

# SwiftUI at Scale

- “But it’s not production ready”
- Wrong!

Thomas Ricouard (@Dimillian) - Staff iOS engineer - Medium / Glose

# Why would you want to use SwiftUI?

Sorry for the boring long list of bullet points

- Because
- It's
- Less
- Code
- And
- More
- Fun

# Why would you want to use SwiftUI?

For real

- It's less code and more fun

# Now that you get it

We can get to some very boring, very concrete  
, yet exciting examples!

# **Exhibit A**

**Design & product come to you with this perfect, most  
beautiful list you ever saw**

# Exhibit A

4:22



FrenchKit 🇫🇷

Talks

- Speaker 1**  
SwiftUI is awesome
- Speaker 2**  
A mistake in SwiftUI
- Speaker 3**  
Cooking with SwiftUI
- Speaker 4**  
We have SwiftUI at home
- Speaker 5**  
I prefer Appkit than SwiftUI
- Speaker 6**  
UIKit is for pro, SwiftUI is for noob
- Speaker 7**  
SwiftUI is not my language
- Speaker 8**  
Why I'll never touch SwiftUI ever again
- Speaker 9**  
Swift compiler in SwiftUI
- Speaker 10**  
SwiftUI, the truth, and the deep state

4:22

FrenchKit 🇫🇷

Speakers

- S1
- S2
- S3
- S4
- S5
- S6

Talks

- Speaker 1**  
SwiftUI is awesome
- Speaker 2**  
A mistake in SwiftUI
- Speaker 3**  
Cooking with SwiftUI
- Speaker 4**  
We have SwiftUI at home
- Speaker 5**  
I prefer Appkit than SwiftUI
- Speaker 6**  
UIKit is for pro, SwiftUI is for noob
- Speaker 7**  
SwiftUI is not my language
- Speaker 8**  
Why I'll never touch SwiftUI ever again

