Test-Driven iOS Development with Swift

Dominik Hauser, @swiftpainless

Three rules (Uncle Bob)

- You are not allowed to write any production code unless it is to pass a failing unit test
- You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures
- You are not allowed to write any more production code than is sufficient to pass the one failing unit test

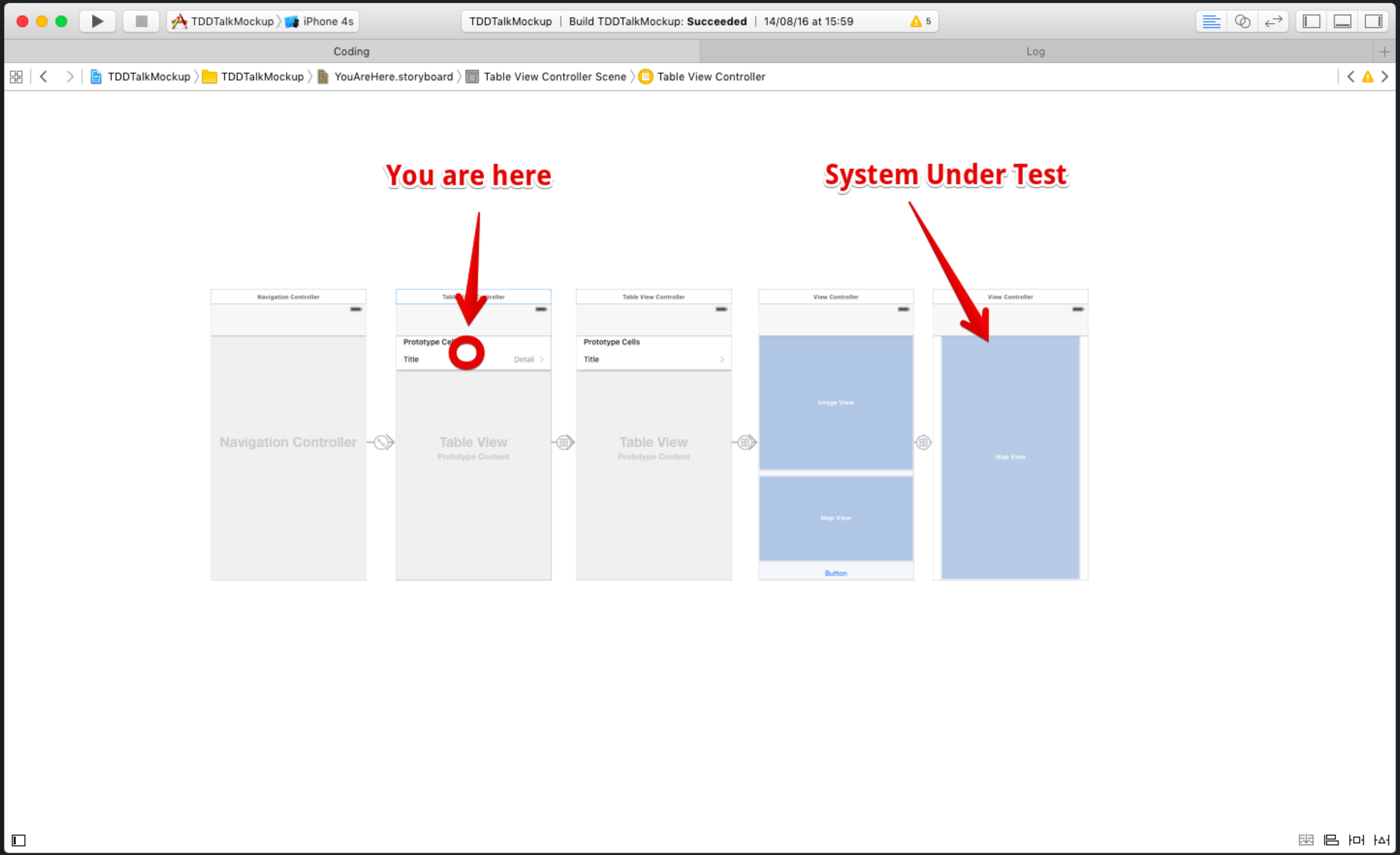
Red, Green, Refactor

Red: Write a failing test

Green: Write the simplest code that makes the test pass

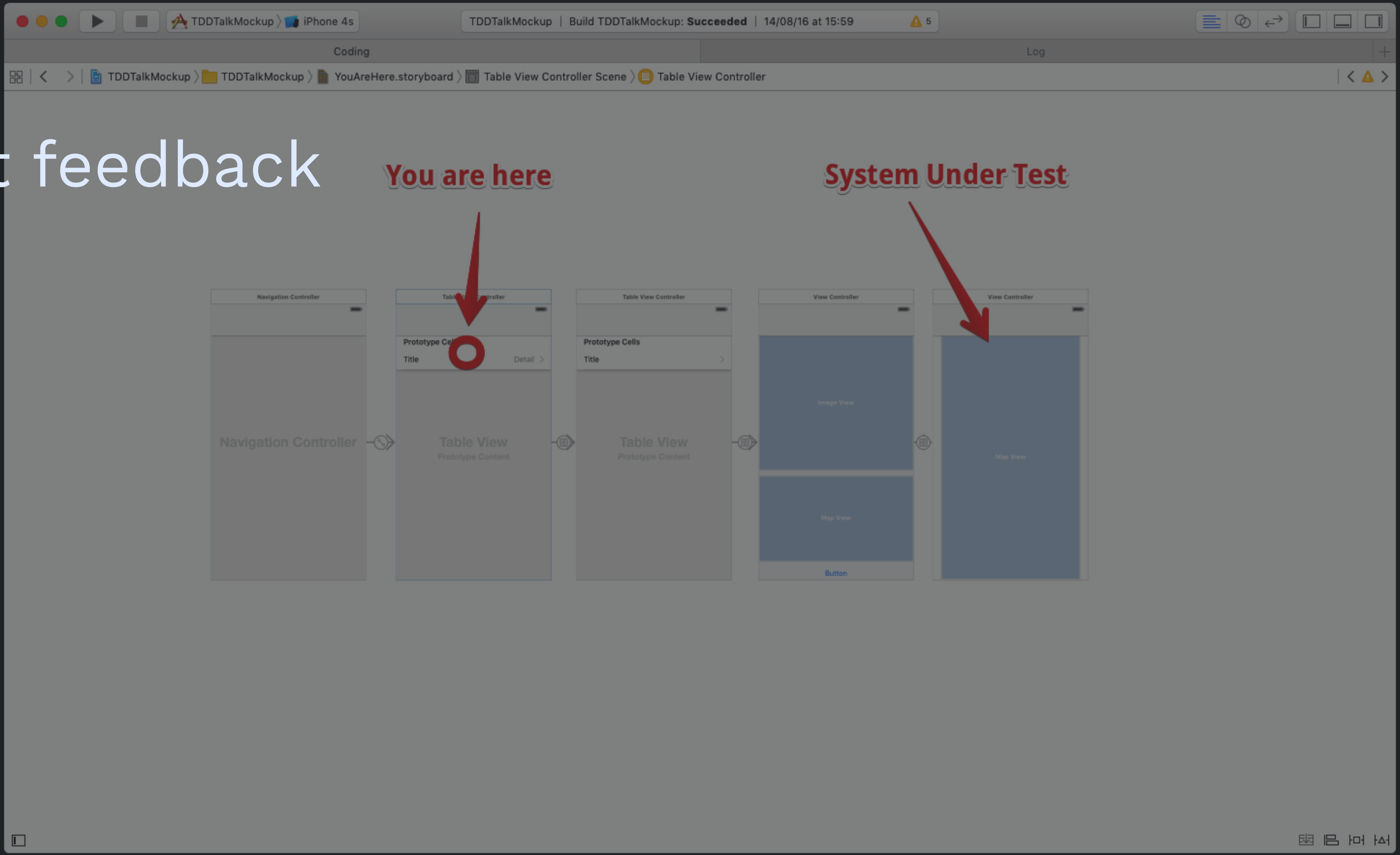
Refactor: Improve the code without breaking the tests

