

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 11 – Lifecycle aware components – Tùy biến back stack với Fragment

Contents

1. Lifecycle aware components	2
1.1. Các bước thực hiện	2
1.2. Kết quả.....	3
2. Tùy biến back stack với Fragment.....	5
2.1. Các bước thực hiện	5
2.2. Kết quả.....	19

1. Lifecycle aware components

1.1. Các bước thực hiện

Tạo project Hello Wolrd như các bài đầu tiên

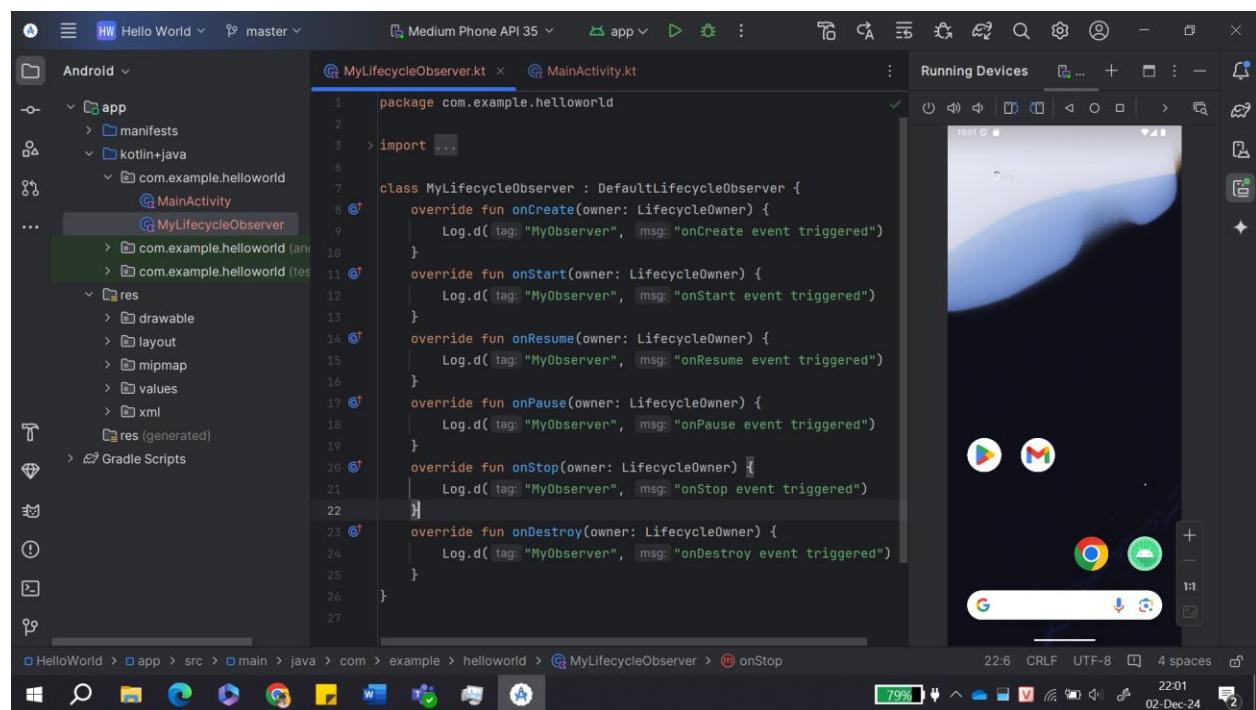
Chuyển đổi compileSDK và targetSDK thành 35 trong trường hợp bị lỗi compile (trong file build.gradle.kts)

```
android {
    namespace = "com.example.helloworld"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.helloworld"
        minSdk = 26
        targetSdk = 35
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }
}
```

Tạo một đối tượng LifecycleObserver thừa kế từ DefaultLifecycleObserver và thêm vào các câu lệnh log tương ứng khi bắt với các trạng thái khác nhau của ứng dụng



Đăng ký đối tượng MyLifecycleObserver vừa tạo với hàm chính MainActivity và thêm hàm xử lý WindowInsets:

```
package com.example.helloworld
import ...
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)

        // Đăng ký LifecycleObserver
        val observer = MyLifecycleObserver()
        lifecycle.addObserver(observer)

        // Xử lý WindowInsets
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }
    }
}
```

`ViewCompat.setOnApplyWindowInsetsListener`: Đây là phương thức giúp bạn lắng nghe sự kiện liên quan đến `WindowInsets`. Bạn có thể xử lý các thay đổi của hệ thống, chẳng hạn như khi thanh trạng thái xuất hiện hoặc khi chuyển từ portrait sang landscape.

`WindowInsetsCompat.Type.systemBars()`: Trả về thông tin về các thanh hệ thống, chẳng hạn như thanh trạng thái và thanh điều hướng.

`v.setPadding(...)`: Sau khi lấy thông tin từ `WindowInsets`, bạn có thể điều chỉnh padding của view để tránh bị che khuất bởi các thanh hệ thống.

1.2. Kết quả

Khi khởi động app ta sẽ bắt được:

The screenshot shows the Android Studio interface with the project 'Hello World' open. In the code editor, the file `MainActivity.kt` is displayed, showing the implementation of the `onCreate` method. In the Logcat tab, the output for the `onCreate` event is shown, indicating that the application has started.

```

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Xử lý WindowInsets
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets -
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.b
            insets
    }
}

```

Logcat Output:

```

2024-12-02 22:05:40.781 10830-10830 GraphicsEnvironment com.example.helloworld
2024-12-02 22:05:40.783 10830-10830 GraphicsEnvironment com.example.helloworld
2024-12-02 22:05:40.783 10830-10830 GraphicsEnvironment com.example.helloworld
2024-12-02 22:05:40.920 10830-10830 CompatChangeReporter com.example.helloworld
2024-12-02 22:05:41.326 10830-10830 AppCompatDelegate com.example.helloworld
2024-12-02 22:05:41.645 10830-10830 CompatChangeReporter com.example.helloworld
2024-12-02 22:05:41.777 10830-10830 MyObserver com.example.helloworld
2024-12-02 22:05:41.907 10830-10830 CompatChangeReporter com.example.helloworld
2024-12-02 22:05:42.087 10830-10830 MyObserver com.example.helloworld
2024-12-02 22:05:42.131 10830-10830 MyObserver com.example.helloworld
2024-12-02 22:05:42.150 10830-10830 MyObserver com.example.helloworld
2024-12-02 22:05:42.181 10830-10830 HWUI com.example.helloworld
2024-12-02 22:05:42.401 10830-10845 EGL_emulation com.example.helloworld

```

Khi thoát ra ngoài màn hình chính ta bắt được onPause và onStop:

The screenshot shows the Android Studio interface with the project 'Hello World' open. In the code editor, the file `MainActivity.kt` is displayed, showing the implementation of the `onCreate` method. In the Logcat tab, the output for the `onPause` and `onStop` events is shown, indicating that the application has been paused and stopped respectively.

```

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Xử lý WindowInsets
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets -
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.b
            insets
    }
}

```

Logcat Output:

```

2024-12-02 22:06:19.225 10830-10830 ActivityThread com.example.helloworld
2024-12-02 22:07:22.183 10830-10830 ActivityThread com.example.helloworld
2024-12-02 22:07:22.249 10830-10830 ActivityThread com.example.helloworld
2024-12-02 22:07:33.959 10830-10830 MyObserver com.example.helloworld
2024-12-02 22:07:34.351 10830-10830 VRI[MainActivity] com.example.helloworld
2024-12-02 22:07:34.382 10830-10830 MyObserver com.example.helloworld

```

Khi quay trở lại app ta bắt được onStart và onResume :

The screenshot shows the Android Studio interface. The top bar displays 'HW Hello World' and 'master'. The left sidebar shows the project structure under 'Android': app (manifests, kotlin+java, com.example.helloworld, res). In the main editor, 'MainActivity.kt' is open, showing Java code for an AppCompatActivity. The code includes an override for onCreate and logic for handling window insets. The bottom right corner of the code editor has a yellow warning icon. The bottom right panel shows a preview of the application running on a 'Medium Phone API 35 (emulator-5554)' device. The Logcat tab at the bottom shows log entries from 'package:mine' for the 'HelloWorld' application, including activity transitions and observer events. The status bar at the bottom right shows the date and time.

