

EXPERIMENT NO: - 05**Name :** Anushka Shahane**Class :** D15A**Roll No :** 54**AIM:** - To apply navigation, routing and gestures in Flutter App.**Theory:** -

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.

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.

Gestures enable the app to respond to user interactions, making the application more dynamic and responsive.

➤ **Navigation and Routing in Flutter**

Navigation is the process of moving between different screens or pages in an app. Flutter provides a simple and effective way to handle this through the use of the `Navigator` widget and routes.

1. Using Navigator Widget

The `Navigator` widget manages a stack of routes, allowing for pushing and popping routes on the stack.

- **Pushing a Route:** To navigate to a new screen, use `Navigator.push()`.
- **Popping a Route:** To go back to the previous screen, use `Navigator.pop()`.

```
ElevatedButton(  
  onPressed: () {  
    Navigator.push(  
      context,    ),  
  ),  
)
```

```

        context,

        MaterialPageRoute(builder: (context) => SecondScreen()),

    );},

);

```

2. Named Routes

Flutter also allows the use of named routes to navigate, which can make the routing process cleaner, especially in larger applications.

```

MaterialApp(
  initialRoute: '/',
  routes: {
    '/': (context) => HomeScreen(), '/second':
      (context) => SecondScreen(),
  },
);

```

Navigate to the route using `Navigator.pushNamed()` `Navigator.pushNamed(context, '/second')`;

Handling Gestures in Flutter

Gestures refer to user interactions with the app, such as taps, swipes, pinches, and drags. Flutter provides several widgets and gesture detectors to handle these interactions.

Tap Gestures

The most common gesture is the tap, which can be handled using the `GestureDetector` widget or specific buttons like `InkWell` or `ElevatedButton`.

Long Press Gesture

For long press gestures, Flutter provides the `onLongPress` callback in `GestureDetector` or `InkWell`.

Swipe and Drag Gestures

Flutter also provides swipe and drag gesture handling. The `onHorizontalDragUpdate` and `onVerticalDragUpdate` callbacks are used for dragging gestures.

Code: - home_screen.dart

```

import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'details_form_screen.dart';

class HomeScreen extends StatefulWidget {
  @override
  _HomeScreenState createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  int _selectedIndex = 0;
  String _selectedDate = "Select date";

  void _onBottomNavTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

  Future<void> _selectDate(BuildContext context) async {
    DateTimeRange? picked = await showDateRangePicker(
      context: context,
      firstDate: DateTime.now(),
      lastDate: DateTime(2030),
    );

    if (picked != null) {
      setState(() {
        _selectedDate =
          "${DateFormat('E dd MMM').format(picked.start)} -
          ${DateFormat('E dd MMM').format(picked.end)}";
      });
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[100],
      appBar: AppBar(
        backgroundColor: Colors.blue[900],
        title: Text(
          'Booking.com',
          style: TextStyle(
            color: Colors.white, fontWeight: FontWeight.bold,
            fontSize: 18),
        ),
      ),
      actions: [
        IconButton(
          icon: Icon(Icons.notifications, color: Colors.white),
          onPressed: () {},
        ),

```

```

        IconButton(
          icon: Icon(Icons.account_circle, color:
Colors.white),
          onPressed: () {},
        ),
      ],
    ),
    body: SingleChildScrollView(
      child: Padding(
        padding: const EdgeInsets.all(12),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            // **Category Title**
            Row(
              mainAxisAlignment:
MainAxisAlignment.center,
              children: [
                Icon(Icons.hotel, color: Colors.blue[900],
size: 24),
                SizedBox(width: 8),
                Text(
                  "Find your stays",
                  style: TextStyle(
                    color: Colors.blue[900],
                    fontSize: 14,
                    fontWeight: FontWeight.bold,
                  ),
                ),
              ],
            ),
            SizedBox(height: 15),

            // **Search Section**
            Container(
              padding: EdgeInsets.all(12),
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(10),
                border: Border.all(color: Colors.orange,
width: 2),
                color: Colors.white,
              ),
              child: Column(
                children: [
                  _buildSearchField(Icons.search, "Enter your
destination"),
                  GestureDetector(
                    onTap: () => _selectDate(context),
                    child: _buildSearchField(
                      Icons.calendar_today,
                      _selectedDate,

