MPL Lab Exp 2

Name: Anurag Gaiwal Roll No. 17
Div: D15A Batch: A

Aim: To design Flutter UI by including common widgets.

Theory:

We can split the Flutter widget into two categories:

1. Visible (Output and Input)
2. Invisible (Layout and Control)

1. Visible widget

The visible widgets are related to the user input and output data. Some of the important types of
this widget are:

1. Text

A Text widget holds some text to display on the screen. We can align the text widget by using
textAlign property, and style property allow the customization of Text that includes font, font
weight, font style, letter spacing, color, and many more.

2. Button

This widget allows you to perform some action on click. Flutter does not allow you to use the
Button widget directly; instead, it uses a type of buttons like a FlatButton and a RaisedButton.

3. Image

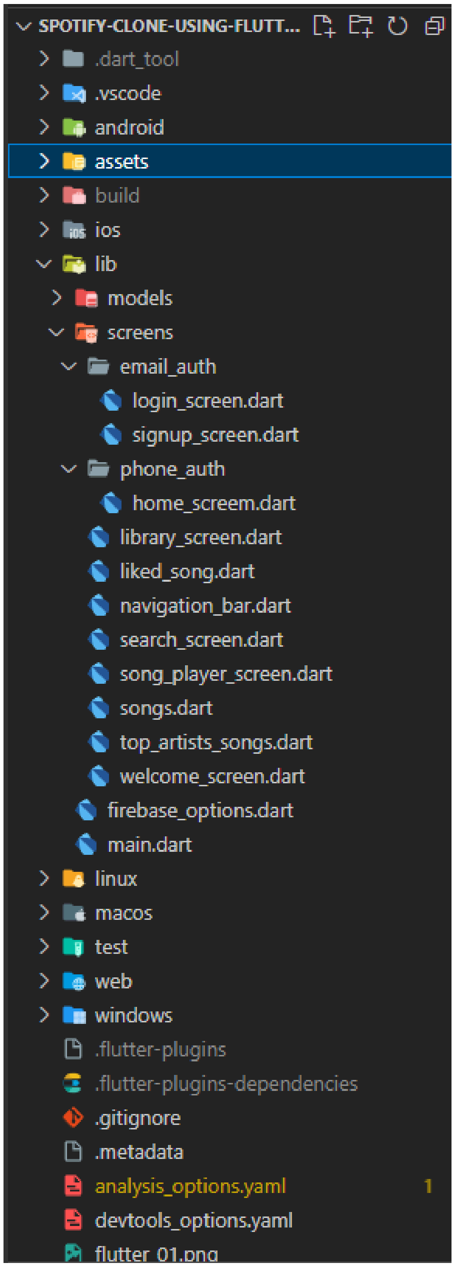
This widget holds the image which can fetch it from multiple sources like from the asset
folder or directly from the URL. It provides many constructors for loading image, which are given below:

o Image: It is a generic image loader, which is used by ImageProvider.

o asset: It load image from your project asset folder.

o file: It loads images from the system folder.
o memory: It load image from memory.
o network: It loads images from the network.
To add an image in the project, you need first to create an assets folder where you keep your images and then add the below line in pubspec.yaml file.

<u>File structure:</u>



```
SPOTIFY-CLONE-USING-FLUTT...
  > .dart_tool
  > .vscode
  > android
  > assets
  > build
  > ios
  v lib
    > models
    v screens
      v email_auth
          login_screen.dart
          signup_screen.dart
      v phone_auth
          home_screem.dart
        library_screen.dart
        liked_song.dart
        navigation_bar.dart
        search_screen.dart
        song_player_screen.dart
        songs.dart
        top_artists_songs.dart
        welcome_screen.dart
      firebase_options.dart
      main.dart
  > linux
  > macos
  > test
  > web
  > windows
    .flutter-plugins
    .flutter-plugins-dependencies
    .gitignore
    .metadata
    analysis_options.yaml              1
    devtools_options.yaml
    flutter_01.png
```

Code:

main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:jio_saavn_auth/firebase_options.dart';
import 'package:jio_saavn_auth/models/song.dart';
import 'package:jio_saavn_auth/screens/welcome_screen.dart';
import 'package:provider/provider.dart';