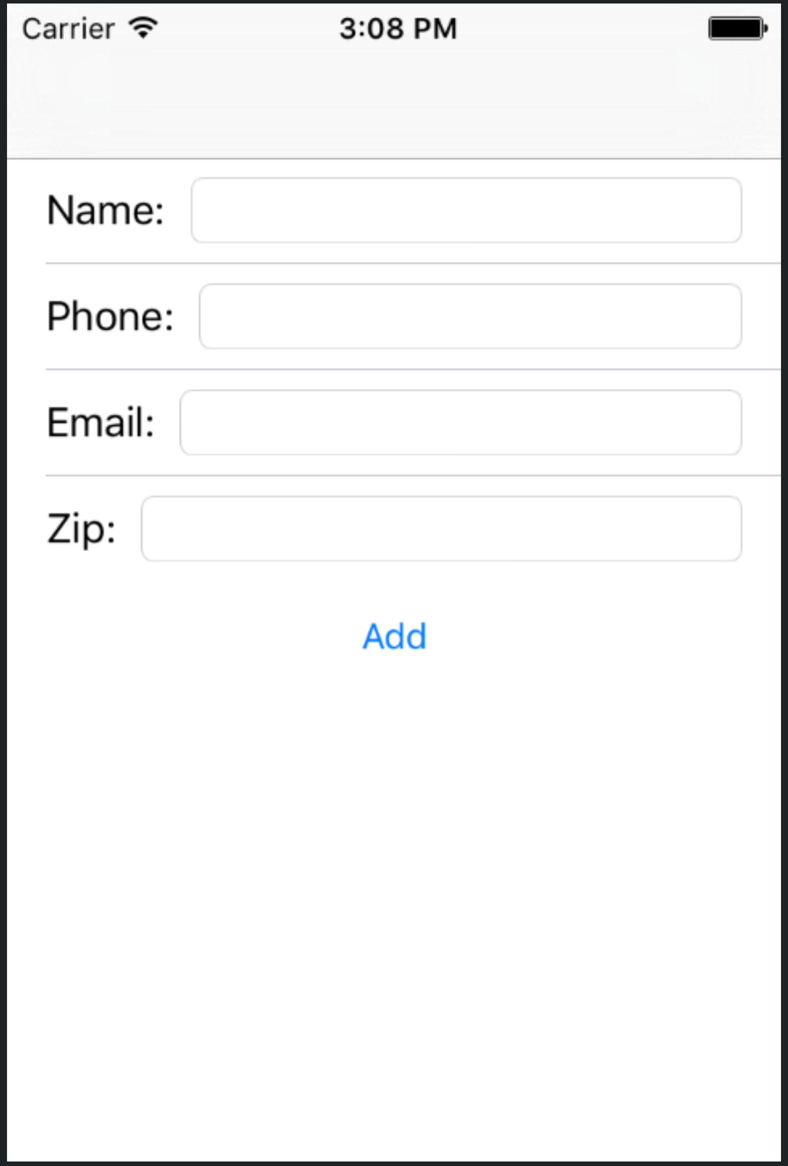
Why Unit Tests?


TDDTalkMockup 〉 iPhone 4s
TDDTalkMockup | Build TDDTalkMockup: Succeeded | 14/08/16 at 15:59
5
Coding
Log
TDDTalkMockup 〉 TDDTalkMockup 〉 YouAreHere.storyboard 〉 Table View Controller Scene 〉 Table View Controller
You are here
System Under Test
Navigation Controller
Navigation Controller
Table View Controller
Prototype Cells
Title
Detail
Table View
Prototype Content
Table View Controller
Prototype Cells
Title
Table View
Prototype Content
View Controller
Image View
Map View
Button
View Controller
Map View

Why Unit Tests?

