



Build Your IT Skill

ណែនាំស្តាប់ពី Local Storage



រៀបរាងដោយ៖

គ្រូរោយធីចិត្តល្អ

អនុបណ្ឌិត RUPP, TKU

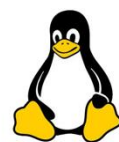
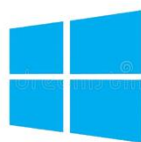
(Software Engineering)

រៀបរៀងដោយ៖

Mr. Pha Phanna, Bs IT

Local Storage

Flutter MVC & MVVM



I. About Local Storage

Local storage គឺជាការរក្សាទុកនូវ data នៅក្នុង Storage របស់ Device ដែលបាន Install App នោះផ្ទាល់។ ចំពោះទិន្នន័យ វាប្តូរជាប់តែនៅពេលដែល យើង Clear data or Uninstall Application ចេញពី Device នោះ ពេលគឺដូចជា App offline ជាដើម។

II. SharedPreferences

Package

```
dependencies:  
  
  shared_preferences: ^2.0.15  
  
  flutter:
```

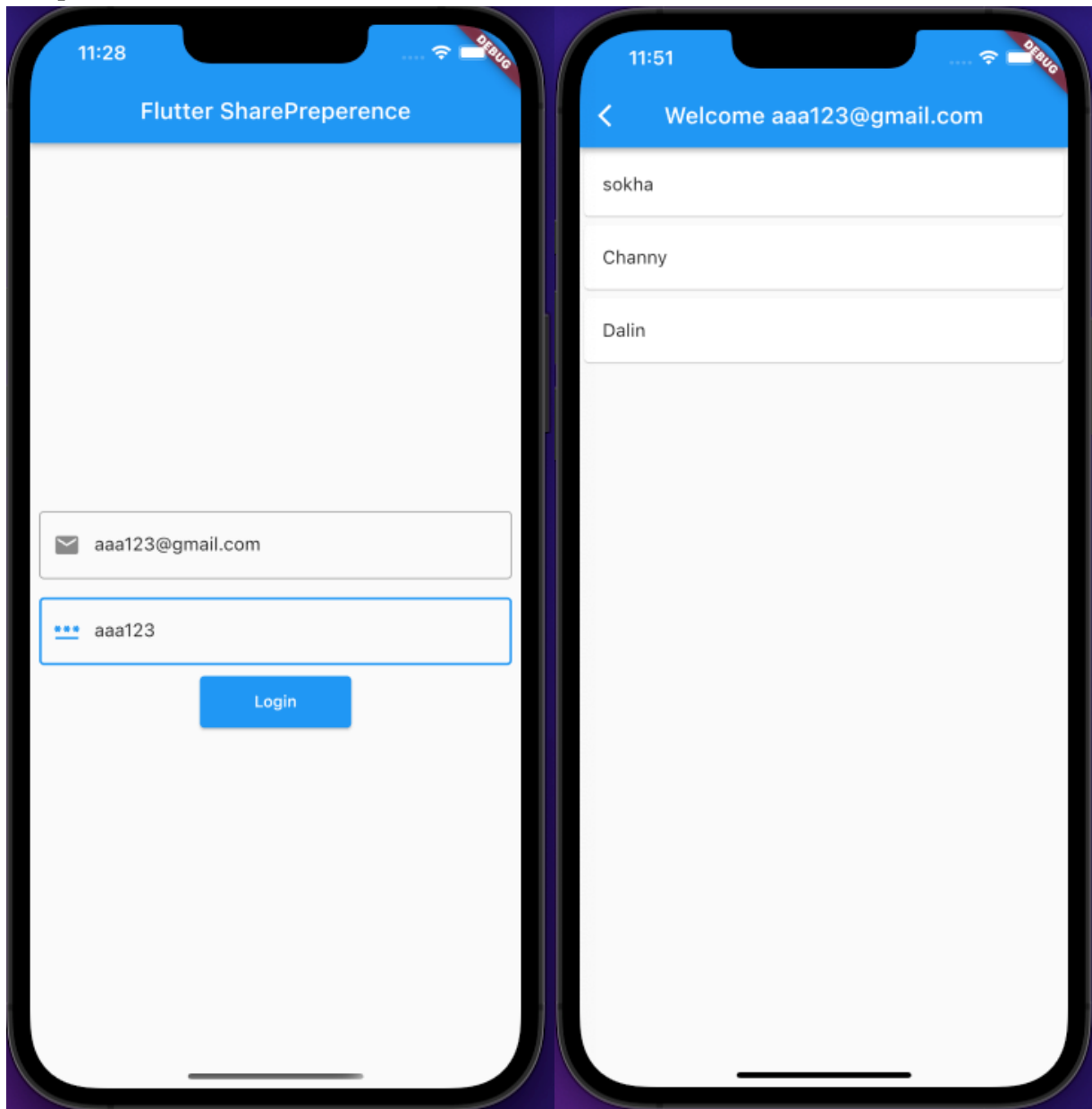
setData

```
void getInitSharePreference() async {  
  
  var prefs = await SharedPreferences.getInstance();  
  
  await prefs.setString('email', 'aaa123@gmail.com');  
  
  await prefs.setString('password', 'aaa123');  
  
  await prefs.setInt('age', 23);  
  
  await prefs.setBool('check', true);  
}
```

getData

```
String? email = "", password = "";  
  
List<String> listName = [];  
  
getValueSharePreference() async {  
  
  var prefs = await SharedPreferences.getInstance();  
  
  
  setState(() {  
  
    email = prefs.getString('email');
```

Example UI



III. **SQLITE**

ជា database ដែលអាច Store data ក្នុង local storage.....របស់ Device បាន។

CRUD Sqlite (Create, read, update, delete)

