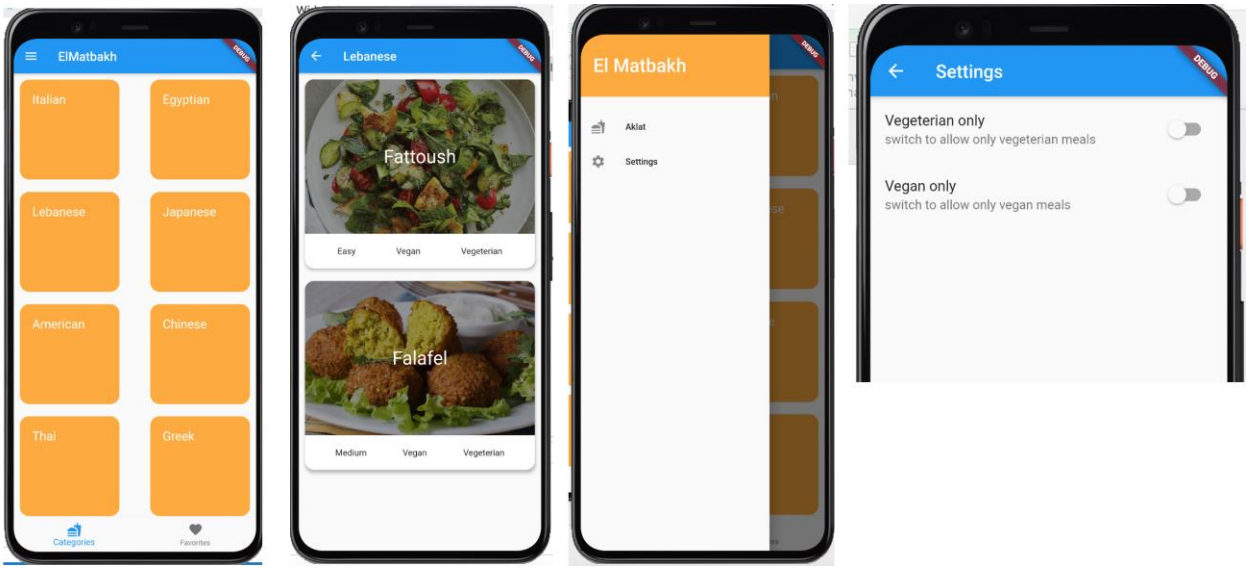
Software mobile development - GIU
Spring 2025

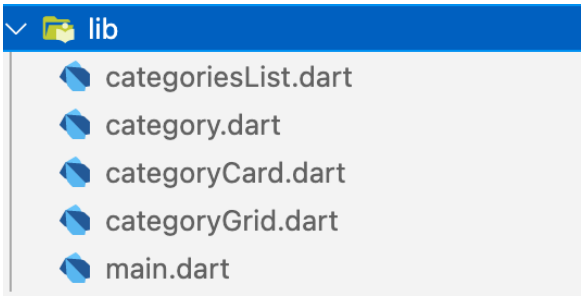
Lab 3

The goal of this lab is to develop a kitchen/recipes application called El Matbakh.



Step 1

We need to create the homepage containing the categories organized in a grid. For that you need to create the following classes.



A) Category.dart

```
class Category {

String id;
String title;

Category({required this.id, required this.title});

}
```

B) categorylist.dart

```
import '../models/category.dart';

var categoriesList = [

Category(id:'c1',title:'Italian'),
Category(id:'c2', title:'Egyptian'),
Category(id:'c3', title:'Lebanese'),
Category(id:'c4', title:'Japanese'),
Category(id:'c5', title:'American'),
Category(id:'c6', title:'Chinese'),
Category(id:'c7', title:'Thai'),
Category(id:'c8', title:'Greek'),
];
```

C) categorycard.dart

```
import 'package:flutter/material.dart';
import './category.dart';

class CategoryCard extends StatelessWidget {
Category cat;

CategoryCard({required this.cat});

@override
Widget build(BuildContext context) {
return Container(
child: Text( cat.title, style: TextStyle(color: Colors.white,
fontSize: 20),),
padding: EdgeInsets.all(20), margin: EdgeInsets.all(10),
decoration: BoxDecoration(
borderRadius: BorderRadius.circular(15), color:
Colors.orangeAccent),);
}}
```

D) Category grid widget

```
import 'package:elmatbakh/categoryCard.dart';
import 'package:flutter/material.dart';
import './categoriesList.dart';

class CategoryGrid extends StatelessWidget {

@override
Widget build(BuildContext context) {
return Scaffold(
appBar: AppBar(title: Text('El Matbakh'),),

body: // write your code here
}}
```

E) our main.dart

```
import 'package:elmatbakh/categoryGrid.dart';
import 'package:flutter/material.dart';

void main() { runApp(MyApp()); }

class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
return MaterialApp(
        title: 'Flutter Demo',
        theme: ThemeData( primarySwatch: Colors.blue,),
        home: CategoryGrid()
);}}
```

Run and check your app.

