COM413 – Mobile App Development – Workbook Practical 2

Introduction

In the previous practical we worked on creating simple UIs within applications and adding functionality. We created and app which had more than one activity that had a multi-functioning interface: web search, phone call and SMS messaging. So now to move on from where we left off.

We will now create a login/register page and use SQLite to record the users and their data, so that they can log into the app.

Basic:

Create a login page, which will also allow users to register:

- Must use SQLite for this
- View the data using DB Browser

We will now create a login page with a separate Register Activity. Again, we will use SQLite to record the users and their data. But we will use the register button as an OnClickListerner to start a new activity called Register, this will open the register layout.

Advanced:

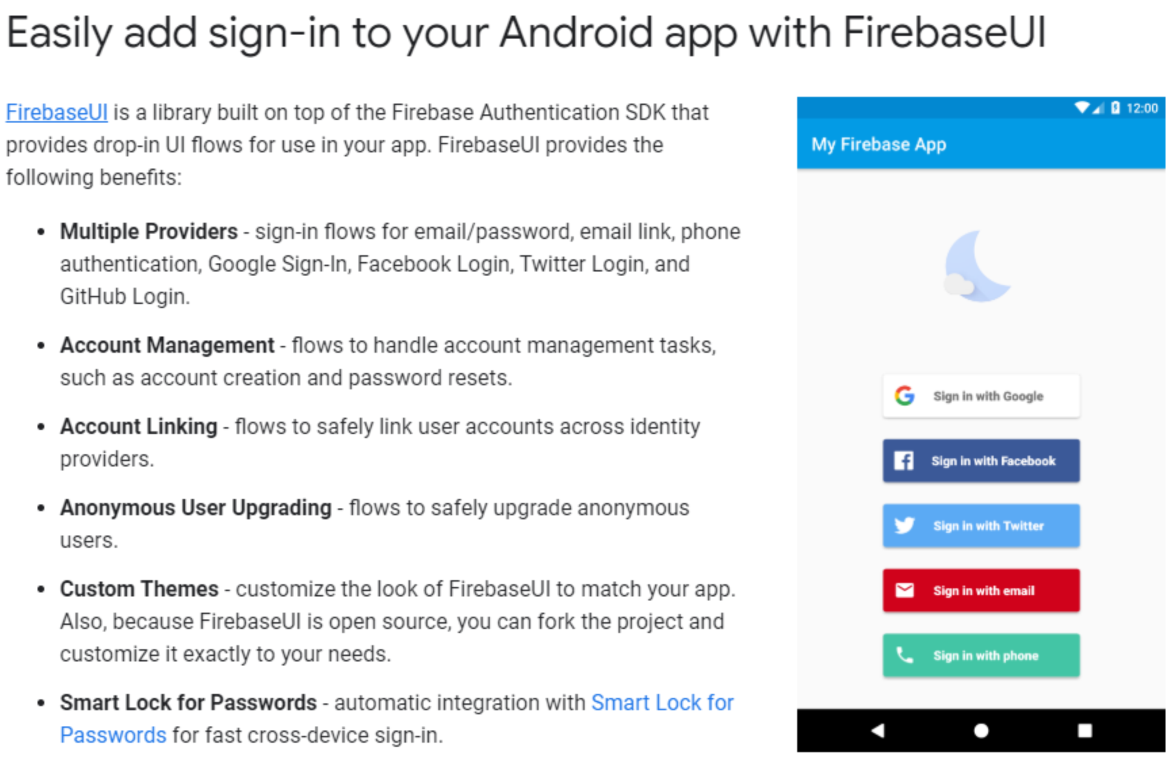
Connecting your app to Firebase

There is a lot of information regarding connecting Firebase to an android app, I will be going over this in the lecture. If you can set up Firebase and connect it to a new project, I will discuss how to use authentication on Friday and next week's Practical we will set-up Firebase Authentication.

Additional:

Connecting your Firebase to other login types.

The advanced functionality is to use Twitter or Facebook to Log in to your app.



Easily add sign-in to your Android app with FirebaseUI

FirebaseUI is a library built on top of the Firebase Authentication SDK that provides drop-in UI flows for use in your app. FirebaseUI provides the following benefits:

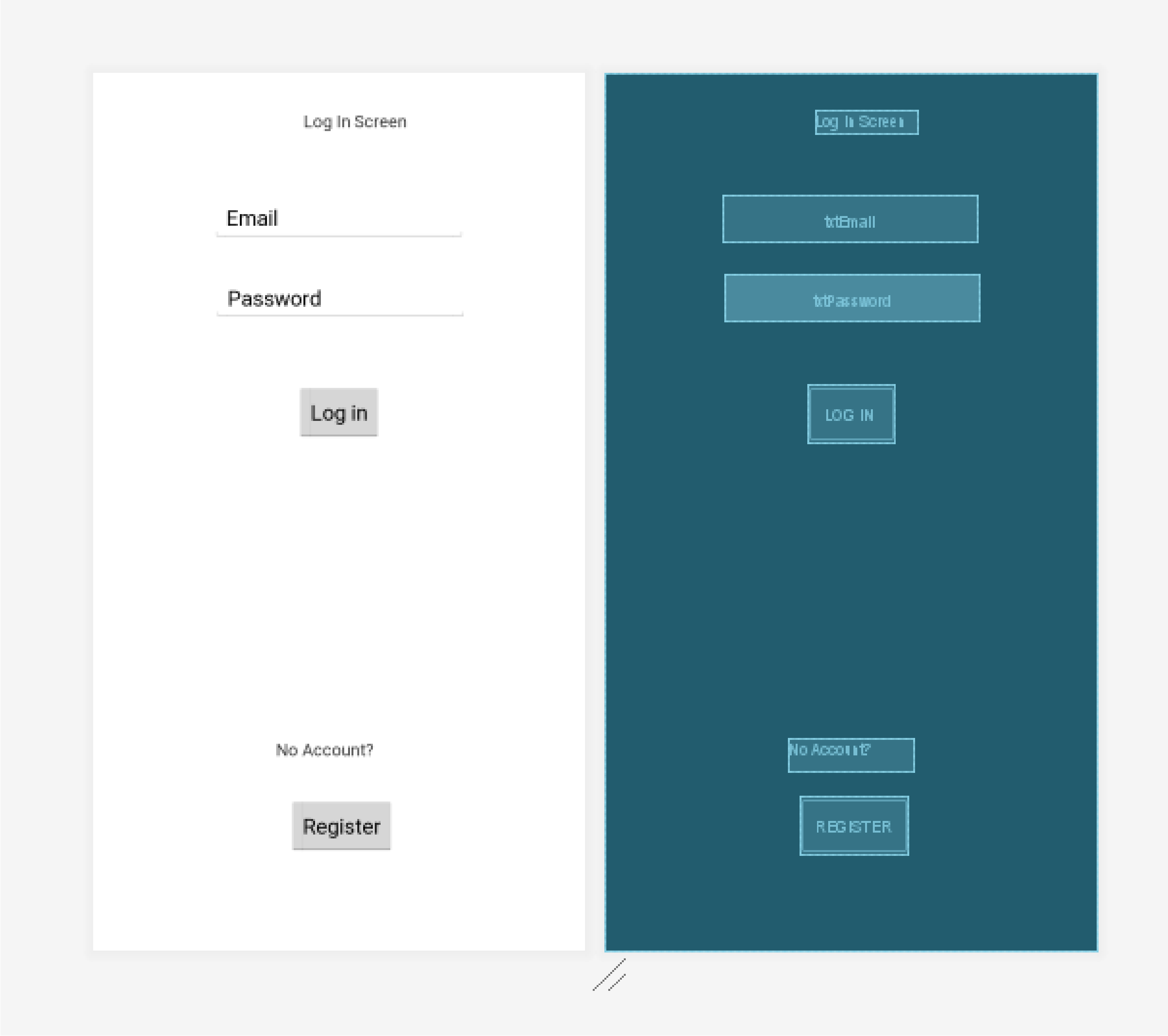
- **Multiple Providers** - sign-in flows for email/password, email link, phone authentication, Google Sign-In, Facebook Login, Twitter Login, and GitHub Login.
- **Account Management** - flows to handle account management tasks, such as account creation and password resets.
- **Account Linking** - flows to safely link user accounts across identity providers.
- **Anonymous User Upgrading** - flows to safely upgrade anonymous users.
- **Custom Themes** - customize the look of FirebaseUI to match your app. Also, because FirebaseUI is open source, you can fork the project and customize it exactly to your needs.
- **Smart Lock for Passwords** - automatic integration with Smart Lock for Passwords for fast cross-device sign-in.

Basic Functionality:

Introduction

1. Create a new project.
2. Start with an empty activity.
3. In the component Tree, delete the textview.

The XML layout:

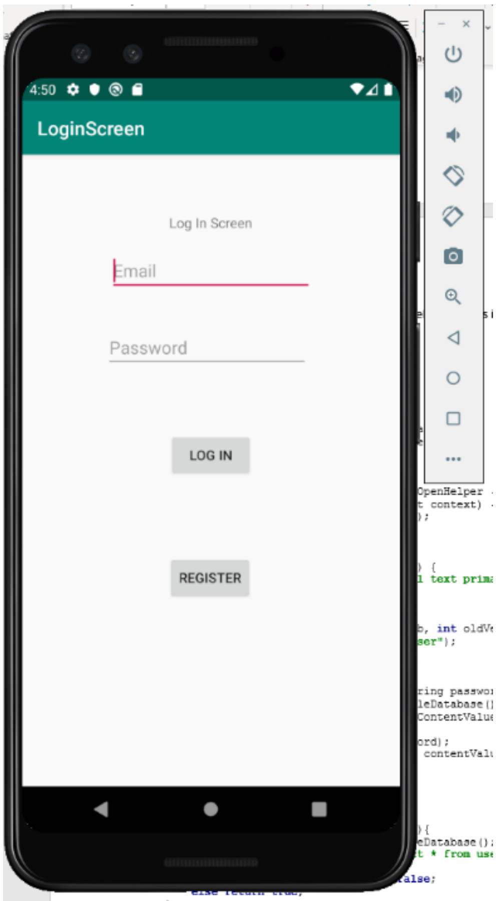


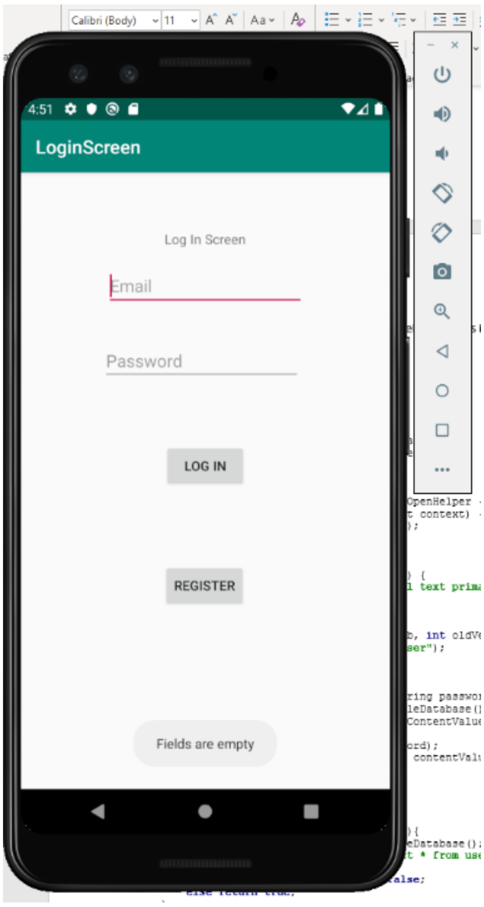
The Java behind the XML:

```java
package com.example.loginscreen;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    DatabaseHelper db;
    EditText txtEmail,txtPassword;
    Button btnLogin, btnRegister;

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

        db = new DatabaseHelper(this);

        txtEmail = (EditText) findViewById(R.id.txtEmail);
        txtPassword = (EditText)findViewById(R.id.txtPassword);

        btnRegister = (Button) findViewById(R.id.btnRegister);
        btnRegister.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String Email = txtEmail.getText().toString();
                String Password = txtPassword.getText().toString();

                if(Email.equals("")||Password.equals("")){
                    Toast.makeText(getApplicationContext(),"Fields are
empty",Toast.LENGTH_SHORT).show();
                }
                else{
                    Boolean checkEmail = db.checkEmail(Email);
                    if(checkEmail==true){
                        Boolean insert = db.insert(Email,Password);
                        if(insert==true){
                            Toast.makeText(getApplicationContext(),"Register
Successful",Toast.LENGTH_SHORT).show();
                        }
                    }
                    else{
                        Toast.makeText(getApplicationContext(),"Email already
Exists",Toast.LENGTH_SHORT).show();
                    }
                }
            }
        });
        //btnLogin = (Button)findViewById(R.id.btnLogin);
    }
}
```

