

Basics on UITableViewController

Fabrice.Kordon@lip6.fr



UITableViewController?



Problem

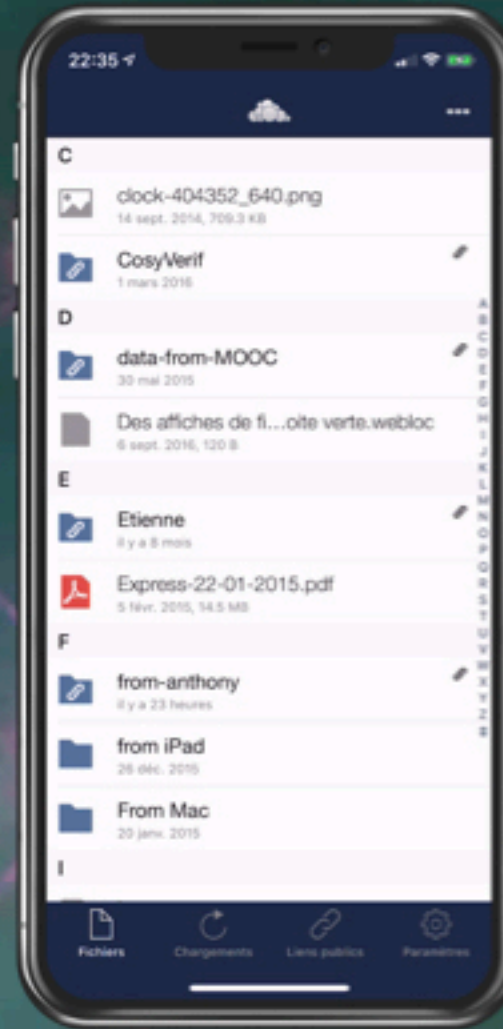
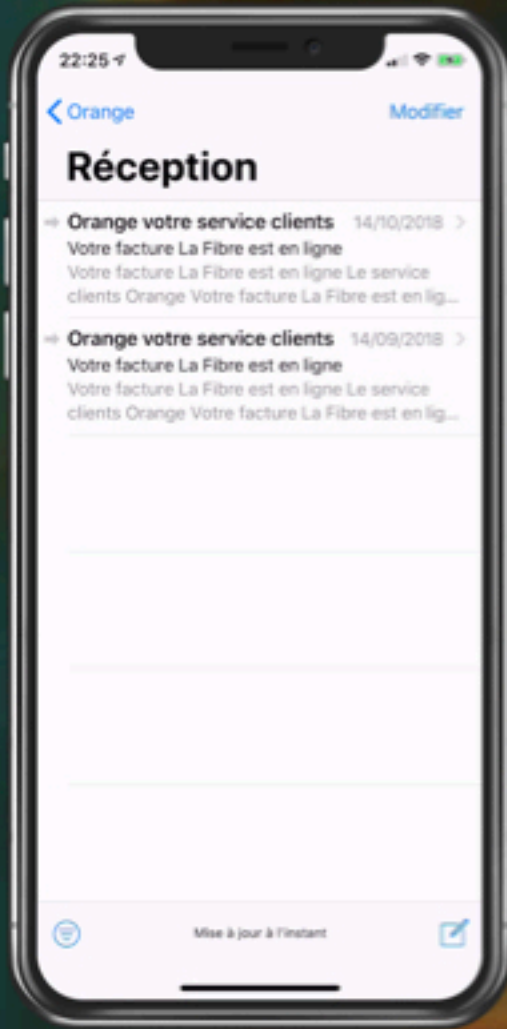
