

BoostCamp 1 주차 발표

박현수 (ios)

목차 (contents)

1. Autolayout 구성 노하우
2. 자주 사용하는 앱의 MVC 패턴에 따른 클래스 구성 상상
3. 네비게이션 컨트롤러 동작양식
4. 네비게이션 컨트롤러가 아닌 다른 화면전환방법
5. 프로젝트

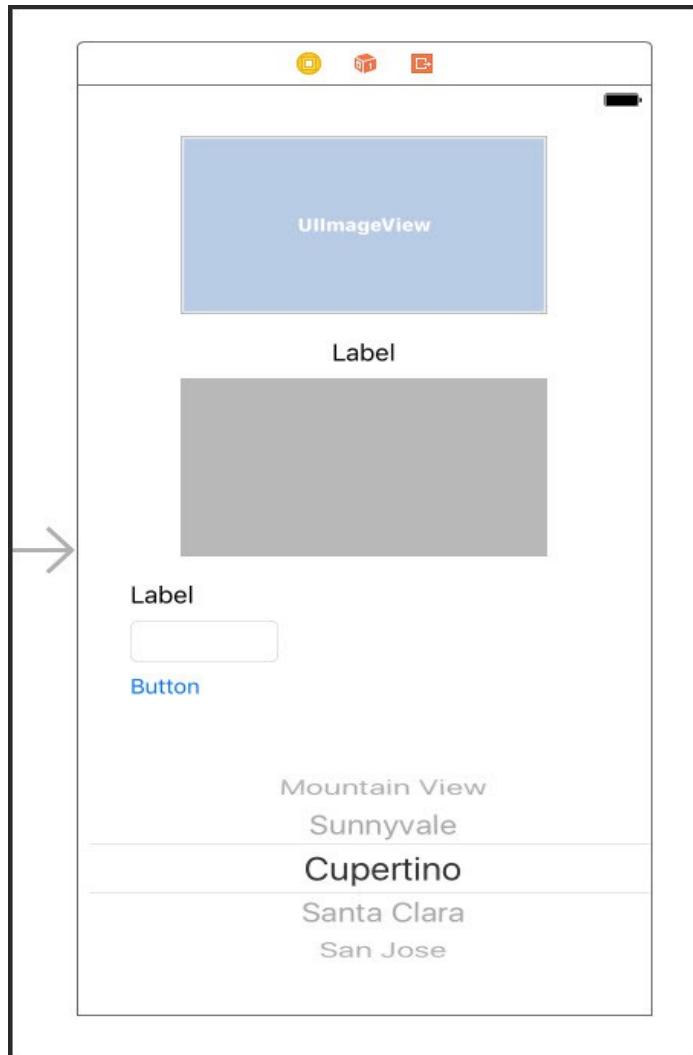
AutoLayout

목표 : Apple 기기의 다양한 크기의 화면에서도 사용자가 제대로 볼 수 있도록 디자인해야함 .

AutoLayout

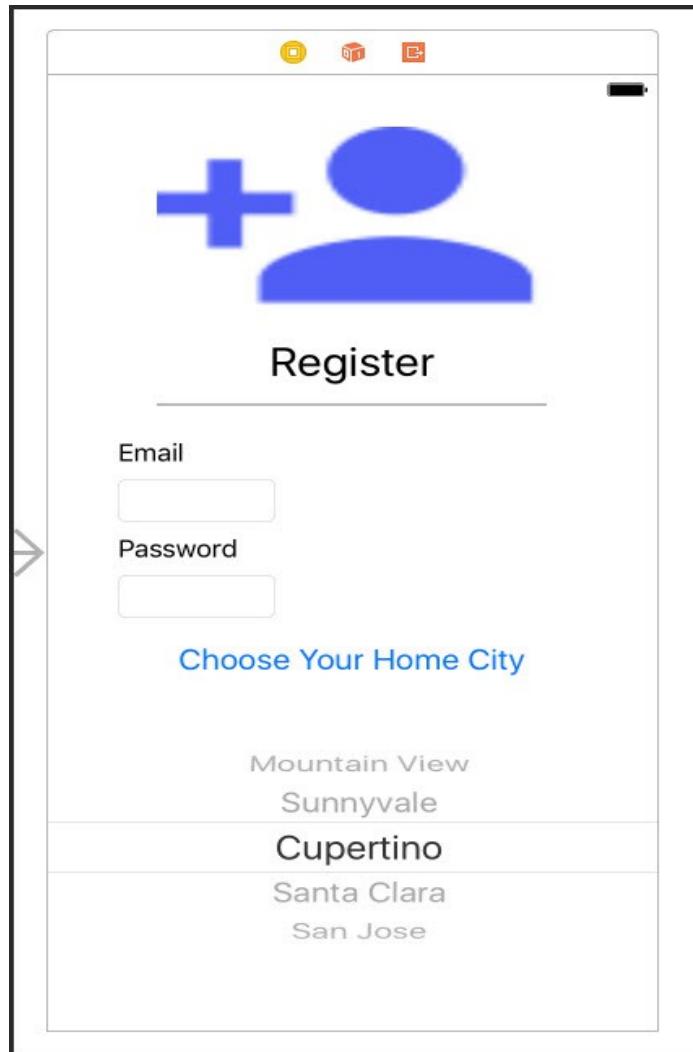
Autolayout 의 기본 ! Constraint

AutoLayout



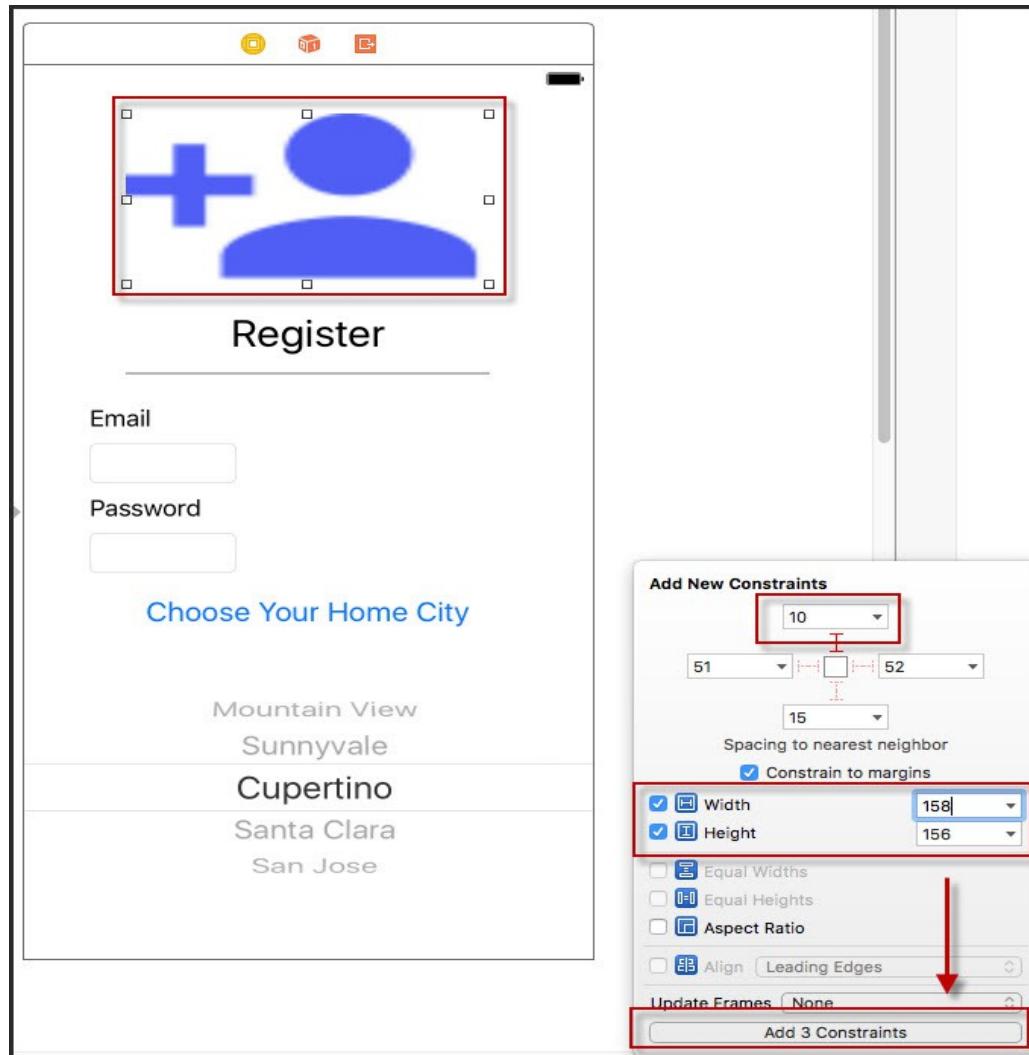
1. 사용자 등록 app
(의존적인 constraint 설정)

