manner.
- [Reliability] - Program will be tested for various use cases, devices, and operating systems to ensure errors are minimized, and that the program has high performance in general use.
- [Recoverability] - Will periodically backup databases to MongoDB Atlas to ensure data integrity.
- [Scalability] - Will use Heroku as the hosting service to scale the program to meet user demands for data capacity and I/O throughput.
- [Manageability] - Admins shall take disciplinary actions on any platform page via an admin only

dropdown where admins can warn or delete a user.

- **[Maintainability]** - Code will be modular and well-documented to streamline updates.

1.7 Technology

- **[MongoDB]** - Cross-platform document-oriented database program used to store login information, user results, leaderboards, and quiz questions.
- **[Express]** - Designed for building web applications and APIs; will also handle the middleware needed for security and authentication purposes
- **[React]** - Front-end library to make a responsive and easy to use web application with a clean user interface.
- **[Node.js]** - Cross-platform back-end JavaScript runtime environment
- **[GraphQL]** - Data and manipulation language for fulfilling queries using existing data
- **[Apollo]** - Open source GraphQL Server

1.8 Training Requirements

Each member will:

1. Go back and analyze their 316 projects in order to refresh their understanding of the technologies mentioned above. Afterwards, a session will be held where each member discusses their project.
2. Create a simple MERN project that will feature a multiple choice quiz where the questions and correct responses will be pulled from a database.

Members will hold each other accountable by verifying with each other that they have completed the simple MERN project as described in the 2nd bullet point. Through analyzing each member's project, other members will know if that member has an understanding of each technology if they were able to fulfill all the specified requirements.

1.9 Action Plan

For the month of September, we will spend a considerable amount of time planning the project. Once we have sharply defined the bounds of the project, we will begin making models that will help organize our ideas. We will create a use-case model which will list all of the specific functions of the application as well as its constraints. Furthermore, we will create mockups of all user interface screens and list out any additional components that the site will need. While we are creating models, all teammates will continue to refresh their skills with the technologies that will be used for the website. By October, we will have data, domain, and software models for our web application.

Our first deployment will be on **October 20th**. By this date all modeling will be complete, each team member will have become sufficiently comfortable with all technologies, and we will be ready for a full implementation. For the first build, we'll deploy a simple barebones project. For the next month, we will continue to develop the application and regularly meet with our TA to keep us on track. We will have multiple builds throughout the month. We will also ask people outside of the team to test features of the website and provide feedback on how the user experience can be improved.

By **December 1st**, we will have a complete implementation of our application. The final product will be deployed, and we will allow users to utilize the application. After the final deployment, we will use feedback from users to fix bugs and continue to improve the product.

2. Use Case Model

2.1 Use Case Listing

Use Case Number	UI Context	Use Case Name
2.1	Sign Up Page	Sign Up
2.2	Sign In Page	Sign In
2.3	Navigation Bar (Dropdown)	Sign Out
2.4	Navigation Bar	Return To Homepage
2.5	Navigation Bar	Search for a Platform
2.6	Navigation Bar	Search for a Quiz
2.7	Navigation Bar	Search for a User
2.8	Navigation Bar	Select a Category for Search
2.9	Navigation Bar	Create Platform
2.10	Navigation Bar	Create Quiz
2.11	Navigation Bar	Navigate to Shop
2.12	Nav Bar (Dropdown)	Navigate to Your Account Page
2.13	Nav Bar (Dropdown)	Navigate to Platform Manager
2.14	Nav Bar (Dropdown)	Navigate to Quiz Manager
2.15	Nav Bar (Dropdown)	Navigate to Favorites
2.16	Nav Bar (Dropdown)	Navigate to Followed Platforms
2.17	Nav Bar (Dropdown)	Navigate to Settings Page
2.18	Search Results Page	Sort search results
2.19	Search Results Page	Filter by Number of Plays
2.20	Search Results Page	Filter by Difficulty
2.21	Search Results Page	Filter by Types
2.22	Homepage / Search Results Page /	Select Platform

	Account Page	
2.23	Homepage / Search Results Page / Account Page / Platform Page	Select User
2.24	Homepage / Platform / Search Results Page / Favorites Page / Account Page	Select a Quiz
2.25	Pre-Quiz Page	Start Quiz
2.26	Pre-Quiz Page / Post-Quiz Page	Navigate back to search
2.27	Pre-Quiz Page	View Quiz Info
2.28	Pre-Quiz Page / Post-Quiz Page	View Leaderboard
2.29	Pre-Quiz Page (Admin)	Delete Quiz (Admin)
2.30	Quiz Page	Select an answer
2.31	Quiz Page	Navigate to next question
2.32	Quiz Page	Navigate to previous question
2.33	Quiz Page	Navigate to a specific question
2.34	Quiz Page	Finish quiz
2.35	Post-Quiz Page	Review Quiz Results
2.36	Post-Quiz Page	Personal Results
2.37	Post-Quiz Page	Retake quiz
2.38	Post-Quiz Page	Rate quiz
2.39	Pre-Quiz / Post-Quiz Page	Favorite / Unfavorite a quiz
2.40	Post-Quiz Page	Comment on a Quiz
2.41	Post-Quiz Page	Reply on a quiz comment
2.42	Post-Quiz Page	Delete a quiz comment
2.43	Post-Quiz Page	Delete a quiz reply
2.44	Platform	Follow Platform
2.45	Platform	Unfollow Platform

2.46	Platform	View Platform Followers
2.47	Platform (Admin)	Delete Platform (Admin)
2.48	Account Page	View User's Platforms
2.49	Account Page	View User's Quizzes
2.50	Account Page	View User's Favorited Quizzes
2.51	Account Page	View User's Followed Platforms
2.52	Account Page	Edit Account Page
2.53	Account Page (Admin)	Ban User (Admin)
2.54	Editing Account	Update Account Picture
2.55	Editing Account	Update Account Banner
2.56	Editing Account	Update Account Background
2.57	Editing Account	Update Account Bio
2.58	Editing Account	Update Featured Platforms
2.59	Editing Account	Update Featured Quizzes
2.60	Platform Manager / Platform	Edit Platform
2.61	Platform Manager	Delete Platform
2.62	Editing Platform	Update Platform Title
2.63	Editing Platform	Update Platform Icon
2.64	Editing Platform	Update Platform Banner
2.65	Editing Platform	Update Platform Description
2.66	Editing Platform	Add Quiz to Platform
2.67	Editing Platform	Remove Quiz from Platform
2.68	Quiz Manager	Edit Quiz
2.69	Quiz Manager	Delete Quiz
2.70	Editing Quiz	Update Quiz Image
2.71	Editing Quiz	Update Quiz Name

2.72	Editing Quiz	Add Quiz Question
2.73	Editing Quiz	Delete Quiz Question
2.74	Editing Quiz	Update Quiz Question Title
2.75	Editing Quiz	Update Quiz Answer Choices
2.76	Editing Quiz	Update Quiz Answer
2.77	Editing Quiz	Update Quiz Timer
2.78	Editing Quiz	Update between Instant and Standard
2.79	Editing Quiz	Update between Shuffle and Ordered
2.80	Shop Screen	Select Item from Shop
2.81	Purchase Screen	Preview Item
2.82	Purchase Screen	Purchase Item
2.83	Settings Page	Update User Display Name
2.84	Settings Page	Change between Dark/Light Mode
2.85	Settings Page	Delete User Account

2.2 Use Case Description

Use-Case Number:	2.1
Use-Case Name:	Sign Up
Actors:	Guests
Preconditions:	Guest has a Google account
Postconditions:	User account is created via SSO and information on the user stored in the database
Story:	<ol style="list-style-type: none"> 1. Guest opens website 2. Guest left-clicks “Sign Up” 3. Guest left-clicks “Sign Up with Google” 4. Guest fills out Google SSO 5. Guest fills out display name 6. Guest left-clicks “Create Account”
Scenario:	<ol style="list-style-type: none"> 1. Jon Doe types “quizsite.com” 2. Jon Doe observes Sign In/Sign Up page 3. Jon left-clicks “Sign Up” 4. Jon left-clicks “Sign Up with Google” 5. Jon fills out the Google SSO 6. Jon enters a display name 7. Jon left-clicks “Create Account” 8. Jon is taken to the homepage
Exceptions:	<ol style="list-style-type: none"> 1. If a user attempts to sign up with a Google account that has already been used to sign up to the website, display an error message stating the email is already in use
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Once per account lifetime
Open Issues:	<ol style="list-style-type: none"> 1. Needs to use Google SSO

Use-Case Number:	2.2
Use-Case Name:	Sign In
Actors:	Users, Verified Users, Admins
Preconditions:	User has an account
Postconditions:	User is signed in and is taken to the home page
Story:	<ol style="list-style-type: none"> 1. User opens website 2. User left-clicks “Sign In” 3. User fills out login info 4. User left-clicks “Sign In”
Scenario:	<ol style="list-style-type: none"> 1. Jon Doe types “quizsite.com” 2. Jon Doe observes Sign In/Sign Up page 3. Jon left-clicks “Sign In” 4. Jon fills out username “Johnny” 5. Jon fills out password “Donny” 6. Jon left-clicks “Sign In” 7. Jon is taken to the home page
Exceptions:	<ol style="list-style-type: none"> 1. If a user enters a combination of username and password that is not held in the database, display an error message stating the username or password is incorrect
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Once per session
Open Issues:	<ol style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.3
Use-Case Name:	Sign Out
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	User is signed out and taken to the homepage
Story:	<ul style="list-style-type: none"> 1. User left-clicks “Sign Out” 2. User is taken to the homepage
Scenario:	<ul style="list-style-type: none"> 1. Jon Doe observes navigation bar 2. Jon left-clicks “Sign Out” 3. Jon is taken to the homepage
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Once per session
Open Issues:	<ul style="list-style-type: none"> 1. CORS to send info to and from browser

Use-Case Number:	2.4
Use-Case Name:	Return to Homepage
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User is taken to the homepage
Story:	<ul style="list-style-type: none"> 1. User left-clicks “Logo” 2. User is taken to the homepage
Scenario:	<ul style="list-style-type: none"> 1. Jon Doe observes navigation bar 2. Jon left-clicks the website logo 3. Jon is taken to the homepage
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	None

Use-Case Number:	2.5
Use-Case Name:	Search for a Platform
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User is taken to a results page based on search parameters
Story:	<ol style="list-style-type: none"> 1. User either left-clicks the category dropdown to choose a category or does not choose anything in which case categories will default to “All” 2. User left-clicks the search bar in the nav bar at the top 3. User either fills out the search or leaves it empty in which case the search will default to any platform in the selected category. 4. User left-clicks the search platform button or hits enter to search 5. User is taken to a results page with platform based on search parameters
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe left-clicks the search bar 3. Joe fills search bar with “CSE” 4. Joe left-clicks the search platform button 5. Joe is taken to a page with platforms that relate to “CSE” and are not filtered by any category
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. Implementing search with categories and search parameter 2. Pulling information from MongoDB/hosting using GraphQL 3. CORS to send info to and from browser

Use-Case Number:	2.6
Use-Case Name:	Search for a Quiz
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User is taken to a results page based on search parameters
Story:	<ol style="list-style-type: none"> 1. User either left-clicks the category dropdown to choose a category or does not choose anything in which case categories will default to “All” 2. User left-clicks the search bar in the nav bar at the top 3. User either fills out the search or leaves it empty in which case the search will default to any quiz in the selected category. 4. User left-clicks the search quiz button or hits enter to search 5. User is taken to a results page with quizzes based on search parameters
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe left-clicks the search bar 3. Joe fills search bar with “CSE” 4. Joe left-clicks the search quiz button 5. Joe is taken to a page with quizzes that relate to “CSE” and are not filtered by any category
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. Implementing search with categories and search parameter 2. Pulling information from MongoDB/hosting using GraphQL 3. CORS to send info to and from browser

Use-Case Number:	2.7
Use-Case Name:	Search for a User
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User is taken to a results page based on search parameters
Story:	<ol style="list-style-type: none"> 1. User either left-clicks the category dropdown to choose a category or does not choose anything in which case categories will default to “All” 2. User left-clicks the “User Search” button 3. User left-clicks the search bar in the nav bar 4. User either fills out the search or leaves it empty in which case the search will default to any user in the selected category 5. User left-clicks the search button or hits enter to search 6. User is taken to a results page based on search parameters
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe left-clicks category, scrolls using mouse and left-clicks “Baseball” 3. Joe Doe left-clicks the “User” button 4. Joe left-clicks the search bar in the nav bar at the top 5. Joe fills search bar with “Joe Doe” 6. Joe left-clicks the search user button 7. Joe is taken to a page with users that have usernames similar to “Joe Doe” and have a quiz in the category of “Baseball”
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. Implementing search with categories and search parameter 2. Pulling information from MongoDB/hosting using GraphQL 3. CORS to send info to and from browser

Use-Case Number:	2.8
Use-Case Name:	Select a Category for Search
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User filters search by category
Story:	<ol style="list-style-type: none"> 1. User left-clicks the category dropdown to choose a category 2. User left-clicks on a category from the dropdown
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe left-clicks category 3. Joe Doe scrolls using mouse 4. Joe Doe left-clicks “Baseball”
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. Implementing search with categories

Use-Case Number:	2.9
Use-Case Name:	Create Platform
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	User is taken to the platform creation page
Story:	<ol style="list-style-type: none"> 1. User left-clicks the “Create” button in the nav bar 2. User left-clicks the “Create Platform” button 3. User is taken to the platform creation page
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe left-clicks the “Create” button in the nav bar 3. Joe Doe left-clicks the “Create Platform” button 4. Joe is taken to a page with tools for platform creation
Exceptions:	<ol style="list-style-type: none"> 1. If a guest tries to make a quiz, they are taken to a page which tells them they can't create quizzes without logging in and has a link to the Log In/Sign Up page
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many timers per account lifetime
Open Issues:	<ol style="list-style-type: none"> 1. The platform creation tools that will be included

Use-Case Number:	2.10
Use-Case Name:	Create Quiz
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	User is taken to the quiz creation page
Story:	<ul style="list-style-type: none"> 1. User left-clicks the “Create” button in the nav bar 2. User left-clicks the “Create Quiz” button in the nav bar 3. User is taken to the quiz creation page
Scenario:	<ul style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe left-clicks the “Create” button in the nav bar 3. Joe Doe left-clicks “Create Quiz” button 4. Joe is taken to a page with tools for quiz creation
Exceptions:	<ul style="list-style-type: none"> 2. If a guest tries to make a quiz, they are taken to a page which tells them they can't create quizzes without logging in and has a link to the Log In/Sign Up page
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	<ul style="list-style-type: none"> 2. Quiz creation tools included

Use-Case Number:	2.11
Use-Case Name:	Navigate to Shop
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	User is taken to the shop page
Story:	<ul style="list-style-type: none"> 1. User left-clicks the “Shop” button in the nav bar 2. User is taken to the shop page
Scenario:	<ul style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe left-clicks “Shop” button 3. Joe is taken to the shop page
Exceptions:	<ul style="list-style-type: none"> 1. If a guest left-clicks the “Shop” button in the nav bar, they are taken to the shop page but aren’t able to purchase any items from it
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	<ul style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.12
Use-Case Name:	Navigate to Your Account Page
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	User is taken to their account page
Story:	<ol style="list-style-type: none"> 1. User left-clicks on their profile picture in the nav bar 2. User left-clicks the “Account” button in the dropdown 3. User is taken to his account page
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe left-clicks on his profile picture 3. Joe Doe observes the dropdown bar 4. Joe Doe left-clicks “Account” button 5. Joe is taken to his own account page
Exceptions:	<ol style="list-style-type: none"> 1. If a guest tries to enter their own account page, they are taken to the Log In/Sign Up page
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.13
Use-Case Name:	Navigate to Platform Manager
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	User is taken to the management page for their own platforms
Story:	<ol style="list-style-type: none"> 1. User left-clicks their profile picture in the nav bar 2. User left-clicks the “Platform Manager” button in the dropdown
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe left-clicks their profile picture 3. Joe Doe observes the dropdown bar 4. Joe Doe left-clicks “Platform Manager” button 5. Joe is taken to the management page for his own platforms
Exceptions:	<ol style="list-style-type: none"> 1. If a guest tries to enter the management page for their own quizzes, they are taken to the Log In/Sign Up page
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	A few times per session
Open Issues:	<ol style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.14
Use-Case Name:	Navigate to Quiz Manager
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	User is taken to the management page for their own quizzes
Story:	<ul style="list-style-type: none"> 3. User left-clicks their profile picture in the nav bar 4. User left-clicks the “Quiz Manager” button in the dropdown
Scenario:	<ul style="list-style-type: none"> 6. Joe Doe observes the nav bar 7. Joe Doe left-clicks their profile picture 8. Joe Doe observes the dropdown bar 9. Joe Doe left-clicks “Quiz Manager” button 10. Joe is taken to the management page for his own quizzes
Exceptions:	<ul style="list-style-type: none"> 2. If a guest tries to enter the management page for their own quizzes, they are taken to the Log In/Sign Up page
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	<ul style="list-style-type: none"> 3. Pulling information from MongoDB/hosting using GraphQL 4. CORS to send info to and from browser

Use-Case Number:	2.15
Use-Case Name:	Navigate to Favorites
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	User is taken to their favorites page
Story:	<ul style="list-style-type: none"> 1. User left-clicks their profile picture in the nav bar 2. User left-clicks the “Favorites” button in the dropdown
Scenario:	<ul style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe left-clicks their profile picture 3. Joe Doe observes the dropdown bar 4. Joe Doe left-clicks “Favorites” button 5. Joe is taken to his favorites page
Exceptions:	<ul style="list-style-type: none"> 1. If a guest tries to enter their favorites page, they are taken to the Log In/Sign Up page
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	<ul style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.16
Use-Case Name:	Navigate to Followed Platforms
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	User is taken to a page that shows platforms they follow
Story:	<ul style="list-style-type: none"> 1. User left-clicks their profile picture in the nav bar 2. User left-clicks the “Followed Platforms” button in the dropdown
Scenario:	<ul style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe observes the dropdown bar 3. Joe Doe left-clicks “Followed Platforms” button 4. Joe Doe is taken to the page of platforms he follows
Exceptions:	<ul style="list-style-type: none"> 1. If a guest tries to enter the page that shows their followed platforms, they are taken to the Log In/Sign Up page
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	<ul style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.17
Use-Case Name:	Navigate to Settings Page
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User is taken to their settings page
Story:	<ol style="list-style-type: none"> 1. User left-clicks their profile picture in the nav bar 2. User left-clicks the “Settings” button in the dropdown
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the nav bar 2. Joe Doe left-clicks their profile picture 3. Joe Doe observes the dropdown bar 4. Joe Doe left-clicks “Settings” button 5. Joe is taken to his settings page
Exceptions:	<ol style="list-style-type: none"> 1. If a guest tries to enter their settings page, they are taken to a page with fewer settings
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.18
Use-Case Name:	Sort search results
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	User has already made a search
Postconditions:	User's search is sorted by a parameter without reloading the page
Story:	<ol style="list-style-type: none"> 1. User left-clicks a sort button and a drop down appears 2. User left-clicks on one of the parameters to sort by 3. The user's search results are sorted by the parameter in descending order
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the search results 2. Joe Doe left-clicks the sort button and a drop down appears 3. Joe Doe observes the sort drop down 4. Joe Doe left-clicks "Difficulty" to sort difficulty in descending order 5. Joe Doe observes the sorted results 6. Joe Doe left-clicks the arrow next to the sort button to sort difficulty by ascending instead
Exceptions:	None
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. Implementing search with sort 2. Implementing tables that can be altered without reloading the page

Use-Case Number:	2.19
Use-Case Name:	Filter by Number of Plays
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	User has already made a search
Postconditions:	User's search is filtered by the number of plays
Story:	<ul style="list-style-type: none"> 1. User fills out the filter information 2. User left-clicks the "Filter" button 3. User is taken to a results page based on search parameters
Scenario:	<ul style="list-style-type: none"> 1. Joe Doe observes the search results page 2. Joe Doe fills out the filter information for the "number of plays more than" with 5000 3. Joe Doe left-clicks the "Filter" button 4. Joe Doe is taken to a page with only quizzes that have been played more than 5000 times
Exceptions:	<ul style="list-style-type: none"> 1. If the value put in the box is not a non-negative integer, display a message stating that a non-negative integer must be put in the filter for "number of plays more than/less than" under the box
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	<ul style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.20
Use-Case Name:	Filter by Difficulty
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	User has already made a search
Postconditions:	User's search is filtered by difficulty
Story:	<ol style="list-style-type: none"> 1. User fills out the filter information for desired difficulty 2. User left-clicks the "Filter" button 3. User is taken to page of search results with filters applied
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the search results page 2. Joe Doe left-clicks the label/checkbox pair "Easy" and "Medium" for "Difficulty" 3. Joe Doe left-clicks the "Filter" button 4. Joe Doe is taken to a page with only quizzes that have difficulty "Easy" or "Medium"
Exceptions:	None
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.21
Use-Case Name:	Filter by Types
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	User has already made a search
Postconditions:	User's search is filtered by types
Story:	<ol style="list-style-type: none"> 1. User selects the types of quizzes they want in the filter input box 2. User left-clicks the "Filter" button 3. User is taken to a page of search results with the filter applied
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the search results page 2. Joe Doe left-clicks the label/checkbox pair "Instant" and "Shuffle" for "Types" 3. Joe Doe left-clicks the "Filter" button 4. Joe Doe is taken to a page with only quizzes that are either instant or shuffle type quizzes
Exceptions:	None
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. Pulling information from MongoDB/hosting using GraphQL 2. CORS to send info to and from browser

Use-Case Number:	2.22
Use-Case Name:	Select Platform
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User is taken to the platform that they selected
Story:	<ol style="list-style-type: none"> 1. User left-clicks on a platform's icon 2. User is taken to that platform
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes a platform icon for a platform about baseball 2. Joe Doe left-clicks on that platform icon 3. Joe Doe is taken to the platform about baseball
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	A few times per session
Open Issues:	<ol style="list-style-type: none"> 1. Will a user be able to make his platform completely private so that other users can't find it?

