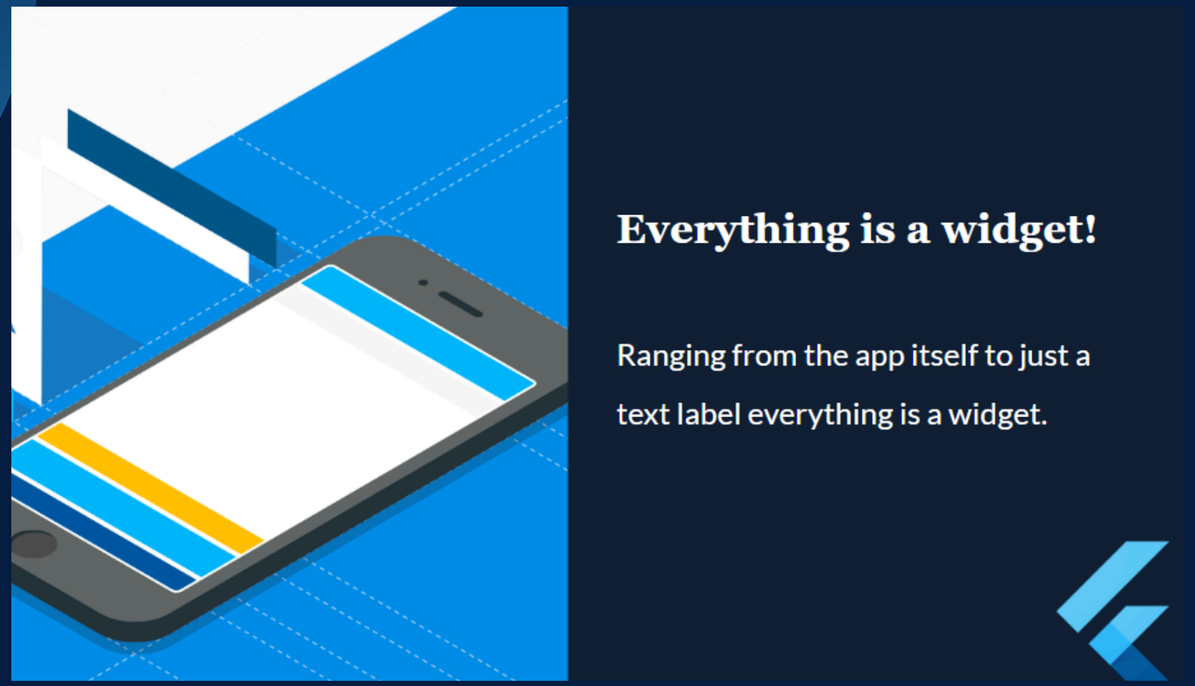
CS4039
SOFTWARE FOR
MOBILE DEVICES

ABDUL WAHAB USMANI

- Lead Software Engineer at Blink Co. Technologies (Jan 2023 - Present)
- Senior Software Engineer at VentureDive (Sep 2021- Dec 2022)
- Software Engineer at Aban Tech(July 2019 - Aug 2021)
- BS(CS) from FAST-NUCES

FLUTTER FRAMEWORK FUNDAMENTALS

FLUTTER UI



🖼️ **Declarative UI**

- Widgets describe how the UI should look for a given state.

