GALGOTIAS UNIVERSITY

Project

FITNESS APP

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Yash Srivastava (18SCSE1010472)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIA UNIVERSITY

GREATER NOIDA- 201306

TABLE OF CONTENTS

H

Abstract

The emerging popularity of so-called " Fitness app" (mobile applications designed to assist users in pursuing a healthy lifestyle by encouraging them to make positive lifestyle decisions) has presented an interesting challenge to mobile application developers. Our application incorporates step and sleep tracking algorithms. In addition, the application tracks the user's mood throughout the day, and, using this data, the user can monitor the correlation between his or her workout,exercise,weight track,body track. In this project a Fitness App for the Android platform, Fitness App, is developed and tested to track these factors.

Introduction

Since the emergence and popularization of smartphones, many mobile applications that track and record data about their users have been created. The classic example of this is the pedometer which utilizes the mobile device's built-in accelerometer to track the number of steps the user takes each day. Applications in this category, that track and record health or activity data about their users, are typically called Wellness or Fitness Apps. These Wellness Apps are designed to assist the user in pursuing a healthy lifestyle by encouraging them to perform positive activities, and improve lifestyle choices. Factors that are typically targeted by such applications include exercise, sleep, and diet. Understanding the nature of this relationship is crucial when designing a Wellness App. Applications like this have the potential to motivate its users into maintaining a cycle of positive lifestyle decisions and/or breaking a cycle of negative lifestyle decisions. Diet, exercise and sleep can influence several physiological pathways associated with depression and a bidirectional relationship likely exists between depression and these lifestyle factors, thereby creating a potentially increasing cycle of influence .

The goal of this project: The purpose of this project was to create a wellness application for the Android platform capable of tracking, recording, and displaying data relevant to a user’s workout,exercise,weight track,body track. This application also enables individuals to become aware of deficiencies in their everyday habits and will hopefully encourage the user to self- regulate towards improvement. The addition of an avatar to represent the user’s current wellness state, 3D graphics and “gamification” features further promotes user engagement and continued usage of the application. The overall goal is to show the user his or her daily habits and help them to make choices that will result in a healthier lifestyle, and therefore, a happier life. This document will comprehensively describe such an application’s research, design, testing, and development.

Overview

1. What is Android?

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers.

Android was developed by the Open Handset Alliance, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.

On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.

The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

2. Features of Android

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-tasking	User can jump from one task to another and same time various application can run simultaneously.
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.

3. 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.

4. Technology used

Front End

- **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.

5. 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 emulator instances.

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 thedeviceoremulator.

Introduction to App

The name of our application is Fitness App.

This app will have four activities.

1. Authentication
2. Dashboard (main page)
3. Users Data
4. Page for adding new profile
5. Page for deleting data

The goal of fitness tracking apps is to collect data about the user's activities. This includes the number of steps taken, stairs climbed, distance run, and other fitness metrics. To make it easy for users to monitor progress, fitness tracking apps provide calendars and charts and save routes.

1. Objective of the Project :

The ultimate objective of fitness is longevity with no disease,no any physical pain, and happiness. It is hard to get this objectiveof life but not impossible. The happiness of life can be obtainedthrough physical fitness. There are many ways people try get physical fitness, some of them are better lifestyle, better diet, gymyoga, aerobics etcHere are some main positive symptoms of fitness

• zero levels of stress and tension

•Physical strength, stamina and flexibility

•Greater powers of concentration and self control

•Better organ functioning

•Sense of balance and internal harmony

•Healthy & glowing skin

•Strong Immune System etc...

2. Problem Statement :

Do you not feel that the sedentary lifestyle that we lead in today's world has imposed a lot of serious health complications in our lives? Moreover, we also hear people say that the technological advancement has brought a huge change in today's life. Now, if you wish to meet a person, you go for video calling services, instead of meeting him personally over a place.

Thus, it has made the world smaller. And, the techies are becoming more and more engrossed with these new advancements in the field of science and technology. And thereby, neglecting their health widely.

This has created a havoc on the health giving rise to the following disorders. Obesity is certainly one of the most bad consequences.

Also, abnormalities in blood pressure and level of blood sugar, depression, cardio vascular disorders, anxiety, different types of cancer and many more. So, what do you think?

The advancement of technology has really made your life smooth or has it made your life more complicated? Well, the answer is definitely a big NO. No, that the technology hasn't made our life complicated.

Yes, there are some negative impacts on health. But, it has provided a lot of alternatives to outnumber the negative ones. Thanks to the technological development that you can now start exercising at your desk right in your workplace.

Several under desk cycles and elliptical bikes have been developed, which no longer create the problem of restricted blood flow and constant inactivity even if you sit for more than 8 hours in your workstation.

Also, it's because of the technological advancement, several fitness mobile apps are also developed, which not help in improving the fitness level, but also help in maintaining a healthy life.

