

Atelier 2: correction

Tâches

- 1) Créer une classe Produit (libelle, description, prix, ,photo)
- 2) La liste contenant les noms des produits doit contenir une liste d'objets de type Produit
- 3) Mettre à jour l'ajout d'un produit
- 4) Dans la liste des produits on doit afficher le nom et la photo des produits
- 5) Lorsqu'on tape sur une ligne les détails du produit doivent être affichés.

Question 1) Ajout de la classe Produit

Créer un dossier model et y ajouter le fichier produit.dart.

```
class Produit {  
  String? id; ← sera le clé dans l'ateli  
  String? libelle;  
  String? description;  
  double? prix;  
  String? photo;  
  Produit({  
    this.id,  
    this.libelle,  
    this.description,  
    this.prix,  
    this.photo,  
  });  
  Constructeur nommé  
  Produit.empty()  
    : id = '',  
      libelle = '',  
      description = '',  
      prix = 0,  
      photo = '';  
}
```

et maintenant

```
dependencies:  
  flutter:  
    sdk: flutter  
  flutter_slidable: ^3.1.1  
  path: ^1.9.0  
  
dev_dependencies:  
  flutter_test:  
    sdk: flutter  
  flutter_lints: ^4.0.0  
  
flutter:  
  uses-material-design: true  
  assets:  
    - assets/images/
```

Ajoutez les dépendances

** Créer le dossier assets/images
et copier dedans les
images.*

dans la classe

Question 3) Ajout d'un produit

On peut soit modifier la page d'ajout d'un produit existante ou en créer une nouvelle, nous allons dans cet atelier utiliser une nouvelle page nommée `add_produit_form.dart`

1. Ajouter le widget `AddProduitForm` (`StatefulWidget`) dans un fichier nommé `add_produit_form.dart`
2. Le constructeur doit accepter comme argument une fonction qui va permettre l'ajout d'un produit.

```
import 'package:flutter/material.dart';
import 'model/produit.dart';

class AddProduitForm extends StatefulWidget {
  final Function(Produit) onSubmit;
  const AddProduitForm({super.key, required this.onSubmit});

  @override
  State<AddProduitForm> createState() => _AddProduitFormState();
}
```

3. Ajouter dans la classe `_AddProduitFormState`,
 - a. `formKey` : Utilisé pour la validation du formulaire.
 - b. `_produit` : Instance de `Produit` initialisée à vide pour stocker les données du produit.
 - c. `_pickedImagePath` : Variable pour stocker le chemin de l'image sélectionnée.
 - d. `_pickImage` : Cette méthode permet à l'utilisateur de sélectionner une image depuis la galerie et met à jour l'état du chemin de l'image sélectionnée (`_pickedImagePath`).

```
class _AddProduitFormState extends State<AddProduitForm> {
  final _formKey = GlobalKey<FormState>();
  final _produit = Produit.empty();
  String? _pickedImagePath;

  Future<void> _pickImage() async {
    final ImagePicker picker = ImagePicker();
    final XFile? pickedFile =
      await picker.pickImage(source: ImageSource.gallery);
    if (pickedFile == null) return;
    setState(() {
      _pickedImagePath = pickedFile.path;
    });
  }
}
```

```

void _saveProduit() {
  if (_formKey.currentState!.validate()) {
    _formKey.currentState!.save();
    _produit.photo = _pickedImagePath;
    _produit.id = DateTime.now().millisecondsSinceEpoch.toString();
    widget.onSubmit(_produit);
    Navigator.pop<Object?>(context: context);
  }
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text('Ajouter un produit')),
    body: Padding(
      padding: const EdgeInsets.all(12.0),
      child: Form(
        key: _formKey,
        child: Column(
          children: [
            TextFormField(
              decoration: const InputDecoration(labelText: 'Libellé:'),
              validator: (value) {
                if (value!.isEmpty) {
                  return 'Veuillez saisir le libellé';
                }
                return null;
              },
              onSave: (value) => _produit.libelle = value,
            ), // TextFormField

```

```

TextFormField(
  decoration: const InputDecoration(labelText: 'Description'),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Veuillez saisir la description';
    }
    return null;
  },
  onSave: (value) => _produit.description = value,
), // TextFormField
TextFormField(
  decoration: const InputDecoration(labelText: 'Prix'),
  keyboardType: TextInputType.number,
  validator: (value) {
    if (value!.isEmpty) {
      return 'Veuillez saisir le prix';
    }
    return null;
  },
  onSave: (value) => _produit.prix = double.parse(value!),
), // TextFormField

```

```

GestureDetector(
  onTap: _pickImage,
  child: Container(
    width: 80,
    height: 80,
    decoration: BoxDecoration(
      border: Border.all(color: Theme.of(context).primaryColor),
      color: Theme.of(context).primaryColorLight,
      shape: BoxShape.circle,
      image: DecorationImage(
        fit: BoxFit.cover,
        image: _pickedImagePath != null
          ? FileImage(File(_pickedImagePath!))
            as ImageProvider<Object>
          : const AssetImage('assets/images/placeholder.jpg'),
      ), // DecorationImage
    ), // BoxDecoration
  )), // Container // GestureDetector

