# Are you a true Terrier?

## Test your knowledge on

# Terrier Trivia

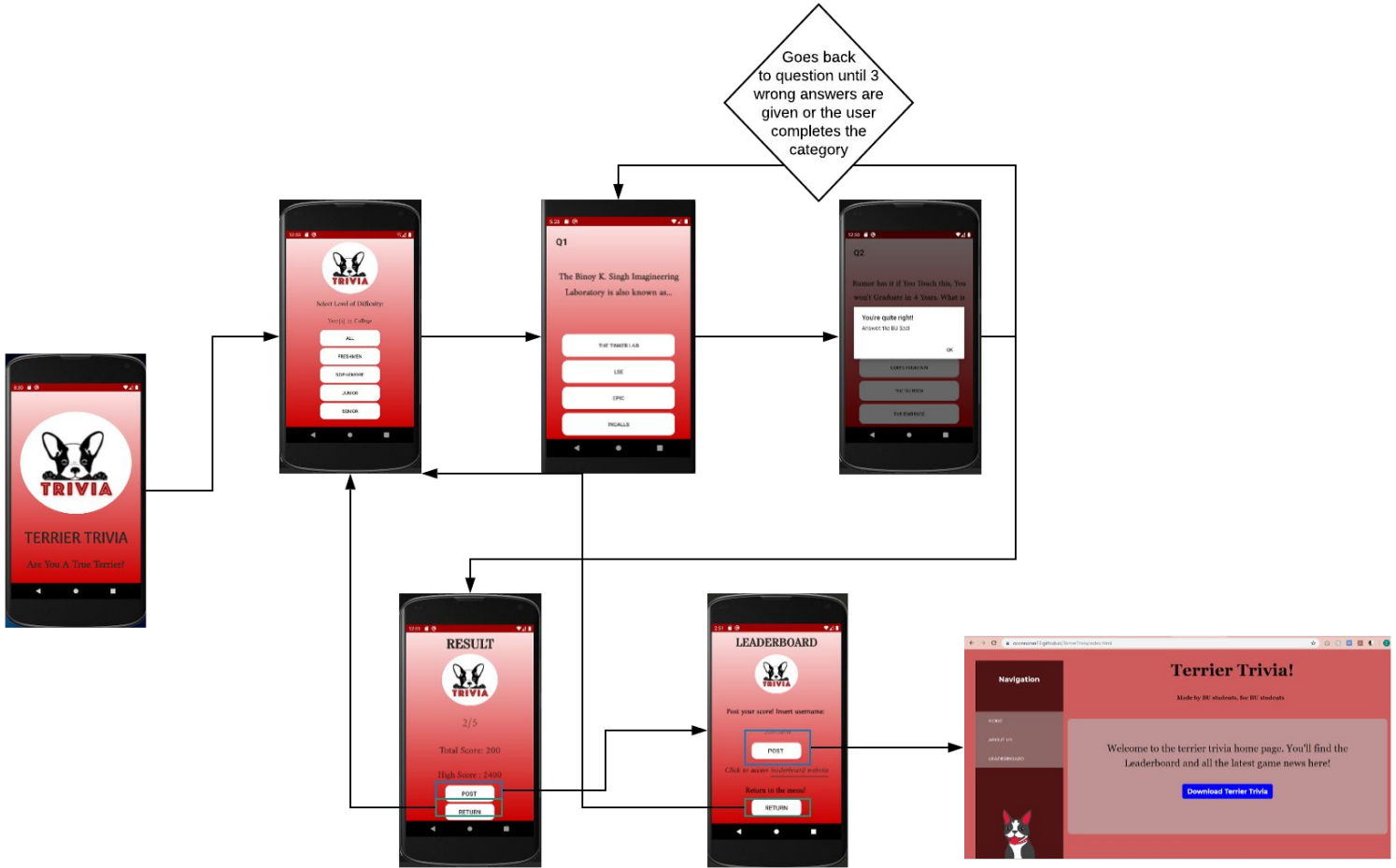

**Contributors:**

| | |
|---|---|
| Pooja Chainani | pooja13@bu.edu |
| Vanessa Giron | van526g@bu.edu |
| Esperanza Rojas Hernandez | erojash@bu.edu |
| Se'Lina Lasher | slasher@bu.edu |
| Alexandra O'Connor | oconnora@bu.edu |


# Who is the audience? What is the app's purpose?

- The 'Terrier Trivia' application is directed for our fellow peers & alumni at Boston University.
- The application is a fun way for students to learn about the history and current information regarding our university.

