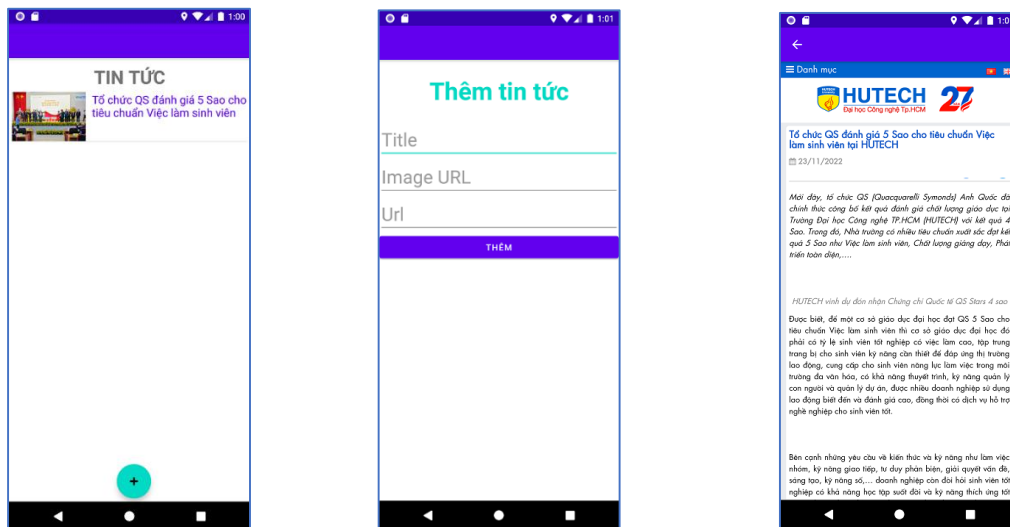


Bài 4— Hướng dẫn thực hiện BottomNavigation + ViewPager2 + Fragments. GVHD: Nguyễn Huy Cường



Bước 0: (cái này không gọi là bước nữa, vì tất cả đều biết)

- ✓ Ứng dụng có kết nối internet nên xin phép quyền truy cập

```
<uses-permission android:name="android.permission.INTERNET" />
```

- ✓ Cần sử dụng thư viện hỗ trợ hình ảnh **Glide** (ở **grade app**) thêm vào đoạn code để import thư viện trong dependencies

```
implementation 'com.github.bumptech.glide:glide:4.12.0'  
annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'
```

- ✓ Phải sync thì mới download và implement được thư viện này

Bước 1: Thiết kế giao diện cho MainActivity, Add Activity, DetailActivity

- ✓ Thiết kế giao diện **MainActivity** để xem tin tức và thêm mới được tin tức gồm có:

TextView, **RecyclerView**, và **ActionButtonFloat** (ở đây không có dám chụp xml layout vì mình thiết kế đại đại bằng constraint cho nhanh 😊)



- ✓ Thêm giao diện **Add** Activity để thêm mới tin tức có 3 EditText: Title, ImageURL và Url
- ✓ Thêm **Detail** Activity để đọc chi tiết 1 tin có: **WebView**

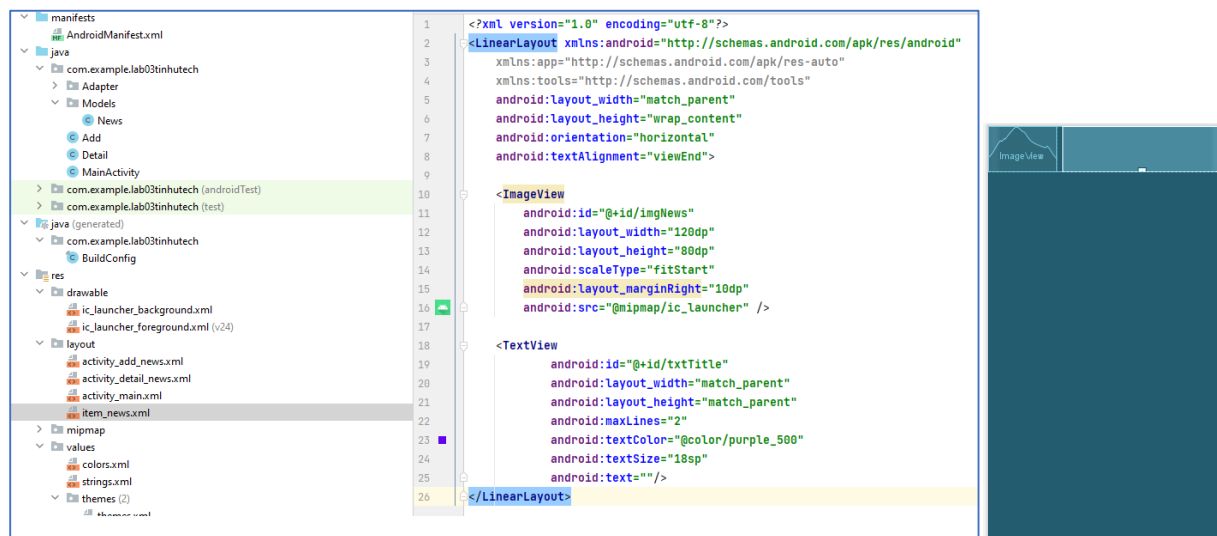
Bước 2: Để hiển thị tin tức trên **recycleView** (bài đã rất cũ, HD sơ sơ lại)

