

ST LAB 8

SAHIL BONDRE: U18CO021

Question: Assume you are a software developer and your company wants you to develop an android app which stores contact information (such as Name, Mobile No, Email, Address etc.) in SQLite database and provides following options to its users.

- (a) Enter a new contact
- (b) Search for an existing contact
- (c) Delete an existing contact
- (d) Update/Edit an existing contact

Submission: A single (doc/pdf) file which will contain your source code and the screenshots of generated outputs. Name of your file should be assignment-number_yourid.

build.gradle

```
plugins {  
    id 'com.android.application'  
}  
  
android {  
    compileSdkVersion 30  
    buildToolsVersion "30.0.2"  
  
    defaultConfig {  
        applicationId "com.example.contactslab8"  
        minSdkVersion 18  
        targetSdkVersion 30  
        versionCode 1  
        versionName "1.0"  
  
        testInstrumentationRunner  
"androidx.test.runner.AndroidJUnitRunner"  
    }  
  
    buildTypes {
```

```

        release {
            minifyEnabled false
            proguardFiles
getDefaultProguardFile('proguard-android-optimize.txt'),
'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {

    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.3.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation
'androidx.test.espresso:espresso-core:3.3.0'

    def room_version = "2.2.6"

    implementation "androidx.room:room-runtime:$room_version"
    annotationProcessor "androidx.room:room-compiler:$room_version"
}

```

Main Activity:

MainActivity.java

```

package com.example.contactslab8;

import android.content.Intent;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;

import androidx.annotation.Nullable;

```

```

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.room.Room;

import com.example.contactslab8.models.Contact;
import com.example.contactslab8.models.ContactDao;
import
com.google.android.material.floatingactionbutton.FloatingActionButton;

import java.util.List;
import java.util.Vector;

public class MainActivity extends AppCompatActivity {

    private List<String> contactNames;
    private List<Contact> contacts;
    private ListView contactListView;
    private Button searchButton;
    private EditText searchEditText;
    private FloatingActionButton fab;
    private ArrayAdapter<String> arrayAdapter;
    private AppDatabase db;
    private ContactDao contactDao;

    private void updateContactList() {
        arrayAdapter = new ArrayAdapter<>(this,
R.layout.activity_listview, R.id.contact_text_view, contactNames);
        contactListView.setAdapter(arrayAdapter);
    }

    private void initialiseUI() {
        contactListView = findViewById(R.id.contacts_list_view);
        searchButton = findViewById(R.id.search_button);
        searchEditText = findViewById(R.id.search_edit_text);
        fab = findViewById(R.id.fab);

        contactListView.setOnItemClickListener((parent, view, position,
id) -> {
            Intent intent = new Intent(this, ViewContactActivity.class);
            intent.putExtra("contact", contacts.get(position));
            startActivityForResult(intent, 2);
        });

        fab.setOnClickListener(v -> {
            Intent intent = new Intent(this, AddActivity.class);

```

```

        startActivityForResult(intent, 1);
    });

    searchButton.setOnClickListener(v -> {
        new Thread(() -> {
            Contact contact =
contactDao.findByName(searchEditText.getText().toString());
            runOnUiThread(() -> {

                if (contact != null) {
                    Intent intent = new Intent(this,
ViewContactActivity.class);
                    intent.putExtra("contact", contact);
                    startActivityForResult(intent, 2);
                } else {
                    new AlertDialog.Builder(this)

.setIcon(android.R.drawable.stat_sys_warning)
                    .setTitle("404")
                    .setMessage("Contact not found!")
                    .setNegativeButton("OK",

(dialogInterface, i) -> {

                        }).show();
                    }
                });
            }).start();
        });
    }

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    db = Room.databaseBuilder(this, AppDatabase.class,
"contacts").build();
    contactDao = db.contactDao();
    initialiseUI();
    new Thread(this::initialiseContacts).start();
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    new Thread(this::initialiseContacts).start();
}

```

```

    }

    private void initialiseContacts() {
        contacts = contactDao.getAll();
        contactNames = new Vector<>(contacts.size());

        for (Contact contact : contacts) {
            contactNames.add(contact.name);
        }
        runOnUiThread(this::updateContactList);
    }
}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:paddingHorizontal="5dp"
    android:paddingVertical="5dp"
    tools:context=".MainActivity">

    <LinearLayout
        android:id="@+id/search_bar"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <EditText
            android:id="@+id/search_edit_text"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_margin="4dp"
            android:layout_weight="1" />

        <Button
            android:id="@+id/search_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="4dp"
            android:text="Search" />

```

```

</LinearLayout>

<ListView
    android:id="@+id/contacts_list_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_below="@+id/search_bar" />

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:contentDescription="Add"
    android:src="@android:drawable/ic_input_add" />

</RelativeLayout>

