Сначала:

1. (например, значок списка или галочки).
2. Сохрани её как PNG-файл, например, ic\_todo.png.
3. Помести файл в папку app/src/main/res/drawable.

#### Оценка 3:

* Реализовано добавление и удаление элементов через кнопки ADD и REMOVE.
* Использован ListView для отображения списка.
* Интерфейс соответствует скриншоту, компоненты имеют осмысленные ID: btnAdd, btnModify, btnRemove, lstItems.

#### Оценка 4:

* Текстовые надписи вынесены в strings.xml.
* Реализовано изменение элемента через AlertDialog с текстовым полем и кнопками "OK" и "Cancel".
* Добавлена проверка на выбор элемента перед изменением или удалением.

#### Оценка 5:

* При удалении элемента отображается диалоговое окно с подтверждением (Yes/No).
* Установлен заголовок и пользовательская иконка (ic\_todo) для диалога удаления.

#### Дополнительные функции:

* Список сохраняет структуру "Item\nSub item" (можно редактировать при необходимости).
* После добавления, изменения или удаления элемента выбор сбрасывается (selectedItemPosition = -1).

Переходим к кодам, их просто вставить и будет счастье.

Activity\_main.xaml:

<?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:padding="16dp">  
  
 <!-- Кнопки -->  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
 <Button  
 android:id="@+id/btnAdd"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/btnAdd" />  
 <Button  
 android:id="@+id/btnModify"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:layout\_marginEnd="8dp"  
 android:text="@string/btnModify" />  
 <Button  
 android:id="@+id/btnRemove"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="@string/btnRemove" />  
 </LinearLayout>  
  
 <!-- Список -->  
 <ListView  
 android:id="@+id/lstItems"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:choiceMode="singleChoice" />  
  
</LinearLayout>

String.xaml:

<resources>  
 <string name="app\_name">Список элементов</string>  
 <string name="btnAdd">ДОБАВИТЬ</string>  
 <string name="btnModify">ИЗМЕНИТЬ</string>  
 <string name="btnRemove">УДАЛИТЬ</string>  
 <string name="dialogAddTitle">Добавить новый элемент</string>  
 <string name="dialogAddButton">ДОБАВИТЬ</string>  
 <string name="dialogModifyTitle">Изменить элемент</string>  
 <string name="dialogModifyButtonOk">ОК</string>  
 <string name="dialogModifyButtonCancel">Отмена</string>  
 <string name="dialogRemoveTitle">Подтверждение удаления</string>  
 <string name="dialogRemoveMessage">Вы уверены, что хотите удалить этот элемент?</string>  
 <string name="dialogRemoveButtonYes">Да</string>  
 <string name="dialogRemoveButtonNo">Нет</string>  
 <string name="errorNoItemSelected">Пожалуйста, выберите элемент!</string>  
 <string name="errorEmptyInput">Элемент не может быть пустым!</string>  
</resources>

MainActivity.java

package com.example.itemlist;  
  
import android.content.SharedPreferences;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ListView;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.ArrayList;  
import java.util.Arrays;  
  
public class MainActivity extends AppCompatActivity {  
  
 private Button btnAdd, btnModify, btnRemove;  
 private ListView lstItems;  
 private ArrayList<String> itemsList;  
 private ArrayAdapter<String> adapter;  
 private int selectedItemPosition = -1;  
 private static final String *PREFS\_NAME* = "ItemListPrefs";  
 private static final String *KEY\_ITEMS* = "items";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 // Инициализация компонентов  
 btnAdd = findViewById(R.id.*btnAdd*);  
 btnModify = findViewById(R.id.*btnModify*);  
 btnRemove = findViewById(R.id.*btnRemove*);  
 lstItems = findViewById(R.id.*lstItems*);  
  
 // Инициализация списка  
 itemsList = new ArrayList<>();  
 if (savedInstanceState != null) {  
 itemsList = savedInstanceState.getStringArrayList("itemsList");  
 selectedItemPosition = savedInstanceState.getInt("selectedPosition");  
 } else {  
 // Загрузка из SharedPreferences  
 SharedPreferences prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
 String savedItems = prefs.getString(*KEY\_ITEMS*, "");  
 if (!savedItems.isEmpty()) {  
 itemsList.addAll(Arrays.*asList*(savedItems.split(";;")));  
 } else {  
 // Начальные данные  
 itemsList.add("Расходы за месяц\nВсего 55.000 рублей");  
 itemsList.add("Список дел");  
 itemsList.add("Список покупок\nХлеб\nМолоко\nМасло");  
 }  
 }  
  
 // Настройка адаптера с подсветкой выбранного элемента  
 adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_activated\_1*, itemsList);  
 lstItems.setAdapter(adapter);  
 lstItems.setChoiceMode(ListView.*CHOICE\_MODE\_SINGLE*);  
  
