

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 6 – Create XML layouts for Android – Calculate the tip

Contents

1. Create XML layouts for Android.....	3
1.1. Before your begin	3
1.2. Start the project.....	3
1.3. Read and understand XML.....	3
1.4. Build the layout in XML	4
1.4.1. Choose file activity_main.xml.....	4
1.4.2. Delete the TextView	5
1.4.3. Add cost of service text field	6
1.4.4. Review the EditText attributes	6
1.4.5. Add the service question.....	7
1.5. Add tip options.....	8
1.5.1. Add Radio group.....	8
1.6. Complete the rest of the layout.....	10
1.6.1. Add a Switch for rounding up the tip	10
1.6.2. Add the Calculate button	11
1.6.3. Add tip result	12
1.7. Adopt good coding practices	13
1.7.1. Extract the strings	13
1.7.2. Reformat the XML	13
2. Calculate the tip	13
2.1. Before your begin	13
2.2. Start app overview	14

2.3. View binding.....	14
2.4. Calculate the tip	15
2.5. Test and debug	17
2.6. Adopt good coding practicves	18

1. Create XML layouts for Android

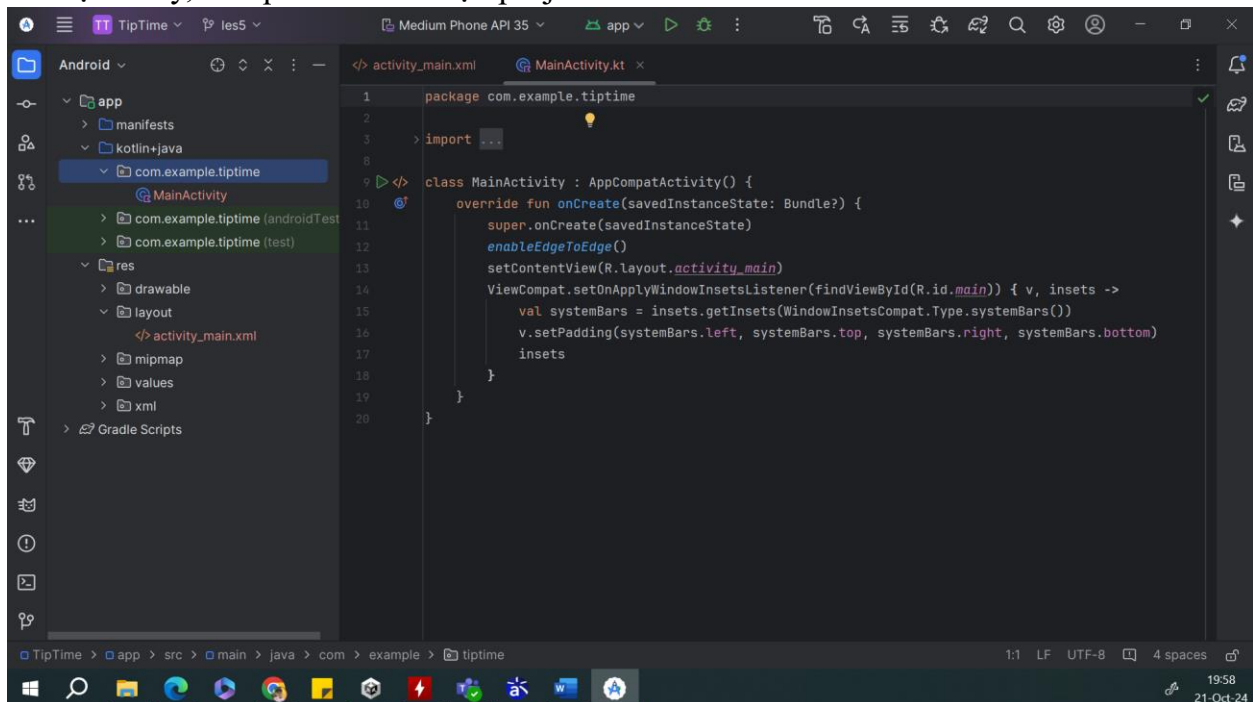
1.1. Before your begin

Kiến thức bạn sẽ học được

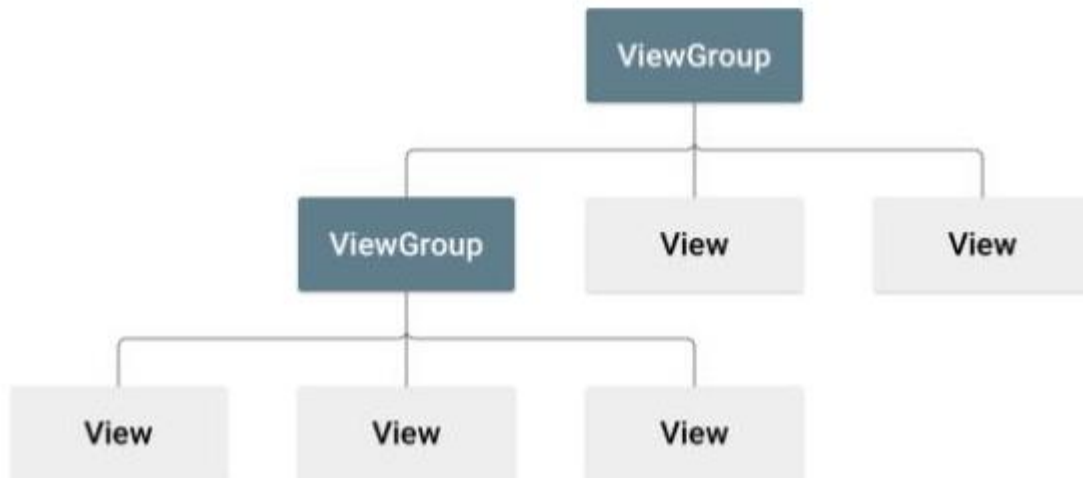
- Biết cách đọc và ghi bố cục XML trong Android
- Biết cách tạo bố cục cho một biểu mẫu đơn giản để người dùng nhập văn bản và đưa ra các lựa chọn

1.2. Start the project

Tạo project Tip Time, các bước thực hiện để tạo project như mọi khi, em xin phép không viết lại ở đây, kết quả của bước tạo project như sau:



1.3. Read and understand XML



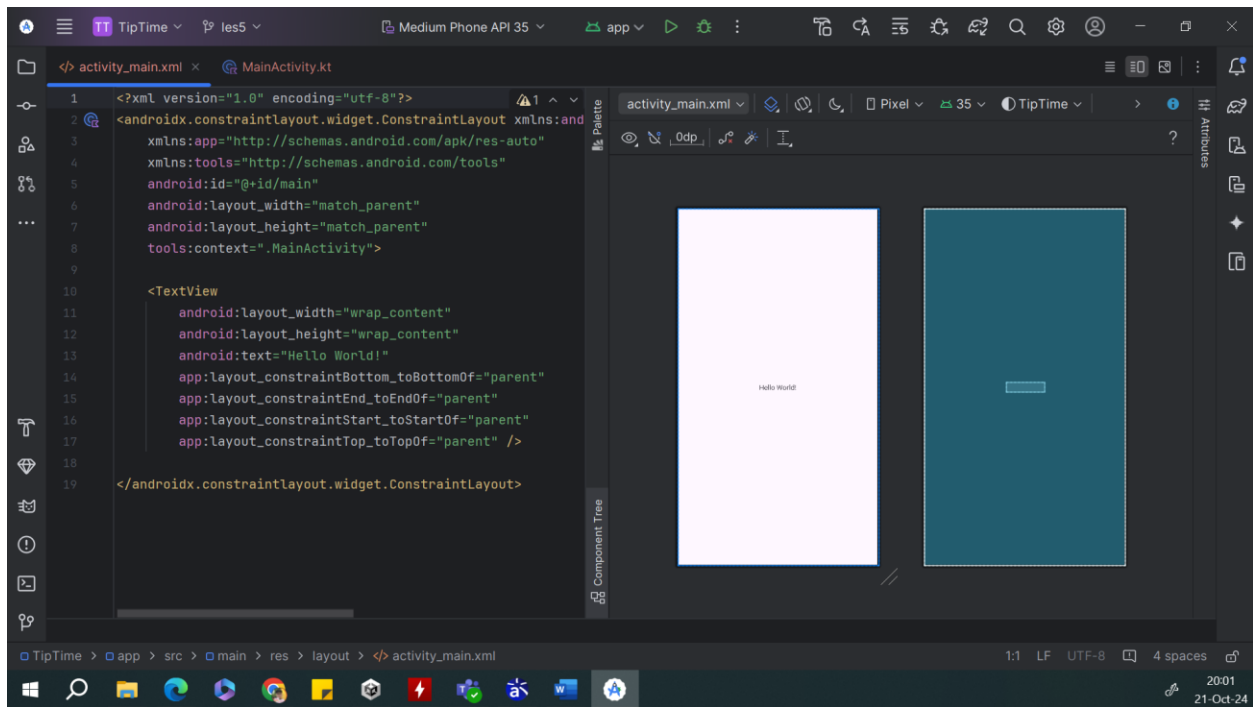
ConstraintLayout (parent)



