VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", Belagavi-590018

A Mini Project Report on
"Quiz Application"

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR

THE AWARD OF DEGREE OF

BACHELOR OF ENGINEERING IN
INFORMATION SCIENCE AND ENGINEERING

SUBMITTED BY

YATHIN. B. N (1JB20IS088)
TEJAS. S (1JB20IS077)

Under the Guidance of

Dr. Pavitra Bai. S
Associate Professor
Dept. of ISE, SJBIT
Bengaluru-60

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

SJB INSTITUTE OF TECHNOLOGY

BGS HEALTH AND EDUCATION CITY, KENGERI, BENGALURU-560060, KARNATAKA, INDIA.

2022-2023

|| Jai Sri Gurudev ||

Sri Adichunchanagiri Shikshana Trust ®

SJB INSTITUTE OF TECHNOLOGY

BGS Health & Education City, Kengeri, Bengaluru – 560 060

Department of Information Science & Engineering

CERTIFICATE

Certified that the Mini-project work entitled "**QUIZ APPLICATION",** is bonafide work carried out by **TEJAS. S (1JB20IS077)** and **YATHIN. B. N (1JB20IS088) ,** a bonafide student of **SJB Institute of Technology,** in partial fulfilment for 6th semester in **INFORMATION SCIENCE AND ENGINEERING** of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**, **BELAGAVI** during the academic year **2022-23.** It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Mini Project prescribed for the said degree.

Dr.Pavitra Bai. S **Associate Professor** **Dept. of ISE, SJBIT**	**Dr. Shashidhara H R** **Professor & Head** **Dept. of ISE, SJBIT**

EXTERNAL VIVA

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

ACKNOWLEDGEMENT

I would like to express my profound thanks to His Divine Soul **Padmabhushan Sri Sri Sri Dr. Balagangadharanatha Maha Swamiji** and His Holiness **Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji** for providing me an opportunity to pursue my academics in this esteemed institution.

I would also like to express my profound thanks to **Revered Sri Sri Dr. Prakashnath Swamiji**, Managing Director, SJB Institute of Technology, for his continuous support in providing amenities to carry out this mini project in this admired institution.

I express my gratitude to **Dr. K. V. Mahendra Prashanth**, Principal, SJB Institute of Technology, for providing excellent facilities and academic ambience, which have helped me in satisfactory completion of mini project work.

I extend my sincere thanks to **Dr. Shashidhara H.R**, Head of the Department, Information Science and Engineering, for providing an invaluable support throughout the period of mini project work.

I express my truthful thanks to **Prof. Dr. Pavitra Bai S,** Mini Project Coordination and guidance**,** who has been a source of inspiration to me and has extended his/her support throughout the mini project duration.

Finally, I take this opportunity to extend my earnest gratitude and respect to my parents, Teaching & Non-teaching staffs of the department, the library staff and all my friends, for their continuous support and encouragement.

YATHIN BN
(1JB20IS088)
TEJAS. S
(1JB20IS077)

ABSTRCT

The Android Quiz Application is a mini-project designed to create an interactive and educational platform for users to test their knowledge through quizzes. The application aims to provide a user-friendly interface, allowing users to register, select quizzes, answer questions, track their progress, and compete with others. Developed using Java and XML in the Android Studio IDE, the application employs Firebase for efficient data storage. The project focuses on usability, responsiveness, and visual appeal to enhance the user experience. By offering a wide range of quiz topics and questions, the application promotes continuous learning and self-improvement. It incorporates features such as a scoring mechanism, displaying your score with segregation and displaying the number of correct and wrong answers the user has answered. The application can be further extended with multimedia support, social sharing options, and personalized quiz recommendations. Through the development of this mini-project, users can enjoy an interactive learning experience while challenging themselves and expanding their knowledge in an easily accessible and engaging manner.

.

Table of Contents

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Android Versions

Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet computers. Initially developed by Android, Inc., which Google backed financially and later bought in 2005. Android is open source and Google releases the code under the Apache License. This open source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. Additionally, Android has a large community of developers writing applications ("apps") that extend the functionality of devices, written primarily in a customized version of the Java programming language.

TABLE 1.1 : Android Versions and Specifications

Version	Code Name	Release Date	API Level	Distribution
1.5	Cupcake	April 30,2009	3	0%
1.6	Donut	September 15,2009	4	0.1%
2.0-2.1	Eclair	October 26,2009	7	1.5%
2.2	Froyo	May 20,2010	8	3.2%
2.3.3-2.3.7	Gingerbread	February 9,2010	10	36.4%
2.3-2.3.2	Gingerbread	December 6,2010	9	0.1%
3.2	Honeycomb	May 10,2011	12	0%
3.2	Honeycomb	July 15,2011	13	0.1%
4.0.x	Ice Cream Sandwich	December 16,2011	15	25.6%
4.1.x	Jelly Bean	July 9,2012	16	29.0%
4.2.x	Jelly Bean	November 13,2012	17	4%

1.1.1 Android software development

Android software development is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit.

ADT (Android Development Tools) is the software used to develop android apps. It basically encases Eclipse IDE, which is a multi-language Integrated development environment (IDE) comprising a base workspace and an extensible plug-in system for customizing the environment.. The latest version comes with ADT plugin preinstalled and bundled to the IDE.

