**BÁO CÁO TUẦN 9**

**Họ và tên:** Nguyễn Kỳ Anh  
**MSSV:** 20225793  
 **GR1**  
**Giáo viên hướng dẫn:** Nguyễn Đức Toàn

# RadioListTitle

## Định nghĩa

* RadioListTitle là một Widget trong flutter, kết hợp một radio button với tiêu đề, phụ đề, và các thành phần giao diện khác để tạo ra một mục chọn trong danh sách các lựa chọn.

## Cấu trúc và thành phần

* Value: giá trị đại diện cho RadioListTitle, dung để xác định lựa chọn.
* groupValue: giá trị hiện tại của nhóm radio
* onChanged: Hàm callback được gọi khi người dung chọn RadioListTitle, cập nhật groupValue.
* Title: Tiêu đề chính thức của mục, thường là một widget Text.
* Secondary: Widget hiển thị bên cạnh (vd: biểu tượng).
* activeColor: Màu của radio button khi được chọn( mặc định là màu xanh).
* controlAffinity: Quy định vị trí của radio button (bên phải hoặc trái tiêu đề).

Vd:

return MaterialApp(

home: Scaffold(

appBar: AppBar(title: Text('Ví dụ RadioListTile')),

body: Column(

children: [

RadioListTile(

title: Text('Lựa chọn 1'),

value: 'option1',

groupValue: \_selectedOption,

onChanged: (value) {

setState(() {

\_selectedOption = value.toString();

});

},

),

RadioListTile(

title: Text('Lựa chọn 2'),

value: 'option2',

groupValue: \_selectedOption,

onChanged: (value) {

setState(() {

\_selectedOption = value.toString();

});

},

),

],

),

),

);

}

}

# CheckBox

## Định nghĩa

* CheckBox: Là một Widget trong flutter dung để tạo một ô tích cho phép người dung chọn hoặc không.
* CheckBox cho phép chọn nhiều tùy chọn độc lập.

## Cấu trúc và thành phần

* Value(bắt buộc): Gía trị Boolean (true hoặc false) xác định trạng thái của checkbox (chọn hoặc không).
* onChanged (bắt buộc): Hàm callback được gọi khi người dung thay đổi trạng thái của checkbox, nhận giá trị boolean mới.
* activeColor: Màu của checkbox khi được chọn
* checkColor: màu của dấu kiểm trong checkbox.
* Trisate: nếu true, checkbox có thể có trạng thái thứ ba (null) ngoài true và false.
* Side: Tùy chỉnh viền của checkbox(vd: màu sắc, độ dày).

VD:

return MaterialApp(

home: Scaffold(

appBar: AppBar(title: Text('Ví dụ Checkbox')),

body: Column(

children: [

CheckboxListTile(

title: Text('Tùy chọn 1'),

value: \_isChecked1,

onChanged: (bool? value) {

setState(() {

\_isChecked1 = value ?? false;

});

},

activeColor: Colors.blue,

checkColor: Colors.white,

),

CheckboxListTile(

title: Text('Tùy chọn 2'),

value: \_isChecked2,

onChanged: (bool? value) {

setState(() {

\_isChecked2 = value ?? false;

});

},

activeColor: Colors.blue,

checkColor: Colors.white,

),

],

),

),

);

}

}

# DatePicker

## Định nghĩa

* DatePicker là một widget trong flutter cho phép người dung chọn một ngày từ giao diện lịch (calendar) hoặc nhập trực tiếp.
* Trong Flutter, DatePicker thường được hiển thị thông qua hàm showDatePicker, tạo một hộp thoại chọn ngày.

## Cấu trúc và thành phần

* Context( bắt buộc): BuildContext để hiển thị hộp thoại.
* initalDate: Ngày mặc định được chọn khi DatePicker mở (thường là ngày hiện tại).
* DateTime.now(): ngày hôm nay.
* firstDate: Ngày sớm nhất mà người dung có thể chọn
* lastDate: Ngày muộn nhất mà người dung có thê chọn.
* currentDate: Ngày hiện tại dùng để làm nổi bật lên lịch.
* initialDatePickerMode: Chế độ hiển thị ban đầu (day hoặc year).
* onDateChanged; Hàm callback khi người dùng thay đổi ngày.
* onConfirm: Hàm callback khi người dung xác nhận ngày.
* Locate: định dạng ngôn ngữ và khu vực (VD: Locale(‘vi’,’VN’) cho tiếng Việt).
* Builder: Tùy chỉnh giao diện của DatePicker.

