OnLineQ

by

160315065 – ŞABAN EMRE BALIKÇI
140315014 - ZEYNEP GÖKÇE GÖRDÜK

DESIGN PROJECT REPORT

CELAL BAYAR
ÜNİVERSİTESİ

Department of Computer Engineering

2018

emrebalikci /onlineq

Abstract

The Quiz Android Mobile application is an application which gives you brief tutorials on various programming languages and technologies to the user who has installed this application. The mobile application also contains sample code, interview questions, and answers.

This application also conducts a quiz to the user who is interested in a particular technology for checking their knowledge on the subject. It has different levels like Basic Level and Advanced Level quiz. This application conducts a quiz in the form of multiple choice questions. After the completion of the quiz the application generates reports based on the quiz conducted.

The user can also test their knowledge on those technologies based on a timed quiz, we can include the above technology questions in this application. We can give a time frame for each question and all the questions are to be answered in that specified time period. Say for example we can give 20 seconds to answer a question. If the user gives an incorrect answer, will be exited out of the quiz.

Table of Contents

Chapter 1 – Introduction 1

1.1 Description of Project 1

1.2 Motivation 1

Chapter 2 - Requirements Analysis 2

2.1 Requirements Collection 2

2.2 Software Requirements Specification 3

Chapter 3 - System Architecture & Design 4

3.1 Android Architecture 4

3.1.1 The Linux Kernel 5

3.1.2 Android Run Time 5

3.1.3 Libraries 5

3.1.4 Android Application Framework 6

3.1.5 Applications 6

3.2 System Architecture 7

3.2.1 System Architecture Diagram 7

3.2.2 System Design Diagram 8

3.2.2.1 Sequence Diagram 8

3.2.2.2 Use Case Diagram 9

3.2.2.3 Class Diagram 10

Chapter 4 - API Guides 11

4.1 App Components 11

4.1.1 Intents & Intent Filters 11

4.2 App Manifest 13

4.2.1 The Manifest File 13

4.2 Text and Input 14

4.2.1 Input Method 14

4.2.2 Quiz Fragment 15

Chapter 5 - Implementation 16

5.1 Screens 16
5.1.1 Login Page 16
5.1.2 Signin Page 17
5.1.3 Category Page 17
5.1.4 Play Page 18
5.1.5 Quiz Page 19
_5.1.5 Quiz Page 19
Chapter 6 – Testing 22
6.1 Unit Testing 22
6.2 Compability Testing 23
6.3 User Testing 23

Chapter 7 - Conclusion 24
7.1 The Applicatiom Achivement 24
7.2 What I have learned ? 26

Chapter 8 - Reference 27

Chapter 1 – Introduction

1.1 Description of Project

The Quiz Android Mobile application is an application which gives you brief tutorials on various programming languages and technologies to the user who has installed this application. training on different technologies. The mobile application also contains sample codes, interview questions and answers.

This application also conducts a quiz to the user who is interested in a particular technology for checking their knowledge of the subject. This application conducts a quiz in the form of multiple choice questions. After the completion of the quiz the application generates reports based on the quiz conducted.

The user can also test their knowledge on those technologies based on a timed quiz, we can include the above technology questions on this application. We can give a time frame for each question and all the questions are to be answered in that specified time period. Say for example we can give 10 seconds to answer a question. If the user gives an incorrect answer he will be exited out of the quiz.

1.2 Motivation

We use smart phones in every day in our lives. The main motive of this project is to use those smart phones for making the availability of the notes and the lectures and conduct quiz using your smart phone.

Chapter 2 - Requirements Analysis

2.1 Requirements Collection

Requirements collection phase is the most important phase in a project as the collection of requirements determines the final outcome of the project. The requirements of the project are done such that the interface of the project is as simple as possible and as flexible as possible to use the application that is being developed. Intensive care is taken in collecting the requirements of the project.

In the system existing there isn't any mobile application which can give online classes for a specific course. There isn't any application which is that effective to conduct classes and quizzes at the same time.

So taking this problem into consideration the present application has been developed and this application allows users to take classes online, read the notes online increase his knowledge on the subject and take the quiz. He/ She can take two types of quizzes based on the type he is interested in. He/ She can take either a timed quiz or a normal quiz. In the normal quiz the user simply takes the quiz and he has to answer all the questions in a row and he can take his own time and when he gives a wrong answer the game gets exited and there will be a popup saying the quiz is over and asks to retake the quiz or restart the game or go to home page. He can also take the timed quiz in which he has to complete the timed quiz in a specified amount of time and after the end of time the application says the given time is over and asks to retake the quiz or restart the game or go to home page.

2.2 Software Requirements Specification

Operating System: Ubuntu

Database: Firebase

Tools: Android Studio

Platform: Android SDK Framework

Technologies Used: Java

Network: Internet Connectivity Required.

Chapter 3 - System Architecture & Design

3.1 Android Architecture

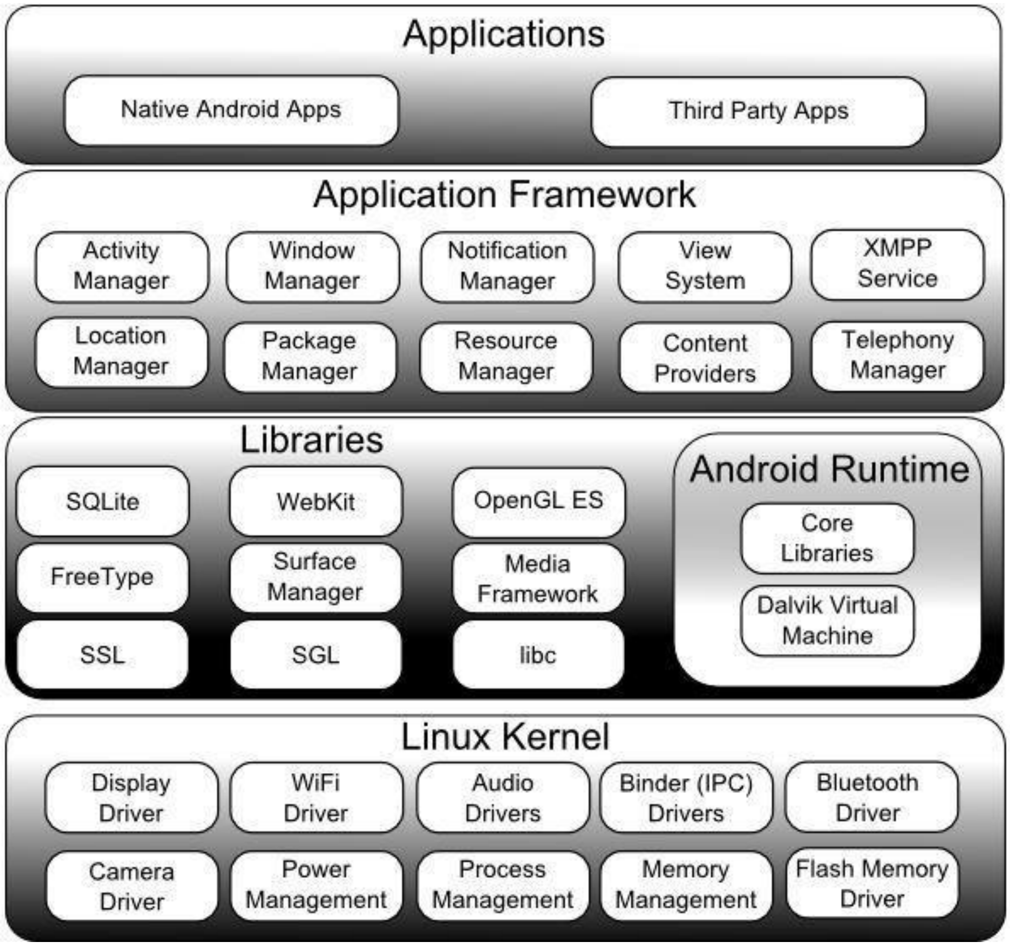


Figure 1 Android Architecture [1]

The different software components in the Android Operating System are arranged in the form of a stack. The different software components include

- The Linux Kernel
- Libraries
- Android Run Time
- Application Framework
- Applications [1]

3.1.1 The Linux Kernel

The Linux Kernel is the bottom layer of the architecture and the whole android operating system is built on the top of the Linux Kernel with some changes in the architecture. It is built on Linux 2.6 Kernel and some changes are made to its architecture. Linux Kernel is the core of the operating system. Process Management, Memory Management and Device Management are some of the functionality provided by the Linux Kernel. [2]

3.1.2 Android Run Time

The Android Emulator simulates various Android phone, tablet, Android Wear, and Android TV devices on your computer. It comes with predefined configurations for popular device types and can transfer data faster than a device connected over USB.

Each instance of the emulator uses an Android Virtual Device (AVD) to configure its size, form factor, Android version, and various hardware characteristics. To effectively test your app, you should create an AVD that models each device type that your app is designed to support.

Each AVD functions as an independent device, with its own private storage for user data, SD card, and so on. By default, the emulator stores the user data, SD card data, and cache in directory specific to that AVD. When you launch the emulator, it loads the user data and SD card data from the AVD directory.

3.1.3 Libraries

When developing apps that support multiple API versions, you may want a standard way to provide newer features on earlier versions of Android or gracefully fall back to equivalent functionality. Rather than building code to handle earlier versions of the platform, you can leverage these libraries to provide that compatibility layer. In addition, the Support Libraries provide additional convenience classes and features not available in the standard Framework API for easier development and support across more devices.

Originally a single binary library for apps, the Android Support Library has evolved into a suite of libraries for app development. Many of these libraries are now a strongly recommended, if not essential, part of app development.

This document provides an overview of the support library to help you understand its components and how to use it effectively in your app.

3.1.4 Android Application Framework

The Android applications directly interact with the Android Framework to run and are managed here. Resource Management, Voice call management and activities like these are handled by the application framework.

The Android framework includes the following key services:

- Activity Manager – The activity stack and the application lifecycle are controlled by activity manager.
- Content Providers – The data is shared and published with other applications using this content providers.
- Resource Manager – The non-code embedded resources such as strings, color settings and user interface layouts are accessed using this resource manager.
- Notifications Manager – The display alerts and notifications to the user are given by this notifications manager.