AutoLayout



UI component 네이밍 및 기본 설정

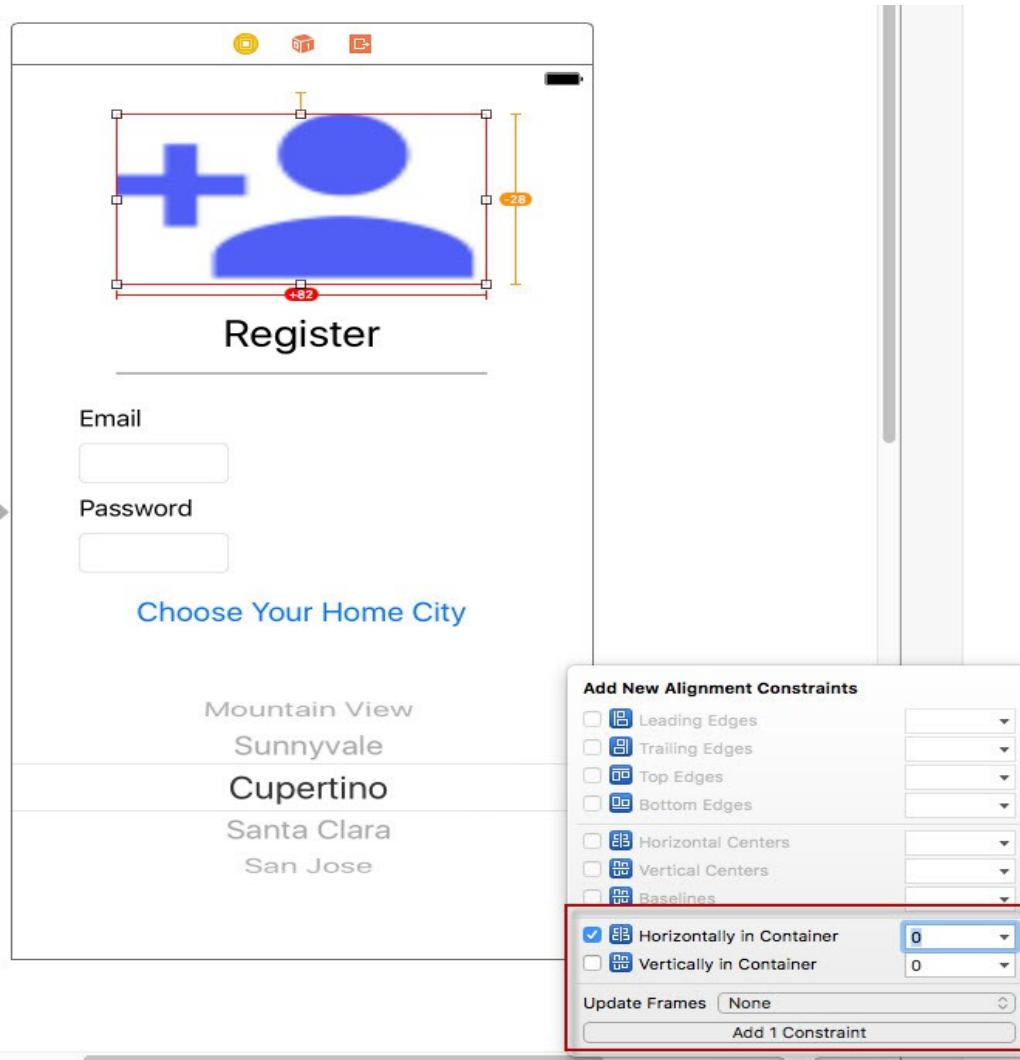
AutoLayout



가장 위에 위치할 이미지 설정

1. top - 10
2. 너비 , 높이 고정

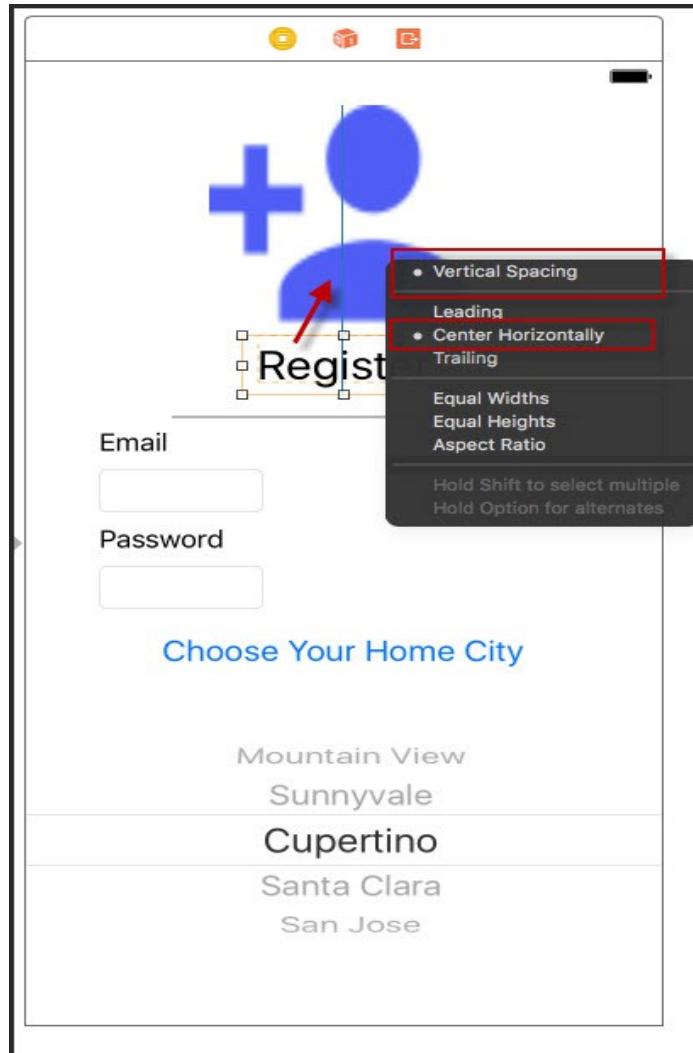
AutoLayout



가장 위에 위치할 이미지 설정

3. 컨테이너에 수평으로 가운데 정렬

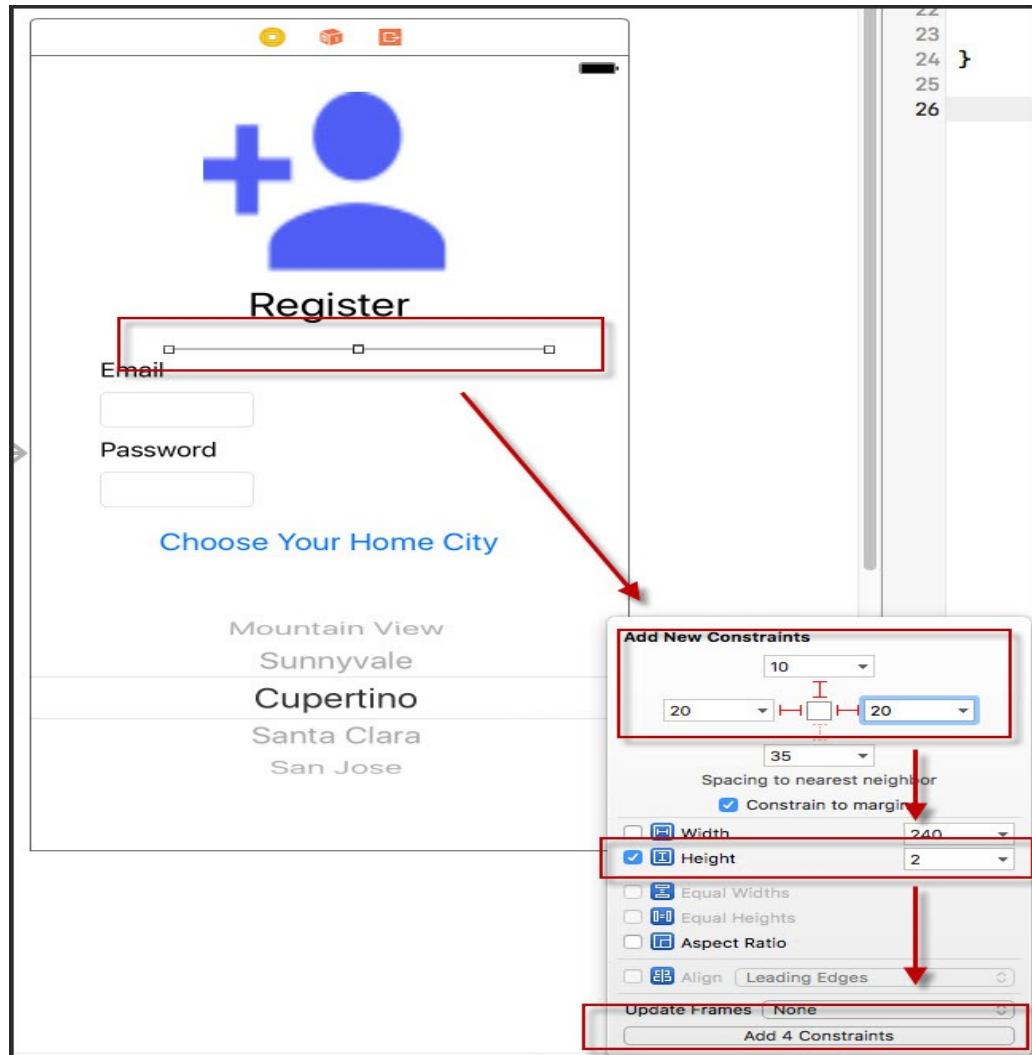
AutoLayout



헤더 제목 (Register) 레이블 설정

1. 위의 이미지와 고정된 spacing
2. 컨테이너에 수평으로 가운데 정렬

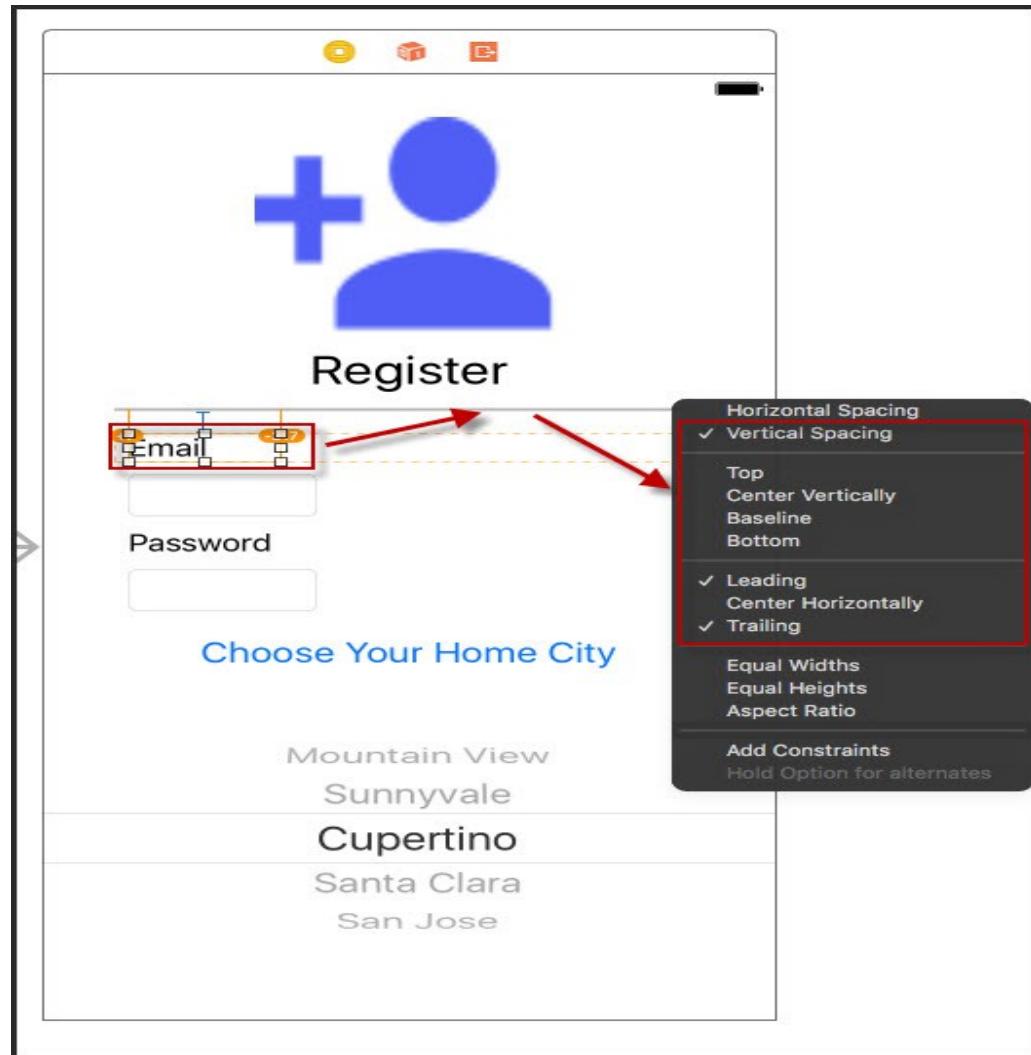
AutoLayout



나눔선 (divider) 설정

1. Register 와 10 만큼 간격
2. 양 사이드와 20 정도 간격
3. 고정된 높이

AutoLayout

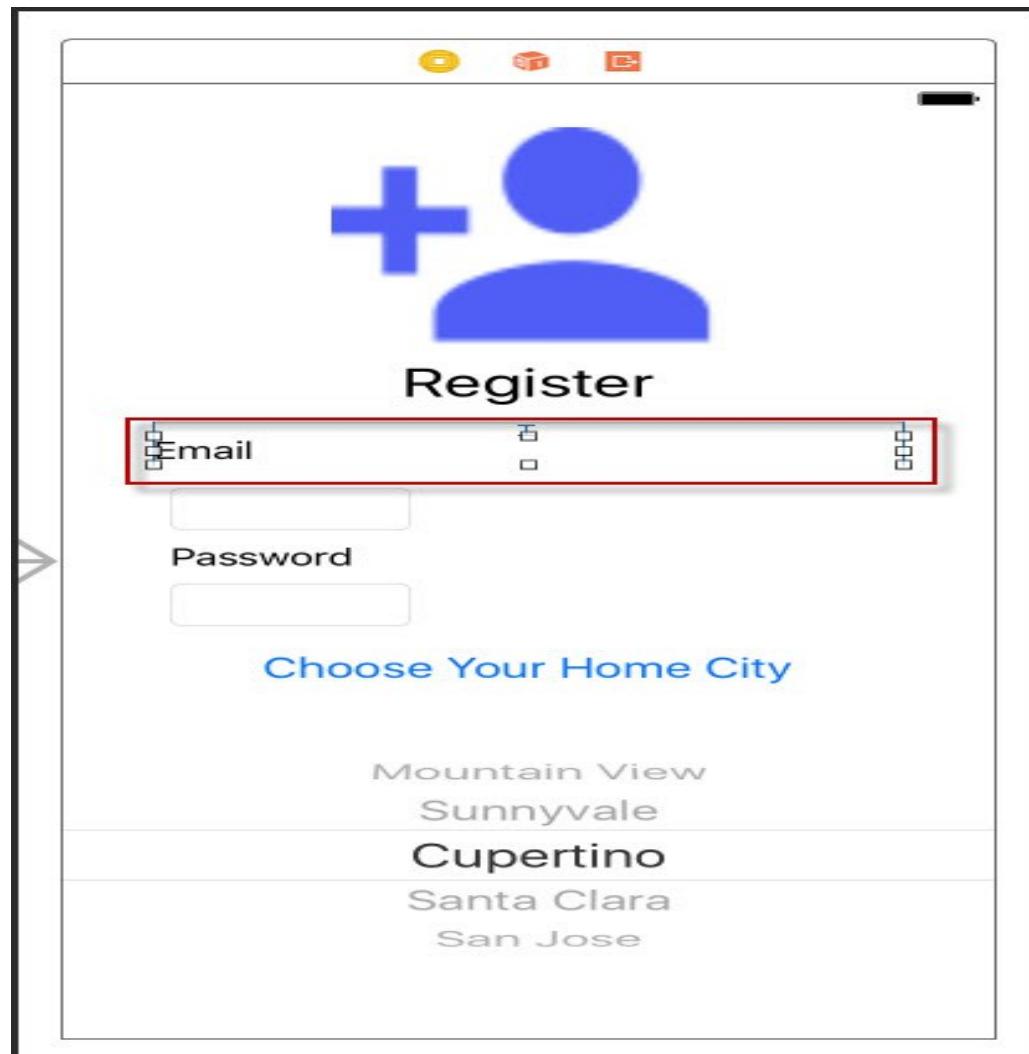


Email 레이블 설정

1. 나눔선과 8 만큼 간격
2. 나눔선과 Leading
3. 나눔선과 Trailing

* Leading 과 Trailing 으로 대상과 너비와 좌우 위치를 동일하게 할 수 있음

AutoLayout

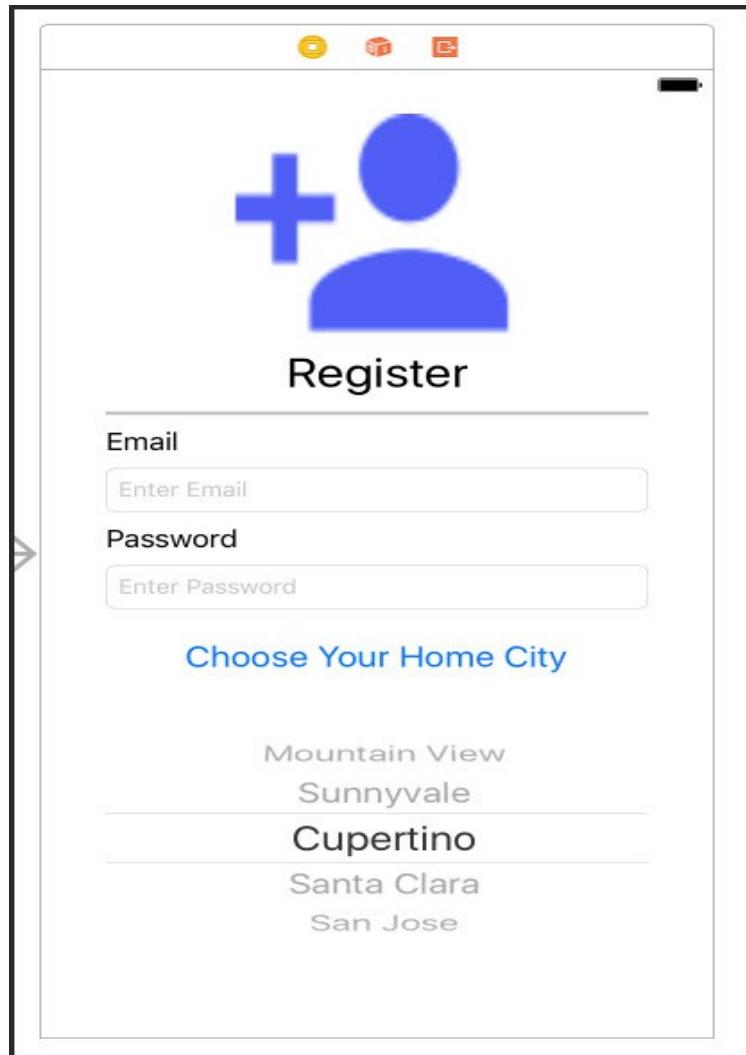


Email 레이블 설정

1. 나눔선과 8 만큼 간격
2. 나눔선과 Leading
3. 나눔선과 Trailing

* Leading 과 Trailing 으로 대상과 너비와 좌우 위치를 동일하게 할 수 있음

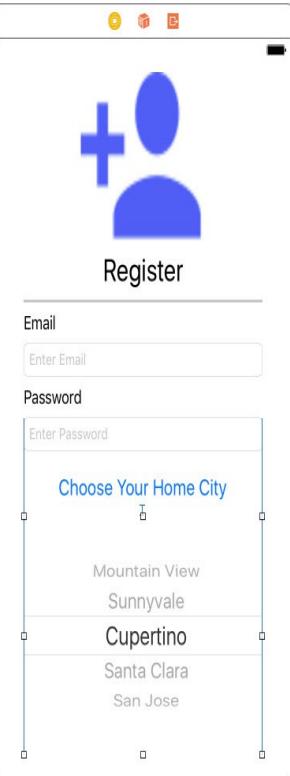
AutoLayout



나머지 레이블 및 텍스트필트 설정

Email 과 동일하게

AutoLayout



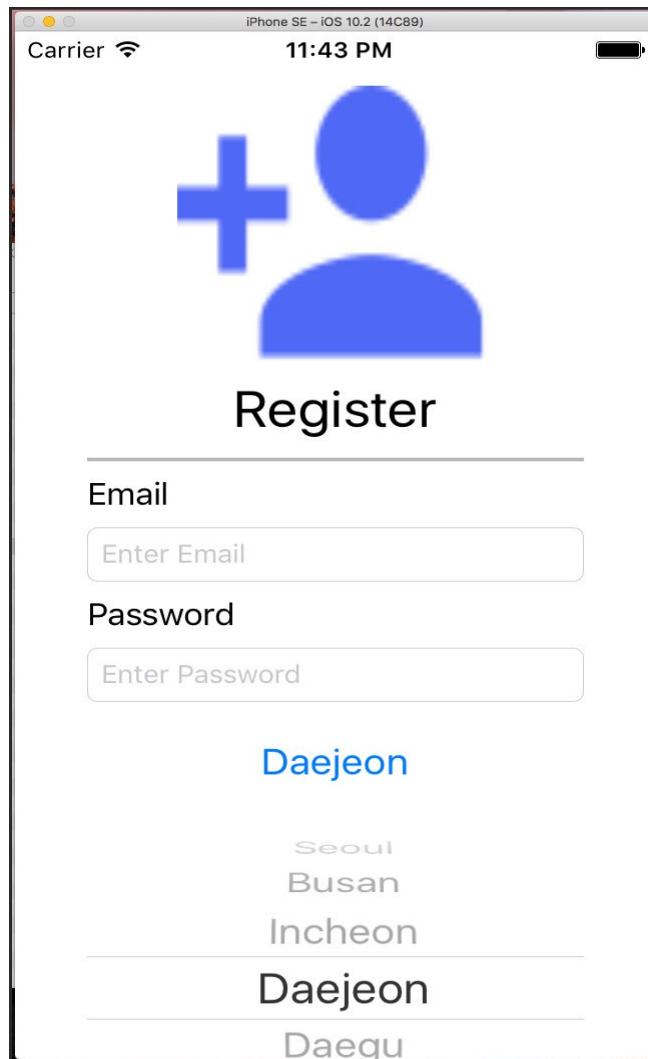
The screenshot shows a mobile application's registration screen. At the top, there is a large blue user icon with a plus sign. Below it, the word "Register" is displayed. The form includes fields for "Email" (with placeholder "Enter Email") and "Password" (with placeholder "Enter Password"). A section titled "Choose Your Home City" contains a UIPickerView. The picker shows the following options: Mountain View, Sunnyvale, Cupertino, Santa Clara, and San Jose. To the right of the UIPickerView, its corresponding Swift code is shown.

