

HotProspects App

TabView & enum & switch



iPhone 12 Pro
iOS 15.4



11:19



Everyone

Hello, World!



Everyone



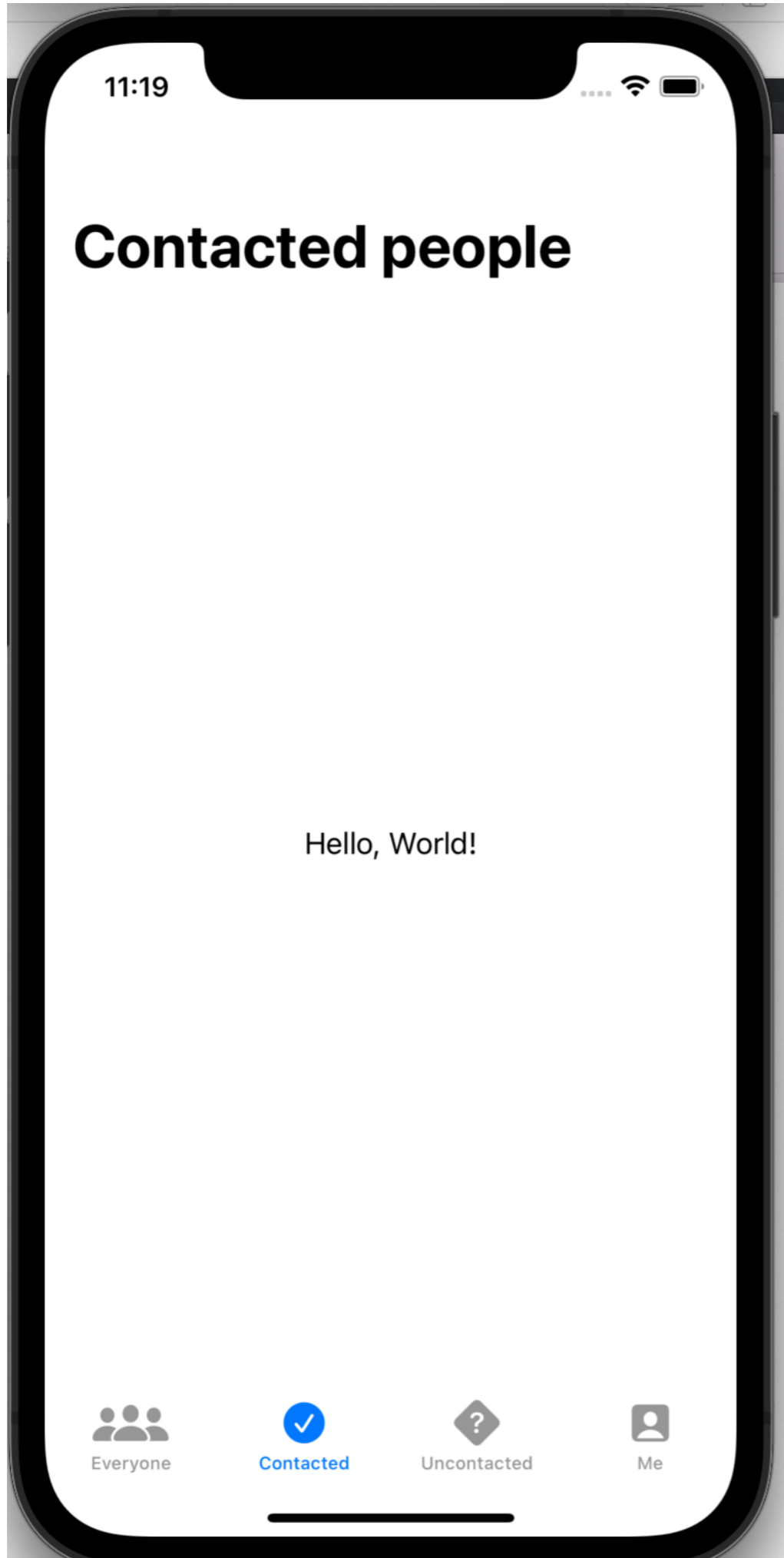
Contacted



Uncontacted



Me



11:20



Hello, World!