This is how the IDE looks like with the important elements marked

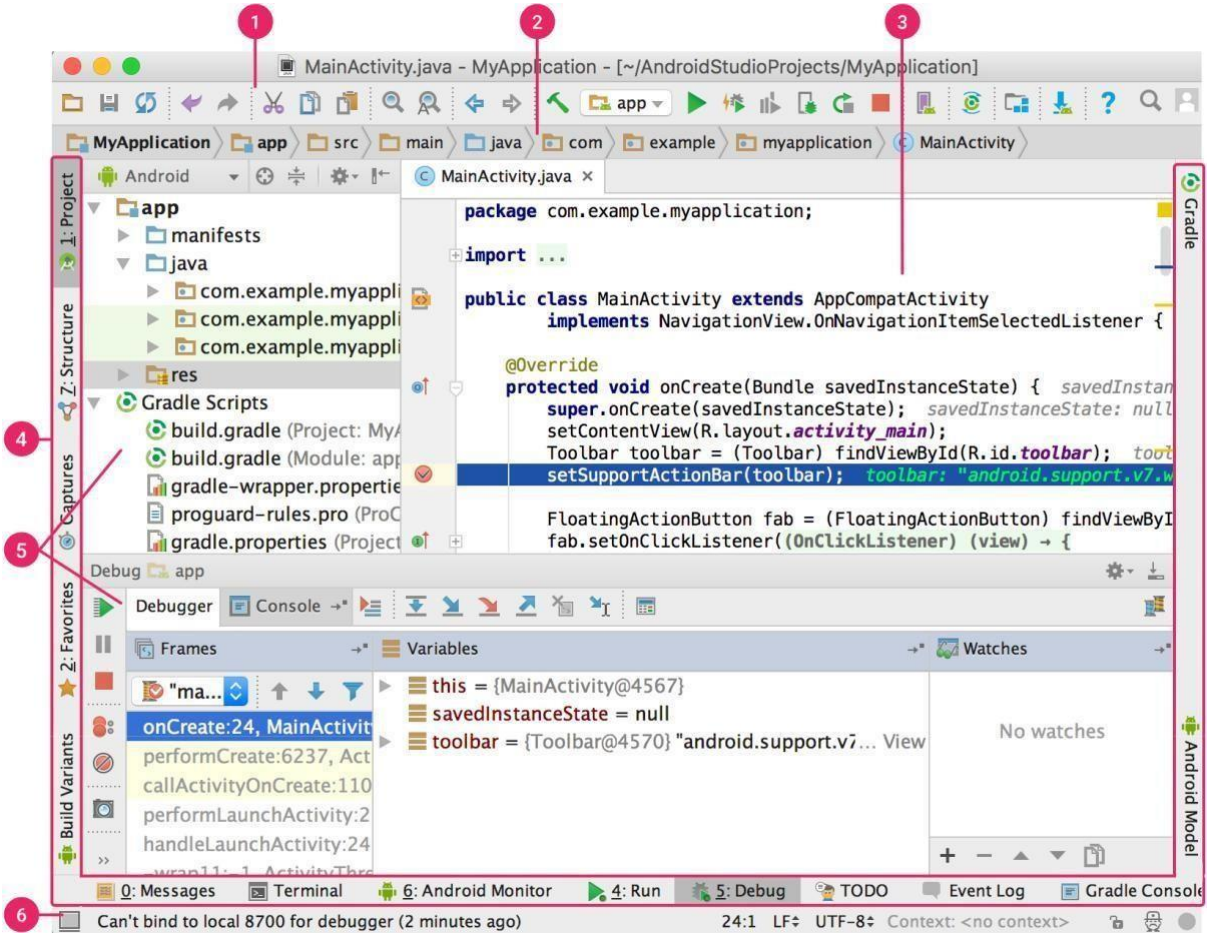

Figure 1.1 Android Studio main window

The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.

The **navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the **Project** window.

The **editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.

The **tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows. The **tool windows** give you access to specific tasks like project management, each, version control, and more. You can expand them and collapse them.

The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

1.1.2 Application programming interface (API)

Application programming interface (API) specifies how some software components should interact with each other. In practice in most of the cases an API is a library that usually includes specification for routines, data structures, object classes, and variables. An API specification can take many forms, including an International Standard such as POSIX, vendor documentation such as the Microsoft Windows API, the libraries of a programming language, e.g., Standard Template Library in C++ or Java API.

Google APIs can be downloaded from Google Code, Google's site for developer tools, APIs and technical resources. The Google Data API] allow programmers to create applications that read and write data from Google services. Currently, these include APIs for Google Apps, Google Analytics, Blogger, Google Base, Google Book Search, Google Calendar, Google Code Search, Google Earth, Google Spreadsheets, Google Notebook, and Picasa Web Albums.

1.1.3 Software Development Kit

SDK (Software Development Kit or "devkit") is typically a set of software development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform. It may be something as simple as an application programming interface (API) in the form of some files to interface to a particular programming language or include sophisticated hardware to communicate with a certain embedded system. Common tools include debugging aids and other utilities often presented in an integrated development environment .

In the latest version of ADT, the android SDK adds on to the IDE automatically as soon as you unzip and load the IDE.

SDK Manager enables us to download Google APIs and use them in our code

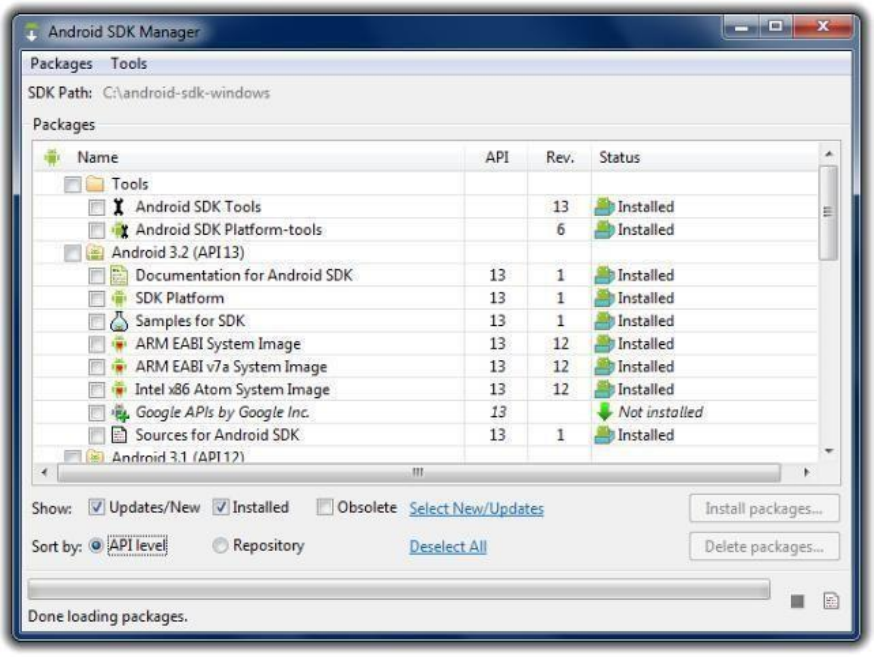

Figure 1.2 SDK Manager

1.1.4 Android Virtual Device (AVD)

Android Virtual Device (AVD) manager enables us to launch virtual android devices/ emulators in our PC and run the app in the emulator, and at the same time we can track and debug each app activity from the Logcat in our IDE.

1.2 ADVANTAGES:

- Improves Efficiency.
- Offers High Scalability.
- Secures the App Data.
- Easy to Maintain.
- Improves Customer Relationship
- Facilitates New Client Data Retrieval.
- Provides Real-time Project Access.

1.3 FEATURES:

- A flexible Gradle-based build system.
- A unified environment where one can develop for all Android devices.
- Apply Changes to push code and resource changes to the running app without restarting the app.
- Extensive testing tools and frameworks.
- Lint tools to catch performance, usability, version compatibility, and other problems C++ and NDK support.
- Built-in support for Google Cloud Platform, making it easy to integrate Google
- Cloud Messaging and App Engine.

Chapter 2

SYSTEM REQUIREMENTS SPECIFICATION

2.1 HARDWRE REQUIREMENTS

- RAM: 8GB recommended, 4 GB minimum.
- HARD DISK: 110 MB of hard disk space required, 40 MB additional hard disk space required for installation (150 MB total).
- MONITOR: 15 VGA Color Monitor
- SYSTEM: Intel Core 2 Duo or Above

2.2 SOFTWARE REQUIREMENTS

2.2.1 Data Requirements

The set of data that is involved in any project is defined using data requirements. For this project, the main data required is the product information to register the application and the item's information. Without this information, the application cannot process the transaction.

2.2.2 Functional Requirements

Functional requirements are properties that must exist in the final system. For any mobile application, we need to download the application from the play store. The application could be either free or paid depending upon the store or merchant. To use the application, the user needs to register and log in to the application after installing by providing login information. Once, he or she logins into the application, they can use all the features.

