A Project Report on “OUR-KAHOOT”

Final Report
Course: CSS 216 Mobile Programming
Team: “Try Young”
Suleyman Demirel University,
Faculty of Engineering and Natural Sciences,
Computer Science

Presented to:
Bissenbay Dauletbayev

Presented by:
Nurzhan Momynkul
Nurdaulet Seitkulov

Table of contents

CERTIFICATE

This is to certify that this Project Report entitled **“OUR-KAHOOT”** which is submitted by A**bdulla Atentayev (190117006), Aisultan Abdykerov (190107035), Nurdaulet Seitkulov (190107097), Nurzhan Momynkul (190107062), Sergey Grichik (190103069)** in the partial fulfillment, for the award of course **“CSS216 Mobile Programming”** of Bachelor of Technology in Department of Computer Science & Engineering, of **UNIVERSITY OF SULEYMAN DEMIREL**. The question embodied in this project work has not previously been submitted for the award of any degree or course at any university/institute, to our knowledge and belief.

Date: 27/12/2021

ACKNOWLEDGEMENT

We are very pleased to present a report on the project work done during the CSS216 “Mobile Programming” course. We are especially grateful to Mr. Bissenbay Dauletbayev for his constant support and guidance throughout our work. Mr. Bissenbay Dauletbayev provided us with this opportunity to work on this mobile project. I appreciate the guidance provided and once again thanks to you. Last but not least, we thank our friends for their contribution to the completion of the project.

Chapter 1

INTRODUCTION

1.1 Overview

In today's world, smartphones have changed our lives and have become an indispensable part of our lives because of their ability to simplify our routine work and thereby save our time. An Android smartphone offers users great functionality and a special experience. Android is a Linux-based operating system that was purchased by Google in 2007. There are many applications, and one of the main reasons there are so many is that Android is open source. On the other hand, Android-based devices such as cell phones, tablets are very user-friendly. Besides Android, there is another OS called IOS, which is used by more than 900 million people. IOS is a mobile operating system for devices manufactured by Apple. iOS runs on the iPhone, iPad, iPod Touch. IOS is best known as the basic software that allows iPhone users to interact with their phones using gestures such as swiping, tapping and pinching.

There is an open-source UI software development package called Flutter that we can use to create mobile apps for both iOS and Android operating systems. Flutter is Google's free open-source UI environment for creating your own mobile apps. Released in 2017, Flutter allows developers to create mobile apps for iOS and Android using a single codebase and programming language. This feature makes it easier and faster to create iOS and Android apps.

In this context, the Project application is developed on the Android and IOS platforms using the Dart programming language. The name of the application is defined as **"OUR-KAHOOT"**. It is an online quiz in which a user with administrative rights can perform administrative tasks such as adding, deleting and editing from the application user interface (UI) and the interviewee can participate in MCQ quizzes with a time limit. Thus, the goals of this project are to make it easier for users to set up quizzes and conduct quizzes using this smartphone-based application. Another goal of this project is to create a user-friendly environment.

1.2 Objectives

The main goal of **“OUR-KAHOOT”** is to create a user-friendly environment for all users and reduce manual effort. In the past, the quiz was done manually, but with the further development of technology, we can generate scores and ask questions automatically. Functional requirements include creating users to take the quiz, automatic generation of kahoots and reports, and administrative tasks such as adding, deleting, updating for users with administrator privileges. In this application, all authority belongs to the administrator (quiz creator), i.e. specifying quiz details with verification, adding questions and answers, and generating a report with scores for each quiz.

1.3 Motivation

Nowadays, most exams, such as organizational recruiting, in-school questions to conclude new topics, and university class tests are paper-based,

which is time-consuming and resource-intensive. The questionnaire is designed, printed, and then data collection, data entry, editing, and cleaning up, which is time consuming and costly. The proposed application is a way to avoid these circumstances that any organization currently faces. Briefly, **"OUR-KAHOOT"** is a game-based learning platform used to test students' knowledge, for formative assessment, or as a break from traditional classroom activities.

Chapter 2

Proposed Model

2.1 Purpose of the project

The main goal of this project is to develop a quiz system called **"OUR-KAHOOT"** using Flutter. The application (OUR-KAHOOT) will provide an online quiz with multiple choice questions (MCQs). This app will support android and ios based operating system. With this app, users or any organization can perform activities such as:

➔ Quiz Creation

➔ Quiz Playing

For better understanding here we made a Flow Chart of the application.

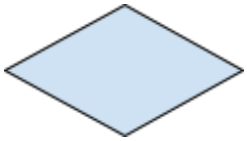
2.2 Flow Chart:

A flowchart is a means of visually representing the flow of data through an information processing system, the operations performed in the system, and the sequence in which they are performed. In this part, we will look at a program flowchart that describes which operations (and in what sequence) are needed to solve a given problem. A program flowchart can be compared to a drawing of a building. As we know, a designer draws a blueprint before starting to build a building. Similarly, a programmer prefers to draw a flowchart before writing a computer program.