Everyone



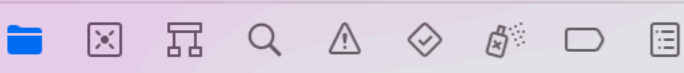
Contacted



Uncontacted

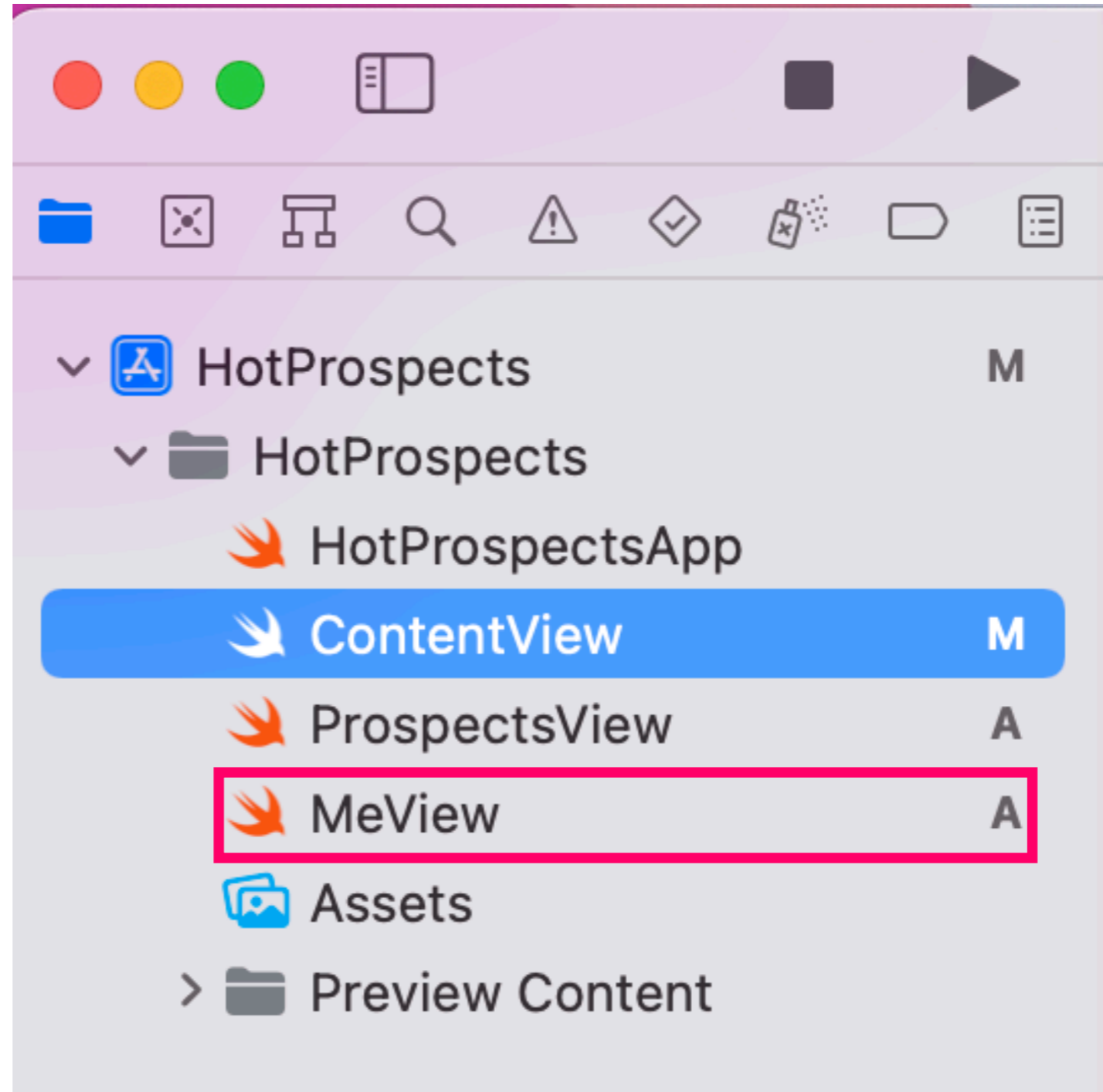


Me



- HotProspects M
- HotProspects
 - HotProspectsApp
 - ContentView M**
 - ProspectsView A
 - MeView A
 - Assets
 - Preview Content

```
1 //
2 // ContentView.swift
3 // HotProspects
4 //
5 // Created by Apple on 2022/5/18.
6 //
7
8 import SwiftUI
9
10 struct ContentView: View {
11     var body: some View {
12         TabView{
13             ProspectsView(filter: .none)
14                 .tabItem{
15                     Label("Everyone",
16                         systemImage:
17                         "person.3")
18                 }
19         }
20     }
21 }
```



MeView

Choose a template for your new file:

iOS

macOS

watchOS

tvOS

DriverKit

Filter

Source



Swift File



Cocoa Touch Class



UI Test Case Class



Unit Test Case Class



Objective-C File



Header File



C File



C++ File



Metal File

User Interface



SwiftUI View



Storyboard



View



Empty



Launch Screen

Cancel

Previous

Next

Favorites

iCloud

- iCloud Drive
- Shared

Tags

- Euler method 1+2
- SCI2020
- IphoneGame
- 藍色
- 重要事項
- 黃色
- 灰色
- All Tags...

