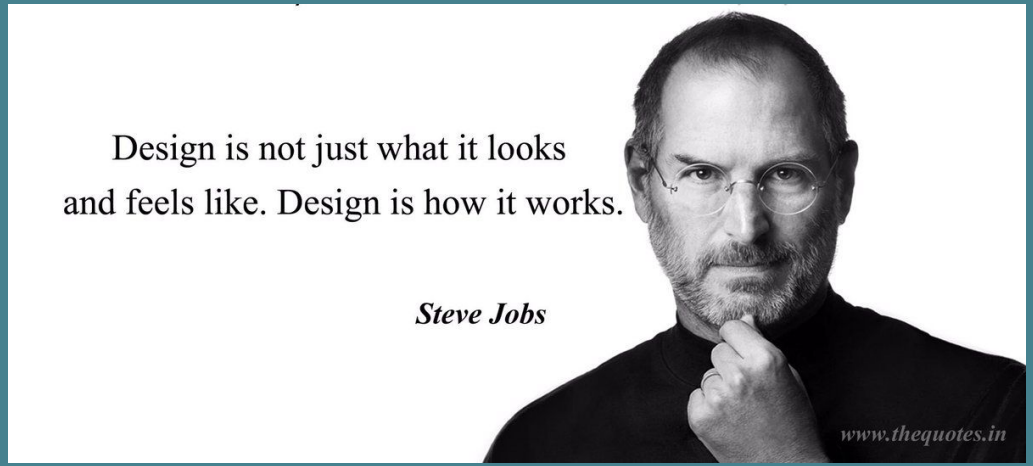
Software Design Patterns

MVC Pattern

Team Members:

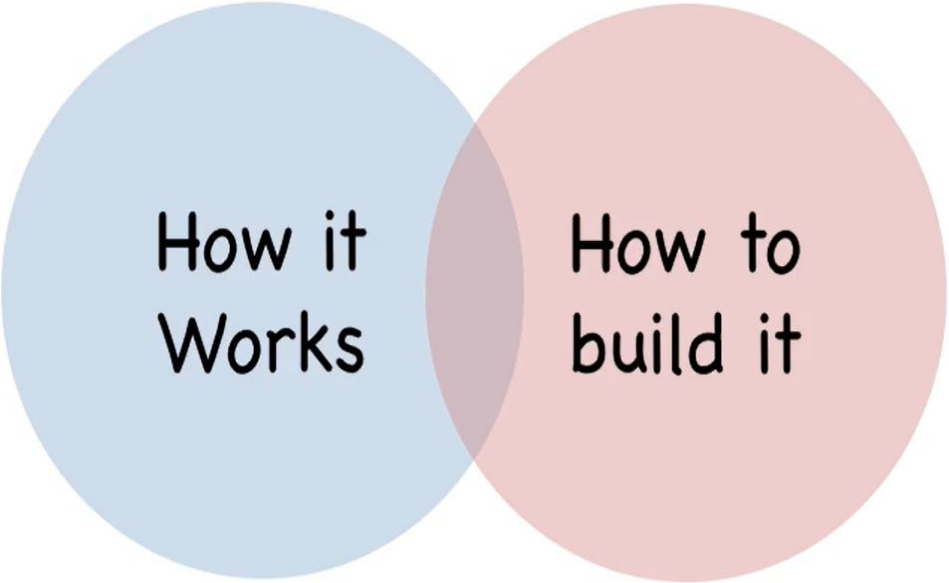
- Mohammed ALI (Developer, designer)
- Safak ALPAY (Designer, Architect, Tester)
- Sencer (left the class)



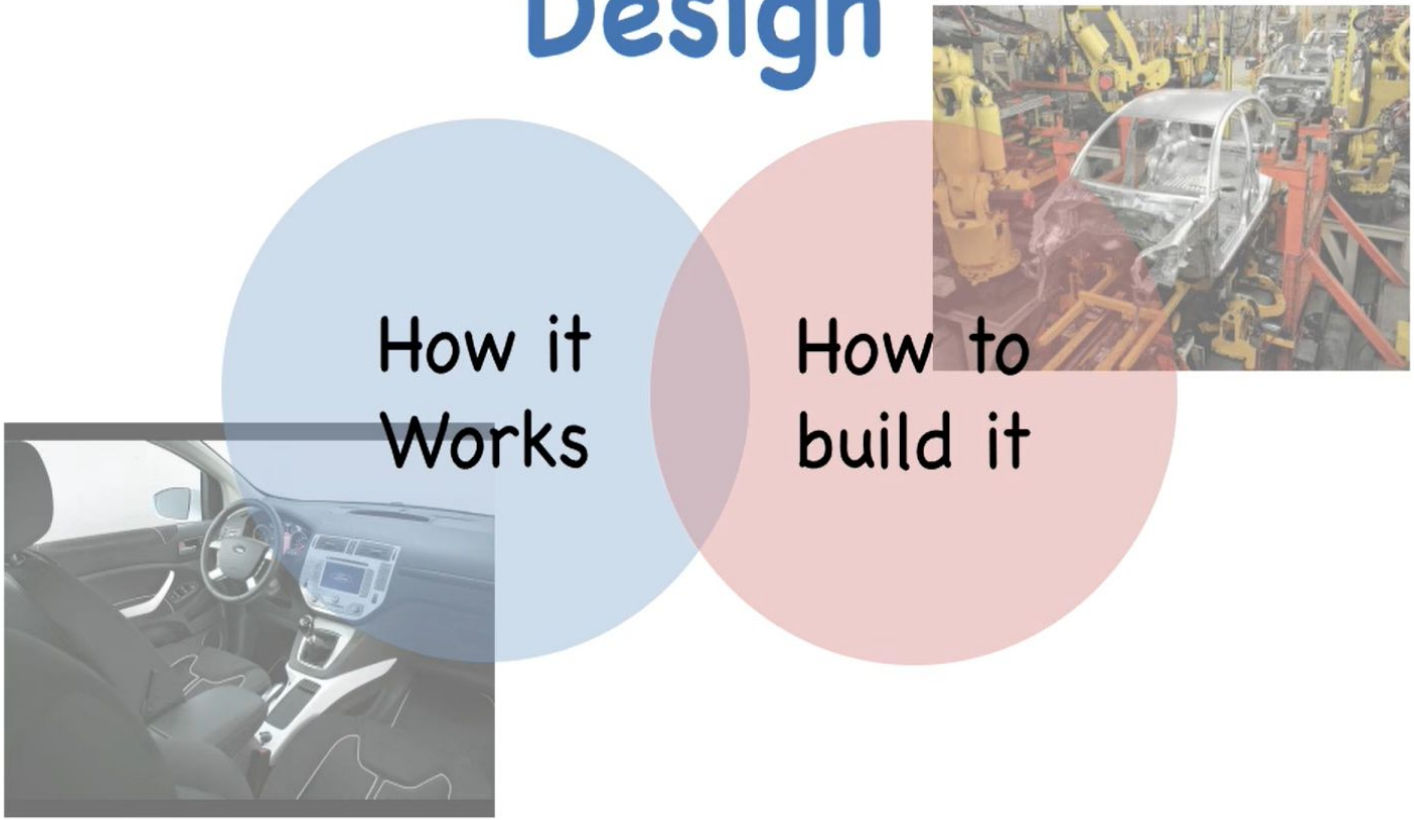
"Good design goes to heaven;
bad design goes everywhere."

Mieke Gerritzen

The Two Aspects of Design



The Two Aspects of Design



Why do need a design architecture patterns?

- Complexity in large projects
- To scale
- To manage states

State Management

Some architecture patterns?

- MVC (Model View Controller)
- SetState
- BloC
- Redux
- Mobex
- inherited widget

- What is the best architecture?

MVC Pattern

Why MVC?

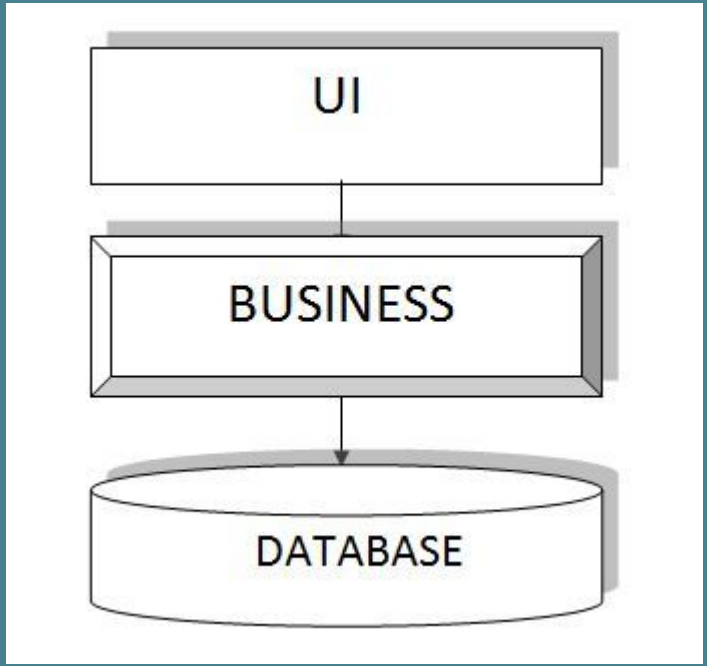
- Most common used architecture pattern
-
- Developed by Apple
- Convenient for App development
- Needed design patterns can be used within
- communication between coders, designers and stakeholders easier

Advenages:

- Easy to Modify
  - The Modification Never Affects The Entire Model
- Ideal for large size of Apps
- Supports Asynchronous Technique
- Easier to Update
- Easy for multiple developers to collaborate and work together
-

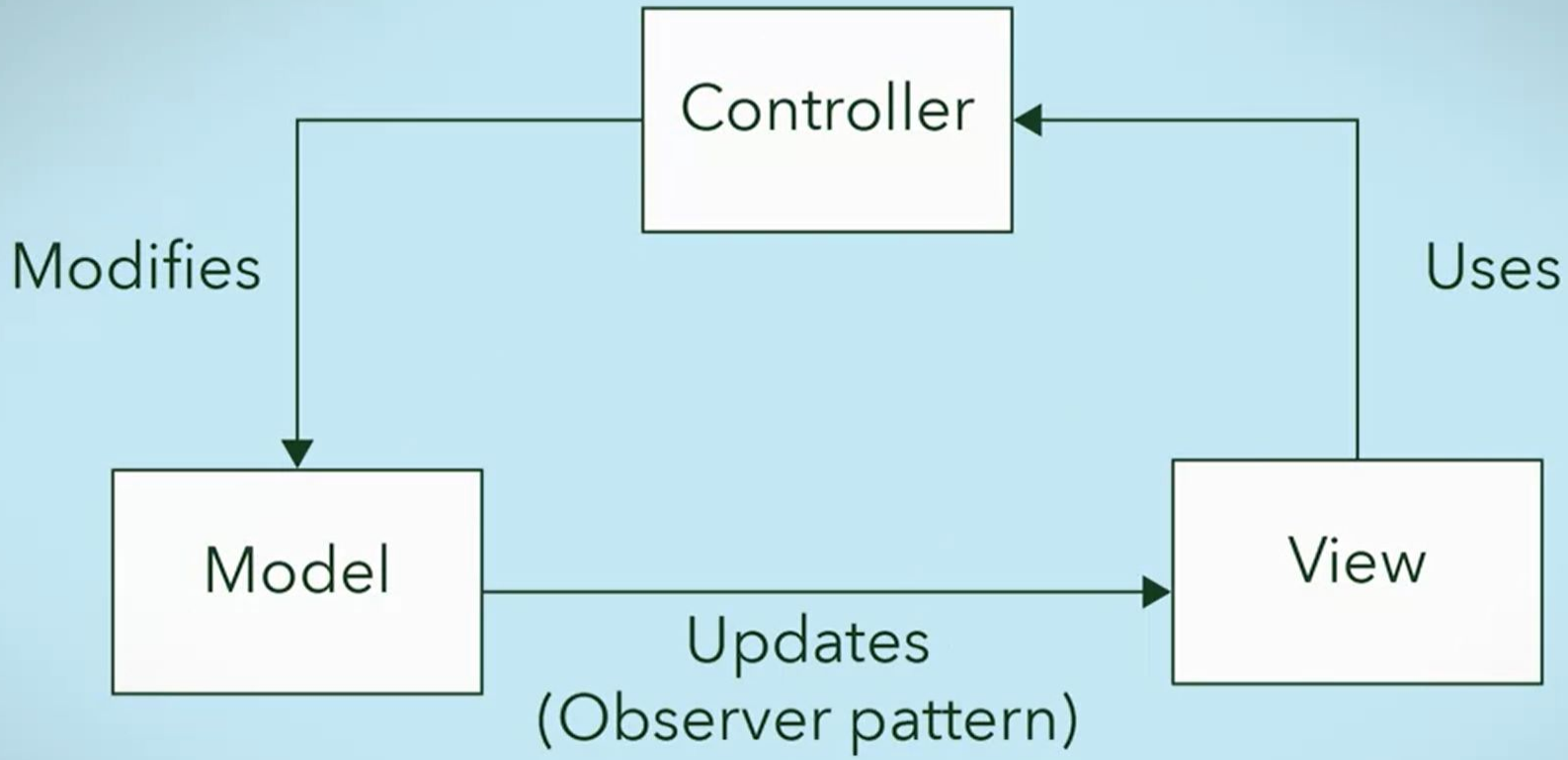
Separate Concerns (SoC)

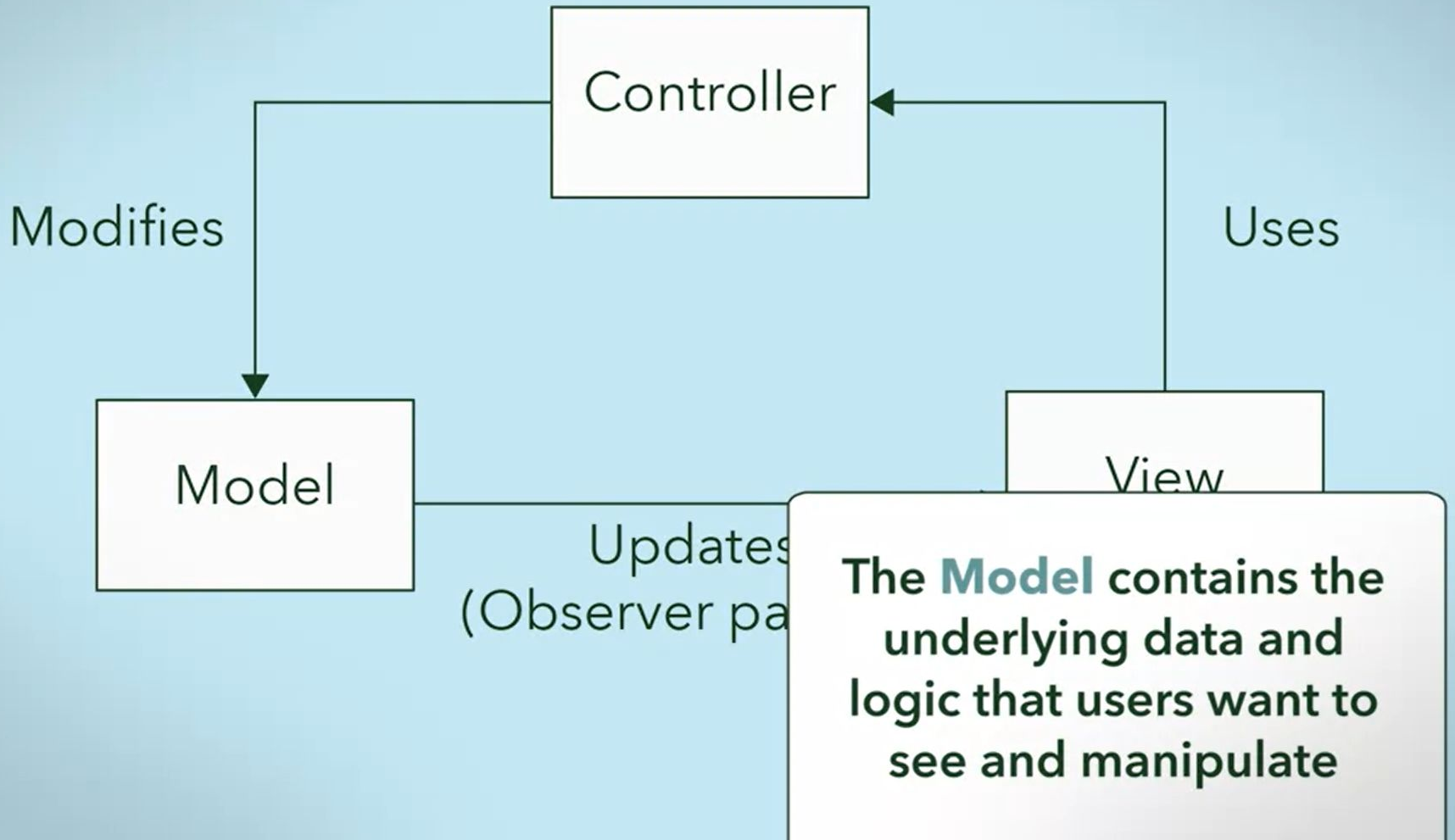
- Each section addresses a separate concern
- Concern is affects the code of a computer program
- Easy to Corporate

