**UNIVERSITY OF MUMBAI**

PROJECT ENTITLED

**“GymDroid”**

SUBMITTED BY

**Mr. Vivek Kamale Ashok**

UNDER GUIDANCE OF

**Mr.Rohan Ghosalkar**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE OF BACHELOR

IN INFORMATION TECHONOLOGY

**PILLAI COLLEGE OF ARTS, COMMERCE &**

**SCIENCE, NEW PANVEL**

**2****017-2018**

**ACKNOWLEDGEMENT**

I **Vivek Kamale Ashok** student of PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE would like to express my sincere gratitude towards Information Technology Department.

I would like to thank lecturer **Rohan Ghosalkar**, and for their constant support during the project. I thank all my colleagues for being with me throughout the project, which leads to a successful completion of my project.

I’d like to thank **Mrs.Rashmi Ghosalkar** for encouraging me and helping me immensely in developing and testing my app.

I shall thank all developers and Youtubers around the world that share their knowledge, work, and wisdom over the Internet.

**PREFACE**

       This project aims at Betterment of fitness for a normal person.This report contains the basic logic used for application development along with the diagrams so that the logic may be apprehended without difficulty. For details information, screen layout can be viewed.

          Although this report is prepared with utmost care, there may be some errors as the project is subjected to further enhancement as per the requirements of the clients.

**INDEX**

|  |  |  |
| --- | --- | --- |
| Chapter Number | Name Of Topics | Page No. |
|  | Task Page | 5 |
| 1. | Introduction  1.1 Organizational Overview  1.2 Why Android  1.3 Limitations Of The Present System  1.4 Proposed System and Its Advantages | 6  7  9  12  12 |
| 2. | System Analysis  2.1 Feasibility study  2.2 Gantt Chart | 13  14  17 |
| 3. | System Design  3.1 System Requirement  3.2 Use Case Diagram  3.3 ERD  3.4 Activity Diagram  3.5 Class Diagram  3.6 Sequence Diagram  3.7Converting ERD To Table | 18  19  21  22  23  25  26  27 |
| 4. | System Coding  4.1 Tables And Views  4.2 Screen Layouts and Codes | 28  29  31 - 85 |
| 6. | Future Enhancements | 86 |
| 8. | Limitations | 88 |
| 9. | Conclusion | 90 |
| 10.. | References and Bibliography | 92 |

|  |  |  |
| --- | --- | --- |
| **Sr No.** | **Task** | **Signature** |
| **1** | SRS Submission |  |
| **2** | Project Flow Charts & Related Diagrams |  |
| **3** | Front End (Font Designing/Web Page Designing/Application Interface Designing) |  |
| **4** | Coding For Related Design |  |
| **5** | Testing |  |
| **6** | Backend Designing |  |
| **7** | Back End Testing |  |
| **8** | Validation |  |
| **9** | Presentation |  |
| **10** | Documentation |  |
|  |  |  |

**INTRODUCTION**

**1.1 ORGANIZATIONAL OVERVIEW**

This project offers systematic and easy way of bettering one’s daily fitness levels by a simple android app.

The objective of the **“GymDroid”** app is to develop a project which helps the people to live a healthy and fit lifestyle.

The project namely **“GymDroid”** is an android application developed in java. The tools being used in the development process are:

* Android Studio
* GenyMotion

**OBJECTIVES OF THE SYSTEM**

* The system should maintain the information about the person.
* It should provide a proper list of basic exercises.
* Provides gym or home based easy exercises.
* Keeping the clients engaged in daily healthy workout routines.
* To provide the reports.

**SCOPE OF THE SYSTEM**

* It maintains the information of the users.
* Any user can use the app.
* It automatically list the registered user’s list.
* It provides all the list of workouts that one can exercise.

**1.2 WHY ANDROID**

**Biggest Addressable Smartphone market**

**According to 2012 survey, android OS accounts for 68.4% of the total IT product market in OS share.**



* **Open Source**

Another great advantage of Android over similar platforms is the fact that Android is open source. Meaning, that no industry player, not even Google, will be able to restrict its development or introduce any changes that would go against your interests. Developers have full access to the phone's functionality, like sending/receiving texts (SMS), using camera or even handling phone calls. And another, probably one of the most important, advantages of being open source is that all the newest revolutionary technologies can be introduced to Android by the various developers who work on it. That's why for Android open source is one of the most important advantages.

* **More Innovative**

iOS powered devices are released usually once or twice a year, as opposed to dozens of Android devices being released all over the world every year. The result of this is new technologies and features being available to the phone users more often, and therefore future releases of my application with enhanced feature sets will be possible.

**1.3 LIMITATION OF PRESENT SYSTEM**

Too many gym based applications have complex workout regimes that can’t be used by a normal beginner. Most apps also have a hectic list of validation process that makes access to real content in a delay.

**1.4 PROPOSED SYSTEM AND ITS ADVANTAGES**

* It will be easy to use.
* Time saving.
* It will eradicate the unhealthy lifestyle of youth.
* No tedious transactions involved.
* Very simple workout regimes.
* It will reduce the manpower.
* It is very helpful in this busy world to stay fit.

**SYSTEM ANALYSIS**

**2.1 FEASIBILITY STUDY**

An important outcome of preliminary investigation is to determine whether the proposed system is feasible or not. A feasibility study is a test of a system proposal according to its work ability, ability to meet users need an effective use of resources. It is conducted to select the best system that meets the performance standards. It reviews the alternative solution and its objective is not solving the problem. Three key consideration are involved in feasibility analysis.

* Economic feasibility
* Technical feasibility
* Operational feasibility

**Economic Feasibility**

It considered the cost/benefit analysis of the proposed project. It also helpful to find out the system development costs. The system is economically feasible.

Economic or financial feasibility is the second part of the resource determination. The effectiveness of the system commonly known as cost benefit analysis, it determining the benefits and savings that are expected from the candidates system. This is an ongoing process that improves accuracy at each phase of the system development life cycle. The basic resources to be considered are as follows:

* Cost of doing entire system study
* Estimate cost of hardware
* Estimated cost of application

As hardware and software resources are already available at quality precision gauges, no additional expenditure to procedure then needs to be incurred.

**Technical Feasibility**

Technical feasibility centres on existing Android device for hardware and software and to what extent it can support the proposed system. The current technical resource available in the organization is capable of handling user requirements

The proposed system is technically feasible to the hotel as its eady with the software necessary for the system. There is no extra cost incurred in the end user training as most of them have knowledge of computers and the system has user friendly interface.

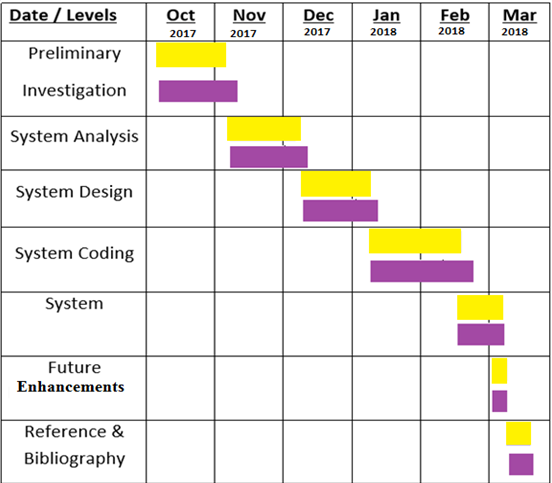
**Operational Feasibility**

It considered the acceptability of the system and checks whethim

* System will be used if it is developed and implemented.
* The user will be able to handle the system easily
* Whether the proposed system will cause any trouble.