That's the reason why we see a steep rise in the trend to build an app for health & fitness

3. Description of Project :

- **Fitness apps provide the nudge in the right direction**

If you really wish to be healthy, all you need is the proper nudge from some authenticated source. This helps you to make certain modifications in your lifestyle and daily habits, if necessary.

And, these fitness app are the best to provide the nudge in the right direction. These keep you motivated and focussed to achieve the desired level of fitness.

- **Get new ideas for your workout regime**

In the past few years the people have become more serious about their health & fitness. That is the reason the number of people who join the gym has increased rapidly.

With that fitness trainers have begun to come up with new workout regimes. But what about those who can't join the gym due to their busy lives? Fitness mobile apps are a perfect solution for those users.

They can learn about new fitness regimes and practice it to gain top-notch results. For example, apps like Body fitness and Jefit suggest a new kind of resistance training workout which is designed specifically for some muscles in the body.

Whereas, apps like Tai Chi and Yoga app provide you with pictures, videos, and instructions for the regime.

- **They create a healthy competition**

Well, the competition is between you and other competitors in this virtual world. The apps nurture the competitive side of your health and fitness and drive you towards achieving it.

The app, Strava helps you to sign up and compete against all other users of the app. It takes all the activities into account and breaks them up in various segments and gives you a winner.

So, there's always that charm and excitement, which isn't only inspiring but motivating as well.

- **Set realistic fitness goals**

Many of us may aspire to have a physique like a supermodel. However, they might not have a proper guidance or a path to achieve it.

Moreover, setting-up an impossible target early on can be harmful for your body and it may demotivate you for the further session.

Mobile fitness apps here play a significant role as they design a regime with realistic targets. They take you to the next level only when you finish the previous one.

By this way these apps help you to sustain your exercise regime which is one of the major challengers for those who are beginners.

- **The apps provide you with challenges based on your body condition**

You heard it right. These fitness tracking apps keep of all your activities, BMR, age and body weight as well. By accumulating all the information it provides you a routine to achieve the maximum outcome from your schedule.

The routine is often quite challenging, which helps you to get your butts off and keep moving to attain them.

4. Features

- User will be able to add new profile.
- User will be able to exports or imports their record.
- User will able to delete the records which are not required.
- Change user personal profile info like picture, name, password.

MODULES

1. Server Integrations

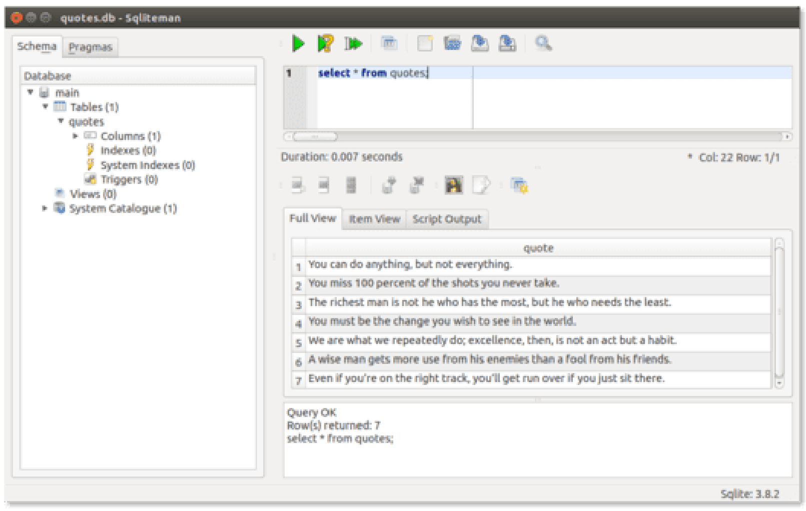
Description:

This module focuses on the integration and connection of server with our app. It has a lot of features such as User Authentication, Databases (Real-Time Database and Cloud Firestore), File Storage, Cloud Functions (A Node. js Environment) and much more. And above all most of the feature are free until a particular threshold and that's the case for large user base.

Design & Architecture

Step 1:

Create a database *quotes.db* using your favorite SQLite database application (DB Browser for SQLite is a portable cross platform freeware, which can be used to create and edit SQLite databases). Create a table '*quotes*' with a single column '*quote*'. Insert some random quotes into the table '*quotes*'.



```
select * from quotes;
```

Duration: 0.007 seconds   Col: 22 Row: 1/1

	quote
1	You can do anything, but not everything.
2	You miss 100 percent of the shots you never take.
3	The richest man is not he who has the most, but he who needs the least.
4	You must be the change you wish to see in the world.
5	We are what we repeatedly do; excellence, then, is not an act but a habit.
6	A wise man gets more use from his enemies than a fool from his friends.
7	Even if you're on the right track, you'll get run over if you just sit there.

```
Query OK
Row(s) returned: 7
select * from quotes;
```

Step 2:

The database can be imported into project either directly as it is, or as a compressed file. The compressed file is recommended, if your database is too large in size. You can create either a *ZIP* compression or a *GZ* compression.