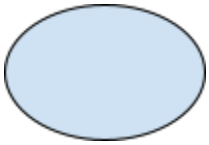
Symbols used to make a flowchart:

Start or end of the program

Input or Output



Decision making and branching



Connector or joining of two parts of program

Flow lines

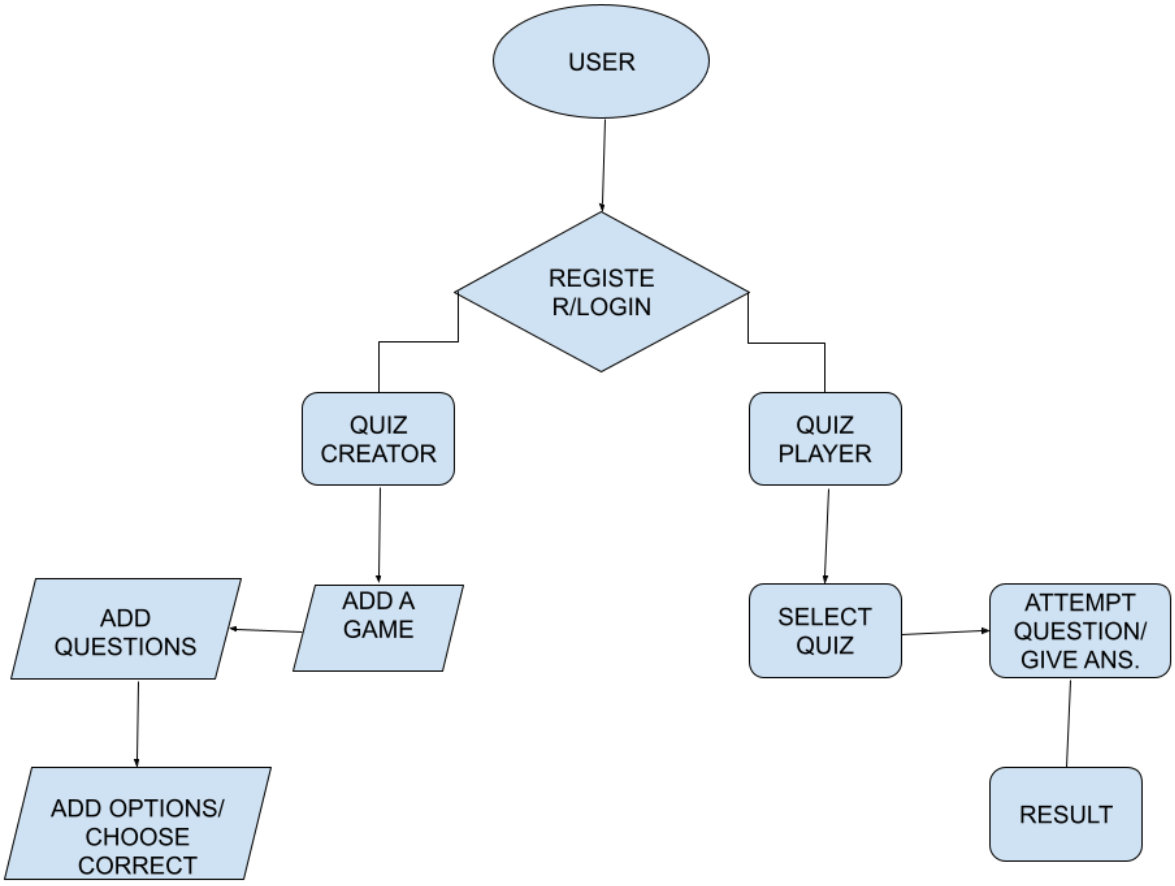


Figure 2.1 Flow Chart for "OUR-KAHOOT"

2.2.1 Description for Flow Chart:

1. First of all, the user goes to the **"Welcome Page"** where he/she can log in or create a new account if he/she is logging in for the first time;
2. When registering, the user undertakes to complete the following forms:
    - ➢ Login
    - ➢ Password
    - ➢ Choosing his/her role (creator or player)
3. If user is a CREATOR, following options will be managed:
    - ➢ Create a quiz (in a ‘ADD GAME’ screen);
    - ➢ Create a question (in a ‘ADD QUESTION’ screen);
    - ➢ Create options with correct answers (in a ‘QUESTION SCREEN’)

    Above options can be added, edited and deleted by creator.
4. If user is a PLAYER, after signing in, all active quizzes list will be shown in listview for giving test.
5. Quiz will execute with certain time limit. Quiz will finish after that time or all questions answered.

Chapter 3

Implementation

3.1 Software and Hardware Requirements

- Computer for coding at least 8GB RAM
- Flutter, PL Dart
- Android SDK
- Local Database
- Android Studio IDE
- Electronic device (smart, mobile phones)

3.1.1 Computer for coding at least 8GB RAM

This computer will be use to code the application. Since Android Studio need heavy RAM usage, to code the mobile application will require the computer to have at least 8gb of RAM to use the SDK and the mobile emulator.

Description	Requirements
Processor	Intel® Core™ i7-4720HQ CPU@ 2.60 GHz 2.59GHz
Graphic processor	Mali-G51 MP4
Operating System	Windows 10
Ram	8.00GB
Hard Drive	1TB HDD
System Type	64-bit Operating System, x64-based processor

Table 3.1: Software requirement

3.1.2 Flutter, PL Dart

Flutter is used to develop mobile applications for Android and iOS. It is a single codebase with the dart language, which is easy to learn and use. It has a Stateful Hot Reload feature that allows you to develop faster and uses native compilers, which means much better performance.

3.1.3 Android SDK

The Android SDK (Software Development Kit) is a set of development tools used to develop applications for the Android platform. The Android SDK includes the following:

- Needed libraries

- Debugger
- Emulator
- Related documentation for Android Application Program Interfaces (API)
- Source Code Example
- Tutorials for Android

Every time Google releases a new version of Android, the corresponding SDK is also released. To be able to write programs with the latest features, developers must download and install the SDK for each phone version The API level is an integer value that uniquely identifies the version of the framework API offered by the Android platform version. The Android platform provides an API that applications can use to interact with the underlying Android system.

3.1.4 Local Database

3.1.5 Android Studio IDE

This software will be used to develop mobile application for the quiz making system. Through using Android Studio editor tool, it can develop a creative user interface for the user.

3.1.6 Mobile Phone

An android/ios smartphone device that has basic specification is needed in this project. User will need to use this device to open and use the mobile application. This device will act as the input of the user when they need to create their kahoot or play.

