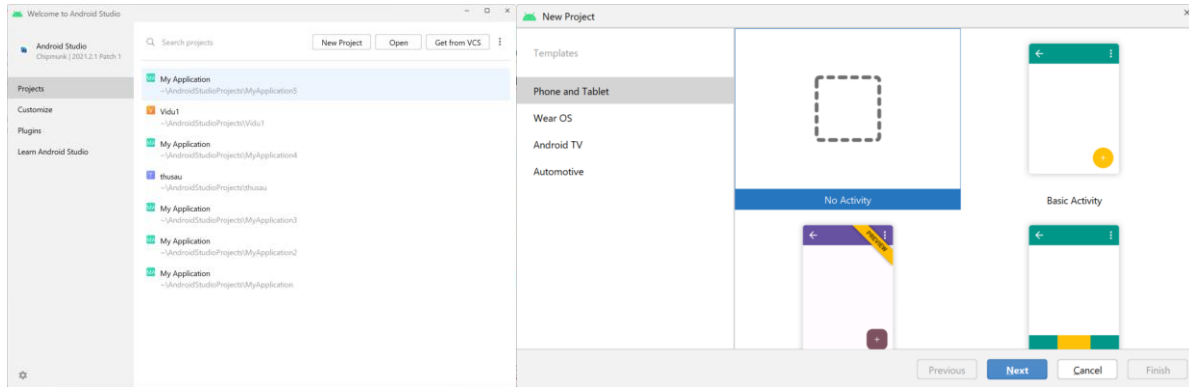


Bài thực hành Android

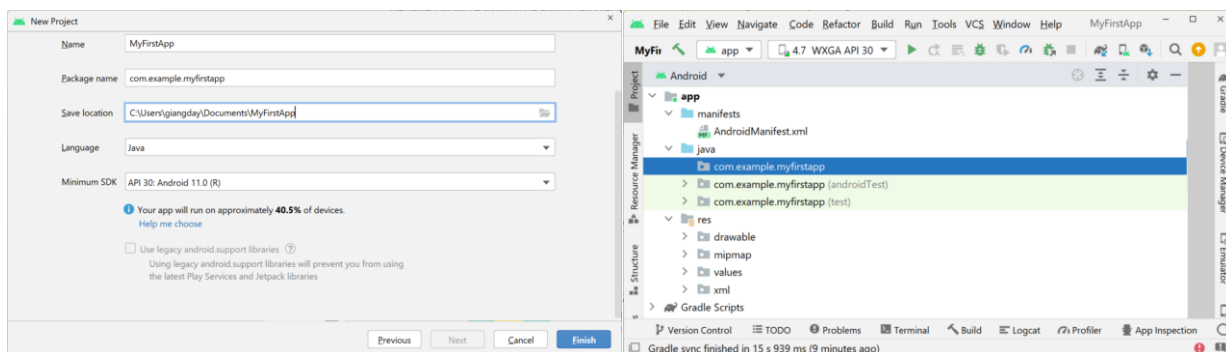
1) Chương 1 : Activities

Bài tập 1 : Tạo ứng dụng đầu tiên

Bắt đầu với giao diện **welcome to Android Studio** , chọn **new project** ta có thể tạo ra một dự án phát triển ứng dụng .



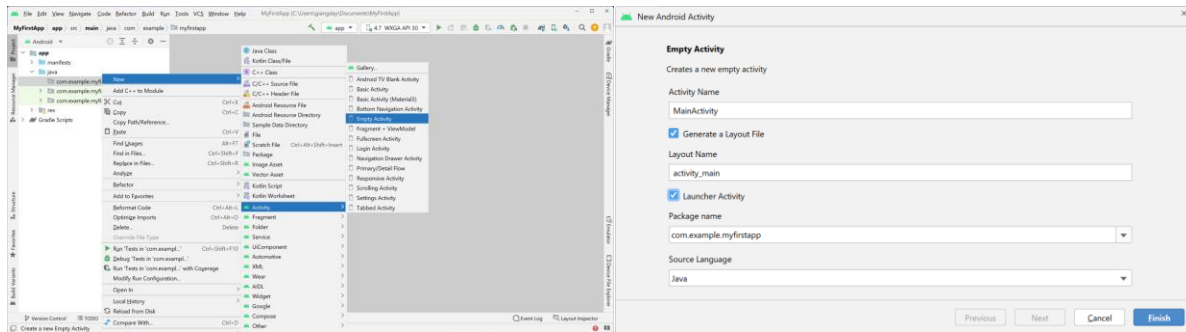
Bước 2 : chọn loại thiết bị : phone, wear, TV, Automotive → chọn một mẫu ứng dụng cho trước. gợi ý ta chọn **phone and tablet** → **No Activity**



Cấu trúc một thư mục application :

- App
 - Manifests
 - Java
 - Res
 - Values
 - layout

Tạo mới một Activity – Empty Activity



Ta chọn ***Generate a Layout File*** để có thể tạo một layout giao diện cho Activity. Lựa chọn ***Launcher Activity*** cho phép Activity mới này trở thành giao diện chính của ứng dụng.