```
11 class ViewController: UIViewController, UIPickerViewDelegate, UIPickerViewDataSource {
12
13 @IBOutlet weak var cityPickerBtn: UIButton!
14 @IBOutlet weak var cityPicker: UIPickerView!
15
16 let cities = ["Seoul", "Busan", "Incheon", "Daejeon", "Daegu",
17 "Gwangju"]
18
19 override func viewDidLoad() {
20     super.viewDidLoad()
21
22     self.cityPicker.dataSource = self
23     self.cityPicker.delegate = self
24 }
25
26 @IBAction func cityBtnClicked(_ sender: Any) {
27     cityPicker.isHidden = false
28 }
29
30 func numberOfComponents(in pickerView: UIPickerView) -> Int {
31     return 1
32 }
33
34 func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {
35     return cities.count
36 }
37
38 func pickerView(_ pickerView: UIPickerView, titleForRow row: Int,
39 forComponent component: Int) -> String? {
40     return cities[row]
41 }
42
43 func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int,
44 inComponent component: Int) {
45     cityPickerBtn.setTitle(cities[row], for: UIControlState.normal)
46     cityPicker.isHidden = true
47 }
```

PickerView 코드 넣기 (선택시 버튼의 텍스트 변화)

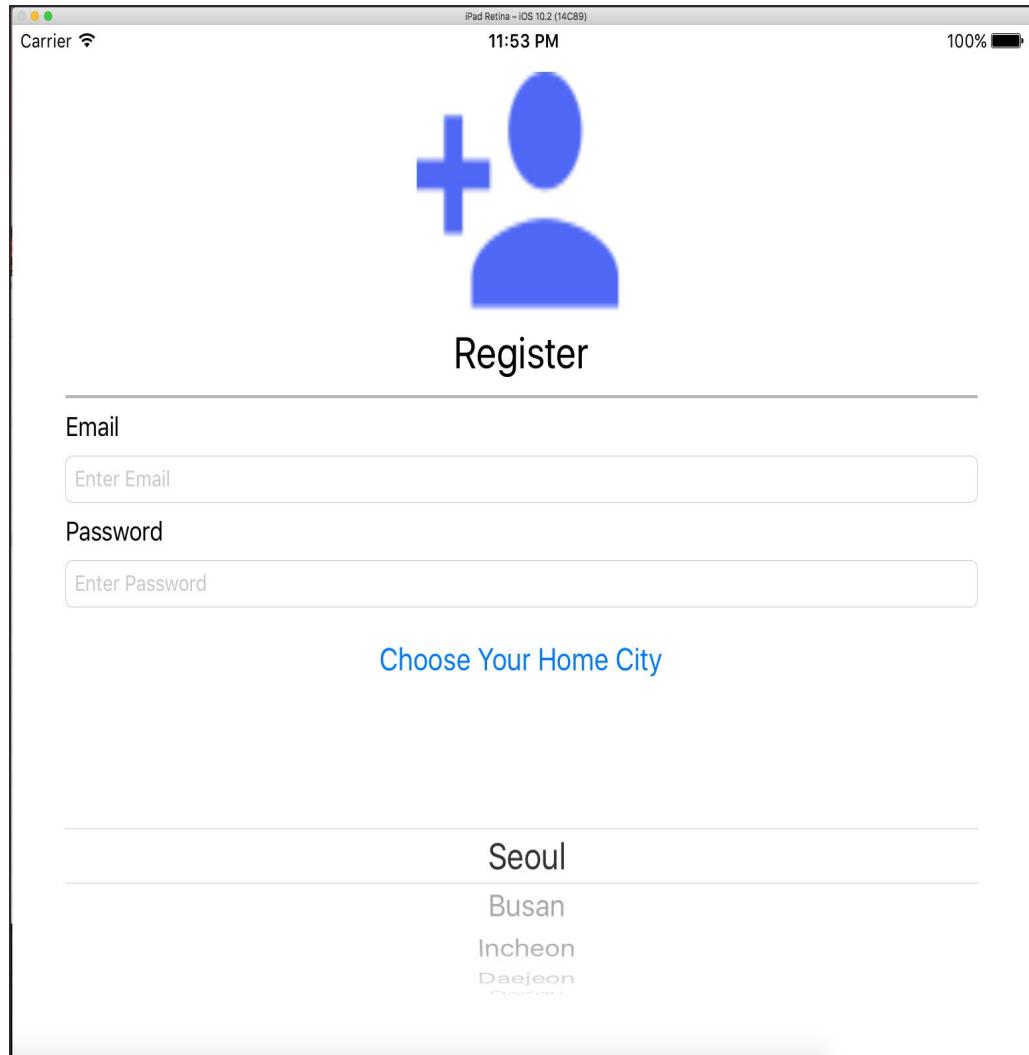
1. 행의 개수
2. 각 행의 제목
3. 선택된 행

AutoLayout



iPhone SE 환경 구동 시

AutoLayout

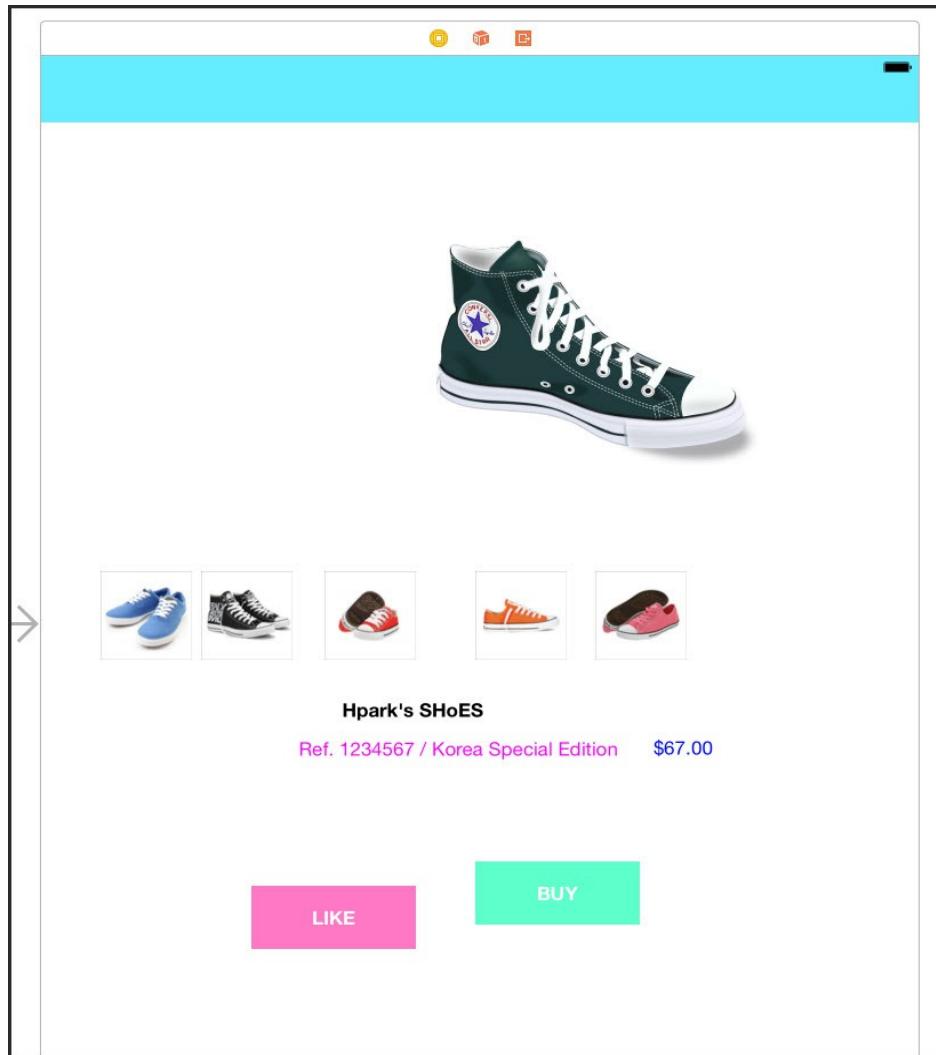


iPad 환경 구동 시

AutoLayout

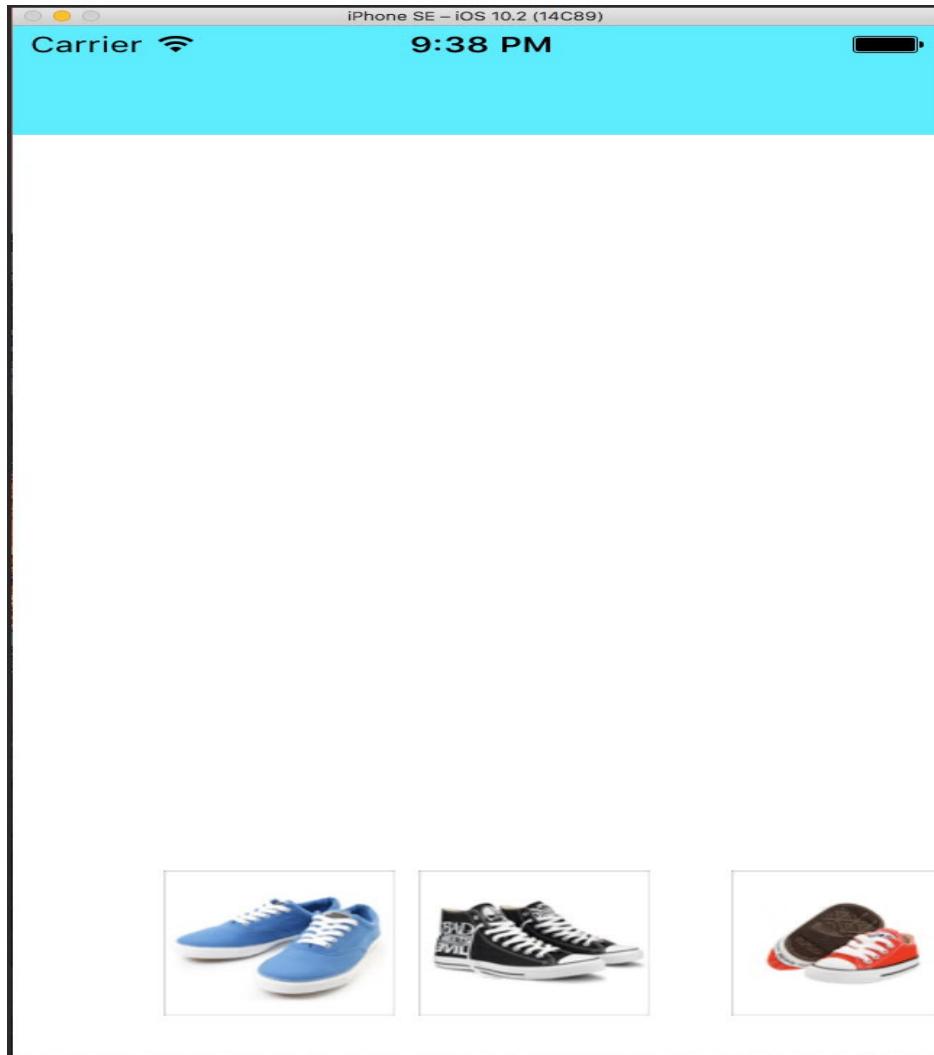
1. 위와 고정된 간격으로 밑부분에 남는 공간이 생김
2. 상호 의존적 - 컴포넌트 하나를 크게 바꾸면 ...

