

A ETE FINAL Project Report

on

ANDROID QUIZ TIME APPLICATION

*Submitted in partial fulfillment of the
requirement for the award of the degree of*

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE & ENGINEERING



Submitted to:

**Dr. ' Basetty Mallikarjuna
Associate Professor Submitted**

By:

S.No	Enrollment Number	Admission Number	Student Name	Degree / Branch	Sem
1	18021050117	18SCSE1050012	TARUN GOYAL	B.TECH/ CSE	7th
2	18021050133	18SCSE1050029	REHAN AKHTAR	B.TECH/ CSE	7th

School of Computing Science and Engineering

Galgotias University, Greater Noida

DEC 2021

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project entitled “Quiz Time Android Application” in partial fulfilment of the requirements for the award of the school of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of Aug 2021 to Dec 2021, under the supervision of “**Dr. A. Daniel Sir**” , Associate Professor, Department of Computer and Science Engineering/Computer Application and Information and Science, of school of Computing and Science Engineering, Galgotias University, Greater Noida. The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Tarun Goyal 18SCSE1050012

Rehan Akhtar 18SCSE1050029

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

DR. A. Daniel

Associate Professor

CERTIFICATE

The Final Project Viva-Voce examination of Tarun Goyal- 18SCSE1050012 and Rehan Akhtar 18SCSE1050029 has been held on 17-12-21 and his/her work is recommended for the award of Bachelor of Technology.

Signature of Examiner(s)

Signature of Supervisor(s)

Signature of Project Coordinator

Signature of Dean

Date: 17 December 2021

Place: Greater Noida

ABSTRACT

CONTENT OF ABSTRACT

In Now's a day, Everyone wants to learn online using smart android application. Having provide more content about education and a part of smart animation, frontend, backend and Google Firebase Console. User want a good app having good reviews, downloaders, and how to person get valuable things from app.

Mainly this Android Application "QUIZ TIME" is based on Android Educational App. All Data Comes from Google Firebase Console. Like a College Educational App, Many Subjects of Different Courses, with full Syllabus and chapters and in each chapter connected with small que around 8 to 10 questions. Firstly on Starting the app, user will go to sign up or sign in part. And according to that choice select a course or a subject. And learn from start to end with level wise. In each level having small quizzes then first attend then go to next level. If your score is above 75%, you got a small bonus keys to open a next level. All ques will come to firebase random chapter wise. After attend a quiz having a option of re-attempt. And also check a rank on leader board. First 20 user finish first course first get a special key to open a any course. So all process saved on google firebase then all data safe and secure.

SYSTEM REQUIREMENT OF THE QUIZ TIME:-

Now, this method is intended in such the way that it takes fewer resources to figure out work correctly. That is the minimum needs that we'd like to require care of:-

- ✓ The system wants a minimum of 4 GB of ram to run all the options.
- ✓ It wants a minimum 1.3 GHz processor to run smoothly.
- ✓ Rest is all up to the user's usage can take care of hardware.
- ✓ For security opposing anti-virus is suggested.
- ✓ The system is made correctly, and all the testing is done as per the requirements. So, the rest of the things depend on the user, and no one can harm the data or the software if the proper care is done.

TABLE OF CONTENTS

S.No	Particulars	Page No
1	Front Page	I
2	Acknowledgement	II
3	Abstract	III
4	Table of Contents	IV
5	List of Figures	V
6	Chapter 1: Introduction about Project	1-1
7	Literature Reviews / Comparative study	2-2
8	Chapter 2: Requirements , Feasibility and Scope/Objective	3-6
9	Chapter 3: Analysis , Activity Time Schedule (PERT)	7-8
10	Chapter 4: Design	9-15
11	Chapter 5: Implementation & Testing	16-44
12	Chapter 6: Limitations and Future Scope of the Project	45-45
13	Conclusions	46-46

14	References	47-47
----	------------	-------

LIST OF FIGURES

Figure No.	Title	Page No
1	Data Flow Diagram	10-10
2	E-R Diagram	10-10
3	Class Diagram	11-11
4	Activity Diagram	12-12

Introduction about Project

Now a days, Everyone wants to learn online using smart android application. Many Application on play store having provide more content, but in our app provide a better content with latest animations, bonus keys, leader board part to check score and rank. Using Android studio, uses java language and xml for designing part and for data safety we have used a Google firebase console to provide a security.

MODULES {FRAMES} OF CLOTHING STORE:-

Mainly this android application “QUIZ TIME” is based on Android Educational App. Data Comes from google firebase console. Provide more facility and course and content in a single smart android application. First sign in and sign up part using mail id or phone number and password. Then after sign in, user choose a course according to need and first read full syllabus having each chapter divided in to levels and each level connected with small quiz. After attend the quiz with 60% mark then only user goes to next. If user not get 60%, then a option of re-attempt with new random que not a last one. After complete one course with above 75% marks. User will get a special super bonus key to open one any course. So all data come to firebase then provide more safety with each user data, so no loss of data, learn and play, enjoying using this smart educational app.

This project aims to develop a Quiz App for users with the goal so that it is very easy to learn your education course part from an extensive number of store available on the play store . With the help of this you can choose any course from our QUIZ APP. Here is no compelling reason to go to the users data loss or bugging problems centres during reading and learning time . To get to this Quiz app all the users will need to have a email and password to login and proceed your choice course . The login credentials for an online learning app, system are under high security and nobody will have the capacity to crack it easily. Upon successful login the users can choose any choice and interested course such as Programming Language course, for small students Math, English GK and Science or many different stream wise courses Etc. And of course you will get your marks at your leader board . It is simple. You will go to your profile account and

check your rank with first 20 users and your rank. It just need a computer .Almost a wide range of course can be learn from this app. It is extremely secure. User security is more first.