3.1.5 Applications

The top of the android stack are the applications. This includes both the applications that are preinstalled apps that are already on the device and the new ones which we develop and install. [7]

3.2 System Architecture

3.2.1 System Architecture Diagram

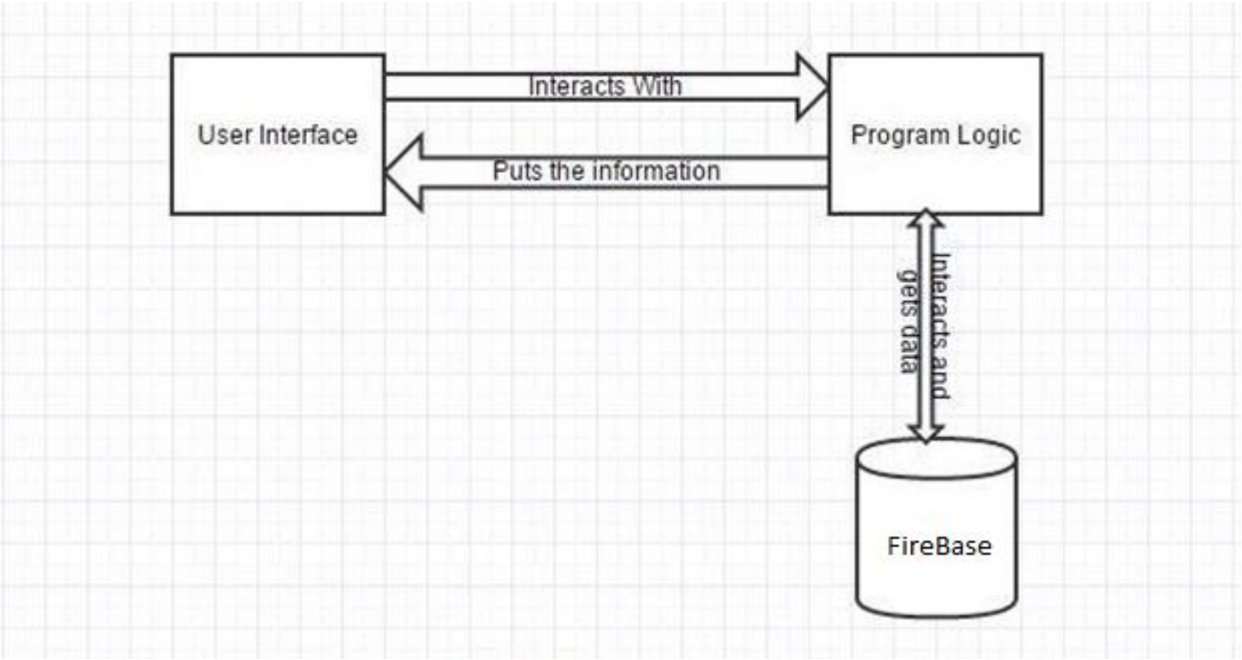


Figure 2 System Architecture

The above architecture diagram clearly shows the architecture of the Project. Using the User Interface the user of the application interacts with the programmed business logic. The business gets the data from the database which is Firebase in this application. This Interacts and gives the data to the programmed logic and the information is processed and displayed back on the user interface.

3.2.2 System Design Diagram

3.2.2.1 Sequence Diagram

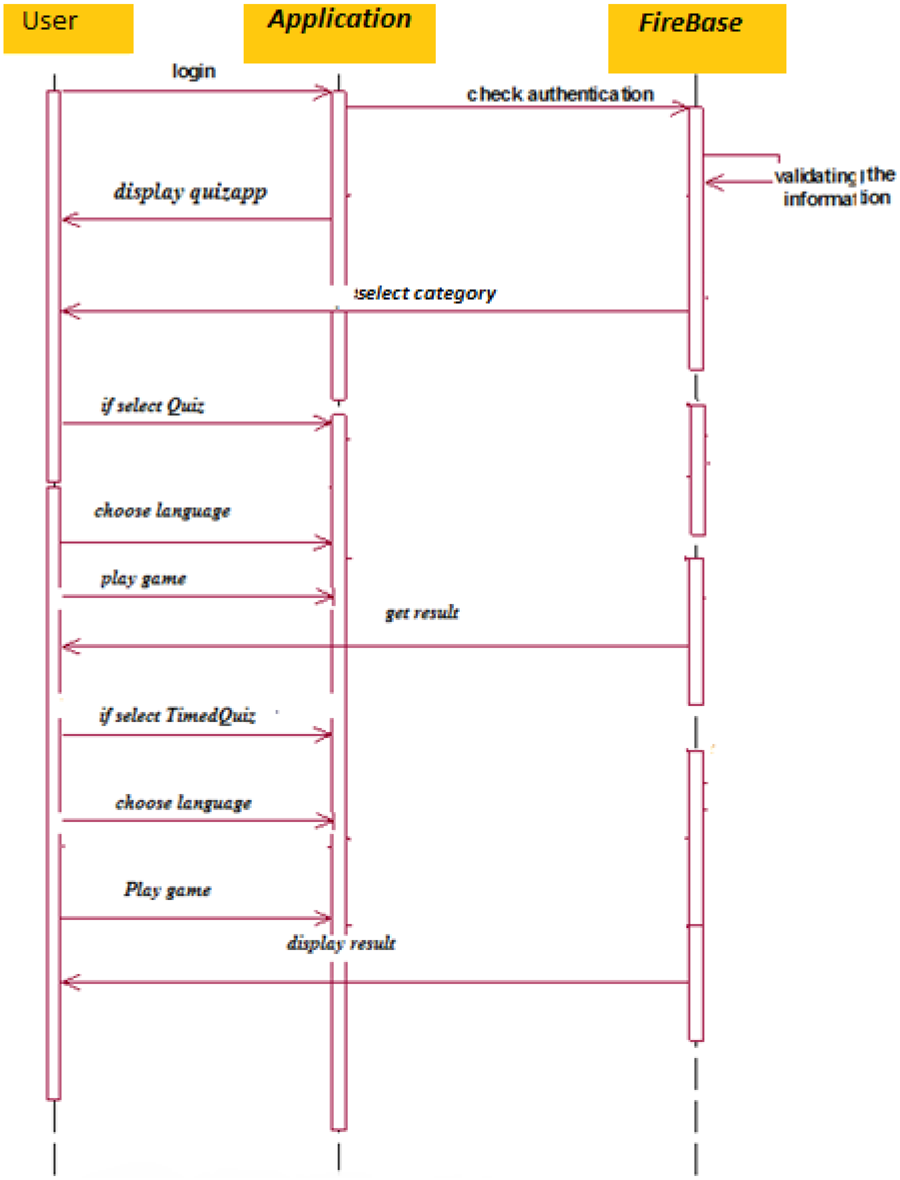


Figure 3 Sequence Diagram

3.2.2.2 Use Case Diagram

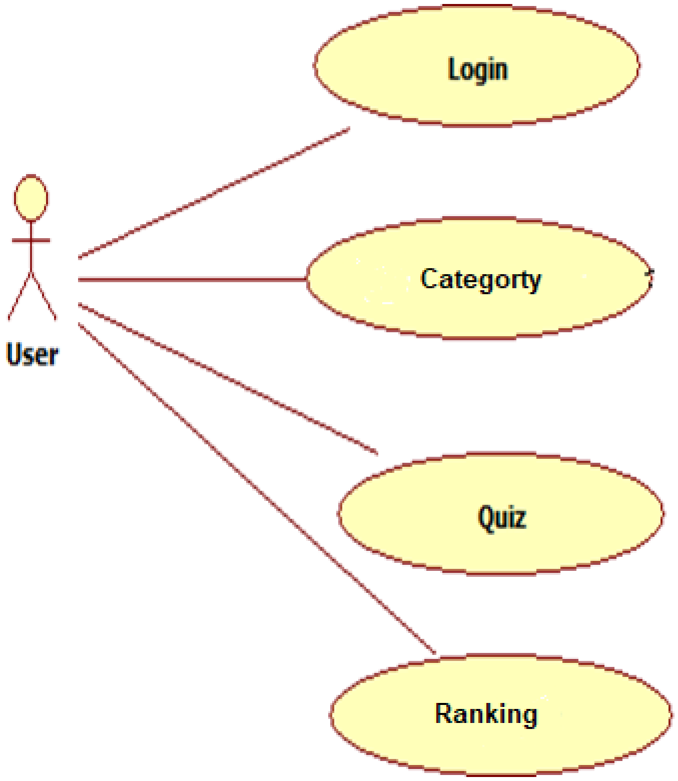


Figure 4 Use Case Diagram

The User of the application interacts with the application interacts with the application. Then he can select one out of three options i.e. Technology Training, Quiz and Timed Quiz. When User selects technology training he can select a technology he can select the pdf or videos for the lecture format. User can also select Quiz, attend the quiz and he/she can retake the quiz on a different technology. User can also select timed quiz and he/she can take the quiz on the technology he/she likes.

3.2.2.3 Class Diagram

The user of the application logins and registers with a username and using the Signin( ) and Login( ) methods. After logging into the application the user has three options now. The user can take ranking or take quiz or timed quiz or take three one after the other.

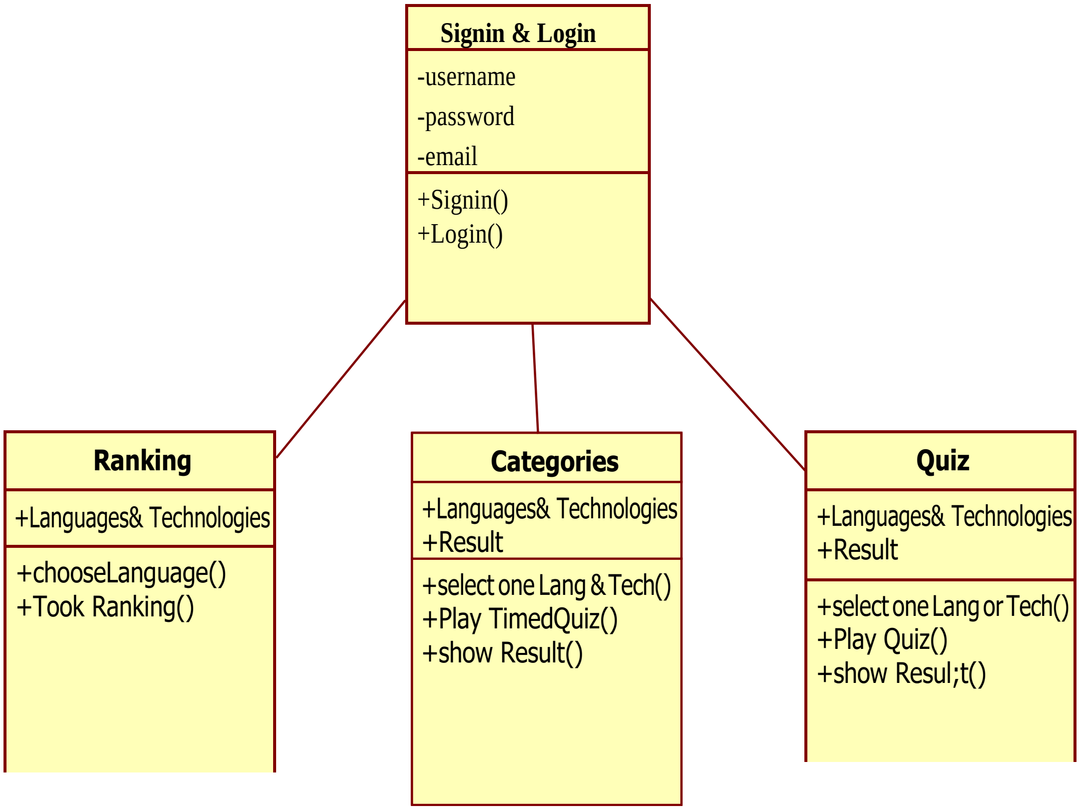


Figure 5 Class Diagram

If the user selects the Training he/she can select the language/technology that he/she is interested in. User can do that using the methods chooselanguage( ) which selects one language out of different technologies that are available and user can taketraining( ), the functionality is defines in this method. If the user selects Timed Quiz he/she can select the language/technology that he/she is interested in. He/she can do that using the methods chooselanguage( ) which selects one language out of different technologies that are available and user can play the timed quiz using the playtimedquiz( ) method and he can view the results using the showresult( ).

