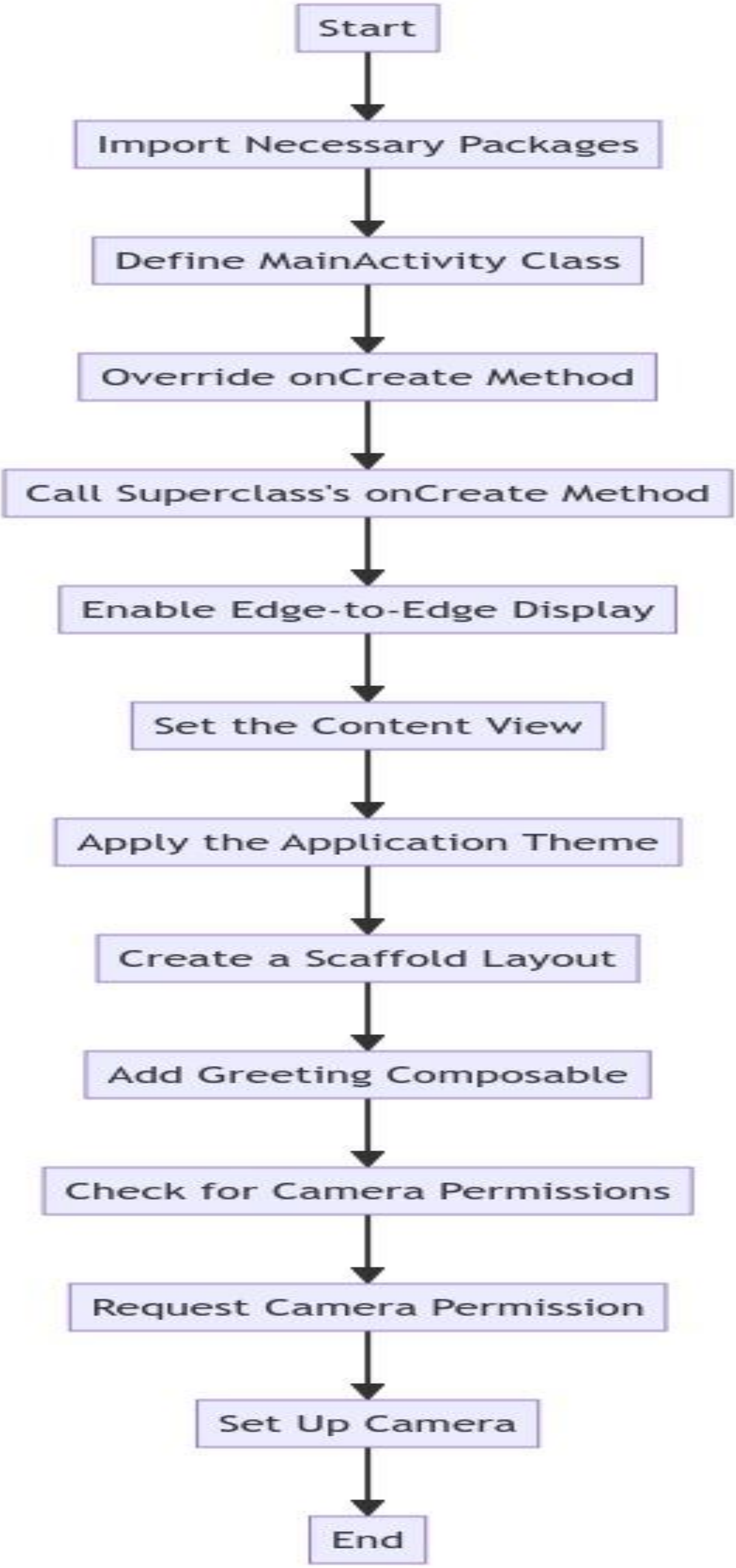
Creating a Flowchart after Integrated With Camera Service

Name of Testers:

1. Anupam
2. Asha
3. Harshvardhan
4. Hrushikesh
5. Manish
6. Krishna
7. Tahir


Start
Import Necessary Packages
Define MainActivity Class
Override onCreate Method
Call Superclass's onCreate Method
Enable Edge-to-Edge Display
Set the Content View
Apply the Application Theme
Create a Scaffold Layout
Add Greeting Composable
Check for Camera Permissions
Request Camera Permission
Set Up Camera
End

Introduction:

This process outlines the steps involved in setting up an Android application using Jetpack Compose, a modern toolkit for building native UIs in Android. Here's a detailed explanation of each step:

1. Start:

The process begins.

2. Import Necessary Packages:

This step involves importing required packages from the AndroidX library. AndroidX is a major improvement to the original Android Support Library, containing several libraries that provide backward-compatible versions of new features and APIs.

3. Define MainActivity Class:

A new class named MainActivity is declared. It inherits from ComponentActivity, which is a base class for activities that use Jetpack Compose.

