CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

MAJOR PROJECT

ON

Wi-Fi Based Mobile Quiz/Online Test(MCQ)

SUBMITTED TO:

Prof. Manpreet Kaur

SUBMITTED BY :

PAMIT DUGGAL (16BCA1440)

AMIT KASHYAP(16BCA1017)

SHANKIT KAPIL(16BCA1360)

MOONIB SARFARAZI(16BCA1031)

DECLARATION BY CANDIDATE

I, hereby, certify that this "**Wi-Fi Based Mobile Quiz/Online Test(MCQ**), is presented in partial fulfillment for the award of **BACHELORS OF COMPUTER APPLICATION** at **CHANDIGARH UNIVERSITY, PUNJAB**. This report is an authentic record of our project work carried out during a period from 05 JAN 2019 to 12 APRIL 2019 .The matter presented in this report has not been submitted by any other University Institute for the award of Bachelors of Computer Application.

Signature of the Student

ACKNOWLEDGEMENT

I take immense pleasure in thanking our H.O.D **Mr. Kamaljeet Singh Saini** for having permitted me to carry out this project work.

I wish to express my deep sense of gratitude to our Guide ……..……… for her able guidance and useful suggestions, which helped us in completing the project work, in time.

Words are inadequate in offering our thanks for their encouragement and cooperation in carrying out the project work.

Finally, yet importantly, we would like to express our heartfelt thanks our beloved parents or their blessings, our friends & classmates for their help and wishes for the successful completion of this project.

CONTENTS

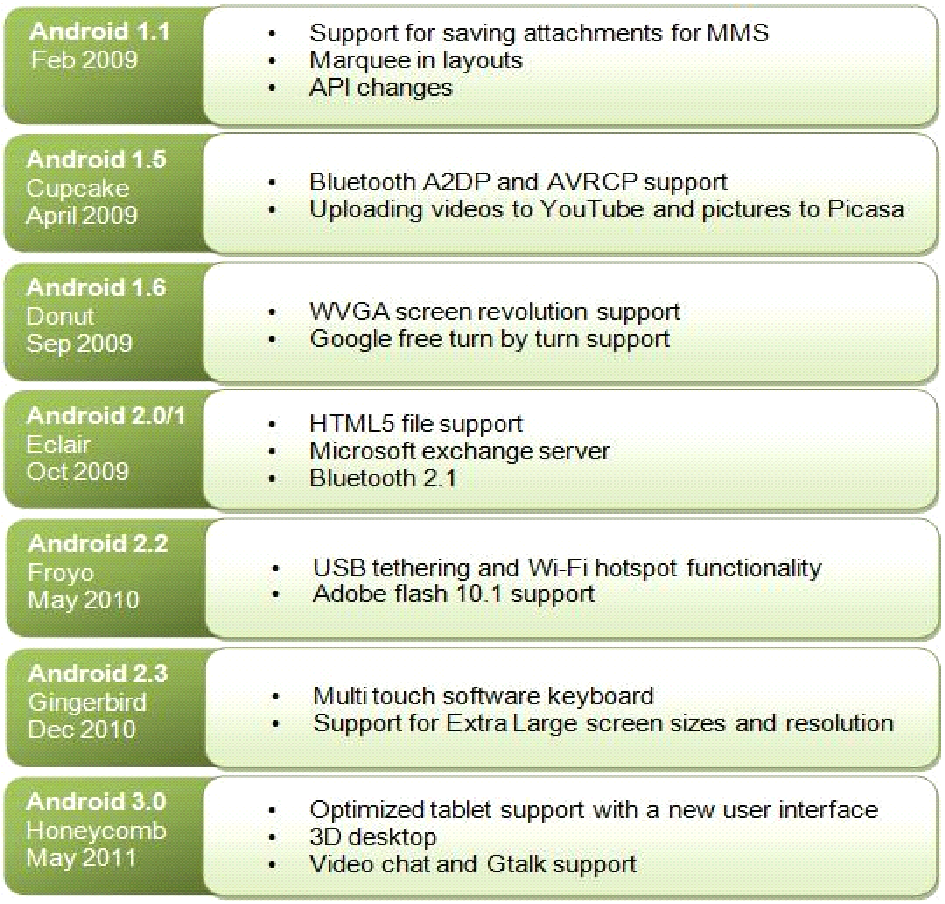
Page No.

Overview

What isAndroid?

Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

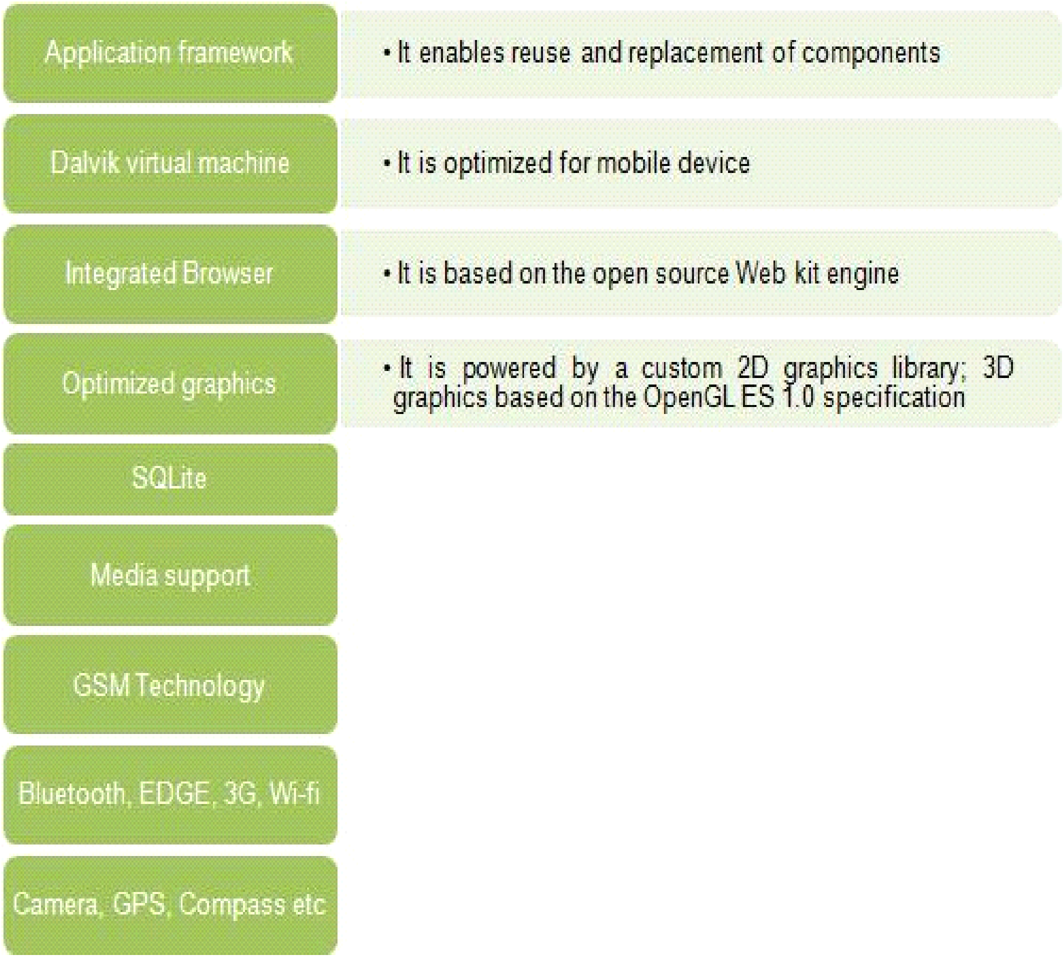
One of the most widely used mobile OS these days is **ANDROID**. **Android** is a software bunch comprising
no only operating system but also middleware and key applications. Android Inc was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. After original release there have been number of updates in the original version
of Android.



Android 1.1 Feb 2009	• Support for saving attachments for MMS • Marquee in layouts • API changes
Android 1.5 Cupcake April 2009	• Bluetooth A2DP and AVRCP support • Uploading videos to YouTube and pictures to Picasa
Android 1.6 Donut Sep 2009	• WVGA screen revolution support • Google free turn by turn support
Android 2.0/1 Eclair Oct 2009	• HTML5 file support • Microsoft exchange server • Bluetooth 2.1
Android 2.2 Froyo May 2010	• USB tethering and Wi-Fi hotspot functionality • Adobe flash 10.1 support
Android 2.3 Gingerbird Dec 2010	• Multi touch software keyboard • Support for Extra Large screen sizes and resolution
Android 3.0 Honeycomb May 2011	• Optimized tablet support with a new user interface • 3D desktop • Video chat and Gtalk support

Features & Specifications

Android is a powerful Operating System supporting a large number of applications in Smart Phones. These applications make life more comfortable and advanced for the users. Hardwares that support Android are mainly based on ARM architecture platform. Some of the current features and specifications of android are:



Application framework	• It enables reuse and replacement of components
Dalvik virtual machine	• It is optimized for mobile device
Integrated Browser	• It is based on the open source Web kit engine
Optimized graphics	• It is powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification
SQLite	
Media support	
GSM Technology	
Bluetooth, EDGE, 3G, Wi-fi	
Camera, GPS, Compass etc	

Android comes with an Android market which is an online software store. It was developed by Google. It allows Android users to select, and download applications developed by third party developers and use them. There are around 2.0 lack+ games, application and widgets available on the market for users.

Android applications are written in java programming language. Android is available as open source for developers to develop applications which can be further used for selling in android market. There are around 200000 applications developed for android with over 3 billion+ downloads. Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model.

For software development, Android provides **Android SDK** (Software development kit). Read more about open source software.

