Andrei Martins
Ivaylo Gatev
Joshua Kornfeld
Rami Bader
Sevde Yanik
Yatharth Upmanue

Table of Contents

1.0 - Vision

1.1 - Introduction

For people who wish to play and/or easily create their own quiz games, Quizigma provides a very intuitive user interface that does not require from the user any knowledge in programming or creation of apps. Quizigma will give its users the chance to play quiz games not just as personal entertainment but also, for example, students who need support for learning materials, like distance learning.

1.2 - Positioning

1.2.0 -Problem Statement

The problem of	many current quiz games require the user to pay (sometimes high) prices to get full access to all the features or complicated interfaces for personalized games.
affects	all the people who just want to play a simple quiz game or create one but lack the knowledge on how to interact with such technology.
the impact of which is	a frustrating experience, especially for students who wish to use it as an extension tool for their learning process.
a successful solution would be	a quiz game that allows people not just to participate but create their own games with an environment interface that is simple to use and would allow a game to be created with just a couple of clicks.

1.2.1 -Product Position Statement

For	the people.
Who	want an easy and fun way to play quiz games.
The Quizigma	is an app and web game.
That	allows people to play quiz games for free with an easy and friendly interface.
Unlike	existing games that have complicated UI or enforce the user to have a premium account to access all functionalities.
Our product	will provide a totally fun and easy way to create personalized quiz games.

1.2.2 - Overall Vision

- Quizigma is an application for creating and playing interactive quizzes. The first window the user will see is a sign up or login selection, where the user login using his/her email and password. If the user selects sign up the user must pick a nickname, email and password.
- Each player will have an account of his/her own that shows the score and the place in the leaderboards for the different quizzes.
- Quizigma is an application that shall provide an interface that allows the users to play the game in 2 modes:
    1. **Creation Mode:** the users create their own quiz games by creating a question and the options to answer. The user should also select to which category the quiz belongs to. A quiz can be set for personal use or for multiple users. After the game is created, a random ID for that will be generated so that the user can share with other people. The person who created a quiz made for multiple users can't compete on the leaderboard.
    2. **Playing Mode:** the users participate in quiz games created by other people or use the game they created for themself. To be able to do that, the user should have the ID number of the game or search for the quiz according to is categorie in the "Category" screen.
- Each game shall have at least 2 questions. Each question shall have at least 2 answers, of which 1 will be selected by the user as the correct one.
- The creator of the game can set how long the participants have to answer each question.
- At the end of the game, a scoreboard will be displayed with the results. The participant with most correct answers will be selected as the winner. In case of a draw, the participant who finished the game first will be selected as the winner.

1.3 - Stakeholder Descriptions

1.3.0 - Stakeholder Summary

Name	Description	Responsibilities
Customers	All the people who will use the application.	Interaction with the app. Creation of the content itself (the quiz game).
Product Owner	Responsible for setting the Backlog and Features of the project.	Describe his requirements for the project.
Development Team	Responsible for the development of the app.	Product Development. Product Testing. Database Creation.
Maintenance Team	Responsible for the maintenance of the product after it's released.	Find and fix bugs, which weren't found during the testing phrase
Database Administrator	Responsible for the management of the database.	Managing the database.

1.3.1 - User Environment

- The game can be played on both Web-Browser and mobile phones(Android and iOS).
- The app can be downloaded from Google Play(for Android) and App Store(iOS).
- An Internet connection is required.

1.4 - Product Overview

1.4.0 - Needs and Features

Need	Priority	Features	Planned Release
Enhance the easy process to create a quiz game with very few clicks.	Required	Input from the user, simple template for quiz maker.	November/2020
Select the winner of the game.	Required	Counter to track the correct select and answers and the time it took for the user to finish the game.	Future releases
Offer a fair game.	Required	Prevent the quiz creator from competing in the quizzes that he created.	Future releases
Language translation	Optional	Translate the UI of the app in different languages.	Future releases
Google and Facebook login	Optional	The user can use their Google and Facebook credential to log in.	Future releases
Medals for 1st, 2nd and 3rd places	Optional	The user receives a medal for finishing 1st, 2nd or 3rd on the leaderboard of a quiz.	Future releases
The quiz game	Optional	The user can add	Future releases

supports short written questions		questions, which need to be answered by typing a word or a short phrase.	