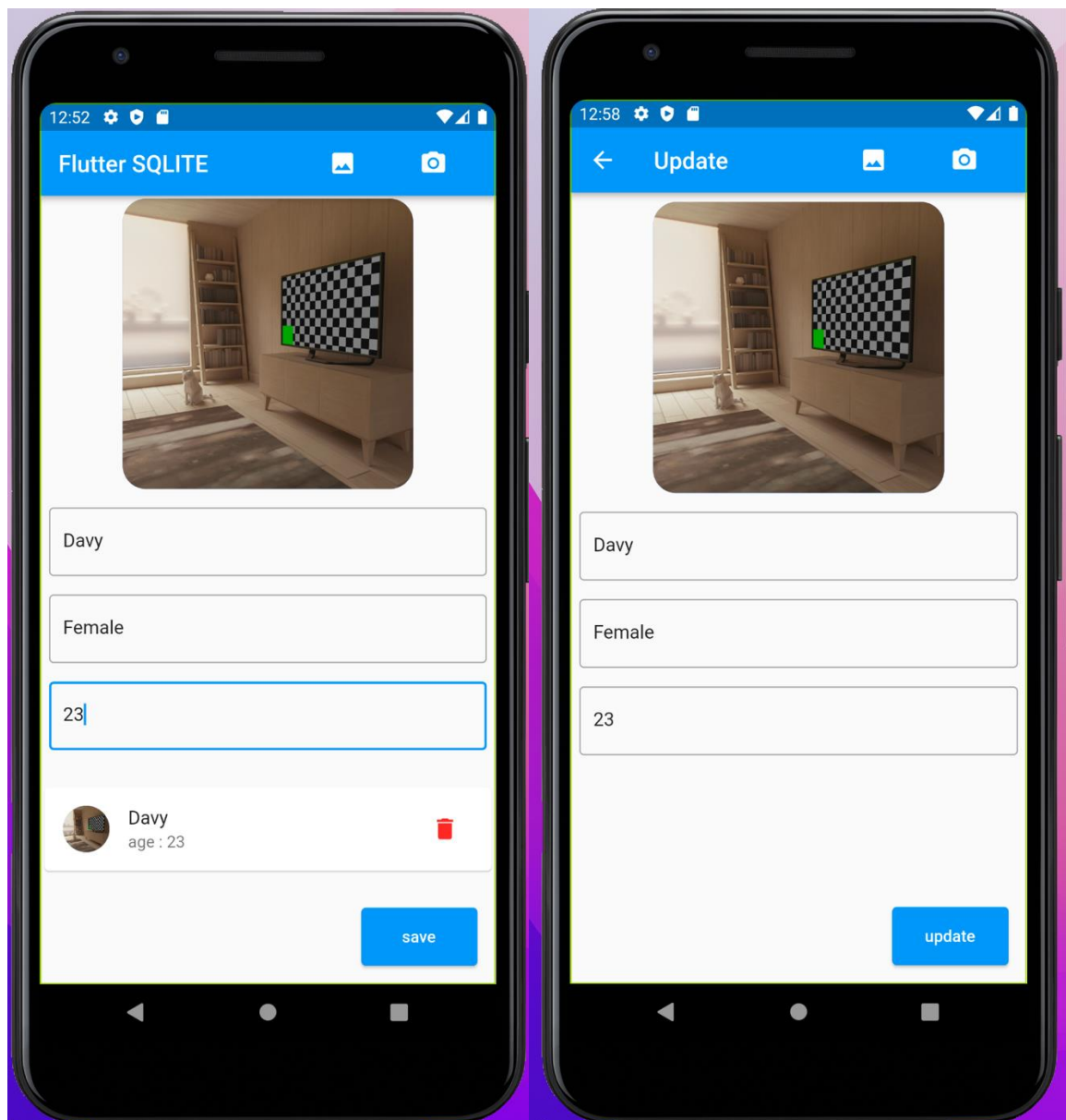
dependencies:

sqlite: ^2.0.3

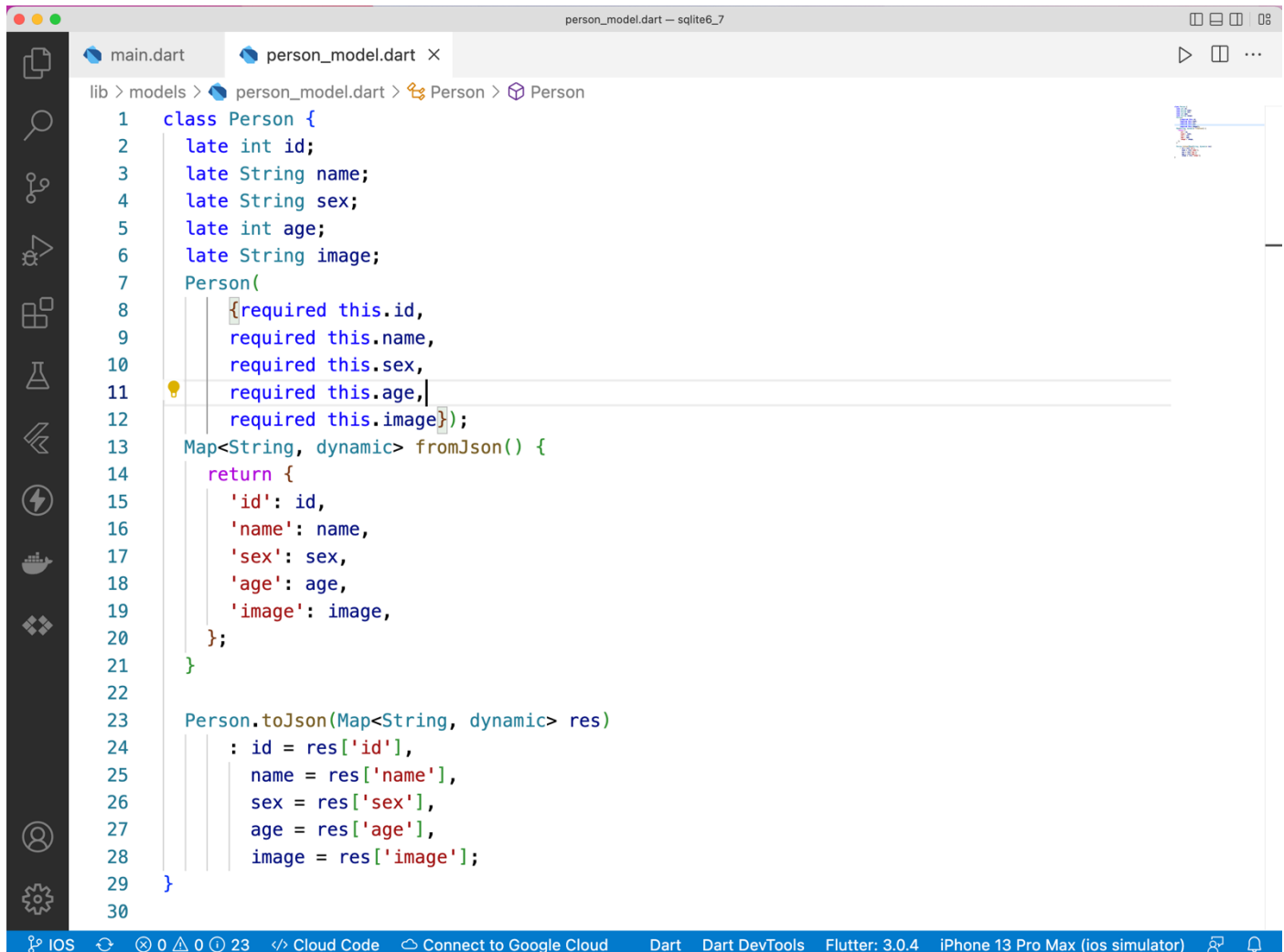
image_picker: ^0.8.5+3

path_provider: ^2.0.11

Example UI



Model



ConnectionDatabase

```
import 'dart:io';

import 'package:path_provider/path_provider.dart';

import 'package:sqflite/sqflite.dart';

import 'package:sqflite/sqflite_api.dart';

import 'package:path/path.dart'; // for use with function join

String table = 'person';
```

```
database_connection.dart — sqlite6_7
main.dart  person_model.dart  database_connection.dart M  pubspec.yaml
lib > connection > database_connection.dart > DataConnection > getPersonData

10 class DataConnection {
11   Future<Database> initializeData() async {
12     Directory tempDir = await getTemporaryDirectory();
13     String tempPath = tempDir.path;  The value of the local variable 'tempPath' isn't used
14     Directory appDocDir = await getApplicationDocumentsDirectory();
15     String appDocPath = appDocDir.path;  The value of the local variable 'appDocPath' isn't
16     String path = await getDatabasesPath();
17     return openDatabase(
18       join(path, 'data.db'),
19       version: 1,
20       onCreate: (db, version) async {
21         await db.execute(
22           'CREATE TABLE $table(id INTEGER PRIMARY KEY, name TEXT,sex TEXT, age  INTEGER,image TEXT)');
23       },
24     );
25   }
26
27   Future<void> insertData(Person person) async {
28     final db = await initializeData();
29     await db.insert(table, person.toJson());
30     print('object was insert to database');  Avoid `print` calls in production code.
31   }
32
33   Future<List<Person>> getPersonData() async {
34     final db = await initializeData();
35     List<Map<String, dynamic>> result = await db.query(table);
36     return result.map((e) => Person.fromJson(e)).toList();
37   }
38   // Future<Person> getPerObjectList()
39
40   Future<void> deletePersonData(int id) async {
41     final db = await initializeData();
42     await db.delete(table, where: 'id=?', whereArgs: [id]);
43   }
44
45   Future<void> updatePersonData(Person person) async {
46     final db = await initializeData();
47     await db.update(table, person.toJson(),
48       where: 'id=?', whereArgs: [person.id]);
49   }
50 }
51
```

