Final Report

Product Name: Expertize
Customer Name: Krista Katzenmeyer
MMLK Gang
Mahia Tasneem, Maya Nichols, Kiara Bowen, LeAnn Lewis
12/07/2018

Table of Contents

SRD

SSD

UI

TR Page

Product Coding Summary Page

System Requirements

1. Introduction

1.1 Purpose of this document

We hope the document can serve as a basis for our projects functionality in a clear and concise form.

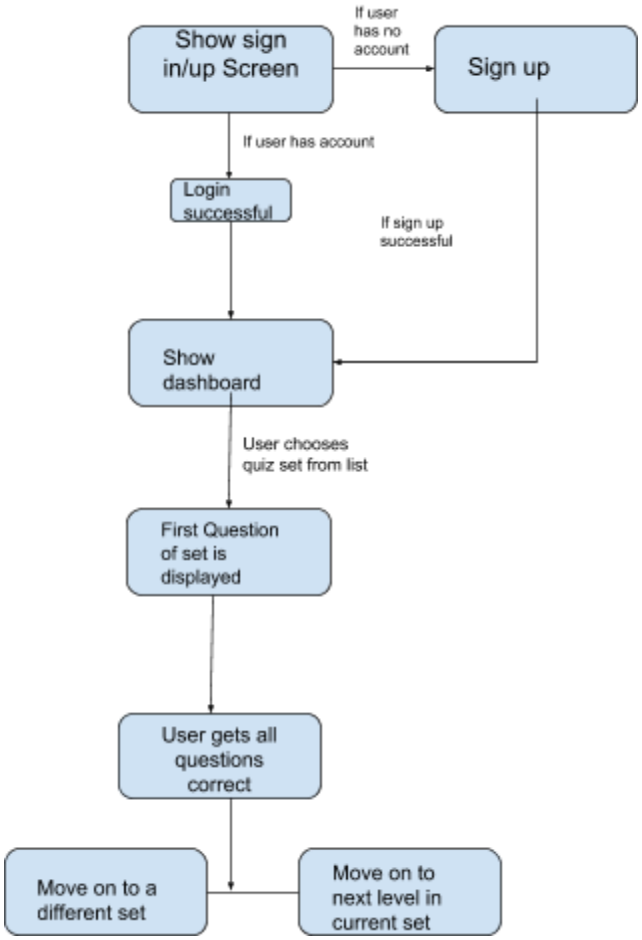
1.2 References

The Unified Modeling Language(UML): A Standard Graphical Notation presentation by Eddie Roache
UML Distilled by Martin Fowler
Software Engineering, 10th Edition by Ian Sommerville

1.3 Purpose of this product

The purpose of this product is to create a fun website application in which users can test their knowledge about many facts concerning a specific topic.

1.4 Product Scope



2. Functional Requirements

See use case diagram referred to in **Section 1.4.**
The following diagrams describe a few user cases for the operation of our website.

Number	1	
Name	Account Creation	
Summary	The website should allow users to create an account.	
Priority	5	
Preconditions	User does not already have an account	
Postconditions	User will have an account that they can log in and out of	
Primary Actor	users	
Secondary Actors	< >	
Trigger	Clicking on a join or create account button	
Main Scenario	**Step**	**Action**
	1	Navigate to the website: Expertize.com
	2	Click on “Create an Account”
	3	Follow the steps to create an account using email and primary information
	4	Verify information and proceed to website, signed in as the user
Extensions	**Step**	**Branching Action**
	1a	User tries to create/access a challenge **:** < navigate to sign up page >
Open Issues	Unique account creation under one email needs to be implemented	

Number	2	
Name	Create Trivia Questions	
Summary	Users should be able to create Trivia Questions for other users to answer	
Priority	3	
Preconditions	All users must have an account and be logged in	
Postconditions	Users will be able to create questions on the topics selected in MCQ	
Primary Actor	users	
Secondary Actors	other users	
Trigger	users will be prompted to a page to create trivia challenges in their selected format and store them in the database	
Main Scenario	**Step**	**Action**
	1	select “ create trivia challenge” from home page
	2	users will be navigated to a new page where they’ll be able to add questions and submit them
Extensions	**Step**	**Branching Action**

	1a	Users can take a default quiz from the database to understand the format to create questions
Open Issues	Ensuring that the questions meet requirements and are appropriate according to different user levels.	

