

Họ tên: Lê Phúc Hưng

MSSV: 20215276

Mã lớp: 151902

Môn học: Phát triển ứng dụng cho thiết bị di động (IT4785)

Source Code: <https://github.com/lephuchung/HustMobile>

Bài: Lesson 8 – Activities and intents

## Contents

1. Introduction .....	2
2. Starter code .....	2
3. Words app overview .....	2
4. Introduction to Intents .....	3
4.1. Explicit Intent (Intent tường minh) .....	3
4.2. Implicit Intent (Intent ngầm định) .....	3
5. Set up Explicit Intent .....	4
6. Set up DetailActivity .....	5
6.1. Clean up .....	5
7. Set up Implicit Intent .....	6
8. Set up Menu and Icons .....	7
9. Implement Menu button .....	8
10. Solution code .....	9
11. Summary .....	9
12. Learn more .....	9

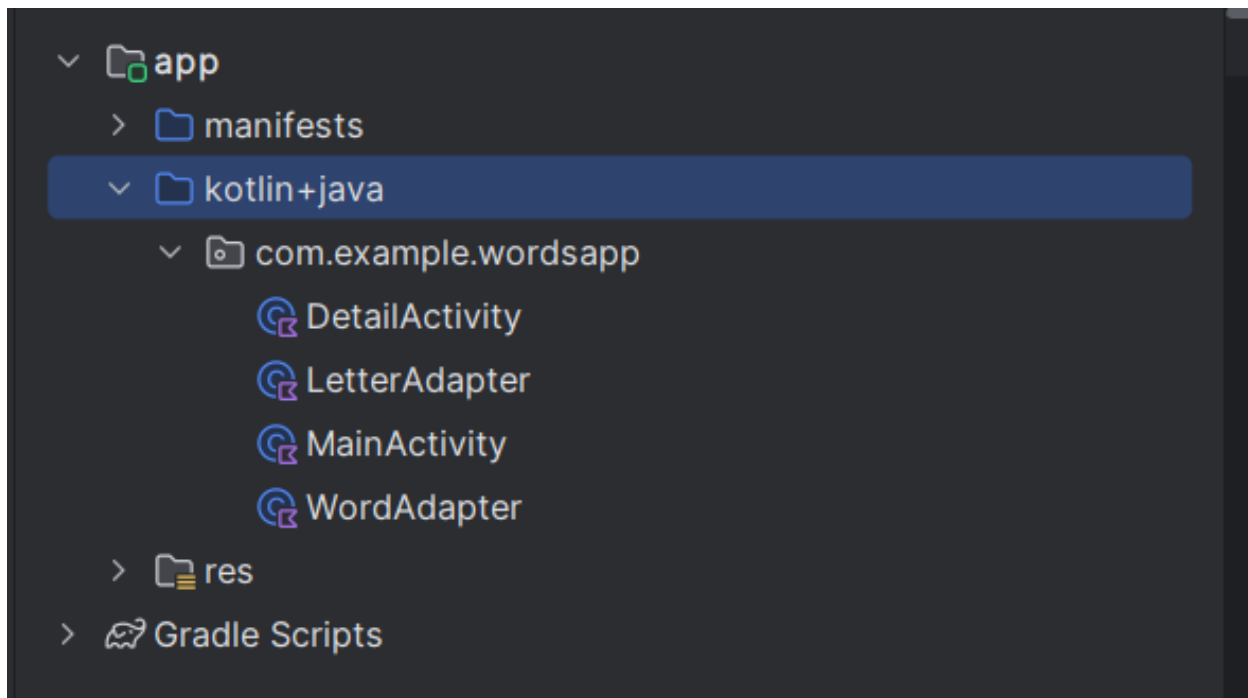
## 1. Introduction

## 2. Starter code

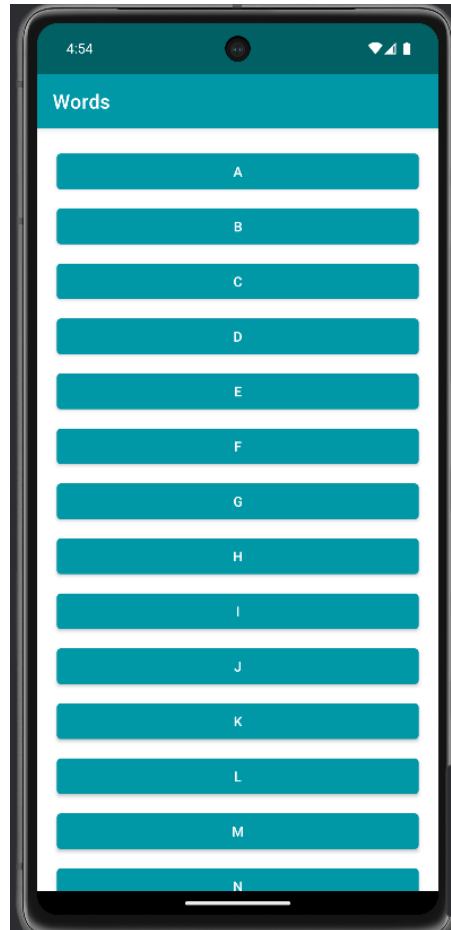
Tải code về thông qua link github: <https://github.com/google-developer-training/android-basics-kotlin-words-app/tree/starter>

## 3. Words app overview

Hiện tại, ứng dụng này bao gồm hai hoạt động (activity), mỗi hoạt động có một khung hiển thị tái chế (recycler view) và một trình chuyển đổi (adapter).