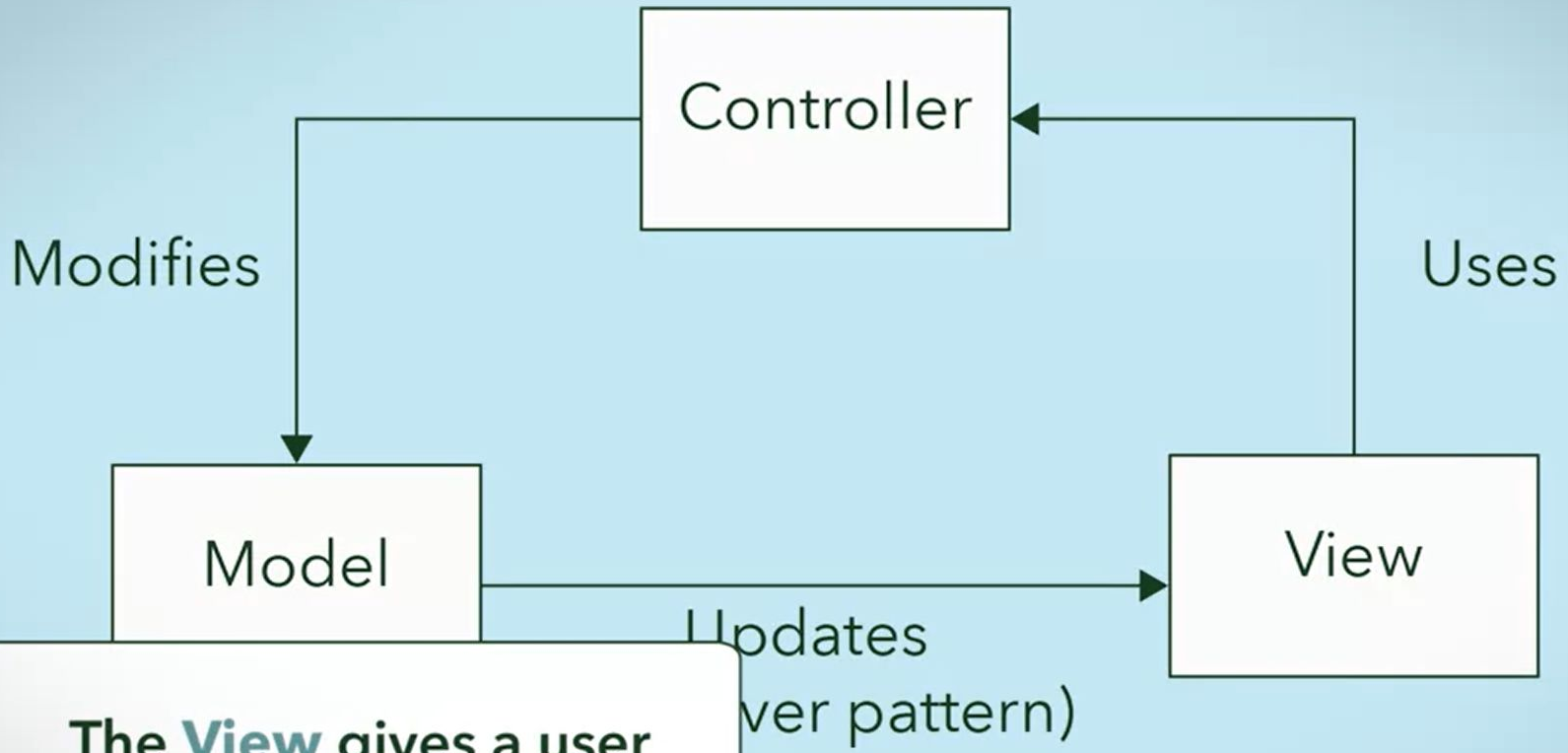


Disadvantages:

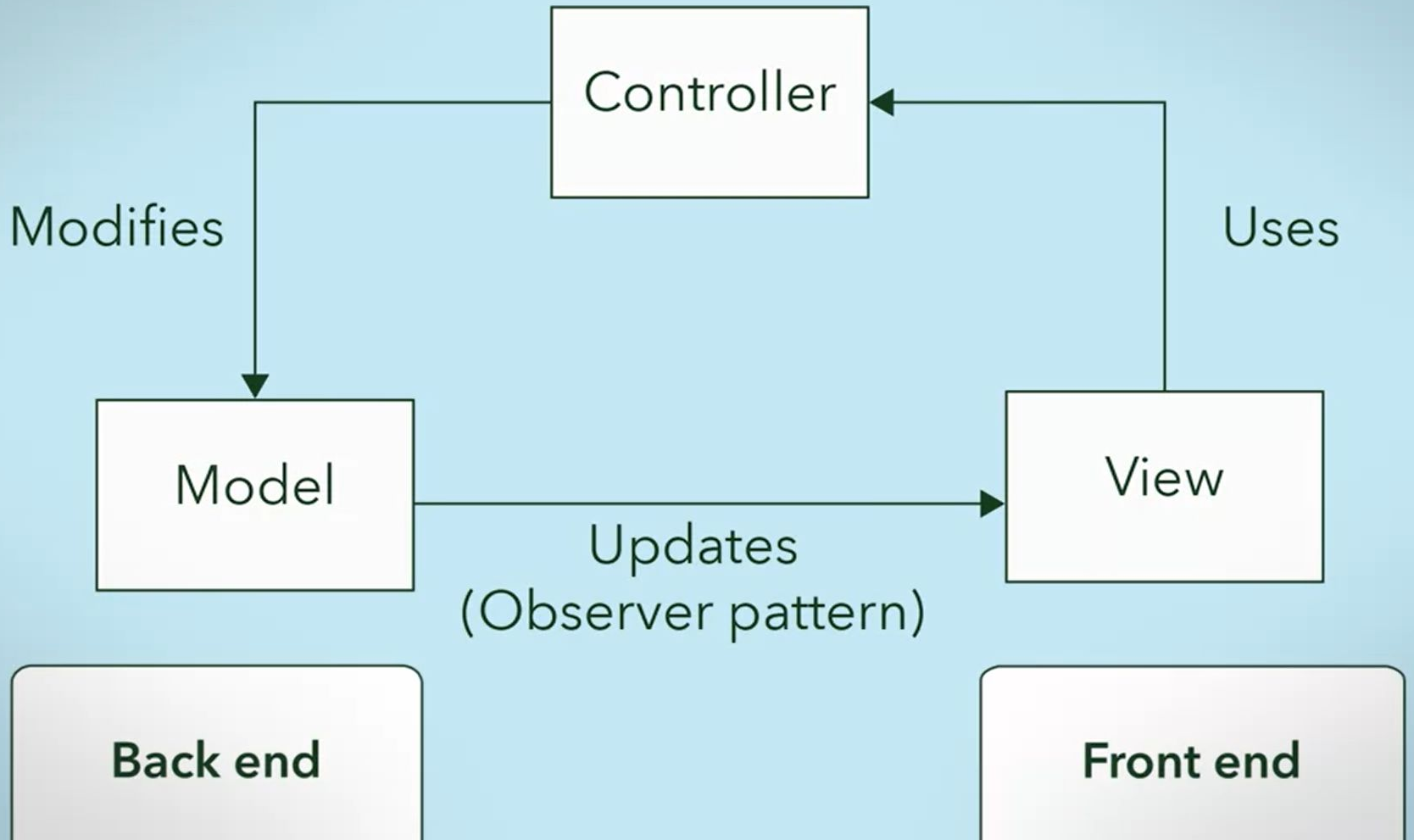
- Not Suitable for small applications
- Passing data between layers sometimes confusing if they are large

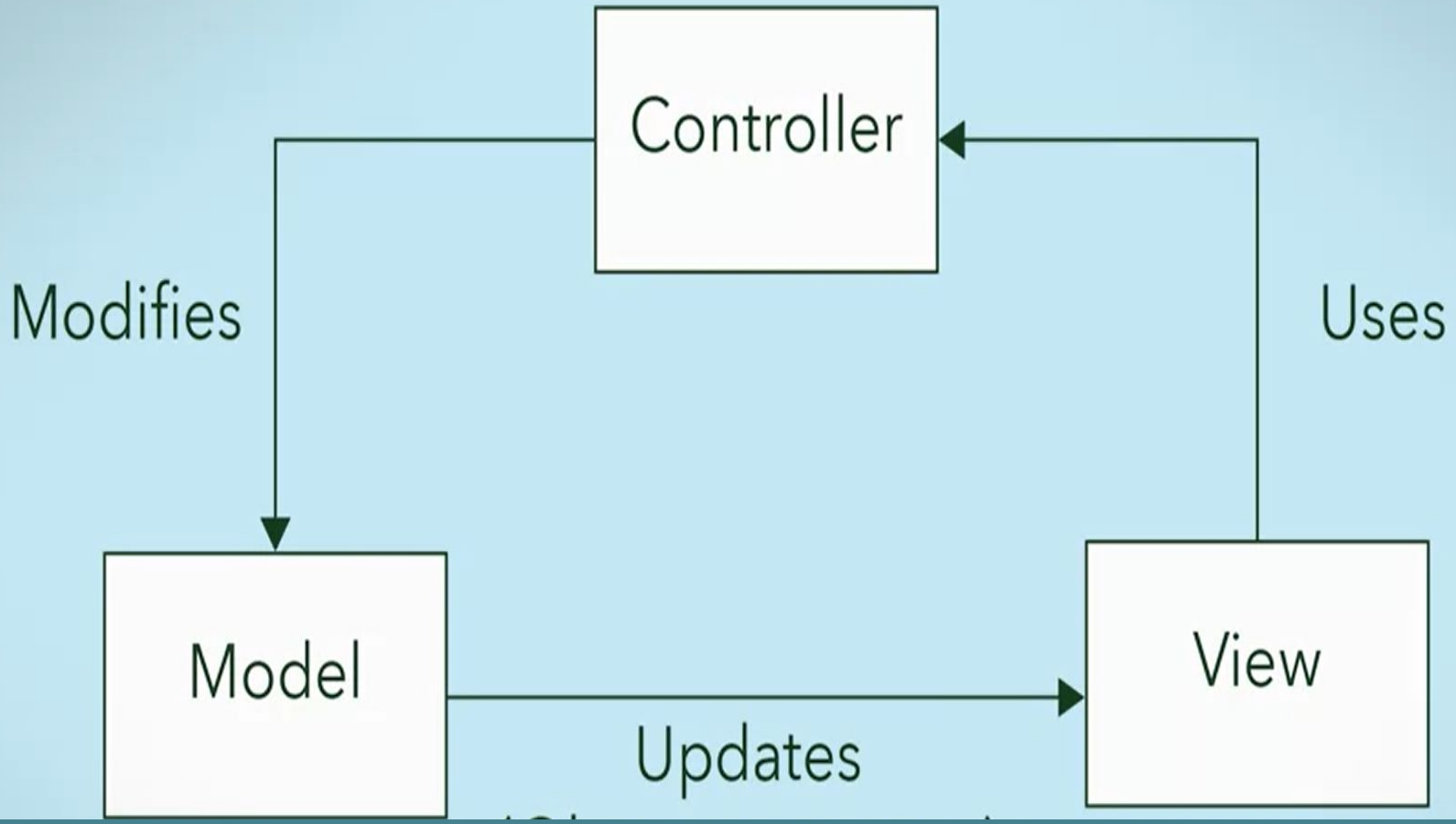

Controller
Modifies
Uses
Model
View
Updates
(Observer pattern)

