CHAPTER 1

INTRODUCTION

1.1 Introduction to Android Application

It is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance and commercially sponsored by Google. It is free and open-source software; its source code is known as Android Open-Source Project (AOSP), which is primarily licensed under the Apache License. As the world's most popular mobile platform, Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It has the largest installed base of any mobile platform and is still growing fast. Every day another million users power up their Android-powered devices for the first time and start looking for apps, games, and other digital content. Android provides a touchscreen user interface (UI) for interacting with apps. Android's UI is mainly based on direct manipulation. People use touch gestures such as swiping, tapping, and pinching to manipulate on-screen objects. In addition to the keyboard, there's a customizable on-screen keyboard for text input. Android can also support game controllers and full-size physical keyboards connected by Bluetooth or USB.

Some general features of android applications are:

- SMS and MMS are available forms of messaging, including threaded text messaging

- Voice-based features like Google search through voice has been available since initial release.

- Android has native support for multi-touch

- Android supports multiple languages.

- It provides support for virtual reality or 2D/3D graphics.

1.2 Introduction to Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

1.3 Mini-Project Description

Making Quiz app is an android-based application, and enables the user to undertake a series of questions on java language. The app is user friendly, and the user shall find it extremely easy to answer the multiple-choice questions. At the end of the quiz, a result report is generated which states the score. The app also presents an option to the current user to play the question round again or quit in between. It is effortlessly powerful & traversable can be troublesome test. The essential requirement may be which we might be making an android application for the portable stage. The real limitation can be the goals & can be restricted manage estimate as the android application is for the versatile environment. The other imperative regards to the versatile can be process power & restricted memory.

1.4 REQUIREMENT SPECIFICATION

1.4.1 HARDWARE REQUIREMENTS:

i) Processor Name: Ryzen 5 Hexa Core

ii) RAM: 8 GB

iii) Processor variant: 5500U

1.4.2 SOFTWARE REQUREMENTS:

i) Operating System: Windows 10

ii) Tool used: Android Studio 4.1

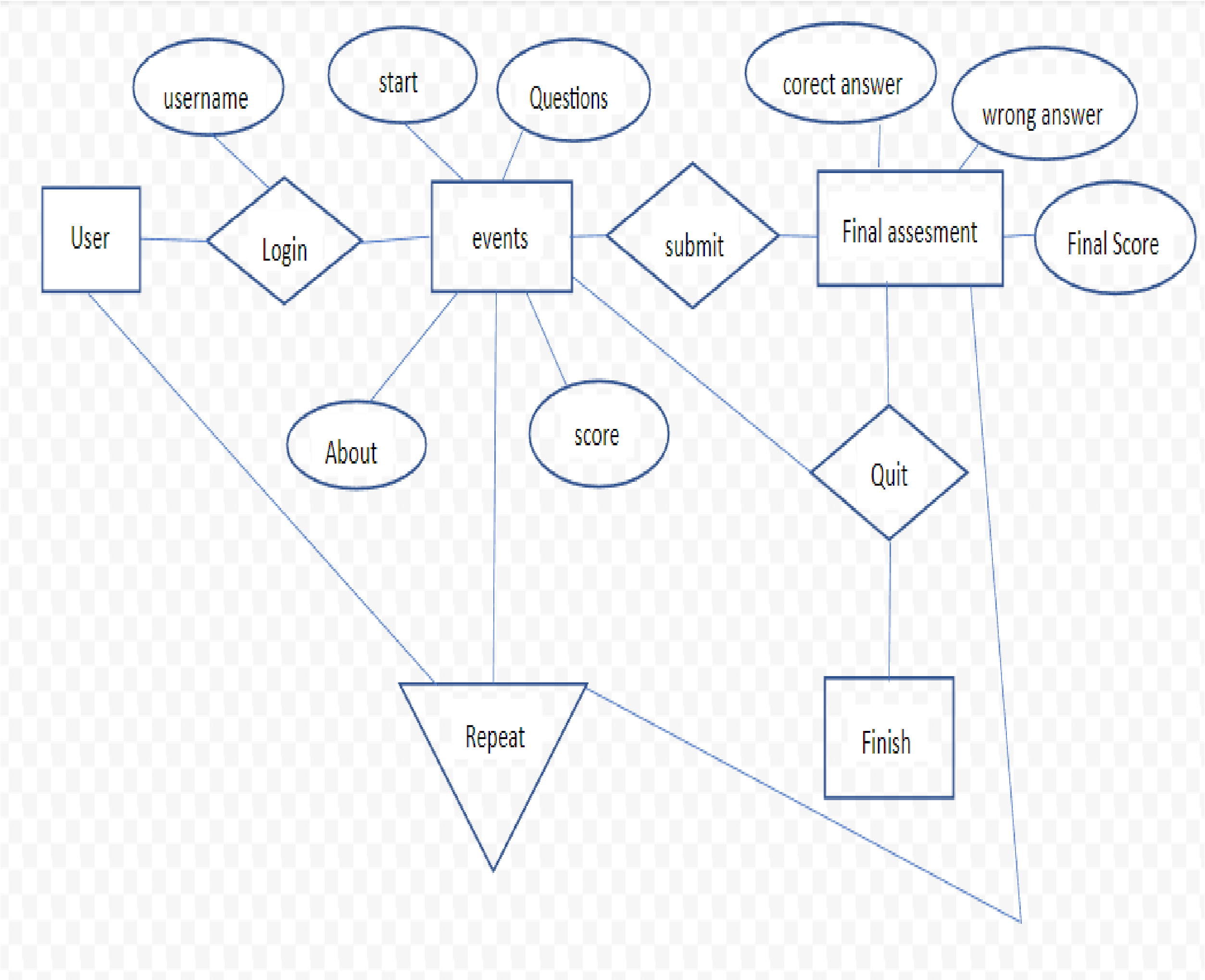
iii) IDE: Android Studio on JetBrains' IntelliJ IDEA