Number	3	
Name	Deleting created challenges	
Summary	Users should be able to delete challenges they created before	
Priority	3	
Preconditions	Only the user who created the challenge will be able to delete it; Need verification from the website administrators before deleting it	
Postconditions	The deleted question will be removed from the database	
Primary Actor	users creating the quiz	
Secondary Actors	other users and system administrators	
Trigger	Quiz creator should navigate to Quiz (Options -> Delete)	
Main Scenario	**Step**	**Action**
	1	A dialogue box opens to request system administrators to delete the selected quiz and options are provided to specify the reason of deletion
	2	If approved, the trivia is removed from the database
Extensions	**Step**	**Branching Action**
	1a	The quiz selected to be deleted will be temporarily hidden from other users to
Open Issues	Ensuring deleted challenges are removed from the database Creating an option for users to cancel deletion until system approval is provided	

Number	4	
Name	Taking challenges	
Summary	Logged in users should be able to take either default challenges from the website or other user-created challenges	
Priority	4	
Preconditions	Users must be logged in	
Postconditions	Users should be able to navigate to selected challenges and earn points as they answer them	
Primary Actor	logged in users taking the challenge	
Secondary Actors	other users	
Trigger	“Expertize?” should navigate to the selected quiz	
Main Scenario	**Step**	**Action**
	1	user should select a genre/topic and difficulty level and choose a quiz from the available quizzes on that topic

	2	users should see a +1 for every correct answer and a -1 for every wrong answer
	3	at the end of the quiz, they should be able to see the total scores earned
Extensions	**Step**	**Branching Action**
	1a	users should only be able to attempt one quiz once a day
Open Issues	Ensuring users are not able to attempt a quiz multiple times	

Requirement 1 Test- Will create multiple accounts on different platforms to make sure it's fully functional

Requirement 2 & 3 Test- Will manually check that the option to create and delete questions is accessible and functional

Requirement 4 Test- Test by actually playing the game and checking that choosing the correct answer allows you to progress through the rest of the questions and receive your score in return

3. Non-functional requirements:

1. Sign up verification will only be done by email
2. Questions can either be in multiple choice format or in fill-in-the blanks format
3. Questions with a multiple choice format will have four options
4. Questions with a fill-in-the-blanks format will have a hint
5. Questions should be in the order of Easy-Medium-Difficult
6. Users will have two attempts to answer each question
7. Users will only be able to try a quiz once
8. Users should be able to view their scores at the end of every quiz

3.1 Using Expertize:

1. Users should log in to the website
2. They can either navigate to:
    a. Create a new Quiz
    b. Take a quiz created by a user
    c. Start a default quiz from the database
3. Quiz creators should:
    a. Choose a genre/topic/category
    b. Declare the level of intensity and format (multiple choice or filling blanks)
    c. Order questions according to difficulty
    d. Have at least 10 questions
4. Quiz takers can:
    a. Access any quiz
    b. Have at most one try to take a quiz
5. Every wrong answer will reduce the current score by 1 and display correct answer
6. At the end of every quiz, users will be navigated to the home screen where they can try more quizzes

4. User Interface

See "User Interface Design Document for *Expertize".*

However, as a short summary*:*
Expertize is a site for people of all ages and backgrounds who have one thing in common, and that is their thirst for knowledge. Our site requires no training other than the ability to use a computer and by extension, the internet. Our system has a simple point and click interface where all the user has to do is click on their chosen answer from the choices provided and if they got it either correct or incorrect (they will be prompted as to which it is), they will move on to the next question. It is of the utmost importance that our system is not only easy to learn but also to use in order to foster a hugely diverse fan base not only in background but also in age range.

The only input/output devices needed for our product is a mouse (or touchpad, joystick etc.), a monitor and an internet connection (simplicity at its finest) as the game will be stored on our servers and remotely accessed by the users.

5. Open Issues

These are the issues that, as a team, we will be working to fine tune in the development of our product:

- How will we regulate high scores on the site and keep all the trivia information updated?
    - Will the high score be based on how long the player took to complete a trivia set? or by how many tries it took them to do so.

Software Systems Design

1. Introduction

1.1 Purpose of this document

This document serves to outline the interior design of Expertize, that is, including but not limited to the databases involved as well as the relationship between what the users see and the code that we write.