Android is a powerful operating system competing with Apple 4GS and supports great features. Few of them are listed below:

Feature	**Description**
Beautiful UI	Android OS basic screen provides a beautiful and intuitive user interface.
Connectivity	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
Storage	SQLite, a lightweight relational database, is used for data storage purposes.
Media support	H.263, H.264, MPEG-4 SP, AMR, AMR- WB, AAC, AAC 5.1, MP3, MID, WAV, JPEG, PNG, GIF, and BMP.
Messaging	SMS and MMS
Web browser	Based on the open-source WebKit layout.
Multi-touch	Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
Resizable widgets	Widgets are resizable, so users can expand them to show more content or shrink them to save space
Multi-Language	Supports single direction and bi-directional text.

GCM	Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution.
Wi-Fi Direct	A technology that lets apps discover and pair directly, over a high-bandwidth peer-to- peer connection.
Android Beam	A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

Android Applications

Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play** or the **Amazon Appstore**. Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide. This tutorial has been written with an aim to teach you how to develop and package Android application. We will start from environment setup for Android application programming and then drill down to look into various aspects of Android applications.

TECHNOLOGY USED:-

Android software development kit

The Android software development kit (SDK) includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux, Mac OS X 10.5.8 or later, and Windows XP or later. As of March 2015, the SDK is not available on Android itself, but the software development is possible by using specialized Android applications.

Android Studio, [9] made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

Java development kit

The Android build process depends on a number of tools from the JDK. Check out the build system overview documentation. The first big piece we need from JDK is

javac- all your source code written in Java needs to be compiled before it can be converted to the DEX format.

Once your code has been compiled, dexed, and packaged into an APK, we need jar signer to sign the APK.

There are some efforts out there to bring Java 8 features to Android, most notably gradle-retrolambda . Some of these require JDK 8 to compile properly.

Tools used for project

Android Studio

Android Studio is the official Integrated development environment (IDE) for Android platform development.

It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0 .

Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.

The first stable build was released in December 2014, starting from version 1.0.

Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows , Mac OS X and Linux ,

and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development.

Android Tool 1: Eclipse w/ADT

Although Eclipse is not the only Java development environment that can be used to develop Android applications, it is by far the most popular. This is partially due to its cost (free!) but mostly due the strong integration of the Android tools with Eclipse. This integration is achieved with the Android Development Tools (ADT) plug-in for Eclipse, which can be downloaded from the Android website.

Android Tool 2: The SDK and AVD Manager

This tool serves a number of important functions. It manages the different versions of the Android SDKs (build targets) that you can develop for, as well as third-party add- ons, tools, devices drivers, and documentation. Its second function is to manage the Android Virtual Device configurations (AVDs) you use to configure emulatorinstances.

Android Tool 3: Android Debug Bridge

The Android Debug Bridge (ADB) connects other tools with the emulator and devices. Besides being critical for the other tools (most especially the Eclipse ADT plug-in) to function, you can use it yourself from the command line to upload and download files, install and uninstall packages, and access many other features via the shell on the device or emulator.

Introduction to App

Our project is on android Application name ‘Wi-Fi Based Mobile Quiz/Online Test(MCQ).

The main objective of “Quiz Contest” is to facilitate a user friendly environment for all users and reduces the manual effort. In past days quiz is conducted manually but in further resolution of the technology we are able to generate the score and pose the queries automatically. The functional requirements include to create users that are going to participate in the quiz,automatic score and report generation and administrative tasks like add, delete, update for admin privilege users. In this application, all the permissions lies with the administrator i.e., specifying the details of the quiz with checking result will show to interviewee or not, addition of question and answers, marks for each question, Set timer for each quiz and generate report with score for each quiz.

System specification

- Mobile Version : Android 4.0 or upper versions
- Android Studio 3.1.0
- 8GB RAM
- 500 GB SSD

SOFTWARE RESOURCE REQUIRMENT DOCUMENT

Introduction

Relevance:

Quiz Contest is an android application that has general questions related to current affairs and computer. It has multiple choice questions with time limit and it also calculate scores of each correct answer. It is good for students of every age group it helps in increasing general knowledge about world ,Sports and computer etc. Don't need register simply give any user name and password it will saved automatically and you can login again with same user name and password don’t have to worry about the past score. The application helps the user to increase his/her knowledge. Since Smartphone mobiles are being widely used by general population and students, the Quiz Contest application can provide on the Student's mobile.

Problem Definition:

Quiz Contest is a application developed to conduct an quiz based on time constraints. Quiz Contest system is accessed by entering the user name and password which is added to the database. Before start of the quiz, the rules and regulations are displayed that includes description of the time limit, number of questions to be answered and scoring methods. Quiz is started by displaying one question with four options each based on computer and general knowledge. if the answer is correct, score is incremented by four and no negative marks for wrong answers . If the time exceeds 20secs next question will come automatically after giving few limited question's answer quiz application will finally direct you to the score page. Final score will be displayed and updated in the database with username.