iv) Emulator: Pixel 4 XL API 30

CHAPTER 2

DESIGN

2.1 Architectural Diagram of the Application



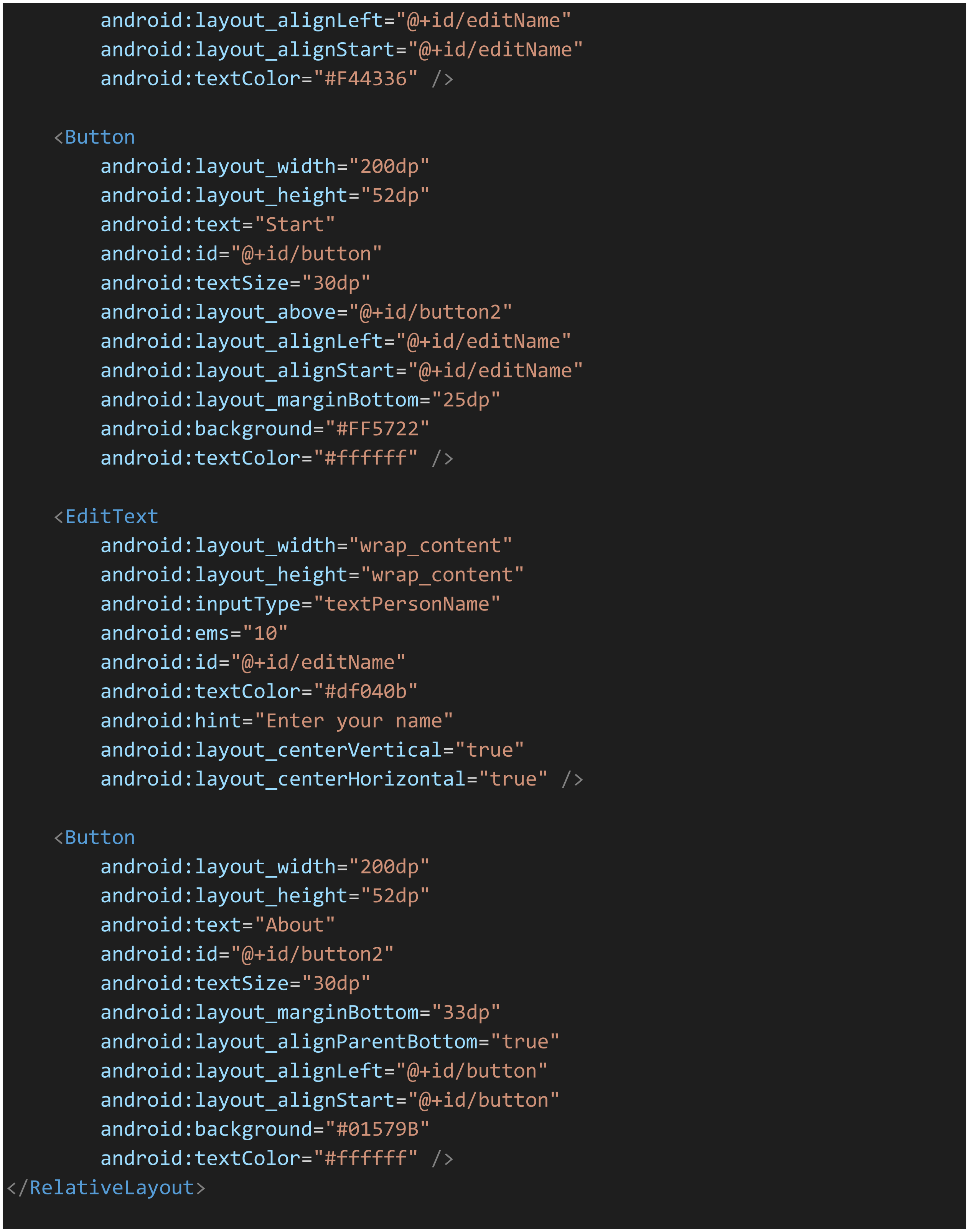
Explanation:

- User has to open the app and once the application is open, it shows the main page which contains the quiz details.
- In the about section, you will get the information of the developers.
- Next in the main page the user should enter his/her name and click on the start button.
- The quiz questions are displayed on the screen with the option of next button and quit button.
- The application allows to quit in between the quiz.
- Once the user selects the answer it shows whether it is correct or wrong with the score displayed.
- Finally, once the whole quiz is done it displays the correct answers, wrong answers and the final score.
- To get back user can click on the restart button.

