

TipCalculator

Core Data

1:29



Bill:

Amount of Bill to pay

Tip Percentage:

0%

5%

10%

15%

20%

Total Bill:0.0

1:30



Bill:

123

Tip Percentage:

0%

5%

10%

15%

20%

Total Bill:129.15

1:31



Bill:

123

Tip Percentage:

0%

5%

10%

15%

20%

Total Bill:135.3

Three Sections

```
struct ContentView: View {
    var body: some View {
        Form {
            Section{
                Text("Hello, world!")
                Text("Hello, world!")
            }
            Section {
                Text("Hello, world!")
            }
            Section {
                Text("Hello, world!")
            }
        }
    }
}
```

```
}
}
}
```

one

3:46

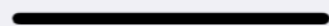


Hello, world!

Hello, world!

Hello, world!

Hello, world!



7

設定狀態變數

狀態變數

checkAmount

代表消費金額

```
struct ContentView: View {  
    @State private var checkAmount = ""  
    @State private var tipPercentage = 1
```

狀態變數

tipPercentage代表稅
率選擇，1代表5%

```
struct ContentView: View {
    var body: some View {
        Form {
            Section{
                Text("Hello, world!")
                Text("Hello, world!")
            }
            Section {
                Text("Hello, world!")
            }
            Section {
                Text("Hello, world!")
            }
        }
    }
}
```

宣告private

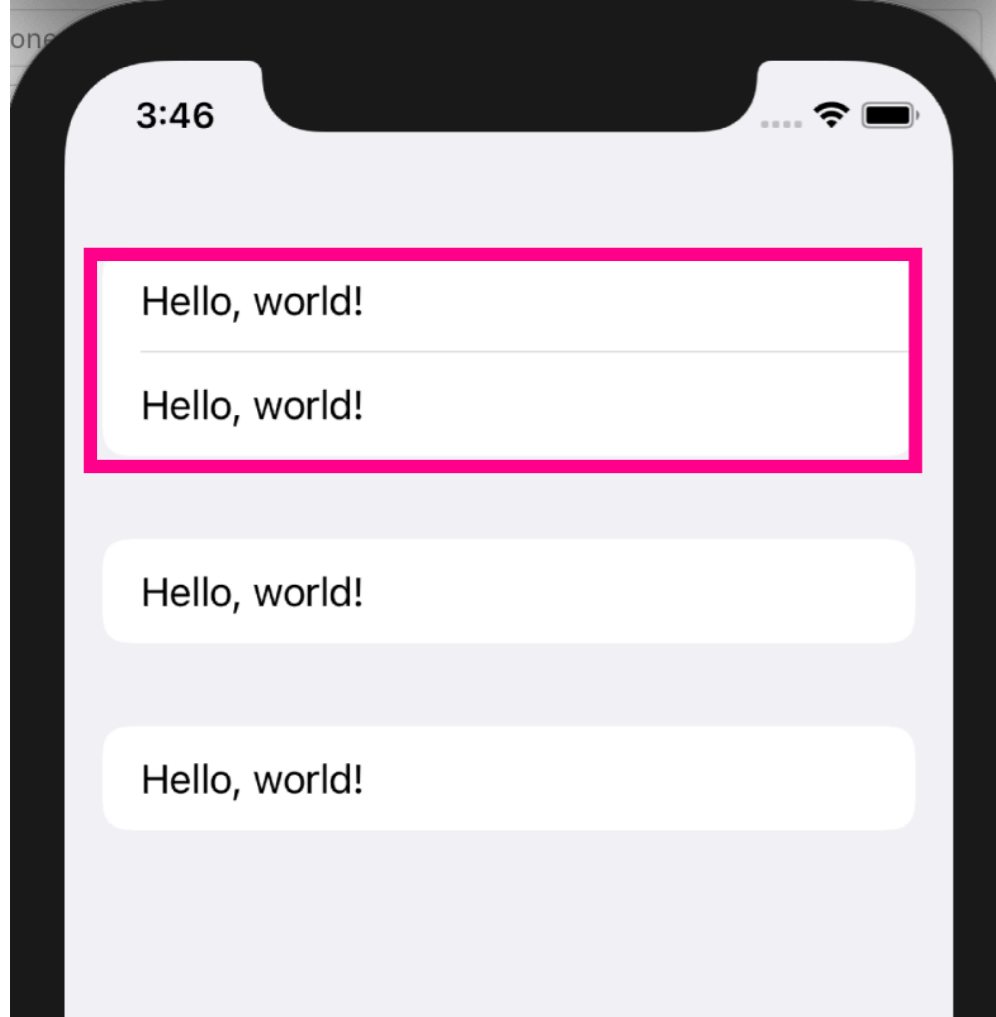
狀態變數

checkAmount

tipPercentage

```
}
}
}
```

修改第一個Section Text and Textfield



修改第一個Section的顯示內容：
改變第一個文字欄位的顯示內容
外，並將第二個文字欄位，改為
TextField，讓使用者輸入字串

```
Section{  
  Text("Bill:")  
  TextField("Amount of Bill to pay", text: $checkAmount)  
}
```

連結text到狀態變數，
checkAmount，使用者輸入的字串會
自動儲存到checkAmount中

Bill:

123|

**修改第三個Section
Text**

顯示變數totalAfterTip的內容

```
Section{  
    Text("Total Bill:" + String(totalAfterTip))  
}
```