# Project Diagram

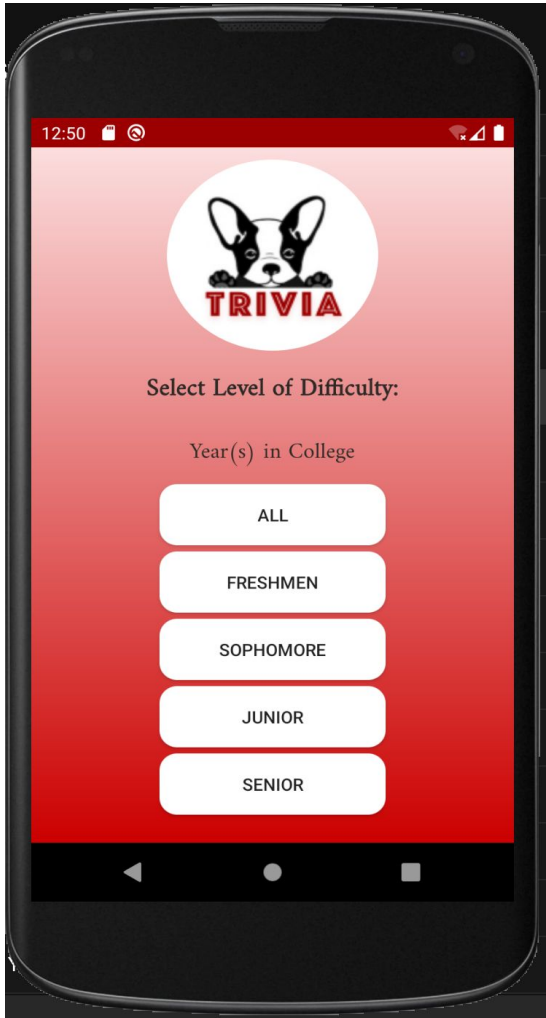


# Splash Activity

- This is the first screen that appears when the user opens the application. It consists of the app logo and slogan and appears for 3 seconds before transitioning into the next screen, which is the Start Activity Screen.