Chương trình trước khi sửa đổi:



## 4. Introduction to Intents

Ta sẽ tìm hiểu một chút về Intents:

Trong ứng dụng Android, Intent là một đối tượng đại diện cho một hành động cần thực hiện. Có hai loại intent chính là implicit (ngầm định) và explicit (tường minh).

### 4.1. *Explicit Intent (Intent tường minh)*

Intent tường minh là loại intent dùng để khởi chạy một activity cụ thể trong ứng dụng của bạn khi bạn đã biết chính xác tên của activity đó. Loại intent này thường được dùng để mở một màn hình khác trong cùng ứng dụng. Ví dụ, nếu bạn muốn từ màn hình chính chuyển đến màn hình chi tiết, bạn sẽ sử dụng explicit intent để chỉ rõ activity cần mở.

### 4.2. *Implicit Intent (Intent ngầm định)*

Intent ngầm định thì trừu tượng hơn, vì bạn không chỉ định activity cụ thể nào, mà chỉ mô tả loại hành động bạn muốn thực hiện. Hệ thống sẽ xem xét các ứng dụng có khả năng thực hiện hành động này và đưa ra tùy chọn cho người dùng. Ví dụ, nếu bạn muốn mở một đường link, gửi email, hoặc thực hiện cuộc gọi điện thoại, bạn sẽ dùng implicit intent và hệ thống Android sẽ tự động xác định ứng dụng nào có thể thực hiện yêu cầu đó.