宣告字串變數
totalAfterTip

設定稅率，將
tipPercentage轉
為百分比

```
struct ContentView: View {  
    @State private var checkAmount = ""  
    @State private var tipPercentage = 1  
    var totalAfterTip: Double {  
        let userTipSelection = (Double(tipPercentage) * 5) / 100  
        let totalBill = Double(checkAmount) ?? 0  
        let amountToPay = totalBill + (totalBill * userTipSelection)  
        return amountToPay  
    }  
}
```

在closure中計算，回傳計算
結果

將checkAmount轉為實數，
計算應付金額

Bill:

123

Total Bill:129.15

**修改第二個Section
設計Picker，讓使用者選
擇稅率**

選擇器的顯示文字

連結狀態變數
tipPercentage

將代表不同選擇的tag值，與狀態變數連結

```
Section{
    Text("Tip Percentage:")
    Picker("Tip Amount",
        selection: $tipPercentage){
        Text("0%").tag(0)
        Text("5%").tag(1)
        Text("10%").tag(2)
        Text("15%").tag(3)
        Text("20%").tag(4)
    }
}
.pickerStyle(SegmentedPickerStyle())
}
```

不同的tag值的顯示文字

```
Section{
    Text("Tip Radix:")
    Picker("Tip Select", selection:
        $tipRadix){
        Text("2").tag(0)
        Text("4").tag(1)
        Text("8").tag(2)
        Text("16").tag(3)
    }
}
.pickerStyle(SegmentedPickerStyle())
}
```

指定SegmentredPickerStyle()

Tip Percentage:

0%

5%

10%

15%

20%

Tip Percentage:

0%

5%

10%

15%

20%

Tip Percentage:

0%

5%

10%

15%

20%

Tip Percentage:

0%

5%

10%

15%

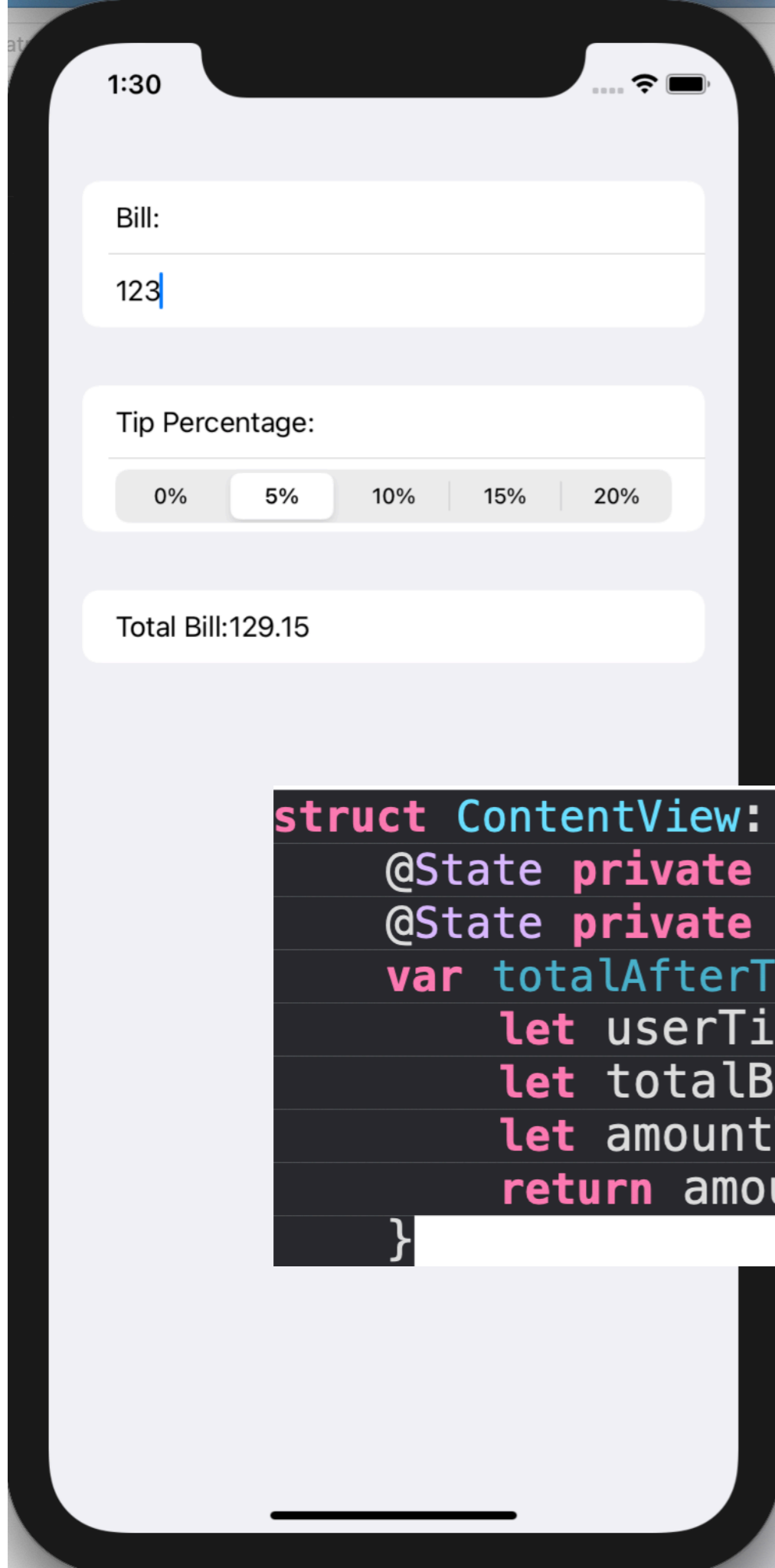
20%

設計變數totalAfterTip的 Return 值

先將tipPercentage轉為Double，再將內容乘5除100

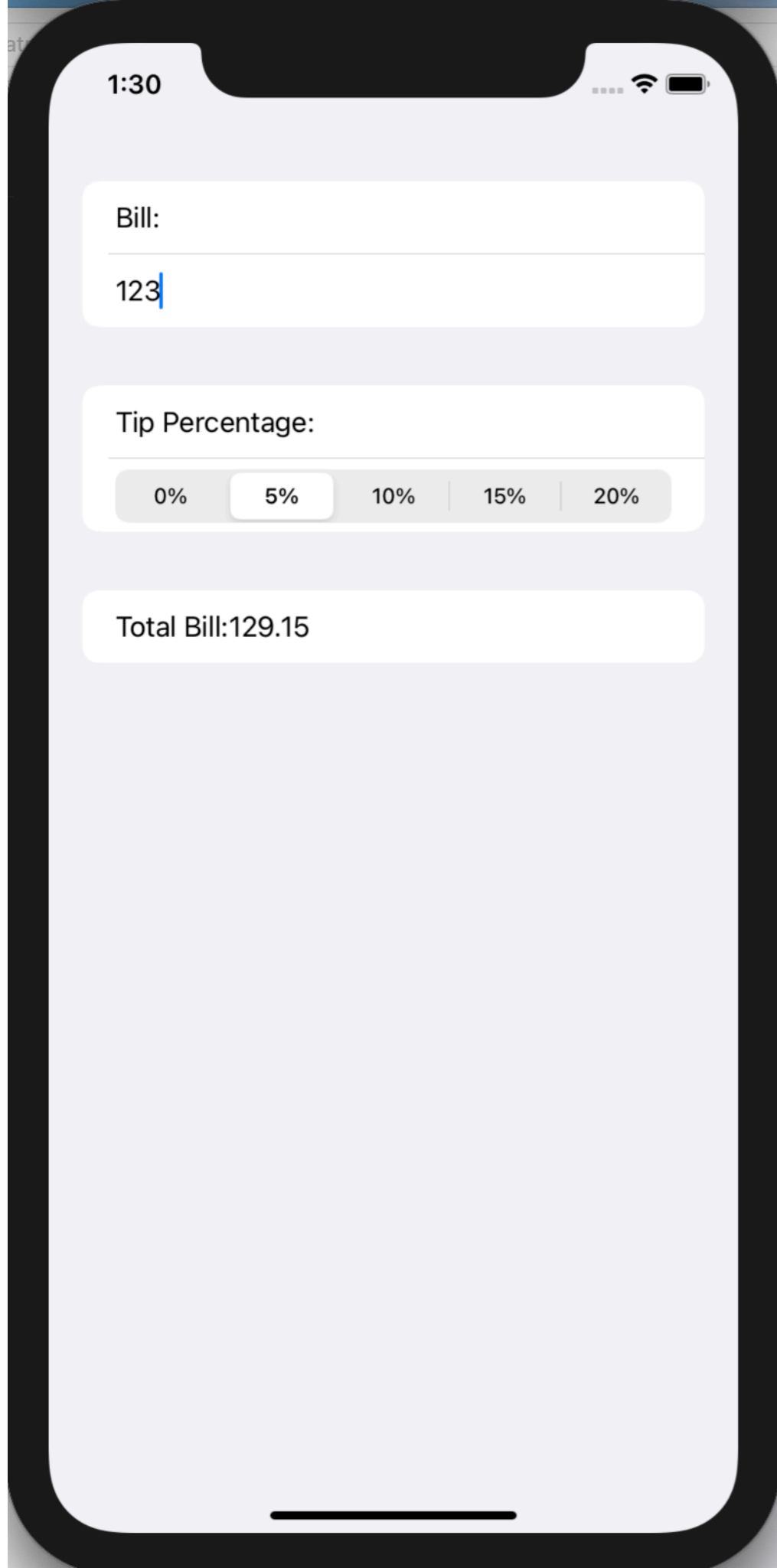
```
var totalAfterTip: Double{  
    let userTipSelection = (Double(tipPercentage) * 5) / 100  
    let totalBill = Double(checkAmount) ?? 0  
    let amountToPay = totalBill + (totalBill * userTipSelection)  
    return amountToPay  
}
```

將應付金額乘以稅率，再加總應付金額



三個變數分別對應到
消費金額輸入
稅制選擇器變動
加稅後的結果輸出

```
struct ContentView: View {  
    @State private var checkAmount = ""  
    @State private var tipPercentage = 1  
    var totalAfterTip: Double {  
        let userTipSelection = (Double(tipPercentage) * 5) / 100  
        let totalBill = Double(checkAmount) ?? 0  
        let amountToPay = totalBill + (totalBill * userTipSele  
        return amountToPay  
    }  
}
```