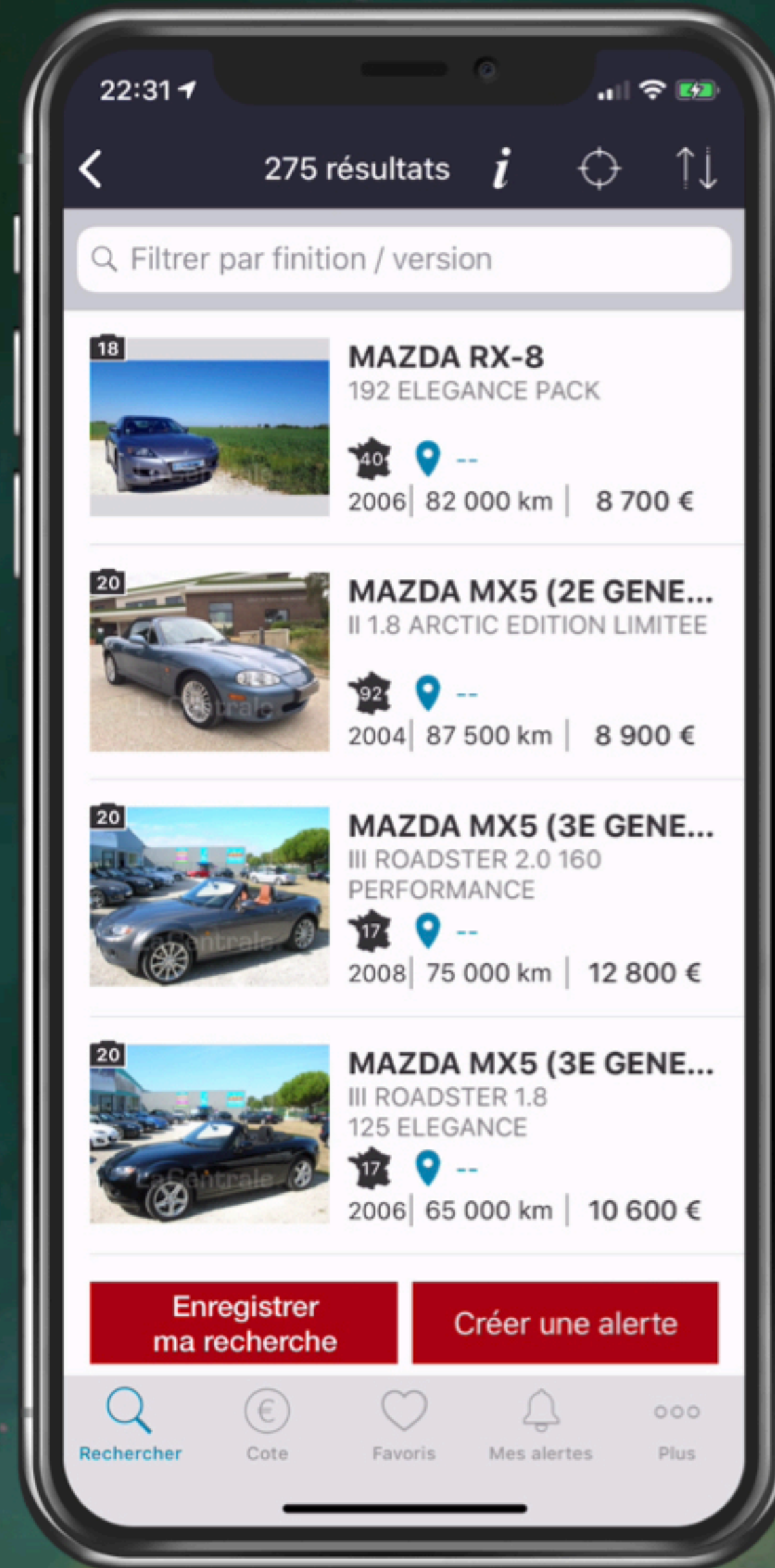
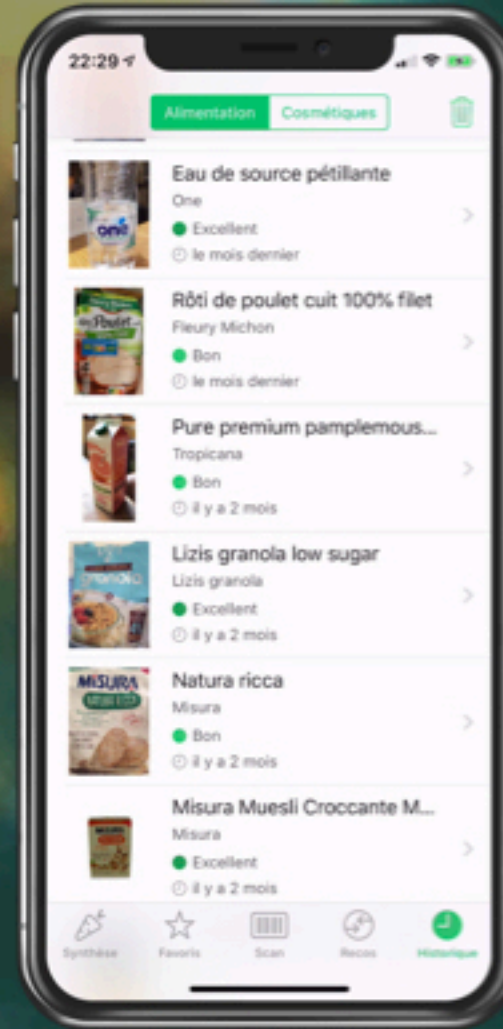
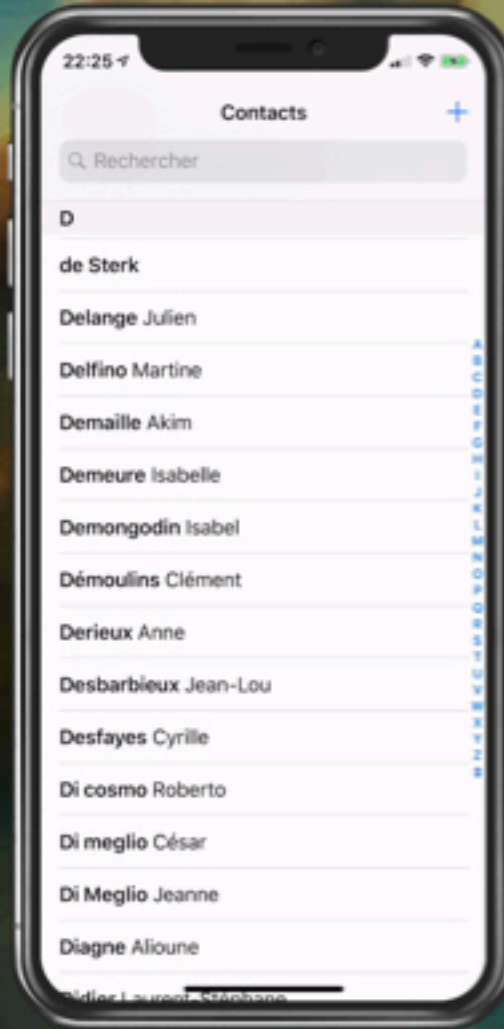
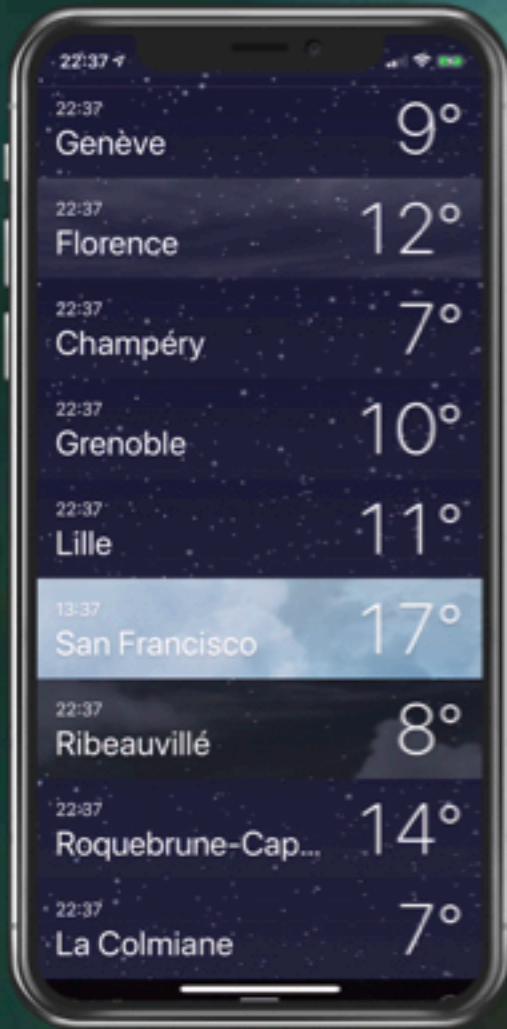
- Handle sets of data
 - ▶ Large
 - ▶ Structured