Save As:

Tags:

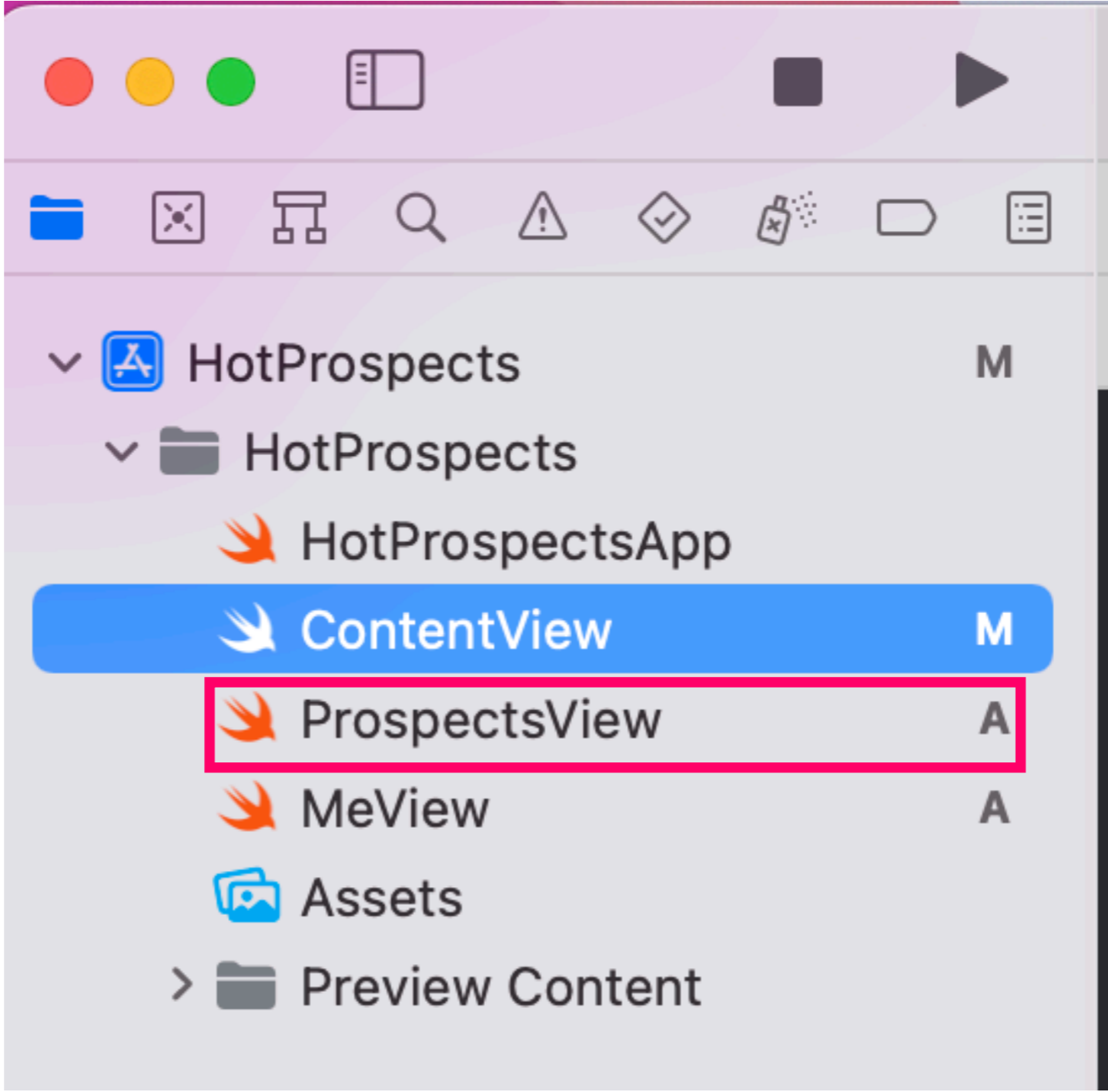
- Assets.xcassets >
- ContentView.swift
- HotProspectsApp.swift
- MeView.swift
- Preview Content >
- ProspectsView.swift