2.2.3 Performance Requirements

Response time, scalability, platform dependencies, tolerance are the performance requirements that should be considered when developing any system. It should be able to deliver the information about any of those issues to the user when the system is no longer able to provide results when the user wants scalable enough to accept new features when we want to expand the application complexity.

2.2.4 System Requirements

The application should be installed into a device, system or any machine in such a way that it should have basic requirements like supporting software and hardware of the device, accessing in-built software, say camera for mobile device, internet permissions, and potential security issues such as virus or any malware detection.[7]

- DEVELOPMENT PLATFORM : WINDOWS 7
- LANGUAGE TOOL : JAVA
- *SOFTWARE USED: Android Studio SDK

Chapter 3

SYSTEM DESIGN

The purpose of the design phase is to develop a clear understanding of what the developer wants people to gain from his or her project. As the developer works on the project, the test for every design decision should be ‘does this future fulfill the ultimate purpose of the project?’ A purpose statement affects the design process by explaining what the developer wants the project to do, rather than describing the project itself. The design document will verify that the current design meets all of the explicit requirements contained in the system model as well as the implicit requirements described by the customer.

3.1 SEQUENCE DIAGRAMS

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects work together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are also known as event diagrams or event scenarios.

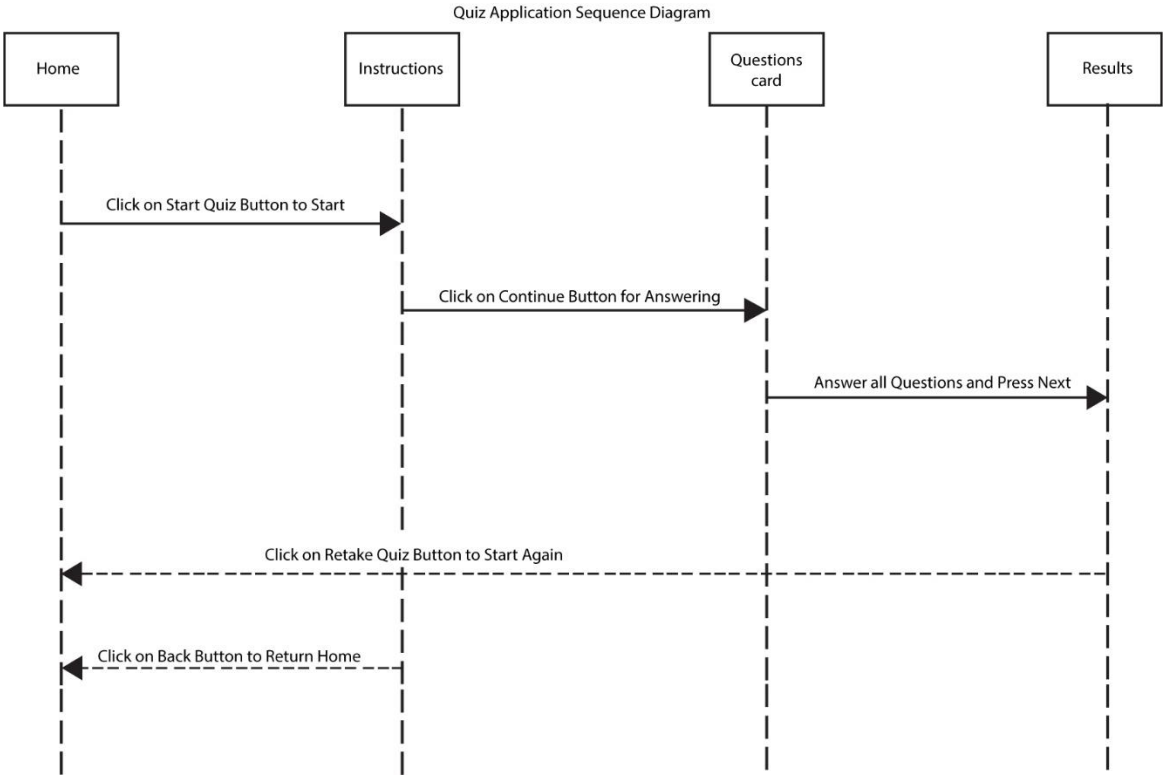

Figure 3.1 Sequence diagram

Live Data: A data holder class that can be observed. Always holds/caches the latest version of data, and notifies its observers when data has changed. LiveData is lifecycle aware. UI components just observe relevant data and don't stop or resume observation. LiveData automatically manages all of this since it's aware of the relevant lifecycle status changes while observing.

View Model: Acts as a communication center between the Repository (data) and the UI. The UI no longer needs to worry about the origin of the data. ViewModel instances survive Activity/Fragment recreation.

Repository: A class that you create that is primarily used to manage multiple data sources.

Entity: Annotated class that describes a database table when working with Room.

Room database: Simplifies database work and serves as an access point to the underlying SQLite database (hides SQLite OpenHere. The Room database uses the DAO to issue queries to the SQLite database.

SQLite database: On device storage. The Room persistence library creates and maintains this database for you.

DAO: Data access object. A mapping of SQL queries to functions. When you use a DAO, you call the methods, and Room takes care of the rest.

Chapter 4

IMPLEMENTATION

4.1 Introduction to Programming Languages, IDE'S, Tools:

4.1.1 Java

As the project is developing an Android Application, the default programming language is Java. All Android applications are built using Java in Android Studio or Eclipse or both. Java is a popular and widely used language throughout the world. As mentioned in, Java is one of the powerful programming languages like C, C++. developed by Sun Microsystems which has many powerful features as described below. The language is also easy to learn, understand and implement. Java is used in various kinds of applications like Web, Desktop, Mobile, and Big Data. Many powerful features are supported by Java including various libraries, application services, graphics library for 2D/3D applications. The language is flexible enough to maintain code complexity, test, implementation, integration and support. Apart from these, there are other key features which make Java more special. It is object oriented programming language, one of the important hierarchies in the programming languages which is used to implement real time applications, it provides for code reusability, it has a platform independence feature including any virtual machines(Write Once Read Everywhere), as in no need to write the 20 code for different OS as the Java Compliers convert the java source files to bytecode and this could be interpreted by any machine and the actual code is compiled irrespective of any machine, OS. It is more secured as the compilers are designed efficiently to figure out any kind of errors.

4.1.2 IDE's, Tools and Technologies:

4.1.2.1 Android Studio

Android Studio is exclusively designed for developing Android applications.Say, suppose we declared few variables or methods that starts with an 'S', it automatically senses everything that starts with an 'S' and makes suggestions. It also supports Git as a version control system to maintain the app changes and push them into github. All java files, layout files (for design) are integrated into a single project easily. After the completion of project, the whole application could be put as an .APK (Android Package) file, in which we can run that APK file in any device and use the application. Other main tools include Android SDK, ADB, and Gradle Build.

4.1.3 Android Software Development Kit (SDK):

One of the main tools used in developing android applications, as it packages many core features into one SDK and it can be used in the application easily. This helps us to avoid writing lot of code, and building applications faster.