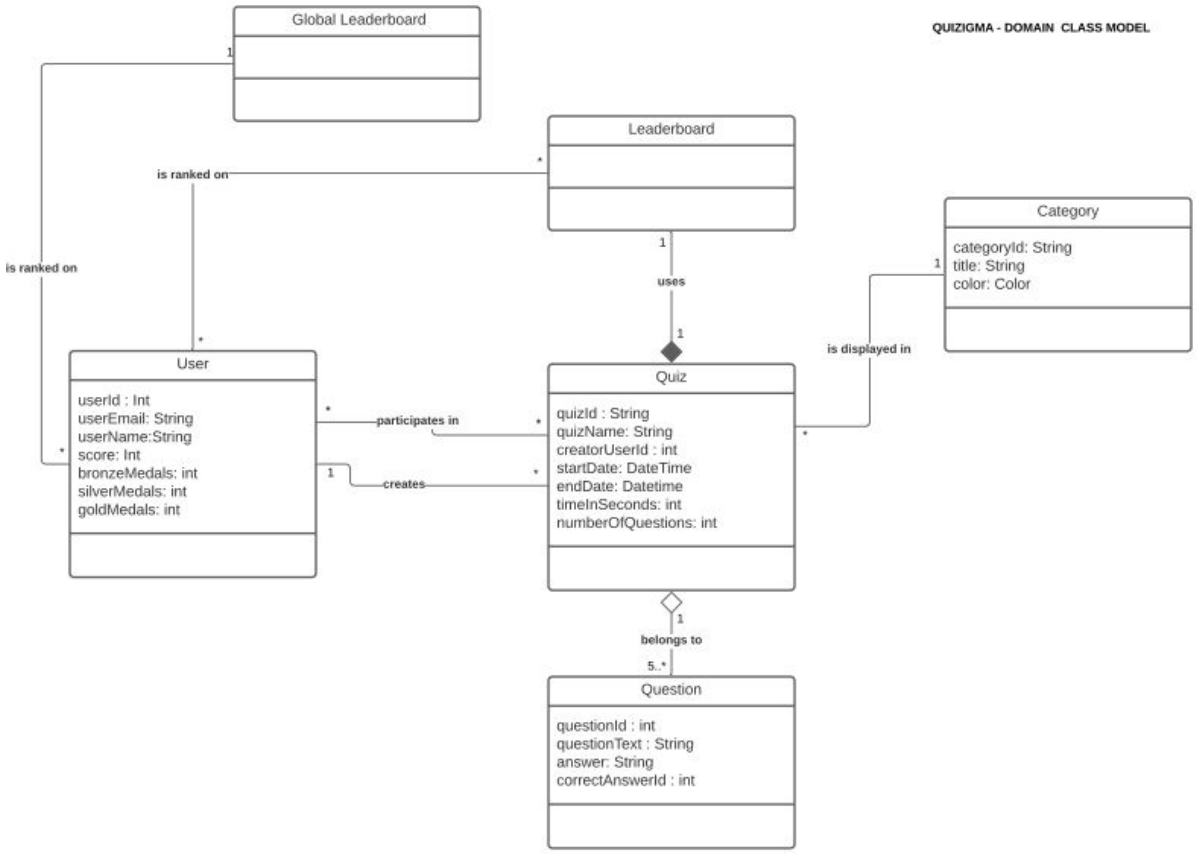
1.5 - Other Product Requirements

Requirement	Priority	Planned Release
The app must run on an Android and iOS phone.	Required	November/2020
The app must run on a Web-Browser.	Required	December/2020
The database server must be running.	Required	January/2021

2.0 - Selected Technologies

- Flutter.
- Dart.
- Firebase Authentication Service.
- Cloud Firestore.
- Firebase Hosting.
- MVC_pattern Flutter package.
- Visual Studio Code.
- Android Studio.