Objective:

The main objective of “Quiz Contest” is to facilitate a user friendly environment for all users and reduces the manual effort. In past days quiz is conducted manually but in further resolution of the technology we are able to generate the score and pose the queries automatically. The functional requirements include to create users that are going to participate in the quiz, automatic score and report generation and administrative tasks like add, delete, update for admin privilege users. In this application, all the permissions lies with the administrator i.e., specifying the details of the quiz with checking result will show to interviewee or not, addition of question and answers, marks for each question, Set timer for each quiz and generate report with score for each quiz.

Table of Comparison :

Parameters	Existing System	Proposed System
	Manual	Automatic
	More time	Less time

	consuming	consuming
Database	Required	Not required
	Less	More

Comparison of Existing System & Proposed System

Software Requirements:

Operating System: Android

Toolkit: Software Development Toolkit(SDK)

Platform : Java and Android

Database: SQLite

Hardware Requirements:

2.3 minimum version (API 8)

Android phone (Having Version Above or 2.3)

DATA DICTIONARY

A set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.

Data dictionary for online quiz application

Table: user info

Purpose: To store candidate appears for the quiz

Field name	Data type	description
Id	Integer	Primary key,autoIncrement
Name	varchar	Name of the user
Password	varchar	User password
ExamCode	varchar	Exam code

Table: Exam Info

Purpose: To store the the test info

Field name	Data type	description
ExamCode	varchar	Primary key, auto-Increment

Date	varchar	date of exam
minutes	Int	Duration in minutes
NoQuestion	int	Number of questions

Table: Test Session

Purpose: To store which time the exam takes place

Field name	Data type	description
id	integer	Primary key, auto-increment
TestID	integer	Foreign:Test.ID
studentID	integer	Foreign:student.ID
TestTakenOn	Date/Time	Field required

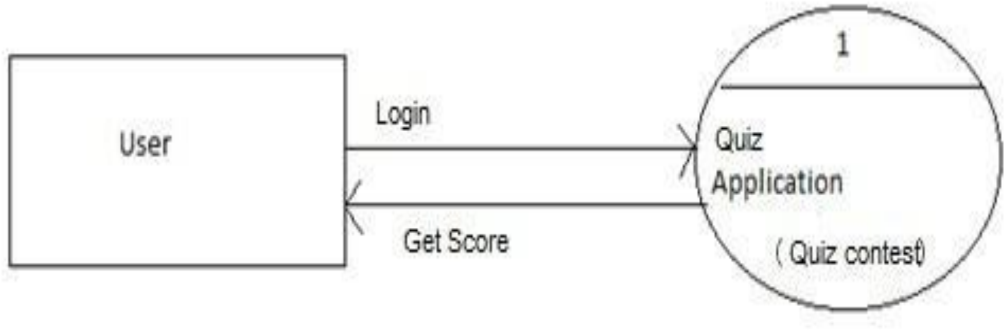
Diagram:

Data Flow Diagram (DFD):

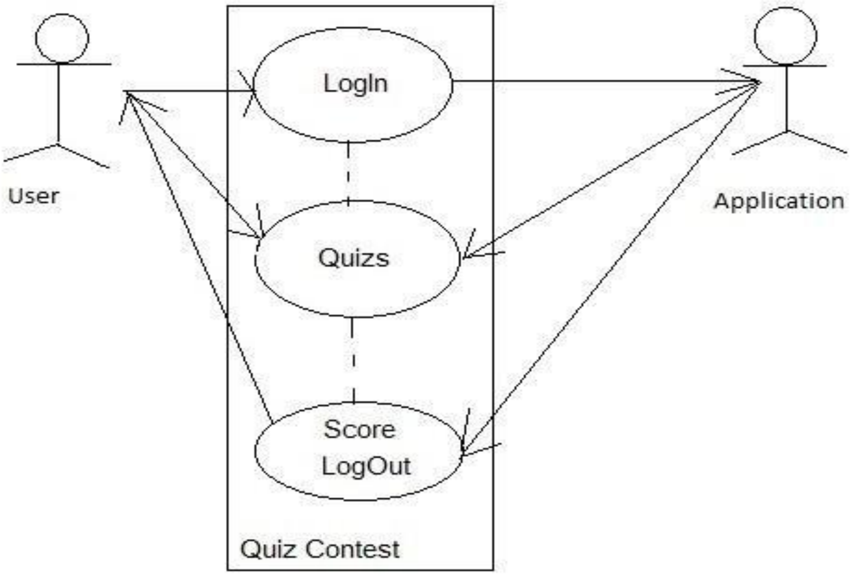
A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modelling its *process* aspects. A DFD is often used as a preliminary step to create an overview of the system. DFDs can also be used for the visualization of data processing.