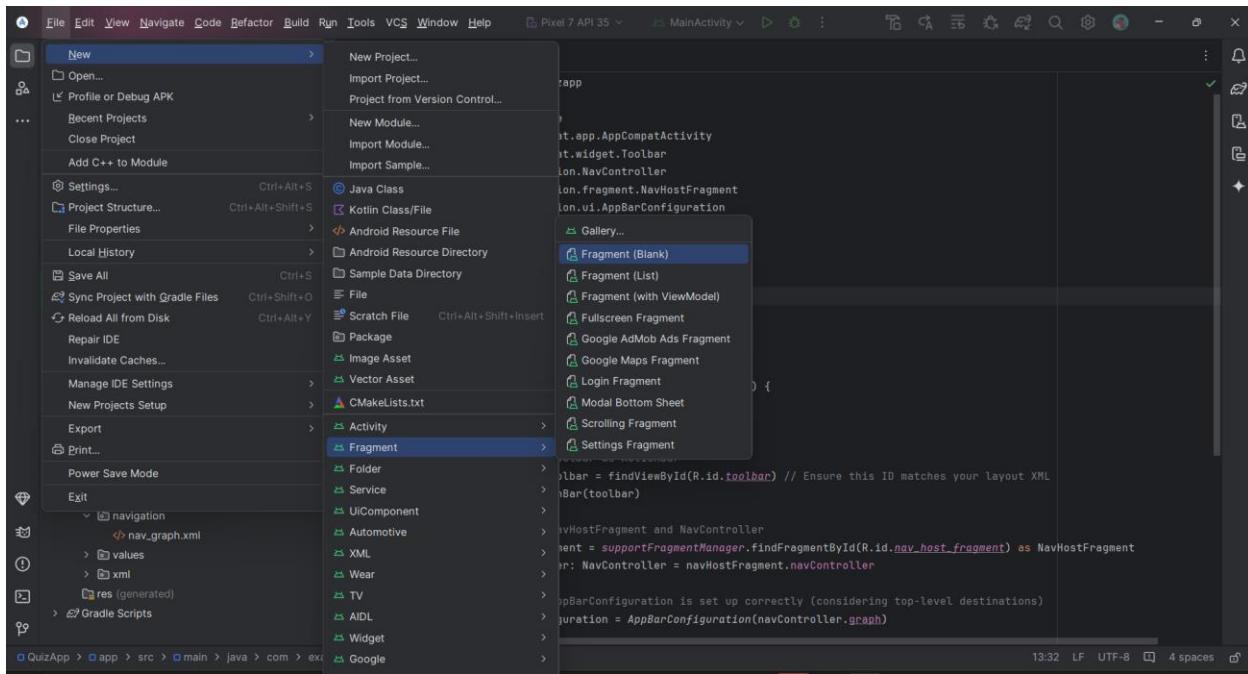
Điều này cho thấy đối tượng đã thành công quan sát được sự thay đổi trong trạng thái của ứng dụng

2. Tùy biến back stack với Fragment

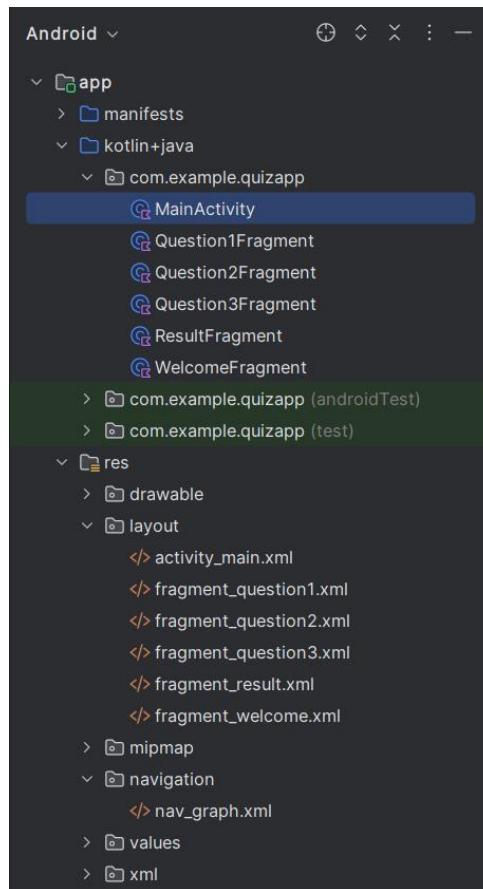
2.1. Các bước thực hiện

Tạo dự án QuizApp như những bài thực hành đầu tiên

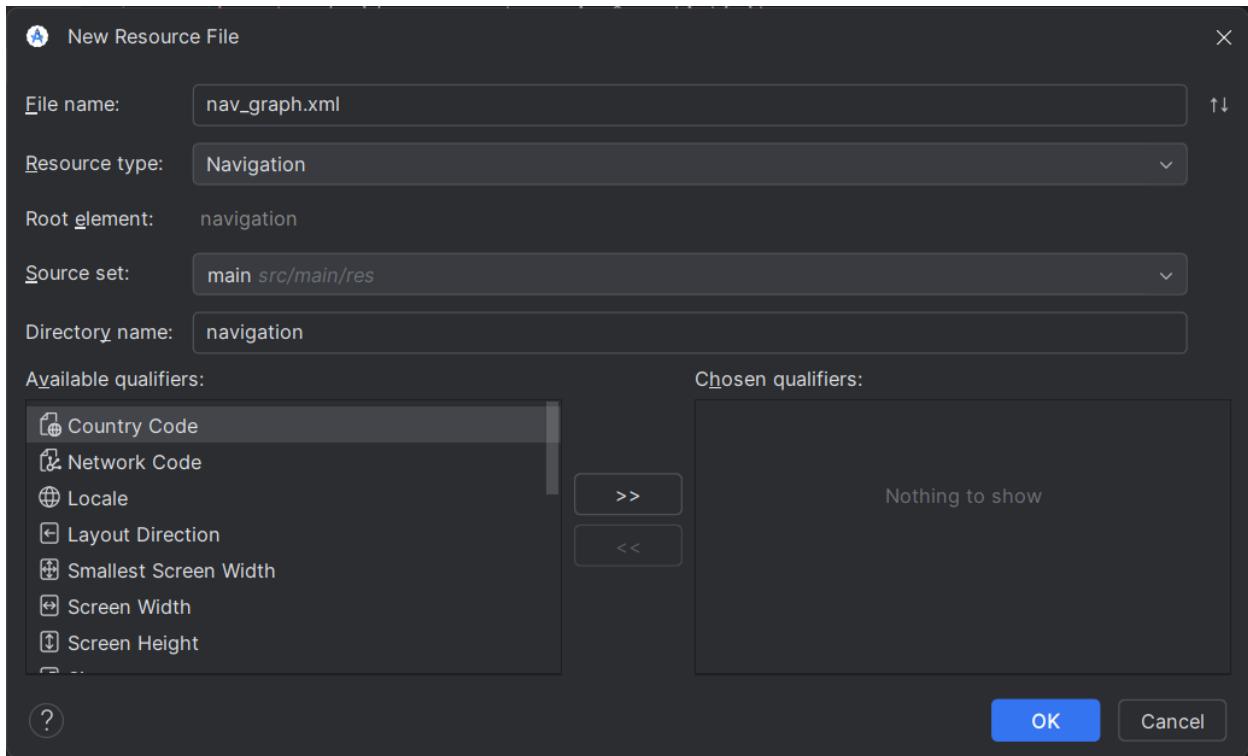
Lần lượt tạo các Fragment (bao gồm Question1Fragment, Question2Fragment, Question3Fragment, WelcomeFragment và ResultFragment)



