

**TRƯỜNG ĐẠI HỌC THỦY LỢI**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**GIÁO TRÌNH**  
**THỰC HÀNH PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI ĐỘNG**

Hà Nội, 2.2025

**MỤC LỤC**

CHƯƠNG 1. Làm quen.....	4
Bài 1) Tạo ứng dụng đầu tiên.....	4
1.1) Android Studio và Hello World.....	4
1.2) Giao diện người dùng tương tác đầu tiên.....	5
1.3) Trình chỉnh sửa bố cục.....	5
1.4) Văn bản và các chế độ cuộn.....	5
1.5) Tài nguyên có sẵn.....	5
Bài 2) Activities.....	5
2.1) Activity và Intent.....	5
2.2) Vòng đời của Activity và trạng thái.....	5
2.3) Intent ngầm định.....	5
Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ.....	5
3.1) Trình gỡ lỗi.....	5
3.2) Kiểm thử đơn vị.....	5
3.3) Thư viện hỗ trợ.....	5
CHƯƠNG 2. Trải nghiệm người dùng.....	6
Bài 1) Tương tác người dùng.....	6
1.1) Hình ảnh có thể chọn.....	6
1.2) Các điều khiển nhập liệu.....	6
1.3) Menu và bộ chọn.....	6
1.4) Điều hướng người dùng.....	6
1.5) RecyclerView.....	6
Bài 2) Trải nghiệm người dùng thú vị.....	6
2.1) Hình vẽ, định kiểu và chủ đề.....	6
2.2) Thẻ và màu sắc.....	6
2.3) Bố cục thích ứng.....	6

Bài 3)	Kiểm thử giao diện người dùng.....	6
3.1)	Espresso cho việc kiểm tra UI.....	6
CHƯƠNG 3.	Làm việc trong nền.....	6
Bài 1)	Các tác vụ nền.....	6
1.1)	AsyncTask.....	6
1.2)	AsyncTask và AsyncTaskLoader.....	6
1.3)	Broadcast receivers.....	6
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền.....	6
2.1)	Thông báo.....	6
2.2)	Trình quản lý cảnh báo.....	6
2.3)	JobScheduler .....	6
CHƯƠNG 4.	Lưu dữ liệu người dùng.....	7
Bài 1)	Tùy chọn và cài đặt.....	7
1.1)	Shared preferences.....	7
1.2)	Cài đặt ứng dụng.....	7
Bài 2)	Lưu trữ dữ liệu với Room.....	7
2.1)	Room, LiveData và ViewModel.....	7
2.2)	Room, LiveData và ViewModel.....	7

## **CHƯƠNG 1. LÀM QUEN**

### **Bài 1) Tạo ứng dụng đầu tiên**

#### **1.1) Android Studio và Hello World**

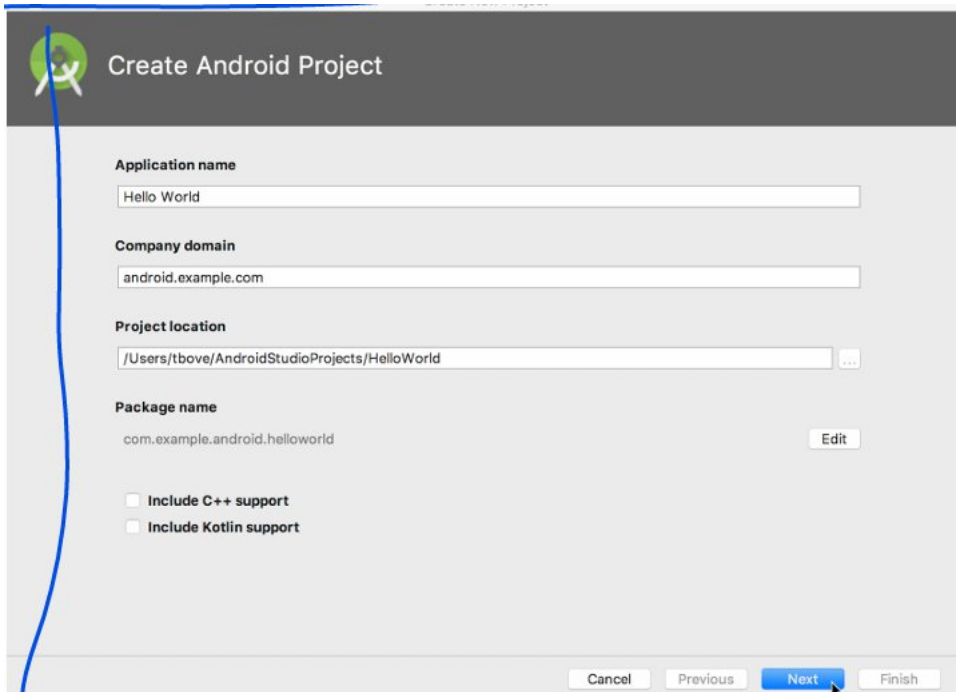
##### **Giới thiệu**

Trong bài thực hành này, bạn sẽ tìm hiểu cách cài đặt Android Studio, môi trường phát triển Android. Bạn cũng sẽ tạo và chạy ứng dụng Android đầu tiên của mình, Hello World, trên một trình giả lập và trên một thiết bị vật lý.

##### **Những gì Bạn nên biết**

Bạn nên có khả năng:

- Hiểu quy trình phát triển phần mềm tổng quát cho các ứng dụng lập trình hướng đối tượng sử dụng một IDE (môi trường phát triển tích hợp) như Android Studio.
- Chứng minh rằng bạn có ít nhất 1-3 năm kinh nghiệm trong lập trình hướng đối tượng, với một phần trong số đó tập trung vào ngôn ngữ lập trình Java. (Các bài thực hành này sẽ không giải thích về lập trình hướng đối tượng hoặc ngôn ngữ Java.



### Những gì Bạn sẽ cần:

- Một máy tính chạy Windows hoặc Linux, hoặc một Mac chạy macOS. Xem trang tải xuống Android Studio để biết yêu cầu hệ thống cập nhật.
- Truy cập Internet hoặc một phương pháp thay thế để tải các cài đặt mới nhất của Android Studio và Java lên máy tính của bạn.