Save Data

```
child: ElevatedButton(  
  child: const Center(child: Text('save')),  
  onPressed: () async {  
    await DataConnection()  
      .insertData(Person(  
        id: Random().nextInt(1000),  
        name: name_controller.text,  
        sex: sex_controller.text,  
        age: int.parse(age_controller.text),
```

Read &Delete Data


```
174   SizedBox(  
175     height: 500,  
176     width: double.infinity,  
177     child: FutureBuilder<List<Person>>(  
178       future: listPerson,  
179       builder: (context, AsyncSnapshot<List<Person>> snapshot) {  
180         if (snapshot.connectionState == ConnectionState.waiting) {  
181           return const Center(  
182             child: CircularProgressIndicator(),  
183           ); // Center  
184         } else if (snapshot.hasError) {  
185           return const Center(  
186             child: Icon(  
187               Icons.info,  
188               color: Colors.red,  
189               size: 30,  
190             ), // Icon  
191           ); // Center  
192         } else {  
193           return ListView.builder(  
194             itemCount: snapshot.data!.length,  
195             itemBuilder: (context, index) {  
196               var per = snapshot.data![index];  
197               return InkWell(  
198                 onTap: () {  
199                   Navigator.push(  
200                     context,  
201                     MaterialPageRoute(  
202                       builder: (context) => UpdateData(person: per),  
203                     )); // MaterialPageRoute
```

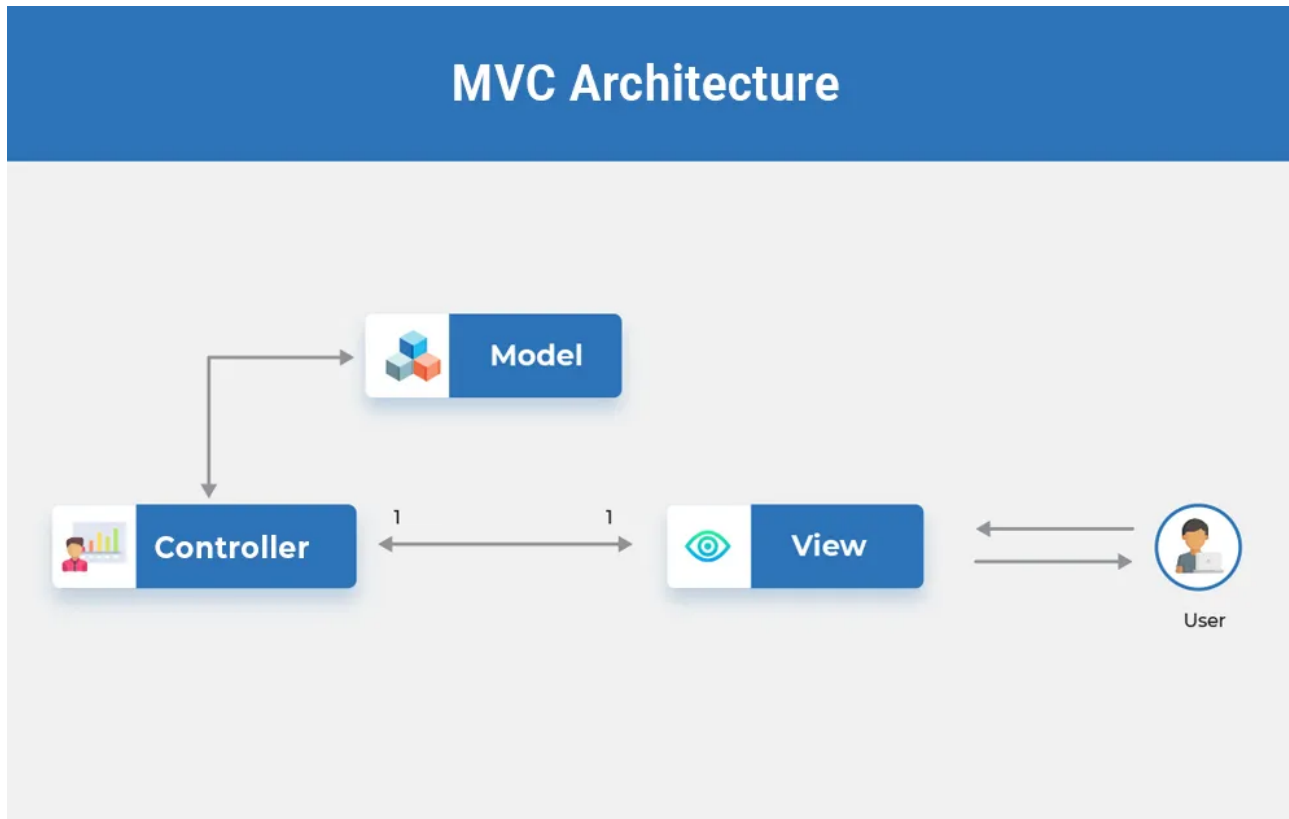
```
204 },
205 child: Card(
206   child: ListTile(
207     leading: CircleAvatar(
208       backgroundImage: FileImage(File(per.image)),
209     ), // CircleAvatar
210     title: Text(per.name),
211     subtitle: Text('age : ${per.age}'),
212     trailing: IconButton(
213       onPressed: () async {
214         await DataConnection()
215           .deletePersonData(per.id)
216           .whenComplete(() => _onRefresh());
217       },
218       icon: const Icon(
219         Icons.delete,
220         color: Colors.red,
221       )), // Icon // IconButton
222   ), // ListTile
223 ), // Card
224 ); // InkWell
225 },
226 ); // ListView.builder
227 }
228 },
229 ), // FutureBuilder
230 ) // SizedBox
```

Update Data

```
child: ElevatedButton(
  child: const Center(child: Text('update')),
  onPressed: () async {
    await DataConnection()
      .updatePersonData(Person(
        id: widget.person.id,
        name: name_controller.text,
        sex: sex_controller.text,
        age: int.parse(age_controller.text),
        image: _image == null ? widget.person.image : _image!.path))
      .whenComplete(() => Navigator.pushAndRemoveUntil(
        context,
```