1.2 References

- MMLK Documentation:
    - *System Requirement Specification Document*
    - *User Interface Design Document*
- *Software Design: Requirements to Code*

2. System Architecture

2.1 Architectural Design

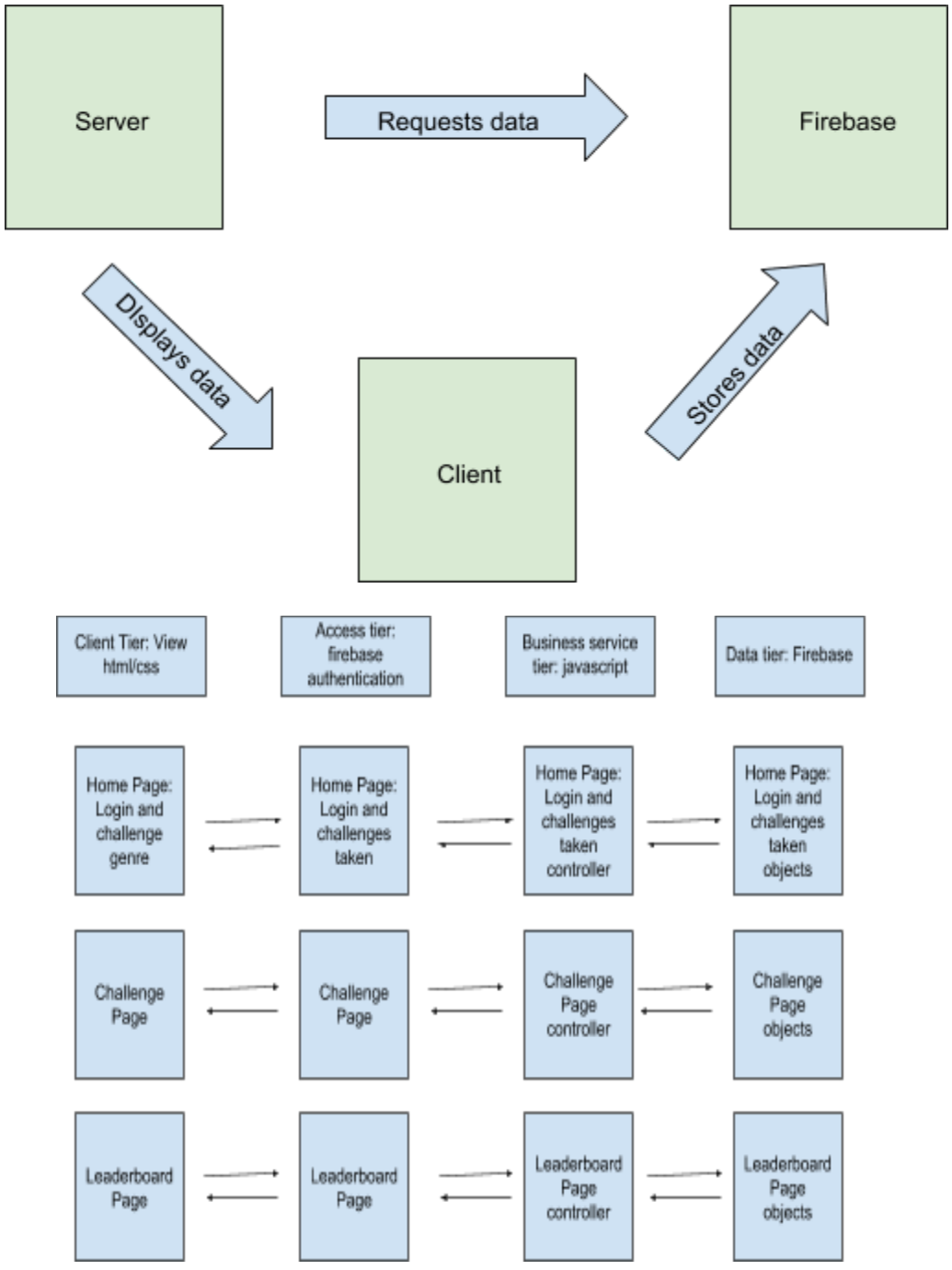
All the system's data will be stored in Firebase. The Server will request the data from the Firebase storage and provide it to the clients through GUI. The user provided information

will also be stored in Firebase. The entire system operations will mainly be navigated back and forth with the use of Firebase.