### Những gì bạn sẽ học

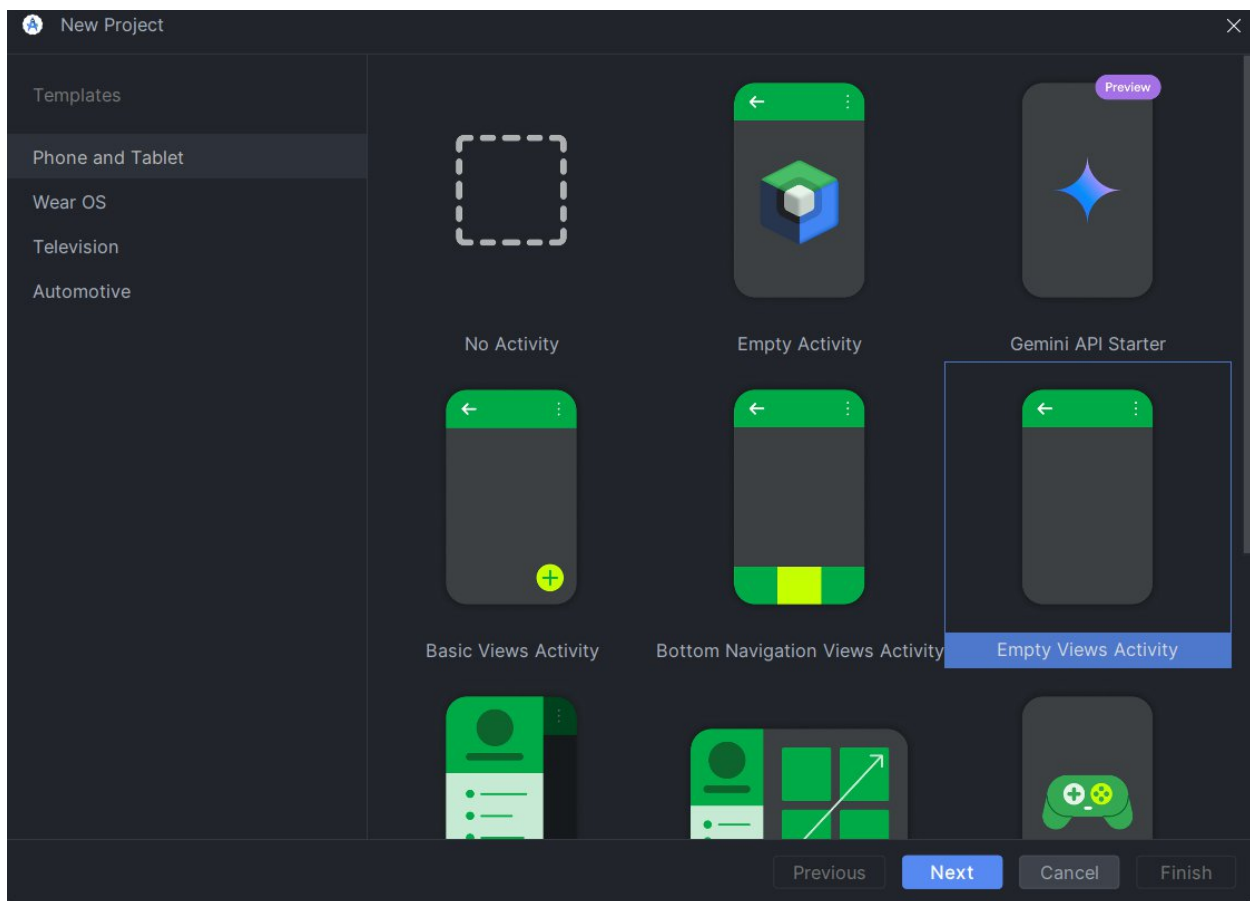
- Cách cài đặt và sử dụng IDE Android Studio.
- Cách sử dụng quy trình phát triển để xây dựng ứng dụng Android.
- Cách tạo một dự án Android từ một mẫu.