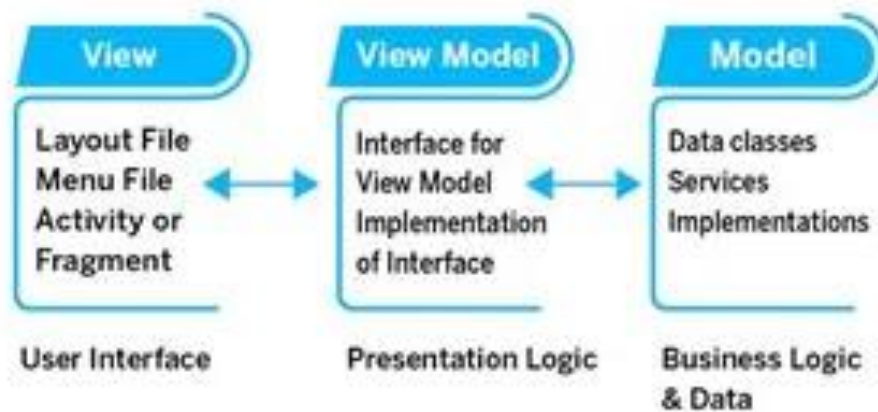
Flutter MVC & MVVM

MVC : Model – View - Controller



MVVM : Model – View – View Model

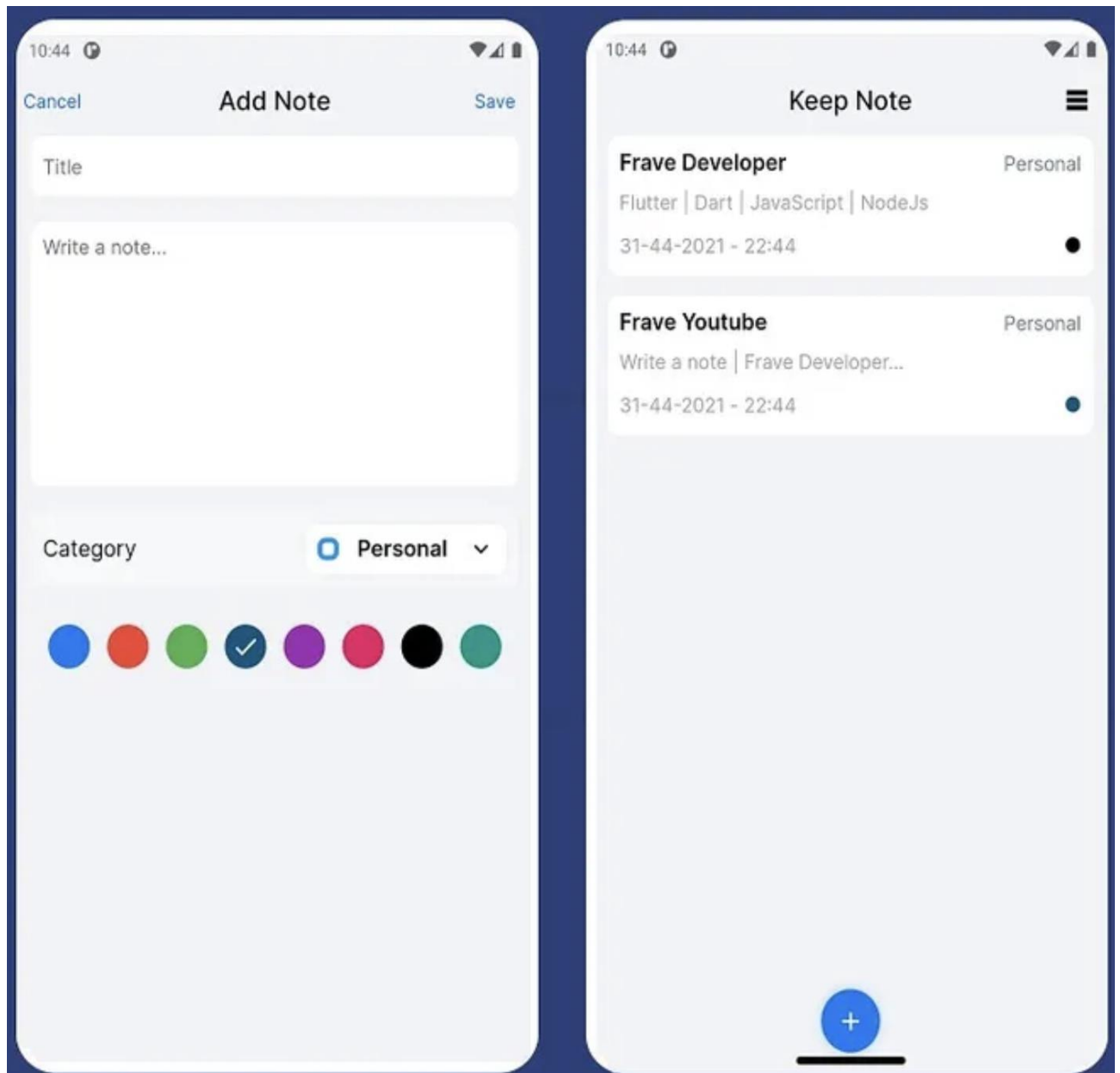
Flow Chart of MVVM



Apply App ដែលប្រើ Sqlite ខាងលើទៅជា ទម្រង់ MVC បន្ថែមទៀត.....

Practice and research

- Create App Note Using SQLite (CRUD)