Use-Case Number:	2.23
Use-Case Name:	Select User
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User is taken to the user's account page that they selected
Story:	<ol style="list-style-type: none"> 1. User left-clicks on a user icon 2. User is taken to that user's account page
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes a quiz box for a quiz about baseball 2. Joe Doe left-clicks on the user icon imbedded within the quiz box 3. Joe Doe is taken to the account page for the user who created the baseball quiz
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	A few times per session
Open Issues:	<ol style="list-style-type: none"> 1. Will a user be able to make his platform completely private?

Use-Case Number:	2.24
Use-Case Name:	Select a Quiz
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User is taken to the pre-quiz page of the quiz they selected
Story:	<ol style="list-style-type: none"> 1. User left-clicks on a quiz 2. User is taken to the pre-quiz page of that quiz
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes a quiz box for a quiz about baseball 2. Joe Doe left-clicks on a quiz box for the baseball quiz 3. Joe Doe is taken to the pre-quiz page of the baseball quiz
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	None

Use-Case Number:	2.25
Use-Case Name:	Start Quiz
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User starts the quiz and is taken to the page with the first question
Story:	<ol style="list-style-type: none"> 1. User left-clicks on the “Start Quiz” button 2. User is taken to the first question of the quiz they just started
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the pre-quiz page of a baseball quiz 2. Joe Doe left-clicks on the “Start Quiz” button 3. Joe Doe is taken to the first question of the baseball quiz
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	None

Use-Case Number:	2.26
Use-Case Name:	Navigate back to search
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User goes back to the search page that led them to the quiz they were just on
Story:	<ol style="list-style-type: none"> 1. User left-clicks on the “Back” button 2. User is taken back to the search page they were previously on
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the pre-quiz page of a baseball quiz 2. Joe Doe left-clicks on the “Back” button 3. Joe Doe is taken back to the search page that led him to the baseball quiz he was just on
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	<ol style="list-style-type: none"> 1. How can we implement allowing a user to save the scrolling position of the search they had previously done?

Use-Case Number:	2.27
Use-Case Name:	View Quiz Info
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User moves to the quiz info page
Story:	<ul style="list-style-type: none"> 1. User left-clicks on the “Quiz Info” button 2. User is taken to the quiz info page for the quiz they are currently on
Scenario:	<ul style="list-style-type: none"> 1. Joe Doe observes the pre-quiz page of a baseball quiz 2. Joe Doe left-clicks on the “Quiz Info” button 3. Joe Doe is taken to a page that shows him general information about the baseball quiz
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	A few times per session
Open Issues:	<ul style="list-style-type: none"> 1. What info will be included in the quiz info page?

Use-Case Number:	2.28
Use-Case Name:	View Leaderboard
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User goes to the leaderboard page of the quiz they are currently on
Story:	<ol style="list-style-type: none"> 1. User left-clicks on the “Leaderboard” button 2. User is taken to the leaderboard page for the quiz they are currently on
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the pre-quiz page of a baseball quiz 2. Joe Doe left-clicks on the “Leaderboard” button 3. Joe Doe is taken to a page that shows the leaderboard for the baseball quiz
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	A few times per session
Open Issues:	<ol style="list-style-type: none"> 1. What will be displayed for the leaderboard if no one has taken the quiz?

Use-Case Number:	2.29
Use-Case Name:	Delete Quiz (Admin)
Actors:	Admins
Preconditions:	User must be logged into their account, User must be an admin
Postconditions:	The quiz the admin has deleted will be removed entirely from the website
Story:	<ol style="list-style-type: none"> 1. Admin observes the pre-quiz page for a quiz 2. Admin left clicks “Delete quiz” button 3. A pop-up appears stating “Are you sure you want to delete this quiz?” 4. Admin left-clicks “Yes” 5. Admin is brought back to the home page. The quiz will no longer exist meaning it will be accessed by any user
Scenario:	<ol style="list-style-type: none"> 1. Admin decides to permanently delete an inappropriate quiz from the website 2. Admin deletes the inappropriate quiz 3. Admin left-clicks the “Delete Quiz” button on the pre-quiz page for the inappropriate quiz 4. A pop-up appears stating “Are you sure you want to delete this quiz?” 5. Admin left-clicks “Yes” 6. Admin is redirected to the home page. The inappropriate quiz will no longer exist meaning it will be accessed by any user
Exceptions:	None
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used infrequently
Open Issues:	None

Use-Case Number:	2.30
Use-Case Name:	Select an answer
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User selects one of the choices on a question
Story:	<ol style="list-style-type: none"> 1. User left-clicks on one of the choices on the question page
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the question page for a baseball quiz 2. The question asks, "When was baseball invented?", Joe Doe left-clicks on the choice that states "1839"
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	None

Use-Case Number:	2.31
Use-Case Name:	Navigate to next question
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User moves to the next question of the quiz they are currently on
Story:	<ol style="list-style-type: none"> 1. User left-clicks on the next question button 2. User moves to the the page with the next question
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the question page for a baseball quiz 2. Joe Doe left-clicks on the next question button 3. Joe Doe is taken to the quiz page for the next question
Exceptions:	<ol style="list-style-type: none"> 1. If the user is on the last question of the quiz, there will be no navigate to next question button
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	None

Use-Case Number:	2.32
Use-Case Name:	Navigate to previous question
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User moves to the previous question of the quiz they are currently on
Story:	<ol style="list-style-type: none"> 1. User left-clicks on the previous question button 2. User moves to the page of the question that came before the question they are currently on
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the question page for a baseball quiz 2. Joe Doe left-clicks on the previous question button 3. Joe Doe is taken to the quiz page for the previous question
Exceptions:	<ol style="list-style-type: none"> 1. If the user is on the first question of the quiz, the previous question button will not appear
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	None

Use-Case Number:	2.33
Use-Case Name:	Navigate to specific question
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User moves to a specific question of the quiz they are currently on
Story:	<ul style="list-style-type: none"> 1. User left-clicks on one of the number buttons, which each correspond to a specific question on the quiz 2. User is taken to the page of the specific question they clicked the number for
Scenario:	<ul style="list-style-type: none"> 1. Joe Doe observes the question page for a baseball quiz 2. Joe Doe left-clicks on the number button that indicates “5” 3. Joe Doe is taken to the fifth question of the baseball quiz
Exceptions:	<ul style="list-style-type: none"> 1. You cannot navigate to the question you are currently on. The icon for that number will be greyed out.
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	<ul style="list-style-type: none"> 1. How will we format the listing of numbers on the UI in a way that looks appealing, and that will also scale with large quizzes?

Use-Case Number:	2.34
Use-Case Name:	Finish quiz
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User finishes the quiz meaning they “end” the quiz and can no longer change their answers
Story:	<ul style="list-style-type: none"> 1. User left-clicks on ”Finish Quiz” button 2. User moves to the post-quiz page
Scenario:	<ul style="list-style-type: none"> 1. Joe Doe observes the last page for a baseball quiz 2. Joe Doe left-clicks on the “Finish Quiz” button 3. Joe Doe is taken to the post-quiz page
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #2
Frequency of use:	Many times per session
Open Issues:	None

Use-Case Number:	2.35
Use-Case Name:	Review Quiz Results
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User moves to the review quiz results page
Story:	<ol style="list-style-type: none"> 1. User left-clicks on "Review Quiz" button 2. User moves to the review quiz page
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the post-quiz page for a baseball quiz 2. Joe Doe left-clicks on the "Review Quiz" button 3. Joe Doe is taken to the review page for the baseball quiz
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	None

Use-Case Number:	2.36
Use-Case Name:	Personal Results
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User moves to the review quiz results page
Story:	<ol style="list-style-type: none"> 1. User left-clicks on "Personal Stats" button 2. User moves to the personal stats page
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the post-quiz page for a baseball quiz 2. Joe Doe left-clicks on the "Personal Stats" button 3. Joe Doe is taken to the personal stats page for the baseball quiz where he sees that he is at the bottom of the leaderboard
Exceptions:	<ol style="list-style-type: none"> 1. A guest's stats will not be saved but they will still be shown the various stats and while they won't be placed on the leaderboard they will be shown where they would have been placed
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Many times per session
Open Issues:	None

Use-Case Number:	2.37
Use-Case Name:	Retake Quiz
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	User moves to the pre-quiz page of the quiz they just finished
Story:	<ol style="list-style-type: none"> 1. User left-clicks on "Retake Quiz" button 2. User moves to the pre-quiz page of the quiz they just finished
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the post-quiz page for a baseball quiz 2. Joe Doe left-clicks on the "Retake Quiz" button 3. Joe Doe is taken to the pre-quiz page for the baseball quiz
Exceptions:	None
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	A few times per session
Open Issues:	None

Use-Case Number:	2.38
Use-Case Name:	Rate Quiz
Actors:	Users, Verified Users, Admins
Preconditions:	User is Signed In
Postconditions:	The rating of the quiz is adjusted based on the user's rating
Story:	<ul style="list-style-type: none"> 1. User left-clicks on the rating slider to specify the rating they want to give the quiz on a scale from 0-5 with steps of 0.5
Scenario:	<ul style="list-style-type: none"> 1. Joe Doe observes the post-quiz page for a baseball quiz 2. Joe Doe left-clicks on the rating slider to represent a 4 star rating 3. Joe Doe rating is solidified by the slider holding the rating Joe gave
Exceptions:	<ul style="list-style-type: none"> 1. A guest will not be able to rate a quiz instead they will be shown a popup to sign up
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	A few times per session
Open Issues:	None

Use-Case Number:	2.39
Use-Case Name:	Favorite/Unfavorite a Quiz
Actors:	Users, Verified Users, Admins
Preconditions:	User is Signed In
Postconditions:	The quiz is added or removed from the user's favorites
Story:	<ol style="list-style-type: none"> 1. User left-clicks on the "Favorite" button 2. The "Favorite" button changes to say "Unfavorite" 3. User left-clicks on the "Unfavorite" button 4. The "Unfavorite" button changes to "Favorite"
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the post-quiz page for a baseball quiz 2. Joe Doe left-clicks on the "Favorite" button 3. The baseball quiz is added to Joe's favorites and the "Favorite" button changes to "Unfavorite" 4. Joe Doe observes the favorite icon 5. Jon Doe decides he does not like the quiz anymore. Jon Doe left-clicks on the "Unfavorite" button. 6. The baseball quiz is taken off of Joe's favorites
Exceptions:	<ol style="list-style-type: none"> 1. A guest will not be able to favorite a quiz, instead they will be shown a popup to sign up
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	A few times per session
Open Issues:	None

Use-Case Number:	2.40
Use-Case Name:	Comment on a Quiz
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in and has finished the quiz
Postconditions:	A comment from the User is added to a quiz's comment section
Story:	<ol style="list-style-type: none"> 1. User left-clicks on the "Comment" button and a text box appears 2. User left-clicks on the text box 3. User types in their comment in the text box 4. User left-clicks "Post" and their comment appears in the comment section
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe observes the post-quiz page for a baseball quiz 2. Joe Doe left-clicks on the "Comment" button 3. Joe Doe left-clicks on the text box that appears 4. Joe Doe types "The best quiz ever" in the text box 5. Joe Doe left-clicks "Post" and their comment is added to the baseball quiz's comment section
Exceptions:	<ol style="list-style-type: none"> 1. A guest will not be able to comment on a quiz, instead they will be shown a popup to sign up
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	A few times per session
Open Issues:	None

Use-Case Number:	2.41
Use-Case Name:	Reply on a quiz comment
Actors:	Users, Verified Users, Admins
Preconditions:	User is signed in
Postconditions:	A comment from the User is displayed as a reply under an already existing comment
Story:	<ol style="list-style-type: none"> 1. User left-clicks the “Reply” button displayed under a user’s comment and a text box appears 2. User left-clicks the text box 3. User types their reply 4. User left-clicks “Reply” and their comment appears as a reply under the original commenter’s comment
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe left-clicks the “Reply” button displayed under QuizCommenter19’s comment “Good quiz :)”. A text box appears below the comment. 2. Joe_Doe left-clicks the text box 3. Joe_Doe types “I agree, this was a great quiz!” 4. Joe_Doe left-clicks “Reply” and their comment appears as a reply under QuizCommenter19’s comment
Exceptions:	<ol style="list-style-type: none"> 1. If the reply is unable to be sent because of server problems, a message will appear reading “Reply failed to send.”
Priority:	Wish List
When available:	Build #5
Frequency of use:	Regularly used
Open Issues:	<ol style="list-style-type: none"> 1. How many replies will be allowed on a single comment? 2. Will replies be collapsable? How will this be implemented?

Use-Case Number:	2.42
Use-Case Name:	Delete a quiz comment
Actors:	Users, Verified Users, Admins
Preconditions:	Comment to be deleted belongs to the user OR Comment deleter is an admin
Postconditions:	The comment to be deleted is removed and no longer visible to any users
Story:	<ol style="list-style-type: none"> 1. User observes comment to be deleted 2. User left-clicks on the trash icon under the comment 3. A message appears reading “Are you sure you want to delete this comment?” 4. User left-clicks “Yes” 5. The confirmation message disappears from the screen and the comment is removed. Any comments below it are moved up to take its place.
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe observes comment that they previously wrote that reads “I did not like this quiz very much” 2. Joe_Doe left-clicks on the trash icon under the comment 3. A message appears reading “Are you sure you want to delete this comment?” 4. Joe_Doe left-clicks “Yes” 5. The confirmation message and deleted comment disappear from the screen. The comments below move up and take its place.
Exceptions:	<ol style="list-style-type: none"> 1. If the comment is unable to be deleted because of server problems, a message will appear reading “Comment failed to be deleted” 2. If user tries to delete comment that isn’t theirs, the “Delete Comment” button will not appear
Priority:	Wish List
When available:	Build #5
Frequency of use:	Used infrequently
Open Issues:	<ol style="list-style-type: none"> 1. Must be able to ensure users can’t delete each other’s comments without having administrator privileges 2. Deleted comments should disappear for everyone and should show no signs of ever having existed

Use-Case Number:	2.43
Use-Case Name:	Delete a quiz reply
Actors:	Users, Verified Users, Admins
Preconditions:	Reply to be deleted belongs to the user OR Reply deleter is an admin
Postconditions:	The reply to be deleted is removed and no longer visible to any users
Story:	<ol style="list-style-type: none"> 1. User observes reply to be deleted 2. User left-clicks on trash icon under the reply 3. A message appears reading “Are you sure you want to delete this reply?” 4. User left-clicks “Yes” 5. The confirmation message disappears from the screen and the reply is removed. Any replies below it are moved up to take its place.
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe observes a reply that they previously wrote that reads “I disagree, this quiz made me terminally ill” 2. Joe_Doe left-clicks on the trash icon under the reply 3. A message appears reading “Are you sure you want to delete this reply?” 4. Joe_Doe left-clicks “Yes” 5. The confirmation message and deleted reply disappear from the screen. The replies below move up and take its place.
Exceptions:	<ol style="list-style-type: none"> 1. If the reply is unable to be deleted because of server problems, a message will appear reading “Reply failed to be deleted” 2. If user tries to delete reply that isn’t theirs, the “Delete Reply” button will not appear
Priority:	Wish List
When available:	Build #5
Frequency of use:	Used infrequently
Open Issues:	<ol style="list-style-type: none"> 1. Must be able to ensure users can’t delete each other’s replies without having administrator privileges 2. Deleted replies should disappear for everyone and should show no signs of ever having existed

Use-Case Number:	2.44
Use-Case Name:	Follow Platform
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account, User must not already be following the platform
Postconditions:	The user will be “following” the specified platform and will see new quiz releases from this platform on their homepage. The platform will also appear in the “Followed Platforms” page for the user.
Story:	<ol style="list-style-type: none"> 1. User observes a platform 2. User left-clicks the Follow button in the platform banner. The user is now following that platform 3. The Follow button turns into an Unfollow button
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe enjoys taking quizzes on the Cars platform and wants to take new quizzes from this platform when they come out 2. Joe_Doe left-clicks the Follow button next to the Cars platform. Joe is now following the Cars platform 3. The Follow button turns into an Unfollow button
Exceptions:	<ol style="list-style-type: none"> 1. A user cannot follow their own platforms. The Follow button will not appear when viewing platforms they own 2. Guests cannot follow platforms. The Follow button will not appear if the user is not signed in
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Several times per session
Open Issues:	<ol style="list-style-type: none"> 1. How will we implement the feature to show Followed Platforms’ new quizzes on the homepage? 2. Will the platform creator be notified when someone follows their platforms? Where will they receive this notification?

Use-Case Number:	2.45
Use-Case Name:	Unfollow Platform
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account, User must be following the platform
Postconditions:	The user will no longer be following the specified platform and will no longer see new quiz releases from this platform on their homepage. The platform will be removed from the Followed Platforms page for the user.
Story:	<ol style="list-style-type: none"> 1. User observes a platform 2. User left-clicks the Unfollow button in the platform banner. A message that reads “Are you sure you want to unfollow [Platform]?” appears 3. User left-clicks “Yes”. The user is no longer following that platform 4. The Unfollow button turns into a Follow button
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe decides they no longer enjoy Tennis quizzes and doesn't want to follow the Tennis platform anymore. 2. Joe_Doe left-clicks the Unfollow button on the Tennis platform page. 3. A message that reads “Are you sure you want to unfollow Tennis?” appears 4. Joe_Doe left-clicks “Yes”. Joe Doe no longer follows the Tennis platform 5. The Unfollow button turns into an Follow button
Exceptions:	<ol style="list-style-type: none"> 1. Because Guests can't follow platforms there will never be an option for a Guest to unfollow a platform.
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used infrequently
Open Issues:	<ol style="list-style-type: none"> 1. Will the platform creator receive a notification if someone unfollows their platform?