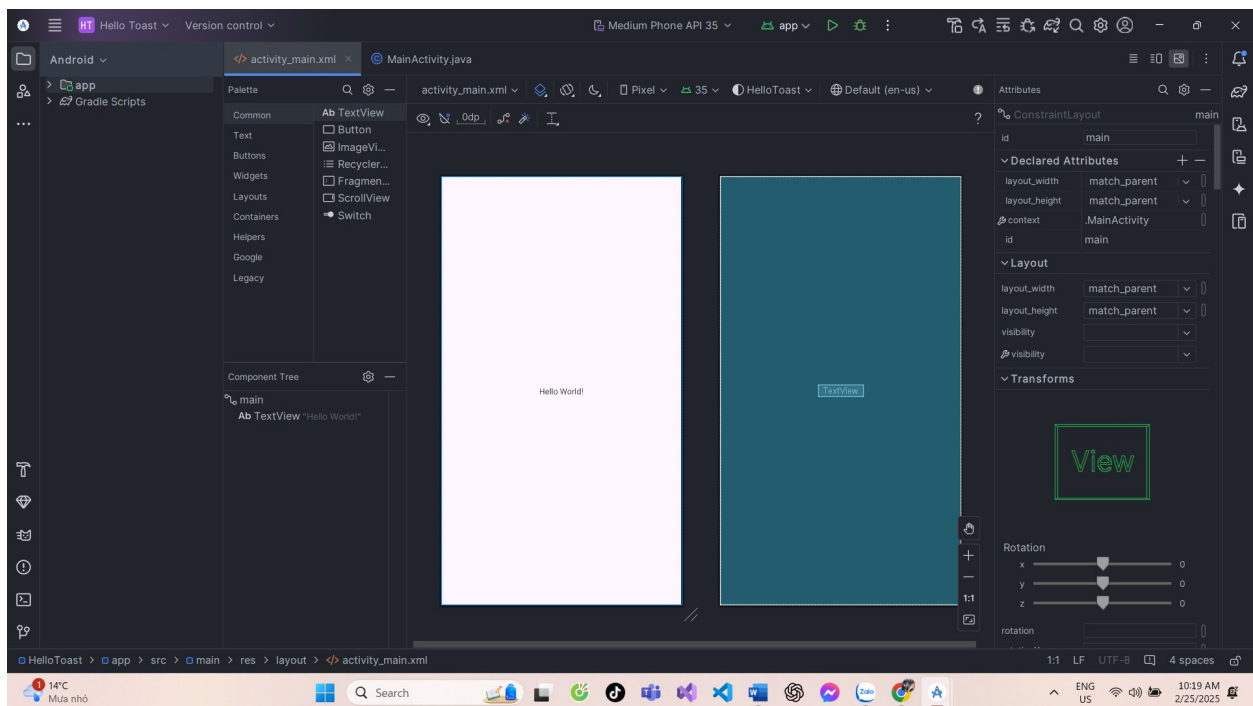
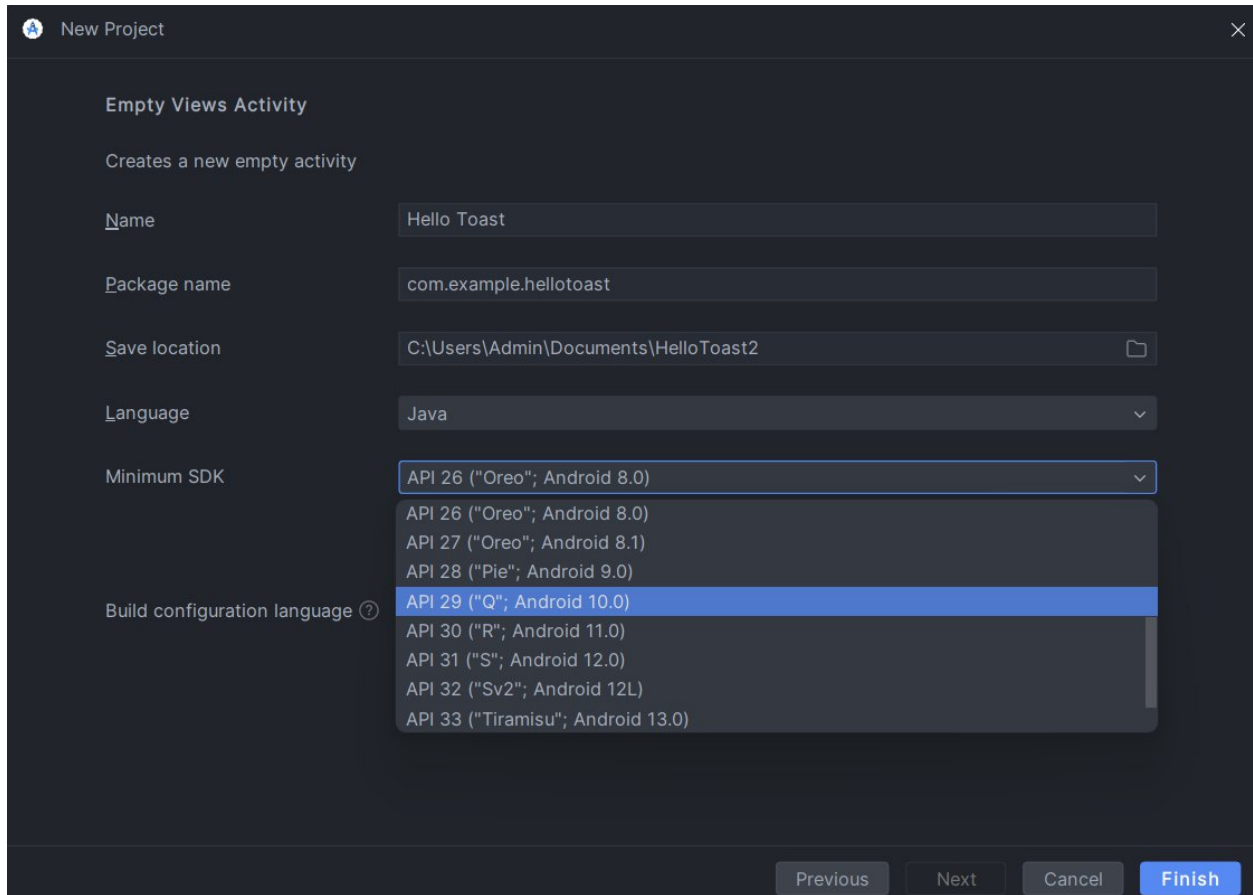
- Cách thêm thông điệp ghi lại vào ứng dụng của bạn để phục vụ mục đích gỡ lỗi.