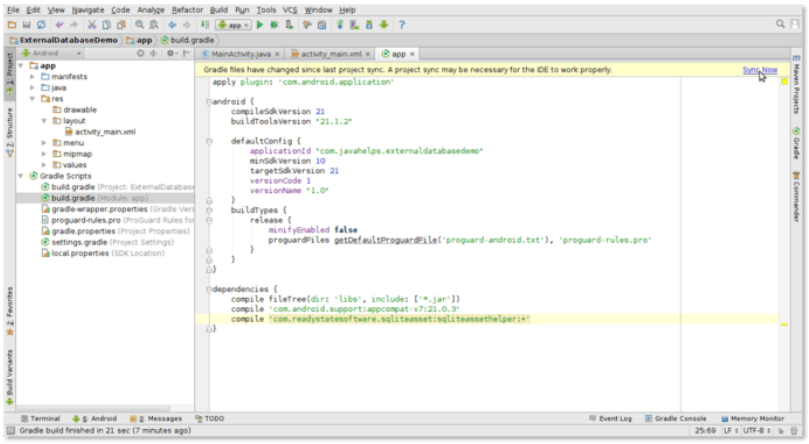
The file name of the compressed db file must be *quotes.db.zip*, if you are using ZIP compression or *quotes.db.gz*, if you are using GZ compression.

Step 3:

Create a new application "*External Database Demo*" with a package name "*com.javahelps.com.javahelps.externaldatabasedemo*".

Step 4:

Open the *build.gradle (Module: app)* file and add the following dependency.



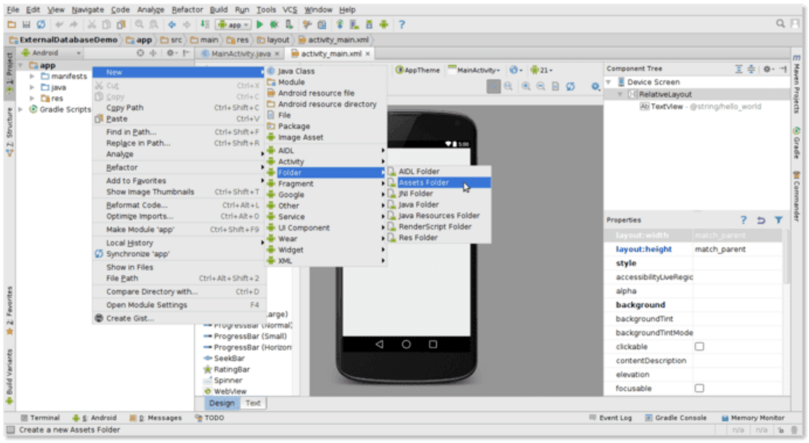
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly. Sync Now