Chapter 4 - API Guides

4.1 App Components

4.1.1 Intents & Intent Filters

The action from another app component is requested using a messaging object known as intent.

The usage of intent in the present application:

```
<?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="@drawable/bg_vertical"
    tools:context="penguin.onlineq.MainActivity">


    <ImageView
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="20dp"
        android:src="@drawable/logo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/wrap_login"/>

    <RelativeLayout
        android:id="@+id/wrap_login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="8dp"
        android:layout_centerInParent="true"
        android:orientation="vertical"      >

        <android.support.v7.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:app="http://schemas.android.com/apk/res-auto"
            android:id="@+id/info_login"

            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            app:cardElevation="4dp">

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="vertical"
                android:layout_marginLeft="8dp"
                android:layout_marginRight="8dp">

                <com.rengwuxian.materialedittext.MaterialEditText
```

```
          android:id="@+id/edtUser"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:hint="User name"
          android:textColor="@color/colorPrimary"

          android:textColorHint="@color/colorPrimary"
          android:textSize="24sp"
          app:met_baseColor="@color/colorPrimary"
          app:met_floatingLabel="highlight"
          app:met_singleLineEllipsis="true" />

        <com.rengwuxian.materialedittext.MaterialEditText
          android:id="@+id/edtPassword"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:hint="Password"
          android:inputType="textPassword"
          android:textColor="@color/colorPrimary"
          android:textColorHint="@color/colorPrimary"
          android:textSize="24sp"
          app:met_baseColor="@color/colorPrimary"
          app:met_floatingLabel="highlight"
          app:met_singleLineEllipsis="true" />


      </LinearLayout>

    </android.support.v7.widget.CardView>

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_below="@+id/info_login"
      android:layout_margin="8dp"
      android:orientation="horizontal"
      android:weightSum="2">

      <Button
        android:id="@+id/btn_sign_up"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Sign Up" />

      <Button
        android:id="@+id/btn_sign_in"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Login" />
    </LinearLayout>
  </RelativeLayout>


</RelativeLayout>
```

Every intent in the manifest file is defined by an intent filter and this is for defining the main page and the page that is to be launched first during the run time of the application. [3]

4.2 App Manifest

The essential information like Functionality and requirements of your android application are described in the app manifest file. The package name which serves as a unique identifier for the application is named by the app manifest. The host application component processes determined by the app manifest. The permission to interact with the applications are declared in the app manifest. This app manifest has the minimum API level that is required to host the application, permissions to access protected parts of that API and it also has the list of libraries that are to be linked. The activities, services, broadcast receivers, and content providers that the application is composed of are described in the app manifest this allows the application to know which screen is to be launched first. [4][9]

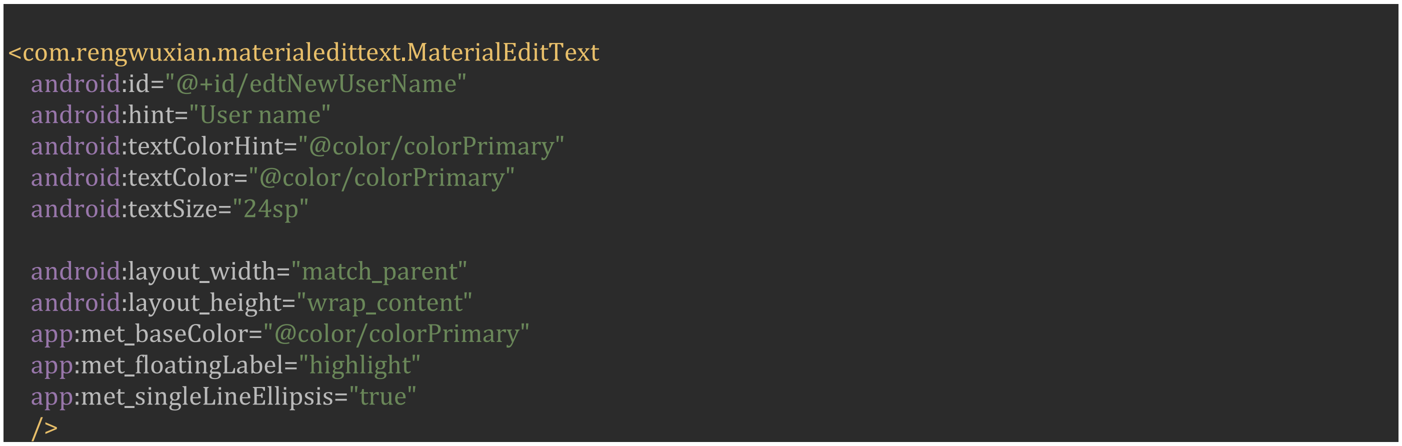
4.2.1 The Manifest File

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="penguin.onlineq">

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <receiver android:name=".BroadcastReciver.AlarmReceiver"/>
    <activity android:name=".Home" />
    <activity android:name=".Start" />
    <activity android:name=".Playing" />
    <activity android:name=".Done" />
    <activity android:name=".ScoreDetail"></activity>

    <service android:name=".BroadcastReciver.MyFirebaseIdService">
        <intent-filter>
            <action android:name="com.google.firebase.MESSAGING_EVENT"/>
        </intent-filter>
    </service>
</application>
</manifest>
```

4.2 Text and Input

4.2.1 Input Method

The text and input and other text related functions are used to edit, copy and paste text in the application. Here is an example of how edit text is used in the application.



```
<com.rengwuxian.materialedittext.MaterialEditText
  android:id="@+id/edtNewUserName"
  android:hint="User name"
  android:textColorHint="@color/colorPrimary"
  android:textColor="@color/colorPrimary"
  android:textSize="24sp"

  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  app:met_baseColor="@color/colorPrimary"
  app:met_floatingLabel="highlight"
  app:met_singleLineEllipsis="true"
  />
```

The hint name given is the player name in the application and this gives the user a hint to take a look at how the name is to be given and when once clicked on this field the user will be able to edit the text the layout i.e. height, width of the text are given here. [6]

4.2.2 Quiz Fragment

```
<LinearLayout
  android:orientation="vertical"
  android:id="@+id/group"
  android:paddingTop="10dp"
  android:layout_width="match_parent"
  android:layout_height="wrap_content">

  <LinearLayout
    android:weightSum="2"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
      android:id="@+id/txtScore"
      android:textColor="@android:color/white"
      android:gravity="center_horizontal"
      android:text="0"
      android:textSize="36sp"
      android:layout_weight="1"
      android:layout_width="0dp"
      android:layout_height="wrap_content" />

    <TextView
      android:id="@+id/txtTotalQuestion"
      android:textColor="@android:color/white"
      android:gravity="center_horizontal"
      android:text="1/10"
      android:textSize="36sp"
      android:layout_weight="1"
      android:layout_width="0dp"
      android:layout_height="wrap_content" />

  </LinearLayout>

  <ProgressBar
    android:id="@+id/progressBar"
    style="@style/Widget.AppCompat.ProgressBar.Horizontal"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:max="5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />


  <Button
    android:id="@+id/btnAnswerA"
    android:text="Answer A"
    android:textColor="@android:color/white"
    android:textStyle="bold"
    android:theme="@style/Widget.AppCompat.Button.Borderless"
    android:background="#FF8A80"
    android:layout_margin="8dp"
    android:foreground="?android:attr/selectableItemBackground"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

Chapter 5 - Implementation

5.1 Screens

5.1.1 Login Page

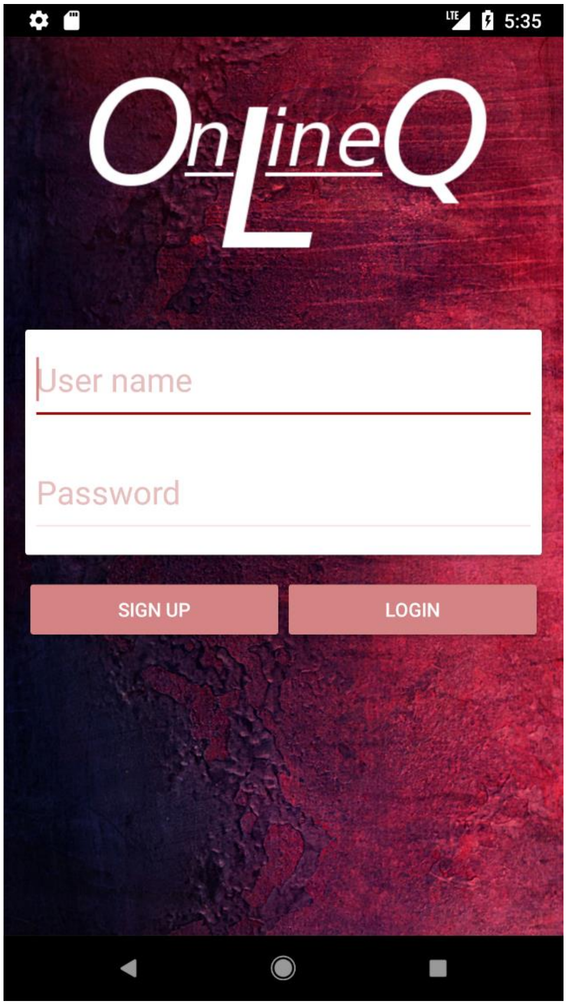


Figure 8 Login Page

This is the first page of the application to login or register to go to the Online Application page.

