Department of Information Technology

Academic Year: 2024-25
Semester: VI
Class / Branch / Div: TE- IT/C
Subject: MAD & PWA Lab
Name of Instructor:Prof. Sujata Oak

Name of Student:
Student ID:
Roll No.
Date of Submission:

Experiment No.:5

Aim: To apply navigation and routing in Flutter App.

Theory:

Flutter Navigation and Routing

Navigation and routing are some of the core concepts of all mobile application, which allows the user to move between different pages. We know that every mobile application contains several screens for displaying different types of information. **For example,** an app can have a screen that contains various products. When the user taps on that product, immediately it will display detailed information about that product.

In Flutter, the screens and pages are known as **routes,** and these routes are just a widget. In Android, a route is similar to an **Activity,** whereas, in iOS, it is equivalent to a **ViewController.**

In any mobile app, navigating to different pages defines the workflow of the application, and the way to handle the navigation is known as **routing.** Flutter provides a basic routing class **MaterialPageRoute** and two methods **Navigator.push()** and **Navigator.pop()** that shows how to navigate between two routes. The following steps are required to start navigation in your application.

Navigator: As the name suggests, Navigator is a widget that helps us to navigate between the routes. The navigator follows stack method when dealing with the routes. Based on the actions made by the user, the routes are stacked one over the other and when pressed back, it goes to the most recently visited route. Navigator is a widget that follows a stack discipline.

Navigating to a Page: Since we have defined our Home, all the remaining is to navigate from home to another route of the app. For that the navigator widget has a method called **Navigator.push()**. This method pushes the route on top of the home, thereby displaying the second route.

Navigating Back to Home: Now we have arrived at our destination, but how do we go back home? For that, the navigator has a method called Navigator.pop(). This helps us to remove the present route from the stack so that we go back to our home route.

Step 1: First, you need to create two routes.

Step 2: Then, navigate to one route from another route by using the Navigator.push() method.

Step 3: Finally, navigate to the first route by using the Navigator.pop() method.

Let us take a simple example to understand the navigation between two routes:

Create two routes

Here, we are going to create two routes for navigation. In both routes, we have created only a **single button.** When we tap the button on the first page, it will navigate to the second page. Again, when we tap the button on the second page, it will return to the first page.

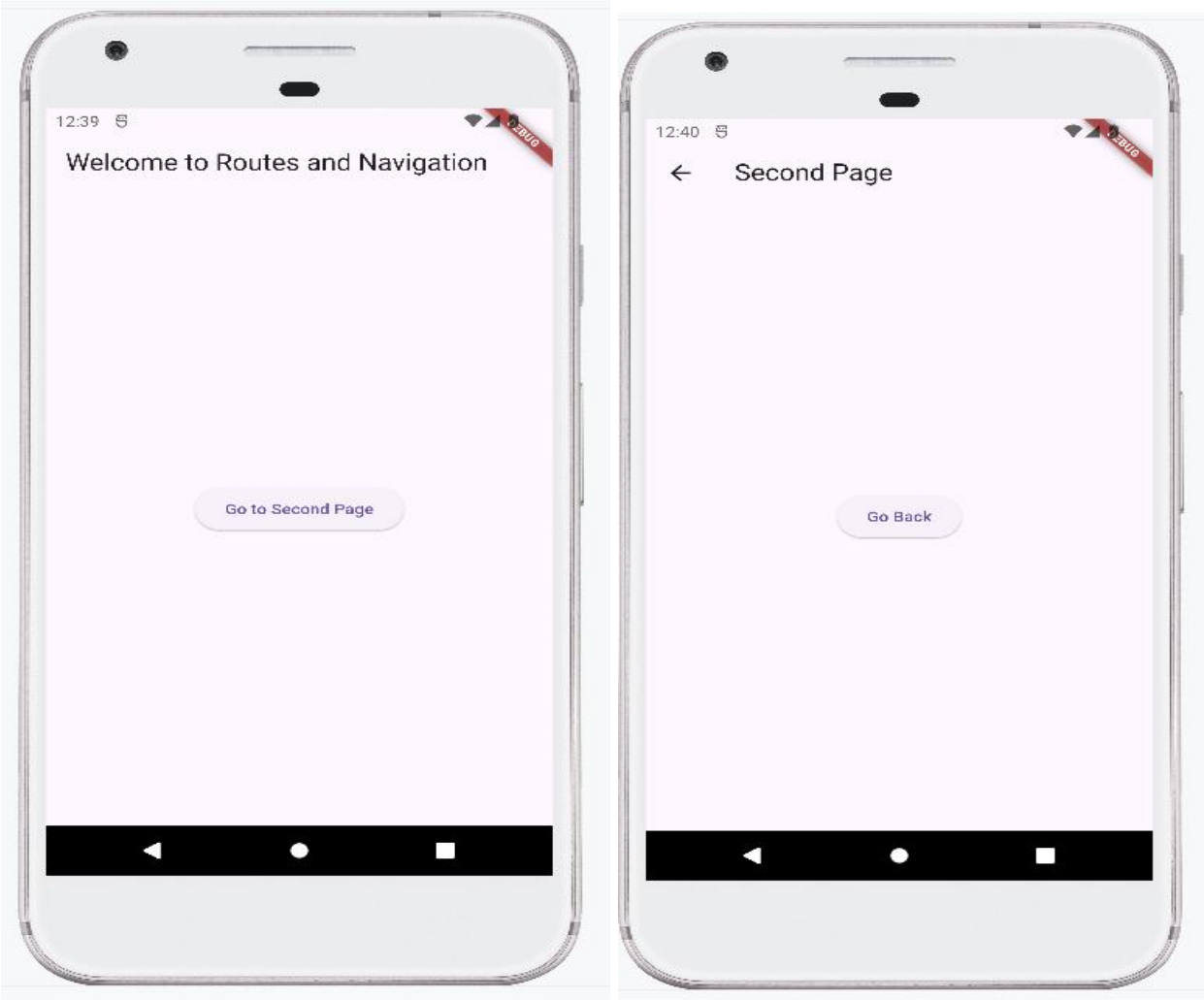
The below code snippet creates two routes in the Flutter application.

main.dart

```
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Navigation Example',
      home: HomePage(), // The starting page
    );
  }
}

// Home Page
class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Welcome to Routes and Navigation")),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            // Navigate to SecondPage
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => SecondPage()),
            );
          },
          child: Text("Go to Second Page"),
        ),
      ),
    );
  }
}

// Second Page
class SecondPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Second Page")),
      body: Center(
```

```
        child: ElevatedButton(
          onPressed: () {
            // Navigate back to the HomePage
            Navigator.pop(context);
          },
          child: Text("Go Back"),
        ),
      ),
    );
  }
}
```

OUTPUT:



Conclusion: In this experiment we have designed a flutter application with navigation functionality.