Experiment No: 2

Experiment No 2 2: To design Flutter UI by including common widgets.	
ROLL NO	**03**
NAME	**Ansari Mohammed Sharjeel**
CLASS	**D15B**
SUBJECT	**MAD & PWA Lab**
LO-MAPPED	

Experiment 2

Aim: To design Flutter UI by including common widgets.

Theory:

What are Flutter Widgets?

Flutter Widgets are essentially the building blocks of Flutter applications. They are the UI elements that are used to construct the user interface of the app. Widgets can be simple, like a button
or a text field, or they can be complex, like a page layout or an animation.

Properties of Widgets:

Immutable: Widgets in Flutter are immutable. Once created, their properties cannot be changed.
Instead, when a property needs to be updated, a new widget with the updated property is created.
Composable: Widgets in Flutter are composable, meaning they can be nested and combined to create
complex UIs. This allows for easy reuse of UI components and building blocks.
Declarative: Flutter uses a declarative approach to UI development, meaning that UIs are described in
terms of their desired state rather than in terms of the steps needed to achieve that state. This makes
it is easier to reason about and maintain UI code.
Stateful and Stateless: Widgets in Flutter can be either stateful or stateless. Stateless widgets are
immutable and do not have any internal state, while stateful widgets can change their appearance or
behavior in response to user interactions or other events.

1. **MaterialApp:** This widget represents the root of your Flutter application. It provides various configurations for your app, such as the title and theme.

2. **Scaffold**: Scaffold is a layout structure widget in Flutter that provides a framework to implement the basic material design layout of the application. It provides functionality like AppBar, FloatingActionButton, Drawer, etc. In this case, we're using an AppBar for the top bar, and a FloatingActionButton for adding new customers.

3. **AppBar:** AppBar is a material design widget used for providing the app bar title, leading icon (typically for navigation), and actions (typically for buttons like search or settings).

4. **Center:** Center is a layout widget used to center its child widget both vertically and horizontally within itself.

5. **Column:** Column is a layout widget used to arrange its children in a vertical array. It allows you to stack widgets vertically on top of each other.

6. **ListView.builder:** ListView.builder is used to efficiently create a scrollable list of widgets. It creates a scrollable list of widgets on-demand, which is useful for long lists where creating every widget upfront would be inefficient. It takes an itemBuilder function which generates the widgets for each item in the list.

7. **ListTile:** ListTile is a widget used to represent a single fixed-height row that contains one to three lines of text, optionally flanked by icons or other widgets, such as check boxes. It's commonly used in lists to represent items.

8. **FloatingActionButton:** FloatingActionButton is a material design floating action button. It's a circular button typically placed in the bottom right corner of the screen. It's used for primary actions in the application, such as adding a new item or initiating a new process.

These are some of the key widgets and concepts used in the provided Flutter UI code. By understanding and mastering these widgets, we can create complex and visually appealing user interfaces for your Flutter applications.

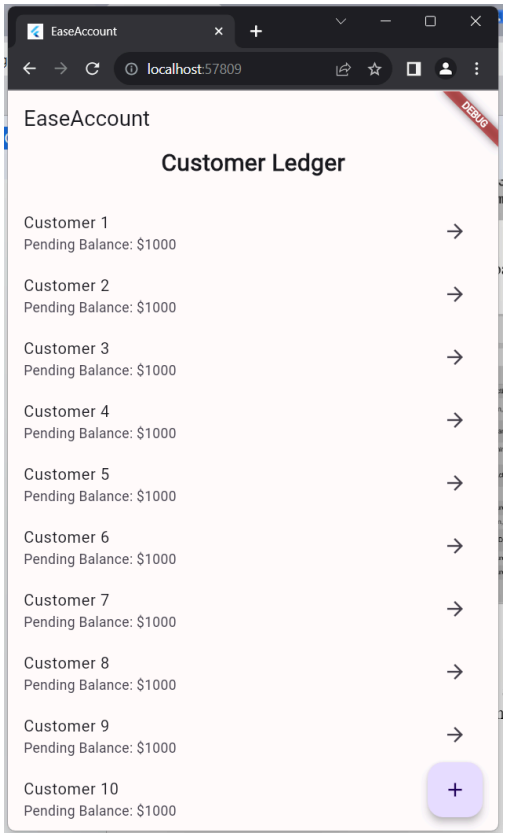
Code:

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

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

class EaseAccountApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'EaseAccount',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: HomePage(),
```

```
    );
  }
}

class HomePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('EaseAccount'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'Customer Ledger',
              style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 20),
            Expanded(
              child: ListView.builder(
                itemCount: 10, // Replace with actual count
                itemBuilder: (context, index) {
                  // Replace with actual ledger data
                  return ListTile(
                    title: Text('Customer ${index + 1}'),
                    subtitle: Text('Pending Balance: \$1000'),
                    trailing: IconButton(
                      icon: Icon(Icons.arrow_forward),
                      onPressed: () {
                        // Navigate to customer details page
                      },
                    ),
                  );
                },
              ),
            ),
          ],
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: () {
          // Add customer action
        },
        child: Icon(Icons.add),
```

```
        ),
      );
    }
}
```

Output:



Conclusion:

Hence, we understood how to design Flutter UI by including the widgets that have been used
in making the Login Screen of our application (Grocery App).