

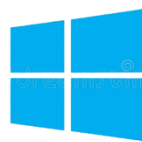


Build Your IT Skill

ETEC IT PROFESSIONAL TRAINING CENTER

Local Storage

Flutter MVC & MVVM



រៀបចំដោយ៖ គា ជាតំណាង
Software Developer

I. About Local Storage

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

II. SharedPreferences

Package

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

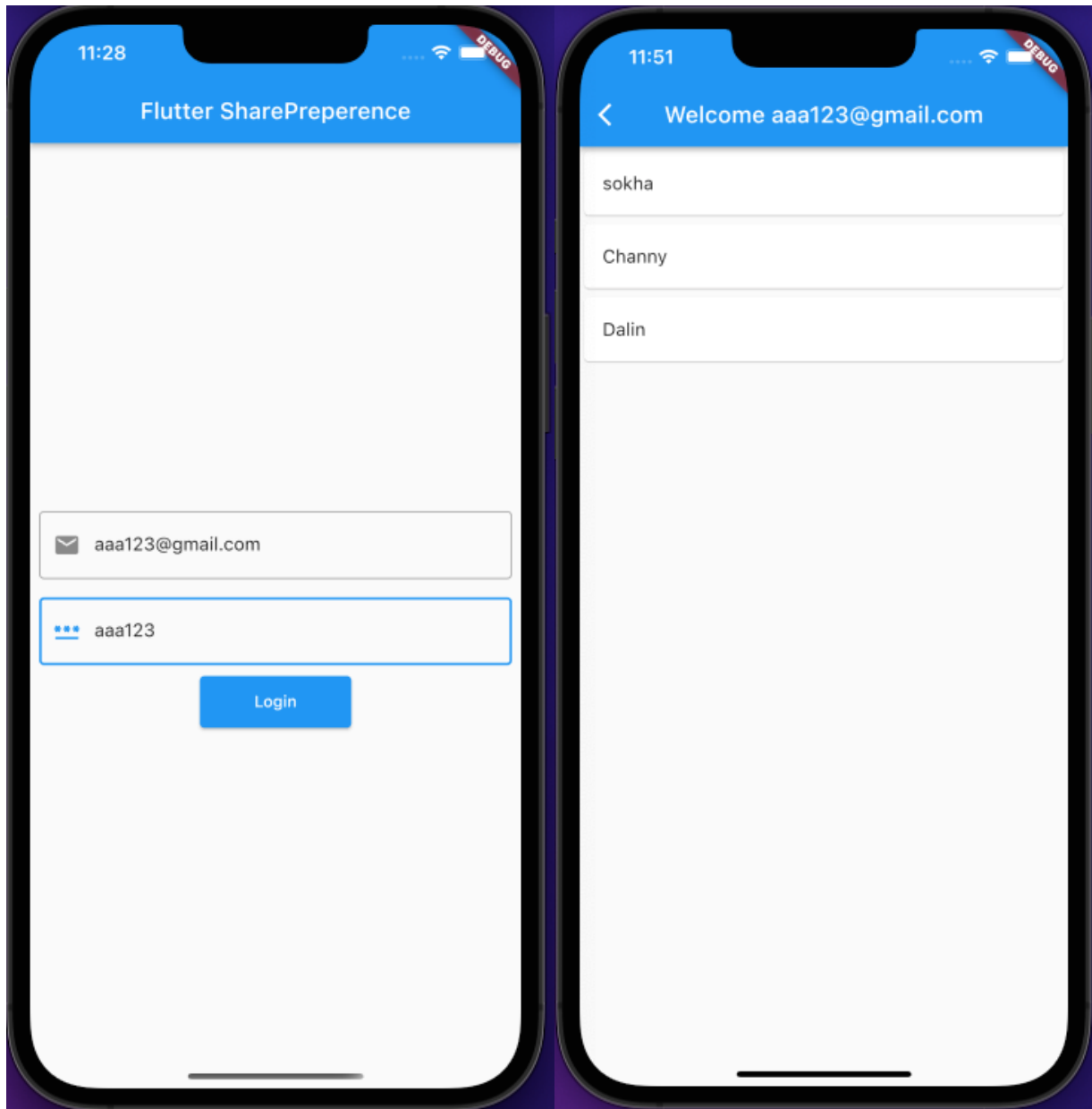
setData

```
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);  
  await prefs.setStringList('name', ['Sokha', 'Channy', 'Dalin']);  
}
```

