

Section : IT Systems Development (DSI)
Module : Framework cross-platform workshop
Teaching unit : Mobile Developpment and Web
Level : 3rd Year (Applied license: LMD)

Workshop 3: Managing form

Objective

Manage fields and form with Flutter for states and validation

Technical requirements

In order to start with Flutter, we need a few tools:

- A PC with a recent Windows version, or a Mac with a recent version of the macOS or Linux operating system. we can also use a Chrome OS machine, with a few tweaks.
- An Android/iOS setup.
- The Flutter SDK. It's free, light, and open source.
- Physical device/emulator/simulator
- Android Studio/IntelliJ IDEA or Visual Studio Code

What to do

- 1- create new flutter project named TP3
- 2- Open your simulator or Android emulator.
- 3- Create à new flutter project
- 4- Create the first new page as follow:

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

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

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        body: Column(
          children: [],
        ),
      ),
    );
  }
}
```

Scaffold is a graphical object widget which has two attributes: AppBar and body

5- Run your project

6- To dismiss the demo label, add the primitive:

```
debugShowCheckedModeBanner: false,
```

after

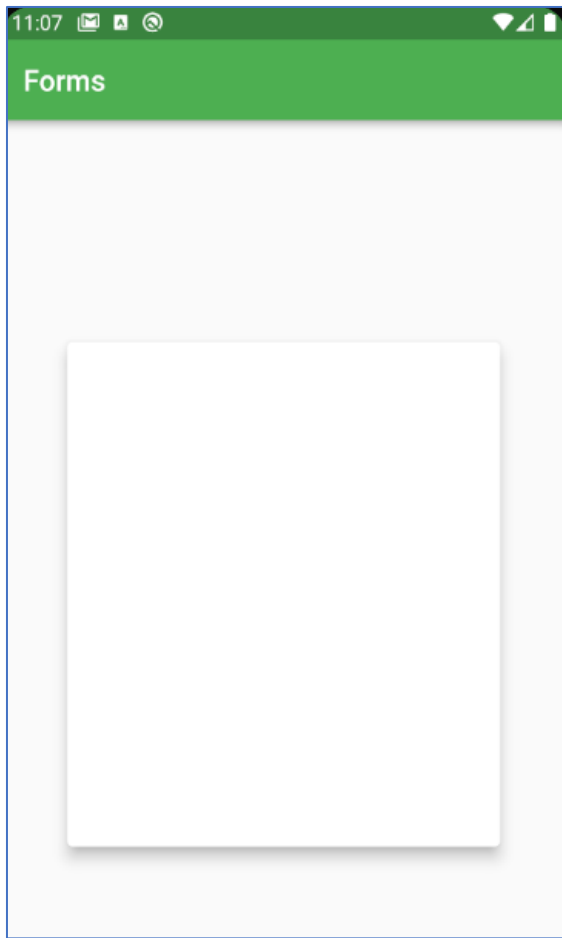
```
return MaterialApp(
```

7- Add to your application a green bar and a title as “Forms” with:

```
appBar: AppBar(
  backgroundColor: Colors.green,
  title: Text("Forms"),
),
```

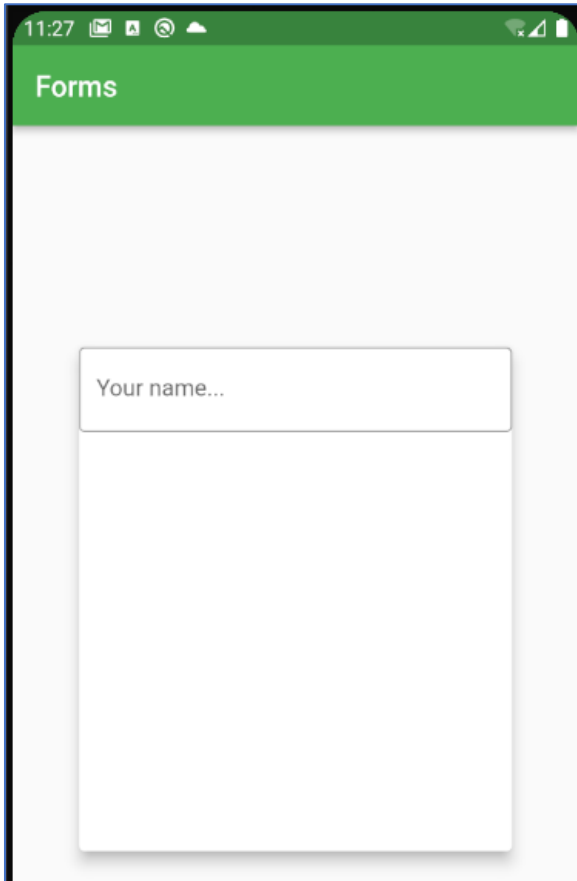
8- Create a card in your form. You can use for example:

```
class _MyAppState extends State<MyApp> {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(
          backgroundColor: Colors.green,
          title: Text("Forms"),
        ),
        body: Center(
          child: Card(
            elevation: 8.0,
            child: Container(
              height: 350.0,
              width: 300.0,
            ),
          ),
        ),
      ),
    );
  }
}
```



