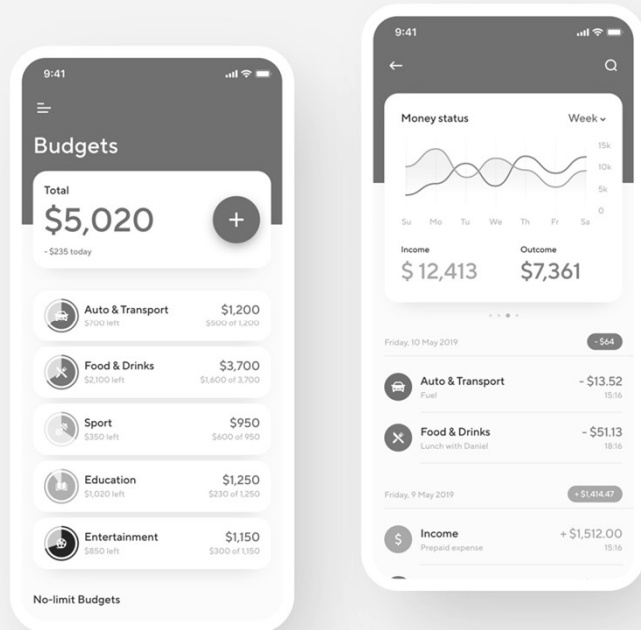


NGUYỄN CHÍ HIẾU

LISTVIEW & MENU



1

NỘI DUNG

1. ListView

2. Menu

2

Giới thiệu ListView

ListView và Adapter

LISTVIEW

Tạo ListView tùy biến

Nguyễn Chí Hiếu

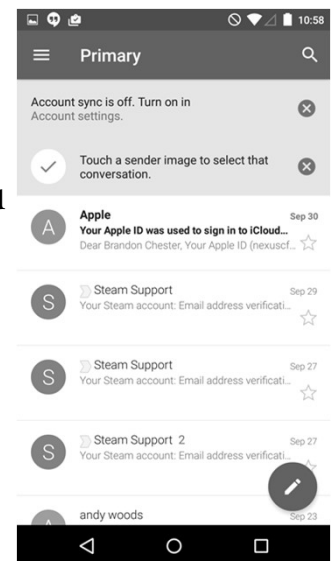
2017 3

3

GIỚI THIỆU LISTVIEW

• ListView

- Là View hiển thị danh sách gồm nhiều phần tử/dòng (hỗ trợ thanh cuộn).
- Dữ liệu của danh sách có thể khai báo tĩnh hay động (đọc từ tập tin, cơ sở dữ liệu, Internet)



Nguyễn Chí Hiếu

2017 4

4

HIỂN THỊ LISTVIEW VỚI DỮ LIỆU TĨNH

• Dữ liệu tĩnh

- Dữ liệu được khởi tạo trước khi chạy ứng dụng.
- Các phần tử của danh sách được khai báo trong *res/values/strings.xml*.
- Trong tập tin thiết kế giao diện, thêm thuộc tính `android:entries="@array/___"`: nạp dữ liệu tĩnh cho ListView

HIỂN THỊ LISTVIEW VỚI DỮ LIỆU TĨNH

• Dữ liệu tĩnh

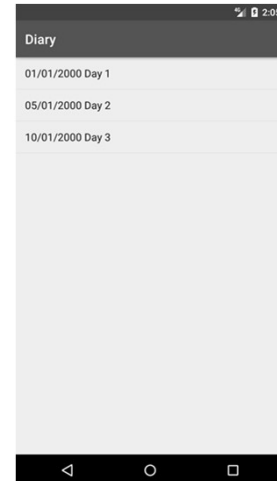
```
<resources>
    ...
    <string-array name="entries">
        <item>01/01/2000 Day 1</item>
        <item>05/01/2000 Day 2</item>
        <item>10/01/2000 Day 3</item>
    </string-array>
</resources>
```

HIỂN THỊ LISTVIEW VỚI DỮ LIỆU TĨNH

• Ví dụ 1: Tạo ứng dụng Diary

- Thêm vào một Activity và đặt tên *EntryListActivity*.
- Trong *activity_entry_list.xml*, thêm *ListView* với các thuộc tính sau:

```
<ListView
    android:id="@+id/listEntry "
    android:entries="@array/entries"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</ListView>
```



Nguyễn Chí Hiếu

2017

7

7

HIỂN THỊ LISTVIEW VỚI DỮ LIỆU ĐỘNG

• Dữ liệu động

- Dữ liệu được đọc hay sinh ra khi chạy ứng dụng và cài đặt trong tập tin **.java*.
- Dữ liệu này có thể đọc từ: tập tin, cơ sở dữ liệu SQLite, Web services, ...