5.1.2 Signin Page

5.1.3 Category Page

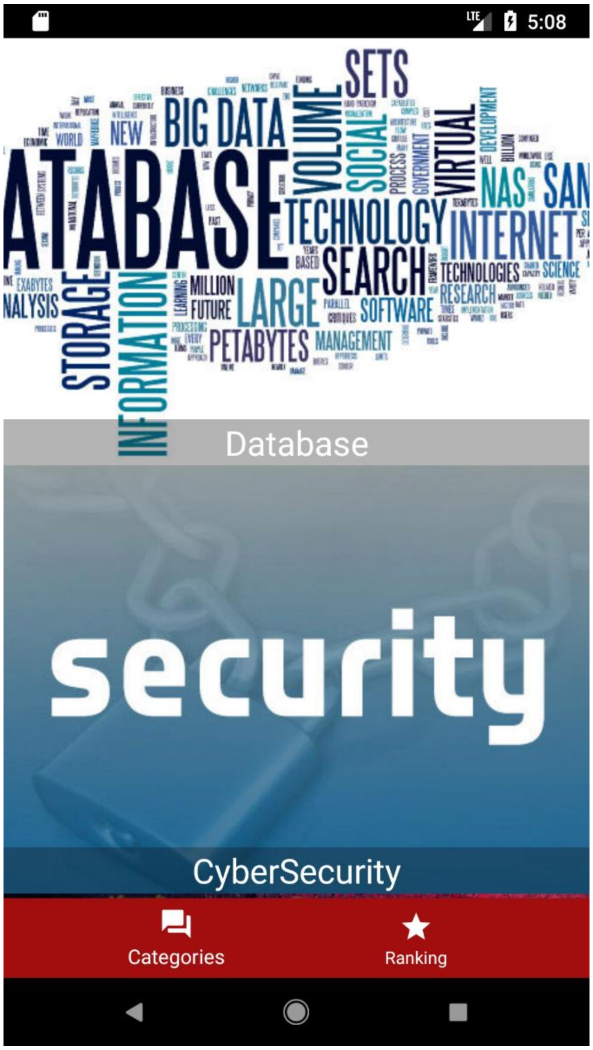


Figure 10 Category entering Page

This is the page select one of the user categories

5.1.4 Play Page

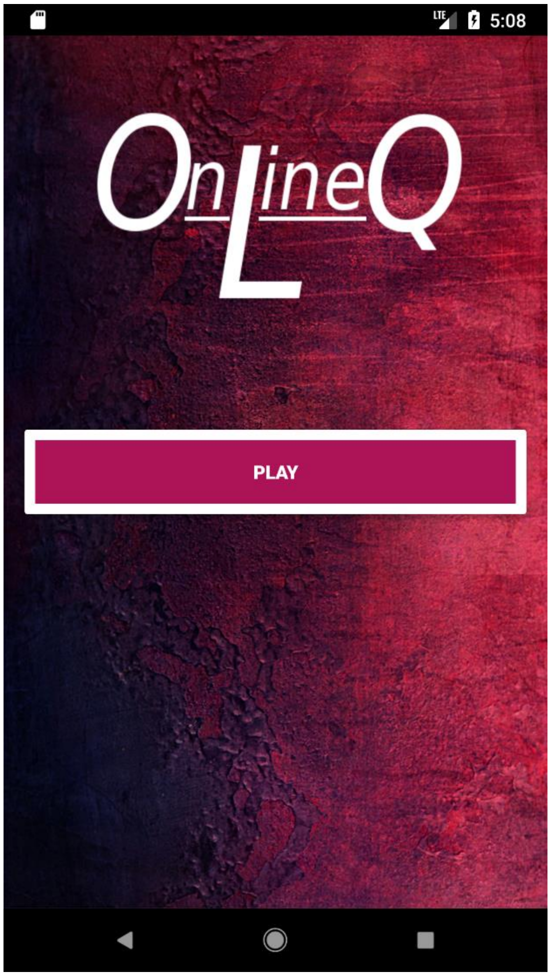


Figure 11 Player Game entering Page

Starts when the user is ready

5.1.5 Quiz Page

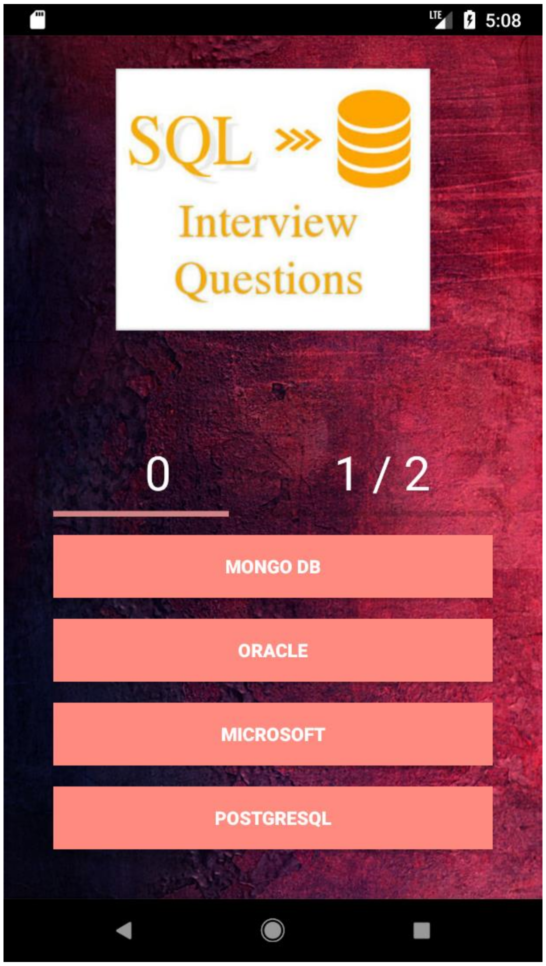
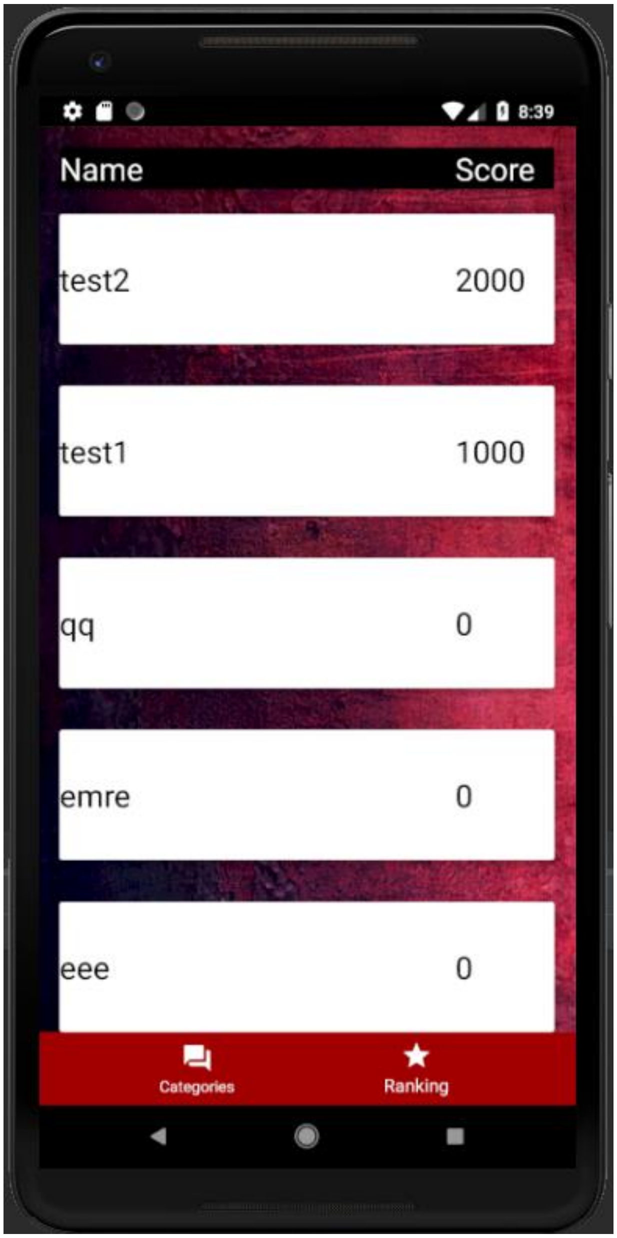


Figure 12 Question Page

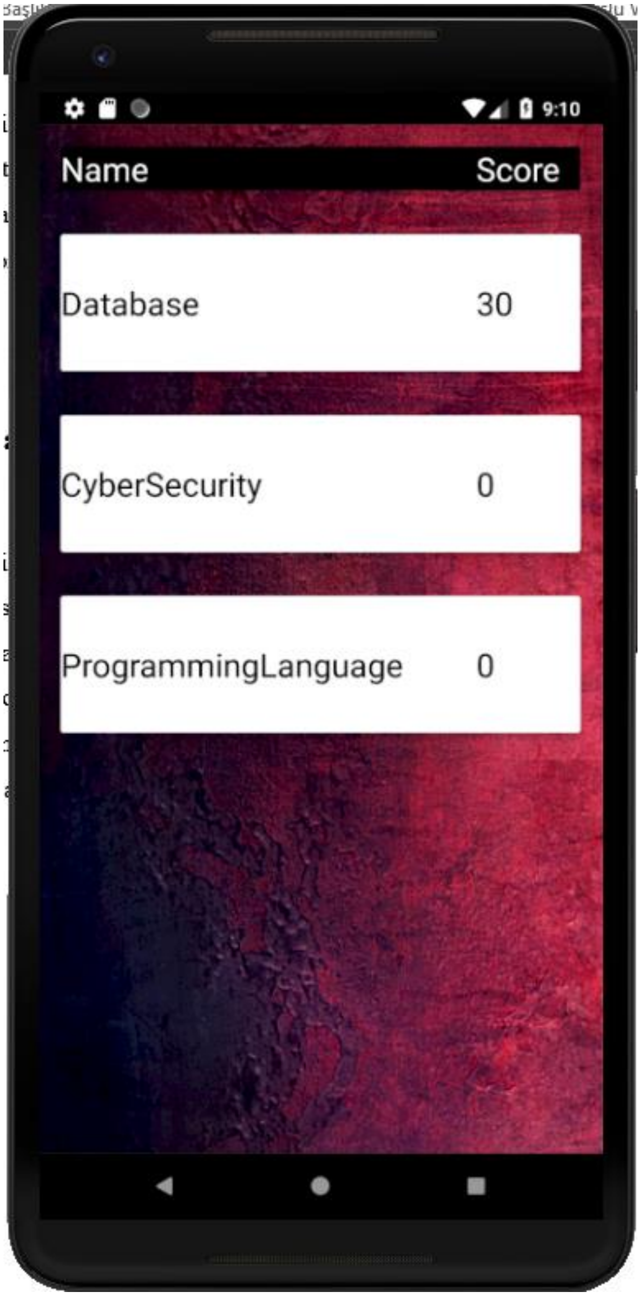
The user quizzes and if the answer is wrong the exam will end



Name	Score
test2	2000
test1	1000
qq	0
emre	0
eee	0

Figure 13 Ranking Page

Shows the total scores of all users.



Name	Score
Database	30
CyberSecurity	0
ProgrammingLanguage	0

Figure 14 Score Page

If the user clicks on Username, they can see their scores in all categories

How does the scoring algorithm in the quiz works?

The scoring algorithm for the quiz is as following:

The first question is worth 100 points. Every next question is worth more points, and these extra points increase with a 100 for every question. So every question becomes more valuable, see the first column below (question points).

Then you have a time bonus. This way you get more points when you answer more quickly. It's a multiplier of the points that a question is worth and the maximum multiplier is 0.8.

See the example below for an indication of the maximum score for a quiz with 7 questions!

Question	Points	Time bonus	Final Score
1	100	80	180
2	300	240	720
3	600	480	1,800
4	1,000	800	3,600
5	1,500	1,200	6,300
6	2,100	1,680	10,080
7	2,800	2,240	15,120

Are you having difficulties with importing questions to your quiz?