2.2 XML Code

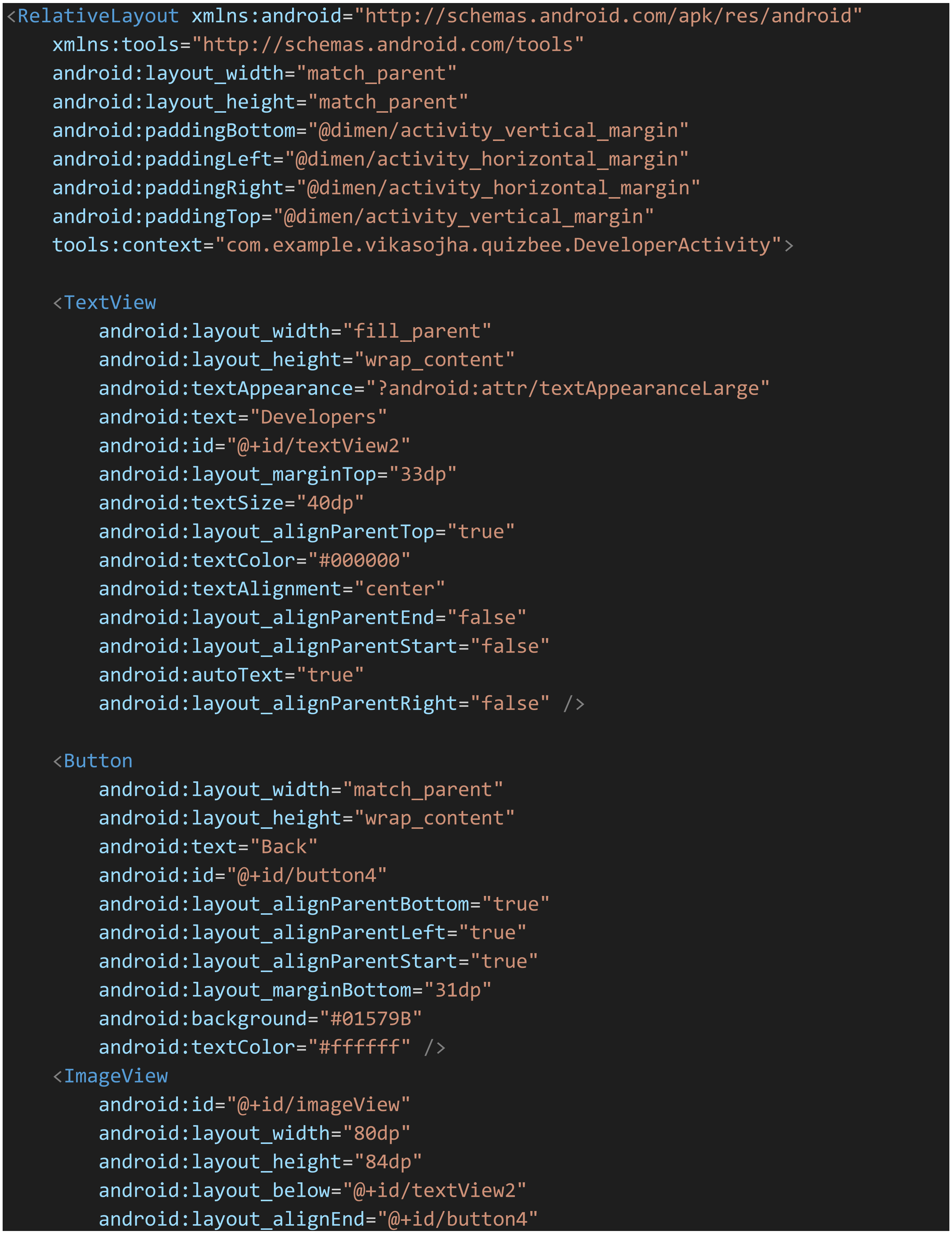
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.vikasojha.quizbee.MainActivity"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/quizapp"
        android:layout_marginTop="48dp"
        android:textSize="60dp"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
```

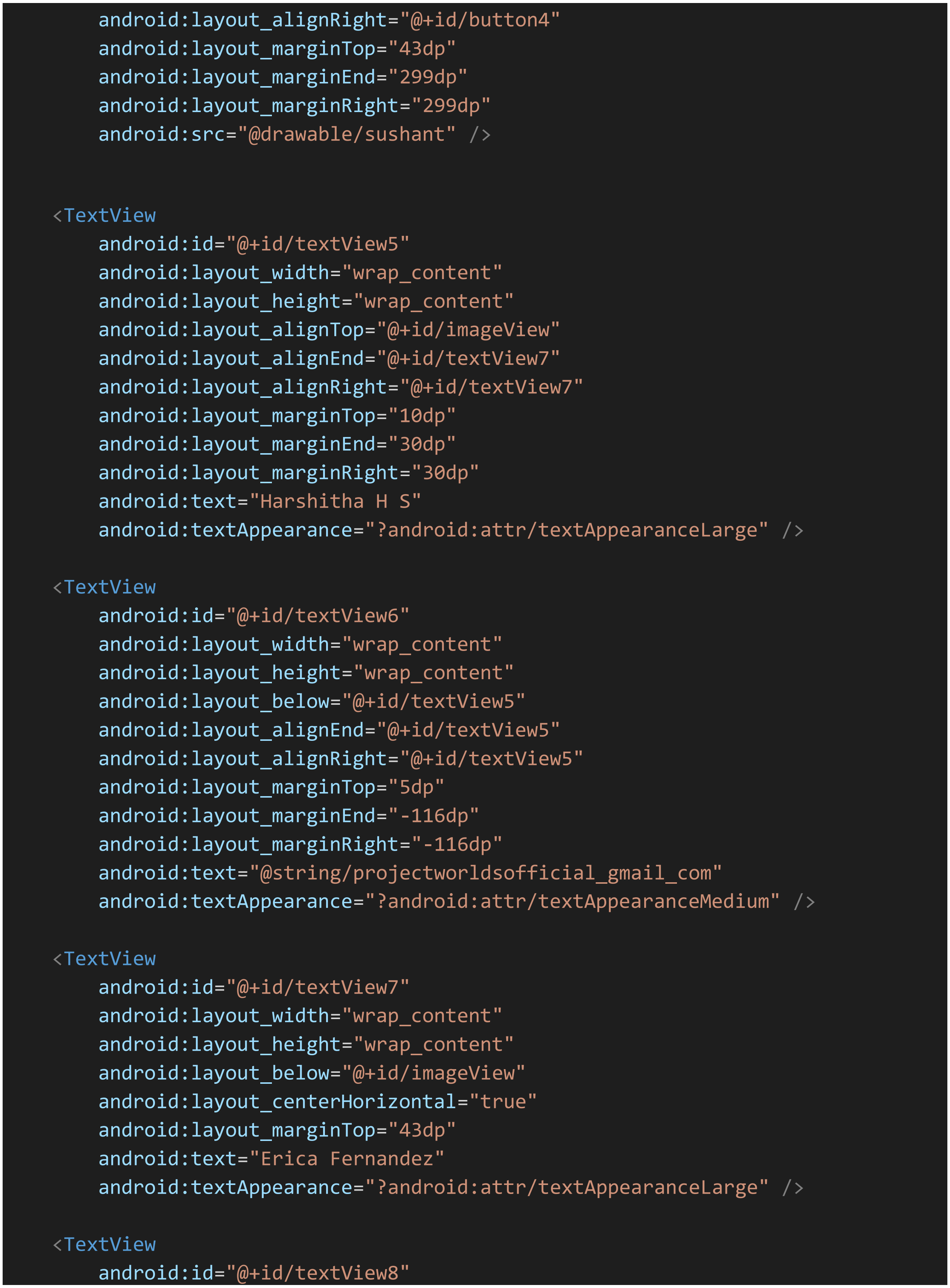


```
        android:layout_alignLeft="@+id/editName"
        android:layout_alignStart="@+id/editName"
        android:textColor="#F44336" />

    <Button
        android:layout_width="200dp"
        android:layout_height="52dp"
        android:text="Start"
        android:id="@+id/button"
        android:textSize="30dp"
        android:layout_above="@+id/button2"
        android:layout_alignLeft="@+id/editName"
        android:layout_alignStart="@+id/editName"
        android:layout_marginBottom="25dp"
        android:background="#FF5722"
        android:textColor="#ffffff" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:id="@+id/editName"
        android:textColor="#df040b"
        android:hint="Enter your name"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