The system is completely menu driven. The user friendliness of the system result in simplicity in the operation. The long term benefit of the system would enhance the working style of the operational user.

**2.2 GANTT CHART**

****

**Estimated: **

**Actual: **

**SYSTEM DESIGN**

**3.1 SYSTEM REQUIREMENT**

The application has been designed keeping basic system requirements into consideration related to hardware and software.

**Hardware Requirements**

* Android Phone
* 1 or faster processor
* RAM 1GB
* ROM 1 GB

**Software Requirements**

* Android operating system Lollipop or above is required.
* Android Emulator Users / Genymotion will need to install and test

**Supported Architectures:**

* + - Minimum supported API 19
    - Target API 26 is supported

**Technology Used:**

**Front end**: XML, Java.

**Back end**: SQLite

**XML**

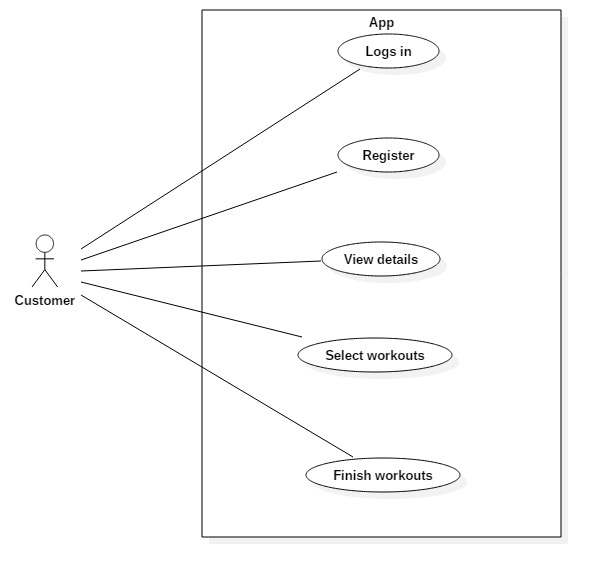
**XML** is an open source mark-up language used to design widgets and Layouts in android.

**Java**

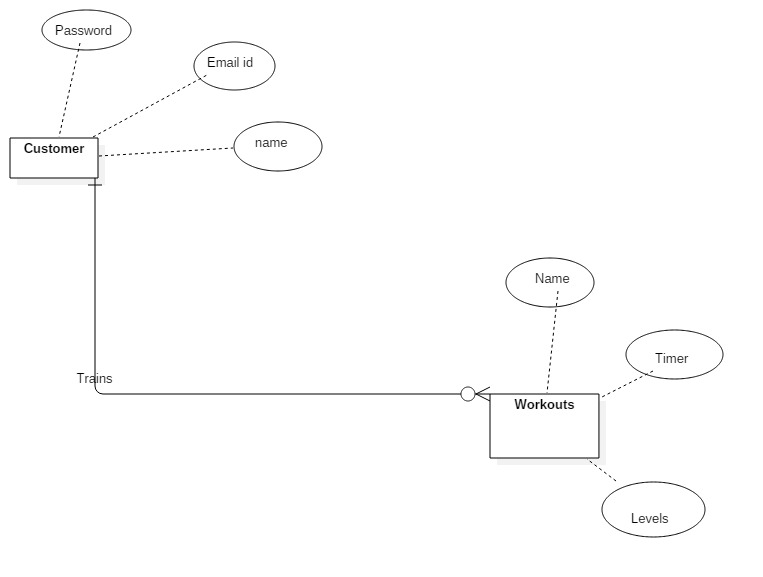
**Java** is an open source high level programing language that I used to write the logic code of android. Widgets and Layouts and Views are also made using java in android programmatically.