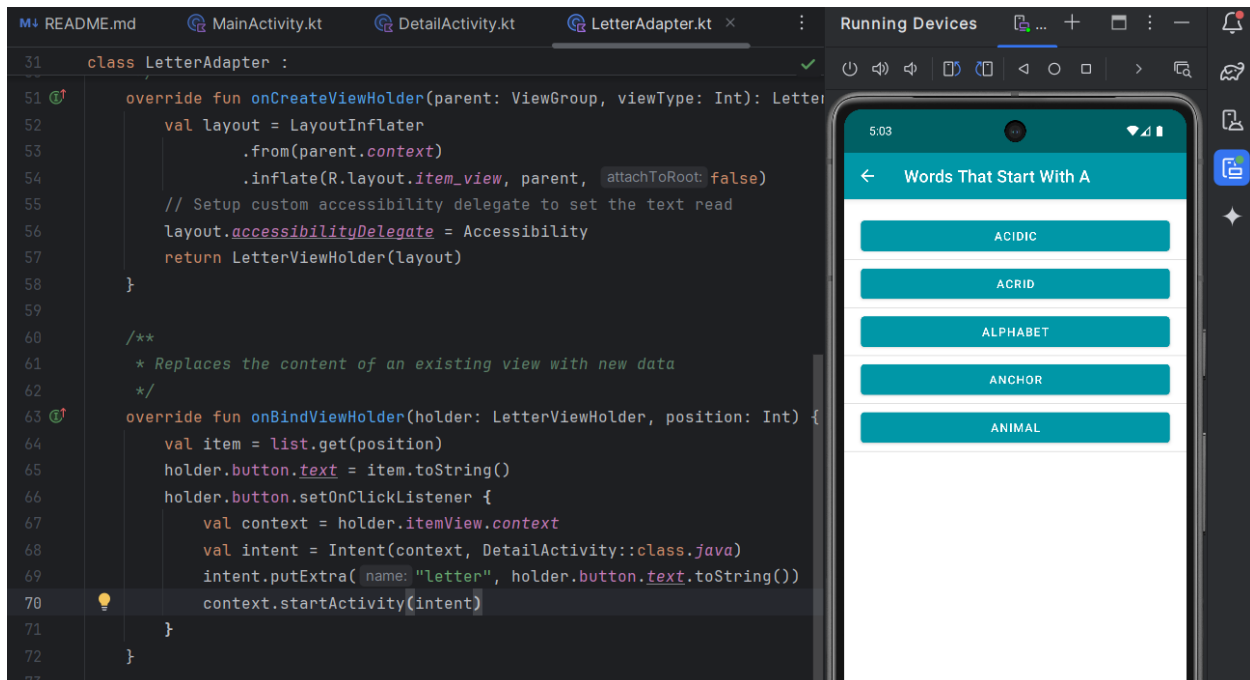
Nói chung, khi muốn hiển thị một activity trong cùng ứng dụng của bạn, bạn sẽ sử dụng explicit intent. Còn khi muốn thực hiện một hành động mà có thể sử dụng một ứng dụng khác (như trình duyệt web, ứng dụng email), bạn sẽ dùng implicit intent.

## 5. Set up Explicit Intent

Trên màn hình đầu tiên, khi nhấn vào một chữ cái, người dùng sẽ được chuyển đến một màn hình thứ hai có danh sách từ vựng. Để làm được ta sẽ thêm onClickListener vào để lắng nghe, biết được người dùng ấn nút nào để hiển thị danh sách các từ bắt đầu bằng chữ đó. onClickListener sẽ được thêm vào onBindViewHolder như sau:

```
63 override fun onBindViewHolder(holder: LetterViewHolder, position: Int) {
64     val item = list.get(position)
65     holder.button.text = item.toString()
66     holder.button.setOnClickListener {
67         val context = holder.itemView.context
68         val intent = Intent(context, DetailActivity::class.java)
69         intent.putExtra(name: "letter", holder.button.text.toString())
70         context.startActivity(intent)
71     }
72 }
```