    <Button
        android:layout_width="200dp"
        android:layout_height="52dp"
        android:text="About"
        android:id="@+id/button2"
        android:textSize="30dp"
        android:layout_marginBottom="33dp"
        android:layout_alignParentBottom="true"
        android:layout_alignLeft="@+id/button"
        android:layout_alignStart="@+id/button"
        android:background="#01579B"
        android:textColor="#ffffff" />
</RelativeLayout>
```

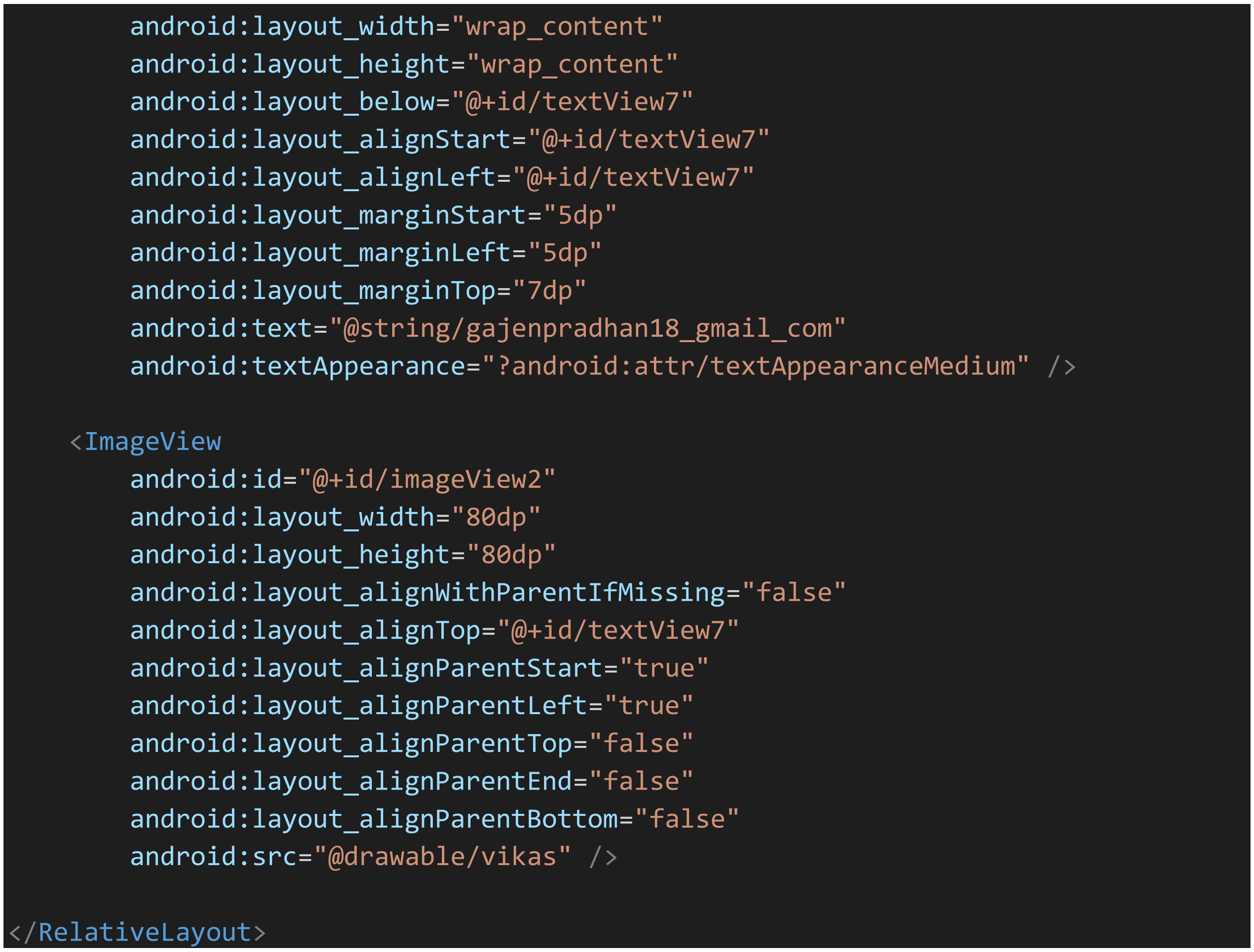
Activitiy_developer.xml



```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.vikasojha.quizbee.DeveloperActivity">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Developers"
        android:id="@+id/textView2"
        android:layout_marginTop="33dp"
        android:textSize="40dp"
        android:layout_alignParentTop="true"
        android:textColor="#000000"
        android:textAlignment="center"
        android:layout_alignParentEnd="false"
        android:layout_alignParentStart="false"
        android:autoText="true"
        android:layout_alignParentRight="false" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Back"
        android:id="@+id/button4"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginBottom="31dp"
        android:background="#01579B"
        android:textColor="#ffffff" />
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="80dp"
        android:layout_height="84dp"
        android:layout_below="@+id/textView2"
        android:layout_alignEnd="@+id/button4"
```



```
        android:layout_alignRight="@+id/button4"
        android:layout_marginTop="43dp"
        android:layout_marginEnd="299dp"
        android:layout_marginRight="299dp"
        android:src="@drawable/sushant" />


    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/imageView"
        android:layout_alignEnd="@+id/textView7"
        android:layout_alignRight="@+id/textView7"
        android:layout_marginTop="10dp"
        android:layout_marginEnd="30dp"
        android:layout_marginRight="30dp"
        android:text="Harshitha H S"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView5"
        android:layout_alignEnd="@+id/textView5"
        android:layout_alignRight="@+id/textView5"
        android:layout_marginTop="5dp"
        android:layout_marginEnd="-116dp"
        android:layout_marginRight="-116dp"
        android:text="@string/projectworldsofficial_gmail_com"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/textView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/imageView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="43dp"
        android:text="Erica Fernandez"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:id="@+id/textView8"
```



```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView7"
        android:layout_alignStart="@+id/textView7"
        android:layout_alignLeft="@+id/textView7"
        android:layout_marginStart="5dp"
        android:layout_marginLeft="5dp"
        android:layout_marginTop="7dp"
        android:text="@string/gajenpradhan18_gmail_com"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:layout_alignWithParentIfMissing="false"
        android:layout_alignTop="@+id/textView7"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="false"
        android:layout_alignParentEnd="false"
        android:layout_alignParentBottom="false"
        android:src="@drawable/vikas" />

</RelativeLayout>
```

Activity_Result

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.vikasojha.quizbee.ResultActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="28dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Large Text"
        android:id="@+id/tvres"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="70dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentStart="true"
        android:background="#F44336"
        android:textColor="#000000"
        android:textAlignment="center"
        android:textSize="19dp" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Restart"
        android:id="@+id/btnRestart"
        android:layout_alignParentBottom="true"
        android:layout_marginBottom="119dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:background="#01579B"
        android:textColor="#ffffff" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="28dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Large Text"
        android:id="@+id/tvres2"
        android:background="#76FF03"
        android:layout_below="@+id/tvres"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="31dp"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"
        android:textColor="#000000"
        android:textAlignment="center"
        android:textSize="19dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="28dp"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Large Text"
        android:id="@+id/tvres3"
        android:background="#FFEB3B"
        android:layout_below="@+id/tvres2"
        android:layout_marginTop="33dp"
        android:layout_alignRight="@+id/tvres2"
        android:layout_alignEnd="@+id/tvres2"
        android:layout_alignParentEnd="true"
        android:layout_alignParentStart="true"
        android:textColor="#000000"
        android:textAlignment="center"
        android:textSize="19dp" />
</RelativeLayout>
```

CHAPTER 3

IMPLEMENTATION

3.1 List of classes imported

- **AppCompatActivity:** Base class for activities that wish to use some of the newer platform features on older Android devices Some of these backported features include: Using the action bar, including action items, navigation modes and more with the setSupportActionBar(Toolbar) API.
- **Intent:** Intent is the message that is passed between components such as activities, content providers, broadcast receivers, services etc. It is generally used with the start Activity() method to invoke activity, broadcast receivers etc.
- **Bundle:** Bundles are generally used for passing data from one activity to another. Basically, here the concept of key-value pair is used where the data that one wants to pass is the value of the map, which can be later retrieved by using the key.
- **Handler:** It will deliver messages and runnable to that Looper's message queue and execute them on that Looper's thread.
- **Log:** Mock Log implementation for testing on non-android host.
- **View:** Java Class that inherits directly from the Java Object class, the root of the Java class hierarchy. Views are the basic building block for all Android GUI elements, and serve as the base class for all GUI widgets.
- **TextView:** A user interface element that displays text to the user.
- **Button:** Button is a subclass of TextView class and Compound Button is the subclass of Button class. There are different types of buttons in android such as Radio Button, Toggle Button, Compound Button etc.
- **Toast:** A toast is a view containing a quick little message for the user. The toast class helps you create and show those.
- **EditText:** A EditText is an overlay over TextView that configures itself to be editable. It is the predefined subclass of TextView that includes rich editing capabilities.
- **AuthFailureError:** Error indicating that there was an authentication failure when performing a Request.