**UML Diagrams**

**3.2 USE CASE DIAGRAM**

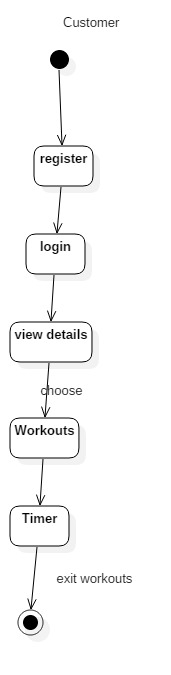
****

**3.3 ER Diagram**

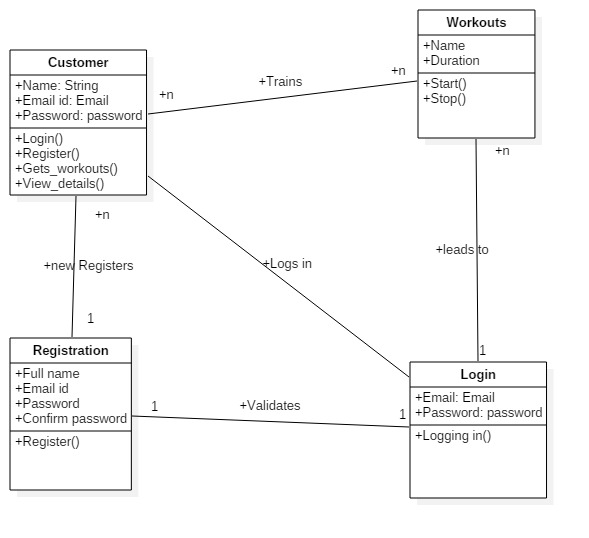
****

**3.4 ACTIVITY DIAGRAM**

**Login Registration Activity**

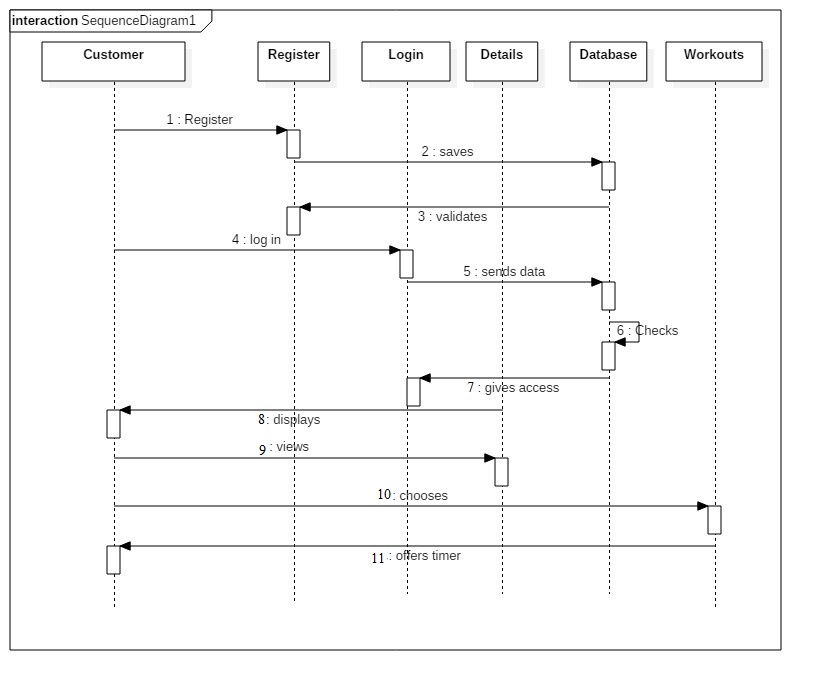
****

**3.5 CLASS DIAGRAM**

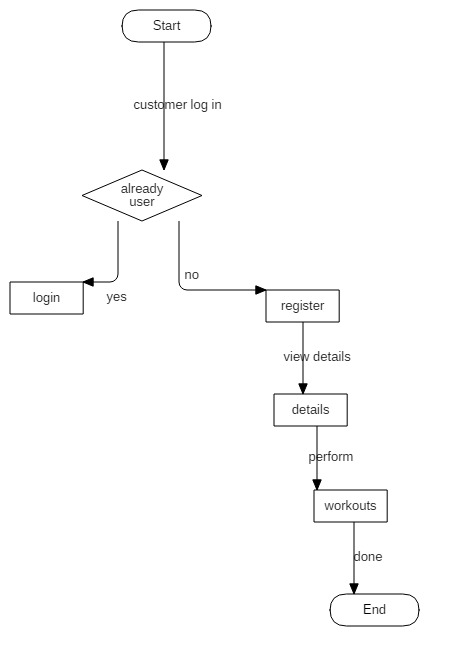


**3.6 SEQUENCE DIAGRAM**

Bill

****

**3.6 Flow chart diagram:**

****

**3.7 Converting ERD to Tables**

|  |  |  |
| --- | --- | --- |
| **SR.NO.** | **TABLE NAME** | **DESCRIPTION** |
| 1. | Customer | Contails customer details |
| 2. | Workouts | It contains list of workouts and timer |
| 3. | Registration/login | It contains register and login data. |

**SYSTEM CODING**

**4.1 LIST OF TABLES AND VIEWS WITH ATTRIBUTES**

**Users Table:**

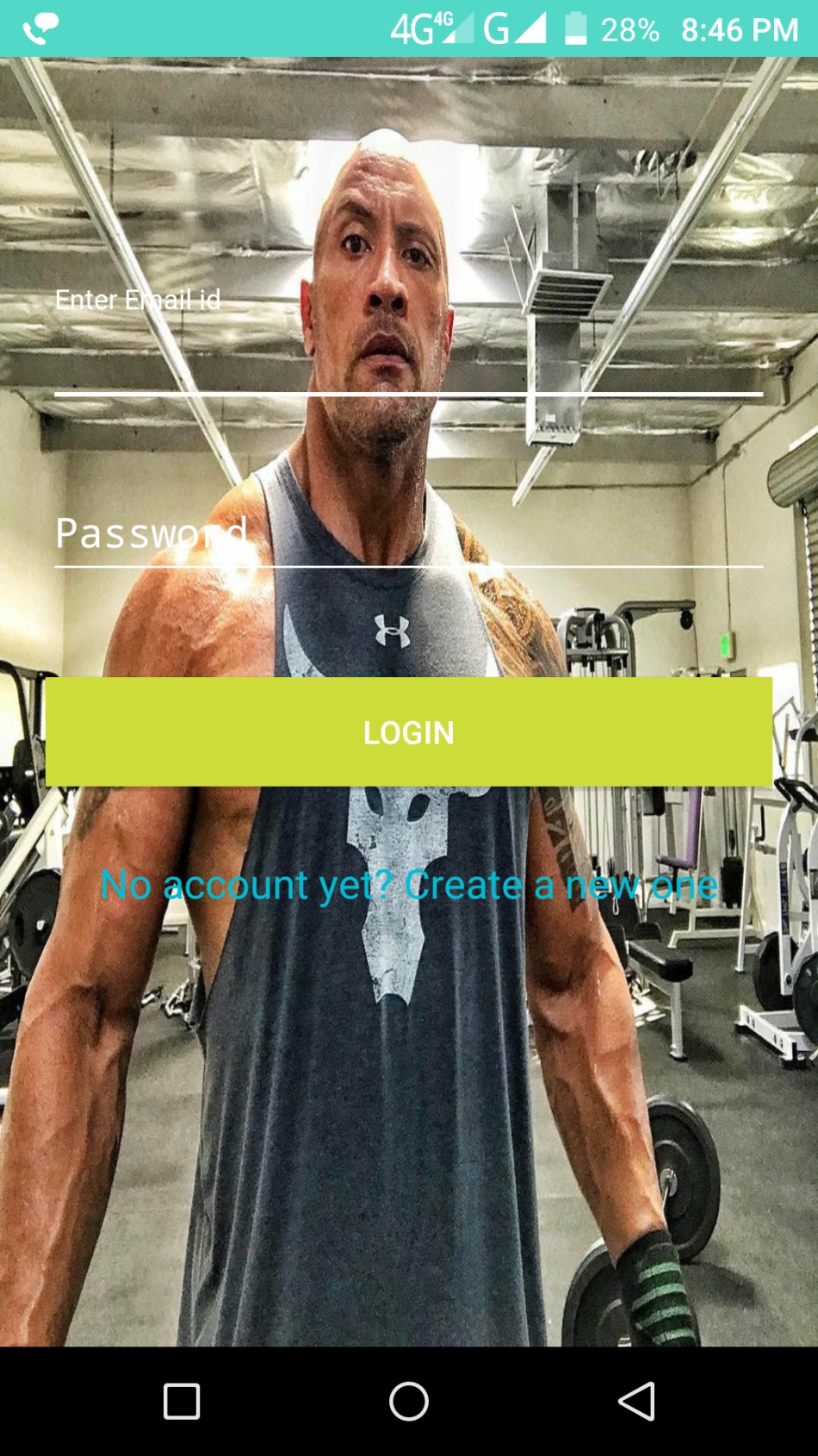
|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Datatype** | **Constraint** |
| Name | varchar(20) | Notnull |
| Email id | Email id | notnull |
| Password | password | Notnull |

**Workout Table**:

|  |  |  |
| --- | --- | --- |
| **Attribute name** | **Datatype** | **Constraint** |
| Name | varchar |  |
| timer | Numeric | 30sec |

**4.2 SCREEN LAYOUTS AND CODE**

* 1. **LoginActivity**

****

**File Name: activity\_login.xml [XML code]**

<?xml version="1.0" encoding="utf-8"?>

<android.support.v4.widget.NestedScrollView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com.tools"

android:id="@+id/nestedScrollView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/people\_eyebrow"

android:paddingBottom="20dp"

android:paddingLeft="20dp"

android:paddingRight="20dp"

android:paddingTop="20dp"

tools:context=".activities.LoginActivity">

<android.support.v7.widget.LinearLayoutCompat

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<android.support.v7.widget.AppCompatImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="40dp">

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

<!-- android:src="@drawable/logo"-->

<android.support.design.widget.TextInputLayout

android:id="@+id/textInputLayoutEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp">

<android.support.design.widget.TextInputEditText

android:id="@+id/textInputEditTextEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_email"

android:inputType="text"

android:maxLines="1"

android:textColor="@android:color/white"/>

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:id="@+id/textInputLayoutPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp">