```

View Contact Activity

ViewContactActivity.java

```

package com.example.contactslab8;

import android.os.Bundle;
import android.telephony.PhoneNumberUtils;
import android.widget.Button;
import android.widget.EditText;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.room.Room;

import com.example.contactslab8.models.Contact;
import com.example.contactslab8.models.ContactDao;

public class ViewContactActivity extends AppCompatActivity {
    private EditText emailEditText;
    private EditText nameEditText;
    private EditText phoneEditText;
    private EditText addressEditText;
    private Button saveButton;

```

```

private Button deleteButton;
private String name;
private String phone;
private String address;
private String email;
private AppDatabase db;
private ContactDao contactDao;
private Contact contact;

private boolean isValidEmail(CharSequence target) {
    if (target == null)
        return false;

    return
android.util.Patterns.EMAIL_ADDRESS.matcher(target).matches();
}

private void showErrorDialog(String message) {
    new AlertDialog.Builder(this)
        .setIcon(android.R.drawable.stat_notify_error)
        .setTitle("Error")
        .setMessage(message)
        .setNegativeButton("OK", (dialogInterface, i) -> {
            //set what would happen when positive button is
clicked
        })
        .show();
}

private void showAddDialogue() {
    new AlertDialog.Builder(this)
        .setIcon(android.R.drawable.ic_menu_save)
        .setTitle("Save")
        .setMessage("Are you sure you want to save contact?")
        .setPositiveButton("Yes", (dialogInterface, i) -> {
            new Thread(() -> {
                contactDao.update(new Contact(contact.uid, name,
email, phone, address));
                runOnUiThread(() -> {
                    setResult(1);
                    finish();
                });
            }).start();
            //set what would happen when positive button is
clicked
        }).setNegativeButton("No", (dialogInterface, i) -> {

```

```

    }).show();
}

private void initialiseUI() {
    emailEditText = findViewById(R.id.email_edit_text);
    nameEditText = findViewById(R.id.name_edit_text);
    phoneEditText = findViewById(R.id.phone_edit_text);
    addressEditText = findViewById(R.id.address_edit_text);
    saveButton = findViewById(R.id.save_button);
    deleteButton = findViewById(R.id.delete_button);

    emailEditText.setText(contact.email);
    nameEditText.setText(contact.name);
    phoneEditText.setText(contact.mobile);
    addressEditText.setText(contact.address);

    saveButton.setOnClickListener(v -> {
        name = nameEditText.getText().toString();
        email = emailEditText.getText().toString();
        phone = phoneEditText.getText().toString();
        address = addressEditText.getText().toString();

        if (name.length() == 0) {
            showErrorDialog("Name cannot be empty");
            return;
        }

        if (email.length() != 0 && !isValidEmail(email)) {
            showErrorDialog("Invalid email");
            return;
        }

        if (!PhoneNumberUtils.isGlobalPhoneNumber(phone)) {
            showErrorDialog("Invalid Phone");
            return;
        }

        showAddDialogue();
    });

    deleteButton.setOnClickListener(v -> {
        showDeleteDialogue();
    });
}

```



```

private void showDeleteDialogue() {
    new AlertDialog.Builder(this)
        .setIcon(android.R.drawable.ic_delete)
        .setTitle("Delete")
        .setMessage("Are you sure you want to delete this
contact?")
        .setPositiveButton("Yes", (dialogInterface, i) -> {
            new Thread(() -> {
                contactDao.delete(contact);
                runOnUiThread(() -> {
                    setResult(1);
                    finish();
                });
            }).start();
            //set what would happen when positive button is
clicked
        }).setNegativeButton("No", (dialogInterface, i) -> {

        }).show();
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_view_contact);
    contact = (Contact) getIntent().getSerializableExtra("contact");

    db = Room.databaseBuilder(this, AppDatabase.class,
"contacts").build();
    contactDao = db.contactDao();
    initialiseUI();
}
}

```

activity_view_contact.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingHorizontal="6dp"
    android:paddingVertical="6dp"
    tools:context=".AddActivity">

```

```
<EditText
    android:id="@+id/name_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Name"
    android:inputType="textPersonName" />

<EditText
    android:id="@+id/email_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:inputType="textEmailAddress" />

<EditText
    android:id="@+id/phone_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Phone Number"
    android:inputType="phone" />

<EditText
    android:id="@+id/address_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Address"
    android:inputType="textPostalAddress" />

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:id="@+id/delete_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="16dp"
        android:layout_toLeftOf="@+id/save_button"
        android:text="Delete" />

    <Button
        android:id="@+id/save_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
```