```

```

Padding(
  padding: const EdgeInsets.symmetric(vertical: 16.0),
  child: Row(
    children: [
      TextButton.icon(
        onPressed: () {
          _formKey.currentState!.reset();
          setState(() {
            _pickedImagePath = null;
          });
        },
        icon: const Icon(Icons.restore_from_trash_rounded),
        label: const Text('Réinitialiser'), // TextButton.icon
      ),
      ElevatedButton(
        onPressed: () {
          if (_formKey.currentState!.validate()) {
            _saveProduit();
          }
        },
        child: const Text('Enregistrer'),
      ), // ElevatedButton
    ],
  ), // Row
), // Padding
],
), // Column
), // Form
), // Padding
); // Scaffold
}

```

- 5) Lorsqu'on tape sur une ligne les détails du produit doivent être affichés (produit_details.dart)

Ajouter le fichier produit_details.dart

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter_app_2024/model/produit.dart';

class ProduitDetails extends StatelessWidget {
  final Produit produit;
  const ProduitDetails({super.key, required this.produit});
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(produit.libelle!,
          style: const TextStyle(fontSize: 16, fontWeight: FontWeight.bold)), // Text
      ), // AppBar
      body: Center(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            crossAxisAlignment: CrossAxisAlignment.center,
            children: [
              Image.file(
                File(produit.photo!),
                height: 160,
                width: 160,
              ), // Image.file
              const SizedBox(height: 16),
              Text('Prix: ${produit.prix}',
                style: const TextStyle(fontSize: 16)), // Text
              const SizedBox(height: 16),
              Text(
                'Description: ${produit.description}',
                style: const TextStyle(fontSize: 16),
              ), // Text
            ],
          ), // Column
        ), // Padding
      ), // Center
    ); // Scaffold
  }
}
```



```

return Scaffold(
  appBar: AppBar(title: Text("Mes contacts")),
  floatingActionButton: FloatingActionButton(
    onPressed: () => Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => AddContactForm(onSubmit: saveContact)), // MaterialPageRoute
    child: Icon(Icons.person),
  ), // FloatingActionButton
  body: ListView.builder(
    itemCount: liste.length,
    itemBuilder: (context, index) {
      return ContactBox(
        contact: liste[index][0],
        selContact: liste[index][1],
        onChanged: (value) => change(value, index),
        delContact: (context) => delContact(index),
      ); // ContactBox
    }); // ListView.builder // Scaffold

```

On passe à ContactBox un objet Contact

Modifications dans le widget ProduitBox

```

import 'dart:io';
import 'package:flutter/material.dart';
import 'package:flutter_app_2024/model/produit.dart';
import 'package:flutter_app_2024/produit_details.dart';
import 'package:flutter_slidable/flutter_slidable.dart';

class ProduitBox extends StatelessWidget {
  final Produit produit;
  final bool selProduit;
  final Function(BuildContext context)? delProduit;
  final Function(bool?)? onChanged;

  const ProduitBox(
    {super.key,
    required this.produit,
    required this.selProduit,
    required this.delProduit,
    required this.onChanged});

```

```
Widget build(BuildContext context) {
  return Padding(
    padding: const EdgeInsets.all(10.0),
    child: Slidable(
      endActionPane: ActionPane(
        motion: const StretchMotion(),
        children: [
          SlidableAction(
            onPressed: delProduit,
            icon: Icons.delete,
            backgroundColor: Colors.red,
            borderRadius: BorderRadius.circular(45),
          ), // SlidableAction
        ],
      ), // ActionPane
    child: InkWell(
      onTap: () {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => ProduitDetails(produit: produit)); // MaterialPageRoute
      ),
    ),
  );
}
```

4) Dans la liste des produits on doit afficher le nom et la photo des produits

Dans `_ProduitsListState`

- La liste des produits doit être vide initialement

```
// Liste des produits
List liste = [];
```

- `saveProduit`

```
void saveProduit(Produit produit) {
  setState(() {
    liste.add([produit, false]);
  });
}
```

- Modifications dans le widget scaffold

```

        child: Container(
          decoration: BoxDecoration(
            color: Colors.yellow,
            borderRadius: BorderRadius.circular(45),
          ), // BoxDecoration
          height: 120,
          child: Row(
            children: [
              Checkbox(
                value: selProduit,
                onChanged: onChanged,
              ), // Checkbox
              Center(
                child:
                  Image.file(width: 50, height: 50, File(produit.photo!)),
              ), // Center
              const SizedBox(width: 10),
              Text(produit.libelle!),
            ],
          ), // Row
        ), // Container
      ), // InkWell
    ), // Slidable
  ); // Padding
}

```