AutoLayout



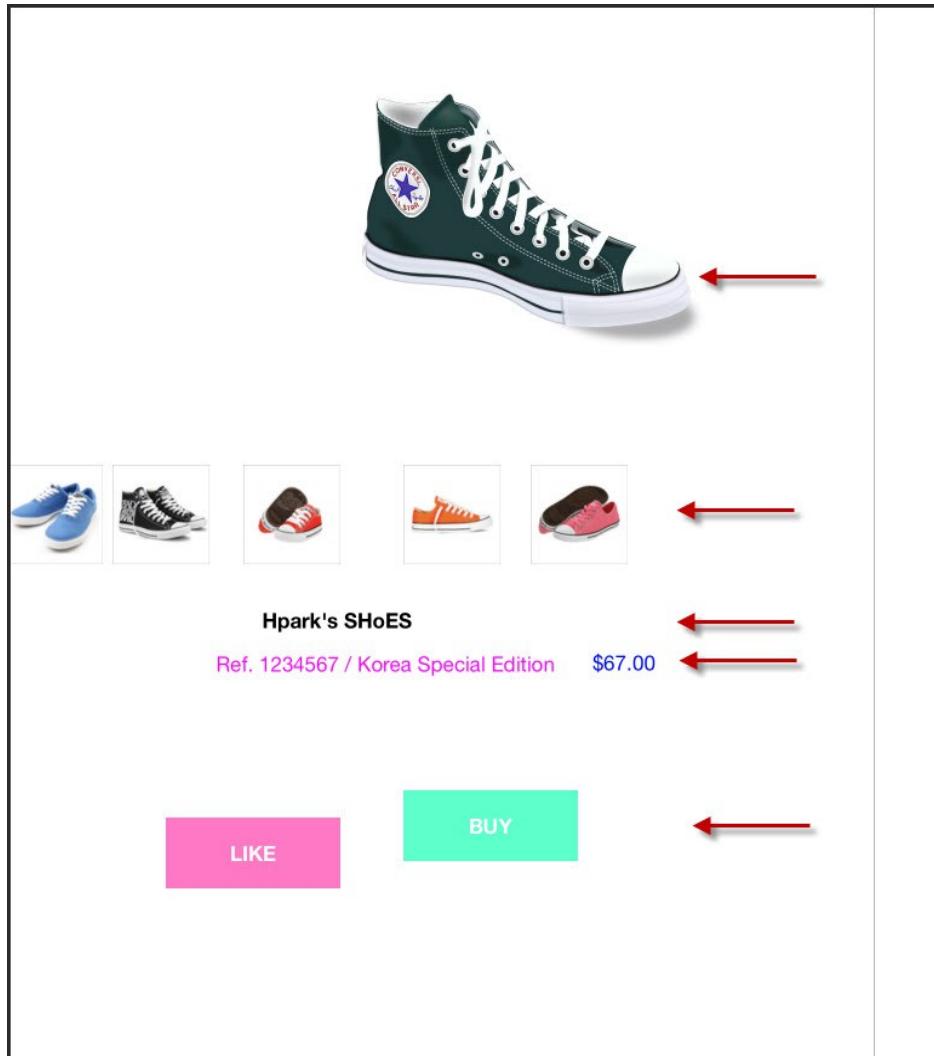
2. 신발 마켓 app
(StackView 설정)

AutoLayout



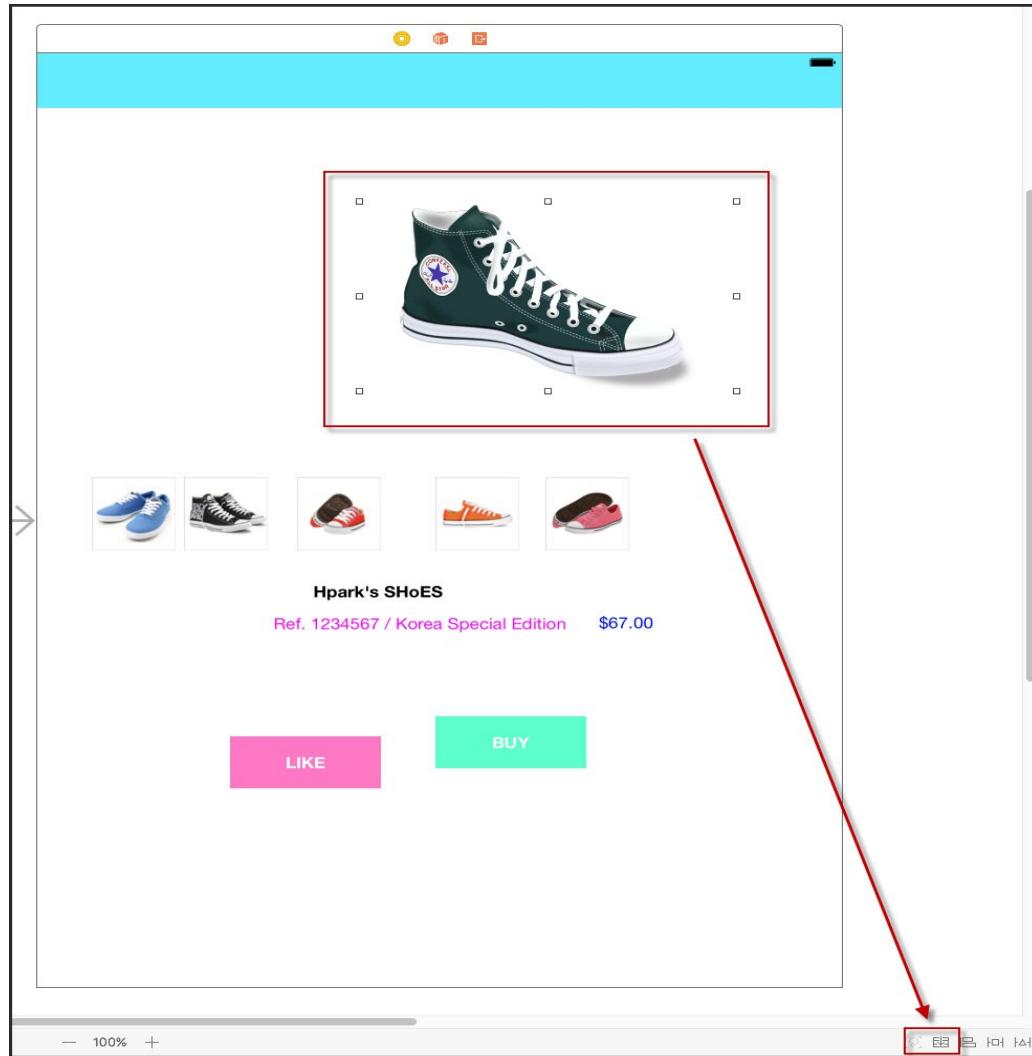
구동 결과

AutoLayout



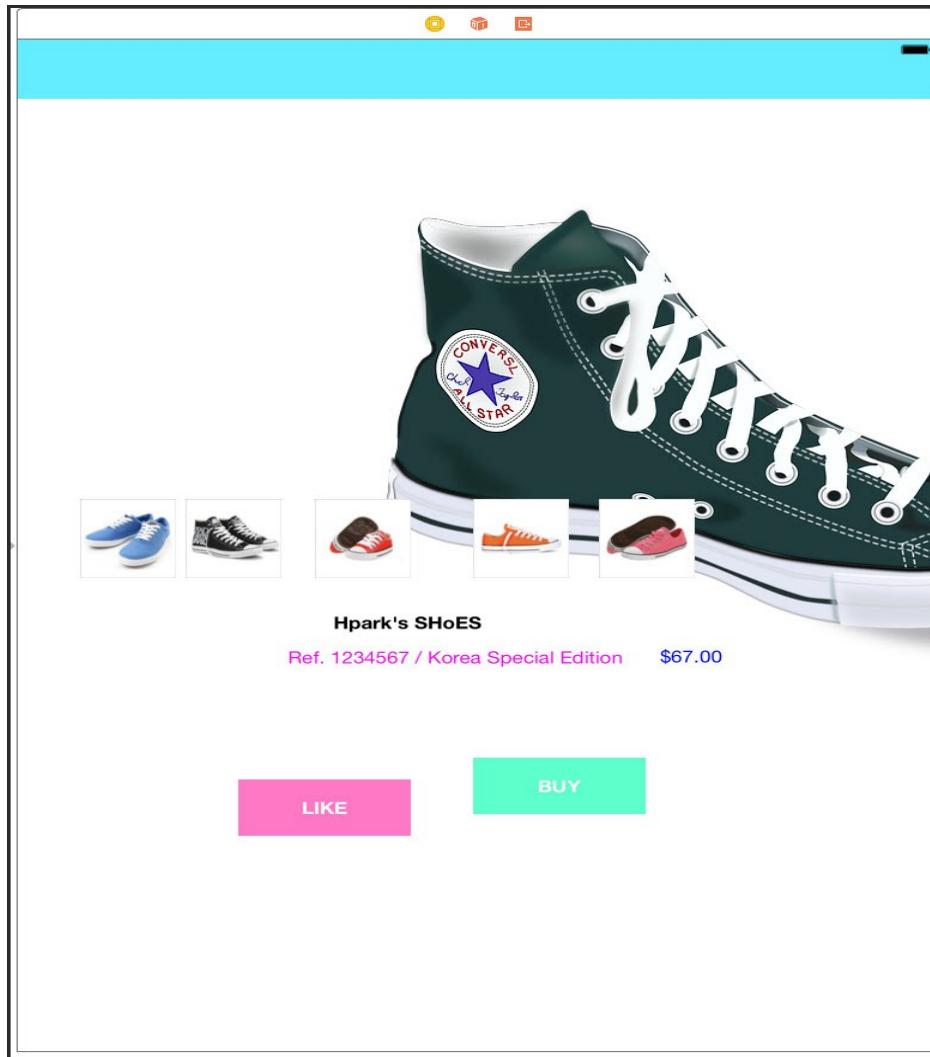
StackView 묶음
» 총 5 개

AutoLayout



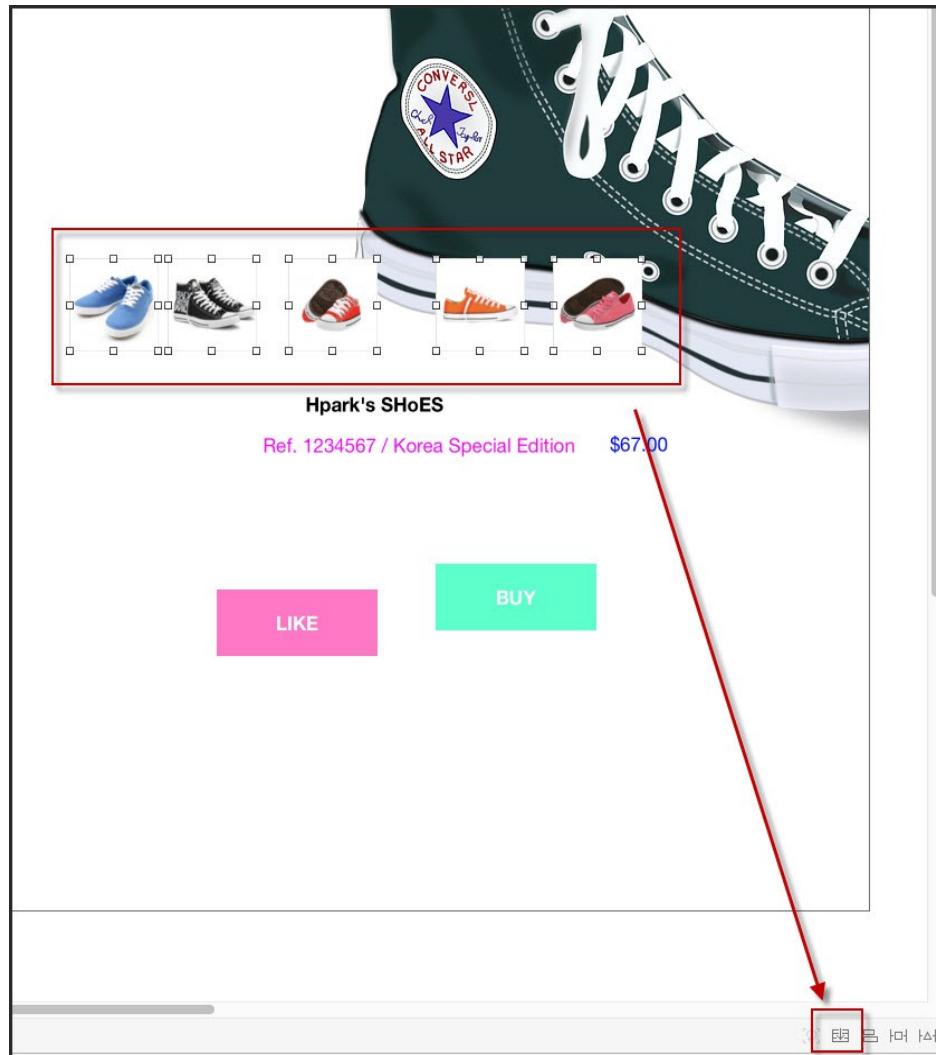
한 뮤음씩
StackView 만들기

AutoLayout



?