Nguyễn Chí Hiếu

2017

8

8

```

private ListView listEntry;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_entry_list);

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

    loadEntries();
}

```

- Trong EntryListActivity.java

HIỂN THỊ LISTVIEW VỚI DỮ LIỆU ĐỘNG

Nguyễn Chí Hiếu

2017 9

9

Nguyễn Chí Hiếu

2. LISTVIEW & ADAPTER

2017 10

10

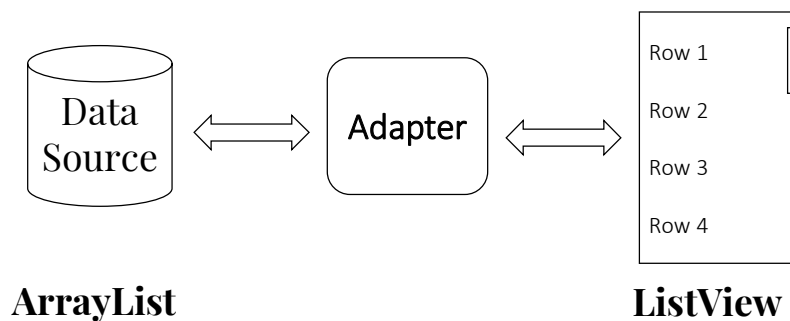
LISTVIEW VÀ ADAPTER

• Adapter

- Là cầu nối giữa Data Source và Adapter View (ListView, Spinner, ...)
- Adapter là lớp đối tượng tạo và quản lý các dòng trong ListView.

LISTVIEW VÀ ADAPTER

• Adapter



LISTVIEW VÀ ADAPTER

- ArrayAdapter

- ArrayAdapter tạo một view bằng cách gọi phương thức *toString()* đối với mỗi đối tượng trong danh sách và hiển thị lên một TextView.
- Phương thức khởi tạo: `ArrayAdapter(Context context, int resource, List<T> objects)`
- Trong đó, resource là layout do Android cung cấp sẵn ([link](#)):
 - `simple_list_item_1`,
 - `simple_list_item_2`, ...

LISTVIEW VÀ ADAPTER

- **Bước 1.** Tạo biến chứa dữ liệu cần hiển thị.
- **Bước 2.** Tạo một biến thuộc lớp ArrayAdapter và khởi tạo với tham số gồm:
 - Context của Activity
 - Kiểu layout (Android cung cấp sẵn/tự thiết kế)
 - Dữ liệu
- **Bước 3.** Gọi phương thức *setAdapter()* gán dữ liệu cho ListView.

• Trong *EntryListActivity.java*, thêm phương thức *loadEntries()* nạp dữ liệu.

```
private void loadEntries() {
    ArrayList<String> entries = new ArrayList<String>();
    entries.add("01/01/2000 Day 1");
    // ...
    if (entries != null) {
        ArrayAdapter<String> adapter =
            new ArrayAdapter<String>(
                this,
                android.R.layout.simple_list_item_1,
                entries);
        listView.setAdapter(adapter);
    }
}
```

LISTVIEW VÀ ADAPTER

Nguyễn Chí Hiếu

2017 15

15

LISTVIEW VÀ ADAPTER

- `setOnClickListener(AdapterView.OnItemClickListener)`: nhấn một phần tử trong ListView
- `setOnItemLongClickListener(AdapterView.OnItemLongClickListener)`: nhấn và giữ một phần tử trong ListView
- `setOnItemSelectedListener(AdapterView.OnItemSelectedListener)`: chọn một phần tử trong ListView
- ...

Nguyễn Chí Hiếu

2017 16

16


```
listEntry.setOnItemClickListener(new
    AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view,
                                int position, long id) {
            Toast.makeText(getContext(),
                ((TextView) view).getText(),
                Toast.LENGTH_SHORT).show();
        }
    });
```

- Xử lý sự kiện của ListView
 - Trong *EntryListActivity.java*, thêm vào sự kiện *setOnClickListener()* cho ListView.

