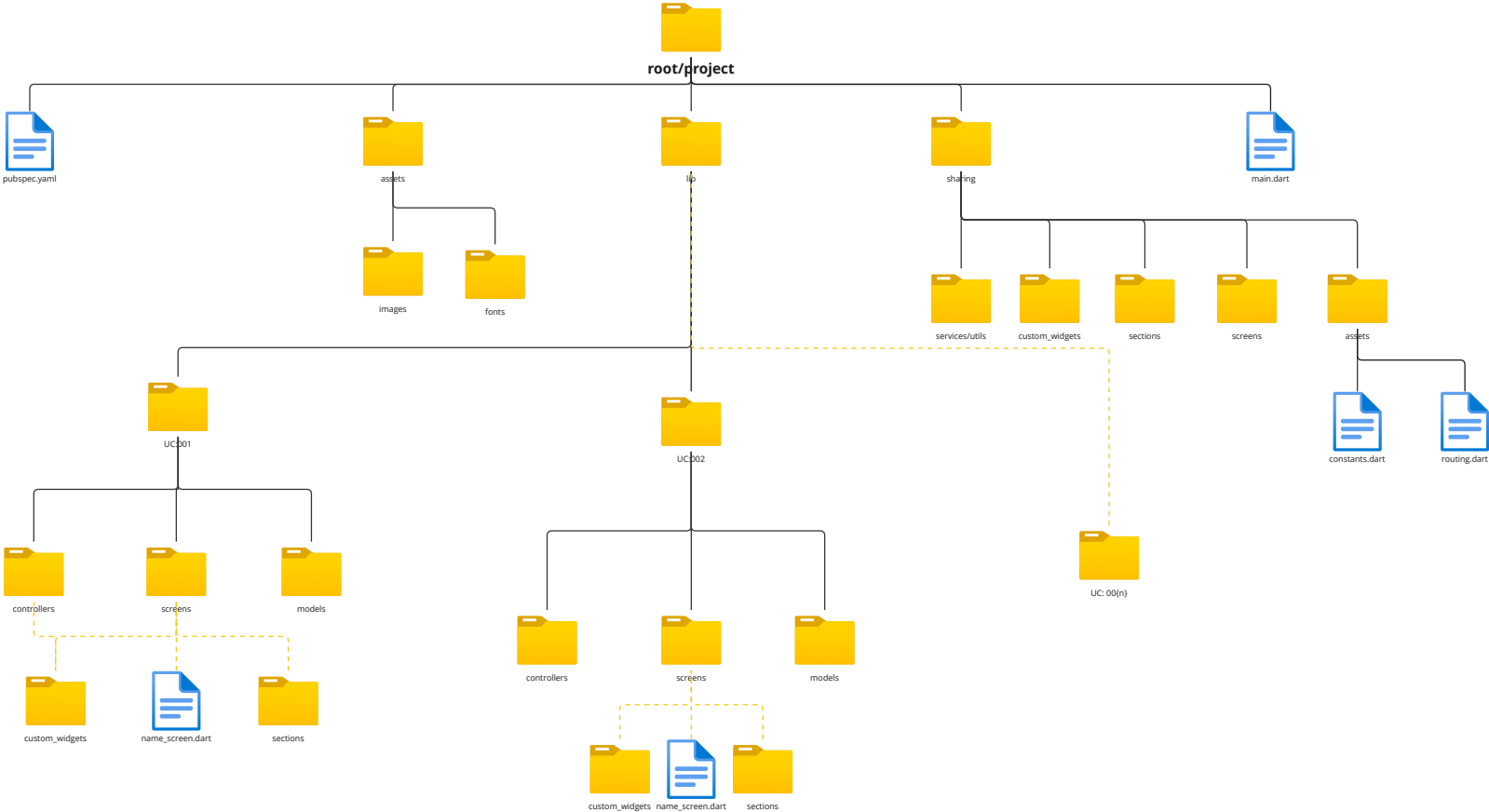


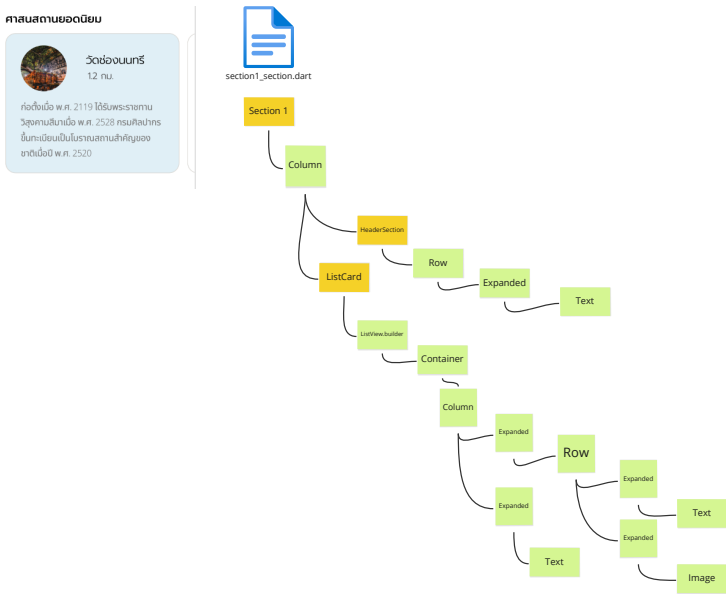
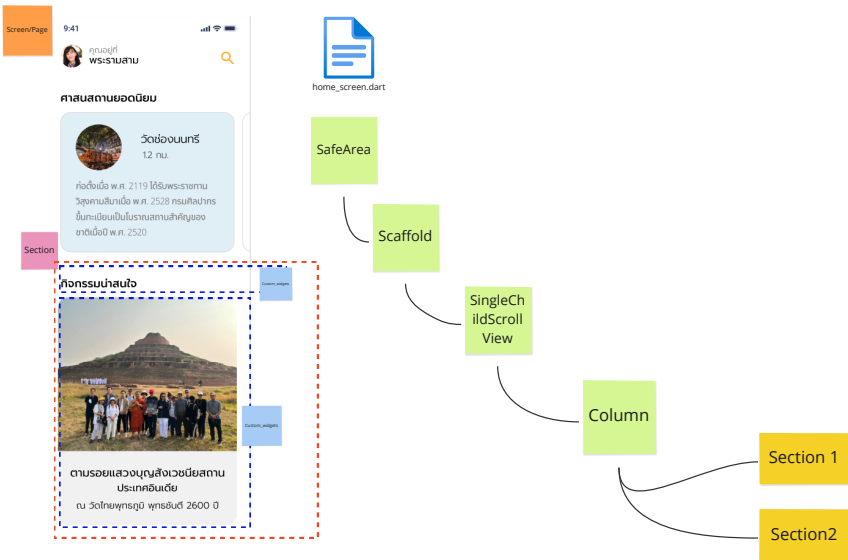
# Flutter Folder/Project Structure



# Frequently used Widget

Standard Widget	Reference link
SafeArea	<a href="https://api.flutter.dev/flutter/widgets/SafeArea-class.html">https://api.flutter.dev/flutter/widgets/SafeArea-class.html</a>
SingleChildScrollView	<a href="https://api.flutter.dev/flutter/widgets/SingleChildScrollView-class.html">https://api.flutter.dev/flutter/widgets/SingleChildScrollView-class.html</a>
Scaffold	<a href="https://api.flutter.dev/flutter/material/Scaffold-class.html">https://api.flutter.dev/flutter/material/Scaffold-class.html</a>
Container	<a href="https://api.flutter.dev/flutter/widgets/Container-class.html">https://api.flutter.dev/flutter/widgets/Container-class.html</a>
SizeBox	<a href="https://api.flutter.dev/flutter/widgets/SizeBox-class.html">https://api.flutter.dev/flutter/widgets/SizeBox-class.html</a>
Text	<a href="https://api.flutter.dev/flutter/widgets/Text-class.html">https://api.flutter.dev/flutter/widgets/Text-class.html</a>
Image	<a href="https://api.flutter.dev/flutter/widgets/Image-class.html">https://api.flutter.dev/flutter/widgets/Image-class.html</a>