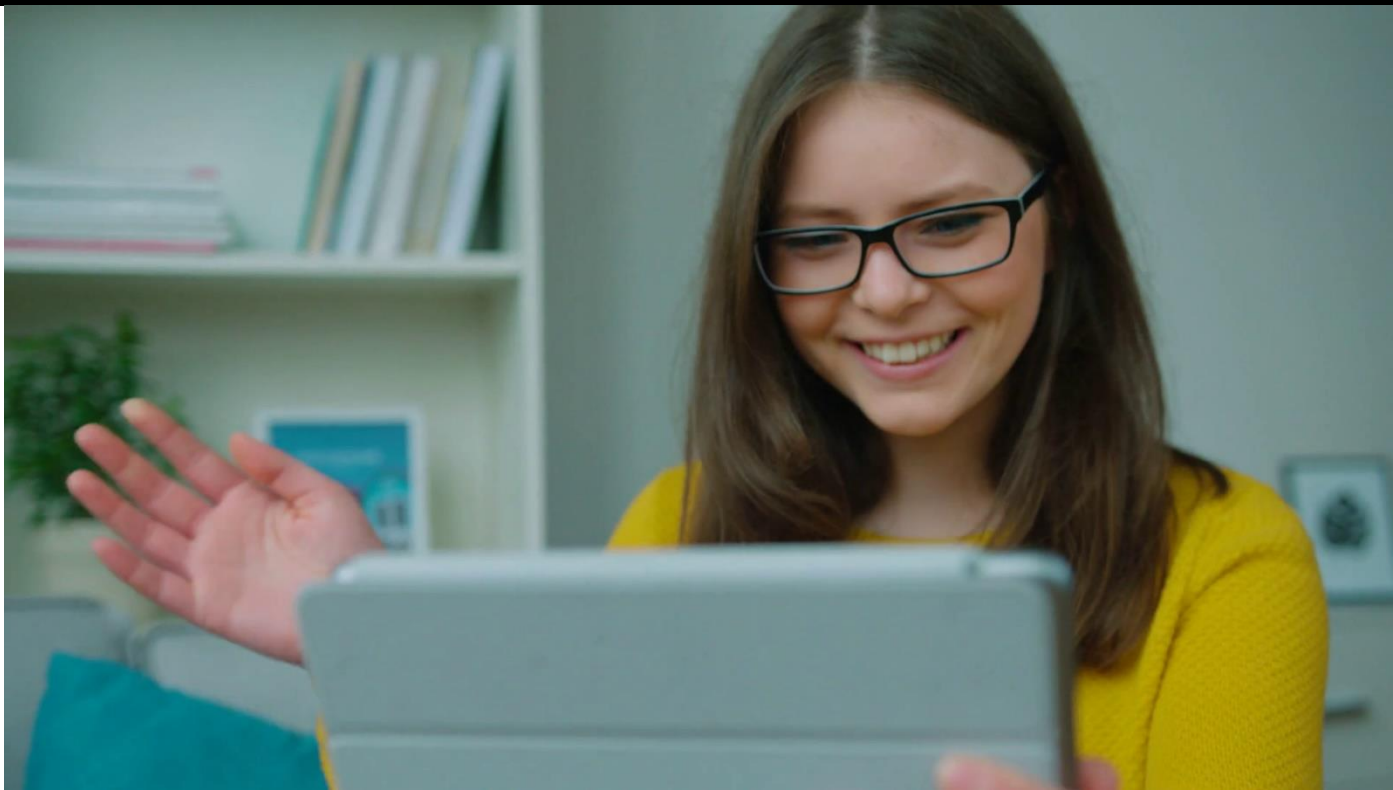


- Change App Icon
- Change App Name



Build Your IT Skill

រំលងសំណួរ State Management



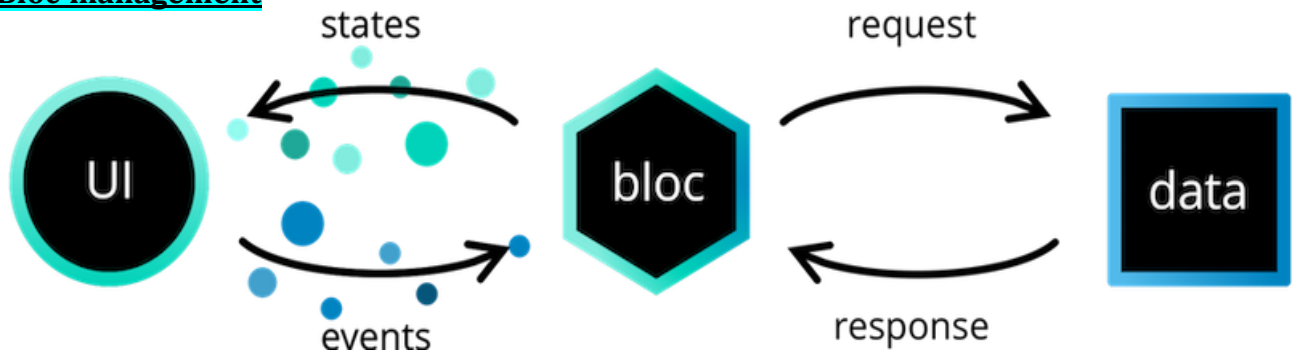
រៀបរាងដោយ៖
គ្រូមេឃឹមប៊ុនណុ
អនុបណ្ឌិត RUPP, TKU
(Software Engineering)
រៀបរៀងដោយ៖
Mr. Pha Phanna, Bs IT

About StateManagement

StateManagement គឺជាការ កាត់ផ្ដាច់ ហើយ Manage រវាង Data និង UI ដែលអាច៖

- អាចអោយ StatelessWidget អាច Change State របស់UI បាន
- អាច initialize data
- កាត់បន្ថយការប្រើប្រាស់ setState()
- អាចប្រើប្រាស់ Provider & Multi-Provideretc.

IV. Bloc management



- Package

dependencies:

flutter_bloc: ^8.1.1



“រៀនត្រូវចេះ, រៀនត្រូវឆ្លាត ,រៀនត្រូវបំបែក
,រៀនត្រូវចេះ
មើលគេងង រៀនត្រូវសួរខ្លួនឯងចងចេះអីខ្លះហើយ”
—គ្រូអាយធីចិត្តល្អ

- **State**

```
part of 'counter_bloc.dart';  
  
class CounterState {  
  CounterState({required this.counter});  
  int counter;  
}
```

- **Event**

```
part of 'counter_bloc.dart';  
  
abstract class CounterEvent extends Equatable {  
  const CounterEvent();  
  
  @override  
  List<Object> get props => [];
```

- **Bloc**

```
import 'dart:io';

import 'package:bloc/bloc.dart';
import 'package:equatable/equatable.dart';

part 'counter_event.dart';
part 'counter_state.dart';

class CounterBloc extends Bloc<CounterEvent, CounterState> {
  CounterBloc() : super(CounterInitial()) {
    on<IncrementEvent>((event, emit) {
      emit(CounterState(counter: state.counter + 1));
    });
  }
}
```

Part Screen1



“ពាក្យថា រៀន គឺ រៀន មិន ចេះ ចប់ នោះ ទេ, ខ្លាច តែ
អ្នក រៀន គត់ ថា រៀន ចប់ អស់ ហើយ,
ចឹង ចំណេះ ចេះ តែ ម៉្លេះ ទ្រង់...”

— គ្រូ អាយធី ចិត្តល្អ


```
main.dart — blocError
main.dart × counter_bloc.dart counter_event.dart
lib > main.dart > main
1 import 'package:bloc_counter2_4/bloc/counter_bloc.dart';
2 import 'package:flutter/material.dart';
3 import 'package:flutter_bloc/flutter_bloc.dart';
4
Run | Debug | Profile
5 void main() {
6   Bloc.observer = AppBlocObserver();
7   runApp(const MyApp());
8 }
9
10 class AppBlocObserver extends BlocObserver {
11   @override
12   void onChange(BlocBase bloc, Change change) {
13     super.onChange(bloc, change);
14     if (bloc is Cubit) print(change); // Avoid `print` calls in production
15   }
16
17   @override
18   void onTransition(Bloc bloc, Transition transition) {
19     super.onTransition(bloc, transition);
20     print(transition); // Avoid `print` calls in production code.
21   }
22 }
23
24 class MyApp extends StatelessWidget {
25   /// {@macro app}
26   const MyApp({Key? key}) : super(key: key);
27
28   @override
29   Widget build(BuildContext context) {
30     return BlocProvider(
31       create: (_) => CounterBloc(),
32       child: const AppView(),
33     ); // BlocProvider
34   }
35 }
36
```

```
37 class AppView extends StatelessWidget {
38   const AppView({Key? key}) : super(key: key);
39
40   @override
41   Widget build(BuildContext context) {
42     return BlocBuilder<CounterBloc, CounterState>(
43       builder: (context, state) {
44         return const MaterialApp(
45           color: Colors.blueAccent,
46           home: MyHomePage(title: 'Counter Blog'),
47         ); // MaterialApp
48       },
49     ); // BlocBuilder
50   }
51 }
52
53
54 class MyHomePage extends StatefulWidget {
55   const MyHomePage({Key? key, required this.title}) : super(key: key);
56
57   final String title;
58
59   @override
60   State<MyHomePage> createState() => _MyHomePageState();
61 }
62
63 class _MyHomePageState extends State<MyHomePage> {
64   @override
65   Widget build(BuildContext context) {
66     return Scaffold(
67       appBar: AppBar(
68         title: Text(widget.title),
69         actions: [
70           MaterialButton(
71             onPressed: () {
72               Navigator.push(
73                 context,
74                 MaterialPageRoute(
75                   builder: (context) => const SecondScreen(),
76                 )); // MaterialPageRoute
77             },
```

```
78     child: const Text(  
79       'Next',  
80       style: TextStyle(fontSize: 28),  
81     ), // Text  
82   ), // MaterialButton  
83   const SizedBox(  
84     width: 40,  
85   ) // SizedBox  
86 ],  
87 ), // AppBar  
88 body: BlocBuilder<CounterBloc, CounterState>(  
89   builder: (context, state) {  
90     return Center(  
91       child: Column(  
92         mainAxisAlignment: MainAxisAlignment.center,  
93         children: <Widget>[  
94           Text(  
95             '${state.counter}',  
96             style: Theme.of(context).textTheme.headline4,  
97           ), // Text  
98         ], // <Widget>[]  
99       ), // Column  
100     ); // Center  
101   },  
102 ), // BlocBuilder  
103 floatingActionButton: Row(  
104   mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
105   children: [  
106     FloatingActionButton(  
107       onPressed: () {  
108         context.read<CounterBloc>().add(DecrementEvent());  
109       },  
110       tooltip: 'Decrement',  
111       child: const Icon(Icons.remove),  
112     ), // FloatingActionButton  
113     FloatingActionButton(  
114       onPressed: () {  
115         context.read<CounterBloc>().add(IncrementEvent());  
116       },  
117       tooltip: 'Increment',  
118       child: const Icon(Icons.add),  
119     ), // FloatingActionButton  
120   ],  
121 ), // Row  
122 ); // Scaffold  
123 }  
124 }  
125
```