getData

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

Example UI



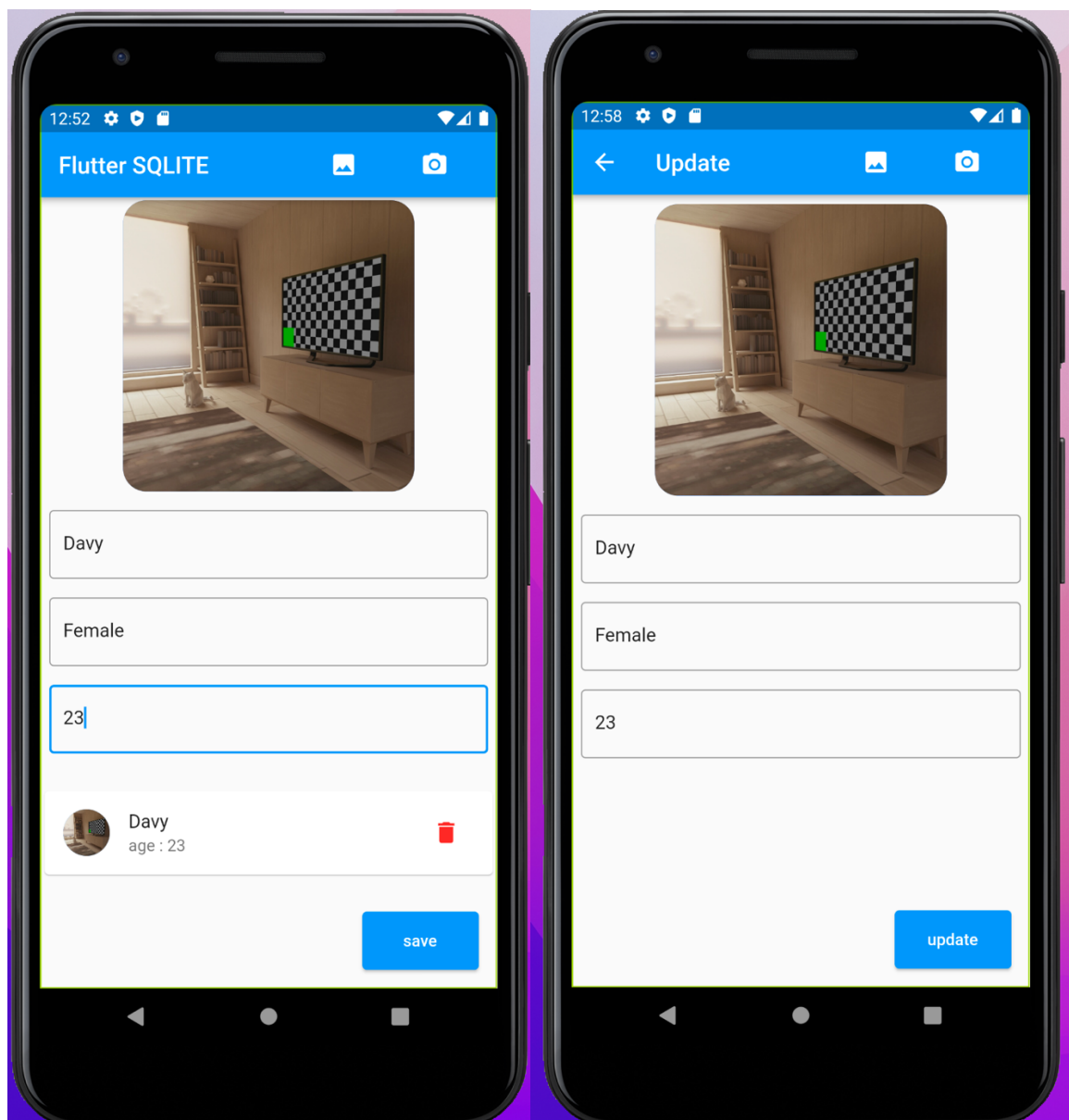
III. SQLITE

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

CRUD Sqlite (Create, read, update, delete)

```
dependencies:  
  sqflite: ^2.0.3  
  image_picker: ^0.8.5+3  
  path_provider: ^2.0.11  
  flutter:  
    sdk: flutter
```

Example UI



Model



ConnectionDatabase

```
import 'dart:io';
import 'package:path_provider/path_provider.dart';
import 'package:sqflite/sqflite.dart';
import 'package:sqflite/sqlite_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.fromJson());
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.toJson(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.fromJson(),
48       where: 'id=?', whereArgs: [person.id]);
49   }
50 }
51

```

0 23 Cloud Code Connect to Google Cloud Dart Dart DevTools Flutter: 3.0.4 iPhone 13 Pro Max (ios simulator)