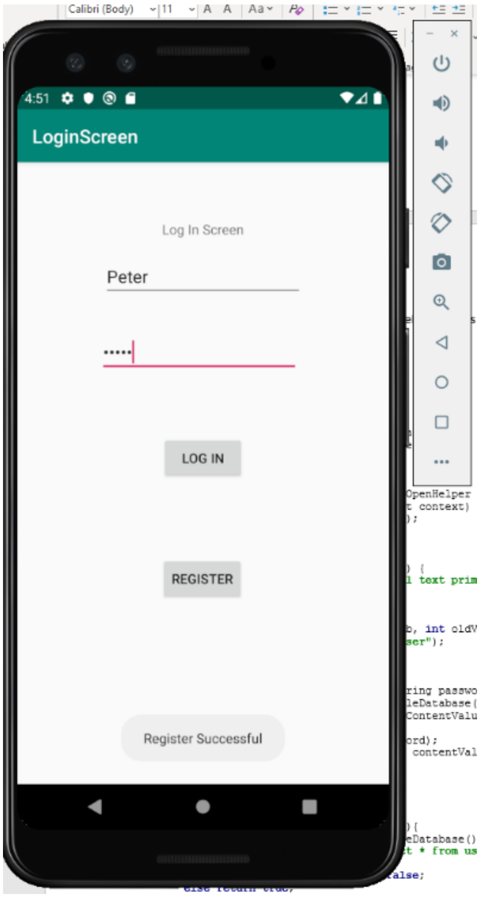
You need to create a new java class called DatabaseHelper. This is based on SQLite:

```java
package com.example.loginscreen;

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

import androidx.annotation.Nullable;

public class DatabaseHelper extends SQLiteOpenHelper {
    public DatabaseHelper(@Nullable Context context) {
        super(context, "Login.db", null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("Create table user(email text primary key, password text)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("drop table if exists user");
    }

    //insert into database
    public boolean insert(String email, String password){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("email",email);
        contentValues.put("password",password);
        long ins = db.insert("user", null, contentValues);
        if(ins==-1) return false;
        else return true;
    }

    //check if email exists
    public Boolean checkEmail(String email){
        SQLiteDatabase db= this.getReadableDatabase();
        Cursor cursor = db.rawQuery("Select * from user where email=?", new
String[]{email});
        if(cursor.getCount()>0) return false;
        else return true;
    }

}
```