1.4. Build the layout in XML

1.4.1. Choose file activity_main.xml

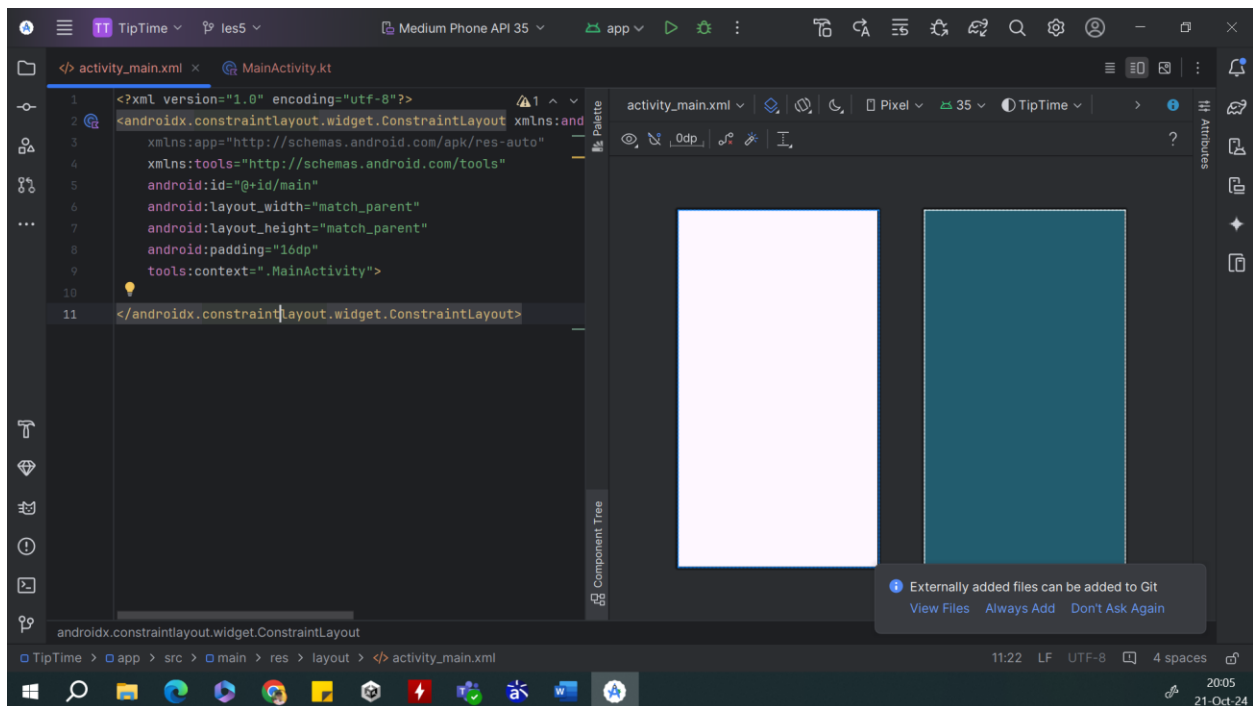
Chọn file activity_main.xml, chuyển sang chế độ split



1.4.2. Delete the TextView

Các bước thực hiện:

1. Xóa phần Text View
2. Thêm padding 16dp vào ConstraintLayout để UI không bị sát lề

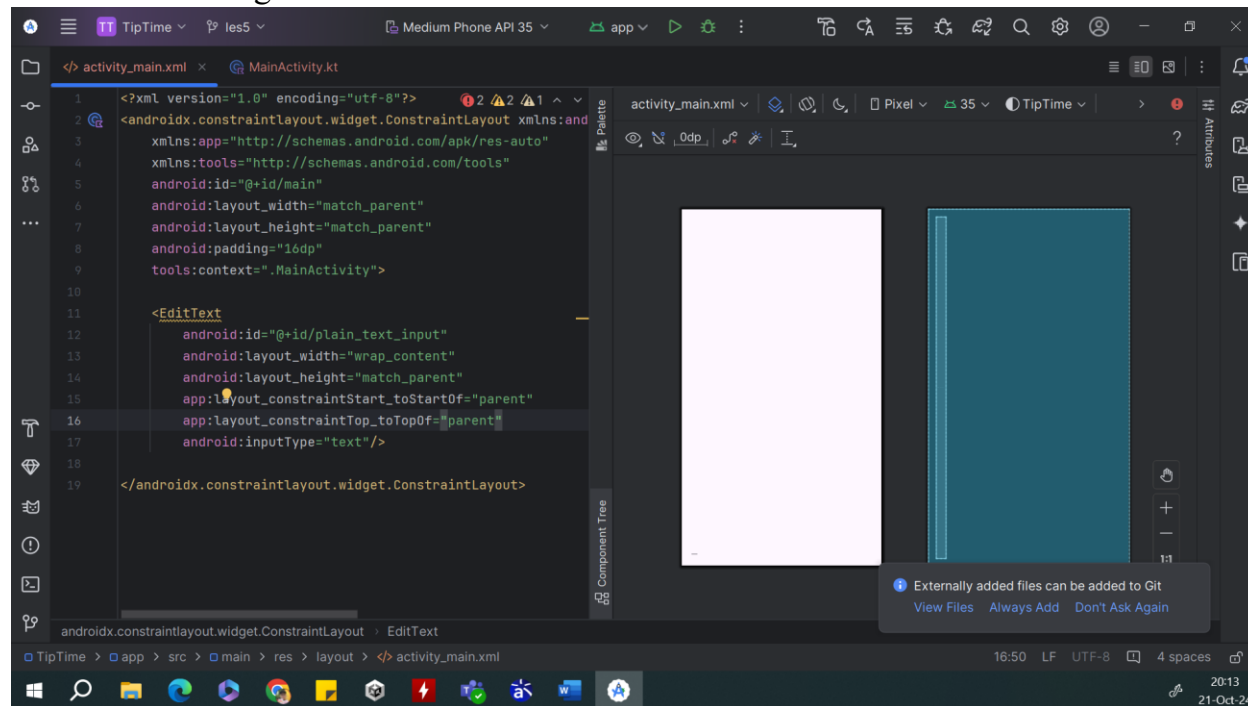


1.4.3. Add cost of service text field

Các bước thực hiện:

1. Thêm code cho phần EditText
2. Thêm code cho các đặc tính của EditText

Code cuối sẽ trông như sau:

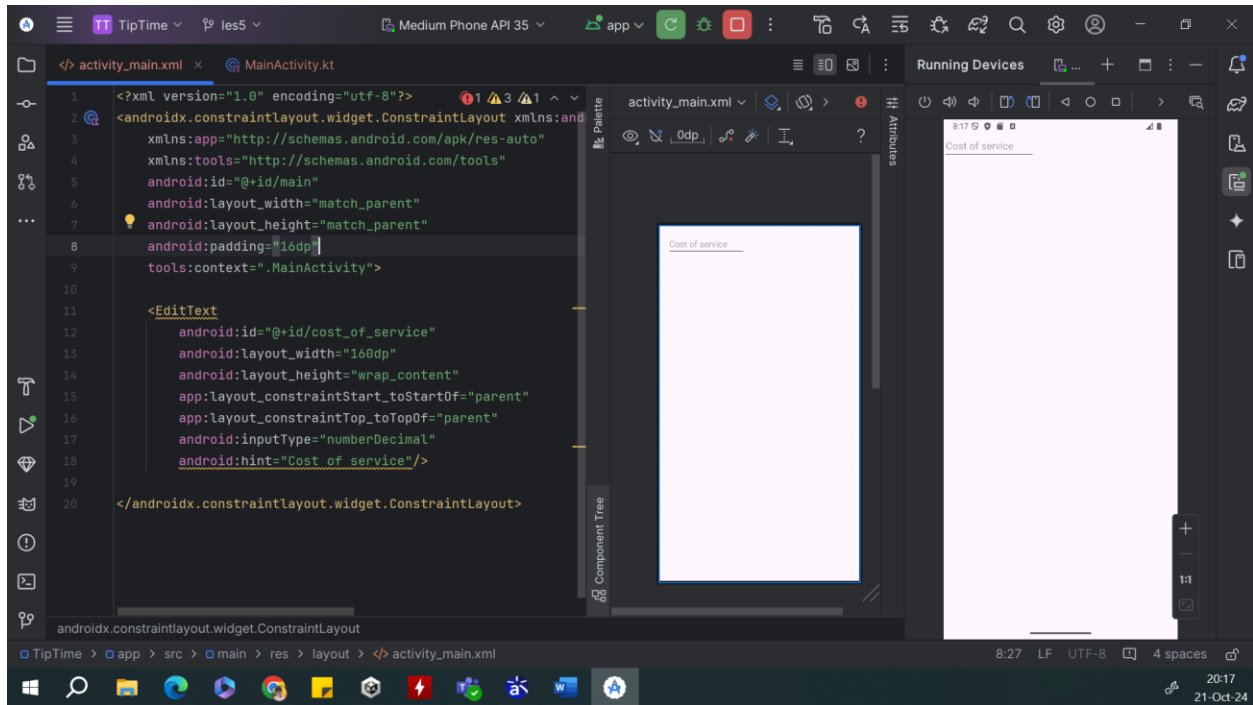


1.4.4. Review the EditText attributes

Các bước thực hiện:

1. Tìm thuộc tính id: @+id/plain_text_input
2. Thay đổi id thành @+id/cost_of_service
3. Sửa layout_width thành 160dp và inputType thành numberDecimal
4. Thêm hint
5. Chạy app

Kết quả:

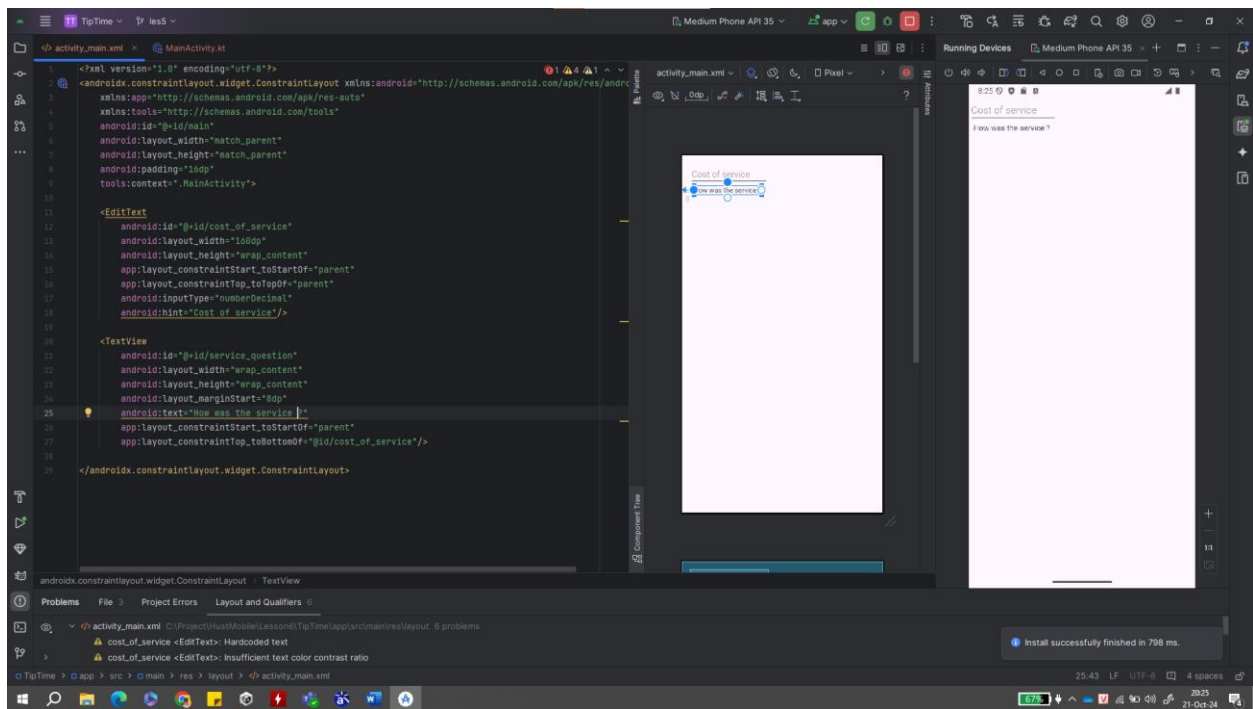


1.4.5. Add the service question

Các bước thực hiện:

1. Sau tag đóng của EditText, thêm dòng mới và gõ <TextView
2. Chọn wrap_content cho cả layout_width và layout_height
3. Thêm thuộc tính text với “How was the service?” và đóng tag
4. Thêm một horizontal constraint vào TextView để nó bắt đầu từ lề của lề của parent, tương tự thêm vertical constraint
5. Thêm ID vào TextView

Kết quả:

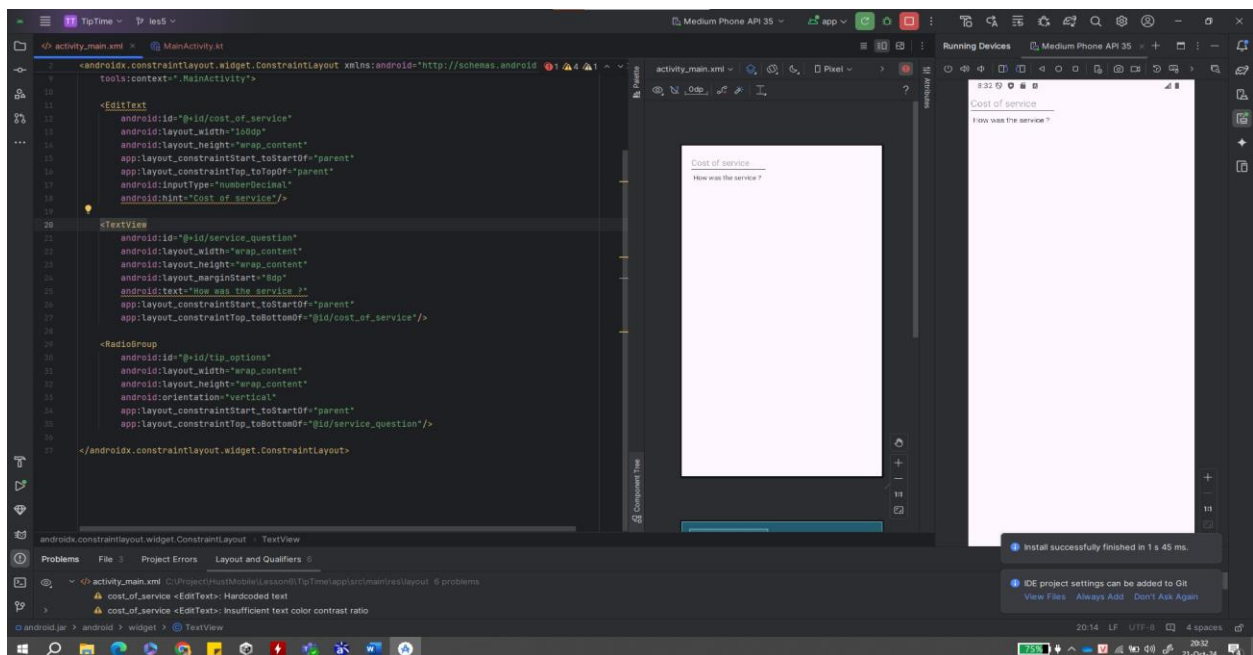


1.5. Add tip options

1.5.1. Add Radio group

Các bước thực hiện:

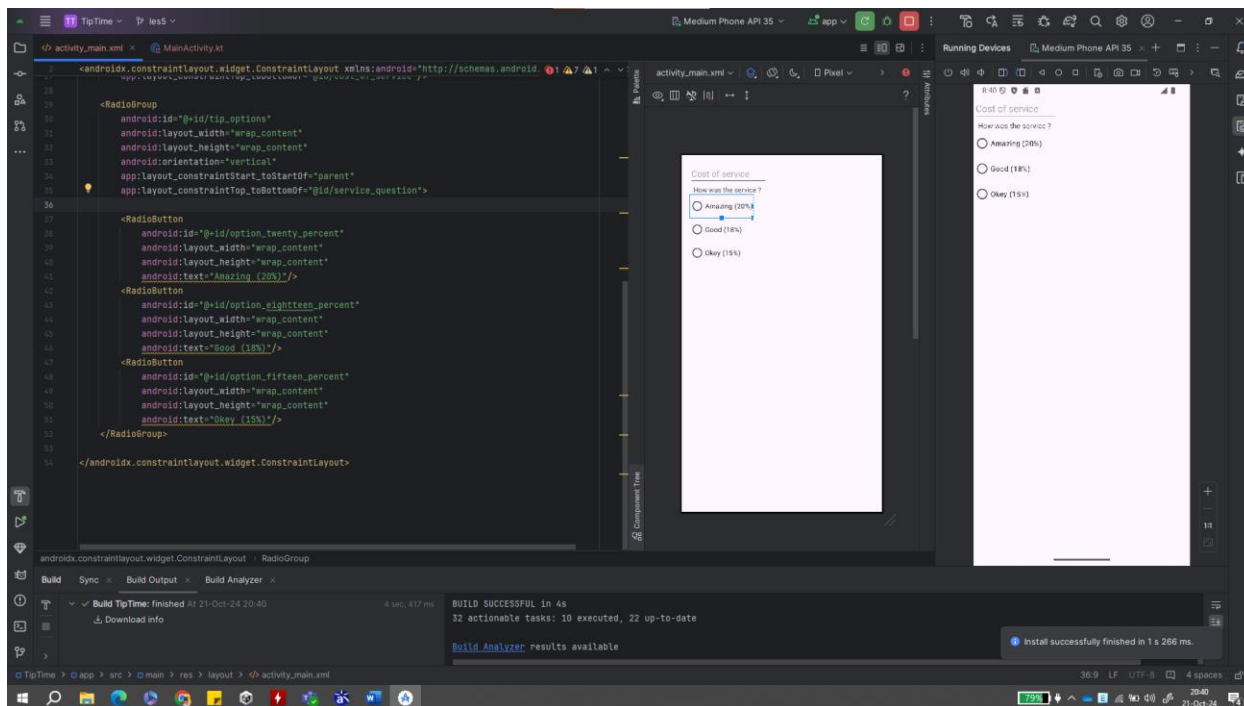
1. Thêm `RadioGroup` và `RadioButton` vào app
2. Đặt `layout_width` và `layout_height` của `RadioGroup` là `wrap_content`, thêm ID là `@+id/tip_options`, đóng tag, thêm các thuộc tính



1.5.2. Add Radio button

Các bước thực hiện:

1. Thêm RadioButton, đặt layout_width và layout_height thành wrap_content, thêm ID là @+id/option_twenty_percent vào RadioButton, đặt text là Amazing (20%), đóng tag
2. Tương tự thêm RadioButton Good (18%) và Okay (15%)
3. Chạy app

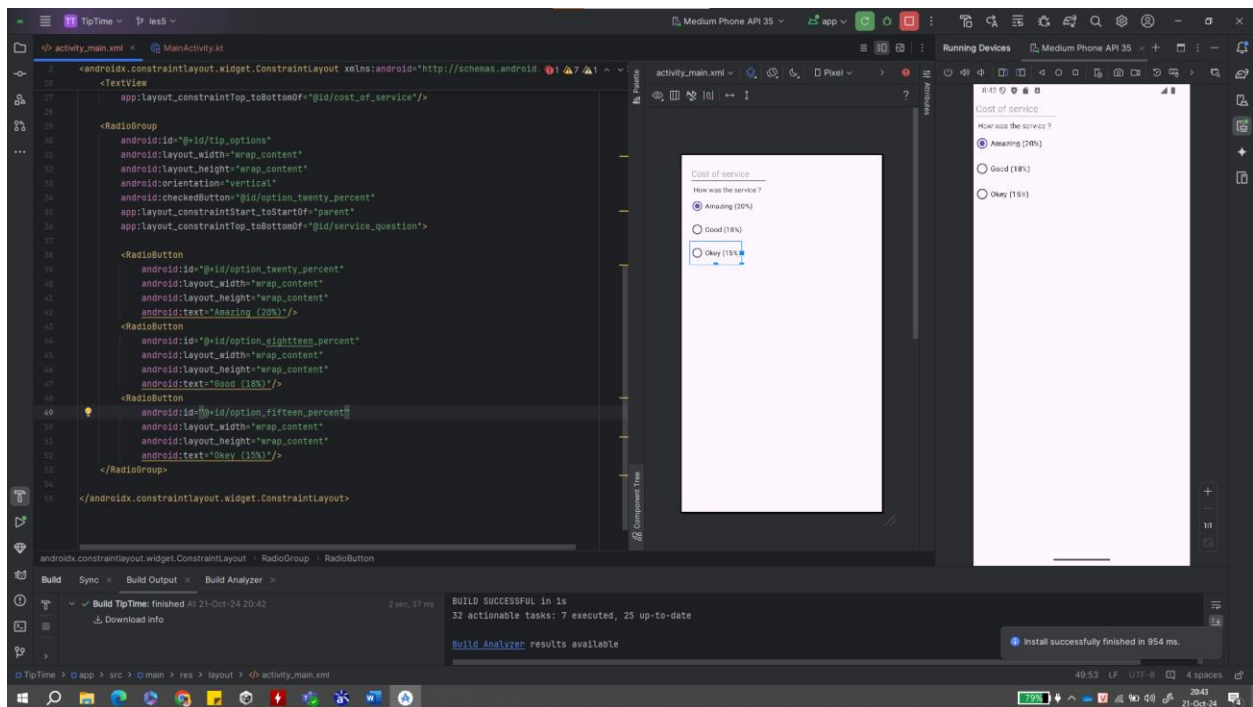


1.5.3. Add Default selection

Các bước thực hiện:

1. Trong RadioGroup, đặt thuộc tính android:checkedButton thành @id/option_twenty_percent

Kết quả:



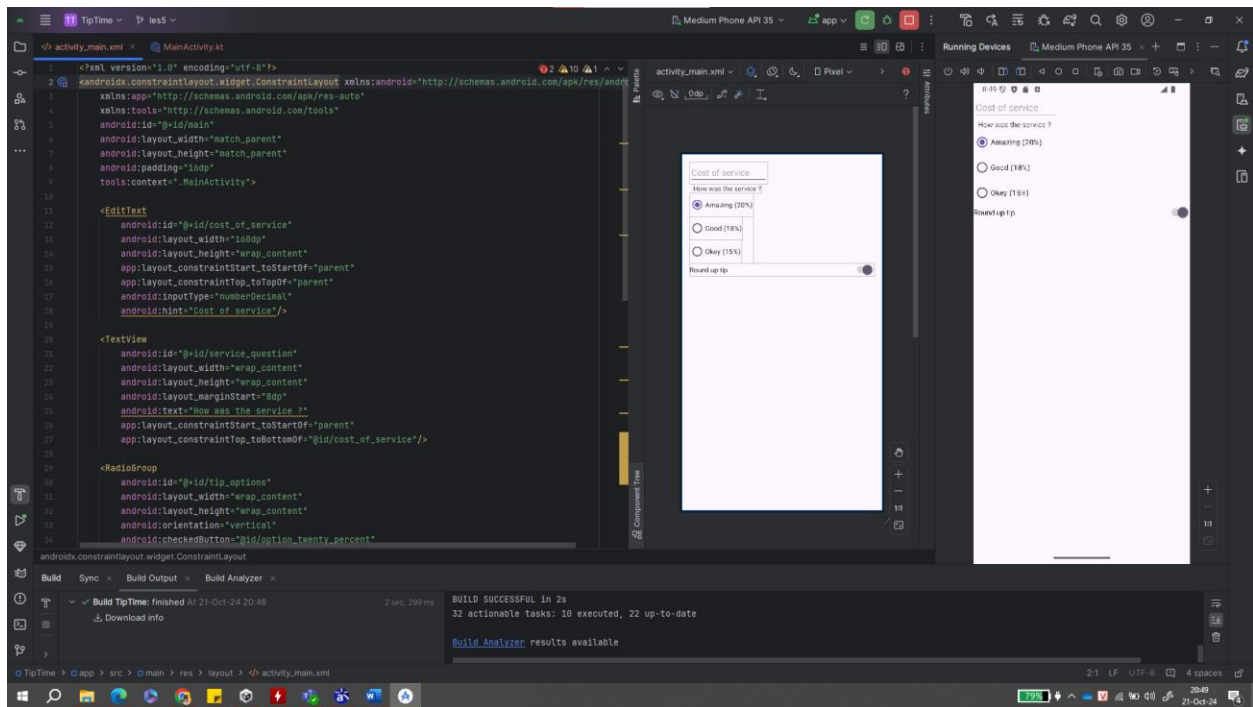
1.6. Complete the rest of the layout

1.6.1. Add a Switch for rounding up the tip

Các bước thực hiện:

