# 项目总结报告

项目名称： 学生信息管理App

学号姓名：

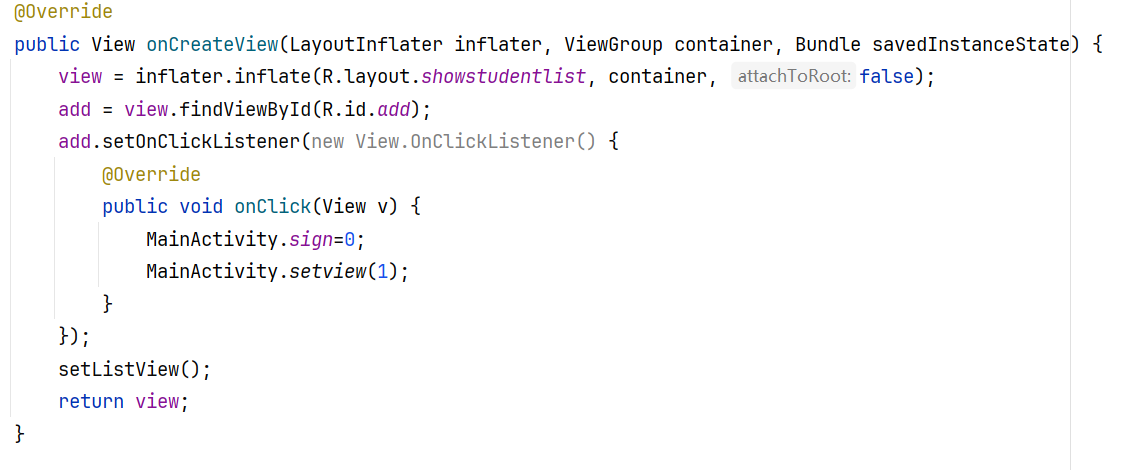
功能设计及运行结果：

该应用程序实现了添加学生，学生列表，点击列表项查看学生详情，还可以在学生详情中对学生信息进行修改。页面使用了frament，使用Room对学生数据进行持久化存储



****

****



这段代码首先通过inflater.inflate(R.layout.showstudentlist, container, false)将布局文件R.layout.showstudentlist转换为View对象，并将其赋值给成员变量view。然后，通过view.findViewById(R.id.add)找到布局中id为add的View，并设置点击事件监听器。在点击事件监听器中，当点击 add 按钮时，会将 MainActivity 的静态变量sign设置为0，调用MainActivity的静态方法setview，并传入参数1。然后，调用setListView()方法来设置ListView的数据源和适配器。最后返回该 View 对象。





这是添加学生的布局文件，由八个线性布局组成，每个线性布局都是水平方向组成页面中的每一个子组件

|  |
| --- |
| <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginRight="10dp"  android:orientation="horizontal">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:text="学号："  ></TextView>   <EditText  android:id="@+id/edit1"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:hint="请输入学号"  ></EditText> </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginRight="10dp"  android:orientation="horizontal">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:text="姓名："  ></TextView>   <EditText  android:id="@+id/edit2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:hint="请输入姓名"  ></EditText> </LinearLayout> <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="15dp"  android:layout\_marginRight="10dp"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:text="性别："></TextView>   <RadioGroup  android:id="@+id/radioGroup"  android:orientation="horizontal"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content">   <RadioButton  android:id="@+id/sexMale"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:text="男" />   <RadioButton  android:id="@+id/sexFemale"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:text="女" />   </RadioGroup> </LinearLayout> <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginRight="10dp"  android:orientation="horizontal">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:text="年龄："  ></TextView>   <EditText  android:id="@+id/edit3"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:hint="请输入年龄"  ></EditText> </LinearLayout> <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="10dp"  android:layout\_marginRight="10dp"  android:orientation="horizontal">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:text="电话："  ></TextView>   <EditText  android:id="@+id/edit4"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textSize="20sp"  android:hint="请输入电话"  ></EditText> </LinearLayout> <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="15dp"  android:layout\_marginRight="10dp"  android:orientation="horizontal">   <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="简介："  android:layout\_gravity="start"  android:textSize="20sp"></TextView>   <EditText  android:id="@+id/edit5"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center"  android:textSize="20sp"  ></EditText> </LinearLayout> <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="10dp"  android:layout\_marginTop="30dp"  android:layout\_marginRight="10dp"  android:orientation="horizontal">   <Button  android:id="@+id/submit"  android:layout\_width="200dp"  android:layout\_height="50dp"  android:text="提交"  ></Button>  <Button  android:id="@+id/back"  android:layout\_width="200dp"  android:layout\_height="50dp"  android:text="返回"  ></Button> </LinearLayout> |