For example, when the user enters their login information, their credentials will be stored to Firebase and later when they try to log back in, those will be requested by the server to pull from the firebase for the corresponding user.

2.2 Decomposition Description

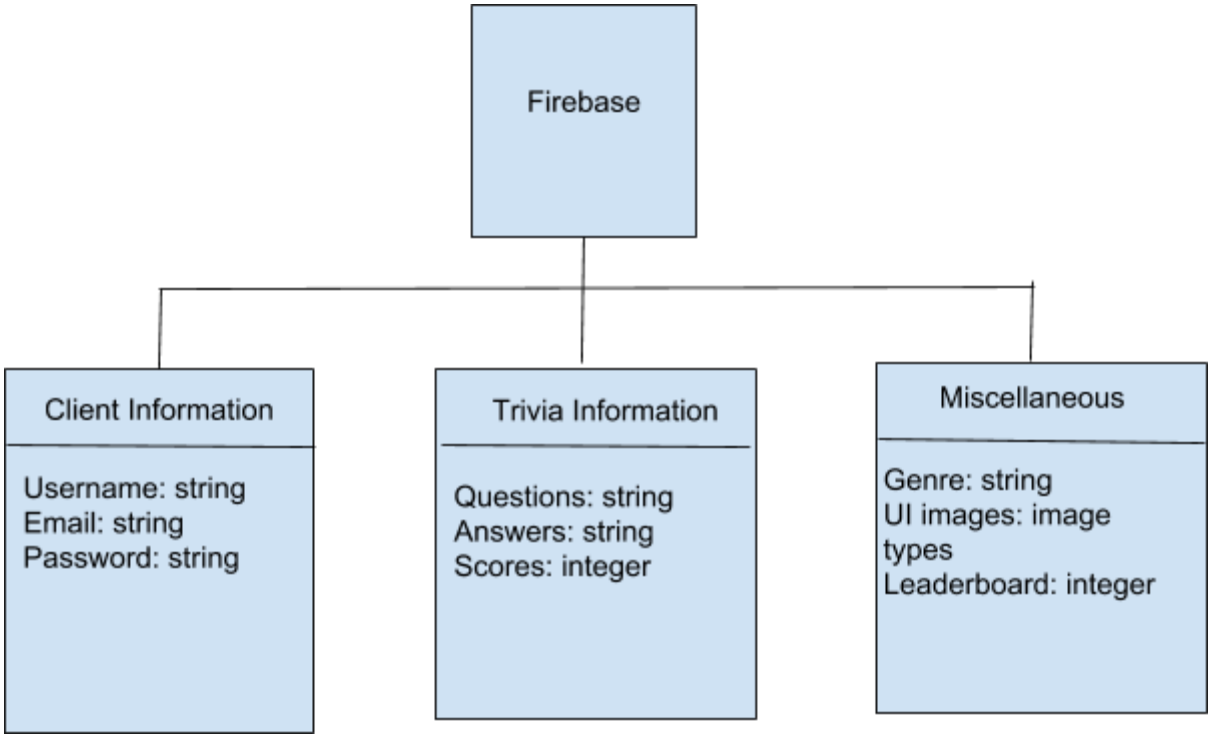
We are using a MVC approach combined with Object Orientation to design our system. The following diagram shows an overview of this. Further UML and MVC diagrams will be added later once the entities and their attributes are confirmed.



3. Persistent Data Design

3.1 Database Descriptions

The databases used by the system are: Firebase



We created this table that helps describe the relationship that Firebase has with the client, trivia, and miscellaneous information. For the client information we will store the username, email, and password as strings. For the trivia information we will store the questions, answers, and scores by using strings and integers. The strings and integers will be used to describe our questions and allow the user to make a response. For our miscellaneous information we'll need to be able to store the genre, UI images, and leaderboard.

3.2 File Descriptions

No files are being used because everything that we need is constructed in an efficient manner instead of an arbitrary and unstructured way. You can tell this with the information from above - each data type is in a specific way not in a random manner.

4. Requirements Matrix

Requirement Number	Requirements	Functions Responsible
1	Account Creation	CreateUser()

2	Create Trivia Questions	NewChallenge()
3	Deleting created challenges	DelChallenge()
4	Taking challenges	StartSet()
5	Saving Progress/time taken to complete	SaveState()

User Interface Design

1. Introduction

1.1 Purpose of this document

This document will outline the approach to the design of our application, “Expertize”. It specifies our design standards and expectations for our app.