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

Level 0 DFD:



Use Case Diagram:



Entity-Relationship Diagram:

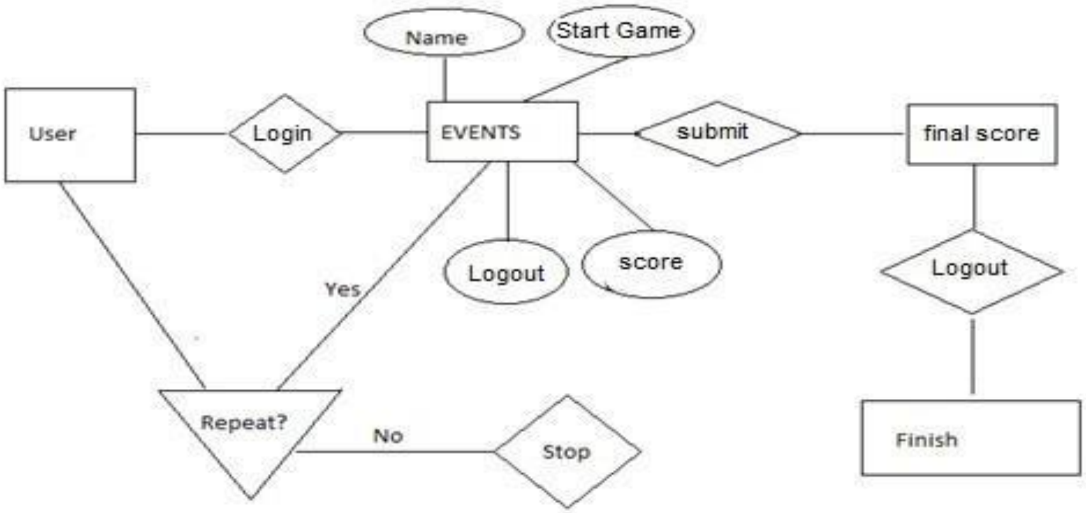


Table Diagram:

Design of database table which is named as Events is given below -

Attribute Name	Attribute type
Id	Int
Name	Text
Info	Text
Period	Text
Period_unit	Text
Start_time	Text

Form Design:

Components-

- Linear Layout (Vertical)
- Linear Layout (Horizontal)

- TextView(Medium)
- EditText
- Button
- CheckBox
- ListVie
- Spinner
- ImageView
- AlertDialog

SQL JOINS

The SQL **Joins** clause is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each.

Consider the following two tables −

Table 1 − USER Table

```
+----+----------+-----+-----------+

| ID | USER NAME| AGE | PASSWORD  |

+----+----------+-----+-----------+

|  1 | Ramesh   |  32 | Ahmedabad |

|  2 | Khilan   |  25 | Delhi     |

|  3 | kaushik  |  23 | Kota      |
|  4 | Chaitali |  25 | Mumbai    |
|  5 | Hardik   |  27 | Bhopal    |
|  6 | Komal    |  22 | MP        |
|  7 | Muffy    |  24 | Indore    |
+----+----------+-----+-----------+
```

Now, let us join these two tables in our SELECT statement as shown below.

```
SQL> SELECT ID,USER NAME, AGE,
   FROM USER, ORDERS
   WHERE  CUSTOMERS.ID = 2,3,4
```

This would produce the following result.

```
+----+----------+-----+
| ID | NAME     | AGE |
+----+----------+-----+
|  3 | kaushik  |  23 |
|  2 | Khilan   |  25 |
|  4 | Chaitali |  25 |
+----+----------+-----+
```

Here, it is noticeable that the join is performed in the WHERE clause. Several operators can be used to join tables, such as =, <, >, <>, <=, >=,

!=, BETWEEN, LIKE, and NOT; they can all be used to join tables. However, the most common operator is the equal to symbol.

There are different types of joins available in SQL −

- INNER JOIN − returns rows when there is a match in both tables.
- LEFT JOIN − returns all rows from the left table, even if there are no matches in the right table.
- RIGHT JOIN − returns all rows from the right table, even if there are no matches in the left table.
- FULL JOIN − returns rows when there is a match in one of the tables.
- SELF JOIN − is used to join a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement.
- CARTESIAN JOIN − returns the Cartesian product of the sets of records from the two or more joined tables.

