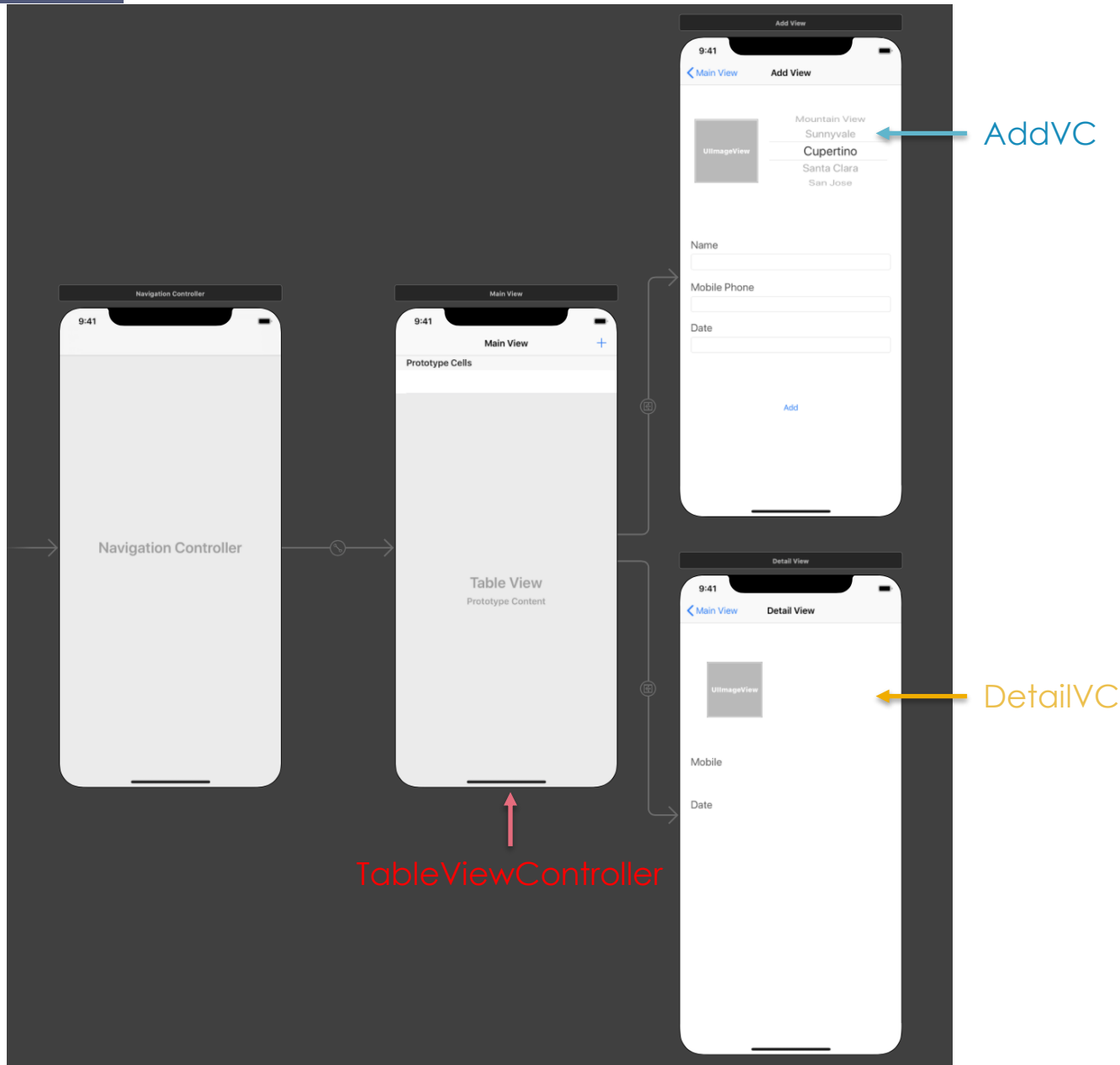


# 테이블 뷰 컨트롤러



20167207 ZHENG JUN

1. 기본 뷰 컨트롤러 화면과 **ViewController.swift** 삭제 2. 테이블 뷰 컨트롤러 새로 추가하기
3. 내비게이션 컨트롤러 추가하기(**is Initial View Controller**체크)
- 4-1 셀과 연결할 **새로운 뷰 컨트롤러**를 생성하기 4-2 셀과 새로운 뷰 컨트롤러를 **연결**하기
- 4-3 셀과 뷰 컨트롤러 연결한 세그웨이의 이름 만들기
- 5-1 버튼과 연결할 **새로운 뷰 컨트롤러** 추가하기. 5-2 버튼과 뷰 컨트롤러 **연결**하기
- 5-3 버튼과 뷰 컨트롤러 연결한 **세그웨이의 이름** 만들기
- 6-1 테이블 뷰 컨트롤러의 화면에 **이름** 만들기 6-2 셀과 연결된 뷰 컨트롤러의 화면에 **이름** 만들기
- 6-3 버튼과 연결된 뷰 컨트롤러의 화면에 **이름** 만들기
7. 테이블 뷰 컨트롤러의 **셀 이름** 만들기 8. 테이블 뷰 컨트롤러화면을 제어하는 **swift 소스 파일** 추가하기
9. 테이블 뷰 컨트롤러 화면과 추가 프로그램을 **연결**하는 방법 10. **Detail View**”화면을 제어하는 **swift 소스 파일** 추가하기
11. **“Detail View**”화면과 추가 프로그램을 **연결**하는 방법 12. 아울렛 변수와 액션함수 추가(**테이블 뷰도 아울렛변수로** 추가)
13. 외부 변수(**제목과 이미지 배열**). 14. 세션의 수와 세션 당 **열의 수** 설정-(함수의 **/\* ...\*/** 주석문제거)
15. 셀의 레이블로 **IndexPath**의 위치에 있는 **문자열** 출력하기
16. 셀의 이미지뷰로 **IndexPath**의 위치에 있는 **이미지** 출력하기. 17. **셀의 삭제**
18. **Delete**를 한글 **“삭제”** 로 변경하기
19. 목록의 **순서 바꾸기**
20. 목록의 **셀** 추가하기-**viewWillAppear()** - **tableViewMemo.reloadData()**
21. 셀을 선택하면 그 **내용** 보여주기



1. 테이블 뷰 컨트롤러에서 셀의 정보를 볼 수 있고 셀을 왼쪽으로 밀거나 Edit 버튼을 눌러 셀을 삭제가 가능하고 추가로 Edit를 클릭해서 목록순서를 바꿀 수 있다
2. AddVC에서 피커 뷰로 추가한 연락처의 이미지를 선택 할 수 있고 이름, 번호, 등록일을 추가 가능하다.
3. DetailVC는 Main View에서 선택한 셀의 세부 내용(이미지, 이름, 번호, 등록일)을 보여준다.

## TableViewController

```
var name = ["홍길동", "김길동", "이길동"]
var mobile = ["010-1111-2222", "010-3333-4444", "010-5555-6666"]
var date = ["2019-11-10", "2019-11-13", "2019-11-23"]
var itemsImageFile = ["1.png", "2.png", "3.png"]
class TableViewController: UITableViewController {
|
|
|   @IBOutlet var tvListView: UITableView!
|   override func viewDidLoad() {
|       super.viewDidLoad()
|       // 왼쪽 edit 바 버튼 추가해서 설정할 수 있음
|       self.navigationItem.leftBarButtonItem = self.editButtonItem
|   }
|   // 테이블당 섹션 하나
|   override func numberOfSections(in tableView: UITableView) -> Int {
|       return 1
|   }
|   // 행수
|   override func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
|       // #warning Incomplete implementation, return the number of rows