Description	**Requirements**

Processor	Dual core, 1,4 GHz
Device	Mobile / Tablet
Operating System	Android, IOS
RAM	64 MB or higher
Internal Storage	16 GB

Table 3.2: Hardware Description

3.2 Development Methodology

sen. This prototyping model is a systems development methodology (SDM) in which a paradigm is created, tested, and then refined. This methodology contains five steps: *planning, analysis, design, prototype implementation, and system implementation.*

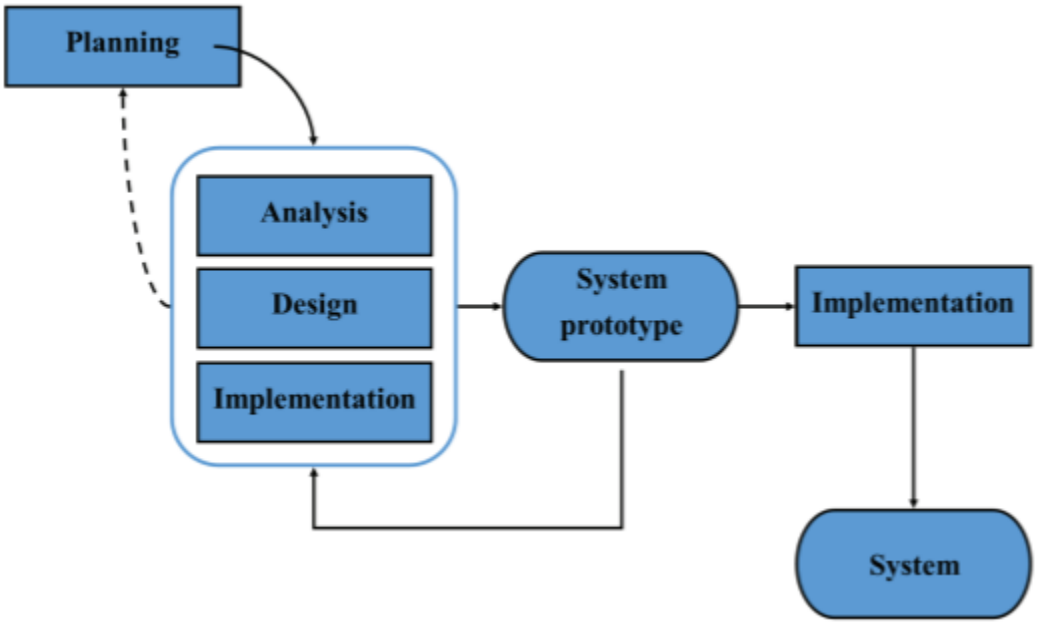


Figure 3.1: Prototyping-based Methodology

Below Table 3.2 Project Calendar and Timeline by 5 step Prototyping-based Methodology

Project Task	Project Week										
	5	6	7	8	9	10	11	12	13	14	15
Planning											
Forming project title through discussion with supervisor											
Researching with existing mobile application											
Analysis											
Identifying project background											
Identifying project problem statement											
Creating system functionality comparison table											
Identifying project objective											
Design											
Identifying the methodologies used in project											
Creating system design using diagrams											
Implementation											
Start coding the system and create a basic prototype											
Improve prototype incrementally											
Test the system											
Deploy the system											
Present and publish on Play Store											

3.3 Implementation

3.3.1 Sign up & Login

A new user creates an account, by navigating to the ‘Sign Up Page’. Here are 4 required options to fill: Login, Password, Password repeating, Choosing role (creator/player). Password validation has configured with the following manner:

```
validator: (value) {
 if (value == null || value.isEmpty) {
   return 'Please enter correct Password';
 }.
```

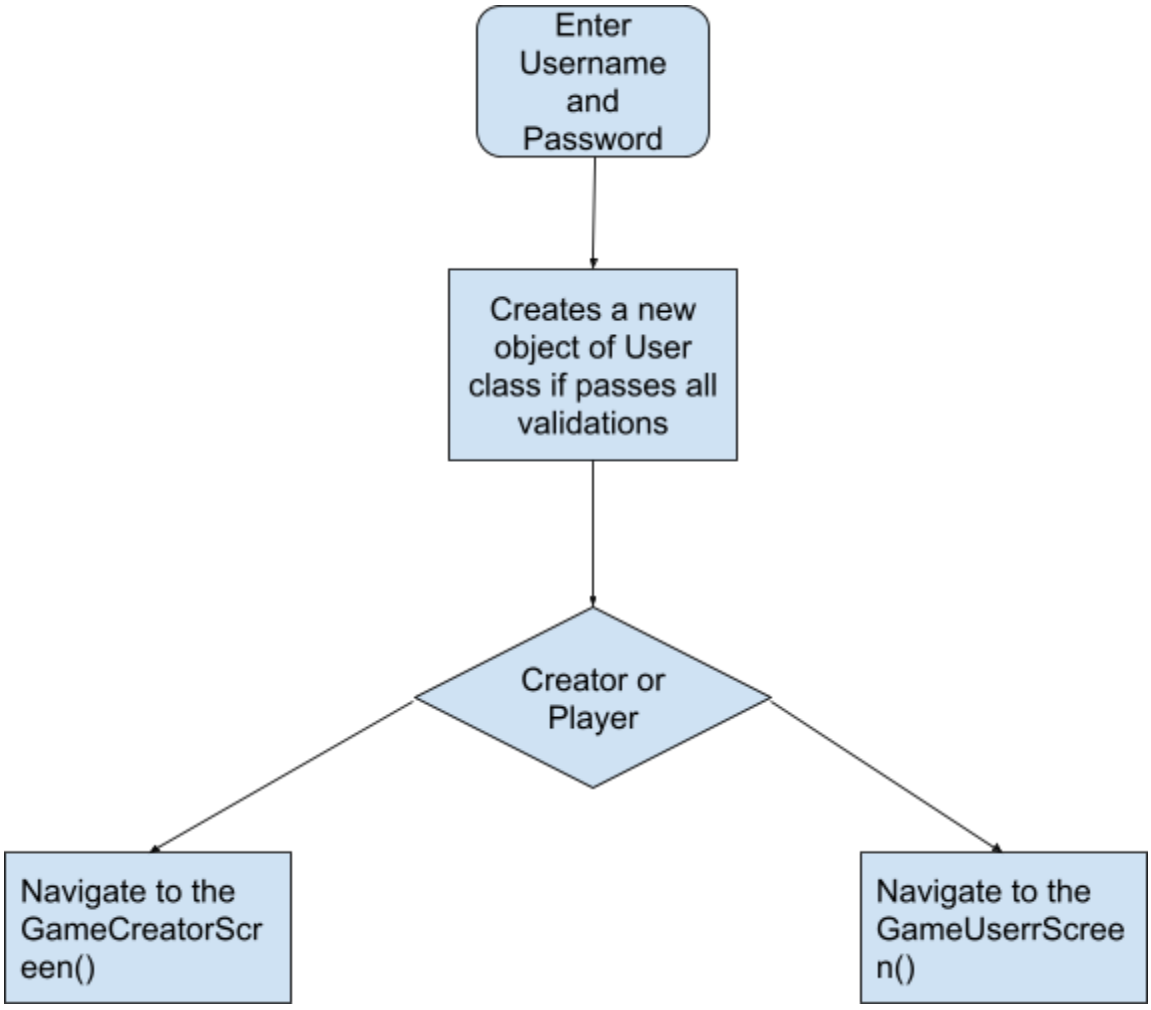
Lastly, new user chooses his role, if he/she is a administrator, then they change the SwitchWidget to the ON.

```
SwitchWidget(
 title: 'Are you creator?',
 value: user.settings.isCreator,
 onChanged: (isCreator) {
   final settings = user.settings.copy(
     isCreator: isCreator,
   );
   setState(() => user = user.copy(settings: settings));
 },
),
```