Part Screen 2

main.dart M

second_screen.dart U X

counter_bloc.dart

counter...

lib > second_screen.dart > SecondScreen > build

```
1 import 'package:bloc_counter2_4/bloc/counter_bloc.dart';
2 import 'package:flutter/material.dart';
3 import 'package:flutter_bloc/flutter_bloc.dart';
4
5 class SecondScreen extends StatelessWidget {
6   const SecondScreen({Key? key}) : super(key: key);
7
8   @override
9   Widget build(BuildContext context) {
10     return Scaffold(
11       appBar: AppBar(
12         title: const Text('Second Screen'),
13       ), // AppBar
14       body: BlocBuilder<CounterBloc, CounterState>(
15         builder: (context, state) {
16           return Center(
17             child: Column(
18               mainAxisAlignment: MainAxisAlignment.center,
19               children: <Widget>[
20                 Text(
21                   '${state.counter}',
22                   style: Theme.of(context).textTheme.headline4,
23                 ), // Text
24               ], // <Widget>[]
25             ), // Column
19
```

3:40

Second Screen

10

-

+

```
26         ); // Center
27     },
28 ), // BlocBuilder
29 floatingActionButton: Row(
30     mainAxisAlignment: MainAxisAlignment.spaceEvenly,
31     children: [
32         FloatingActionButton(
33             onPressed: () {
34                 context.read<CounterBloc>().add(DecrementEvent());
35             },
36             tooltip: 'Decrement',
37             child: const Icon(Icons.remove),
38         ), // FloatingActionButton
39         FloatingActionButton(
40             onPressed: () {
41                 context.read<CounterBloc>().add(IncrementEvent());
42             },
43             tooltip: 'Increment',
44             child: const Icon(Icons.add),
45         ), // FloatingActionButton
46     ],
47 ), // Row
48 ); // Scaffold
49 }
50 }
```

V. GetX management

dependencies:

get: ^4.6.5

- **Controller**

```
class Controller extends GetxController {

    var counter = 0.obs();

    increment() {

        counter = counter + 1;

        update();

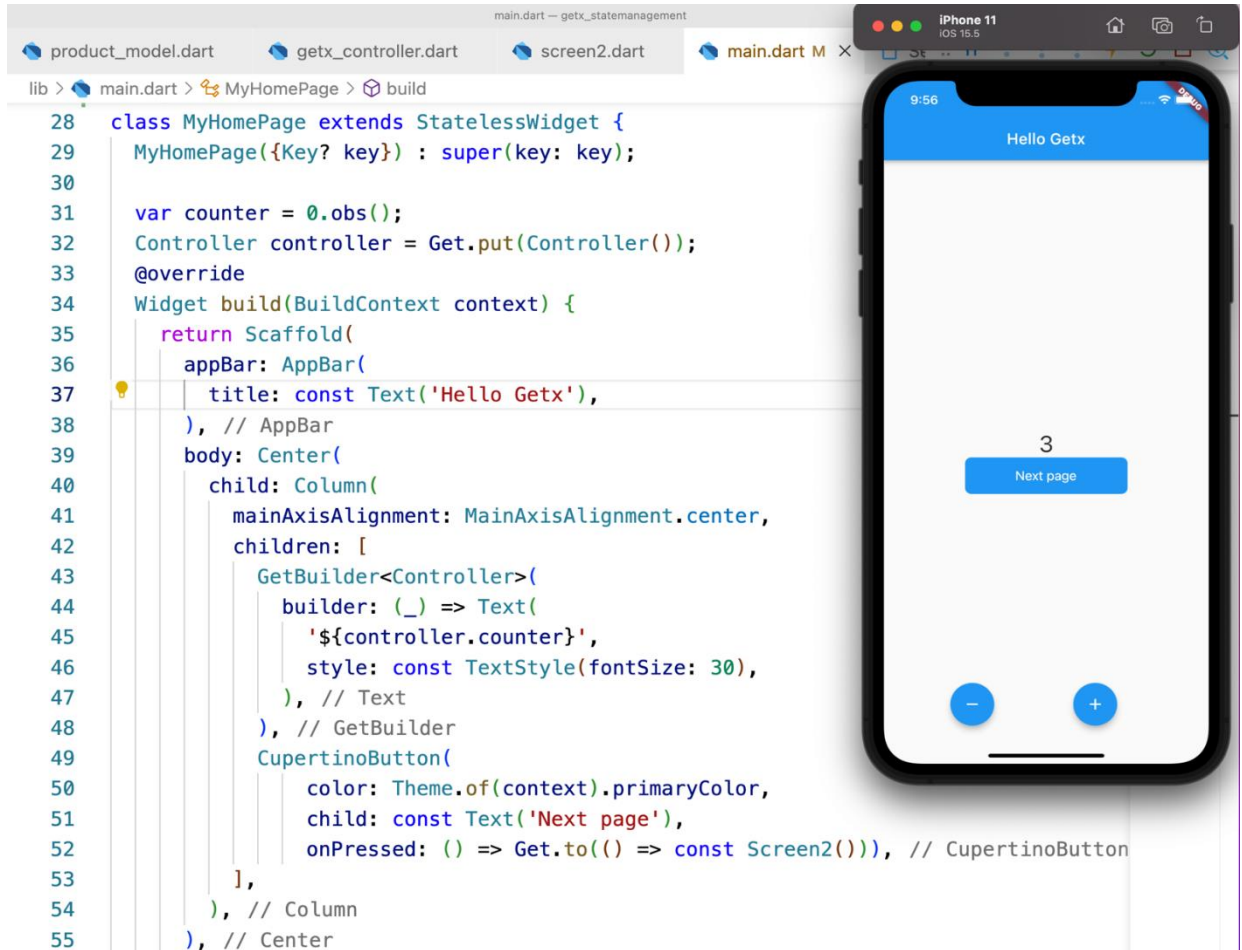
    }

    decrement() {
```

- **MaterialApp**

```
void main() {  
  runApp(const MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      theme: ThemeData(  
        primaryColor: Colors.blue,  
        fontFamily: 'Roboto',  
        textTheme: TextTheme(  
          bodyText1: TextStyle(  
            color: Colors.black,  
            fontSize: 16,  
            fontWeight: FontWeight.normal,  
          ),  
          bodyText2: TextStyle(  
            color: Colors.black,  
            fontSize: 14,  
            fontWeight: FontWeight.normal,  
          ),  
          title: TextStyle(  
            color: Colors.black,  
            fontSize: 24,  
            fontWeight: FontWeight.bold,  
          ),  
        ),  
      ),  
    );  
  }  
}
```

- **Scaffold**



The screenshot shows an IDE with the following Dart code in `main.dart`:

```
28 class MyHomePage extends StatelessWidget {  
29   MyHomePage({Key? key}) : super(key: key);  
30  
31   var counter = 0.obs();  
32   Controller controller = Get.put(Controller());  
33   @override  
34   Widget build(BuildContext context) {  
35     return Scaffold(  
36       appBar: AppBar(  
37         title: const Text('Hello Getx'),  
38       ), // AppBar  
39       body: Center(  
40         child: Column(  
41           mainAxisAlignment: MainAxisAlignment.center,  
42           children: [  
43             GetBuilder<Controller>(  
44               builder: (_) => Text(  
45                 '${controller.counter}',  
46                 style: const TextStyle(fontSize: 30),  
47               ), // Text  
48             ), // GetBuilder  
49             CupertinoButton(  
50               color: Theme.of(context).primaryColor,  
51               child: const Text('Next page'),  
52               onPressed: () => Get.to(() => const Screen2()), // CupertinoButton  
53             ),  
54           ], // Column  
55         ), // Center  
56       ),  
57     );  
58   }  
59 }
```