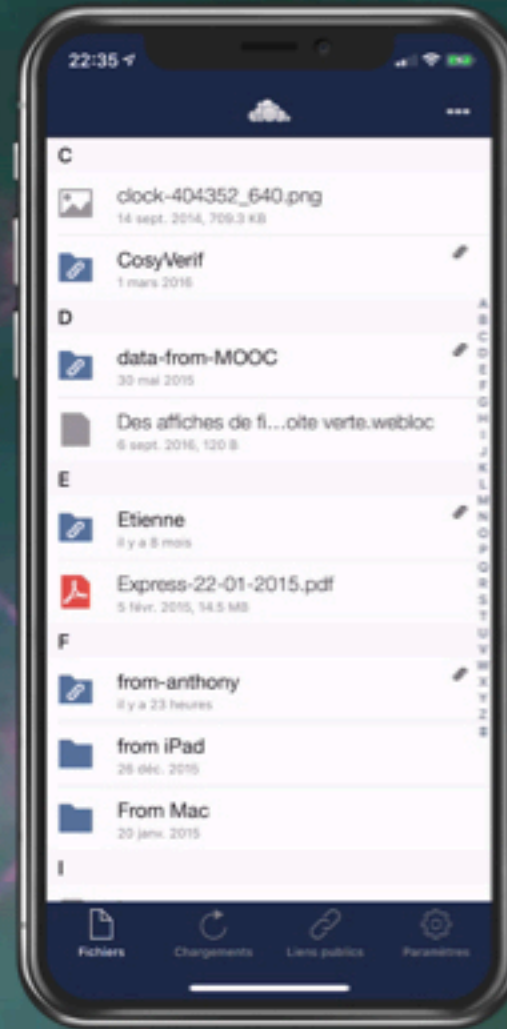
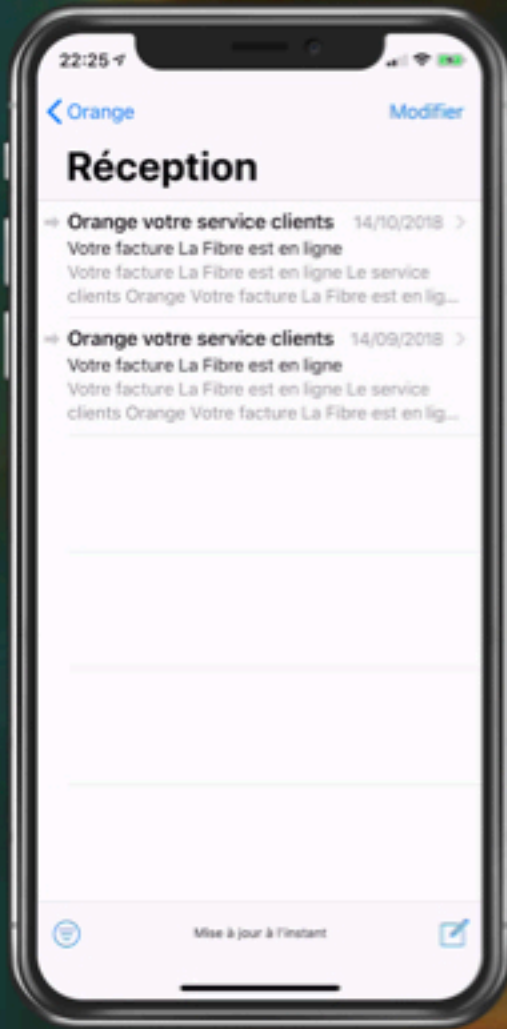
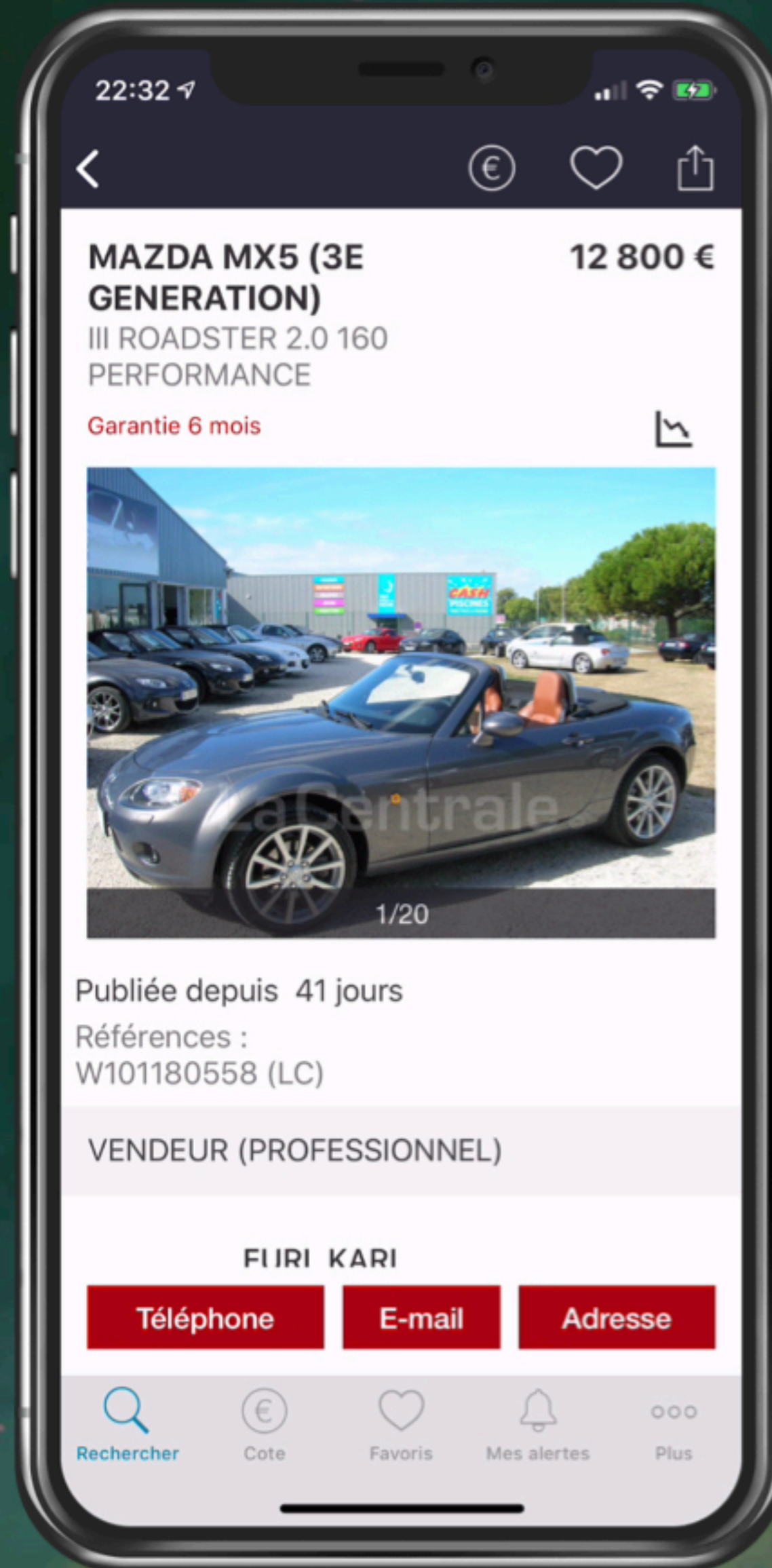