1. Thêm code cho phần Switch
2. Chạy chương trình

Kết quả:

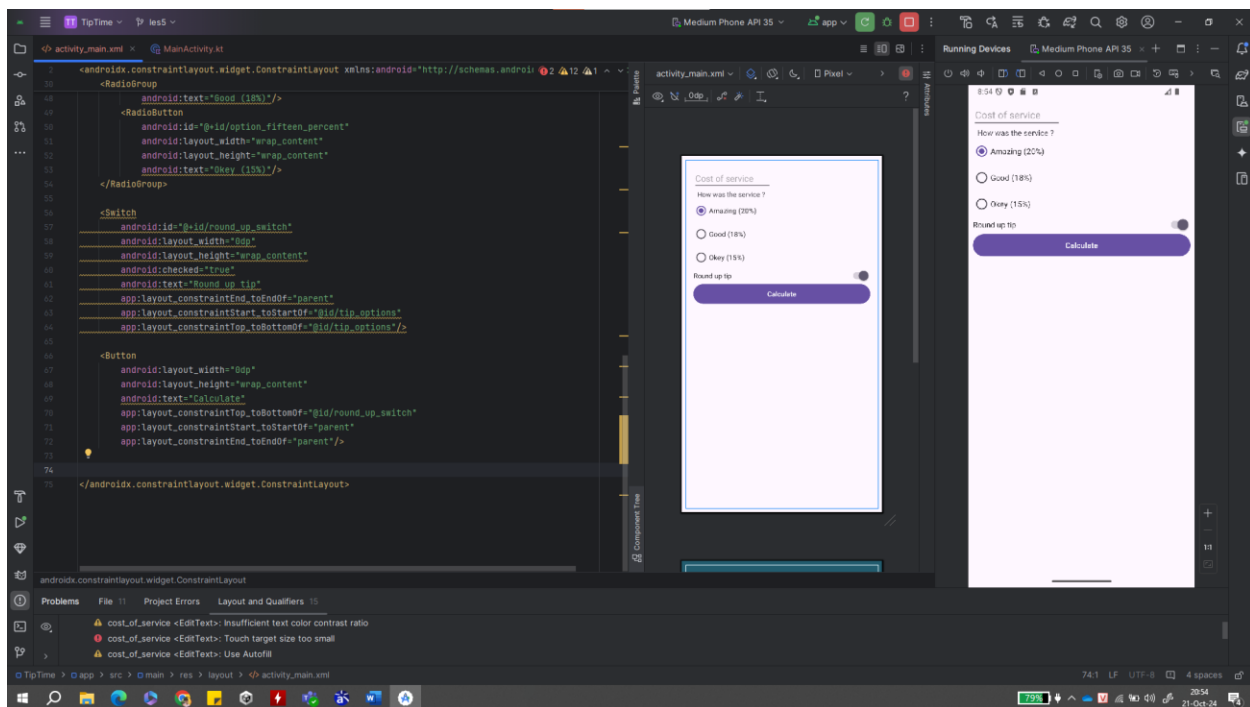


1.6.2. Add the Calculate button

Các bước thực hiện:

1. Thêm Button
2. Sửa thêm các thuộc tính
3. Chạy chương trình

Kết quả:

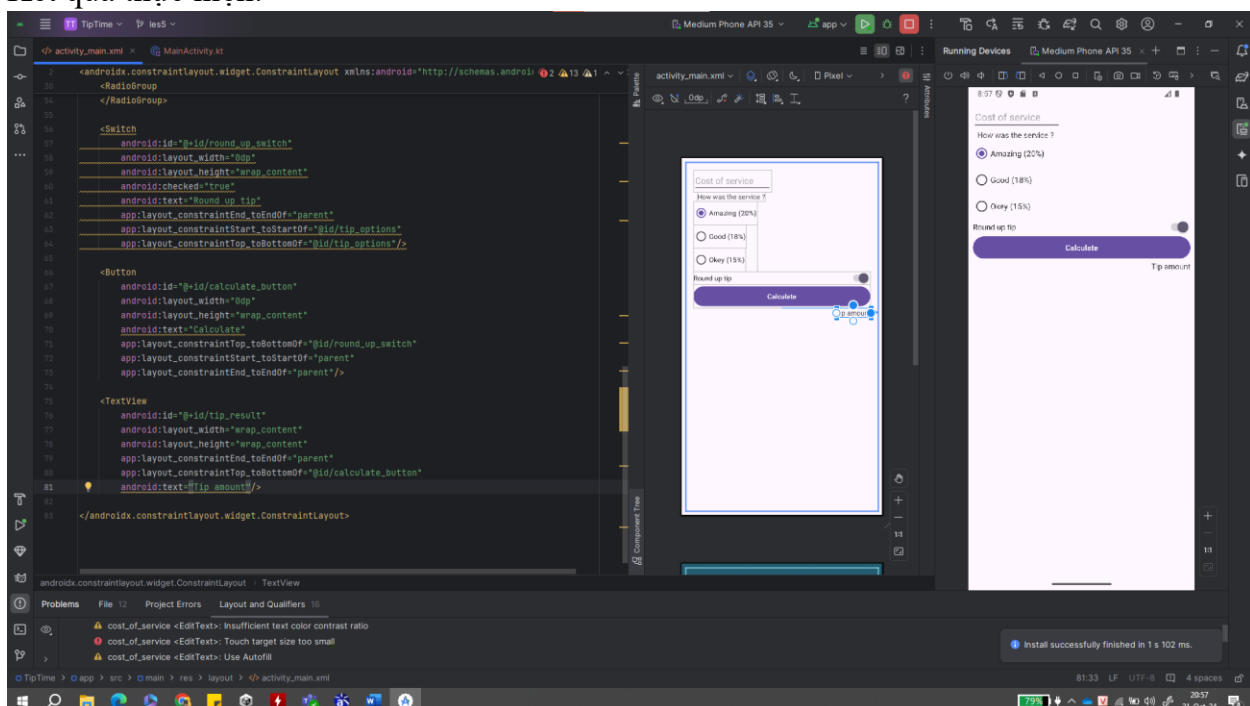


1.6.3. Add tip result

Các bước thực hiện:

1. Thêm TextView để hiển thị kết quả tip
2. Chạy ứng dụng

Kết quả thực hiện:

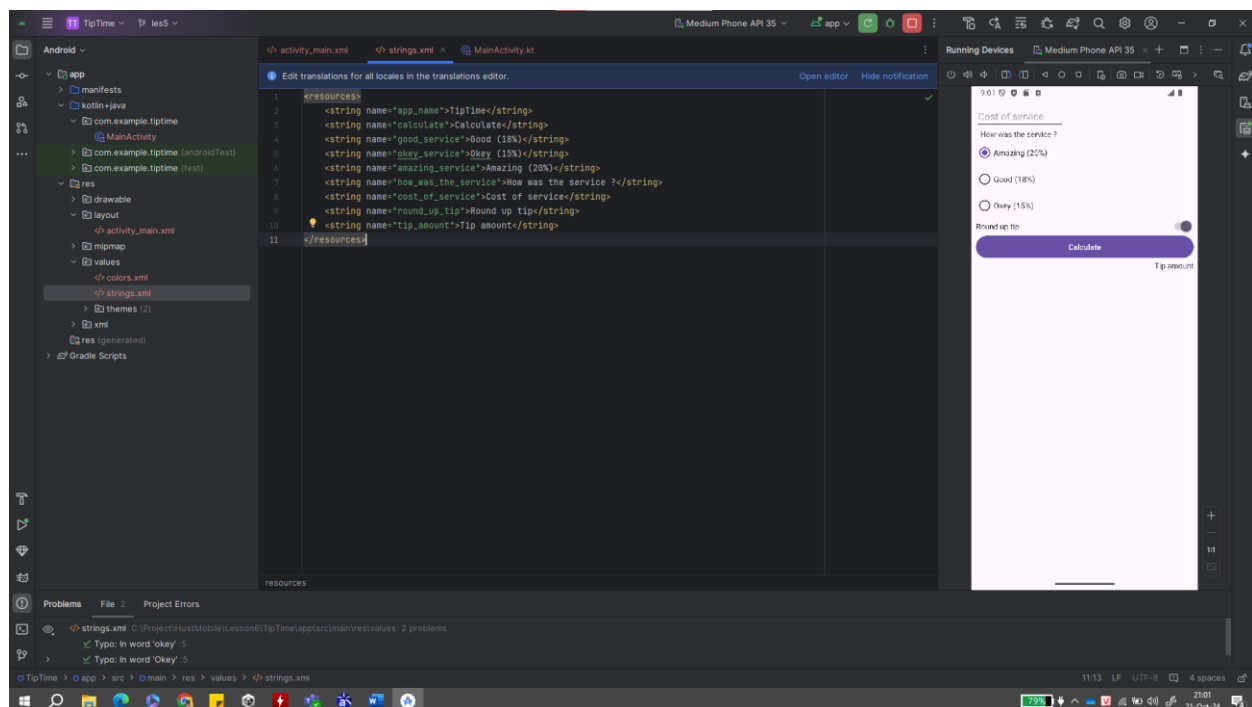


1.7. Adopt good coding practices

1.7.1. Extract the strings

Các bước thực hiện:

1. Nhấn vào bóng đèn bên cạnh dòng code bị warning (với những strings)
2. Mở app > res > values > strings.xml, thêm như mẫu



1.7.2. Reformat the XML

Các bước thực hiện:

1. Trong activity_main.xml, chọn Edit > Select All
2. Chọn Code > Reformat Code

2. Calculate the tip

2.1. Before your begin

Kiến thức bạn sẽ học được

- Cấu trúc cơ bản của ứng dụng Android.
- Cách đọc giá trị trong giao diện người dùng để nhập vào mã và thao tác với các giá trị đó.
- Cách sử dụng tính năng liên kết khung hiển thị (view binding) thay vì findViewById() để dễ dàng viết mã tương tác với các khung hiển thị.
- Cách xử lý số thập phân trong Kotlin qua loại dữ liệu Double.

- Cách định dạng số dưới dạng đơn vị tiền tệ.
- Cách sử dụng tham số chuỗi để tự động tạo chuỗi.
- Cách dùng Logcat trong Android Studio để phát hiện vấn đề trong ứng dụng.

2.2. Start app overview

Mở project trước đó Tip Time

Giới thiệu về Gradle:

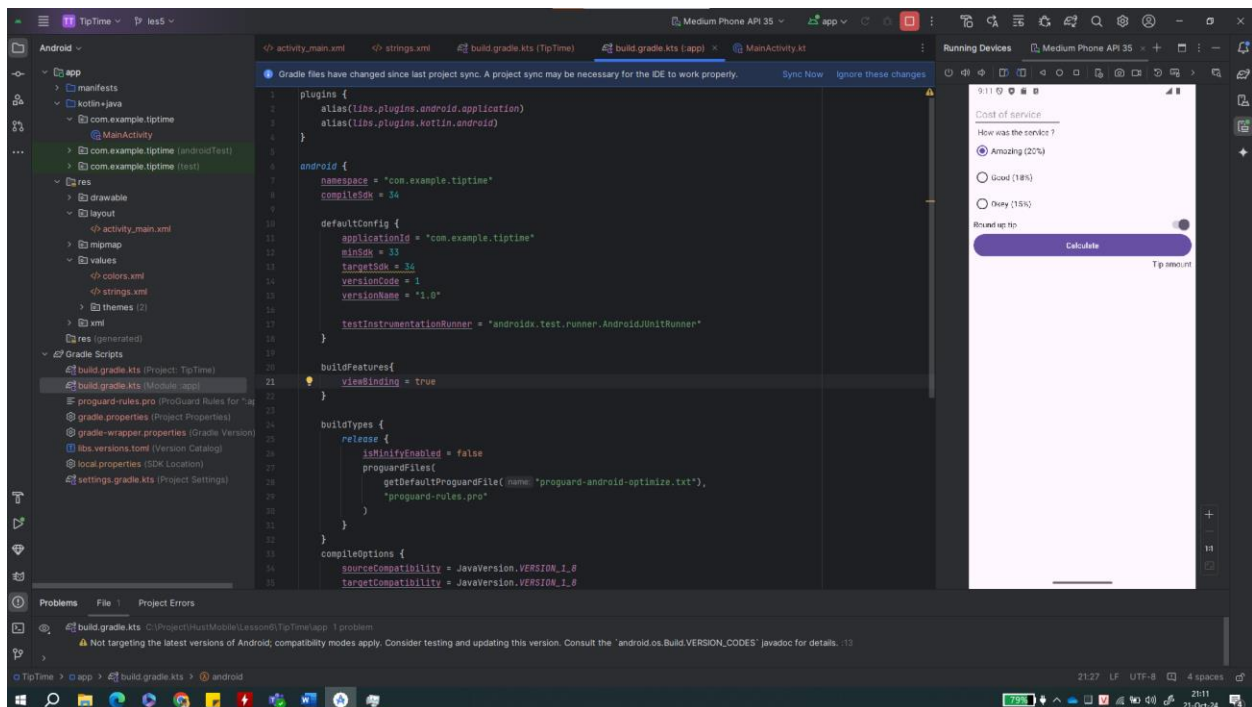
Gradle là một hệ thống xây dựng tự động mà Android Studio sử dụng. Bất cứ khi nào bạn thay đổi mã, thêm tài nguyên hoặc thực hiện các thay đổi khác đối với ứng dụng, Gradle sẽ tìm hiểu xem những gì đã thay đổi và thực hiện các bước cần thiết để tạo lại ứng dụng. Gradle cũng cài đặt ứng dụng của bạn trong trình mô phỏng hoặc thiết bị thực và kiểm soát việc thực thi ứng dụng.

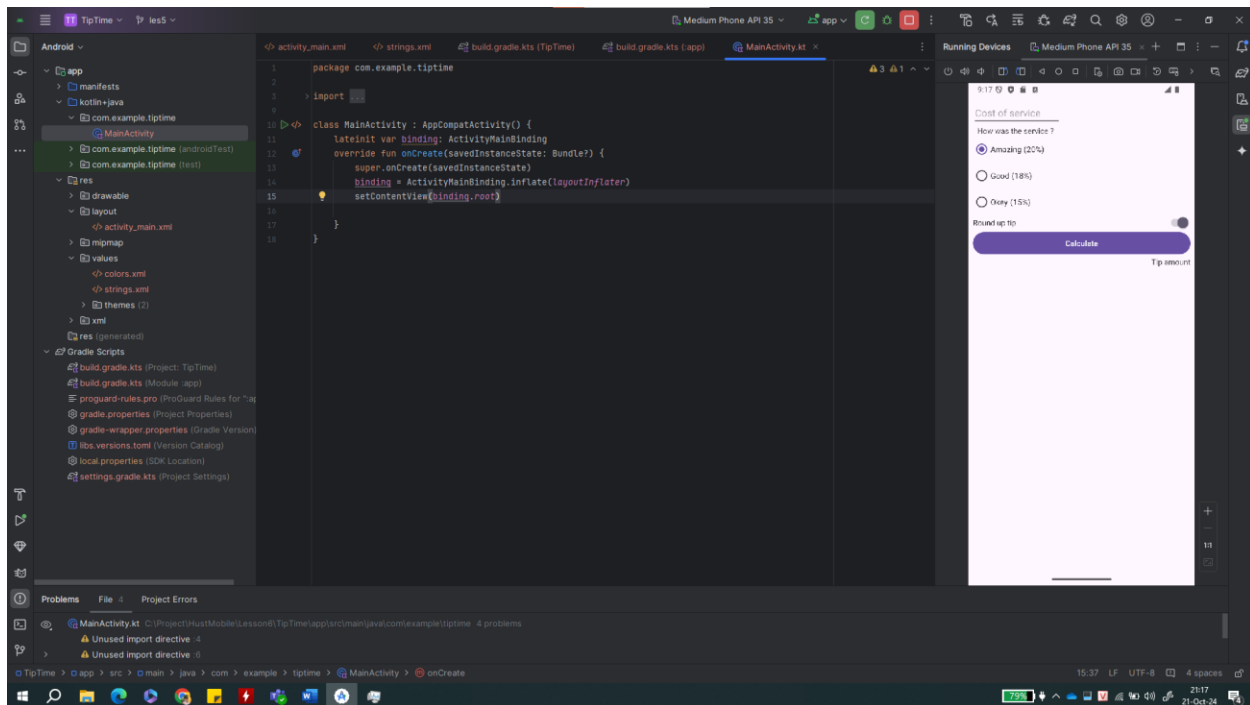
2.3. View binding

Các bước thực hiện:

1. Bật liên kết thành phần hiển thị
2. Chọn đồng bộ hóa ngay (Sync now)
3. Cập nhật MainActivity

Kết quả thực hiện:





Giải thích:

- Dòng dưới đây khai báo biến cấp cao nhất trong lớp này cho đối tượng liên kết. Biến này được định nghĩa ở cấp này vì sẽ được sử dụng trên nhiều phương thức trong lớp MainActivity.