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.javahelps.externaldatabasedemo"
        minSdkVersion 10
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
    compile 'com.readystatesoftware.sqliteasset:sqliteassethelper:+'
}
```

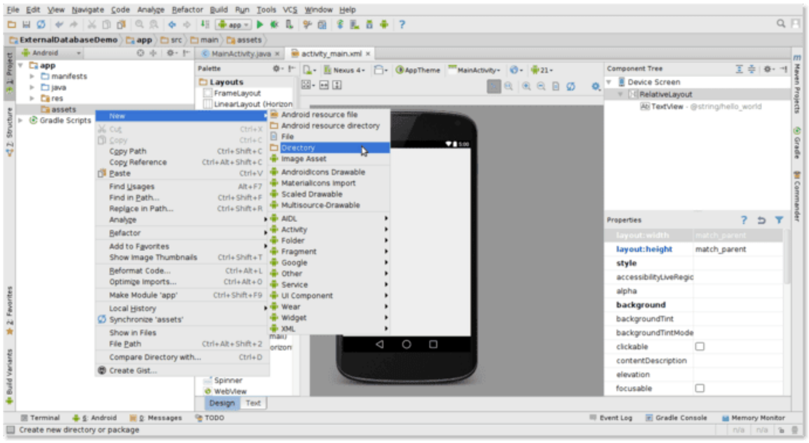
Gradle build finished in 21 sec (7 minutes ago)

Once you have saved the *build.gradle* file click on the '*Sync Now*' link to update the project. You can synchronize the *build.gradle*, by right clicking on the *build.gradle* file and selecting "*Synchronize build.gradle*' option as well.

Step 5:
Right click on the *app* folder and create new *assets* folder.

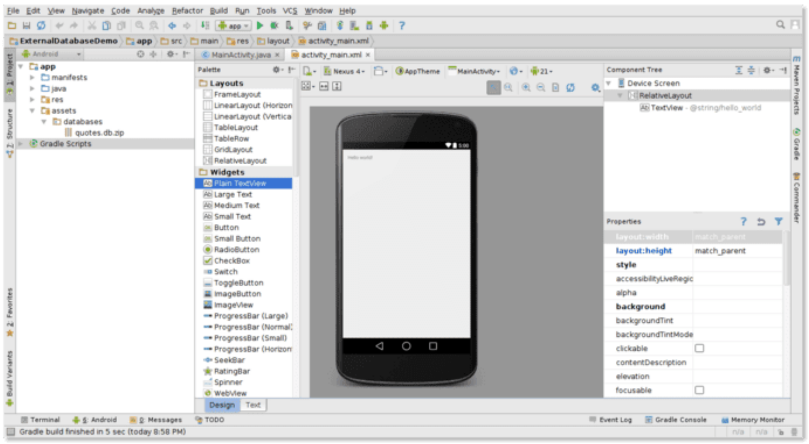


Step 6:
Create a new folder '*databases*' inside the *assets* folder.



Step 7:

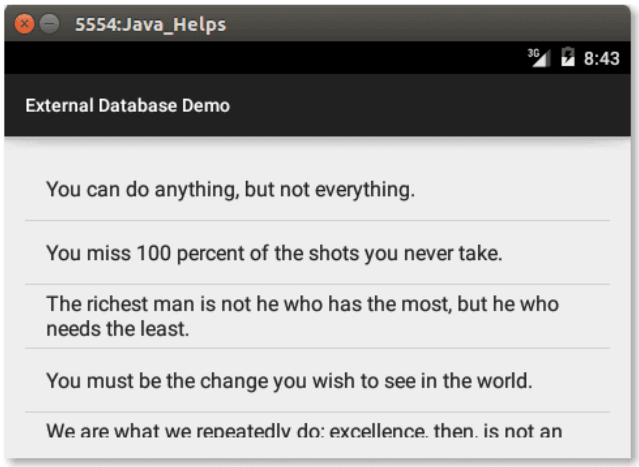
Copy and paste the *quotes.db.zip* file inside the *assets/databases* folder.



Step 8:
Create a new class '*DatabaseOpenHelper*'

Step 9:

Save all the changes and run the application.


5554:Java_Helps

8:43

External Database Demo

You can do anything, but not everything.

You miss 100 percent of the shots you never take.

The richest man is not he who has the most, but he who needs the least.

You must be the change you wish to see in the world.

We are what we repeatedly do; excellence, then, is not an

Implementation & Source Code

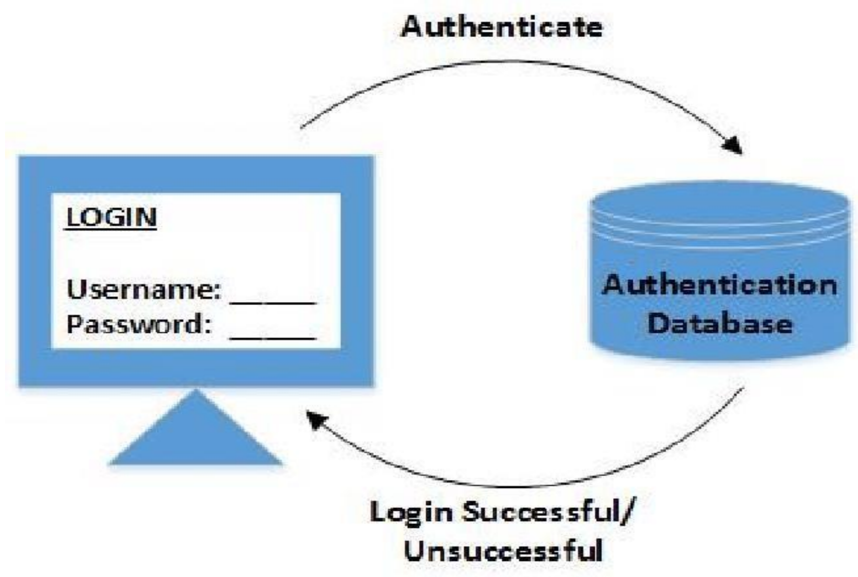
```
dependencies {
    compile 'com.readystatesoftware.sqliteasset:sqliteassethelper:+'
}
package com.javahelps.externaldatabasedemo;

package com.javahelps.externaldatabasedemo;

import android.content.Context;
import com.readystatesoftware.sqliteasset.SQLiteAssetHelper;

public class DatabaseOpenHelper extends SQLiteAssetHelper {
    private static final String DATABASE_NAME = "quotes.db";
    private static final int DATABASE_VERSION = 1;

    public DatabaseOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}
package com.javahelps.externaldatabasedemo;

import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

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

public class DatabaseAccess {
    private SQLiteOpenHelper openHelper;
    private SQLiteDatabase database;
    private static DatabaseAccess instance;

    /**
     * Private constructor to aboid object creation from outside classes.
     *
     * @param context
     */
    private DatabaseAccess(Context context) {
        this.openHelper = new DatabaseOpenHelper(context);
    }