改變
應付金額輸入
就會改變
總金額

改變
稅率選擇器
也會改變
總金額

```
PlaySound > PlaySound > ContentView.swift > body
1 //
2 // ContentView.swift
3 // PlaySound
4 //
5 // Created by David Razmadze on 5/25/20.
6 // Copyright © 2020 David Razmadze. All rights reserved.
7 //
8
9 import SwiftUI
10
11 struct ContentView: View {
12     var body: some View {
13         VStack {
14             Button(action: {
15                 playSound(key: "sound_effect_1")
16             }) {
17                 Text("Play")
18                     .foregroundColor(Color.green)
19                     .font(.system(size: 32))
20             }
21         }
22     }
23 }
24
25 struct ContentView_Previews: PreviewProvider {
26     static var previews: some View {
27         ContentView()
28     }
29 }
30
```

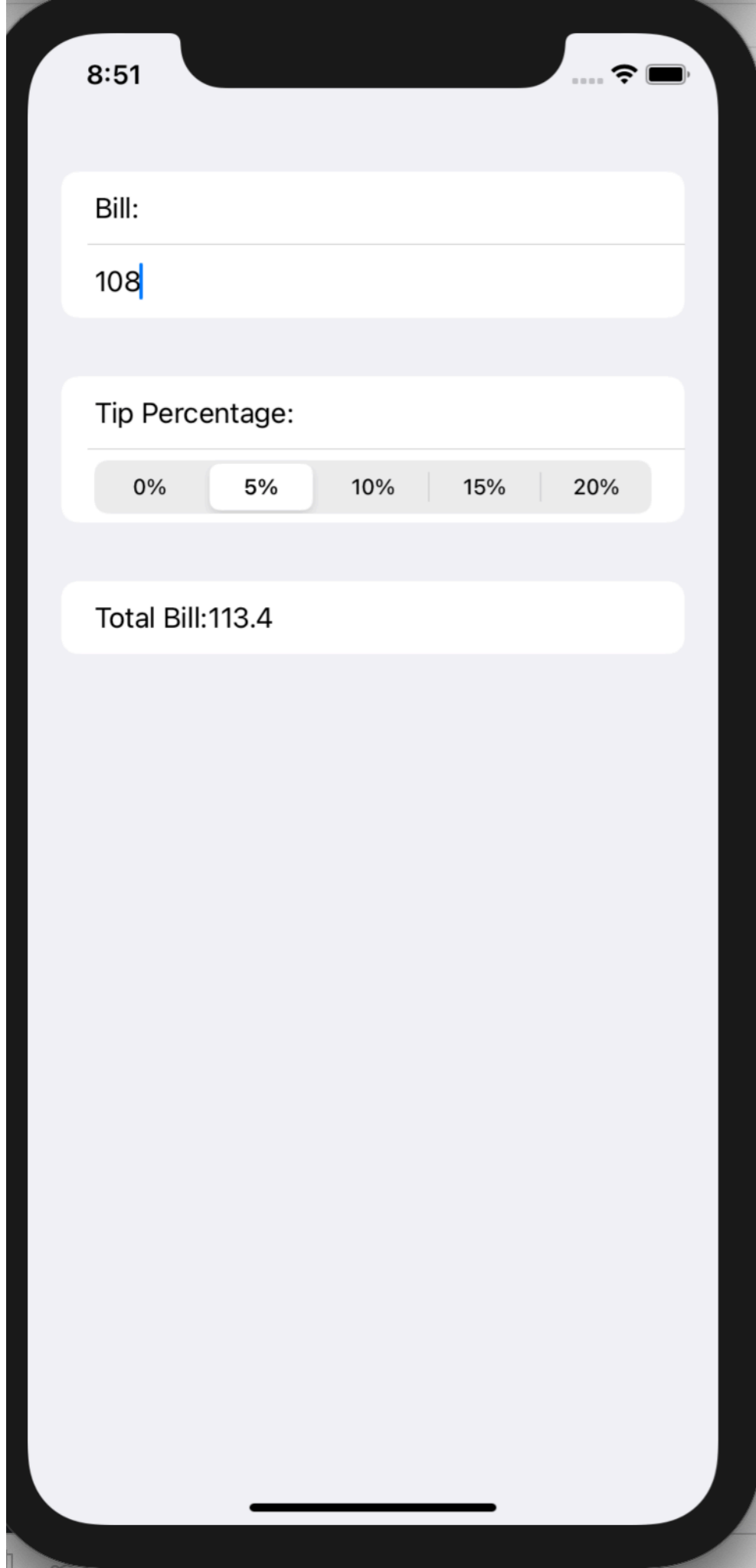


Build Succeeded



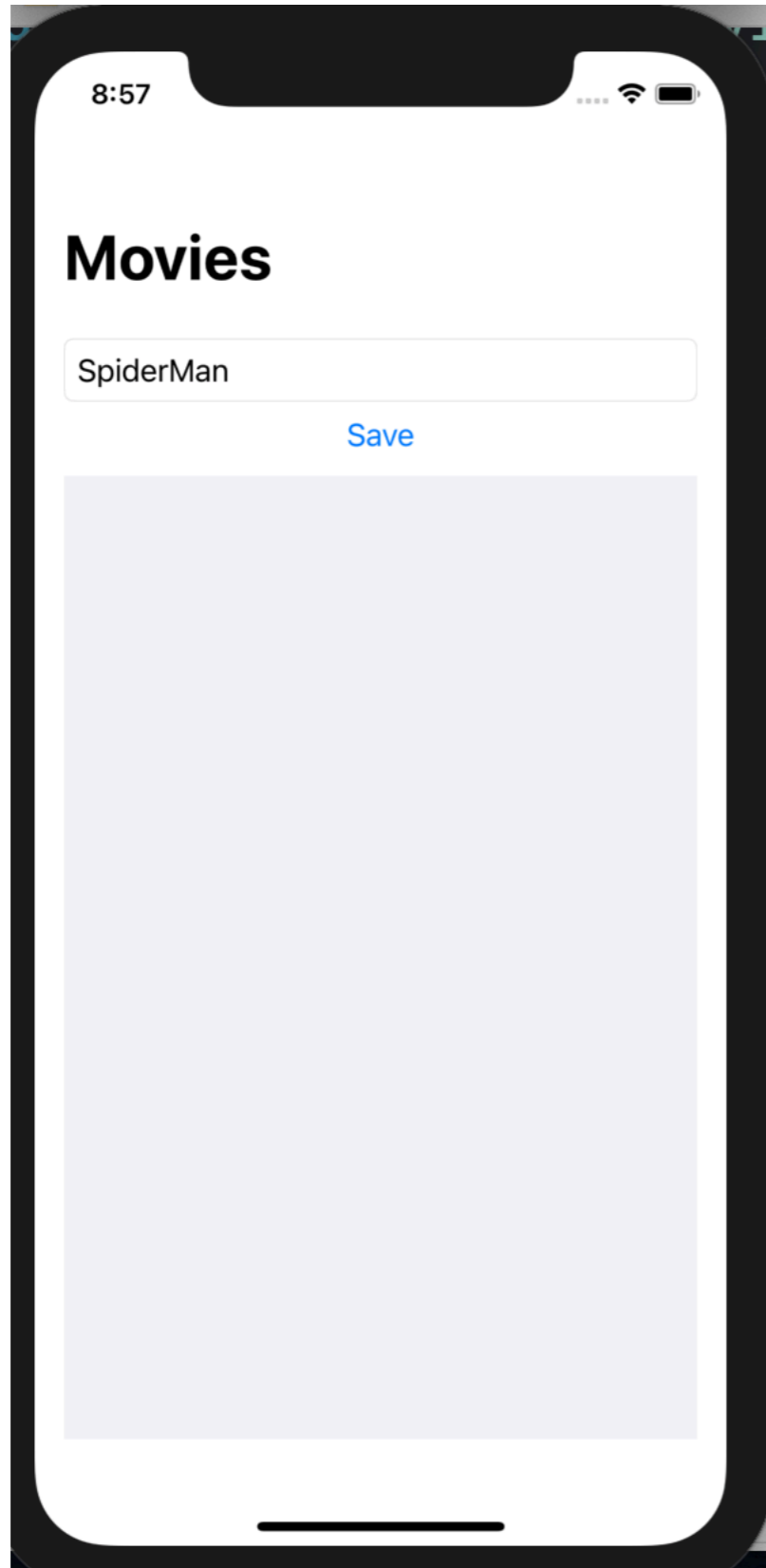
```
1 //
2 // PlaySound.swift
3 // PlaySound
4 //
5 // Created by David Razmadze on 5/25/20.
6 // Copyright © 2020 David Razmadze. All rights reserved.
7 //
8
9 import Foundation
10 import AVFoundation
11
12 var player: AVAudioPlayer!
13
14 func playSound(key: String) {
15     let url = Bundle.main.url(forResource: key, withExtension: "mp3")
16
17     // Do nothing if this url is empty
18     guard url != nil else {
19         return
20     }
21
22     do {
23         player = try AVAudioPlayer(contentsOf: url!)
24         player?.play()
25     } catch {
26         print("error")
27     }
28
29 }
30
```