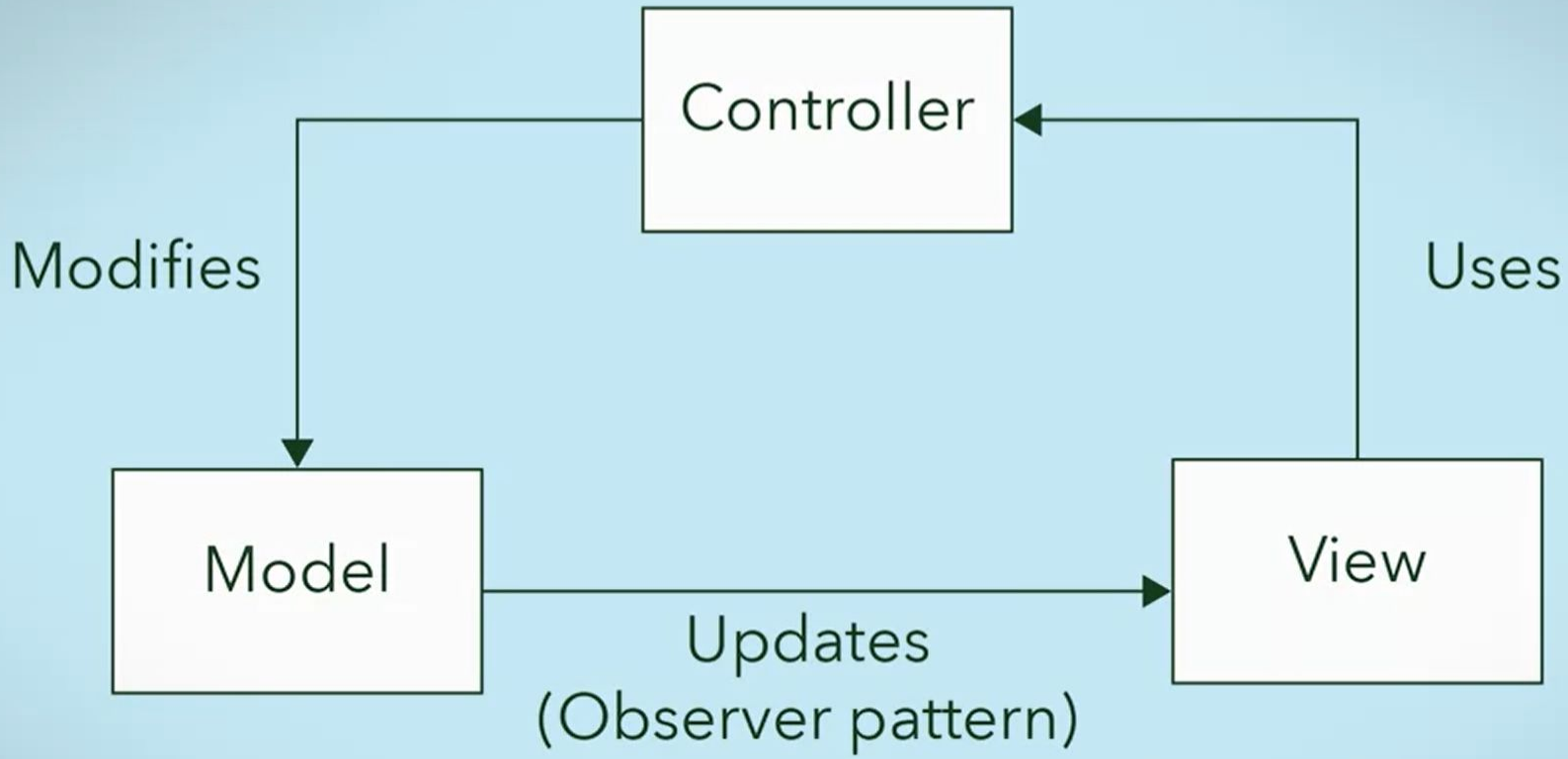

Controller
Modifies
Uses
Model
View
Updates
(Observer pa
The Model contains the underlying data and logic that users want to see and manipulate



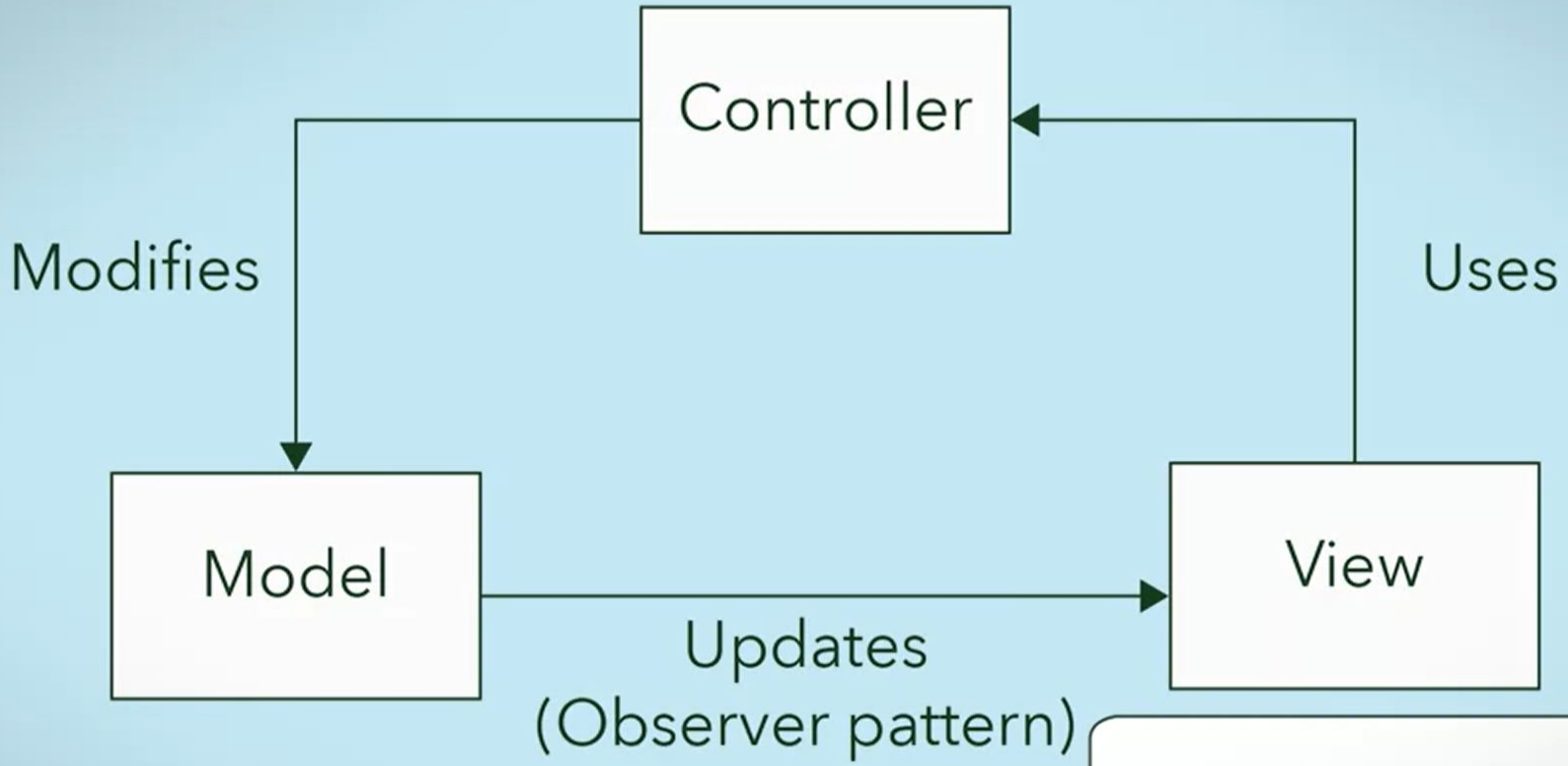
The View gives a user a way to see the model, or at least parts of it


Controller
Modifies
Uses
Model
View
Updates
(Observer pattern)
Back end
Front end


Controller
Modifies
Uses
Model
View
Updates


Controller
Modifies
Uses
Model
View
Updates
(Observer pattern)