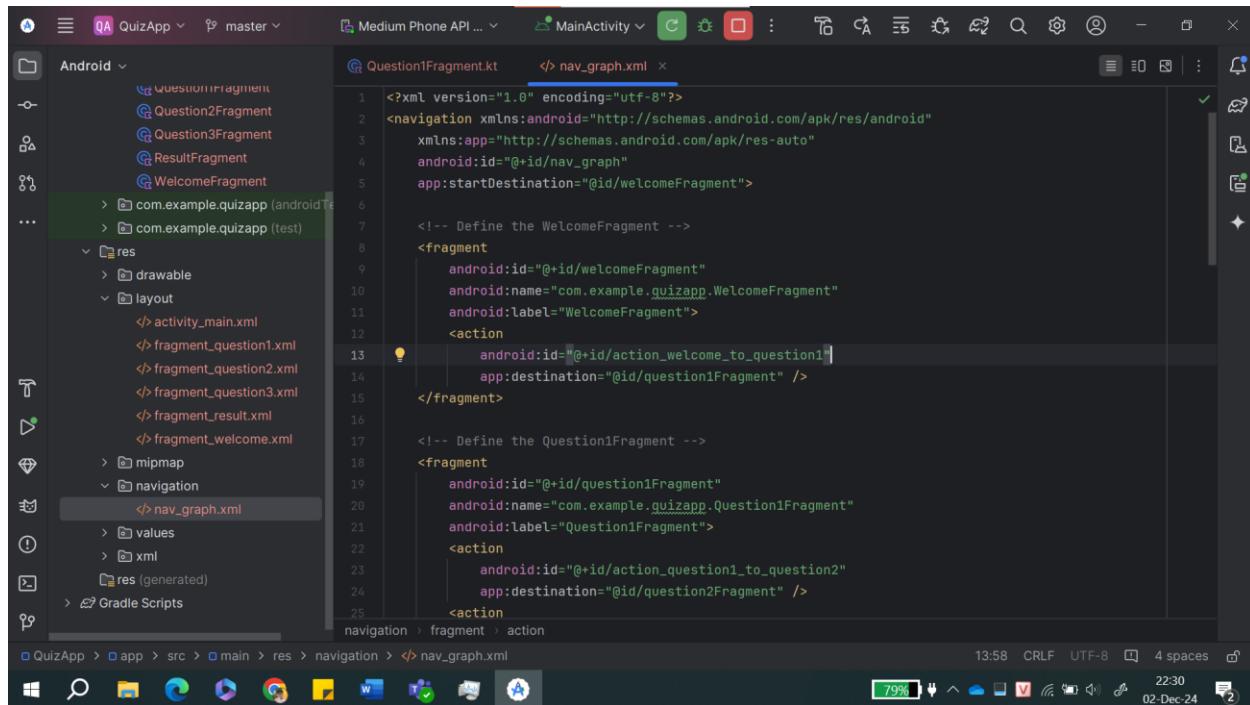
Việc tạo như trên sẽ tạo ra tổng là 10 file gồm 5 file .kt và 5 file .xml.



Tiếp theo là tạo nav_graph.xml để ứng dụng biết được các dòng chuyển đổi fragment khi ta thao tác với nút.



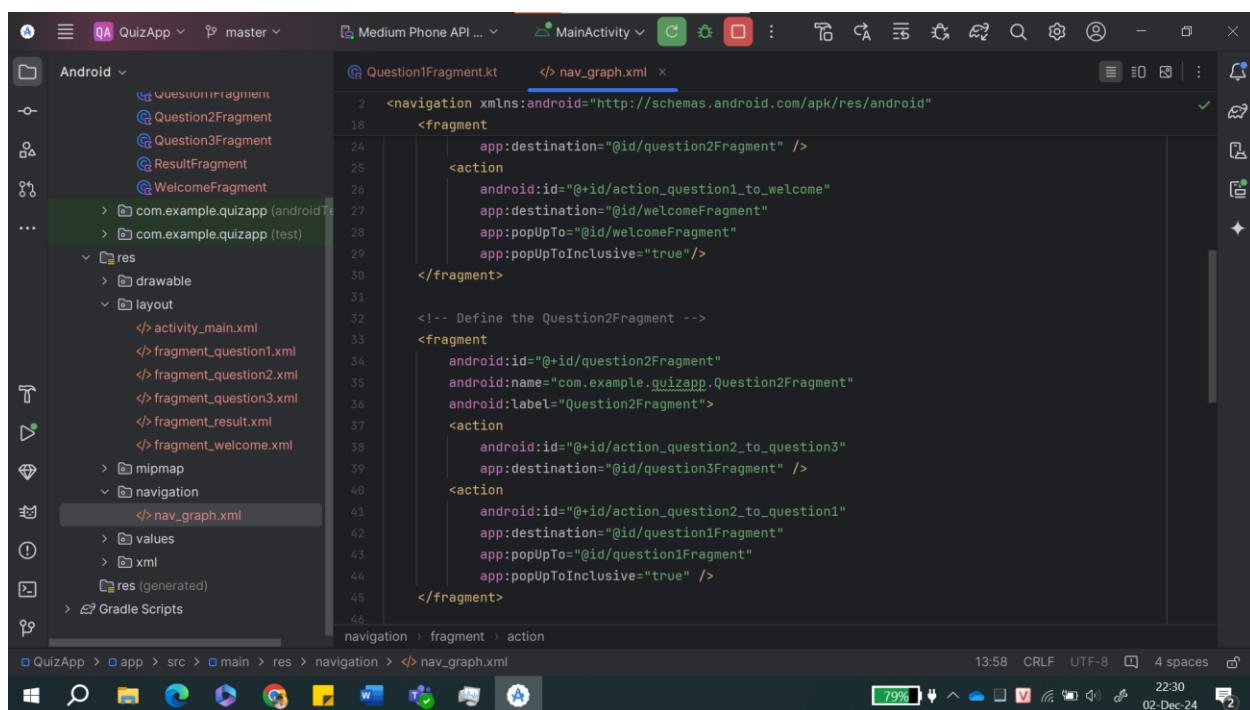
Trong nav_graph.xml ta sẽ lần lượt định nghĩa các fragment gồm id, name, nhãn của chúng cùng với các hành động chúng làm như chuyển trang tới và lui.



```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/nav_graph"
    app:startDestination="@+id/welcomeFragment">

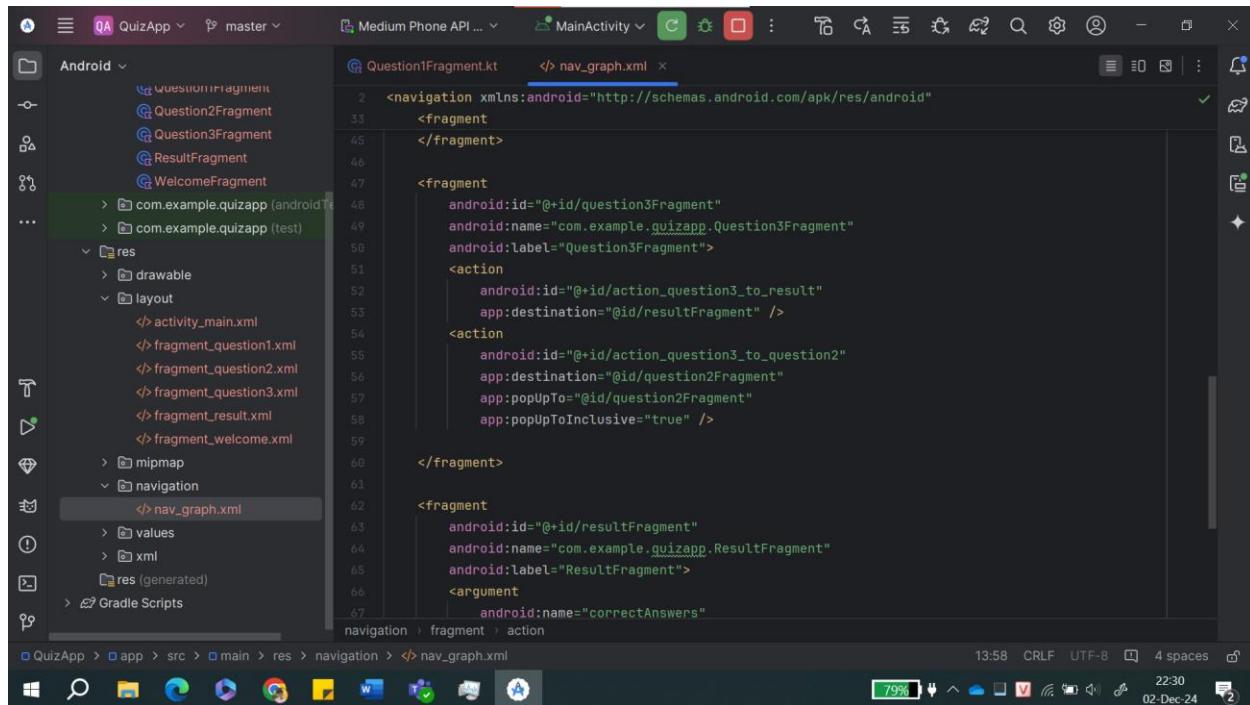
    <!-- Define the WelcomeFragment -->
    <fragment
        android:id="@+id/welcomeFragment"
        android:name="com.example.quizapp.WelcomeFragment"
        android:label="WelcomeFragment">
        <action
            android:id="@+id/action_welcome_to_question1"
            app:destination="@+id/question1Fragment" />
    </fragment>

    <!-- Define the Question1Fragment -->
    <fragment
        android:id="@+id/question1Fragment"
        android:name="com.example.quizapp.Question1Fragment"
        android:label="Question1Fragment">
        <action
            android:id="@+id/action_question1_to_question2"
            app:destination="@+id/question2Fragment" />
    </fragment>
</navigation>
```