Core Data

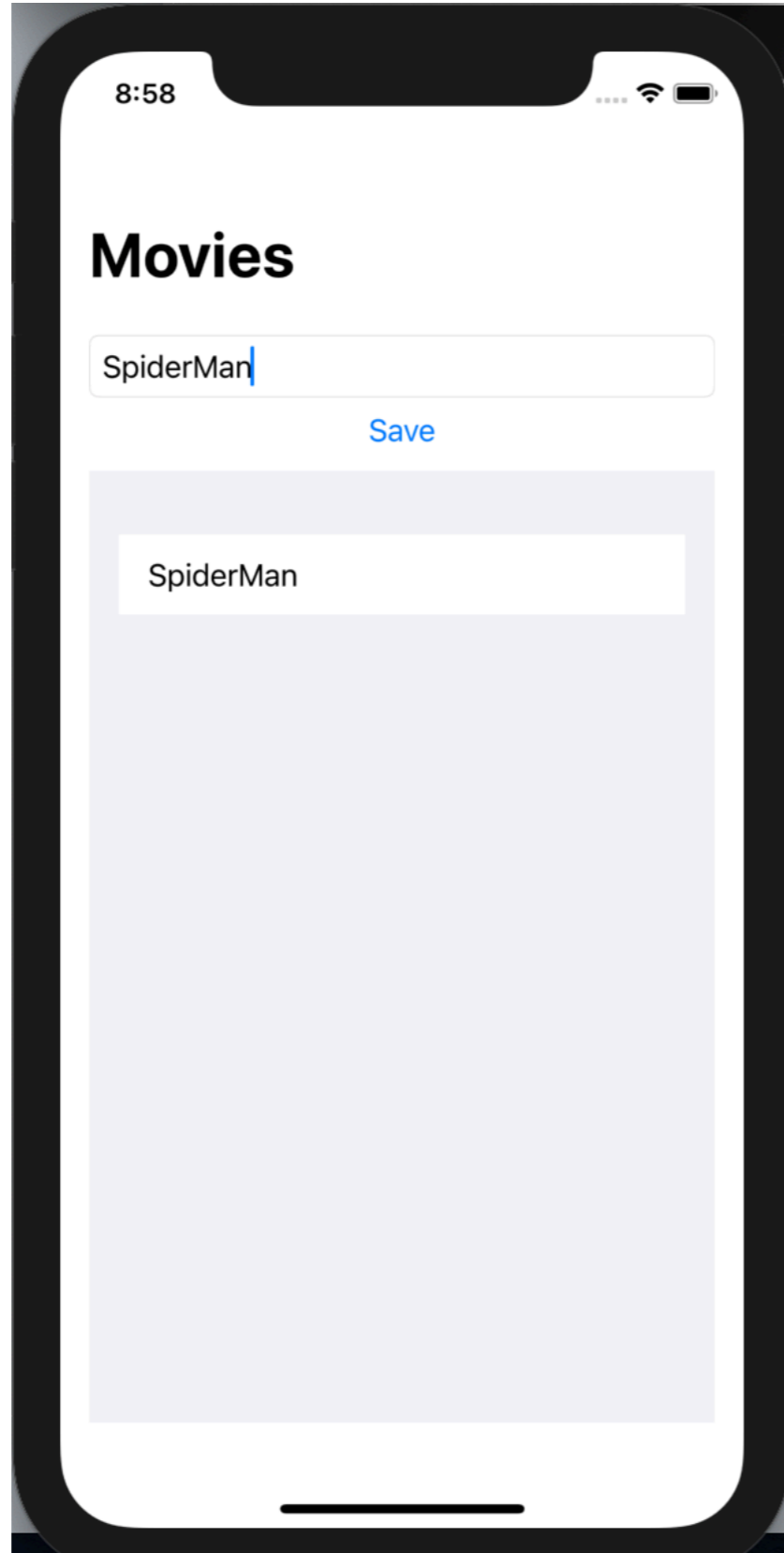


如何將消費總額記錄下來，形成資料紀錄？下次使用時再同樣記錄下來？

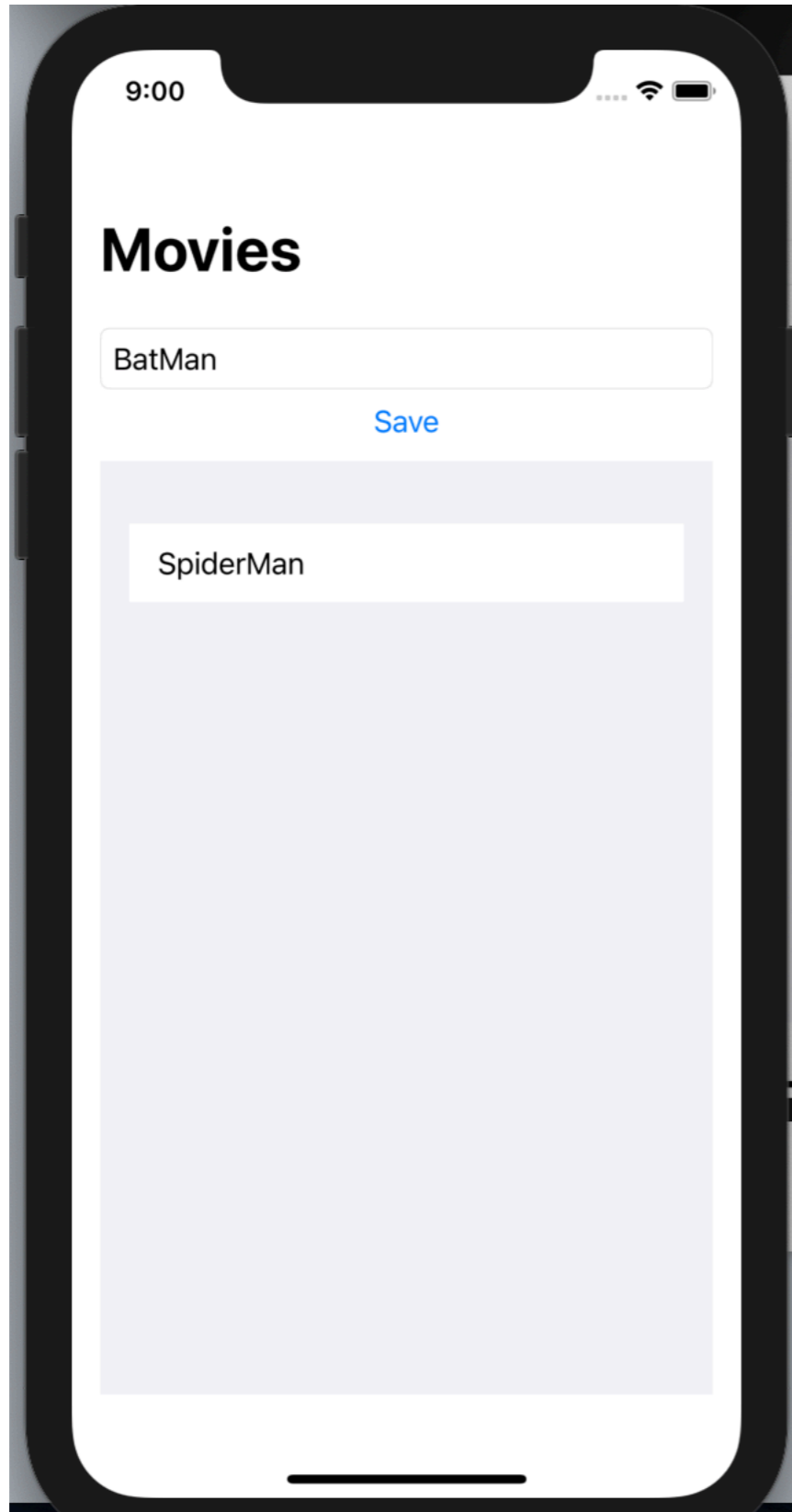