<android.support.design.widget.TextInputEditText

android:id="@+id/textInputEditTextPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_password"

android:inputType="textPassword"

android:maxLines="1"

android:textColor="@android:color/white"/>

</android.support.design.widget.TextInputLayout>

<android.support.v7.widget.AppCompatButton

android:id="@+id/appCompatButtonLogin"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp"

android:background="#CDDC39"

android:text="@string/text\_login"/>

<android.support.v7.widget.AppCompatTextView

android:id="@+id/textViewLinkRegister"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="30dp"

android:gravity="center"

android:text="@string/text\_not\_member"

android:textColor="#00BCD4"

android:textSize="18dp"/>

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

</android.support.v4.widget.NestedScrollView>

**File name: LoginActivity [Java code]**

package com.example.ram.gymdroid.activities;

import android.content.Intent;

import android.os.Bundle;

import android.support.design.widget.Snackbar;

import android.support.design.widget.TextInputEditText;

import android.support.design.widget.TextInputLayout;

import android.support.v4.widget.NestedScrollView;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.AppCompatButton;

import android.support.v7.widget.AppCompatTextView;

import android.util.Log;

import android.view.View;

import com.example.ram.gymdroid.R;

import com.example.ram.gymdroid.helpers.InputValidation;

import com.example.ram.gymdroid.sql.DatabaseHelper;

public class LoginActivity extends AppCompatActivity implements View.OnClickListener

{

private final AppCompatActivity activity = LoginActivity.this;

private NestedScrollView nestedScrollView;

private TextInputLayout textInputLayoutEmail;

private TextInputLayout textInputLayoutPassword;

private TextInputEditText textInputEditTextEmail;

private TextInputEditText textInputEditTextPassword;

private AppCompatButton appCompatButtonLogin;

private AppCompatTextView textViewLinkRegister;

private InputValidation inputValidation;

private DatabaseHelper databaseHelper;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

getSupportActionBar().hide();

initViews();

initListeners();

initObjects();

}

/\*\*

\* This method is to initialize views

\*/

private void initViews() {

nestedScrollView = (NestedScrollView) findViewById(R.id.nestedScrollView);

textInputLayoutEmail = (TextInputLayout) findViewById(R.id.textInputLayoutEmail);

textInputLayoutPassword = (TextInputLayout) findViewById(R.id.textInputLayoutPassword);

textInputEditTextEmail = (TextInputEditText) findViewById(R.id.textInputEditTextEmail);

textInputEditTextPassword = (TextInputEditText) findViewById(R.id.textInputEditTextPassword);

appCompatButtonLogin = (AppCompatButton) findViewById(R.id.appCompatButtonLogin);

textViewLinkRegister = (AppCompatTextView) findViewById(R.id.textViewLinkRegister);

}

/\*\*

\* This method is to initialize listeners

\*/

private void initListeners() {

appCompatButtonLogin.setOnClickListener(this);

textViewLinkRegister.setOnClickListener(this);

}

/\*\*

\* This method is to initialize objects to be used

\*/

private void initObjects() {

databaseHelper = new DatabaseHelper(activity);

inputValidation = new InputValidation(activity);

}

/\*\*

\* This implemented method is to listen the click on view

\*

\* @param v

\*/

@Override

public void onClick(View v) {

switch (v.getId()) {

case R.id.appCompatButtonLogin:

verifyFromSQLite();

break;

case R.id.textViewLinkRegister:

// Navigate to RegisterActivity

Intent intentRegister = new Intent(getApplicationContext(), RegisterActivity.class);

startActivity(intentRegister);

break;

}

}

/\*\*

\* This method is to validate the input text fields and verify login credentials from SQLite

\*/

private void verifyFromSQLite() {

if (!inputValidation.isInputEditTextFilled(textInputEditTextEmail, textInputLayoutEmail, getString(R.string.error\_message\_email)))

{

return;

}

if (!inputValidation.isInputEditTextEmail(textInputEditTextEmail, textInputLayoutEmail, getString(R.string.error\_message\_email)))

{

return;

}

if (!inputValidation.isInputEditTextFilled(textInputEditTextPassword, textInputLayoutPassword, getString(R.string.error\_message\_email)))

{

return;

}

if (databaseHelper.checkUser(textInputEditTextEmail.getText().toString().trim()

, textInputEditTextPassword.getText().toString().trim())) {

Intent accountsIntent = new Intent(activity, UsersListActivity.class);

accountsIntent.putExtra("EMAIL", textInputEditTextEmail.getText().toString().trim());

emptyInputEditText();

startActivity(accountsIntent);

} else {

// Snack Bar to show success message that record is wrong

Snackbar.make(nestedScrollView, getString(R.string.error\_valid\_email\_password), Snackbar.LENGTH\_LONG).show();

}

}

/\*\*

\* This method is to empty all input edit text

\*/

private void emptyInputEditText() {

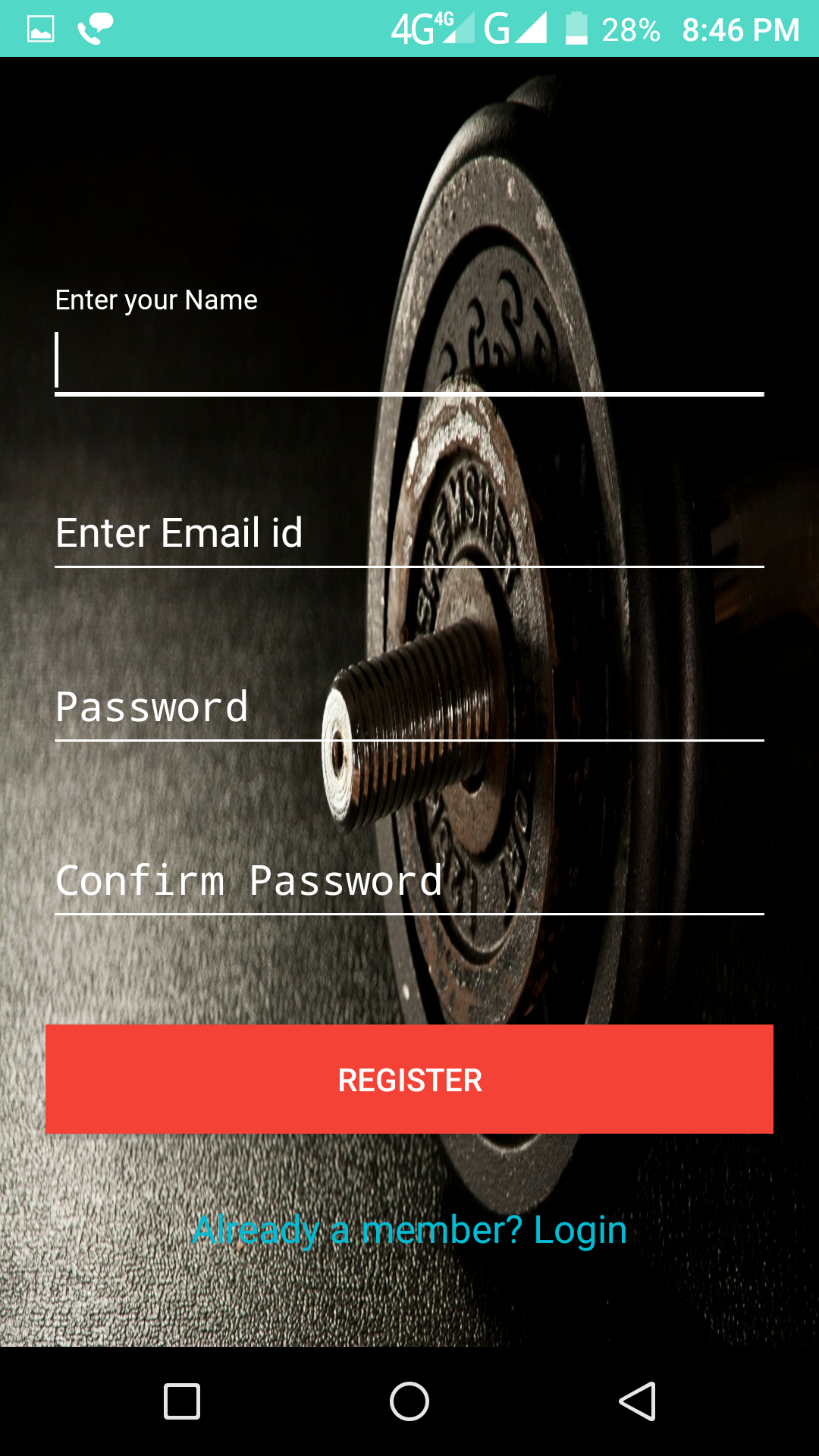
textInputEditTextEmail.setText(null);

textInputEditTextPassword.setText(null);