Code trên sẽ cho ra kết quả chạy như sau



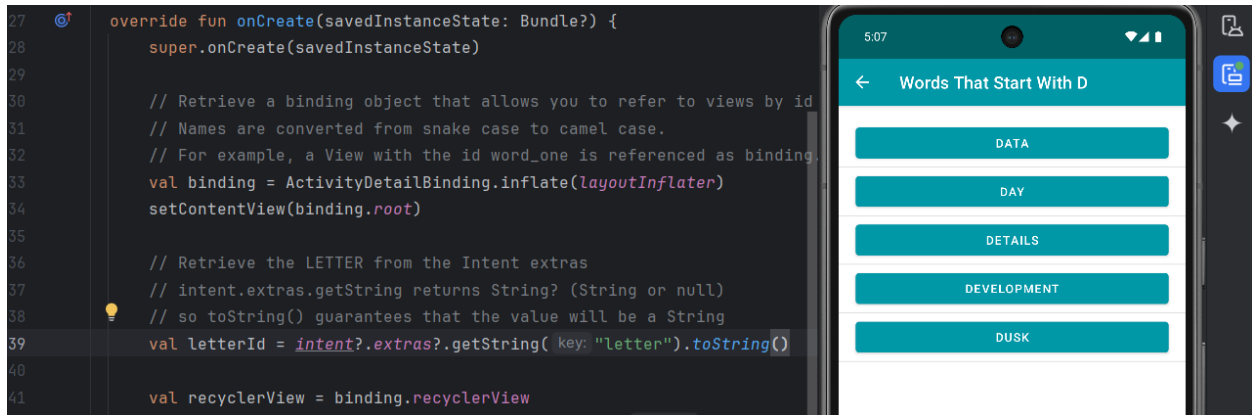
Tuy nhiên khi ấn vào nút chữ cái nào cũng bị hiện ra danh sách các từ bắt đầu bằng chữ A và ta sẽ sửa vấn đề này trong phần 6

## 6. Set up DetailActivity

Trong phương thức `onCreate` của `DetailActivity`, sau lệnh gọi đến `setContentView`, ta thay thế chữ cái đã mã hoá cứng bằng mã để nhận `letterId` được truyền đến từ `intent`.

```
// Retrieve the LETTER from the Intent extras
// intent.extras.getString returns String? (String or null)
// so toString() guarantees that the value will be a String
val letterId = intent?.extras?.getString( key: "letter").toString()
```

Việc này sẽ cho ứng dụng biết thay đổi giá trị `letterId` khi ta ấn các chữ cái khác nhau, code lần này sẽ chạy đúng



### 6.1. Clean up

Ta sẽ tái cấu trúc mã bằng cách sử dụng *companion object* (đối tượng đồng hành) trong Kotlin, giúp quản lý các hằng số trong ứng dụng dễ dàng và sạch sẽ hơn, đặc biệt khi làm việc với các tên extra (extra keys) cho `Intent`.

Các bước thực hiện:

1. Tạo hằng số trong Companion Object: Tạo companion object trong `DetailActivity` và khai báo hằng số `LETTER`.
2. Sử dụng hằng số: Thay thế các chuỗi mã hóa cứng bằng cách gọi hằng số `DetailActivity.LETTER`.
3. Cập nhật các lệnh gọi Intent: Khi chuyển dữ liệu qua Intent, ta sử dụng `putExtra(DetailActivity.LETTER, ...)`, giúp mã rõ ràng và dễ dàng bảo trì hơn.

```
companion object {
    const val LETTER = "letter"
}
```

```
// so toString() guarantees that the value will be a String
val letterId = intent?.extras?.getString(LETTER).toString()
```

```
override fun onBindViewHolder(holder: LetterViewHolder, position: Int) {
    val item = list.get(position)
    holder.button.text = item.toString()
    holder.button.setOnClickListener {
        val context = holder.itemView.context
        val intent = Intent(context, DetailActivity::class.java)
        intent.putExtra(DetailActivity.LETTER, holder.button.text.toString())
        context.startActivity(intent)
    }
}
```