The most common reason why uploads fail is that the CSV is not in a UTF-8 format. You can set your Excel to save CSV files as UTF-8 in the following way:

1. Go to "Save as" and select the dropdown "Extra" and then select "Weboptions"
2. The go to "Encoding" tab and set it to UNICODE UTF-8

Please note that Excel on MAC cannot save in UTF-8 (nobody knows why). A simple work-around is uploading the file to Google Docs and then downloading it as a CSV file, it will then automatically have a UTF-8 encoding.

Chapter 6 - Testing

6.1 Unit Testing

Unit Testing is a software testing method in which small parts of the code are separated from other parts and the functionality is checked whether it works as expected or not.

Table 1 Unit Testing

Test Case	**Expected Result**	**Pass/ Fail**
Start Page	Should display 2 buttons, Quiz results and Quiz Page	Pass
Pressing Quiz results button	Should display quiz results	Pass
Pressing Quiz button	Should display home page which has 3 buttons: Technology training, Quiz and Timed Quiz	Pass
Pressing Technology Training button	Should display a list box which has 4 list box items, C, C++, Java and Android	Pass
Selecting C technology	Should open a dialog box which has topics	Pass
Selecting a topic	Selecting a topic should take you to a page which has 2 buttons pdf and Video	Pass
Pressing pdf button	Should open the respective pdf file and give the contents.	Pass
Pressing video button	Should open the respective	Pass

	video file and play the video.	
Selecting other topics	Should do same as for the first topic selection.	Pass
Selecting other technologies like C++, Java and Android	Should respond same as that is done for C.	Pass
Pressing Quiz button	Should display a list box which has 4 list box items, C, C++, Java and Android	Pass
After selection of technology	Should start the quiz	Pass
After starting the quiz	Radio buttons should work	Pass
In quiz	If we press next without selecting an option the game should be over and dialog box should be displayed.	Pass
In After completion quiz	Should say game over and display result.	Pass
Pressing Timed Quiz button	Should display a list box which has 4 list box items, C, C++, Java and Android	Pass
After selection of technology	Should start the Timed quiz	Pass
After starting the timed quiz	The timer should work	Pass
In timed quiz	If we press next without selecting an option the game should be over and dialog box should be displayed.	Pass
In After completion quiz	Should say game over and display result.	Pass

If Time completed	Should say your given time is completed	Pass

6.2 Compatibility Testing

This application was tested and used on different devices like LG G3, Google Nexus 4. The application worked fine and is stable. The application worked fine both in landscape and portrait modes and there isn't any problem with the resolution or compatibilty.

6. 3 User Testing

The present application was tested by my friends who are using different mobile devices that has Lollipop Android and that seemed to be working fine and they were satisfied with the performance and responsiveness of the application and how the app worked.

Chapter 7 - Conclusion

7.1 The Application Achievements

The application Tutorials and Quiz Android Mobile application is an unique android application that is for a video lectures, tutorials and quiz. The main objective of the project is achieved and can be used to see video classes and posting of notes on the application. The other big feature of the application is also to conduct online quiz and timed quiz. These both are achieved using this application.

7.2 What have I learnt ?

The application Tutorials and Quiz Android Mobile application has been implemented successfully on different Android devices and this enhanced my confidence in Android development. This is the first time that I am developing an Android Application and this gave me an idea of an Android Application Development, it helped me learn its database SQLite and how to use API's in Android Development and I would like to learn more about its development in the coming days and would like to try and develop more apps in Android.

Chapter 8 - References

[1] Android Architecture, Shiju P John February 3, 2017
http://www.eazytutz.com/android/android-architecture/

[2] An Overview of the Android Architecture, Online, July 03, 2014
http://www.techotopia.com/index.php/An_Overview_of_the_Android_Architecture

[3] Intent and Intent Filters, Online, 2017
http://developer.android.com/guide/components/intents-filters.html

[4] App Manifest, Online, 2017
http://developer.android.com/guide/topics/manifest/manifest-intro.html

[5] Unit Testing, Online, July 2009
http://en.wikipedia.org/wiki/Unit_testing

[6] Creating an Input Method, Online, 2015 http://developer.android.com/guide/topics/text/creating-input-method.html#DefiningIME

[7] Android Architecture, Online, 2017
http://www.tutorialspoint.com/android/android_architecture.htm

[8] Android Architecture, Pritesh Taral, 2017
http://www.c4learn.com/android/android-os-architecture/

[9] Working with AndroidManifest.xml, Online, 2017
http://developer.xamarin.com/guides/android/advanced_topics/working_with_androidmanifest.xml/

[10] Android Application, Online, 2017
http:/www.uml-diagrams.org/android-application-uml-deployment-diagram-example.html

Code For Project :

```
package penguin.onlineq.BroadcastReciver;

import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.RingtoneManager;
import android.net.Uri;
import android.support.v4.app.NotificationCompat;

import penguin.onlineq.MainActivity;
import penguin.onlineq.R;

public class AlarmReceiver extends BroadcastReceiver{

    @Override
    public void onReceive(Context context, Intent intent) {
        long when = System.currentTimeMillis();
        NotificationManager notificationManager=
(NotificationManager)context.getSystemService(Context.NOTIFICATION_SERVICE);

        Intent notificationIntent =new Intent(context, MainActivity.class);
        notificationIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

        PendingIntent
pendingIntent=PendingIntent.getActivity(context,0,notificationIntent,PendingIntent.FLAG_UPDATE_CURRENT);
        Uri alarmsound= RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);

        NotificationCompat.Builder builder= new NotificationCompat.Builder(context)
                .setSmallIcon(R.mipmap.ic_launcher_round)
                .setContentTitle("OnLineQ - Online Quiz Up")
                .setSound(alarmsound)
                .setAutoCancel(true)
                .setWhen(when)
                .setContentIntent(pendingIntent)
                .setVibrate(new long[]{1000,1000,1000,1000,1000});
        notificationManager.notify(0,builder.build());

    }
}
```

```
package penguin.onlineq.BroadcastReciver;

import android.util.Log;

import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.iid.FirebaseInstanceIdService;

public class MyFirebaseIdService  extends FirebaseInstanceIdService{
    @Override
    public void onTokenRefresh() {
        super.onTokenRefresh();
        String refreshedToken =FirebaseInstanceId.getInstance().getToken();
        sendRegistrationToServer(refreshedToken);
    }

    private void sendRegistrationToServer(String refreshedToken) {
        Log.d("Token",refreshedToken);

    }
}
```

```
package penguin.onlineq.BroadcastReciver;

import android.content.Intent;
import android.support.v4.content.LocalBroadcastManager;
import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;
import penguin.onlineq.Common.Common;

public class MyFirebaseMessagingService extends FirebaseMessagingService {
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        handleNotification(remoteMessage.getNotification().getBody());
        super.onMessageReceived(remoteMessage);
    }

    private void handleNotification(String body) {
        Intent pushNotification =new Intent(Common.STR_PUSH);
        pushNotification.putExtra("message",body);
        LocalBroadcastManager.getInstance(this).sendBroadcast(pushNotification);

    }
}
```

```
package penguin.onlineq.Common;

import java.util.ArrayList;
import java.util.List;
import penguin.onlineq.Model.Question;
import penguin.onlineq.Model.User;

/**
 * Created by emrebalikci on 10.12.2017.
 */

public class Common {
    public static String categoryId,categoryName;
    public static User currentUser;
    public static List<Question> questionList=new ArrayList<>();

    public  static  final  String STR_PUSH="pushNotification";


}
```

```
package penguin.onlineq.Interface;

import android.view.View;

/**
 * Created by emrebalikci on 09.12.2017.
 */

public interface ItemClickListener {

    void onClick(View view,int position,boolean isLongClick);
}
```

```
package penguin.onlineq.Interface;

public interface  RankingCallBack<T>{
    void callBack(T ranking);
}
```

```
package penguin.onlineq.Model;

/**
 * Created by emrebalikci on 09.12.2017.
 */

public class Category {
   private  String Name;
   private  String Image;


   public Category(){

   }

   public Category(String name, String image) {
      Name = name;
      Image = image;
   }

   public String getName() {
      return Name;
   }

   public void setName(String name) {
      Name = name;
   }

   public String getImage() {
      return Image;
   }

   public void setImage(String image) {
      Image = image;
   }
}
```

```
package penguin.onlineq.Model;

/**
 * Created by emrebalikci on 26.12.2017.
 */
public class Question {
   private String Question;
   private String AnswerA;
   private String AnswerB;
   private String AnswerC;
   private String AnswerD;
   private String CategoryId;
   private String IsImageQuestion;
   private String CorrectAnswer;

   public Question() { }

   public Question(String question, String answerA, String answerB, String answerC, String answerD, String categoryId,
String isImageQuestion, String correctAnswer) {
      Question = question;
      AnswerA = answerA;
      AnswerB = answerB;
      AnswerC = answerC;
      AnswerD = answerD;
      CategoryId = categoryId;
      IsImageQuestion = isImageQuestion;
      CorrectAnswer = correctAnswer;
```

```
    }

    public String getCorrectAnswer() {
        return CorrectAnswer;
    }

    public void setCorrectAnswer(String correctAnswer) {
        CorrectAnswer = correctAnswer;
    }

    public String getQuestion() {
        return Question;
    }

    public void setQuestion(String question) {
        Question = question;
    }

    public String getAnswerA() {
        return AnswerA;
    }

    public void setAnswerA(String answerA) {
        AnswerA = answerA;
    }

    public String getAnswerB() {
        return AnswerB;
    }

    public void setAnswerB(String answerB) {
        AnswerB = answerB;
    }

    public String getAnswerC() {
        return AnswerC;
    }

    public void setAnswerC(String answerC) {
        AnswerC = answerC;
    }

    public String getAnswerD() {
        return AnswerD;
    }

    public void setAnswerD(String answerD) {
        AnswerD = answerD;
    }

    public String getCategoryId() {
        return CategoryId;
    }

    public void setCategoryId(String categoryId) {
        CategoryId = categoryId;
    }

    public String getIsImageQuestion() {
        return IsImageQuestion;
    }

    public void setIsImageQuestion(String isImageQuestion) {
        IsImageQuestion = isImageQuestion;
}
```

```java
package penguin.onlineq.Model;

/**
 * Created by emrebalikci on 27.12.2017.
 */

public class QuestionScore {
    private String Question_Score;
    private String User;
    private String Score;
    private String CategoryId;
    private String CategoryName;

    public QuestionScore(String question_Score, String user, String score, String categoryId, String categoryName) {
        Question_Score = question_Score;
        User = user;
        Score = score;
        CategoryId = categoryId;
        CategoryName = categoryName;
    }

    public QuestionScore(){

    }

    public String getQuestion_Score() {
        return Question_Score;
    }

    public void setQuestion_Score(String question_Score) {
        Question_Score = question_Score;
    }

    public String getUser() {
        return User;
    }

    public void setUser(String user) {
        User = user;
    }

    public String getScore() {
        return Score;
    }

    public void setScore(String score) {
        Score = score;
    }

    public String getCategoryId() {
        return CategoryId;
    }

    public void setCategoryId(String categoryId) {
        CategoryId = categoryId;
    }

    public String getCategoryName() {
        return CategoryName;
    }

    public void setCategoryName(String categoryName) {
        CategoryName = categoryName;
    }
}
```

```
package penguin.onlineq.Model;

public class Ranking {
    private String userName;
    private  long score;

    public Ranking(){

    }

    public Ranking(String userName, long score) {
        this.userName = userName;
        this.score = score;
    }

    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public long getScore() {
        return score;
    }

    public void setScore(long score) {
        this.score = score;
    }
}
```

```
package penguin.onlineq.Model;

/**
 * Created by EMRE on 20.10.2017.
 */

public class User {
    private String userName;
    private String password;
    private String email;

    public User() {

    }

    public User(String userName, String password, String email) {
        this.userName = userName;
        this.password = password;
        this.email = email;
    }

    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public String getPassword() {
        return password;
    }
```

```java
    public void setPassword(String password) {
        this.password = password;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

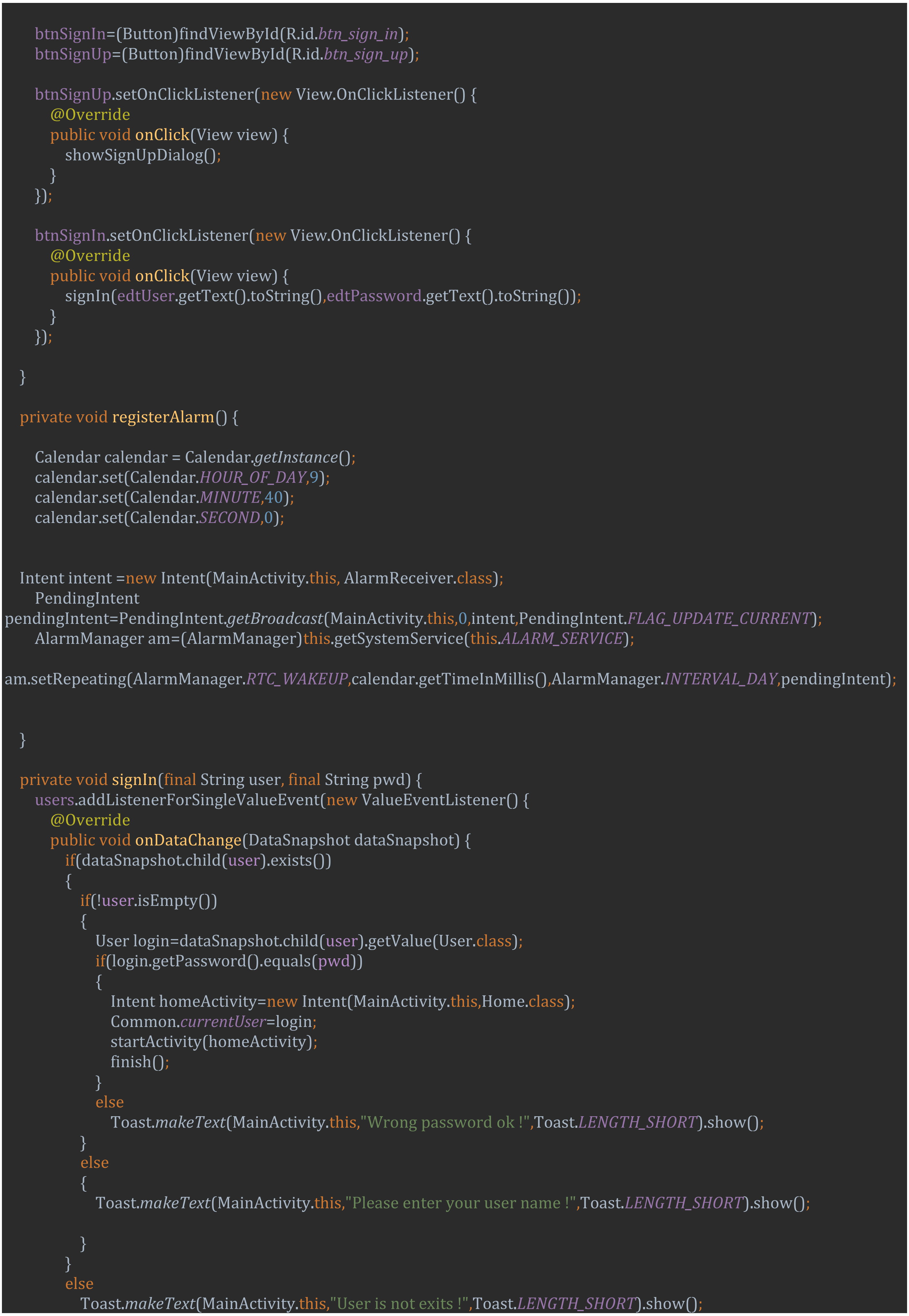
```java
package penguin.onlineq;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.Toast;