}

}

**2.RegisterActivity**

****

**File Name: activity\_register.xml [XML code]**

<?xml version="1.0" encoding="utf-8"?>

<android.support.v4.widget.NestedScrollView

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/nestedScrollView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/dumble"

android:paddingBottom="20dp"

android:paddingLeft="20dp"

android:paddingRight="20dp"

android:paddingTop="20dp"

tools:context=".LoginActivity">

<android.support.v7.widget.LinearLayoutCompat

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<android.support.v7.widget.AppCompatImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal"

android:layout\_marginTop="40dp">

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

<!-- android:src="@drawable/logo"-->

<android.support.design.widget.TextInputLayout

android:id="@+id/textInputLayoutName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp">

<android.support.design.widget.TextInputEditText

android:id="@+id/textInputEditTextName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_name"

android:inputType="text"

android:maxLines="1"

android:textColor="@android:color/white"/>

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:id="@+id/textInputLayoutEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp">

<android.support.design.widget.TextInputEditText

android:id="@+id/textInputEditTextEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_email"

android:inputType="text"

android:maxLines="1"

android:textColor="@android:color/white"/>

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:id="@+id/textInputLayoutPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp">

<android.support.design.widget.TextInputEditText

android:id="@+id/textInputEditTextPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_password"

android:inputType="textPassword"

android:maxLines="1"

android:textColor="@android:color/white"/>

</android.support.design.widget.TextInputLayout>

<android.support.design.widget.TextInputLayout

android:id="@+id/textInputLayoutConfirmPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp">

<android.support.design.widget.TextInputEditText

android:id="@+id/textInputEditTextConfirmPassword"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="@string/hint\_confirm\_password"

android:inputType="textPassword"

android:maxLines="1"

android:textColor="@android:color/white"/>

</android.support.design.widget.TextInputLayout>

<android.support.v7.widget.AppCompatButton

android:id="@+id/appCompatButtonRegister"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp"

android:background="#F44336"

android:text="@string/text\_register"/>

<android.support.v7.widget.AppCompatTextView

android:id="@+id/appCompatTextViewLoginLink"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="30dp"

android:gravity="center"

android:text="Already a member? Login"

android:textColor="#00BCD4"

android:textSize="17sp"/>

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

</android.support.v4.widget.NestedScrollView>

**Filename: RegisterActivity.java [Java code]**

package com.example.ram.gymdroid.activities;

/\*\*

\* Created by ram on 04-03-2018.

\*/

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.design.widget.Snackbar;

import android.support.design.widget.TextInputEditText;

import android.support.design.widget.TextInputLayout;

import android.support.v4.widget.NestedScrollView;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.AppCompatButton;

import android.support.v7.widget.AppCompatTextView;

import android.view.View;

import com.example.ram.gymdroid.R;

import com.example.ram.gymdroid.helpers.InputValidation;

import com.example.ram.gymdroid.model.User;

import com.example.ram.gymdroid.sql.DatabaseHelper;

public class RegisterActivity extends AppCompatActivity implements View.OnClickListener

{

private final AppCompatActivity activity = RegisterActivity.this;

private NestedScrollView nestedScrollView;

private TextInputLayout textInputLayoutName;

private TextInputLayout textInputLayoutEmail;

private TextInputLayout textInputLayoutPassword;

private TextInputLayout textInputLayoutConfirmPassword;

private TextInputEditText textInputEditTextName;

private TextInputEditText textInputEditTextEmail;

private TextInputEditText textInputEditTextPassword;

private TextInputEditText textInputEditTextConfirmPassword;

private AppCompatButton appCompatButtonRegister;

private AppCompatTextView appCompatTextViewLoginLink;

private InputValidation inputValidation;

private DatabaseHelper databaseHelper;

private User user;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_register);

getSupportActionBar().hide();

initViews();

initListeners();

initObjects();

}

/\*\*

\* This method is to initialize views

\*/

private void initViews() {

nestedScrollView = (NestedScrollView) findViewById(R.id.nestedScrollView);

textInputLayoutName = (TextInputLayout) findViewById(R.id.textInputLayoutName);

textInputLayoutEmail = (TextInputLayout) findViewById(R.id.textInputLayoutEmail);

textInputLayoutPassword = (TextInputLayout) findViewById(R.id.textInputLayoutPassword);

textInputLayoutConfirmPassword = (TextInputLayout) findViewById(R.id.textInputLayoutConfirmPassword);

textInputEditTextName = (TextInputEditText) findViewById(R.id.textInputEditTextName);

textInputEditTextEmail = (TextInputEditText) findViewById(R.id.textInputEditTextEmail);

textInputEditTextPassword = (TextInputEditText) findViewById(R.id.textInputEditTextPassword);

textInputEditTextConfirmPassword = (TextInputEditText) findViewById(R.id.textInputEditTextConfirmPassword);

appCompatButtonRegister = (AppCompatButton) findViewById(R.id.appCompatButtonRegister);

appCompatTextViewLoginLink = (AppCompatTextView) findViewById(R.id.appCompatTextViewLoginLink);

}

/\*\*

\* This method is to initialize listeners

\*/

private void initListeners() {

appCompatButtonRegister.setOnClickListener(this);

appCompatTextViewLoginLink.setOnClickListener(this);

}

/\*\*

\* This method is to initialize objects to be used

\*/

private void initObjects() {

inputValidation = new InputValidation(activity);

databaseHelper = new DatabaseHelper(activity);

user = new User();

}

/\*\*

\* This implemented method is to listen the click on view

\*

\* @param v

\*/

@Override

public void onClick(View v) {

switch (v.getId()) {

case R.id.appCompatButtonRegister:

postDataToSQLite();

break;

case R.id.appCompatTextViewLoginLink:

finish();

break;

}

}

/\*\*

\* This method is to validate the input text fields and post data to SQLite

\*/

private void postDataToSQLite()

{

if (!inputValidation.isInputEditTextFilled(textInputEditTextName, textInputLayoutName, getString(R.string.error\_message\_name)))

{

return;

}

if (!inputValidation.isInputEditTextFilled(textInputEditTextEmail, textInputLayoutEmail, getString(R.string.error\_message\_email)))

{

return;

}

if (!inputValidation.isInputEditTextEmail(textInputEditTextEmail, textInputLayoutEmail, getString(R.string.error\_message\_email))){

return;

}

if (!inputValidation.isInputEditTextFilled(textInputEditTextPassword, textInputLayoutPassword, getString(R.string.error\_message\_password))){

return;

}

if (!inputValidation.isInputEditTextMatches(textInputEditTextPassword, textInputEditTextConfirmPassword,

textInputLayoutConfirmPassword, getString(R.string.error\_password\_match))) {

return;

