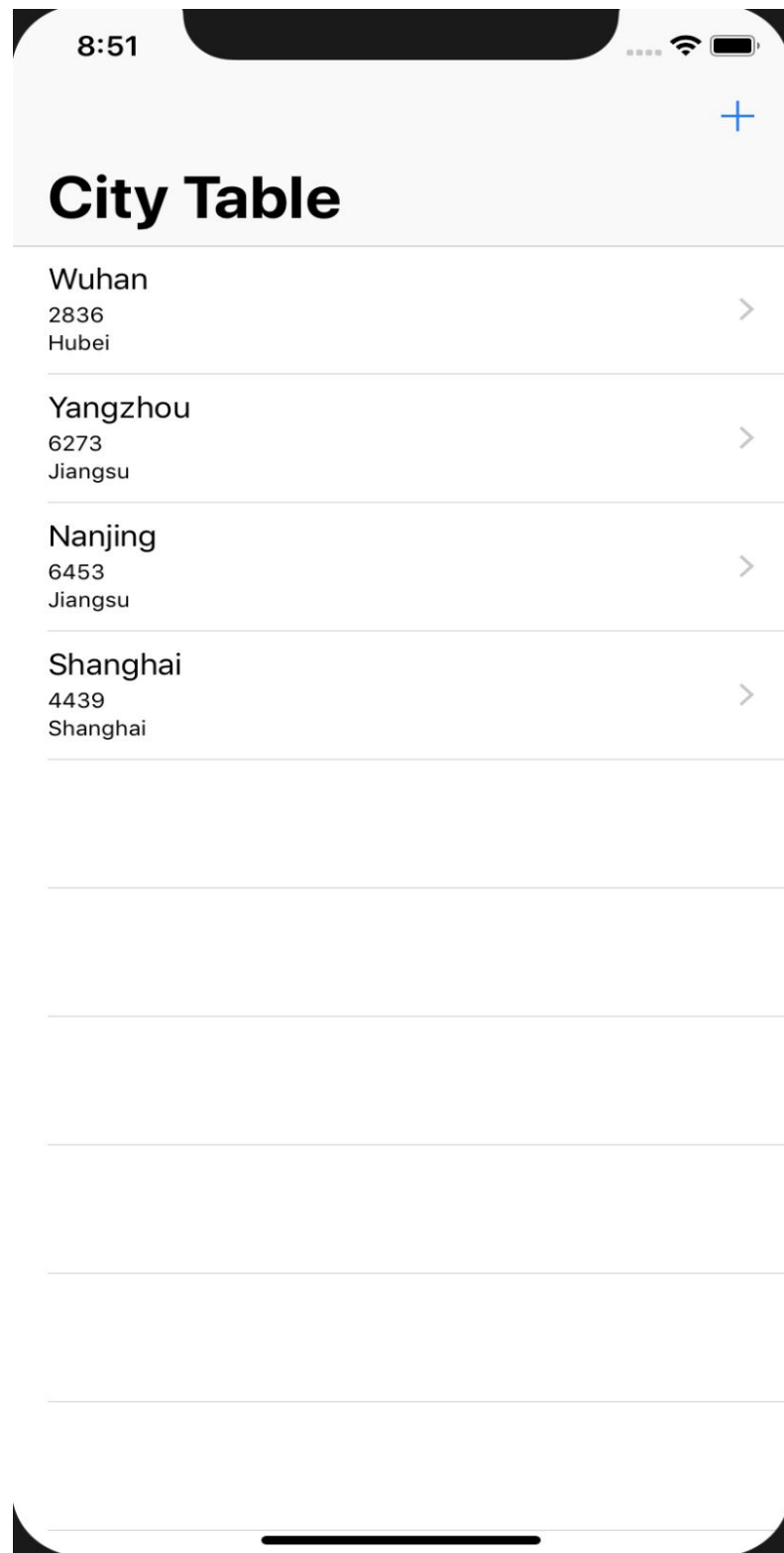