AutoLayout



다시한번 ..

AutoLayout



Hpark's SHoES

Ref. 1234567 / Korea Special Edition \$67.00

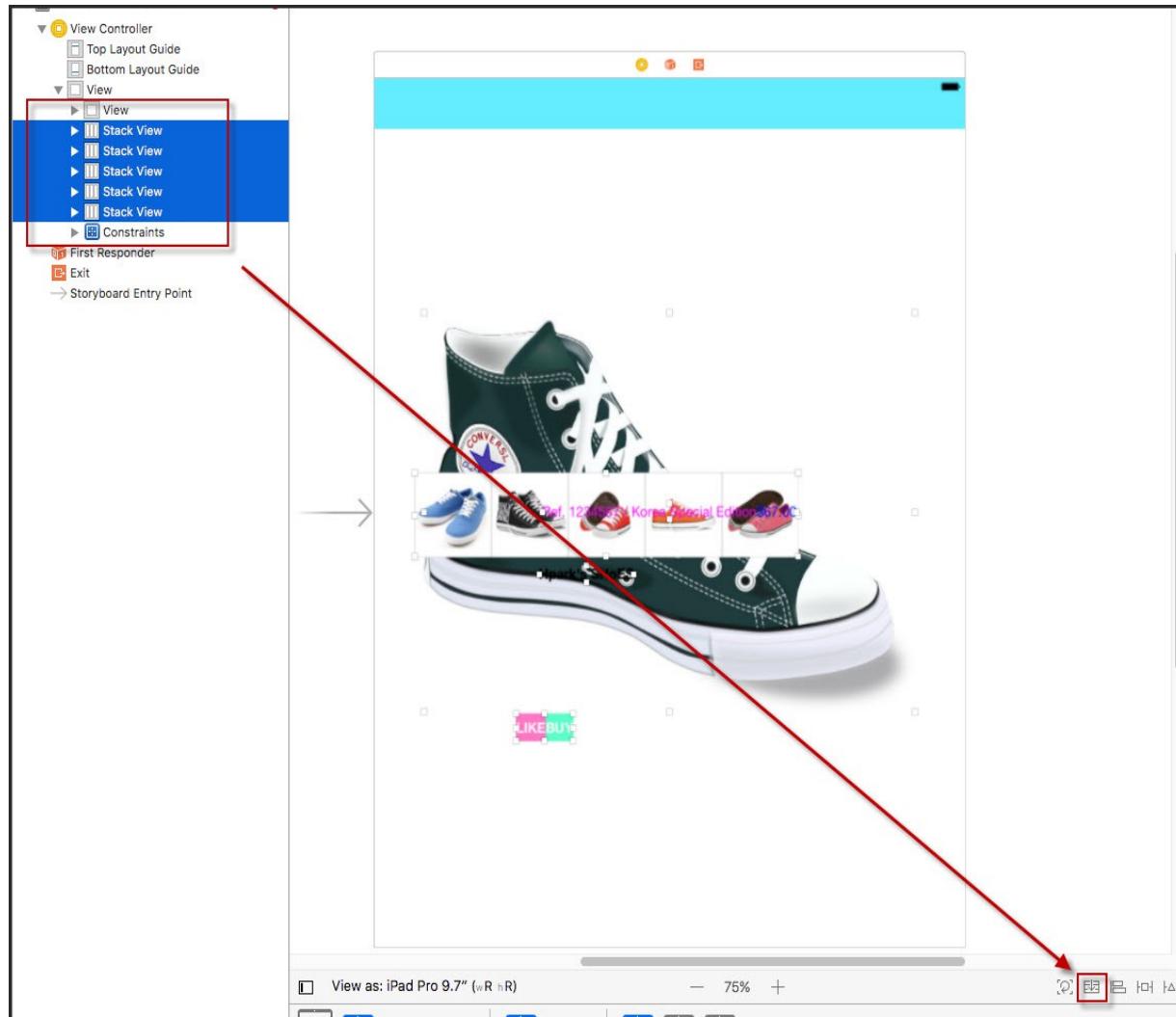
LIKE BUY

AutoLayout



..?

AutoLayout



5 개의
StackView 를 더
큰 StackView 로
묶음

AutoLayout



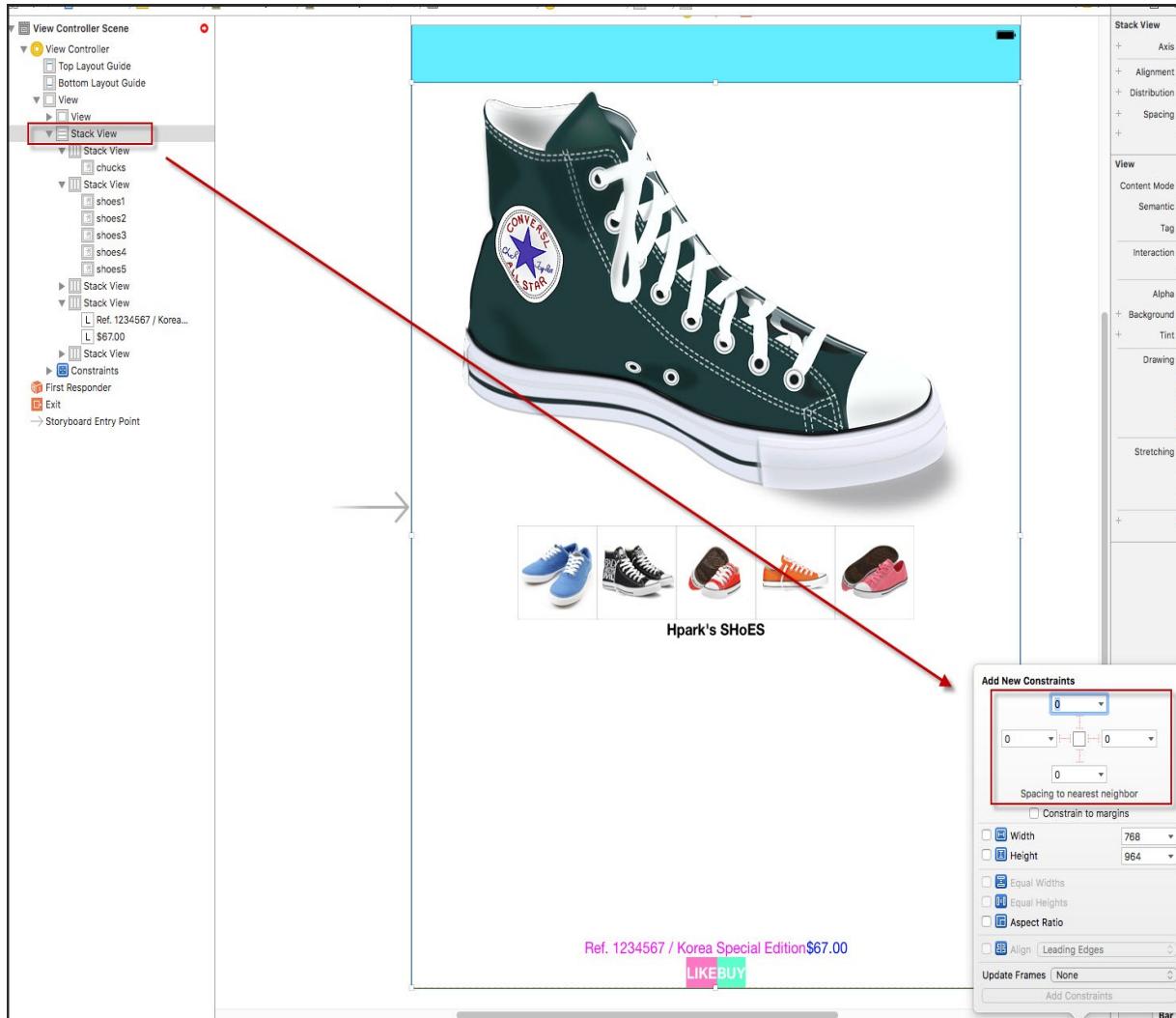
Hpark's SHoES

Ref. 1234567 / Korea Special Edition \$67.00

LIKEBUY

StackView 정렬

AutoLayout



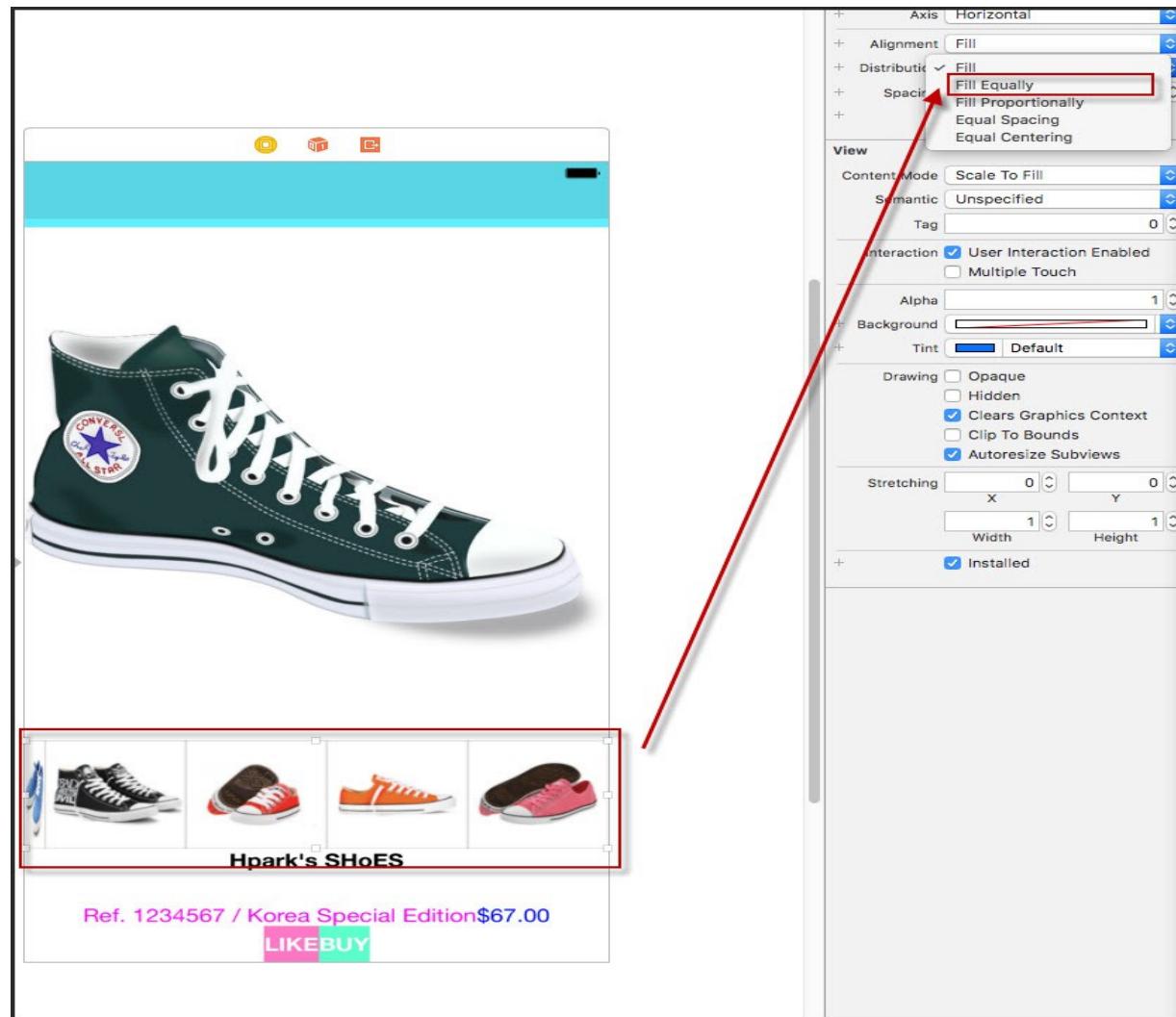
모든 개체를 포함한
StackView 를 컨
테이너에
(0,0,0,0) 으로
constraint 설정