- **Request:** Base class for all network requests.
- **Response:** Encapsulates a parsed response for delivery.
- **Volley Error:** Exception style class encapsulating Volley errors.
- **String Request:** A canned request for retrieving the response body at a given URL as a String.
- **Menu Item:** Interface for direct access to a previously created menu item.
- **Gson:** This is the main class for using Gson. Gson is typically used by first constructing a Gson instance and then invoking to Json (Object) or from Json (String, Class) methods on it. Gson instances are Thread-safe so you can reuse them freely across multiple threads.
- **TypeToken:** Forces clients to create a subclass of this class which enables retrieval the type information even at runtime.
- **Type:** Type is the common super interface for all types in the Java programming language. These include raw types, parameterized types, array types, type variables and primitive types.
- **ArrayList:** Implements all optional list operations, and permits all elements, including null.
- **HashMap:** Hash table-based implementation of the Map interface. This implementation provides all of the optional map operations, and permits null values and the null key.
- **Map:** This interface takes the place of the Dictionary class, which was a totally abstract class rather than an interface.
- **PreferenceManager:** Used to help create Preference hierarchies from activities or XML. In most cases, clients should use.
- **List:** An ordered collection (also known as a sequence).
- **ContentUris:** Utility methods useful for working with Uri objects that use the "content" (content://) scheme.
- **Cursor:** This interface provides random read-write access to the result set returned by a database query.
- **Uri:** Immutable URI reference. A URI reference includes a URI and a fragment, the component of the URI following a '#'.
- **Build:** Information about the current build, extracted from system properties.

- **Envirionment:** Provides access to environment variables.
- **DocumentsContract:** Defines the contract between a documents provider and the platform.
- **OnClickListener:** Interface definition for a callback to be invoked when a view is clicked.
- **PermissionChecker:** This class provides permission check APIs that verify both the permission and the associated app op for this permission if such is defined.
- **AndroidNetworking:** AndroidNetworking entry point. You must initialize this class before use. The simplest way is to just do {#code AndroidNetworking.initialize(context)}.
- **Priority:** Priority levels recognized by the request server.
- **UriSyntaxException:** Checked exception thrown to indicate that a string could not be parsed as a URI reference.
- **JSONException:** Thrown to indicate a problem with the JSON API.
- **JSONObject:** A modifiable set of name/value mappings. Names are unique, non-null strings.
- **RecyclerView:** A flexible view for providing a limited window into a large data set.
- **LayoutInflater:** Instantiates a layout XML file into its corresponding View objects.
- **ImageView:** Displays image resources, for example Bitmap or Drawable resources.
- **ViewGroup:** The view group is the base class for layouts and views containers.
- **RequestView:** A request dispatch queue with a thread pool of dispatchers.

3.2 List of important library functions

- **findViewById():** Finds a view that was identified by the id attribute from the XML that was processed in OnCreate(Bundle).
- **getText():** To get the text entered in the EditText views.
- **toString():** It's a method in java's Object class, which is the superclass of every java object. It is meant for returning textual representation of an object.
- **insertData():** It's a method in java's Object class, which is the superclass of every java object. It is meant for returning textual representation of an object.
- **makeText():** The text that should appear to the user. The duration that the toast should remain on the screen.
- **getName():** Method class is helpful to get the name of methods, as a String. To get the name of all methods of a class, get all the methods of that class object.
- **putExtras():** Adds extended data to the intent. It has two parameters, first one specifies the name of the extra data,and the second parameter is the data itself.
- **getInstance():** Provider class is used to return a Signature object that implements the specified signature algorithm.
- **setOnClickListener():** OnClickListener and wires the listener to the button using setOnClickListener . As a result, the system executes the code you write in onClick after the user presses the button.
- **setAdapter():** Instead, list view requests views on demand from a ListAdapter as needed, such as to display new views as the user scrolls up or down. In order to display items in the list, call setAdapter to associate an adapter with the list.
- **getIntent():** This method returns the intent that shared the activity.
- **isEmpty():** This method returns true if, and only if, length () is 0, false otherwise.
- **equals ():** This method compares the string to the specified object.
- **setEnabled():** This method sets the enabled state of the view, true if enabled, false otherwise.
- **show ():** This method shows the view for the specified duration.