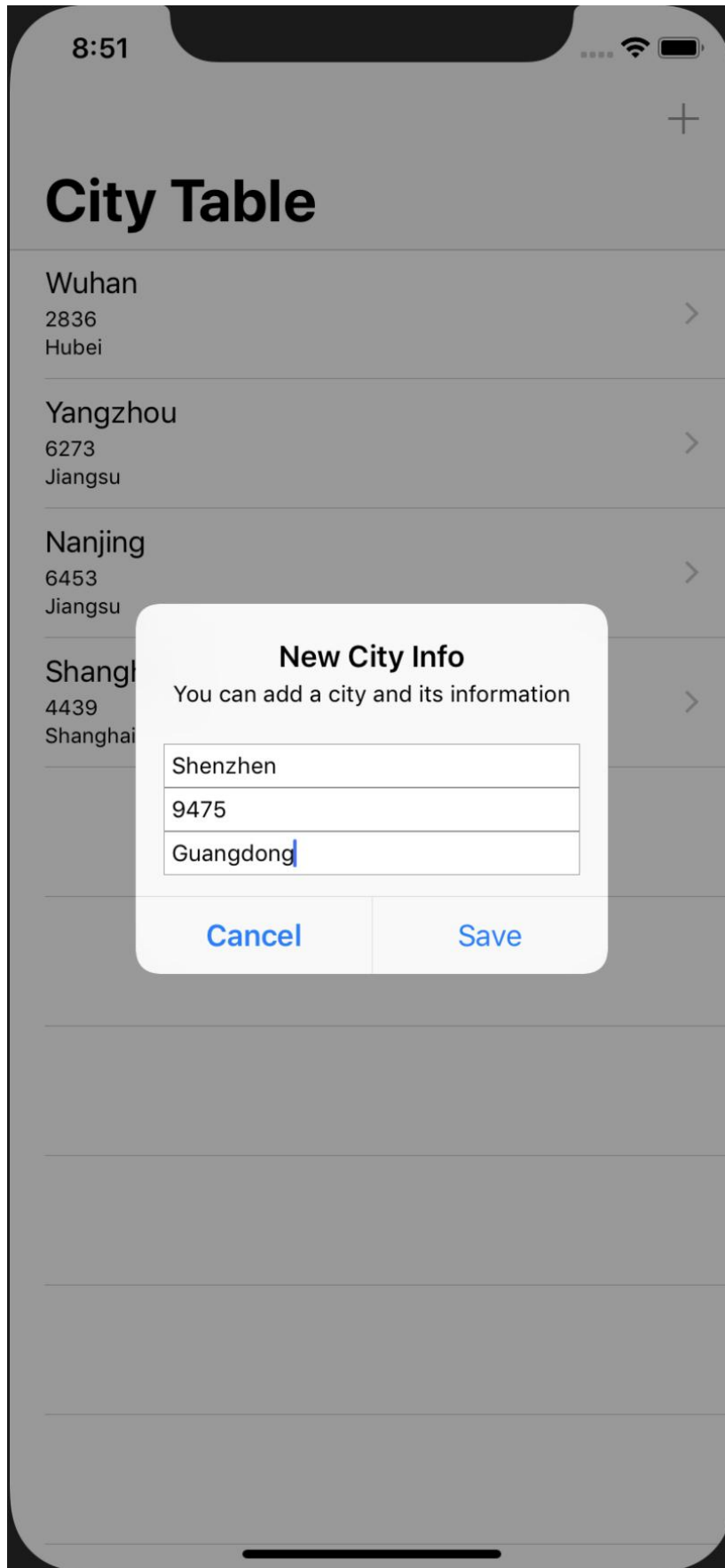Objective-C Assignment3