LISTVIEW VÀ ADAPTER

2017 17

TẠO LISTVIEW TÙY BIẾN

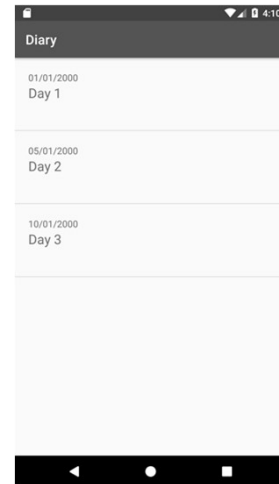
- ArrayAdapter mặc định chỉ sinh ra các dữ liệu dạng chuỗi tương ứng mỗi dòng của ListView.
- Mỗi dòng của ListView cho phép kết hợp từ view lại với nhau nhằm tạo ra ListView phù hợp yêu cầu của ứng dụng.

2017 18

TẠO LISTVIEW TÙY BIẾN

• **Ví dụ 2.** Thiết kế giao diện ứng dụng như hình bên. Mỗi phần tử/dòng của ListView gồm 2 TextView:

- Date: hiển thị ngày
- Content: hiển thị nội dung



TÙY BIẾN LISTVIEW

- **Bước 1.** Thêm một layout để hiển thị tương ứng với một phần tử trong ListView và đặt tên *entry_list_content.xml*.

```
<LinearLayout ...
    android:orientation="vertical">
    <TextView
        android:id="@+id/textDate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/textContent"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:textSize="20sp" />
</LinearLayout>
```

TẠO LISTVIEW TÙY BIẾN

Nguyễn Chí Hiếu

2017 21

21

TẠO LISTVIEW TÙY BIẾN

- **Bước 2.** Thêm một lớp đối tượng mới và đặt tên **EntryModel** với 2 thuộc tính date và content.

```
public class EntryModel {
    private String date;
    private String content;
    public EntryModel() {}
    public EntryModel(String date, String content)
    {
        this.date = date;
        this.content = content;
    }
}
```

Nguyễn Chí Hiếu

2017 22

22

TẠO LISTVIEW TÙY BIẾN

- **Bước 3.** Thêm một lớp **EntryAdapter** kế thừa từ lớp **ArrayAdapter** và phương thức khởi tạo cho lớp này.

```
public class EntryAdapter extends
    ArrayAdapter<EntryModel> {
    private final int mResourceId;
    public EntryAdapter(Context context,
        int resource,
        List<EntryModel> objects) {
        super(context, resource, objects);
        mResourceId = resource;
    }
}
```

Nguyễn Chí Hiếu

2017 23

23

TẠO LISTVIEW TÙY BIẾN

- Phương thức khởi tạo EntryAdapter(_, _, _)
- context
- resource: id của layout
- object: data source

Nguyễn Chí Hiếu

2017 24

24

TẠO LISTVIEW TÙY BIẾN

- Phương thức khởi tạo EntryAdapter(_, _, _)

```
public class EntryAdapter extends
    ArrayAdapter<EntryModel> {
    private final int mResourceId;
    public EntryAdapter(Context context,
        int resource,
        List<EntryModel> objects) {
        super(context, resource, objects);
        mResourceId = resource;
    }
}
```

Nguyễn Chí Hiếu

2017 25

25

- **Bước 4.** Override phương thức *getView()*

```
@Override
public View getView(int position,
    View convertView, ViewGroup parent) {
    View row = convertView;
    EntryModel entry = getItem(position);
    if (row == null) {
        row = LayoutInflater.from(getContext())
            .inflate(mResourceId, parent, false);
    }
    TextView textDate = (TextView) row.findViewById(R.id.textDate);
    textDate.setText(entry.getDate());
    //...
    return row;
}
```

Nguyễn Chí Hiếu

2017 26

26

• **Bước 5.** Trong *EntryListActivity.java*, viết lại phương thức *loadEntries()*

```
private void loadEntries() {
    ArrayList<EntryModel> entries = new ArrayList<>();
    EntryModel entry = null;
    entry = new EntryModel("01/01/2000", "Day 1");
    entries.add(entry);
    // ...
    if (entries != null) {
        EntryAdapter adapter = new EntryAdapter(
            getApplicationContext(),
            R.layout.entry_list_content,
            entries);
        listEntry.setAdapter(adapter);
    }
}
```