Literature Reviews/Comparative study

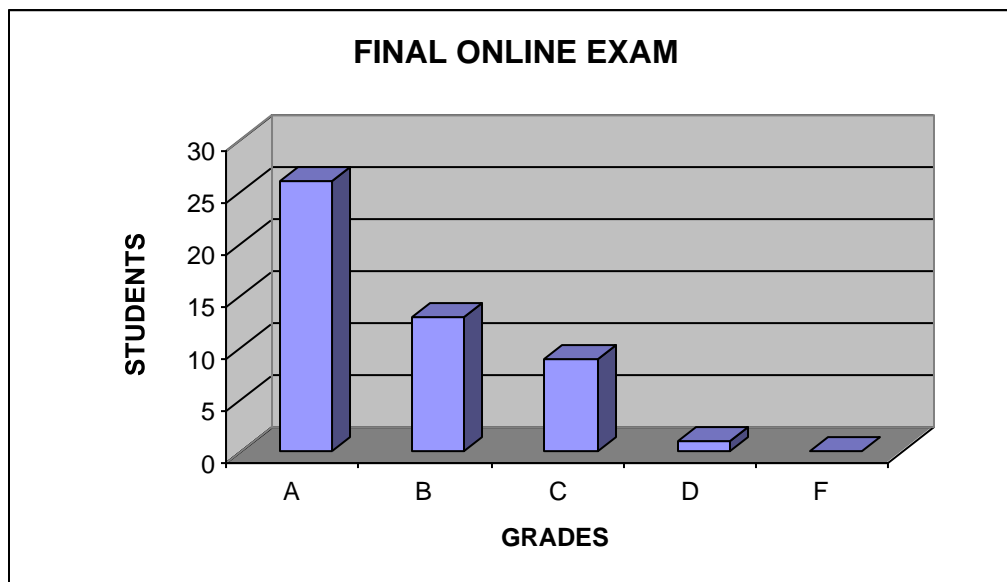
A literature review revealed research and studies based on implementation of a variety technologies in college courses. “The internet has opened many possibilities for the classroom instruction but it can also be a barrier to teaching as well” (Bugeja, 2006). The new innovative technologies provide opportunities to improve learning and create a more exciting and motivating environment (Connors, 2007).

According to a case study by Ralph, Buskirk, and Schmidt (2007) regarding the use of online projects, students in favor of online projects indicated that the accessibility to the professor for fast and easy feedback was a great asset. Futhermore, the study revealed that when implementing technology students were concerned with the expense of the technology, the necessity for internet access, and the reliability of the technology.

Research on student perceptions and satisfaction with online courses provide insights to student reactions and satisfaction with implementation of an online exam. Hale (2007) found that student satisfaction surveys reveal that the most important reason for taking a distance education course is its convenience. In addition, Steinman (2007) indicated that “students’ perceptions of online courses can be negative if they experience large transactional distance with the instructor and with other students and can influence whether a student will stay in or drop out of a class.”

1). METHODOLOGY, RESULTS, AND IMPLICATIONS

Two classes were utilized as a case study for implementing an online exam. The course already uses a computer simulation and although it is not required the courses have an online forum available for students use if they choose. The students are given two exams during the course: a midterm and a final. A total of 49 students participated in the two exams. The Midterm Exam covered eleven chapters of the textbook. The Midterm Exam consisted of 60 multiple choice questions generated from a textbook testbank. Two versions of the exam were created one for each class. The questions on the exam were in order with the textbook. Students were given 3 hours to complete the exam and were allowed to use their textbooks and notes on the exam. The exam was administered during the regular class session and the students took the traditional paper and pencil exam. The instructor collected the exams at the end of the three hour time limit, hand scored them, and returned the following week in class



2). Faculty and Student Feedback:-

This implementation of an online exam is the first time the instructor ever utilized any exam other than a traditional paper and pencil exam taken with a proctor to oversee the students completing the exam. The instructor indicated the following concerns to the use of an online exam:

- Cheating
- Reliability of technology
- Ease of use for instructor
- Ease of use for students
- Student satisfaction with online exam

Requirements, Feasibility and Scope/Objective

Now, this method is intended in such the way that it takes fewer resources to figure out work correctly. That is the minimum needs that we'd like to require care of:-

FRONT-END/BACK-END

➤ Front-End:

Using Java Programming Language , Extensible Markup Language, And by using smart android animations and tools.

➤ **Back-End:**

Google Firebase, Database Console

Hardware Requirements

- ✓ The system wants a minimum of Two GB of ram to run all the options.
- ✓ It wants a minimum 1.3 GHz processor to run smoothly.
- ✓ Rest is all up to the user's usage can take care of hardware.

Software Requirements

- ✓ Android Studio 11.3
- ✓ Google Firebase Console
- ✓ FOR SECURITY OPPOSING ANTI-VIRUS IS SUGGESTED.

FEASIBILTY ANALYSIS

Here are many Developers that operate as a Android Application whether they provide Students and Children, Online Course and Quizzes. However, most of these Developers operate based on traditional methods that need to be developed in order to attract Play Store Permissions. Hence, we have come up with an innovative idea of Special courses, quizzes android based on the most up-to-date techniques and scientific methods to satisfy all developers and help student to their knowledge effectively. In addition, highly experienced developers are needed to implement such modern techniques. We will show you how to conduct a feasibility study for a online quiz applications.

From the inception of ideas for software system, until it is implemented and provided to users and even after that the system undergoes gradual Developments and evaluations.

The application is said to have life cycle composed of several phases. At the feasibility stage, it is desirable that two or three different configuration will be pursued that satisfy the key technical requirement but which represent different Level of ambition and cost.

Feasibility is the determination of whether or not a project is worth doing. A Feasibility study is carried out select a best system that mate performance Requirements.

The data collected during primary investigation examines system feasibilities that is likelihood that the system will be beneficial to the organization? Four tests for Feasibility study are as follows:-

Technical Feasibility:

- ✓ This is concerned with specifying equipment and software that will successfully satisfy the use considerably.
- ✓ The feasibility to produce output in a given time because system is fast enough to handle multiple users.
- ✓ Response time under certain circumstances and ability to process a certain volume of transaction of a particular speed.
- ✓ Feasibility to communicate data to distant location.

Economical Feasibility:

- ✓ Economic analysis is the most frequently used technique used for evaluating the effectiveness of a proposed system.
- ✓ More commonly known as cost/benefit analysis the procedure is to determine the benefits and savings that are expected from a proposed system and compared them with cost.

- ✓ Though the cost of installing the system may appear high, it is one time investment.

Operational Feasibility:

It is mainly related to human organizational as social aspects. The points to be considered are -
The system interface is standard, user friendly and provides extensive help. Hence no special training is not required.

Social Feasibility:

Social feasibility is determination of whether a proposed project will be acceptable to people or not, so this project is totally Social and Feasible and Education Based .

Scope:-

The basic need of users and children being is online courses and quizzes. According to archaeologists and anthropologists, earliest learning most likely included grass or leaves tied around the body. With the passage of time, quizzes performed a range of educational and social functions. It also functions as a form of adornment and an expression of personal taste, style and social status. However, the main function of quiz application is to increase the comfort of the learning or increase the knowledge .

The small child are more into online quizzes . . All big Android Education companies now come up with standard courses and users know the value of courses which would help to learn them.. This is not the same with the present generation. So it makes better sense to have online Education Quiz App . In addition to this, a lot more options are available when Learning from this app for is done online. The users gets a lot of information on the courses, quizzes, and ranking b/w lot of users provide more competition which is available. The online learning nowadays are more secure.