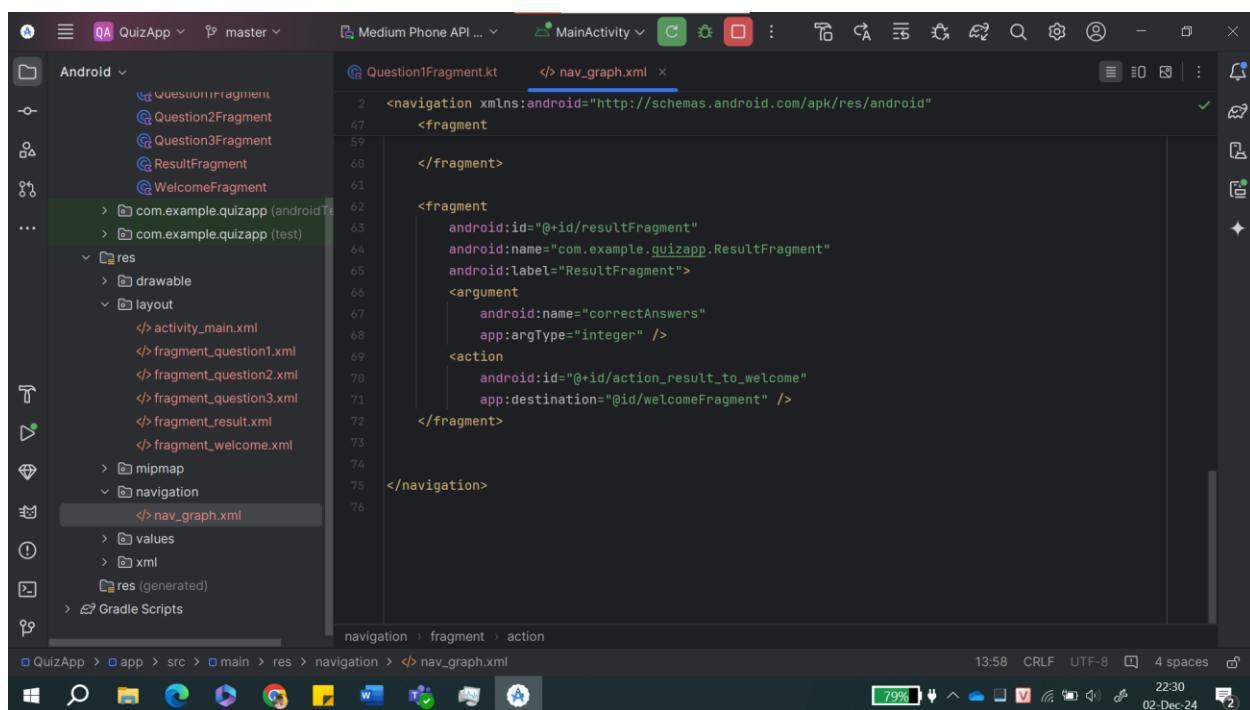
```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/nav_graph"
    <!-- Define the WelcomeFragment -->
    <fragment
        android:id="@+id/welcomeFragment"
        android:name="com.example.quizapp.WelcomeFragment"
        android:label="WelcomeFragment">
        <action
            android:id="@+id/action_welcome_to_question2"
            app:destination="@+id/question2Fragment" />
        <action
            android:id="@+id/action_question1_to_welcome"
            app:destination="@+id/welcomeFragment"
            app:popUpTo="@+id/welcomeFragment"
            app:popUpToInclusive="true"/>
    </fragment>

    <!-- Define the Question2Fragment -->
    <fragment
        android:id="@+id/question2Fragment"
        android:name="com.example.quizapp.Question2Fragment"
        android:label="Question2Fragment">
        <action
            android:id="@+id/action_question2_to_question3"
            app:destination="@+id/question3Fragment" />
        <action
            android:id="@+id/action_question2_to_question1"
            app:destination="@+id/question1Fragment"
            app:popUpTo="@+id/question1Fragment"
            app:popUpToInclusive="true" />
    </fragment>
</navigation>
```



The screenshot shows the Android Studio interface with the project 'QuizApp' open. The navigation graph XML file, 'nav_graph.xml', is displayed in the code editor. The code defines a navigation graph with three fragments: Question1Fragment, Question3Fragment, and ResultFragment. It includes actions for navigating between these fragments and arguments for passing data.

```
<navigation xmlns:android="http://schemas.android.com/apk/res/android">
    <fragment>
        <action
            android:id="@+id/question3_to_result"
            app:destination="@+id/resultFragment" />
        <action
            android:id="@+id/action_question3_to_question2"
            app:destination="@+id/question2Fragment"
            app:popUpTo="@+id/question2Fragment"
            app:popUpToInclusive="true" />
    </fragment>
    <fragment>
        <action
            android:id="@+id/result_to_welcome"
            app:destination="@+id/welcomeFragment" />
    </fragment>
</navigation>
```



The screenshot shows the same Android Studio interface after changes have been made to the navigation graph XML file. The 'ResultFragment' section has been modified to include an argument named 'correctAnswers' with type 'integer'. The 'action' section now points to the 'welcomeFragment' instead of the 'resultFragment'.

```
<navigation xmlns:android="http://schemas.android.com/apk/res/android">
    <fragment>
        <action
            android:id="@+id/question3_to_result"
            app:destination="@+id/resultFragment" />
        <action
            android:id="@+id/action_question3_to_question2"
            app:destination="@+id/question2Fragment"
            app:popUpTo="@+id/question2Fragment"
            app:popUpToInclusive="true" />
    </fragment>
    <fragment>
        <argument
            android:name="correctAnswers"
            app:type="integer" />
        <action
            android:id="@+id/action_result_to_welcome"
            app:destination="@+id/welcomeFragment" />
    </fragment>
</navigation>
```

Sau khi định ra được các luồng người dùng sẽ di chuyển ta chuyển tới lập trình giao diện (các file .xml) và xử lý logic (các file .kt)

Trước hết ta cần sửa lại vài thông tin trong build.gradle.kts

```
plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
    id("androidx.navigation.safeargs.kotlin") version "2.8.4" // Use the appropriate version
}

android {
    namespace = "com.example.quizapp"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.quizapp"
        minSdk = 26
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"
    }

    testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
}
```

Chỉnh compileSdk = 35 để tránh lỗi cũng như dùng phiên bản navigation.safeargs đúng.

```
dependencies {

    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.appcompat)
    implementation(libs.material)
    implementation(libs.androidx.activity)
    implementation(libs.androidx.constraintlayout)
    implementation(libs.androidx.navigation.fragment.ktx)
    implementation(libs.androidx.navigation.ui.ktx)
    implementation(libs.androidx.navigation.fragment.ktx.v270) // Use latest version
    implementation(libs.androidx.navigation.ui.ktx.v270)
    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
}
```

Thêm một vài dependencies cần thiết

Tiếp đến là sửa 2 file MainActivity.kt và .xml để set một FragmentContainerView chứa các Fragment của chương trình

The screenshot shows the Android Studio interface with the project 'QuizApp' open. The left sidebar displays the project structure under the 'Android' tab, specifically the 'app' module. The right panel shows the XML code for 'activity_main.xml'. The code defines a relative layout containing a toolbar and a navigation host fragment.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"

    <!-- Add the Toolbar here -->
    <androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:title="Quiz App" />

    <!-- NavHostFragment -->
    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:navGraph="@navigation/nav_graph" />

</RelativeLayout>
```

The screenshot shows the Android Studio interface with the project 'QuizApp' open. The left sidebar displays the project structure under the 'Android' tab, specifically the 'app' module. The right panel shows the Java code for 'MainActivity.kt'. The code defines the MainActivity class, which extends AppCompatActivity, and sets up the navigation host fragment.