## 7. Set up Implicit Intent

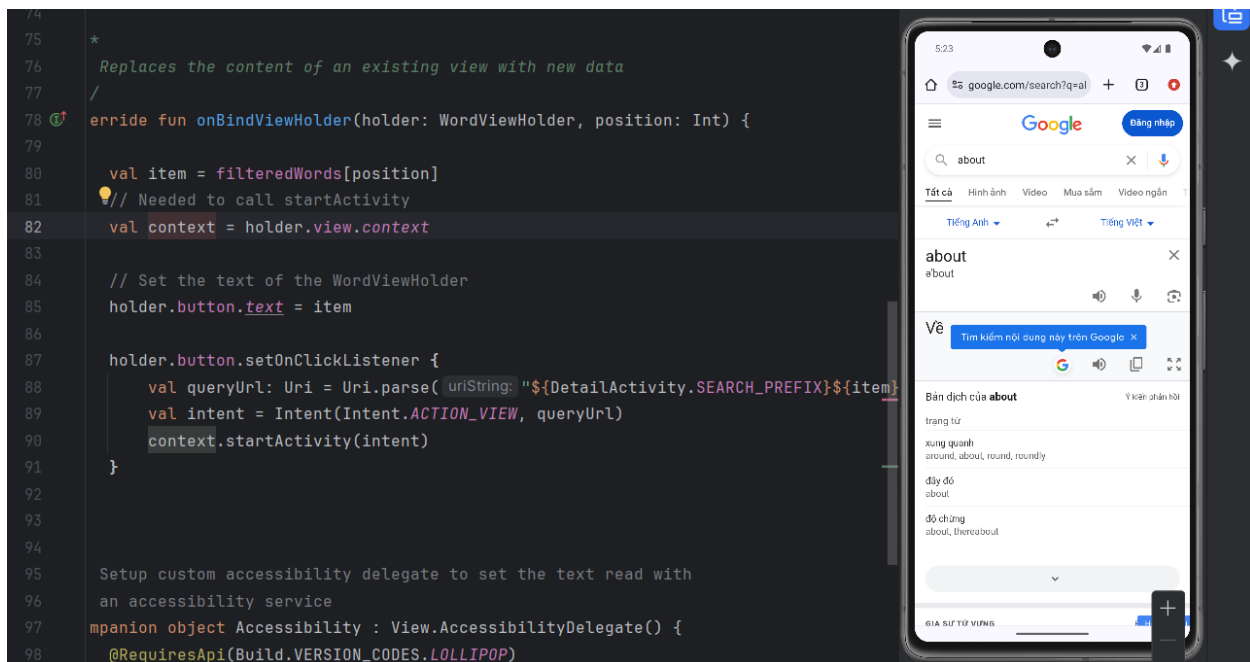
Giờ ta sẽ làm cho khi ấn vào 1 từ, ta có thể chuyển hướng sang chrome và tìm kiếm từ đó trên google

```
companion object {
    const val LETTER = "letter"
    const val SEARCH_PREFIX = "https://www.google.com/search?q="
}
```