Analysis, Activity Time Schedule (PERT)

Analysis

The analysis must achieve three primary objectives:

- ❖ To describe what the user requires.
- ❖ To establish the basis for the enhancement of a software design.
- ❖ To define a set of requirements that can be validated once the software is completely enhanced.
- ❖ At the core of the model lies the *data dictionary*, which is a repository that contains descriptions of all the data objects consumed or produced by the software .Three different diagrams surround the core.
- ❖ The entity relation diagram depicts relationships between data objects.
- ❖ The data flow diagram provides an indication of how the data is transformed as they move through the system.
- ❖ The state transition diagram indicates how the system behaves as a consequence of external events.

Activity Time Schedule (PERT)

The Program Evaluation Review Technique, or PERT, is a visual tool used in project planning. Using the technique helps project planners identify start and end dates, as well as interim required tasks and timelines. The information is displayed as a network in chart form.

PERT helps project planners identify:

- Start and end dates
- Anticipated total required completion time
- All activities, referred to as events on the chart, that impact the completion time
- The required sequence of events
- The probability of completion by a certain date

The PERT Process

PERT has a set series of steps in mapping out a complex project, which include:

1. List all the tasks and milestones (a.k.a. events) required for completion of the project
2. Determine the required sequence of tasks
3. Design a chart to visually display all the steps
4. Estimate the time required for each task
5. Identify the critical path – the longest series of tasks in the project
6. Adjust the chart to reflect progress made once the project starts

A PERT chart uses numbered circles or rectangles to represent milestones and straight lines with arrows at the end to represent tasks to be completed. The direction of the arrows, and the numbers, indicate the required sequence. Typically, the numbers increase by 10 at each milestone, so that new tasks can be added along the way without requiring the whole chart to be redrawn and numbered.

Design

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top down approach, designers first identify and link major program

components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections.

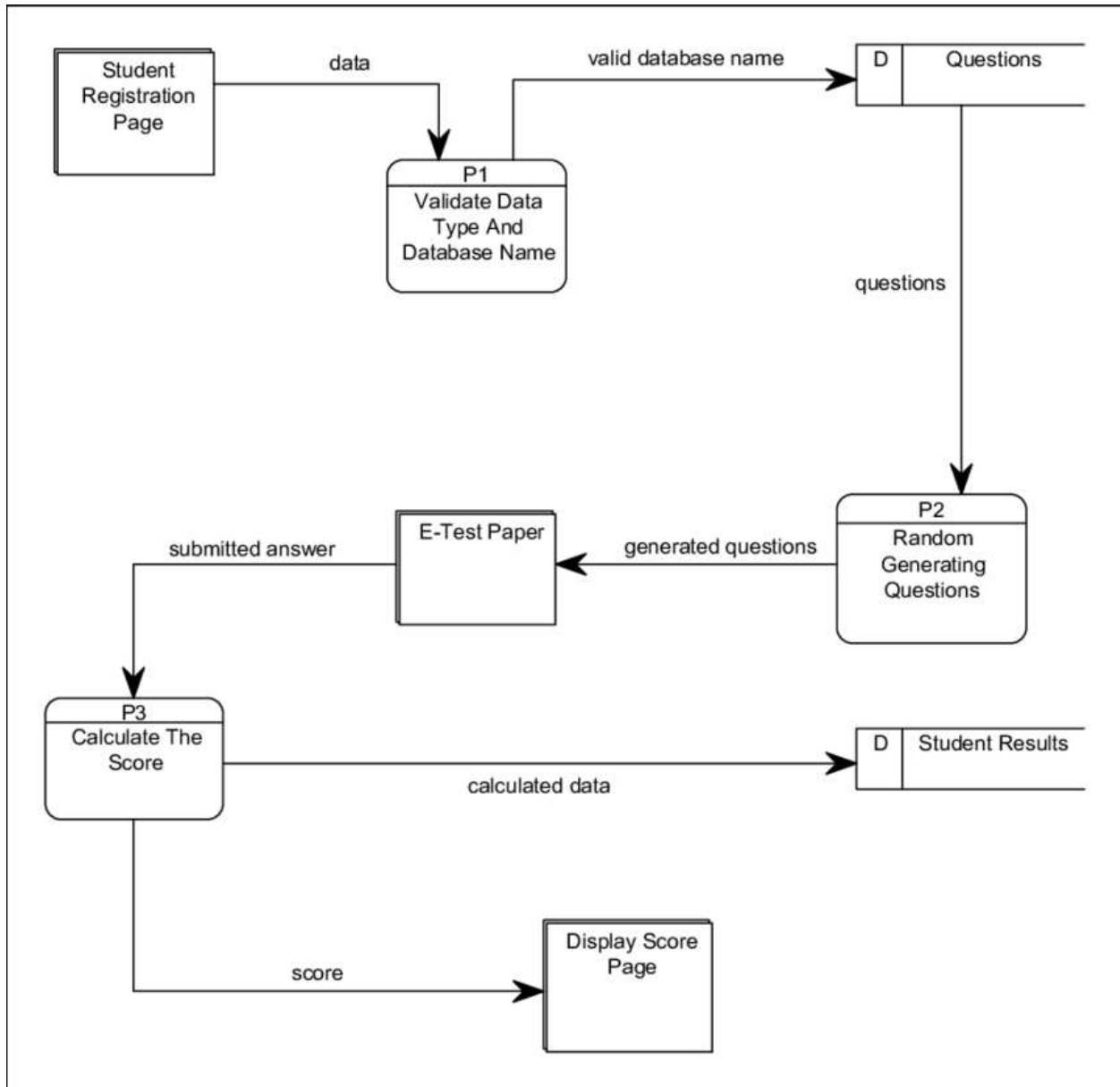
Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design.

Designers should carefully document completed designs. Detailed documentation enhances a programmer's ability to develop programs and modify them after they are placed in production. The documentation also helps management ensure final programs are consistent with original goals and specifications. Organizations should create initial testing, conversion, implementation, and training plans during the design phase. Additionally, they should draft user, operator, and maintenance manuals.