Use-Case Number:	2.46
Use-Case Name:	View Platform Followers
Actors:	Guest, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	The user will be brought to a page that lists all of the users that follow that specific platform.
Story:	<ol style="list-style-type: none"> 1. User observes a platform page 2. User left-clicks the “Followers” button 3. User is brought to page that lists all of the users that follow the platform
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe is viewing a Tennis platform page 2. Joe_Doe wants to see all of followers of this Tennis platform 3. Joe_Doe left-clicks the “Followers” button on the platform page which brings them to a page that lists all of the users that follow this Tennis platform page.
Exceptions:	<ol style="list-style-type: none"> 1. If the user has no followers, instead of seeing a list of follow platforms they will see a label that states “User has no followers”
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used intermittently
Open Issues:	None

Use-Case Number:	2.47
Use-Case Name:	Delete Platform (Admin)
Actors:	Admins
Preconditions:	User must be logged into their account, User must be an admin
Postconditions:	The platform the admin has deleted will be removed entirely from the website
Story:	<ol style="list-style-type: none"> 1. Admin observes a platform page 2. Admin left-clicks “Delete platform” button 3. A pop-up appears stating “Are you sure you want to delete this platform?” 4. Admin left-clicks “Yes” 5. Admin is brought back to the home page. The platform will no longer exist meaning it will be accessed by any user
Scenario:	<ol style="list-style-type: none"> 1. Admin decides to permanently delete an inappropriate platform from the website 2. Admin deletes the inappropriate platform 3. Admin left-clicks the “Delete Platform” button on the platform page for the inappropriate platform 4. A pop-up appears stating “Are you sure you want to delete this platform?” 5. Admin left-clicks “Yes” 6. Admin is redirected to the home page. The inappropriate platform will no longer exist meaning it will be accessed by any user
Exceptions:	None
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used infrequently
Open Issues:	None

Use-Case Number:	2.48
Use-Case Name:	View User's Platforms
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	The user will be brought to a page that lists all of the platforms that another user has created
Story:	<ol style="list-style-type: none"> 1. User observes a user's account page 2. User left-clicks the "View All Platforms" button 3. User is brought to page that lists all of the platform that the account owner has created
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe is viewing EvilJoeDoe's account page 2. Joe_Doe wants to see all of EvilJoeDoe's platforms 3. Joe_Doe left-clicks "View All Platforms" button on the account page which brings them to a page that lists all of the platforms EvilJoeDoe has created
Exceptions:	<ol style="list-style-type: none"> 1. If the user has no platforms, instead of seeing a list of platforms the page will say "This user has no platforms"
Priority:	Essential
When available:	Build #3
Frequency of use:	A few times per session
Open Issues:	None

Use-Case Number:	2.49
Use-Case Name:	View User's Quizzes
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	The user will be brought to a page that lists all of the quizzes that another user has created
Story:	<ol style="list-style-type: none"> 1. User observes a user's account page 2. User left-clicks the "View All Quizzes" button 3. User is brought to page that lists all of the quizzes that the account owner has created
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe is viewing EvilJoeDoe's account page 2. Joe_Doe wants to see all of EvilJoeDoe's quizzes 3. Joe_Doe left-clicks "View All Quizzes" button on the account page which brings them to a page that lists all of the quizzes EvilJoeDoe has created
Exceptions:	<ol style="list-style-type: none"> 1. If the user has no quizzes, instead of seeing a list of quizzes the page will say "This user has no quizzes"
Priority:	Essential
When available:	Build #3
Frequency of use:	A few times per session
Open Issues:	None

Use-Case Number:	2.50
Use-Case Name:	View User's Favorited Quizzes
Actors:	Guests, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	The user will be brought to a page that lists all of the quizzes that another user has favorited
Story:	<ol style="list-style-type: none"> 1. User observes a user's account page 2. User left-clicks the "View All Favorited Quizzes" button 3. User is brought to page that lists all of the quizzes that the account owner has favorited
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe is viewing QuizFavoriter10's account page 2. Joe_Doe wants to see all of QuizFavoriter10's favorited quizzes 3. Joe_Doe left-clicks "View All Favorited Quizzes" button on the account page which brings them to a page that lists all of the quizzes QuizFavoriter10 has favorited
Exceptions:	<ol style="list-style-type: none"> 1. If the user has no favorited quizzes, instead of seeing a list of favorited quizzes the page will say "This user has no favorite quizzes"
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used intermittently
Open Issues:	None

Use-Case Number:	2.51
Use-Case Name:	View User's Followed Platforms
Actors:	Guest, Users, Verified Users, Admins
Preconditions:	None
Postconditions:	The user will be brought to a page that lists all of the platforms a user is following
Story:	<ol style="list-style-type: none"> 1. User observes a user's account page 2. User left-clicks the "Following" button 3. User is brought to page that lists all of the platforms that the account owner is following
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe is viewing PlatformFollower10's account page 2. Joe_Doe wants to see all of PlatformFollower10's followed platforms 3. Joe_Doe left-clicks the "Following" button on the account page which brings them to a page that lists all of the platforms PlatformFollower10 is following
Exceptions:	<ol style="list-style-type: none"> 1. If the user has no followed platforms, instead of seeing a list of followed platforms they will see a label that states "[User] is not following any platforms"
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used intermittently
Open Issues:	None

Use-Case Number:	2.52
Use-Case Name:	Edit Account Page
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account
Postconditions:	The user will be shown options that allow them to make adjustments to their account page
Story:	<ol style="list-style-type: none"> 1. User observes their own account page 2. User left-clicks the “Edit Account Page” button 3. Icons appear next to the user’s profile picture, banner, background, bio, and Featured Quizzes section indicating that these elements can be edited. A “Save Changes” button appears at the bottom right of the screen
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe wants to makes adjustments to their account page 2. Joe_Doe left-clicks the “Edit Account Page” button 3. The edit icon appears next to their profile picture, banner, background, bio, and Featured Quizzes. A “Save Changes” button appears at the bottom right of the screen
Exceptions:	<ol style="list-style-type: none"> 1. If the user is a Guest, the “Edit Account Page” button will not appear, as they do not have an account
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used intermittently
Open Issues:	<ol style="list-style-type: none"> 1. Where will we place the “Edit Account” button on the page? 2. How do we place the Edit icons to make them visible but still make the UI look sharp? Instead of Edit icons can we just allow for a user to hover over areas that they want to change?

Use-Case Number:	2.53
Use-Case Name:	Ban User
Actors:	Admins
Preconditions:	User must be logged into their account, User must be an admin
Postconditions:	The banned user's platforms will be taken down and their account will cease to exist
Story:	<ol style="list-style-type: none"> 1. Admin observes user's account page 2. Admin left-clicks "Ban User" button 3. A pop-up appears stating "Are you sure you want to ban this user?" 4. Admin left-clicks "Yes" 5. Admin is brought back to the home page. Banned user's platforms will no longer be viewable, and their quizzes will be removed from any search results
Scenario:	<ol style="list-style-type: none"> 1. Admin decides to permanently ban BadUser14 from the website for consistently being toxic towards other community members 2. Admin opens BadUser14's account page 3. Admin left-clicks the "Ban User" button on BadUser14's account page 4. A pop-up appears stating "Are you sure you want to ban this user?" 5. Admin left-clicks "Yes" 6. Admin is redirected to the home page. BadUser14's account will no longer be viewable, and their quizzes will be removed from any search results
Exceptions:	<ol style="list-style-type: none"> 1. An admin cannot ban themselves. The "Ban User" button will not appear when viewing their own account page
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used infrequently
Open Issues:	<ol style="list-style-type: none"> 1. Where will the ban user button be on the account page? 2. Will there only be an option to permaban users? Or will admins be able to ban a user for a specified amount of time? 3. Will banned user's comments

Use-Case Number:	2.54
Use-Case Name:	Update Account Picture
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account
Postconditions:	User's profile picture will be changed to the new image they upload
Story:	<ol style="list-style-type: none"> 1. User observes their account page 2. User hovers over their profile picture revealing an Edit icon 3. User left-clicks the Edit icon 4. User is prompted to upload a new image for their profile picture 5. User left-clicks "Upload Image" 6. User selects an image from their files to upload 7. User left-clicks the "Save" button to save the change. Their profile picture will change to the image they uploaded.
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe decides they want to change their profile picture to a picture of a duck 2. Joe_Doe hovers over their profile picture revealing an Edit icon 3. Joe_Doe left-clicks the Edit icon, which opens a prompt to upload an image 4. Joe_Doe left-clicks "Upload Image" which opens their File Explorer 5. Joe_Doe scrolls through their files to find a picture of a duck and selects it 6. Joe_Doe left-clicks the "Save" button. Their profile picture changes to the duck picture that they just uploaded
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot upload a profile picture, as they don't have an account. They will instead be given a default profile picture that all guests will have
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Used infrequently
Open Issues:	<ol style="list-style-type: none"> 1. What are the size restrictions on the uploaded image? 2. How will non-square images be handled? 3. What file formats will be allowed? 4. Which API can be used to make uploading images easier?

Use-Case Number:	2.55
Use-Case Name:	Update Account Banner
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account
Postconditions:	User's banner will be changed to the new image they upload
Story:	<ol style="list-style-type: none"> 1. User observes their account page 2. User hovers over their banner image revealing an Edit icon 3. User left-clicks the Edit icon 4. User is prompted to upload a new image for their banner 5. User left-clicks "Upload Image" 6. User selects an image from their files to upload 7. User left-clicks the "Save" button to save the change. Their profile picture will change to the image they uploaded.
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe decides they want to change their banner to a picture of an airplane 2. Joe_Doe hovers over their banner revealing an Edit icon 3. Joe_Doe left-clicks the Edit icon, which opens a prompt to upload an image 4. Joe_Doe left-clicks "Upload Image" which opens their File Explorer 5. Joe_Doe scrolls through their files to find a picture of an airplane and selects it 6. Joe_Doe left-clicks the "Save" button. Their banner image changes to the airplane picture that they just uploaded
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot upload a profile banner, as they don't have an account. They will instead be given a default banner that all guests will have
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used infrequently
Open Issues:	<ol style="list-style-type: none"> 1. What are the size restrictions on the uploaded image? 2. How will non-landscape images be handled? 3. What file formats will be allowed? 4. Which API can be used to make uploading images easier?

Use-Case Number:	2.56
Use-Case Name:	Update Account Background
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account, User must be in the “Editing Account” interface
Postconditions:	User’s background will be changed to the new background they select
Story:	<ol style="list-style-type: none"> 1. User observes their account page 2. User clicks “Edit Account” button, which reveals an Edit icon for the account’s background 3. User left-clicks Edit icon, which opens a small menu for updating the background color/theme of their account page 4. User left-clicks the color/theme they want to use 5. User left-clicks “Save” to save the changes they’ve made. The account’s background changes to reflect the option they selected
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe decides they don’t like their account’s gray background and want to change their background to the color pink 2. Joe_Doe left-clicks “Edit Account” button, revealing an Edit icon for the account’s background 3. Joe_Doe left-clicks Edit icon, revealing a small menu for updating the background color/theme of the account 4. Joe_Doe left-clicks on the color pink, selecting it 5. Joe_Doe left-clicks “Save”. The account’s background changes to the color pink
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot update their background, as they don’t have an account. They will instead be given a default background that all guests will have 2. Verified Users will be given more options for updating their background
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used infrequently
Open Issues:	<ol style="list-style-type: none"> 1. Are the only options available for changing a background provided by the site itself?

Use-Case Number:	2.57
Use-Case Name:	Update Account Bio
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account, User must be in the “Editing Account” interface
Postconditions:	User’s bio will be changed to the new text they enter
Story:	<ol style="list-style-type: none"> 1. User observes their account page 2. User left-clicks “Edit Account” button, which reveals an Edit icon for their account’s bio 3. User left-clicks the Edit icon, which opens a text box 4. User left-clicks the text box and writes their new bio 5. User left-clicks “Save” to save the changes they’ve made. The account’s bio changes to reflect the changes they’ve made
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe decides they want to say a little bit about themselves on their profile rather than having a blank bio 2. Joe_Doe left-clicks “Edit Account” button 3. Joe_Doe left-clicks “text box and types “23y/o alpha male huge fan of video games and cars. looking for SO” 4. Joe_Doe left-clicks “Save” to make the changes visible on their profile
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot update their bio, as they don’t have an account. They will instead be given a default blank bio instead 2. If a user tries to type more characters into their bio than the approved character limit, the site will prevent them from typing further
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #3
Frequency of use:	Used infrequently
Open Issues:	<ol style="list-style-type: none"> 1. What will be the character limit? 2. Will a user be able to add links to their bio?

Use-Case Number:	2.58
Use-Case Name:	Update Featured Platforms
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account, User must be in the “Editing Account” interface
Postconditions:	User’s Featured Platforms section in their account page will now show the new platforms that they’ve selected
Story:	<ol style="list-style-type: none"> 1. User observes their account page 2. User left-clicks “Edit Account” button, which reveals an Edit icon next to the Featured Platform section 3. User left-clicks the Edit icon, which opens a menu containing all of the user’s created platforms 4. User left-clicks the platforms they want to display on their platform page, placing a selection box over each clicked quiz 5. User left-clicks “Save” to save the changes they’ve made. The Featured Platforms section of the platform now displays the platforms that the user has just selected.
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe decides they want to showcase their baseball, football, and tennis platforms on their platform page 2. Joe_Doe left-clicks “Edit Account” button 3. Joe_Doe left -clicks Edit icon next to “Featured Platform” 4. Joe_Doe left-clicks “Baseball Platform”, “Football Platform”, and “Tennis Platform” icons in the menu 5. Joe_Doe left-clicks “Save” to save the changes they’ve made. The Featured Quizzes section of the platform now displays “Baseball Platform”, “Football Platform”, and “Tennis Platform”
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot update their featured platforms, as they don’t have an account page 2. If a user doesn’t have any quizzes, the menu for adding featured platform will be blank
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used intermittently

Use-Case Number:	2.59
Use-Case Name:	Update Featured Quizzes
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account, User must be in the “Editing Account” interface
Postconditions:	User’s Featured Quizzes section in their account page will now show the new quizzes that they’ve selected
Story:	<ol style="list-style-type: none"> 1. User observes their account page 2. User left-clicks “Edit Account” button, which reveals an Edit icon next to the Featured Quizzes section 3. User left-clicks the Edit icon, which opens a menu containing all of the user’s created quizzes 4. User left-clicks the quizzes they want to display on their platform page, placing a selection box over each clicked quiz 5. User left-clicks “Save” to save the changes they’ve made. The Featured Quizzes section of the platform now displays the quizzes that the user has just selected.
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe decides they want to showcase their baseball, football, and tennis quizzes on their platform page 2. Joe_Doe left-clicks “Edit Account” button 3. Joe_Doe left-clicks Edit icon next to “Featured Quizzes” 4. Joe_Doe left-clicks “Baseball Quiz”, “Football Quiz”, and “Tennis Quiz” icons in the menu 5. Joe_Doe left-clicks “Save” to save the changes they’ve made. The Featured Quizzes section of the platform now displays “Baseball Quiz”, “Football Quiz”, and “Tennis Quiz”
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot update their featured quizzes, as they don’t have an account page 2. If a user doesn’t have any quizzes, the menu for adding featured quizzes will be blank
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used intermittently
Open Issues:	<ol style="list-style-type: none"> 1. How many featured quizzes can be displayed at once?

Use-Case Number:	2.60
Use-Case Name:	Edit Platform
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account
Postconditions:	User will enter the interface for editing a platform
Story:	<ol style="list-style-type: none"> 1. User observes Platform Manager page 2. User left-clicks the Edit icon next to the platform they want to make changes to 3. User is taken to the interface for editing a platform
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe wants to make changes to his Football platform 2. Joe_Doe left-clicks the Edit icon next to his Football platform in the Platform Manager Page 3. Joe_Doe is taken to the interface for editing his Football platform
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot edit platforms, as they aren't able to create them
Priority:	Essential
When available:	Build #2
Frequency of use:	Used intermittently
Open Issues:	<ol style="list-style-type: none"> 1. How will we lay out the buttons for editing all the features on the platform?

Use-Case Number:	2.61
Use-Case Name:	Delete Platform
Actors:	Users, Verified Users, Admins
Preconditions:	User must be logged into their account
Postconditions:	User deletes a platform along with any quizzes that existed on the platform
Story:	<ol style="list-style-type: none"> 1. User observes Platform Manager page 2. User left-clicks the Delete icon next to the platform they want to remove 3. A message appears that reads “Are you sure you want to delete this platform”? 4. User left-clicks “Confirm” 5. The platform is removed, along with all quizzes on it. Other platforms in the Platform Manager page are shifted upwards to take the place of the deleted platform.
Scenario:	<ol style="list-style-type: none"> 1. Joe_Doe decides he hates football now and no longer wants to own a platform dedicated to it 2. Joe_Doe left-clicks the Delete icon next to the Football platform in the Platform Manager Page 3. A message appears that reads “Are you sure you want to delete this platform”? 4. Joe_Doe left-clicks “Confirm” 5. Joe_Doe’s football platform is removed, along with all of the quizzes associated with it
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot delete platforms, as they aren’t able to create them 2. If a user clicks “Cancel” when the confirmation message appears, the platform is not deleted.
Priority:	Essential
When available:	Build #2
Frequency of use:	Used infrequently
Open Issues:	<ol style="list-style-type: none"> 1. Will there be a way to recover deleted platforms?

Use-Case Number:	2.62
Use-Case Name:	Update Platform Title
Actors:	Users, Verified Users, Admins
Preconditions:	User is currently editing a platform on the platform editor page.
Postconditions:	User updates the title of the platform
Story:	<ol style="list-style-type: none"> 1. User is viewing the platform editor page displaying a platform that is currently being edited. 2. User left-clicks on the platform title text box 3. User types in a new title into the text box 4. User left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe wants to make sure users know his Football platform is about American Football 2. Joe Doe left-clicks on the already existing title, “Football”, which is a text box because Joe is in the platform editor 3. Joe Doe types in “American Football” in the text box 4. Joe Doe left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot edit platform titles, as they aren’t able to create platforms
Priority:	Essential
When available:	Build #2
Frequency of use:	Many times per account lifetime
Open Issues:	None

Use-Case Number:	2.63
Use-Case Name:	Update Platform Icon
Actors:	Users, Verified Users, Admins
Preconditions:	User is currently editing a platform on the platform editor page.
Postconditions:	User updates the icon of the platform
Story:	<ol style="list-style-type: none"> 1. User is viewing the platform editor page displaying a platform that is currently being edited. 2. User hovers over the platform icon revealing an Edit Icon 3. User left-clicks the Edit Icon 4. A pop up appears where the User can click on “Browse” to select an image from his local files. 5. User selects an image from his local files and left-clicks on upload. 6. The image is uploaded and replaces the old image icon on the platform editor screen. 7. User left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe wants to update the icon for his Basketball platform to be Kobe Bryant 2. Joe Doe hovers over the platform icon revealing an Edit icon 3. Joe Doe left-clicks the Edit icon, opening a prompt to upload an image 4. Joe Doe selects an image of Kobe Bryant from his local files and left-clicks upload 5. Joe Doe left-clicks on “Submit Changes” at the bottom of the editor to publish the changes 6. The platform image is updated to Kobe Bryant
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot edit platform icons, as they aren't able to create platforms
Priority:	Essential
When available:	Build #2
Frequency of use:	Many times per account lifetime
Open Issues:	<ol style="list-style-type: none"> 1. Will there be restrictions on the dimensions of the icons?

Use-Case Number:	2.64
Use-Case Name:	Update Platform Banner
Actors:	Users, Verified Users, Admins
Preconditions:	User is currently editing a platform on the platform editor page.
Postconditions:	User updates the banner of the platform
Story:	<ol style="list-style-type: none"> 1. User is viewing the platform editor page displaying a platform that is currently being edited. 2. User hovers over the platform banner revealing an Edit Icon 3. User left-clicks the Edit Icon 4. A pop up appears where the User can click on “Browse” to select an image from his local files. 5. User selects an image from his local files and left-clicks on upload. 6. The image is uploaded and replaces the old banner on the platform editor screen. 7. User left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe wants to update the banner for his Basketball platform to display the best basketball players of all time 2. Joe Doe hovers over the platform banner revealing an Edit icon 3. Joe Doe left-clicks the Edit icon, opening a prompt to upload an image 4. Joe Doe selects an image of a bunch of Michael Jordans from his local files and left-clicks upload 5. Joe Doe left-clicks on “Submit Changes” at the bottom of the editor to publish the changes 6. The platform banner is updated to a bunch of Michael Jordans
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot edit platform banners, as they aren't able to create platforms
Priority:	Essential
When available:	Build #2
Frequency of use:	Many times per account lifetime
Open Issues:	<ol style="list-style-type: none"> 1. Will there be restrictions on the dimensions of the banners?

Use-Case Number:	2.65
Use-Case Name:	Update Platform Description
Actors:	Users, Verified Users, Admins
Preconditions:	User is currently editing a platform on the platform editor page.
Postconditions:	User updates the description of the platform
Story:	<ol style="list-style-type: none"> 1. User is viewing the platform editor page displaying a platform that is currently being edited. 2. User left-clicks on the platform description text box 3. User makes any appropriate changes to the platforms description 4. User left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe wants to make sure his platform description matches his new platform title, “American Football” 2. Joe Doe left-clicks on the already existing description which is a text box because Joe is in the platform editor 3. Joe Doe types “American” in front of any instances of “Football” 4. Joe Doe left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot edit platform descriptions, as they aren’t able to create platforms
Priority:	Essential
When available:	Build #2
Frequency of use:	Many times per account lifetime
Open Issues:	None

Use-Case Number:	2.66
Use-Case Name:	Add Quiz to Platform
Actors:	Users, Verified Users, Admins
Preconditions:	User is currently editing a platform on the platform editor page.
Postconditions:	The quiz the user has selected is added to the platform
Story:	<ol style="list-style-type: none"> 1. User is viewing the platform editor page displaying a platform that is currently being edited. 2. User left-clicks the “Add Quiz” button, which opens a menu containing all of the user’s created quizzes 3. User left-clicks the quiz they want to add to their platform page 4. User left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe decides they want to add a new quiz to their football platform 2. Joe Doe left-clicks the “Add Quiz” button, bringing up a menu of all of Joe Doe’s quizzes 3. Joe Doe left-clicks on the “Football Stats” quiz, which adds the quiz to the platform 4. Joe Doe left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot add quizzes to their platform, as they aren’t able to create platforms 2. If a user doesn’t have any quizzes, the menu for adding featured quizzes will be blank
Priority:	Essential
When available:	Build #2
Frequency of use:	Used intermittently
Open Issues:	None

Use-Case Number:	2.67
Use-Case Name:	Remove Quiz from Platform
Actors:	Users, Verified Users, Admins
Preconditions:	User is currently editing a platform on the platform editor page.
Postconditions:	The quiz the user has selected is removed from the platform
Story:	<ol style="list-style-type: none"> 1. User is viewing the platform editor page displaying a platform that is currently being edited. 2. User hovers over a quiz 3. User left-clicks the remove icon which removes the quiz from the platform 4. User left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Scenario:	<ol style="list-style-type: none"> 1. Joe Doe decides they want to remove the “Football Stats” quiz because it is no longer good enough for Joe’s standards 2. Joe Doe hovers over the “Football Stats” quiz which brings up a remove icon 3. Joe Doe left-clicks on the remove icon which removes the “Football Stats” quiz from the platform 4. Joe Doe left-clicks on “Submit Changes” at the bottom of the editor to publish the changes
Exceptions:	<ol style="list-style-type: none"> 1. Guest users cannot remove quizzes from their platform, as they aren’t able to create platforms in the first place
Priority:	Essential
When available:	Build #2
Frequency of use:	Used intermittently
Open Issues:	None

Use-Case Number:	2.68
Use-Case Name:	Edit Quiz
Actors:	Users, Verified Users
Preconditions:	User is on the quiz manager and has at least one published quiz.
Postconditions:	User is taken to a page where he can edit the selected quiz.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz manager screen and sees a quiz that needs to be edited. 2. User left-clicks on the edit icon next to the quiz to be edited. 3. User is redirected to a quiz editor page where the selected quiz is ready to be edited.
Scenario:	<ol style="list-style-type: none"> 1. The NBA account would like to update their quiz “Facts about Giannis Antetokounmpo” to include questions about his most recent Finals run. 2. The NBA account clicks on the edit icon next to the “Facts about Giannis Antetokounmpo” quiz. 3. The NBA account is redirected to the quiz editor page where they can edit the Giannis quiz.
Exceptions:	<ol style="list-style-type: none"> 1. Users that have not published any quizzes will not be able to edit quizzes.
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Several Times per Lifetime
Open Issues:	<ol style="list-style-type: none"> 1. It must be determined how leaderboards will be impacted by quiz edits. Will leaderboards be completely reset or will they be left as is. Most likely, it will depend on the severity of the edits.