`lateinit var binding: ActivityMainBinding`

Từ khoá `lateinit` là kiến thức mới. Từ khoá này đảm bảo mã của bạn sẽ khởi chạy biến trước khi sử dụng biến đó. Nếu bạn không làm như vậy, ứng dụng của bạn sẽ gặp sự cố.

- Dòng này sẽ khởi chạy đối tượng `binding` mà bạn sẽ dùng để truy cập vào Views trong bố cục `activity_main.xml`.

`binding = ActivityMainBinding.inflate(layoutInflater)`

- Thiết lập thành phần hiển thị nội dung của hoạt động (activity). Thay vì truyền giá trị nhận dạng tài nguyên của bố cục, `R.layout.activity_main`, thao tác này sẽ chỉ định gốc của hệ phân cấp khung hiển thị trong ứng dụng `binding.root`.

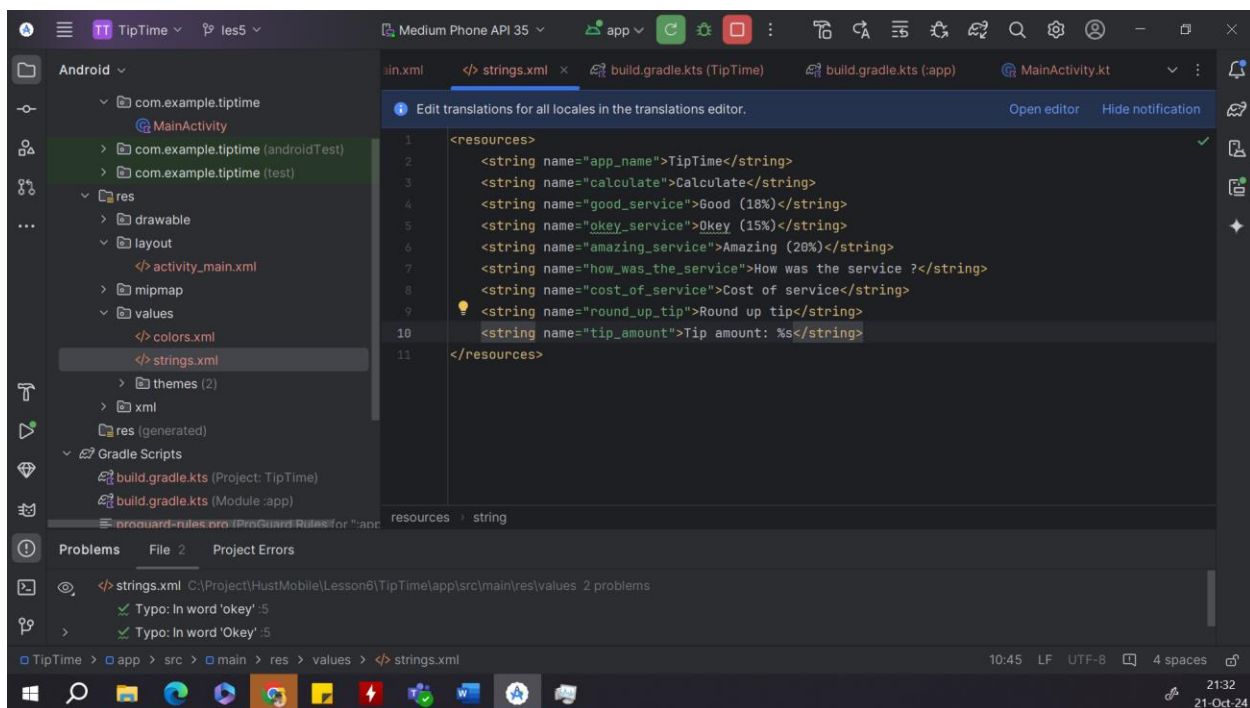
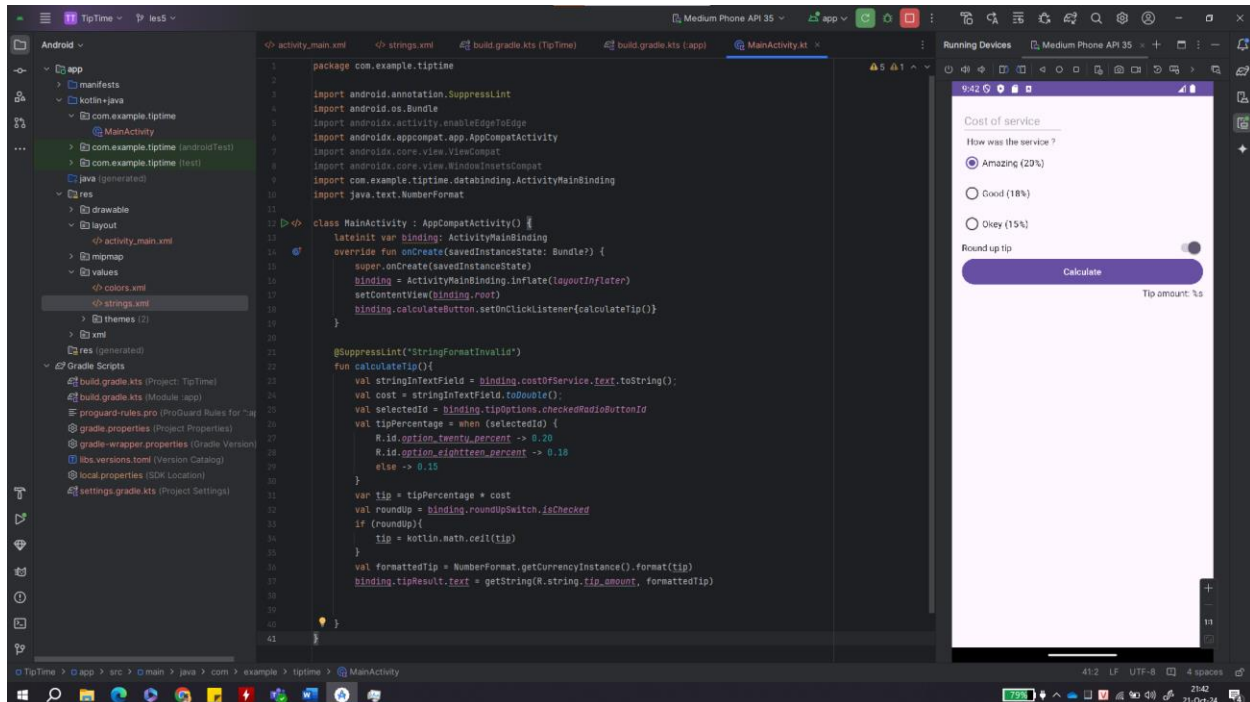
`setContentView(binding.root)`