```



```

        icon: Icon(Icons.person, size: 22), label: "My account"),
      ],
    ),
  );
}

```

```

Widget _buildSearchField(IconData icon, String hintText,
  {bool isReadOnly = false}) {
  return Padding(
    padding: EdgeInsets.symmetric(vertical: 4),
    child: TextField(
      readOnly: isReadOnly,
      decoration: InputDecoration(
        prefixIcon: Icon(icon, color: Colors.black54, size: 20),
        hintText: hintText,
        hintStyle: TextStyle(fontSize: 12),
        border: OutlineInputBorder(borderRadius:
BorderRadius.circular(5)),
      ),
    ),
  );
}

```

```

Widget _buildDiscountCard(String title, String subtitle) {
  return Container(
    width: 150,
    margin: EdgeInsets.only(right: 8),
    padding: EdgeInsets.all(8),
    decoration: BoxDecoration(
      color: Colors.white,
      border: Border.all(color: Colors.blue),
      borderRadius: BorderRadius.circular(8),
    ),
    child: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Text(title,
          style: TextStyle(fontWeight: FontWeight.bold,
fontSize: 12)),
        SizedBox(height: 4),
        Text(subtitle, style: TextStyle(fontSize: 10)),
      ],
    ),
  );
}

```

```

Widget _buildDealCard(String location, String price,
String imageUrl) {
  return Container(
    width: 180,
    margin: EdgeInsets.only(right: 8),

```

```

padding: EdgeInsets.all(8),
decoration: BoxDecoration(
  color: Colors.white,
  borderRadius: BorderRadius.circular(8),

```

```

    boxShadow: [

```

```

      BoxShadow(color: Colors.grey.withOpacity(0.2),
blurRadius: 4)
    ],
  ),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      ClipRRect(

```

```

borderRadius: BorderRadius.circular(8),

```

```

    child: Image.network(
      imageUrl,
      height: 80,
      width: double.infinity,
      fit: BoxFit.cover,
    ),
  ),
  SizedBox(height: 4),
  Text(location,
    style: TextStyle(fontWeight: FontWeight.bold,
fontSize: 12)),
  SizedBox(height: 4),
  Text(price, style: TextStyle(fontSize: 11, color:
Colors.black54)),
    ],
  ),
);
}
}