Thêm Prefix cho tìm kiếm vì tất cả các từ đều bắt đầu từ URI này

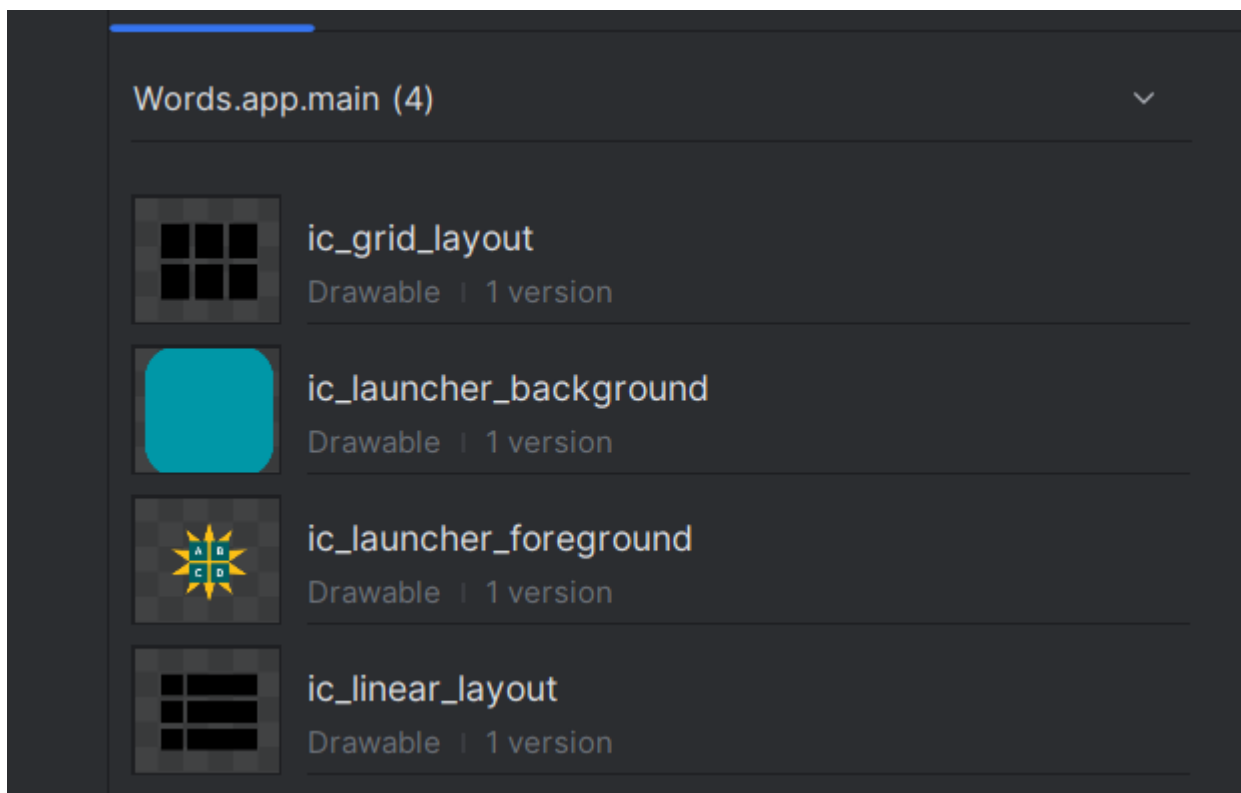
```
holder.button.setOnClickListener {
    val queryUrl: Uri = Uri.parse("${DetailActivity.SEARCH_PREFIX}${item}")