🔄 **Reactive Updates**

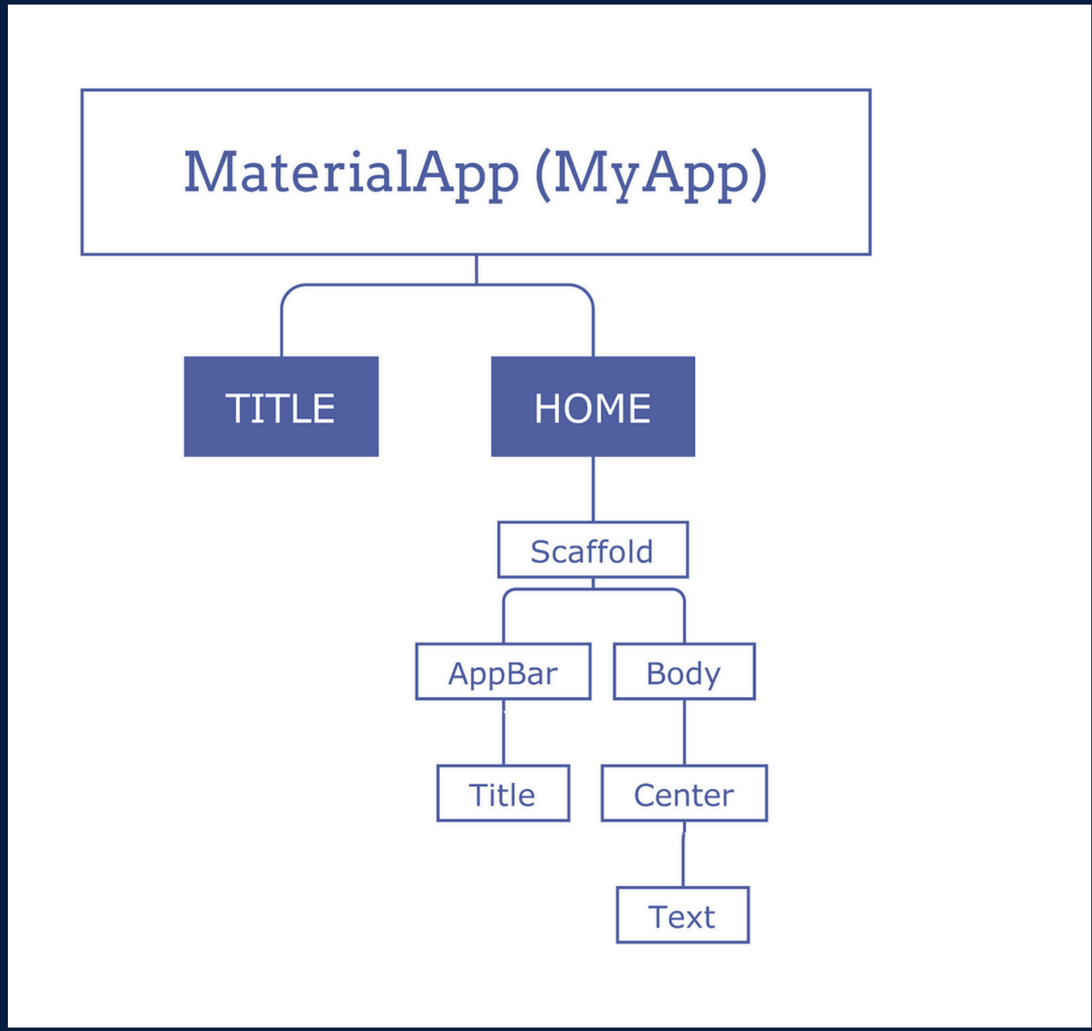
- When the state changes → Flutter rebuilds only affected widgets.

💡 **Inspired by React**

- Uses a reactive model for smooth, efficient updates.

CENTRAL IDEA IS THAT YOU BUILD YOUR UI OUT OF WIDGETS

They form a tree like hierarchy.



HELLO WORLD

- runApp() → Root of widget tree.
- Center → Aligns child in middle.
- Text → Displays styled text.

```
void main() {
  runApp(
    const Center(
      child: Text(
        'Hello, world!',
        textDirection: TextDirection.ltr,
        style: TextStyle(color: Colors.blue),
      ),
    ),
  );
}
```

STATELESS VS STATEFUL WIDGETS

Stateless Widget

- No internal state.
- Output depends only on input data.

Stateful Widget

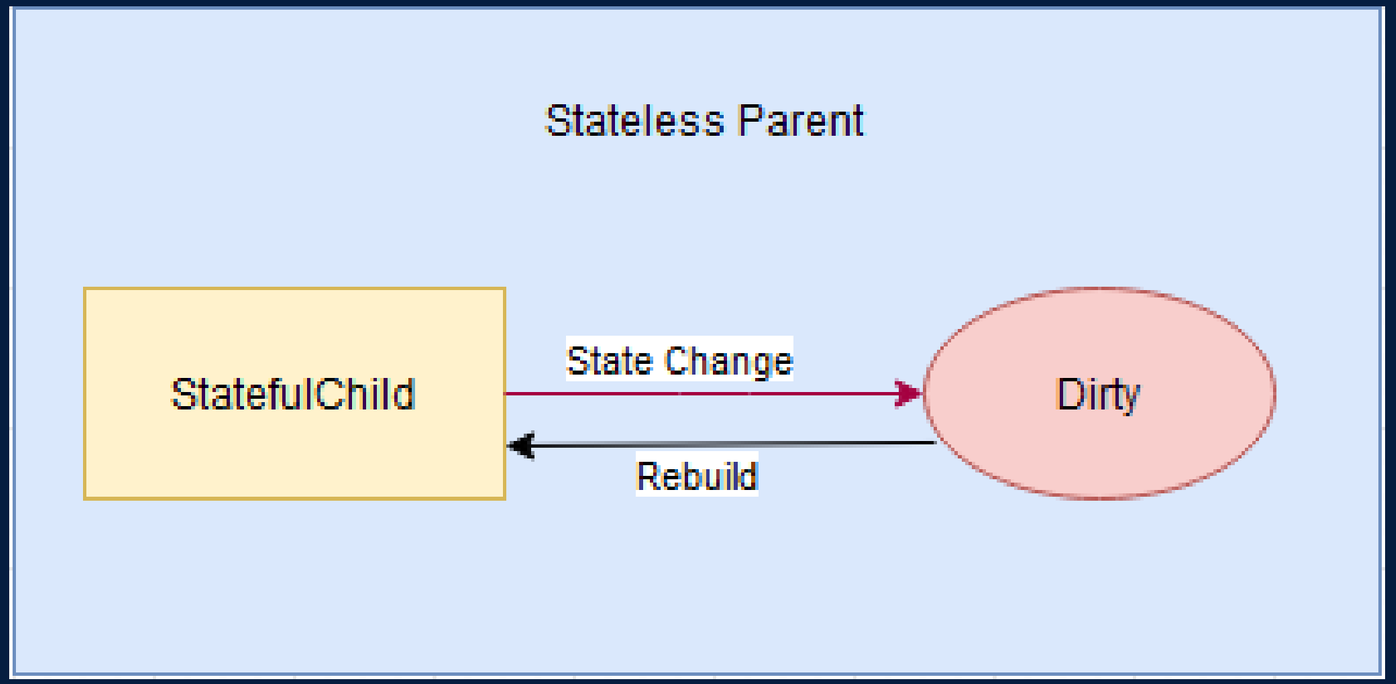
- Stores data that can change over time.
- Uses setState() to rebuild and update the UI.