# Exhibit A

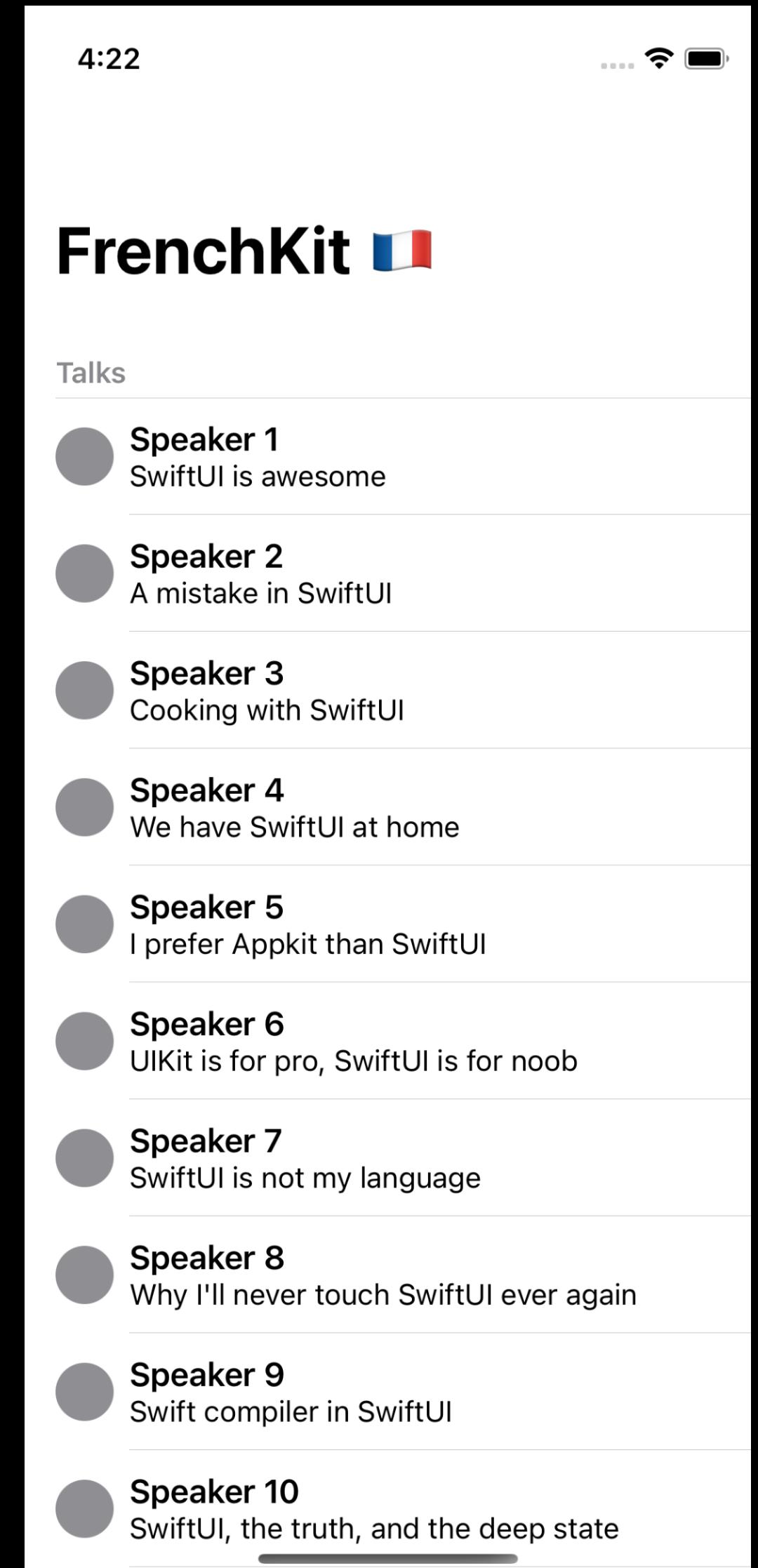
## With UIKit

1. Setting up the UITableView using the flavour of the day
  - A. Delegate/Datasource pattern
  - B. Diffable Datasource
  - C. Cell configuration
2. Making UITableViewCell with nice auto layout code or XIB
3. Place everything by hand
4. Realise that with the new design you would be safer with an UICollectionView