```
package com.example.quizapp
import ...

class MainActivity : AppCompatActivity() {

    var correctAnswers: Int = 0
    var isQuestion1Correct: Int = 0
    var isQuestion2Correct: Int = 0
    var isQuestion3Correct: Int = 0

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Set up the Toolbar as ActionBar
        val toolbar: Toolbar = findViewById(R.id.toolbar) // Ensure this ID matches your layout XML
        setSupportActionBar(toolbar)

        // Set up the NavHostFragment and NavController
        val navHostFragment = supportFragmentManager.findFragmentById(R.id.nav_host_fragment) as NavHostFragment
        val navController: NavController = navHostFragment.navController

        // Ensure the AppBarConfiguration is set up correctly (considering top-level destinations)
        val appBarConfiguration = AppBarConfiguration(navController.graph)
    }
}
```

The screenshot shows the Android Studio interface with the following details:

- Title Bar:** Shows "QuizApp" and "master".
- Toolbars:** Standard Android Studio toolbars.
- Project Structure:** Shows the project tree under "Android":
 - app:** Contains "manifests", "kotlin-kt", and "com.example.quizapp". "com.example.quizapp" contains "MainActivity", "Question1Fragment", "Question2Fragment", "Question3Fragment", "ResultFragment", and "WelcomeFragment".
 - res:** Contains "drawable", "layout" (with XML files like "activity_main.xml", "fragment_question1.xml", "fragment_question2.xml", "fragment_question3.xml", "fragment_result.xml", and "fragment_welcome.xml"), "mipmap", and "navigation".
- Main Activity Editor:** Displays the code for "MainActivity.kt". The code sets up a NavController and ActionBar, and handles the Up/Back button.
- Status Bar:** Shows "11:17 LF UTF-8 4 spaces", "79%", and the date "02-Dec-24".

Sửa thêm một chút trong file .kt là đặt các biến để lưu kết quả cho mỗi câu hỏi tối đa là 1 và tối thiểu là 0

Giờ ta sẽ đến Welcome Fragment, Welcome Fragment chỉ đơn giản là một giao diện chào người dùng với nút bắt đầu tham gia quiz nên không có quá nhiều xử lý logic, chỉ cần đặt lại các điểm số về 0 trước khi thực hiện quiz mới

The screenshot shows the Android Studio interface with the code editor open to the `WelcomeFragment.kt` file. The code defines a fragment that inflates a layout and uses NavController to navigate to another fragment.

```
1 package com.example.quizapp
2
3 import ...
4
5 class WelcomeFragment : Fragment() {
6
7     override fun onCreateView(
8         inflater: LayoutInflater, container: ViewGroup?,
9         savedInstanceState: Bundle?
10    ): View? {
11         val view = inflater.inflate(R.layout.fragment_welcome, container, false)
12
13         // Use NavController to navigate to Question1Fragment on "Start Quiz"
14         view.findViewById<Button>(R.id.start_quiz_button).setOnClickListener {
15             findNavController().navigate(R.id.action_welcome_to_question1) // Use the action from the navigation graph
16         }
17
18         return view
19     }
20
21 }
```

The screenshot shows the Android Studio interface with the XML layout editor open to the `fragment_welcome.xml` file. It defines a ConstraintLayout containing a TextView and a Button.

```
<?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=".WelcomeFragment">

    <TextView
        android:id="@+id/welcome_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="120dp"
        android:text="Welcome to the Quiz App"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/start_quiz_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="52dp"
        android:text="Start Quiz" />

```

```

<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Button
        android:id="@+id/start_quiz_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="52dp"
        android:text="Start Quiz"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="56dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/welcome_text"
        tools:src="@tools:sample/avatars[9]" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Tiếp theo đến các Fragment câu hỏi, các fragment này khá tương tự nhau đều bao gồm đặt đúng sai cho các câu hỏi, xử lý logic nút submit và back cũng như tính điểm

```

package com.example.quizapp
import ...
class Question1Fragment : Fragment() {
    private var selectedAnswer: Int? = null // Lưu trữ lựa chọn của người dùng
    private var isCorrectAnswerSelected = false // Cờ để theo dõi nếu đáp án đúng được chọn

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val view = inflater.inflate(R.layout.fragment_question1, container, false)

        // Xử lý khi chọn đáp án
        view.findViewById(R.id.option_1_button).setOnClickListener {
            selectAnswer(answer: 1, isCorrect: false) // Đáp án sai
        }

        view.findViewById(R.id.option_2_button).setOnClickListener {
            selectAnswer(answer: 2, isCorrect: true) // Đáp án đúng
        }

        view.findViewById(R.id.option_3_button).setOnClickListener {
            selectAnswer(answer: 3, isCorrect: false) // Đáp án sai
        }
    }
}