Example: Counter App

Tap button → number increases instantly.

STATEFUL CHILD OF STATELESS PARENT

- Parent can be stateless while child can be stateful.
- Allows to rebuild only stateful child on state change.
- Use stateful custom widget to avoid making root widget stateful.



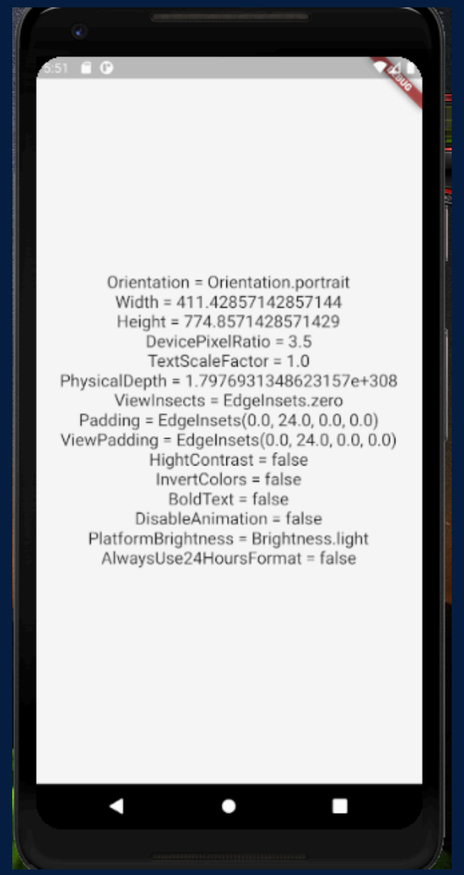
RESPONSIVE USER INTERFACE

FLUTTER SUPPORT IN MAKING RESPONSIVE UI

- MediaQuery
- LayoutBuilder
- OrientationBuilder
- Other responsive widgets

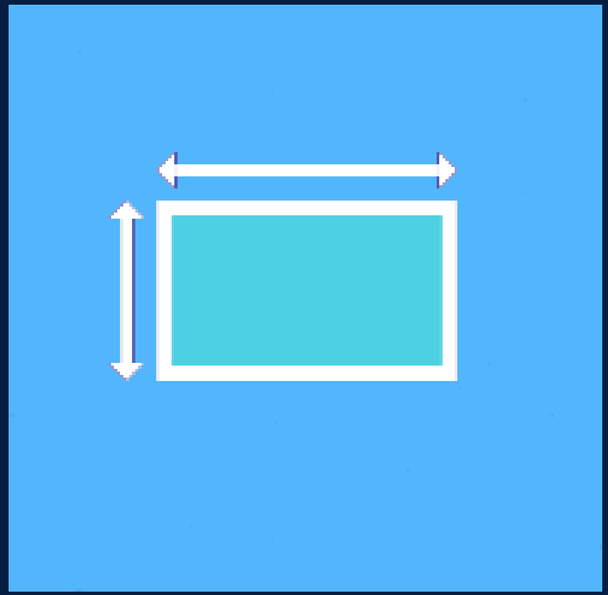
MediaQuery

- Provides device data using MediaQuery.of(context) to allow make decisions on layout
  - Orientation
  - Size (in logical pixels)
  - devicePixelRatio
  - viewInsets
- Rebuilds widget on data change.



LayoutBuilder

- Used to define child widget size based on parent widget's constraints
- Provides parent's widget constraints as **BoxConstraints**
- BoxConstraints contains
  - 📏 minWidth → Minimum allowed width of a widget
  - 📐 maxWidth → Maximum allowed width of a widget
  - 📏 minHeight → Minimum allowed height of a widget
  - 📐 maxHeight → Maximum allowed height of a widget



OrientationBuilder

- Builds a widget tree that can depend on the parent widget's orientation (distinct from the device orientation).
- Orientation