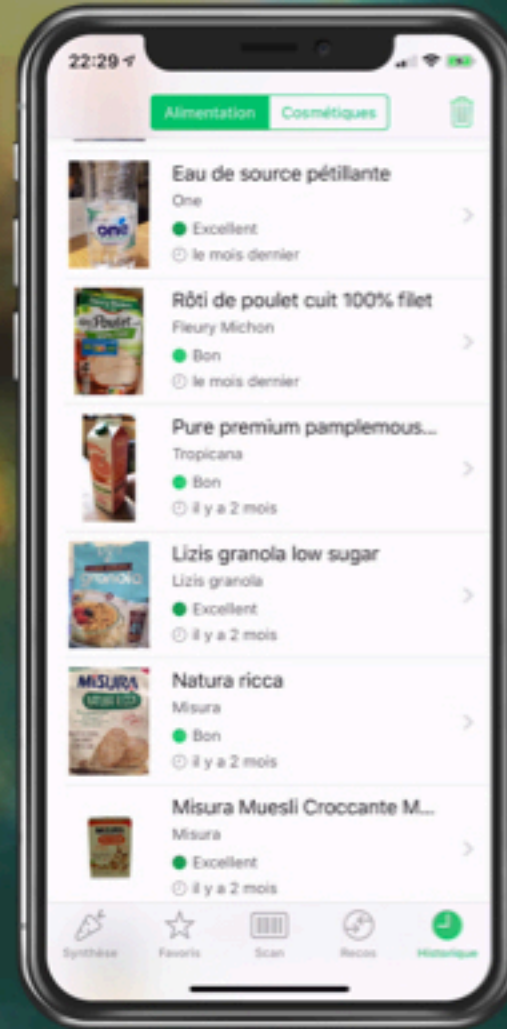
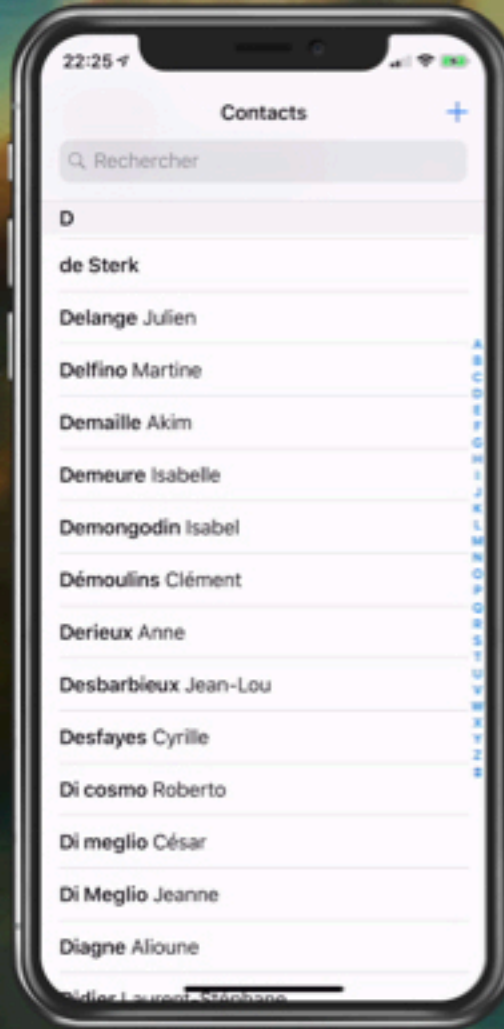
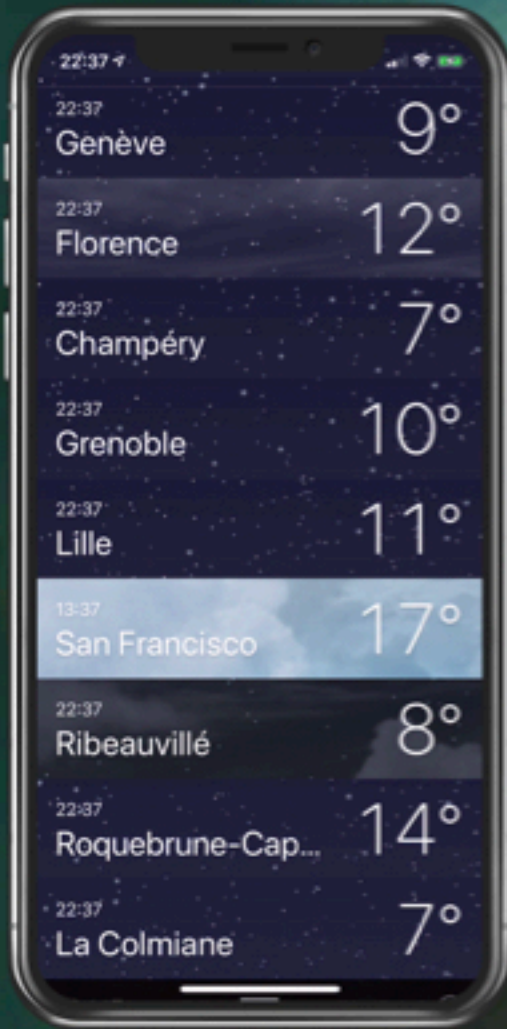
The solution?