|       return name.count
|   }
|
|   // 이름과 이미지를 셀에 삽입
|   override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
|       let cell = tableView.dequeueReusableCell(withIdentifier: "myCell", for: indexPath)
|
|       cell.textLabel?.text = name[(indexPath as NSIndexPath).row]
|       cell.imageView?.image = UIImage(named: itemsImageFile[(indexPath as NSIndexPath).row])
|
|       return cell
|   }
}
```

## TableViewController

```

// 목록 삭제 함수
override func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {
    if editingStyle == .delete {
        // 삭제
        name.remove(at: (indexPath as IndexPath).row)
        mobile.remove(at: (indexPath as IndexPath).row)
        date.remove(at: (indexPath as IndexPath).row)
        itemsImageFile.remove(at: (indexPath as IndexPath).row)
        tableView.deleteRows(at: [indexPath], with: .fade)
    } else if editingStyle == .insert {
    }
}

// "delete" 를 "삭제" 로 변경
override func tableView(_ tableView: UITableView, titleForDeleteConfirmationButtonForRowAt indexPath: IndexPath) -> String? {
    return "삭제"
}

// 목록 순서 바꾸기
override func tableView(_ tableView: UITableView, moveRowAt fromIndexPath: IndexPath, to: IndexPath) {
    // 이동할 아이템의 정보 저장
    let nameToMove = name[(fromIndexPath as NSIndexPath).row]
    let mobileToMove = mobile[(fromIndexPath as NSIndexPath).row]
    let dateToMove = date[(fromIndexPath as NSIndexPath).row]
    let itemImageToMove = itemsImageFile[(fromIndexPath as NSIndexPath).row]
    // 이동할 아이템 삭제
    name.remove(at: (fromIndexPath as NSIndexPath).row)
    mobile.remove(at: (fromIndexPath as NSIndexPath).row)
    date.remove(at: (fromIndexPath as NSIndexPath).row)
    itemsImageFile.remove(at: (fromIndexPath as NSIndexPath).row)
    // 삭제 된 아이템을 이동할 위치로 삽입
    name.insert(nameToMove, at: (to as NSIndexPath).row)
    mobile.insert(mobileToMove, at: (to as NSIndexPath).row)
    date.insert(dateToMove, at: (to as NSIndexPath).row)
    itemsImageFile.insert(itemImageToMove, at: (to as NSIndexPath).row)
}