```

Code : hotel_list.dart

```
import 'package:flutter/material.dart';
import '../models/destination_model.dart';
import '../services/hotel_service.dart';

class HotelListScreen extends StatefulWidget {
  final String destination;
  final String checkIn;
  final String checkOut;
  final String rooms;

  const HotelListScreen({
    Key? key,
    required this.destination,
    required this.checkIn,
    required this.checkOut,
    required this.rooms,
  }) : super(key: key);

  @override
  _HotelListScreenState createState() =>
    _HotelListScreenState();
}

class _HotelListScreenState extends
State<HotelListScreen> {
  late Future<List<Destination>>
    _futureDestinations;
  Set<String> favoriteHotels = {};

  @override
  void initState() {
    super.initState();
    _futureDestinations =
HotelService.fetchDestinations(
  destinationQuery: widget.destination,
  checkIn: widget.checkIn,
  checkOut: widget.checkOut,
  rooms: widget.rooms,
);
  }
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.white,
    appBar: AppBar(
      backgroundColor: Colors.blue[900],
      title: Container(
        padding: EdgeInsets.symmetric(vertical: 8,
horizontal: 12),
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(13),
          border: Border.all(color: Colors.orange, width: 6),
        ),
        child: Row(
          children: [
            // Icon(Icons.arrow_back, color: Colors.black),
            SizedBox(
              width: 10,
              height: 10,
            ),
            Text(
              "${widget.destination} · ${widget.checkIn} -
${widget.checkOut}",
              style: TextStyle(
                color: Colors.black,
                fontWeight: FontWeight.w100,
                fontSize: 15),
            ),
          ],
        ),
      ),
    body: Column(
      children: [
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: Row(
            mainAxisAlignment:
MainAxisAlignment.spaceBetween,
            children: [
              Icon(Icons.swap_vert, color: Colors.black), //
Sort Icon
              Icon(Icons.tune, color: Colors.black), // Filter
Icon
              Icon(Icons.map, color: Colors.black), // Map
Icon
            ],
          ),
        ),
      ],
    ),
  );
}
```

```

],
  ),
),
Expanded(
  child: FutureBuilder<List<Destination>>(
    future: _futureDestinations,
    builder: (context, snapshot) {
      if (snapshot.connectionState ==
ConnectionState.waiting) {
        return Center(child:
CircularProgressIndicator());
      } else if (snapshot.hasError) {
        return Center(child: Text("Error:
${snapshot.error}"));
      } else if (!snapshot.hasData ||
snapshot.data!.isEmpty) {
        return Center(child: Text("No hotels
found"));
      } else {
        final destinations = snapshot.data!;
        return ListView.builder(
          itemCount: destinations.length,
          itemBuilder: (context, index) {
            final dest = destinations[index];
            return Card(
              margin:
                EdgeInsets.symmetric(horizontal:
12, vertical: 6),
              shape: RoundedRectangleBorder(
                borderRadius:
BorderRadius.circular(15),
              ),
              elevation: 5,
              child: Container(
                decoration: BoxDecoration(
                  color: Colors.white,
                  borderRadius:
BorderRadius.circular(15),
                  boxShadow: [
                    BoxShadow(
                      color:
Colors.grey.withOpacity(0.3),
                      spreadRadius: 2,
                      blurRadius: 5,
                      offset: Offset(0, 3),
                    ),
                  ],
                ),
              ),
            ),
          ),
        ),
      ),
    ),
  ),
),

```

```

const SizedBox(height: 5),
  Text(
    "We'll use this to create an account if you
don't have one yet.",
    style: GoogleFonts.montserrat(
      fontSize: 14,
      color: Colors.grey[600],
    ),
  ),
const SizedBox(height: 30),

// Email Entry Box
Container(
  padding: const
EdgeInsets.symmetric(horizontal: 12),
  decoration: BoxDecoration(
    borderRadius: BorderRadius.circular(8),
    border: Border.all(color:
Colors.grey.shade400),
  ),
  child: TextField(
    decoration: InputDecoration(
      labelText: "Enter your name",
      labelStyle: GoogleFonts.montserrat(
        fontSize: 14, color: Colors.grey[600]),
      border: InputBorder.none,
    ),
  ),
),
const SizedBox(height: 30),

// Continue Button
CustomButton(
  text: "Continue",
  onTap: () => Navigator.pushNamed(context,
Routes.signUp),
  backgroundColor: const Color(0xFF003580),
  textColor: Colors.white,
),
const SizedBox(height: 40),

```

```

Row(
  children: [
    Expanded(child: Divider(color: Colors.grey.shade400)),
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 10),
      child: Text("OR",
        style: GoogleFonts.montserrat(
          fontSize: 14, color: Colors.black87)),
    ),
    Expanded(child: Divider(color: Colors.grey.shade400)),
  ],
),
const SizedBox(height: 30),

// Continue with Google
CustomButton(
  text: "Continue with Google",
  onTap: () {},
  isOutlined: true,
  textColor: Colors.black,
  borderColor: Colors.grey.shade400,
  iconPath: "assets/images/google.png", // Add Google icon
),
const SizedBox(height: 10),

// Continue with Facebook
CustomButton(
  text: "Continue with Facebook",
  iconPath: "assets/images/facebook.png", // Add Facebook icon
),
const SizedBox(height: 10),
Center(
  child: TextButton(
    onPressed: () => Navigator.pushNamed(context, Routes.signIn),
    child: Text(
      "Already have an account? Sign In",
      style: GoogleFonts.montserrat(
        fontSize: 14,
        fontWeight: FontWeight.w500,
        color: Colors.blue,
      ),
    ),
  ),
),
),
}
}