    val intent = Intent(Intent.ACTION_VIEW, queryUrl)
    context.startActivity(intent)
}
```

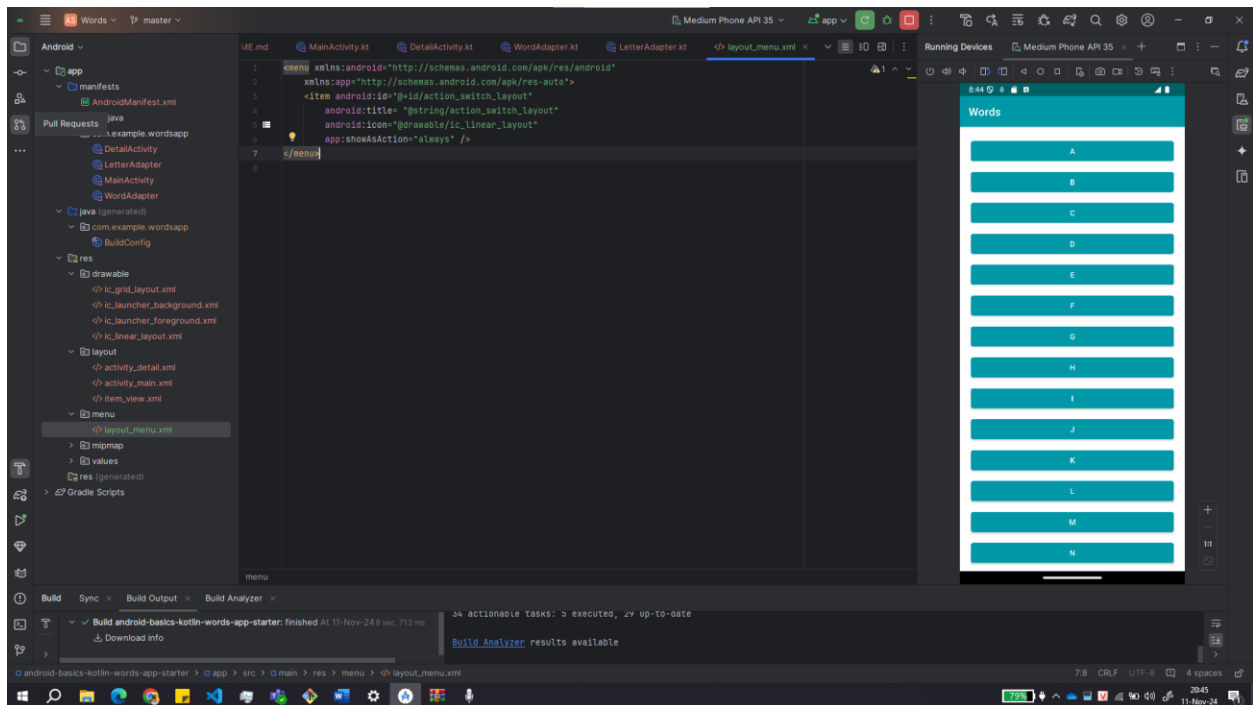
Thêm setOnClickListener vào để bắt event và sau đó khởi tạo intent mới cuối cùng là startActivity để thực thi tìm kiếm