4. Override onCreate Method:

The onCreate method is defined within the MainActivity class. This method is a lifecycle callback invoked when the activity is being created.

5. Call Superclass's onCreate Method:

Within the onCreate method of MainActivity, the onCreate method of the superclass (ComponentActivity) is called. This ensures that the superclass's initialization logic is executed properly.

6. Enable Edge-to-Edge Display:

This step involves customizing the user interface to use the full screen space, also known as edge-to-edge display. This can enhance the user experience by maximizing the available screen real estate.

7. Set the Content View:

The setContent method is used to define the UI layout using a composable function. Composables are the building blocks of Jetpack Compose UIs, allowing developers to create UI components declaratively.

8. Apply the Application Theme:

The app's theme is applied to the content. Themes define the visual style of an app, including colors, typography, and other stylistic elements.

9. Create a Scaffold Layout:

A basic material design layout structure is set up using the Scaffold composable. The Scaffold provides a layout structure that includes app bars, floating action buttons, and other common UI components.

10. Add Greeting Composable:

A Greeting composable is added, which likely displays a greeting message to the user. The composable may accept parameters such as the user's name and modifier to customize its appearance.

11. Check for Camera Permissions:

The app checks if the camera permission is granted. This is essential if the app intends to access the device's camera for any functionality.

12. Request Camera Permission:

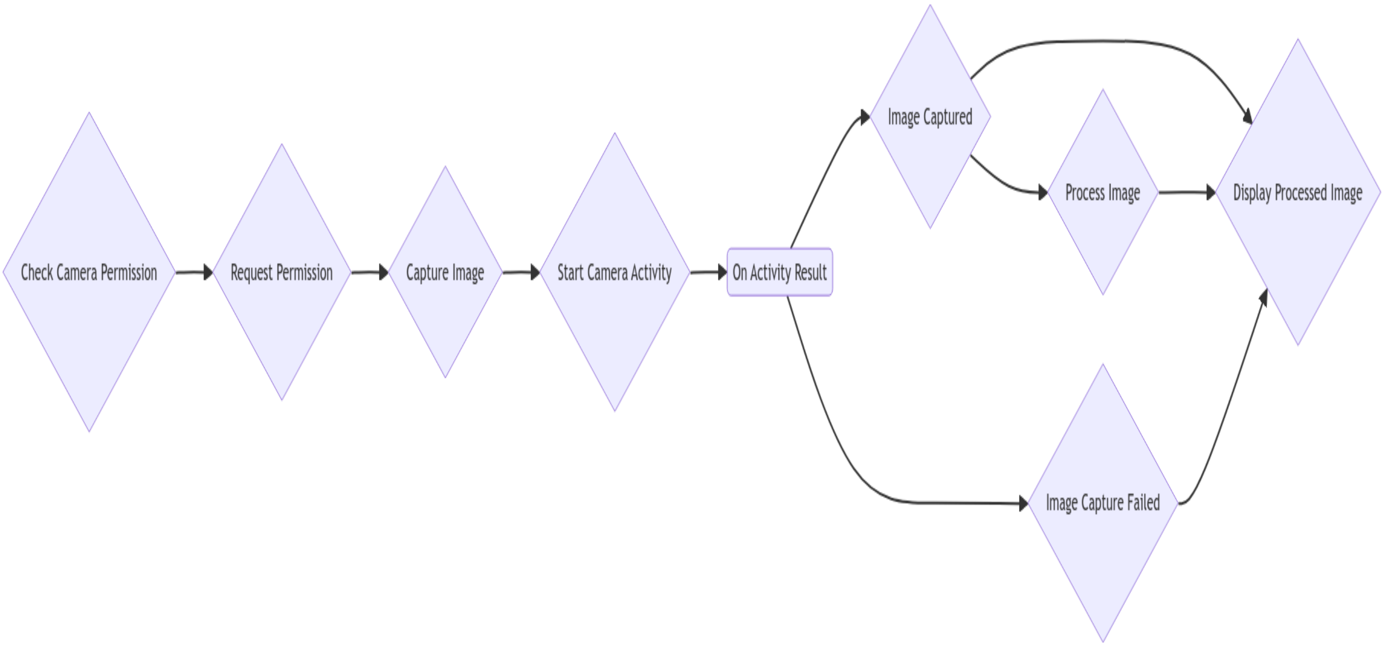
If the camera permission is not granted, the app sets up a permission request launcher and handles the result. This ensures that the app requests permission from the user dynamically when needed.

13. Set Up Camera:

Finally, the camera provider is initialized, and the camera is bound to the lifecycle. This step ensures that the camera functionality is properly set up and managed throughout the app's lifecycle.

14. End: The process concludes:

Overall, this process outlines the initialization and setup steps required to create an Android app using Jetpack Compose, including defining the activity, setting up the UI, managing permissions, and integrating camera functionality.



THANK YOU