AutoLayout



iPhone 구동 상태

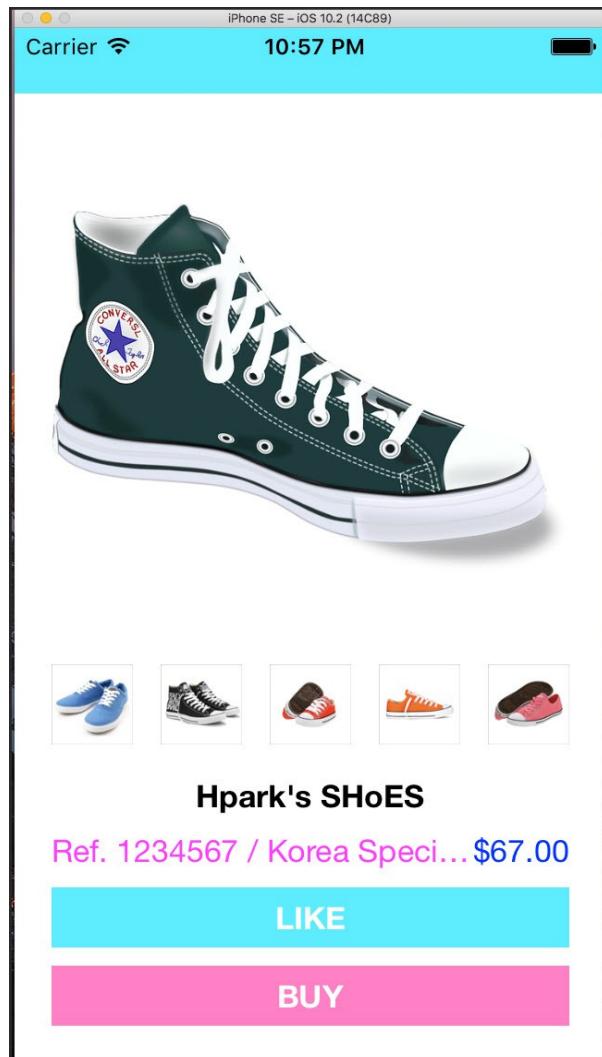
AutoLayout



각각의 컴포넌트에
필요한 추가설정

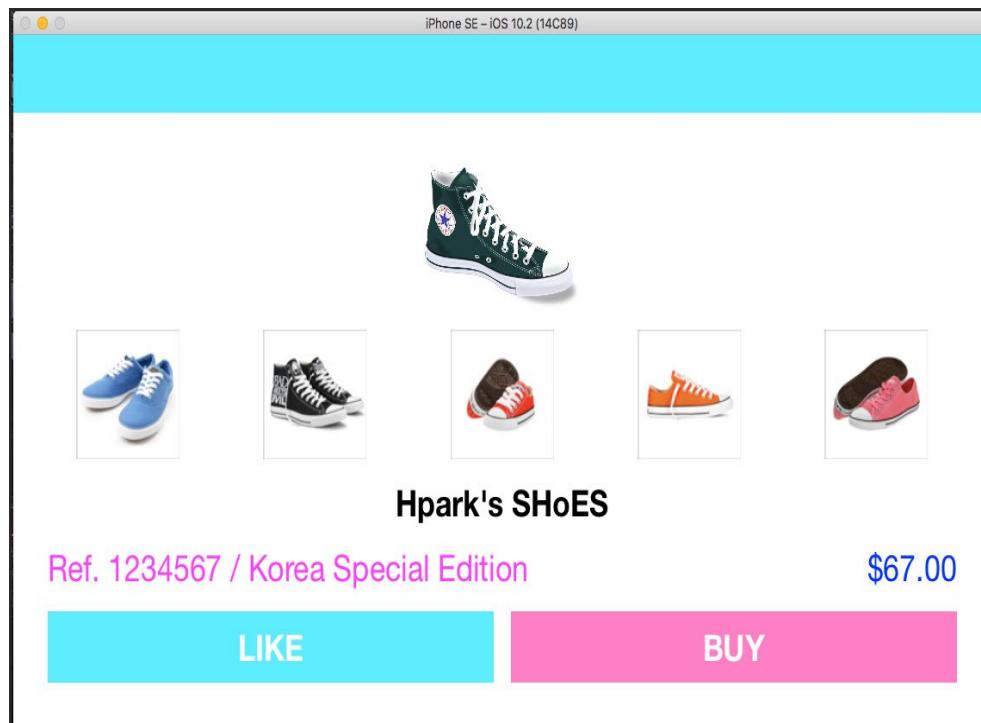
- Alignment
- Spacing
- Content Hugging
(늘어나기 싫은 정도)
- Content Compression
Resistance
(줄어들기 싫은 정도)

AutoLayout



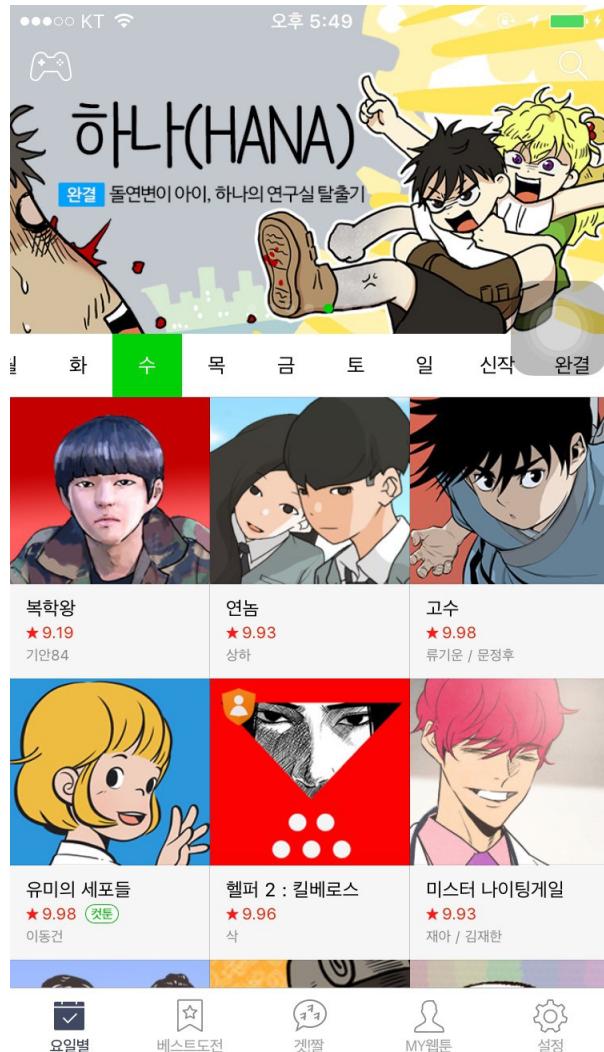
iPhone 구동 상태

AutoLayout



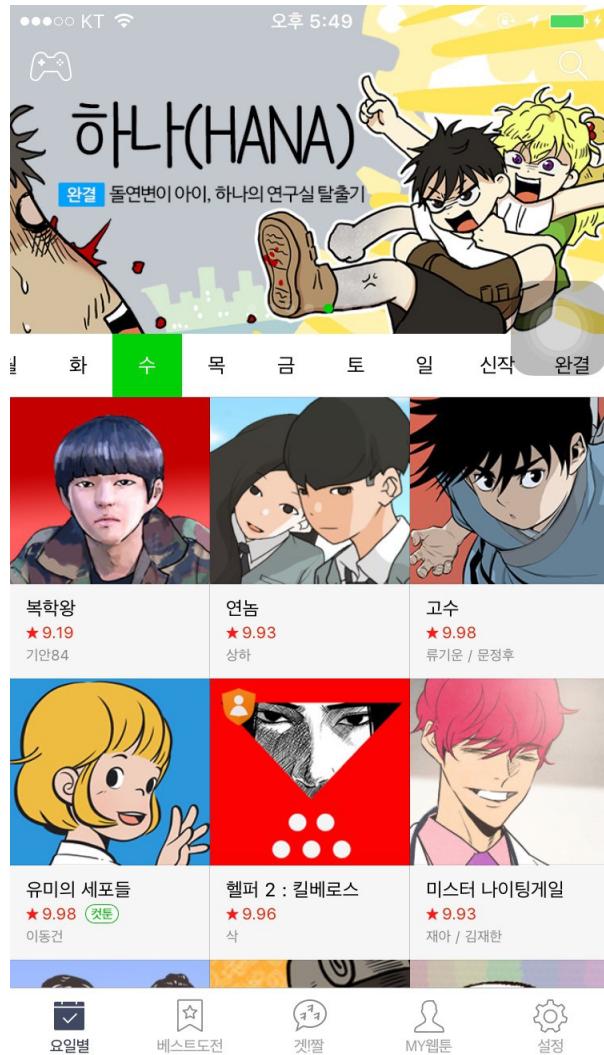
iPhone 구동 상태
(LandScape 모드)

MVC 패턴



자주 사용하는 앱
(네이버 웹툰)

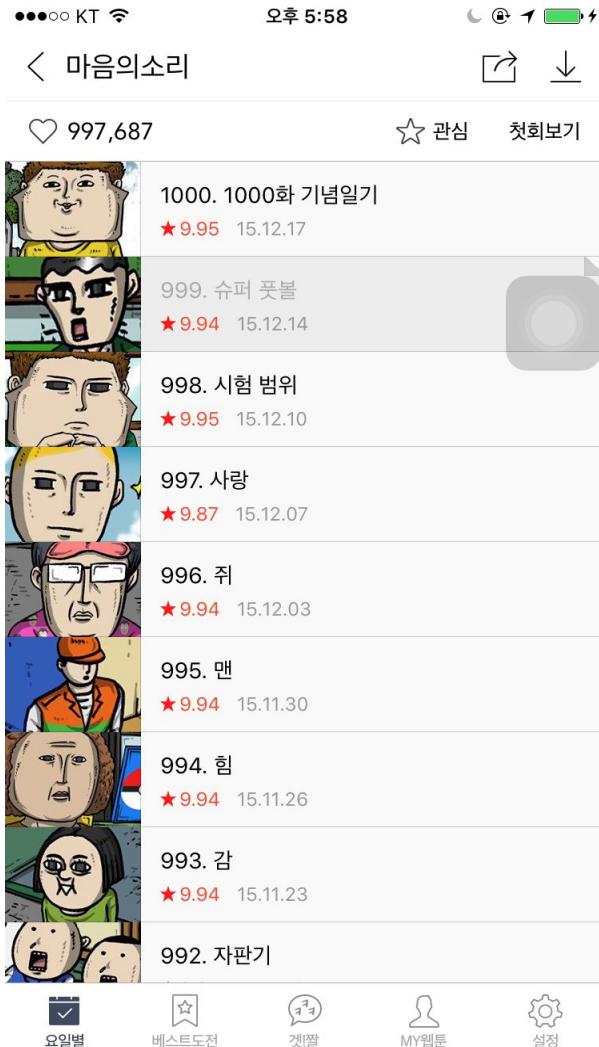
MVC 패턴 - Controller



Class MainView 컨트롤러

- ImageView?
메인 이미지 전시
- ScrollView?
요일별
- TableView?
웹툰 작품 전시
- Toolbar?
다른 웹툰 관련 서비스

MVC 패턴 - Controller



Class ContentListView 컨트롤러

- Navigation Bar?
돌아가기, 저장 및 공유
- TableView?
웹툰 작품 전시
- Tool Bar?
다른 웹툰 관련 서비스

MVC 패턴 - Controller



Class ContentView 컨트롤러

- Navigation Bar?
돌아가기 및 부가기능
- Scroll View?
웹툰 1 회분 전시
- Tool Bar?
댓글창 및 회간 이동

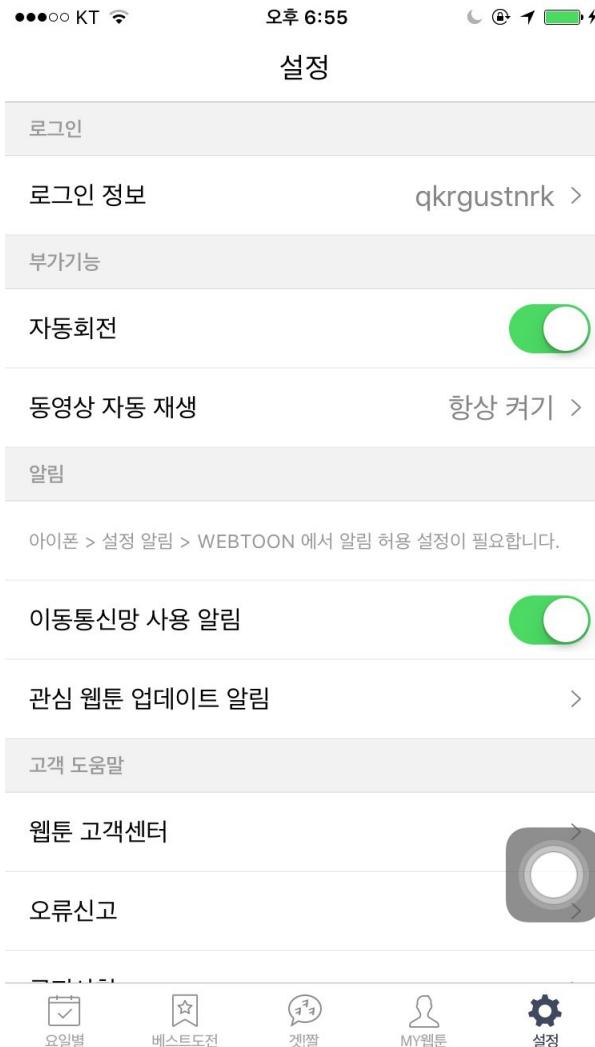
MVC 패턴 - Controller



Class CommentsView 컨트롤러

- Navigation Bar?
돌아가기 및 부가기능
- Tab Bar?
Best 댓글 및 전체 댓글
- TableView?
전체 댓글 전시
- TextField
댓글작성

MVC 패턴 - Controller



Class SettingsView 컨트롤러

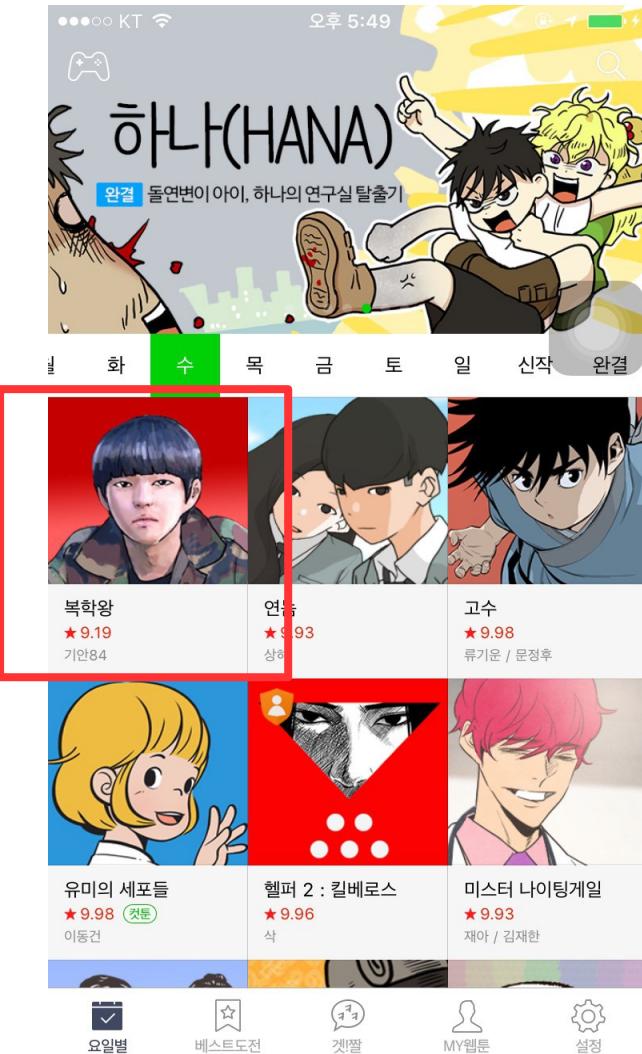
- TableView?