— Fast feedback

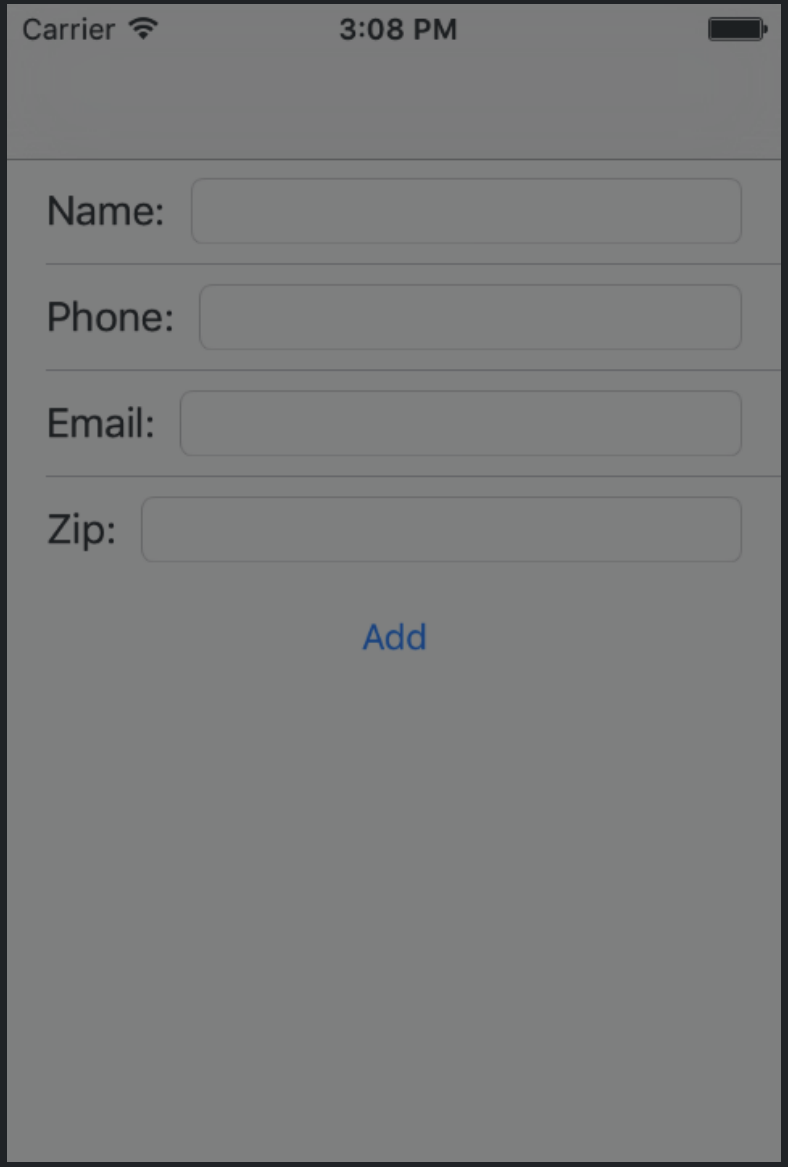

TDDTalkMockup | Build TDDTalkMockup: Succeeded | 14/08/16 at 15:59
TDDTalkMockup 〉 TDDTalkMockup 〉 YouAreHere.storyboard 〉 Table View Controller Scene 〉 Table View Controller
You are here
System Under Test
Navigation Controller
Table View
Table View