- UITableViewController
 - ▶ You know in fact...

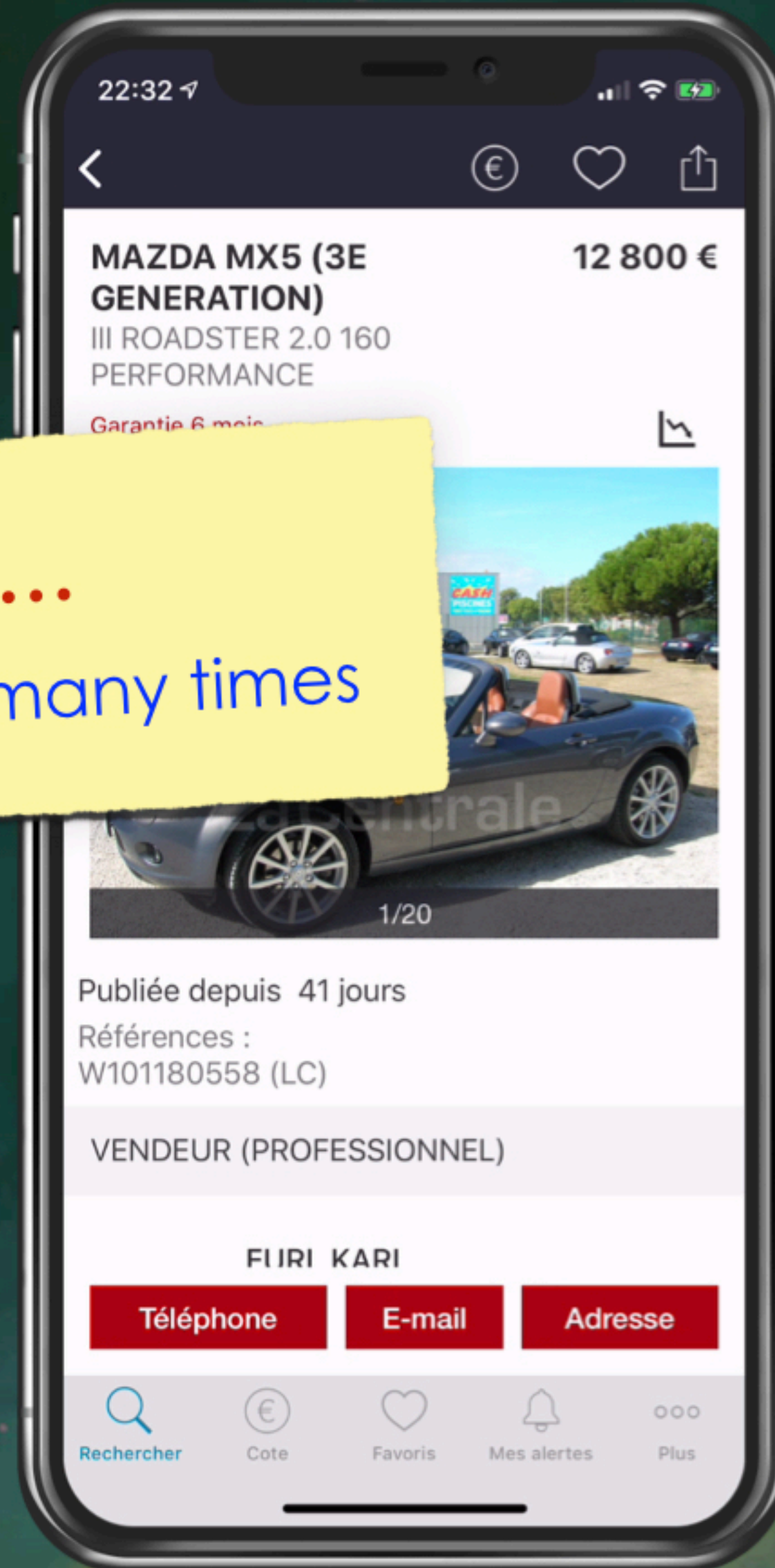
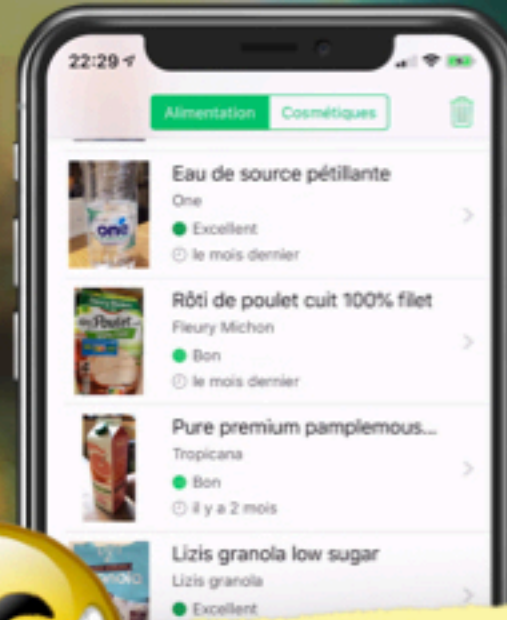
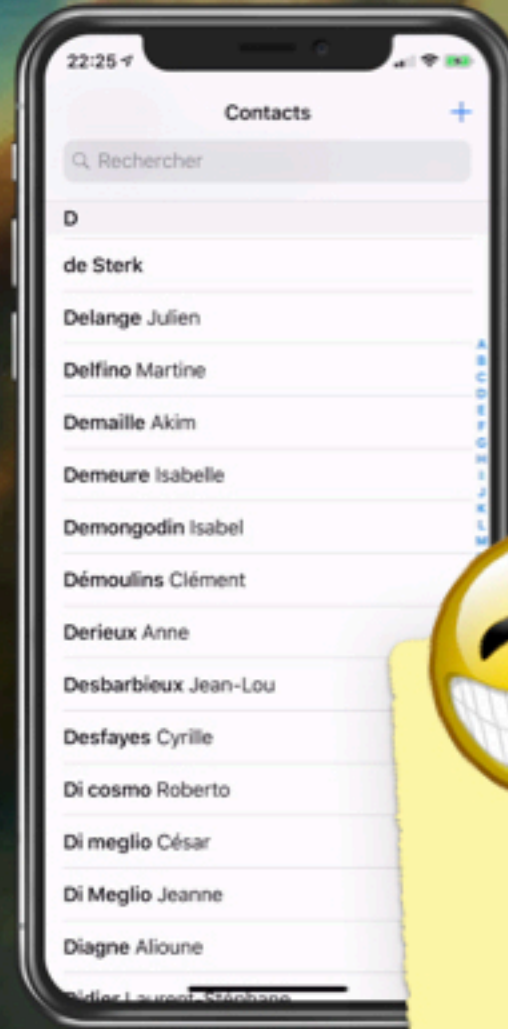
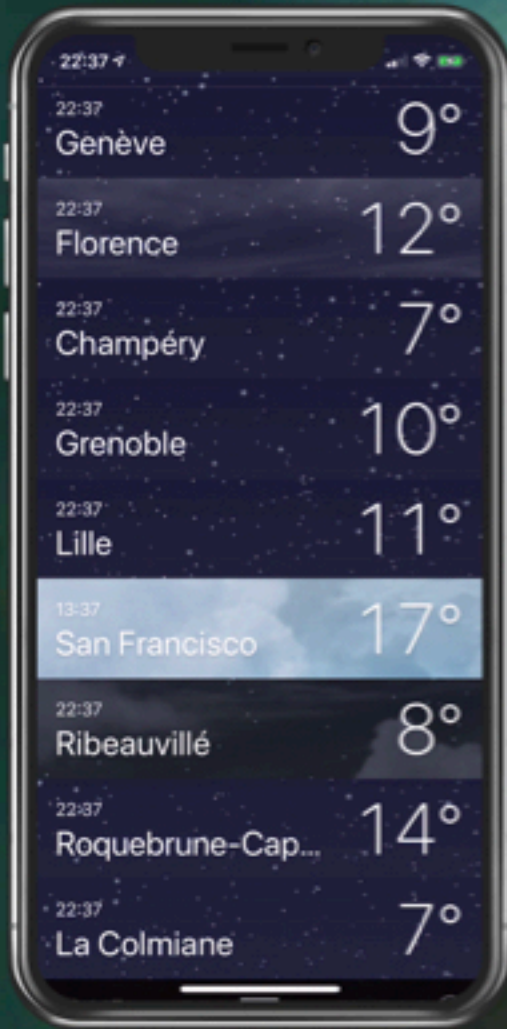
UITableViewController?