Future<void> main() async {
    WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: DefaultFirebaseOptions.currentPlatform,
  );
  // runApp(
  //   ChangeNotifierProvider(
  //     create: (context) => LikedSongsModel(),
  //     child: MyApp(),
  //   ),
  // );
  runApp(
    MultiProvider(
      providers: [
        ChangeNotifierProvider(create: (context) => LikedSongsModel()),
        // Add other providers if needed
      ],
      child: MyApp(),
    ),
  );
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) => LikedSongsModel(),
      child: MaterialApp(
        title: 'Spotify',
        home: WelcomePage(), // Use your main screen here
      ),
    );
  }
}
```

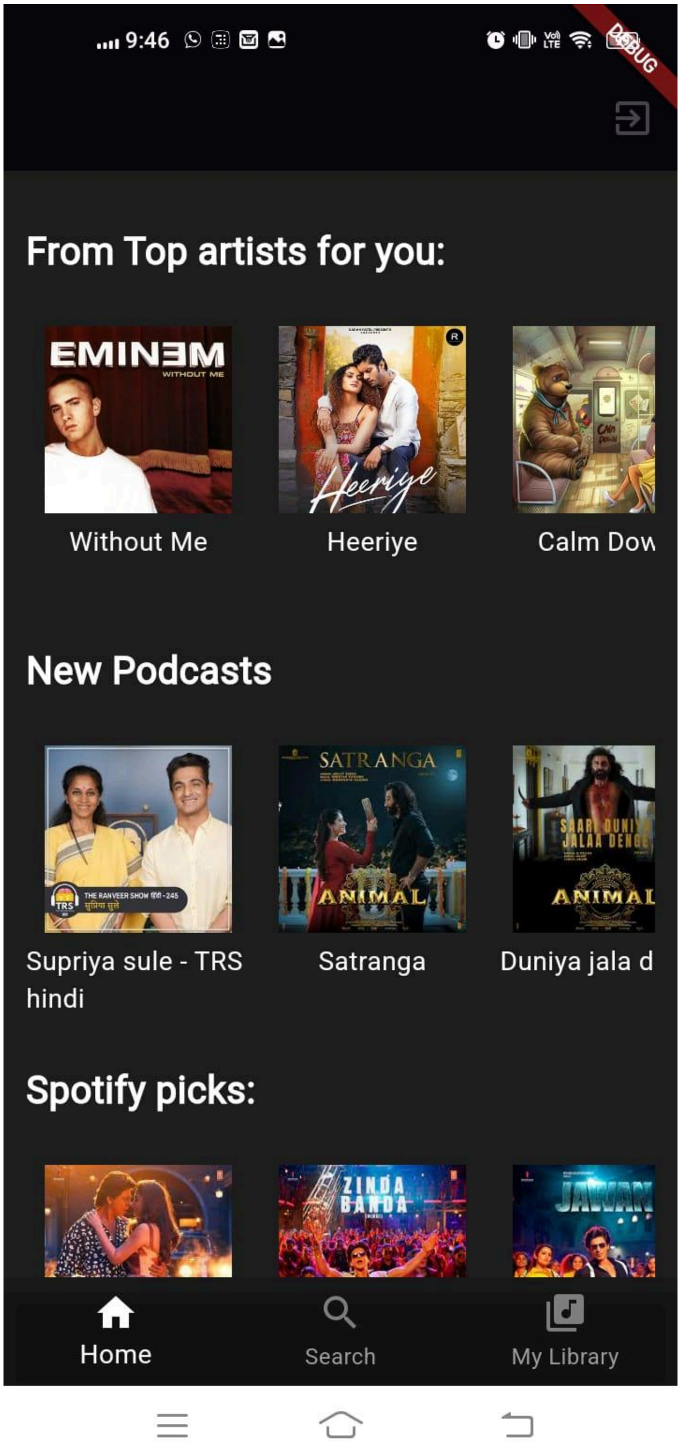
Home_screen.dart:

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:jio_saavn_auth/screens/email_auth/login_screen.dart';
import "package:jio_saavn_auth/screens/library_screen.dart";
import 'package:jio_saavn_auth/screens/navigation_bar.dart';
import 'package:jio_saavn_auth/screens/search_screen.dart';
import 'package:jio_saavn_auth/screens/song_player_screen.dart';
import 'package:jio_saavn_auth/screens/top_artists_songs.dart';

class Song {
  final String name;
  final String path;
  final String imagePath;

  Song({
    required this.name,
    required this.path,
    required this.imagePath,
  });
}

List<Song> recommendedSongs = [
  Song(
      name: 'Alone',
      path: 'assets/songs/song1.mp3',
      imagePath: 'assets/song1.png'),
  Song(
      name: 'Kesariya',
      path: 'assets/songs/song2.mp3',
      imagePath: 'assets/song2.png'),
  Song(
      name: 'We own it',
      path: 'assets/songs/song3.mp3',
      imagePath: 'assets/song3.png'),
];

// Add this function outside the HomeScreen class
Future<List<String>> fetchTopSongsForArtist(String artistName) async {
  // Simulate fetching data, replace this with actual data fetching logic
  await Future.delayed(Duration(seconds: 2)); // Simulating a delay
  // Hardcoded top songs for each artist
  Map<String, List<String>> artistTopSongs = {
    'Without Me': ['song1.mp3', 'song2.mp3', 'song3.mp3'],
    'Heeriye': ['song4.mp3', 'song5.mp3', 'song6.mp3'],
    'Calm Down': ['song7.mp3', 'song8.mp3', 'song9.mp3'],
    // Add more artists and their top songs as needed
  };

  return artistTopSongs[artistName] ?? [];
}

class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}
```

```
class _HomeScreenState extends State<HomeScreen> {
  int _selectedIndex = 0;

  void logout() async {
    await FirebaseAuth.instance.signOut();

    Navigator.popUntil(context, (route) => route.isFirst);
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => LoginScreen()),
    );
  }

  Color _getColor(int index) {
    return _selectedIndex == index
        ? Colors.white
        : const Color.fromARGB(255, 128, 128, 128);
  }

  Widget _buildSuggestedItem(BuildContext context, Song song) {
    return GestureDetector(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => SongPlayerScreen(
              songPaths: [song.path],
              songName: song.name,
              imagePath: song.imagePath,
            ),
          ),
        );
      },