2.1 Tạo DataModel cho 1 tin tức, gọi là lớp **News**

```
public class News {  
    public String Title;  
    public String ImageUrl;  
    public String Url;  
  
    public News(String title, String Imageurl, String url) {  
        Title = title;  
        ImageUrl = Imageurl;  
        Url = url;  
    }  
}
```

2.2 Thiết kế custom item layout cho 1 tin tức (gồm hình và chữ --simple)

Bài 4– Hướng dẫn thực hiện BottomNavigation + ViewPager2 + Fragments. GVHD: Nguyễn Huy Cường



2.3*** Tạo Adapter [Quan trọng nhất, còn mấy thứ trước ghi bước cho sang thôi]

- ✓ Tạo 1 class “NewsAdapter” kế thừa từ Adapter trong thư viện RecyclerView **RecyclerView.Adapter**, Với lớp ViewHolder nằm trong NewsAdapter

```
1 related problem
public class NewsAdapter extends RecyclerView.Adapter<NewsAdapter.ViewHolder>
{
}
}
```

- ✓ Implements các hàm

```
1 related problem
public class NewsAdapter extends RecyclerView.Adapter<NewsAdapter.ViewHolder>
{
}
}
```

Implement methods
Make 'NewsAdapter' abstract
Create Test
Create subclass

- ✓ Implements ViewHolder

```
1 related problem
public class NewsAdapter extends RecyclerView.Adapter<NewsAdapter.ViewHolder>
{
    @NonNull
```

Create class 'ViewHolder'
Create interface 'ViewHolder'
Create enum 'ViewHolder'

- ✓ Thêm danh sách các tin tức vào adapter

```
public class NewsAdapter extends RecyclerView.Adapter<NewsAdapter.ViewHolder>
{
    List<News> listNews;
    public NewsAdapter(List<News> _listNews) {
        this.listNews = _listNews;
    }
}
```

✓ Ảnh xạ view cho ViewHolder tương ứng

```
public class ViewHolder extends RecyclerView.ViewHolder {
    ImageView imageView;
    TextView txtTitle;
    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        imageView = itemView.findViewById(R.id.imgNews );
        txtTitle = itemView.findViewById(R.id.txtTitle );
    }
}
```

✓ Hoàn chỉnh

Đây là lúc chúng ta cần dùng đến Glide, để load ảnh từ internet và hiển thị chúng lên ImageView, chúng ta sẽ thực hiện trong phương thức onBindViewHolder() trong RecyclerView

```
@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View newsView = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_news, parent, attachToRoot);
    return new ViewHolder(newsView);
}

@Override
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    News objNew = listNews.get(position);
    Context mContext = holder.itemView.getContext();
    holder.txtTitle.setText(objNew.Title);
    Glide.with(mContext).load(objNew.ImageUrl).placeholder(R.drawable.ic_launcher_background).into(holder.imageView);
}

@Override
public int getItemCount() {
    return listNews.size();
}
```

Giải thích: Sử dụng thư viện Glide để lấy hình ảnh trên internet và đưa vào imageView trên holder

2.4 Sử dụng ở MainActivity

```
public class MainActivity extends AppCompatActivity {
    List<News> listNews;
    NewsAdapter adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        listNews = new ArrayList<News>();
        listNews.add(new News( title: "Tổ chức QS đánh giá 5 Sao cho tiêu chuẩn Việc làm sinh viên tại HUTECH",
                                Imageurl: "https://file1.hutech.edu.vn/file/editor/homepage1/261780-trao-qs.JPG",
                                url: "https://www.hutech.edu.vn/homepage/tin-tuc/tin-hutech/14607357-to-chuc-qs-danh-gia-5-sao-cho-tieu-chuan-viec-lam-sinh-vien"

        ));

        RecyclerView rcvNews = findViewById(R.id.recyclerView);
        adapter = new NewsAdapter( listNews);
        RecyclerView.ItemDecoration itemDecoration = new DividerItemDecoration( context: this, DividerItemDecoration.VERTICAL);
        rcvNews.addItemDecoration(itemDecoration);
        rcvNews.setLayoutManager(new LinearLayoutManager( context: this));
        rcvNews.setAdapter(adapter);
    }
}
```