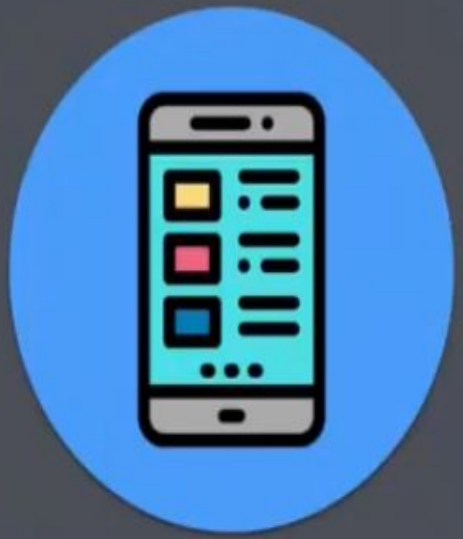
Controller



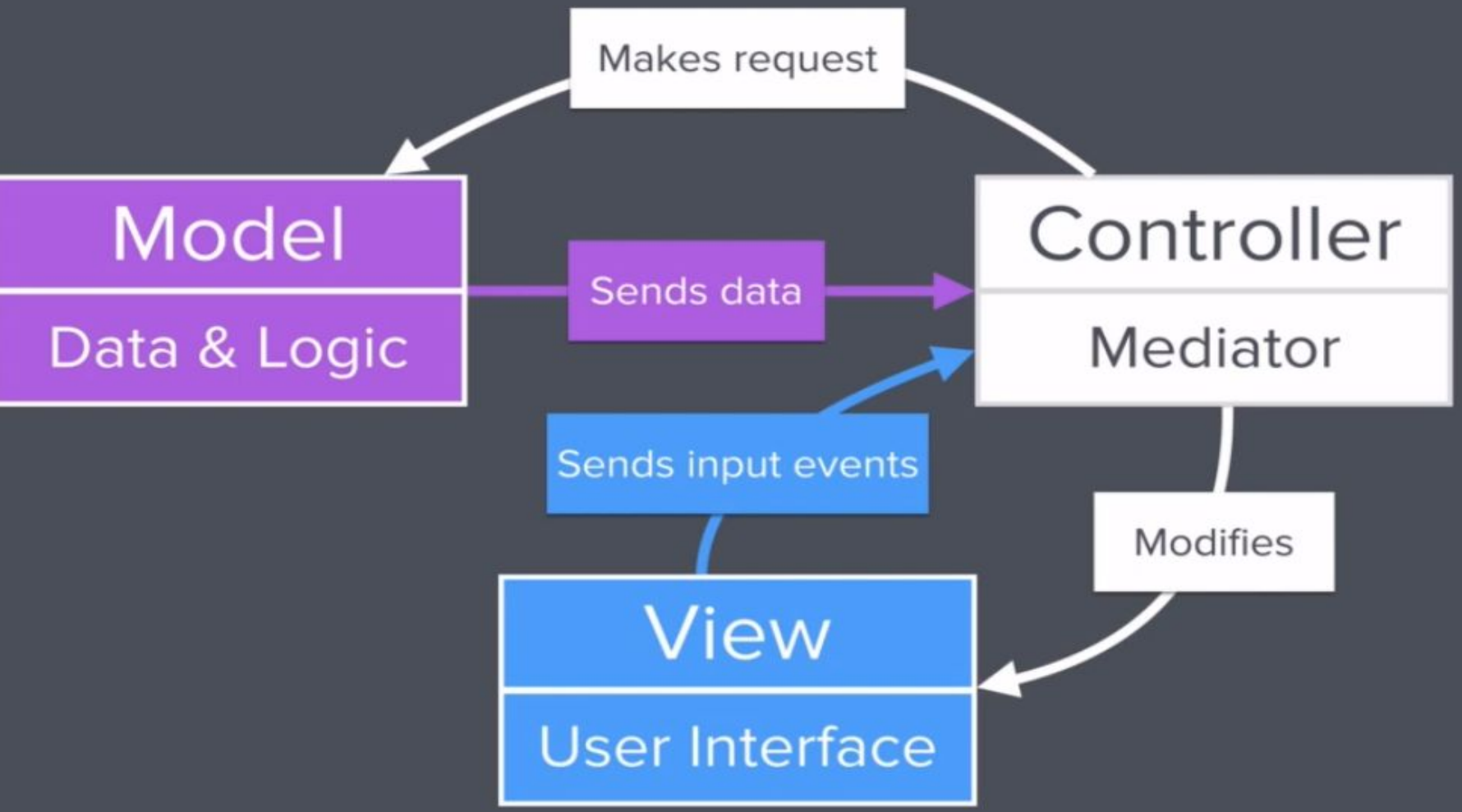
Separation of Concerns

Model

View



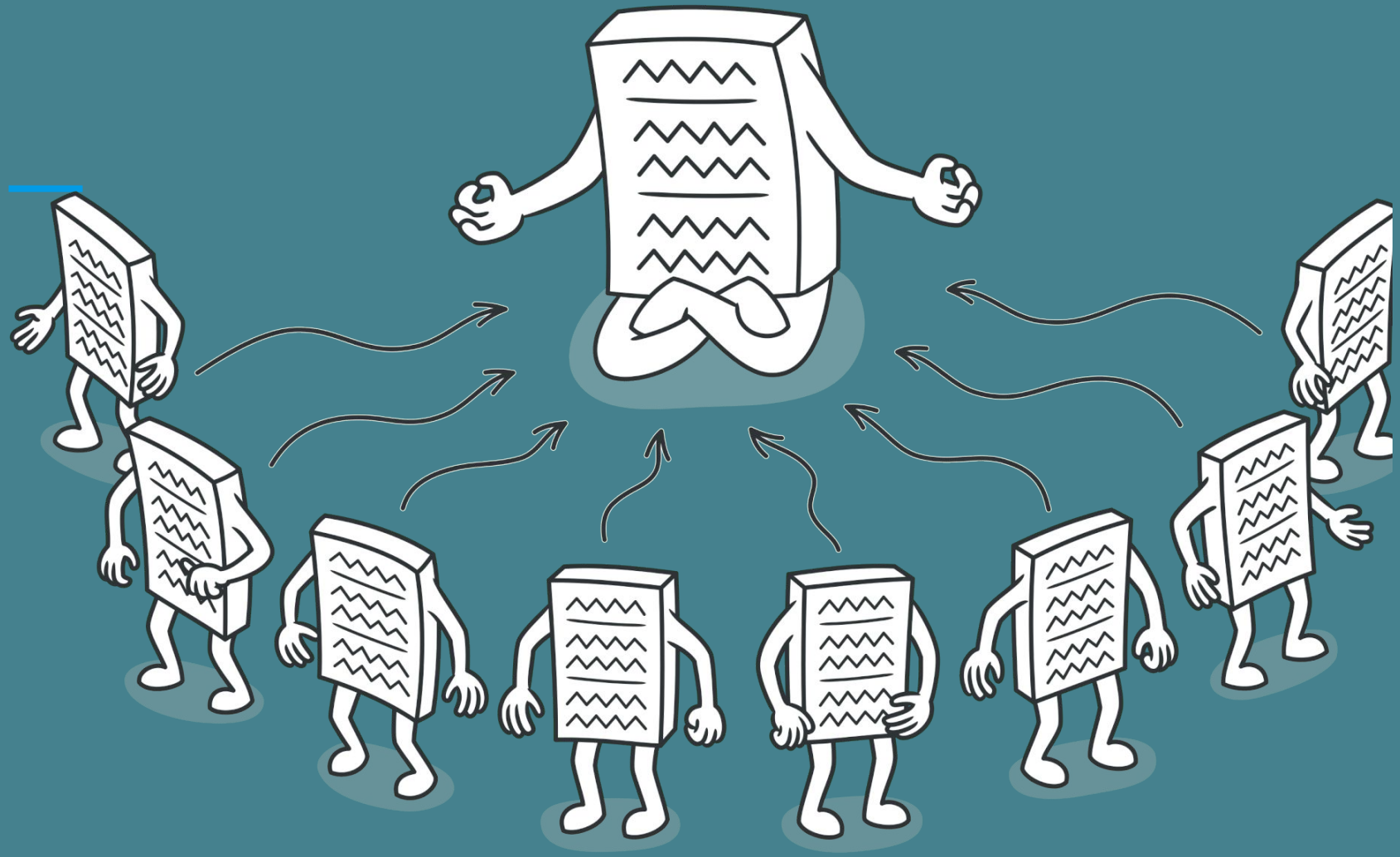
Controller


Makes request
Model
Data & Logic
Sends data
Controller
Mediator
Sends input events
Modifies
View
User Interface

Model-View-Controller Pattern Components:

- The **Model** contains the underlying data, state, and logic that the user wants to see and manipulate.
- The **View** presents the model information to the user in the way they expect it and allows them to interact with it.
- The **Controller** interprets the user's interaction with elements in the view and modifies the model itself.

Singleton pattern

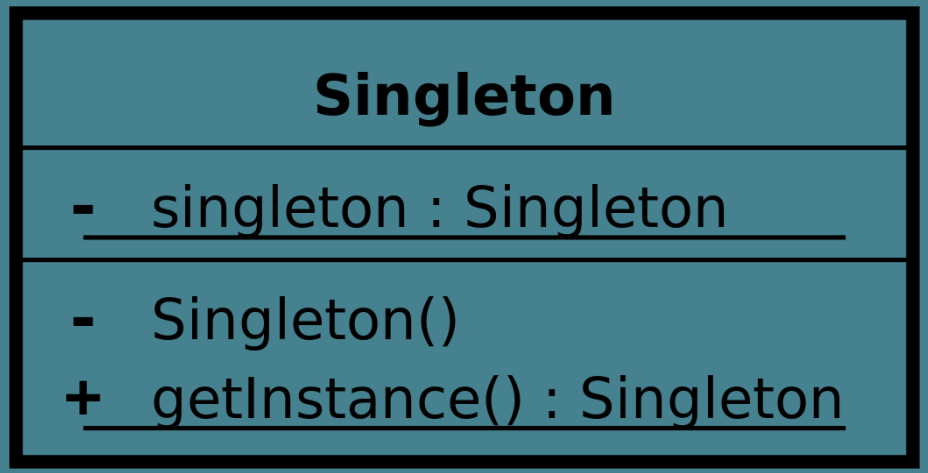


Definition

- In software engineering, the singleton pattern is a software design pattern that restricts the instantiation of a class to one "single" instance. This is useful when exactly one object is needed to coordinate actions across the system. The term comes from the mathematical concept of a singleton.

https://en.wikipedia.org/wiki/Singleton_pattern

UML Structure



Where we use it? And Why we use it?