    /**
     * Return a singleton instance of DatabaseAccess.
     *
     * @param context the Context
     * @return the instance of DabaseAccess
     */
    public static DatabaseAccess getInstance(Context context) {
        if (instance == null) {
            instance = new DatabaseAccess(context);
        }
        return instance;
    }

    /**
     * Open the database connection.
     */
    public void open() {
        this.database = openHelper.getWritableDatabase();
```

```java
    }

    /**
     * Close the database connection.
     */
    public void close() {
        if (database != null) {
            this.database.close();
        }
    }

    /**
     * Read all quotes from the database.
     *
     * @return a List of quotes
     */
    public List<String> getQuotes() {
        List<String> list = new ArrayList<>();
        Cursor cursor = database.rawQuery("SELECT * FROM quotes", null);
        cursor.moveToFirst();
        while (!cursor.isAfterLast()) {
            list.add(cursor.getString(0));
            cursor.moveToNext();
        }
        cursor.close();
        return list;
    }
}


import android.content.Context;

import com.readystatesoftware.sqliteasset.SQLiteAssetHelper;

public class DatabaseOpenHelper extends SQLiteAssetHelper {
    private static final String DATABASE_NAME = "quotes.db";
    private static final int DATABASE_VERSION = 1;

    public DatabaseOpenHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
}


public class MainActivity extends ActionBarActivity {
    private ListView listView;

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

        this.listView = (ListView) findViewById(R.id.listView);
        DatabaseAccess databaseAccess = DatabaseAccess.getInstance(this);
        databaseAccess.open();
        List<String> quotes = databaseAccess.getQuotes();
        databaseAccess.close();

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, quotes);
        this.listView.setAdapter(adapter);
    }
}
```

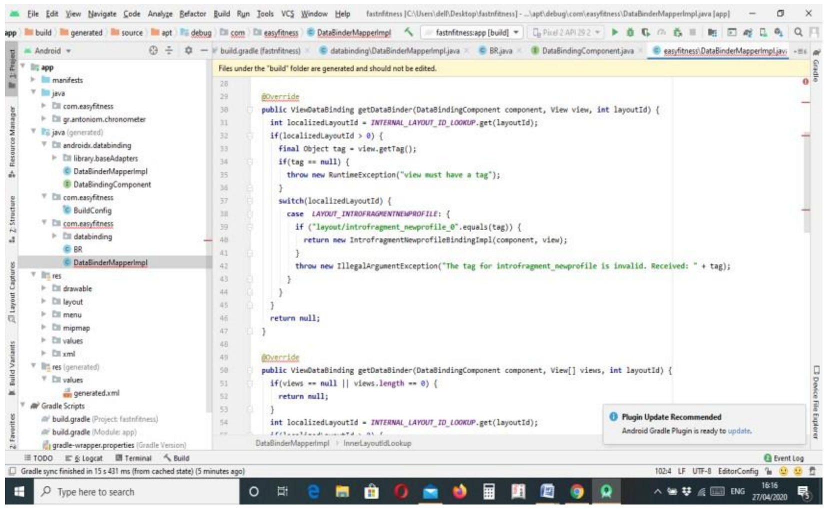
2. User Authentication

Description:
This module focuses on authentication system. Where user will be able to create new account and also update his profile. For this module we will be using Database Authentication which provide lot of feature among which we are only implementing signup through Email and Password.

Design & Architecture



Implementation & Source Code

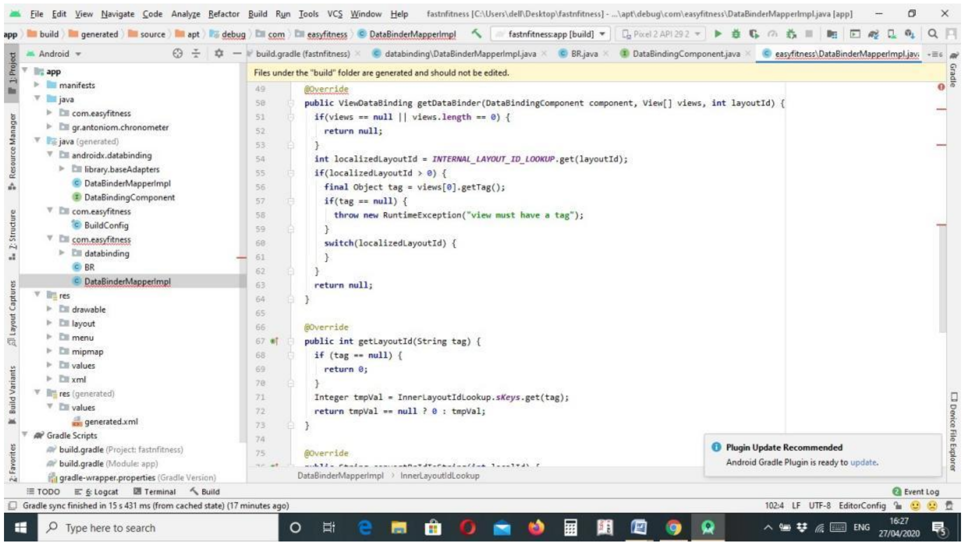

Files under the "build" folder are generated and should not be edited.