import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.squareup.picasso.Picasso;

import penguin.onlineq.Common.Common;
import penguin.onlineq.Interface.ItemClickListener;
import penguin.onlineq.Model.Category;
import penguin.onlineq.ViewHolder.CategoryViewHolder;


public class CategoryFragment extends Fragment {

    RecyclerView listCategory;
    RecyclerView.LayoutManager layoutManager;
    FirebaseRecyclerAdapter<Category,CategoryViewHolder> adapter;

    FirebaseDatabase database;
    DatabaseReference categories;

    public  static CategoryFragment newInstance(){
        CategoryFragment categoryFragment=new CategoryFragment();
        return  categoryFragment;
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        database=FirebaseDatabase.getInstance();
        categories=database.getReference("Category");


    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle
savedInstanceState) {

        View myfragment=inflater.inflate(R.layout.fragment_category,container,false);

        listCategory=(RecyclerView)myfragment.findViewById(R.id.listCategory);
        listCategory.setHasFixedSize(true);
        layoutManager=new LinearLayoutManager(container.getContext());
        listCategory.setLayoutManager(layoutManager);

        loadCategories();


        return myfragment;
    }

    private void loadCategories() {
        adapter=new FirebaseRecyclerAdapter<Category, CategoryViewHolder>(
                Category.class,
                R.layout.category_layout,
                CategoryViewHolder.class,
                categories
        ) {
            @Override
            protected void populateViewHolder(CategoryViewHolder viewHolder, final Category model, int position) {
                 viewHolder.category_name.setText(model.getName());
                Picasso.with(getActivity())
                        .load(model.getImage())
                        .into(viewHolder.category_image);

                viewHolder.setItemClickListener(new ItemClickListener() {
                    @Override
                    public void onClick(View view, int position, boolean isLongClick) {
                      // Toast.makeText(getActivity(),String.format("%d |
%s",adapter.getRef(position).getKey(),model.getName()),Toast.LENGTH_SHORT).show();
                        Intent startGame=new Intent(getActivity(),Start.class);
                        Common.categoryId=adapter.getRef(position).getKey();
                        Common.categoryName=model.getName();
                        startActivity(startGame);

                    }
                });
            }
        };
        adapter.notifyDataSetChanged();
        listCategory.setAdapter(adapter);

    }
}
```

```
package penguin.onlineq;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
```

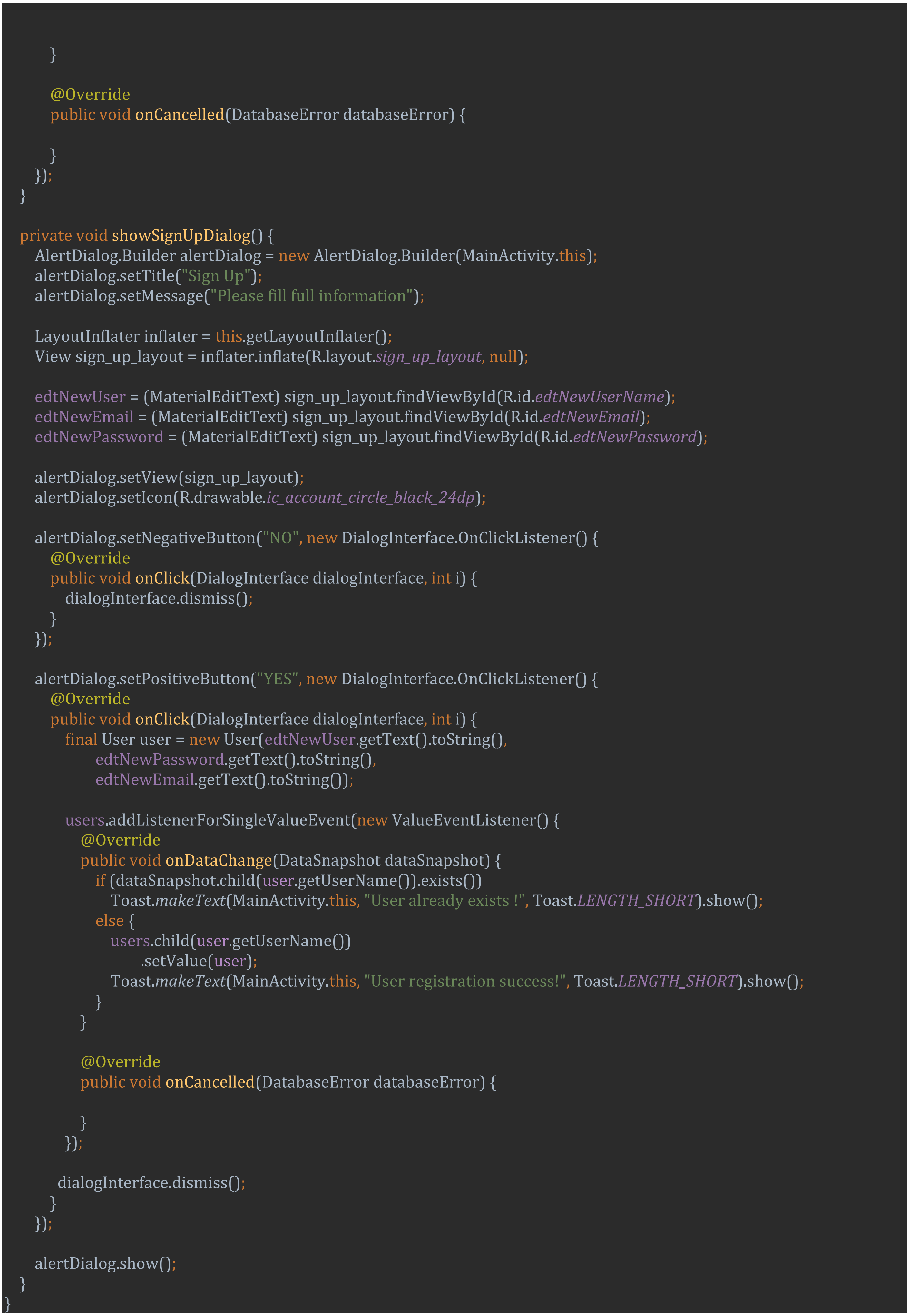
```java
import android.widget.TextView;

import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import penguin.onlineq.Common.Common;
import penguin.onlineq.Model.QuestionScore;

public class Done extends AppCompatActivity {

    Button btnTryAgain;
    TextView txtResultScore,getTxtResultQuestion;
    ProgressBar progressBar;