The app preview on the iPhone 11 shows a blue header with "Hello Getx", a white body with a large "3" and a "Next page" button, and a bottom bar with minus and plus buttons.

```
56 floatingActionButton: Row(  
57   mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
58   children: [  
59     FloatingActionButton(  
60       heroTag: 'Decrement',  
61       onPressed: () => controller.decrement(),  
62       child: const Icon(Icons.remove),  
63     ), // FloatingActionButton  
64     FloatingActionButton(  
65       heroTag: 'Increment',  
66       onPressed: () => controller.increment(),  
67       child: const Icon(Icons.add),  
68     ), // FloatingActionButton  
69   ],  
70 ), // Row  
71 ); // Scaffold  
72 }  
73 }  
74
```

session2* 0 0 0 7 Debug my code Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 3.0.4 iPhone 11 (ios simulator)

ឧទាហរណ៍ ជាមួយនិង List

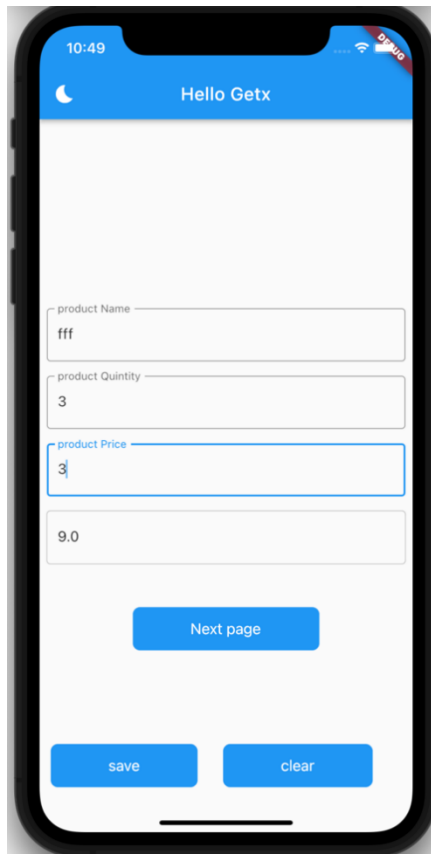
- **ProductController**

```
class ProductController extends GetxController {  
  
  var products = <Product>[].obs();  
  
  addProductCard(Product product) {  
  
    products.add(product);  
  
    print('object added');  
  
    update();  
  
  }  
}
```

- ProductModel

```
class Product {  
    late int code;  
    late String name;  
    late double price;  
    late int qty;  
    late double total;  
    Product(  
        {required this.code,  
        required this.name,
```

- Scaffold




```
main.dart — getx_state management
t_model.dart  getx_controller.dart  screen2.dart  main.dart M  product_getx_


lib > main.dart > _MyHomePageState > build

34 class _MyHomePageState extends State<MyHomePage> {
35   TextEditingController nameController = TextEditingController();
36   TextEditingController priceController = TextEditingController();
37   TextEditingController qtyController = TextEditingController();
38   TextEditingController totoalController = TextEditingController();
39   //-----
40   ProductController productController = Get.put(ProductController());
41
42   @override
43   Widget build(BuildContext context) {
44     return Scaffold(
45       appBar: AppBar(
46         leading: InkWell(
47           child: const Icon(Icons.bedtime),
48           onTap: () {},
49         ), // InkWell
50         title: const Text('Hello Getx'),
51       ), // AppBar
52       body: Center(
53         child: Column(
54           mainAxisAlignment: MainAxisAlignment.center,
55           children: [
56             Padding(
57               padding: const EdgeInsets.all(8.0),
58               child: TextFormField(
59                 controller: nameController,
60                 decoration: const InputDecoration(
61                   label: Text('product Name'), border: OutlineInputBorder(), //
62                 ), // TextFormField
```

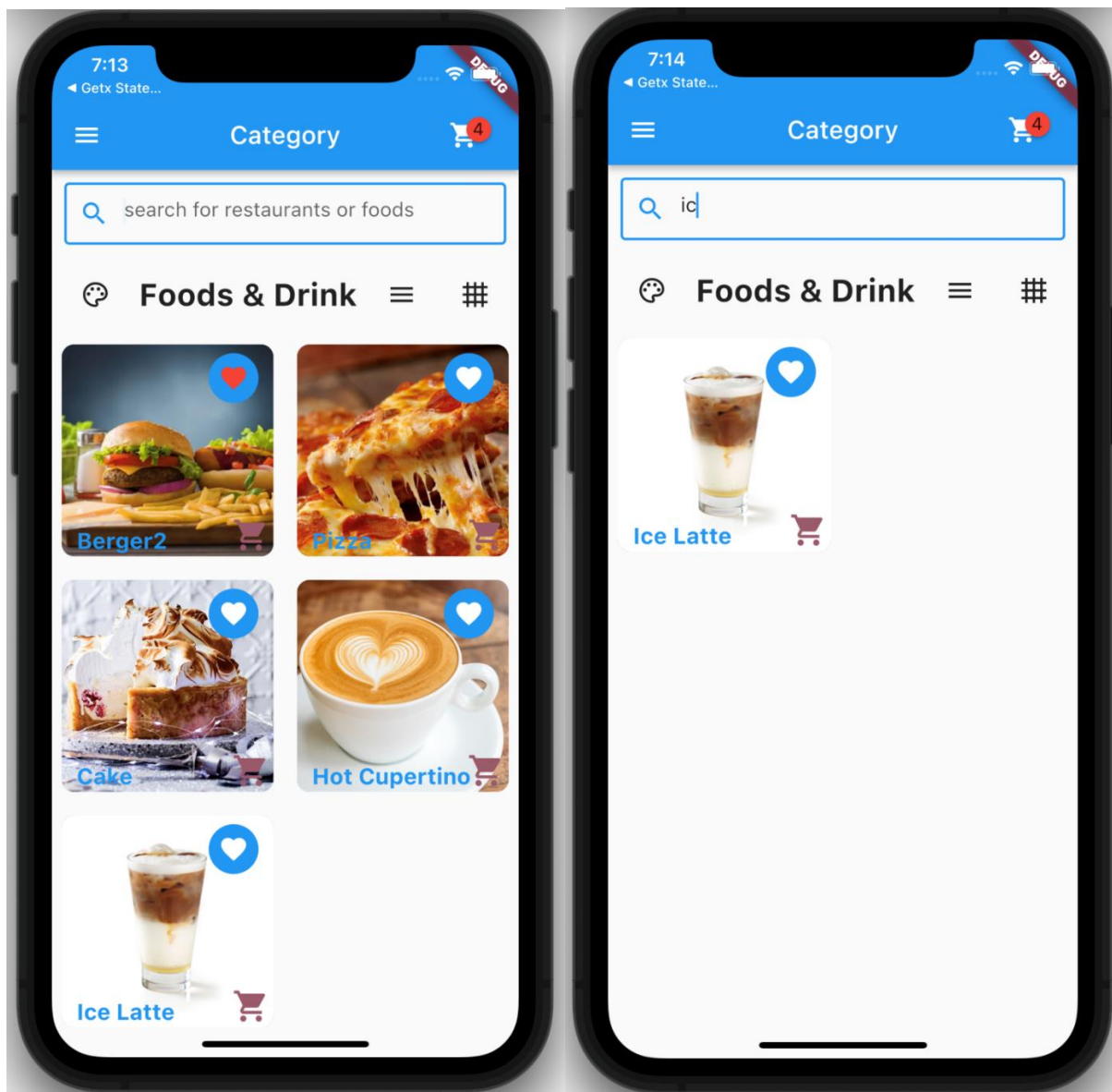
```
63     ), // Padding
64     Padding(
65       padding: const EdgeInsets.all(8.0),
66       child: TextFormField(
67         controller: qtyController,
68         decoration: const InputDecoration(
69           label: Text('product Quantity'),
70           border: OutlineInputBorder(), // InputDecoration
71         ), // TextFormField
72     ), // Padding
73     Padding(
74       padding: const EdgeInsets.all(8.0),
75       child: TextFormField(
76         onChanged: ((newValue) {
77           setState(() {
78             int qty = int.parse(qtyController.text);
79             double price = double.parse(newValue.toString());
80             totoalController.text = (price * qty).toString();
81           });
82         }),
83         controller: priceController,
84         decoration: const InputDecoration(
85           label: Text('product Price'), border: OutlineInputBorder(), //
86         ), // TextFormField
87     ), // Padding
88     Padding(
89       padding: const EdgeInsets.all(8.0),
90       child: TextFormField(
91         enabled: false,
92         controller: totoalController,
93         decoration: const InputDecoration(
94           hintText: 'total', border: OutlineInputBorder(), // InputDecor
95         ), // TextFormField
96     ), // Padding
97     const SizedBox(
98       height: 40,
99     ), // SizedBox
100    CupertinoButton(
101      color: Theme.of(context).primaryColor,
102      child: const Text('Next page'),
103      onPressed: () => Get.to(() => ListProductScreen()), // CupertinoBu
104    ],
105  ), // Column
106 ), // Center
```