```


Code: hotel_service.dart

```

import 'dart:convert';
import 'package:http/http.dart' as http;
import '../models/destination_model.dart';

class HotelService {
  // Updated endpoint using your curl command
  static const String _endpoint =
    "https://booking-com15.p.rapidapi.com/api/v1/hotels/searchDestination";

  static Future<List<Destination>> fetchDestinations({
    required String destinationQuery,
    required String checkIn, // include if supported by your API
    required String checkOut, // include if supported by your API
    required String rooms, // include if supported by your API
  }) async {
    // Build the URL with query parameters.
    // The curl shows the API accepts a "query" parameter.
    // If your API supports additional parameters, they can be appended here.
    final Uri url = Uri.parse(_endpoint).replace(queryParameters: {
      'query': destinationQuery,
      'checkin': checkIn,
      'checkout': checkOut,
      'rooms': rooms,
    });

    // Headers required by the API. (Remember to secure your API key in production.)
    final headers = {
      'x-rapidapi-host': 'booking-com15.p.rapidapi.com',
      'x-rapidapi-key': '94e5999c47msh7a64d356faaa760p1b75eajsn392f227c4818',
      'Content-Type': 'application/json',
    };

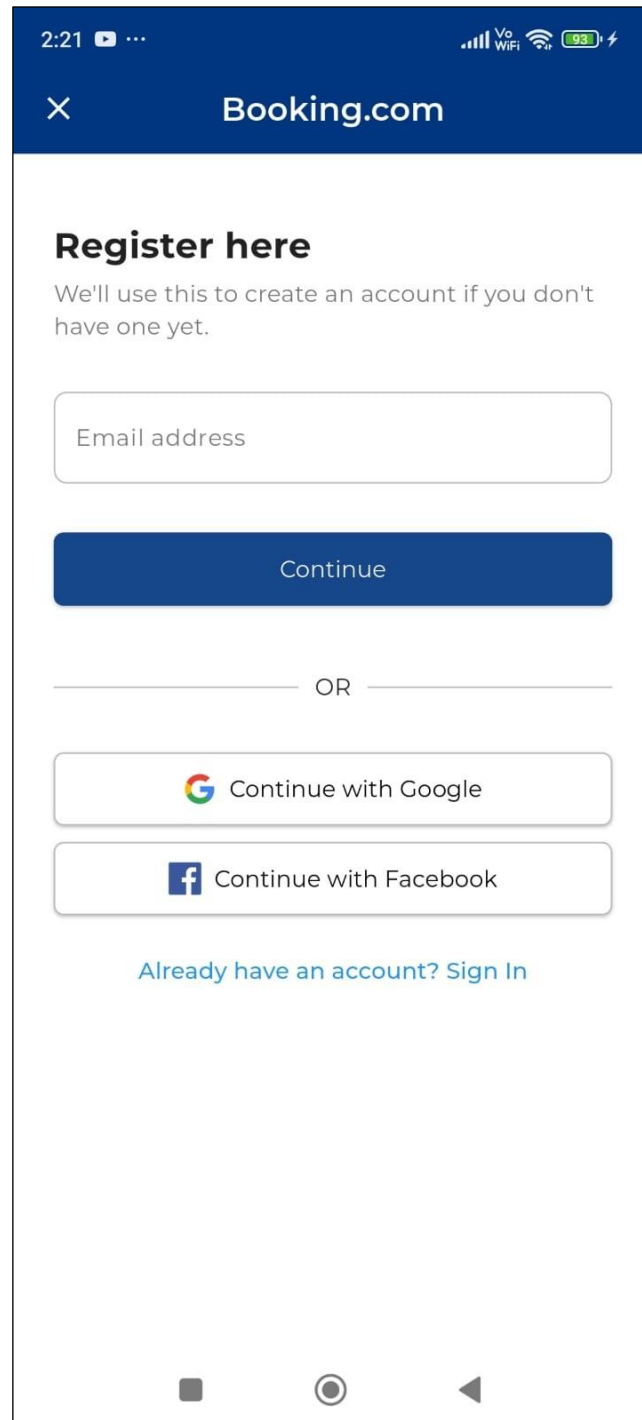
    final response = await http.get(url, headers: headers);

    if (response.statusCode == 200) {
      final Map<String, dynamic> jsonResponse = json.decode(response.body);
      // Expecting a response similar to:
      // {
      //   "status": true,
      //   "message": "Success",
      //   "timestamp": 1739725005441,
      //   "data": [ { ... }, { ... }, ... ]
      // }
      if (jsonResponse['status'] == true) {
        final List<dynamic> data = jsonResponse['data'];
        return data.map((item) => Destination.fromJson(item)).toList();
      } else {
        throw Exception("API error: ${jsonResponse['message']}");
      }
    } else {
      throw Exception(
        "Failed to load destinations. HTTP Status: ${response.statusCode}");
    }
  }
}