5. Rewriting all your code to fit UICollectionView
6. Lot of code, headache, rage, raising  seems like a good idea

# Exhibit A

## With SwiftUI

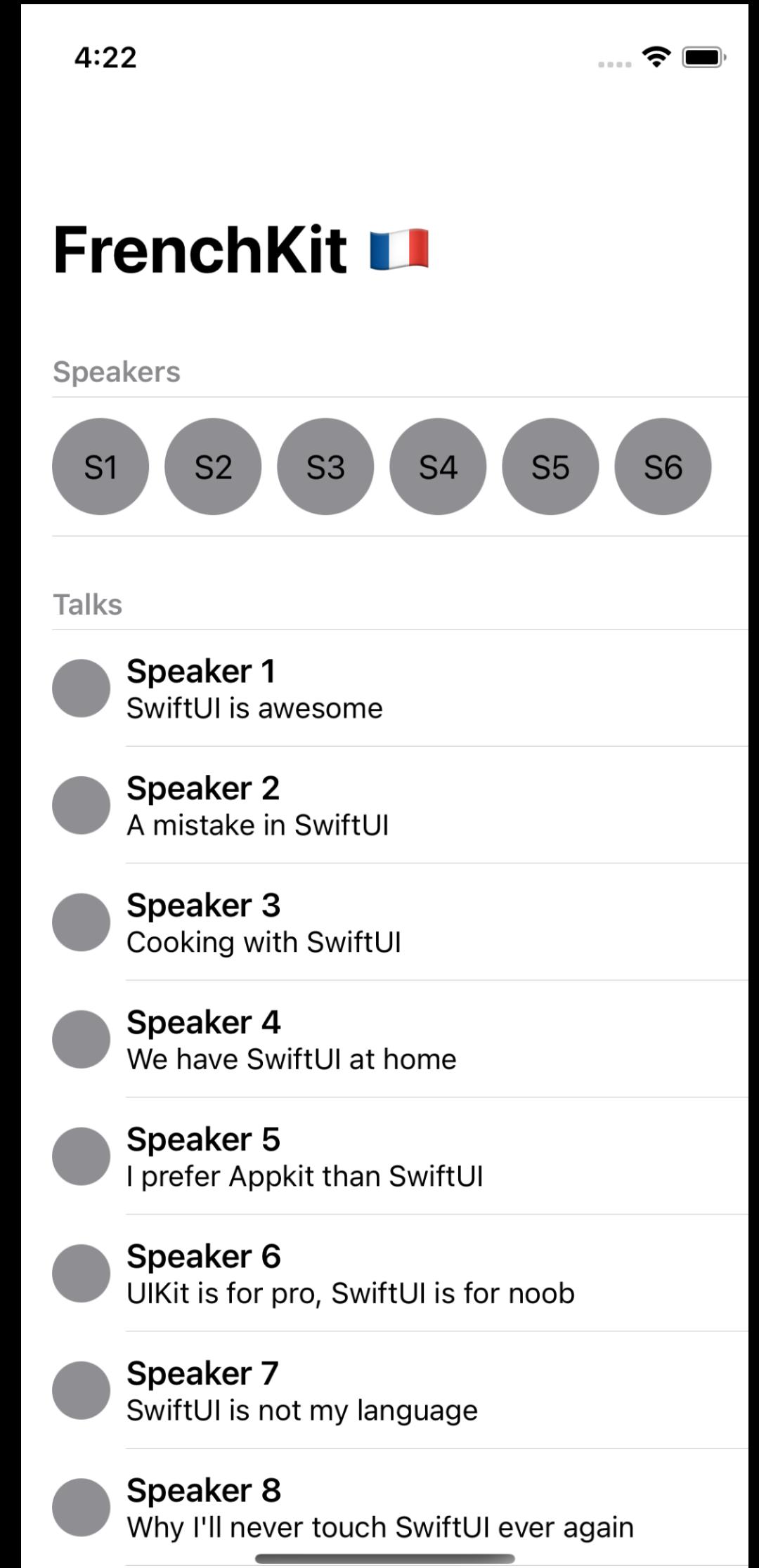
```
List {  
    Section("Talks") {  
        ForEach(speakers) { speaker in  
            SpeakerRow(speaker: speaker)  
        }  
    }  
}
```