1.2 References

- MMLK Documentation:
  - *System Requirement Specification Document*
- *Software Design: Requirements to Code*

2. User Interface Standards

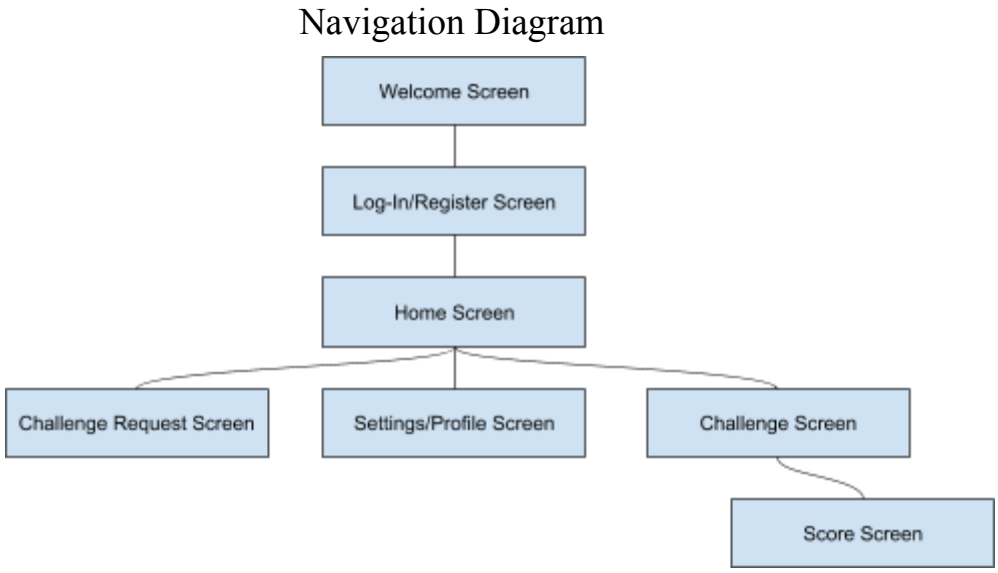
The general color of the pages on our site will be a dark background (mostly black which fades into a very dark blue in some places). This is done in order to give our site more of a videogame-like look and feel as well as enhance the visuals of our challenges as they will be a lighter color.
Our site will include some standard buttons to help the user navigate it such as the login button, home button, and, of course, the icons for the challenges will act as buttons as well.

The login button (on the login screen) will have the standard look of many login buttons (a small rectangle with rounded edges with the word “login” on it) and prompt you to use one of your google accounts to sign in to the site. The home button, which will be located in the upper right-hand corner of all your screens (except the login screen) will basically be the word “home” written in white writing with the background of whichever page you are on, and will navigate the user back to the main screen of the site (where you start after logging in). Finally, the primary image for each of the challenges (their icons) will act as a button of sorts because once clicked they will open to reveal your question set.
Finally, we will have a button to the right of the “Home” button labelled “Requests” (in the same style as the home button). This button will allow user to request that a challenge set be added to the site by filling out a short form that will be sent to us, the

administrators. These buttons are all present and function to help the user navigate our site in the most hassle-free way possible.

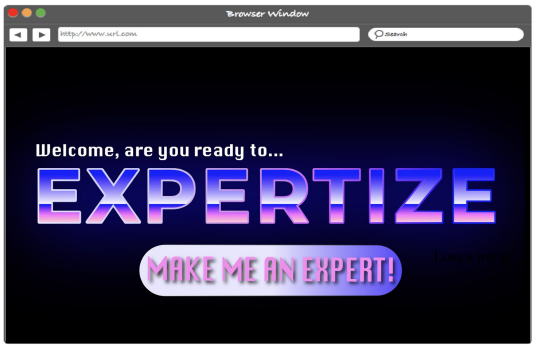
The login screen (as can be seen in Figure 1, section 2.1) offers you the option of signing in with your google account as this is much more streamlined and easier than having the users create and maintain a whole new account on our servers. The most common component of each screen is the color scheme and, once you are past the login screen, buttons that allow you to switch between a challenge you are attempting and your home screen.

