

Experiment 05**Aim: To apply navigation, routing and gestures in Flutter App****Theory:**

Flutter uses Navigator for route management and gesture widgets like GestureDetector for user interaction. This provides seamless screen transitions and intuitive UX.

- Navigation to HomePage after successful registration.
- You might also have gestures like onTap or onPressed, though not clearly visible here.

Navigation & Routing:

Navigation and routing in Flutter typically involve the use of Navigator.push() or Navigator.pushReplacement() to move between screens.

1. Navigating to Landing Screen (on logout):

```
Navigator.pushReplacement(  
  context,  
  MaterialPageRoute(builder: (context) => LandingScreen()),  
);
```

- Type: pushReplacement (replaces the current route)
- Purpose: Navigates to LandingScreen after logging out

2. Navigating to Profile List Screen (on tapping a society):

```
Navigator.push(  
  context,  
  MaterialPageRoute(  
    builder: (context) => ProfileListScreen(  
      userId: widget.userId,  
      userType: widget.userType,  
      societyId: societyId,  
    ),  
  ),  
);
```

- Type: push (adds a new route to the stack)
- Purpose: Opens the ProfileListScreen with relevant data for the selected society

Gesture Handling:

Gesture handling refers to responding to user interactions like tapping.

1. Tapping a List Item (Society Card):

```
onTap: () {  
  Navigator.push(  
    context,  
    MaterialPageRoute(  
      builder: (context) => ProfileListScreen(  
        userId: widget.userId,  
        userType: widget.userType,  
        societyId: societyId,  
      ),  
    ),  
  );  
}
```

- Gesture: onTap of a ListTile
- Effect: Triggers navigation to a detailed profile list of the selected society

2. Tapping Floating Action Button to Add Society:

```
floatingActionButton: widget.userType == "Manager"  
  ? FloatingActionButton(  
    onPressed: _addSociety,  
    ...  
  )  
  : null,
```

- Gesture: Tap on the FAB
- Effect: Opens a dialog to add a new society (managers only)

3. Tapping the "Add" Button in Dialog:

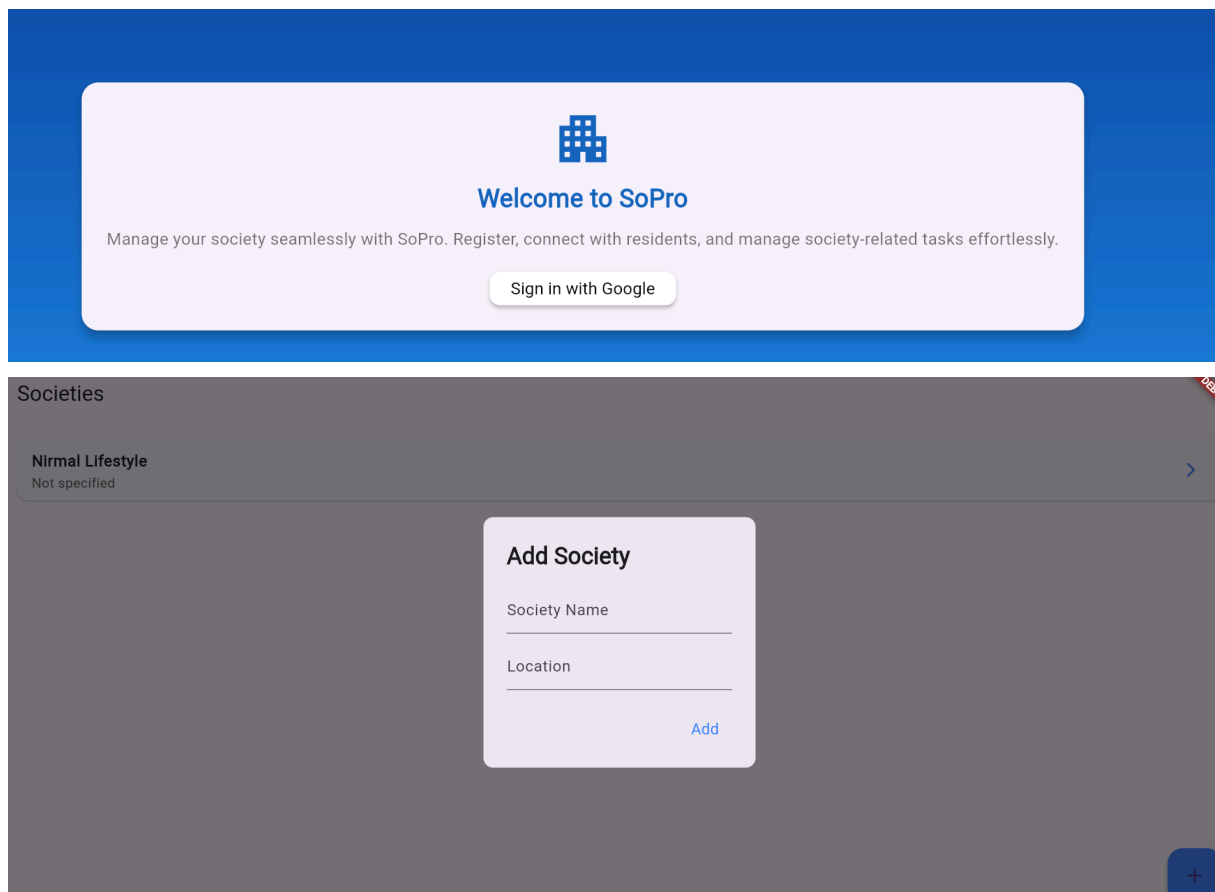
```
TextButton(  
  onPressed: () async {  
    ...  
    Navigator.pop(context);  
  },  
  child: Text("Add"),  
)
```