    FirebaseDatabase database;
    DatabaseReference question_score;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_done);
        database=FirebaseDatabase.getInstance();
        question_score=database.getReference("Question_Score");


        txtResultScore=(TextView)findViewById(R.id.txtTotalScore);
        getTxtResultQuestion=(TextView)findViewById(R.id.txtTotalQuestion);
        progressBar=(ProgressBar)findViewById(R.id.doneProgressBar);
        btnTryAgain=(Button)findViewById(R.id.btnTryAgain);


        btnTryAgain.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent =new Intent(Done.this,Home.class);
                startActivity(intent);
                finish();
            }
        });

        Bundle extra=getIntent().getExtras();
        if(extra!=null)
        {
            int score=extra.getInt("SCORE");
            int totalQuestion=extra.getInt("TOTAL");
            int correctAnswer=extra.getInt("CORRECT");
            txtResultScore.setText(String.format("SCORE : %d",score));
            getTxtResultQuestion.setText(String.format("PASSED : %d / %d",correctAnswer,totalQuestion));


            progressBar.setMax(totalQuestion);
            progressBar.setProgress(correctAnswer);


            //Upload point to DB

            question_score.child(String.format("%s _ %s", Common.currentUser.getUserName(),
                                          Common.categoryId))
                    .setValue(new QuestionScore(String.format("%s _ %s", Common.currentUser.getUserName(),
                            Common.categoryId),
                            Common.currentUser.getUserName(),
                            String.valueOf(score),
                            Common.categoryId,
                            Common.categoryName));
        }
    }
}
package penguin.onlineq;

public class Home extends AppCompatActivity {


    BottomNavigationView bottomNavigationView;

    BroadcastReceiver mRegistrationBroadcastReceiver;
    @Override
    protected void onPause() {
        LocalBroadcastManager.getInstance(this).unregisterReceiver(mRegistrationBroadcastReceiver);

        super.onPause();

    }
    @Override
    protected void onResume() {
        LocalBroadcastManager.getInstance(this).registerReceiver(mRegistrationBroadcastReceiver,new
IntentFilter("registrationComplete"));
        LocalBroadcastManager.getInstance(this).registerReceiver(mRegistrationBroadcastReceiver,new
IntentFilter(Common.STR_PUSH));

        super.onResume();

    }

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

        registrationNotification();
        bottomNavigationView=(BottomNavigationView)findViewById(R.id.navigation);

        bottomNavigationView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                Fragment selectedFragment=null;
                switch (item.getItemId())
                {
                    case R.id.action_category:
                        selectedFragment=CategoryFragment.newInstance();
                        break;
                    case R.id.action_ranking:
                        selectedFragment=RankingFragment.newInstance();
                        break;
                }

                FragmentTransaction transaction=getSupportFragmentManager().beginTransaction();
                transaction.replace(R.id.frame_layout,selectedFragment);
                transaction.commit();
                return true;
            }

        });

        setDefaultFragment();
    }

    private void registrationNotification() {
```

```
mRegistrationBroadcastReceiver =new BroadcastReceiver() {
  @Override
  public void onReceive(Context context, Intent intent) {
    if(intent.getAction().equals(Common.STR_PUSH)){
      String message=intent.getStringExtra("message");
      showNotification("Emre BALIKCI",message);

    }
  }
};


  }

  private void showNotification(String title, String message) {
    Intent intent =new Intent(getApplicationContext(),MainActivity.class);
    PendingIntent
contentIntent=PendingIntent.getActivity(getBaseContext(),0,intent,PendingIntent.FLAG_UPDATE_CURRENT);
    NotificationCompat.Builder builder=new NotificationCompat.Builder(getBaseContext());
    builder.setAutoCancel(true)
        .setDefaults(Notification.DEFAULT_ALL)
        .setWhen(System.currentTimeMillis())
        .setSmallIcon(R.mipmap.ic_launcher_round)
        .setContentTitle(title)
        .setContentText(message)
        .setContentIntent(contentIntent);

    NotificationManager
notificationManager=(NotificationManager)getBaseContext().getSystemService(Context.NOTIFICATION_SERVICE);
    notificationManager.notify(new Random().nextInt(),builder.build());

  }

  private void setDefaultFragment() {
    FragmentTransaction transaction=getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.frame_layout,CategoryFragment.newInstance());
    transaction.commit();

  }
}
```

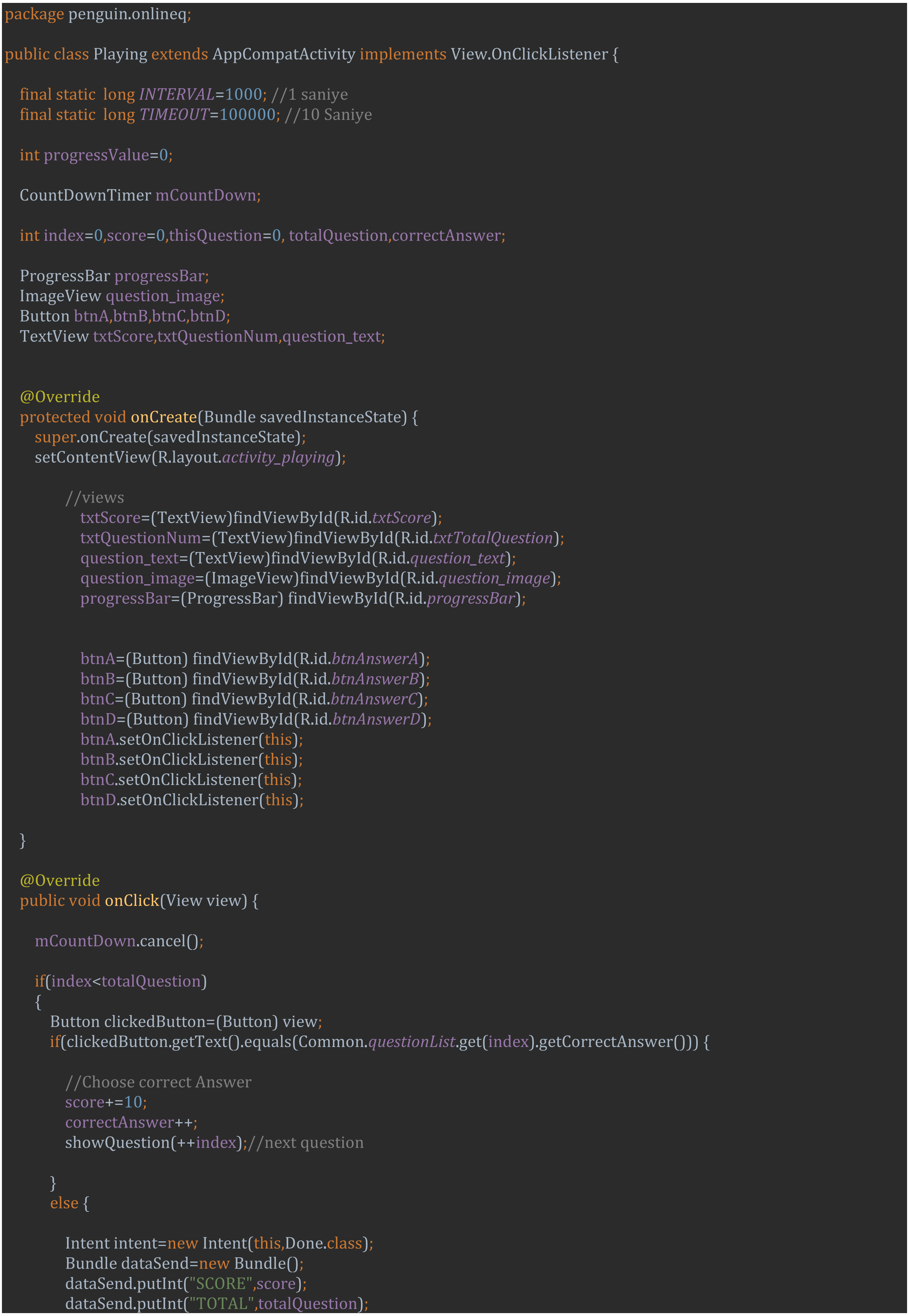
```
package penguin.onlineq;

public class MainActivity extends AppCompatActivity {

MaterialEditText edtNewUser,edtNewPassword,edtNewEmail;//for signup
MaterialEditText edtUser,edtPassword;//for Sign in

Button btnSignUp,btnSignIn;
  FirebaseDatabase database;
  DatabaseReference users;

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

    registerAlarm();

    database=FirebaseDatabase.getInstance();
    users=database.getReference("Users");

    edtUser=(MaterialEditText)findViewById(R.id.edtUser);
    edtPassword=(MaterialEditText)findViewById(R.id.edtPassword);
```



```
    btnSignIn=(Button)findViewById(R.id.btn_sign_in);
    btnSignUp=(Button)findViewById(R.id.btn_sign_up);

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

    btnSignIn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        signIn(edtUser.getText().toString(),edtPassword.getText().toString());
      }
    });

  }

  private void registerAlarm() {

    Calendar calendar = Calendar.getInstance();
    calendar.set(Calendar.HOUR_OF_DAY,9);
    calendar.set(Calendar.MINUTE,40);
    calendar.set(Calendar.SECOND,0);


  Intent intent =new Intent(MainActivity.this, AlarmReceiver.class);
    PendingIntent
pendingIntent=PendingIntent.getBroadcast(MainActivity.this,0,intent,PendingIntent.FLAG_UPDATE_CURRENT);
    AlarmManager am=(AlarmManager)this.getSystemService(this.ALARM_SERVICE);

am.setRepeating(AlarmManager.RTC_WAKEUP,calendar.getTimeInMillis(),AlarmManager.INTERVAL_DAY,pendingIntent);


  }

  private void signIn(final String user, final String pwd) {
    users.addListenerForSingleValueEvent(new ValueEventListener() {
      @Override
      public void onDataChange(DataSnapshot dataSnapshot) {
        if(dataSnapshot.child(user).exists())
        {
          if(!user.isEmpty())
          {
            User login=dataSnapshot.child(user).getValue(User.class);
            if(login.getPassword().equals(pwd))
            {
              Intent homeActivity=new Intent(MainActivity.this,Home.class);
              Common.currentUser=login;
              startActivity(homeActivity);
              finish();
            }
            else
              Toast.makeText(MainActivity.this,"Wrong password ok !",Toast.LENGTH_SHORT).show();
          }
          else
          {
            Toast.makeText(MainActivity.this,"Please enter your user name !",Toast.LENGTH_SHORT).show();

          }
        }
        else
          Toast.makeText(MainActivity.this,"User is not exits !",Toast.LENGTH_SHORT).show();
```



```
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {

            }
        });
    }

    private void showSignUpDialog() {
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(MainActivity.this);
        alertDialog.setTitle("Sign Up");
        alertDialog.setMessage("Please fill full information");