```
class _MyAppState extends State<MyApp> {  
  TextEditingController _nameController = TextEditingController();  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      home: Scaffold(  
        appBar: AppBar(  
          backgroundColor: Colors.green,  
          title: Text("Forms"),  
        ),  
        body: Center(  
          child: Card(  
            elevation: 8.0,  
            child: Container(  
              height: 350.0,  
              width: 300.0,  
              child: Column(  
                children: [  
                  TextFormField(  
                    controller: _nameController,  
                    decoration: InputDecoration(  
                      hintText: "Your name...",  
                      border: OutlineInputBorder(),  
                    ),  
                ],  
              ),  
            ),  
          ),  
        ),  
      ),  
    ),  
  ),  
)
```

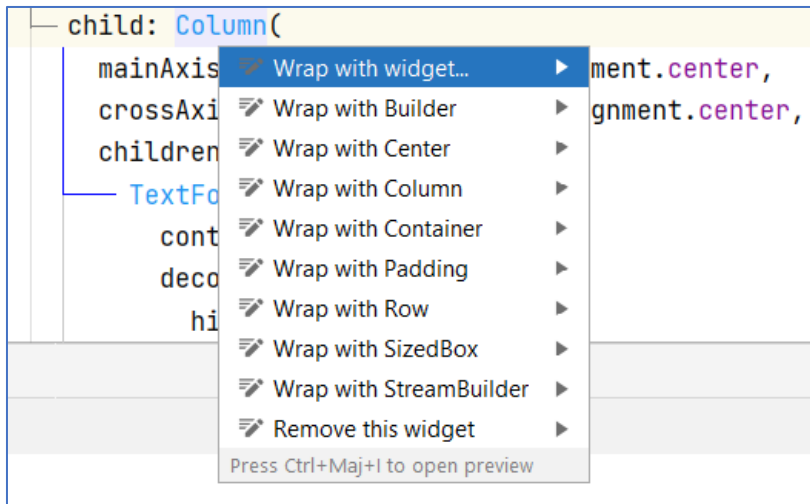
```
    ),  
  ),  
),  
);  
}  
}
```



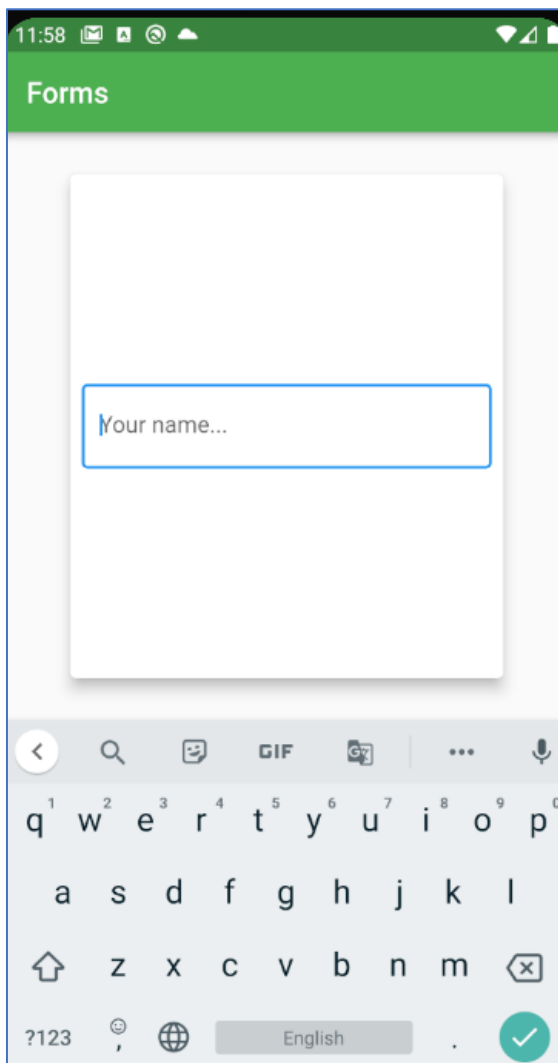
```
child: Column(  
  mainAxisAlignment: MainAxisAlignment.center,  
  crossAxisAlignment: CrossAxisAlignment.center,  
  children: [  
    TextFormField(  

```

9- On the Column child press alt+enter and choose Wrap with Padding



Your AVD will display

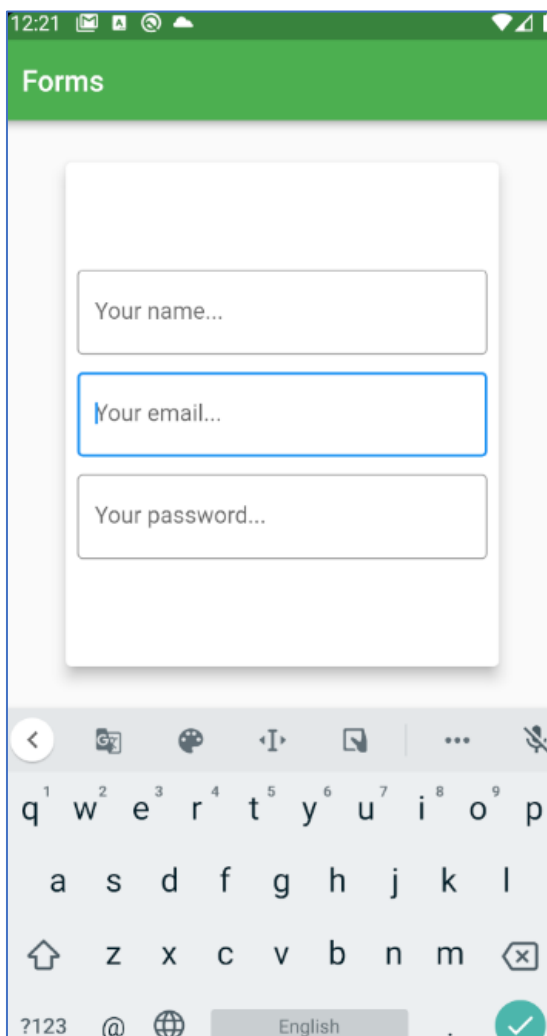


10- Add two other fields for email and password

```
TextEditingController _emailController = TextEditingController();
TextEditingController _passwordController = TextEditingController();
```

11- Use SizedBox to separate these fields

