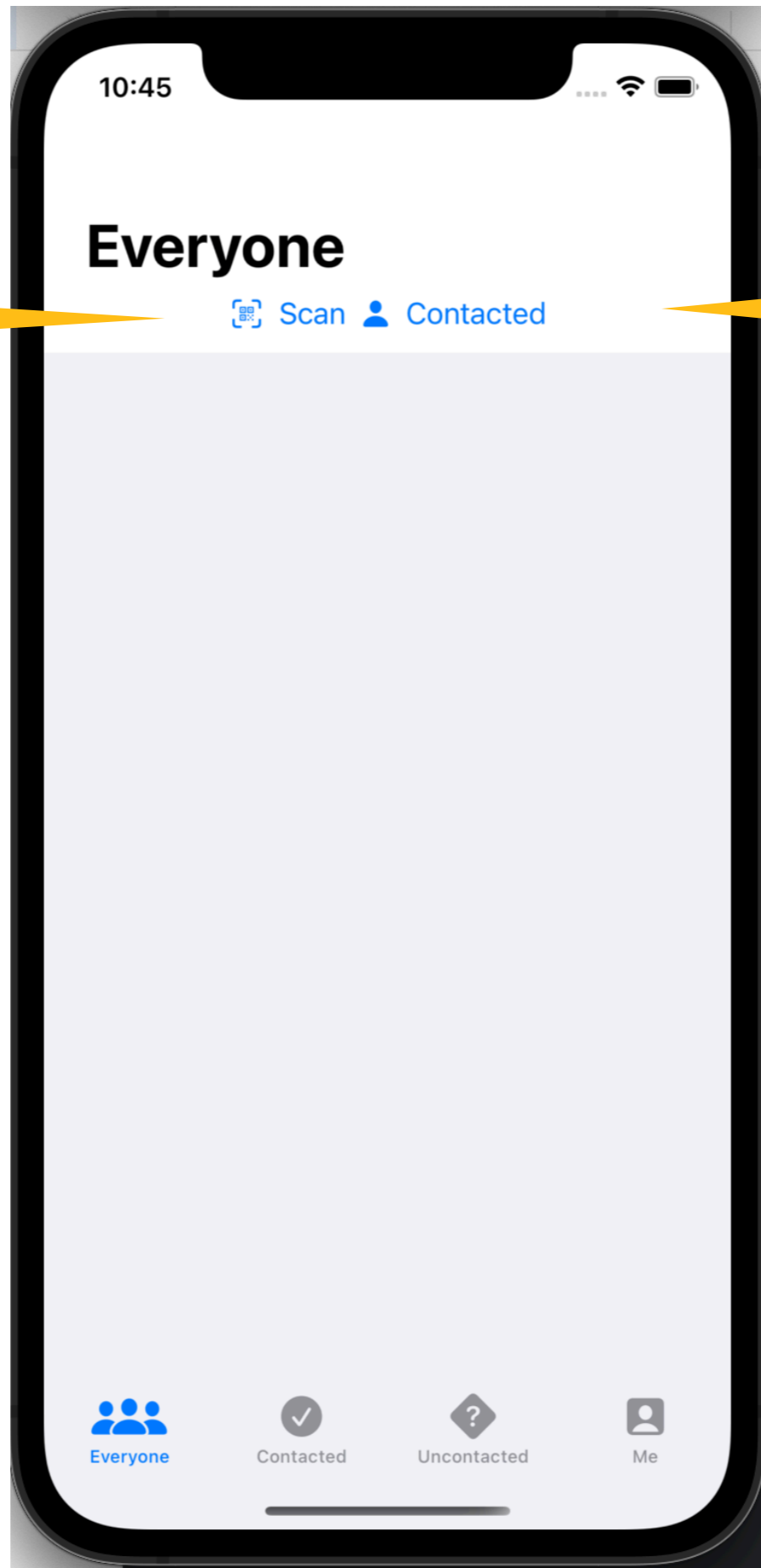




# HotProspects App II

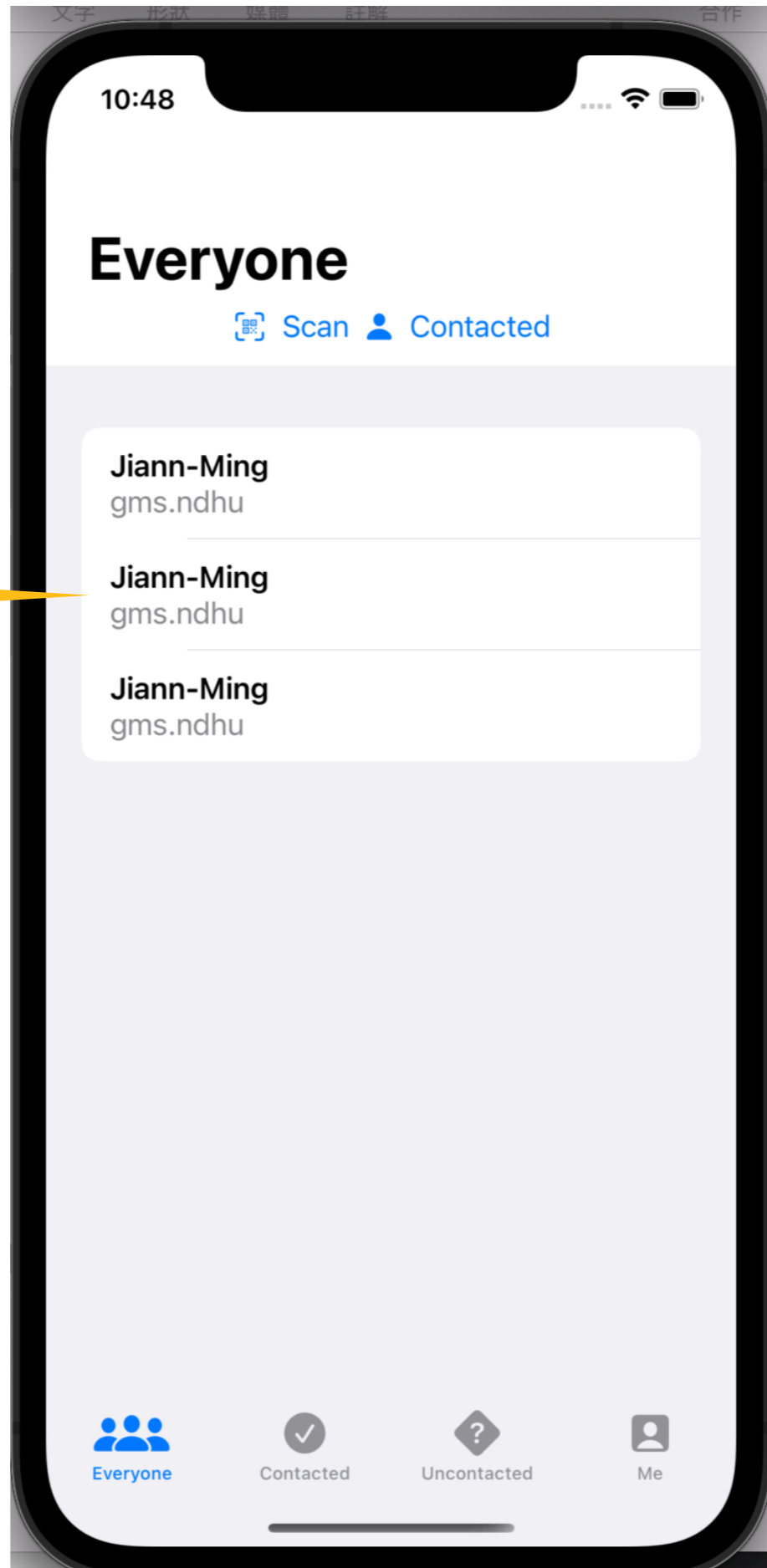
資料篩選

篩選  
未聯絡資料

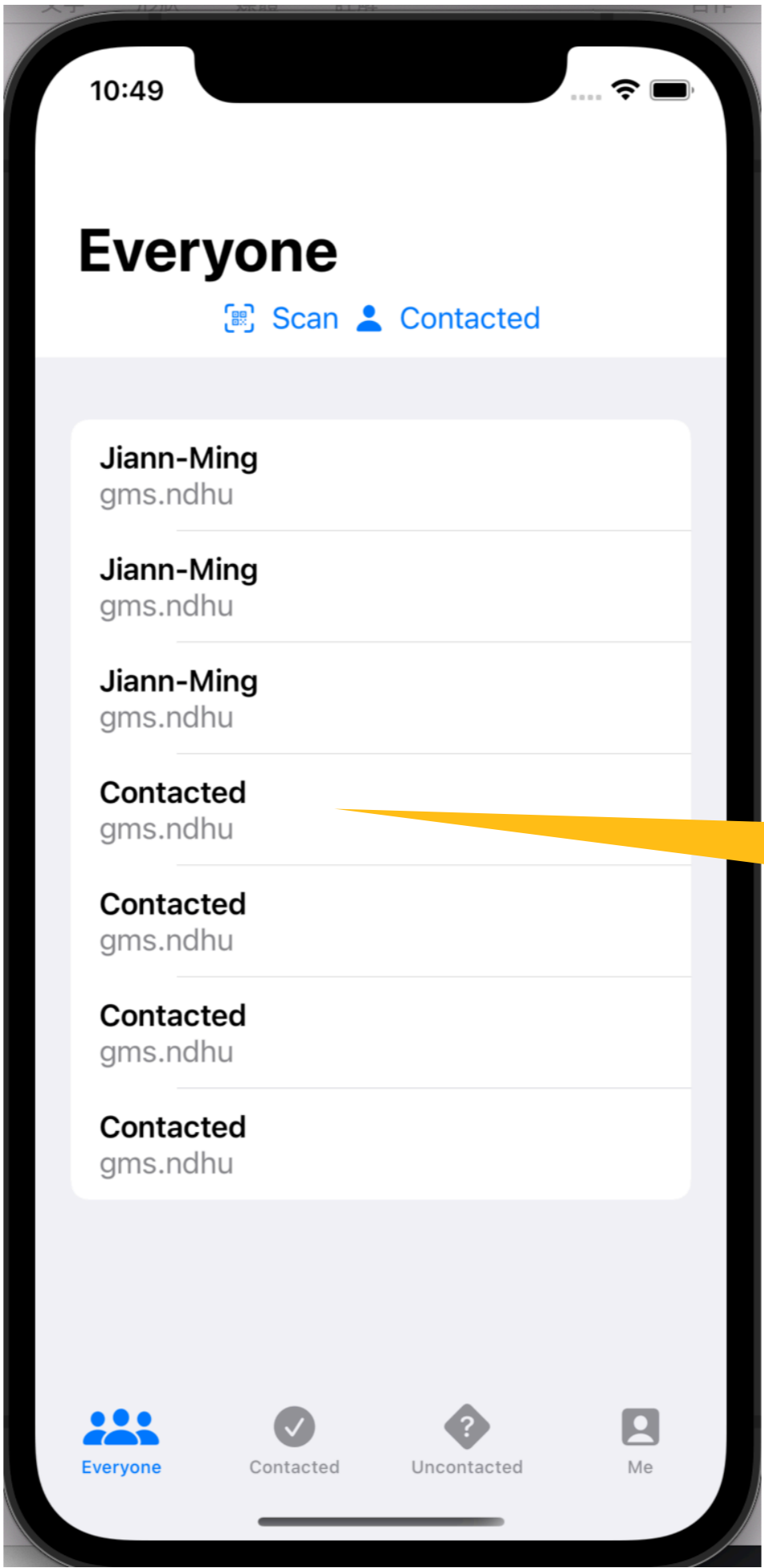


新增

新增已聯絡  
資料



新增多筆  
資料



顯示所有  