SvgPicture	<a href="https://pub.dev/documentation/flutter_svg/latest/svg/SvgPicture-class.html">https://pub.dev/documentation/flutter_svg/latest/svg/SvgPicture-class.html</a>
ElevatedButton	<a href="https://api.flutter.dev/flutter/material/ElevatedButton-class.html">https://api.flutter.dev/flutter/material/ElevatedButton-class.html</a>
GestureDetector	<a href="https://api.flutter.dev/flutter/widgets/GestureDetector-class.html">https://api.flutter.dev/flutter/widgets/GestureDetector-class.html</a>
AppBar	<a href="https://api.flutter.dev/flutter/material/AppBar-class.html">https://api.flutter.dev/flutter/material/AppBar-class.html</a>
ListView.builder	<a href="https://api.flutter.dev/flutter/widgets/ListView-class.html">https://api.flutter.dev/flutter/widgets/ListView-class.html</a>
Expanded	<a href="https://api.flutter.dev/flutter/widgets/Expanded-class.html">https://api.flutter.dev/flutter/widgets/Expanded-class.html</a>
Flexible	<a href="https://api.flutter.dev/flutter/widgets/Flexible-class.html">https://api.flutter.dev/flutter/widgets/Flexible-class.html</a>
Layout Widget	Reference link
AppBar	<a href="https://api.flutter.dev/flutter/material/AppBar-class.html">https://api.flutter.dev/flutter/material/AppBar-class.html</a>
Column	<a href="https://api.flutter.dev/flutter/widgets/Column-class.html">https://api.flutter.dev/flutter/widgets/Column-class.html</a>