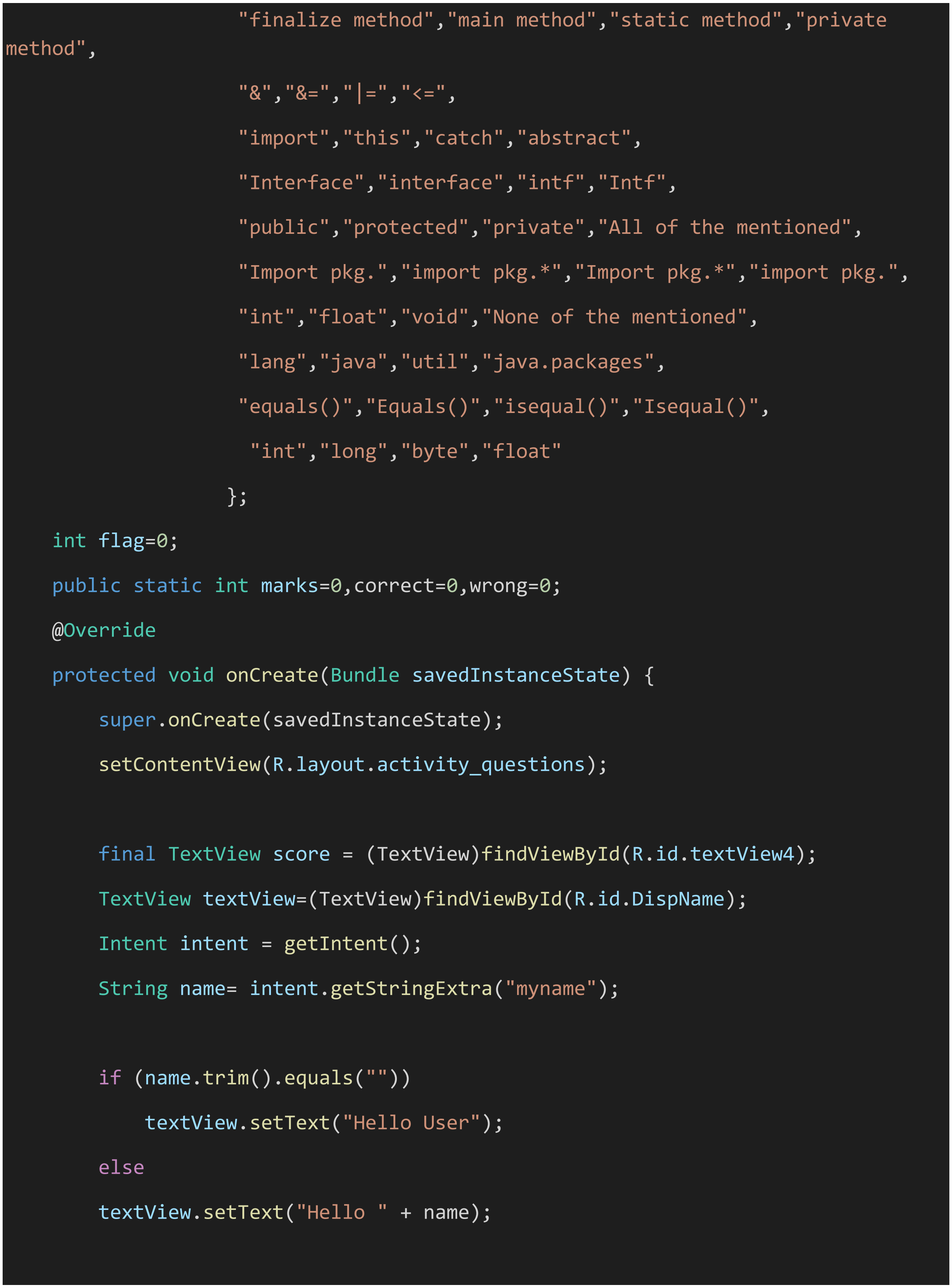
3.3 Java Code

MainActivity.java

```
package com.example.vikasojha.quizbee;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

        Button startbutton=(Button)findViewById(R.id.button);
        Button aboutbutton=(Button)findViewById(R.id.button2);
        final EditText nametext=(EditText)findViewById(R.id.editName);

                startbutton.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        String name=nametext.getText().toString();
                        Intent intent=new
Intent(getApplicationContext(),QuestionsActivity.class);
                        intent.putExtra("myname",name);
                        startActivity(intent);
                    }
                });

                aboutbutton.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                    Intent intent=new
Intent(getApplicationContext(),DeveloperActivity.class);
                    startActivity(intent);
            }
        });}}
```

QuestionsActivity.java

```
package com.example.vikasojha.quizbee

import …

public class QuestionsActivity extends AppCompatActivity {

    TextView tv;

    Button submitbutton, quitbutton;

    RadioGroup radio_g;

    RadioButton rb1,rb2,rb3,rb4;


    String questions[] = {

                            "Which method can be defined only once in a
program?",

                            "Which of these is not a bitwise operator?",

                            "Which keyword is used by method to refer to the
object that invoked it?",

                            "Which of these keywords is used to define interfaces
in Java?",

                            "Which of these access specifiers can be used for an
interface?",

                            "Which of the following is correct way of importing
an entire package ‘pkg’?",

                            "What is the return type of Constructors?",

                            "Which of the following package stores all the
standard java classes?",

                            "Which of these method of class String is used to
compare two String objects for their equality?",

                            "An expression involving byte, int, & literal numbers
is promoted to which of these?"

                         };

    String answers[] = {"main method","<=","this","interface","public","import
pkg.*","None of the mentioned","java","equals()","int"};

    String opt[] = {
```



```
                    "finalize method","main method","static method","private
method",
                    "&","&=","|=","<=",
                    "import","this","catch","abstract",
                    "Interface","interface","intf","Intf",
                    "public","protected","private","All of the mentioned",
                    "Import pkg.","import pkg.*","Import pkg.*","import pkg.",
                    "int","float","void","None of the mentioned",
                    "lang","java","util","java.packages",
                    "equals()","Equals()","isequal()","Isequal()",
                     "int","long","byte","float"
                   };
    int flag=0;
    public static int marks=0,correct=0,wrong=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_questions);

        final TextView score = (TextView)findViewById(R.id.textView4);
        TextView textView=(TextView)findViewById(R.id.DispName);
        Intent intent = getIntent();
        String name= intent.getStringExtra("myname");

        if (name.trim().equals(""))
            textView.setText("Hello User");
        else
        textView.setText("Hello " + name);
        submitbutton=(Button)findViewById(R.id.button3);

        quitbutton=(Button)findViewById(R.id.buttonquit);

        tv=(TextView) findViewById(R.id.tvque);


        radio_g=(RadioGroup)findViewById(R.id.answersgrp);

        rb1=(RadioButton)findViewById(R.id.radioButton);

        rb2=(RadioButton)findViewById(R.id.radioButton2);

        rb3=(RadioButton)findViewById(R.id.radioButton3);

        rb4=(RadioButton)findViewById(R.id.radioButton4);

        tv.setText(questions[flag]);

        rb1.setText(opt[0]);

        rb2.setText(opt[1]);

        rb3.setText(opt[2]);

        rb4.setText(opt[3]);

        submitbutton.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                //int color = mBackgroundColor.getColor();

                //mLayout.setBackgroundColor(color);


                if(radio_g.getCheckedRadioButtonId()==-1)

                {

                    Toast.makeText(getApplicationContext(), "Please select one
choice", Toast.LENGTH_SHORT).show();

                    return;

                }

                RadioButton uans = (RadioButton)
findViewById(radio_g.getCheckedRadioButtonId());

                String ansText = uans.getText().toString();
```

```
                    Intent in = new
Intent(getApplicationContext(),ResultActivity.class);
                    startActivity(in);
                }
                radio_g.clearCheck();
            }
        });

        quitbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new
Intent(getApplicationContext(),ResultActivity.class);
                startActivity(intent);
            }
        });
    }

}
import …
public class QuestionsActivity extends AppCompatActivity {
    TextView tv;
    Button submitbutton, quitbutton;
    RadioGroup radio_g;
    RadioButton rb1,rb2,rb3,rb4;

    String questions[] = {
                            "Which method can be defined only once in a
program?",
                            "Which of these is not a bitwise operator?",
                            "Which keyword is used by method to refer to the
object that invoked it?",
                            "Which of these keywords is used to define interfaces
in Java?",
                            "Which of these access specifiers can be used for an
interface?",
                            "Which of the following is correct way of importing
an entire package ‘pkg’?",
                            "What is the return type of Constructors?",
                            "Which of the following package stores all the
standard java classes?",
                            "Which of these method of class String is used to
compare two String objects for their equality?",
                            "An expression involving byte, int, & literal numbers
is promoted to which of these?"
                         };
                   };
    int flag=0;
    public static int marks=0,correct=0,wrong=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_questions);

        final TextView score = (TextView)findViewById(R.id.textView4);
        TextView textView=(TextView)findViewById(R.id.DispName);
        Intent intent = getIntent();
        String name= intent.getStringExtra("myname");

        if (name.trim().equals(""))
            textView.setText("Hello User");
        else
        textView.setText("Hello " + name);


        submitbutton=(Button)findViewById(R.id.button3);
        quitbutton=(Button)findViewById(R.id.buttonquit);
        tv=(TextView) findViewById(R.id.tvque);


        radio_g=(RadioGroup)findViewById(R.id.answersgrp);
        rb1=(RadioButton)findViewById(R.id.radioButton);
        rb2=(RadioButton)findViewById(R.id.radioButton2);
        rb3=(RadioButton)findViewById(R.id.radioButton3);
        rb4=(RadioButton)findViewById(R.id.radioButton4);
        tv.setText(questions[flag]);
        rb1.setText(opt[0]);
        rb2.setText(opt[1]);
        rb3.setText(opt[2]);
        rb4.setText(opt[3]);
        submitbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //int color = mBackgroundColor.getColor();
                //mLayout.setBackgroundColor(color);


                if(radio_g.getCheckedRadioButtonId()==-1)
                {
                    Toast.makeText(getApplicationContext(), "Please select one
choice", Toast.LENGTH_SHORT).show();
                    return;
                }
                RadioButton uans = (RadioButton)
findViewById(radio_g.getCheckedRadioButtonId());
```

```
                String ansText = uans.getText().toString();
//                Toast.makeText(getApplicationContext(), ansText,
Toast.LENGTH_SHORT).show();
                if(ansText.equals(answers[flag])) {
                    correct++;
                    Toast.makeText(getApplicationContext(), "Correct",
Toast.LENGTH_SHORT).show();
                }
                else {
                    wrong++;
                    Toast.makeText(getApplicationContext(), "Wrong",
Toast.LENGTH_SHORT).show();
                }

                flag++;

                if (score != null)
                    score.setText(""+correct);

                if(flag<questions.length)
                {
                    tv.setText(questions[flag]);
                    rb1.setText(opt[flag*4]);
                    rb2.setText(opt[flag*4 +1]);
                    rb3.setText(opt[flag*4 +2]);
                    rb4.setText(opt[flag*4 +3]);
                }
                else
                {
                    marks=correct;
                    Intent in = new 
Intent(getApplicationContext(),ResultActivity.class);

                    startActivity(in);

                }

                radio_g.clearCheck();

            }

        });


        quitbutton.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                Intent intent=new 
Intent(getApplicationContext(),ResultActivity.class);

                startActivity(intent);

            }

        });

    }



}
        public void onClick(View v) {
          Intent intent=new Intent(getApplicationContext(),ResultActivity.class);
          startActivity(intent);
        }
      });
    }

}
```