Step 2

We need to add some navigation to our app.

A) **Create** a kitchen screen page

```
class KitchenScreen extends StatelessWidget {
Category category;
KitchenScreen({required this.category});

@override
Widget build(BuildContext context) {
return Scaffold
(appBar: AppBar(title: Text(category.title) ,),
body: Container(child: Text('Kitchen page for
${category.title}'),),
);}}
```

B) **Modify** your category card class to add the clicking functionality and the navigation to the screen page.

```
class CategoryCard extends StatelessWidget {
Category cat;

CategoryCard({required this.cat});

navigateToKitchenPage(BuildContext myContext){

 Navigator.of(myContext).push(MaterialPageRoute(builder:
    (ctxDummy){return KitchenScreen(category:cat);}));
 }

@override
Widget build(BuildContext context) {
return GestureDetector(
onTap: () => navigateToKitchenPage(context),
child: Container(// same content as before);}}
```

Run and check you app.

Step 3

Using named routes

A) **Add** the routes to your main.dart file by **updating** the build method to look like this.

```
class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,),

       initialRoute: '/',
       routes: {
        // write your code here
        },
    );
  }
}
```

B) **Update** your category card so that it uses push named instead of push.

```
class CategoryCard extends StatelessWidget {

  Category cat;

  CategoryCard({required this.cat});

navigateToKitchenPage(BuildContext myContext){

    Navigator.of(myContext).pushNamed('/kitchenRoute',
arguments: {'category': cat});
}

  @override
  Widget build(BuildContext context) {
    // same as before
}
```

C) Update your kitchen screen to receive those arguments in the new way rather than in the constructor.

```
class KitchenScreen extends StatelessWidget {

@override
Widget build(BuildContext context) {

// VARIABLE AND CONSTRUCTOR REMOVED (YOU CAN COMMENT THEM OUT)

// NEW PARTS
final routeArgs = ModalRoute.of(context)!.settings.arguments as
Map<String,Category>;

final extractedCategory = routeArgs['category'];

return Scaffold
( // UPDATED PARTS

appBar: AppBar(title: Text(extractedCategory!.title) ,),

body: Container(child:
Text('Kitchen page for ${extractedcategory.title}'),),
);}}
```

Step 4

Updating the kitchen screen page to be more appealing and shows all recipes belonging to that kitchen.

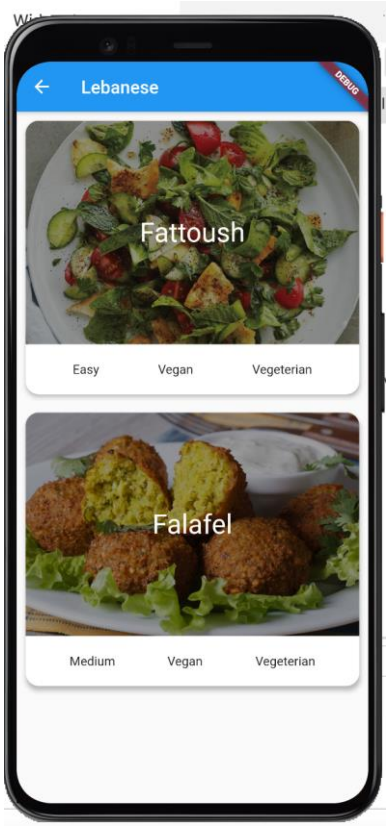
A) Create a Recipe class.

```
class Recipe {
  final String id;
  final String title;
  final List<String> categoryId;
  final int difficulty;
  final List<String> ingredients;
  final String steps;
  final bool isVegan;
  final bool isVegeterian;
  final String imageURL;

  Recipe(
      {required this.id,
      required this.title,
      required this.categoryId,
      required this.difficulty,
      required this.ingredients,
      required this.steps,
      required this.isVegan,
      required this.isVegeterian,
      required this.imageURL});
}
```