```

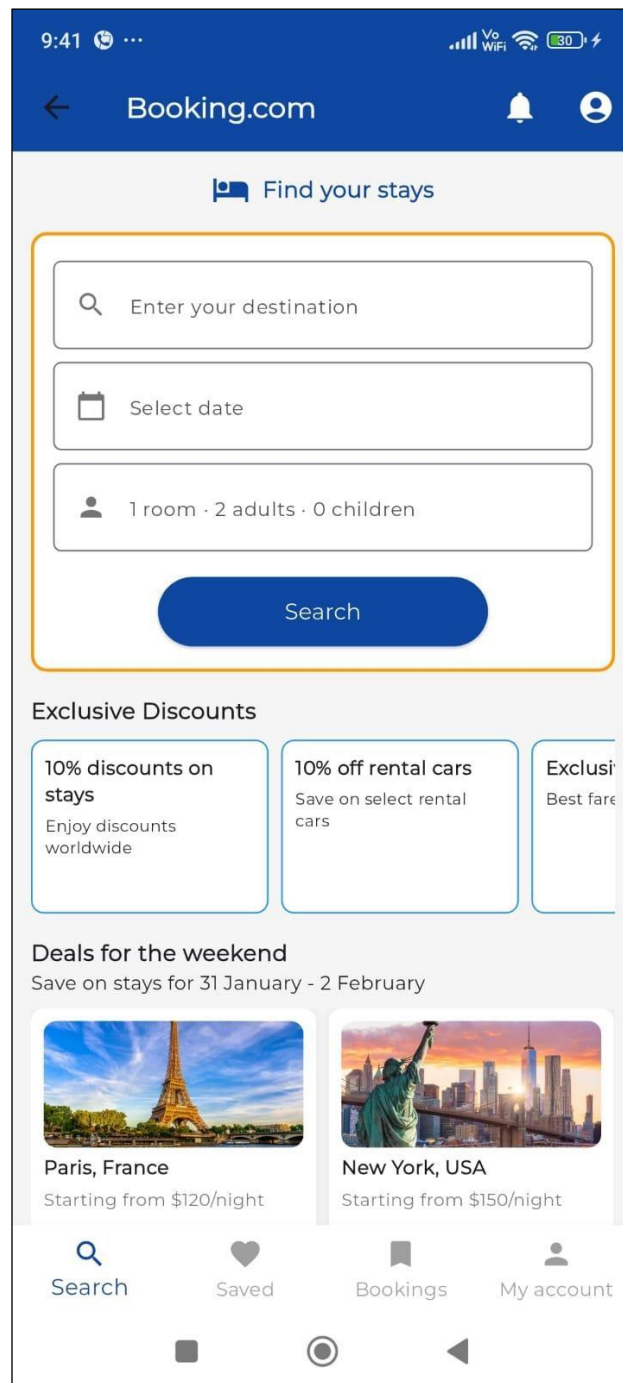
OUTPUT :

