**KIẾN TRÚC ỨNG DỤNG FLUTTER**

Trong bài học này, chúng ta sẽ tìm hiểu kiến trúc của ứng dụng Flutter (architecture of the Flutter framework.)

Widgets

Khái niệm cốt lõi nhất trong Flutter framework đó là **Trong Flutter, mọi thứ đều là widget**. Widget (tiện tích) là thành phần giao diện cơ bản nhất tạo nên toàn bộ giao diện người dùng của ứng dụng.

Trong *Flutter*, bản thân chính ứng dụng đã là một widget. Mỗi ứng dụng chính là một  **top-level widget** và nó bao gồm một hoặc nhiều các widget con, mỗi widget này lại có thể bao gồm một hoặc nhiều widget con khác. Nhờ sự kết hợp linh hoạt này chúng ta có thể tạo ra bất kì ứng dụng phức tạp nào.

Ví dụ, chúng ta có thể nhìn vào cấu trúc widget của ứng dụng **hello world** (được học ở bài trước) thông qua sơ đồ dưới đây:

A picture containing text, screenshot, diagram, font

Description automatically generated

Giải thích sơ đồ trên

* *MyApp* là một widget được tạo ra bằng widget gốc của Flutter, ***MaterialApp***.
* *MaterialApp* có các thuộc tính của màn hình home và mô tả giao diện người dùng, nó lại được tạo ra bởi một widget khác, *MyHomePage*.
* *MyHomePage* được tạo bởi một widget gốc của flutter, *Scaffold*
* *Scaffold* có 2 thuộc tính – *body* và *appBar*
* *body* chứa giao diện chính còn *appBar* chứa phần đầu (header) của ứng dụng
* *Header UI* là một widget gốc của flutter, *AppBar* và *Body UI*  sử dụng *Center* widget.
* *Center* widget có một thuộc tính, *Child*, nó chứa phần nội dung chính là một *Text* widget

Gestures

Flutter widget hỗ trợ tương tác thông qua một widget đặc biệt gọi là *GestureDetector*. *GestureDetector* là một tiện ích không hiển thị trên giao diện nhưng có khả năng nắm bắt các thao tác của người dùng như nhấp, kéo, vuốt, chạm.... Phần lớn widget gốc của Flutter hỗ trợ tương tác giao diện thông qua *GestureDetector*. Chúng ta sẽ tìm hiểu chi tiết về gesture (cử chỉ) trong các bài học tiếp theo.

Khái niệm State

Flutter widgets quản lý các *State* (trạng thái) thông qua một widget đặc biệt *StatefulWidget*. Tất cả các Widget cần *StatefulWidget* để quản lý các trạng thái và kết nối với các widget khác. Flutter widgets là một dạng **reactive (**kỹ thuật reactive hay reactive programming là gì thì nó hơi khó giải thích, các bạn tìm đọc trên mạng nhé) gốc. Nó tương tự như **reactjs** và *StatefulWidget* sẽ tự động thay đổi giao diện hiển thị khi thay đổi trang thái. Sự thay đổi giao diện này được tối ưu bằng cách tìm kiếm sự khác biệt giữa UI widget cũ và mới sau đó chỉ vẽ lại những thay đổi cần thiết.

Layers

Một khái niệm quan trọng của Flutter framework đó là các thành phần sẽ được nhóm lại theo độ phức tạp và được sắp xếp rõ ràng trong các tầng có độ phức tạp giảm dần. Một **layer** (tầng,lớp) được tạo thành bằng việc sử dụng các class tiếp theo ngay canh nó. Top của tất cả các layer là các widget đặc biệt cho *Android* và *iOS*. Layer tiếp theo là widget gốc của flutter. Tiếp lữa là *Rendering layer*, đây là level thấp nhất trong việc sinh các thành phần của flutter app. Layer tiếp theo là nền tảng gốc hệ điều hành.

Tổng quan về layer của Flutter được mô tả trong sơ đồ dưới đây:

A picture containing text, receipt, screenshot, font

Description automatically generated

Tổng kết

Tổng kết những điểm chính về kiến trúc của Flutter

* Trong Flutter, tất cả đều quy về các widget, một widget phức hợp sẽ bao gồm các widget khác bên trong
* Các tính năng tương tác sẽ đước tích hợp bất cứ khi nào nhờ *GestureDetector* widget.
* Trạng thái của các widget được quản lý cập nhật bởi *StatefulWidget* widget.
* Flutter cung cấp thiết kế class để bất kỳ lớp nào có thể được lập trình tùy thuộc vào độ phức tạp của tác vụ