Code Implementation

MainActivity.java

```
package com.example.suhanshu.kvz;

import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.DialogInterface;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.example.suhanshu.kvz.BroadcastReceiver.AlarmReceiver;
import com.example.suhanshu.kvz.Model.User;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.rengwuxian.materialedittext.MaterialEditText;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity {
    MaterialEditText editNewUser, editNewPassword, editNewEmail;
    MaterialEditText editUser, editPassword;
    Button btnSignUp, btnSignIn;
    DatabaseReference users;
    DatabaseReference category;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnSignIn = findViewById(R.id.sign_in);
        btnSignUp = findViewById(R.id.sign_up);
        editUser = findViewById(R.id.Username);
        editPassword = findViewById(R.id.password);
        users = FirebaseDatabase.getInstance().getReference("Users");
        btnSignUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showDialog();
            }
        });

        registerAlarm();
        btnSignIn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                users.addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                        if
(dataSnapshot.child(editUser.getText().toString()).exists()) {
                            User user =
dataSnapshot.child(editUser.getText().toString()).getValue(User.class);
                            if
(user.getPassword().equals(editPassword.getText().toString())) {
                                Intent intent = new Intent(MainActivity.this,
Home.class);
                                Common.currentUser=user;
                                startActivity(intent);
                                finish();
                            } else
                                Toast.makeText(MainActivity.this, "Password Wrong",
Toast.LENGTH_SHORT).show();
                        } else
                            Toast.makeText(MainActivity.this, "Please Register",
Toast.LENGTH_SHORT).show();
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError databaseError) {

                    }
                });
            }
        });

    }

    private void registerAlarm() {
        Calendar calendar=Calendar.getInstance();
        calendar.set(Calendar.HOUR_OF_DAY,0);// 9 hour
        calendar.set(Calendar.MINUTE,47);
        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.INTERV
AL_DAY,pendingIntent);
    }

    private void showDialog() {
        AlertDialog.Builder alertdialog = new AlertDialog.Builder(MainActivity.this);
        alertdialog.setTitle("Sign Up");
        alertdialog.setMessage("Please fill the credentials");
        LayoutInflater layoutInflater = this.getLayoutInflater();
        View view = layoutInflater.inflate(R.layout.signup, null);
        editNewEmail = view.findViewById(R.id.newemail);
        editNewUser = view.findViewById(R.id.newUsername);
        editNewPassword = view.findViewById(R.id.newpassword);
        alertdialog.setView(view);
        alertdialog.setIcon(R.drawable.ic_account_circle_black_24dp);
        alertdialog.setNegativeButton("No", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });
        alertdialog.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {

                final User user = new User(editNewUser.getText().toString()
                        , editNewPassword.getText().toString()
                        , editNewEmail.getText().toString());

                users.addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                        if (dataSnapshot.child(user.getUserName()).exists()) {
                            Toast.makeText(MainActivity.this, "User Already Exist",
Toast.LENGTH_SHORT).show();
                        } else {
                            users.child(user.getUserName())
                                    .setValue(user);
                            Toast.makeText(MainActivity.this, "Registration
Successfully", Toast.LENGTH_SHORT).show();

                        }
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError databaseError) {

                    }
                });
                dialog.dismiss();
            }
        });
        alertdialog.show();

    }
}
```

Start.java

```
package com.example.suhanshu.kvz;

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

import com.example.suhanshu.kvz.Model.Question;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.Collections;

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

    private void loadQuestion(String categoryId) {
        // first clear list if have old questions
        if (Common.list_question.size() > 0) {
            Common.list_question.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.list_question.add(ques);
                        }
                    }

                    @Override
                    public void onCancelled(DatabaseError databaseError) {

                    }
                });
        Collections.shuffle(Common.list_question);
    }
}
```

MainActivity.xml

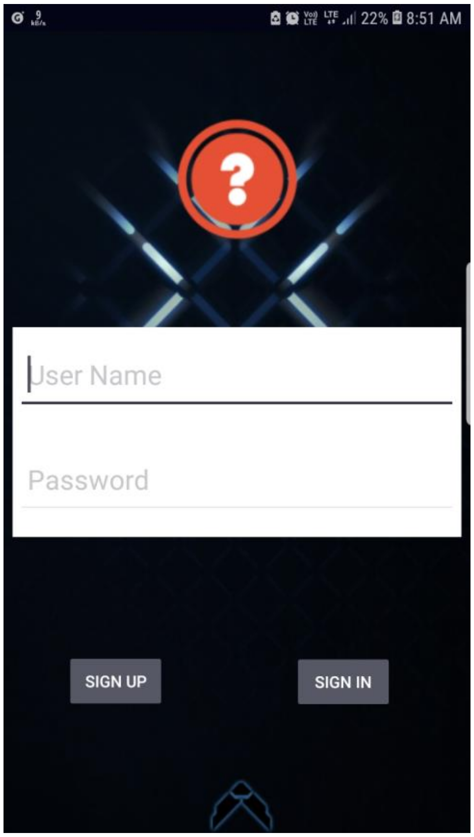
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/wall1"
    tools:context=".MainActivity">

    <ImageView
        android:layout_width="159dp"
        android:layout_height="116dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:src="@drawable/logo"
        app:layout_constraintBottom_toTopOf="@+id/linearLayout"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/sign_in"
        style="@style/Base.Widget.AppCompat.Button.Colored"
        android:layout_width="wrap_content"
        android:layout_height="52dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="@string/sign_in"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.805"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.845" />