Use-Case Number:	2.69
Use-Case Name:	Delete Quiz
Actors:	Users, Verified Users, Admin
Preconditions:	Quiz to be deleted belongs to the user OR quiz deleter is an admin. User must have at least one published quiz.
Postconditions:	The quiz to be deleted is removed and no longer visible to any users
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz manager screen and sees a quiz that needs to be deleted. 2. User left-clicks on the delete icon next to the quiz to be deleted. 3. A confirmation box pops up saying “Are you sure you want to delete this quiz?”. 4. User left-clicks on the confirm button. 5. The confirmation box disappears and the quiz to be deleted is removed from the quiz manager. All quizzes below are moved up to take its place.
Scenario:	<ol style="list-style-type: none"> 1. Marvel accidentally published their quiz “Spiderman: No Way Home Character Quiz” before the movie was released in theaters. The quiz maker was tasked with deleting the quiz ASAP. 2. The quiz maker for Marvel looks through the listing of Marvel quizzes on the quiz manager and finds the Spiderman quiz. 3. The quiz maker left-clicks on the delete icon next to the quiz. 4. The quiz maker left-clicks the confirm button on the confirmation box. 5. The Spiderman quiz is deleted and the quiz maker gets to keep his job.
Exceptions:	<ol style="list-style-type: none"> 1. If the quiz is unable to be deleted because of server problems, a message will appear reading “Quiz failed to be deleted”.
Priority:	Essential, must be implemented.
When available:	Build #3
Frequency of use:	Used Infrequently
Open Issues:	<ol style="list-style-type: none"> 1. If a quiz is deleted while someone is currently taking the quiz how will we handle that?

Use-Case Number:	2.70
Use-Case Name:	Update Quiz Image
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	The image for the quiz being edited is updated for all users to see.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User clicks on the text “Update Quiz Image” which is displayed below the icon of the current quiz image. 3. A pop up appears where the User can click on “Browse” to select an image from his local files. 4. User selects an image from his local files and left-clicks on upload. 5. The image is uploaded and replaces the old image icon on the quiz editor screen. 6. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. NASA notices that their current image for the quiz “Planet Facts” includes Pluto. They would like to update the picture so that it does not include Pluto anymore. 2. NASA’s appointed user is currently editing the quiz. 3. NASA’s quiz maker clicks on the text “Update Quiz Image”. 4. The quiz maker selects an image of the planets without Pluto and clicks on upload. 5. The quiz maker clicks on “Submit Changes”. 6. The quiz image is updated and Pluto is gone from the picture.
Exceptions:	<ol style="list-style-type: none"> 1. If an image fails to be uploaded because of server problems, a message will appear reading “Image failed to be uploaded”
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Used Infrequently
Open Issues:	<ol style="list-style-type: none"> 1. Will we have restrictions on resolution and the format of the images being uploaded? 2. What API will be used for image uploading and how will we store them in our database.

Use-Case Number:	2.71
Use-Case Name:	Update Quiz Name
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	The name of the quiz being edited is updated for all users to see.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User clicks on the text field containing the current name of the quiz. 3. User types in the new name he wants to replace the previous name of the quiz. 4. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. The Scripps National Spelling Bee has a typo in their quiz “Spelling Bee Winner of Every Yaer”. They would like to fix this typo. 2. The quiz-maker in charge is currently editing the quiz on the quiz editor page. 3. The quiz-maker types into the name textfield “Spelling Bee Winner of Every Year” 4. The quiz-maker clicks on “Submit Changes” and the new name of the quiz is published.
Exceptions:	<ol style="list-style-type: none"> 1. Names must be limited in characters so that quiz names do not take up more than one line on any page of the application.
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Infrequently
Open Issues:	<ol style="list-style-type: none"> 1. Will we use an API to verify that the names do not break user guidelines?

Use-Case Number:	2.72
Use-Case Name:	Add Quiz Question
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	The new question is added to the quiz for all users to see.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User clicks on the text “Add Quiz Question” located directly below the last question of the quiz. 3. Text boxes for a new question and answers are appended at the end with default values. 4. Users update these fields with the new questions and answer choices. 5. User left clicks the “Correct Answer” button next to the correct choice to mark it as the correct answer. 6. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. User “Mathwiz01” would like to add another integral to his quiz “Hardest Calculus Quiz of All Time”. 2. “Mathwiz01” is currently editing the quiz in the quiz editor page. 3. “Mathwiz01” clicks on the text “Add Quiz Question”. 4. “Mathwiz01” updates the new fields with the new question and possible answer choices. 5. “Mathwiz01” left-clicks the “Correct Answer” button next to the correct choice to mark it as the correct answer. 6. “Mathwiz01” clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Exceptions:	<ol style="list-style-type: none"> 1. Users cannot select multiple answer choices for a question.
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Frequently
Open Issues:	None

Use-Case Number:	2.73
Use-Case Name:	Delete Quiz Question
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	The question is deleted from the quiz.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User clicks on the trash icon next to the question they wish to delete. 3. A confirmation box appears that says “Confirm Deletion?”. 4. Users click on confirm and the box disappears along with the question. All questions below are shifted up. 5. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. “Mathwiz01” would like to delete an integral from his quiz because he feels that it is too easy. 2. “Mathwiz01” clicks on the trash icon next to the simple integral question. 3. “Mathwiz01” clicks on confirm on the popup box. The question disappears. 4. “Mathwiz01” clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Exceptions:	<ol style="list-style-type: none"> 1. Users must have at least one question for their quiz so the final question cannot be deleted. The trash icon is disabled in this case.
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Infrequently
Open Issues:	<ol style="list-style-type: none"> 1. How will deleting a question affect leaderboards that show how many questions a user got correct?

Use-Case Number:	2.74
Use-Case Name:	Update Quiz Question Title
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	The question on the quiz is updated for all users to see.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User finds the field-box for the question title to update. 3. User types in the changes to the question. 4. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. User “Mathwiz01 ” made a mistake on one of his derivatives and needs to update the question title accordingly. 2. “Mathwiz01” is currently editing the quiz in the quiz editor page. 3. “Mathwiz01” clicks on the text-field for the question to update. 4. “Mathwiz01” updates the field to read “Derivative of x?” 5. “Mathwiz01” clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Exceptions:	<ol style="list-style-type: none"> 1. Questions have a character limit that cannot be exceeded
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Infrequently
Open Issues:	<ol style="list-style-type: none"> 1. How will updating questions affect leaderboards?

Use-Case Number:	2.75
Use-Case Name:	Update Quiz Answer Choices
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	The answer choices for the question are updated for all users to see.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User finds the field-boxes for the answers to update. 3. User types in the changes to the answer choices. 4. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. User “Mathwiz01 ” made a mistake in one of the answer choices to his integral question and needs to update the answer choice accordingly. 2. “Mathwiz01” is currently editing the quiz in the quiz editor page. 3. “Mathwiz01” clicks on the text-field for the answer choice to update. 4. “Mathwiz01” updates the field to read “x^2” 5. “Mathwiz01” clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Exceptions:	<ol style="list-style-type: none"> 1. Answer choices should have a character limit that cannot be exceeded.
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Infrequently
Open Issues:	<ol style="list-style-type: none"> 1. How will updating answer choices affect leaderboards?

Use-Case Number:	2.76
Use-Case Name:	Update Quiz Answer
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	The answer for the question is updated for all users to see.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User finds the answer choices for the question to update. 3. User clicks on the check box next to the correct answer. 4. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. User “Mathwiz01 ” made a mistake for the answer to one of his integral questions and needs to update the answer accordingly. 2. “Mathwiz01” is currently editing the quiz in the quiz editor page. 3. “Mathwiz01” finds the answer choices to the question to update. 4. “Mathwiz01” clicks on the check box next to the third answer. 5. “Mathwiz01” clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Exceptions:	<ol style="list-style-type: none"> 1. There can only be one correct answer per question. Once a checkbox is selected all other checkboxes are blank.
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Infrequently
Open Issues:	<ol style="list-style-type: none"> 1. How will updating answers affect leaderboards?

Use-Case Number:	2.77
Use-Case Name:	Update Quiz Timer
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	Quiz timer is updated for all users to see.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User clicks on the “Quiz Timer” field which allows users to update the number of minutes and number of seconds for the quiz. 3. User updates the number of minutes and seconds for the quiz. 4. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. User “Mathwiz01” wants to allow users to have more time for his extremely difficult quiz. 2. “Mathwiz01” is currently editing the quiz in the quiz editor page. 3. “Mathwiz01” clicks on the “Quiz Timer” field. He updates the timer to have 10 more minutes. 4. “Mathwiz01” clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Exceptions:	<ol style="list-style-type: none"> 1. Quiz timer cannot exceed 1 hour and must be a positive value.
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Infrequently
Open Issues:	<ol style="list-style-type: none"> 1. How will this be implemented in conjunction with the question timer? If the user has a quiz timer, will they be allowed to have a timer for each question?

Use-Case Number:	2.78
Use-Case Name:	Update between Instant and Standard Answers
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	In instant format, quiz answers are revealed after every question, while in standard format, quiz answers are revealed after completion of quiz.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User sees the toggle between instant format and standard format. 3. User clicks on the format that fits the quiz. 4. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. User “Mathwiz01” wants to allow users to have instant feedback to his answers. 2. “Mathwiz01” is currently editing the quiz in the quiz editor page and sees the toggle between instant and standard format. Currently standard is selected. 3. “Mathwiz01” clicks on the “Instant” format button. 4. “Mathwiz01” clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Exceptions:	<ol style="list-style-type: none"> 1. Users have to choose one format, they cannot choose both.
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Infrequently
Open Issues:	<ol style="list-style-type: none"> 1. For instant format quizzes, will users be able to navigate to previous questions. One way we could do this is allow users to go back to view questions without allowing changes to answers.

Use-Case Number:	2.79
Use-Case Name:	Update between Shuffled and Ordered Questions
Actors:	Users, Verified Users
Preconditions:	User is currently editing a quiz on the quiz editor page.
Postconditions:	In shuffled format, quiz questions are given at random, while in ordered format, quiz questions are given sequentially.
Story:	<ol style="list-style-type: none"> 1. User is viewing the quiz editor page displaying a quiz that is currently being edited. 2. User sees the toggle between shuffled format and ordered format. 3. User clicks on the format that fits the quiz.. 4. User clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Scenario:	<ol style="list-style-type: none"> 1. User “Mathwiz01” wants to shuffle the order his calculus questions are revealed. 2. “Mathwiz01” is currently editing the quiz in the quiz editor page and sees the toggle between shuffled and ordered format. Currently ordered format is selected. 3. “Mathwiz01” clicks on the shuffle format button. 4. “Mathwiz01” clicks on “Submit Changes” on the bottom of the editor to publish his changes.
Exceptions:	<ol style="list-style-type: none"> 1. Users have to choose one format, they cannot choose both.
Priority:	Essential, must be implemented
When available:	Build #3
Frequency of use:	Used Infrequently
Open Issues:	None

Use-Case Number:	2.80
Use-Case Name:	Select Item From Shop
Actors:	Users, Verified Users
Preconditions:	User is currently viewing the shop screen.
Postconditions:	Purchase screen is opened for the selected item.
Story:	<ol style="list-style-type: none"> 1. User is viewing the reward screen which lists out all the rewards available. 2. Users click on a reward. 3. User is redirected to the purchase screen for the reward
Scenario:	<ol style="list-style-type: none"> 1. User “Skaterdude17” is viewing the rewards. He sees an animated banner of a skater doing a kickflip. 2. “Skaterdude17” clicks on the animated banner. 3. “Skaterdude17” is redirected to the purchase screen for the banner.
Exceptions:	<ol style="list-style-type: none"> 1. Some rewards may be unavailable or out of stock. These rewards will be disabled and highlighted in grey.
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Several Times per Lifetime
Open Issues:	None

Use-Case Number:	2.81
Use-Case Name:	Preview Item
Actors:	Users, Verified Users
Preconditions:	User is currently viewing the purchase screen for the reward they selected.
Postconditions:	A preview of the reward on a sample platform is displayed.
Story:	<ul style="list-style-type: none"> 1. User is currently viewing the reward on the purchase screen. 2. User clicks on the button “Preview Reward”. 3. User is redirected to a sample platform page with a preview of their reward.
Scenario:	<ul style="list-style-type: none"> 1. User “Skaterdude17” is viewing the animated skateboard banner on the purchase page. 2. “Skaterdude17” clicks on “Preview Reward”. 3. “Skaterdude17” is redirected to a sample platform page with a preview of the animated banner on display.
Exceptions:	<ul style="list-style-type: none"> 1. Some rewards do not contain previews. Generally only rewards that customize a platform will have a preview button.
Priority:	Wishlist
When available:	Build #5
Frequency of use:	Several Times per Lifetime
Open Issues:	<ul style="list-style-type: none"> 1. If a reward does not have a preview available, what should be in place of the “Preview Reward” button. Would it make sense to have nothing there or just a disabled version of the button?

Use-Case Number:	2.82
Use-Case Name:	Purchase Item
Actors:	Users, Verified Users
Preconditions:	User is currently viewing the purchase screen for the reward they selected. They must have enough currency to purchase the item.
Postconditions:	The item is purchased and the user's currency is updated to match the purchase.
Story:	<ol style="list-style-type: none"> 1. User is currently viewing the reward on the purchase screen. 2. User clicks on the button "Purchase Reward". 3. The popup box appears saying "Are you sure you would like to purchase this item?" 4. User clicks on confirm. The item is purchased and the user's currency is subtracted by the cost of the item.
Scenario:	<ol style="list-style-type: none"> 1. User "Skaterdude17" is viewing the animated skateboard banner on the purchase page. 2. "Skaterdude17" clicks on "Purchase Reward". 3. "Skaterdude17" now owns the animated skateboard banner and the cost of the banner is subtracted from his currency.
Exceptions:	<ol style="list-style-type: none"> 1. If the user does not have enough item to purchase the reward
Priority:	Value Adding, significantly improves the app but could work without it
When available:	Build #4
Frequency of use:	Several Times per Lifetime
Open Issues:	<ol style="list-style-type: none"> 1. How will our currency system work? What constitutes a user gaining currency?

Use-Case Number:	2.83
Use-Case Name:	Update User Display Name
Actors:	Users, Verified Users
Preconditions:	User is currently on the settings page showing user information
Postconditions:	User's display name is updated to the new name they inputted
Story:	<ol style="list-style-type: none"> 1. User is currently viewing the account information section of the settings page. 2. User clicks on the text field for the display name. 3. User enters a new display name. 4. User clicks on "Submit Changes"
Scenario:	<ol style="list-style-type: none"> 1. User "Skaterdude17" is viewing the account information section of the settings page. He would like to update his display name. 2. "Skaterdude17" clicks on the display name field. 3. "Skaterdude17" types into the text field "Skaterdude18". 4. "Skaterdude17" clicks on save changes. 5. "Skaterdude17" is now "Skaterdude18".
Exceptions:	<ol style="list-style-type: none"> 1. Display name must be a unique name that no user in the database already has. 2. If the user submits his original display name, an alert will popup saying "No changes have been made".
Priority:	Essential, must be implemented.
When available:	Build #3
Frequency of use:	Several Times per Lifetime
Open Issues:	None

Use-Case Number:	2.84
Use-Case Name:	Change between Dark Mode and Light Mode
Actors:	Users, Verified Users
Preconditions:	User currently on the settings page.
Postconditions:	The mode previously chosen is switched to the new mode of display.
Story:	<ul style="list-style-type: none"> 1. User is currently on the settings page. 2. User sees the slider between light mode and dark mode. 3. User clicks on the slider and the mode is changed.
Scenario:	<ul style="list-style-type: none"> 1. User “ComputerGeek” does not like the brightness of lightmode. He goes to his settings page. 2. “ComputerGeek” sees the slider is set to light mode. He clicks on the slider to set the display to dark mode. 3. The user interface goes into dark mode dimming the brightness of the screen.
Exceptions:	None
Priority:	Wishlist
When available:	Build #5
Frequency of use:	Once or twice per lifetime.
Open Issues:	<ul style="list-style-type: none"> 1. Most likely will need a front-end API to make the switch between modes seamless.

Use-Case Number:	2.85
Use-Case Name:	Delete User Account
Actors:	Users, Verified Users
Preconditions:	None
Postconditions:	User account is deleted and all content from the user is removed from the site.
Story:	<ol style="list-style-type: none"> 1. User is currently on the settings page. 2. User clicks on the “Delete Account” button found at the bottom of the page. 3. User is redirected to another page with a warning about the repercussions of deleting an account. 4. User clicks on the “Proceed with Account Deletion” button found at the bottom of the warning. 5. A confirmation button appears with the text “FINAL WARNING: ARE YOU SURE YOU WOULD LIKE TO DELETE YOUR ACCOUNT”. 6. User clicks on “Yes”. User account is signed out and deleted. 7. User is redirected as a guest back to the homepage.
Scenario:	<ol style="list-style-type: none"> 1. User “QuizLover101” hates the website. He does not want to be an account holder any longer. 2. User “QuizLover101” clicks on the “Delete Account” button on the settings page. 3. User “QuizLover101” continues to confirm deletion. He presses “yes” on the final confirmation box. 4. User “QuizLover101”’s account is deleted and he is signed out.
Exceptions:	<ol style="list-style-type: none"> 1. If the account is unable to be deleted because of server problems, a message will appear reading “Account failed to be deleted”.
Priority:	Essential, must be implemented.
When available:	Build #4
Frequency of use:	Once per account lifetime.
Open Issues:	<ol style="list-style-type: none"> 1. Will we have a way to recover accounts? Perhaps a time limit before the account information is wiped out of the database.