}

if (!databaseHelper.checkUser(textInputEditTextEmail.getText().toString().trim()))

{

user.setName(textInputEditTextName.getText().toString().trim());

user.setEmail(textInputEditTextEmail.getText().toString().trim());

user.setPassword(textInputEditTextPassword.getText().toString().trim());

databaseHelper.addUser(user);

// Snack Bar to show success message that record saved successfully

Snackbar.make(nestedScrollView, getString(R.string.success\_message), Snackbar.LENGTH\_LONG).show();

emptyInputEditText();

} else {

// Snack Bar to show error message that record already exists

Snackbar.make(nestedScrollView, getString(R.string.error\_email\_exists), Snackbar.LENGTH\_LONG).show();

}

}

/\*\*

\* This method is to empty all input edit text

\*/

private void emptyInputEditText() {

textInputEditTextName.setText(null);

textInputEditTextEmail.setText(null);

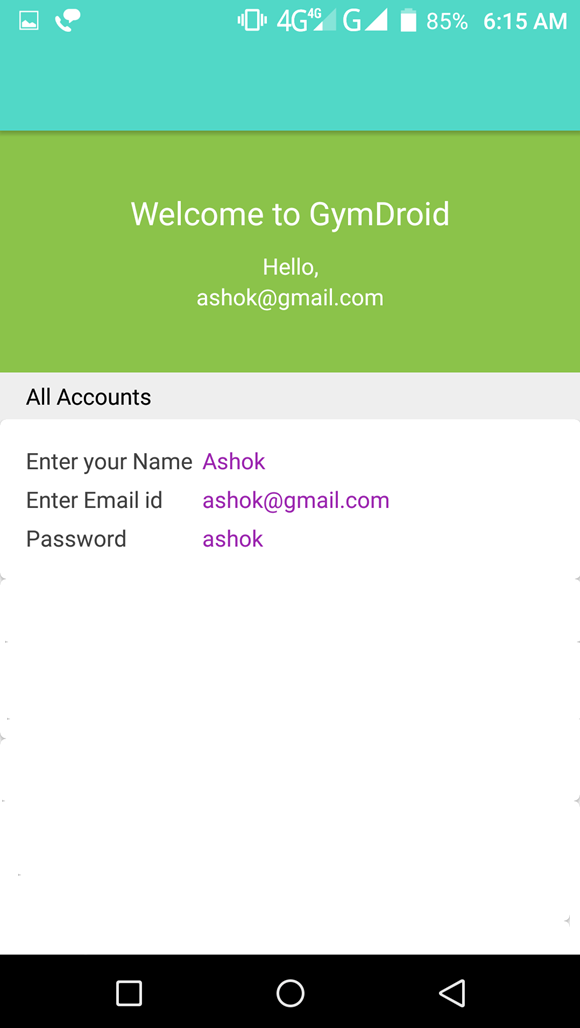
textInputEditTextPassword.setText(null);

textInputEditTextConfirmPassword.setText(null);

}

}

**3] Welcome Activity**

****

**Filename: activity\_users\_list.xml [XML code]**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<android.support.v7.widget.LinearLayoutCompat

android:layout\_width="match\_parent"

android:layout\_height="150dp"

android:background="#8BC34A"

android:gravity="center"

android:orientation="vertical">

<android.support.v7.widget.AppCompatTextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/text\_title"

android:textSize="20sp"/>

<android.support.v7.widget.AppCompatTextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:text="@string/text\_hello"/>

<android.support.v7.widget.AppCompatTextView

android:id="@+id/textViewName"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

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

<android.support.v7.widget.AppCompatTextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:paddingBottom="5dp"

android:paddingLeft="16dp"

android:paddingTop="5dp"

android:text="@string/text\_accounts"

android:textColor="@android:color/black"/>

<android.support.v7.widget.RecyclerView

android:id="@+id/recyclerViewUsers"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

</LinearLayout>

**Filename: UsersListActivity.java [Java code]**

package com.example.ram.gymdroid.activities;

import android.os.AsyncTask;

import android.os.Bundle;

import android.support.annotation.Nullable;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.AppCompatTextView;

import android.support.v7.widget.DefaultItemAnimator;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import com.example.ram.gymdroid.R;

import com.example.ram.gymdroid.adapters.UsersRecyclerAdapter;

import com.example.ram.gymdroid.model.User;

import com.example.ram.gymdroid.sql.DatabaseHelper;

import java.util.ArrayList;

import java.util.List;

public class UsersListActivity extends AppCompatActivity {

private AppCompatActivity activity = UsersListActivity.this;

private AppCompatTextView textViewName;

private RecyclerView recyclerViewUsers;

private List<User> listUsers;

private UsersRecyclerAdapter usersRecyclerAdapter;

private DatabaseHelper databaseHelper;

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_users\_list);

getSupportActionBar().setTitle("");

initViews();

initObjects();

}

/\*\*

\* This method is to initialize views

\*/

private void initViews() {

textViewName = (AppCompatTextView) findViewById(R.id.textViewName);

recyclerViewUsers = (RecyclerView) findViewById(R.id.recyclerViewUsers);

}

/\*\*

\* This method is to initialize objects to be used

\*/

private void initObjects() {

listUsers = new ArrayList<>();

usersRecyclerAdapter = new UsersRecyclerAdapter(listUsers);

RecyclerView.LayoutManager mLayoutManager = new LinearLayoutManager(getApplicationContext());

recyclerViewUsers.setLayoutManager(mLayoutManager);

recyclerViewUsers.setItemAnimator(new DefaultItemAnimator());

recyclerViewUsers.setHasFixedSize(true);

recyclerViewUsers.setAdapter(usersRecyclerAdapter);

databaseHelper = new DatabaseHelper(activity);

String emailFromIntent = getIntent().getStringExtra("EMAIL");

textViewName.setText(emailFromIntent);

getDataFromSQLite();

}

/\*\*

\* This method is to fetch all user records from SQLite

\*/

private void getDataFromSQLite() {

// AsyncTask is used that SQLite operation not blocks the UI Thread.

new AsyncTask<Void, Void, Void>() {

@Override

protected Void doInBackground(Void... params) {

listUsers.clear();

listUsers.addAll(databaseHelper.getAllUser());

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

usersRecyclerAdapter.notifyDataSetChanged();

}

}.execute();

}

}

**4] Input validation**

**Filename: InputValidation.java**

package com.example.ram.gymdroid.helpers;

import android.app.Activity;

import android.content.Context;

import android.support.design.widget.TextInputEditText;

import android.support.design.widget.TextInputLayout;

import android.view.View;

import android.view.WindowManager;

import android.view.inputmethod.InputMethodManager;

public class InputValidation

{

private Context context;

/\*\*

\* constructor

\*

\* @param context

\*/

public InputValidation(Context context) {

this.context = context;

}

/\*\*

\* method to check InputEditText filled .

\*

\* @param textInputEditText

\* @param textInputLayout

\* @param message

\* @return

\*/

public boolean isInputEditTextFilled(TextInputEditText textInputEditText, TextInputLayout textInputLayout, String message) {

String value = textInputEditText.getText().toString().trim();

if (value.isEmpty()) {

textInputLayout.setError(message);

hideKeyboardFrom(textInputEditText);

return false;

} else {

textInputLayout.setErrorEnabled(false);

}

return true;

}

/\*\*

\* method to check InputEditText has valid email .