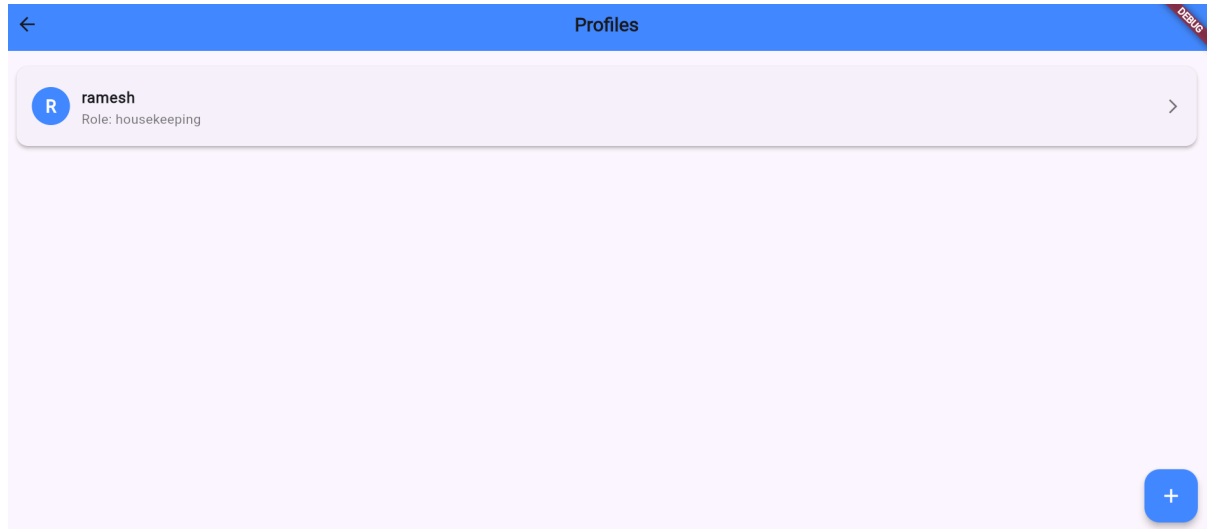
- Gesture: Tap on the TextButton
- Effect: Adds the society to Firestore and closes the dialog

Summary:

Type	Location in Code	Purpose
Navigation	Navigator.pushReplacement(...)	Redirects to LandingScreen after logout
Navigation	Navigator.push(...)	Opens ProfileListScreen when a society is tapped
Gesture	onTap in ListTile	Handles society list item selection
Gesture	onPressed of FloatingActionButton	Opens dialog to add a new society
Gesture	onPressed of TextButton inside AlertDialog	Adds the new society and dismisses dialog

Let me know if you want a visual flow of the navigation or want this section commented inline in your code!





Conclusion:

Effective routing and gestures enabled smooth multi-page transitions within the Society Management platform.