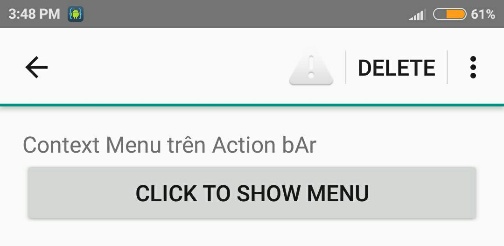
**Contextual Action Bar (CAB), Menu Phụ trên Action bar.**

Vẫn khởi tại xml menu: menu/menu\_context.xml



**Nhấn Button hiện Menu.**

private ActionMode mActionMode;

@Override

protected void onCreate(Bundle savedInstanceState) {

...

View button = findViewById(R.id.button);

button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

//Đang hiện Menu thì ko thực hiện

if (mActionMode != null) {

return;

}

// Bắt đầu CAB bằng cách dùng ActionMode.Callback

mActionMode = startActionMode(mActionModeCallback);

view.setSelected(true);

}

});

}

// Button Click => hiện menu

private ActionMode.Callback mActionModeCallback = new ActionMode.Callback() {

@Override

public boolean onCreateActionMode(ActionMode mode, Menu menu) {

// Gọi khi tạo ActionMode, startActionMode() gọi

//Gọi MenuResource thông qua Inflater

MenuInflater inflater = mode.getMenuInflater();

inflater.inflate(R.menu.menu\_main, menu);

return true;

}

@Override

public boolean onPrepareActionMode(ActionMode mode, Menu menu) {

// Gọi khi ActioMode shown, sau onCreateActionMode

// Có thể gọi nhiều lần nếu Mode = Invalidated

return false; // Return false if nothing is done

}

@Override

public boolean onActionItemClicked(ActionMode mode, MenuItem item) {

// Gọi khi chọn Menu Item

switch (item.getItemId()) {

case R.id.action\_click\_me:

Toast.makeText(getBaseContext(), "add action", Toast.LENGTH\_SHORT).show();

mode.finish(); // Action picked, so close the CAB

return true;

default:

return false;

}

}

@Override

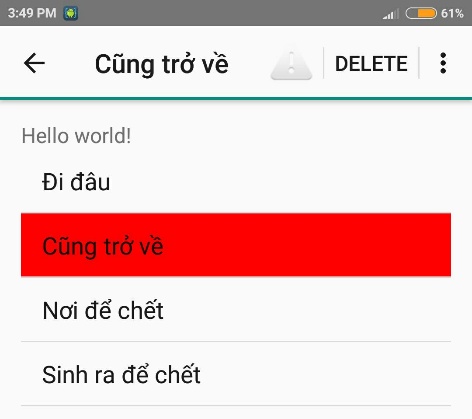
public void onDestroyActionMode(ActionMode mode) {

// Gọi khi Thoát ActionMode

mActionMode = null;

}

};

**Nhấn giữ ListView Item hiện Menu**

Bước 1: Tạo XML menu: menu/menu\_content.xml

Bước 2: tạo ListView trong Activity

<ListView ...

android:id="@+id/listView"

android:background="?android:attr/activatedBackgroundIndicator"

android:listSelector="#FF00"/>

Bước 3: Tạo trong Activity java

ListView listView;

private String[] arrayString;

private ArrayAdapter<String> adapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

//...

initListView();

setListViewEvents();

}

private void initListView() {

listView = (ListView) findViewById(R.id.listView);

arrayString = getResources().getStringArray(R.array.nghia\_array);

adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_1, arrayString);

listView.setChoiceMode(ListView.CHOICE\_MODE\_MULTIPLE\_MODAL);

listView.setAdapter(adapter);

}

Bước 4: tạo sự kiện ListView, hiện menu.

private void setListViewEvents() {

// Nhấn chọn Item => show menu

listView.setMultiChoiceModeListener(new AbsListView.MultiChoiceModeListener() {

@Override

public void onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked) {

// Gọi khi Item select/ Deselect

// Cập nhật Tiêu đề trên CAB

mode.setTitle(arrayString[position]);