      child: Container(
        width: 120,
        margin: EdgeInsets.only(right: 5),
        child: Column(
          children: [
            Image.asset(
              song.imagePath,
              height: 100,
              width: 100,
              fit: BoxFit.cover,
            ),
            SizedBox(height: 5),
            Text(
              song.name,
              style: TextStyle(color: Color.fromARGB(255, 255, 255, 255)),
            ),
          ],
        ),
      ),
    );
  }

  Widget _buildImageRow(List<Song> songs) {
    return SizedBox(
      height: 150,
      child: ListView.builder(
        scrollDirection: Axis.horizontal,
        itemCount: songs.length,
        itemBuilder: (BuildContext context, int index) {
          return GestureDetector(
            onTap: () async {
              print('Tapped on ${songs[index].name}');

              // Fetch the top songs for the selected artist
        List<String> topSongs =
          await fetchTopSongsForArtist(songs[index].name);
        print('Top songs for ${songs[index].name}: $topSongs');

        // Navigate to TopArtistSongsScreen with the fetched top songs
        Navigator.push(
         context,
         MaterialPageRoute(
          builder: (context) => TopArtistSongsScreen(
           artistName: songs[index].name,
           topSongs: topSongs,
          ),
         ),
        );
       },
       child: _buildSuggestedItem(context, songs[index]),
      );
     },
    ),
   );
  }

  @override
  Widget build(BuildContext context) {
   List<Song> topArtists = [
    Song(
      name: 'Without Me',
      path: 'assets/songs/eminem.mp3',
      imagePath: 'assets/travis.png'),
    Song(
      name: 'Heeriye',
      path: 'assets/songs/arjit.mp3',
      imagePath: 'assets/sonu.png'),
    Song(
      name: 'Calm Down',
      path: 'assets/songs/selena.mp3',
      imagePath: 'assets/drake.png'),
];

List<Song> newReleases = [
  Song(
      name: 'Supriya sule - TRS hindi',
      path: 'assets/songs/new2.mp3',
      imagePath: 'assets/trsh_supriya_sule.jpg'),
  Song(
      name: 'Satranga',
      path: 'assets/songs/new3.mp3',
      imagePath: 'assets/satranga.png'),
  Song(
      name: 'Duniya jala denge',
      path: 'assets/songs/new3.mp3',
      imagePath: 'assets/sariduniya.png'),
];

List<Song> jioSaavnPicks = [
  Song(
      name: 'Challeya',
      path: 'assets/songs/pick1.mp3',
      imagePath: 'assets/challeya.png'),
  Song(
      name: 'Zinda Banda',
      path: 'assets/songs/pick2.mp3',
      imagePath: 'assets/zindabanda.png'),
  Song(
      name: 'Jawan Title Track',
      path: 'assets/songs/pick3.mp3',
      imagePath: 'assets/jawan.png'),
];
```

```
return Scaffold(
  appBar: AppBar(
    automaticallyImplyLeading: false,
    backgroundColor: Colors.black,
    actions: [
      IconButton(
        onPressed: () {
          logout();
        },
        icon: Icon(Icons.exit_to_app),
      ),
    ],
  ),
  backgroundColor: Color.fromARGB(255, 30, 30, 30),
  body: SingleChildScrollView(
    child: Padding(
      padding: const EdgeInsets.all(12.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Center(