```
        android:text="Save" />
    </RelativeLayout>
```

```
</LinearLayout>
```

Add Activity

AddActivity.java

```
package com.example.contactslab8;

import android.os.Bundle;
import android.telephony.PhoneNumberUtils;
import android.widget.Button;
import android.widget.EditText;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.room.Room;

import com.example.contactslab8.models.Contact;
import com.example.contactslab8.models.ContactDao;

public class AddActivity extends AppCompatActivity {

    private EditText emailEditText;
    private EditText nameEditText;
    private EditText phoneEditText;
    private EditText addressEditText;
    private Button saveButton;
    private String name;
    private String phone;
    private String address;
    private String email;
    private AppDatabase db;
    private ContactDao contactDao;

    private boolean isValidEmail(CharSequence target) {
        if (target == null)
            return false;

        return
            android.util.Patterns.EMAIL_ADDRESS.matcher(target).matches();
    }
}
```

```

    }

    private void showErrorDialog(String message) {
        new AlertDialog.Builder(this)
            .setIcon(android.R.drawable.stat_notify_error)
            .setTitle("Error")
            .setMessage(message)
            .setNegativeButton("OK", (dialogInterface, i) -> {
                //set what would happen when positive button is
clicked
            })
            .show();
    }

    private void showAddDialogue() {
        new AlertDialog.Builder(this)
            .setIcon(android.R.drawable.ic_menu_save)
            .setTitle("Save")
            .setMessage("Are you sure you want to add new contact?")
            .setPositiveButton("Yes", (dialogInterface, i) -> {
                new Thread(() -> {
                    contactDao.insertAll(new
Contact(System.currentTimeMillis(), name, email, phone, address));
                    runOnUiThread(() -> {
                        setResult(1);
                        finish();
                    });
                }).start();
                //set what would happen when positive button is
clicked
            }).setNegativeButton("No", (dialogInterface, i) -> {

            }).show();
    }

    private void initialiseUI() {
        emailEditText = findViewById(R.id.email_edit_text);
        nameEditText = findViewById(R.id.name_edit_text);
        phoneEditText = findViewById(R.id.phone_edit_text);
        addressEditText = findViewById(R.id.address_edit_text);
        saveButton = findViewById(R.id.save_button);

        saveButton.setOnClickListener(v -> {
            name = nameEditText.getText().toString();
            email = emailEditText.getText().toString();
            phone = phoneEditText.getText().toString();

```

```

        address = addressEditText.getText().toString();

        if (name.length() == 0) {
            showErrorDialog("Name cannot be empty");
            return;
        }

        if (email.length() != 0 && !isValidEmail(email)) {
            showErrorDialog("Invalid email");
            return;
        }

        if (!PhoneNumberUtils.isGlobalPhoneNumber(phone)) {
            showErrorDialog("Invalid Phone");
            return;
        }

        showAddDialogue();
    });
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add);
    db = Room.databaseBuilder(this, AppDatabase.class,
"contacts").build();
    contactDao = db.contactDao();
    initialiseUI();
}
}

```

activity_add.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingHorizontal="6dp"
    android:paddingVertical="6dp"
    tools:context=".AddActivity">

    <EditText
        android:id="@+id/name_edit_text"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:inputType="textPersonName" />

<EditText
    android:id="@+id/email_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:inputType="textEmailAddress" />

<EditText
    android:id="@+id/phone_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Phone Number"
    android:inputType="phone" />

<EditText
    android:id="@+id/address_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Address"
    android:inputType="textPostalAddress" />

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <Button
        android:id="@+id/save_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:text="Save" />

</RelativeLayout>

</LinearLayout>

```

SQL Models

Contact.java

```
package com.example.contacts1ab8.models;

import androidx.room.ColumnInfo;
import androidx.room.Entity;
import androidx.room.PrimaryKey;

import java.io.Serializable;

@Entity
public class Contact implements Serializable {
    @PrimaryKey
    public long uid;
    @ColumnInfo(name = "name")
    public String name;
    @ColumnInfo(name = "email")
    public String email;
    @ColumnInfo(name = "mobile")
    public String mobile;
    @ColumnInfo(name = "address")
    public String address;

    public Contact(long uid, String name, String email, String mobile,
String address) {
        this.uid = uid;
        this.name = name;
        this.email = email;
        this.mobile = mobile;
        this.address = address;
    }
}
```