\*

\* @param textInputEditText

\* @param textInputLayout

\* @param message

\* @return

\*/

public boolean isInputEditTextEmail(TextInputEditText textInputEditText, TextInputLayout textInputLayout, String message)

{

String value = textInputEditText.getText().toString().trim();

if (value.isEmpty() || !android.util.Patterns.EMAIL\_ADDRESS.matcher(value).matches()) {

textInputLayout.setError(message);

hideKeyboardFrom(textInputEditText);

return false;

} else {

textInputLayout.setErrorEnabled(false);

}

return true;

}

public boolean isInputEditTextMatches(TextInputEditText textInputEditText1, TextInputEditText textInputEditText2, TextInputLayout textInputLayout, String message)

{

String value1 = textInputEditText1.getText().toString().trim();

String value2 = textInputEditText2.getText().toString().trim();

if (!value1.contentEquals(value2)) {

textInputLayout.setError(message);

hideKeyboardFrom(textInputEditText2);

return false;

} else {

textInputLayout.setErrorEnabled(false);

}

return true;

}

/\*\*

\* method to Hide keyboard

\*

\* @param view

\*/

private void hideKeyboardFrom(View view) {

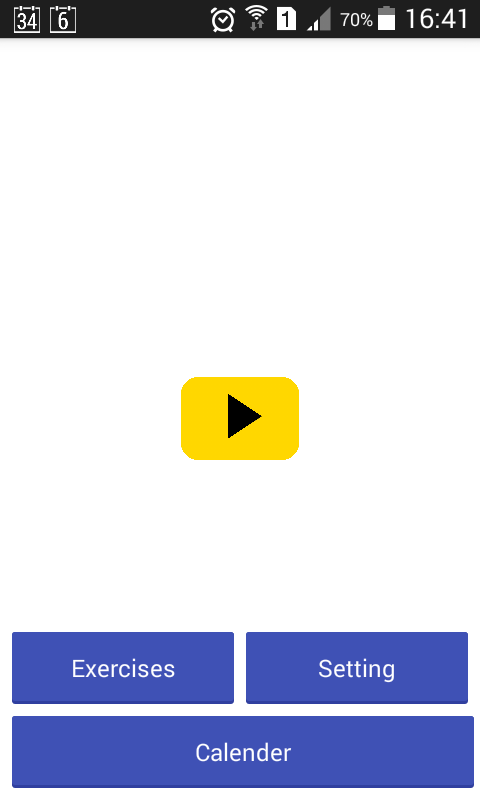
InputMethodManager imm = (InputMethodManager) context.getSystemService(Activity.INPUT\_METHOD\_SERVICE);

imm.hideSoftInputFromWindow(view.getWindowToken(), WindowManager.LayoutParams.SOFT\_INPUT\_STATE\_ALWAYS\_HIDDEN);

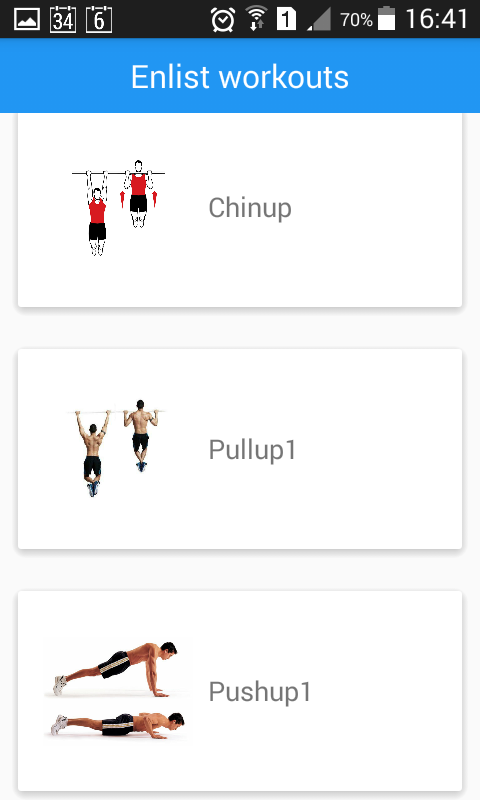
}

}

**5] Main Screen**

****

**6] List of exercises**

****

**7] Android Manifest file**

**Filename: AndroidManifest.xml**

<?xml version="1.0" encoding="utf-8"?>

<manifest

xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.ram.gymdroid">

<application

android:allowBackup="true"

android:icon="@drawable/square\_dumble"

android:label="@string/app\_name"

android:supportsRtl="true"

android:roundIcon="@drawable/square\_dumble"

android:theme="@style/AppTheme">

<activity

android:name=".LoginActivity"

android:screenOrientation="portrait">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity

android:name=".RegisterActivity"

android:screenOrientation="portrait"/>

<activity

android:name=".UsersListActivity"

android:screenOrientation="portrait"/>

</application>

</manifest>

**8] Gradle files:**

**Filename: build.gradle (Module: app)**

apply plugin: 'com.android.application'

android {

compileSdkVersion 25

buildToolsVersion "25.0.3"

defaultConfig {

applicationId "com.example.ram.gymdroid"

minSdkVersion 19

targetSdkVersion 25

versionCode 1

versionName "1.0"

testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

}

buildTypes {

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

dependencies {

compile 'com.android.support:appcompat-v7:23.4.0'

compile 'com.android.support:design:23.4.0'

compile 'com.android.support:cardview-v7:23.4.0'

compile fileTree(dir: 'libs', include: ['\*.jar'])

androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {

exclude group: 'com.android.support', module: 'support-annotations'

})

compile 'com.android.support:appcompat-v7:25.3.1'

compile 'com.android.support.constraint:constraint-layout:1.0.2'

testCompile 'junit:junit:4.12'

}

**9] Database file**

**Filename: DatabaseHelper.java [Java code]**

package com.example.ram.gymdroid.sql;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import com.example.ram.gymdroid.model.User;

import java.util.ArrayList;

import java.util.List;

public class DatabaseHelper extends SQLiteOpenHelper {

// Database Version

private static final int DATABASE\_VERSION = 1;

// Database Name

private static final String DATABASE\_NAME = "UserManager.db";

// User table name

//4/33 page

private static final String TABLE\_USER = "user";

// User Table Columns names

private static final String COLUMN\_USER\_ID = "user\_id";

private static final String COLUMN\_USER\_NAME = "user\_name";

private static final String COLUMN\_USER\_EMAIL = "user\_email";

private static final String COLUMN\_USER\_PASSWORD = "user\_password";

// create table sql query

private String CREATE\_USER\_TABLE = "CREATE TABLE " + TABLE\_USER + "("

+ COLUMN\_USER\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," + COLUMN\_USER\_NAME + " TEXT,"

+ COLUMN\_USER\_EMAIL + " TEXT," + COLUMN\_USER\_PASSWORD + " TEXT" + ")";

// drop table sql query

private String DROP\_USER\_TABLE = "DROP TABLE IF EXISTS " + TABLE\_USER;

/\*\*

\* Constructor

\*

\* @param context

\*/

public DatabaseHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL(CREATE\_USER\_TABLE);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

//Drop User Table if exist

db.execSQL(DROP\_USER\_TABLE);

// Create tables again

onCreate(db);

}

/\*\*

\* This method is to create user record

\*

\* @param user

\*/

public void addUser(User user) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues values = new ContentValues();

values.put(COLUMN\_USER\_NAME, user.getName());

values.put(COLUMN\_USER\_EMAIL, user.getEmail());

values.put(COLUMN\_USER\_PASSWORD, user.getPassword());

// Inserting Row

db.insert(TABLE\_USER, null, values);

db.close();

}

/\*\*

\* This method is to fetch all user and return the list of user records

\*

\* @return list

\*/

public List<User> getAllUser() {

// array of columns to fetch

String[] columns = {

COLUMN\_USER\_ID,

COLUMN\_USER\_EMAIL,

COLUMN\_USER\_NAME,

COLUMN\_USER\_PASSWORD

};

// sorting orders

String sortOrder =

COLUMN\_USER\_NAME + " ASC";

List<User> userList = new ArrayList<User>();

SQLiteDatabase db = this.getReadableDatabase();

// query the user table

/\*\*

\* Here query function is used to fetch records from user table this function works like we use sql query.