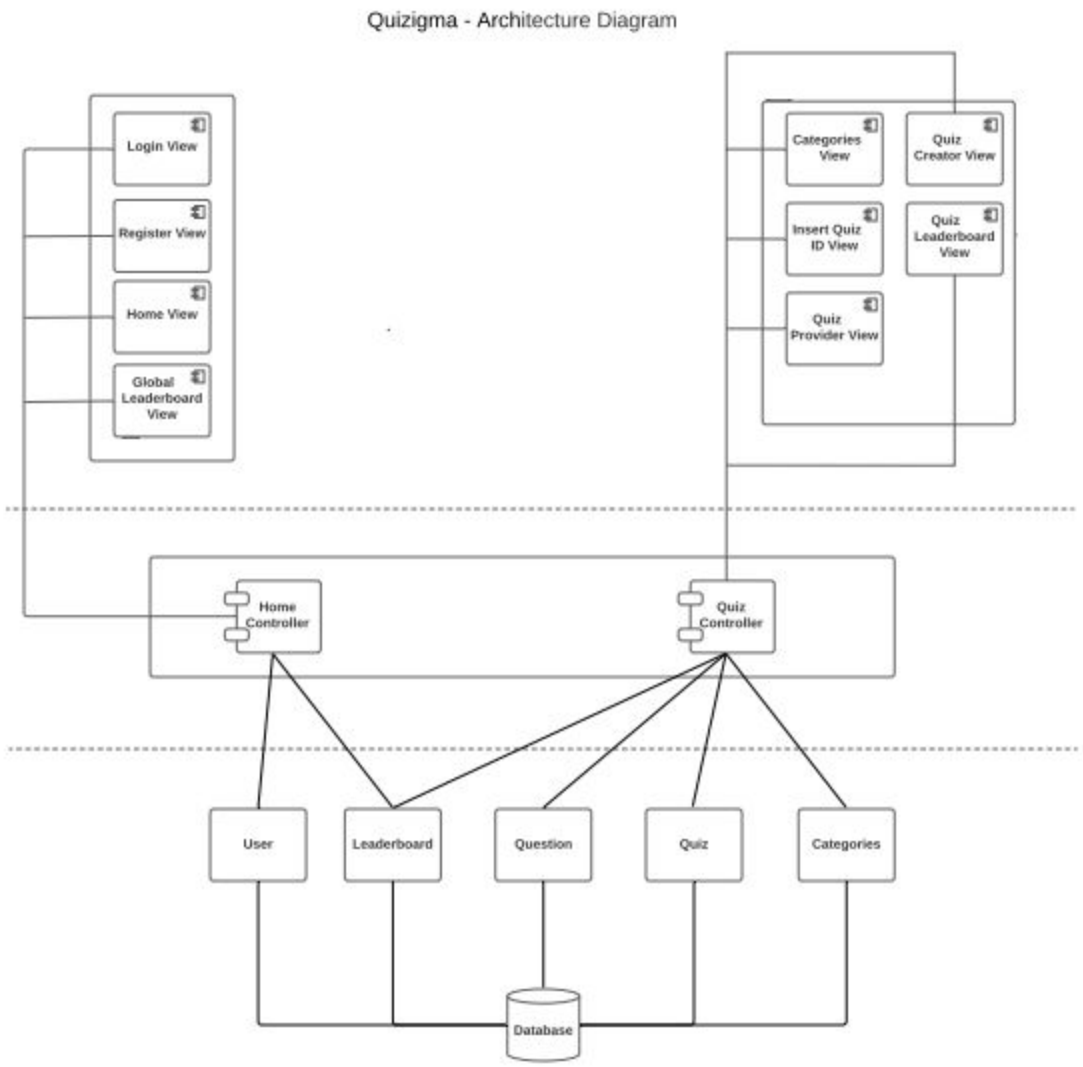
3.0 - Domain Class Diagram



The main class is the "Quiz" class, which interacts with the "User". A "User" can create or participate in a Quiz game. Each Quiz has its own "Leaderboard" which the user is ranked on. There is also a second "Global Leaderboard" which interacts with the "User" and ranks the overall results of all the quizzes that the uer participated (for example, in how many quizzes the user is the number 1 player). The "Medal" attributes inside the "User" is an achievement for how good the performance of the user was. There are 3 options: Bronze Medals (third place), Silver Medals (second place) and Gold Medals (first place).

The attributes "startDate" and "endDate" control the time limit a Quiz game should be available, for users who want to make a game available for only a limited period of time. Each Quiz belongs to a specific "Category". The class "Question" belongs to each individual quiz and stores the text of the question to be displayed, the available answer options and which one is the correct answer chosen by the creator of the quiz.