# Widget Composition Tree



# Flutter Code Structure

```
import 'package:buddhism/get_recommended_route/screens/homepage_screen/sections/profileheader.dart';
import
'package:buddhism/get_recommended_route/screens/homepage_screen/sections/recommendactivities.dart';
import
'package:buddhism/get_recommended_route/screens/homepage_screen/sections/recommendddonation.dart';
import
'package:buddhism/get_recommended_route/screens/homepage_screen/sections/recommendtemple.dart';
import 'package:flutter/material.dart';
```

→ All necessary dependencies part

```
class HomeScreen extends StatelessWidget {
  const HomeScreen({Key? key}) : super(key: key);
  Bool isPressed = false;
```

→ Necessary variables' declaration parts

```
@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      body: SingleChildScrollView(
        child: Padding(
          padding:
            const EdgeInsets.symmetric(horizontal: 30.0, vertical: 30.0),
          child: Column(
            children: const [
              ProfileHeader(),
              SizedBox(height: 50,),
              RecommendTemple(),
              SizedBox(height: 50,),
              RecommendActivities(),
              SizedBox(height: 50,),
              RecommendDonation(),
            ],
          ),
        ),
      ),
    ),
  );
}
```

→ Flutter Widget coding part (Corresponding with Widget Composition Tree)

# Structure for Clean code structure

## Coding Standard: Flutter (BBL Team)

### 1) Naming Convention

- Classes, enums, typedefs, and extensions name must be in **\*\*UpperCamelCase\*\***.
- For CustomWidgets : ClassName (UpperCamelCase) + BBL  
Example : CardFarBBL
- Packages ,Libraries, Directories, and source files name must be in **\*\*snake\_case\*\*** (lowercase\_with\_underscores) with type of file
- For non-CustomWidgets  
Example : homepage\_screen
- For customWidgets : Instead of using snake\_case, BBL as a suffix will be used.  
Example : cardFarBBL
- Variables, constants, parameters, and named parameters must be in **\*\*lowerCamelCase\*\***.
- Attribute constants must be in **\*\*lowerCamelCase\*\*** and start with **\*\*k\*\***.
- Private variable must start with **\*\*underscore(\_)\*\***.

### 2) Use only absolute imports for files in lib/

```
...  
//Don't  
import '../.../constants.dart';  
  
//Do  
import 'package:food/screen/dishDetail.dart';  
...
```

### 3) Variable Declaration

- การประกาศตัวแปรทั้งหมดให้อยู่ส่วนบนของหน้า
- ใน body ไม่ต้องใส่ **\*\*const\*\*** นำหน้าตัวแปร

