Remember :

Flutter apps are made of Widgets.

You can declare stateful and stateless widgets.

The widgets can be declared in the same dart file as classes and variables or in different dart files.

The widgets are used by using the variable or by import the dart file and using the class.

Widgets can have multiple widgets inside it which are then used in different widgets creating a chain of flow between widgets.

The **routing** is done by push pop methods creating a stack of flow from one activity to another.

Generally, when we run the app we have a Default Process and a Main UI thread in main UI thread perform less timely tasks. For timely and complex separate from main thread create a Worker or Background thread like database ops, file ops, networking ops.

Since android is multi threaded threads can run parallel to each other but Dart is single threaded so the ops run in main thread only but we have APIs like Future, Async and Await to perform async ops

Future – to do something in future ex: you can define time after how many sec to run

Async – this keyword is defined with function to tell that it will perform async ops

Await - it is used along the declaration of future object to tell that wait and don’t execute next until the current runs

Instead of using Future API with async and await function it can be used with “Then” keyword where the commands are executed only when the Future returns something

Stateless widgets

Overrides **Build** method and returns widget

Used when UI depends on the information defined in widget itself

Stateful widgets

Classes that inherit stateful widget are immutable but the state is mutable

Override **createState** method and returns a state

When state changes **setState**() method is called telling the framework to redraw the widget

State of widget is the information that can be read synchronously when widget is built and might change during it’s lifetime

Used when UI needs to change dynamically

How to use stateful widgets:

A screenshot of a cell phone

Description automatically generated

Usign Databases(SQFLite):

It is a plugin, it is serverless and contain the data on the device

It’s dependency need to be declared in the pubspec.yaml

SQFLite deals in map objects so income and going transactions are map objects

We will need Model Class to represent data object and Database helper to perform CRUD ops