2.1 User Interface Walkthrough

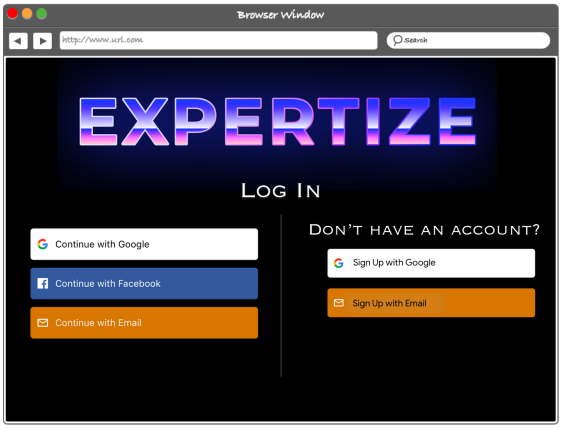


This diagram represents the paths that can be taken from each screen of our application.

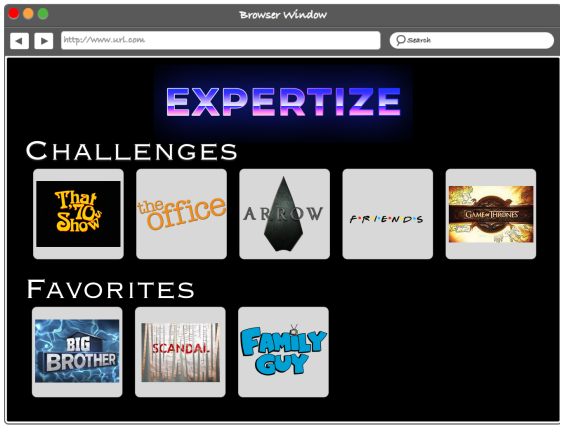
Users will start off on the Expertize Welcome Screen. On this page users that are already logged in will be automatically transferred to the home page. New users or users that aren't logged in will be brought to a welcome screen.



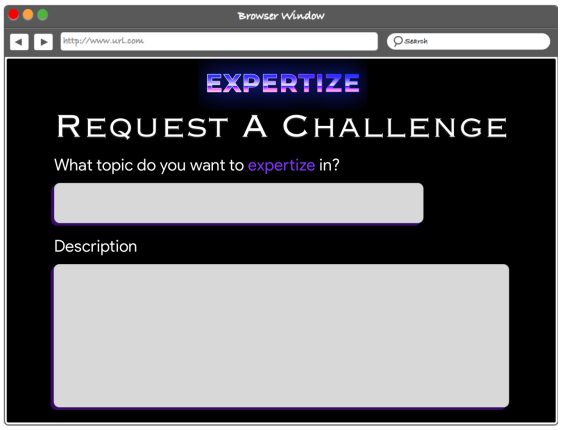
From there you are brought to our Log-In/Register Screen. This screen contains an option to either sign in if you have an existing account or create an account if you are new.



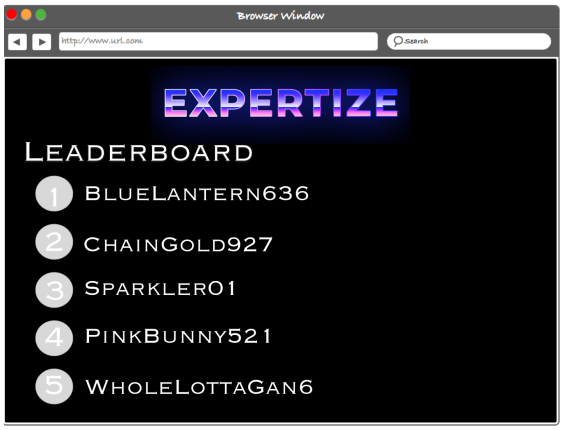
On our Home Screen, users will see available trivia challenges that they can take. They'll also see a section with their "Favorites" or challenges that they've taken frequently.



From the Home Screen, users can either go to their Profile Settings Screen, a Challenge Screen, or the Challenge Request Screen. On the Profile Settings Screen, users can change or view account info. The Challenge Screen will begin the whole trivia game. Users will be presented with a question pertaining to the topic and answer choices. Answering incorrectly will just not allow the user to earn any points whereas answering correctly will advance you to a new question and increase your score. On the Challenge Request Screen, users will be asked to enter a topic they'd like to see added to Expertize and given the option to describe exactly what type of questions they'd like this set to pertain to.