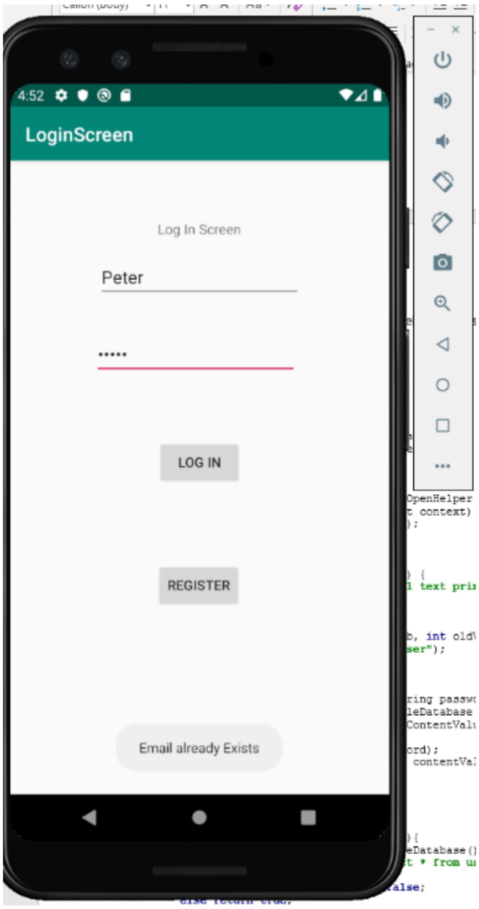
Fields are empty



Peter

•••••

Register Successful



Peter

•••••

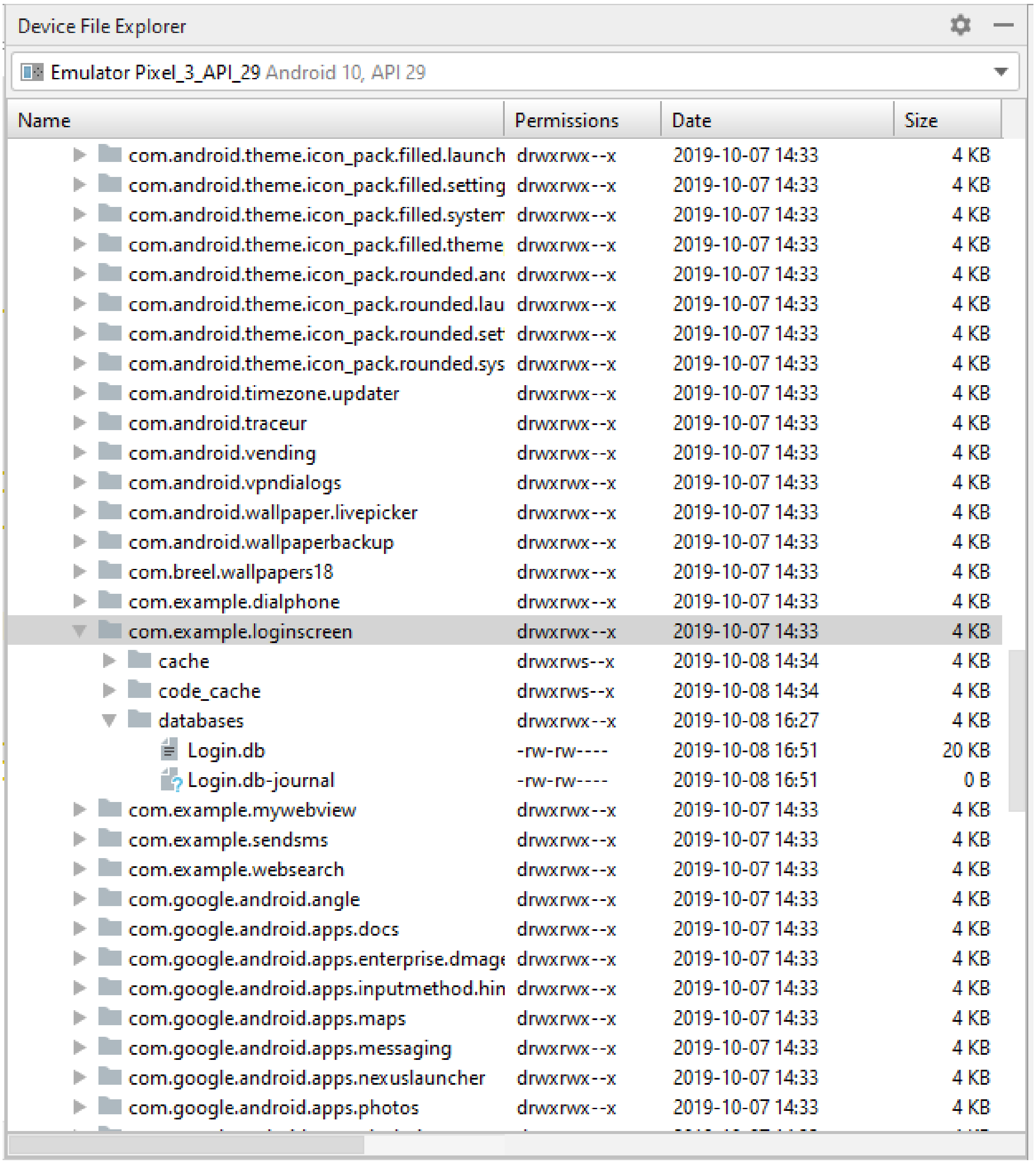
Email already Exists

If we look further:

Click View -> Tool Windows -> Device File Explorer

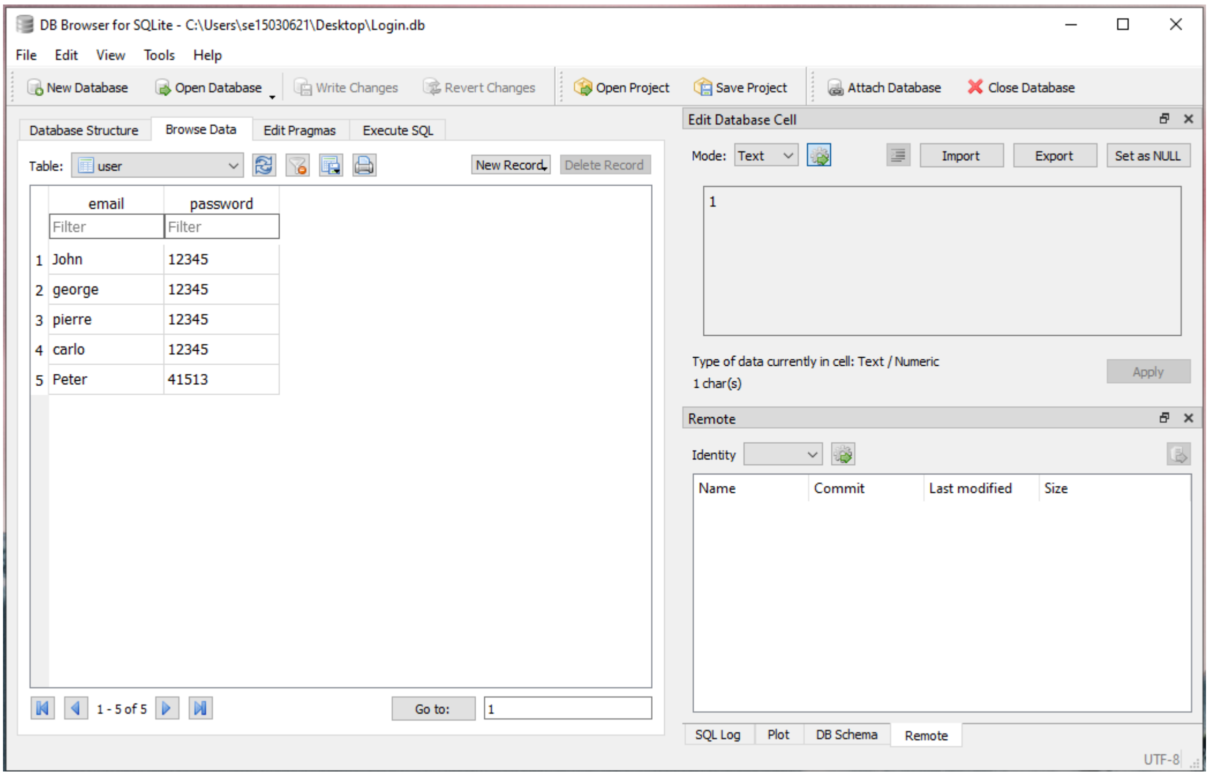
Load your AVD Emulator

Data -> Data -> com.example.loginscreen



Emulator Pixel_3_API_29 Android 10, API 29

Name	Permissions	Date	Size
▶ com.android.theme.icon_pack.filled.launch	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.theme.icon_pack.filled.setting	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.theme.icon_pack.filled.systerr	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.theme.icon_pack.filled.theme	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.theme.icon_pack.rounded.anc	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.theme.icon_pack.rounded.lau	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.theme.icon_pack.rounded.set	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.theme.icon_pack.rounded.sys	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.timezone.updater	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.traceur	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.vending	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.vpndialogs	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.wallpaper.livepicker	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.android.wallpaperbackup	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.breel.wallpapers18	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.example.dialphone	drwxrwx--x	2019-10-07 14:33	4 KB
▼ com.example.loginscreen	drwxrwx--x	2019-10-07 14:33	4 KB
▶ cache	drwxrws--x	2019-10-08 14:34	4 KB
▶ code_cache	drwxrws--x	2019-10-08 14:34	4 KB
▼ databases	drwxrwx--x	2019-10-08 16:27	4 KB
Login.db	-rw-rw----	2019-10-08 16:51	20 KB
Login.db-journal	-rw-rw----	2019-10-08 16:51	0 B
▶ com.example.mywebview	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.example.sendsms	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.example.websearch	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.google.android.angle	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.google.android.apps.docs	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.google.android.apps.enterprise.dmage	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.google.android.apps.inputmethod.hin	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.google.android.apps.maps	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.google.android.apps.messaging	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.google.android.apps.nexuslauncher	drwxrwx--x	2019-10-07 14:33	4 KB
▶ com.google.android.apps.photos	drwxrwx--x	2019-10-07 14:33	4 KB