When user completes registration, presses to the SIGNUP button and it automatically calls checkRegistration method. If all data is correct program adds user to UserController.

After this, user can Sign in to the system by entering login and password. For validation checkLogin() method is used. If user is creator, checkLogin() method routes to GameCreatorScreen() and vice versa.

```
if (newUser.settings.isCreator == true) {
 Navigator.of(context)
      .push(MaterialPageRoute(builder: (_) => const
GameCreatorScreen()));
} else {
 Navigator.of(context)
      .push(MaterialPageRoute(builder: (_) => const
GameUserScreen()));
}
```



3.3.2 Adding Game

We implemented void addGame() method to create a new game. It creates an instance of Game object with specific 'ID' and Game 'Name', and adds it to the GameController.

```
final id = Uuid().v4();
Game game = Game(id: id, name: text);
GameController.addGame(game);
```

All games are shown in a ListView and to delete a game user should dismiss a screen from right to left:

```
direction: DismissDirection.endToStart,
onDismissed: (_) {
 GameController.deleteGame(game.id);
 setState(() {});
},
```

If we press the game, it navigates to the QuestionCreatorScreen()

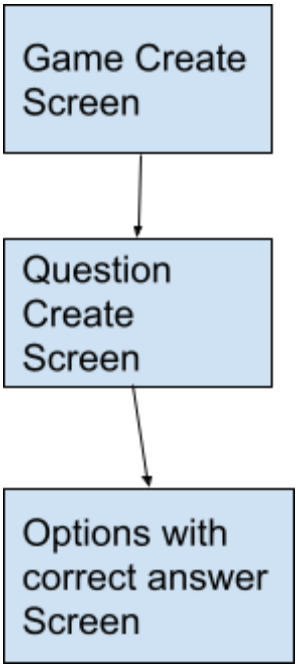
```
Navigator.of(context).push(MaterialPageRoute(
   builder: (_) => const QuestionCreatorScreen()));
```

In this page we implemented addTask() method to create some Tasks:

```
final id = Uuid().v4();
Task task = Task(id: id, questionText: text);
GameController.addTask(task);
```

After this, user adds options, where need to be chosen at least one correct answer. Correct answer is saved by index of option and sets true for correctness (property of Answer class).

```
Widget _buildAnswerTile(Answer answer, index) {
 if (answer.correctness == true) {
   selectedValue = index;
 }
```