From the Challenge Screen, users will eventually be brought to the Score Screen. On this screen, users can view their score and see the top scorers in the challenge set they're on. The user will be able to view their friends or other players placement and hopefully, it'll boost their drive to play in order to become first on the leaderboard.



2.2 Data Validation

Data Item	Data Type	Limit	Format
Login information(google email)	String and integer	The limit is however long your google email is (aka it inherits the limits and validation of gmail authentication)	Printable ASCII

Trivia genre	String	Genre should be selected from a list of options	Printable ASCII
Trivia Questions/ Answers	String/Integer/Boolean	String values for answers should be lowercase	Printable ASCII
Score	Integer	Can be a maximum of 50.	Printable ASCII

Testing Report (TR)

1. Introduction

1.1 Purpose of This Document

To help users and creators ensure proper functionality of all aspects of the product.

1.2 References

- MMLK Documentation:
  - *System Requirement Specification Document*
  - *User Interface Design Document*
- *Software Design: Requirements to Code*

2. Testing Process

2.1 Description

We decided to test the backend API that implements the core functionality and writes to the database by comparing the expected changes with the test results for all the use cases, with variations for boundary cases.

We have a unit-test suite that tests all the endpoints on the backend with most of the boundary cases. In addition, we have used Jasmine and Karma on the frontend to test the individual Angular components. Lastly, we have used manual testing for Integration testing and end-to-end testing.

2.2 Testing Sessions

Testing Session	Date	Location	Time Started	Time Ended	Performed by	Use Case(s)
1	11/20/2018	Google	4:00pm	5:00pm	Mahia Tasneem LeAnn Lewis	Account Creation using Google, Facebook and Twitter **Use Case 1**
2	11/22/2018	Google	7:00pm	8:00pm	Mahia Tasneem Kiara Bowen	Js function to pull quiz questions, answers and hints from the database **Use Case 4**
3	11/22/2018	Google	8:00pm	9:00pm	Mahia Tasneem Maya Nichols	Dynamically pulling questions based on which challenge/show was requested **Use Case 4**
4	11/25/2018	Google	4:00pm	6:00pm	Mahia Tasneem	Create Challenges from user input and store in firebase **Use Case 2**
5	12/01/2018	Google	4:00pm	5:00pm	Mahia Tasneem LeAnn Lewis Maya Nichols Kiara Bowen	Deleting challenges **Use Case 3**

2.3 Impressions of the Process

Overall, the testing process was very effective to our product. We managed to recognize the errors in time (for the most part) and had ample time to fix them. We also recognized the

features that were working and got the time to make improvements to them. The testing process helped us achieve a successful product outcome.

Since our project depended solely on authentication integration with google, facebook and twitter (therefore we did not have to store usernames ourselves) and reading and writing information to a single database, we did not have to write too many tests. However, we still got a very good idea of how important the testing process is as it aids in finding and fixing errors early in the developmental process.The system relies on the Google Authentication for user login because we feel that this would be safer and more secure than anything we could write ourselves without proper roll out and testing.

We have used various combinations of html and css files for creating and displaying the UI for the homepage, as well as the rest of the website and gameplay. In addition,
we have carried extensive manual testing to ensure that our homepage works properly.

Best Modular Unit: Login, Take challenges
Reason: All the functionalities work perfectly, leaving us opportunities to further improvise.

Worst Modular Unit: Create Challenges
Reason: User input may override the information existing in firebase and result in existing challenges not working

3. Test Results

Test cases that ran and the actual result was identical to the expected result:
Tester: Mahia Tasneem
Testing sessions 1,2, 3 and 4

Test cases that did not run or pass:
Tester: Kiara Bowen
Defects: Facebook login authentication was missing developer requirements
Suggested Repair: Update developer requirements in the main js file of the program.

Changes made in the program:
UI Layout & Design, “Delete Challenges” & Facebook Login feature had to be taken out.