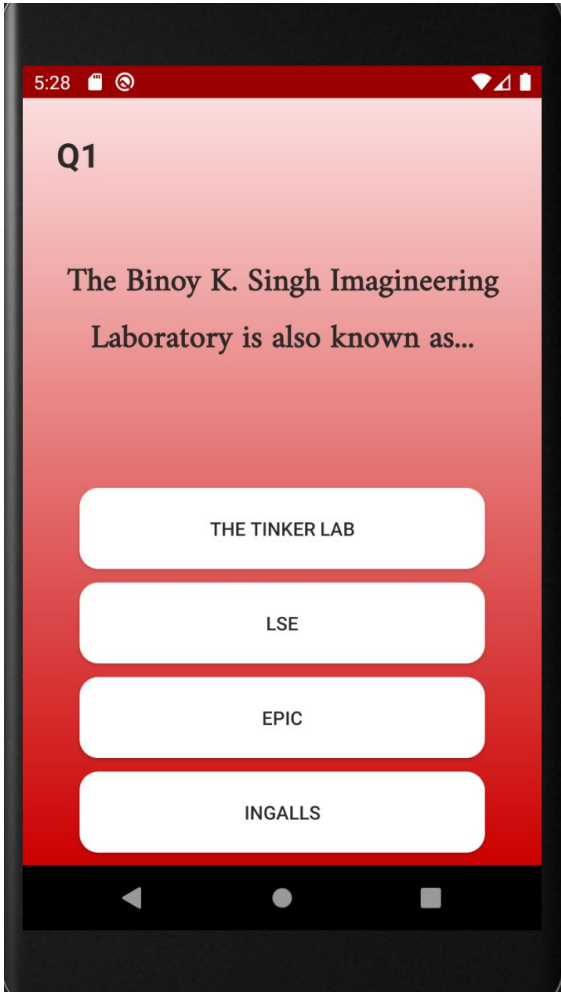
# Start Activity

- This is the main menu where the student have to select what level of difficulty they would like to play the trivia.
  - Freshmen is the easiest level
  - Senior is the most difficult level
- The ‘ALL’ button has all of the 100 questions.
- ‘FRESHMEN’, ‘SOPHOMORE’, ‘JUNIOR’ and ‘SENIOR’ difficulty levels each have a pool of 25 questions.
- The background design of the application contains the official primary colors of Boston University.
  - Scarlet red and white.



# Main Activity

- This is the the main part of the trivia application.
- The question and the four buttons that contain the choices are displayed on the screen.
- This was done by an array of strings that followed this format:
  - {"Question","Right Answer","Choice1","Choice2","Choice3"}
- While playing the trivia, the questions are randomized and are never repeated during a single game.
- The four options are also randomized every time a new game starts.
- After a player clicks on an option, two features occur: sound effects and random commentary



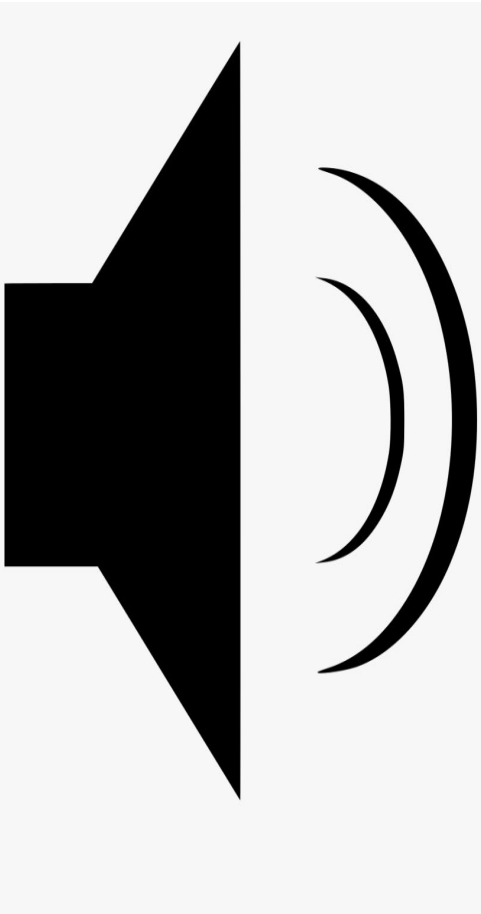
# Answer Check

- Answers are first checked in the Main Activity. A while loop that occurs until the user gets three questions wrong or finishes all the questions in their category.
- Wrong answer count is checked inside the while loop in an if else statement.
  - If the selected answer is answer selected matches the correct answer then the correct answer count is incremented as well as the total amount of questions answered.
  - Else, the wrong answer count is incremented and the total amount of questions answered as well
    - Once 3 wrong answers are counted the game ends and the user is brought to the results page