```
@Override
public ViewDataBinding getDataBinder(DataBindingComponent component, View view, int layoutId) {
  int localizedLayoutId = INTERNAL_LAYOUT_ID_LOOKUP.get(layoutId);
  if(localizedLayoutId > 0) {
    final Object tag = view.getTag();
    if(tag == null) {
      throw new RuntimeException("view must have a tag");
    }
    switch(localizedLayoutId) {
      case  LAYOUT_INTROFRAGMENTNEWPROFILE: {
        if ("layout/introfragment_newprofile_0".equals(tag)) {
          return new IntrofragmentNewprofileBindingImpl(component, view);
        }
        throw new IllegalArgumentException("The tag for introfragment_newprofile is invalid. Received: " + tag);
      }
    }
  }
  return null;
}

@Override
public ViewDataBinding getDataBinder(DataBindingComponent component, View[] views, int layoutId) {
  if(views == null || views.length == 0) {
    return null;
  }
  int localizedLayoutId = INTERNAL_LAYOUT_ID_LOOKUP.get(layoutId);
```

Plugin Update Recommended
Android Gradle Plugin is ready to update.

Step1: Create a DatabaseAuth object and initialize it by getting instance of it.

Step2: Now call creatUserWith Name() function with Name and password as argument.



Files under the "build" folder are generated and should not be edited.

```java
@Override
public ViewDataBinding getDataBinder(DataBindingComponent component, View[] views, int layoutId) {
  if(views == null || views.length == 0) {
    return null;
  }
  int localizedLayoutId = INTERNAL_LAYOUT_ID_LOOKUP.get(layoutId);
  if(localizedLayoutId > 0) {
    final Object tag = views[0].getTag();
    if(tag == null) {
      throw new RuntimeException("view must have a tag");
    }
    switch(localizedLayoutId) {
    }
  }
  return null;
}

@Override
public int getLayoutId(String tag) {
  if (tag == null) {
    return 0;
  }
  Integer tmpVal = InnerLayoutIdLookup.sKeys.get(tag);
  return tmpVal == null ? 0 : tmpVal;
}

@Override
```

Plugin Update Recommended
Android Gradle Plugin is ready to update.

Gradle sync finished in 15 s 431 ms (from cached state) (17 minutes ago)

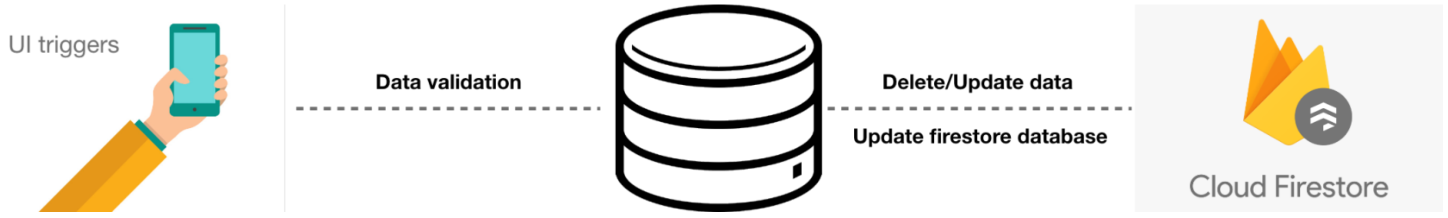
Step3: Now attach onCompleteListner, and update the ui accordingly.

3. Updating and Deleting Profile

Description:

In this module we will be implementing the flawless experience of updating and deleting data, both on front-end and backend server which in our case is Database.

Design & Architecture



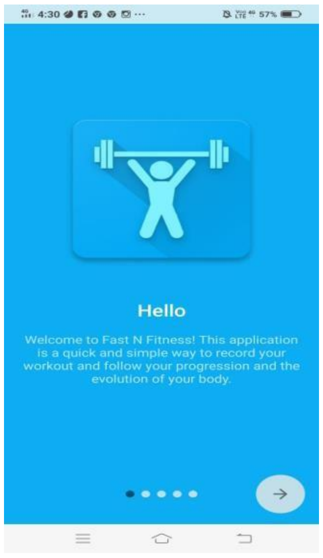
Implementation & Source Code

```
package com.easyfitness.bodymeasures;

import android.content.Context;
import android.database.Cursor;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.CursorAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.cardview.widget.CardView;

import com.easyfitness.BtnClickListener;
import com.easyfitness.R;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;

import static android.text.format.DateFormat.getDateFormat;

public class BodyMeasureCursorAdapter extends CursorAdapter {

    BtnClickListener mDeleteClickListener = null;
    private LayoutInflater mInflater;
    private Context mContext = null;

    public BodyMeasureCursorAdapter(Context context, Cursor c, int flags, BtnClickListener mD) {
        super(context, c, flags);
        mContext = context;
        mDeleteClickListener = mD;
        mInflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }

    @Override
    public void bindView(View view, Context context, Cursor cursor) {
        TextView t0 = view.findViewById(R.id.LIST_BODYMEASURE_ID);
        t0.setText(cursor.getString(0));

        TextView t1 = view.findViewById(R.id.LIST_BODYMEASURE_DATE);
        Date date;
        try {
            SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");
            dateFormat.setTimeZone(TimeZone.getTimeZone("GMT"));
            date = dateFormat.parse(cursor.getString(1));

            //SimpleDateFormat dateFormat2 = new SimpleDateFormat("dd/MM/yyyy");
            //dateFormat2.setTimeZone(TimeZone.getTimeZone("GMT"));
            //t1.setText(DateFormat.getDateInstance().format(date));
            DateFormat dateFormat3 = getDateFormat(mContext.getApplicationContext());
            dateFormat3.setTimeZone(TimeZone.getTimeZone("GMT"));
            t1.setText(dateFormat3.format(date));
        } catch (ParseException e) {
            t1.setText("");
            e.printStackTrace();
        }

        TextView t2 = view.findViewById(R.id.LIST_BODYMEASURE_WEIGHT);
        t2.setText(cursor.getString(3));

        CardView cdView = view.findViewById(R.id.CARDVIEW);

        int mFirstColorOdd = 0;
        if (cursor.getPosition() % 2 == mFirstColorOdd) {
            cdView.setBackgroundColor(context.getResources().getColor(R.color.background_even));
        } else {
            cdView.setBackgroundColor(context.getResources().getColor(R.color.background));
        }

        ImageView deletImg = view.findViewById(R.id.deleteButton);
        deletImg.setTag(cursor.getLong(0));
        deletImg.setOnClickListener(v -> {
            if (mDeleteClickListener != null)
                mDeleteClickListener.onBtnClick((long) v.getTag());
        });

    }

    @Override
    public View newView(Context context, Cursor cursor, ViewGroup parent) {
        return mInflater.inflate(R.layout.bodymeasure_row, parent, false);
    }