4.1.4 Android Debug Bridge (ADB):

Android SDK uses ADB tool as a connection device which allows us to connect the Android Devices or Emulator with the machine via USB. After developing or while developing applications, we can connect with the device to check how the application runs. Later, we can debug and run the applications.

4.1.5 Gradle Build:

Gradle Scripts are the recent feature that is added to Android Studio. It is basically an automated build system which is used to automate the various phases involved in designing an application that includes design, development, test, debug, and publish. We need to configure the project and modules by mentioning all the supported jar files, SDK's, version name, level, compiled SDK version, build tools version. to ensure that the developed app is compatible with the testing device/emulator. Gradle is also similar to Ant and Maven which helps in maintaining java projects (repositories).

4.1.6 Android Device Monitor:

If we want to access all the hidden files that are generated when we run the application, we can use the monitor. We can select any project and explore the files that are related to that project. But, as they are hidden files, we need root permissions to access them. Suppose, if we run the app in device, we need to root the device and run commands in adb shell to get permissions.

4.1.7 SDK Manager:

It is one of the main tools to maintain the updates of all the installed components required to run the project. It also notifies us when the project is not compatible with device or any other compatibility issues and to download any component that is required.

4.1.7 AVD Manager:

It is used to create virtual devices of any desired API level to support higher level SDK's incase our device does not support. Using emulators to test the application is difficult as it might be little slower when compared to real device.

4.1.8 Create an Entity:

Room allows you to create tables via an Entity. Let's do this now.
Create a new Kotlin class file called Word containing the Word data class. This class will describe the Entity (which represents the SQLite table) for your words. Each property in the class represents a column in the table. Room will ultimately use these properties to both create the table and instantiate objects from rows in the database.

4.2 XML AND .java CODES OF THE PROJECT

4.2.1 Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/primary"
    tools:context=".MainActivity">

    <RelativeLayout
        android:id="@+id/topBar"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_marginStart="30dp"
        android:layout_marginEnd="30dp">

        <LinearLayout
```

```
android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_alignParentBottom="true">

        <TextView
            android:id="@+id/currentQuestionTV"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Question 1"
            android:textColor="#E6FFFFFF"
            android:textStyle="bold"
            android:textSize="25sp"/>

        <TextView
            android:id="@+id/totalQuestionTV"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="/10"
            android:textColor="#80FFFFFF"
            android:textSize="20sp" />
    </LinearLayout>

    <ImageView
        android:layout_toStartOf="@+id/quizTimer"
        android:layout_width="18dp"
        android:layout_height="18dp"
        android:src="@drawable/stopwatch"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="3dp"/>

    <TextView
        android:id="@+id/quizTimer"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="00:00:00"
        android:textSize="16sp"
        android:textColor="@color/secondary"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="5dp" />

</RelativeLayout>
<View
    android:id="@+id/divider"
    android:layout_width="match_parent"
    android:layout_height="0.8dp"
    android:layout_below="@+id/topBar"
    android:background="#66FFFFFF"
    android:layout_marginStart="30dp"
    android:layout_marginEnd="30dp"/>

<LinearLayout
    android:layout_below="@id/divider"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="30dp"
    android:layout_marginEnd="30dp"
    android:layout_marginTop="60dp"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#FFFFFF"
        android:textSize="28sp"
```

```
    android:textStyle="bold"
     android:id="@+id/questionTV"/>

<!---Option 1-->
<RelativeLayout
    android:id="@+id/option1Layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="60dp"
    android:background="@drawable/round_back_white50_10">

    <TextView
        android:id="@+id/option1TV"
        android:layout_toStartOf="@+id/option1Icon"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="15sp"
        android:text="Option 1"
        android:textColor="#FFFFFF" />
    <ImageView
        android:id="@+id/option1Icon"
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:src="@drawable/round_back_white50_100"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:layout_marginEnd="20dp"/>
</RelativeLayout>

<!---Option 2-->
<RelativeLayout
    android:id="@+id/option2Layout"
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:background="@drawable/round_back_white50_10">

    <TextView
        android:id="@+id/option2TV"
        android:layout_toStartOf="@+id/option2Icon"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="15sp"
        android:text="Option 2"
        android:textColor="#FFFFFF" />
    <ImageView
        android:id="@+id/option2Icon"
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:src="@drawable/round_back_white50_100"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:layout_marginEnd="20dp"/>
</RelativeLayout>

<!---Option 3-->
<RelativeLayout
    android:id="@+id/option3Layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:background="@drawable/round_back_white50_10">

    <TextView
        android:id="@+id/option3TV"
```

```
android:layout_toStartOf="@+id/option3Icon"

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="15sp"
        android:text="Option 3"
        android:textColor="#FFFFFF" />
    <ImageView
        android:id="@+id/option3Icon"
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:src="@drawable/round_back_white50_100"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:layout_marginEnd="20dp"/>
</RelativeLayout>

<!---Option 4-->
<RelativeLayout
    android:id="@+id/option4Layout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:background="@drawable/round_back_white50_10">

    <TextView
        android:id="@+id/option4TV"
        android:layout_toStartOf="@+id/option4Icon"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="15sp"
        android:text="Option 4"
        android:textColor="#FFFFFF" />
```

```
<ImageView
        android:id="@+id/option4Icon"
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:src="@drawable/round_back_white50_100"
        android:layout_alignParentEnd="true"
        android:layout_centerVertical="true"
        android:layout_marginEnd="20dp"/>
    </RelativeLayout>

  </LinearLayout>

  <androidx.appcompat.widget.AppCompatButton
    android:id="@+id/nextQuestionBtn"
    android:layout_width="wrap_content"
    android:layout_height="50dp"
    android:paddingStart="50dp"
    android:paddingEnd="50dp"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="30dp"
    android:textAllCaps="false"
    android:textColor="#FFFFFF"
    android:background="@drawable/round_back_secondary_100"
    android:text="Next Question"/>

</RelativeLayout>
```

4.2.2 Activity_start.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:gravity="center"
android:orientation="vertical"
android:background="#101530"
tools:context=".StartActivity">

<ImageView
   android:layout_width="80dp"
   android:layout_height="wrap_content"
   android:adjustViewBounds="true"
   android:src="@drawable/logo" />

<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="QuizFlix App"
   android:textColor="#FFFFFF"
   android:textSize="25sp"
   android:layout_marginTop="50sp"
   android:textStyle="bold"
   />

<TextView
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:text="Play and Fun !!"
```

```
  android:layout_marginTop="5dp"
   android:textColor="#99FFFFFF" />

<androidx.appcompat.widget.AppCompatButton
   android:id="@+id/startQuizBtn"
   android:layout_width="match_parent"
   android:layout_height="50dp"
   android:layout_marginStart="50dp"
   android:layout_marginEnd="50dp"
   android:layout_marginTop="50dp"
   android:text="Start Quiz"
   android:textColor="#FFFFFF"
   android:textAllCaps="false"
   android:background="@drawable/round_back_secondary_100" />

<androidx.appcompat.widget.AppCompatButton
   android:id="@+id/quitBtn"
   android:layout_width="match_parent"
   android:layout_height="50dp"
   android:layout_marginStart="50dp"
   android:layout_marginEnd="50dp"
   android:layout_marginTop="25dp"
   android:text="Go Home"
   android:textColor="#FFFFFF"
   android:textAllCaps="false"
   android:background="@drawable/round_back_sec_str_100" />

 </LinearLayout>
```