 // Восстановление выбора  
 if (selectedItemPosition != -1) {  
 lstItems.setItemChecked(selectedItemPosition, true);  
 }  
  
 // Выбор элемента  
 lstItems.setOnItemClickListener((parent, view, position, id) -> {  
 selectedItemPosition = position;  
 lstItems.setItemChecked(position, true);  
 });  
  
 // Обработчик кнопки ADD  
 btnAdd.setOnClickListener(v -> showAddDialog());  
  
 // Обработчик кнопки MODIFY  
 btnModify.setOnClickListener(v -> {  
 if (selectedItemPosition == -1) {  
 new AlertDialog.Builder(MainActivity.this)  
 .setMessage(getString(R.string.*errorNoItemSelected*))  
 .setPositiveButton("OK", null)  
 .show();  
 return;  
 }  
 showModifyDialog();  
 });  
  
 // Обработчик кнопки REMOVE  
 btnRemove.setOnClickListener(v -> {  
 if (selectedItemPosition == -1) {  
 new AlertDialog.Builder(MainActivity.this)  
 .setMessage(getString(R.string.*errorNoItemSelected*))  
 .setPositiveButton("OK", null)  
 .show();  
 return;  
 }  
 showRemoveDialog();  
 });  
 }  
  
 // Сохранение состояния при повороте экрана  
 @Override  
 protected void onSaveInstanceState(Bundle outState) {  
 super.onSaveInstanceState(outState);  
 outState.putStringArrayList("itemsList", itemsList);  
 outState.putInt("selectedPosition", selectedItemPosition);  
 }  
  
 // Сохранение списка в SharedPreferences  
 @Override  
 protected void onPause() {  
 super.onPause();  
 SharedPreferences prefs = getSharedPreferences(*PREFS\_NAME*, *MODE\_PRIVATE*);  
 SharedPreferences.Editor editor = prefs.edit();  
 editor.putString(*KEY\_ITEMS*, String.*join*(";;", itemsList));  
 editor.apply();  
 }  
  
 // Диалог для добавления нового элемента  
 private void showAddDialog() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle(getString(R.string.*dialogAddTitle*));  
  
 final EditText input = new EditText(this);  
 builder.setView(input);  
  
 builder.setPositiveButton(getString(R.string.*dialogAddButton*), (dialog, which) -> {  
 String newItem = input.getText().toString().trim();  
 if (newItem.isEmpty()) {  
 new AlertDialog.Builder(MainActivity.this)  
 .setMessage(getString(R.string.*errorEmptyInput*))  
 .setPositiveButton("OK", null)  
 .show();  
 return;  
 }  
 itemsList.add(newItem);  
 adapter.notifyDataSetChanged();  
 selectedItemPosition = -1;  
 lstItems.clearChoices();  
 });  
  
 builder.setNegativeButton(getString(R.string.*dialogModifyButtonCancel*), (dialog, which) -> dialog.cancel());  
 builder.show();  
 }  
  
 // Диалог для изменения элемента  
 private void showModifyDialog() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle(getString(R.string.*dialogModifyTitle*));  
  
 final EditText input = new EditText(this);  
 input.setText(itemsList.get(selectedItemPosition));  
 builder.setView(input);  
  
 builder.setPositiveButton(getString(R.string.*dialogModifyButtonOk*), (dialog, which) -> {  
 String modifiedItem = input.getText().toString().trim();  
 if (modifiedItem.isEmpty()) {  
 new AlertDialog.Builder(MainActivity.this)  
 .setMessage(getString(R.string.*errorEmptyInput*))  
 .setPositiveButton("OK", null)  
 .show();  
 return;  
 }  
 itemsList.set(selectedItemPosition, modifiedItem);  
 adapter.notifyDataSetChanged();  
 selectedItemPosition = -1;  
 lstItems.clearChoices();  
 });  
  
 builder.setNegativeButton(getString(R.string.*dialogModifyButtonCancel*), (dialog, which) -> dialog.cancel());  
 builder.show();  
 }  
  
 // Диалог для подтверждения удаления  
 private void showRemoveDialog() {  
 AlertDialog.Builder builder = new AlertDialog.Builder(this);  
 builder.setTitle(getString(R.string.*dialogRemoveTitle*))  
 .setIcon(R.drawable.*ic\_todo*)  
 .setMessage(getString(R.string.*dialogRemoveMessage*))  
 .setPositiveButton(getString(R.string.*dialogRemoveButtonYes*), (dialog, which) -> {  
 itemsList.remove(selectedItemPosition);  
 adapter.notifyDataSetChanged();  
 selectedItemPosition = -1;  
 lstItems.clearChoices();  
 })  
 .setNegativeButton(getString(R.string.*dialogRemoveButtonNo*), (dialog, which) -> dialog.cancel());  
 builder.show();  
 }  
}