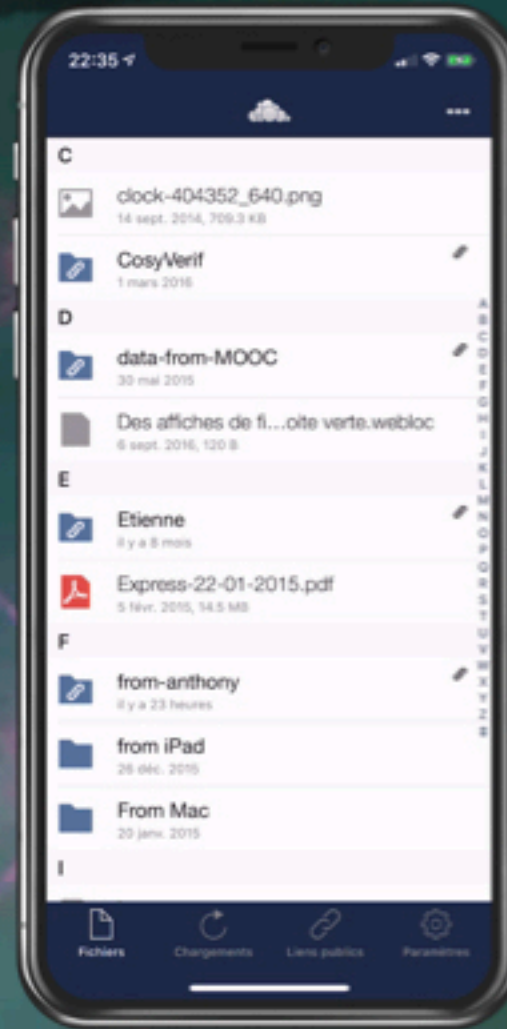
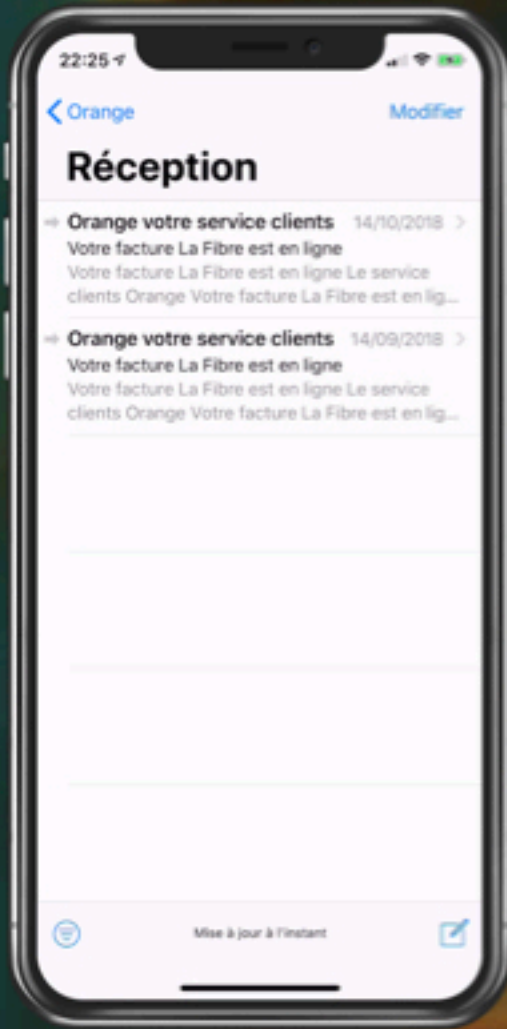
UITableViewController?