            child: Align(
              alignment: Alignment.center,
              child: Image.asset(
                'assets/jiosaavn_name.png',
                height: 150,
                width: 200,
              ),
            ),
          ),
          SizedBox(height: 10),
          SizedBox(height: 20),
          Text(
            'Recommended Songs for you:',
            style: TextStyle(
              fontSize: 20,
              fontWeight: FontWeight.bold,
              color: Color.fromARGB(255, 255, 255, 255),
            ),
          ),
          SizedBox(height: 25),
          _buildImageRow(recommendedSongs),
          SizedBox(height: 20),
          Text(
            'From Top artists for you:',
            style: TextStyle(
              fontSize: 20,
              fontWeight: FontWeight.bold,
              color: Color.fromARGB(255, 255, 255, 255),
            ),
          ),
          SizedBox(height: 25),
          _buildImageRow(topArtists),
          SizedBox(height: 20),
          Text(
            'New Podcasts',
            style: TextStyle(
              fontSize: 20,
              fontWeight: FontWeight.bold,
              color: Color.fromARGB(255, 255, 255, 255),
            ),
          ),
          SizedBox(height: 25),
          _buildImageRow(newReleases),
          SizedBox(height: 20),
          Text(
            'Spotify picks:',
                          style: TextStyle(
                            fontSize: 20,
                            fontWeight: FontWeight.bold,
                            color: Color.fromARGB(255, 255, 255, 255),
                          ),
                        ),
                        SizedBox(height: 25),
                        _buildImageRow(jioSaavnPicks),
                      ],
                    ),
                  ),
                ),
                bottomNavigationBar: CustomBottomNavigationBar(
                  selectedIndex: _selectedIndex,
                  onItemTapped: (index) {
                    setState(() {
                      _selectedIndex = index;
                    });
                    if (_selectedIndex == 1) {
                      Navigator.push(
                        context,
                        MaterialPageRoute(builder: (context) => SearchScreen()),
                      );
                    } else if (_selectedIndex == 2) {
                      Navigator.push(
                        context,
                        MaterialPageRoute(builder: (context) => MyLibrary()),
                      );
                    }
                  },
                ),
              );
            }
          }
```

Home screen:


9:46
DEBUG
From Top artists for you:
EMINEM
WITHOUT ME
Without Me
Heeriye
Heeriye
Calm Dow
New Podcasts
THE RANVEER SHOW हिंदी - 245
सुप्रिया सुळे
Supriya sule - TRS hindi
SATRANGA
ANIMAL
Satranga
SAARI DUNIYA JALAA DENGE
ANIMAL
Duniya jala d
Spotify picks:
ZINDA BANDA
JAWAN
Home
Search
My Library

Conclusion:

Flutter’s widget architecture offers great flexibility for building complex UIs. Understanding key widgets and concepts is essential for effective Flutter development.