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/white"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.503"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <com.rengwuxian.materialedittext.MaterialEditText
            android:id="@+id/Username"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:layout_marginRight="8dp"
            android:hint="@string/user_name"
            android:padding="5dp"
            android:textColor="@color/colorPrimary"
            android:textColorHint="@color/colorPrimary"
            android:textSize="24sp"
            app:met_baseColor="@color/colorPrimary"
            app:met_floatingLabel="highlight"
            app:met_primaryColor="@color/colorPrimary"
            app:met_singleLineEllipsis="true" />

        <com.rengwuxian.materialedittext.MaterialEditText
            android:id="@+id/password"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="8dp"
            android:layout_marginRight="8dp"
            android:hint="@string/password"
            android:padding="5dp"
            android:textColor="@color/colorPrimary"
            android:textColorHint="@color/colorPrimary"
            android:textSize="24sp"
            app:met_baseColor="@color/colorPrimary"
            app:met_floatingLabel="highlight"
            app:met_primaryColor="@color/colorPrimary"
            app:met_singleLineEllipsis="true" />


    </LinearLayout>

    <Button
        android:id="@+id/sign_up"
        style="@style/Base.Widget.AppCompat.Button.Colored"
        android:layout_width="wrap_content"
        android:layout_height="52dp"
        android:layout_marginBottom="8dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="@string/sign_up"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.153"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.844" />

</android.support.constraint.ConstraintLayout>
```

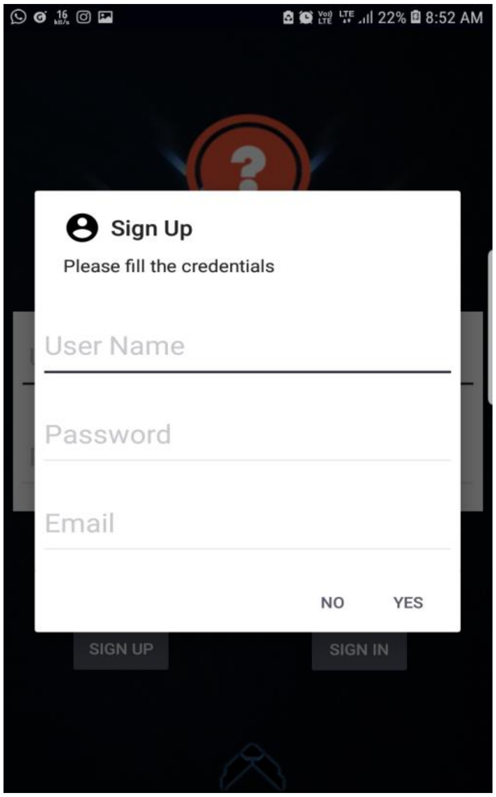
AndroidManifest.xml

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

    <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=".BroadcastReceiver.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=".Services.MyFirebaseIdService">
            <intent-filter>
```

```
                <action android:name="com.google.firebase.INSTANCE_ID_EVENT" />
            </intent-filter>
        </service>
        <service android:name=".Services.MyFirebaseMessagingService">
            <intent-filter>
                <action android:name="com.google.firebase.MESSAGING_EVENT" />
            </intent-filter>
        </service>
    </application>

</manifest>
```