# Feature: Sound Effect

- Depending if the player gets the question correct or incorrect, a sound is played.
- The sound effects are implemented by:
  - Storing two audio files (correctsound.mp3 and incorrectsound.mp3) in the android file called ‘raw’.
  - In the ‘checkAnswer’ function of the class Main Activity, the MediaPlayer class is used to control the playback of the two audio samples.
  - Within the ‘checkAnswer’ function, there is an ‘if’ statement that compares if the choice made by player equals the correct answer to the question given.
  - If the choice made the player is correct, the ‘correctsound.mp3’ audio is accessed from the ‘raw’ file and stored in a new variable called ‘correctsoundMP’.
  - If the choice made the player is incorrect, the ‘incorrectsound.mp3’ audio is accessed from the ‘raw’ file and stored in a new variable called ‘incorrectsoundMP’.
  - In order to play the sound samples, ‘correctsoundMP’ and ‘incorrectsoundMP’ call the MediaPlayer built-in start() function.

```
if(btnText.equals(rightAnswer)){
    //Sound effect when answer is correct
    final MediaPlayer correctsoundMP = MediaPlayer.create( context: this, R.raw.correctsound);
            correctsoundMP.start();

}else{
    //Sound effect when answer is incorrect
  final MediaPlayer wrongsoundMP = MediaPlayer.create( context: this, R.raw.wrongsound);
          wrongsoundMP.start();
```



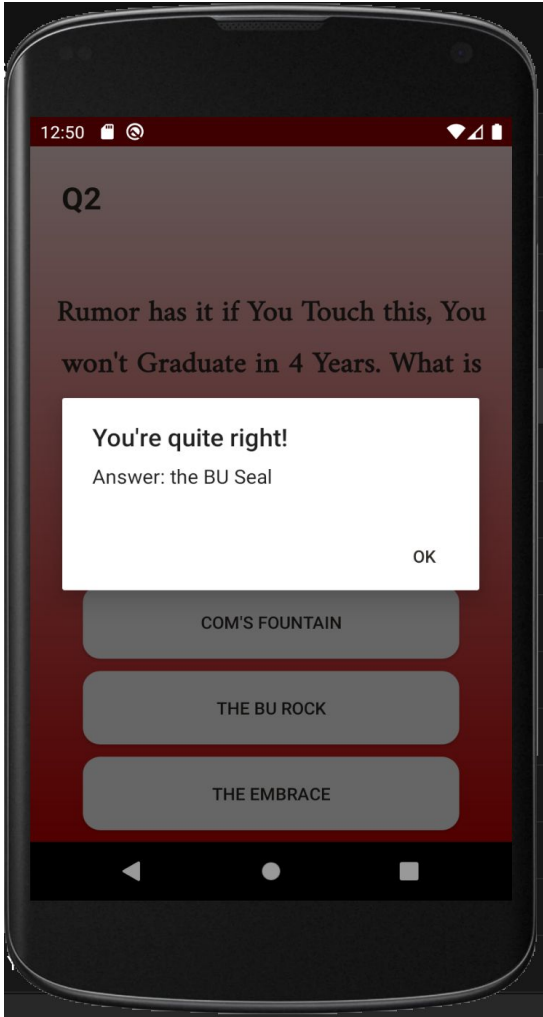
# Feature: Random Commentary

- Once the user selects an answer, a text block appears and it will inform them if they got the question correct or incorrect and tells them the correct answer.
- For every question, a random comment will appear after answering.
- The random commentary was done by making two string arrays.

```
// The correctResponses and incorrectResponses strings are used for the commentary after a user answers a question
String correctResponses[] = {"Correct!", "Great Job!", "Excellent!", "Fantastic!", "Absolutely!", "That's spot on!", "That's it!",
                        "You're dead right!","Easy peasy!","You're on fire!", "You're quite right!","Yes, that's very correct!"};

String incorrectResponses[] = {"Incorrect...", "Wrong...","You were close!","Not quite...","Not exactly...","You're mistaken...",
                         "Nope!","Better luck next time!","Oof..."};
```