Nguyễn Chí Hiếu

2017 27

27

XỬ LÝ SỰ KIỆN CỦA LISTVIEW

• Trong *EntryListActivity.java*, thay đổi mã nguồn sự kiện *setOnItemClickListener()*

```
listEntry.setOnItemClickListener(new AdapterView
    .OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView,
        View view, int position, long id) {
        EntryModel entry = (EntryModel) listEntry
            .getItemAtPosition(position);
        Toast.makeText(getApplicationContext(), entry.getDate(),
            Toast.LENGTH_SHORT).show();
    }
});
```

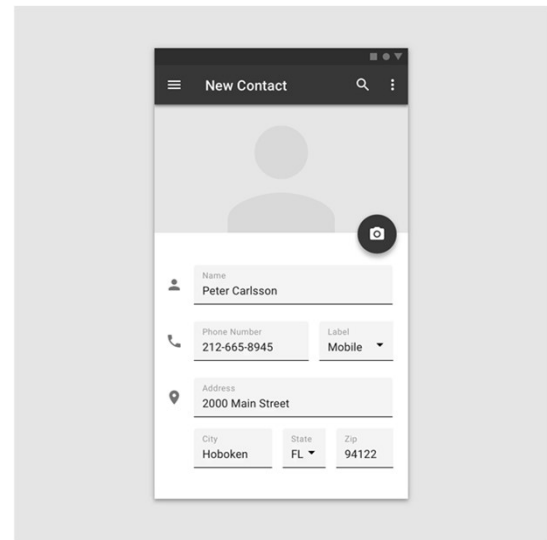
Nguyễn Chí Hiếu

2017 28

28

3. MENU

- Giới thiệu Menu
- Options Menu
- Context Menu
- Popup Menu

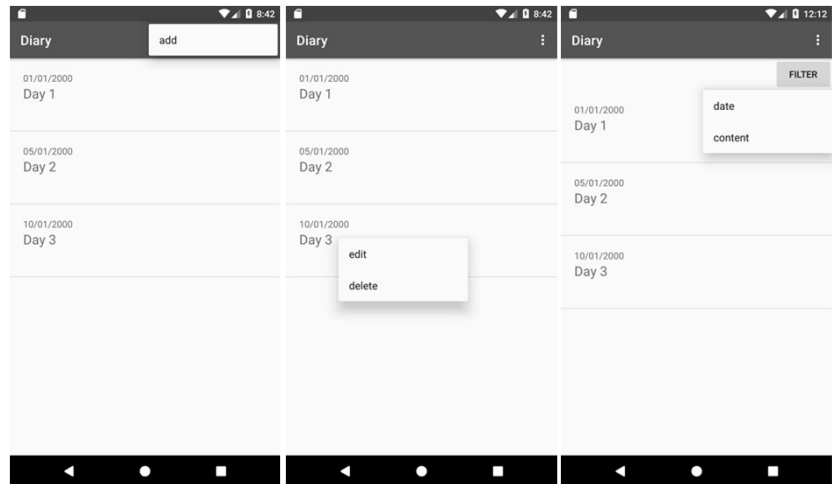


GIỚI THIỆU MENU

- Menu
 - Là một trong những thành phần quan trọng khi thiết kế giao diện ứng dụng.
 - Hiện thị một số thao tác chuyển hướng hay xử lý đối với một Activity hay View nào đó của ứng dụng.

GIỚI THIỆU MENU

- Phân loại
 - Options Menu
 - Context Menu
 - Popup Menu



Nguyễn Chí Hiếu

2017 31

31

GIỚI THIỆU MENU

- Thêm menu mới cho ứng dụng trong res/menu
 - Thêm menu từ **.xml* hay **.java*
 - Thêm các ảnh cho menu (nếu cần)
 - Các thành phần của menu
 - menu
 - item
 - group

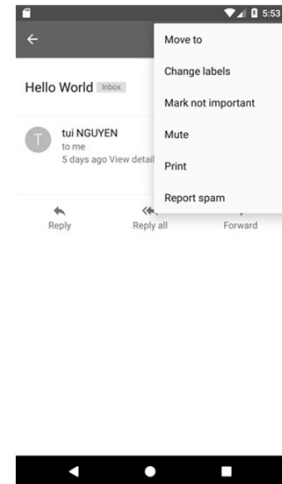
Nguyễn Chí Hiếu

2017 32

32

OPTIONS MENU

- Tập hợp các menu cơ bản của một Activity.
- Giúp thực hiện nhanh một số thao tác: tìm kiếm, chia sẻ, cài đặt hệ thống, ...
- Override phương thức *onCreateOptionsMenu()* để thêm vào Activity.