Screenshots





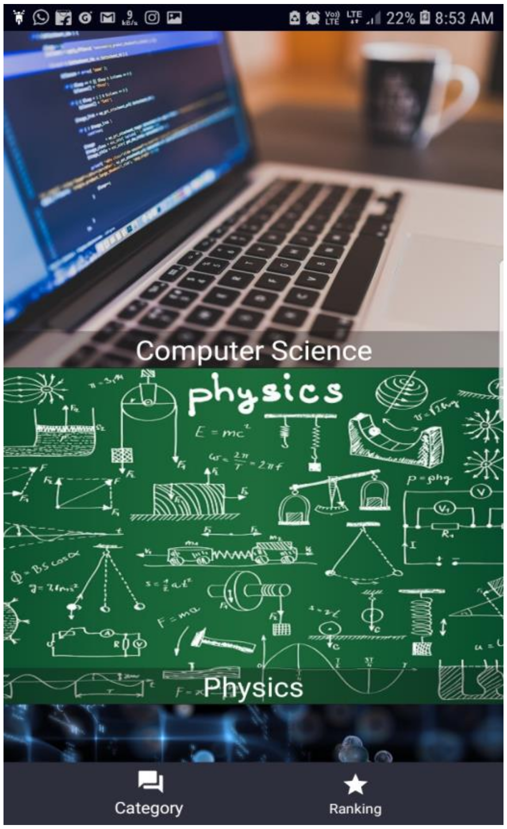
Sign Up

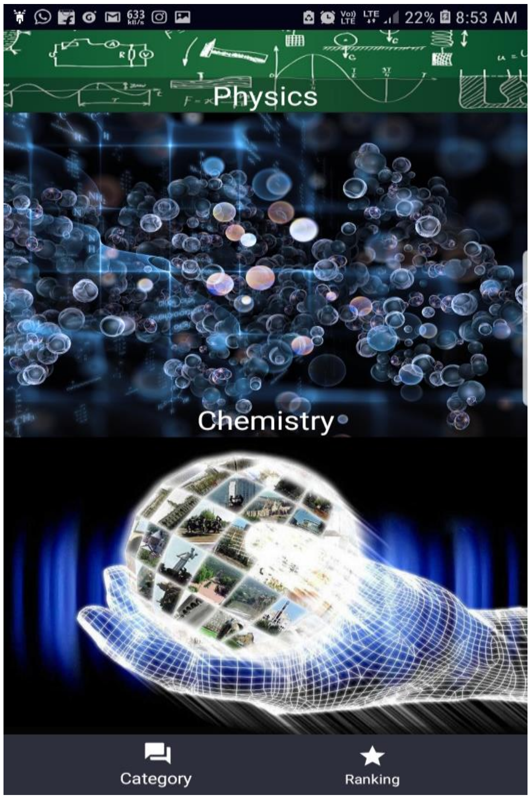
Please fill the credentials

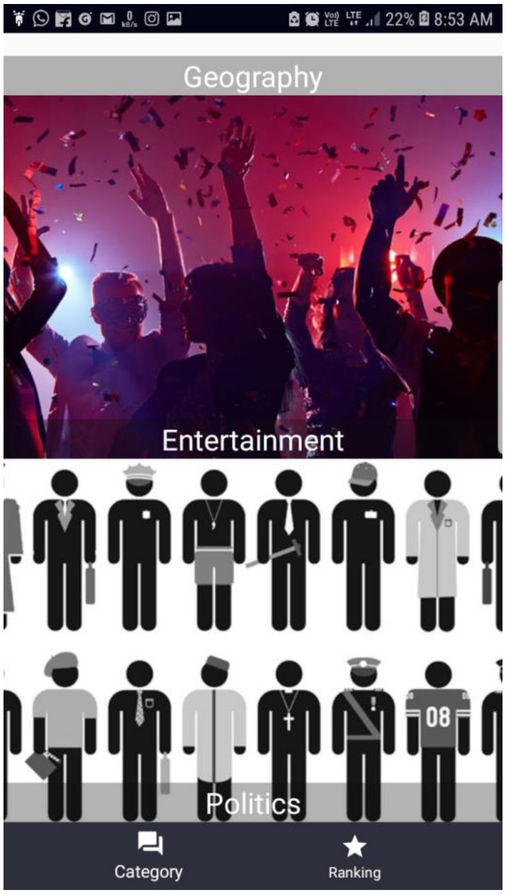
User Name

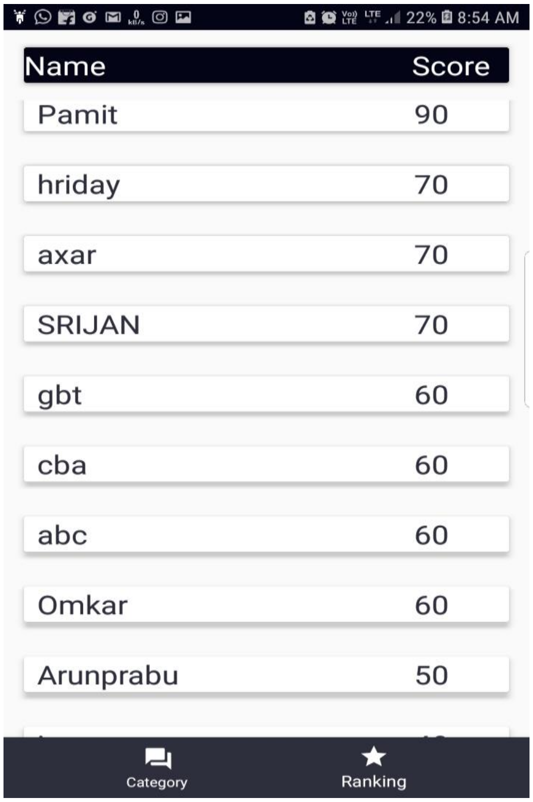
Password

Email




22% 8:53 AM
Physics
Chemistry
Category
Ranking


22% 8:53 AM
Geography
Entertainment
Politics
Category
Ranking



Name	Score
Pamit	90
hriday	70
axar	70
SRIJAN	70
gbt	60
cba	60
abc	60
Omkar	60
Arunprabu	50

Conclusion

Currently there are many medication reminder systems which are operable manually. Due to increased manual work, the available system becomes more time consuming. So in the given work, an attempt has been made to implement fully automatic medication reminder system. It eases the user's task of recalling when to take the medicine by reminding them of the particular medicine at the correct time thereby reducing the much prevalent manual work.

Bibliography

- www.google.com
- www.stackoverflow.com
- www.scripbd.com
- www.slideshare.com
- www.devloper.android.com
- https://www.javatpoint.com/java-mail-api-tutorial
- https://stacktips.com/tutorials/android/speech-to-text-in-android