VD:

class MyApp extends StatefulWidget {

@override

\_MyAppState createState() => \_MyAppState();

}

class \_MyAppState extends State<MyApp> {

DateTime? \_selectedDate;

Future<void> \_selectDate(BuildContext context) async {

final DateTime? picked = await showDatePicker(

context: context,

initialDate: DateTime.now(),

firstDate: DateTime(2000),

lastDate: DateTime(2030),

locale: Locale('vi', 'VN'),

);

if (picked != null && picked != \_selectedDate) {

setState(() {

\_selectedDate = picked;

});

}

}

@override

Widget build(BuildContext context) {

return MaterialApp(

home: Scaffold(

appBar: AppBar(title: Text('Ví dụ DatePicker')),

body: Center(

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Text(

\_selectedDate == null

? 'Chưa chọn ngày'

: 'Ngày đã chọn: ${\_selectedDate!.toString().substring(0, 10)}',

),

ElevatedButton(

onPressed: () => \_selectDate(context),

child: Text('Chọn ngày'),

),

],

),

),

),

);

}

}

# UI quản lý.

* Gồm 1 class chính “QuanLy” liên kết các class phụ để tạo thành giao diện quản lý. Chia các widget thành các class giúp dễ quản lý các phần lớn nhỏ trong giao diện.

## Class QuanLy

* Import các class trong giao diện
* Hàm initState() quản lý data và update trên giao diện khi người dung ấn trọn ngày muốn xem.
* Hàm \_updateData(): giúp updata các dữ liệu trong giao diện được khai báo trong class database.
* Sử dụng Colum danh sách sắp xếp theo cột để thêm từng class giao diện của từng phần card vào trong class chính.
* SizedBox(height: 10): cách đều mỗi một giao diện con 10px.
* bottomNavigationBarItem: dung để thêm các Icon thanh appBar dưới màn hình.

## Class appBar

* Image.asset: thêm logo cho giao diện.
* **leading:**
* Hiển thị logo từ 'assets/images/Logo.jpg'.
* Kích thước: width: 200, height: 200 (có thể gây tràn khung).
* Padding: 10.0.
* **title:**
* Văn bản: "GoodFood".
* Kiểu chữ: đậm, màu đỏ.
* **actions:**
* Nút thông báo: IconButton với Icons.notifications, onPressed rỗng.
* Nút trợ giúp: IconButton với Icons.help, mở hộp thoại HuongDan.
* **flexibleSpace**:
* Gradient nền: từ xanh lam (0xFF4FC3F7) đến xanh nhạt (0xFFB2EBF2).
* backgroundColor: trong suốt, elevation: 0.
* **bottom**:
* PreferredSize cao 50.0.
* Chứa DropdownButton:
  + Hiển thị danh sách ngày từ ngay.
  + Giá trị hiện tại: selectedValue.
  + Callback: onChanged khi chọn giá trị mới.
  + Giao diện: nền trắng, chữ đen (cỡ 16), gạch chân xám, mũi tên xám.

## Class Card\_DoanhThu

* Card:
* Màu nền: Colors.white.withOpacity(0.9) (trắng mờ).
* Viền: Bo tròn (borderRadius: 10).
* Độ nổi: 4.
* **Tiêu đề**:
* Văn bản: "Doanh thu".
* Kiểu chữ: Cỡ 18, đậm (FontWeight.bold).
* **TextButton**:
* 3 nút: "Giờ" (màu xanh), "Ngày", "Thứ" (màu đen).
* onPressed: Hiện rỗng, chưa có chức năng.
* Căn đều khoảng cách (MainAxisAlignment.spaceBetween).
* **Thông báo chưa có dữ liệu**:
* Văn bản chính: "Chưa có dữ liệu" (cỡ 16, đậm, màu xám).
* Văn bản phụ: "Doanh thu cửa hàng của bạn sẽ hiển thị tại đây" (cỡ 14, màu xám).
* Căn giữa (CrossAxisAlignment.center).
* **Khoảng cách:**
* Sử dụng MediaQuery.of(context).size.height \* 0.1 để tạo khoảng cách động theo chiều cao màn hình.

