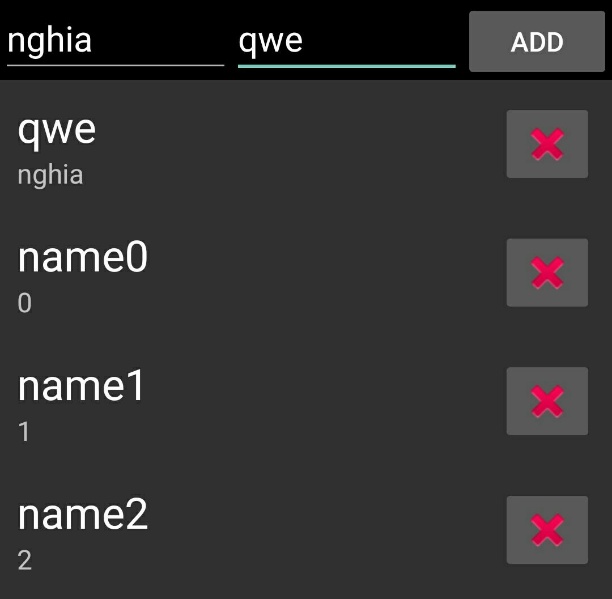
**RECYCLE VIEW**

Hiển thị danh sách lên giao diện như ListView.

Hiển thị đa dạng hơn.

Bước 1: Thêm RecycleView trong: **build.gradle**

dependencies {

// RecyclerView

implementation 'com.android.support:recyclerview-v7:26.1.0'

}

Bước 2: Tạo trên giao diện activity layout:

<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"

tools:context="com.imark.nghia.nghiacontainer\_recyclerview\_likelistview.MainActivity">

//Thêm mới

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:background="#000000">

<EditText

android:id="@+id/main\_edit\_text\_code"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:hint="Name"/>

<EditText

android:id="@+id/main\_edit\_text\_name"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:hint="Code"/>

<Button

android:id="@+id/main\_button\_add"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Add"/>

</LinearLayout>

//Hiển thị danh sách

<android.support.v7.widget.RecyclerView

android:id="@+id/main\_recycler\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:scrollbars="vertical"/>

</LinearLayout>

Tạo giao diện Item: layout/item\_recyclerview.xml

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center\_vertical"

android:padding="10dp">

<ImageButton

android:id="@+id/button\_delete"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:src="@android:drawable/ic\_delete"

android:layout\_alignParentRight="true"

android:layout\_alignParentEnd="true"

android:layout\_centerInParent="true"/>

<TextView

android:id="@+id/text\_view\_name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textAppearance="?android:textAppearanceLarge"

android:layout\_toLeftOf="@+id/button\_delete"

android:layout\_toStartOf="@+id/button\_delete"/>

<TextView

android:id="@+id/text\_view\_code"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/text\_view\_name"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:layout\_toLeftOf="@+id/button\_delete"

android:layout\_toStartOf="@+id/button\_delete"/>

</RelativeLayout>

Bước 2: Tạo ItemModel để lưu trữ Item: ItemModel.java

public class ItemModel {

public String Code;

public String Name;

public ItemModel(String code, String name) {

this.Code = code;

this.Name = name;

}

}

Khởi tạo Adapter kết nối Item: CustomRecyclerAdapter.java

public class CustomRecyclerAdapter extends RecyclerView.Adapter<CustomRecyclerAdapter.RecyclerViewHolder> {

private List<ItemModel> mListItem;

public CustomRecyclerAdapter(List<ItemModel> listItem) {

this.mListItem = listItem;

}

@Override

public RecyclerViewHolder onCreateViewHolder(ViewGroup viewGroup, int position) {

// Khởi tạo giao diện

LayoutInflater inflater = LayoutInflater.from(viewGroup.getContext());

View itemView = inflater.inflate(R.layout.item\_recyclerview, viewGroup, false);

return new RecyclerViewHolder(itemView);

}

@Override

public void onBindViewHolder(RecyclerViewHolder recyclerViewHolder, int position) {

// Hiển thị giá trị lên giai diện

ItemModel itemModel = mListItem.get(position);

recyclerViewHolder.mTextView\_Name.setText(itemModel.Name);

recyclerViewHolder.mTextView\_Code.setText(itemModel.Code);

}

@Override

public int getItemCount() {

return mListItem.size();

}

/\*\*ViewHolder for item view of list\*/

public class RecyclerViewHolder extends RecyclerView.ViewHolder {

TextView mTextView\_Code, mTextView\_Name;

ImageButton mButton\_Delete;

public RecyclerViewHolder(View itemView){

super(itemView);

// ----- Find View ----

mTextView\_Code = (TextView) itemView.findViewById(R.id.text\_view\_code);

mTextView\_Name = (TextView) itemView.findViewById(R.id.text\_view\_name);

mButton\_Delete = (ImageButton) itemView.findViewById(R.id.button\_delete);

// ----- Init ----- Nút xoá Item

mButton\_Delete.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

mListItem.remove(getAdapterPosition());

notifyItemRemoved(getAdapterPosition());

}

});

}

}

}

Bước 3: Sử dụng trong Activity

EditText mEditText\_Name, mEditText\_Code;

Button mButton\_Add;

private RecyclerView mRecyclerView;

List<ItemModel> mListItem;

CustomRecyclerAdapter mAdapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

//...

findViews();

initVars();

}

private void findViews() {

mEditText\_Code = (EditText) findViewById(R.id.main\_edit\_text\_code);

mEditText\_Name = (EditText) findViewById(R.id.main\_edit\_text\_name);

mButton\_Add = (Button) findViewById(R.id.main\_button\_add);

mRecyclerView = (RecyclerView) findViewById(R.id.main\_recycler\_view);

}

private void initVars() {

//Tao Danh Sach

mListItem = new ArrayList<ItemModel>(){{

for(int i=0; i<100; i++){

add(new ItemModel(""+i, "name"+i));

}

}};

// If the size of views will not change as the data changes.

mRecyclerView.setHasFixedSize(true);

// Setting the LayoutManager.

LinearLayoutManager layoutManager = new LinearLayoutManager(this);

mRecyclerView.setLayoutManager(layoutManager);

mAdapter = new CustomRecyclerAdapter(mListItem);

mRecyclerView.setAdapter(mAdapter);

//Thêm Item tại Vị trí đầu tiên

mButton\_Add.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

mListItem.add(0, new ItemModel(

mEditText\_Code.getText()+"",

mEditText\_Name.getText()+""));

mAdapter.notifyDataSetChanged();

}

});

}