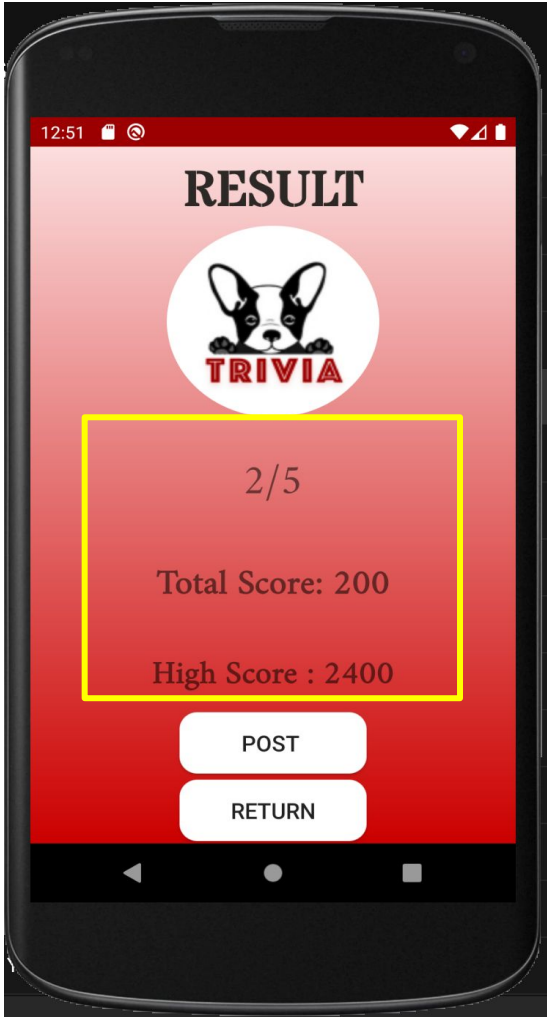
- A random number from within the string array's index range is generated and the string chosen from the array string is stored in the button "alertTitle".

```
            correctSoundMP.start();

            //Randomizes the correct reponses from the string
            int randomIndex = new Random().nextInt( bound: 11 + 1);
            alertTitle = correctResponses[randomIndex];
            rightAnswerCount++;
        }else{
            //Sound effect when answer is incorrect
            final MediaPlayer wrongsoundMP = MediaPlayer.create( conte
            wrongsoundMP.start();

            //Randomizes the incorrect reponses from the string
            int randomIndex = new Random().nextInt( bound: 8 + 1);
            alertTitle = incorrectResponses[randomIndex];
            wrongAnswerCount++;
        }
```



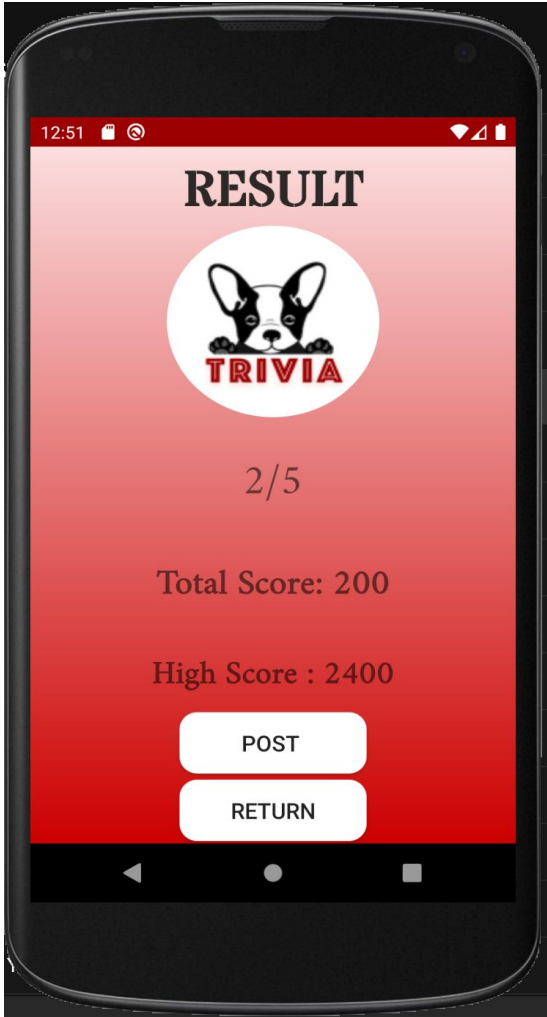
# Total Score & High Score

- The right answer count, category and total question count are passed from the Main Activity to the Result Activity.
- The right answer count is concatenated together with the total question count to display the amount fraction seen to the right.
- The right answer count and category are then passed to an if else statement to calculate and display the total score. Depending on the category the right answer count is multiplied by a certain number to display the total score
- The total score is then passed to another if else statement which determines if the user beat their previous high score. If they did the new high score is displayed. Else the high score stays the same.