    Landscape: wider than tall

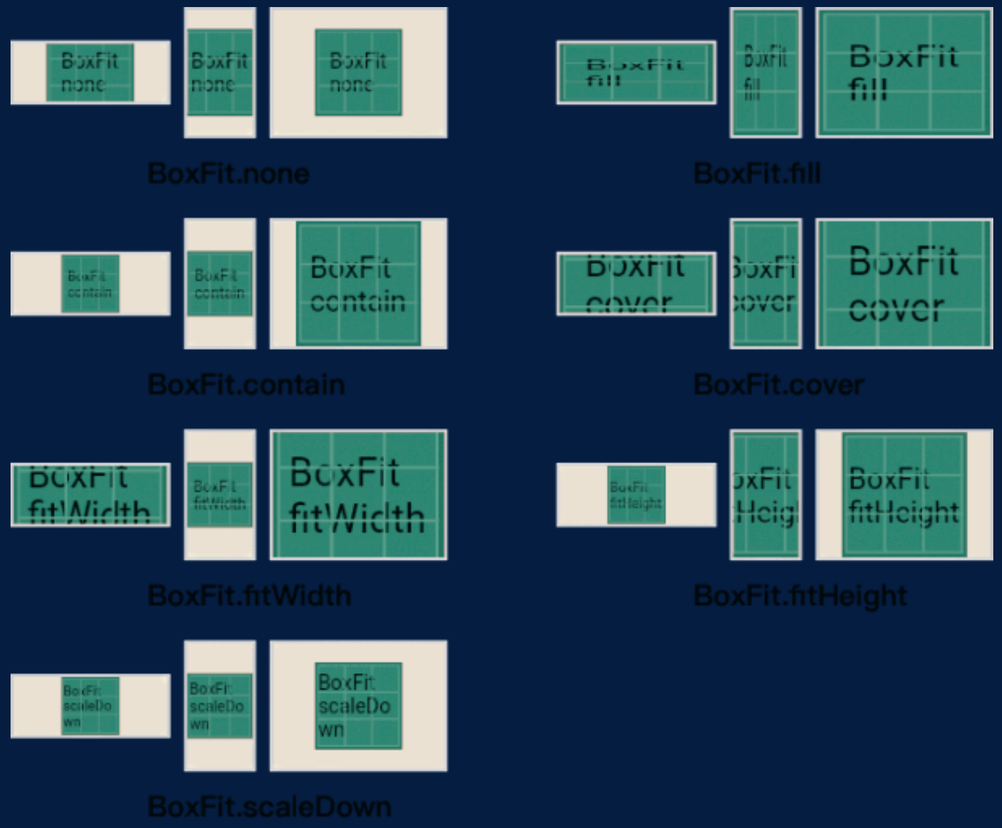
    Portrait: taller than wide

```
Orientation get orientation {
  return size.width > size.height ? Orientation.landscape : Orientation.portrait;
}
```

Other responsive widgets

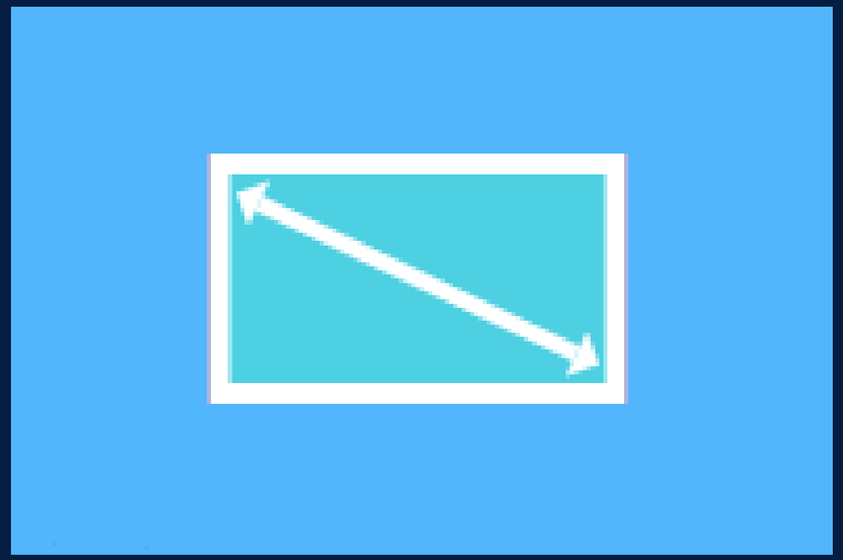
- FittedBox

  Scales and positions its child within itself according to fit .