각종 설정

- Tool Bar?

다른 웹툰 관련 서비스

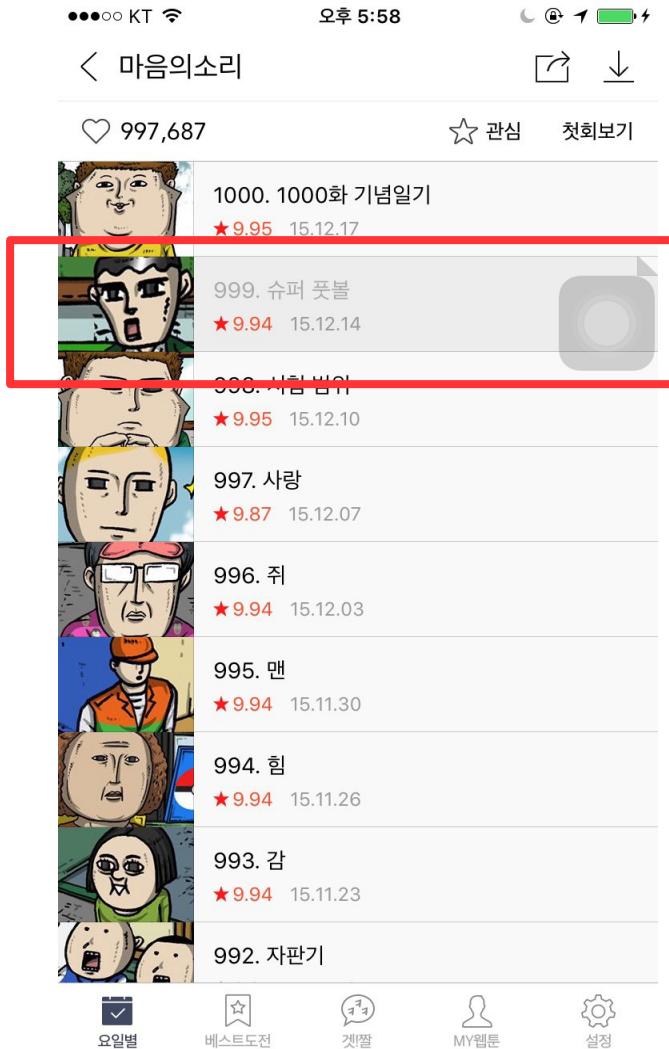
MVC 패턴 - Model



Class pieceOfWebtoon

- mainImageUrl
- title
- averageStarRating
- totalHearts
- author
- contents
- dayWebtoon
- isCompleted
- ...

MVC 패턴 - Model

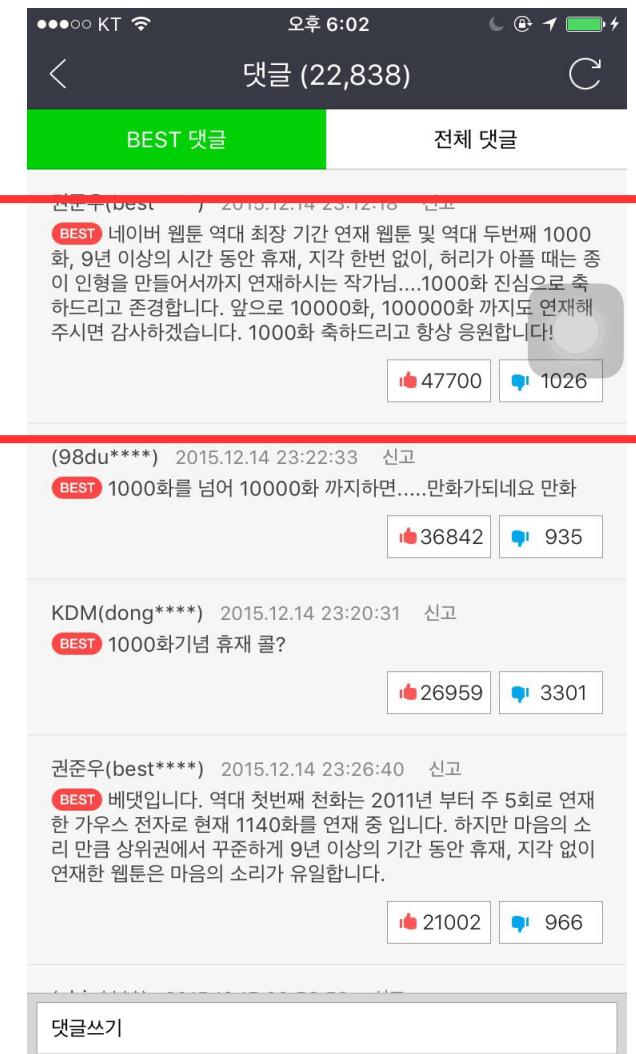


Class Content

- mainImageUrl
- title
- averageStarRating
- numHearts
- dayOfPosting
- comments
- content

...

MVC 패턴 - Model

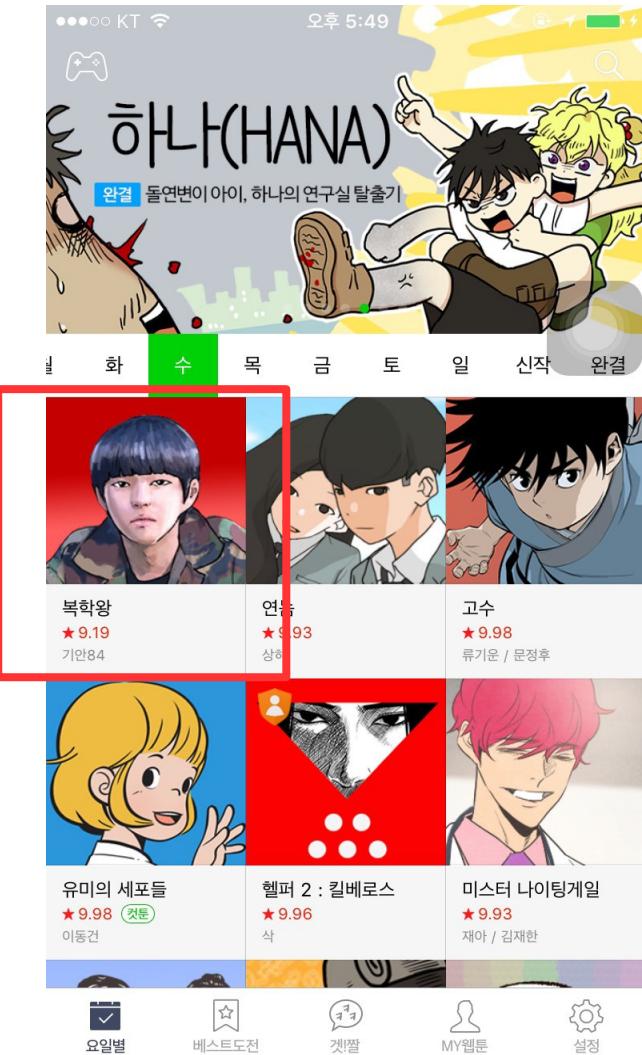


Class Comment

- author
- numThumbsUp
- numThumbsDown
- commentContent
- isBest
- dayOfCommentPosting

...

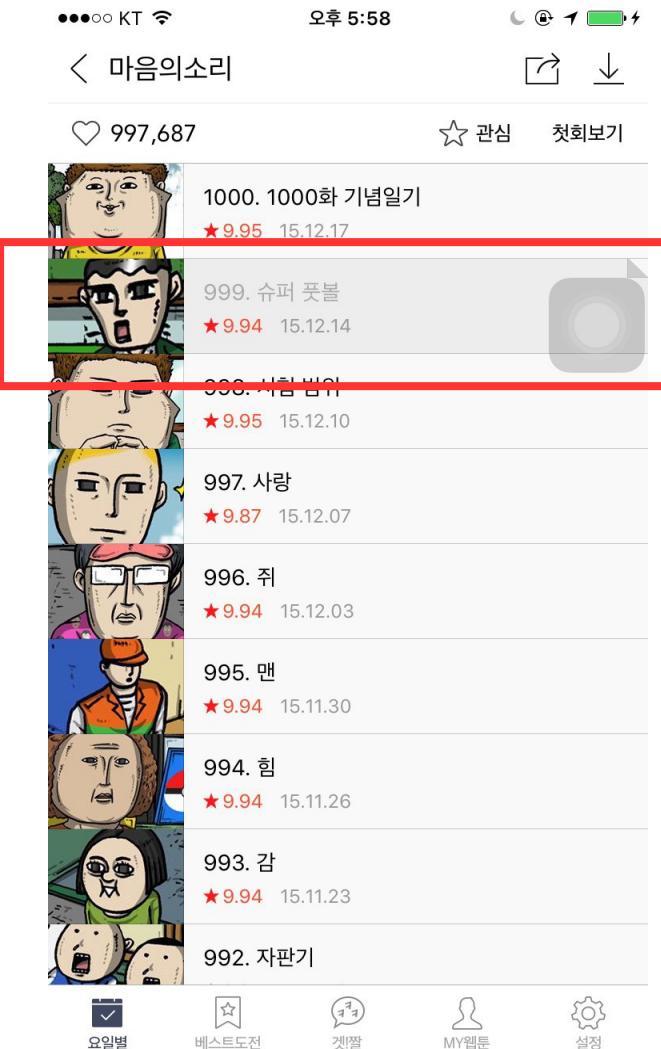
MVC 패턴 - View



Class WebtoonCell

- Image - 웹툰 메인 이미지
- Label - 제목, 별점수, 작가이름

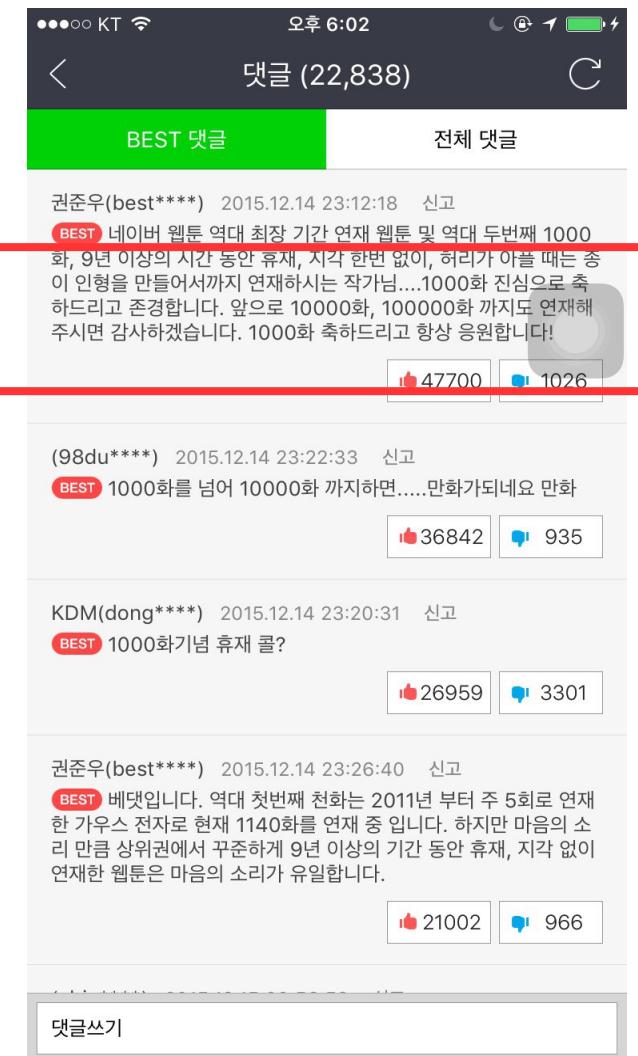
MVC 패턴 - View



Class ContentCell

- Image - 회당 메인 이미지
- Label - 제목, 별점수, 올린 날짜

MVC 패턴 - View



Class CommentsCell

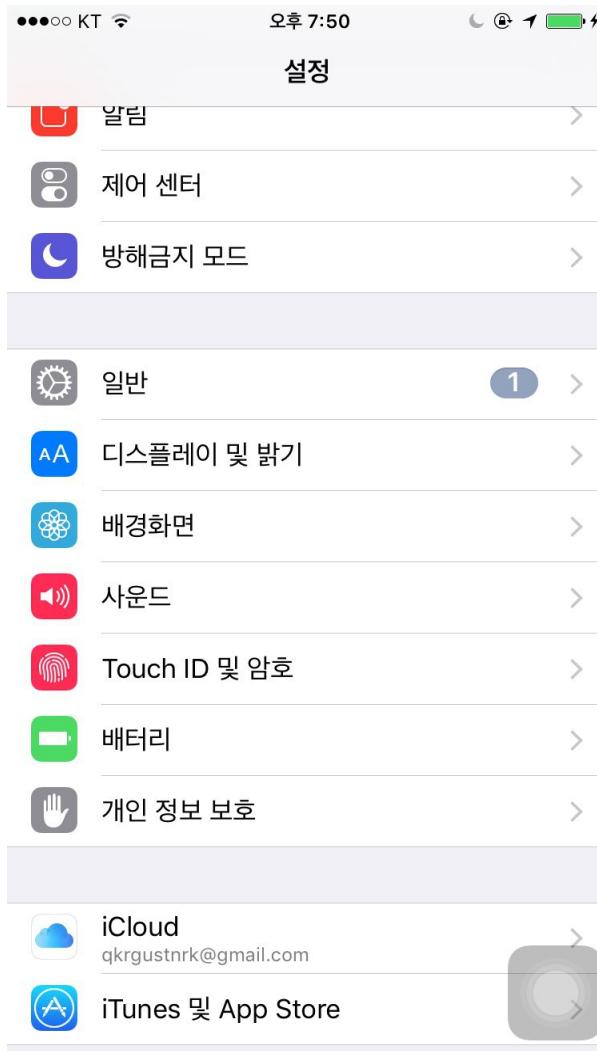
- Label - 올린이, 아이디, 올린 날짜
- Button - 신고, 좋아요, 싫어요