3.3.3 GameUserScreen

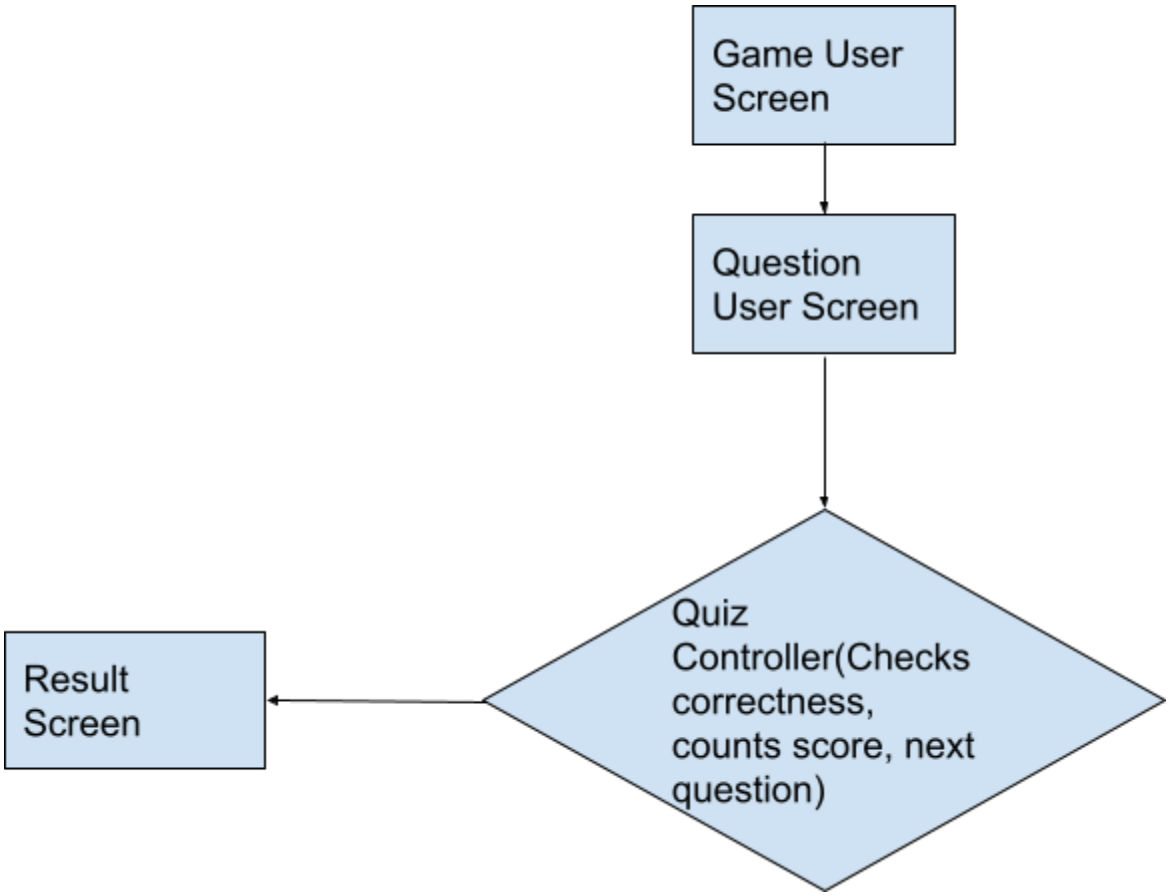
In this page player will see active, available games in ListView. It checks Game's key and by key in this screen appears created Game by Administrator. After starting a Quiz, all process will be controlled by the QuizController class. Here player will be faced with animation where he/she has 250 milliseconds for each question. If no options chosen, it automatically goes to the next question.

```
void nextTask() {
 if (_taskNumber != numOfTasks) {
   _isAnswered = false;
   _pageController.nextPage(
       duration: const Duration(milliseconds: 250), curve:
Curves.ease);
   _animationController.reset();
   _animationController.forward().whenComplete(nextTask);
 } else {
```

```
    Get.to(const ResultScreen());
  }
}
```

Correctness of answer will be checked by the index in QuizController that Administrator already set.

```
void checkAns(Task task, int selectedIndex) {
 if (_isAnswered) return;
 _isAnswered = true;
 _selectedAns = selectedIndex;
```



Chapter 4

User Manual

4.1 System Requirement

Description	Requirements
Processor	Dual core, 1,4 GHz
Device	Mobile / Tablet
Operating System	Android, IOS
RAM	64 MB or higher
Internal Storage	16 GB

Table 4.1 Minimum System Requirement

4.2 Prerequisite

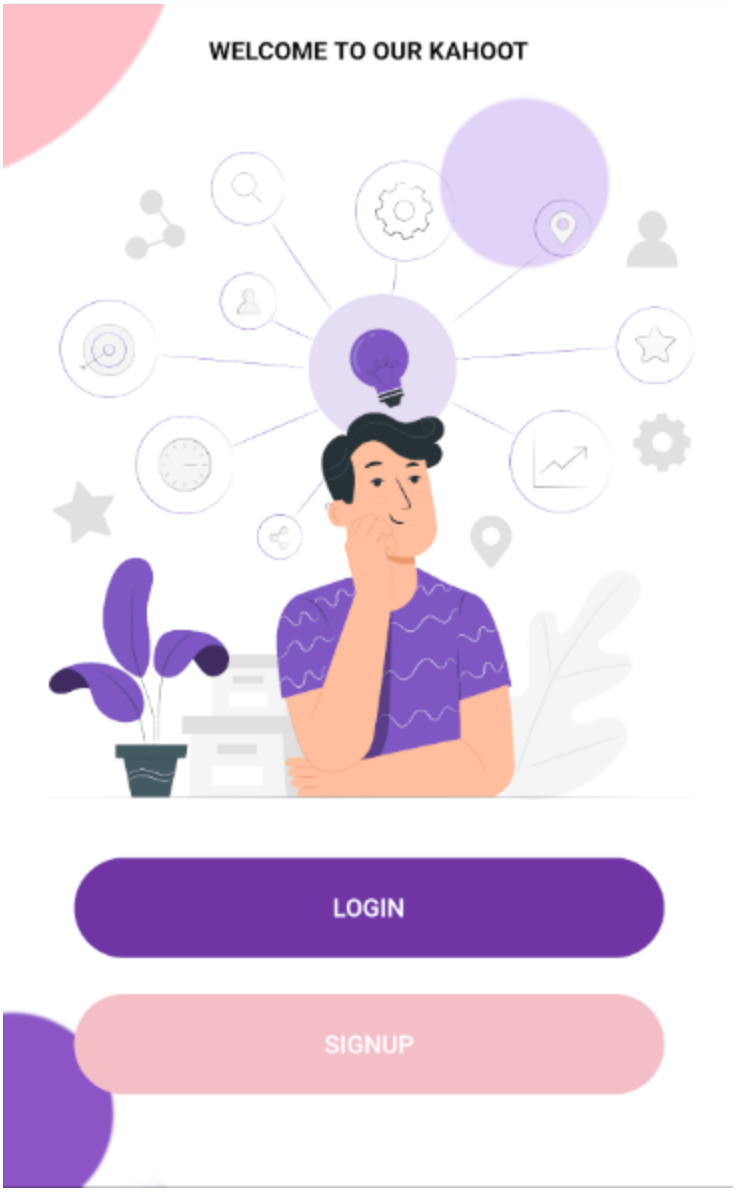
- Internet Connection must active
- Valid Username with password must require to use this application