Quá trình tạo ra Activity sẽ tạo mới tập tin layout (activity_main.xml) và phần nội dung mới trong tập tin AndroidManifest.xml để có thể khai báo và cấu hình cho MainActivity được nhận thấy trong ứng dụng.

Giao diện layout : activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

</androidx.constraintlayout.widget.ConstraintLayout>
```

Thông tin của MainActivity trong AndroidManifest.xml

```
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Thiết kế giao diện cho MainActivity bằng 1 trong 3 giao diện :

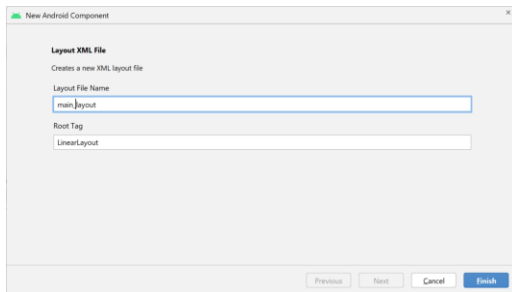
- Design : thiết kế giao diện trực quan bằng kéo thả
- Code : thiết kế giao diện bằng cách thay đổi nội dung tập tin XML
- Split : Kết hợp hai giao diện Design và Code

Kiểu Layout :

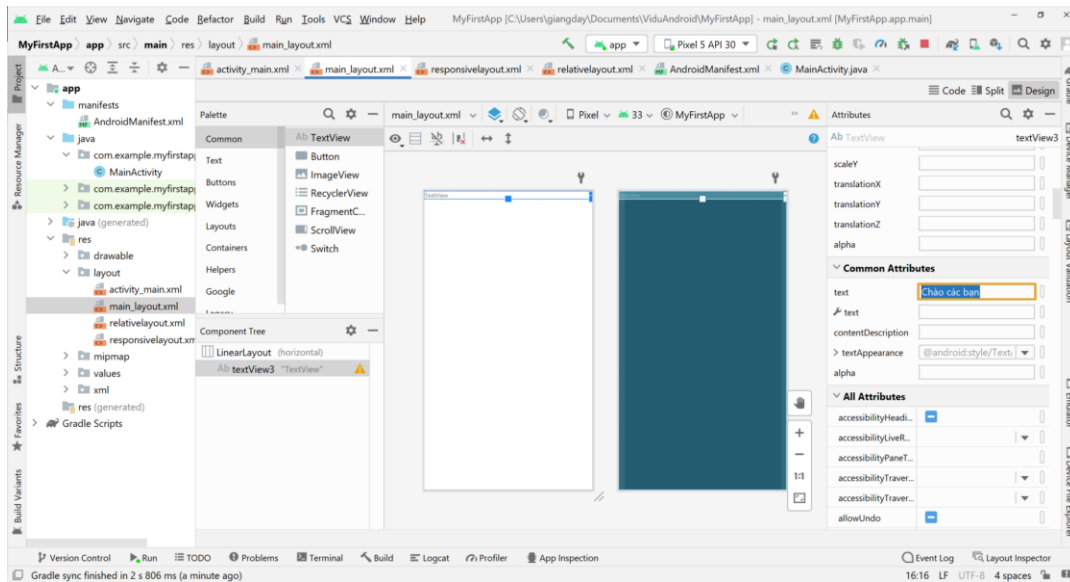
- Linear layout
- Relative layout
- Constraint layout
- TableLayout

Trong bài tập này ta thiết kế giao diện bằng Design. Tạo ra một Linear Layout cho MainActivity theo các bước như sau :

- **Bước 1 :** tạo tập tin activity_main.xml bằng thao tác *file* → *new* → *XML* → *Layout XML file* tên tập tin là main_layout



- **Bước 2 :** sử dụng giao diện design để thêm một *textview* main_layout với nội dung là **Chào các bạn**