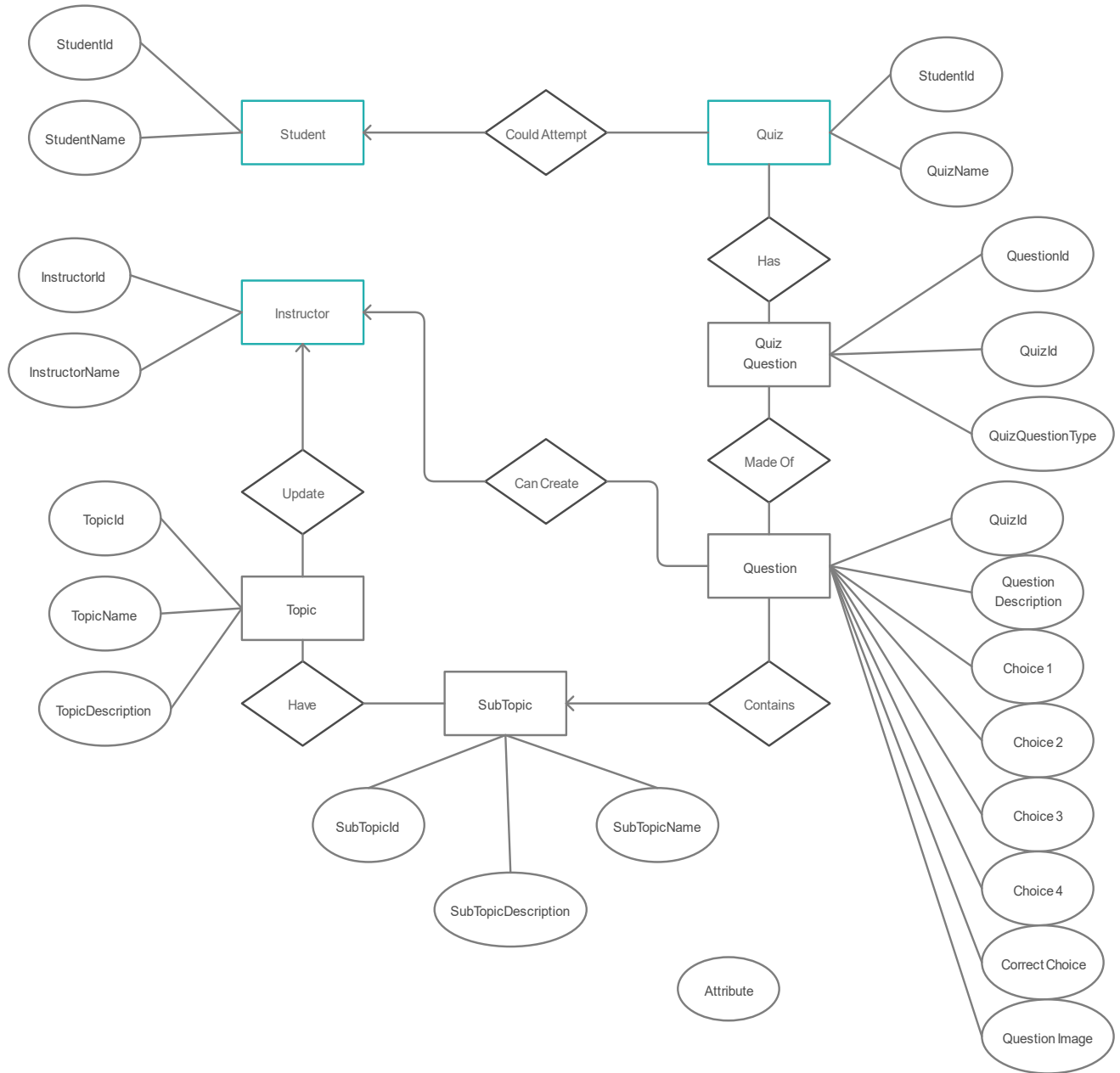
For design of the project:

1. First Database has to be designed which can be used to handle all the requirements of the users.
2. The basic structure of the project has to be designed.

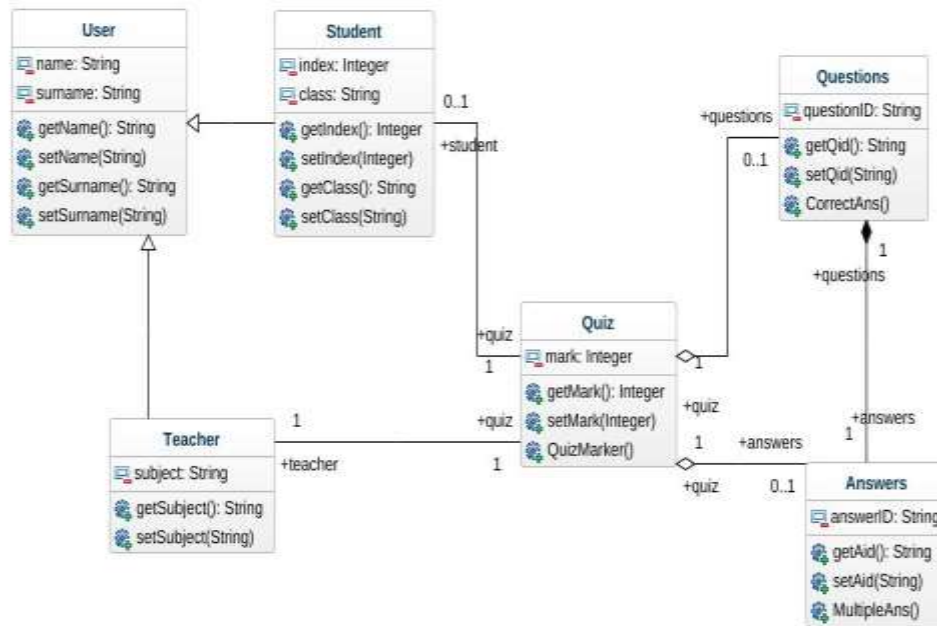
DFD DIAGRAM FOR QUIZ TIME ANDROID APPLICATION:



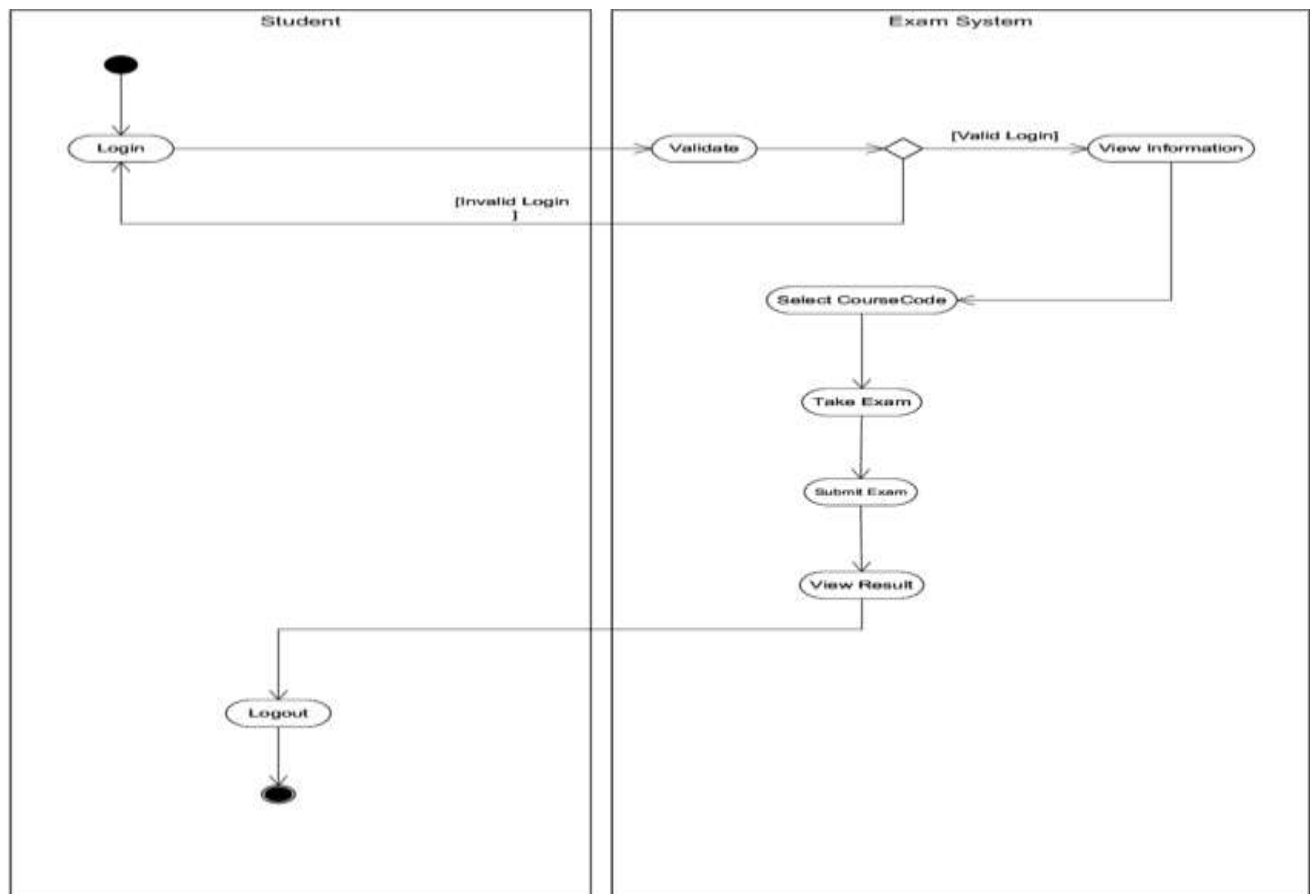
ER DIAGRAM FOR QUIZ TIME ANDROID APPLICATION:



CLASS DIAGRAM FOR QUIZ TIME ANDROID APPLICATION:



ACTIVITY DIAGRAM FOR QUIZ TIME ANDROID APPLICATION:



REGISTRATION ACTIVITY:



LOGIN ACTIVITY:



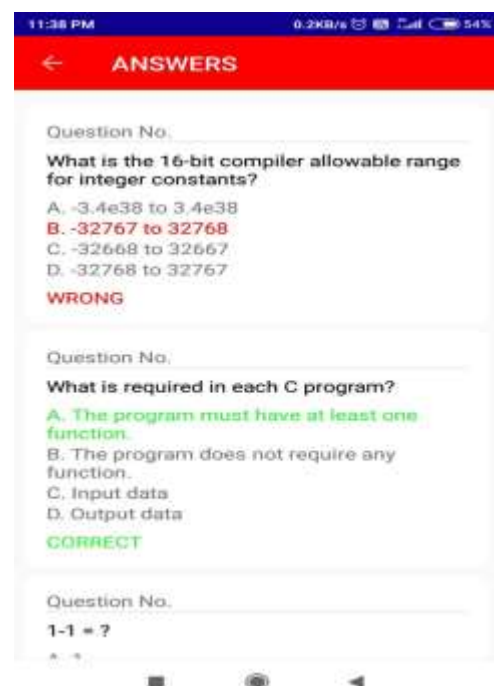
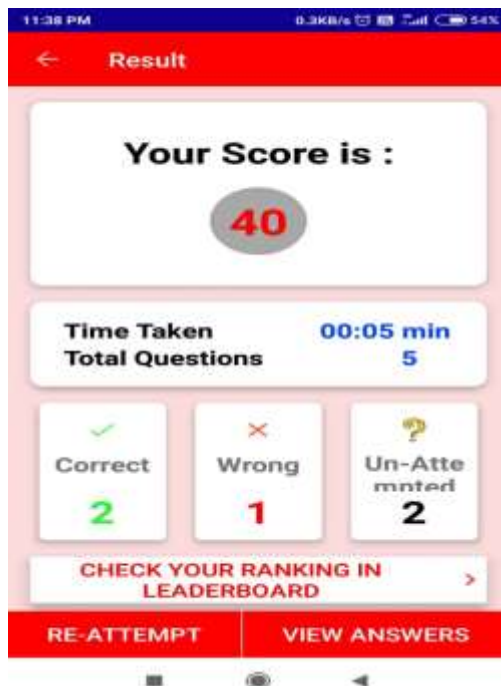
COURSE AND QUIZ ACTIVITY:



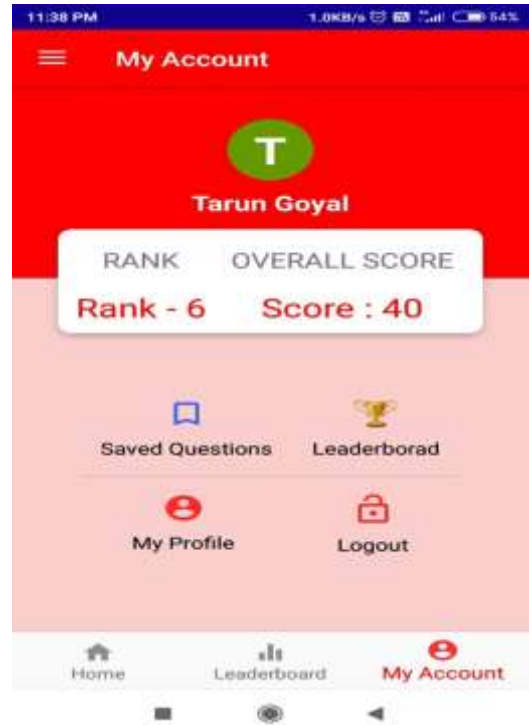
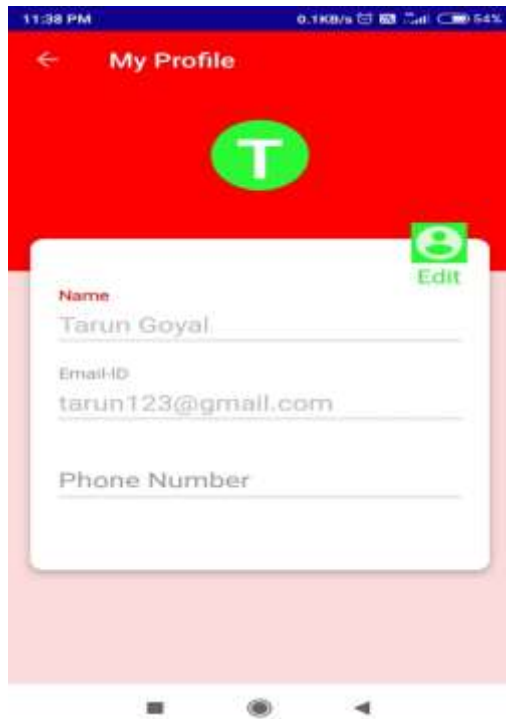
START QUIZ AND QUESTIONS ACTIVITY:



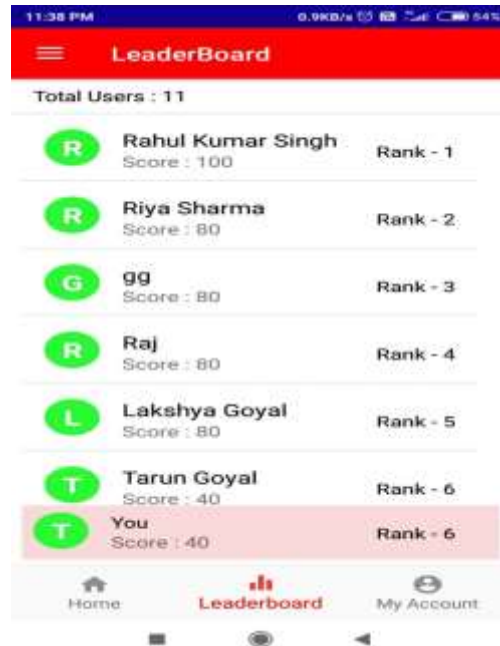
RESULT/SCORE AND ANSWERS ACTIVITY:



PROFILE AND ACCOUNT DETAILS ACTIVITY



LEADERBOARD ACTIVITY



Implementation & Testing

MODULES OF QUIZ TIME APPLICATION:-

Registration Module:

If you are a new user, you must register first. By Clicking on Sign up Button. After Clicking on Sign up Button you're able to find a new Activity Page where you have to fill all the essential information like Full Name, Email Address, Password, Confirm Password and Read the Terms and Condition carefully.

After filling all the required information then you have to submit the information by clicking on SIGNUP Button.

Without Registration you are not able to move inside the app and not able to use any features of our QUIZ APP.

After Filling All Information

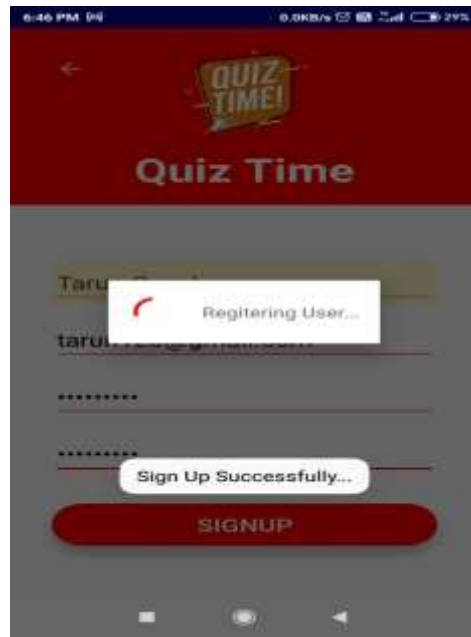


A screenshot of the Quiz Time app's registration screen. The header is red with a white back arrow, a 'QUIZ TIME!' logo, and the text 'Quiz Time'. Below the header, there are four input fields: 'Full Name' (containing 'Tarun Goyal'), 'Password', and 'Confirm Password'. A red 'SIGNUP' button is at the bottom.



A screenshot of the Quiz Time app's registration screen after submission. The header is red with a white back arrow, a 'QUIZ TIME!' logo, and the text 'Quiz Time'. Below the header, the 'Full Name' field is highlighted in yellow and contains 'Tarun Goyal'. The 'Email Address' field contains 'tarun123@gmail.com'. The 'Password' and 'Confirm Password' fields are masked with dots. A red 'SIGNUP' button is at the bottom.

After clicking on SIGNUP Account button a TOAST Message is Display “Sign Up Successfully” :



Coding for Regsitratio Activity:-

```
package goyal.world.quiztime;

import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Dialog;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class SignUpActivity extends AppCompatActivity {

    private EditText name, email, password, confirmPassword;
    private Button signUpBtn;
    private ImageView backBtn;
    private FirebaseAuth mAuth;
```

```

private String emailStr, passStr, confirmPassStr, nameStr;
private Dialog progressDialog;
private TextView dialogText;

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

    name = findViewById(R.id.user_name);
    email = findViewById(R.id.email_id);
    password = findViewById(R.id.password);
    confirmPassword = findViewById(R.id.confirm_password);
    signUpBtn = findViewById(R.id.signUp_btn);
    backBtn = findViewById(R.id.back_btn);

    progressDialog = new Dialog(SignUpActivity.this);
    progressDialog.setContentView(R.layout.dialog_layout);
    progressDialog.setCancelable(false); // so that user can not cancel this
dialog //set the Layout for progress window
    progressDialog.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);

    dialogText = progressDialog.findViewById(R.id.dialog_text);
    dialogText.setText("Regitering User...");

    mAuth = FirebaseAuth.getInstance();

    backBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish(); // backBtn: back to the previous activity(this activity is
finish...)
        }
    });

    signUpBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (validate()) {
                signupNewUser();
            }
        }
    });
}

private boolean validate()
{
    nameStr = name.getText().toString().trim(); //trim use for delete the space
from string(i.e. Start and End Space Only)
    passStr = password.getText().toString().trim();
    emailStr = email.getText().toString().trim();
    confirmPassStr = confirmPassword.getText().toString().trim();
}
}