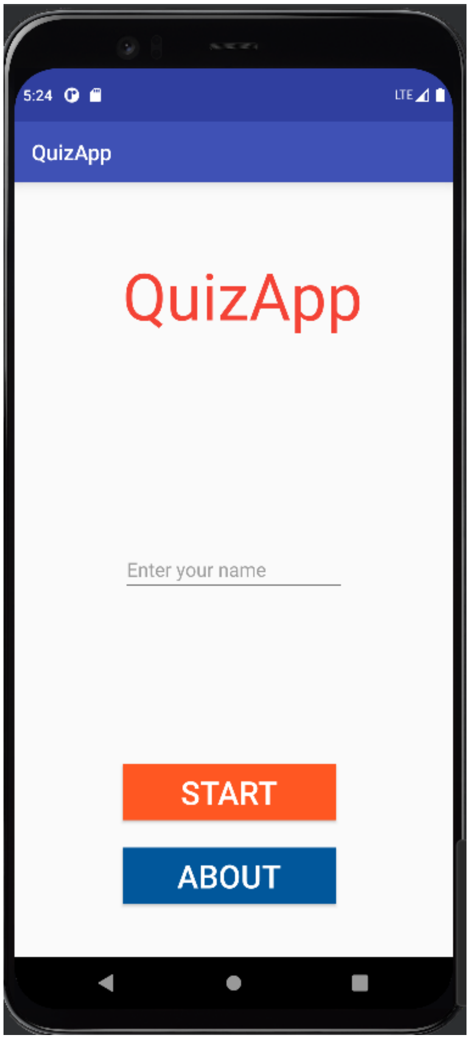
DeveloperActivity.java

```
package com.example.vikasojha.quizbee;

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

public class DeveloperActivity extends AppCompatActivity {
    Button btnRestart;

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

        btnRestart = (Button) findViewById(R.id.button4);


        btnRestart.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent in2 = new
Intent(getApplicationContext(),MainActivity.class);
                startActivity(in2);
            }
        });
    }
}
```