3. UI Model

3.1 User Interface View Listing

Use Case #	Name	Description
3.2.1	Home Page	Page all users initially go to when entering the site, which displays featured and recommended platforms and quizzes, as well as a timeline that shows new content from platforms/users that the user is following
3.2.2	Quiz Categories Page	Page that lists popular quiz categories, and takes users to a page that displays quizzes relating to these topics
3.2.3	Search Results Page	Page which displays all quizzes or platforms that match search parameters
3.2.4	Pre-Quiz Page	Page with information about the quiz the user is about to take and a button to start the quiz
3.2.5	Quiz Taking Page	Page displayed when taking a quiz that includes quiz questions, a clock and navigation between questions
3.2.6	Post-Quiz Results Page	Page displayed after taking a quiz that displays the user's score, stats, and rewards as well as the leaderboard
3.2.7	Post-Quiz Answers Page	Page displayed after taking a quiz that shows the answers for the questions
3.2.8	Login Page	Page with a link to a Google SSO to authenticate the user
3.2.9	Sign Up Page	Page that lets the user choose their display name on their registration
3.2.10	Settings Page	Page that allows a user to edit various settings, such as display name, theme, and profile picture
3.2.11	Quiz Editing Page	Page where a platform owner can edit a quiz, by changing the questions, answers, or quiz options

3.2.12	Platform Editing Page #1	Platform state where a platform owner can manage their platform to feature more of their quizzes to visitors or change aesthetics
3.2.13	Platform Editing Page #2	Platform state where a platform owner can add a playlist or change aesthetics
3.2.14	Platform Editing Page #3	View where a platform owner is adding a featured quiz to a featured playlist in their platform.
3.2.15	Platform Page	Page shown to visitors including featured quizzes, navbar, follower count and a follow button
3.2.16	Account Editing Page #1	View where an account owner cannot feature any more platforms or quizzes while editing their account
3.2.17	Account Editing Page #2	View seen where more platforms can be featured on the user's account while editing it
3.2.18	Account Editing Page #3	View seen when the user is in the process of featuring a platform on their account while editing it
3.2.19	Account Page	Page seen by users visiting the account of another user
3.2.20	List of all Platforms Page	Page displaying all the platforms of the account owner
3.2.21	List of all Quizzes Page	Page displaying all the quizzes of the account owner
3.2.22	List of Badges Page	Page displaying all the badges of the account owner
3.2.23	Shop Page	Page displaying decorative elements for accounts/profiles which can be bought for in-game currency.
3.2.24	Purchase Page	Page asking for confirmation of the purchase of a store item.
3.2.25	Preview Page	Preview of decorative item on the user's own account page

3.2 User Interface View Listing

3.2.1 Home Page

Trivia Tree, Category ▾ Search for a quiz... Quiz ▾ Categories ⚡ Create Account

Featured Quizzes

 Sports Footballfan20	 Mario Quiz MarioTime	 Computer Parts gamerster11	 Western Animation Clint S	 Music TravisFan	 Celeste Levels Maddy Linn	 Anime Dinoman
--	--	--	---	---	---	---

What's New

 Slink posted a new Quiz: Childhood Cartoons Trivia
 Cronker created a new Platform: Gamer Quizzes
 Nintendo Platform posted a new Quiz: Name that Pokemon!
 Mythology posted a new Quiz: Roman Gods
 Kanye West Fans posted a new Quiz: Kanye's Top 10 Songs of 2021
 Wumbo Jumbo posted a new Quiz: Hand Bones
 Jason21 created a new Platform: My First Platform
 Italian Cuisine posted a new Quiz: Gourmet Pastas

Featured Platforms

 Football Footballfan20	 Best Baseball Quizzes Footballfan20	 Technology gamer man	 Japanese Food ChrisSteeb	 Best Cartoons Footballfan20	 Horses TheFlyingTree	 Classical Painting VanBro200
--	--	--	---	--	--	---

Recommended Quizzes

 Football Footballfan20	 Pro Sports Footballfan20	 Classic Music Footballfan20	 Disney Footballfan20	 Cuisine Footballfan20	 Best Cereals Footballfan20	 Plants Footballfan20
--	--	---	--	---	--	--

3.2.2 Categories Page

Trivia Tree, Category ▾ Search for a quiz... Quiz ▾ Categories ⚡ Create Account

Categories

 Sports	 Video Games	 Technology	 Food	 Music	 Anime
 Art	 Television	 Vehicles	 Lifestyle	 Travel	 History
 Cars	 Science	 Computers	 Books	 Animation	 Geography

3.2.3 Search Results Page

Trivia Tree, Music ▾ video game music Quiz ▾  Create Account

Search Results for “video game music” Quizzes

Filters

- Minimum Rating: ★★☆☆☆
- Difficulties: All Difficulties ▾
- Quiz Types: Standard ▾
- Timer: Any Time Limit ▾

	Celeste Songs Do you know these classic Celeste songs? Take this quiz to find out!! Difficult ⚙️ ⏳ 10:00	★ 5	Celeste	 CelesteGamer200
	Undertale Music Quiz Guess this selection of over 100+ Undertale songs. Intermediate ⚙️ ⏳ ∞	★ 4.2	Undertalers	 gamer man
	Legend of Zelda Music Songs from Ocarina of Time, Windwaker, and Link's Awakening. Easy ⚙️ ⏳	★ 2	TravisZeldaPlatform	 Travis Activision
	Nintendo Music Music from Nintendos biggest games (Mario, Zelda, and many more!) Easy ⚙️ ⏳ 30:00	★ 1.4	Nintendo	 NintendogsGuy12
	Do You Know Persona Songs? Only true Persona fans will be able to guess all of these songs Difficult ⚙️	★ 4	Persona5Place	 PersonerFan
	Donkey Kong Country Beats Music quiz from the ORIGINAL DKC and DKC2 for the Gameboy Intermediate ⚙️	★ 5	DonkeyKongMusic	 DoubleEdgedFrog

3.2.4 Pre-Quiz Page

Trivia Tree, Category ▾ Search for a quiz... Quiz ▾  Categories  Create Account

Nintendo Music Quiz

We all remember listening to Mario Bros for the DS as kids. But can you remember all of the songs? Take this quiz to find out!!

	4 Minutes	Shuffled Questions
	5 Questions	Standard Quiz Type
Back to Search	Start Quiz	

 **Creator**
MarioGamer100

1,263 Plays
21 Favorites

3.2.5 Quiz Taking Screen Page

The screenshot shows a quiz interface. On the left, there's a sidebar with the Nintendo logo and the title "Nintendo Music Quiz" by "MarioGamer100". A 2x7 grid table shows question numbers 1 through 14. Questions 1, 3, 5, 7, 9, 11, and 13 have answers listed: 1. Lakitu Jam, 3. Pipe Theme Song, 5. The Mario Brothers, and 13. Whomp Town 3000. Question 2 is empty. The current question is 1, asking "What is the song that plays at Whomp's Fortress?". The options are A. Lakitu Jam, B. Pipe Theme Song, C. The Mario Brothers, and D. Whomp Town 3000. A timer at the top shows 00:11:833. At the bottom right is a "Next Question" button.

3.2.6 Post-Quiz Page (Results)

The screenshot shows the post-quiz results page for the "Nintendo Music Quiz" created by "MarioGamer100". The quiz stats are: You Scored: 7/14 = 50%, average score is 4, and you did better than 32% of test takers. There are five yellow star rating icons. To the right, a clock icon shows 3 Minutes 27 Seconds, a clipboard icon shows "Attempt Number 1", and a coin icon shows "20 Earned". Below these are two buttons: "View Results" and "View Answers". On the right, there's a "Quiz Leaderboard" table:

Rank	User	Score
1.	CelesteGamer200	100%
2.	DoubleEdgedFrog	100%
3.	TheAnonymousMan	100%
4.	RedDeadSoul21	100%
5.	Love1D_101	80%
6.	IceHoopz	80%
7.	BarrackHusseinObama	80%
8.	asdfghjkl	60%

3.2.7 Post-Quiz Page (Answers)

Trivia Tree Category ▾ Search for a quiz... Quiz 🔍 Categories Cart ⋮ Create Account

Nintendo Music Quiz

Creator MarioGamer100

Retry Quiz

1. What is the song that plays at Whomp's Fortress?

2. What is the most popular Nintendo song?

A. Super Smash Bro's Brawl Main Theme
B. Bean Bean Kingdom
C. Velvet Kingdom
D. Peach Kingdom

3. Which song plays during Mario's fight with Bowser?

[View Results](#) [View Answers](#)

Quiz Leaderboard

1.	CelesteGamer200	100%
2.	DoubleEdgedFrog	100%
3.	TheAnonymousMan	100%
4.	RedDeadSoul21	100%
5.	Love1D_101	80%
6.	IceHoopz	80%
7.	BarrackHusseinObama	80%
8.	asdfghjkl	60%

3.2.8 Login Page

Trivia Tree Category ▾ Search for a quiz... Quiz 🔍 Categories Cart ⋮ Create Account

Where the world takes quizzes

A site designed and developed with both quiz creators and quiz takers in the forefront. Experience the elegance of a elegant and responsive UI while expressing your knowledge when creating and taking quizzes.

Sign in with Google



3.2.9 Sign Up Page

The screenshot shows a dark-themed web interface. At the top, there's a navigation bar with 'Trivia Tree' on the left, followed by 'Category' and a search bar containing 'Search for a quiz...'. To the right of the search bar are 'Quiz' and a magnifying glass icon. Further right are icons for a shopping cart, three dots, and 'Create Account'. Below the navigation bar, the main content area has a light gray background. It features a text input field containing 'SBU_Student' with a green circular progress bar to its right. Below this is a large blue 'Confirm' button.

Choose A Display Name

(Display names can be changed later in the settings)



Confirm

3.2.10 Settings Page

The screenshot shows a dark-themed web interface. At the top, there's a navigation bar with 'Trivia Tree' on the left, followed by 'Category' and a search bar containing 'Search for a quiz...'. To the right of the search bar are 'Quiz' and a magnifying glass icon. Further right are 'Categories' and a shopping cart icon. On the far right is a user profile icon labeled 'SBU_Fan' with a small American flag icon. Below the navigation bar, the main content area has a light gray background. It features a large bold 'Settings' heading. Underneath, there are several input fields and buttons: a 'Display Name:' field with 'SBU_Fan', an 'Email:' field with 'quizztakersemail@gmail.com', a 'Theme:' section with radio buttons for 'Light Mode' (selected) and 'Dark Mode', a 'Profile Picture:' section with a placeholder image of a dog wearing a cap and a 'Upload File' button, and two large red buttons at the bottom: 'Delete Account' on the left and 'Save Changes' on the right.

Settings

Display Name: SBU_Fan

Email: quizztakersemail@gmail.com

Theme: Light Mode Dark Mode



Profile Picture:

3.2.11 Quiz Editing Page

Trivia Tree, Category ▾ Search for a quiz... Quiz ▾ Categories ⚡ SBU_Fan 🇺🇸

SBU Game Developers

Quiz Title *
This is a quiz about the game developers club at Stony Brook, a club I recently joined. How many questions can you get right?

Quiz Description

Platform: None

Quiz Icon: clubicon.png 

Question 1

Which game won the 2021 Game Programming Competition?

A. Hell's Fury
B. Dr. Botany
C. Time Raider
D. Twin

+ Add New Choice

Go To Question

Options

Quiz Type: Standard Instant

Question Ordering: Ordered Shuffled

Quiz Timer Format: HHMMSS
00:30:00

Question Timer Format: HHMMSS
00:30:00

3.2.12 Platform Editing Page #1

Trivia Tree, Category ▾ Search for a quiz... Quiz ▾ Peter2000 🇺🇸

Stony Brook Platforms Quizzes Badges

 Select Effect

Stony Brook 124 Followers

Subjects

 Science  Math  Classic Music  Coding  Music

Sports

 Football  E-Sports  Soccer  Baseball  Basketball  Volleyball  Tennis  E-Sports

3.2.13 Platform Editing Page #2

The screenshot shows the Trivia Tree platform editing interface for the 'Stony Brook' account. At the top, there's a navigation bar with links for 'Category', 'Search for a quiz...', 'Quiz', 'Categories', and user profile 'SBU_Fan'. Below the navigation is a banner with the text 'STRONGER TOGETHER' and a large red heart icon. On the left, there's a circular profile picture of Stony Brook with a red and white design, followed by the account name 'Stony Brook' and '124 Followers'. Below this is a 'Sports' section with icons for Football, Basketball, Soccer, Baseball, Basketball, Volleyball, Tennis, and E-Sports. A 'Create playlist' button is also present. On the right side of the banner, there are buttons for 'Select Effect' and 'Upload File', and a 'Follow' button.

3.2.14 Platform Editing Page #3

This screenshot shows the same Trivia Tree platform editing interface for 'Stony Brook', but with a modal window open. The modal is titled 'Select a quiz to add:' and contains a search bar with 'SBU Game Dev' and a list item 'SBU Game Developers'. It includes 'Cancel' and 'Add' buttons. The rest of the page is identical to the previous screenshot, featuring the 'STRONGER TOGETHER' banner, the 'Sports' section, and the 'Create playlist' button.

3.2.15 Platform Page

The screenshot shows the Trivia Tree platform page for the user "Stony Brook". At the top, there's a navigation bar with links for "Category", "Search for a quiz...", "Quiz", "Categories", "Cart", "More", and "SBU_Fan". Below the navigation is a banner with the text "STRONGER TOGETHER" and a red heart icon. On the left, there's a profile section for "Stony Brook" with 124 followers, featuring a circular logo with a red and white design. To the right of the profile is a "Follow" button. Below this, there's a section titled "Subjects" with icons for Science, Math, Classic Music, Coding, and Music. Further down is a "Sports" section with icons for Football, Basketball, Soccer, Baseball, Hockey, Volleyball, Tennis, and E-Sports. Navigation arrows are present on either side of the sports icons.

3.2.16 Account Editing Page #1

The screenshot shows the account editing page for the user "SBU_Fan". The top navigation bar is identical to the one in the previous screenshot. The main area features a profile card for "SBU_Fan" (Gamer / Quiz Taker) with options to "Select Effect" or "Upload File". Below this are two sections: "Featured Platforms" (Technology, Japanese Food, Best Cartoons, Horses, Classical Painting) and "Featured Quizzes" (Mario Quiz, Computer Parts, Western Animation, Music, Celeste Levels). On the right, there's a "Biography" section with the text "I love among us. Among Us is a GOATED (Game Of The Yeared) game.", a "Badges" section showing three earned badges ("Who's that Faculty?", "Stony Brook D1 Quiz"), and a summary of "3 Badges Earned". At the bottom right are "Cancel" and "Confirm" buttons.

3.2.17 Account Editing Page #2

Trivia Tree Category ▾ Search for a quiz... Quiz Q Categories Cart More SBU_Fan 

SBU_Fan
Platforms
Quizzes
Badges



SBU_Fan
 Gamer / Quiz Taker

Select Effect
Upload File

Featured Platforms

 Technology
 Japanese Food
 Best Cartoons
 Horses
 Add Platform

Featured Quizzes

 Mario Quiz
 Computer Parts
 Western Animation
 Music
 Celeste Levels

Change Background

Biography

I love among us. Among Us is a GOATED (Game Of The Yearned) game.

Badges

 Who's that Faculty?

 Stony Brook D1 Quiz

3 Badges Earned
Cancel
Confirm

3.2.18 Account Editing Page #3

Trivia Tree Category ▾ Search for a quiz... Quiz Q Categories Cart More SBU_Fan 

SBU_Fan
Platforms
Quizzes
Badges



SBU_Fan
 Gamer / Quiz Taker

Select Effect
Upload File

Featured Platforms

 Technology
 Japanese Food
 Best Cartoons
 Horses
 Add Platform

Featured Quizzes

 Mario Quiz
 Computer Parts
 Western Animation
 Music
 Celeste Levels

Change Background

Biography

I love among us. Among Us is a GOATED (Game Of The Yearned) game.

Select a platform to add:



 Stony Brook

Who's that Faculty?

Stony Brook Quiz

3 Badges Earned
Cancel
Add

3 Badges Earned
Cancel
Confirm

3.2.19 Account Page

The screenshot shows the Trivia Tree account page for a user named "SBU_Fan". The top navigation bar includes links for "Category", "Search for a quiz...", "Quiz", "Categories", and a user icon for "SBU_Fan". Below the navigation is a horizontal menu with tabs: "SBU_Fan", "Platforms", "Quizzes", and "Badges". The main content area features a profile section with a circular profile picture of a dog wearing a baseball cap, the username "SBU_FAN", and the title "Gamer / Quiz Taker". Below this are two sections: "Featured Platforms" (Technology, Japanese Food, Best Cartoons, Horses, Stony Brook) and "Featured Quizzes" (Mario Quiz, Computer Parts, Western Animation, Music, Celeste Levels). To the right is a "Biography" section with the text: "I love among us. Among Us is a GOATED (Game Of The Yeared) game." and a "Badges" section showing three earned badges: "Who's that Faculty?", "Stony Brook D1 Quiz", and "3 Badges Earned". A link to "Edit Account Page" is located at the bottom left.

3.2.20 List of Platforms

The screenshot shows the Trivia Tree list of platforms page for a user named "Peter2000". The top navigation bar includes links for "Category", "Search for a quiz...", "Quiz", "Categories", and a user icon for "SBU_Fan". Below the navigation is a horizontal menu with tabs: "Peter2000", "Platforms", "Quizzes", and "Badges". The main content area features a profile section with a circular profile picture of a dog wearing a baseball cap, the username "SBU_Fan", and the title "Gamer / Quiz Taker". Below this is a section titled "All Platforms" displaying various platform categories with their respective icons and names: Football, Best Baseball Quizzes, Technology, Japanese Food, Best Cartoons, Horses, Classical Painting, Wolfie Fans, Platform For My Dog, Mario 64, and Stony Brook.

3.2.21 List of Quizzes

The screenshot shows the Trivia Tree website interface. At the top, there's a navigation bar with 'Trivia Tree' on the left, a search bar 'Search for a quiz...', and categories like 'Quiz', 'Categories', and a shopping cart icon. On the right, it shows the user profile 'SBU_Fan' with a small profile picture. Below the navigation is a horizontal menu with tabs: 'Peter2000', 'Platforms', 'Quizzes' (which is highlighted in blue), and 'Badges'. The main content area features a banner for 'SBU_Fan' labeled 'Gamer / Quiz Taker' with a profile picture of a dog wearing a cap. Below the banner, the heading 'All Quizzes' is displayed. There are seven quiz categories shown as cards:

- Digital Artists
- E-Sports Players
- Name the Eva...
- Presidents
- Coding Quiz
- Flamingo Breeds
- Science and...

At the bottom left, there's a card for 'Volcanoes'.

3.2.22 List of Badges

The screenshot shows the Trivia Tree website interface, similar to the previous one but under the 'Badges' tab. The top navigation and user profile are identical. The horizontal menu now includes 'Peter2000', 'Platforms', 'Quizzes', and 'Badges' (highlighted in blue). The main content area features a banner for 'SBU_Fan' labeled 'Gamer / Quiz Taker'. Below the banner, the heading 'All Badges' is displayed. There are five badge categories shown as cards:

- Stony Brook D1 Quiz
- Basketball Stats Quiz
- Name That Pokemon
- Stony Brook 3000
- Hardest Quiz of All Time

At the bottom left, there's a card for 'Apes Quiz'.

3.2.23 Shop Page

The screenshot shows the 'THE TREE SHOP' interface. At the top right is a gold coin icon with the number '1245'. Below it is a navigation bar with tabs: 'Banner Effects' (selected), 'Icon Effects', 'Backgrounds', and 'Weekly Specials'. The main area displays four banner preview cards:

- Colorful Lights**: A winter scene with Santa Claus, snow-covered trees, and colorful lights. Price: 550.
- Neon Lights**: A city skyline silhouette at night with red and blue neon lights. Price: 500.
- Fire**: A group of people's legs wearing various sneakers standing on a fire-themed border. Price: 600.
- Flowers**: A purple mesh ball surrounded by flowers. Price: 500.

A large black arrow points to the right on the right side of the cards. Below the cards are three small numbered circles (1, 2, 3).

3.2.24 Purchase Page

The screenshot shows the purchase page for the 'Neon Lights' banner effect. At the top right is a gold coin icon with the number '1245'. The title 'THE TREE SHOP' is at the top center. To the right of the title is another gold coin icon with '1245'.

The main content area features a large preview image of the 'Neon Lights' banner effect, which shows a city skyline silhouette against a red sunset background with glowing neon lights.

Banner Effect

Neon Lights

\$ 500

[Back to Shop](#) [Purchase](#)

[Preview Banner](#)

3.2.25 Preview Page

The screenshot shows a user profile page for 'SBU_FAN'. At the top, there's a navigation bar with 'Trivia Tree' (highlighted in red), 'Category', 'Search for a quiz...', 'Quiz' (with a dropdown arrow), a search icon, a shopping cart icon, three vertical dots, and an orange 'Create Account' button.

The main content area features a large profile banner with a circular icon showing two plushies (one blue, one white) and the text 'SBU_FAN' in bold, followed by 'Gamer / Quiz Taker'. Below this are several sections:

- Featured Platforms:** Technology (blue icon), Japanese Food (yellow bowl icon), Best Cartoons (pink pig icon), Horses (purple horse icon), Stony Brook (red shield icon).
- Featured Quizzes:** Mario Quiz (Mario icon), Computer Parts (monitor icon), Western Animation (Cartoon character icon), Music (red musical note icon), Celeste Levels (Celeste character icon).
- Biography:** A short bio: "I love among us. Among Us is a GOATED (Game Of The Yeared) game."
- Badges:** Two badges are shown: "Who's that Faculty?" (gold medal icon) and "Stony Brook D1 Quiz" (gold medal icon). Below them, it says "3 Badges Earned".

4. Domain Model

4.1 Similar Problems

CSE 316 Project: Each of our team members have created a full web application in CSE 316 using the **MERN** stack, which is the main technology stack we'll be using to create Trivia Tree. Storing account information and user data in a **MongoDB** database was a key component in these projects, and is something we'll need to do in order to store user's quizzes, platforms, and badges. Single Sign On was accomplished using **Passport**, which Trivia Tree will do in the same way. One team member utilized **Apollo** for fetching and modifying items in the database, which is how data for users, platforms, quizzes, etc. will be obtained and stored.

SBU Hacks 2021 Project: Our team participated in the SBU Hacks 2021 event, building a very simple quizzing application using the **MERN** stack. We utilized **Chakra UI** as a front end styling library, which provided tools for creating clean looking user interfaces. We set up a database using **MongoDB Atlas**, which defined schemas for storing usernames and quiz points. **Mongoose** was set up to allow us to communicate with the MongoDB database. The quiz app also utilized **React-Router** in order to link different pages to each other in the form of a single page application.

