**Quiz App Documentation**

Patrick Snow

Cybersecurity Major

Charleston Southern University

Dr. Sean Hayes

Expected Graduation Date: Spring 2023

**Statement of Purpose**

The purpose of this project is to test my knowledge of computer science and cyber security. The project can also be used as another alternative to using other quiz apps such as Quizlet.

**Problem Statement**

My website will be more interactive with quizzes compared to most that are shared on social media in which it will constantly have fresh material because the users will be able to create their own custom quizzes and share them. The main reason why people are more likely to ignore a quiz app is because it will be viewed as a waste of their time.

Quiz applications could benefit from a lot of new changes. The first benefit of my website is that it will have multiple tests for many categories that can be listed such as Marvel, Star Wars, animals, movies, shows, and music. The amount of categories that are built is to help attract as many users as it can. The second benefit of my website is that it will be more entertaining because there will be endless choices of quizzes for the user to play with considering there will be some already provided as well as other tests contributed from the previous users that visit the site. The final benefit of the app is that it will be more interactive than most quiz apps because it will allow the users to create as many custom quizzes they want while also browsing tests created by other users. My quiz app could have a rating system with a filtered list to sort out terribly made quizzes. Most websites with a quiz are either boring, not interactive with fresh materials, or not diverse enough which is why mine will be more cost-effective, more entertaining, less tedious, and will have a diverse amount of tests.

**Research & Background**

I already had plenty of experiences with multiple quiz applications. I wanted to use my knowledge of computer science and cyber security to make an application that is similar to an app such as Quizlet but with some added features. The features I would like to add but did not get to were color blind accessibility, text-to-speech for the blind or low-vision users, and larger text. The research into this project helped me to learn more about new codes, UI software development kit, and cloud storage.

Flutter is a fairly new UI software development kit in which it was released in 2017 by Google. The main reason why I wanted to try this UI SDK was that it did not need any platform-specific UI components to render its UI (Dziuba, 2022). Flutter saves time and effort with the same end product because it can share the UI and business logic (Dziuba, 2022). The second best feature of working with Flutter is the “hot reload” feature which shows applied changes instantly of the application. Flutter has a huge list of ready-to-use widgets that also help save time on making a framework. The other features include customizable UI, platform-specific logic implementation and Flutter Web. Flutter Web can run the same source-code that was used to make a Flutter application (Dziuba, 2022).

Firebase supports Flutter in creating highly functional and versatile web, Android, and iOS platform applications (Batschinski, 2022). Firebase was also made by Google in which it is a backend platform for building applications. The three main uses of Firebase are to build efficiently, release and monitor an application, and engage with the users. The main features I utilized with Firebase were the databases, authentication, and push messages. Firebase services are cloud-hosted which is great for developers because they will not have to worry about scaling it manually.

The final research that was needed for this project was on the programming language called Dart. It is an open-source general-purpose programming language meant for servers and created by Google. Most developers who are already familiar with object-oriented programming will most likely find Dart easy to use. Dart is newer compared to a programming language such as JavaScript so it has a smaller network. Since Dart has basically started the process of becoming a stable programming language then it is also going to be confusing for developers to use it (Dart vs JavaScript, 2021).

Flutter, Firebase, and Dart are interesting new tools to be utilized in the future. At the moment, it seems they are still rapidly evolving so will its programming syntaxes. It was fun to interact and learn with a new UI SDK and programming language. Although at times it was difficult to understand why certain codes functioned the way they did. The “hot-reload” feature was helpful in seeing the changes to the application instantaneously. Firebase database storage was my first time making an application interact with a database storage that was not stored locally.

**Proposed Implementation Language, Development Kits, and Equipment Needed**

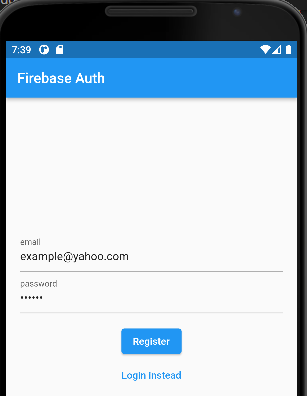
It was coded through the language of Dart. The database language was Firebase. The development kit that was to be used for this project is called Flutter. The application that was used to code all of it together was Visual Studio Code .