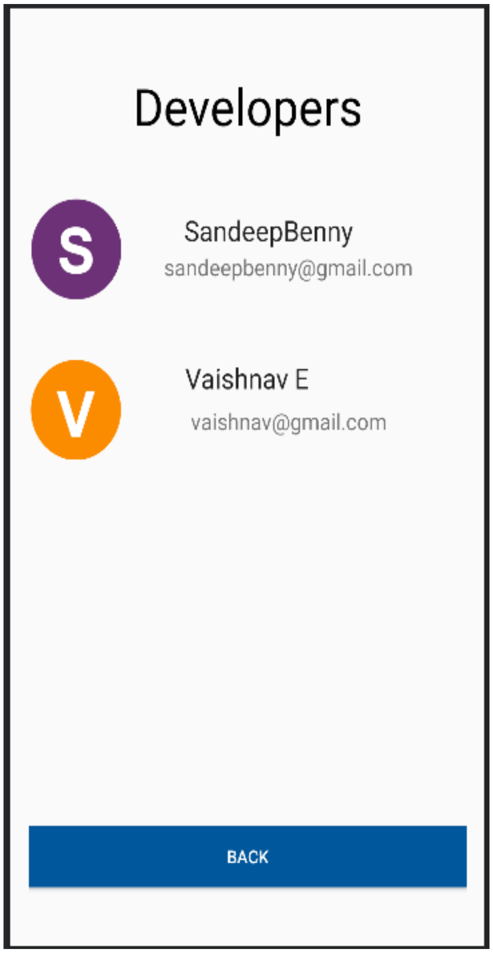
CHAPTER 4

RESULTS

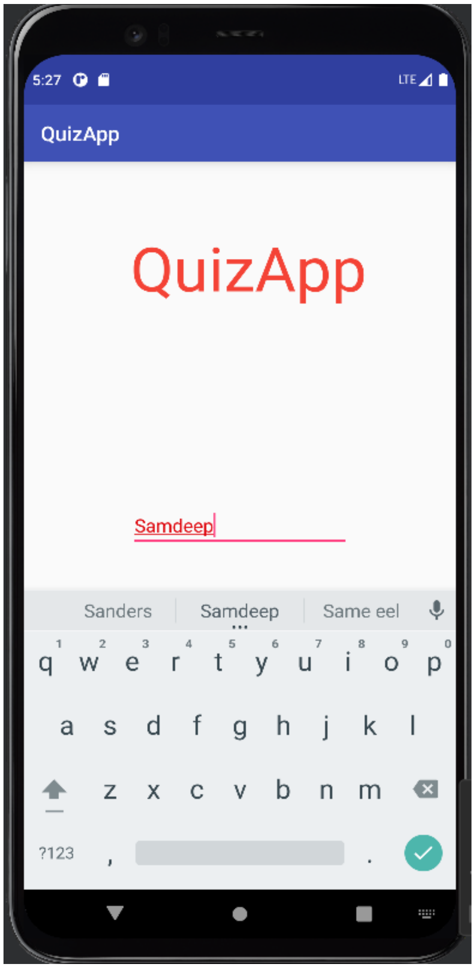
4.1 SCREENSHOTS



4.1 Main page of the app where one can access the app

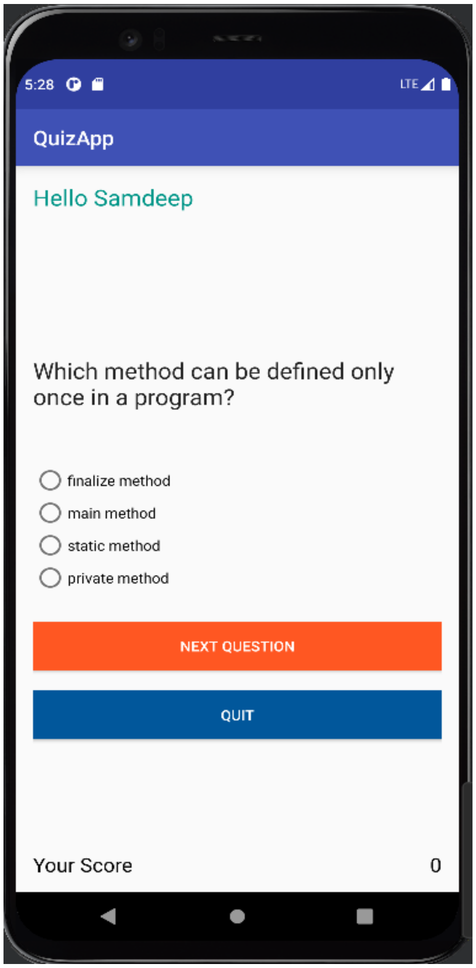


4.2 Developers page giving information about developers



Samdeep

4.3 Login page



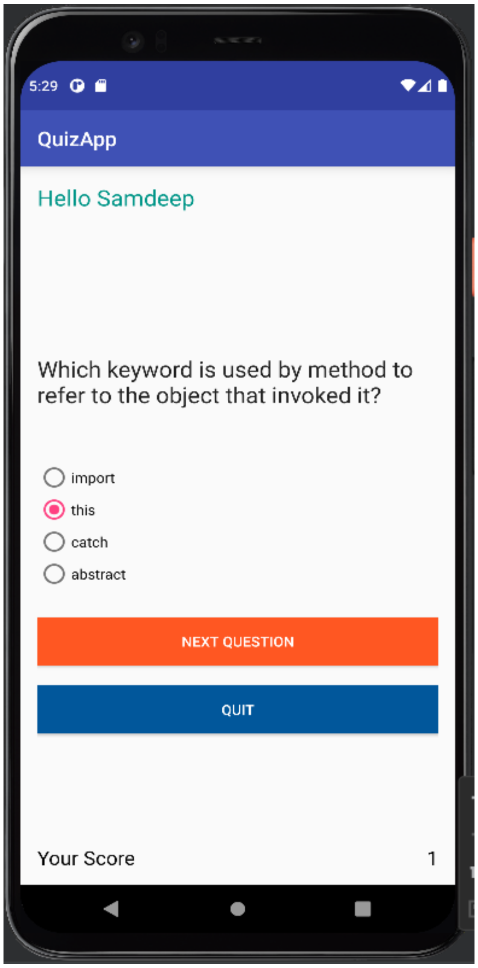
Hello Samdeep

Which method can be defined only once in a program?

○ finalize method
○ main method
○ static method
○ private method

Your Score 0

4.4 Displaying quiz questions

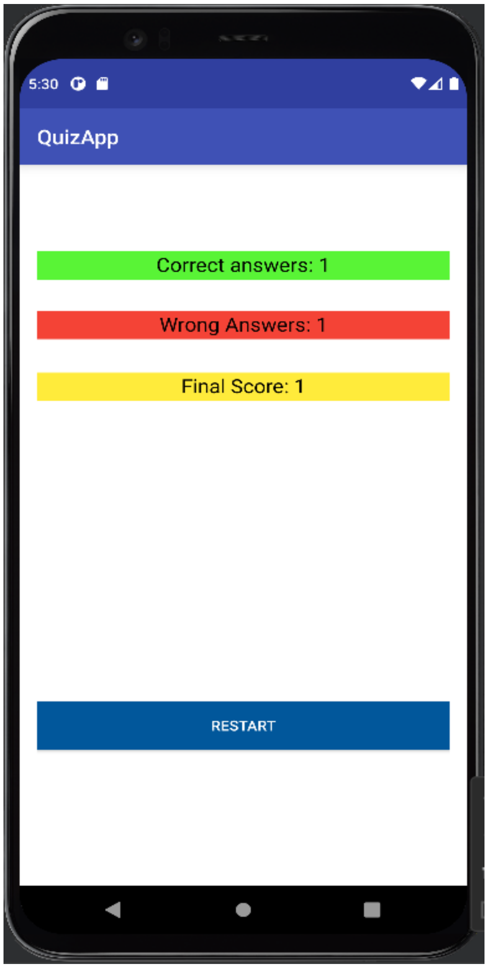


Hello Samdeep

Which keyword is used by method to refer to the object that invoked it?

○ import
● this
○ catch
○ abstract

Your Score 1



Correct answers: 1

Wrong Answers: 1

Final Score: 1

4.5 Screen showing score with the option selected wrong answer with the option

4.6 Final score board showing correct and a restart button

CHAPTER 5

CONCLUSION & FUTURE ENHANCEMENTS

5.1 CONCLUSION

The entire project has been developed as per the requirements stated by the user, it is found to be bug free as per the testing standards that is implemented. The package was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

Automation of the entire system improves the efficiency. It provides a friendly user interface which

proves to be better when compared to the existing system.

Using Android studio gives us more flexibility, but it requires more time. The quiz app is very important in current as well as future learning systems. It can be easily adopted by the educational institutions and universities, companies, etc. To make the quiz more flexible and safer.

The results are instantly shown to the students. Along with the wrong as well as correct answers. The marks also are displayed. The system used here gives appropriate access to the authorized users depending on their permissions.

5.2 FUTURE ENHANCEMENT

In the case of numerous students, this application aids to easy updating of information of all the students within the fast course of time. Here the system is not only efficient but also great at the security aspects. The data security and reliability of the system are the striking features. This also helps maintain the terms and privacy conditions.

The system works according to the modern way of usage through a computer, laptop, tablet or phone system. The System has adequate scope for modification in future if it is necessary.

REFERENCES

Bibliography and references:

- Google for problem solving
- www.androidstudio.com
- https://chrome.google.com
- Native mobile development Shaun Lewis & Mike Dunn
- Learn Java android development by Apress