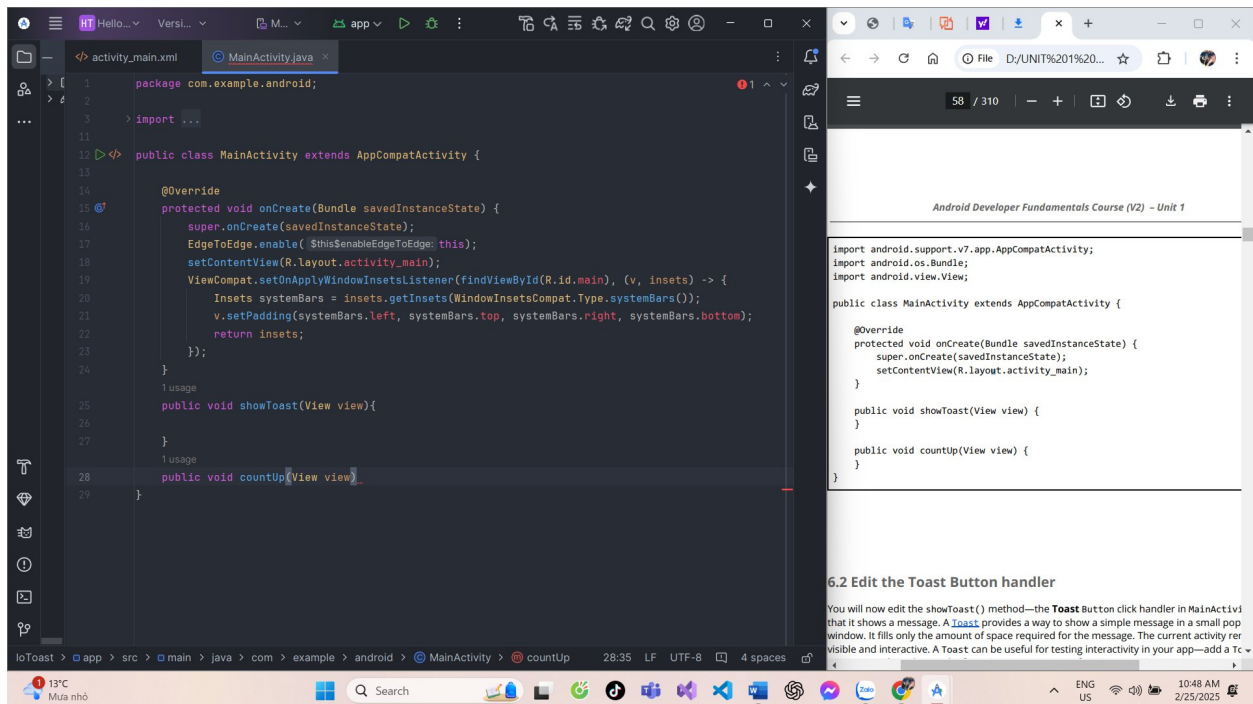
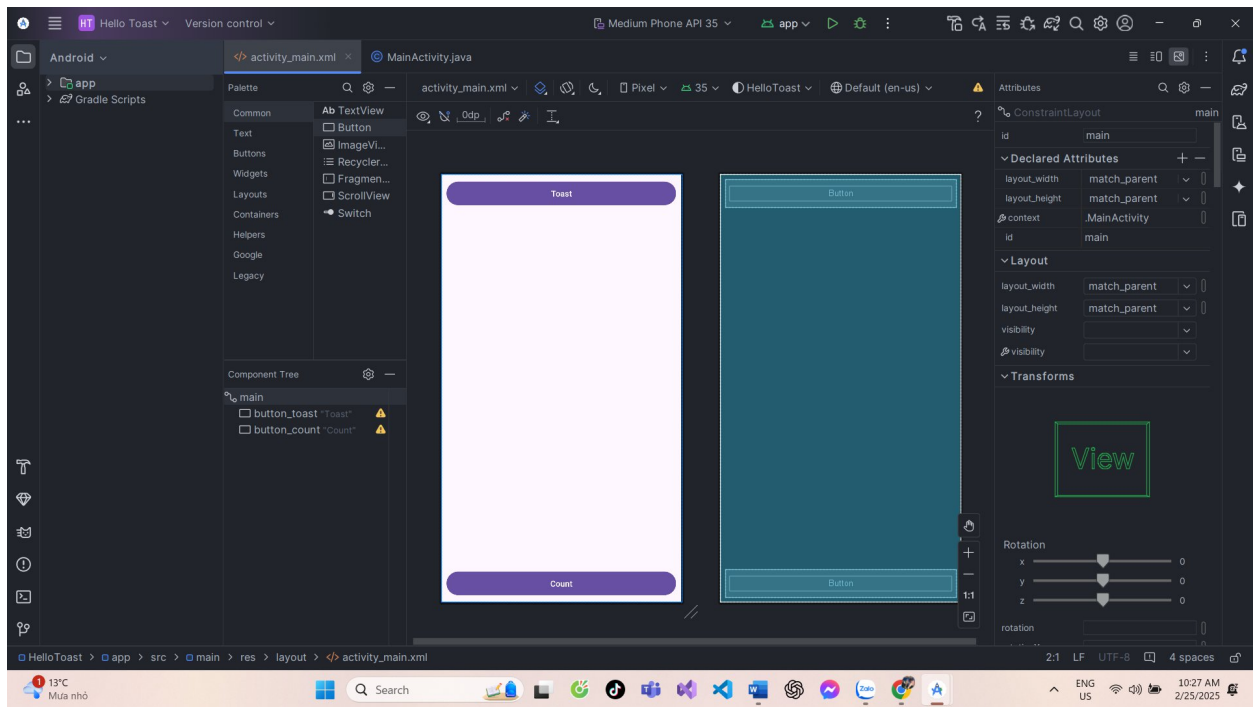
## Những gì bạn sẽ làm

- Cài đặt môi trường phát triển **Android Studio**.
- Tạo một trình giả lập (thiết bị ảo) để chạy ứng dụng của bạn trên máy tính.
- Tạo và chạy ứng dụng **Hello World** trên các thiết bị ảo và vật lý.
- Khám phá cấu trúc dự án.
- Tạo và xem các thông điệp ghi lại từ ứng dụng của bạn.
- Khám phá tệp **AndroidManifest.xml**