```

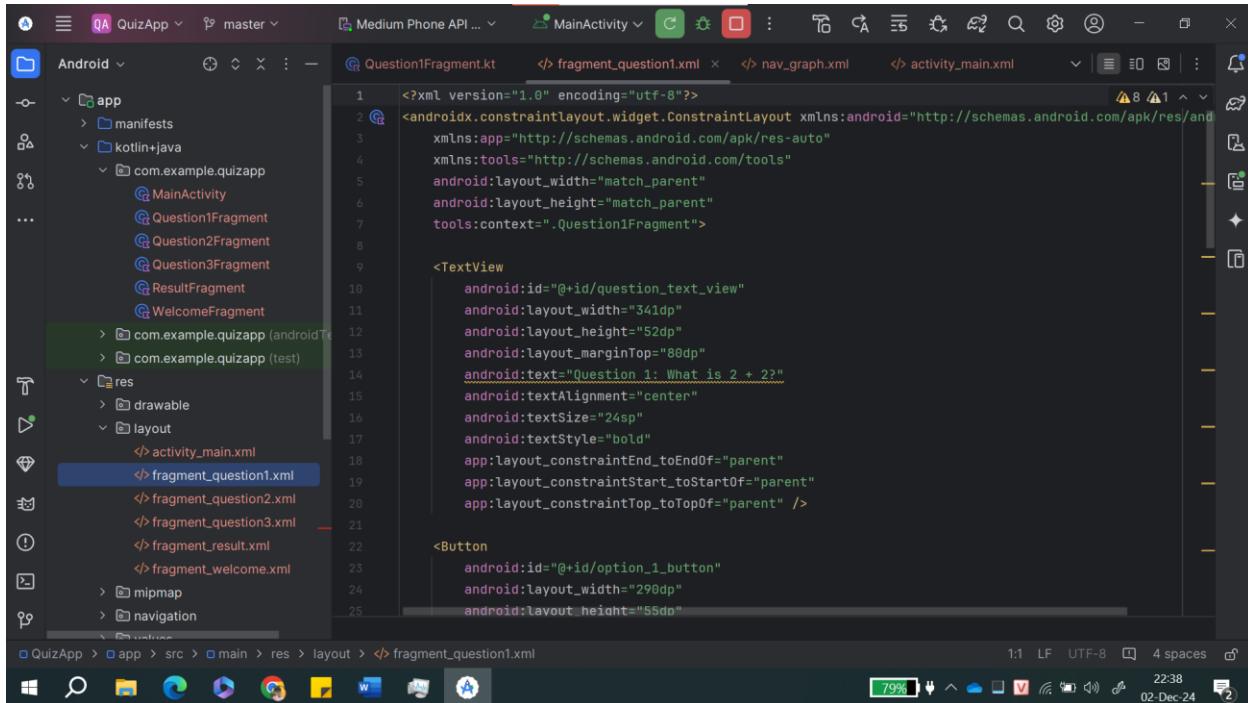
The screenshot shows the Android Studio interface with the Question1Fragment.kt file open. The code handles button click events and navigation between fragments.

```
12     class Question1Fragment : Fragment() {
13         override fun onCreateView(
14             inflater: LayoutInflater,
15             container: ViewGroup?,
16             savedInstanceState: Bundle?
17         ): View? {
18             val view = inflater.inflate(R.layout.fragment_question1, container, false)
19
20             // Xử lý nút Submit
21             view.findViewById(R.id.option_4_button).setOnClickListener {
22                 selectAnswer( answer: 4, isCorrect: false ) // Đáp án sai
23             }
24
25             // Xử lý nút Back
26             view.findViewById(R.id.back_button).setOnClickListener {
27                 findNavController().navigateUp() // Quay lại trang trước
28             }
29
30             return view
31         }
32
33         private fun selectAnswer(answer: Int, isCorrect: Boolean) {
34             selectedAnswer = answer // Lưu trữ đáp án được chọn
35             isCorrectAnswerSelected = isCorrect // Cập nhật trạng thái nếu đáp án đúng
36             Toast.makeText(context, text: "Bạn đã chọn đáp án $answer", Toast.LENGTH_SHORT).show()
37         }
38
39         override fun onResume() {
40             super.onResume()
41             // Đặt lại biến isQuestion1Correct về 0 mỗi khi truy cập Question1Fragment
42             (activity as? MainActivity)?.isQuestion1Correct = 0
43         }
44
45         private fun navigateToNextQuestion() {
46             val mainActivity = activity as MainActivity
47
48             if (isCorrectAnswerSelected) {
49                 mainActivity.isQuestion1Correct = 1
50             }
51
52             findNavController().navigate(R.id.action_question1_to_question2)
53         }
54     }
55
56     private fun selectAnswer(answer: Int, isCorrect: Boolean) {
57         selectedAnswer = answer // Lưu trữ đáp án được chọn
58         isCorrectAnswerSelected = isCorrect // Cập nhật trạng thái nếu đáp án đúng
59         Toast.makeText(context, text: "Bạn đã chọn đáp án $answer", Toast.LENGTH_SHORT).show()
60     }
61
62     override fun onResume() {
63         super.onResume()
64         // Đặt lại biến isQuestion1Correct về 0 mỗi khi truy cập Question1Fragment
65         (activity as? MainActivity)?.isQuestion1Correct = 0
66     }
67
68     private fun navigateToNextQuestion() {
69         val mainActivity = activity as MainActivity
70
71         if (isCorrectAnswerSelected) {
72             mainActivity.isQuestion1Correct = 1
73         }
74
75         findNavController().navigate(R.id.action_question1_to_question2)
76     }
77 }
```

The screenshot shows the Android Studio interface with the Question1Fragment.kt file open. The code has been modified to handle the selection of the fourth option and update the main activity's state.

```
12     class Question1Fragment : Fragment() {
13
14         private fun selectAnswer(answer: Int, isCorrect: Boolean) {
15             selectedAnswer = answer // Lưu trữ đáp án được chọn
16             isCorrectAnswerSelected = isCorrect // Cập nhật trạng thái nếu đáp án đúng
17             Toast.makeText(context, text: "Bạn đã chọn đáp án $answer", Toast.LENGTH_SHORT).show()
18         }
19
20         override fun onCreateView(
21             inflater: LayoutInflater,
22             container: ViewGroup?,
23             savedInstanceState: Bundle?
24         ): View? {
25             val view = inflater.inflate(R.layout.fragment_question1, container, false)
26
27             // Xử lý nút Submit
28             view.findViewById(R.id.option_4_button).setOnClickListener {
29                 selectAnswer( answer: 4, isCorrect: false ) // Đáp án sai
30             }
31
32             // Xử lý nút Back
33             view.findViewById(R.id.back_button).setOnClickListener {
34                 findNavController().navigateUp() // Quay lại trang trước
35             }
36
37             return view
38         }
39
40         override fun onResume() {
41             super.onResume()
42             // Đặt lại biến isQuestion1Correct về 0 mỗi khi truy cập Question1Fragment
43             (activity as? MainActivity)?.isQuestion1Correct = 0
44         }
45
46         private fun navigateToNextQuestion() {
47             val mainActivity = activity as MainActivity
48
49             if (isCorrectAnswerSelected) {
50                 mainActivity.isQuestion1Correct = 1
51             }
52
53             findNavController().navigate(R.id.action_question1_to_question2)
54         }
55     }
56
57     private fun selectAnswer(answer: Int, isCorrect: Boolean) {
58         selectedAnswer = answer // Lưu trữ đáp án được chọn
59         isCorrectAnswerSelected = isCorrect // Cập nhật trạng thái nếu đáp án đúng
60         Toast.makeText(context, text: "Bạn đã chọn đáp án $answer", Toast.LENGTH_SHORT).show()
61     }
62
63     override fun onResume() {
64         super.onResume()
65         // Đặt lại biến isQuestion1Correct về 0 mỗi khi truy cập Question1Fragment
66         (activity as? MainActivity)?.isQuestion1Correct = 0
67     }
68
69     private fun navigateToNextQuestion() {
70         val mainActivity = activity as MainActivity
71
72         if (isCorrectAnswerSelected) {
73             mainActivity.isQuestion1Correct = 1
74         }
75
76         findNavController().navigate(R.id.action_question1_to_question2)
77     }
78 }
```

File layout của chúng sẽ có cùng một dạng như sau:



QuizApp

MainActivity

Question1Fragment.kt

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Question1Fragment">

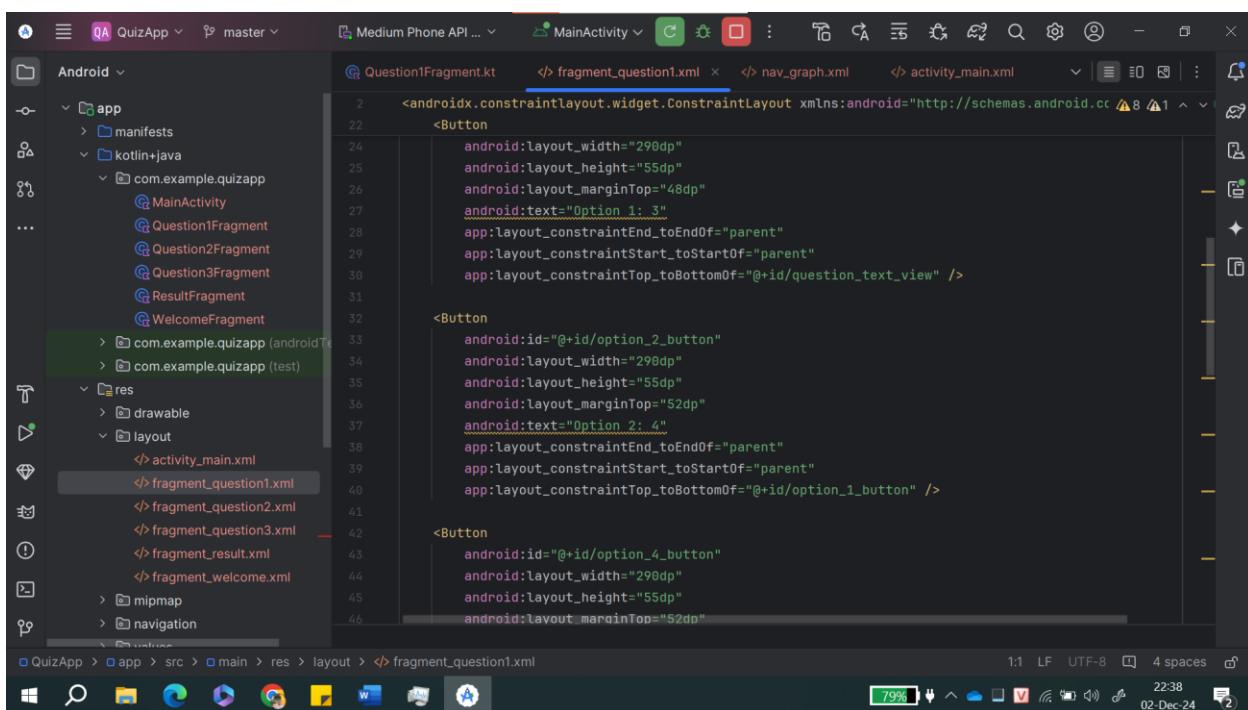
    <TextView
        android:id="@+id/question_text_view"
        android:layout_width="341dp"
        android:layout_height="52dp"
        android:layout_marginTop="80dp"
        android:text="Question 1: What is 2 + 2?"
        android:textAlignment="center"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/option_1_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="8dp"
        android:text="Option 1: 3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/question_text_view" />