- **Bước 3 :** *MainActivity.java* thiết lập liên kết giữa activity và layout bằng câu lệnh : `setContentView(R.layout.main_layout);`;
- **Bước 4 :** Chạy và xem kết quả chương trình trên máy giả lập.
- **Bước 5 :** Sinh viên tìm hiểu về các giá trị thuộc tính (attribute) của textview hiển thị trên Layout.
 - **Text**
 - **Fontfamily**
 - **TextSize**
- Sinh viên quan sát nội dung XML :

```
<?xml version="1.0" encoding="utf-8"?>
<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">

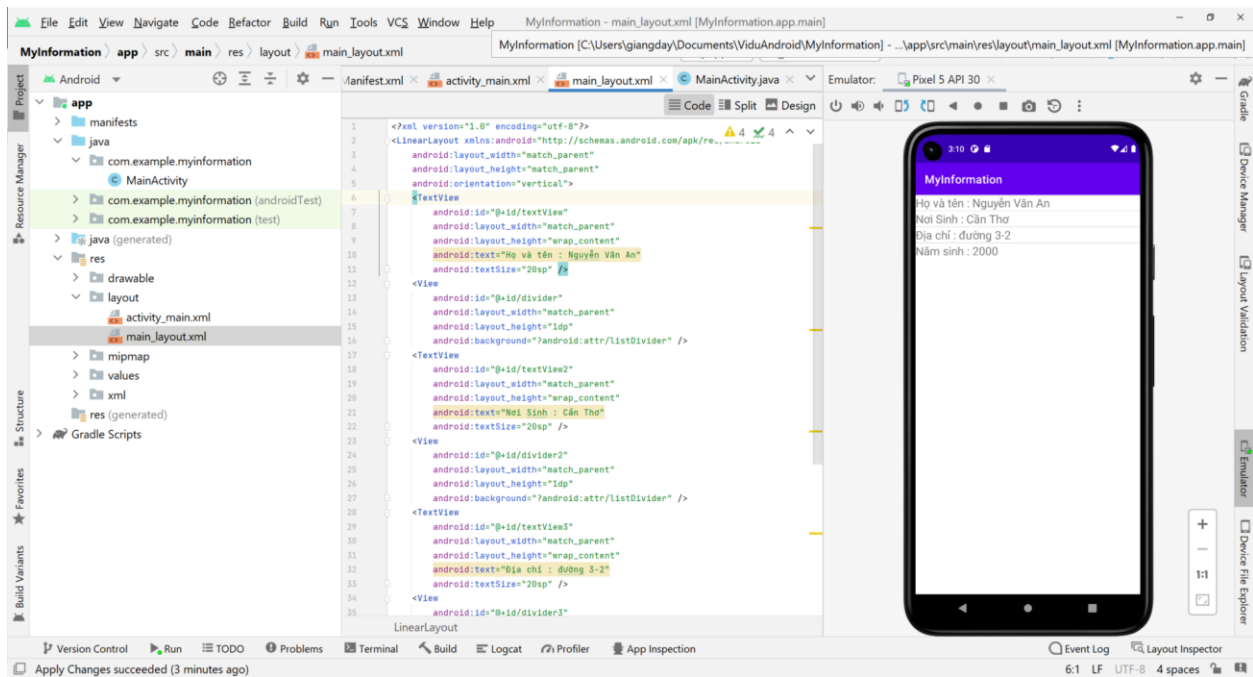
  <TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:fontFamily="sans-serif"
    android:text="Chào các bạn"
    android:textSize="34sp" />
</LinearLayout>
```

Bài tập 2 : Thiết kế ứng dụng hiển thị thông tin cá nhân

Nội dung cần thực hiện là hiển thị thông tin các nhân gồm : mã số sinh viên, họ và tên, nơi sinh, năm sinh

Các bước cần thực hiện :