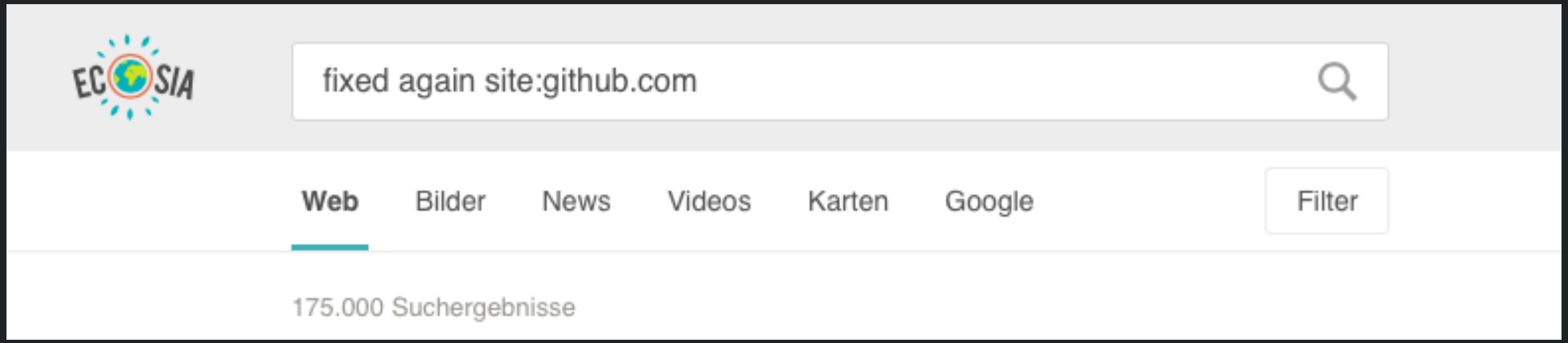


Name:	
Phone:	
Email:	
Zip:	

Why Unit Tests?

- Fast feedback
- Edge cases



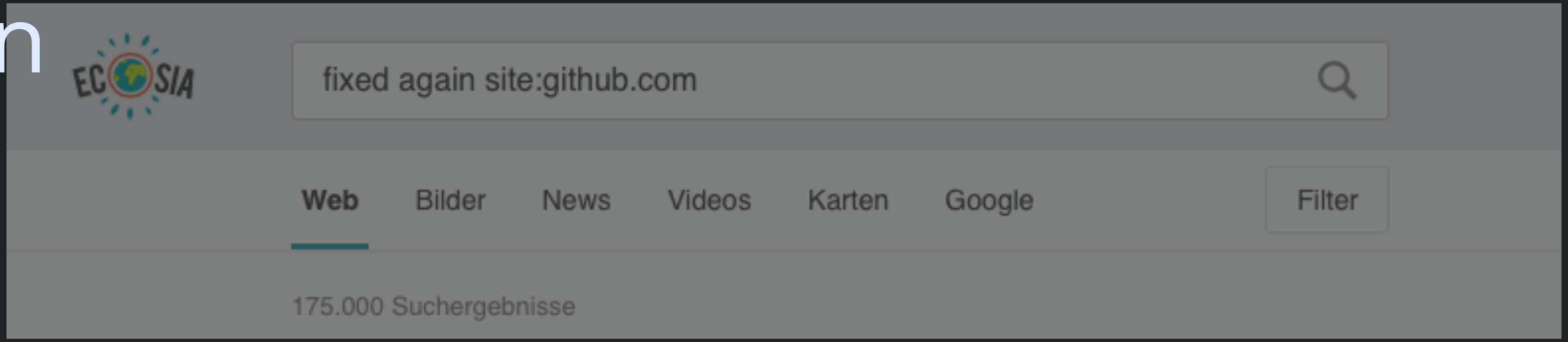


fixed again site:github.com

175.000 Suchergebnisse

Why Unit Tests?

- Fast feedback
- Edge cases
- Regression


fixed again site:github.com

175.000 Suchergebnisse

Why TDD?

Why TDD?

- Automatic testability
- Automatic modularity
- High test coverage
- Refactoring
- Living documentation

Keep in mind

Keep in mind

— No silver bullet
— Slower at first
— Training curve

🤓 Live Coding 🤓