2016302580127 曹相成 2016 级卓越 1 班

2019.5.18
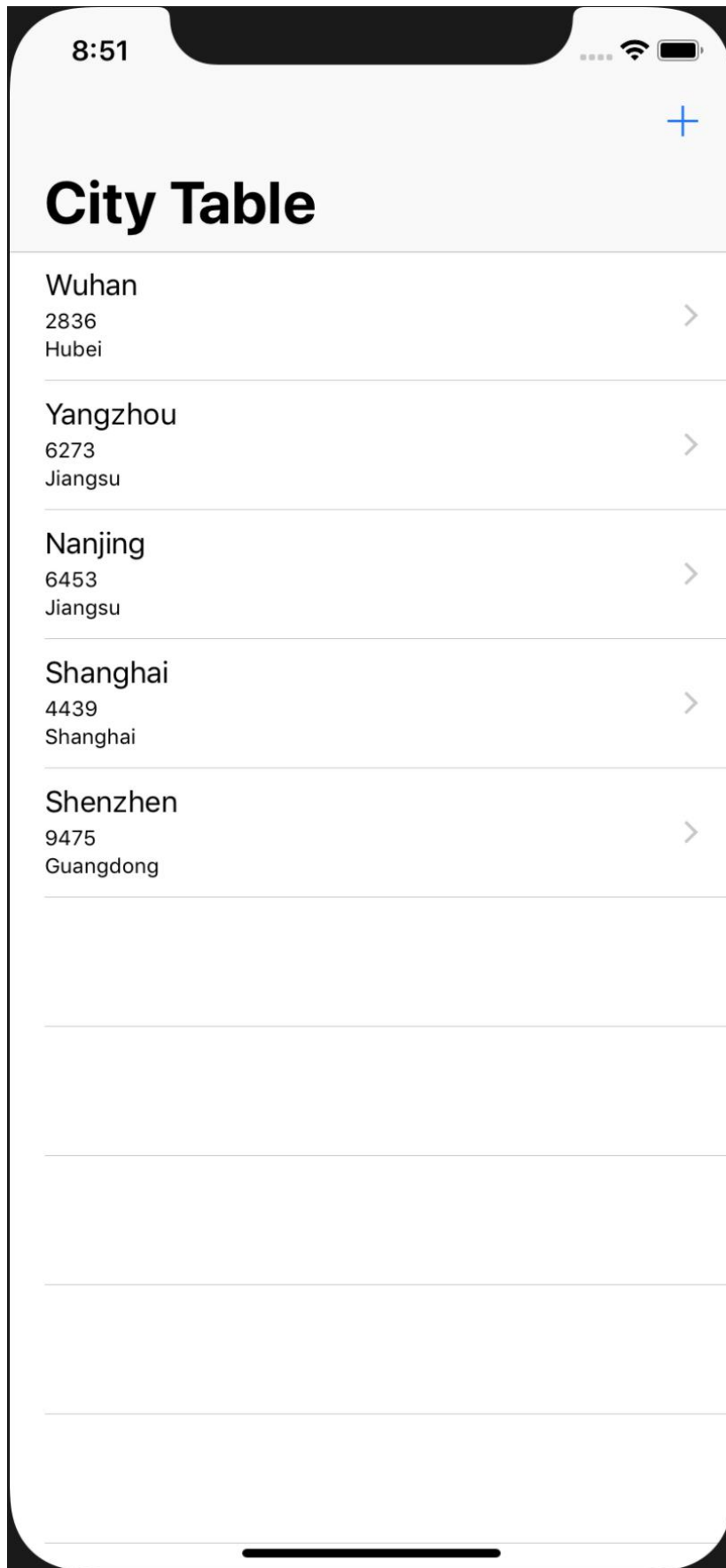
I used an iPhone XR Simulator and the result is as following:

This is the view which I have already added four pieces of data already.

8:51

City Table

Wuhan
2836
Hubei

Yangzhou
6273
Jiangsu

Nanjing
6453
Jiangsu

Shangh
4439
Shanghai

New City Info
You can add a city and its information

Shenzhen

9475

Guangdong

Cancel   Save

This is when I add a new one.