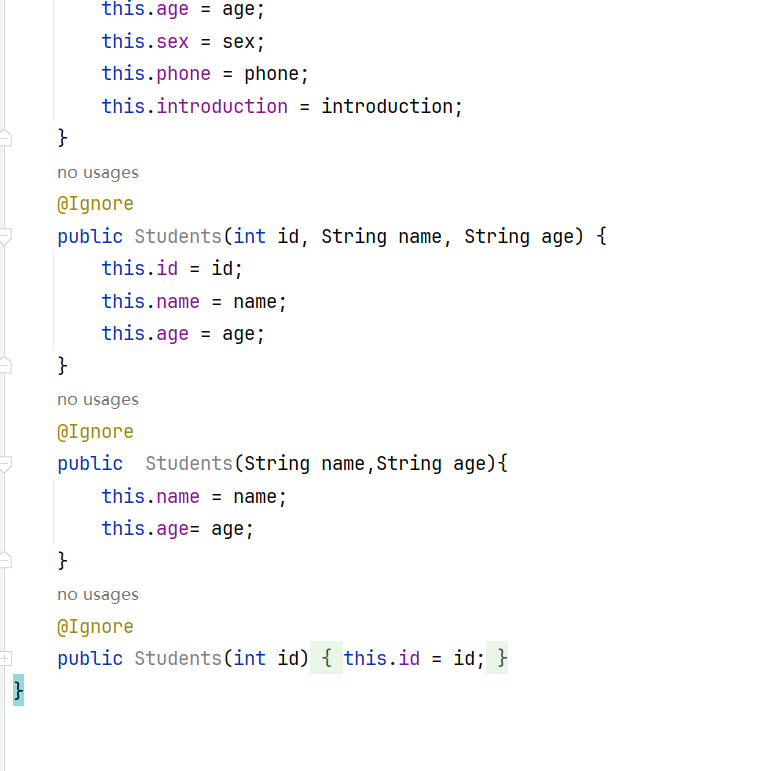
这段代码是一个布局文件，包含了一个顶部的按钮和标题栏以及一个列表视图。根布局是一个垂直方向的LinearLayout，通过 xmlns:android="http://schemas.android.com/apk/res/android"和xmlns:tools="http://schemas.android.com/tools"声明了命名空间，根布局设置了宽度和高度都为 match\_parent，即填充满父容器，xmlns:app="http://schemas.android.com/apk/res-auto"声明了自定义属性的命名空间，根布局中有一个水平方向的 LinearLayout，用于包含顶部的按钮和标题栏，该LinearLayout设置了宽度为match\_parent、高度为wrap\_content，并使用android:background="@drawable/myedittext2"` 设置背景样式。TextView，宽度设置为0d，利用android:layout\_weight="1"进行权重分配，高度为 50dp，文字居中显示，大小为 19dp，颜色为黑色。用于显示标题栏的文本内容为学生列表。





这段代码使用Room持久化库的Android数据库实体类。@Entity(tableName = "students")`用于声明该类是一个 Room 实体类，并指定了数据库表名为 "students"。public class Students定义了一个Students类，用于表示数据库中的学生数据。@PrimaryKey(autoGenerate = true)表示 id 字段是主键，并且通过autoGenerate = true 设置自增。@ColumnInfo(name = "id", typeAffinity = ColumnInfo.INTEGER)指定了id字段在数据库中对应的列名为 "id"，类型为INTEGER。name, age, sex, phone, introduction分别表示学生的姓名、年龄、性别、电话号码和简介。它们都使用了@ColumnInfo注解，指定了在数据库中的列名和数据类型。public Students(int id, String name, String age, String sex, String phone, String introduction)用于初始化Student 类的实例对象。@Ignore注解的三个构造函数被用来忽略Room 框架处理的情况。有些时候我们不需要传递所有字段的数据，可以通过这些 Ignore 注解的构造函数来创建对象。例如：public Students(int id, String name, String age)就是忽略了 sex、phone、introduction 字段。





实现思路及心得：

学生信息管理app在设计时就包括对 学生信息的增删改查功能，以及对学生信息进行展示和编辑的界面。首先，需要设计数据库表来存储学生信息，包括学号、姓名、年龄、性别等字段。使用Room库来实现数据库操作。项目中创建学生信息的数据模型类，用于表示学生信息，并提供相应的方法来获取和设置学生信息。设计学生信息管理界面的布局，包括显示学生信息列表的界面和编辑学生信息的界面，使用EditText等控件来编辑学生信息。编写数据库操作的方法，包括增删改查等功能。通过调用数据库操作方法来实现对学生信息的管理。编写界面逻辑的代码，包括在界面中显示学生信息列表，并响应用户的操作来添加、删除、编辑和查看学生信息；这个大作业是一个很好的理论结合实践的方法，通过实践可以加深对这门课的理解和掌握。在开发过程中，我学会了如何设计数据库结构、编写数据库操作代码，以及如何设计界面布局和实现界面逻辑。同时，我也学会了如何进行测试和调试，以确保程序的稳定性和可靠性；这个项目不足的地方就是没有提供登录功能和头像上传，可以改进成一个需要登陆使用的系统，并且开放头像上传功能和支持不同人员的登录。经过这学期的学习，我不仅学会了安卓开发的基础知识和技能，还了解了安卓开发的一些学习方法。通过课程的学习和实践我对安卓开发有了更深入的了解，并且能够独立开发一些简单的安卓应用程序。同时，我也学会了如何查阅安卓开发文档和资料。安卓课程的学习对我在安卓开发领域的成长和进步起到了非常重要的作用。