使用Core Data可以建構資料模型，擴充APP的應用功能



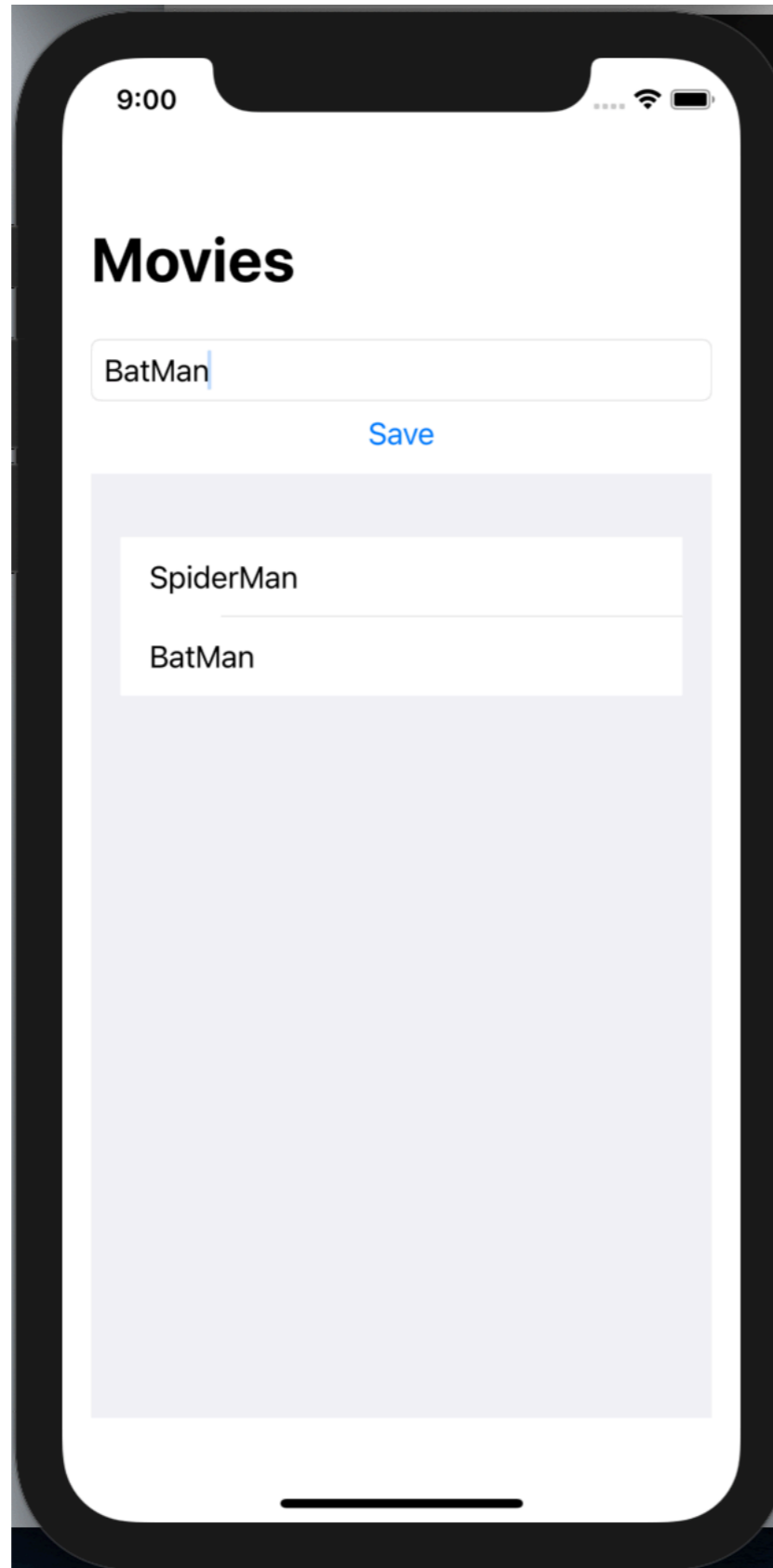
輸入"SpiderMan"



將"SpiderMan"儲存在Core
Data資料模型中



輸入“BatMan”