```
TextFormField(
  controller: _emailController,
  keyboardType: TextInputType.emailAddress,
  decoration: InputDecoration(
    hintText: "Your email...",
    border: OutlineInputBorder(),
  ),
),
SizedBox(
  height: 12.0,
),
TextFormField(
  controller: _passwordController,
  obscureText: true,
  decoration: InputDecoration(
    hintText: "Your password...",
    border: OutlineInputBorder(),
  ),
),
```

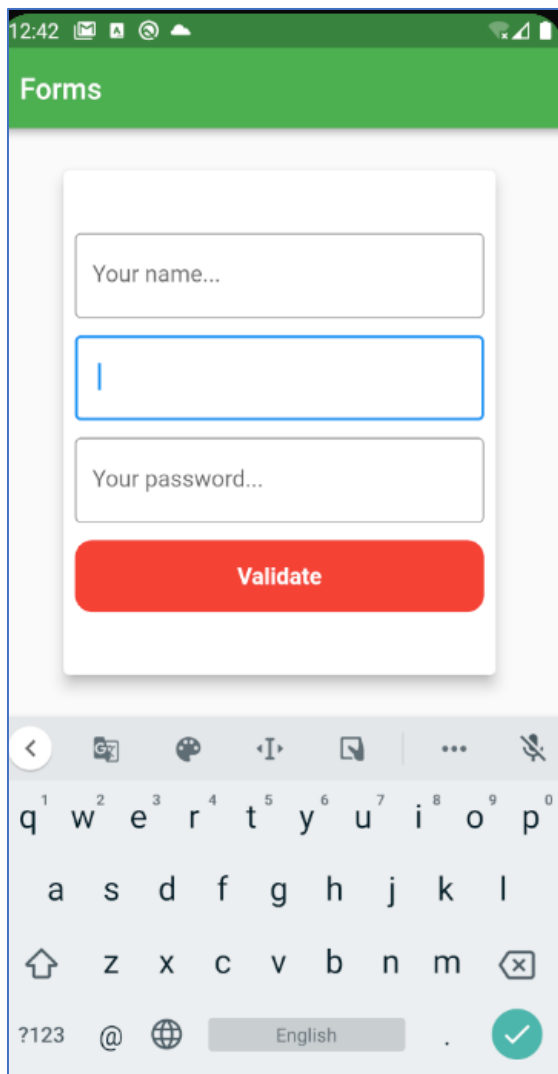


12- Create a button with the following values:

```

    SizedBox(
      height: 12.0,
    ),
    InkWell(
      child: Container(
        height: 50.0,
        decoration: BoxDecoration(
          color: Colors.red,
          borderRadius:
            BorderRadius.all(Radius.circular(12.0)),
        ),
      child: Center(
        child: Text(
          "Validate",
          style: TextStyle(
            color: Colors.white,
            fontSize: 16.0,
            fontWeight: FontWeight.w700,
          ),
        ),
      ),
    ),
  ),
),
)

```



13- Add validators for each field.

For the name field the validator is:

```
validator: (String value) {
  if (value.isEmpty) {
    return "Please enter your name...";
  }
  return null;
},
```

For the email field the validator is:

```
validator: (String value) {
  if (value.isEmpty || !value.contains(".")) {
    return "Please enter valid email address!";
  }
  return null;
},
```

For the password field the validator is:

```
validator: (String value) {
  if (value.isEmpty || value.length < 6) {
    return "Please enter strong password: (";
```

```

    }
    return null;
  },
  14- Add the name of function _validateFields

```

```

InkWell(
  onTap: _validateFields,

```

Add a Form as a child with the key: `_formKey`,