From the welcome page once registration and login is done it navigates to home page

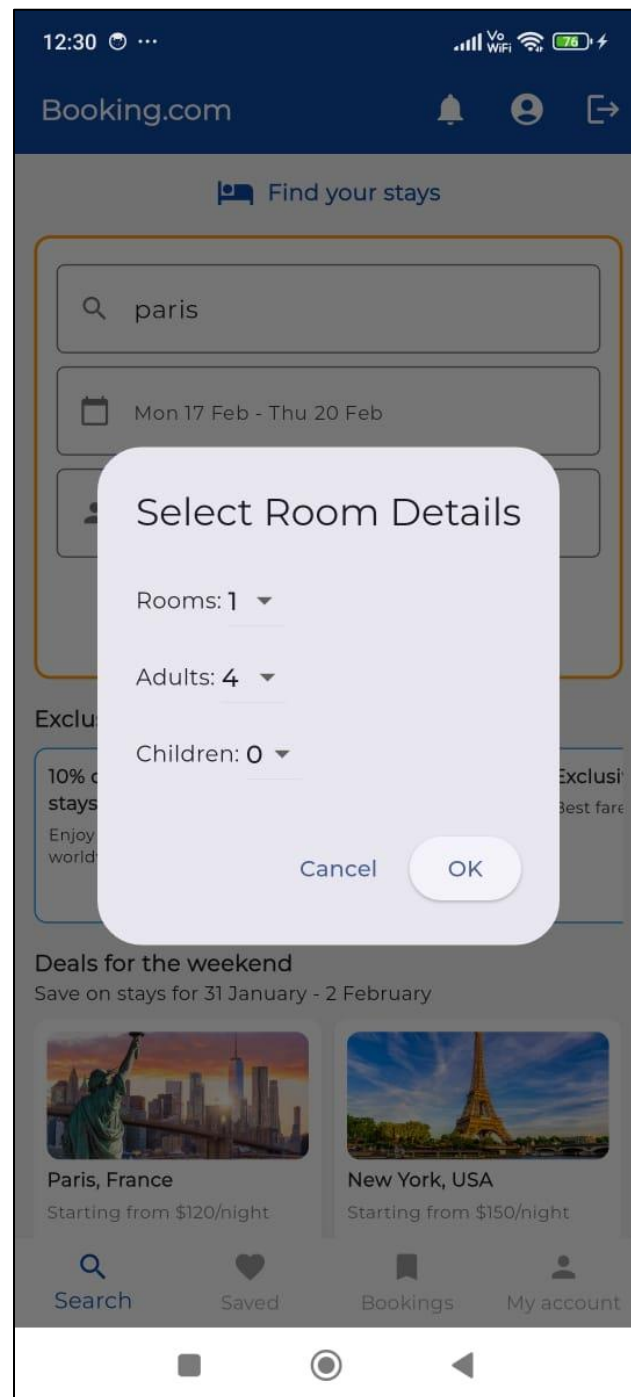


The image shows a mobile application interface for Booking.com. At the top, there is a dark blue header with a white 'X' icon on the left and the text 'Booking.com' in the center. Below the header, the main content area is white. It starts with the heading 'Register here' in bold black text. Underneath, a smaller line of text says 'We'll use this to create an account if you don't have one yet.' Below this text is a white rectangular input field with the placeholder text 'Email address'. Under the input field is a solid blue button with the word 'Continue' in white text. Below the button is a horizontal line with the word 'OR' in the center. Under the line are two more buttons: the first one has the Google 'G' logo and the text 'Continue with Google', and the second one has the Facebook 'f' logo and the text 'Continue with Facebook'. At the bottom of the form area, there is a link that says 'Already have an account? Sign In' in blue text. At the very bottom of the screen, there are three small icons: a square, a circle, and a triangle, which are standard Android navigation bar icons.