```

## TableViewController

```
// Main View 로 돌아올 때 호출하는 함수
override func viewWillAppear(_ animated: Bool) {
    // 테이블 뷰 다시 불러오는 함수
    tableView.reloadData()
}

// Main View --> DetailVC 로 값 전달
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    if segue.identifier == "sgDetail" {
        let cell = sender as! UITableViewCell
        let indexPath = self.tableView.indexPath(for: cell)
        let detailView = segue.destination as! DetailVC
        // DetailVC의 함수 사용
        detailView.receiveItem(name: name[((indexPath as NSIndexPath)?.row)!], mobile: mobile[((indexPath as NSIndexPath)?.row)!], date: date[((indexPath as NSIndexPath)?.row)!], image: itemsImageFile[((indexPath as NSIndexPath)?.row)!])
    }
}
```

# 01 테이블 뷰 컨트롤러

## AddVC

```
class AddVC: UIViewController UIPickerViewDelegate, UIPickerViewDataSource {
    // 초기화
    let MAX_ARRAY_NUM = 3
    let PICKER_VIEW_COLUMN = 1
    let PICKER_VIEW_HEIGHT:CGFloat = 40
    var imageArray = [UIImage?]()
    var imageFileName = [ "1.png", "2.png", "3.png" ]
    var fileName = ""
```

```
@IBAction func btnAddItem(_ sender: Any) {
    // 배열에 추가
    name.append(nameTF.text!)
    mobile.append(mobileTF.text!)
    date.append(DateTF.text!)
    itemsImageFile.append(fileName)
    // 클릭하고 변수 초기화
    nameTF.text=""
    mobileTF.text=""
    DateTF.text=""
    _ = navigationController?.popViewController(animated: true)
}
```

```
// 컴포넌트 수 설정
func numberOfComponents(in pickerView: UIPickerView) -> Int {
    return PICKER_VIEW_COLUMN
}

// 컴포넌트 높이 설정
func pickerView(_ pickerView: UIPickerView, heightForComponent component: Int) -> CGFloat {
    return PICKER_VIEW_HEIGHT
}

// 열의 개수 설정
func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {
    return imageFileName.count
}

// 이미지 뷰의 프레임 크기 변경 --> 컴포넌트의 각 열에 이미지 뷰를 리턴
func pickerView(_ pickerView: UIPickerView, viewForRow row: Int, forComponent component: Int, reusing view: UIView?) -> UIView {
    let imageView = UIImageView(image:imageArray[row])
    imageView.frame = CGRect(x: 0, y: 0, width: 40, height: 40)
    return imageView
}

// 피커 뷰가 선택 될때 실행
func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int, inComponent component: Int) {
    imageView.image = imageArray[row]
    fileName = imageFileName[row]
}
```

```
override func viewDidLoad() {
    super.viewDidLoad()
    // 이미지 배열
    for i in 0 ..< MAX_ARRAY_NUM {
        let image = UIImage(named: imageFileName[i])
        imageArray.append(image!)
    }
    imageView.image = imageArray[0]
    fileName = imageFileName[0]

    // 현재 시간 --> 년 월 일
    let d = NSDate()
    let dateForm = DateFormatter()
    dateForm.dateFormat = "yyyy-MM-dd"
    DateTF.text = dateForm.string(from: d as Date)
}
```

```
import UIKit

class DetailVC: UIViewController {

    var name1 = ""
    var mobile1 = ""
    var date1 = ""

    @IBOutlet var name1bl: UILabel!
    @IBOutlet var mobile1bl: UILabel!
    @IBOutlet var date1bl: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.
        name1bl.text = name1
        mobile1bl.text = mobile1
        date1bl.text = date1
    }

    func reciveItem(name: String, mobile: String, date: String)
    {
        name1 = name
        mobile1 = mobile
        date1 = date
    }
}
```



## 01