```



```

    if (nameStr.isEmpty())
    {
        name.setError("Enter Your Name");
        return false;
    }

    if (emailStr.isEmpty())
    {
        email.setError("Enter Your E-Mail ID");
        return false;
    }

    if (passStr.isEmpty())
    {
        password.setError("Enter Password");
        return false;
    }

    if (confirmPassStr.isEmpty())
    {
        confirmPassword.setError("Enter Confirm Password");
        return false;
    }

    if (passStr.compareTo(confirmPassStr) != 0) // 0 means for same
    {
        Toast.makeText(SignUpActivity.this, "Password and Confirm Password should
be same !", Toast.LENGTH_SHORT).show();
        return false;
    }

    return true;
}

private void signupNewUser()
{
    progressDialog.show();

    mAuth.createUserWithEmailAndPassword(emailStr, passStr)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @RequiresApi(api = Build.VERSION_CODES.KITKAT)
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {

                    Toast.makeText(SignUpActivity.this, "Sign Up
Successfully...", Toast.LENGTH_SHORT).show();

                    DbQuery.createUserData(emailStr, nameStr, new
MyCompleteListener(){

                        @Override
                        public void onSuccess() {

                            DbQuery.loadData(new MyCompleteListener() {
                                @Override
                                public void onSuccess() {
                                    progressDialog.dismiss();
                                }
                            });
                        }
                    });
                }
            }
        });
}

```

```

        Intent intent = new
Intent(SignUpActivity.this, LoginActivity.class);
        startActivity(intent);
        SignUpActivity.this.finish();
    }

    @Override
    public void onFailure() {
        Toast.makeText(SignUpActivity.this,
"Something went wrong ! please Try Again Later !",
        Toast.LENGTH_SHORT).show();
        progressDialog.dismiss();
    }
});

}

@Override
public void onFailure() {

        Toast.makeText(SignUpActivity.this, "Something
went wrong ! please Try Again Later !",
        Toast.LENGTH_SHORT).show();
        progressDialog.dismiss();
    }
});

} else {
    // If sign in fails, display a message to the user.
    progressDialog.dismiss();
    Toast.makeText(SignUpActivity.this, "Authentication
failed.",
        Toast.LENGTH_SHORT).show();
}
});
}
}
}

```

Login Module:

After registration you are able to take all the benefits of Quiz APP and able to see all Courses and Quizzes’.

Here either Boy Girl Small Child All have to log in with their unique Email Address and Password. After this, they will be directed to the primary user interface from where they have further options.

After Filling the Correct Email Id and Password and Login Successful Part:



After Login you're able to difference section for courses like C,C++, Java, Python if you clicked on C then your page will move to C Course section you're able to see difference –different Level related to C courses like Intro of C, Programming Structure, flow of c program, data type and keywords or more like that if you click on Java then your page will move to Java Course section and you're able to see different –different product related to Java Programming Language like Intro, What is JDK, JRE and JVM in Java, Java Variables and Data Type or more .

Coding for LogIn Activity:-

```
package goyal.world.quiztime;

import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Dialog;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RelativeLayout;
import android.widget.TextView;
```

```

import android.widget.Toast;

import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.snackbar.Snackbar;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GoogleAuthProvider;

public class LoginActivity extends AppCompatActivity {

    private EditText email, password;
    private Button loginBtn;
    private TextView forgotPassBtn, signUpBtn;
    private FirebaseAuth mAuth;
    private Dialog progressDialog;
    private TextView dialogText;
    private RelativeLayout gSignB;
    private GoogleSignInClient mGoogleSignInClient;
    private int RC_SIGN_IN = 104;

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

        email = findViewById(R.id.email);
        password = findViewById(R.id.password);
        loginBtn = findViewById(R.id.login_btn);
        forgotPassBtn = findViewById(R.id.forgot_password);
        signUpBtn = findViewById(R.id.signUp_btn);
        gSignB = findViewById(R.id.g_signB);

        progressDialog = new Dialog(LoginActivity.this);
        progressDialog.setContentview(R.layout.dialog_layout);
        progressDialog.setCancelable(false); // so that user can not cancel this
dialog
        //set the layout for progress window
        progressDialog.getWindow().setLayout(ViewGroup.LayoutParams.WRAP_CONTENT,
ViewGroup.LayoutParams.WRAP_CONTENT);

        dialogText = progressDialog.findViewById(R.id.dialog_text);
        dialogText.setText("Signing In...");

        mAuth = FirebaseAuth.getInstance();

        // Configure Google Sign In
        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestIdToken(getString(R.string.default_web_client_id))
                .requestEmail()
                .build();

```

```

mGoogleSignInClient = GoogleSignIn.getClient(this, gso);

loginBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        if (validateData())
        {
            login();
        }
    }
});

signupBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(LoginActivity.this, SignUpActivity.class);
        startActivity(intent); // will not close this login activity so if
        user want to comeback the login activity he can do that also...
    }
});

gSignB.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        googleSignIn();
    }
});

}

private boolean validateData()
{

    if(email.getText().toString().isEmpty())
    {
        email.setError("Enter E-Mail ID");
        return false;
    }

    if (password.getText().toString().isEmpty())
    {
        password.setError("Enter Password");
        return false;
    }

    return true;
}

private void login()
{
    progressDialog.show();

    mAuth.signInWithEmailAndPassword(email.getText().toString().trim(),
password.getText().toString().trim())
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {

```

```

        // Sign in success, update UI with the signed-in user's
information
        Toast.makeText(LoginActivity.this, "Login
Successfully...", Toast.LENGTH_SHORT).show();

        DbQuery.loadData(new MyCompleteListener() {
            @Override
            public void onSuccess() {
                progressDialog.dismiss();
                Intent intent = new Intent(LoginActivity.this,
MainActivity.class);

                startActivity(intent);
                finish();
            }

            @Override
            public void onFailure() {
                progressDialog.dismiss();
                Toast.makeText(LoginActivity.this, "Something
went wrong ! Please try again...",
                    Toast.LENGTH_SHORT).show();
            }
        });
    } else {
        // If sign in fails, display a message to the user.
        progressDialog.dismiss();
        Toast.makeText(LoginActivity.this,
task.getException().getMessage(),
            Toast.LENGTH_SHORT).show();
    }
    });
}

private void googleSignIn()
{
    Intent signInIntent = mGoogleSignInClient.getSignInIntent();
    startActivityForResult(signInIntent, RC_SIGN_IN);
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    // Result returned from launching the Intent from
GoogleSignInApi.getSignInIntent(...);
    if (requestCode == RC_SIGN_IN) {
        Task<GoogleSignInAccount> task =
GoogleSignIn.getSignedInAccountFromIntent(data);
        try {
            // Google Sign In was successful, authenticate with Firebase
            GoogleSignInAccount account = task.getResult(ApiException.class);
            //Log.d(TAG, "firebaseAuthWithGoogle:" + account.getId());
            firebaseAuthWithGoogle(account.getIdToken());
        }
    }
}