UITableViewController?



 **For sure...**
You met this one many times



Managing a Table

Thanks to a UITableViewController

- Embeds a UITableView

Presents data (view + controller)

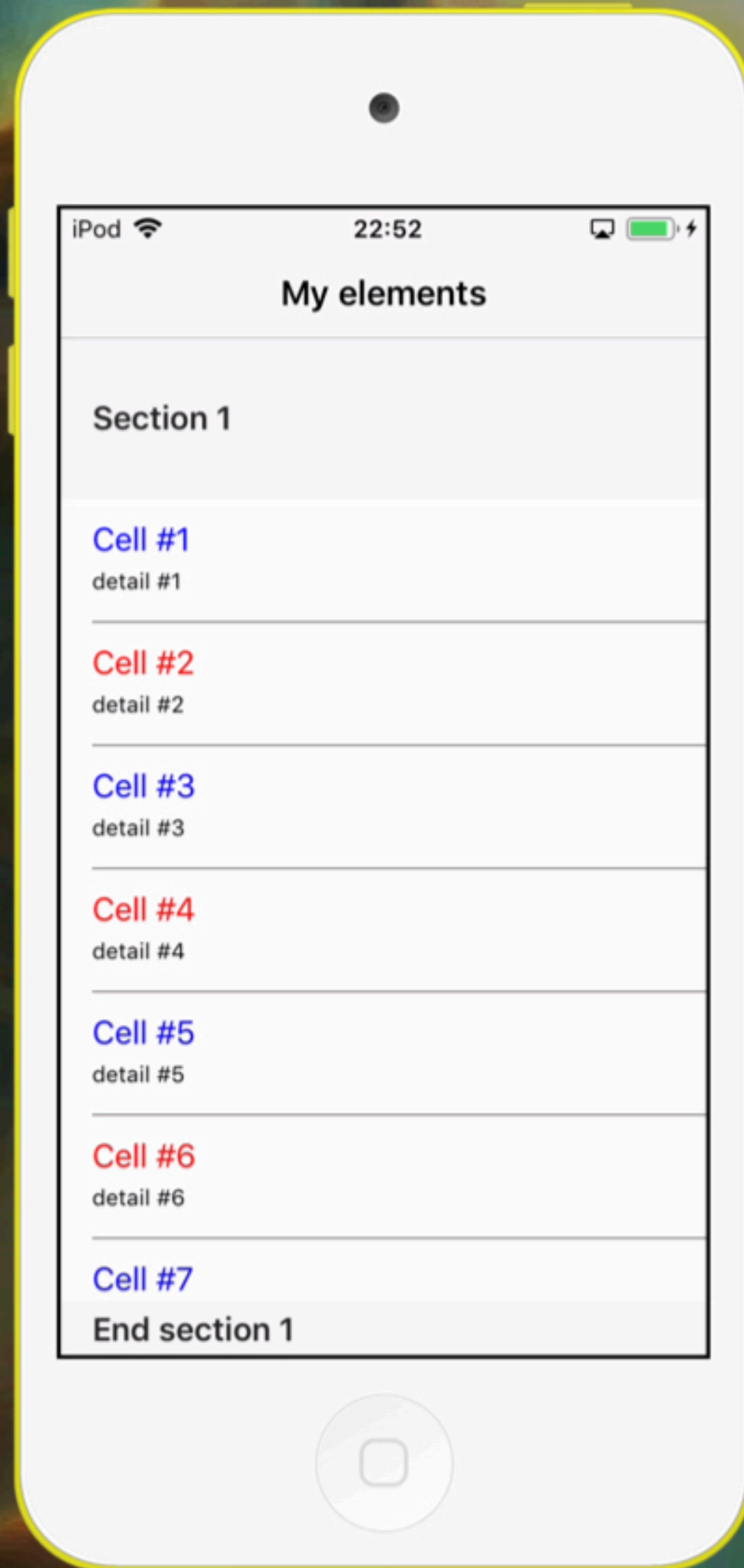
- One column / several lines (cells)
- Dispatch into sections
- Vertical scrolling
- Optimised memory management
 - ▶ Detailed in a further video
 - ▶ Useful for large collections of data