### 1.2) Giao diện người dùng tương tác đầu tiên







```

1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/main"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:background="#FFFFFF"
9      tools:context=".MainActivity">
10
11      <Button
12          android:id="@+id/button_toast"
13          android:layout_width="0dp"
14          android:layout_height="wrap_content"
15          android:layout_marginStart="8dp"
16          android:layout_marginTop="8dp"
17          android:layout_marginEnd="8dp"
18          android:text="Toast"
19          android:textColor="#EFEFE9"
20          android:textColorLink="@color/white"
21          app:layout_constraintEnd_toEndOf="parent"
22          app:layout_constraintStart_toStartOf="parent"
23          app:layout_constraintTop_toTopOf="parent"
24          android:onClick="showToast"/>
25
26      <Button
27          android:id="@+id/button_count"
28          android:layout_width="0dp"
29          android:layout_height="wrap_content"
30          android:layout_marginStart="8dp"
31          android:layout_marginEnd="8dp"
32          android:layout_marginBottom="8dp"

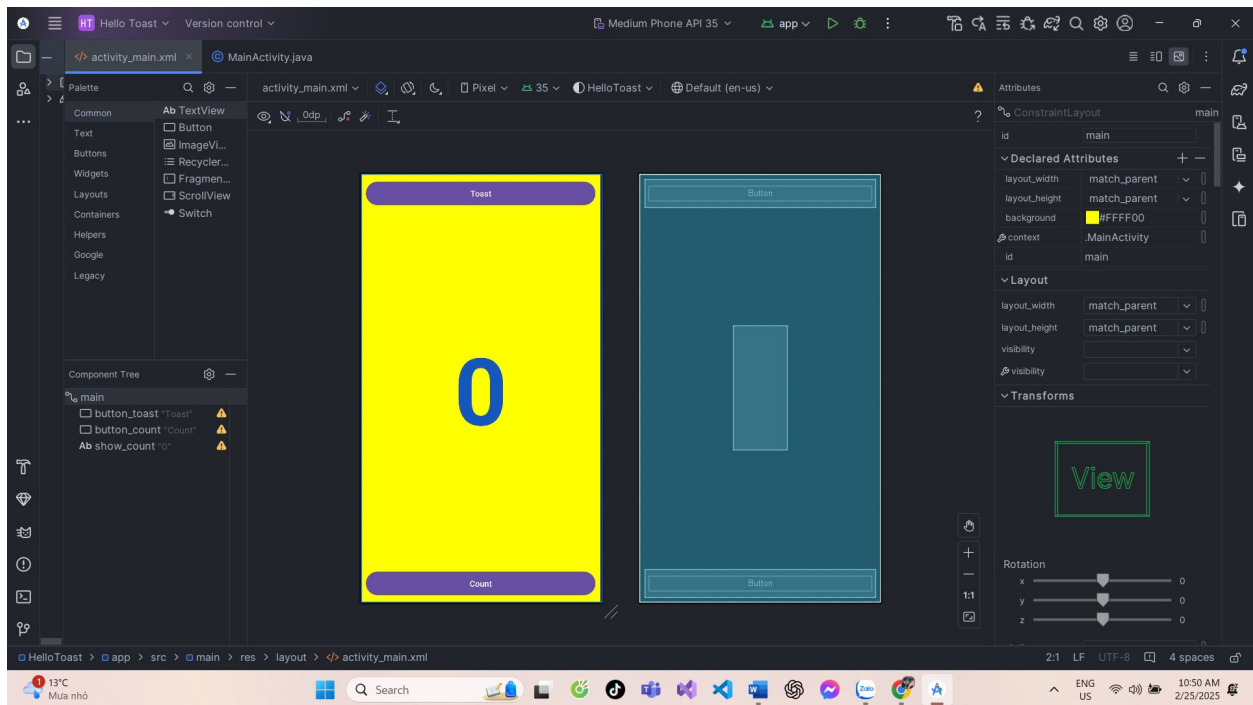
```