        LayoutInflater inflater = this.getLayoutInflater();
        View sign_up_layout = inflater.inflate(R.layout.sign_up_layout, null);

        edtNewUser = (MaterialEditText) sign_up_layout.findViewById(R.id.edtNewUserName);
        edtNewEmail = (MaterialEditText) sign_up_layout.findViewById(R.id.edtNewEmail);
        edtNewPassword = (MaterialEditText) sign_up_layout.findViewById(R.id.edtNewPassword);

        alertDialog.setView(sign_up_layout);
        alertDialog.setIcon(R.drawable.ic_account_circle_black_24dp);

        alertDialog.setNegativeButton("NO", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                dialogInterface.dismiss();
            }
        });

        alertDialog.setPositiveButton("YES", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                final User user = new User(edtNewUser.getText().toString(),
                        edtNewPassword.getText().toString(),
                        edtNewEmail.getText().toString());

                users.addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        if (dataSnapshot.child(user.getUserName()).exists())
                            Toast.makeText(MainActivity.this, "User already exists !", Toast.LENGTH_SHORT).show();
                        else {
                            users.child(user.getUserName())
                                    .setValue(user);
                            Toast.makeText(MainActivity.this, "User registration success!", Toast.LENGTH_SHORT).show();
                        }
                    }

                    @Override
                    public void onCancelled(DatabaseError databaseError) {

                    }
                });

              dialogInterface.dismiss();
            }
        });

        alertDialog.show();
    }
}
```



```
package penguin.onlineq;

public class Playing extends AppCompatActivity implements View.OnClickListener {

  final static  long INTERVAL=1000; //1 saniye
  final static  long TIMEOUT=100000; //10 Saniye

  int progressValue=0;

  CountDownTimer mCountDown;

  int index=0,score=0,thisQuestion=0, totalQuestion,correctAnswer;

  ProgressBar progressBar;
  ImageView question_image;
  Button btnA,btnB,btnC,btnD;
  TextView txtScore,txtQuestionNum,question_text;


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

        //views
          txtScore=(TextView)findViewById(R.id.txtScore);
          txtQuestionNum=(TextView)findViewById(R.id.txtTotalQuestion);
          question_text=(TextView)findViewById(R.id.question_text);
          question_image=(ImageView)findViewById(R.id.question_image);
          progressBar=(ProgressBar) findViewById(R.id.progressBar);


          btnA=(Button) findViewById(R.id.btnAnswerA);
          btnB=(Button) findViewById(R.id.btnAnswerB);
          btnC=(Button) findViewById(R.id.btnAnswerC);
          btnD=(Button) findViewById(R.id.btnAnswerD);
          btnA.setOnClickListener(this);
          btnB.setOnClickListener(this);
          btnC.setOnClickListener(this);
          btnD.setOnClickListener(this);

  }

  @Override
  public void onClick(View view) {

    mCountDown.cancel();

    if(index<totalQuestion)
    {
      Button clickedButton=(Button) view;
      if(clickedButton.getText().equals(Common.questionList.get(index).getCorrectAnswer())) {

        //Choose correct Answer
        score+=10;
        correctAnswer++;
        showQuestion(++index);//next question

      }
      else {

        Intent intent=new Intent(this,Done.class);
        Bundle dataSend=new Bundle();
        dataSend.putInt("SCORE",score);
        dataSend.putInt("TOTAL",totalQuestion);
      dataSend.putInt("CORRECT",correctAnswer);
      intent.putExtras(dataSend);
      startActivity(intent);
      finish();

    }

     txtScore.setText(String.format("%d ",score));
  }
}

private void showQuestion(int index) {

    if(index<totalQuestion)
    {
      thisQuestion++;
      txtQuestionNum.setText(String.format("%d / %d ",thisQuestion,totalQuestion));
      progressBar.setProgress(0);
      progressValue=0;

      if(Common.questionList.get(index).getIsImageQuestion().equals("true"))
      {
        //if  is image

        Picasso.with(getBaseContext())
            .load(Common.questionList.get(index).getQuestion())
            .into(question_image);
        question_image.setVisibility(View.VISIBLE);
        question_text.setVisibility(View.INVISIBLE);
      }

      else {
        question_text.setText(Common.questionList.get(index).getQuestion());
        //if is text we will set image to invisible
        question_image.setVisibility(View.INVISIBLE);
        question_text.setVisibility(View.VISIBLE);
      }
      btnA.setText(Common.questionList.get(index).getAnswerA());
      btnB.setText(Common.questionList.get(index).getAnswerB());
      btnC.setText(Common.questionList.get(index).getAnswerC());
      btnD.setText(Common.questionList.get(index).getAnswerD());

      mCountDown.start();//Start timmer
    }

    else
    {
      //if it is final question
      Intent intent=new Intent(this,Done.class);
      Bundle dataSend=new Bundle();
      dataSend.putInt("SCORE",score);
      dataSend.putInt("TOTAL",totalQuestion);
      dataSend.putInt("CORRECT",correctAnswer);
      intent.putExtras(dataSend);
      startActivity(intent);
      finish();

    }

}
```

```
    @Override
    protected void onResume() {
        super.onResume();

        totalQuestion=Common.questionList.size();

        mCountDown=new CountDownTimer(TIMEOUT,INTERVAL) {
            @Override
            public void onTick(long minisec) {
                progressBar.setProgress(progressValue);
                progressValue++;
            }

            @Override
            public void onFinish() {
                mCountDown.cancel();
                showQuestion(++index);
            }
        };
        showQuestion(index);
    }
}
```

```
package penguin.onlineq;

public class RankingFragment extends Fragment {

    View myfragment;

    RecyclerView rankingList;
    LinearLayoutManager layoutManager;
    FirebaseRecyclerAdapter<Ranking,RankingViewHolder> adapter;


    FirebaseDatabase database;
    DatabaseReference questionScore,rankingTb1;
    int sum=0;

    public  static RankingFragment newInstance(){
        RankingFragment rankingFragment=new RankingFragment();
        return  rankingFragment;
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        database= FirebaseDatabase.getInstance();
        questionScore=database.getReference("Question_Score");
        rankingTb1=database.getReference("Ranking");
    }

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle
savedInstanceState) {

        myfragment=inflater.inflate(R.layout.fragment_ranking,container,false);

        //init View
        rankingList=(RecyclerView)myfragment.findViewById(R.id.rankingList);
        layoutManager=new LinearLayoutManager(getActivity());
       rankingList.setHasFixedSize(true);
       //cunku firebase in orderbychild methodu asekron sıralama yapıcak
        //recycler data tersine ihtiyacım var
    layoutManager.setReverseLayout(true);
    layoutManager.setStackFromEnd(true);
    rankingList.setLayoutManager(layoutManager);

     updateScore(Common.currentUser.getUserName(), new RankingCallBack<Ranking>() {
       @Override
       public void callBack(Ranking ranking) {
         rankingTb1.child(ranking.getUserName()).setValue(ranking);
        // showRanking();
       }

     });
     adapter=new FirebaseRecyclerAdapter<Ranking, RankingViewHolder>(

         Ranking.class,
         R.layout.layout_ranking,
         RankingViewHolder.class,
         rankingTb1.orderByChild("score")
     ) {

       @Override
       protected void populateViewHolder(RankingViewHolder viewHolder, final Ranking model, int position) {
        viewHolder.txt_name.setText(model.getUserName());
        viewHolder.txt_score.setText(String.valueOf(model.getScore()));

        viewHolder.setItemClickListener(new ItemClickListener() {
          @Override
          public void onClick(View view, int position, boolean isLongClick) {
               Intent scoreDetail=new Intent(getActivity(),ScoreDetail.class);
               scoreDetail.putExtra("viewUser",model.getUserName());
               startActivity(scoreDetail);

          }
        });
       }
     };
     adapter.notifyDataSetChanged();
     rankingList.setAdapter(adapter);

    return myfragment;
  }

  private void updateScore(final String userName, final RankingCallBack<Ranking> callback) {
    questionScore.orderByChild("user").equalTo(userName).addListenerForSingleValueEvent(new ValueEventListener()
{
      @Override
      public void onDataChange(DataSnapshot dataSnapshot) {
        for(DataSnapshot data:dataSnapshot.getChildren())
        {
          QuestionScore ques= data.getValue(QuestionScore.class);
          sum +=Integer.parseInt(ques.getScore());
          //tum scoreları topladık.toplama islermi buraya gelicek.Firebase asekron veritabanı

        }
        Ranking ranking=new Ranking(userName,sum);
        callback.callBack(ranking);
      }

      @Override
      public void onCancelled(DatabaseError databaseError) {
      }
    });
  }
}
```

```
package penguin.onlineq;

public class ScoreDetail extends AppCompatActivity {

  FirebaseDatabase database;
  DatabaseReference question_score;
  RecyclerView scoreList;
  RecyclerView.LayoutManager layoutManager;
  FirebaseRecyclerAdapter<QuestionScore,ScoreDetailViewHolder> adapter;


  String viewUser="";

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

    //firebase
    database=FirebaseDatabase.getInstance();
    question_score=database.getReference("Question_Score");

    //view
    scoreList=(RecyclerView)findViewById(R.id.scoreList);
    scoreList.setHasFixedSize(true);
    layoutManager=new LinearLayoutManager(this);
    scoreList.setLayoutManager(layoutManager);

    if(getIntent() !=null)
      viewUser=getIntent().getStringExtra("viewUser");
    if(!viewUser.isEmpty()) {
      loadScoreDetail(viewUser);
    }
  }
  private void loadScoreDetail(String viewUser) {

  adapter=new FirebaseRecyclerAdapter<QuestionScore, ScoreDetailViewHolder>(
        QuestionScore.class,
        R.layout.score_detail_layout,
        ScoreDetailViewHolder.class,
        question_score.orderByChild("user").equalTo(viewUser)
  ) {
    @Override
    protected void populateViewHolder(ScoreDetailViewHolder viewHolder, QuestionScore model, int position) {

      viewHolder.txt_name.setText(model.getCategoryName());
      viewHolder.txt_score.setText(model.getScore());

    }
   };
  adapter.notifyDataSetChanged();
  scoreList.setAdapter(adapter);
  }
}
```

```java
public class Start extends AppCompatActivity {


    Button btnPlay;

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

    database=FirebaseDatabase.getInstance();
    questions=database.getReference("Questions");
    loadQuestions(Common.categoryId);

    btnPlay=(Button)findViewById(R.id.btnPlay);
    btnPlay.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
        Intent  intent= new Intent(Start.this,Playing.class);
        startActivity(intent);
        finish();
      }
    });

  }

  private void loadQuestions(String categoryId) {

      //First,clear List if have old question
      if(Common.questionList.size()>0)
        Common.questionList.clear();

    questions.orderByChild("CategoryId").equalTo(categoryId).addValueEventListener(new ValueEventListener() {
      @Override
      public void onDataChange(DataSnapshot dataSnapshot) {
          for (DataSnapshot postSnapshot:dataSnapshot.getChildren()){
            Question ques=postSnapshot.getValue(Question.class);
            Common.questionList.add(ques);
          }
      }

      @Override
      public void onCancelled(DatabaseError databaseError) {

      }
    });


    //Random list
    Collections.shuffle(Common.questionList);

  }
}
```