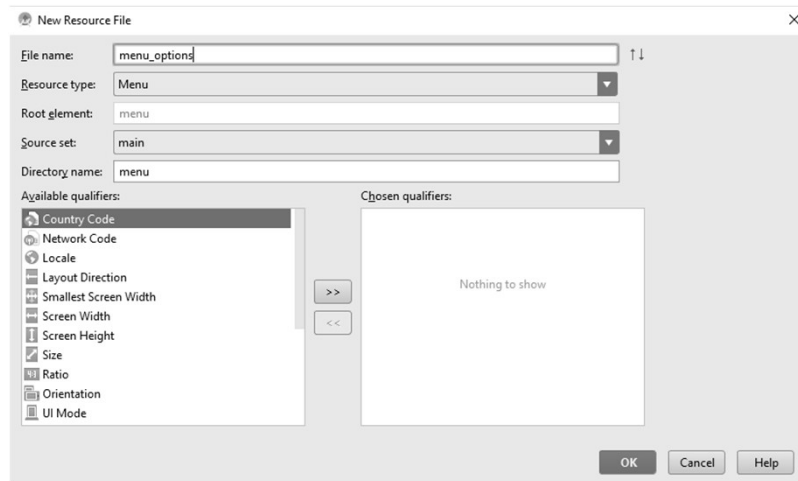


OPTIONS MENU

- **Bước 1.** Thêm một menu mới: new\Android resource file\chọn menu và đặt tên.
- **Bước 2.** Thiết kế giao diện menu trong res\menu.
- **Bước 3.** Override phương thức *onCreateOptionsMenu()*
- **Bước 4.** Xử lý sự kiện khi chọn một menu *onOptionsItemSelected()*

OPTIONS MENU

- **Bước 1.** Thêm một menu mới: new\Android resource file\chọn menu và đặt tên.



Nguyễn Chí Hiếu

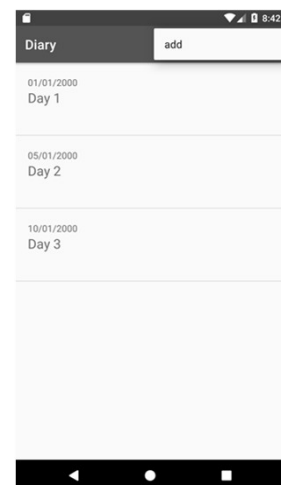
2017 35

35

OPTIONS MENU

- **Bước 2.** Thiết kế giao diện menu.

```
<menu
xmlns:android="http://schemas.android.com
/apk/res/android">
    <item android:id="@+id/menuAdd"
        android:title="add"/>
</menu>
```



Nguyễn Chí Hiếu

2017 36

36

OPTIONS MENU

• Bước 3. Override phương thức *onCreateOptionsMenu()*

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_options, menu);

    return super.onCreateOptionsMenu(menu);
}
```

Nguyễn Chí Hiếu

2017 37

37

OPTIONS MENU

• Bước 4. Xử lý sự kiện khi chọn một menu *onOptionsItemSelected()*

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menuAdd:
            Toast.makeText(getApplicationContext(),
                "add",
                Toast.LENGTH_SHORT).show();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

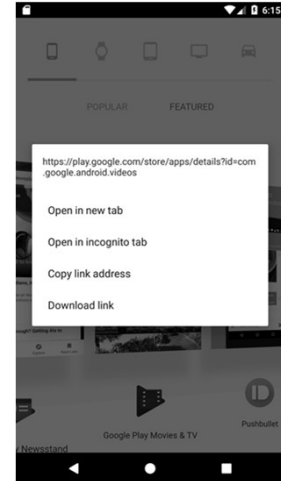
Nguyễn Chí Hiếu

2017 38

38

CONTEXT MENU

- Menu ngữ cảnh tương ứng với một view nào đó trong Activity.
- Trong *onCreate()*, phải đăng ký context menu cho một view xác định:
`registerForContextMenu(_)`
- Các thao tác thêm, xử lý sự kiện tương tự Options Menu.



Nguyễn Chí Hiếu

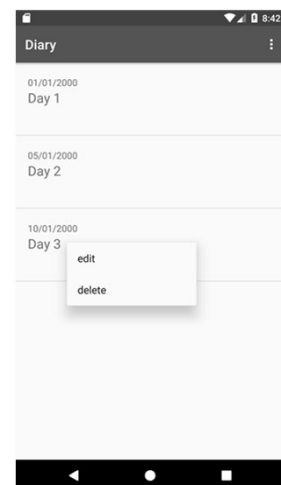
2017 39

39

CONTEXT MENU

- **Bước 1, 2.** Tương tự Context Menu.