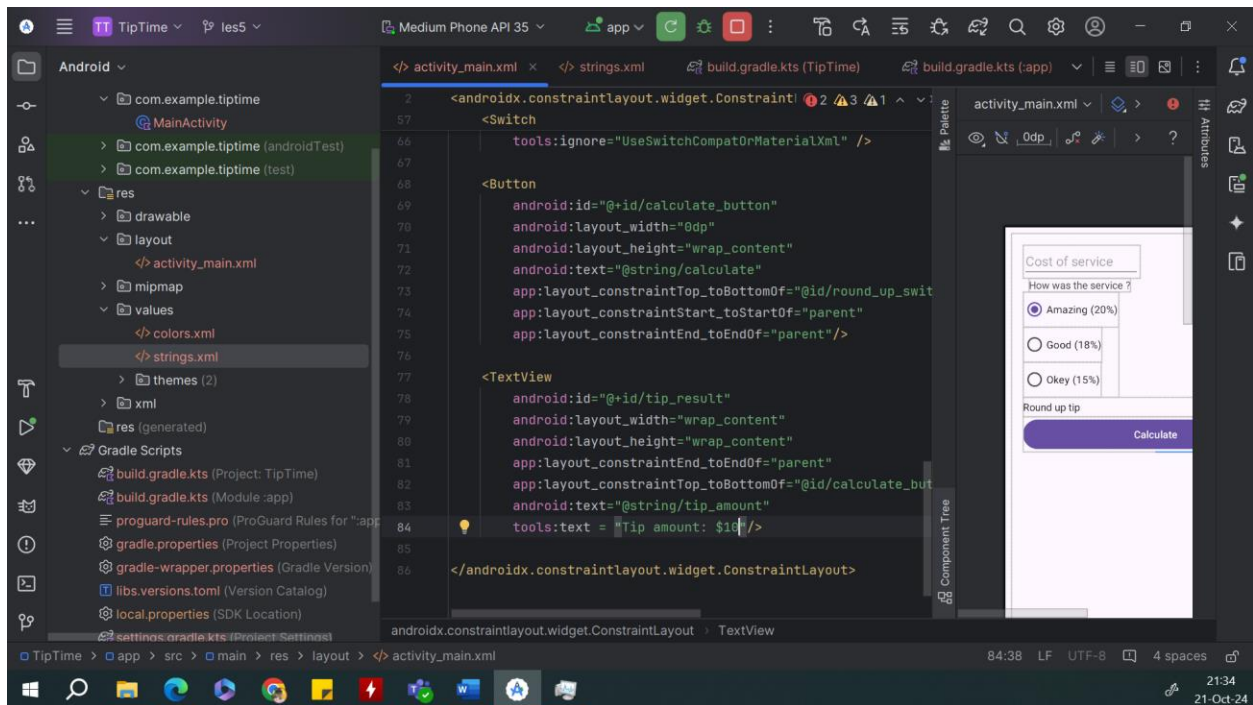
2.4. Calculate the tip

Các bước thực hiện:

1. Cập nhật MainActivity để bắt sự kiện submit từ button Calculate
2. Cập nhật string để hiển thị giá trị của số tiền

3. Thêm một dòng cho thuộc tính tools:text, thiết lập thành Tip Amount: \$10. Vì đây chỉ là một phần giữ chỗ nên bạn không cần trích xuất chuỗi này thành tài nguyên. Phần giữ chỗ sẽ không xuất hiện khi bạn chạy ứng dụng.





2.5. Test and debug

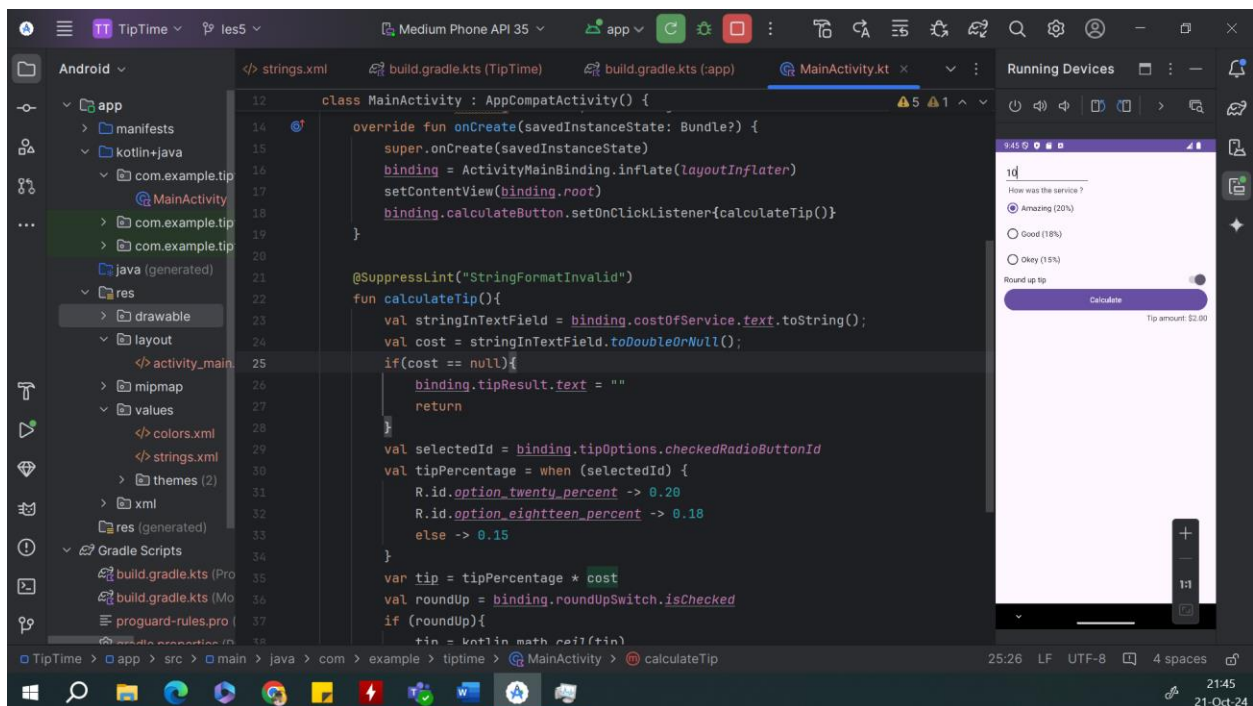
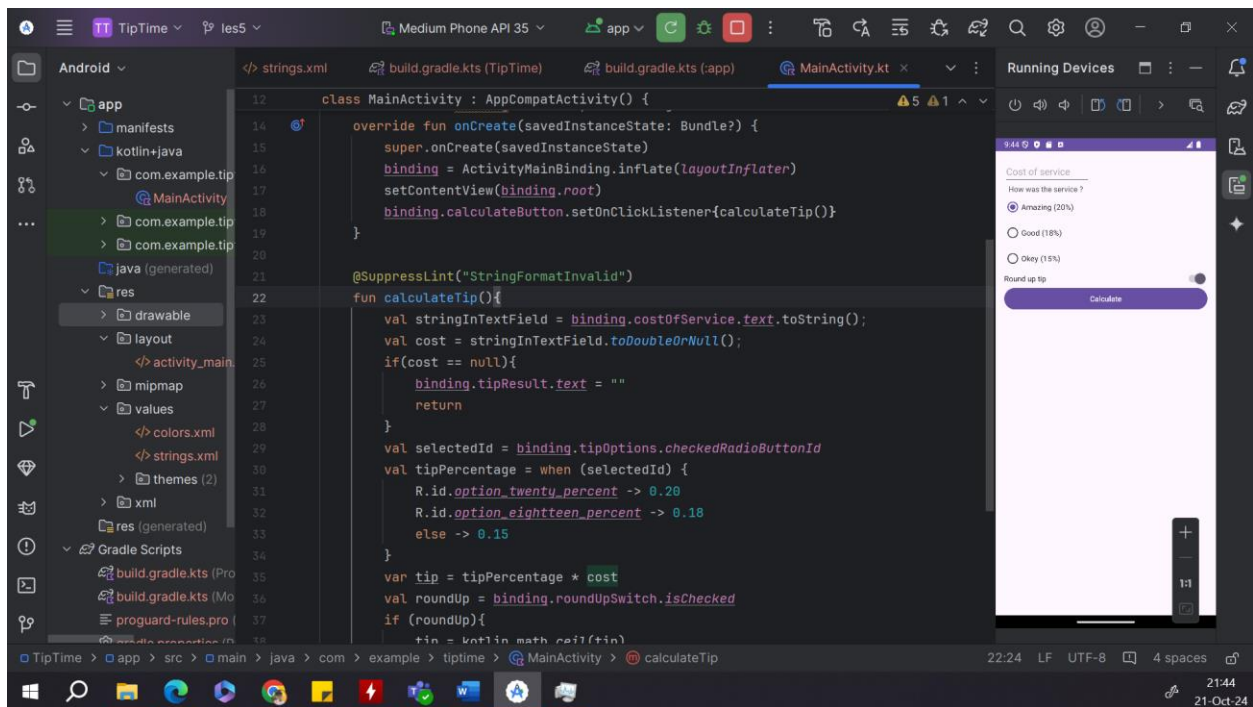
Các bước thực hiện:

Cập nhật code để bắt được 1 số ngoại lệ như khi người dùng không nhập thông tin vào TextField

Chạy chương trình:

- Khi không nhập thông tin vào ấn submit console sẽ không log ra lỗi nữa
- Chạy chương trình bình thường với luồng thành công kết quả vẫn chính xác

Kết quả thực hiện:



2.6. Adopt good coding practives

Format lại code để dễ đọc và bảo trì hơn