```

child: Padding(
  padding: const EdgeInsets.all(8.0),
  child: Form(
    key: _formKey,
    child: Column(

```

The function must be as follow

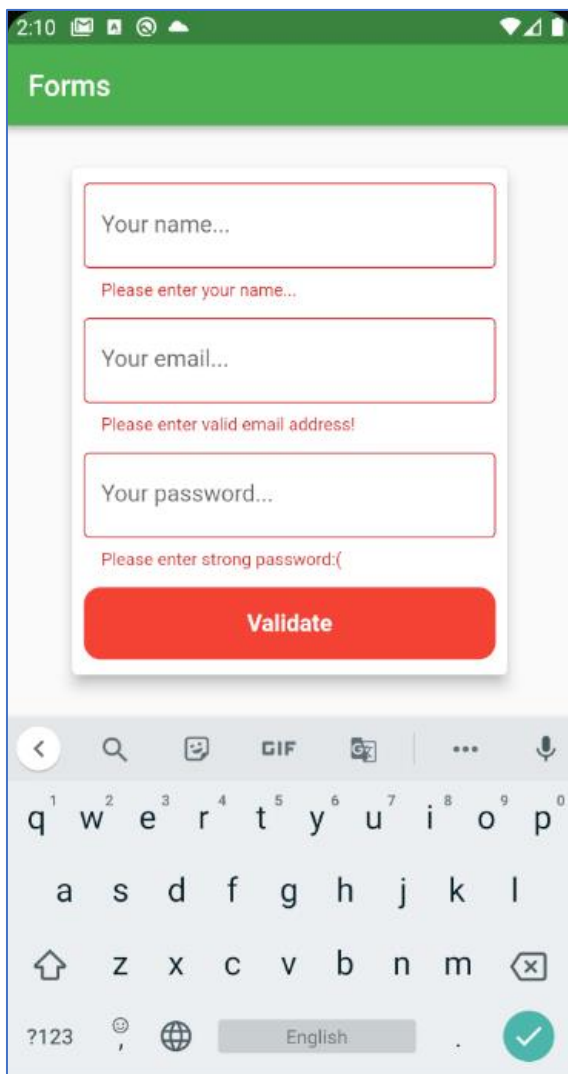
```

final _formKey = GlobalKey<FormState>();

_validateFields() {
  if (_formKey.currentState.validate()) {}
}

```

When you click on the validate button you obtain :



Update `_validateFields()` function to display a snack bar for the message

```
final GlobalKey<ScaffoldState> scaffoldKey = GlobalKey<ScaffoldState>();

_validateFields() {
  final validSnackBar = SnackBar(
    content: Text(
      'Your fields are correct',
      style: TextStyle(fontSize: 16.0),
    ),
    action: SnackBarAction(
      label: "OK",
      onPressed: () {},
    ),
    backgroundColor: Colors.green,
  );
  final invalidSnackBar = SnackBar(
    content:
      Text('Your fields are not correct', style: TextStyle(fontSize:
16.0)),
    action: SnackBarAction(
      label: "Not Valid!",
      textColor: Colors.white,
      onPressed: () {},
    ),
    backgroundColor: Colors.red,
  );
  if (_formKey.currentState.validate()) {
    scaffoldKey.currentState.showSnackBar(validSnackBar);
  } else {
    scaffoldKey.currentState.showSnackBar(invalidSnackBar);
  }
}
```

and don't forget to add **key: scaffoldKey**, under `home: Scaffold(`

```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner: false,
    home: Scaffold(
      key: scaffoldKey,
      appBar: AppBar(
        backgroundColor: Colors.green,
        title: Text("Forms"),
      ),
    ),
  );
}
```

The screenshot shows a mobile application interface with a green header bar labeled "Forms". Below the header is a white form card with three input fields. The first field contains "aaaa", the second contains "faouzi76é001@yahoo.fr", and the third contains ".....". A blue cursor is positioned at the end of the third field. Below the fields is a red button labeled "Validate". At the bottom of the screen, a green bar displays the message "Your fields are correct" and a blue "OK" button.

3:14

Forms

aaaa

faouzi76é001@yahoo.fr

.....

Validate

Your fields are correct OK