```
<menu xmlns:android="http://schemas.android
        .com/apk/res/android">
    <item android:id="@+id/menuEdit"
          android:title="edit" />
    <item android:id="@+id/menuDelete"
          android:title="delete" />
</menu>
```
- **Bước 3.** Trong *onCreate()*, đăng ký Context Menu cho ListView.
`registerForContextMenu(listEntry);`



Nguyễn Chí Hiếu

2017 40

40

CONTEXT MENU

• Bước 4. Override phương thức *onCreateContextMenu()*

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
    ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);

    if (v.getId() == R.id.listEntry) {
        getMenuInflater().inflate(R.menu.menu_context, menu);
    }
}
```

Nguyễn Chí Hiếu

2017 41

41

CONTEXT MENU

• Bước 5. Xử lý sự kiện *onContextItemSelected()*

```
@Override
public boolean onContextItemSelected(MenuItem item) {
    AdapterView.AdapterContextMenuInfo info =
        (AdapterView.AdapterContextMenuInfo) item getMenuInfo();
    EntryModel entry = (EntryModel) listEntry
        .getItemAtPosition(info.position);
    switch (item.getItemId()) {
        case R.id.menuEdit:
            //... Toast.makeText()
            return true;
        // ...
        default:
            return super.onContextItemSelected(item);
    }
}
```

Nguyễn Chí Hiếu

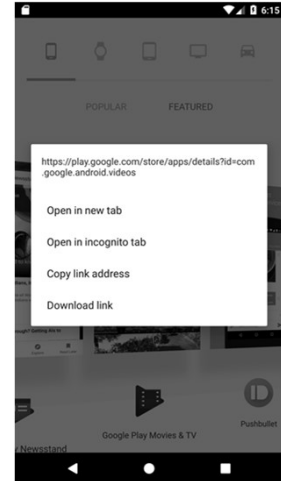
2017 42

42

POPUP MENU

- Có thể tạo và hiển thị Popup Menu cho bất kỳ sự kiện nào của một view.
- Hiển thị bên dưới một view.
- Popup Menu thuộc thư viện widget.

```
import android.widget.PopupMenu;
```



Nguyễn Chí Hiếu

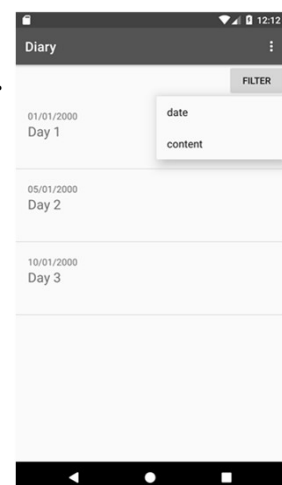
2017 43

43

POPUP MENU

- **Bước 1, 2.** Tương tự Options Menu.

```
<menu xmlns:android="http://schemas.android
        .com/apk/res/android">
    <item android:id="@+id/menuDateFilter"
          android:title="date"/>
    <item android:id="@+id/menuContentFilter"
          android:title="content"/>
</menu>
```



Nguyễn Chí Hiếu

2017 44

44

POPUP MENU

- **Bước 3.** Trong *onCreate()*, đăng ký Popup Menu cho Button.

```
buttonFilter.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        PopupMenu popupMenu = new PopupMenu(
            getApplicationContext(),
            view);
        popupMenu.getMenuInflater()
            .inflate(R.menu.menu_popup, popupMenu.getMenu());

        popupMenu.show();
    }
});
```

Nguyễn Chí Hiếu 2017 45

45

POPUP MENU

- **Bước 4.** Xử lý sự kiện *onMenuItemClick()*

```
@Override
public boolean onMenuItemClick(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menuDateFilter:
            Toast.makeText(getApplicationContext(),
                "date",
                Toast.LENGTH_SHORT).show();
            return true;
        // ...
    }
    return false;
}
```

Nguyễn Chí Hiếu 2017 46

46

BÀI TẬP

- Thiết kế giao diện và cài đặt các ứng dụng sau:
 - Hiển thị danh sách các mục trong nhật ký tương tự ví dụ trên lớp.
 - Hiển thị danh sách sinh viên chứa: ảnh đại diện, MSSV, họ tên, ngày sinh.