Product Coding Summary

Team: MMLK Gang **Date:** 12/07/2018

I. Overall Product Information

Programming language(s): Javascript, HTML and CSS

GUI or text-based? GUI

Database software used (if any): Firebase realtime database

Operating system: Windows and Mac

Other software used (if any): Bootstrap

Total physical lines of original (written by team) code (including HTML) in program, **including comments**: 1700
Total physical lines of original (written by team) code (including HTML) in program **without comments**: 1200

Project Manager:LeAnn Lewis

II. Programmer Information

Programmer Name	Names of Modules Coded	Approximate Percent of Program Written Overall
Mahia Tasneem	App.js, Homepage.html, Comedy Shows.html, RomanceShows.html, ThrillerShows.html, CreateChallenge.html, ShowChallenge.html	35%
Kiara Bowen	index.html, App.js, Homepage.html, Comedy Shows.html, RomanceShows.html, ThrillerShows.html, CreateChallenge.html, Quizpage.html, ShowChallenge.html	35%
Maya Nichols	App.js, TakeChallenge.html	15%
Leann Lewis	App.js, tests.js, index.html	15%

III. Comments (if any)
No more comments at the moment.

IV. Link to Codebase

https://github.com/mahiatasneem26/ExpertizeGames.git

V. Link to Live App

https://expertize-games.firebaseapp.com

VI. Certification

I have read the information in the table above and agree as to its accuracy. Any comments I wish to make are given below next to my name.

Printed Name	Signature	Date	Comments
Mahia Tasneem	Mahia Tasneem	12/07/18	No additional notes.
Maya Nichols	Maya Nichols	12/07/2018	No additional notes.
Kiara Bowen	Kiara Bowen	12/07/2018	No additional notes.
LeAnn Lewis	LeAnn Lewis	12/07/2018	No additional notes.

Appendix A – Agreement Between Customer and Contractor

The parties, to be later mentioned, hereby sign this document agreeing that the product in question *Expertize Games*, the online trivia gaming hub, will be maintained and updated by its creators and administrators at MMLK. The customer agrees to only use the site for its stated purposes and within the parameters of not only "safe internet practices" but also within the confines of the law. On the part of the administrators, they agree to keep the site as factual as possible and to make any updates/fix any maintenance issues necessary in as timely a manner as its day-to-day operations permit. The system will be backed up monthly initially however this is subject to change as its user-base expands.

If for any reason any of the terms, conditions or other parts of this document need to amended, all parties will come together once more to ensure that the new terms of this agreement are fair and agreed upon by all. in order to ensure this, any changes affecting the administrative and only the administrative side will be voted upon by the parties within MMLK and a majority vote will decide the fate of those changes. Any changes that will affect both the site and, by extension the customers, will be brought before everyone, and, as before, the majority vote will decide the ruling. If at any point there is a tie in the voting, the Product Manager's vote will be the deciding factor.

By signing below, the following parties are stating that they have read and agreed to the terms stated above and also all members of the team have read and agree on the content of not only this document but also the format in which it was written and have no major concerns to be noted:

LeAnn Lewis (Administrator):	**LeAnn Lewis**	**Dec. 5, 2018**
	Signature	**Date**
Mahia Tanseem (Administrator):	**Mahia Tasneem**	**Dec. 5, 2018**
	Signature	**Date**
Maya Nichols (Administrator):	**Maya Nichols**	**Dec. 5, 2018**
	Signature	**Date**
Kiara Bowen (Administrator):	**Kiara Bowen**	**Dec. 5, 2018**
	Signature	**Date**
Krista Katzenmeyer (Customer):	**Krista Katzenmeyer**	**Dec. 5, 2018**
	Signature	**Date**

Appendix B – Team Review Sign-off

By signing below all members of the team have read and agree on the content of not only this document but also the format in which it was written and have no major concerns to be noted (any minor concerns will be left in the comments table below).

LeAnn Lewis (Administrator):	**LeAnn Lewis**	**Dec. 5, 2018**
	Signature	**Date**
Mahia Tanseem (Administrator):	**Mahia Tasneem**	**Dec. 5, 2018**
	Signature	**Date**
Maya Nichols (Administrator):	**Maya Nichols**	**Dec. 5, 2018**
	Signature	**Date**
Kiara Bowen (Administrator):	**Kiara Bowen**	**Dec. 5, 2018**
	Signature	**Date**

Team Members:	**Comments:**
Maya Nichols	
Mahia Tanseem	
Kiara Bowen	
LeAnn Lewis	

Appendix C – Document Contributions

Kiara Bowen- Requirements Matrix, Functional Requirements, Purpose of UI, Navigation Diagram, Mocks

Maya Nichols- File Descriptions, Description for mocks, Non Functional Requirements

Mahia Tasneem- System Architecture, Database Descriptions, Functional Requirements, Testing

LeAnn Lewis- Requirements Matrix, Interface Open Issues, UI Standards, Data Validation, Testing