Bước 3: Xử lý FloatingActionButton ở MainActivity để gọi và chờ kết quả từ Add

Để ý: Do cần phải lấy kết quả (cụ thể là tin tức, đưa lên recycleView) được thêm mới từ Add Activity, nên sẽ sử dụng **ActivityResultLauncher**.

```
findViewById(R.id.floatingActionButton).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(MainActivity.this, Add.class);
        //startActivity(intent);
        getNewsResult.launch(intent);
    }
});

ActivityResultLauncher<Intent> getNewsResult =
    registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult result) {
            // lấy dữ liệu từ Receiver phản hồi, ...
        }
    });
```

Bước 4: Xử lý Save ở Add Activity

Bài 4– Hướng dẫn thực hiện BottomNavigation + ViewPager2 + Fragments. GVHD: Nguyễn Huy Cường

```
//Ảnh xạ View & lấy ra intent
EditText txtTitle =(EditText)findViewById(R.id.txtTitle);
EditText txtImageUrl =(EditText)findViewById(R.id.txtImageUrl);
EditText txtUrl =(EditText)findViewById(R.id.txtUrl);
Intent intent = getIntent();

//đưa data vào intent qua bundle
findViewById(R.id.btnAddNews).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Bundle bundle = new Bundle();
        News obj = new News(txtTitle.getText().toString(), txtImageUrl.getText().toString(), txtUrl.getText().toString());
        bundle.putSerializable("KEY_SER_NEWS", obj);
        intent.putExtras(bundle);
        setResult( resultCode: 6666, intent);
        finish();
    }
});
```

Ở đây ông Thầy hình như thích 6666 và sử dụng “KEY_SER_NEWS” theo mục đích cá nhân (để các bạn hiểu rằng cái này do project mình qui định, bạn nào thích số nào cứ đặt bừa cũng được...). Nút Save sẽ lấy thông tin trên Add để đưa vào 1 đối tượng **News** là **obj**. Chúng ta sử dụng bundle để mang dữ liệu qua việc Serializable. Do đó News cũng phải implements Serializable. finish() để trở về.

Bước 5: Lấy dữ liệu ở MainActivity trong **getNewsResult** và insert dữ liệu vào recyclerView ở vị trí đầu tiên (position =0)

```
ActivityResultLauncher<Intent> getNewsResult =
    registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult result) {
            // lấy dữ liệu từ Receiver phản hồi
            if(result.getResultCode() == 6666)
            {
                Bundle bundle = result.getData().getExtras();
                if(bundle!= null)
                {
                    News obj = (News) bundle.getSerializable( key: "KEY_SER_NEWS");
                    listNews.add( 0, obj);
                    adapter.notifyItemInserted( position: 0);
                }
            }
        }
    });
```

Giải thích: - Khi chờ phản hồi lấy được intent của phản hồi là result.getData(). Từ đó lấy được **bundle** và lấy đối tượng obj của News đã được thêm.

- Đưa obj vào danh sách listNews và báo cho adapter của RecyclerView thay đổi cho phần từ thêm mới đầu tiên

Chạy và kiểm tra RecyclerView.

Bước 6: Khi onClick trên 1 item ở RecyclerView, tôi chỉ tiết tin tức => cần viết thêm **setOnClickListener** tại **onBindViewHolder** cho 1 itemView trong ViewHolder

```
37      @Override
38      public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
39          News objNew = listNews.get(position);
40          Context mContext = holder.itemView.getContext();
41          holder.txtTitle.setText(objNew.Title);
42          Glide.with(mContext)
43              .load(objNew.ImageUrl)
44              .placeholder(R.drawable.ic_launcher_background)
45              .into(holder.imageView);
46          holder.itemView.setOnClickListener(new View.OnClickListener() {
47              @Override
48              public void onClick(View v) {
49                  Intent intent = new Intent(v.getContext(), Detail.class);
50                  intent.putExtra("URL_Web", objNew.Url);
51                  v.getContext().startActivity(intent);
52              }
53          });
54      }
```

Giải thích: ở onClick trên mỗi tin ở RecyclerView, sẽ gọi “Detail” Activity và truyền dữ liệu là 1 đường LINK URL (của tin tức đó). Nên sử dụng **putExtra** cho nhanh chóng.

Bước 7:

7.1 Tự tìm cách làm thế nào để có thể hiện ra “back” ở Detail và quay lại ?

7.2 Hãy để ý dòng code số 40 từ bức hình bước 6

```
Context mContext = holder.itemView.getContext();
```

onBindViewHolder thì được gọi liên tục (render các item tùy theo vị trí position). Các bạn giúp làm lại tốt hơn ?

Có nhiều bạn cứ thắc mắc tại sao chỉ đến bước 7 ?

Đó chỉ là tình cờ

... Đừng nên thắc mắc tại sao

-----Chúc các bạn thành công !-----