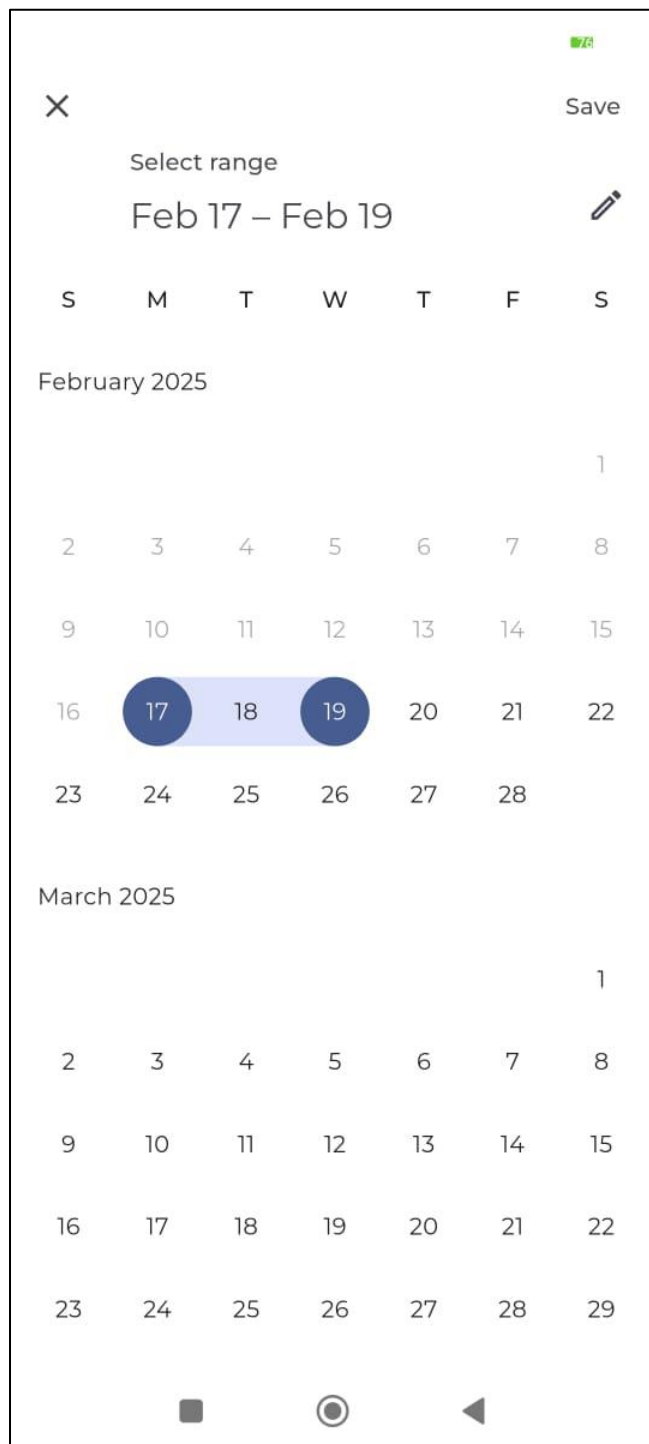
From the home page the user can add destination , date and rooms



Gesture of rooms is enabled



Gesture of calender is enabled



Then after clicking on search the user navigates to hotels page