- We use it this pattern to save database connection and not to lose it.
- If we don’t use his pattern we need to start connection every time.

Other potential usage in feature:

Store user attributes like its status For example last online time, settings.

Java Code

```
public final class Singleton {
    // other attributes if you want to.
    private static final Singleton INSTANCE = new Singleton();
    private Singleton() {}
    public static Singleton getInstance() {
        return INSTANCE;
    }
}
```

Dart code

- In Dart

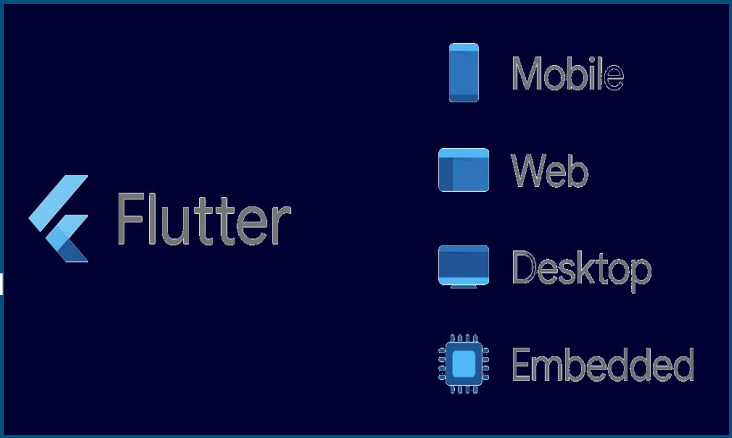
```
class Singleton {
  static final Singleton _singleton = Singleton._internal();

  factory Singleton() {
    return _singleton;
  }

  Singleton._internal();
}
```

Technology we used:

- Dart programming language
  - New and elegant
  - Created and widely by **Google**
  - Nice language for writing modern UI
  - Easy to learn

- Flutter Framework
- Android Studio IDE, VSCode
- Linux os, Window

What is Flutter?



- Flutter is cross platform SDK
- Flutter is written in Dart programming langu
- Open source created by Google
- Stable (Now)
- Android, IOS, Chrome OS
  - Desktop window, Linux, mac os
  - Web
  - Embedded
- In short, Flutter is a truly complete SDK for creating applications
  “Write once, and deploy everywhere”

Ref; Flutter in action, flutter documentation

What is “Evimiz app” (idea)

- We live together
  - Share their exponese
    - homemates
    - travel mates
    - family numbers.

  - Easy to track with time
  - Can be developed more
  - Real world apps
  - Can be in production soon

File Edit Selection View Go Run Terminal Help • main.dart - lib - Visual Studio Code

EXPLORER

OPEN EDITORS 3 UNSAVED
- main.dart U
- model.dart... M
- adding_ne... M

LIB
- controller
- models
  - model.dart M
- views
  - adding_new_it... M
  - final_result.dart M
  - missing_things.... A
  - MyHomePage.... U
  - pass_data.dart A
- home_singleton.... M
- main.dart U

OUTLINE
TIMELINE
DEPENDENCIES

main.dart U | model.dart M | adding_new_items.dart M

main.dart > _FirstScreenState > build

```
36
37  @override
38  Widget build(BuildContext context) {
39    return Scaffold(
40      appBar: AppBar(title: Text('First screen')),
41      body: Center(
42        child: Column(
43          mainAxisAlignment: MainAxisAlignment.center,
44          children: [
45            Padding(
46              padding: const EdgeInsets.all(32.0),
47              child: Text(
48                item,
49                style: TextStyle(fontSize: 24),
50              ), // Text
51            ), // Padding
52            RaisedButton(
53              child: Text(
54                'Go to second screen',
55                style: TextStyle(fontSize: 24),
56              ), // Text
57              onPressed: () {
58                // _awaitReturnValueFromAddNewItem(context);
59              },
60            ) // RaisedButton
61          ],
62        ), // Column
63      ), // Center
64    ); // Scaffold
65  }
```

master*+ | 0 0 21 | Live Share | Ln 50, Col 17 | Spaces: 2 | UTF-8 | CRLF | Dart | Dart from Flutter: 2.0.3

Search | ENG INTL | 08:12 28/05/2021

File Edit Selection View Go Run Terminal Help • adding_new_items.dart - lib - Visual Studio Code

EXPLORER

OPEN EDITORS 3 UNSAVED
- main.dart U
- model.dart... M
- adding_ne... M

LIB
- controller
  - controller.dart M
- models
  - model.dart M
- views
  - adding_new_it... M
  - final_result.dart M
  - missing_things.... A
  - MyHomePage.... U
  - pass_data.dart A
- home_singleton.... M
- main.dart U

OUTLINE
TIMELINE
DEPENDENCIES

main.dart U | model.dart M | adding_new_items.dart M

views > adding_new_items.dart > _AddNewItemState > build

```
18  @override
19  Widget build(BuildContext context) {
20    return Scaffold(
21      appBar: AppBar(title: Text('Adding things'),centerTitle:true ),
22      body: ListView(
23        children: [
24          Center(
25            child: RaisedButton.icon(
26              icon: Icon(Icons.add),
27              label: Text("Add"),
28              color: Colors.blueAccent,
29              onPressed: () {
30                _sendDataBack(context);
31              },
32            ), // RaisedButton.icon
33          ), // Center
34          Container(
35            width: 50.0,
36
37            child: Padding(
38
39              padding: const EdgeInsets.all(32.0),
40              child: TextField(
41                decoration: new InputDecoration(labelText: "Enter item name",hintStyle: TextStyle(fontSize: 15)),
42
43                controller: textFieldController,
44                style: TextStyle(
45                  fontSize: 15,
46                  color: Colors.white,
47                ), // TextStyle
```

master*+ ⊗ 0 ⚠ 0 ⓘ 21 Live Share | Ln 31, Col 1 | Spaces: 2 | UTF-8 | CRLF | Dart | Dart from Flutter: 2.0.3

Search | ENG INTL | 08:14 28/05/2021

File Edit Selection View Go Run Terminal Help • adding_new_items.dart - lib - Visual Studio Code

EXPLORER

OPEN EDITORS 3 UNSAVED
- main.dart U
- model.dart... M
- adding_ne... M

LIB
- controller
  - controller.dart M
- models
  - model.dart M
- views
  - adding_new_it... M
  - final_result.dart M
  - missing_things.... A
  - MyHomePage.... U
  - pass_data.dart A
- home_singleton.... M
- main.dart U

main.dart U | model.dart M | adding_new_items.dart M

views > adding_new_items.dart > _AddNewItemState > build

```
                ), // Container
                onChanged: (newValue) {
                  setState(() {
                    dropdownValue = newValue;
                  });
                },
                items:users
                    .map<DropdownMenuItem<String>>((String payer) {
                  return DropdownMenuItem<String>(
                    value: payer,
                    child: Text(payer),
                  ); // DropdownMenuItem
                }).toList(),
              )], // DropdownButton
            ), // Row
          ), // Container
          SizedBox(height: 100),

        ],
      ), // ListView
    ); // Scaffold
  }

  // get the text in the TextField and send it back to the FirstScreen
  void _sendDataBack(BuildContext context) {
    String textToSendBack =
        textFieldController.text + "#" + textFieldController2.text +"#" +dropdownValue;
    Navigator.pop(context, textToSendBack);
  }
```

OUTLINE
TIMELINE
DEPENDENCIES

master*+ ⊗ 0 ⚠ 0 ⓘ 21 Live Share | Ln 31, Col 1 | Spaces: 2 | UTF-8 | CRLF | Dart | Dart from Flutter: 2.0.3

Search | ENG INTL | 08:14 28/05/2021

File Edit Selection View Go Run Terminal Help

MyHomePage.dart - lib - Visual Studio Code

EXPLORER

OPEN EDITORS 4 UNSAVED

- main.dart U
- model.dart... M
- adding_ne... M
- final_result.... M
- MyHomePa... U

LIB

- controller
  - controller.dart M
- models
  - model.dart M
- views
  - adding_new_it... M
  - final_result.dart M
  - missing_things.... A
  - MyHomePage.... U
  - pass_data.dart A
- home_singleton.... M
- main.dart U

OUTLINE

TIMELINE

DEPENDENCIES

main.dart U | model.dart M | adding_new_items.dart M | final_result.dart M | MyHomePage.dart U

views > MyHomePage.dart > ...

```
26  @override
27  void initState() {
28    super.initState();
29  }
30
31  @override
32  Widget build(BuildContext context) {
33    return Scaffold(
34        appBar: AppBar(
35          title: Text(" May"),
36          centerTitle: true,
37        ), // AppBar
38        body: Column(
39          children: <Widget>[
40            Expanded(
41              child: Container(
42                color: Colors.black12,
43                child: Column(
44                  children: [
45                    Theme(
46                      data: Theme.of(context)
47                          .copyWith(dividerColor: Colors.white38),
48                      child: DataTable(showBottomBorder: true, columns: [
49                        DataColumn(label: Text('Member', style: headers)),
50                        DataColumn(
51                          label: Text('paid', style: headers),
52                        ), // DataColumn
53                        DataColumn(label: Text('Status', style: headers))
54                      ], rows:[
```

master*+ | 0 0 21 | Live Share | Ln 1, Col 1 | Spaces: 2 | UTF-8 | CRLF | Dart | Dart from Flutter: 2.0.3