\* SQL query equivalent to this query function is

\* SELECT user\_id,user\_name,user\_email,user\_password FROM user ORDER BY user\_name;

\*/

Cursor cursor = db.query(TABLE\_USER, //Table to query

columns, //columns to return

null, //columns for the WHERE clause

null, //The values for the WHERE clause

null, //group the rows

null, //filter by row groups

sortOrder); //The sort order

// Traversing through all rows and adding to list

if (cursor.moveToFirst()) {

do {

User user = new User();

user.setId(Integer.parseInt(cursor.getString(cursor.getColumnIndex(COLUMN\_USER\_ID))));

user.setName(cursor.getString(cursor.getColumnIndex(COLUMN\_USER\_NAME)));

user.setEmail(cursor.getString(cursor.getColumnIndex(COLUMN\_USER\_EMAIL)));

user.setPassword(cursor.getString(cursor.getColumnIndex(COLUMN\_USER\_PASSWORD)));

//Adding user record to list

//5/33

userList.add(user);

} while (cursor.moveToNext());

}

cursor.close();

db.close();

// return user list

return userList;

}

/\*\*

\* This method to update user record

\*

\* @param user

\*/

public void updateUser(User user) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues values = new ContentValues();

values.put(COLUMN\_USER\_NAME, user.getName());

values.put(COLUMN\_USER\_EMAIL, user.getEmail());

values.put(COLUMN\_USER\_PASSWORD, user.getPassword());

// updating row

db.update(TABLE\_USER, values, COLUMN\_USER\_ID + " = ?",

new String[]{String.valueOf(user.getId())});

db.close();

}

/\*\*

\* This method is to delete user record

\*

\* @param user

\*/

public void deleteUser(User user) {

SQLiteDatabase db = this.getWritableDatabase();

// delete user record by id

db.delete(TABLE\_USER, COLUMN\_USER\_ID + " = ?",

new String[]{String.valueOf(user.getId())});

db.close();

}

/\*\*

\* This method to check user exist or not

\*

\* @param email

\* @return true/false

\*/

public boolean checkUser(String email) {

// array of columns to fetch

String[] columns = {

COLUMN\_USER\_ID

};

SQLiteDatabase db = this.getReadableDatabase();

// selection criteria

String selection = COLUMN\_USER\_EMAIL + " = ?";

// selection argument

String[] selectionArgs = {email};

// query user table with condition

/\*\*

\* Here query function is used to fetch records from user table this function works like we use sql query.

\* SQL query equivalent to this query function is

\* SELECT user\_id FROM user WHERE user\_email = 'jack@androidtutorialshub.com';

\*/

Cursor cursor = db.query(TABLE\_USER, //Table to query

columns, //columns to return

selection, //columns for the WHERE clause

selectionArgs, //The values for the WHERE clause

null, //group the rows

null, //filter by row groups

null); //The sort order

int cursorCount = cursor.getCount();

cursor.close();

db.close();

if (cursorCount > 0) {

return true;

}

return false;

}

/\*\*

\* This method to check user exist or not

\*

\* @param email

\* @param password

\* @return true/false

\*/

public boolean checkUser(String email, String password) {

// array of columns to fetch

String[] columns = {

COLUMN\_USER\_ID

};

SQLiteDatabase db = this.getReadableDatabase();

// selection criteria

String selection = COLUMN\_USER\_EMAIL + " = ?" + " AND " + COLUMN\_USER\_PASSWORD + " = ?";

// selection arguments

//6/33

String[] selectionArgs = {email, password};

// query user table with conditions

/\*\*

\* Here query function is used to fetch records from user table this function works like we use sql query.

\* SQL query equivalent to this query function is

\* SELECT user\_id FROM user WHERE user\_email = 'jack@androidtutorialshub.com' AND user\_password = 'qwerty';

\*/

Cursor cursor = db.query(TABLE\_USER, //Table to query

columns, //columns to return

selection, //columns for the WHERE clause

selectionArgs, //The values for the WHERE clause

null, //group the rows

null, //filter by row groups

null); //The sort order

int cursorCount = cursor.getCount();

cursor.close();

db.close();

if (cursorCount > 0) {

return true;

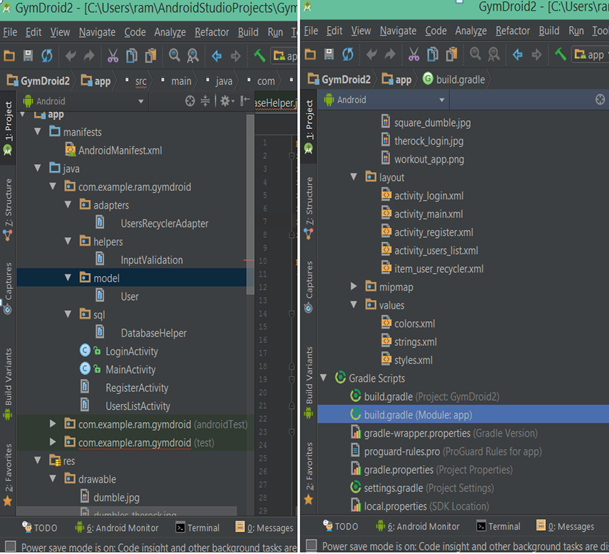
}

return false;

}

}

**10] 1: Project view hierarchy**



**FUTURE ENHANCEMENT**

* In futher release, users’ profile image functionality will be added.
* User will be able to bookmark the selected workouts.
* Alarm setup will be added and timer facility will be improved based on difficulty levels.
* A graphical chart functionality will be added to view daily progress.
* Diet plans will be added.
* Users will be able to rate their favourite workouts.

**LIMITATION**

* Clients cannot connect to the internet for viewing their progress.
* No progress review.
* No diet plans
* No gender based exercises
* No feedback for the admin

. **BIBLIOGRAPHY**

**References**

In order to complete this project these references were used by me:

Websites links:

* http://yasirameen.com
* <https://materialuicolors.co/>
* https://romannurik.github.io/AndroidAssetStudio/
* <https://stackoverflow.com>

Youtube:

# EDMT dev

# https://www.youtube.com/channel/UCllewj2bGdqB8U9Ld15INAg

* Aws Rh

<https://www.youtube.com/channel/UCoQp_Duwqh0JWEZrg4DT2Ug>

* [ProgrammingKnowledge](https://www.youtube.com/channel/UCs6nmQViDpUw0nuIx9c_WvA)

<https://www.youtube.com/watch?v=MWZ2rVFOQWw&t=0s&list=PL371UnkOYBXfDeopqoZ418416k854sZkg&index=19>

Android App:

* Udemy Courses

https://www.udemy.com/