Group

- Targets
- HotProspects
 -
 -
 -

- HotProspects M
 - HotProspects
 - HotProspectsApp
 - ContentView M
 - ProspectsView A
 - MeView A
 - Assets
 - Preview Content

```
HotProspects > HotProspects > MeView > No Selection  
3 // HotProspects  
4 //  
5 // Created by Apple on 2022/5/18.  
6 //  
7  
8 import SwiftUI  
9  
10 struct MeView: View {  
11     var body: some View {  
12         Text("Hello, World!")  
13     }  
14 }  
15  
16 struct MeView_Previews: PreviewProvider {  
17     static var previews: some View {  
18         MeView()  
19     }  
20 }  
21
```



ProspectsView

Choose a template for your new file:

iOS

macOS

watchOS

tvOS

DriverKit

Filter

Source



Swift File



Cocoa Touch Class



UI Test Case Class



Unit Test Case Class



Objective-C File



Header File



C File



C++ File



Metal File

User Interface



SwiftUI View



Storyboard



View



Empty



Launch Screen

Cancel

Previous

Next

```
10 struct ProspectsView: View {
11     enum FilterType {
12         case none, contacted, uncontacted
13     }
14     let filter: FilterType
15     var body: some View {
16         NavigationView{
17             Text("Hello, World!")
18             .navigationTitle(title)
19         }
20     }
21
22     var title: String{
23         switch filter {
24             case .none:
25                 return "Everyone"
26             case .contacted:
27                 return "Contacted people"
28             case .uncontacted:
29                 return "Uncontacted people"
30         }
31     }
32 }
```

```
10 struct ProspectsView: View {
11     enum FilterType {
12         case none, contacted, uncontacted
13     }
14     let filter: FilterType
15     var body: some View {
16         NavigationView{
17             Text("Hello, World!")
18             .navigationTitle(title)
19         }
20     }
21
22     var title: String{
23         switch filter {
24             case .none:
25                 return "Everyone"
26             case .contacted:
27                 return "Contacted people"
28             case .uncontacted:
29                 return "Uncontacted people"
30         }
31     }
32 }
```

宣告列舉型態
FilterType

宣告常數參數
filter

```
10 struct ProspectsView: View {
11     enum FilterType {
12         case none, contacted, uncontacted
13     }
14     let filter: FilterType
15     var body: some View {
16         NavigationView{
17             Text("Hello, World!")
18             .navigationTitle(title)
19         }
20     }
21
22     var title: String{
23         switch filter {
24             case .none:
25                 return "Everyone"
26             case .contacted:
27                 return "Contacted people"
28             case .uncontacted:
29                 return "Uncontacted people"
30         }
31     }
32 }
```

使用變數title設定
navigationTitle

```
10 struct ProspectsView: View {
11     enum FilterType {
12         case none, contacted, uncontacted
13     }
14     let filter: FilterType
15     var body: some View {
16         NavigationView{
17             Text("Hello, World!")
18             .navigationTitle(title)
19         }
20     }
21
22     var title: String{
23         switch filter {
24             case .none:
25                 return "Everyone"
26             case .contacted:
27                 return "Contacted people"
28             case .uncontacted:
29                 return "Uncontacted people"
30         }
31     }
32 }
```

宣告變數 title
以switch filter設定內容

```
34 struct ProspectsView_Previews: PreviewProvider {
35     static var previews: some View {
36         ProspectsView(filter: .none)
37     }
38 }
```

Content View

```
struct ContentView: View {
    var body: some View {
        TabView{
            ProspectsView(filter: .none)
                .tabItem{
                    Label("Everyone", systemImage: "person.3")
                }
            ProspectsView(filter: .contacted)
                .tabItem{
                    Label("Contacted", systemImage: "checkmark.circle")
                }
            ProspectsView(filter: .uncontacted)
                .tabItem{
                    Label("Uncontacted", systemImage: "questionmark.diamond")
                }
            MeView()
                .tabItem{
                    Label("Me", systemImage: "person.crop.square")
                }
        }
    }
}
```

```
TabView{
  ProspectsView(filter: .none)
    .tabItem{
      Label("Everyone", systemImage: "person.3")
    }
  ProspectsView(filter: .contacted)
    .tabItem{
      Label("Contacted", systemImage: "checkmark.circle")
    }
  ProspectsView(filter: .uncontacted)
    .tabItem{
      Label("Uncontacted", systemImage: "questionmark.diamond")
    }
  MeView()
    .tabItem{
      Label("Me", systemImage: "person.crop.square")
    }
}
```