4.2 Technology Set

Front-End Technologies:

[React] - front-end library to create our user interface using functional components and hooks to

- manipulate state information

[React-Router] - used to develop single page web applications by allowing switching between routes specified by the url. Users can redirect to different react components while saving state information

[Chakra UI] - front-end styling library to create simple, modular, and accessible components using React

Back-End Technologies:

[Node.JS] - Backend runtime environment for running a web server

[Express] - Web application framework for Node that handles routing and http requests

[Apollo] - Used for making API calls to the backend

[GraphQL] - Data query language and manipulation language where all desired data can be found with one query

[Mongoose] - object data modeling library for MongoDB that is used to model your data using schemas

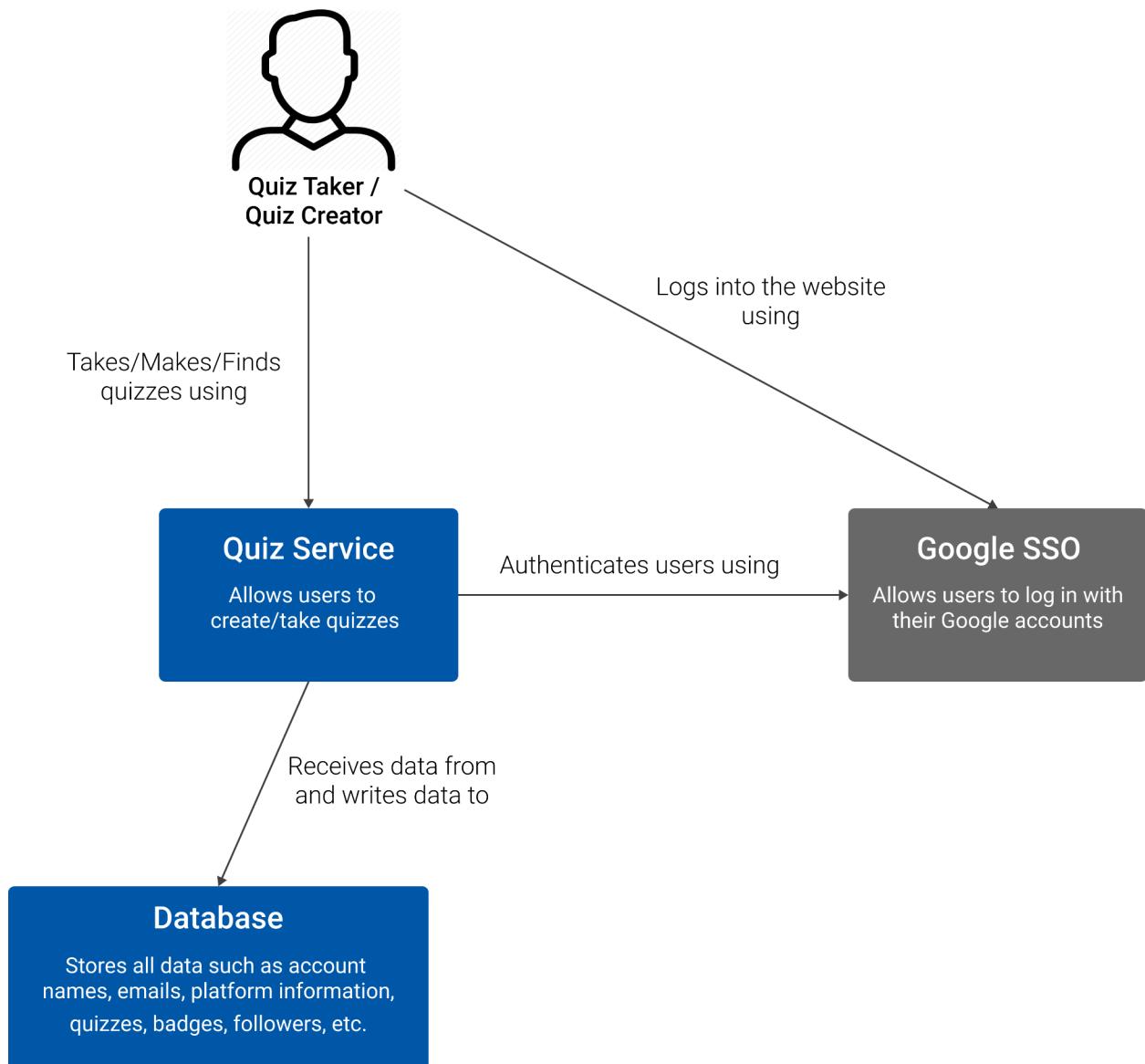
[MongoDB Atlas] - cloud-based NoSQL database that stores data in documents that employ a format similar to JSON

[Passport] - a middleware for Node that takes care of authentication requests from an OAuth provider such as Google or facebook

4.3 Architecture

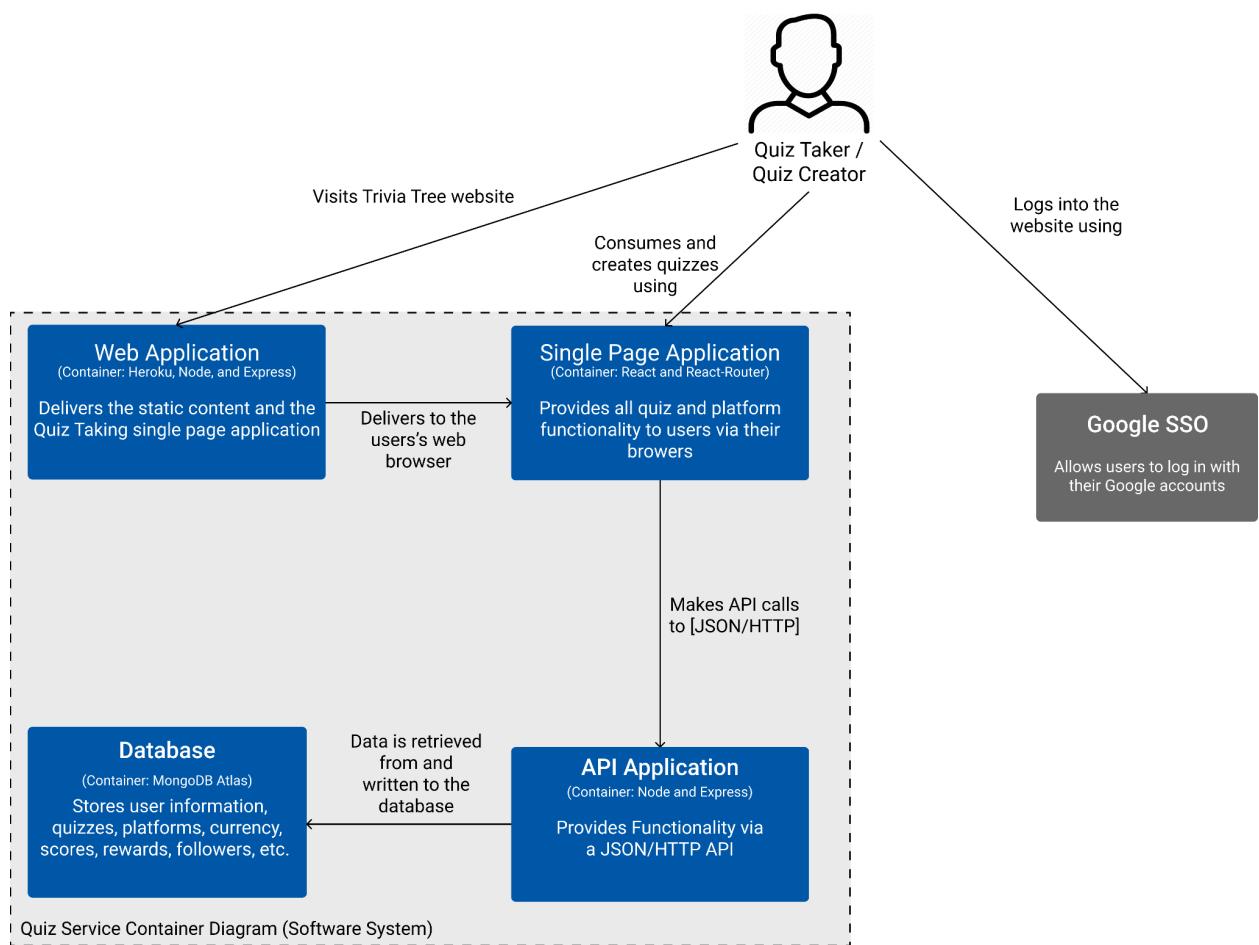
C1 Model

1. System Context Overview



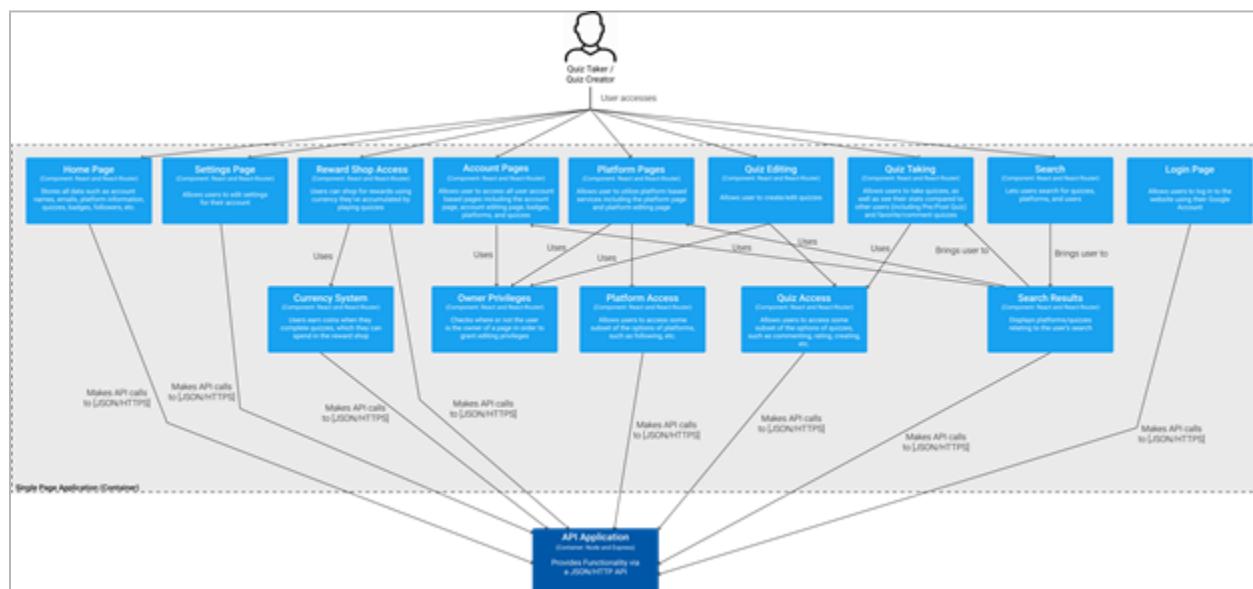
C2 Model

2. Containers



C3 Models

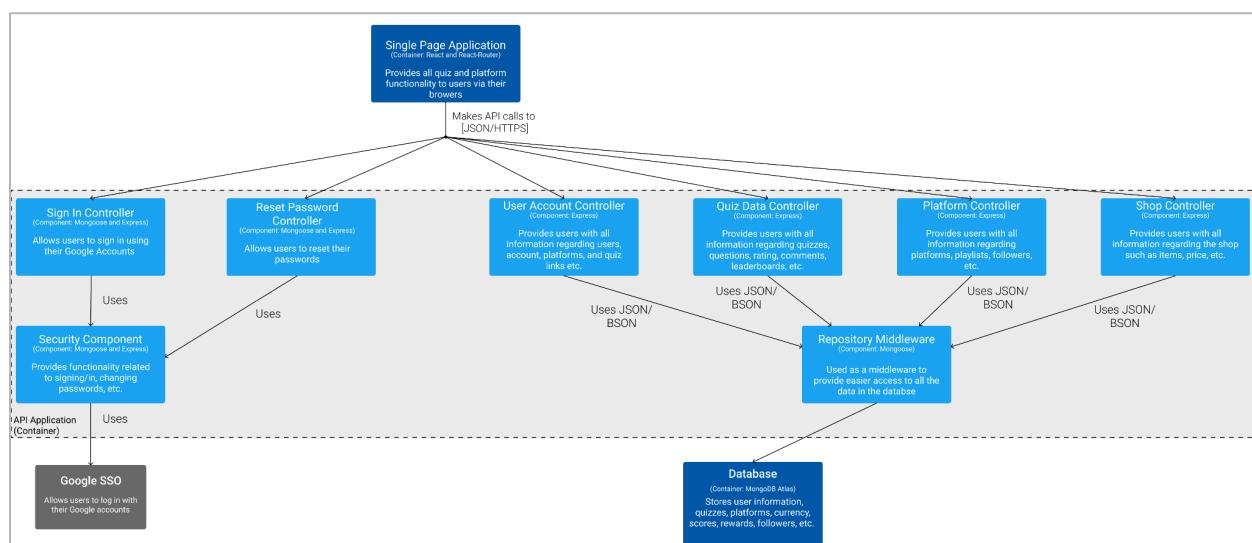
Single Page Application Components:



View in Full Resolution

<https://drive.google.com/file/d/16LPkBV4RGcVsK1bTKIQW4I1PzZEemT6v/view?usp=sharing>

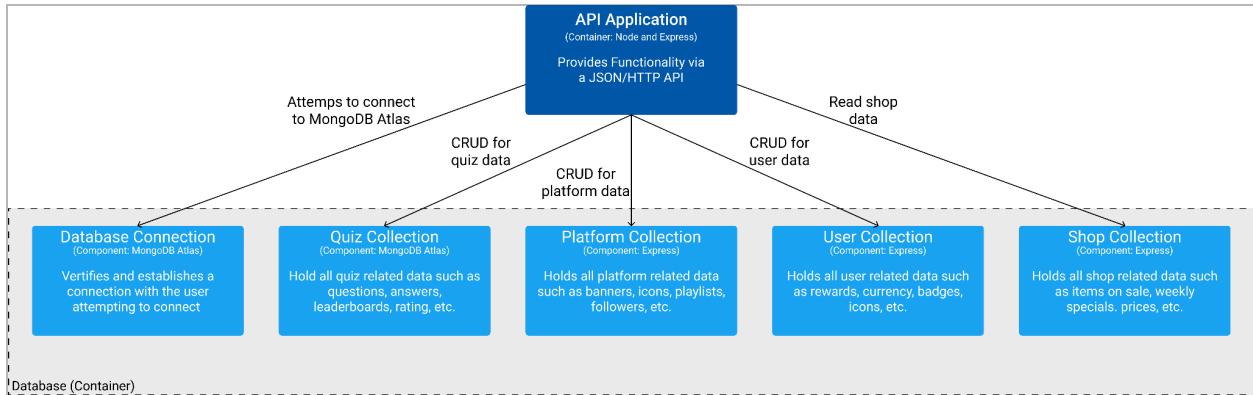
API Application:



View in Full Resolution

https://drive.google.com/file/d/1qT7Nzii0Bl5knY_GSNsjKqZUiBzYiMVs/view?usp=sharing

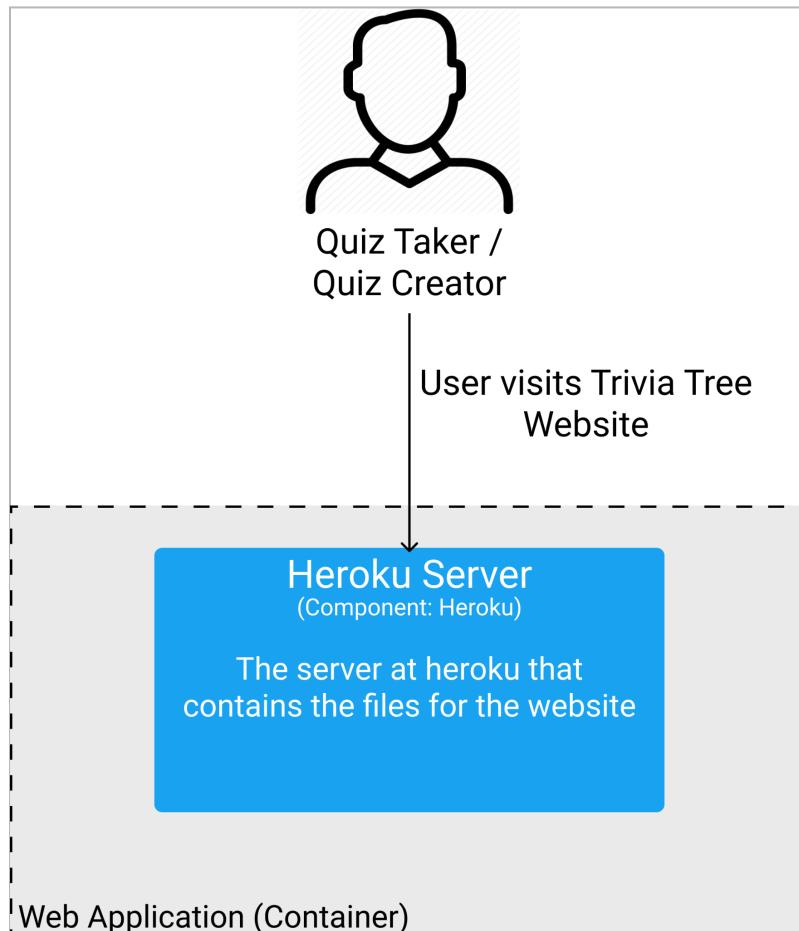
Database Components:



View in Full Resolution

https://drive.google.com/file/d/1idRIF5eleUR-ehn3Se0F5p_a1WcEAdBm/view?usp=sharing

Web Application Components:

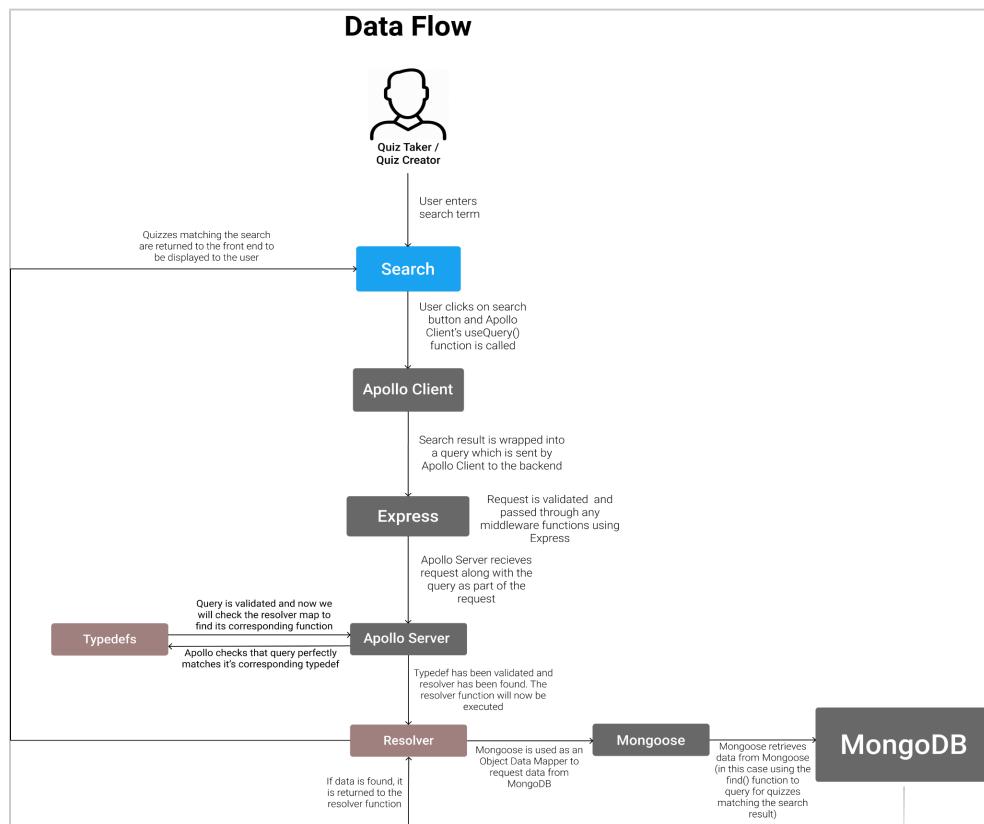


4.4 Dataflow

In describing our data flow, we will use the example of a user searching for quizzes by using our search bar. When a user types into the search bar and clicks on the search icon, a request will originate from our frontend through Apollo Client. In our code we will often come across lines like this

```
const data = useQuery(queries.GET_QUIZZES_FROM_SEARCH, {variables:search_input});.
```

Once a query like this is completed, data will contain a return object bearing the name of the query. For example a query called getQuizzesFromSearch will return some data and be accessed via **data.getQuizzesFromSearch**.



Our query originates from the front end and will be passed to the backend server as a request via Express. Our request will be validated by Express and pass through any of its middleware functions. Apollo Server will receive the request along with our query. Apollo Server will verify that our query perfectly matches its typedef. Finally our query will be mapped to its resolver function where the resolver will be executed. Mongoose is used as an Object Data Mapper (ODM) and translates between objects in our code and the representation of those objects in MongoDB. Mongoose retrieves the data from MongoDB using its predefined functions (i.e. `find()`, `findAll()`, `update()`). The data is sent back to the resolver function which in turn returns data to the originating request in the frontend. The frontend finally has data containing quizzes matching the search, and the data will be displayed on our applications search results page.