## Card\_SoLuongkh

* **Card**:
* Màu nền: Colors.white.withOpacity(0.9) (trắng mờ).
* Viền: Bo tròn (borderRadius: 10).
* Độ nổi: 10 (giống CardHangbanchay, CardHangtonkho, CardHuymon).
* **Tiêu đề**:
* Văn bản: "Số lượng khách hàng".
* Kiểu chữ: Cỡ 18, đậm (FontWeight.bold).
* **Thông báo chưa có dữ liệu**:
* Văn bản chính: "Chưa có dữ liệu" (cỡ 16, đậm, màu xám).
* Văn bản phụ: "Số lượng khách hàng sẽ hiển thị tại đây" (cỡ 16, màu xám).
* Căn giữa (CrossAxisAlignment.center).
* **Khoảng cách**:

SizedBox với MediaQuery.of(context).size.height \* 0.1 tạo khoảng cách động.

## Card\_HangTonKho

* **Card**:
* Màu nền: Colors.white.withOpacity(0.9) (trắng mờ).
* Viền: Bo tròn (borderRadius: 10).
* Độ nổi: 10 (giống CardHangbanchay, cao hơn CardDoanhthu).
* **Tiêu đề**:
* Văn bản: "Tồn kho".
* Kiểu chữ: Cỡ 20, đậm (FontWeight.bold).
* **Hàng (Row) 1**:
* Nhãn: "Tổng tiền hàng" (cỡ 18, đậm).
* Giá trị: data['inventoryTotal'] (cỡ 16, đậm, màu hồng đậm Color.fromARGB(255, 255, 64, 134)).
* Căn đều khoảng cách (MainAxisAlignment.spaceBetween).
* **Hàng (Row) 2**:
* Nhãn: "Số lượng hàng" (cỡ 16, đậm).
* Giá trị: data['inventoryQuantity'] (cỡ 16, đậm, màu hồng tươi Color.fromARGB(255, 255, 7, 106)).
* Căn đều khoảng cách.
* **Divider**: Thanh phân cách giữa hai hàng thông tin.

## Card\_HangBanChay

* + **Card**:
    - Màu nền: Colors.white.withOpacity(0.9) (trắng mờ).
    - Viền: Bo tròn (borderRadius: 10).
    - Độ nổi: 10 (cao hơn CardDoanhthu).
  + **Tiêu đề**:
    - Văn bản: "Hàng bán chạy".
    - Kiểu chữ: Cỡ 18, đậm (FontWeight.bold).
  + **ElevatedButton**:
    - Nút "Theo doanh thu": Nền xanh, chữ trắng, viền bo tròn (radius 20).
    - Nút "Theo số lượng": Nền trắng, chữ đen, viền bo tròn (radius 20).
    - onPressed: Hiện rỗng, chưa có chức năng.
    - Căn trái (MainAxisAlignment.start), khoảng cách ngang 20.0.
  + **Thông báo chưa có dữ liệu**:
    - Văn bản chính: "Chưa có dữ liệu" (cỡ 18, đậm, màu xám).
    - Văn bản phụ: "Top hàng bán chạy sẽ được hiển thị tại đây" (cỡ 16, màu xám).
    - Căn giữa (CrossAxisAlignment.center).
  + **Khoảng cách**:
    - MediaQuery.of(context).size.height \* 0.1 tạo khoảng cách động theo chiều cao màn hình.

## Card\_HuyMon

* **Card**:
* Màu nền: Colors.white.withOpacity(0.9) (trắng mờ).
* Viền: Bo tròn (borderRadius: 20).
* Độ nổi: 10 (giống CardHangbanchay, CardHangtonkho).
* **Tiêu đề**:
* Văn bản: "Hủy món".
* Kiểu chữ: Cỡ 18, đậm, màu đen.
* **Danh sách món hủy**:
* Tạo động từ cancelItems bằng asMap().entries để thêm Divider giữa các mục (trừ mục cuối).
* Mỗi mục hiển thị: tên, tỷ lệ, số lượng, doanh thu.
* **\_huyMon**:
* Tên và tỷ lệ: Cỡ 16, màu đen mờ (Color.fromARGB(171, 0, 0, 0)).
* Số lượng và doanh thu: Cỡ 16, màu xám.
* IconButton: Mũi tên phải, hiện rỗng (onPressed: () {}).