DB Browser for SQLite - C:\Users\se15030621\Desktop\Login.db

Table: user

	email	password
	Filter	Filter
1	John	12345
2	george	12345
3	pierre	12345
4	carlo	12345
5	Peter	41513

1 - 5 of 5

Go to: 1

Mode: Text

1

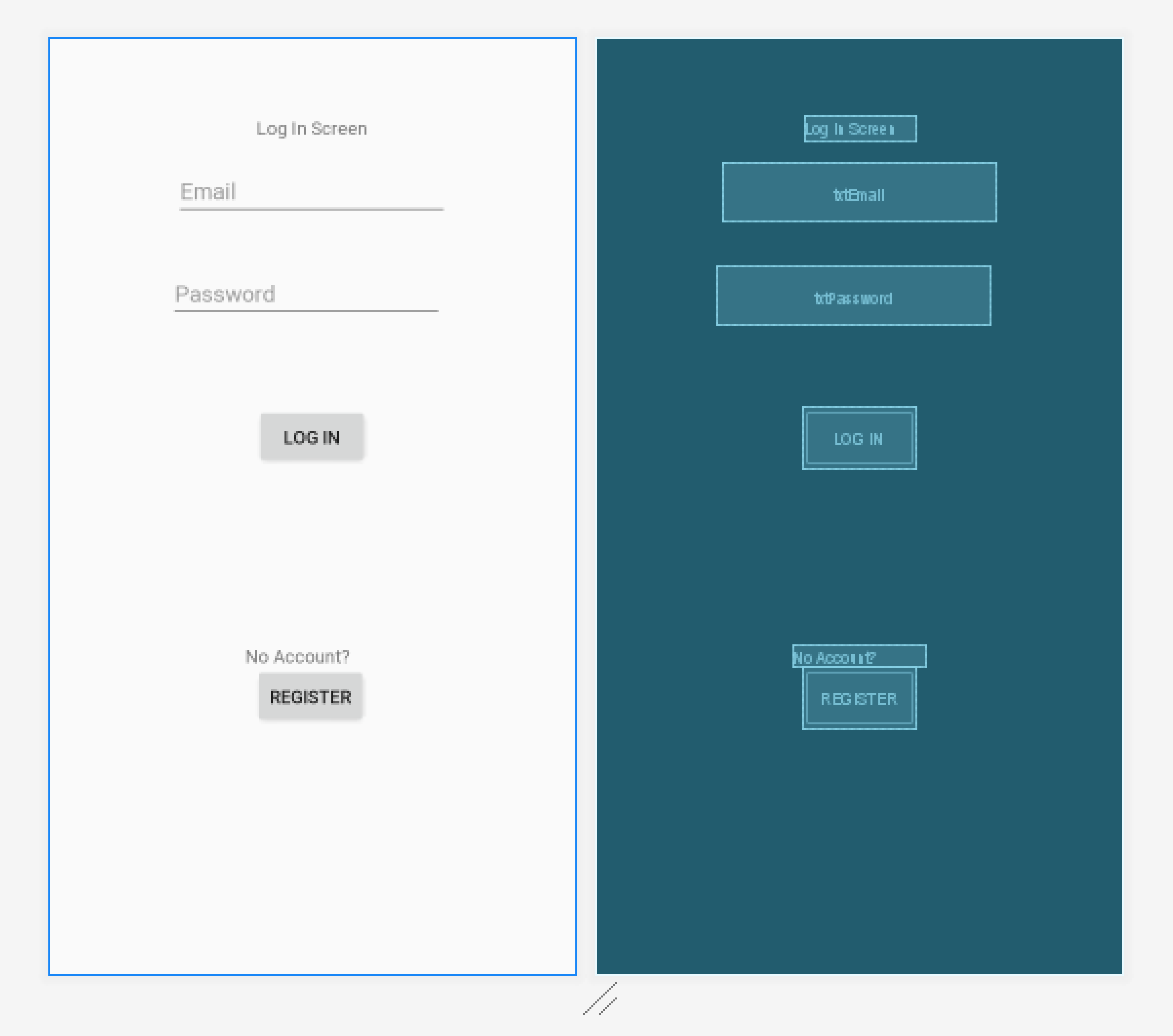
Type of data currently in cell: Text / Numeric