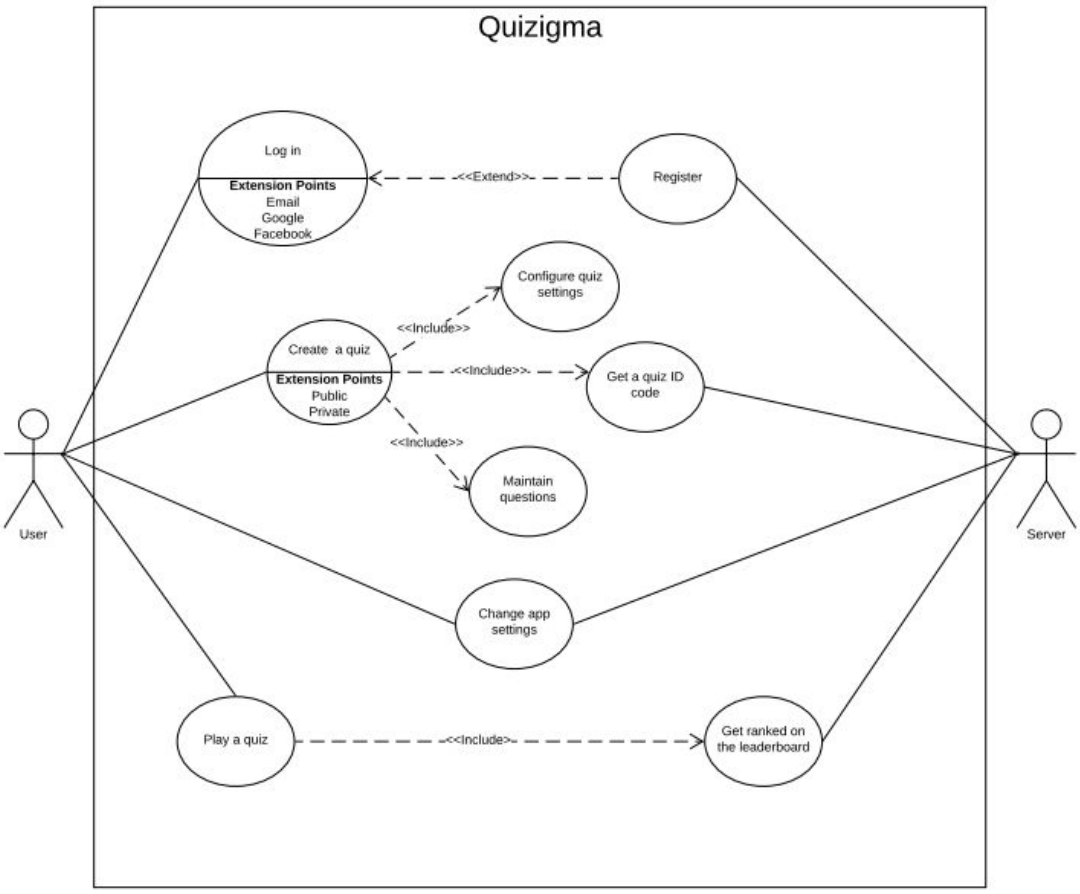
4.0 - Architecture Diagram



In the architecture design a "MVC_pattern" approach is used where the Controllers are responsible for the business logic of the program. Each Controller interacts with a set of "Views". Each "View" is responsible for a specific UI function. All UIs related to home screens like login, register account and main screen will be handled by the "Home Controller". The "Quiz Controller" handles the views responsible for all quiz game related UIs such as categories screen, leaderboard, quiz creator and playing a quiz through its ID.

Last but not least there is the Model of the database. All information regarding the users and quizzes are stored in database Models (representing Question, Leaderboard, User, Quiz and Categories). Each Model is responsible for one specific entity that relates with a Controller. In this way, the Controller gets the information from the database, applies the logic and delivers it back to the View so it can be displayed to the user.

5.0 - Use Case Diagram



6.0 - Non-Functional Requirements

Supported device types:

- Mobile phones (smartphones).
- Web-Browser.
- The application should work in both Portrait and Landscape modes.

Supported platforms and OS versions:

iOS - At least iPhone 6S.

Android - At least Android 6.0 (Marshmallow).

Browsers - Google Chrome, Mozilla Firefox, Internet Explorer, Microsoft Edge, Safari and Opera.

Types of testing:

- Functional testing.

Testing will be supported with functional requirements, test plan, acceptance criteria. The end product should fully pass the acceptance criteria user stories.

Localization:

At the beginning only English language will be supported. Future releases will support multiple languages.

Security

Password requirements:

- The minimum length of the password will be 6 characters.

Usability

Make sure the system is easy and understandable enough to motivate users to use the mobile application.

Availability

Application data is available only online.

Performance

- The system shall not have any visible lags, hangs or freezers
- All buttons and actions are responsive
- While the device is getting information from the server, updating/loading screens' information or data, a spinner or any other visual sign should be displayed so the user understands that a certain action is being performed

Reliability and Productivity

Users shall not lose any data even in negative cases such as interrupted app performance (incoming calls, notifications, app crashes, switching between apps).

7.0 - Source code

The original code can be found under the following link:
https://github.com/IvayloGatev/Quizigma

8.0 - References

- The Net Ninja, Flutter & Firebase Tutorial - https://www.youtube.com/watch?v=sfA3NWDBPZ4&list=PL4cUxeGkcC9j--TKIdkb3ISfRbJeJYQwC&ab_channel=TheNetNinja

- Flutter documentation - https://flutter.dev/docs