4.2.3 Instructions_dialog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
xmlns:app="http://schemas.android.com/apk/res-auto">

<androidx.cardview.widget.CardView
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   app:cardUseCompatPadding="true"
   app:cardCornerRadius="10dp">

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="vertical"
      android:paddingBottom="20dp">

      <TextView
         android:layout_width="match_parent"
         android:layout_height="55dp"
         android:gravity="center"
         android:text="Instructions"
         android:textColor="#FFFFFF"
         android:textSize="20sp"
         android:textStyle="bold"></TextView>

      <TextView
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_marginStart="20dp"
         android:layout_marginEnd="20dp"
```

```
        android:textSize="16sp"
        android:id="@+id/instructionsTV"/>
    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/continueBtn"
        android:layout_width="wrap_content"
        android:layout_height="50dp"
        android:paddingStart="50dp"
        android:paddingEnd="50dp"
        android:textAllCaps="false"
        android:textColor="#FFFFFF"
        android:layout_marginTop="40dp"
        android:layout_gravity="center"
        android:text="Continue"
        android:background="@drawable/round_back_primary_100"/>
  </LinearLayout>
  </androidx.cardview.widget.CardView>

</RelativeLayout>
```

4.2.4 Activity_quiz_results.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
    android:orientation="vertical"
   tools:context=".QuizResult"
   android:background="@color/primary">

<TextView
   android:layout_width="match_parent"
   android:layout_height="68dp"
   android:gravity="center"
   android:textColor="#4CAF50"
   android:textSize="25sp"
   android:textStyle="bold"
   android:text="Quiz Result" />

<LinearLayout
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_marginBottom="60dp"
   android:gravity="center">

   <ImageView
      android:layout_width="188dp"
      android:layout_height="wrap_content"
      android:adjustViewBounds="true"
      android:src="@drawable/winner" />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Congratulations"
      android:letterSpacing=".1"
      android:textStyle="bold"
      android:textSize="30sp"
      android:textColor="#FFC107"
      android:layout_marginTop="10dp"/>

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:gravity="center"
      android:layout_marginTop="5dp"
      android:layout_marginStart="50dp"
      android:layout_marginEnd="50dp"
      android:letterSpacing=".1"
      android:text="You have completed the Quiz successfully"
      android:textSize="16sp"
```

```
 android:textColor="#CCFFFFFF"/>

<TextView
    android:layout_marginTop="30dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:letterSpacing="0.2"
    android:textColor="#80FFFFFF"
    android:text="YOUR SCORE"/>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginTop="5dp"
    android:gravity="center">

    <TextView
        android:id="@+id/scoreTV"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:letterSpacing="0.2"
        android:text="0"
        android:textColor="@color/color3"
        android:textSize="30sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/totalScoreTV"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:letterSpacing="0.2"
        android:text="/10"
        android:textColor="@color/white"
        android:textSize="30sp"
        android:textStyle="bold" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="50dp"
    android:layout_marginEnd="50dp"
    android:layout_marginTop="30dp"
    android:weightSum="2"
    android:orientation="horizontal">

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:orientation="vertical">
```

```
<TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:letterSpacing="0.2"

    android:textColor="#80FFFFFF"
            android:text="CORRECT"/>

        <TextView
            android:id="@+id/correctTV"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:text="0"
            android:textSize="20sp"
            android:textStyle="bold"
            android:textColor="@android:color/holo_green_light" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:letterSpacing="0.2"
            android:textColor="#80FFFFFF"
            android:text="INCORRECT"/>

        <TextView
            android:id="@+id/incorrectTV"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:textSize="20sp"
            android:textStyle="bold"
            android:text="0"
            android:textColor="@android:color/holo_red_light" />
    </LinearLayout>

  </LinearLayout>

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:weightSum="2"
    android:layout_marginEnd="30dp"
    android:layout_marginTop="50dp"
    android:layout_marginStart="30dp">
```

```
        <androidx.appcompat.widget.AppCompatButton

            android:id="@+id/shareBtn"
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_marginEnd="15dp"


            android:layout_weight="1"
            android:background="@drawable/round_back_white_10"
            android:textColor="@color/black"
            android:text="Share Result"
            android:textAllCaps="false" />

        <androidx.appcompat.widget.AppCompatButton
            android:id="@+id/reTakeQuizBtn"
            android:layout_width="0dp"
            android:layout_height="50dp"
            android:layout_marginStart="15dp"
            android:layout_weight="1"
            android:background="@drawable/round_back_secondary_10"
            android:text="Re-Take Quiz"
            android:textColor="#FFFFFF"
            android:textAllCaps="false" />
      </LinearLayout>

    </LinearLayout>

</LinearLayout>
```

4.2.5 StartActivity.java

```
package com.example.quizflix;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class StartActivity extends AppCompatActivity {

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_start);

final AppCompatButton startQuizBtn = findViewById(R.id.startQuizBtn);
final AppCompatButton quitBtn = findViewById(R.id.quitBtn);

startQuizBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
//start quiz application MainActivity
startActivity(new Intent(StartActivity.this,MainActivity.class));
}
});

quitBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
```

```
  //exit application - finish
          finish();
           }
       });
   }
 }
```

4.2.6 InstructionDialog.java

```
  package com.example.quizflix;

  import android.app.Dialog;
  import android.content.Context;
  import android.os.Bundle;
  import android.view.View;
  import android.widget.TextView;
  import androidx.annotation.NonNull;
  import androidx.appcompat.widget.AppCompatButton;

 public class InstructionsDialog extends Dialog{

private int instructionPoints = 0;
public InstructionsDialog(@NonNull Context context) {
    super(context);
}

@Override
protected void onCreate(Bundle savedInstanceState){
    super.onCreate(savedInstanceState);
    setContentView(R.layout.instructions_dailog_layout);

    final AppCompatButton continueBtn = findViewById(R.id.continueBtn);
```

```
final TextView instructionsTV = findViewById(R.id.instructionsTV);
setInstructionPoint(instructionsTV,"1. You will get maximum 3 minutes to complete
the quiz \n \n 2. You will get 1 point on every correct answer");
//setInstructionPoint(instructionsTV,"2. You will get 1 point on every correct answer");

continueBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
dismiss();
      }
  });
}

private void setInstructionPoint(TextView instructionsTV,String instructionPoint) {

if(instructionPoints == 0){
instructionsTV.setText(instructionPoint);
}
else {
instructionsTV.setText(instructionsTV.getText()+"\n\n"+instructionPoint);
     }
   }
}
```