[View in Full Resolution](#)

<https://drive.google.com/file/d/12l35uRYlcCMaMKCqjanX4xqdKrn6P7aN/view?usp=sharing>

4.5 Training Verification

All Members

Participated in SBUHacks 2021, building a small-scale quizzing application called Space Quizzer. Users are able to choose from a selection of 4 space-related quizzes, each with 5 questions. Their score for these quizzes is kept track of in the top-left hand corner of the screen. A basic connection between the server and database was also established, however it was not utilized in the final version of the application.

- **Technologies Used:** React, Express, MongoDB Atlas, NodeJS, ChakraUI, Mongoose
- **Completed:** September 24-25, 2021
- **Devpost Link:** <https://devpost.com/software/space-quizzer>
- **Heroku Deployment:** <https://space-quizzer.herokuapp.com/>
- **Github Link:** <https://github.com/timothyhsu8/sbuhacks2021>

Matthew Li-Ah-Kim

- **Training Description:** Created and deployed a basic app with a homepage that leads to a login page. The login page verifies email: blah@gmail with password: 1 with any name against the database to login users. The user can then answer one question that brings them back to the homepage. The backend does not work with Heroku, so the functionality can be seen if one runs it locally using Github. Deploying a frontend and backend simultaneously on heroku is not trivial, so it was not done for this training verification.
- **Tutorials:** Read Various MongooseJS.com tutorials. devcenter.Heroku.com deployment tutorials.
- **Technologies Used:** Mongoose, MongoDB, React, NodeJS, ExpressJS.
- **Completed:** October 5-7, 2021
- **Heroku Deployment:** <https://evening-peak-09339.herokuapp.com/>
- **Github Link:** <https://github.com/Mattliahkim/TrainingVerification>

Timothy Hsu

- **Training Description:** Built a simple single page application for taking multiple choice quizzes with a basic system for tracking the names and scores of each user in the database. Users will first enter a username to login; if the user does not exist, a new one will be created in the database. There are two multiple choice quizzes that can be taken, after which the user's score will be updated and displayed on the leaderboard.
- **Tutorials:**
 - Full Stack Mern Project by Javascript Mastery
<https://www.youtube.com/watch?v=ngc9gnGgUdA>
 - How to deploy a MERN Stack App to Heroku
<https://www.youtube.com/watch?v=5PaUiPyBDJY>
 - Chakra UI Getting Started <https://chakra-ui.com/docs/getting-started>
 - Various MERN Stack application and Heroku deployment tutorials
- **Technologies Used:** React, Express, MongoDB Atlas, NodeJS, ChakraUI, Mongoose
- **Completed:** October 1-7, 2021
- **Heroku Deployment:** <https://timhsu-mini-quiz-app.herokuapp.com/>
- **Github Link:** <https://github.com/timothyhsu8/MiniQuizSite>

Tamzid Chowdhury

- **Training Description:** Did frontend work using React and ChakraUI to build the main homepage for the Space Quizzer SBUHacks 2021 project. Set up the backend architecture for the Trivia Tree website that will be deployed in Build #1, utilizing Node, Express, Mongoose, Apollo, and GraphQL, alongside connecting the server to a MongoDB Atlas database.
- **Tutorials:**
 - Apollo/GraphQL Fundamentals Course <https://odyssey.apollographql.com/lift-off-part1/>
 - Spring CSE 316 Frontend/Backend Setup
<https://github.com/wolfie-VIP/CSE316-Spring21-HW3#readme>
- **Technologies Used:** React, Express, MongoDB Atlas, Apollo, GraphQL, NodeJS, ChakraUI, Mongoose
- **Completed:** September 24-25 and October 6-7, 2021
- **Github Link:** <https://github.com/tamzid-chowdhury/Trivia-Tree>

Daniel Laszczyc

- **Training Description:** Created a single page application that allows users to create and take quizzes. Users can create multiple choice quizzes with however many questions they desire. Any quizzes created will appear in the homepage for other users to try. Since this is just a demo, any user can delete any quiz.
- **Tutorials:**
 - React / GraphQL Course - Build a social media app
<https://www.youtube.com/watch?v=n1mdAPFq2Os>
 - Chakra UI Getting Started
<https://chakra-ui.com/docs/getting-started>
 - Deploying to heroku with subfolders
<https://www.youtube.com/watch?v=gCjrEGAZCkw>
- **Technologies Used:** React, Express, MongoDB Atlas, NodeJS, ChakraUI, Mongoose, Apollo, GraphQL
- **Completed:** October 1-7, 2021
- **Heroku Deployment:** <https://daniel-quiz-training.herokuapp.com/>
- **Github Link:** <https://github.com/DanielLaszczyc/CSE416Training>

5. Data Model

5.1 Logical Design

User Schema

```
{  
    "_id": ObjectId,  
    "email": String,  
    "displayName": String,  
    "theme": Boolean, //for light mode vs dark mode  
    "background": Item,  
    "ownedBackgrounds": [Item]  
    "icon": Binary Data,  
    "banner": Binary Data,  
    "iconEffect": Item,  
    "ownedIconEffects": [Item],  
    "bannerEffect": Item,  
    "ownedBannerEffects": [Item],  
    "bio": String,  
    "quizzesMade": [Quizzes],  
    "quizzesTaken": [QuizData],  
    "platformsMade": [Platforms],  
    "featuredQuizzes": [Quizzes],  
    "featuredPlatforms": [Platforms],  
    "ownedBadges": [{  
        title: String,  
        placement: Number  
    }],  
    "featuredBadges": [{  
        title: String,  
        placement: Number  
    }],  
    "num_badges": Number,  
    "following": [Platforms],  
    "currency": Number,  
    "cart": [Items]  
    "verified": Boolean,  
    "admin": Boolean,  
    "newNotifications": [{  
        "_id": ObjectID,  
        "platform_id": ObjectID,  
        "user_id": ObjectID,  
        "message": String  
    }]  
}
```

Platform Schema

```
{  
    "_id": ObjectId,  
    "user_id": ObjectId,  
    "name": String,  
    "tags": [String],  
    "icon": Binary Data,  
    "banner": Binary Data,  
    "background": String  
    "playlists": [  
        name: String  
        quizzes: [Quizzes]  
    ],  
    "followers": [ObjectId]  
}
```

Quiz Schema

```
{  
    "_id": ObjectId,  
    "user_id": ObjectId,  
    "platform_id": ObjectId,  
    "title": String,  
    "icon": Binary Data  
    "categories": [String]  
    "tags": [String]  
    "description": String  
    "numQuestions": Number  
    "questions": [  
        "_id": ObjectId,  
        "questionType": String,  
        "question": String,  
        "answerChoices": [String],  
        "answer": [String] //array for questions with multiple answer choices  
    ],  
    "quizTimer": Number,  
    "questionTimer": Number,  
    "quizShuffled": Boolean,  
    "quizInstant": Boolean,  
    "comments": [  
        user_id: ObjectId,  
        comment: String  
    ],  
    "rating": Number,  
    "averageScore": Number,
```

```
        "medianScore": Number  
    }
```

Quiz Data Schema

```
{  
    "_id": ObjectId,  
    "quiz_id": ObjectId,  
    "user_id": ObjectId,  
    "elapsedTime": Number,  
    "score": Number,  
    "answerChoices": [[String]],  
    "questions": [[String]],  
    "dateAttempted": Date,  
    "rating": Number,  
    "isLiked": Boolean  
}
```

Item Schema

```
{  
    "_id": ObjectID,  
    "category": String,  
    "name": String,  
    "image": String,  
    "price": Number,  
    "weeklySpecial": Boolean  
}
```

Category Schema

```
{  
    "_id": ObjectId,  
    "name": String,  
    image: Binary Data  
}
```

5.2 Data Structures

Arrays: Used to store lists of similar data such as users, platforms, quizzes, and quiz questions. A lot of the data for Trivia Tree will be accessed linearly, such as a user traversing through a quiz, rendering platform/quiz icons on the Homepage, or displaying search results to the user; arrays are perfect for these cases. They also provide an easily accessible and modifiable way to store quiz questions and quiz answers, with the array indices corresponding to each question number. JavaScript also provides a number of useful functions for arrays such as map() or filter() which will be useful for certain actions, such as searching for quizzes.

JSON Objects: Used to store data using name-value pairs in the MongoDB database. Useful for storing related data of different types, such as all data associated with users, platforms, or quizzes. JSON Objects are also easily understandable, with all of the important data being associated with a name that describes what the data is; this is important because there will be a significant amount of information in the schemas, and we'll need to access specific data from these structures.

5.3 Local Storage

Session time: This will store the time at which the user entered the website so that we can gather data on how long a user stays on the website for.

- Session time will be stored in a Date() object which will be stored in LocalStorage. We can find the user's session time by subtracting their start time from a new Date() object.

Shopping Cart: Keeps track of the items that the user currently has in their shopping cart, so if they leave the webpage it will save that information.

- Items in the shopping cart will be stored in an array.

Types of Quizzes Taken: Stores categories of quizzes taken, in order to recommend quizzes to the user based on quizzes they've played previously. This will also be stored in the database.

- The types of quizzes taken (categories/tags of the quiz) will be stored in an array.

5.4 Data Dictionary

Attributes

<Category.categoriesList> - An array of categories from the category collection in the database

--Category Schema--

<Category.name> - The name of a category.

- This value can never be null. A category must have a name.

<Category.image> - A string that represents the path to the categories image, which is an image associated with the category

- This value can never be null. A category will be given a default image if none is provided.

--End Category Schema--

<Homepage.featuredPlatforms> - The array of featured quizzes given by the site

<Homepage.featuredQuizzes> - The array of featured quizzes given by the site

<Homepage.recommendedQuizzes> - The array of all the recommended quizzes of the given account.

<Homepage.whatsNew> - The array of the “What’s New” entries of the given account.

--Item Schema--

<Item.category> - A string that describes which category this item belongs to whether it be icon effects, banner effects, etc.

- This value can never be null. An item must belong to a category because this is how the frontend will know which store tab the item belongs to.

<Item.image> - A string that represents the path to the items image, this image will showcase the item.

- This value can never be null. Without an image tied to an item a user would not know what they are purchasing.

<Item.name> - A string that represents the name of the item.

- This value can never be null. An item must have a name, this is how users will distinguish between different items.

<Item.price> - Number denoting the cost of a specific item in the shop.

- This value cannot be negative.

<Item.weeklySpecial> - Boolean value denoting whether this item is a weekly special or not.

- By default, this value is false.

--End User Schema--

<Navbar.category> - A string which holds the category that a user has selected to act as a parameter in their search query.

<Navbar.search> - A string which holds the input from the search text box. This string will be used as the primary parameter in a search query.

<Navbar.searchType> - A string which holds the type of content the user has selected to search for whether that be quizzes, platforms, or users.

--Platform Schema--

<Platform.background> - The color code for a platform's background.

- If this value is null, it will display as a default gray background.

<Platform.banner> - The string that represents the path to the platform's banner.

- If this value is null, it will display as a default gray banner.

<Platform.following> - The boolean that represents whether a user is following this platform or not.

<Platform.followers> - An array of users who follow the platform.

<Platform.icon> - A string that represents the path to the platform's icon.

- This value can never be null. If a platform doesn't have a profile picture, it will be a default one.

<Platform.name> - The name of the platform.

- This can never be null and cannot contain special symbols ("\$", "%", "^", etc.). A platform must have a name, it is how users will distinguish between different platforms.

<Platform.playlists> - The playlists of the platform which hold an array of objects which themselves contain the name of a playlist and an array of quizzes associated with that playlist.

- This array can be empty if the platform owner hasn't added any playlists to the platform.

<Platform.tags> - An array of strings that describe the tags that are associated with the platform in order to help find the platform during searches.

- This array can be empty if the platform owner never added any tags to the platform.

<Platform.user_id> - The ObjectId that is linked to the user who is the owner of the platform.

- This can never be null. A platform must have an owner tied to it.

--End Platform Schema--

<PlatformEditing.quizList> - The array that represents all the owned quizzes of the owner of the platform.

<PostQuiz.allResponses> - The array of the user's answers.

<PostQuiz.questionList> - The array of arrays of quiz questions, with each question having 4 choices, an answer, and a number for displaying correct answers.

<PostQuiz.leaderboard> - The array of arrays that are the users that fill the leaderboard with their information, including rank, username, profile pic path, profile pic effect, and percentage score.

--Quiz Schema--

<Quiz.averageScore> - A number that represents the average score out of all the users who have taken a quiz.

- This value can be null if no users have taken the quiz.

<Quiz.categories> - An array of strings that describe what categories this quiz falls under. The categories in this array automatically become tags for the quiz.

- This value can be null if the owner of the quiz has not associated the quiz with any categories.

<Quiz.comments> - An array that holds an object which contains the user who wrote the comment and the text of the comment itself

- This array can be null or empty if no user has commented on a quiz.

<Quiz.description> - A string that provides a description for the quiz.

<Quiz.icon> - A string that represents the path to the quiz's icon.

- This value can never be null. If a platform doesn't have a profile picture, it will be a default one.

<Quiz.medianScore> - A number that represents the median score out of all the users who have taken a quiz.

- This value can be null if no users have taken the quiz.

<Quiz.numQuestions> - Holds the number of questions in a quiz. Will help users estimate how long a quiz will take to complete.

<Quiz.platform_id> - The ObjectId that is linked to a platform that this quiz is on.

- This value can be null if the owner of the quiz never put this quiz on any of their platforms.

<Quiz.questions> - An array that holds objects which describe all aspects of a question.

<Quiz.questions.answer> - An array of strings that describes the correct answers for a question. The array can have multiple answers in case the question is a ‘select all that apply’ question.

- This array can never be null or empty. A question must have at least one answer.

<Quiz.questions.answerChoices> - An array of strings that are each one of the choices in a question

- This array can never be null or empty. A question must have at least one choice for the user to select.

<Quiz.questions.question> - A string that holds the actual question.

- This value can never be null. A question must have a string that lets the user know what the question is asking.

<Quiz.questions.questionType> - Describes what type of question this is whether it be multiple choice, select all that apply, etc.

<Quiz.questionTimer> - A number that represents how long a user has to finish each question.

- By default this value will be 0. A quiz can not have both a quiz and question timer. The owner of the quiz will have to choose whether they want the quiz to have an overall timer or if each question will have a timer.

<Quiz.quizInstant> - A boolean that describes whether a user who has taken a quiz will know whether they get a question right or wrong right after confirming their answer.

- By default, this value will be false

<Quiz.quizShuffled> - A boolean that describes whether the questions of a quiz will be shuffled when a user takes the quiz.

- By default, this value will be false

<Quiz.quizTimer> - A number which represents how long a user has to finish an entire quiz.

- By default, this value will be 0. A quiz can not have both a quiz and question timer. The owner of the quiz will have to choose whether they want the quiz to have an overall timer or if each question will have a timer.

<Quiz.rating> - A number that represents the average rating of a quiz.

- This value can be null if no users have rated the quiz.

<Quiz.tags> - An array of strings that describe the tags that are associated with the quiz in order to help find the quiz during searches.

- This array can be empty if the quiz owner never added any tags to the quiz.

- This value can be null if the owner of the quiz never added a description to the quiz.

<Quiz.title> - The title/name of the quiz.

- This value can never be null and cannot contain special symbols (“\$”, “%”, “^”, etc.). A quiz must have a title, as this is how users will distinguish between different quizzes

<Quiz.user_id> - The ObjectId that is linked to a user who is the owner of the quiz.

- This value can never be null. A quiz must have an owner tied to it.

--End Quiz Schema--

--Quiz Data Schema--

<QuizData.answerChoices> - 2D array of Strings that denote the correct answer for each quiz question.

- The length of this array should match the length of the questions array.

<QuizData.dateAttempted> - Date object for exactly when the user took this quiz.

<QuizData.elapsedTime> - Number that represents the amount of time a user took to complete a quiz.

- This value can never be negative.

<QuizData.isLiked> - Boolean value that represents whether a user has “liked” a quiz or not.

- By default, this value is false.

<QuizData.questions> - 2D array of Strings for each question in a quiz.

- The length of this array should match the length of the answerChoices array.

<QuizData.quiz_id> - The ObjectId of a specific quiz.

- All QuizData objects must have a quiz_id associated with them.

<QuizData.rating> - Number that represents the rating that the user has given this quiz.

- This value can be null if the user did not rate this quiz.

- If non-null, this value must be between [1,5].

<QuizData.score> - Percentage of questions that the user answered correctly for a specific quiz.

- This value can never be negative.

<QuizData.user_id> - The ObjectId of the user who took the quiz.

- All QuizData objects must have a user_id associated with them.

--End Quiz Data Schema--

<QuizTaking.allResponses> - The array of the user’s answers.

<SettingsPage.email> - The email of the user, but it cannot be changed

- This value can never be null.

<SearchResults.filters> - The array of filters placed on the search, saved for the next search. Includes rating, difficulty, quiz type, timer.

<SearchResults.resultList> - The array of arrays that includes quizzes found by the search, each quiz has an owner, platform, quiz name, rating, difficulty, quiz type, timer.

<Shop.account> - The string that represents the username of the account the shop page is related to.

- This value can never be null, there is always an account associated with an instance of the shop page.

<Shop.backgrounds> - The array of shop background items that will be offered to the user depending on what items they already own.

<Shop.banner> - The string that represents the path to the given page banner.

<Shop.bannerEffects> - The array of shop banner effect items that will be offered to the user.

<Shop.currency> - The integer that represents the amount of currency the given user has.

- By default this value is 0.

- This value cannot be negative

<Shop.iconEffects> - The array of shop icon effect items that will be offered to the user.

<Shop.weeklySpecials> - The array of shop weekly special items that will be offered to the user.

<ShopConfirm.account> - The string that represents the username of the account the preview is related to.

- This value can never be null, there is always an account associated with an instance of the shop confirm page.

<ShopConfirm.banner> - The string that represents the path to the given page's banner.

<ShopConfirm.currency> - The integer that represents the amount of currency the given user has.

- By default this value is 0.

- This value cannot be negative

<ShopConfirm.currentItem> - Array with information about the item currently being purchased, including the string that represents the path to the image of the item, price, name.

<ShopPreview.banner> - The string that represents the path to the given account's banner.

<ShopPreview.bannerEffect> - The string that represents the path to the shop's shown banner effect.

<ShopPreview.profilePicEffect> - The string that represents the path to the given account's profile pic effect.

--User Schema--

<User.admin> - Boolean value representing whether a user is an administrator or not.

- By default, this value is false.

<User.background> - The Item object that represents a user's account background.

- This value can never be null. If no background is selected, it will default to a gray one.

<User.banner> - The string that represents the path to the given account's banner.

- This value can never be null. If no banner is uploaded, the user will be given a default gray banner.

<User.bannerEffect> - The Item object that represents a user's banner effect icon.

- If this value is null, no banner effect will be displayed.

<User.bio> - String that represents the bio of a given account.

- This value can be no longer than 256 characters. If the character limit is exceeded, the user will be prevented from saving.

<User.cart> - The current items in the shopping cart of the given user

- Data stored in this must be the **id** of an item that the user doesn't own.

<User.currency> - Number for the amount of currency a given user has

- By default, this value is 0.

- This value can never be a negative number.

<User.displayName> - The username of the given user.

- This value can never be null and cannot contain special symbols ("\$", "%", "^", etc.)

<User.email> - The email of the given user

- This value can never be null.

<User.featuredBadges> - Array of badges that a user is displaying on their account page.

- If this value is null, no badges will appear in the "Featured Badges" section of the user's account page.

<User.featuredQuizzes> - Array of Quiz Objects representing all of the quizzes a user is displaying in the "Featured Quizzes" section of their account page.

- If this value is null, the "Featured Quizzes" section will not appear on the user's account page.

<User.featuredPlatforms> - Array of Platform Objects representing all of the quizzes a user is displaying in the "Featured Platforms" section of their account page.

- If this value is null, the "Featured Platforms" section will not appear on the user's account page.

<User.following> - Array of all the platforms that a user is following.

<User.icon> - The profile picture of a given user.

- This value can never be null. If no profile picture is uploaded, the user will be given a default profile picture.

<User.iconEffect> - The Item object that represents a user's profile picture effect.

- If this value is null, no profile picture effect will be displayed.

<User.newNotifications> - Array that holds all of the notifications for a given user. Notifications show up under the "What's New" section on the homepage, and will display activities such as new quiz postings from platforms that a user follows.

<User.num_badges> - Number of badges that a user has accumulated.

- By default, this value is 0.
- This value can never be a negative number.

<User.ownedBackgrounds> - The array of Items that represent the backgrounds that a user owns.

- This value can be null if the user hasn't purchased any special backgrounds.

<User.ownedBadges> - Array of all badges that a user owns.

<User.ownedBannerEffects> - The array of Item objects that represents the banner effects that a user owns.

- This value can be null if the user hasn't purchased any banner effects from the shop.

<User.ownedIconEffects> - The array of Item objects that represent the profile picture effects that the user owns.