+

City Table

Wuhan
2836
Hubei
>

Yangzhou
6273
Jiangsu
>

Nanjing
6453
Jiangsu
>

Shanghai
4439
Shanghai
>
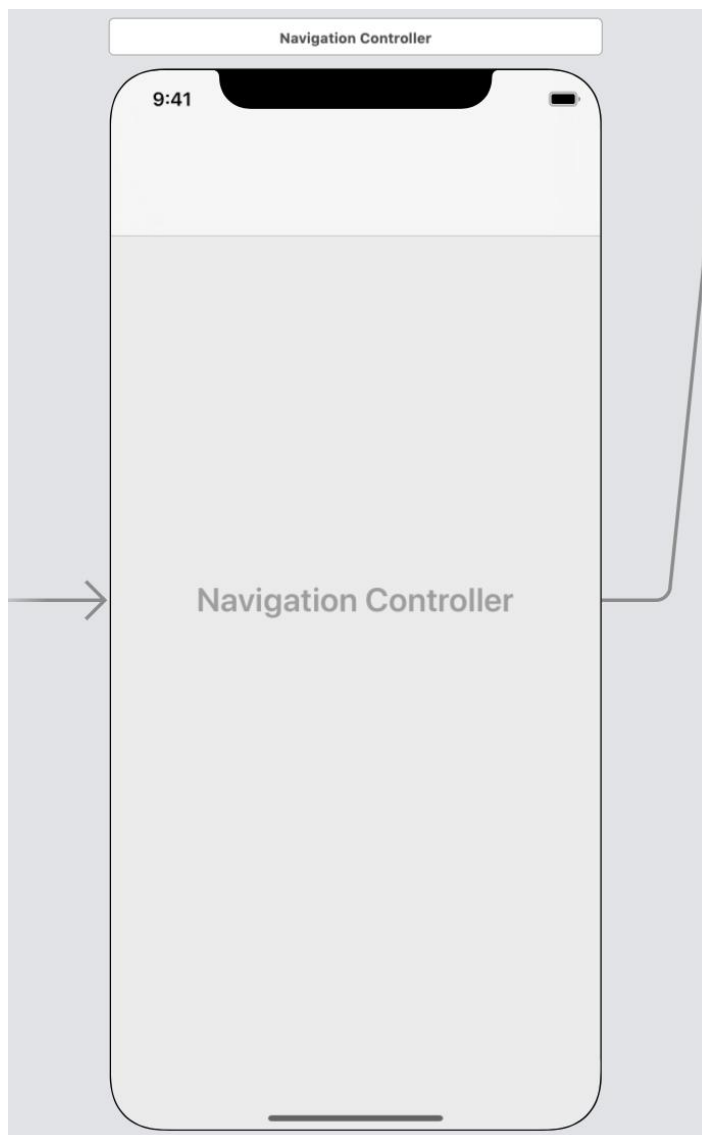
Shenzhen
9475
Guangdong
>

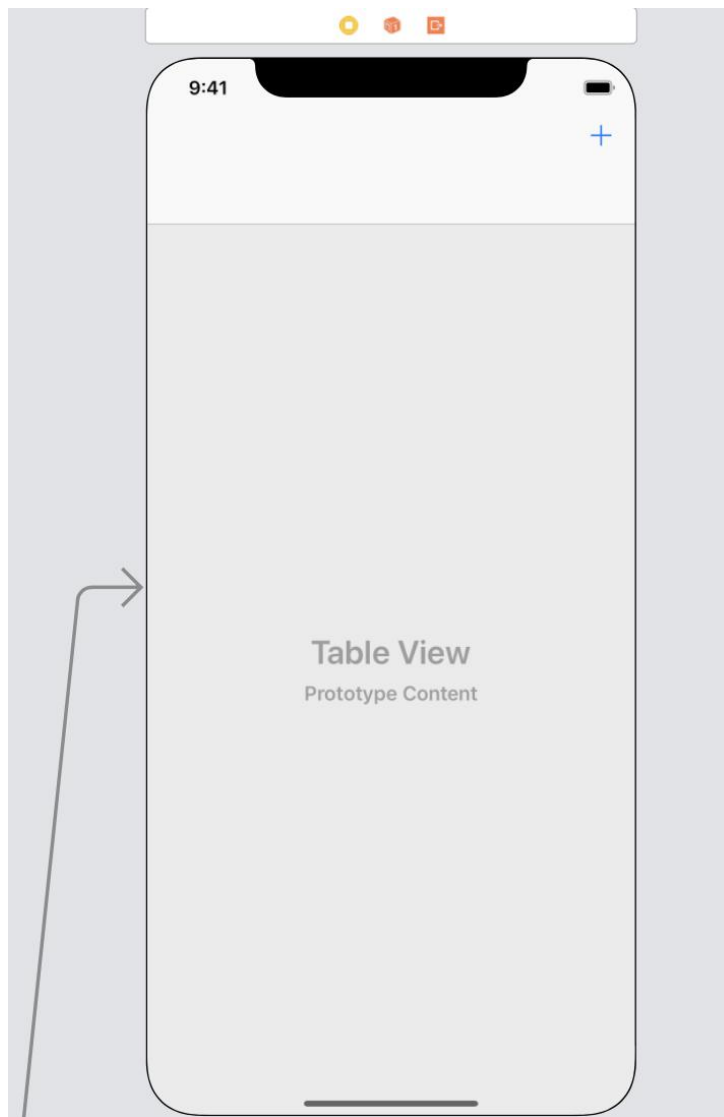This is the result after the operation.