4.2.7 QuestionList.java

```
package com.example.quizflix;

import java.io.Serializable;

public class QuestionsList implements Serializable {
private final String question,option1,option2,option3,option4;
private final int answer;
private int userSelectedAnswer;

public QuestionsList(String question, String option1, String option2, String option3, String
option4, int answer) {
this.question = question;
this.option1 = option1;
this.option2 = option2;
this.option3 = option3;
this.option4 = option4;
this.answer = answer;
this.userSelectedAnswer = 0;
}

public String getQuestion() {
return question;
}

public String getOption1() {
return option1;
}

public String getOption2() {
return option2;
}

public String getOption3() {
return option3;
}

public String getOption4() {
return option4;
}

public int getAnswer() {
return answer;
}
```

```
    public int getUserSelectedAnswer() {
    return userSelectedAnswer;
    }

    public void setUserSelectedAnswer(int userSelectedAnswer) {
  this.userSelectedAnswer = userSelectedAnswer;
  }
}
```

4.2.8 MainActivity.java

```
    package com.example.quizflix;

    import androidx.annotation.NonNull;
    import androidx.appcompat.app.AppCompatActivity;
    import androidx.appcompat.widget.AppCompatButton;

    import android.content.Intent;
    import android.graphics.Color;
    import android.graphics.drawable.ColorDrawable;
    import android.os.Bundle;
    import android.os.CountDownTimer;
    import android.view.View;
    import android.widget.ImageView;
    import android.widget.RelativeLayout;
    import android.widget.TextView;
    import android.widget.Toast;

    import com.google.firebase.database.DataSnapshot;
    import com.google.firebase.database.DatabaseError;
    import com.google.firebase.database.DatabaseReference;
    import com.google.firebase.database.FirebaseDatabase;
    import com.google.firebase.database.ValueEventListener;

    import java.io.Serializable;
    import java.util.ArrayList;
    import java.util.List;
    import java.util.Locale;
    import java.util.concurrent.TimeUnit;

    public class MainActivity extends AppCompatActivity {

        private final List<QuestionsList> questionsLists = new ArrayList<>();
        private TextView quizTimer;
        private RelativeLayout option1Layout,option2Layout,option3Layout,option4Layout;
        private TextView option1TV,option2TV,option3TV,option4TV;
        private ImageView option1Icon,option2Icon,option3Icon,option4Icon;
```

```
        private TextView totalQuestionTV;
        private TextView currentQuestion;

private TextView questionTV;
private final DatabaseReference databaseReference =
FirebaseDatabase.getInstance().getReferenceFromUrl("https://quizflix-61e2f-default-
rtdb.firebaseio.com/");
//countdown timer
private CountDownTimer countDownTimer;

//current question position. By default 0 = first-question
private int currentQuestionPosition = 0;

//selected option number - value must be between 1-4
private int selectedOption = 0;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

quizTimer = findViewById(R.id.quizTimer);
option1Layout = findViewById(R.id.option1Layout);
option2Layout = findViewById(R.id.option2Layout);
option3Layout = findViewById(R.id.option3Layout);
option4Layout = findViewById(R.id.option4Layout);

option1TV = findViewById(R.id.option1TV);
option2TV = findViewById(R.id.option2TV);
option3TV = findViewById(R.id.option3TV);
option4TV = findViewById(R.id.option4TV);

option1Icon = findViewById(R.id.option1Icon);
option2Icon = findViewById(R.id.option2Icon);
option3Icon = findViewById(R.id.option3Icon);
option4Icon = findViewById(R.id.option4Icon);

questionTV = findViewById(R.id.questionTV);
totalQuestionTV = findViewById(R.id.totalQuestionTV);
currentQuestion = findViewById(R.id.currentQuestionTV);



// instructions for Quiz
InstructionsDialog instructionsDialog = new InstructionsDialog(MainActivity.this);
instructionsDialog.setCancelable(false);
instructionsDialog.getWindow().setBackgroundDrawable(new
ColorDrawable(Color.TRANSPARENT));
instructionsDialog.show();
```

```
final AppCompatButton nextBtn = findViewById(R.id.nextQuestionBtn);
databaseReference.addListenerForSingleValueEvent(new ValueEventListener() {
@Override
public void onDataChange(@NonNull DataSnapshot snapshot) {
final int getQuizTime = Integer.parseInt(snapshot.child("time").getValue(String.class));
for(DataSnapshot questions : snapshot.child("questions").getChildren()){
String getQuestion = questions.child("question").getValue(String.class);
String getOption1 = questions.child("option1").getValue(String.class);
String getOption2 = questions.child("option2").getValue(String.class);
String getOption3 = questions.child("option3").getValue(String.class);
String getOption4 = questions.child("option4").getValue(String.class);
int getanswer = Integer.parseInt(questions.child("answer").getValue(String.class));

QuestionsList questionsList = new
QuestionsList(getQuestion,getOption1,getOption2,getOption3,getOption4,getanswer);
questionsLists.add(questionsList);
}
//total questions to textview
totalQuestionTV.setText("/"+questionsLists.size());

//start timer
startQuizTimer(getQuizTime);

//start with first question
selectQuestion(currentQuestionPosition);

}

@Override
public void onCancelled(@NonNull DatabaseError error) {
Toast.makeText(MainActivity.this,"Failed to get data from
Firebase",Toast.LENGTH_SHORT).show();
}
});

option1Layout.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
selectedOption = 1;

selectOption(option1Layout,option1Icon);
}
});

option2Layout.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
selectedOption = 2;

selectOption(option2Layout,option2Icon);
```

```
}
});

option3Layout.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
selectedOption = 3;

selectOption(option3Layout,option3Icon);

}
});

option4Layout.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
selectedOption = 4;

selectOption(option4Layout,option4Icon);

}
});

nextBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
// check if user has selected an option or not
if(selectedOption !=0){
// set the user selected option
questionsLists.get(currentQuestionPosition).setUserSelectedAnswer(selectedOption);

// reset selected option to 0
selectedOption = 0;
currentQuestionPosition++; //increase current question postion ,getting next question

if(currentQuestionPosition < questionsLists.size()){
selectQuestion(currentQuestionPosition); // select next question
}
else{

// list has no questions
countDownTimer.cancel(); //stop timer
finishQuiz();
}

}
else{
Toast.makeText(MainActivity.this,"Please select an option",Toast.LENGTH_SHORT).show();
}
```

```
            }
});

}

private void finishQuiz(){

//create Result activity
Intent intent = new Intent(MainActivity.this,QuizResult.class);

//create bundle to pass quizQuestionsList
Bundle bundle = new Bundle();
bundle.putSerializable("questions",(Serializable) questionsLists);

intent.putExtras(bundle);
startActivity(intent);

finish();
}

private void startQuizTimer(int maxTimeInSeconds){

countDownTimer = new CountDownTimer(maxTimeInSeconds * 1000,1000) {
@Override
public void onTick(long millsUntilFinished) {
long getHour = TimeUnit.MILLISECONDS.toHours(millsUntilFinished);
long getMinute = TimeUnit.MILLISECONDS.toMinutes(millsUntilFinished);
long getSecond = TimeUnit.MILLISECONDS.toSeconds(millsUntilFinished);

String generateTime = String.format(Locale.getDefault(),"%02d:%02d:%02d",getHour,
getMinute-TimeUnit.HOURS.toMinutes(getHour),
getSecond-TimeUnit.MINUTES.toSeconds(getMinute));

quizTimer.setText(generateTime);
}

@Override
public void onFinish() {
finishQuiz();
}
};

countDownTimer.start();
}

private void selectQuestion(int questionListPosition){

//reset options for new question
resetOptions();
//getting the questions with details in the frontend
```

```
        questionTV.setText(questionsLists.get(questionListPosition).getQuestion());
        option1TV.setText(questionsLists.get(questionListPosition).getOption1());
        option2TV.setText(questionsLists.get(questionListPosition).getOption2());
        option3TV.setText(questionsLists.get(questionListPosition).getOption3());
        option4TV.setText(questionsLists.get(questionListPosition).getOption4());
        currentQuestion.setText("Question"+(questionListPosition+1));

    }

    private void resetOptions() {

        option1Layout.setBackgroundResource(R.drawable.round_back_white50_10);
        option2Layout.setBackgroundResource(R.drawable.round_back_white50_10);
        option3Layout.setBackgroundResource(R.drawable.round_back_white50_10);
        option4Layout.setBackgroundResource(R.drawable.round_back_white50_10);

        option1Icon.setImageResource(R.drawable.round_back_white50_100);
        option2Icon.setImageResource(R.drawable.round_back_white50_100);
        option3Icon.setImageResource(R.drawable.round_back_white50_100);
        option4Icon.setImageResource(R.drawable.round_back_white50_100);

    }

    private void selectOption(RelativeLayout selectedOptionLayout,ImageView selectedOptionIcon){
        // reset options to select now
        resetOptions();
        selectedOptionIcon.setImageResource(R.drawable.checked);
        selectedOptionLayout.setBackgroundResource(R.drawable.round_back_selected_option);
    }
}
```