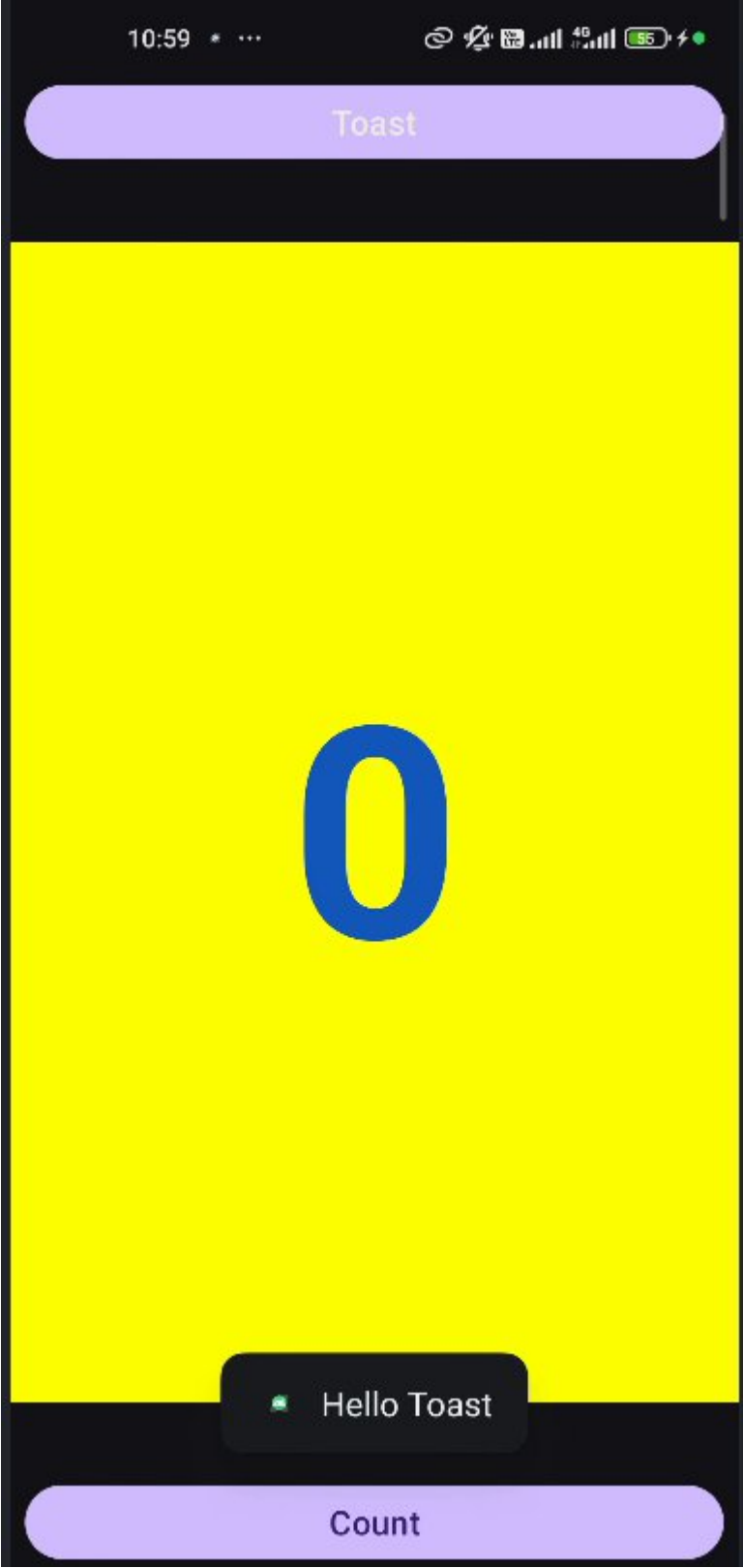
```

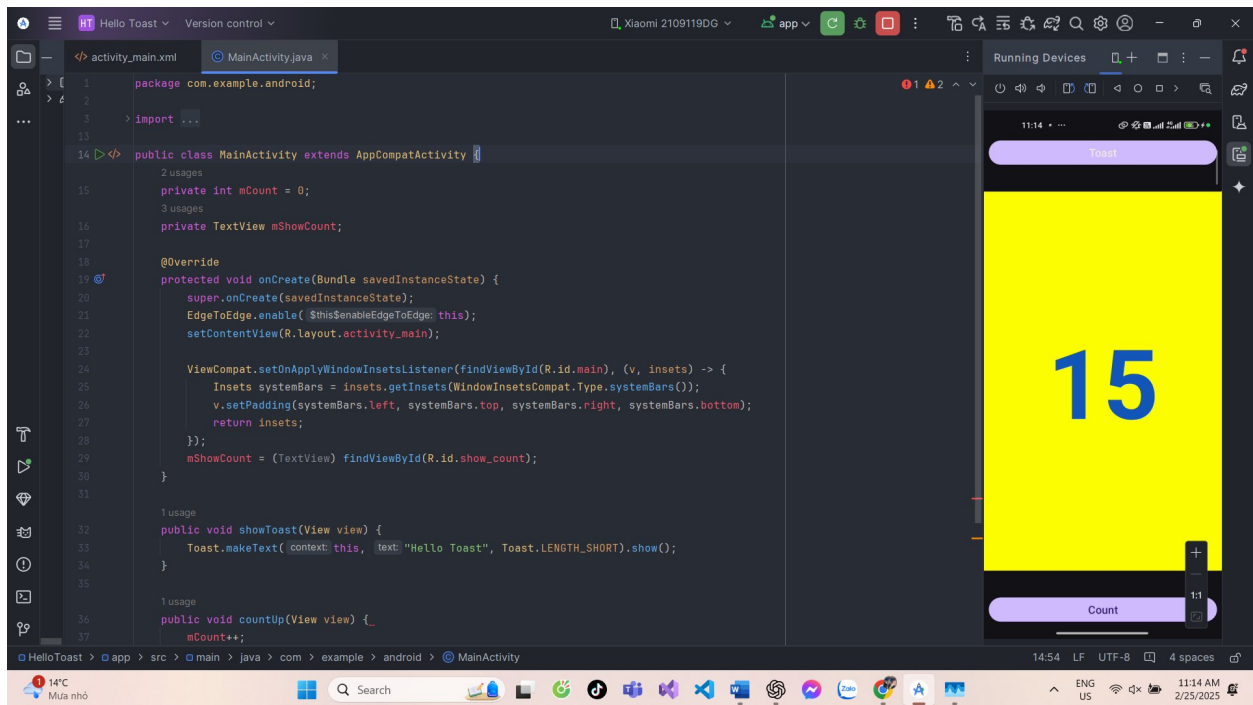
2  <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
26
28          android:layout_width="0dp"
29          android:layout_height="wrap_content"
30          android:layout_marginStart="8dp"
31          android:layout_marginEnd="8dp"
32          android:layout_marginBottom="8dp"
33          android:text="Count"
34          app:layout_constraintBottom_toBottomOf="parent"
35          app:layout_constraintEnd_toEndOf="parent"
36          app:layout_constraintStart_toStartOf="parent"
37          android:onClick="countUp"/>
38
39      <TextView
40          android:id="@+id/show_count"
41          android:layout_width="wrap_content"
42          android:layout_height="wrap_content"
43          android:layout_marginStart="8dp"
44          android:layout_marginTop="8dp"
45          android:layout_marginEnd="8dp"
46          android:layout_marginBottom="8dp"
47          android:background="#FFFFFF"
48          android:gravity="center_vertical"
49          android:text="0"
50          android:textColor="#155780"
51          android:textSize="160sp"
52          android:textStyle="bold"
53          app:layout_constraintBottom_toTopOf="@+id/button_count"
54          app:layout_constraintEnd_toEndOf="parent"
55          app:layout_constraintStart_toStartOf="parent"
56          app:layout_constraintTop_toBottomOf="@+id/button_toast" />
57  </androidx.constraintlayout.widget.ConstraintLayout>

```









### 1.3) Trình chỉnh sửa bố cục

### 1.4) Văn bản và các chế độ cuộn

### 1.5) Tài nguyên có sẵn

## Bài 2) Activities

### 2.1) Activity và Intent

### 2.2) Vòng đời của Activity và trạng thái

### 2.3) Intent ngầm định

## Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ

### 3.1) Trình gỡ lỗi

### 3.2) Kiểm thử đơn vị

### 3.3) Thư viện hỗ trợ

## CHƯƠNG 2. TRẢI NGHIỆM NGƯỜI DÙNG

### Bài 1) Tương tác người dùng

#### 1.1) Hình ảnh có thể chọn

##### Giới thiệu:

Giao diện người dùng (UI) xuất hiện trên màn hình của thiết bị Android bao gồm một hệ thống phân cấp các đối tượng gọi là **View**. Mỗi phần tử trên màn hình đều là một **View**.

Lớp **View** là khối xây dựng cơ bản cho tất cả các thành phần UI. **View** là lớp cơ sở của các lớp cung cấp thành phần giao diện tương tác, chẳng hạn như **Button**. **Button** là một phần tử UI mà người dùng có thể nhấn hoặc bấm để thực hiện một hành động.