Search

ENG INTL 08:15 28/05/2021

File Edit Selection View Go Run Terminal Help

MyHomePage.dart - lib - Visual Studio Code

EXPLORER

OPEN EDITORS 4 UNSAVED

- main.dart U
- model.dart... M
- adding_ne... M
- final_result.... M
- MyHomePa... U

LIB

- controller
  - controller.dart M
- models
  - model.dart M
- views
  - adding_new_it... M
  - final_result.dart M
  - missing_things.... A
  - MyHomePage.... U
  - pass_data.dart A
- home_singleton.... M
- main.dart U

OUTLINE

TIMELINE

DEPENDENCIES

main.dart U | model.dart M | adding_new_items.dart M | final_result.dart M | MyHomePage.dart U

views > MyHomePage.dart > ...

```
53                DataColumn(label: Text('Status', style: headers))
54              ], rows:[
55                DataRow(cells: [
56                  DataCell(Text(users[0], style: memberStyle)),
57                  DataCell(Text(data[users[0]].toString(),
58                      style: memberStyle)), // Text // DataCell
59                  DataCell(Text(status1, style: memberStyle)),
60                ]), // DataRow
61                DataRow(cells: [
62                  DataCell(
63                    Text(users[1], style: memberStyle),
64                  ), // DataCell
65                  DataCell(Text(data[users[1]].toString(),
66                      style: memberStyle)), // Text // DataCell
67                  DataCell(Text(status2, style: memberStyle)),
68                ]), // DataRow
69                DataRow(
70                  cells: [
71                    DataCell(Text(users[2], style: memberStyle)),
72                    DataCell(Text(data[users[2]].toString(),
73                        style: memberStyle)), // Text // DataCell
74                    DataCell(Text(status3, style: memberStyle)),
75                  ],
76                ), // DataRow
77                DataRow(
78                  cells: [
79                    DataCell(Text('Total/each ', style: headers)),
80                    DataCell(Text(total.toString(), style: headers)),
81                    DataCell(Text((total/3).toStringAsFixed(2)+ '', style: headers)),
82                  ]
```

master*+ | 0 0 21 | Live Share | Ln 1, Col 1 | Spaces: 2 | UTF-8 | CRLF | Dart | Dart from Flutter: 2.0.3

Search

ENG INTL 08:16 28/05/2021

File Edit Selection View Go Run Terminal Help • MyHomePage.dart - lib - Visual Studio Code

main.dart U | model.dart M | adding_new_items.dart M | final_result.dart M | MyHomePage.dart U

views > MyHomePage.dart > _MyHomePageState > _awaitReturnValueFromAddNewItem

```dart
  void _awaitReturnValueFromAddNewItem(BuildContext context) async {
    final result = await Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => AddNewItem(),
        )); // MaterialPageRoute
    setState(() {
      List input = result.split("#");

      items.insert(0, input[0]);

      prices.insert(0, input[1]);
      buyer.insert(0, input[2]);

      if (input[2] == users[2]) {
        data[users[2]] = data[users[2]] + double.parse(input[1]);
      } else if (input[2] == users[1]) {
        data[users[1]] = data[users[1]] + double.parse(input[1]);
        // print(data[input[1] + input[2]]);
      } else if (input[2] == users[0]) {
        data[users[0]] = data[users[0]] + double.parse(input[1]);
        // print(data[input[1] + input[2]]);
      } else {
        print("Erorr");
      }
      total = data[users[2]] + data[users[1]] + data[users[0]];
```

EXPLORER

OPEN EDITORS 5 UNSAVED
- main.dart U
- model.dart... M
- adding_ne... M
- final_result.... M
- MyHomePa... U

LIB
- controller
  - controller.dart M
- models
  - model.dart M
- views
  - adding_new_it... M
  - final_result.dart M
  - missing_things.... A
  - MyHomePage.... U
  - pass_data.dart A
- home_singleton.... M
- main.dart U

OUTLINE
TIMELINE
DEPENDENCIES

master*+ ⊗ 0 ⚠ 0 ⓘ 21 Live Share Ln 207, Col 7 Spaces: 2 UTF-8 CRLF Dart Dart from Flutter: 2.0.3

Search ENG INTL 08:16 28/05/2021

File Edit Selection View Go Run Terminal Help

• model.dart - lib - Visual Studio Code

EXPLORER

OPEN EDITORS 2 UNSAVED

- main.dart U
- model.dart... M

LIB

- controller
- models
  - model.dart M
- views
  - adding_new_it... M
  - final_result.dart M
  - missing_things.... A
  - MyHomePage.... U
  - pass_data.dart A
- home_singleton.... M
- main.dart U

OUTLINE

TIMELINE

DEPENDENCIES

main.dart U | model.dart M

models > model.dart > User

```dart
class Home {
  int homeId;
  String nameOfHome;
  int members;
  double total;
  double forEach;
  String month;
  List history;
  var dateOfCreation = DateTime.now();
  Home(this.homeId, this.nameOfHome, this.members, this.total, this.forEach,
      this.month, this.history, this.dateOfCreation);
}
class User {
  int userId;
  String name;
  double paid;
  double total;
  List history;
  User(this.userId, this.name, this.paid, this.total, this.history);
}
class Items {
  int id;
  String name;
  double cost;
  int date;

  Items({this.id, this.name, this.cost, this.date});
}
```

master*+ ⊗ 0 ⚠ 0 ⓘ 21 Live Share

Ln 20, Col 2 Spaces: 2 UTF-8 CRLF Dart Dart from Flutter: 2.0.3

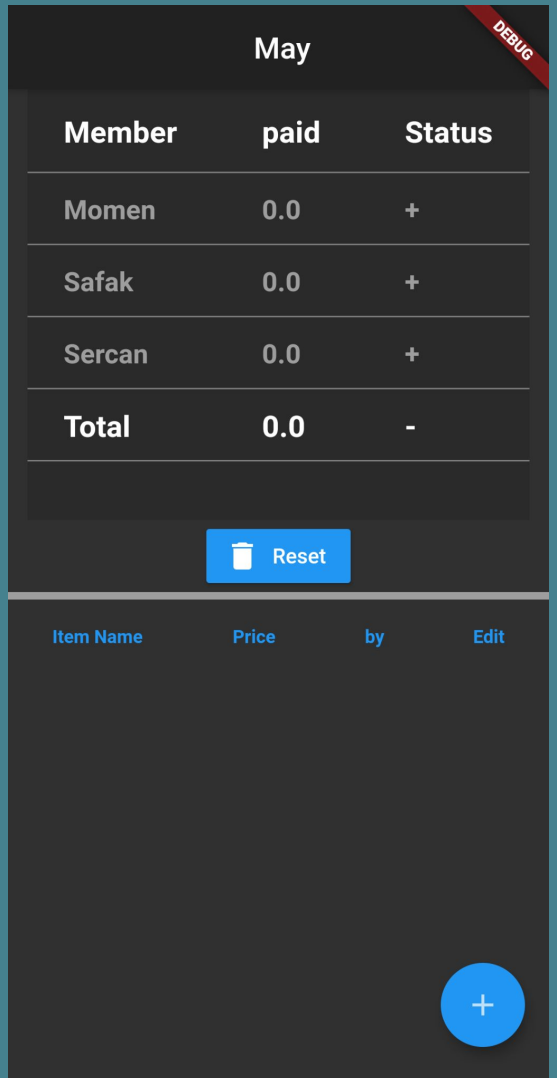
Search

ENG INTL 08:10 28/05/2021

Month account(may)

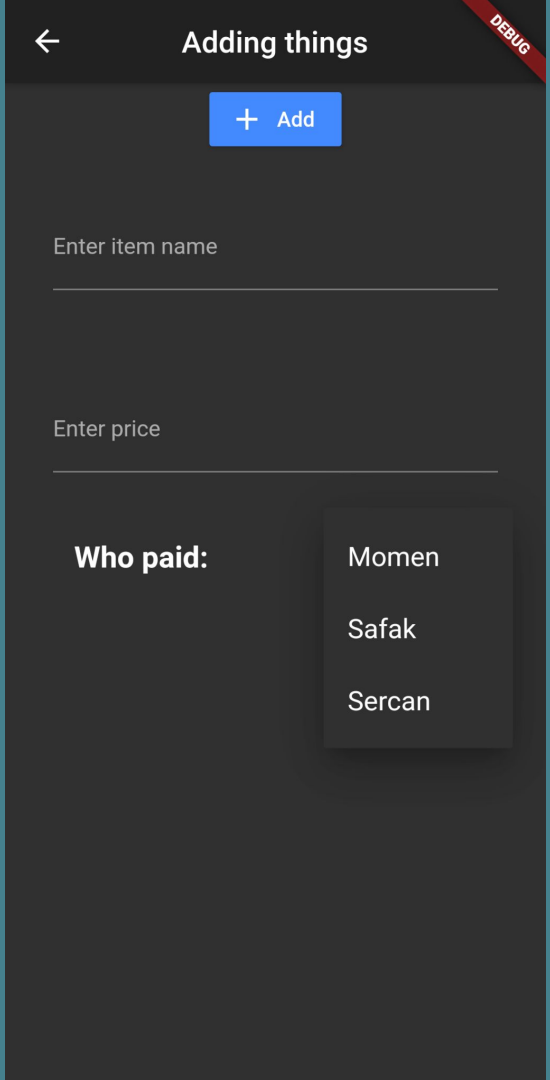
- Home members:
  - Momen
  - Safak
  - Sercan