# Exhibit A

## With SwiftUI

```
List {  
    Section("Speakers") {  
        ScrollView(.horizontal) {  
            LazyHStack {  
                ForEach(speakers) { speaker in  
                    SpeakerHeadRow(speaker: speaker)  
                }  
            }  
        }  
    }  
    Section("Talks") {  
        ForEach(speakers) { speaker in  
            SpeakerRow(speaker: speaker)  
        }  
    }  
}
```



# Exhibit B



# Exhibit B

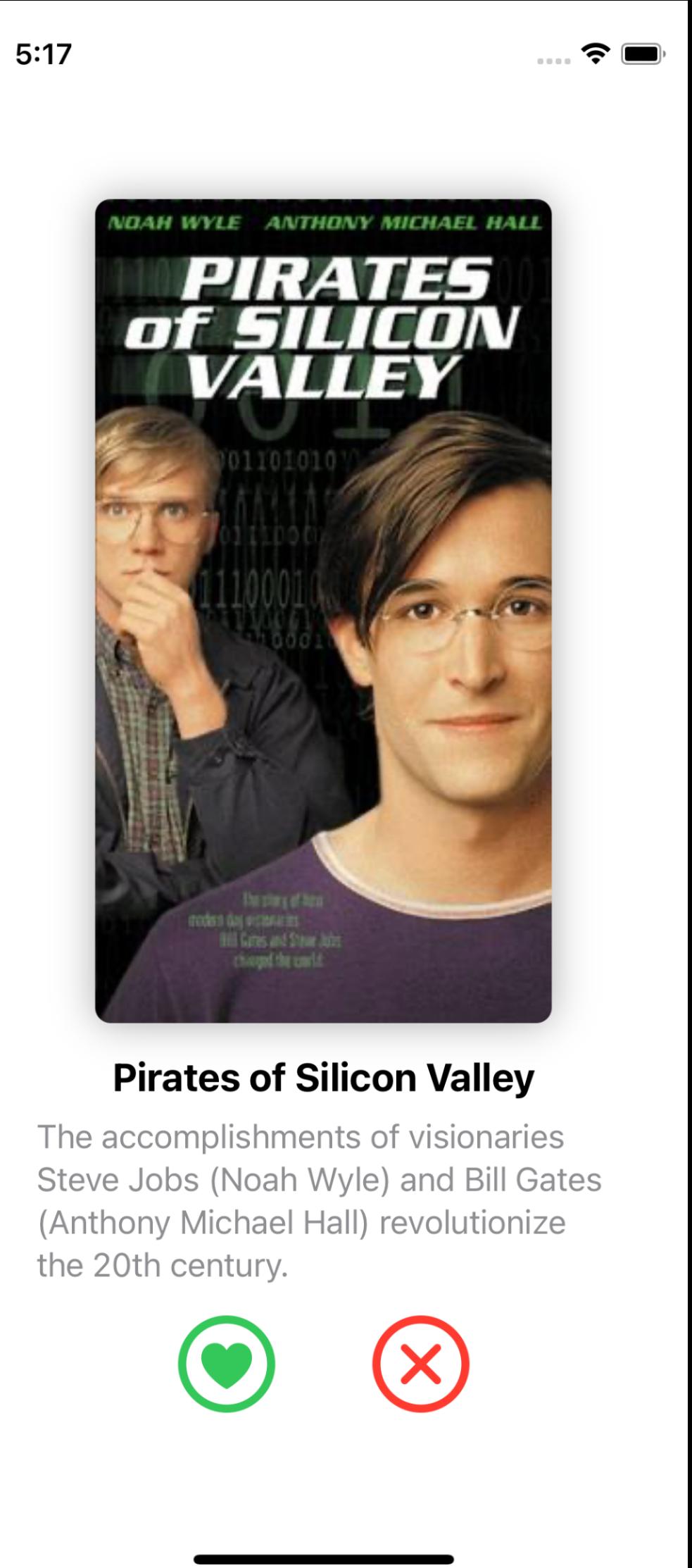
## With UIKit

1. Instantiate and configure all your views
2. addSubview everything
3. Add auto layout constraints
  - A. With some library like Cartography
  - B. With your own helpers
  - C. Using the ASCII format (!!?)
4. Run your app multiple times on multiple devices
5. Sure you could have done it with UIStackView but it still lot of boring code

# Exhibit B

## With SwiftUI

```
VStack(spacing: 16) {  
    AsyncImage(url: URL(string: "..."))  
    VStack(spacing: 8) {  
        Text("...")  
        Text("...")  
    }  
    .frame(maxWidth: 300)  
    HStack {  
        Image(systemName: "heart.circle")  
        Spacer()  
        Image(systemName: "xmark.circle")  
    }  
    .frame(maxWidth: 150)  
}
```



A ExhibitB

ExhibitB > iPhone 13 mini

Running ExhibitB on iPhone 13 mini

ContentView

ExhibitB > ContentView > body

```
9
10 struct ContentView: View {
11     var body: some View {
12         VStack(spacing: 16) {
13             AsyncImage(url: URL(string: "https://upload.wikimedia.org/wikipedia/en/3/30/Movieposterposv.jpg"))
14                 .cornerRadius(8)
15                 .shadow(radius: 10)
16             VStack(spacing: 8) {
17                 Text("Pirates of Silicon Valley")
18                     .fontWeight(.bold)
19                     .font(.title3)
20                 Text("The accomplishments of visionaries Steve Jobs (Noah Wyle) and Bill Gates (Anthony Michael Hall) revolutionize the 20th century.")
21                     .font(.body)
22                     .foregroundColor(.gray)
23             }
24
25             HStack {
26                 Image(systemName: "heart.circle")
27                     .resizable()
28                     .foregroundColor(.green)
29                     .frame(width: 50, height: 50)
30                 Spacer()
31                 Image(systemName: "xmark.circle")
32                     .resizable()
33                     .foregroundColor(.red)
34                     .frame(width: 50, height: 50)
35             }
36             .frame(maxWidth: 150)
37         }
38     }
39 }
40
41 struct ContentView_Previews: PreviewProvider {
42     static var previews: some View {
43         ContentView()
44     }
45 }
```