將“BatMan”儲存在Core Data
資料模型中

Core Data資料模型的刪除功能

9:02



Movies

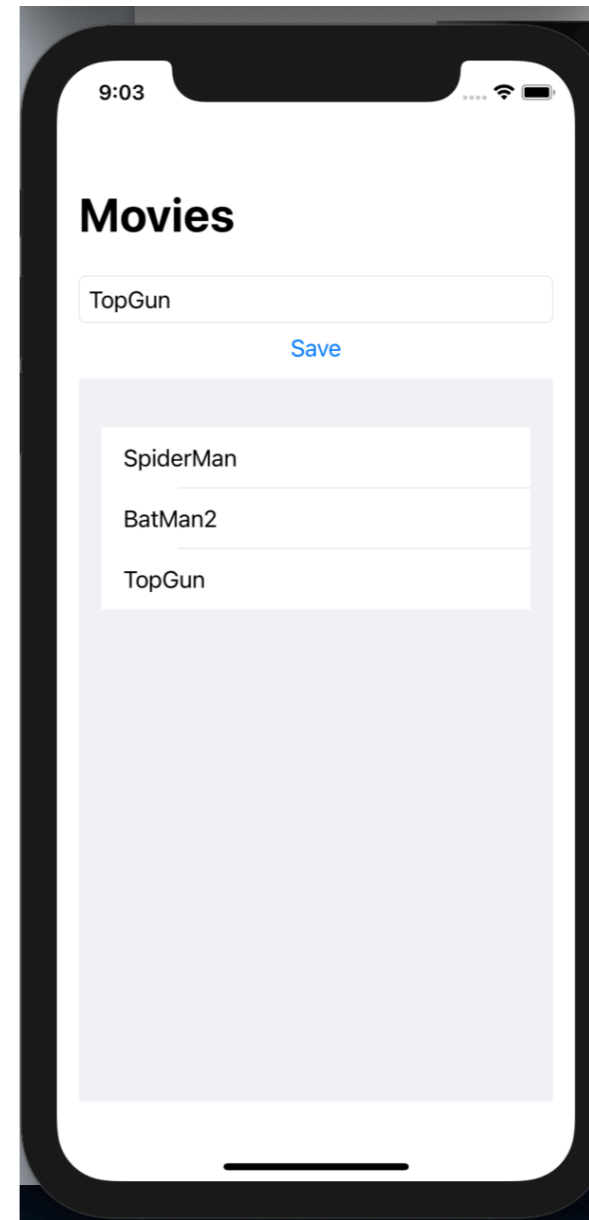
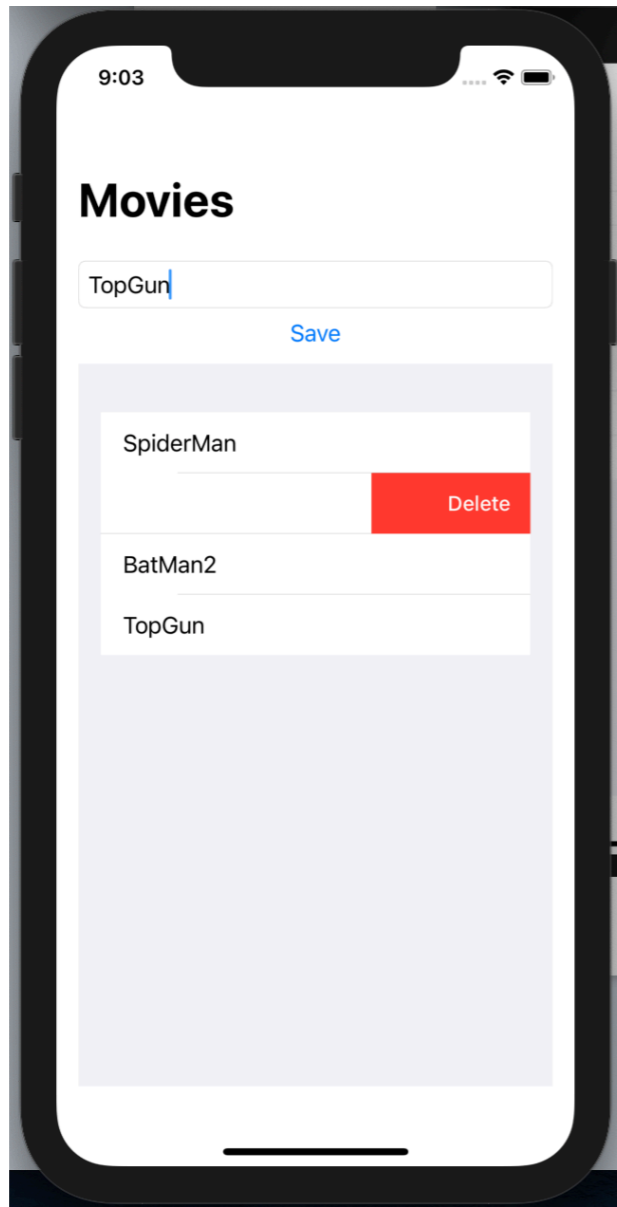
Save

SpiderMan

BatMan

BatMan2

TopGun



Hello CoreData > Hello CoreData > Hello CoreDataModel.xcdatamodeld > Hello CoreDataModel.xcdatamodel > E Movie

ENTITIES

- E Movie

FETCH REQUESTS

CONFIGURATIONS

- C Default

Attributes

Attribute	Type
S title	String

Relationships

Relationship	Destination	Inverse
--------------	-------------	---------

Fetches Properties

Fetches Property	Predicate
------------------	-----------

Outline Style Add Entity

資料模型

欄位與型態