- This value can be null if the user hasn't purchased any icon effects.

<User.platformsMade> - Array of Platform objects representing all of the platforms a user has created.

<User.quizzesMade> - Array of Quiz objects representing all of the quizzes a user has created.

<User.quizzesTaken> - Array of QuizData objects representing all of the quizzes a user has taken.

<User.theme> - User's preference for dark mode or light mode.

<User.verified> - Boolean value representing whether a user is verified or not.

- By default, this value is false.

--End User Schema--

Objects

<Admin> - User with an additional authority to moderate content

<Guest> - User without authority to make content or access any pages that require an account.

<Platform> - Page with featured quiz playlists, a follow button as well as the account navbar

<Account> - Page containing featured platforms and quizzes as well as a short bio, badge list, and account navbar to other sections of the account.

<Quiz> - Page containing questions, a timer, links to specific questions, and 4 responses.

<User> - Actor with the authority to take quizzes, as well as the ability to create quizzes and platforms. This actor also has access to all pages that require an account including their account page as well as the ability to edit that account page. Furthermore, the actor has data associated with their account stored in the database that persists between sessions such as badges and settings.

Files

Badges - Badge images

Banner - Account banner images.

BannerEffect - Banner effect images.

Categories- Category images.

IconEffect - Icon effect images images

Profile Picture - Profile picture images

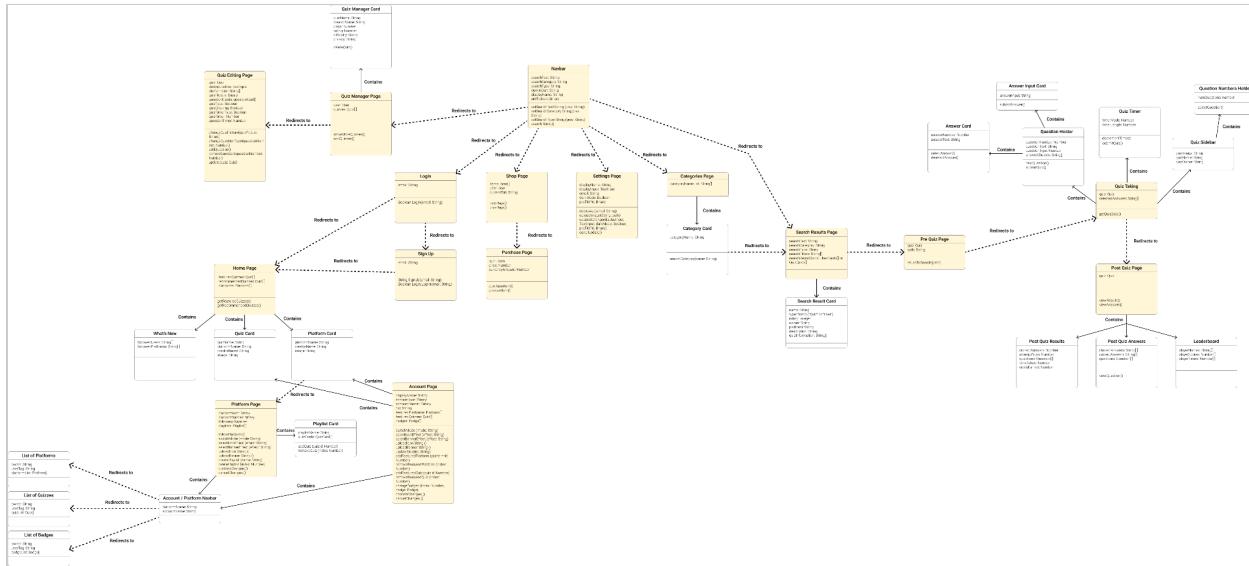
Tables

QuizData - Displays tables of different data for the owner of a quiz including who has taken the quiz, the scores on the quiz, when the quiz has been played, etc.

PlatformData - Displays tables of different data for the owner of a platform including followers, how many people visit the platform and when, which quizzes are played, etc.

6. Software Model

6.1 Front End Components



[View In Full Resolution](#)

<https://drive.google.com/file/d/138wGV9qx3iQ4ro6lM5jDpiYgdp9VJFr/view?usp=sharing>

6.2 Routing

General Routes

/ (home)
 /loginpage
 /signuppage
 /settingspage

Search Routes

/search/:category/:searchQuery

Quiz Routes

/createQuiz
 /quizmanager
 /prequizpage/:quizId
 /quizztakingpage/:quizId
 /editQuiz/:quizId
 /postquizpage/:quizId/:quizAttemptId

Platform Routes

```

/create-platform
/platform-manager
/platform/:platformId
/platform/:platformId/edit

```

Account Routes

```
/accountpage/:userId
```

Shop Routes

```

/shoppingpage
/previewpage/:userId/:item

```

6.3 Back-End API

Schema	User	Platform	Quiz	Quiz Attempt	Item	Category
Queries	<p>getUser(userID: ID): User</p> <p>getUsers(userID: [ID]): [User]</p> <p>getNewNotifications(userID: ID): User</p> <p>isVerified(userID: ID): Boolean</p> <p>isAdmin(userID: ID): Boolean</p>	<p>getPlatforms(platformIDs: [ID]): Platform</p> <p>getPlatform(platformID: ID): Platform</p> <p>getUserPlatforms(userID: ID): Platform</p> <p>platformUpdate(platform: Platform): Boolean</p> <p>createPlatform(name: String, owner: User): Boolean</p> <p>deletePlatform(platform: Platform): Boolean</p> <p>createPlaylist(name: String): Boolean</p> <p>addToPlaylist(playlistID: ID, quizID: ID): Boolean</p> <p>removeFromPlaylist(playlistID: ID, quizID: ID): Boolean</p> <p>deletePlaylist(playlistID: ID): Boolean</p> <p>updateIcon(icon: String): Boolean</p> <p>updateBanner(banner: String): Boolean</p> <p>broadcastNotification(message: String): Integer</p>	<p>getQuizzes (quizIDs: [ID]): [Quiz]</p> <p>getQuiz (quizID: ID): Quiz</p> <p>getOwnerQuizzes (userID: ID): [Quiz]</p> <p>getQuizzesFromTags (tags: [String]): [Quiz]</p> <p>getComments (quizID: ID): [Comment]</p> <p>getQuizStatistics (quizID: ID): Statistics</p> <p>createQuiz (Quiz): Quiz</p> <p>deleteQuiz (quizID: ID): Boolean</p> <p>updateQuizTitle(title: String): String</p> <p>updateQuizDescription(description: String): String</p> <p>updateQuizTags(tags: [String]): String</p> <p>updateQuizQuestions(quizQuestions: quizInput): Quiz</p> <p>updateQuizSettings(quizSettings: settingsInput): Quiz</p> <p>updateQuizIcon(icon: String): Boolean</p> <p>updateQuizRating(rating: Float): Boolean</p> <p>broadcastNotification(message: String): Integer</p>	<p>getQuizAttempt (quizDataID: ID): QuizAttempt</p> <p>getQuizAttemptsForUser (quizDataID: ID, userID: ID): QuizAttempt</p> <p>getAttemptsForQuiz (quizID: ID): [QuizAttempt]</p> <p>createQuizAttemptData(data: QuizAttemptData): QuizAttempt</p> <p>updateWeeklySpecials(itemList: [Item]): Boolean</p> <p>updateUserItems(userID: ID, itemList: [Item]): Boolean</p>	<p>getUsersItems(currentUser: User): [Item]</p>	<p>getCategories (categoryIDs: [ID]): Category</p> <p>getCategory (categoryId: ID): Category</p> <p>addCategory(category: Category): Boolean</p> <p>deleteCategory(categoryID: ID): Boolean</p> <p>deleteCategories(categoryID: [ID]): Boolean</p>
Mutations	<p>Login(): User</p> <p>Logout(): Boolean</p> <p>addBackground(background: Item): Boolean</p> <p>addIconEffect(iconEffect: Item): Item</p> <p>addBannerEffect(bannerEffect: Item): Item</p> <p>addPlatformToFollow(platform: Platform): Boolean</p> <p>addNewNotification(userID: User, message: String): User</p> <p>updateDisplayname(name: String): Boolean</p> <p>updateTheme(theme: Boolean): Boolean</p> <p>updateIcon(icon: String): Boolean</p> <p>updateBanner(icon: String): Boolean</p> <p>updateBackground(item: Item): Boolean</p> <p>updateIconEffect(item: Item): Boolean</p> <p>updateBannerEffect(item: Item): Boolean</p> <p>updateBio(bio: String): Boolean</p> <p>updateFeaturedQuizzes(quizIDs: [Quiz]): Boolean</p> <p>updateCurrency(currency: Currency): Boolean</p> <p>updatePlatform(platforms: [Platform]): Boolean</p> <p>updateCurrency(amount: Integer): Boolean</p> <p>deleteAccount(userID: User): Boolean</p>					

[View In Full Resolution](#)

<https://drive.google.com/file/d/1x6ILRO6JR5aYLHDKqizHv3GngrTF5j7b/view?usp=sharing>

7. Progress Reviews

Section 7.1 Deployment Details

- **Version Control:** <https://github.com/tamzid-chowdhury/Trivia-Tree>
- **Deployment Platform:** Heroku: <https://triviatree.herokuapp.com/>
- **CI/CD:** Jest
- **Build Process:**
 - Commit messages are a full sentence starting with a verb.
 - No force pushing (ex. git push --force) can be dangerous.
 - Everyone is using Visual Studio Code as their development environment.
 - Communicate with the team before every push. Ensure that there won't be any conflicts.
 - If there is a merge conflict, communicate with the team to resolve the conflict. Don't make any assumptions or try overwriting existing code before figuring it out as a team.
 - No extremely small pushes (preferably) in order to prevent frequent pulling from the team.
 - Each team member will work on separate tasks to prevent overlapping of work and problems with version control.
 - What constitutes a change-worthy commit? The new code fixes a bug, completes a task, or does a sizable portion of work. All commits must be stable, meaning they don't break anything else.

Use Cases

Build #1:

- Simple CRUD Operations accessible through the front end
- Set up GraphQL and Apollo
- Quiz Taking HTML
- Post-Quiz/Results/Answers HTML
- Account Page HTML
- Deploy project
- Set up routes

Build #2:

- View Quiz Info - **Tamzid**
- Start Quiz - **Tamzid**
- CI /CD Setup - **Tim**
- Select A Quiz - **Daniel**
- Search Results Page Pulling Data - **Daniel**
- Search Results HTML - **Tim**
- Return To Homepage - **Tim**
- Create a Quiz HTML - **Daniel**
- Pre-Quiz HTML - **Tamzid**
- Navigate To Your Account Page - **Tamzid**
- Navigate To Previous Question - **Tamzid**
- Navigate To Next Question - **Tamzid**
- Select An Answer - **Tamzid**
- Navigate Back To Search - **Tamzid**
- Login/Sign In Page HTML - **Tim**
- Login system with GraphQL and Route Authorization - **Daniel**
- Finish Quiz - **Tamzid**
- Categories HTML - **Matthew**
- Refine Post Quiz - **Matthew**
- Edit Quiz - **Daniel**
- Add Quiz Question - **Daniel**
- Retake Quiz - **Matthew**
- Personal Results - **Matthew**
- Navigate To Specific Question - **Tamzid**
- Update Quiz Timer - **Daniel**
- Update Quiz Question Title - **Daniel**
- Update Quiz Name - **Daniel**
- Update Quiz Answer Choices - **Daniel**
- Update Quiz Answer - **Daniel**
- Create Quiz - **Daniel**

Build #3:

- View Leaderboard - **Tamzid**
- Quiz Editing HTML - **Daniel**
- Search for a Quiz - **Tim**
- Delete Quiz Question - **Daniel**
- Select A Category - **Matthew**
- Review Quiz Results -
Matthew/Tamzid
- Update Quiz Image - **Daniel**
- Add Quiz Timer - **Tamzid**
- Update Questions Timer - **Daniel**
- Update between Shuffled and Ordered Questions - **Daniel**
- Update between Instant and Standard Answers - **Daniel**
- Navigate To Shop - **Tamzid**
- Navigate To Settings Page - **Matthew**
- Shop Page HTML - **Tamzid**
- Settings HTML - **Matthew**
- Platforms, Quizzes, Badges viewable on Account Page - **Tim**
- Platform/Platform Editing HTML - **Tim**
- Select Platform - **Tim**
- User Card HTML -
- Homepage HTML - **Tim**
- Select User - **Tim**
- Quiz Card HTML - **Tim**

Build #4:

- View Platform Followers - **Tamzid**
- Update Platform Title - **Tim**
- Update Platform Icon - **Tim**
- Update Platform Description - **Tim**
- Update Platform Colors - **Tim**
- Update Platform Banner - **Tim**
- Update Featured Quizzes - **Daniel**
- Update Featured Platforms - **Daniel**
- Update Account Picture - **Daniel**
- Update Account Banner - **Matthew**
- Update Account Bio - **Daniel**
- Update Account Background
- Unfollow Platform - **Tamzid**
- Sort Search Results
- Navigate To Favorites - **Tamzid**
- Navigate Back To Rewards Screen - **Matthew**
- Follow Platform - **Tamzid**
- Favorite/Unfavorite A Quiz - **Tamzid**
- Create Platform - **Tim**
- Edit Account Page - **Matthew**
- Edit Platform - **Tim**
- Add Quiz To Platform
- Navigate To Followed Platforms - **Tamzid**
- Search for a Platform - **Tim**
- Platform Card HTML - **Tim**
- Quiz Create and Edit Page Optimization - **Daniel**

Build #5:

- Update User Display Name - **Tamzid**
- Remove Quiz From Platform - **Tim**
- Preview Item - **Matthew**
- Filter By Time - **Tim**
- Filter By # of Plays - **Tim**
- Filter By # of Favorites - **Tim**
- Rate Quiz - **Tamzid**
- Delete Quiz - **Daniel**
- Delete Platform - **Tim**
- View User's Quizzes - **Tim**
- View User's Platforms - **Tamzid**
- Search for a User - **Tim**
- Navigate To Quiz Manager - **Daniel**
- Navigate To Rewards - **Matthew**
- Navigate To Platform Manager - **Tim**
- Update Featured Quizzes - **Tamzid**
- Update Featured Platforms - **Tamzid**
- Navigate To Favorites - **Tamzid**
- Follow Platform - **Tamzid**
- Unfollow Platform - **Tamzid**
- View Platform Followers - **Tamzid**
- Navigate To Followed Platforms - **Tamzid**
- Unfavorite Quiz - **Tamzid**

Build #6:

- User's Followed Platforms - **Tamzid**
- User's Favorited Quizzes - **Tamzid**
- Item Preview Screen - **Matthew**
- Select Item From Shop - **Tim**
- Purchase Screen - **Tim**
- Delete User Account - **Daniel**
- Purchase Item - **Tim**
- Select All That Apply Questions - **Daniel**
- Shuffled Question Ordering - **Daniel**
- Instant Quiz Type - **Daniel**
- (Platform) Create Playlist - **Tim**
- (Platform) Add Quiz To Playlist - **Tim**
- (Platform) Delete Playlist - **Tim**
- Post Quiz Recommendations - **Tamzid**
- User Based Quiz Recommendations - **Tamzid**
- Change Between Dark Mode and Light Mode - **Matthew**
- Comment On A Quiz - **Tamzid**
- Reply on a Quiz Comment - **Tamzid**
- Delete A Quiz Comment - **Tamzid**
- Delete A Quiz Reply - **Tamzid**

Section 7.4 Build (Project Setup and Deployment)

<https://triviatree.herokuapp.com/>

8. Post Development and Presentation

Section 8.1 Post Mortem

Our team succeeded in communication, with each teammate tackling separate tasks and being very vocal about what they were currently working on in order to prevent conflicts in code. When problems arose, we would consult other team members for assistance. And we held team meetings whenever we needed to decide whether a certain feature described in the Use Case document should be implemented or scrapped, or if we wanted to add new features to the site. Our team also found a comfortable workflow, with each team member handling the implementation of major features with multiple use cases associated with them (E.g. Platforms, comments, backend, etc.). This prevented confusion regarding how a section was implemented by another teammate, and allowed one person to “specialize” in that section of the project.

Creating Trivia Tree was a great learning experience, and made us all more comfortable with the full process of creating a software product from start to finish- receiving the requirements, planning out the design and architecture, and implementing the actual product, all while regularly communicating with the team.

Appendix A: Meeting Minutes

Date	Duration	Worked On	Members Present
8/30/21	1 Hour	Systems Analysis	All
9/3/21	3 Hours	Systems Analysis	All
9/5/21	1 Hour	Systems Analysis	All
9/10/21	3 Hours	Use Cases	All
9/11/21	3 Hours	Use Cases	All
9/13/21	2 Hours	Use Cases	All
9/14/21	2 Hours	Use Cases	All
9/17/21	4 Hours	UI Model	All
9/19/21	3 Hours	UI Model	All
9/20/21	3 Hours	UI Model	All
9/22/21	3 Hours	Video Presentation Preparation	All

9/24/21	2 Hours	Hackathon Training Requirement	All
9/25/21	41 Minutes	Hackathon Training Requirement	All
9/27/21	3 Hours	Domain Model	All
9/29/21	3 Hours	Domain Model	All
9/30/21	3 Hours	Domain Model	All
10/3/21	2 Hours	Data Model	Daniel, Tamzid, Matthew
10/5/21	3 Hours	Data Model	All
10/11/21	3 Hours	Software Model	All
10/13/21	1 Hour	Software Model	All
10/15/21	2 Hours	Software Model	All
10/22/21	1 Hour	Discussed state of the project	All
10/26/21	31 Minutes	Divided tasks, moved around use cases	All
11/2/21	1 Hour	Meeting to explain authentication	Daniel, Tamzid, Timothy
11/12/21	1 Hour	Discussed use cases	All
11/21/21	1 Hour	Discussed state of the project	All
12/4/21	25 Minutes	Discussed state of the project, discussed use cases	All

Appendix B: Code Presentation Slides

https://docs.google.com/presentation/d/1jhvQ44sMWT4Id-xe3dR_t3_rBeT_vUGhK7Lfin_DOFA/edit?usp=sharing

Appendix C: Final Presentation Slides

No presentation slides used for the final presentation

Appendix D: Team Member Contributions

Daniel Laszczyc

- Worked on the early development of the backend, specifically related to our database meaning designing a framework for queries, mutations, user validation, and authentication.
- Implemented SSO allowing users to sign in through Google OAuth.
- Created the pages responsible for creating and editing quizzes and connected both pages to the backend.
- Created the functionality to edit an account page such as editing their profile picture, banner, and bio. Also connected the account page to the backend allowing for any changes made to be saved to the database.
- Created the quiz manager page which is where a user can view all the quizzes they've created and either view their statistics, edit them, or delete them.
- Designed our image storing system using cloudinary which allowed us to save images for account pages, quizzes, and platforms.
- Worked on smaller features such as shuffled questions, instant quizzes, and rating system.

Matthew Li-Ah-Kim

- Created initial webpages for the shop, shop item account preview, settings, post quiz page, editing account page, and connected them to the application by creating the routes and user navigation in the application.
- Created initial system for shop navigation going from shop item to user page page with the user page having the item applied to their account temporarily.
- Implemented the dark mode for the application using Chakra-UI's built-in functionality, changing some styles manually to fit the theme, and also to keep all components visible. Also connected dark mode to login and logout, so the theme is changed automatically when logging in and out.

Tamzid Chowdhury

- Worked on functionality for the quiz taking page, including navigating between questions, implementation of the timer, and submitting a quiz attempt to the backend.
- Implemented quiz leaderboard functionality, displaying quiz results and answers, and rating a quiz on the post quiz page. Also implemented the comment system functionality for the post quiz page, including leaving comments, replying to comments, and storing comments in the backend.
- Implemented forums on the platform page, along with making forum posts, replying, liking forum posts and posting images. Also implemented platform following.
- Implemented the pre-quiz page which displays quiz information before the start of a quiz and allows a user to favorite a quiz. Also implemented functionality for updating settings in the settings page.
- Created the recommended quizzes feature, which displays quizzes relevant to a user's interests on the homepage and post-quiz page.
- Created the sign up page, which lets a user choose a display name, profile picture, light/dark mode theme, and categories they're interested in.
- Worked on Account Page functionalities, such as featuring quizzes/platforms and viewing a user's favorite quizzes and followed platforms

Timothy Hsu

- Implemented platform page along with the functionalities for updating the profile picture, banner, name, description and adding/removing quizzes on the frontend and backend.
- Handled refining UI on all pages of the website.
- Added playlist functionality on platforms, along with options to move playlists up and down, rename playlists, delete playlists, and add/remove quizzes from playlists on the frontend and backend.
- Implemented the search results page along with search results cards for quizzes, platforms, and users. Included functionalities to sort and filter quizzes by different characteristics.
- Implemented purchasing items from the shop page, along with applying and saving items on a user's account page.
- Created artwork for the landing page banner, as well as implemented quiz/platform of the day to go with it.
- Created layout for account page, search results, platform manager, login page, navbar, quiz cards and platform cards