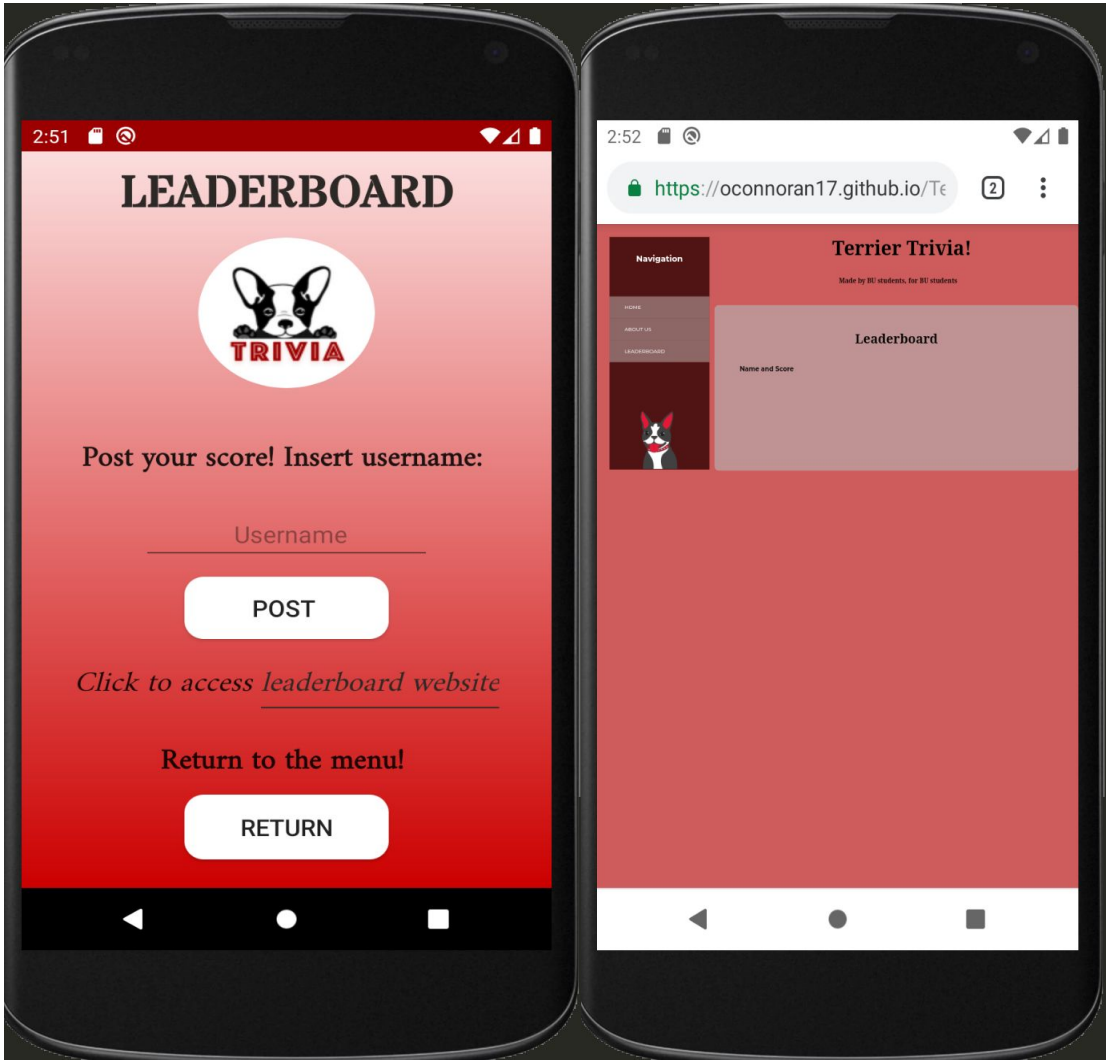
# Result Activity

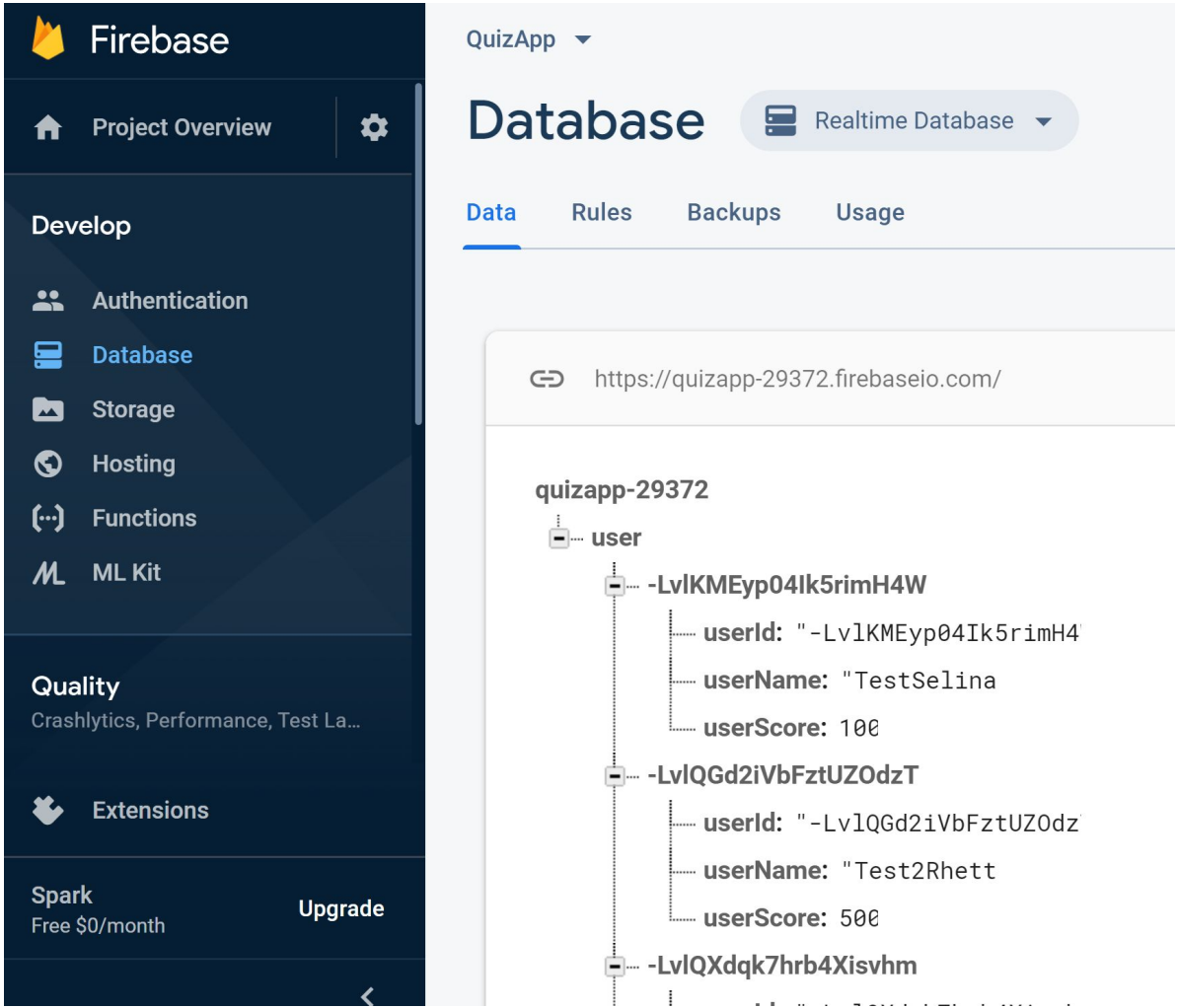
- After a user gets three questions wrong, they are taken to the result page that displays:
  - The amount of questions they got correct / total questions attempted
  - The total score
    - Each of the levels have a different amount of points the player can get.
      - Freshmen level questions: 100 points each
      - Sophomore level questions: 200 points each
      - Junior level questions: 300 points each
      - Senior level questions: 400 points each
  - The high score
    - The high score is saved in the application. It gets replaced when the player beats their previous score.
- The player then has the option to either post their score or return back to the main menu to start a new game.