Some Thoughts of the Experiment:

1. This is an experiment use Core Data and tableView to complete a city system to show the name of the city, the postcode and the province. First, we need to create an entity called City which is similar to the class in database.

2. Next, we need to design the table view:

3. Next, just code!

```swift
@IBAction func addName(_ sender: UIBarButtonItem) {
    let alert = UIAlertController(title: "New City Info", message: "You can add a city and its information",
        preferredStyle: .alert)

    alert.addTextField{ (textfield: UITextField) -> Void in
        //textfield.placeholder = "input name"
    }
    // 给alertTwo添加第二个输入框
    alert.addTextField{ (textfield: UITextField) -> Void in
        //textfield.placeholder = "input postCode"
    }



    let saveAction = UIAlertAction(title: "Save", style: .default) {
        [unowned self] action in guard let textField1 = alert.textFields?[0],
                                    let textField2 = alert.textFields?[1],
                                    let textField3 = alert.textFields?[2],
        let nameToSave = textField1.text, let postCodeToSave = textField2.text, let provinceToSave =
            textField3.text else {
            return
        }

        self.save(name: nameToSave, postCode: postCodeToSave, province: provinceToSave)
        self.tableView.reloadData()

    }

    let cancelAction = UIAlertAction(title: "Cancel", style: .cancel)
    alert.addTextField()

    alert.addAction(saveAction)
    alert.addAction(cancelAction)

    self.present(alert, animated: true)
}
```

This function just use to add new data.

```swift
func save(name: String, postCode: String, province: String) {
    guard let appDelegate = UIApplication.shared.delegate as? AppDelegate else {
        return
    }

    let managedContext = appDelegate.persistentContainer.viewContext

    let entity = NSEntityDescription.entity(forEntityName: "City", in: managedContext)!

    let city = NSManagedObject(entity: entity, insertInto: managedContext)

    city.setValue(name, forKeyPath: "name")
    city.setValue(postCode, forKey: "postCode")
    city.setValue(province,forKey:"province")

    do {
        try managedContext.save()
        citys.append(city)
    } catch let error as NSError {
        print("Could not save. \(error), \(error.userInfo)")
    }
}
```

This function just use to save the data entered by users.

```swift
extension ViewController: UITableViewDataSource, UITableViewDelegate {
    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return citys.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
        let city = citys[indexPath.row]
        var cell = tableView.dequeueReusableCell(withIdentifier: "cell")
        if cell == nil {
            cell = UITableViewCell(style: .subtitle, reuseIdentifier: "cell")
            cell?.accessoryType = .disclosureIndicator
        }

        let s2 = String.init(city.value(forKeyPath: "province") as? String ?? "")

        cell?.detailTextLabel?.numberOfLines = 2
        cell?.textLabel?.text = city.value(forKeyPath:"name") as? String
        cell?.detailTextLabel?.text = city.value(forKeyPath: "postCode") as? String
        cell?.detailTextLabel?.text?.append("\n"+s2)

        return cell!
    }
}
```

This function is used to show the tableView. The most interesting and difficult part here is to arrange the textLabel and detailTextLabel well.

Some Questions:

The textplace.placeholder has some problems when I used. So I remarked it. As you can see in the following picture, I will discuss it with teacher in next class.

```swift
@IBAction func addName(_ sender: UIBarButtonItem) {
    let alert = UIAlertController(title: "New City Info", message: "You can add a city and its information",
        preferredStyle: .alert)

    alert.addTextField{ (textfield: UITextField) -> Void in
        //textfield.placeholder = "input name"
    }
    // 给alertTwo添加第二个输入框
    alert.addTextField{ (textfield: UITextField) -> Void in
        //textfield.placeholder = "input postCode"
    }
```

Some References:

1.https://www.raywenderlich.com/7569-getting-started-with-core-data-tutorial