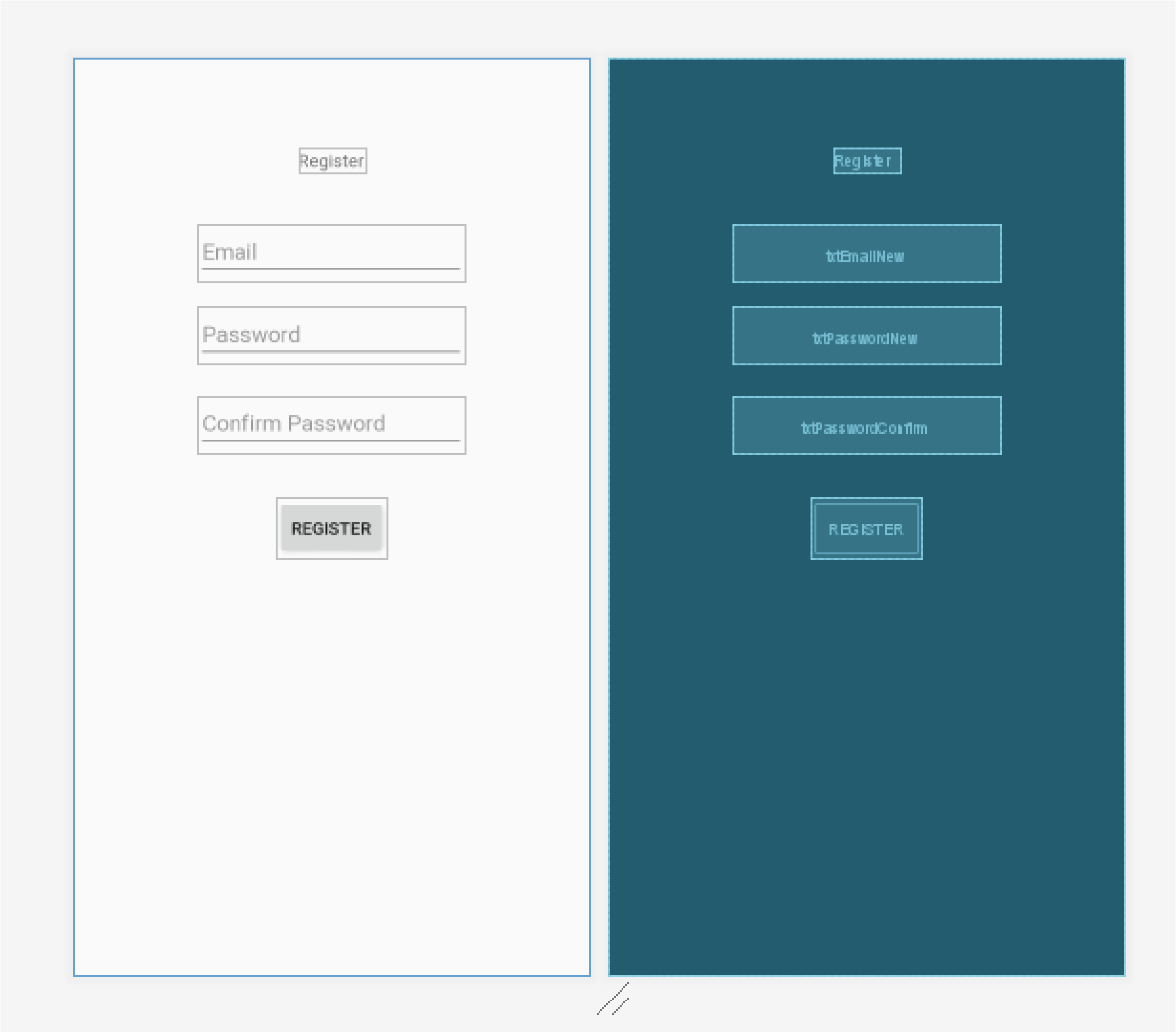
1 char(s)

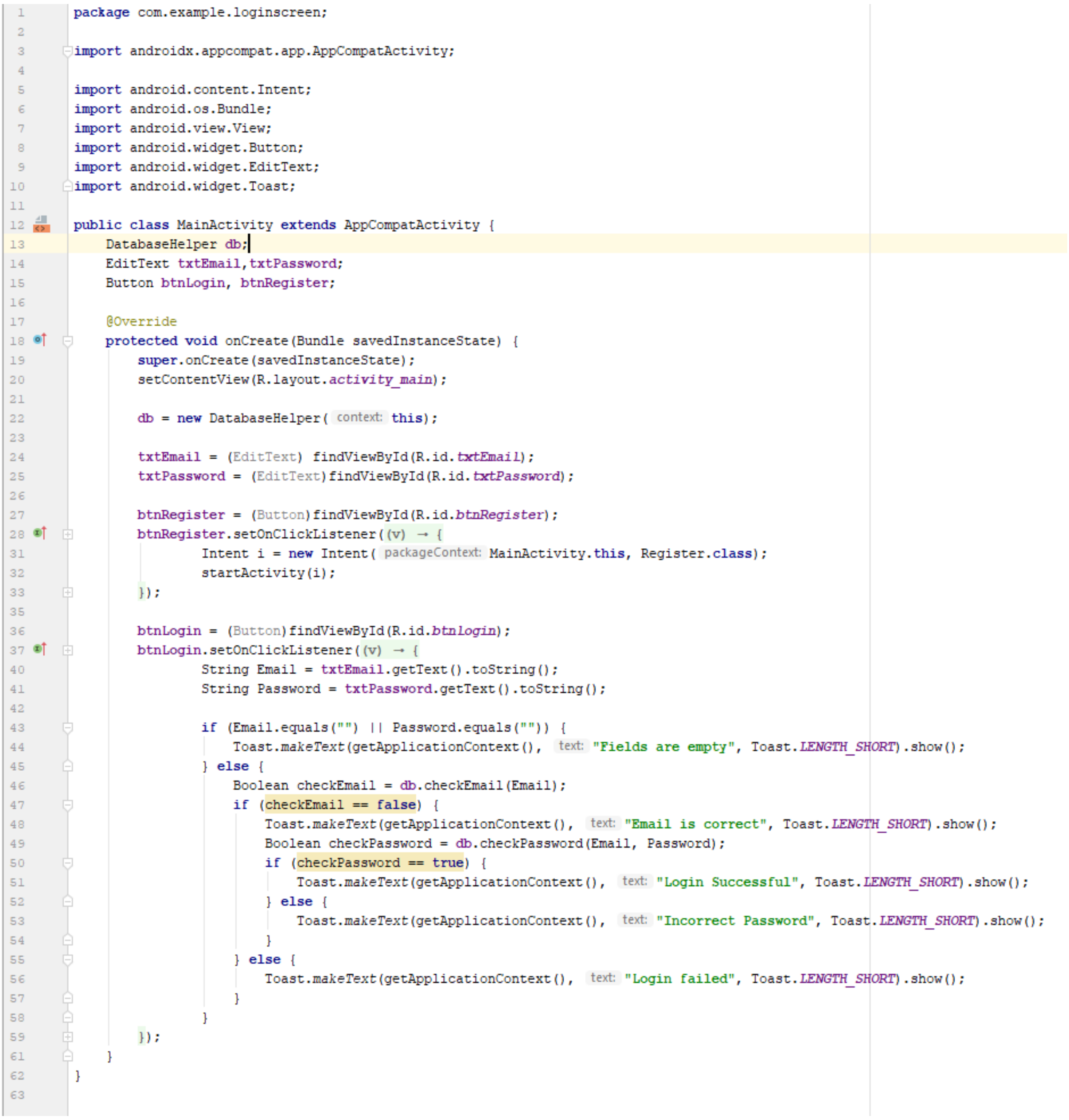
We will now create a login page with a separate Register Activity. Again, we will use SQLite to record the users and their data. But we will use the register button as an OnClickListerner to start a new activity called Register, this will open the register layout.