}

@Override

public boolean onCreateActionMode(ActionMode mode, Menu menu) {

// tạo menu CAB qua Inflater

MenuInflater inflater = mode.getMenuInflater();

inflater.inflate(R.menu.menu\_context, menu);

return true;

}

@Override

public boolean onPrepareActionMode(ActionMode mode, Menu menu) {

// Here you can perform updates to the CAB due to

// an invalidate() request

return false;

}

@Override

public boolean onActionItemClicked(ActionMode mode, MenuItem item) {

// Gọi khi Item Menu Click

switch (item.getItemId()) {

case R.id.action\_click\_me:

Toast.makeText(getBaseContext(), "Action Add", Toast.LENGTH\_SHORT).show();

mode.finish(); // Action picked, so close the CAB

return true;

default:

return false;

}

}

@Override

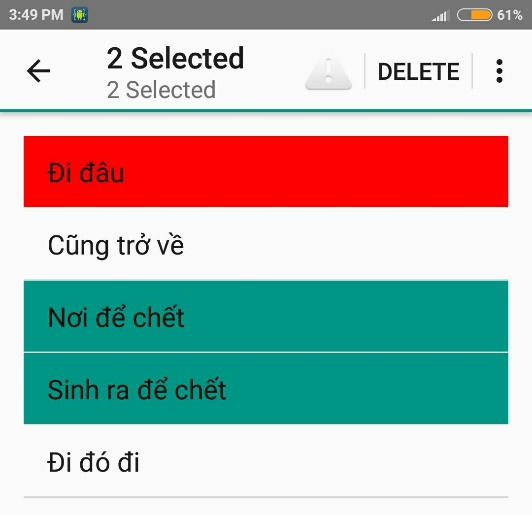
public void onDestroyActionMode(ActionMode mode) {

// Gọi khi CAB remove. Dùng để Select/ Unselected

}

});

}

**Đếm Item chọn trong ListView**

Lặp lại bước 1, 2, 3

Bước 4: tạo sự kiện ListView, hiện menu.

private void setListViewEvents() {

// Long click Listview => show menu

mListView.setMultiChoiceModeListener(new AbsListView.MultiChoiceModeListener() {

@Override

public void onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked) {

// Here you can do something when items are selected/de-selected,

// Đếm & hiện Item chọn

final int checkedCount = mListView.getCheckedItemCount();

mode.setTitle(checkedCount + " Selected");

mode.setSubtitle(checkedCount + " Selected");

}

@Override

public boolean onCreateActionMode(ActionMode mode, Menu menu) {

// tạo CAB menu qua Inflater

MenuInflater inflater = mode.getMenuInflater();

inflater.inflate(R.menu.menu\_context, menu);

return true;

}

@Override

public boolean onPrepareActionMode(ActionMode mode, Menu menu) {

// Here you can perform updates to the CAB due to

// an invalidate() request

return false;

}

@Override

public boolean onActionItemClicked(ActionMode mode, MenuItem item) {

// Respond to clicks on the actions in the CAB

switch (item.getItemId()) {

case R.id.action\_click\_me:

Toast.makeText(getBaseContext(), "Action Add", Toast.LENGTH\_SHORT).show();

mode.finish(); // Action picked, so close the CAB

return true;

case R.id.action\_delete:

//Calls getSelectedIds method from ListViewAdapter Class

// Xóa

int len = mListView.getCount();

SparseBooleanArray checked = mListView.getCheckedItemPositions();

for (int i = 0; i < len; i++)

if (checked.get(i)) {

// thông báo xóa

String selectedItem = mListString.get(i);

Toast.makeText(getBaseContext(), "Xóa: " + selectedItem, Toast.LENGTH\_SHORT).show();

// Xóa item

mListString.remove(i);

mAdapter.notifyDataSetChanged();

}

mode.finish(); //close CAB

return true;

default:

return false;

}

}

@Override

public void onDestroyActionMode(ActionMode mode) {

// Here you can make any necessary updates to the activity when

// the CAB is removed. By default, selected items are deselected/unchecked.

}

});

}