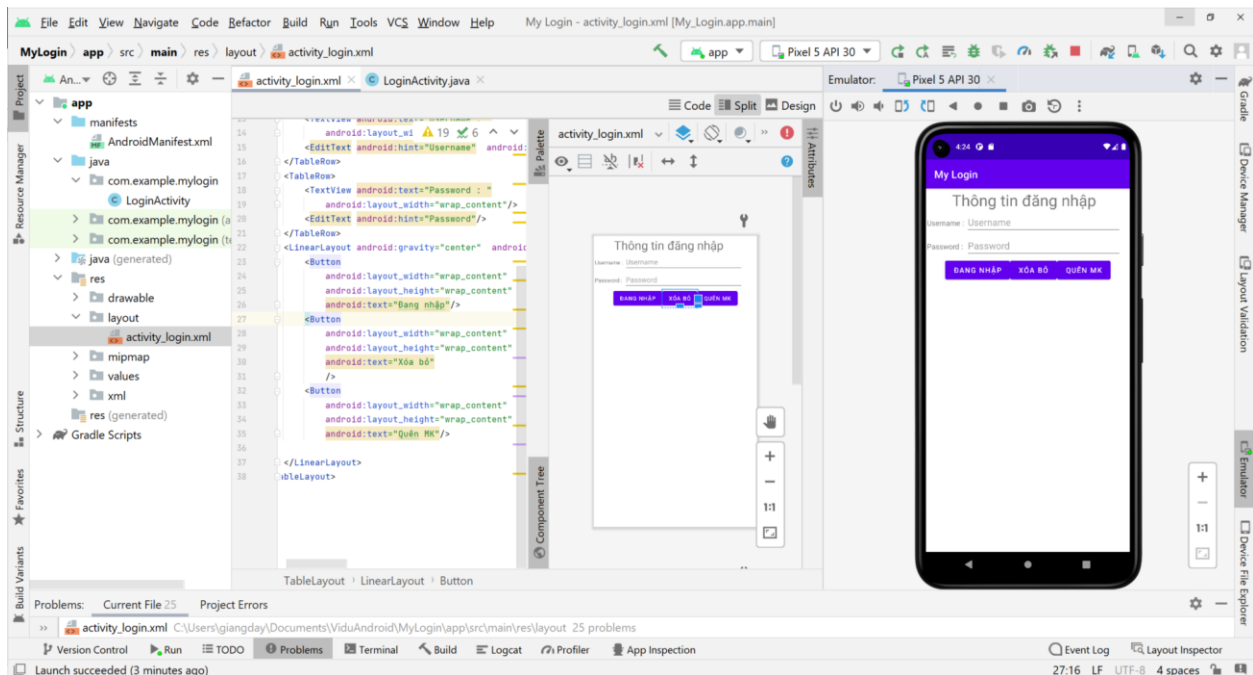
- **Bước 1 :** Tạo project với tên *MyInformation*
- **Bước 2 :** Tạo Main_Activity
- **Bước 3 :** Tạo Lineat layout
 - o Cho phép trình bày các đối tượng nội dung theo chiều dọc bằng thuộc tính **Orientation : Vertical**
 - o Lần lượt thêm các TextView và xác định nội dung và các thuộc tính cần thiết.
- **Bước 4 :** kết nối layout và Main_Activity
- **Bước 5 :** tiến hành triển khai ứng dụng lên máy ảo.



Bài tập 3 : Thiết kế giao diện đăng nhập

Nội dung thực hành : thiết kế một login_activity, sử dụng TableLayout để có thể trình bày các thành phần dễ dàng.

TableLayout cung cấp khả năng trình bày dạng bảng các đối tượng bên trong nó. Để trình bày một dòng trong bảng ta cần thể trình bày **TableRow**. Các thành phần trong một dòng xác định số cột. tổng số cột của bảng được xác định bằng số phần tử lớn nhất của một dòng.



Để có một TableLayout có 03 dòng

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TableRow> </TableRow>
    <TableRow> </TableRow>
    <TableRow> </TableRow>

</TableLayout >
```

Để có thể hiển thị một văn bản trên hai ô liên tiếp của một dòng ta có thuộc tính layout_span

```
<TextView    android:text="Thông tin đăng nhập" android:textSize="30dp" android:layout_span="2"
    android:gravity="center"
/>
```

Trong `TableLayout` ta có thể kết hợp `LinearLayout` để có thể hiển thị một lúc nhiều đối tượng theo một chiều (ngang hoặc dọc)

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:layout_margin="5px">
    <TableRow>
        <TextView
            android:text="Thông tin đăng nhập"
            android:textSize="30dp" android:layout_span="2"
            android:gravity="center"/>
    </TableRow>
    <TableRow>
        <TextView android:text="Username : "
            android:layout_width="wrap_content"/>
        <EditText android:hint="Username" android:layout_width="300dp"/>
    </TableRow>
    <TableRow>
        <TextView android:text="Password : "
            android:layout_width="wrap_content"/>
        <EditText android:hint="Password"/>
    </TableRow>
    <LinearLayout android:gravity="center" android:orientation="horizontal">
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Đăng nhập"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Xóa bỏ"
            />
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Quên MK"/>
    </LinearLayout>
</TableLayout>
```

Câu hỏi bổ sung : tác dụng của các thuộc tính và giá trị của nó:

- `Layout_width / Layout_Height`
- `Layout_magrin`
- `Gravity`
- `Hint`
- `orientation`

Bài tập 4 : Sinh viên thực hiện lại bài 2 bằng cách áp dụng `TableLayout` hoặc `ConstrainLayout`

Bài tập 5 : Sinh viên thiết kế giao diện phục vụ chức năng giải phương trình bậc 2.

Bài tập 6 : Bằng công cụ tìm kiếm qua mạng, Sinh viên tiếp tục tìm hiểu các giao diện `ConstrainLayout`, `RelativeLayout`.