Content View

Preview paused

PIRATES  
of  
SILICON  
VALLEY

NOAH WYLE ANTHONY MICHAEL HALL

PIRATES OF SILICON VALLEY

Pirates of Silicon Valley

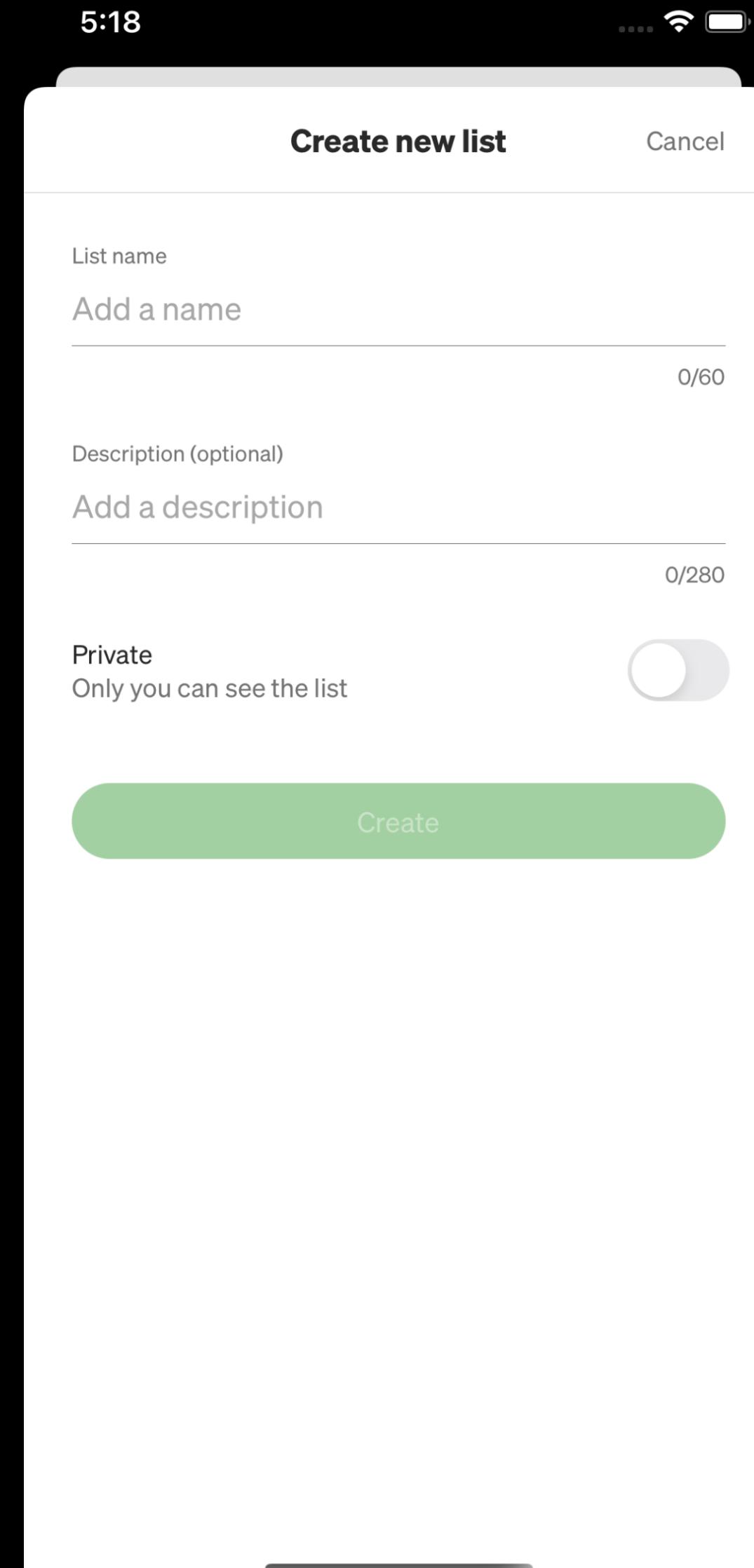
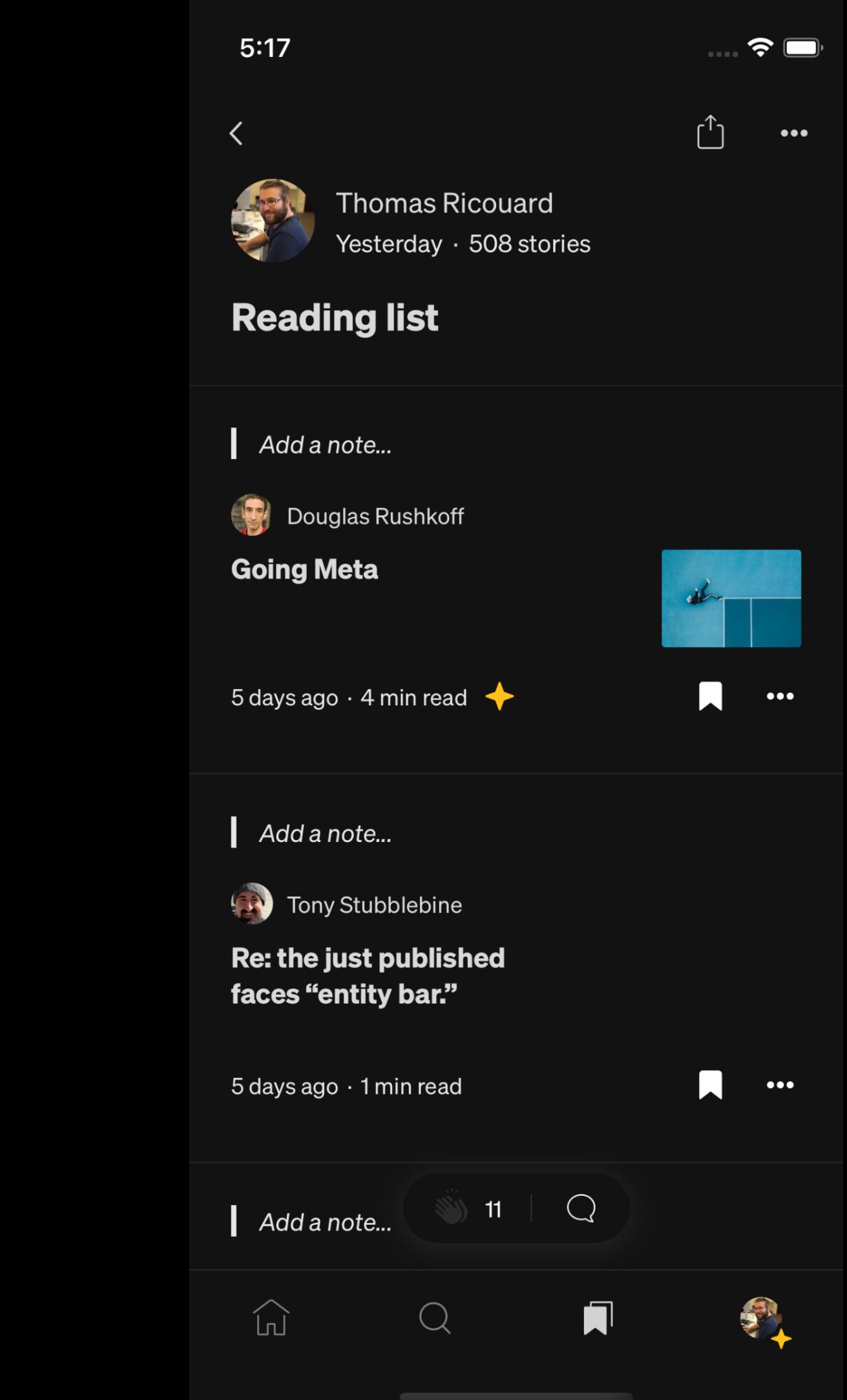
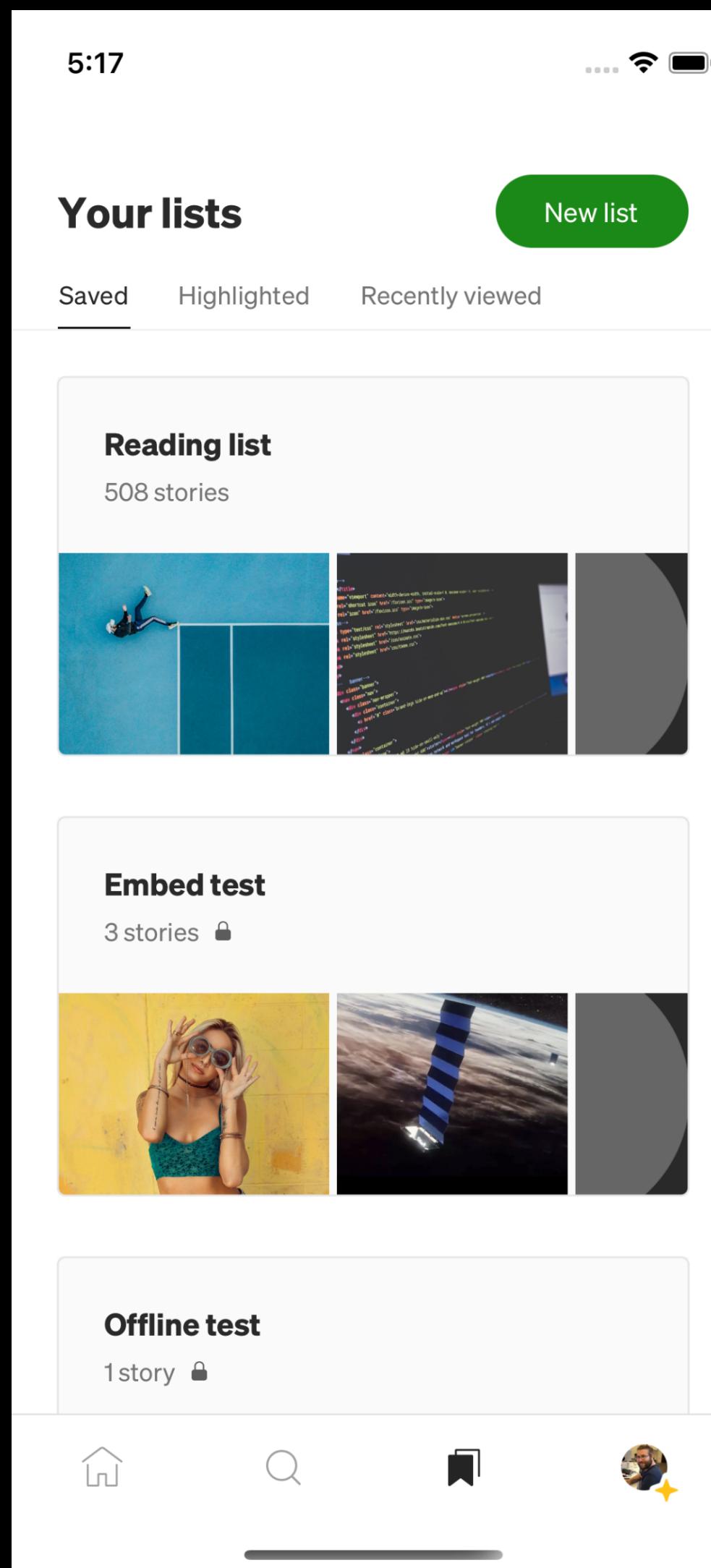
The accomplishments of visionaries Steve Jobs (Noah Wyle) and Bill Gates (Anthony Michael Hall) revolutionize the 20th century.