ContactDao.java

```
package com.example.contactslab8.models;

import androidx.room.Dao;
import androidx.room.Delete;
import androidx.room.Insert;
import androidx.room.Query;
import androidx.room.Update;

import java.util.List;

@Dao
public interface ContactDao {
    @Query("SELECT * FROM contact")
    List<Contact> getAll();

    @Query("SELECT * FROM contact WHERE uid IN (:contactIds)")
    List<Contact> loadAllByIds(int[] contactIds);

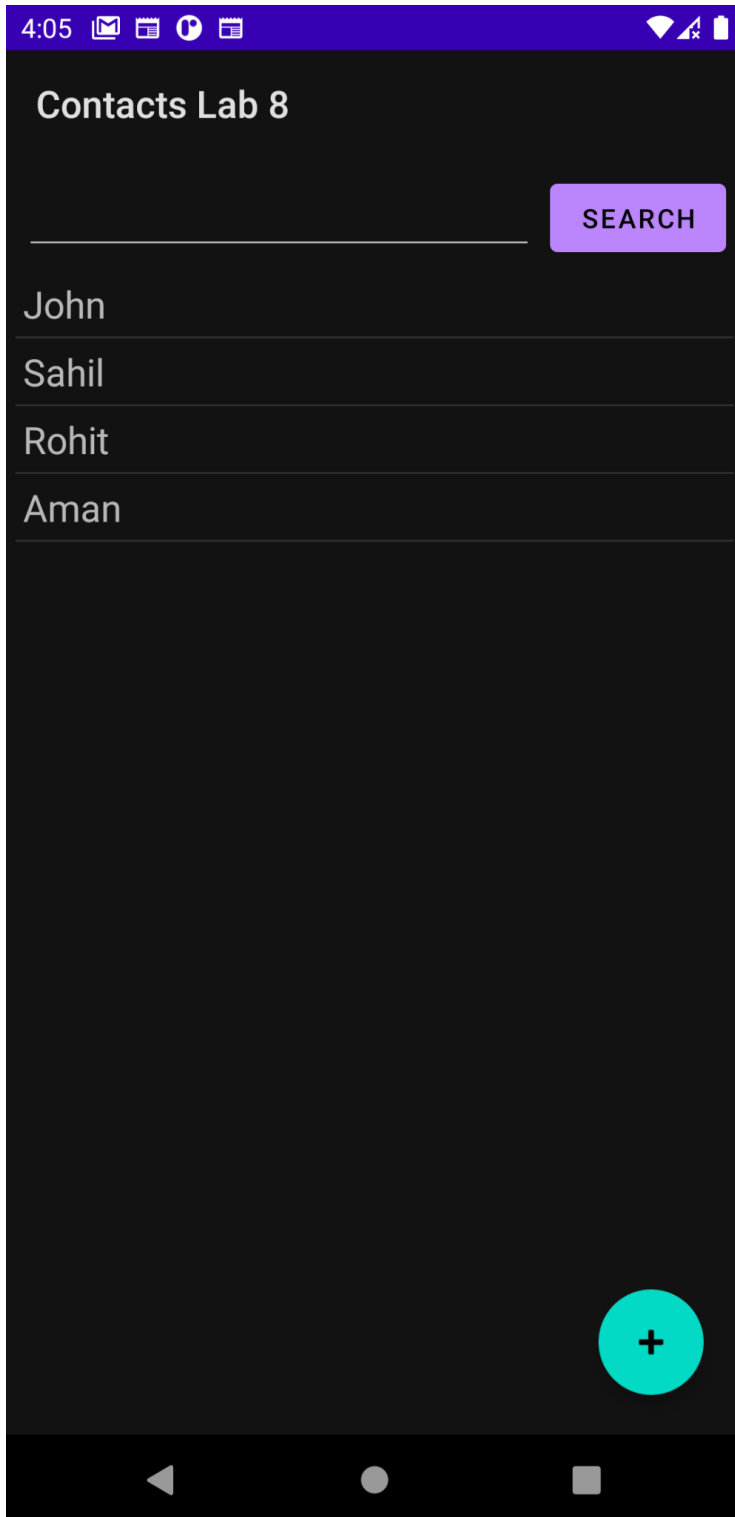
    @Query("SELECT * FROM contact WHERE name LIKE :name LIMIT 1")
    Contact findByName(String name);

    @Insert
    void insertAll(Contact... contacts);

    @Delete
    void delete(Contact contact);

    @Update
    void update(Contact contact);
}
```


Screenshots





Contacts Lab 8

John

john@email.com

123465789

Address

DELETE

SAVE



4:05



Contacts Lab 8

John

john@email.com

123465789

Address

DELETE

SAVE



Save

Are you sure you want to save contact?

NO

YES



4:06



Contacts Lab 8

Name

Email

Phone Number

Address

SAVE

← Contacts Lab 8

Sahil


sahilbondre@gmail.com

1234567989

Mumbai

DELETE

SAVE

 **Delete**

Are you sure you want to delete this contact?

NO

YES