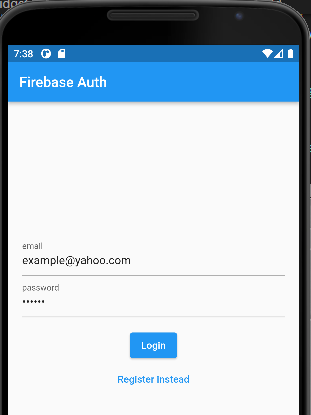
**Project Requirements**

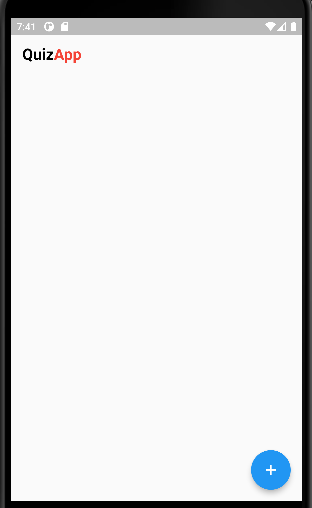
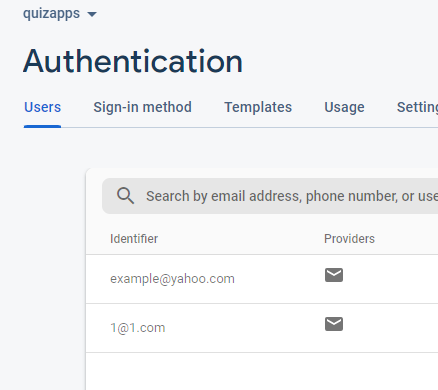
The minimum requirement for the product is to be a fully functional quiz maker application with at least an online cloud storage database. All of the pages need to work as they were desired in order to make the quiz maker app. If one page or function does not work then it will not be able to reach all of the desired functions and pages. The errors or exceptions should be caught in the case of an empty text field. The quizId, quizTitle, quizDescription are used to instantiate a new quiz instance. The quiz is not made until at least one question with the four options or answers are added to the instance. Finally, the desired quiz will be created by all of the data tied to the quizId to make its form onto the home page.

**Project Implementation Description & Explanation**

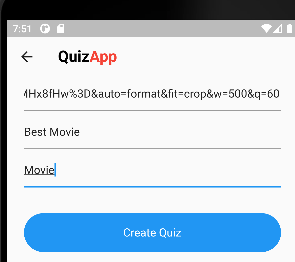
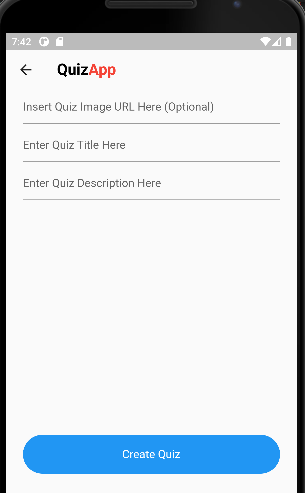
The website will provide several quizzes on the home page. In order for the user to access the home page they will be required to either create a new account or simply sign in with an existing account that is already made and stored on the website's storage database. See Fig.1, Fig.2, and Fig. 3. Refer to lib/pages/profile\_page.dart. 





Once, the user logs in then they can pick any of the quizzes that are provided. See Fig. 3 Refer to lib/auth.dart for the authentication method and lib/pages/home.dart for home page.



The floatingActionButton (shown in bottom right corner of Fig. 4) upon being clicked on Create Quiz page. The Create page which will allow the user to create their own quiz. It will first ask the user to enter an image URL, quiz title, and quiz description into the text field as seen on Fig. 5 and Fig. 6. Refer to lib/pages/create\_quiz.dart for Quiz Page.

The next page after the user submits their quiz header is where the user will be able to add in their own questions and multiple choice answers as seen in Fig.7 and Fig. 8. Finally, after the user finishes customizing their personality quiz and submits it will be added to the Firebase Database cloud storage as shown in Fig. 9. Refer to lib/addQuestion.dart for Question Page.