### 4) Standard Widgets

- การสร้าง Screen ให้ใช้ **\*\*Scaffold\*\***
- การสร้างหน้าที่มี Scroll ให้ใช้ **\*\*SingleChildScrollView\*\***
- ให้ใช้ **\*\*Column\*\*** สำหรับแบ่งเนื้อหาในแนวตั้ง
- ถ้ามีการซ้อนกันของ Widget ให้ใช้ **\*\*Stack\*\***
- ใน Stack ให้ใช้ **\*\*Positioned\*\*** ในการวางซ้อนตำแหน่งต่างๆ
- ให้ใช้ **\*\*Image.asset\*\*** สำหรับรูปที่นำเข้ามาจากเครื่อง และ **\*\*Image.network\*\*** สำหรับรูปที่นำเข้ามาจากอินเทอร์เน็ต
- ใน Column ให้ใช้แค่ **\*\*Container\*\***, **\*\*Expanded\*\*** และ **\*\*Row\*\***
- การสร้าง card หรือปุ่มที่ไม่ใช่ Text, Icon และ Image ให้ใช้ **\*\*GestureDetector\*\***
- การสร้างปุ่มที่มีไอคอนให้ใช้ **\*\*ElevatedButton\*\***
- ให้ใช้ **\*\*ClipRRect\*\*** สำหรับปรับแต่งรูปร่างของกรอบรูป (ใช้เมื่อจำเป็นเท่านั้น)
- การสร้าง List ที่เราต้องการ slide (ซ้าย-ขวา หรือ บน-ล่าง) ให้ใช้ **\*\*ListView.builder()\*\***

### 5) Function

- ใน 1 function ต้องทำงานไม่เกิน 1 อย่าง
- ใน 1 function รับ arguments ได้ไม่เกิน 3 ตัว
- ไม่ควรใช้ flag argument (argument ที่เป็น boolean)

### 6) Comment

- ไม่ควรมี comment code ที่ไม่ได้ใช้งาน
- เขียน comment เมื่อจำเป็นต้องอธิบาย function ที่มีการทำงานที่ซับซ้อน
- สำหรับ function ที่ไม่ซับซ้อนให้ตั้งชื่อ function ให้สื่อถึงการทำงานแผนการเขียน comment
- เขียน **\*\*TODO comment\*\*** สำหรับงานส่วนที่ยังไม่เสร็จ

### 7) Formatting

- ถ้า statement ยาวไม่เกิน 1 บรรทัด ไม่ต้องใส่ **\*\*comma(",")\*\*** หลัง () หรือ []
- ถ้า statement ยาวมากกว่า 1 บรรทัด ให้ใส่ **\*\*comma(",")\*\*** หลัง () หรือ [] ทุกครั้ง

### 8) Object and Data Structure

- ถ้ามี function ที่ทำงานเหมือนหรือคล้ายกันในแต่ละไฟล์ ให้ย้ายไปรวมกันเป็นไฟล์ใหม่ที่มีรูปแบบ **\*\*Polymorphic\*\*** หรือ **\*\*Procedural\*\***
- ถ้าต้องการสร้าง function ใหม่ แต่ฟังก์ชันนั้นทำงานเหมือนหรือคล้ายกัน function ที่มีอยู่แล้วให้ใส่รูปแบบ Polymorphic หรือ Procedural ให้ดีหรือ inherit function นั้นมาใช้แทน

### 9) Naming

- ตั้งชื่อตัวแปรให้มีความหมายและสื่อความหมายชัดเจน
- ตั้งชื่อ function ให้มีความหมายและสื่อความหมายถึงการทำงานชัดเจน
- ตั้งชื่อตัวแปรให้สามารถอ่านออกเสียงได้
- ควรตั้งชื่อตัวแปรให้สื่อถึงตัวชี้วัดและหน่วย
- ไม่ควรตั้งชื่อตัวแปรที่มีความหมายซ้ำซ้อนกัน
- ไม่ควรตั้งชื่อ class เป็นคำกริยา
- ควรตั้งชื่อ function ให้มีคำกริยารวมอยู่ด้วย
- ควรตั้งชื่อตัวแปรให้ context รวมอยู่ในชื่อด้วย

### 10) Others

- ให้ใช้ **\*\*EdgeInsets.all\*\*** เมื่อระยะซ้าย, ขวา, บนและล่างเท่ากัน และ **\*\*EdgeInsets.symmetric\*\*** เมื่อระยะซ้ายเท่ากันขวา หรือบนเท่ากันล่าง นอกเหนือจากนั้นให้ใช้ **\*\*EdgeInsets.only\*\***
- สร้าง **\*\*Container\*\*** แต่อย่าทำเป็น
- ให้ใช้ **\*\*padding\*\*** แทน margin