資料

新增多筆  
已聯絡資料

10:50



## Contacted people

 Scan  Contacted

Contacted  
gms.ndhu

Contacted  
gms.ndhu

Contacted  
gms.ndhu

Contacted  
gms.ndhu

篩選  
已聯絡資料



Everyone



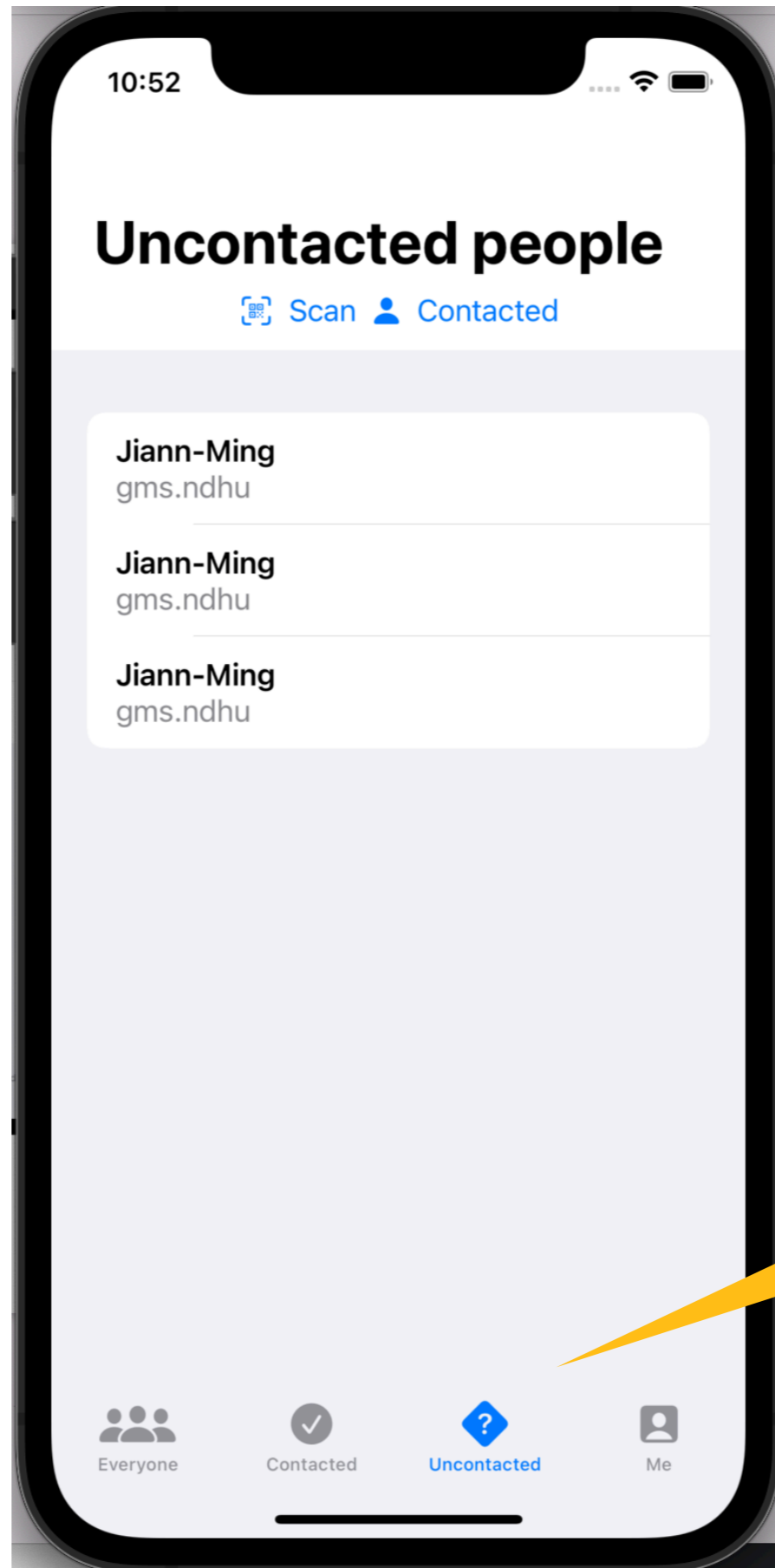
Contacted



Uncontacted



Me



篩選  
未聯絡資料

# 步驟一：新增view Prospect

```
7
8 import SwiftUI
9
10 class Prospect: Identifiable, Codable {
11     var id = UUID()
12     var name = "Anonymous"
13     var emailAddress = ""
14     var isContacted = false
15 }
16
17 @MainActor class Prospects: ObservableObject {
18     @Published var people: [Prospect]
19
20     init(){
21         people = []
22     }
23 }
```