We would need two XML layouts:

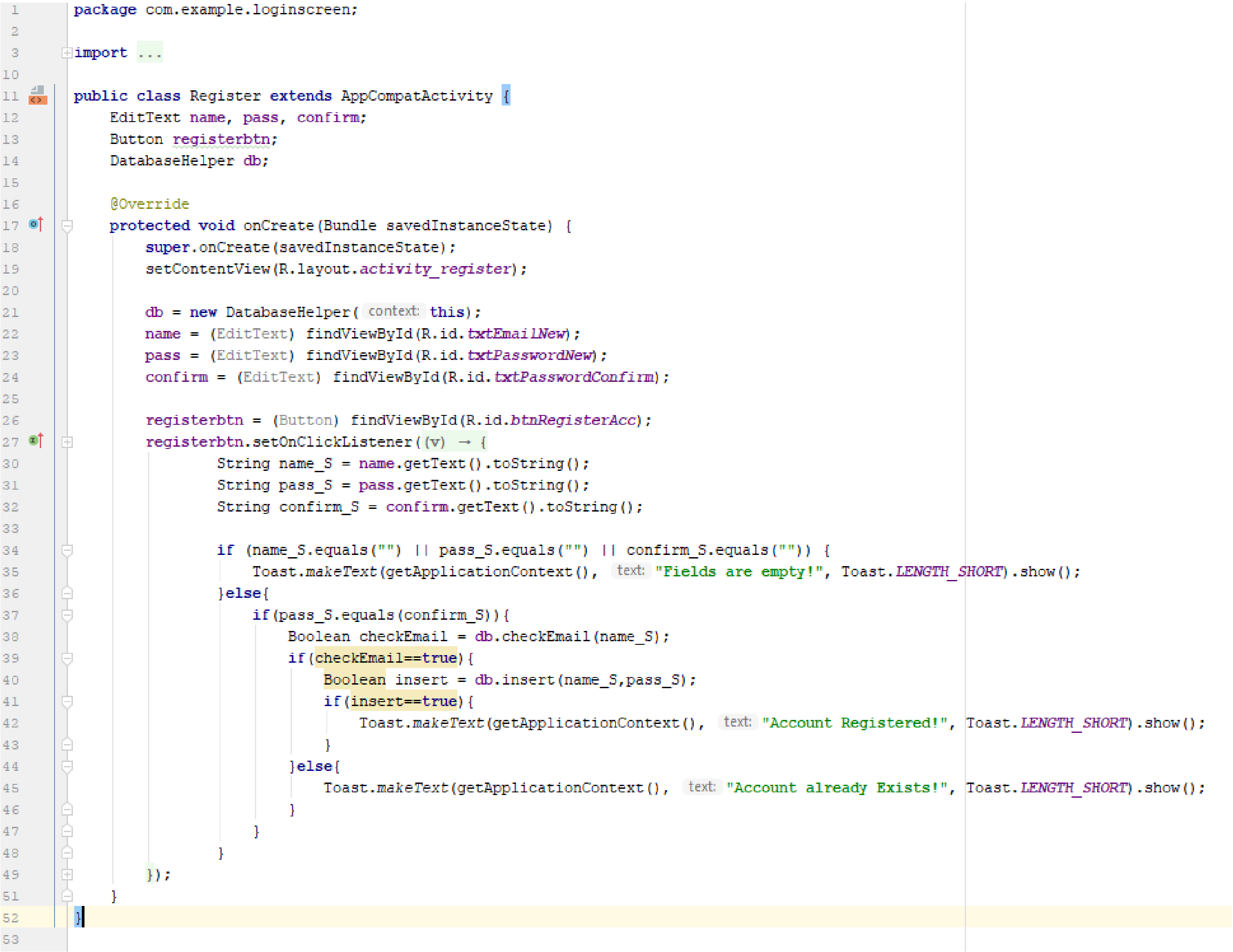
Main XML:



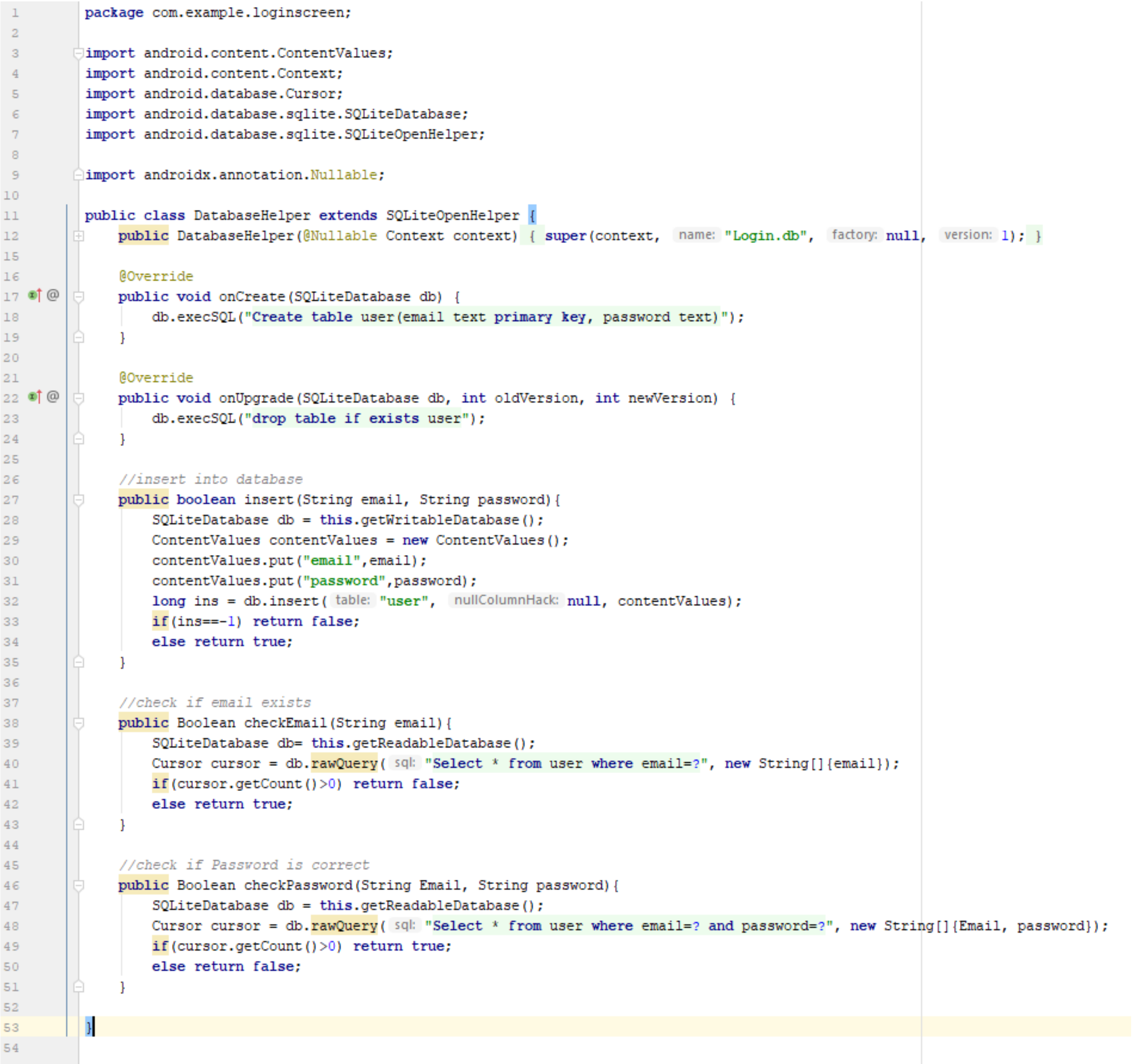

Register
Email
Password
Confirm Password
REGISTER
Register
txtEmailNew
txtPasswordNew
txtPasswordConfirm
REGISTER



```java
package com.example.loginscreen;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    DatabaseHelper db;
    EditText txtEmail,txtPassword;
    Button btnLogin, btnRegister;

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

        db = new DatabaseHelper( context: this);

        txtEmail = (EditText) findViewById(R.id.txtEmail);
        txtPassword = (EditText)findViewById(R.id.txtPassword);

        btnRegister = (Button)findViewById(R.id.btnRegister);
        btnRegister.setOnClickListener((v) → {
                Intent i = new Intent( packageContext: MainActivity.this, Register.class);
                startActivity(i);
        });

        btnLogin = (Button)findViewById(R.id.btnlogin);
        btnLogin.setOnClickListener((v) → {
                String Email = txtEmail.getText().toString();
                String Password = txtPassword.getText().toString();

                if (Email.equals("") || Password.equals("")) {
                    Toast.makeText(getApplicationContext(),  text: "Fields are empty", Toast.LENGTH_SHORT).show();
                } else {
                    Boolean checkEmail = db.checkEmail(Email);
                    if (checkEmail == false) {
                        Toast.makeText(getApplicationContext(),  text: "Email is correct", Toast.LENGTH_SHORT).show();
                        Boolean checkPassword = db.checkPassword(Email, Password);
                        if (checkPassword == true) {
                            Toast.makeText(getApplicationContext(),  text: "Login Successful", Toast.LENGTH_SHORT).show();
                        } else {
                            Toast.makeText(getApplicationContext(),  text: "Incorrect Password", Toast.LENGTH_SHORT).show();
                        }
                    } else {
                        Toast.makeText(getApplicationContext(),  text: "Login failed", Toast.LENGTH_SHORT).show();
                    }
                }
        });
    }
}
```



```java
package com.example.loginscreen;

import ...

public class Register extends AppCompatActivity {
    EditText name, pass, confirm;
    Button registerbtn;
    DatabaseHelper db;

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

        db = new DatabaseHelper( context: this);
        name = (EditText) findViewById(R.id.txtEmailNew);
        pass = (EditText) findViewById(R.id.txtPasswordNew);
        confirm = (EditText) findViewById(R.id.txtPasswordConfirm);

        registerbtn = (Button) findViewById(R.id.btnRegisterAcc);
        registerbtn.setOnClickListener((v) → {
                String name_S = name.getText().toString();
                String pass_S = pass.getText().toString();
                String confirm_S = confirm.getText().toString();

                if (name_S.equals("") || pass_S.equals("") || confirm_S.equals("")) {
                    Toast.makeText(getApplicationContext(),  text: "Fields are empty!", Toast.LENGTH_SHORT).show();
                }else{
                    if(pass_S.equals(confirm_S)){
                        Boolean checkEmail = db.checkEmail(name_S);
                        if(checkEmail==true){
                            Boolean insert = db.insert(name_S,pass_S);
                            if(insert==true){
                                Toast.makeText(getApplicationContext(),  text: "Account Registered!", Toast.LENGTH_SHORT).show();
                            }
                        }else{
                            Toast.makeText(getApplicationContext(),  text: "Account already Exists!", Toast.LENGTH_SHORT).show();
                        }
                    }
                }
        });
    }
}

```



```java
package com.example.loginscreen;

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

import androidx.annotation.Nullable;

public class DatabaseHelper extends SQLiteOpenHelper {
    public DatabaseHelper(@Nullable Context context) { super(context, name: "Login.db", factory: null, version: 1); }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("Create table user(email text primary key, password text)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("drop table if exists user");
    }

    //insert into database
    public boolean insert(String email, String password){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put("email",email);
        contentValues.put("password",password);
        long ins = db.insert( table: "user", nullColumnHack: null, contentValues);
        if(ins==-1) return false;
        else return true;
    }

    //check if email exists
    public Boolean checkEmail(String email){
        SQLiteDatabase db= this.getReadableDatabase();
        Cursor cursor = db.rawQuery( sql: "Select * from user where email=?", new String[]{email});
        if(cursor.getCount()>0) return false;
        else return true;
    }

    //check if Password is correct
    public Boolean checkPassword(String Email, String password){
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor = db.rawQuery( sql: "Select * from user where email=? and password=?", new String[]{Email, password});
        if(cursor.getCount()>0) return true;
        else return false;
    }

}
```