4.3 Getting Started

Tap the icon to open the application

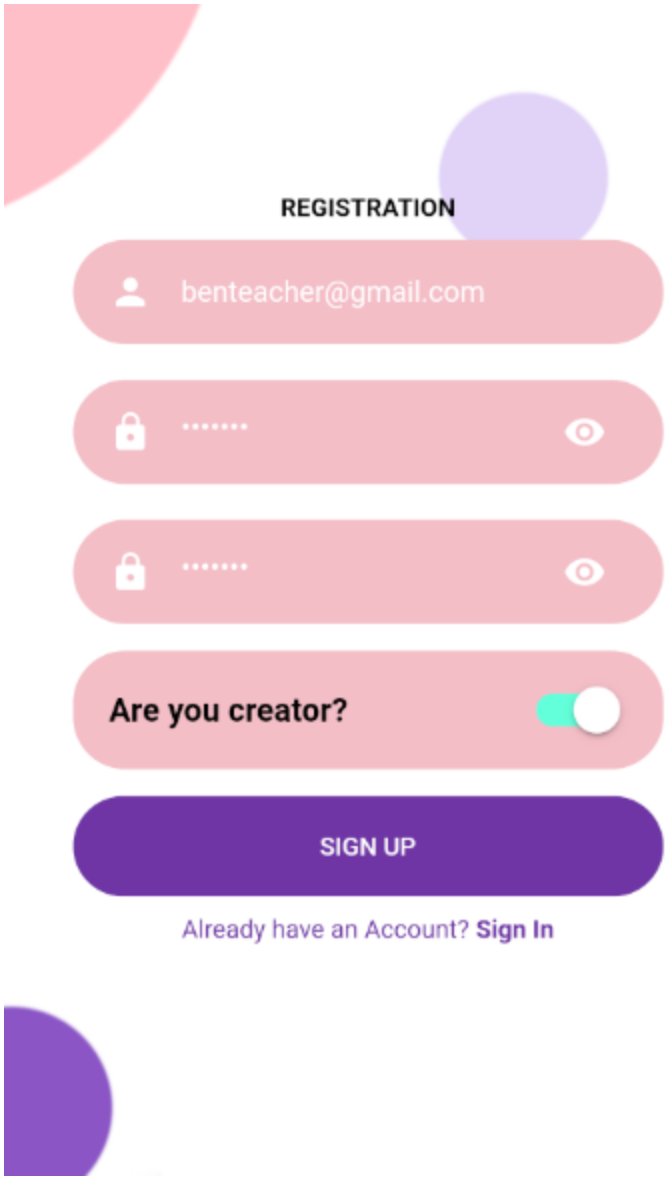
4.3.1 Welcome Page

Welcome screen will appear first, when the user start the app:



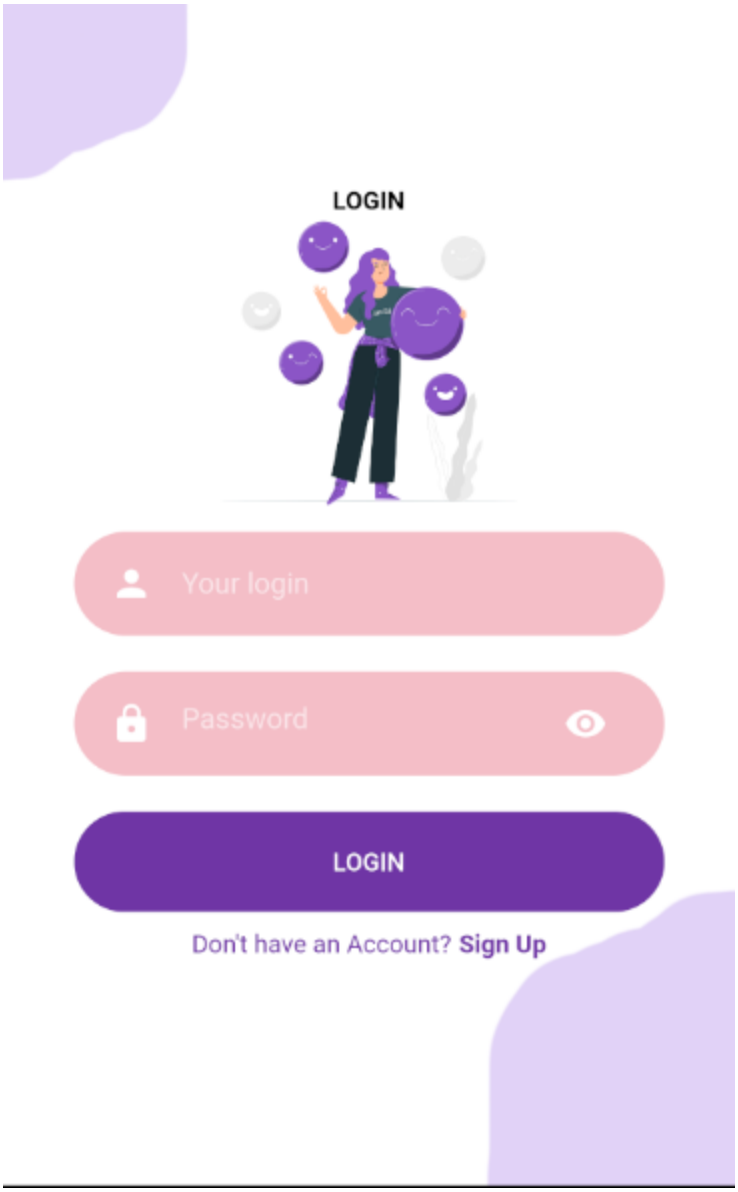
4.3.2 User Registration

Registration screen will appear when we press the button “SIGNUP”, after registration user will be able to choose & take quiz which he / she want.