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),  
        image: _image!.path))  
      .whenComplete(() => _onRefresh());  
  },  
,  
,  
,
```

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

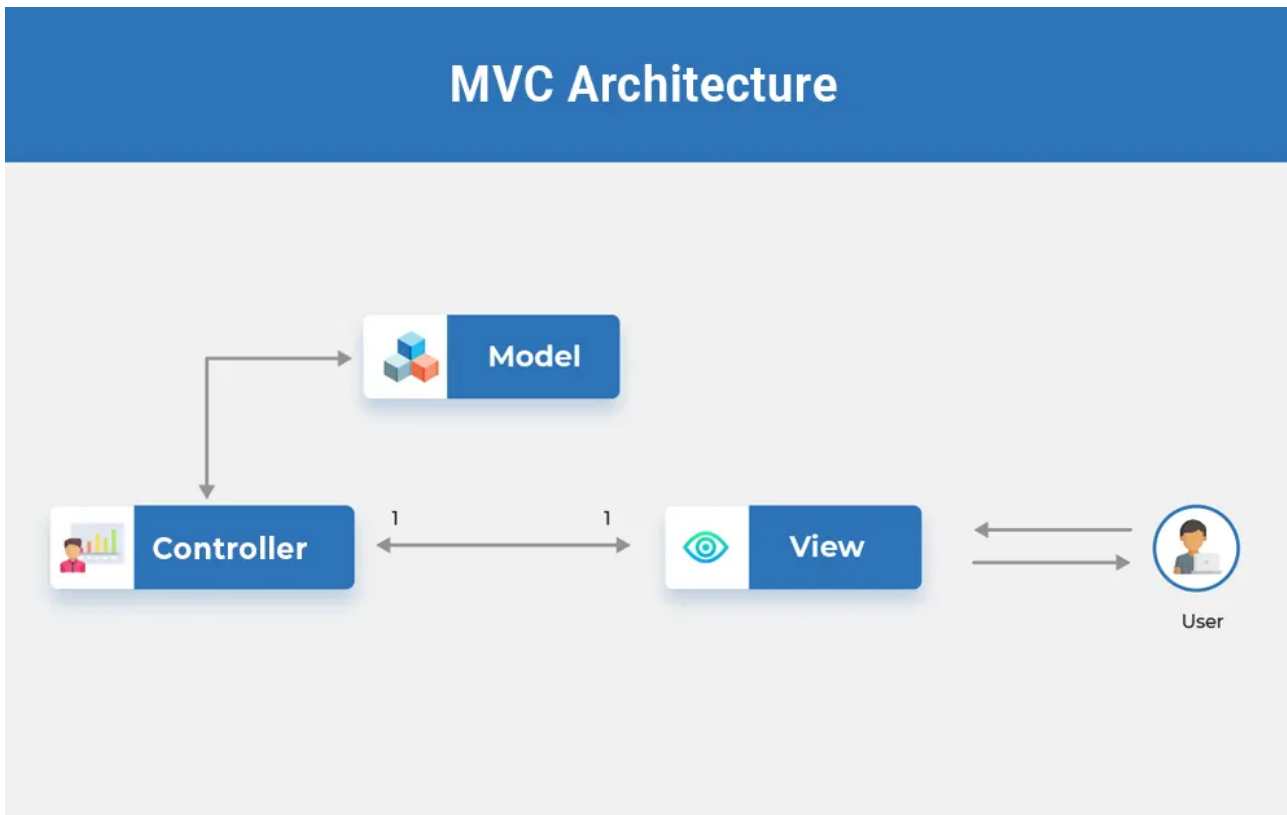
```

