

CO2 -Assignment

2100030127 – Devu Govardhan

First, create a layout file for your user registration form (activity_register.xml)

```
<!-- activity_register.xml -->
```

```
<RelativeLayout
```

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

```
roid"    android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
android:padding="16dp">
```

```
    <EditText
```

```
        android:id="@+id/editTextName"
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:hint="Name"/>
```

```
    <EditText
```

```
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextName"
        android:hint="Email"/>
```

```
    <Button
        android:id="@+id/buttonRegister"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextEmail"
        android:layout_centerHorizontal="true"
        android:text="Register"/>
```

```
</RelativeLayout>
```

Create a SQLiteOpenHelper subclass to manage your database (DatabaseHelper.java):

```

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

public class DatabaseHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "users.db";
private static final int DATABASE_VERSION = 1;

    public static final String TABLE_NAME = "users";
public static final String COLUMN_ID = "id";    public
static final String COLUMN_NAME = "name";    public
static final String COLUMN_EMAIL = "email";

    private static final String TABLE_CREATE =
"CREATE TABLE " + TABLE_NAME + " (" +

        COLUMN_ID + " INTEGER PRIMARY KEY
AUTOINCREMENT, " +

        COLUMN_NAME + " TEXT, " +
        COLUMN_EMAIL + " TEXT" +

        ")";

```

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

```
@Override public void  
onCreate(SQLiteDatabase db) {  
    db.execSQL(TABLE_CREATE);  
}
```

```
@Override public void onUpgrade(SQLiteDatabase db,  
int oldVersion, int newVersion) {    db.execSQL("DROP  
TABLE IF EXISTS " + TABLE_NAME);    onCreate(db);  
}  
}
```

Create a class to handle CRUD operations
(UserDataSource.java):

```
import android.content.ContentValues; import  
android.content.Context;
```

```
import android.database.Cursor; import  
android.database.SQLException; import  
android.database.sqlite.SQLiteDatabase;
```

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

```
public class UserDataSource {
```

```
    private SQLiteDatabase database;    private  
    DatabaseHelper dbHelper;
```

```
    public UserDataSource(Context context) {  
        dbHelper = new DatabaseHelper(context);  
    }
```

```
    public void open() throws SQLException {  
        database = dbHelper.getWritableDatabase();  
    }
```

```
public void close() {  
dbHelper.close(); }
```

```
public void addUser(String name, String email) {  
ContentValues values = new ContentValues();  
values.put(DatabaseHelper.COLUMN_NAME, name);  
values.put(DatabaseHelper.COLUMN_EMAIL, email);  
database.insert(DatabaseHelper.TABLE_NAME, null, values);  
}
```

```
public List<User> getAllUsers() {  
List<User> users = new ArrayList<>();  
Cursor cursor =  
database.query(DatabaseHelper.TABLE_NAME,  
null, null, null, null, null, null);  
cursor.moveToFirst(); while  
(!cursor.isAfterLast()) { User user =  
cursorToUser(cursor);  
users.add(user);  
cursor.moveToNext();  
}
```

```
        cursor.close();  
        return users;  
    }
```

```
    private User cursorToUser(Cursor cursor) {  
        User user = new User();
```

```
        user.setId(cursor.getLong(cursor.getColumnIndex(DatabaseHelper.COLUMN_ID)));
```

```
        user.setName(cursor.getString(cursor.getColumnIndex(DatabaseHelper.COLUMN_NAME)));
```

```
        user.setEmail(cursor.getString(cursor.getColumnIndex(DatabaseHelper.COLUMN_EMAIL)));
```

```
        return user;  
    }
```

```
    public void deleteUser(User user) {  
        long id = user.getId();  
  
        database.delete(DatabaseHelper.TABLE_NAME,  
            DatabaseHelper.COLUMN_ID + " = " + id, null);  
    }  
}
```