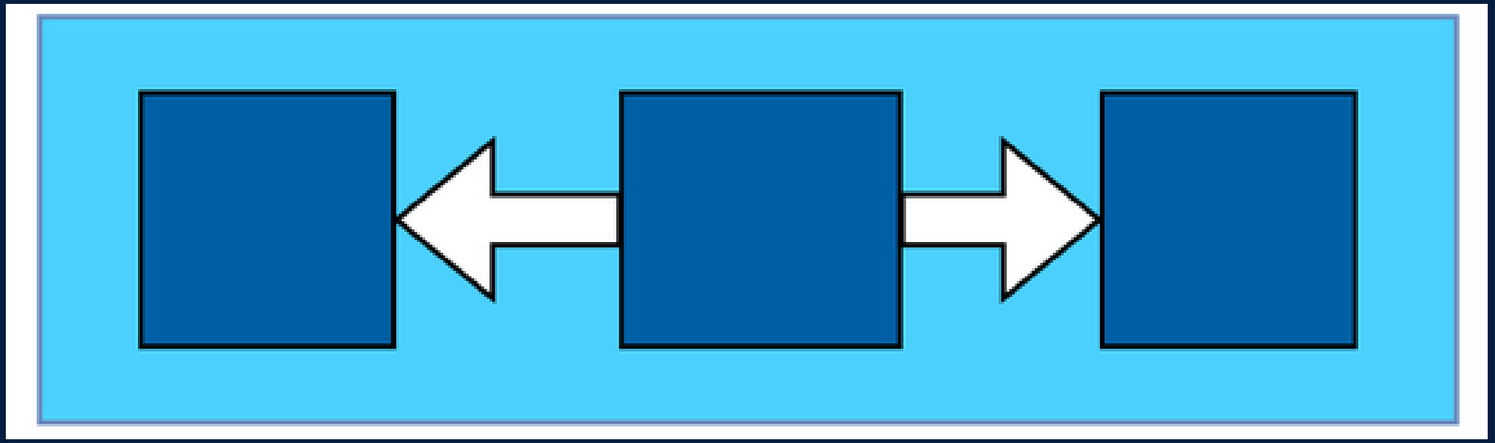
Other responsive widgets

- Aspect Ratio
  - A widget that attempts to size the child to a specific aspect ratio.
  - Used when width and height should be proportional instead of actual values.



Other responsive widgets

- Expanded
  - Helping you fill Rows and Columns.
  - Must be used under the **Flex** widget i.e. Row, Column, and Flex.
  - Use **flex** attribute to define to proportion of space the child should contain.
  - Gets the space after inflexible widgets are laid out

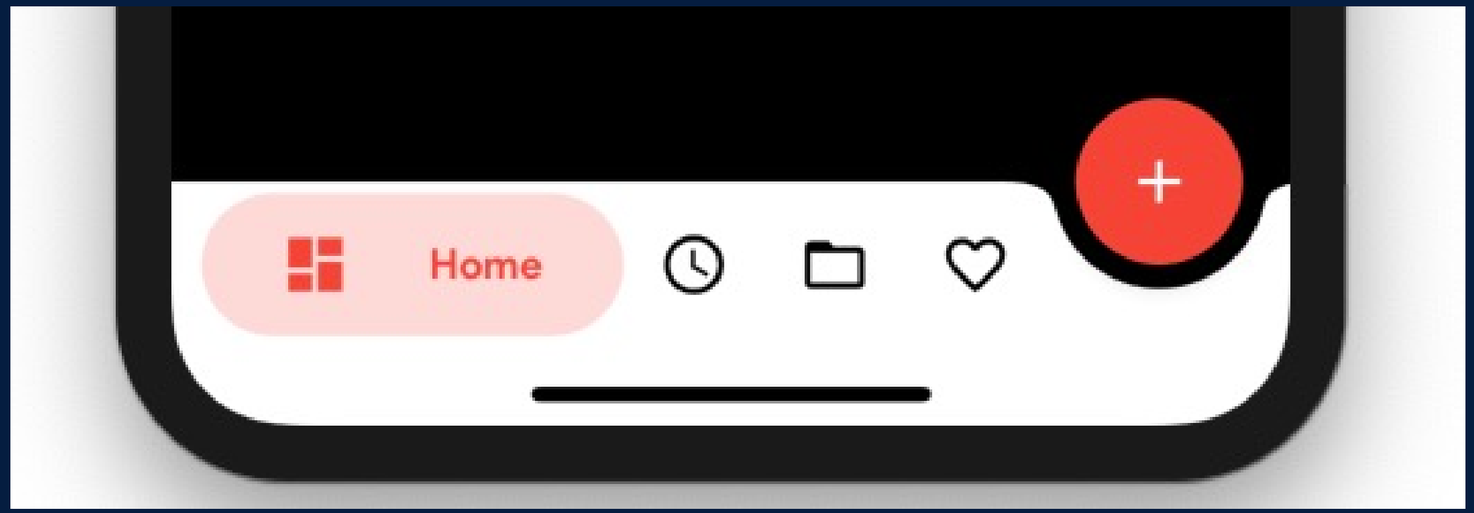


CONSTRUCTOR PARAMETERS

- Required parameters
  - Must be provided and in order.
  - Must be listed before optional parameters
  - Cannot have default value
- Optional parameters
  - Positional parameters []
  - Named Parameters {}
  - Can have default value