## 定義類別 Prospect

```
7
8 import SwiftUI
9
10 class Prospect: Identifiable, Codable {
11     var id = UUID()
12     var name = "Anonymous"
13     var emailAddress = ""
14     var isContacted = false
15 }
16
17 @MainActor class Prospects: ObservableObject {
18     @Published var people: [Prospect]
19
20     init(){
21         people = []
22     }
23 }
```

Identifiable  
Codable

## 定義主角類別 Prospects

```
7
8 import SwiftUI
9
10 class Prospect: Identifiable, Codable {
11     var id = UUID()
12     var name = "Anonymous"
13     var emailAddress = ""
14     var isContacted = false
15 }
16
17 @MainActor class Prospects: ObservableObject {
18     @Published var people: [Prospect]
19
20     init(){
21         people = []
22     }
23 }
```

ObservableObject

主角  
類別

# 步驟二、新增 View ProspectsView

```
7
8 import SwiftUI
9
10 struct ProspectsView: View {
11     enum FilterType {
12         case none, contacted, uncontacted
13     }
14     @EnvironmentObject var prospects: Prospects
15     let filter: FilterType
16     var body: some View { ... }
17
18     // ...
19
20     // ...
21
22     // ...
23
24     // ...
25
26     // ...
27
28     // ...
29
30     // ...
31
32     // ...
33
34     // ...
35
36     // ...
37
38     // ...
39
40     // ...
41
42     // ...
43
44     // ...
45
46     // ...
47
48     // ...
49
50     // ...
51
52     // ...
53
54     // ...
55
56     // ...
57
58     var title: String{ ... }
59
60     // ...
61
62     // ...
63
64     // ...
65
66     // ...
67
68     var filteredProspects: [Prospect] { ... }
69
70     // ...
71
72     // ...
73
74     // ...
75
76     // ...
77
78     // ...
79 }
80
81 struct ProspectsView_Previews: PreviewProvider {
82     static var previews: some View {
83         ProspectsView(filter: .none)
84             .environmentObject(Prospects())
85     }
86 }
```

# A. 宣告環境物件變數與 外傳常數參數

列舉三種篩選  
狀況

```
7
8 import SwiftUI
9
10 struct ProspectsView: View {
11     enum FilterType {
12         case none, contacted, uncontacted
13     }
14     @EnvironmentObject var prospects: Prospects
15     let filter: FilterType
16     var body: some View { ... }
17
18     var title: String{ ... }
19
20     var filteredProspects: [Prospect] { ... }
21 }
22
23 struct ProspectsView_Previews: PreviewProvider {
24     static var previews: some View {
25         ProspectsView(filter: .none)
26             .environmentObject(Prospects())
27     }
28 }
```

```

7
8 import SwiftUI
9
10 struct ProspectsView: View {
11     enum FilterType {
12         case none, contacted, uncontacted
13     }
14     @EnvironmentObject var prospects: Prospects
15     let filter: FilterType
16     var body: some View { ... }
17
18     // ...
19
57
58     var title: String{ ... }
68     var filteredProspects: [Prospect] { ... }
79 }
80
81 struct ProspectsView_Previews: PreviewProvider {
82     static var previews: some View {
83         ProspectsView(filter: .none)
84             .environmentObject(Prospects())
85     }
86 }

```

宣告環境物件，  
使變數prospects為  
主角類別

```
7
8 import SwiftUI
9
10 struct ProspectsView: View {
11     enum FilterType {
12         case none, contacted, uncontacted
13     }
14     @EnvironmentObject var prospects: Prospects
15     let filter: FilterType
16     var body: some View { ... }
17
18     var title: String{ ... }
19
20     var filteredProspects: [Prospect] { ... }
21 }
22
23 struct ProspectsView_Previews: PreviewProvider {
24     static var previews: some View {
25         ProspectsView(filter: .none)
26             .environmentObject(Prospects())
27     }
28 }
```