4.2.9 QuizResult.java

```
package com.example.quizflix;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.AppCompatButton;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import org.w3c.dom.Text;

import java.util.ArrayList;
import java.util.List;

public class QuizResult extends AppCompatActivity {
```

```
private List<QuestionsList> questionsList = new ArrayList<>();
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_quiz_result);

final TextView scoreTV = findViewById(R.id.scoreTV);
final TextView totalScoreTV = findViewById(R.id.totalScoreTV);
final TextView correctTV = findViewById(R.id.correctTV);
final TextView incorrectTV = findViewById(R.id.incorrectTV);
final AppCompatButton shareBtn = findViewById(R.id.shareBtn);
final AppCompatButton reTakeQuizBtn = findViewById(R.id.reTakeQuizBtn);

//retrieving questions from MainActivity
questionsList = (List<QuestionsList>) getIntent().getSerializableExtra("questions");

totalScoreTV.setText("/"+questionsList.size());
scoreTV.setText(getCorrectAnswers() + "");
correctTV.setText(getCorrectAnswers() + "");
incorrectTV.setText(String.valueOf(questionsList.size()-getCorrectAnswers()));

//Share Result
shareBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
Intent sendIntent = new Intent(Intent.ACTION_SEND);
sendIntent.setType("text/plain");
sendIntent.putExtra(Intent.EXTRA_SUBJECT, "QuizFlix");
sendIntent.putExtra(Intent.EXTRA_TEXT,"My score ="+scoreTV.getText());

Intent shareIntent = Intent.createChooser(sendIntent,"Share Via");
startActivity(shareIntent);
}
});

reTakeQuizBtn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
//Retake Quiz
startActivity(new Intent(QuizResult.this,MainActivity.class));
finish();
}
});

}

private int getCorrectAnswers() {
int correctAnswer = 0;
for(int i=0;i<questionsList.size();i++){
```

```
        int getUserSelectedOption = questionsList.get(i).getUserSelectedAnswer();
        int getQuestionAnswer = questionsList.get(i).getAnswer();

        //check if user selected answer matches the correct answer
        if(getUserSelectedOption == getQuestionAnswer){
            correctAnswer++;
        }
    }
    return correctAnswer;
   }
}
```

Chapter 5

RESULT AND SNAPSHOT

5.1 Home page

Home page of the application gives the display of the quiz applcation's logo for some milli seconds

Figure 5.1 Home Page of Application

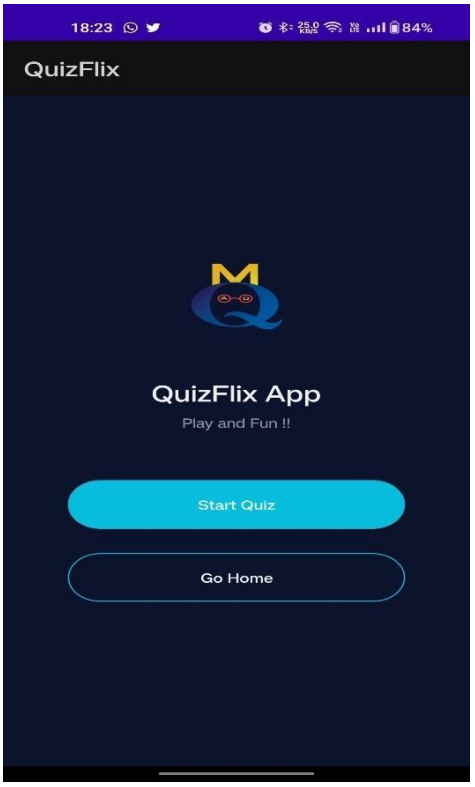

Figure 5.2 Start Quiz page

5.2 Start Quiz page

The start quiz page in the application gives the user two option either to start the quiz by clicking "Start Quiz" or to exit the application by clicking on "Go Home"

5.3 Instruction page

The Instruction page displays the basic instructions before starting the quiz explaining that user gets 3 minutes for finishing the quiz and user gets 1 point for each right option answered.

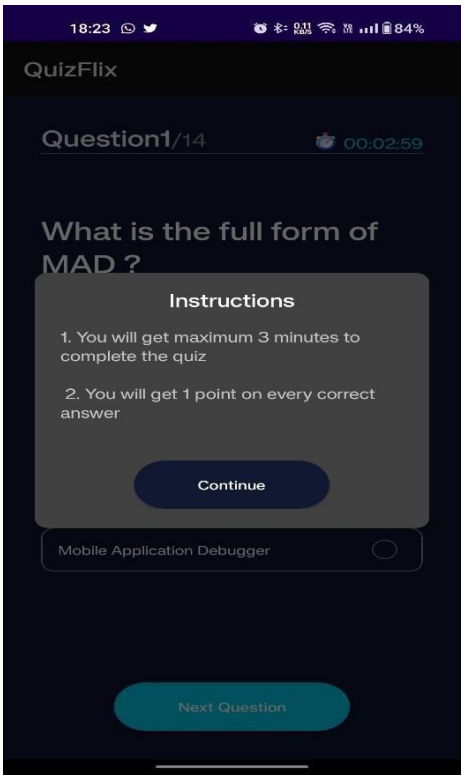

Figure 5.3 Instruction Page

Figure 5.4 Question card Page

5.4 Question card Page

The Question card page displays the question an the option for the quiz the questions and answers are fetched from the database which are referenced from the Firebase database.