B) Create a dummy RecipeList class (copy and paste the below)

```dart
import '../models/recipe.dart';
import 'package:flutter/material.dart';

var recipesList = [
  Recipe(
      id: 'r1',
      title: 'Fattoush',
      categoryId: ['c3'],
      difficulty: 1,
      ingredients: ['Lettuce', 'Tomato', 'Summac','Onion', 'Olive oil'],
      steps: 'Mix all together',
      isVegan: true,
      isVegeterian: true,
imageURL:'https://assets.bonappetit.com/photos/57af6bea53e63daf11a4e565/16:9/w_1280,c_limit/fattoush.jpg' ),
  Recipe(
      id: 'r2',
      title: 'Falafel',
      categoryId: ['c2','c3'],
      difficulty: 2,
      ingredients: ['Fava beans', 'Hummus', 'Spices','frying oil'],
      steps: 'Mix ingredients into balls and fry them',
      isVegan: true,
      isVegeterian: true ,imageURL:
'https://toriavey.com/images/2011/01/TOA109_18.jpeg'),
  Recipe(
      id: 'r3',
      title: 'Chicken Alfredo',
      categoryId: ['c1'],
      difficulty: 2,
      ingredients: ['pasta', 'chicken' ,'alredo sauce'],
      steps: 'Boil pasta, prepare chicken and pour sauce over',
      isVegan: false,
      isVegeterian: false, imageURL:'https://bellyfull.net/wp-content/uploads/2021/02/Chicken-Alfredo-blog-4.jpg' )
];
```

C) Update the kitchen screen to have a list of cards with images and titles stacked on top. Figure below.



C.1) Create a recipe card class.

```
class RecipeCard extends StatelessWidget {
  Recipe rec;

  RecipeCard({required this.rec});

  void goToSpecificRecipe() {}

String getDifficulty(int diff){

if(diff==1){return 'Easy';}
else if(diff==2) {return 'Medium';}
else return 'Hard';


}
  @override
  Widget build(BuildContext context) {
    return InkWell(
      onTap: goToSpecificRecipe,
      child: Card(
        shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(15)),
        elevation: 4,
        margin: EdgeInsets.all(10),
        child: Column(
          children: [
            // child 1 of column is image + title
            Stack(
              children: [
// child 1 of stack is the recipe image
                ClipRRect(
                  borderRadius: BorderRadius.only(
                      topLeft: Radius.circular(15),
                      topRight: Radius.circular(15)),
                  child: Image.network(

                    rec.imageURL,
                    height: 250,
                    width: double.infinity,
                    fit: BoxFit.cover,

                  ),
                ),
// child 2 of stack is the recipe title
                Positioned.fill(
                  child: ClipRRect(
                    borderRadius: BorderRadius.only(
```

```
                        topLeft: Radius.circular(15),
                        topRight: Radius.circular(15)),
                    child: Container(
                      color: Colors.black38,
                      child: Center(

                          //color: Colors.black38,
                          child: Text(
                        rec.title,
                        softWrap: true,
                        overflow: TextOverflow.fade,
                        style: TextStyle(color: Colors.white,
fontSize: 30),
                        textAlign: TextAlign.center,
                      )),
                    ),
                  ),
                  bottom: 0,
                )
              ],
            ),

            // child 2 of colum is the white row
            Container(
              margin: EdgeInsets.all(15),
              padding: EdgeInsets.all(5),
              child: Row(
                mainAxisAlignment:
MainAxisAlignment.spaceAround,
                children: [
                  Text(getDifficulty(rec.difficulty)),
                  Text(rec.isVegan ? 'Vegan' : 'Non-Vegan'),
                  Text(rec.isVegeterian ? 'Vegeterian' : 'Non-
Vegeterian')
                ],
              ),
            )
          ],
        ),
      ),
    );
  }
}
```

C.2) **Update** your kitchen screen page to use the card you just created in a list.

```
  @override
  Widget build(BuildContext context) {


    final routeArgs =
        ModalRoute.of(context)!.settings.arguments as
Map<String, Category>;

    final category = routeArgs['category'];

    final recipesInThatCategory = recipesList.where((element) {
      return element.categoryId.contains(category!.id);
    }).toList();




    return Scaffold(
      appBar: AppBar(
        title: Text(category!.title),
      ),
      body: ListView.builder(
        // write your code here
      ),
    );
  }
}
```