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,
        MaterialPageRoute(
          builder: (context) => const MyApp(),
        ),
        (route) => false));
  },
),

```

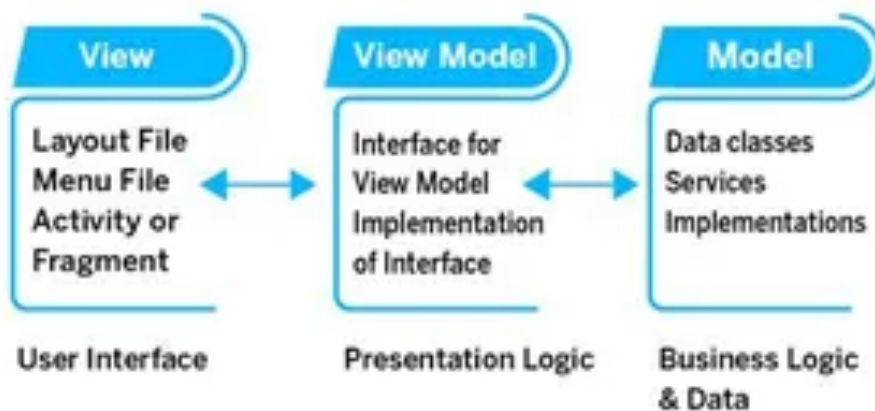

Flutter MVC & MVVM

MVC : Model – View - Controller



MVVM : Model – View – View Model

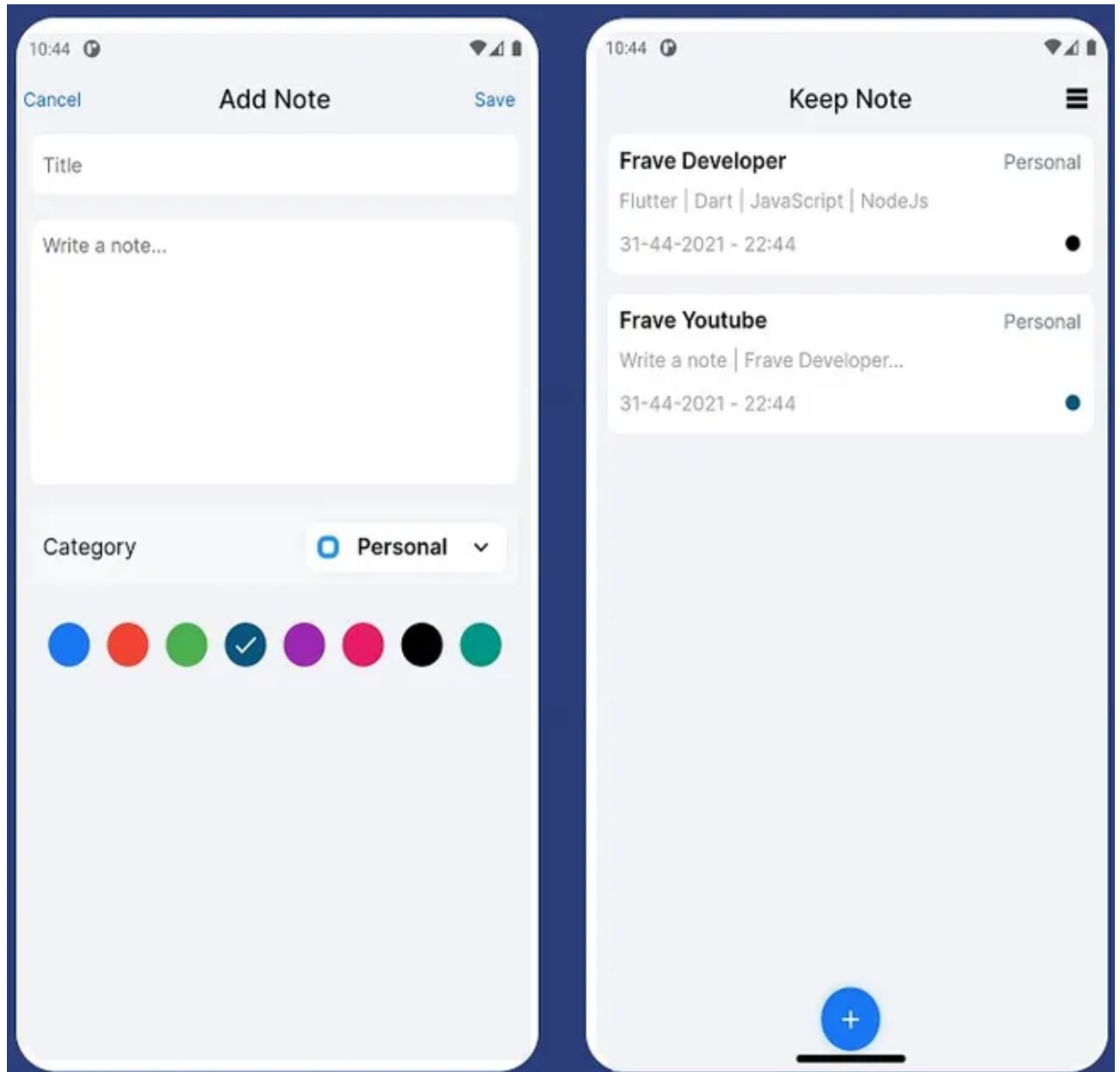
Flow Chart of MVVM



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

Practice and research

- Create App Note Using SQLite (CRUD)



- Change App Icon
- Change App Name

Thank You!! Good luck !!