```

```

    } catch (ApiException e) {
        // Google Sign In failed, update UI appropriately
        Toast.makeText(LoginActivity.this, e.getMessage(),
Toast.LENGTH_SHORT).show();
        // ...
    }
}

private void firebaseAuthWithGoogle(String idToken) {

    progressDialog.show();

    AuthCredential credential = GoogleAuthProvider.getCredential(idToken, null);
    mAuth.signInWithCredential(credential)
        .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
            @RequiresApi(api = Build.VERSION_CODES.KITKAT)
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in user's
information
                    Toast.makeText(LoginActivity.this, "Google SignIn
Successfully", Toast.LENGTH_SHORT).show();
                    // Log.d(TAG, "signInWithCredential:success");

                    FirebaseUser user = mAuth.getCurrentUser();

                    if (task.getResult().getAdditionalUserInfo().isNewUser())
                    {
                        DbQuery.createUserData(user.getEmail(),
user.getDisplayName(), new MyCompleteListener() {
                            @Override
                            public void onSuccess() {

                                DbQuery.loadData(new MyCompleteListener() {
                                    @Override
                                    public void onSuccess() {
                                        progressDialog.dismiss();

                                        Intent intent = new
Intent(LoginActivity.this, MainActivity.class);
                                        startActivity(intent);
                                        LoginActivity.this.finish();
                                    }

                                    @Override
                                    public void onFailure() {
                                        progressDialog.dismiss();
                                        Toast.makeText(LoginActivity.this,
"Something went wrong ! Please try again...",
                                        Toast.LENGTH_SHORT).show();
                                    }
                                });
                            }

                            @Override
                            public void onFailure() {
                                progressDialog.dismiss();

```

```

        Toast.makeText(LoginActivity.this, "Something
went wrong ! Please try again...",
                        Toast.LENGTH_SHORT).show();
    }
    });
}
else
{
    DbQuery.LoadData(    new MyCompleteListener() {
        @Override
        public void onSuccess() {
            progressDialog.dismiss();

            Intent intent = new
Intent(LoginActivity.this, MainActivity.class);
            startActivity(intent);
            LoginActivity.this.finish();
        }

        @Override
        public void onFailure() {
            progressDialog.dismiss();
            Toast.makeText(LoginActivity.this, "Something
went wrong ! Please try again...",
                        Toast.LENGTH_SHORT).show();
        }
    });
}

} else {
    progressDialog.dismiss();
    // If sign in fails, display a message to the user.
    Toast.makeText(LoginActivity.this,
task.getException().getMessage(), Toast.LENGTH_SHORT).show();
}

// ...
});
}
}
}

```

TESTING

Software testing

Software testing is the process used to measure the quality of developed computer software. Usually, quality is constrained to such topics as correctness, completeness, security, but can also

include more technical requirements as described under the ISO standard ISO 9126, such as capability, reliability, efficiency, portability, maintainability, compatibility, and usability.

Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate.

White box, black box, and grey box testing

White box and black box testing are terms used to describe the point of view that a test engineer takes when designing test cases. **Black box** testing treats the software as a black-box without any understanding as to how the internals behave. Thus, the tester inputs data and only sees the output from the test object. This level of testing usually requires thorough test cases to be provided to the tester who then can simply verify that for a given input, the output value (or behaviour), is the same as the expected value specified in the test case.

White box testing, however, is when the tester has access to the internal data structures, code, and algorithms. For this reason, unit testing and debugging can be classified as white-box testing and it usually requires writing code, or at a minimum, stepping through it, and thus requires more skill than the black-box tester. If the software in test is an interface or API of any sort, white-box testing is almost always required.

In recent years the term grey box testing has come into common usage. This involves having access to internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level. Manipulating input data and formatting output do not qualify as grey-box because the input and output are clearly outside of the black-box we are calling the software under test.

Grey box testing could be used in the context of testing a client-server

Environment when the tester has control over the input, inspects the value in a SQL database, and the output value, and then compares all three (the input, sql value, and output), to determine if the data got corrupt on the database insertion or retrieval.

Level of testing

- **Unit testing** tests the minimal software component, or module. Each unit (basic component) of the software is tested to verify that the detailed design for the unit has been correctly implemented. In an Object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

- **Integration testing** exposes defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

- **Functional testing** tests at any level (class, module, interface, or system) for proper functionality as defined in the specification.

- **System testing** tests a completely integrated system to verify that it meets its requirements.

- **System integration testing** verifies that a system is integrated to any external or third party systems defined in the system requirements.

- **Acceptance testing** can be conducted by the end-user, customer, or client to validate whether or not to accept the product. Acceptance testing may be performed as part of the hand-off process between any two phases of development.

- **Alpha testing** is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site.

Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

- **Beta testing** comes after alpha testing. Versions of the software, known as beta versions, are released to a limited audience outside of the company. The software is released to groups of people so that further testing can ensure the product has few faults or bugs.

Limitations and Future Scope of the Project

Limitations

- ❖ Some Time Data Glitching Problem Because More Users At one time work.
- ❖ This is the Android Based Application so every machine needs to install this application.
- ❖ All Android Phones is required to install the Application .
- ❖ For Installing the Application the system need minimum 2 GB of RAM and 1.3 GHz Processor.

Future Scope

Every developer have aims to improve their system or project for the next project. For this system, the improvements are needed on question type such as essay and random short answer. A lot of improvements need to be done on the interface part and admin, lecturer and student module. The process of performance system on the coding with database server will be more efficient if this improvements are achieved. Besides that, the improvements on the interface part and functionality system can be prevent the error during running the system

- ❖ Android applications make it a lot easy for users to learn each and every course and get Knowledge free, but also for a developer as well. Users no more worry about the quality, size and other aspects since everything is already mentioned over there.
- ❖ People will get All Courses with no price.

Conclusions

We have successfully implemented the Android QUIZ TIME APPLICATION.

This project more focus on the problems who faced by the lecturers due to the manual system which are available now is not efficient. For instance, the losing of quiz paper, no pencil and no pen to do quiz on time. The lecturer cannot setup quiz for specific time and have to do quiz in class. Lecturer also need spend more time on grading. While, student also cannot take quiz from any location. This problem can be solved by Online Quiz System. In addition, this kind of system can be replaced the manual system in order to increase the efficiency of the quiz management.

The process in build up this system takes a long time. So that, the insufficient time problems should be faced and solved. In order to produce a good system, the time management is important to ensure the good quality of the system. The improvement should be done and insufficient time should be solved in order to make sure this system can run well and meet the user requirements.

References

- **Android Developer Course is available at -**
 - <https://developer.android.com/>
- **Java Documentation is available at -**
 - <https://docs.oracle.com/en/java/javase/14/>
- <https://www.javatpoint.com/>
- Sk. Imran Hossain Shoyeb. Android based Online Quiz Application. Journal from Department of Computer Science and Engineering, East West University, Dhaka-1212, Bangladesh, May 2015.