5.5 Question card page

The Question card page displays the question an the option for the quiz the questions and answers are fetched from the database which are referenced from the Firebase database.

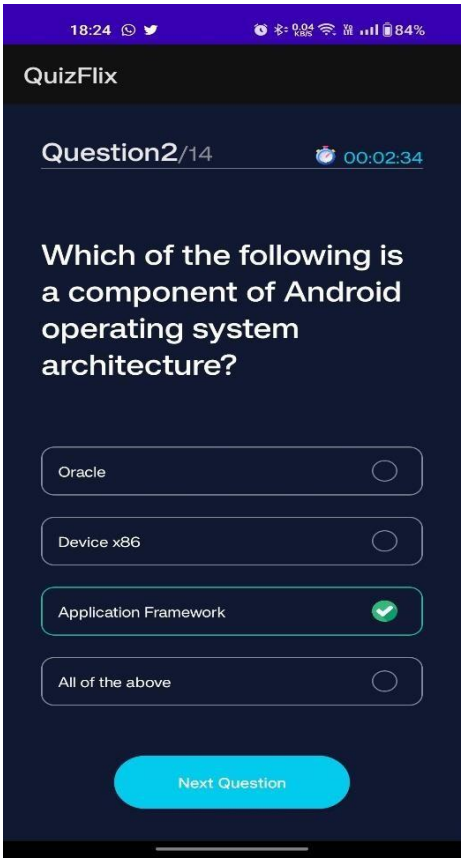

Figure 5.5 Question card Page

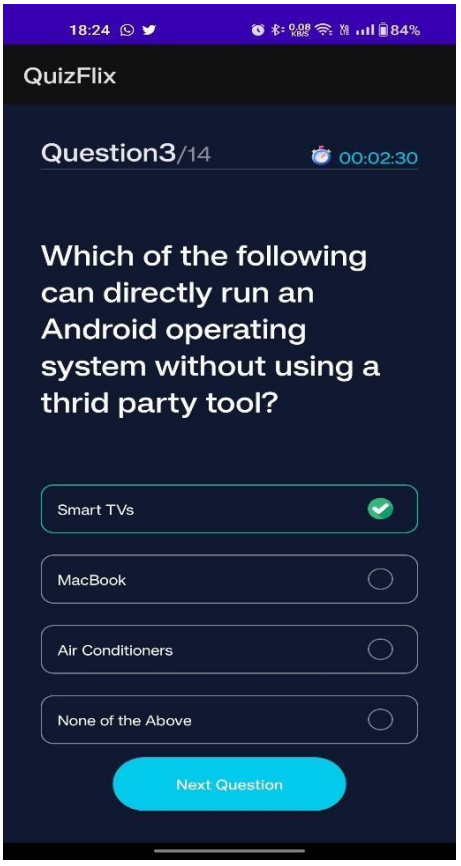

Figure 5.6 Option selected

5.6 Option selected

When the option is selected the circle image in the option gets selected and a tick marks fills the circle which represents that the option which has been selected.

5.7 Result Page

After all the questions have been answered then the result page will be displayed and will give the details of the number of questions attended right and number of questions attended wrong.

Figure 5.7 Result page

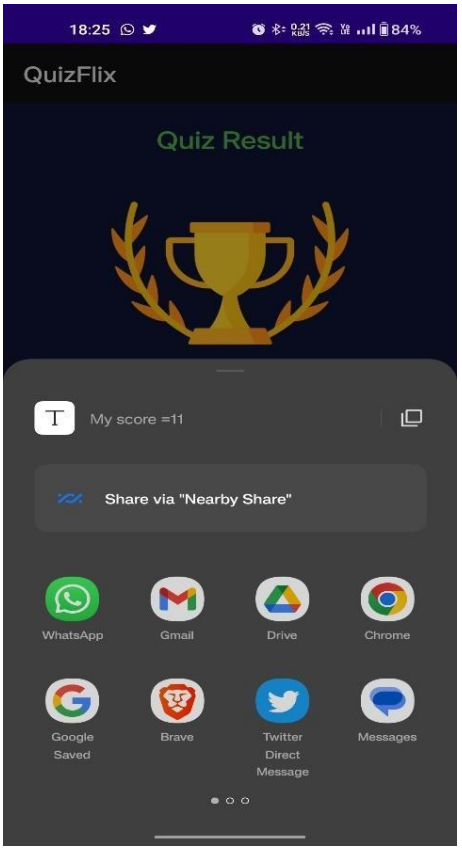

Figure 5.8 Score share Page

5.8 Score share Page

The share button on the result page will lead to the option to share the score through the messaging apps which sends a link from which other people can also take up the quiz.

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

In conclusion, this quiz app provides an engaging and educational experience for users. With its diverse range of topics and challenging questions, it encourages continuous learning and fosters intellectual growth. Whether you're a student, a trivia enthusiast, or simply seeking some fun, this app offers an enjoyable way to test your knowledge and expand your horizons.

6.2 Advantages:

• Flexibility of anytime, anywhere learning.

• Immediate feedback for quick assessment and improvement.

• Interactive and engaging experience enhances knowledge retention.

6.3 Future Enhancement:

- Personalized learning paths based on user performance and preferences
- Integration of social features for multiplayer quizzes and collaborative challenges
- Enhanced gamification elements with badges, levels, and rewards
- Utilization of artificial intelligence and machine learning for adaptive quizzes
- Integration of multimedia content such as videos and interactive simulations
- Improved accessibility features for users with disabilities
- Integration with learning management systems for seamless integration into educational environments
- Real-time analytics and performance tracking for users and administrators
- Expansion of question types to include multimedia-based questions, interactive scenarios, and case studies
- Integration with external resources and APIs to provide a broader range of content and topics.

Chapter 7

REFERENCES

[1] P. Jain, "What Makes Java a Powerful Programming Language," 11 February 2013. [Online]. Available: https://www.weblinkindia.net/blog/what-makes-java-a powerful-programming-language. [Accessed 22 June 2023].

[2] A. Rongala, "Benefits of Java over Other Programming Languages," 7 May 2015. [Online]. Available: https://www.invensis.net/blog/it/benefits-of-java-over-other programming-languages/. [Accessed 22 June 2023].

[3] "Java SE Downloads," [Online]. Available: http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html. [Accessed 22 June 2023].

[4] [3] "Android Developers," [Online]. Available: https://developer.android.com/index.html. [Accessed 22 June 2023].

[5] C. Aliferi, "Android Barcode and Qr Scanner Example," 2 December 2014. [Online]. Available: https://examples.javacodegeeks.com/android/android-barcode and-qr-scanner-example/. [Accessed 22 June 2023].

[6] "Android Pie Chart using MpAndroidChart Library Tutorial," Numetric Technologies, 29 January 2016. [Online]. Available: Android Pie Chart using AchartEngine Library Tutorial. [Accessed 22 June 2023].