宣告傳入的常數參數

# B. 設計變數 `title` 的 `closure` (封包)

```

7
8 import SwiftUI
9
10 struct ProspectsView: View {
11     enum FilterType {
12         case none, contacted, uncontacted
13     }
14     @EnvironmentObject var prospects: Prospects
15     let filter: FilterType
16     var body: some View { ... }
57
58     var title: String{ ... }
68     var filteredProspects: [Prospect] { ... }
79 }
80
81 struct ProspectsView_Previews: PreviewProvider {
82     static var previews: some View {
83         ProspectsView(filter: .none)
84             .environmentObject(Prospects())
85     }
86 }

```

展開變數 title的closure

```
14  @EnvironmentObject var prospects: Prospects
15  let filter: FilterType
16  var body: some View { ... }
57
58  var title: String{
59      switch filter {
60      case .none:
61          return "Everyone"
62      case .contacted:
63          return "Contacted people"
64      case .uncontacted:
65          return "Uncontacted people"
66      }
67  }
```

依據傳入的常數參數filter，  
設定變數title的，顯示為  
View的標題

# C. 篩選環境物件變數 prospects中的資料

依據傳入的常數參數filter，篩選環境物件變數prospects中的資料，並儲存在filteredProspects陣列中

```
15     let filter: FilterType
16     var body: some View { ... }
57
58     var title: String{ ... }
68     var filteredProspects: [Prospect] {
69         switch filter {
70             case .none:
71                 return prospects.people
72             case .contacted:
73                 return prospects.people.filter
74                     { $0.isContacted }
75             case .uncontacted:
76                 return prospects.people.filter
77                     { !$0.isContacted }
78         }
79     }
80 }
```

# D.設計body View

```
14  @EnvironmentObject var prospects: Prospects
15  let filter: FilterType
16  var body: some View {
17      NavigationView{
18          VStack{
19              HStack{...}
39              List {...}
50              .navigationTitle(title)
51          }
52      }
53  }
54
55  var title: String{...}
```

使用  
NavigationView  
設計 List

# E.設計List

```
List {  
    forEach(filteredProspects) {prospect in  
        VStack(alignment: .leading){  
            Text(prospect.name)  
                .font(.headline)  
            Text(prospect.emailAddress)  
                .foregroundColor(.secondary)  
        }  
    }  
}
```

```
}
```

```
}
```

# F.設計資料新增按鈕

```
HStack{
  Button {
    let prospect = Prospect()
    prospect.name = "Jiann-Ming"
    prospect.emailAddress = "gms.ndhu"
    prospect.isContacted = false
    prospects.people.append(prospect)
  } label: {
    Label("Scan", systemImage: "qrcode.viewfinder")
  }
  Button {
    let prospect = Prospect()
    prospect.name = "Contacted"
    prospect.emailAddress = "gms.ndhu"
    prospect.isContacted = true
    prospects.people.append(prospect)
  } label: {
    Label("Contacted", systemImage: "person.fill")
  }
}
```

# 步驟三、設計

## Content View

```
7
8 import SwiftUI
9
10 struct ContentView: View {
11     @StateObject var prospects = Prospects()  Expression requi...
12     var body: some View {
13         TabView{...}
31         .environmentObject(prospects)
32     }
33 }
34
35 struct ContentView_Previews: PreviewProvider {
36     static var previews: some View {
37         ContentView()
38     }
39 }
40
```


# A. 宣告狀態物件變數

```
7
8 import SwiftUI
9
10 struct ContentView: View {
11     @StateObject var prospects = Prospects()
12     var body: some View {
13         TabView{...}
31         .environmentObject(prospects)
32     }
33 }
34
35 struct ContentView_Previews: PreviewProvider {
36     static var previews: some View {
37         ContentView()
38     }
39 }
40
```



Expression requi...

# B.使用TabView，傳入 環境物件變數

```
7
8 import SwiftUI
9
10 struct ContentView: View {
11     @StateObject var prospects = Prospects()  Expression requi...
12     var body: some View {
13         TabView{...}
31         .environmentObject(prospects)
32     }
33 }
34
35 struct ContentView_Previews: PreviewProvider {
36     static var previews: some View {
37         ContentView()
38     }
39 }
40
```

# C. 設計 Tab View 的 Items

```
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")
    }
}
.environmentObject(prospect)
```

# **D.新增MeView**