    <Button
        android:id="@+id/option_2_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="52dp"
        android:text="Option 2: 4"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/option_1_button" />

    <Button
        android:id="@+id/option_3_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="52dp"
        android:text="Option 3: 5"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/option_2_button" />

    <Button
        android:id="@+id/option_4_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="52dp"
        android:text="Option 4: 6"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/option_3_button" />
```



QuizApp

MainActivity

Question1Fragment.kt

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Question1Fragment">

    <TextView
        android:id="@+id/question_text_view"
        android:layout_width="341dp"
        android:layout_height="52dp"
        android:layout_marginTop="80dp"
        android:text="Question 1: What is 2 + 2?"
        android:textAlignment="center"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/option_1_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="8dp"
        android:text="Option 1: 3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/question_text_view" />

    <Button
        android:id="@+id/option_2_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="52dp"
        android:text="Option 2: 4"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/option_1_button" />

    <Button
        android:id="@+id/option_3_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="52dp"
        android:text="Option 3: 5"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/option_2_button" />

    <Button
        android:id="@+id/option_4_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="52dp"
        android:text="Option 4: 6"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/option_3_button" />
```

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:id="@+id/option_3_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="52dp"
        android:text="Option 4: 5"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/option_3_button" />
    <Button
        android:id="@+id/option_3_button"
        android:layout_width="290dp"
        android:layout_height="55dp"
        android:layout_marginTop="52dp"
        android:text="Option 3: 1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/option_2_button" />
    <Button
        android:id="@+id/back_button"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="40dp"
        android:text="Back"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/submit_button"
        app:layout_constraintHorizontal_bias="0.443"
        app:layout_constraintStart_toStartOf="parent" />
```

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:id="@+id/back_button"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="40dp"
        android:text="Back"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toStartOf="@+id/submit_button"
        app:layout_constraintHorizontal_bias="0.443"
        app:layout_constraintStart_toStartOf="parent" />
    <Button
        android:id="@+id/submit_button"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="32dp"
        android:layout_marginBottom="40dp"
        android:text="Submit"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />
```

Cuối cùng ta sẽ hiện kết quả qua ResultFragment:

The screenshot shows the Android Studio interface with the ResultFragment.kt file open in the editor. The code implements a Fragment that calculates the total score from three questions and displays it. It also handles the back button to return to the main activity and the exit button to finish the application.

```
package com.example.quizapp
import ...
class ResultFragment : Fragment() {
    private var correctAnswers: Int = 0
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        val view = inflater.inflate(R.layout.fragment_result, container, false)
        // Lấy tổng điểm từ MainActivity
        val MainActivity = activity as MainActivity
        correctAnswers = MainActivity.isQuestion1Correct + MainActivity.isQuestion2Correct + MainActivity.isQuestion3Correct
        // Hiển thị kết quả
        val resultTextView = view.findViewById<TextView>(R.id.result_text_view)
        resultTextView.text = "You got $correctAnswers correct answers!"
        // Xử lý nút "Back to Home"
        view.findViewById<Button>(R.id.back_to_home_button).setOnClickListener {
            // Đặt lại điểm số về 0 trong MainActivity
            MainActivity.correctAnswers = 0
        }
    }
}
```

The screenshot shows the modified ResultFragment.kt file in Android Studio. The code now includes logic to handle the back button by navigating to the WelcomeFragment and the exit button by finishing the activity.

```
class ResultFragment : Fragment() {
    override fun onCreateView(
        view: View
    ) {
        view.findViewById<Button>(R.id.back_to_home_button).setOnClickListener {
            // Đặt lại điểm số về 0 trong MainActivity
            MainActivity.correctAnswers = 0
            // Điều hướng về WelcomeFragment
            findNavController().navigate(R.id.action_result_to_welcome)
        }
        view.findViewById<Button>(R.id.exit_button).setOnClickListener {
            // Đóng ứng dụng
            activity?.finish()
            System.exit(0)
        }
        return view
    }
}
```

File trên sẽ xử lý tính toán số câu trả lời đúng và cộng tổng xong hiển thị ra màn hình:

The screenshot shows the Android Studio interface with the code editor open to the `fragment_result.xml` file. The code defines a linear layout containing a text view and two buttons. The text view displays a message about correct answers. The first button is labeled "Back to Home" and the second is labeled "Exit". The XML code is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp">

    <TextView
        android:id="@+id/result_text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="You got 0 correct answers!"
        android:textSize="20sp"
        android:layout_marginBottom="32dp" />