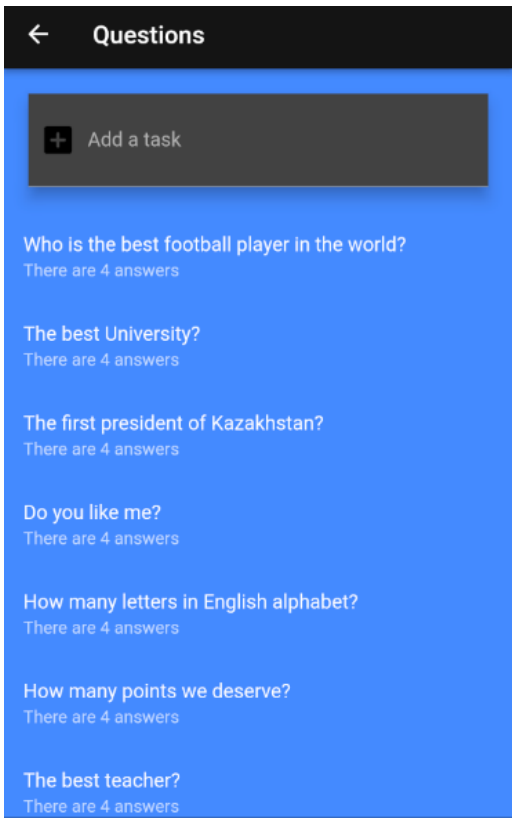
4.3.3 Login

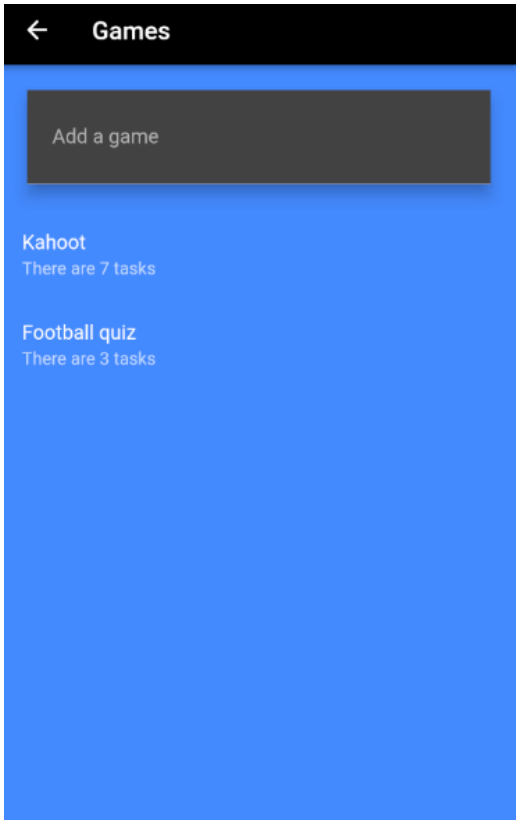
Once the user is successfully registered, the login screen appears. Enter your user name and password in the desired field. The tooltip text tells you where to enter the username and password. After entering your username and password, click the login button.

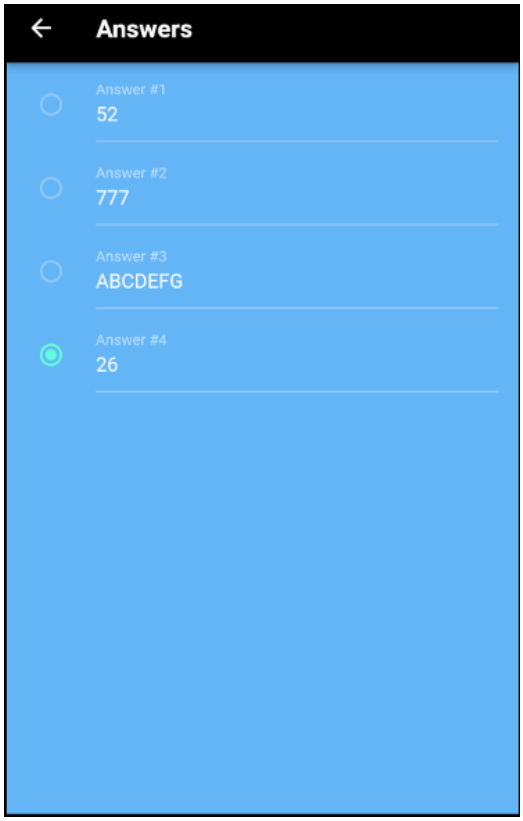


4.3.4 Game creator screen

User (creator) after successful login, will come to the page where you can add a new game, then add some questions and finally, add options with correct answers:

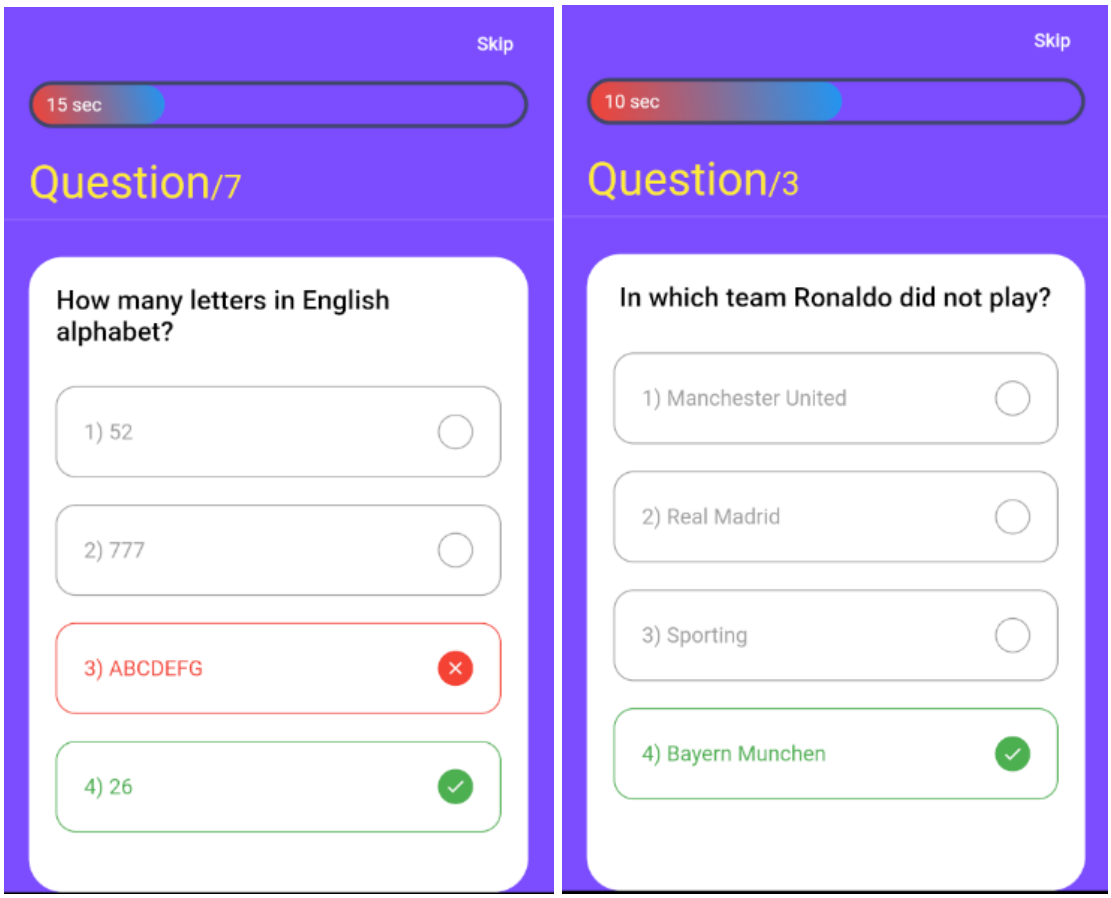

Questions
Add a task
Who is the best football player in the world?
There are 4 answers
The best University?
There are 4 answers
The first president of Kazakhstan?
There are 4 answers
Do you like me?
There are 4 answers
How many letters in English alphabet?
There are 4 answers
How many points we deserve?
There are 4 answers
The best teacher?
There are 4 answers


Games
Add a game
Kahoot
There are 7 tasks
Football quiz
There are 3 tasks


Answers
Answer #1
52
Answer #2
777
Answer #3
ABCDEFG
Answer #4
26

4.3.5 Game Screen

Quiz Player after successful login, will come to the page with where located active quizzes. Player can choose one of them and play it. After choosing a one game, player will go through with all questions, he/she is able to skip the question and has a time limit for each question. When kahoot ends, appears Result Screen, where Player can see how many points he/she got.



Chapter 5

Conclusion

5.1 Project Achievements

From the validation and testing plan, the actual results of the application have achieved good results because all the expected results were achieved and the goals of the project were met. In the end, this project is very useful for educational purposes or your own purposes for testing knowledge in any area. Our app increases engagement, motivation, enjoyment and concentration and improves learning effectiveness and classroom dynamics. In addition, the **"OUR-KAHOOT"** system allows you to create any quizzes in minutes and conduct a live kahoot by streaming your screen to an Apple TV or via video conferencing apps.

5.2 Project Limitations

Although our app allows the user to see the total score at the end of the quiz, the user cannot see the answers to each question after quiz ends. The player is able to see the correct and incorrect answers only during the game, when the player answers or the given time expires.

5.3 Implementation Issues and Challenges

There are several technical problems with this project. First of all, it is necessary to define some development tools before starting to implement the project. It is necessary to consider many platforms of integrated development environments (IDEs), such as Android Studio, Visual Studio require

additional plugins to integrate into the system when developing mobile applications. Between Android Studio and Visual Studio, Android Studio is rated as the best IDE platform among all IDE platforms. In addition, there are two types of mobile operating systems, Android and iOS, which are mostly used in the world. Given the ability of users to use the proposed application in their mobile devices, the flutter operating system was chosen for this project. Since flutter can encode the android and iOS operating system in the same language, it can provide both android and iOS users with the appropriate apk to install. On the other hand, the problems this app will face will be related to the lack of packages to implement in the app. Since flutter is new to the market, time was needed to learn this language when developing this app.

5.4 Future Work

Some improvements can be made in the future. First of all, change the login authentication method to use email and a password. We need to add a Google Sign-in form and register with a phone number. This gives an advantage to the app, because nowadays any app has this feature.
We need to add a method to add a photo or video to the body of the question, that means, quiz creator can create questions and use media in their questions.
We need to add a review mode for the user-player, meaning the player can look at all their question-answers after the quiz ends. This helps the person analyze and work on mistakes.

This is the last option, but optional. We can add a "Dark Theme" to our app. In recent years, a lot of mobile apps have been using this mode to improve UI/UX Design. With such a feature, the user can be more pleasant and there is more probability that the user will evaluate our app in store.

REFERENCES:

- Alessandria, S. and Kayfitz, B., 2021. *Flutter Cookbook*. [S.l.]: Packt Publishing.
- https://www.developer.android.com/
- https://www.geeksforgeeks.org/flutter-tutorial/
- https://www.javatpoint.com/flutter
- https://www.tutorialspoint.com/flutter/index.htm
- https://www.google.com/
- https://www.udemy.com/share/101rfI3@Vj4KeZT6CmrGxdhHJHdLqjPePBkbu2qYhGrJYvxi3eSiY78Wyg_pC0_AjBZStdVd/
- https://www.youtube.com/watch?v=1ukSR1GRtMU&list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