Line: 20 Col: 20

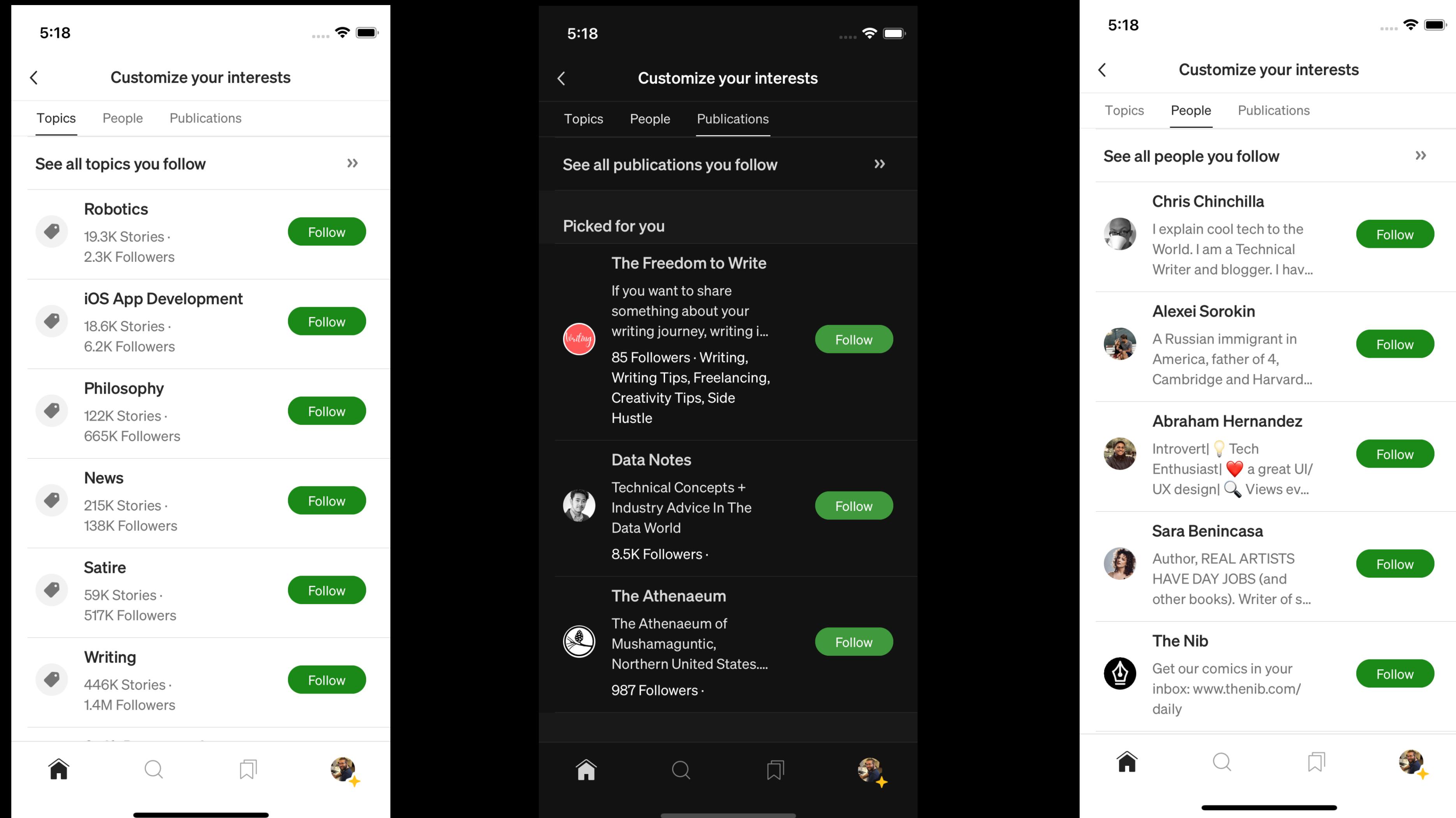
# **But why at scale then?**

**The Medium app is full of SwiftUI and it's in use by millions!**

# The Lists feature is in SwiftUI



# Recommendations UI is in SwiftUI



# And a lot more!

The image displays four screenshots of a mobile application interface, likely from the Medium app, showing various features:

- Screenshot 1 (Edit profile):** Shows the profile editing screen with a placeholder profile picture, a "Save" button, and tabs for "Name" and "Bio". The bio text is a long string of placeholder text.
- Screenshot 2 (Your lists):** Shows the "Your lists" screen with tabs for "Saved", "Highlighted" (which is selected), and "Recently viewed". It displays three highlighted posts by Thomas Ricouard with their content visible.
- Screenshot 3 (Activity):** Shows the "Activity" feed with several items:
  - Danilo Silva de Souza and Tony Stubblebine clapped for **Introducing RunewordsApp** 16 minutes ago.
  - K&K + 5 others followed you 4 hours ago.
  - Mayne Wong + 3 others clapped for **How do we use SwiftUI in the Medium iOS application** 9 hours ago.
  - Peter Duan + 10 others followed you 16 hours ago.
  - İlhan Sarı clapped for **SwiftUI AttributedString is not there yet** Yesterday.
  - Z clapped for **How do we use SwiftUI in the Medium iOS application** Yesterday.
  - Brian Davis followed you (with a green plus icon).
- Screenshot 4 (Responses):** Shows the "Responses (3)" screen with three comments:
  - Jim Lai** (May 14, 2022): "it's nice to see more Redux in SwiftUI. Thought I might also do a short opinion piece as an example to the principals that you outlined..." with a "Read more..." link and a reply count of 4.
  - Michael Long** (Jun 7, 2022): "There's a vast difference between clean SwiftUI and CLEAN SwiftUI." with a "Read more..." link and a reply count of 1.
  - Manuel Meyer** (May 28, 2022): "I actually set out to create a declarative variant of the Clean Architecture and have arrived at something very similar to redux as-well — I would love to here your feed..." with a "Read more..." link.A text input field at the bottom asks "What are your thoughts?"

# It's not about SwiftUI for SwiftUI

1. We can take full advantages of Swift Packages
2. Faster compile time
3. SwiftUI previews
4. Less code!
5. It's just much faster to make a new features
6. Not all screens or features are made for SwiftUI

# Was it a mistake?

Sometimes yes, 99% of the time no

- Harder to test
- SwiftUI is linked to the one shipped with iOS
- Runtime differences
- High level components have a lot of “free” behaviour
- Harder to customize in some cases.

# Thank you!