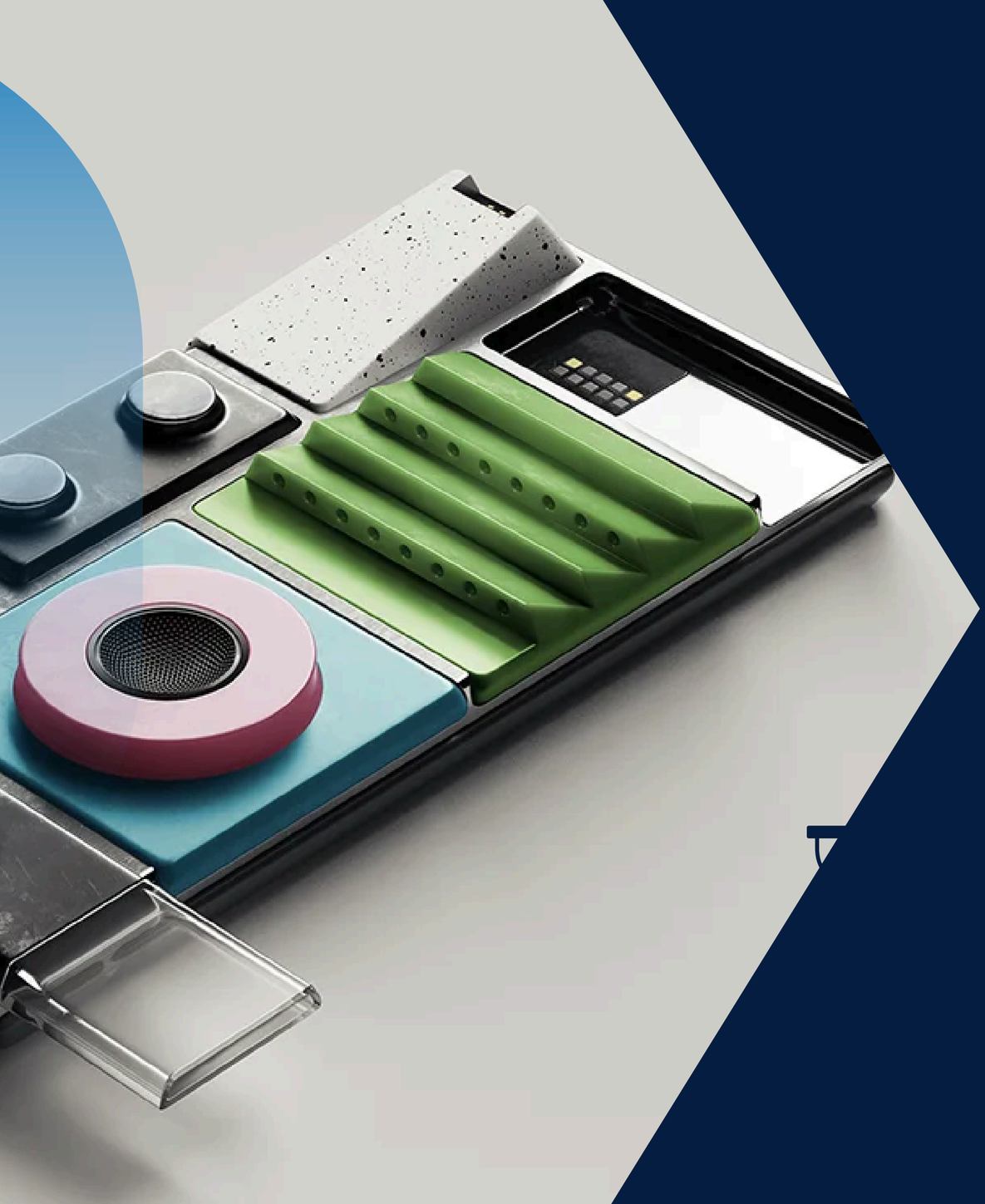
REUSABLE COMPONENTS

- Custom Widget
  - Composition over Inheritance



REUSABLE COMPONENTS

- Custom Painting
  - CustomPaint - A Flutter Widget
  - CustomPainter - uses Canvas to draw 2d shapes