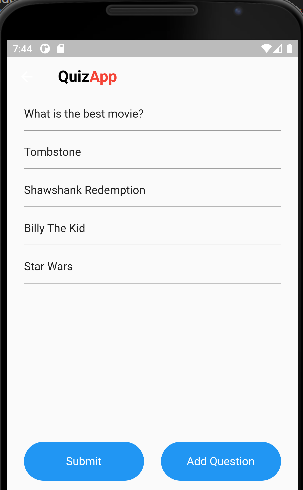
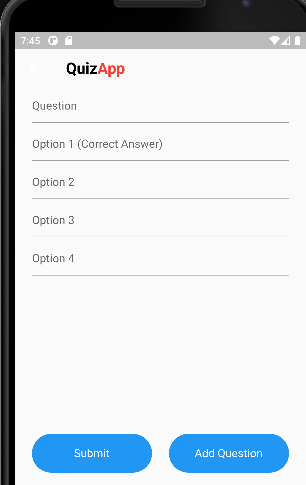
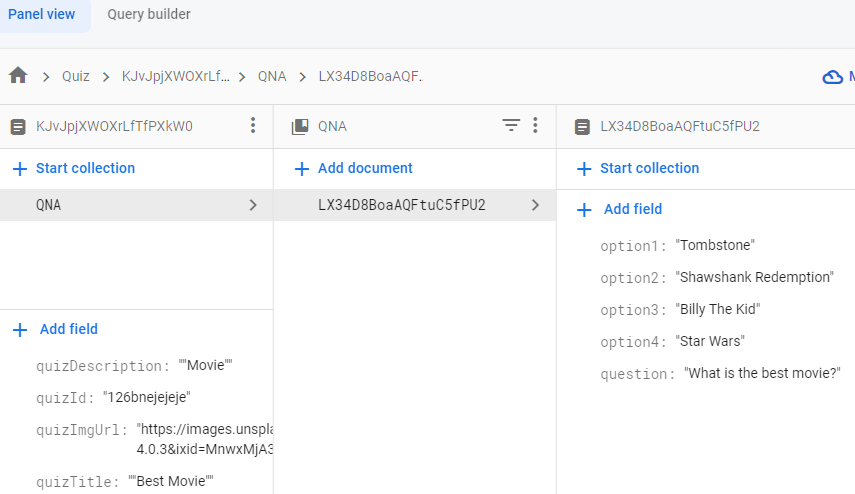


Fig. 9 Firebase Database

**Test Plan**

The test plan is graded on a scale from 1-5 for each scenario. 1 being very unsatisfactory to 5 being very satisfactory. Each test scenario will follow the test step with the test data.

| **Test Scenario** | **Test Case** | **Pre-conditions** | **Test Step** | **Test Data** | **Expected Result** |
| --- | --- | --- | --- | --- | --- |
| Check Login functionality | Check response on entering valid username and password. | Must have already created an account. | 1. Launch website 2. Enter username 3. Enter password 4. Click submit button | Username: JJKaisen2099@yahoo.com  Password:  luffy1Yonko | Login must be successful. |
| After login, user shall be taken to correct page | Check response on successfully logging in. | Must have already logged in with a pre-existing account. | 1. Login | None | Must be taken to the correct page (Home page). |
| Check Create button functionality | Check response on clicking the Create button. | Must have either logged in with an existing account. Does not display for guest login. | 1. Login 2. Click Create button | None | The button must take the user to the Create page. |
| Check Create Quiz page functionality | Check response on entering all of the ten questions and answers for each question. | Must have clicked on the Create button. | 1. Click on Create button 2. Enter all of the questions 3. Enter all of the corresponding answers | None | The page must allow the user to create a quiz with a minimum of ten questions with at least five answer choices for each question. |
| Check the links functionality | Check response on clicking a specific link. | Must already be on the Browse page. | 1. Click on a link to a particular quiz. | None | The links to the created quizzes should take the user to the correct particular quiz that is chosen |
| Check the Results page functionality | Check response on answering all of the questions | Must be on any quiz. | 1. Access a quiz 2. Answer all of the questions 3. Click on the submit button | None | The correct results should be displayed on the results page |
| Check the Account/profile page functionality | Check response on clicking the account button in the navigation bar | Must be already logged in (either guest or a pre-existing account) | 1. Login 2. Click the Account button | None | The button should take the user to the Account/profile page. |
| Check the share and save results functionality | Check response on sharing the URL of the result page and save it to the account | Must have at least logged in with a pre-existing account. | 1. Login 2. Finish a quiz 3. Share URL 4. Click on save results button | None | The results should be able to be shared or saved to their account/profile |
| Check results delete functionality | Check response on clicking the delete button next to a particular quiz on the Account page | Must have at least logged in with a pre-existing account and saved a result from a quiz | 1. Login 2. Click on Account button link 3. Choose a particular quiz 4. Click on delete button for the quiz 5. Click Yes button to confirm the deletion of the result | Choose the first quiz on the Account page to delete | The link to the particular quiz and results should no longer appear on the Account page. |
| Check database functionality | Check the response on accessing an account by logging into the website | Must have a pre-existing account that is already stored into the database | 1. Login | Username: JJKaisen2099  Password:  luffy1Yonko | Login must be successful. In which it does determine that an account was stored in the database. |
| Check the website’s attractiveness | Check the response of the test user(s) favorability on the website | Must have interacted with the website | 1. Interact with all of the functions of the website 2. Fill out survey | None | At least the majority rating the attractiveness as 4 or 5. |
| Check the website’s ease of use | Check the response of the test user(s) favorability on the website | Must have interacted with the website | 1. Interact with all of the functions of the website 2. Fill out survey | None | At least the majority rating the attractiveness as 4 or 5. |
| Check the website’s response time | Check the response of the test user(s) favorability on the website | Must have interacted with the website | 1. Interact with all of the functions of the website 2. Fill out survey | None | At least the majority rating the attractiveness as 4 or 5. |