## 8. Set up Menu and Icons

Trong phần này ta sẽ thêm các tùy chọn cho người dùng, ví dụ như chuyển dạng list thành grid





Đề file .xml trên ta cần chuyển sang bước 9 để sửa code

## 9. Implement Menu button

Các bước thực hiện :

Tạo thuộc tính theo dõi trạng thái

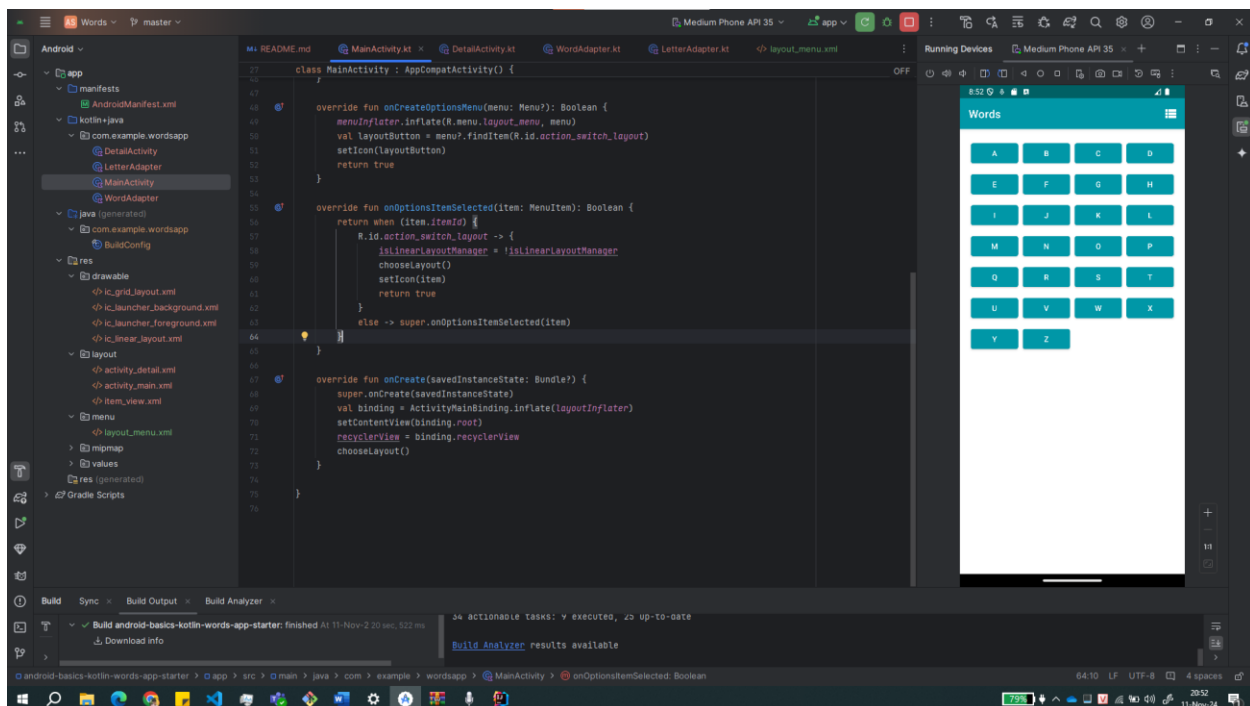
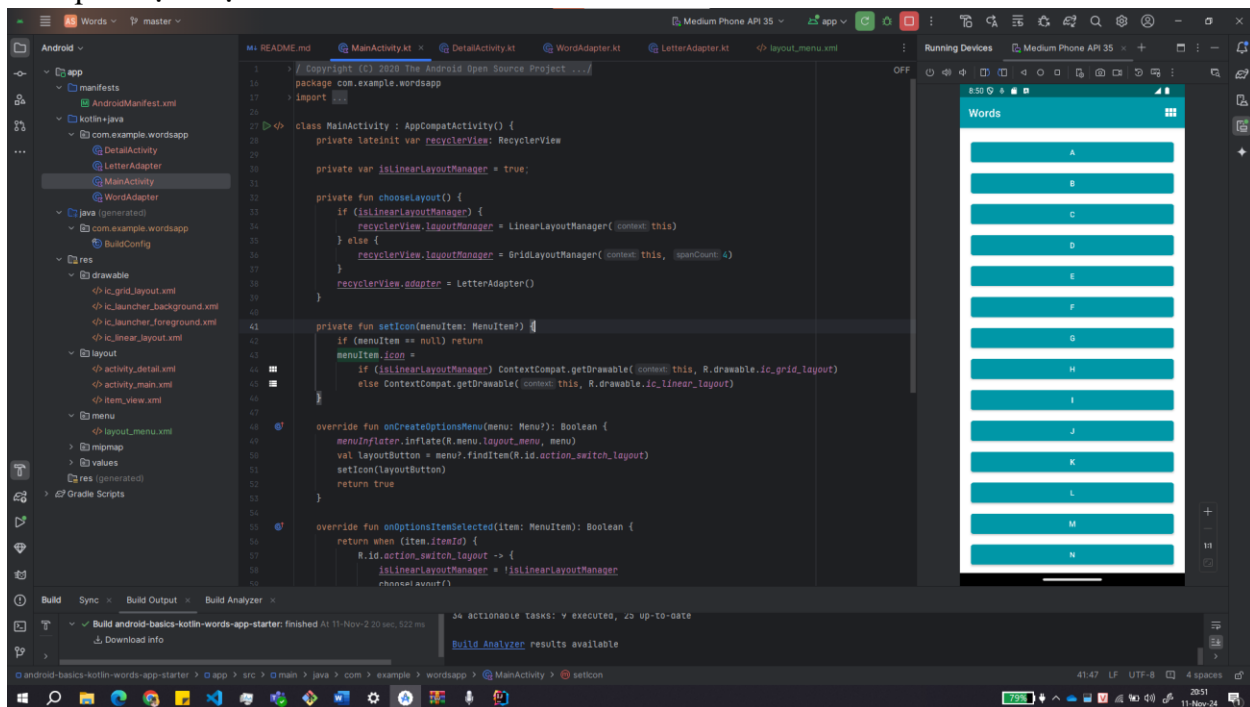
Thêm hàm chọn layout

Khi chuyển trạng thái thì cần đổi icon

- onOptionsItemSelected: nơi bạn mở rộng trình đơn tùy chọn và thực hiện các bước thiết lập bổ sung
- onOptionsItemSelected: nơi bạn sẽ thực sự gọi đến `chooseLayout()` khi nút này được chọn.



## Kết quả thực hiện:



## 10. Solution code

## 11. Summary

## 12. Learn more