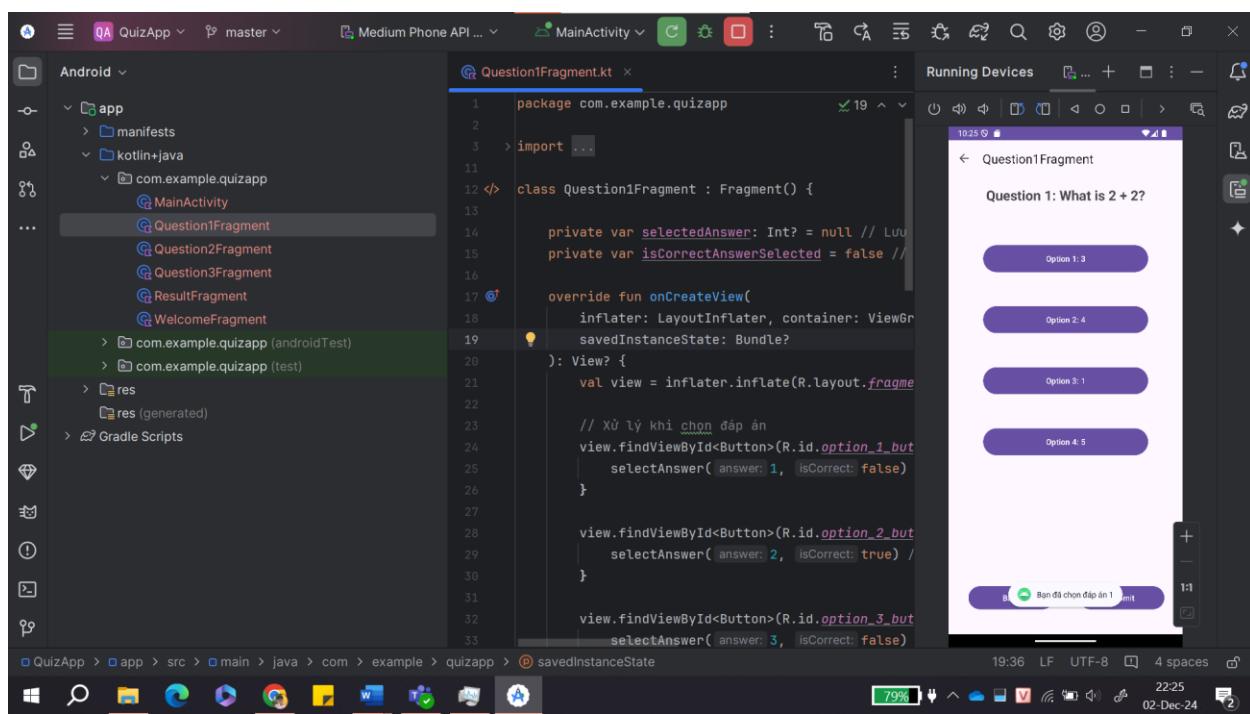
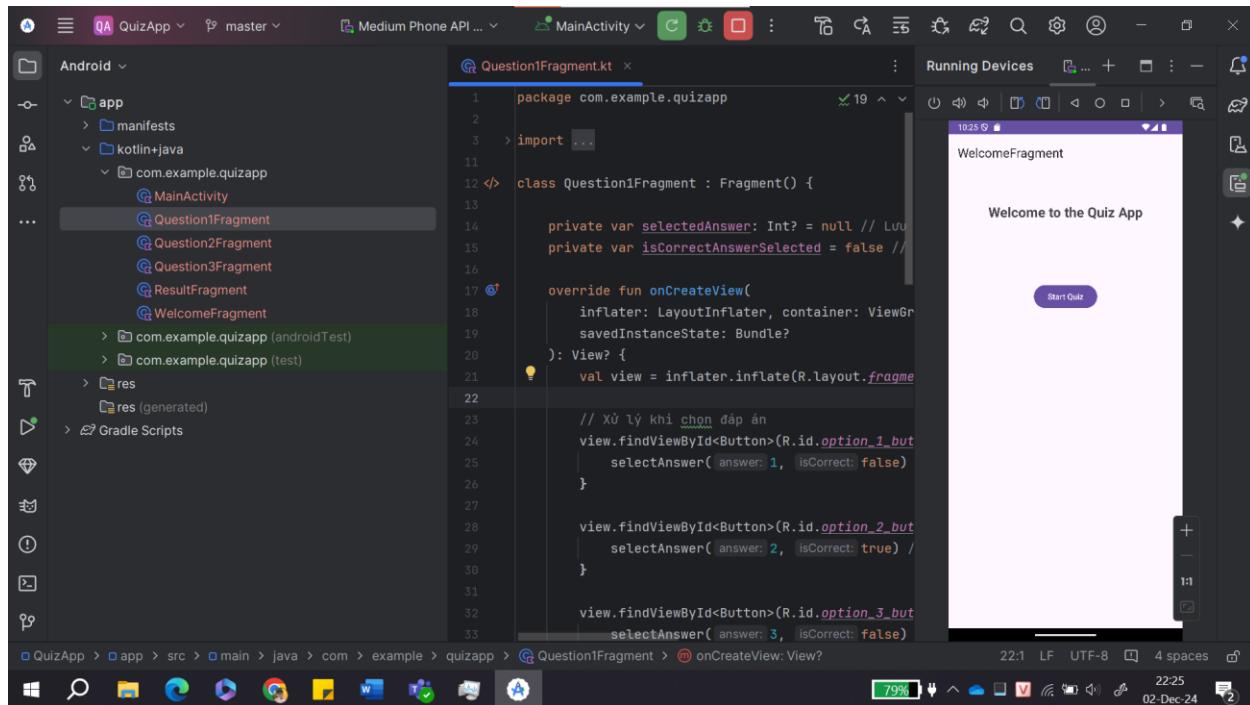
    <Button
        android:id="@+id/back_to_home_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Back to Home"
        android:layout_marginBottom="16dp" />

    <Button
        android:id="@+id/exit_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Exit" />

</LinearLayout>
```

This screenshot is similar to the one above, but the cursor is positioned at the end of the final closing tag, `</LinearLayout>`. The rest of the XML code is identical to the previous screenshot.

2.2. Kết quả



The screenshot shows the Android Studio interface. The left sidebar displays the project structure under the 'Android' tab, with the 'app' module selected. The main editor window shows the code for `Question1Fragment.kt`. The code initializes variables for selected answer and incorrect answer selection, then handles button clicks to select answers and update the UI. The right panel shows a running device with the application's user interface. A question is displayed: "Question 2: What is the capital of France?". Four options are listed: Option 1: London, Option 2: Paris, Option 3: Madrid, and Option 4: Berlin. A message at the bottom of the screen says "Bạn đã chọn đáp án 2". The status bar at the bottom indicates the time is 22:26 and the date is 02-Dec-24.

```
package com.example.quizapp
import ...
class Question1Fragment : Fragment() {
    private var selectedAnswer: Int? = null // Lưu
    private var isCorrectAnswerSelected = false // 
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup,
        savedInstanceState: Bundle?
    ): View? {
        val view = inflater.inflate(R.layout.fragment_question1, container, false)
        // Xử lý khi chọn đáp án
        view.findViewById<Button>(R.id.option_1_button).setOnClickListener {
            selectAnswer(answer: 1, isCorrect: false)
        }
        view.findViewById<Button>(R.id.option_2_button).setOnClickListener {
            selectAnswer(answer: 2, isCorrect: true)
        }
        view.findViewById<Button>(R.id.option_3_button).setOnClickListener {
            selectAnswer(answer: 3, isCorrect: false)
        }
    }
}
```

The screenshot shows the Android Studio interface. The left sidebar displays the project structure under the 'Android' tab, with the 'app' module selected. The main editor window shows the code for `Question1Fragment.kt`, which is identical to the previous screenshot. The right panel shows a running device with the application's user interface. A question is displayed: "Question 3: What is the square root of 16?". Four options are listed: Option 1: 2, Option 2: 4, Option 3: 5, and Option 4: 8. A message at the bottom of the screen says "Bạn đã chọn đáp án 2". The status bar at the bottom indicates the time is 22:26 and the date is 02-Dec-24.

```
package com.example.quizapp
import ...
class Question1Fragment : Fragment() {
    private var selectedAnswer: Int? = null // Lưu
    private var isCorrectAnswerSelected = false // 
    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup,
        savedInstanceState: Bundle?
    ): View? {
        val view = inflater.inflate(R.layout.fragment_question1, container, false)
        // Xử lý khi chọn đáp án
        view.findViewById<Button>(R.id.option_1_button).setOnClickListener {
            selectAnswer(answer: 1, isCorrect: false)
        }
        view.findViewById<Button>(R.id.option_2_button).setOnClickListener {
            selectAnswer(answer: 2, isCorrect: true)
        }
        view.findViewById<Button>(R.id.option_3_button).setOnClickListener {
            selectAnswer(answer: 3, isCorrect: false)
        }
    }
}
```

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "Android". The "app" module contains "manifests", "kotlin+java", and "res" (generated). The "com.example.quizapp" package contains "MainActivity", "Question1Fragment" (selected), "Question2Fragment", "Question3Fragment", "ResultFragment", and "WelcomeFragment".
- Code Editor:** The main editor window displays the "Question1Fragment.kt" file. The code is as follows:

```
1 package com.example.quizapp
2
3 import ...
4
5 class Question1Fragment : Fragment() {
6
7     private var selectedAnswer: Int? = null // Lưu
8     private var isCorrectAnswerSelected = false // 
9
10    override fun onCreateView(
11        inflater: LayoutInflater, container: ViewGroup,
12        savedInstanceState: Bundle?
13    ): View? {
14        val view = inflater.inflate(R.layout.fragment_question1, container, false)
15
16        // Xử lý khi chọn đáp án
17        view.findViewById<Button>(R.id.option_1_button).setOnClickListener {
18            selectAnswer( answer: 1, isCorrect: false )
19        }
20
21        view.findViewById<Button>(R.id.option_2_button).setOnClickListener {
22            selectAnswer( answer: 2, isCorrect: true ) />
23        }
24
25        view.findViewById<Button>(R.id.option_3_button).setOnClickListener {
26            selectAnswer( answer: 3, isCorrect: false )
27        }
28
29    }
30
31    private fun selectAnswer( answer: Int, isCorrect: Boolean ) {
32        if (isCorrect) {
33            ...
34        }
35    }
36}
37
38
```

- Running Device Preview:** A floating window titled "ResultFragment" shows the results of a quiz. It displays the message "You got 2 correct answers!" and two buttons: "Back to Home" and "Exit".
- Bottom Bar:** The bottom bar shows the file path "QuizApp > app > src > main > java > com > example > quizapp > Question1Fragment > onCreateView: View?", battery level at 79%, and the date/time "02-Dec-24 22:26".