Bạn có thể biến bất kỳ **View** nào, chẳng hạn như **ImageView**, thành một phần tử UI có thể nhấn. Để làm điều này, bạn cần lưu trữ hình ảnh cho **ImageView** trong thư mục **drawable** của dự án.

Trong bài thực hành này, bạn sẽ học cách sử dụng hình ảnh như một phần tử tương tác mà người dùng có thể nhấn hoặc bấm vào.

##### Những kiến thức bạn cần biết

Bạn nên có khả năng:

- Tạo một dự án Android Studio từ một mẫu có sẵn và tạo bố cục chính.
- Chạy ứng dụng trên trình giả lập hoặc thiết bị được kết nối.
- Tạo và chỉnh sửa các thành phần UI bằng trình chỉnh sửa bố cục (Layout Editor) và mã XML.
- Truy cập các phần tử UI từ mã bằng **findViewById()**.
- Xử lý sự kiện khi nhấn nút (Button click).
- Hiển thị đoạn tin nhắn thông báo **Toast message**.
- Thêm hình ảnh vào thư mục **drawable** của dự án.

##### Bạn sẽ học được gì:

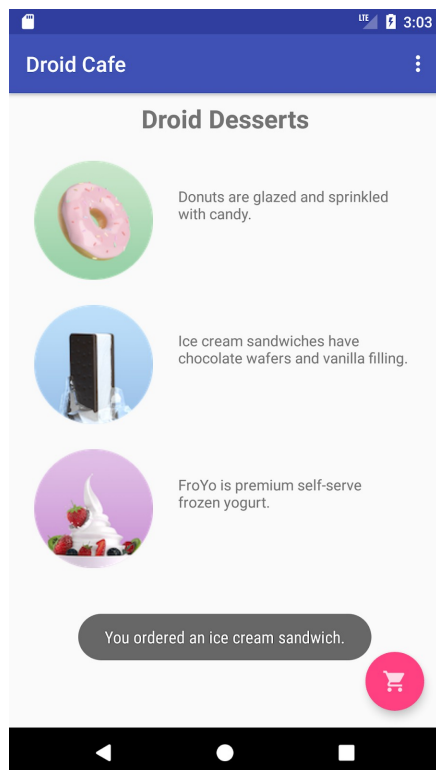
- Cách sử dụng hình ảnh như một phần tử tương tác để thực hiện một hành động.
- Cách đặt thuộc tính cho **ImageView** trong trình chỉnh sửa bố cục.
- Cách thêm phương thức **onClick()** để hiển thị **Toast message**.

## Bạn sẽ làm gì:

- Tạo một dự án Android Studio mới cho một ứng dụng đặt món tráng miệng giả lập mà nó sử dụng hình ảnh như một phần tử tương tác.
- Thiết lập trình xử lý sự kiện **onClick()** cho hình ảnh để hiển thị các **Toast message** khác nhau.
- Thay đổi nút hành động nổi (**Floating Action Button**) do mẫu cung cấp để mà nó hiển thị biểu tượng khác nhau và chạy sang các **Activity** khác.

## Tổng quan về ứng dụng

Trong bài thực hành này, bạn sẽ tạo và xây dựng một ứng dụng mới bằng cách sử dụng mẫu **Basic Activity** để mô phỏng ứng dụng đặt món tráng miệng. Người dùng có thể nhấn vào hình ảnh để thực hiện một hành động - trong trường hợp hiển thị một **Toast message** như hình minh họa bên dưới. Người dùng cũng có thể nhấn vào nút giỏ hàng để chuyển sang **Activity** tiếp theo.



## Nhiệm vụ 1: Thêm hình ảnh vào bố cục

Bạn có thể làm một hình ảnh cho một giống như một nút bằng cách thêm thuộc tính `android:onClick` vào tệp bố cục XML. Ví dụ, bạn có thể làm cho một hình ảnh hoạt động giống như một nút bằng cách thêm `android:onClick` vào **ImageView**.

Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng mẫu để đặt món tráng miệng từ một quán cà phê. Sau khi khởi tạo một dự án mới dựa trên mẫu **Basic Activity**, bạn sẽ chỉnh sửa **TextView** hiển thị **"Hello World"** bằng nội dung phù hợp và thêm các hình ảnh có thể nhấn.

### Khởi tạo dự án mới

1. Mở Android Studio và tạo một dự án mới với tên ứng dụng là **Droid Cafe**.
2. Chọn mẫu **Basic Activity**, giữ nguyên tên mặc định của **Activity** (**MainActivity**). Đảm bảo rằng các tùy chọn **Generate Layout file** và **Backwards Compatibility (AppCompat)** đều đã được chọn.
3. Nhấn **Finish** để hoàn tất tạo dự án.

Khi dự án mở, bạn sẽ thấy hai tệp bố cục trong thư mục **res > layout**:

**activity\_main.xml** chứa thanh ứng dụng (App Bar) và nút hành động nổi (bạn chưa cần thay đổi trong nhiệm vụ này) và **content\_main.xml** chứa tất cả các phần còn lại của bố cục.

4. Mở **content\_main.xml**, chọn tab **Design** (nếu chưa được chọn) để hiển thị trình chỉnh sửa bố cục.
5. Chọn **TextView** ("Hello World") trong bố cục và mở bảng **Attributes**.