**Test Results**

Testing was done by using the test steps and/or test data to see if it was able to achieve the expected result. The failed results mainly occurred because they were not added yet.

| **Test Scenario** | **Test 1** | **Test 2** | **Test 3** | **Test 4** | **Grade** |
| --- | --- | --- | --- | --- | --- |
| Check Login functionality | Passed | Passed | Passed | Passed | 5 |
| After login, user shall be taken to correct page | Passed | Passed | Passed | Passed | 5 |
| Check Create button functionality | Passed | Passed | Passed | Passed | 5 |
| Check Create Quiz page functionality | Passed | Passed | Passed | Passed | 5 |
| Check the links functionality | Fail | Fail | Fail | Fail | 1 |
| Check the Results page functionality | Fail | Fail | Fail | Fail | 1 |
| Check the Account/profile page functionality | Fail | Fail | Fail | Fail | 1 |
| Check the share and save results functionality | Fail | Fail | Fail | Fail | 1 |
| Check results delete functionality | Fail | Fail | Fail | Fail | 1 |
| Check database functionality | Passed | Passed | Passed | Passed | 5 |
| Check the website’s attractiveness | Passed | Fail | Fail | Fail | 2 |
| Check the website’s ease of use | Passed | Passed | Passed | Passed | 5 |
| Check the website’s response time | Passed | Passed | Passed | Passed | 5 |

**Challenges Overcome**

The main challenge of this project was to determine how I was going to make my desired application. I wanted to try something new and out of my comfort zone. I discovered Flutter first and learned that the best server-side language for it was Firebase and the best programming language for it was Dart. Flutter had many of the desired packages and it was more object-oriented which is a better choice in making an application.

During CSCI 498, I started learning how to code with Flutter, Firebase, and Dart for when it was version 2.0 at the time. Spring 2022 semester ended and so did version 2.0 of Flutter. Version 3 was a big update in which it further integrated Firebase into Flutter. Version 2 was deprecated so I had to update a lot of the dependencies and syntaxes. I could no longer run certain functions and dependencies due to them being deprecated so I had to research the new dependencies that were developed for version 3.0. The biggest challenge I was not able to overcome was migrating all of my programming from version 2.0 to 3.0 due to the small amount of tutorials and learning materials for version 3.0 since it is a new UI SDK and it just got a huge update.

**Future Enhancements**

The future enhancements for this project would be better UI, a search function, sharing quiz with friends and family, and more login methods. The UI could be made to look more professional as user interaction events and background. The search function would allow the user to find a quiz by the creator’s name or quiz title. The sharing function would allow the user to share the URL of the quiz with friends and family. The login and register page could also benefit with more login methods such as Google, Facebook, and Twitter.

**References**

Batschinski, G. (2022, August 10). *What is firebase? all secrets unlocked*. Back4App Blog. Retrieved October 15, 2022, from https://blog.back4app.com/firebase/

*Dart vs JavaScript: Top 6 differences you should know*. EDUCBA. (2021, March 3). Retrieved October 15, 2022, from https://www.educba.com/dart-vs-javascript/

Dziuba, A. (2022, August 11). *Top 8 flutter advantages*. Relevant Software. Retrieved October 15, 2022, from https://relevant.software/blog/top-8-flutter-advantages-and-why-you-should-try-flutter-on-your-next-project/