}
```

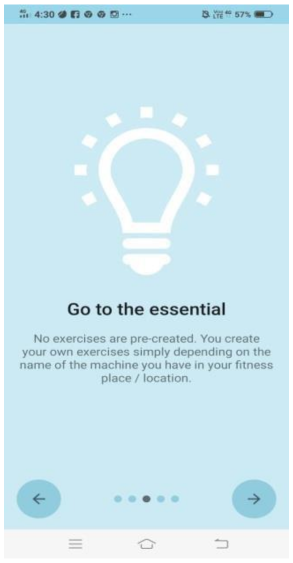
Output & Discussion

Screenshots:

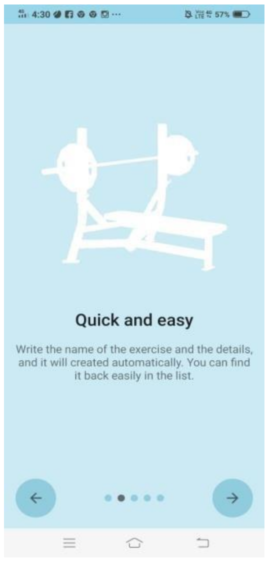
Starting of app……………..


Hello

Welcome to Fast N Fitness! This application is a quick and simple way to record your workout and follow your progression and the evolution of your body.

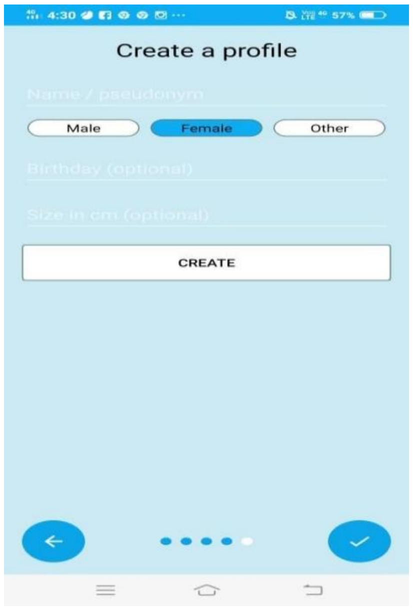

Go to the essential

No exercises are pre-created. You create your own exercises simply depending on the name of the machine you have in your fitness place / location.


Quick and easy

Write the name of the exercise and the details, and it will created automatically. You can find it back easily in the list.

Sign up/Login screen……………


Create a profile

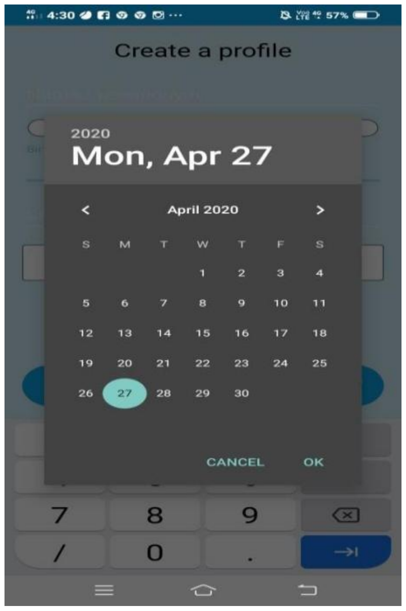
Name / pseudonym

Male [✓ Female] Other

Birthday (optional)

Size in cm (optional)

CREATE


Create a profile

2020

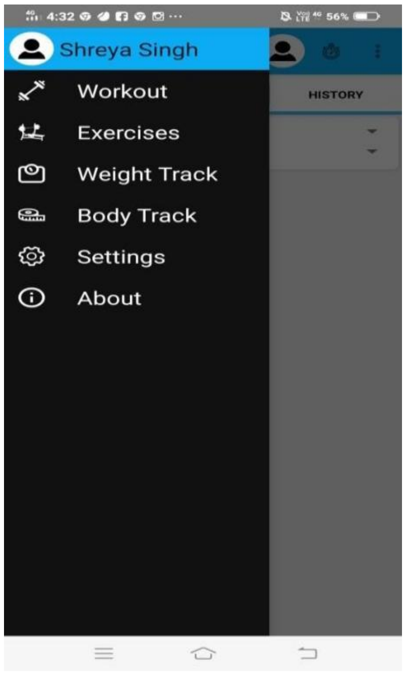
Mon, Apr 27

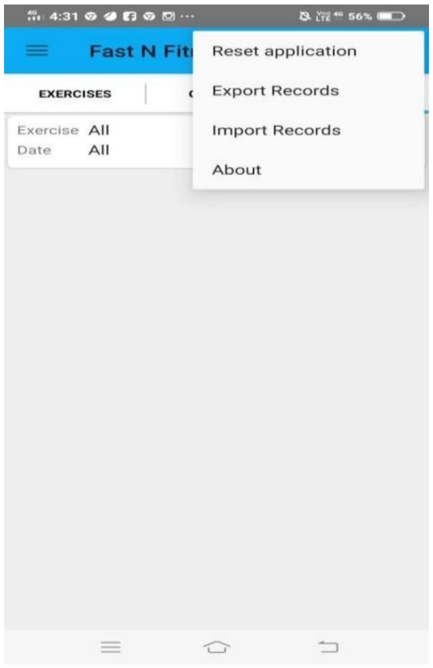
April 2020

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

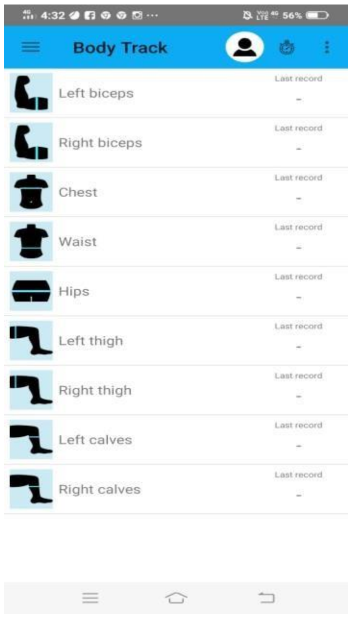
CANCEL OK

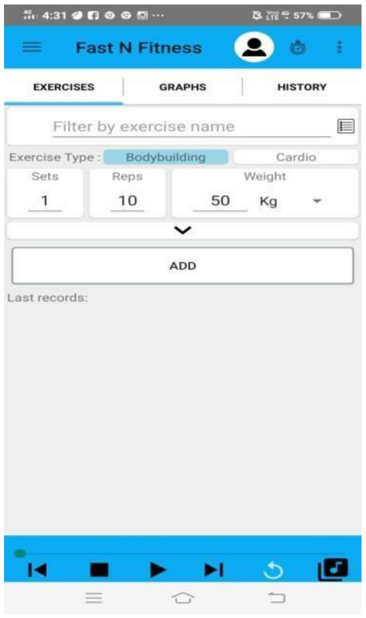
Dashboard……… **Menu for exporting/ importing Records………**





Body track…… **User Records…….**





CONCLUSION

The Android SDK ships with numerous other tools. Many of which are used for special development cases. However, the tools listed above will be used with just about every project, regardless of the type of app being developed. For more information on these and other tools available, check out the Android Tools section of the Android website. Also, new tools and improved tools are released on a fairly regular basis, so make sure you keep all of the packages updated with the AVD and SDK Manager. Finally, above and beyond the Android tools we've discussed, your best resource is the Android Developer website. Complete with up-to-date SDK downloads, source documentation, tutorials, technical articles, and the Android blog with the latest news, this website provides critical knowledge and support for Android developers.

Carrying this project further there are lots of feature which can be integrated in our app.

In the next version of app we will be integrating GPS in our app which will enable user to count their daily steps.