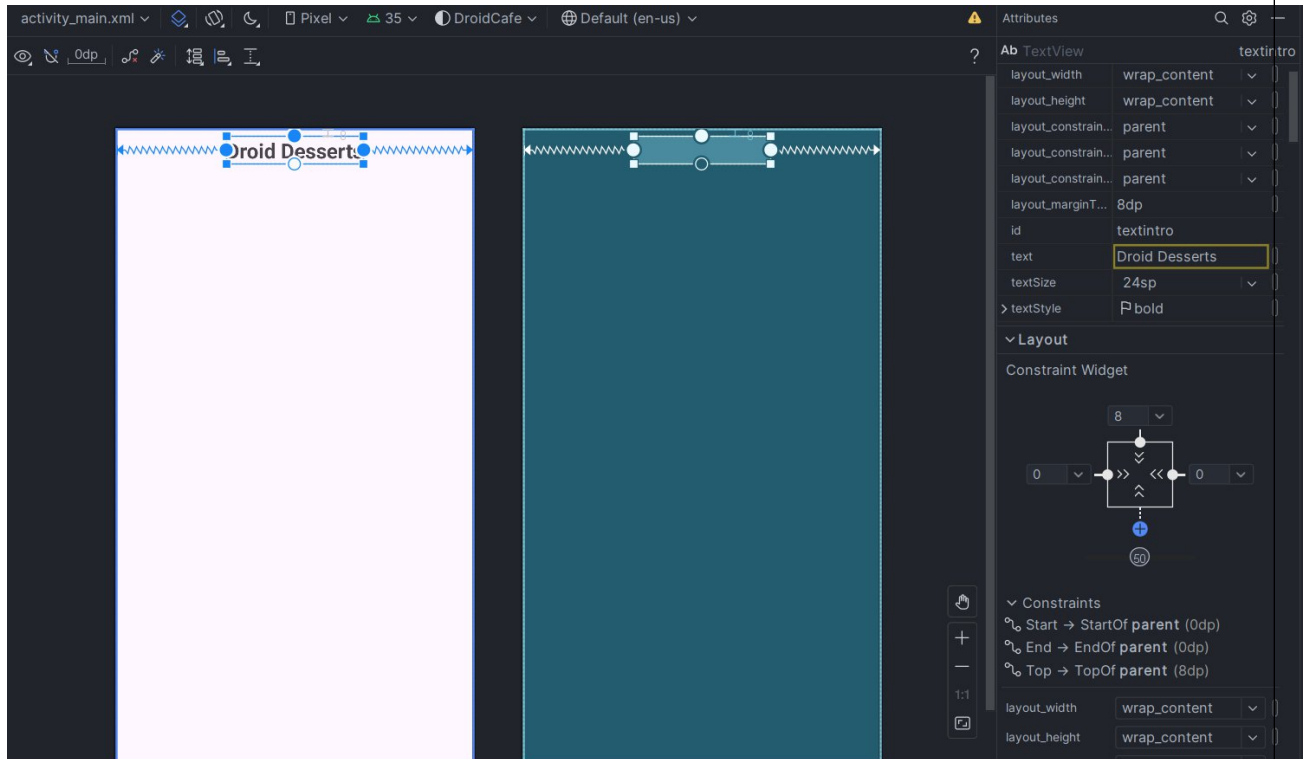
Attribute field	Enter the following:
-----------------	----------------------

6. Thay đổi các thuộc tính như sau:

ID	textintro
Text	Đổi “Hello World” thành “Droid Desserts”
textStyle	B(bold)
textSize	24sp

Các thay đổi trên sẽ:

- Thêm thuộc tính android:id với giá trị textintro.
  - Thay đổi văn bản.
  - Làm đậm văn bản.
  - Thiết lập kích thước chữ là 24sp.
7. Xóa ràng buộc mà nó kéo dài từ dưới cùng của textintro TextView đến cuối của bố cục, từ đó để TextView gắn vào đầu bố cục, và chọn 8 (dp) cho lề trên cùng như hình dưới đây



8. Trong bài học trước, bạn đã học cách trích xuất tài nguyên chuỗi từ chuỗi văn bản theo nghĩa đen. Nhấp vào tab Văn bản để chuyển sang mã XML và trích xuất chuỗi "Droid Desserts" trong TextView và nhập intro\_text làm tên tài nguyên chuỗi

## 1.2) Các điều khiển nhập liệu

Ba hình ảnh (donut\_circle.png, froyo\_circle.png và icecream\_circle.png) được cung cấp cho ví dụ này, bạn có thể tải xuống. Ngoài ra, bạn có thể thay thế hình ảnh của riêng mình thành tệp PNG, nhưng chúng phải có kích thước khoảng 113 x 113 pixel để sử dụng trong ví dụ này.

Bước này cũng có thể giới thiệu một kỹ thuật mới trong quá trình chỉnh sửa bố cục: sử dụng nút Sửa (Fix) trong thông báo cảnh báo để trích xuất tài nguyên chuỗi.

1. Để sao chép hình ảnh vào dự án của bạn, trước tiên hãy đóng dự án.



2. Sao chép các tệp hình ảnh vào thư mục drawable của dự án. Tìm thư mục drawable trong dự án bằng cách sử dụng đường dẫn này: `project_name > app > src > main > res > drawable`.
3. Mở lại dự án của bạn.
4. Mở tệp `content_main.xml` và nhấp vào tab Thiết kế (nếu chưa chọn).
5. Kéo `ImageView` vào bố cục, chọn hình ảnh `donut_circle` cho nó và giới hạn nó ở `TextView` trên cùng và ở phía bên trái của bố cục với lề là 24 (24dp) cho cả hai giới hạn, như được hiển thị trong hình động bên dưới.

**1.3) Menu và bộ chọn**

**1.4) Điều hướng người dùng**

**1.5) RecyclerView**

**Bài 2) Trải nghiệm người dùng thú vị**

**2.1) Hình vẽ, định kiểu và chủ đề**

**2.2) Thẻ và màu sắc**

**2.3) Bố cục thích ứng**

**Bài 3) Kiểm thử giao diện người dùng**

**3.1) Espresso cho việc kiểm tra UI**

### **CHƯƠNG 3. LÀM VIỆC TRONG NỀN**

**Bài 1) Các tác vụ nền**

**1.1) AsyncTask**

**1.2) AsyncTask và AsyncTaskLoader**

**1.3) Broadcast receivers**

**Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền**

**2.1) Thông báo**

**2.2) Trình quản lý cảnh báo**

**2.3) JobScheduler**

### **CHƯƠNG 4. LƯU DỮ LIỆU NGƯỜI DÙNG**

**Bài 1) Tùy chọn và cài đặt**

**1.1) Shared preferences**

**1.2) Cài đặt ứng dụng**

**Bài 2) Lưu trữ dữ liệu với Room**

**2.1) Room, LiveData và ViewModel**

**2.2) Room, LiveData và ViewModel**