# Leaderboard Activity

- If the player decides to post their score, they are directed to the leaderboard.
- Once the player enters their username, the post button with upload the username and their score to the leadership board website.
- The player has the option to check the leaderboard website by clicking on the link that is underlined under the post button.
- The player then clicks on the return button to take them back to the main menu to select a new level.



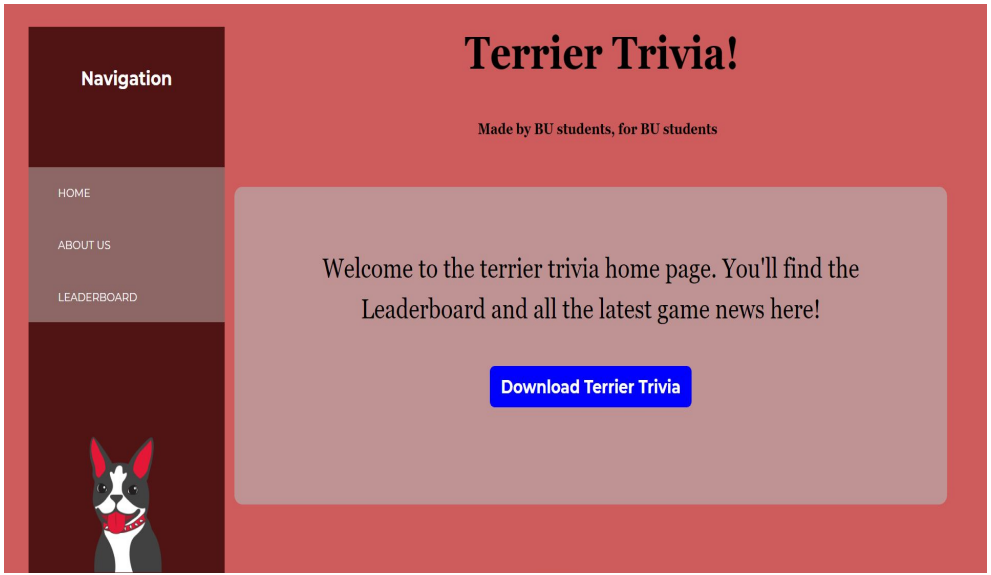



# Firebase Database

- Usernames and Scores Stored in Firebase
- Firebase is a NoSQL database, meaning data is stored in documents rather tables
- We chose Firebase because it had an easy to use platform and was because it was compatible with Android SDK
- Website queries the database the database every time that it's loaded keep scores recent
- May incorporate a function to sort the database

# Terrier Trivia Website

- The app's website incorporates web design principles such as style and user friendly features such as appealing colors and a navigation bar.
- The index page includes a welcome message as well as a download button that directs the user to downloading the app
- The about page includes a short statement on who the team members are and thank you to our extremely helpful TF Miguel Mark
- The leaderboard displays current top players' names and scores that draws information from 'app.js'



# Go Terriers!

Thank you to our mentors Neha Rai and TF Miguel Mark for all their help.

Finally, we hope you enjoy our app and its website!