-



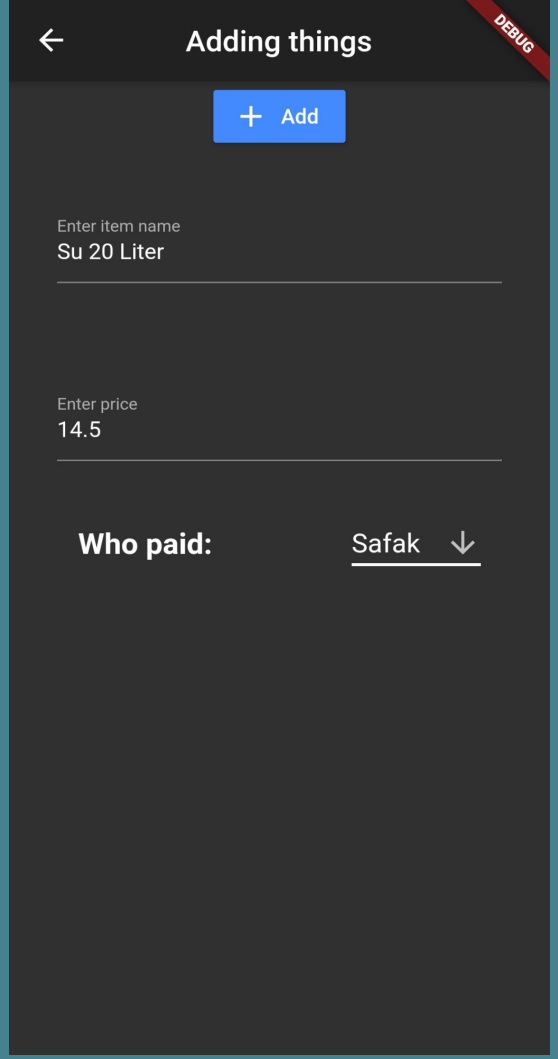
Adding new Items

- Name
- Price
- Who paid



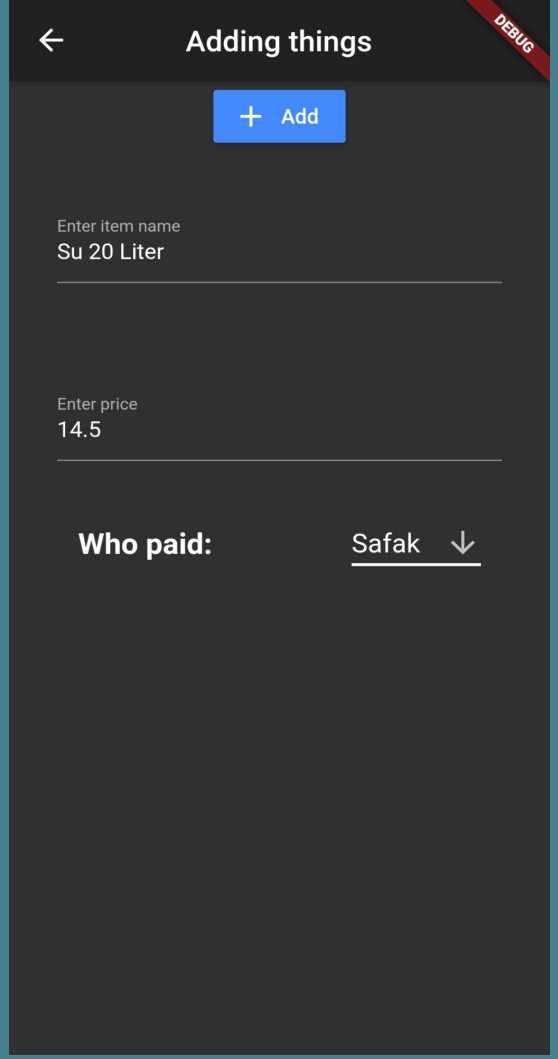
Adding new Items

- Name
- Price
- Who paid



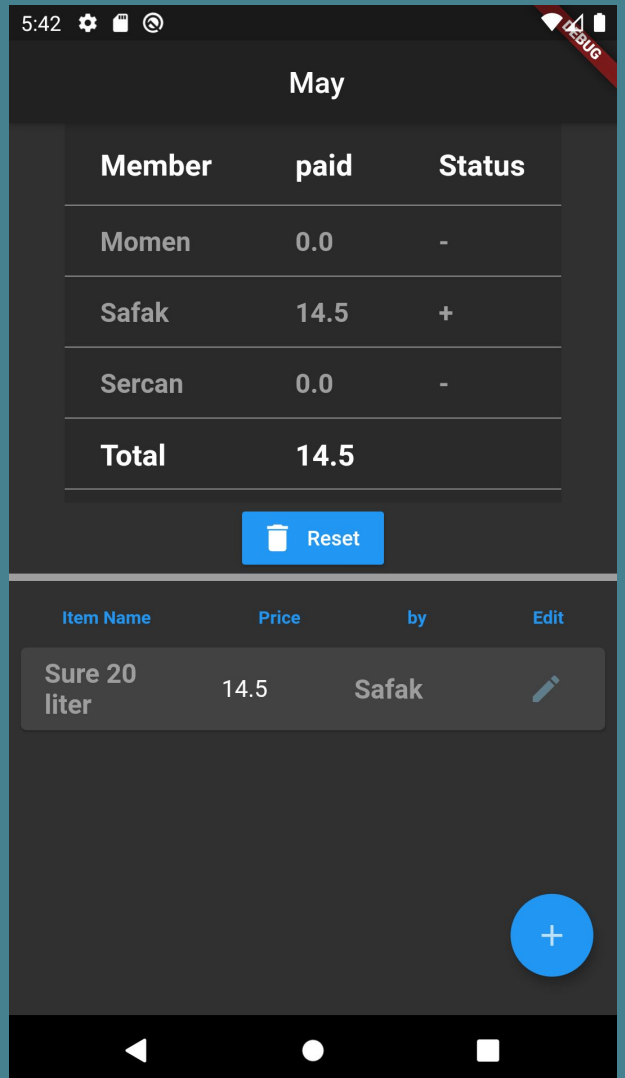
Adding new Items

- Name
- Price
- Who paid



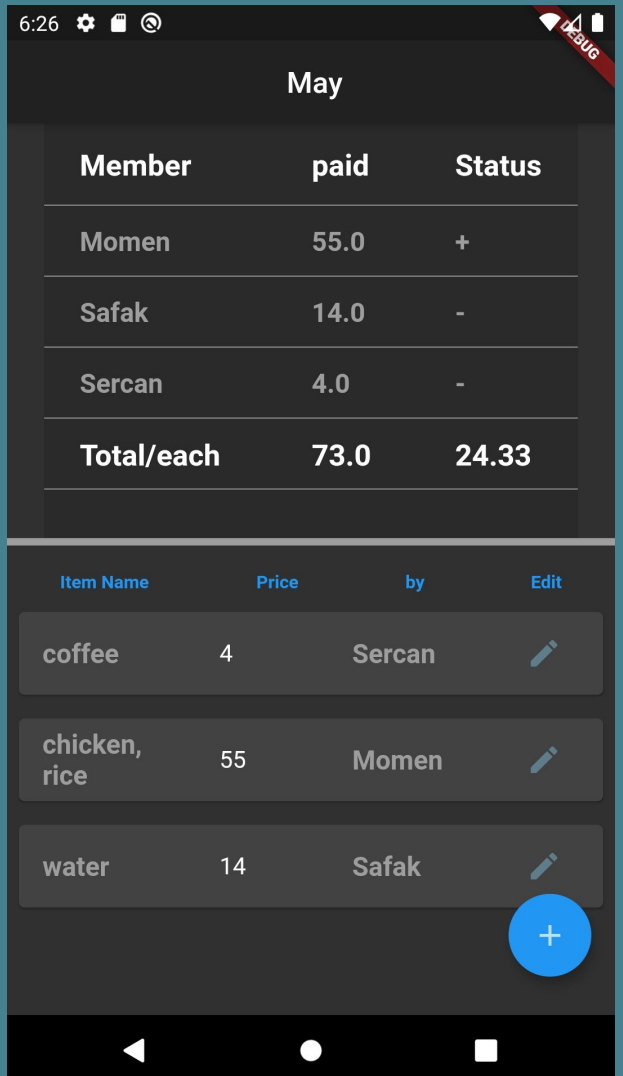
Inserting

- Momen is -
- Safak is +
- Sercan -
- Total = 14.5



Inserting

- Momen is +
- Safak is -
- Sercan -
- Total = 73.0
- Each one: 24.33



Future improvement:

- Support multiple languages (Turkish, Arabic)
- Push notifications
- Using factory patterns for item data
- In production soon !

Conclusion

- Good software design can save a lot of money and time
- Select Software patterns wisely
- Don’t repeat yourself

Thank you !

- Mohammed ALI
- Safak ALPAY

Any questions?