Used in numerous applications

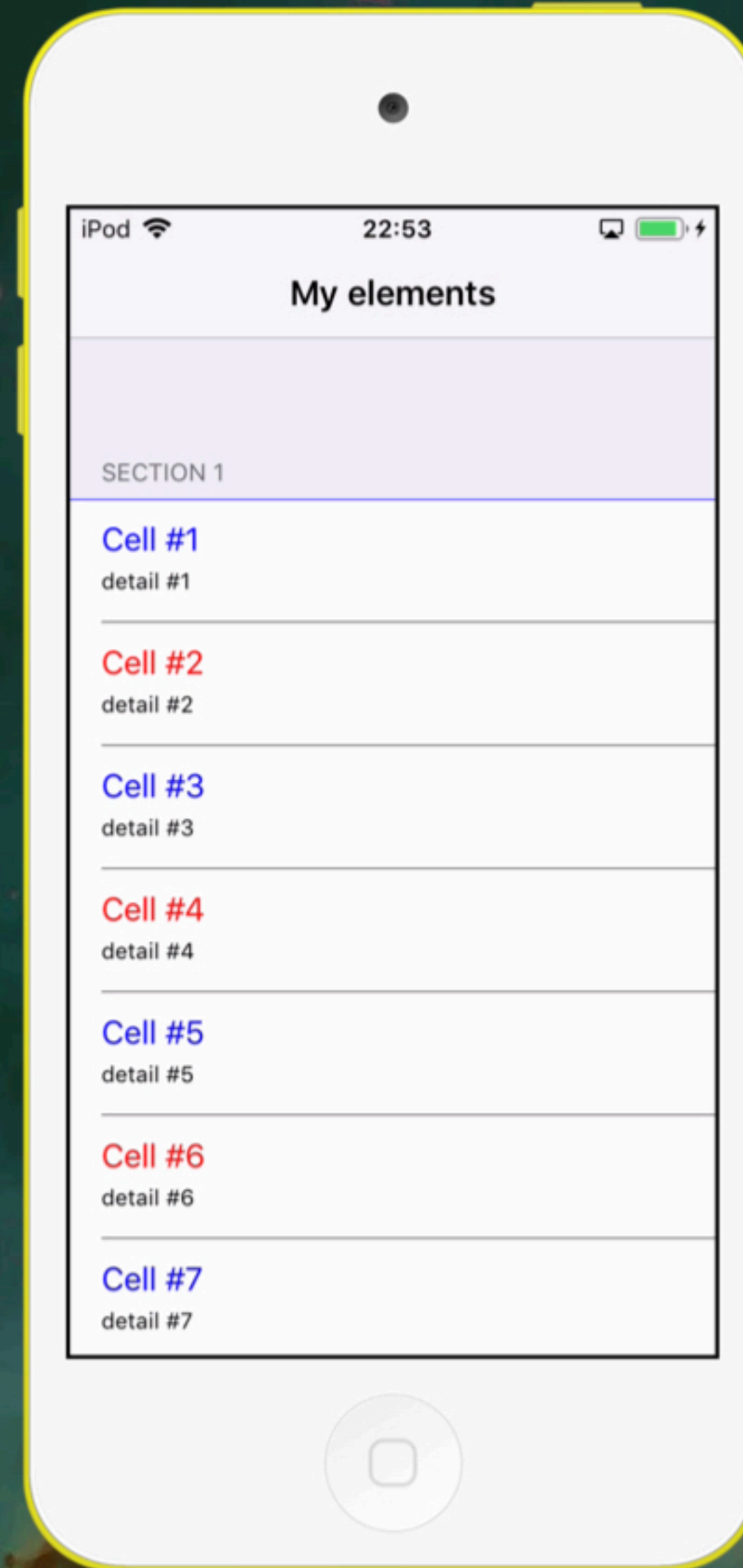
- You should use it too!

Display Styles

plain



grouped



Data management

Intensive use of delegation

Display of data

- Allocation for displayed information only
- Methods implemented by a «dataSource»
 - ▶ UITableViewDataSource
 - ▶ Does this remind you something?

This sources must answer to questions like

- How many sections?
- How many line in section «s»?
- What is the content of line «l» in section «s»?
- What is the header of section «s»?

A bit about UITableViewDataSource

7



Some methods

```
func numberOfSections(in tableView: UITableView) -> Int
func tableView(_ tableView: UITableView,
               numberOfRowsInSection section: Int) -> Int
func tableView(_ tableView: UITableView,
               cellForRowAt indexPath: IndexPath) -> UITableViewCell

func tableView(_ tableView: UITableView,
               titleForHeaderInSection section: Int) -> String?
func tableView(_ tableView: UITableView,
               titleForFooterInSection section: Int) -> String?
```

• And many more...



Important!

- UITableViewController inherits from UIViewController
- Orientation and side-by-side handled too
- ▶ This is a consistent «MVC-group»

As a conclusion...

Display of the UITableView

- Handled in the super structure

UITableView allows to

- Display structured data
- To take advantage of the screen size (small or large)
- To customise cells

And even more...

- It is able to handle efficient display (to be studied later)
 - ▶ Important for fast scrolling
- Edition mechanism embedded
 - ▶ Another protocol : UITableViewDelegate

One more thing...



Recommandation (strong suggestion from



You should encapsulate a UITableViewController in a UINavigationController

- ▶ To be studied in a few videos
- ▶ Take it as a convention so far