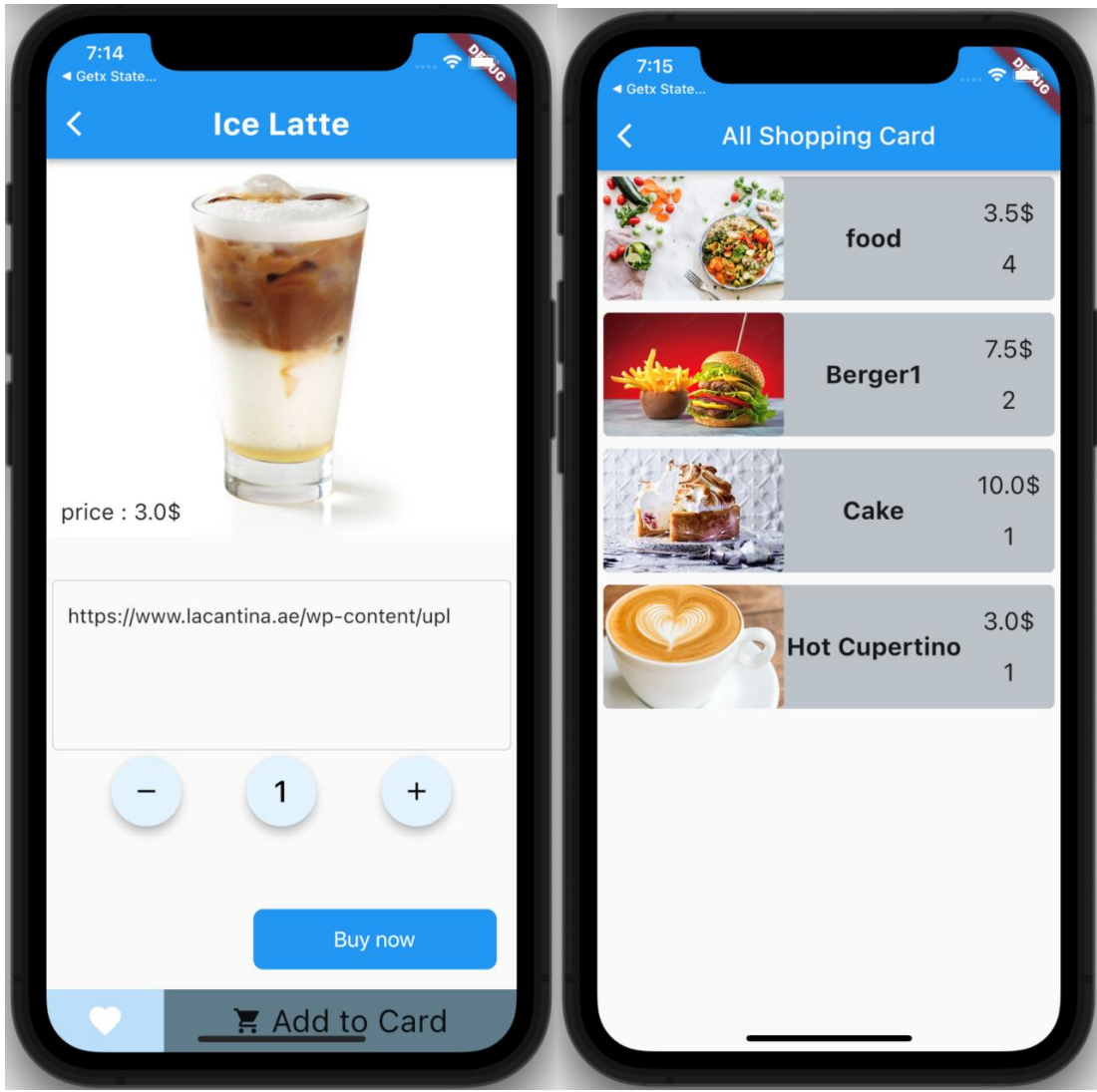
```
107 floatingActionButton: Row(
108   mainAxisAlignment: MainAxisAlignment.spaceEvenly,
109   children: [
110     CupertinoButton(
111       color: Theme.of(context).primaryColor,
112       child: const Text('save'),
113       onPressed: () => productController.addProductCard(Product(
114         code: Random().nextInt(200),
115         name: nameController.text,
116         price: double.parse(priceController.text),
117         qty: int.parse(qtyController.text),
118         total: double.parse(totoalController.text),
119       )), // Product // CupertinoButton
120     CupertinoButton(
121       color: Theme.of(context).primaryColor,
122       child: const Text('clear'),
123       onPressed: () {}), // CupertinoButton
124   ],
125 ), // Row
126 ); // Scaffold
127 }
128 }
```

```
5 class ListProductScreen extends StatelessWidget {
6   ListProductScreen({Key? key}) : super(key: key);
7   final ProductController productController = Get.find();
8   @override
9   Widget build(BuildContext context) {
10    return Scaffold(
11      appBar: AppBar(
12        title: const Text('List Product'),
13      ), // AppBar
14      body: ListView.builder(
15        itemCount: productController.products.length,
16        itemBuilder: (context, index) {
17          var pro = productController.products[index];
18          return ListTile(
19            leading: CircleAvatar(
20              child: Text(pro.code.toString()),
21            ), // CircleAvatar
22            title: Text(
23              pro.name,
24              style: const TextStyle(fontSize: 25),
25            ), // Text
26            subtitle: Text(pro.total.toString()),
27          ); // ListTile
28        },
29      ), // ListView.builder
30    ); // Scaffold
31  }
32 }
```



Apply with GetX





**“ពេលវេលាមានតំលៃណាស់, ត្រូវចេះថែរក្សា
ពេលវេលាអោយបានល្អ និងច្បាស់លាស់ម្តងៗ
និងមានតំលៃតាមពេលវេលាទាំងនោះ”**

—គ្រូអោយធីចិត្តល្អ