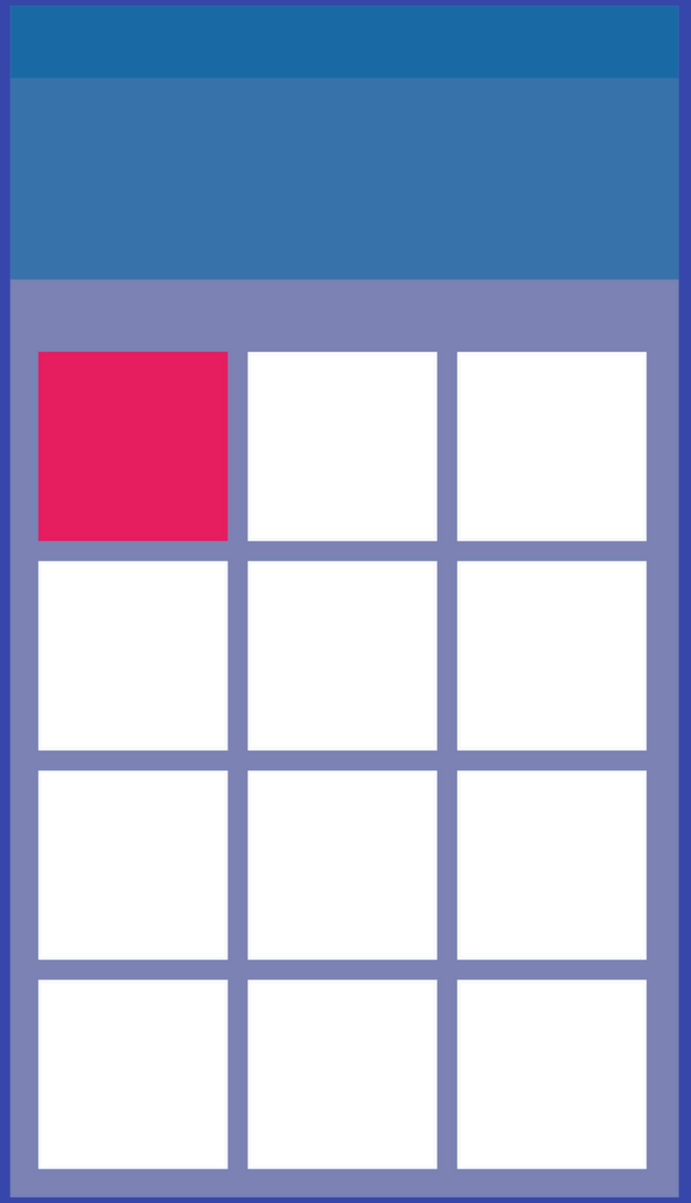
WIDGET CATALOG

Flutter’s collection of visual, structural, platform, and interactive widgets.

MULTI-CHILD WIDGETS

Row, Column, Stack etc...

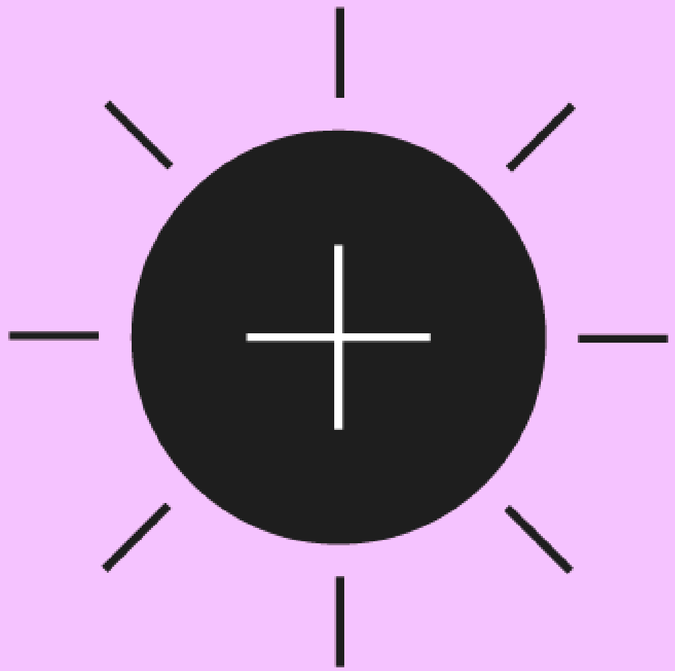
Have List<Widget> as children.



INTERACTIVE WIDGETS

Respond to events such as

touch, edit etc...



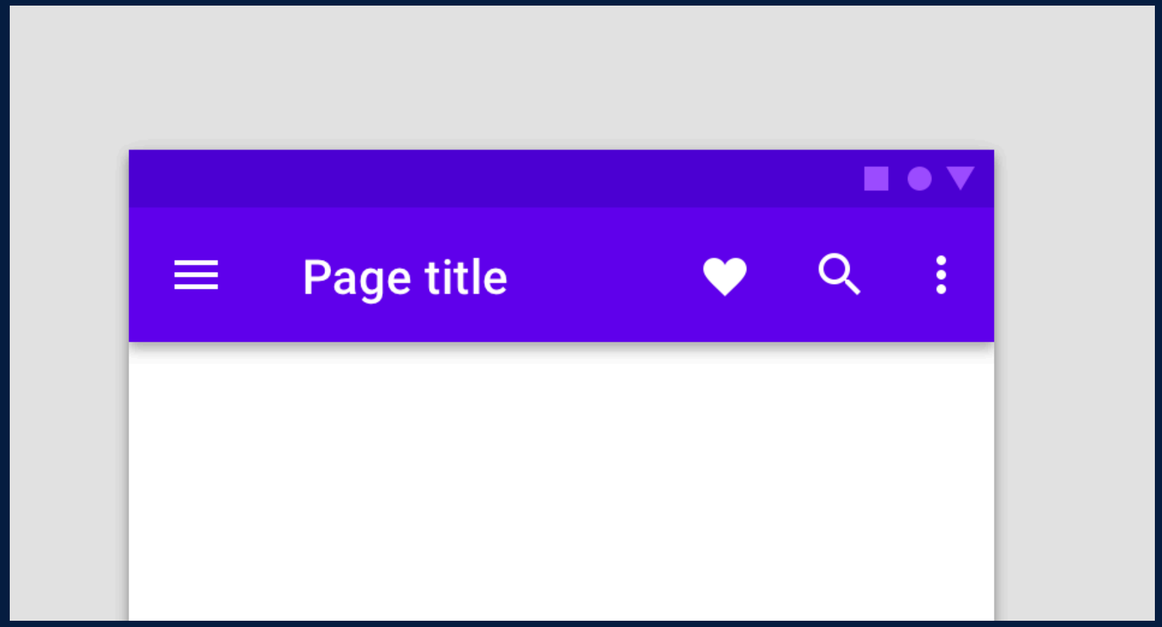
MATERIAL COMPONENTS

- Use MaterialApp for consistent theming.
- Common widgets:

  Scaffold (page structure)

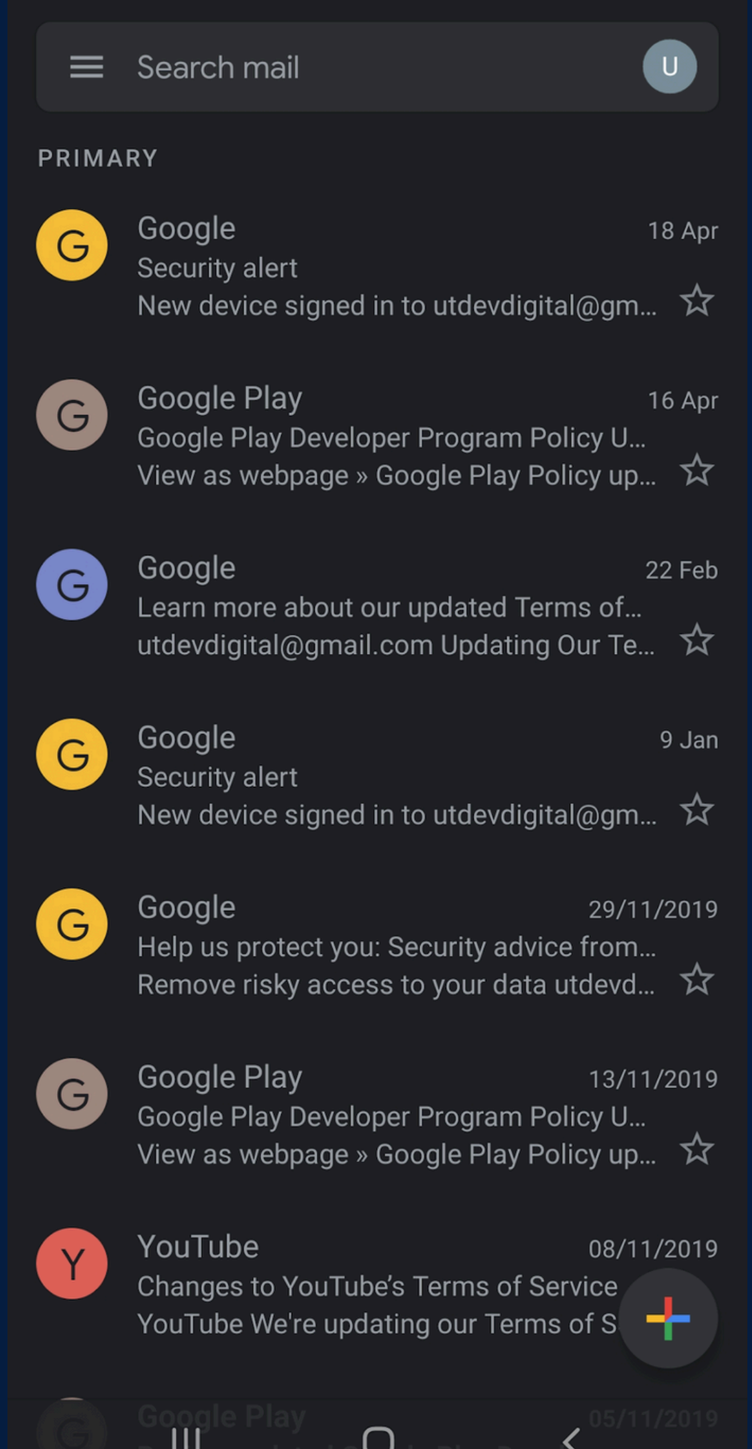
  AppBar (top navigation)

  FloatingActionButton (actions)



BUILDING UI LAYOUTS

GMAIL HOME


Search mail
U
PRIMARY
Google
18 Apr
Security alert
New device signed in to utdevdigital@gm...
Google Play
16 Apr
Google Play Developer Program Policy U...
View as webpage » Google Play Policy up...
Google
22 Feb
Learn more about our updated Terms of...
utdevdigital@gmail.com Updating Our Te...
Google
9 Jan
Security alert
New device signed in to utdevdigital@gm...
Google
29/11/2019
Help us protect you: Security advice from...
Remove risky access to your data utdevd...
Google Play
13/11/2019
Google Play Developer Program Policy U...
View as webpage » Google Play Policy up...
YouTube
08/11/2019
Changes to YouTube's Terms of Service
YouTube We're updating our Terms of S
Google Play
05/11/2019