MVC 패턴 - View

새로 커스터마이징 하는 UI 컴포넌트들도 관련있는 컴포넌트를 상속하여 클래스를 구성하여 View에 포함시킬 수 있다.

```
Ex) class DancingBtn : UIButton {  
    /* CODE */  
}
```

Navigation 동작양식



iPhone 설정 앱

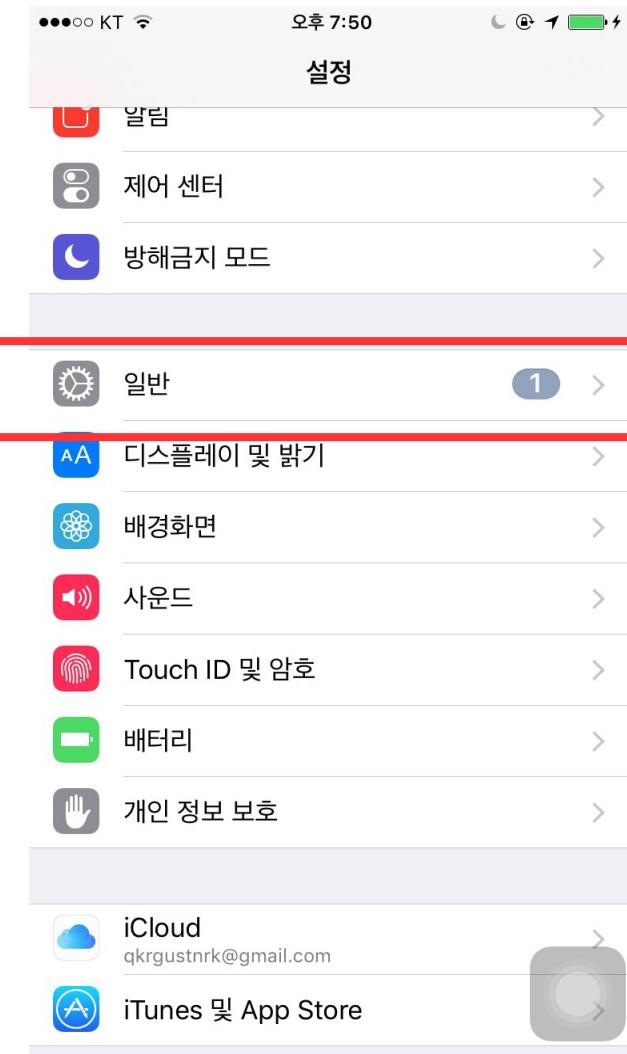
Navigation 동작양식



iPhone 설정 앱 출입

설정

Navigation 동작양식



iPhone 설정 앱 출입
» 일반



일반

설정

Navigation 동작양식



iPhone 설정 - 일반

일반

설정

Navigation 동작양식



iPhone 설정
- 일반
» 언어 및 지역

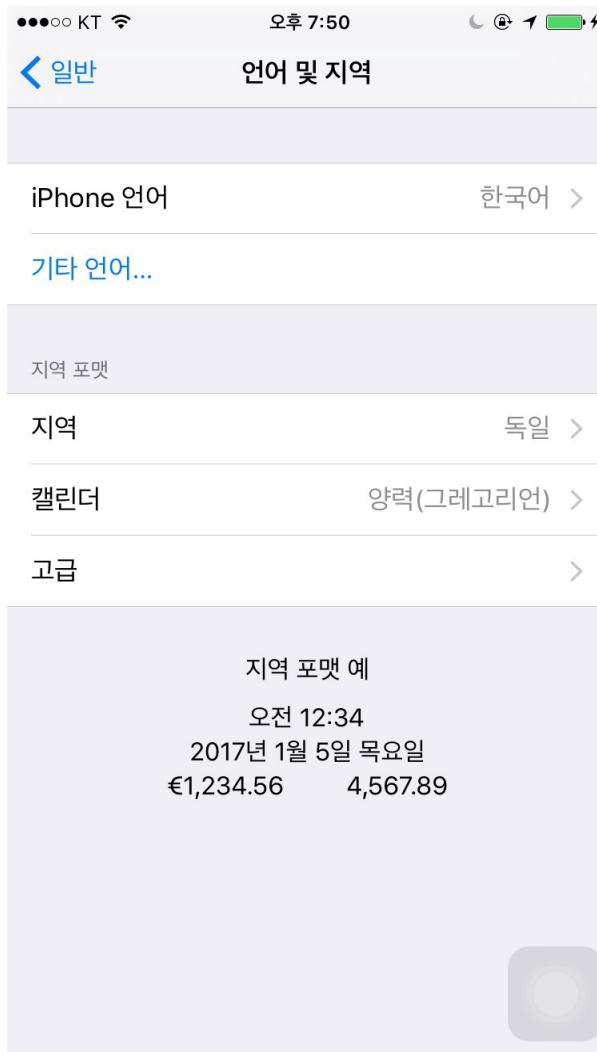


언어 및 지역

일반

설정

Navigation 동작양식



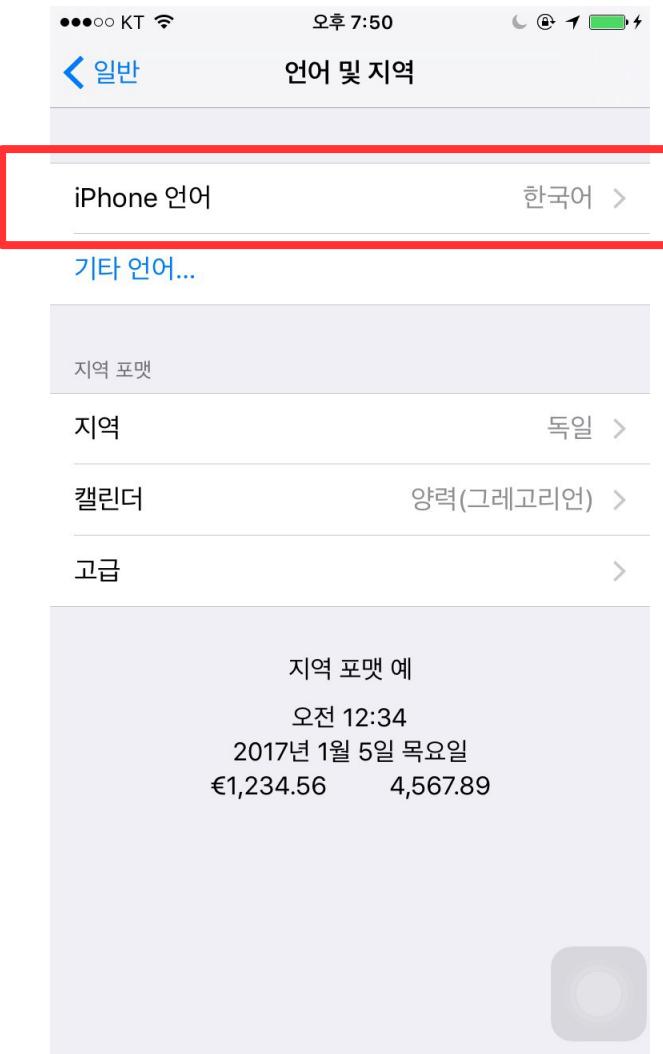
iPhone 설정
- 일반
- 언어 및 지역

언어 및 지역

일반

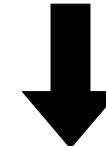
설정

Navigation 동작양식



iPhone 설정

- 일반
 - 언어 및 지역
-)) iPhone 언어



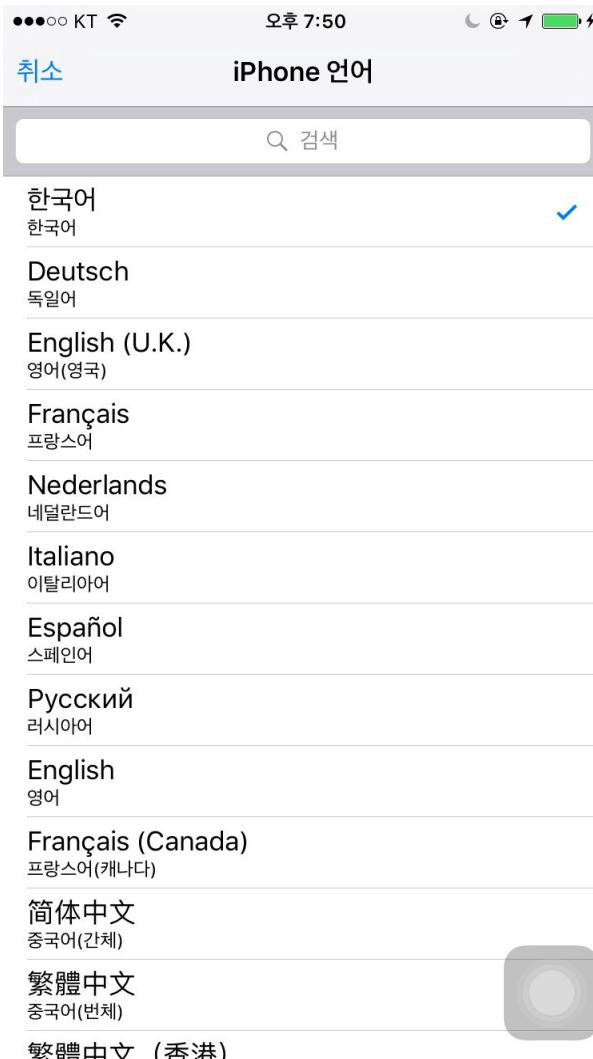
iPhone 언어

언어 및 지역

일반

설정

Navigation 동작양식



iPhone 설정

- 일반
- 언어 및 지역
- iPhone 언어

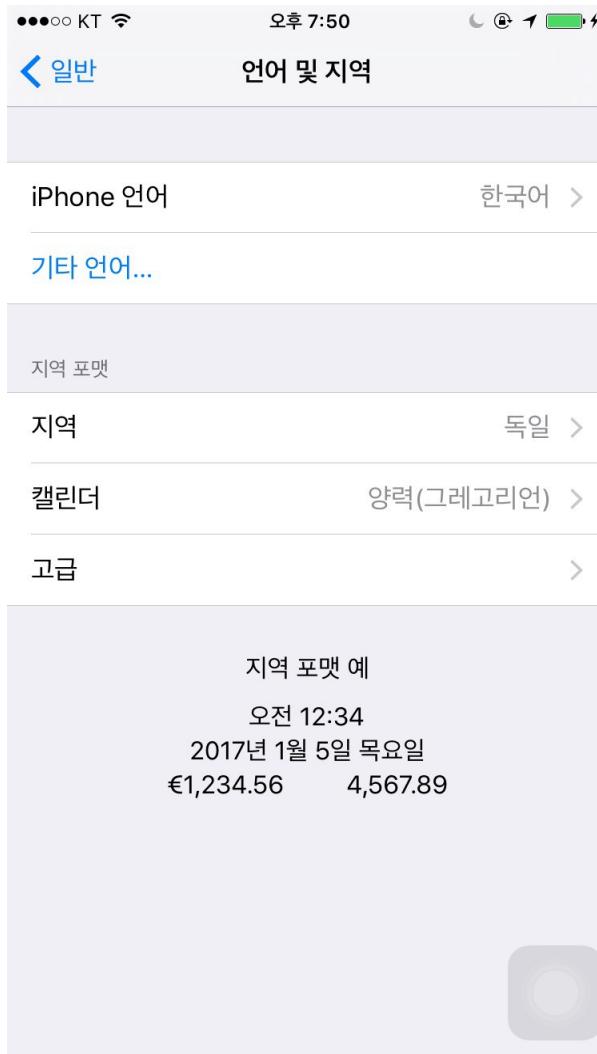
iPhone 언어

언어 및 지역

일반

설정

Navigation 동작양식



iPhone 설정
- 일반
- 언어 및 지역 <<



iPhone 언어

언어 및 지역

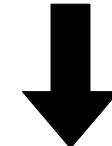
일반

설정

Navigation 동작양식



iPhone 설정
- 일반
- 언어 및 지역
}} 캘린더



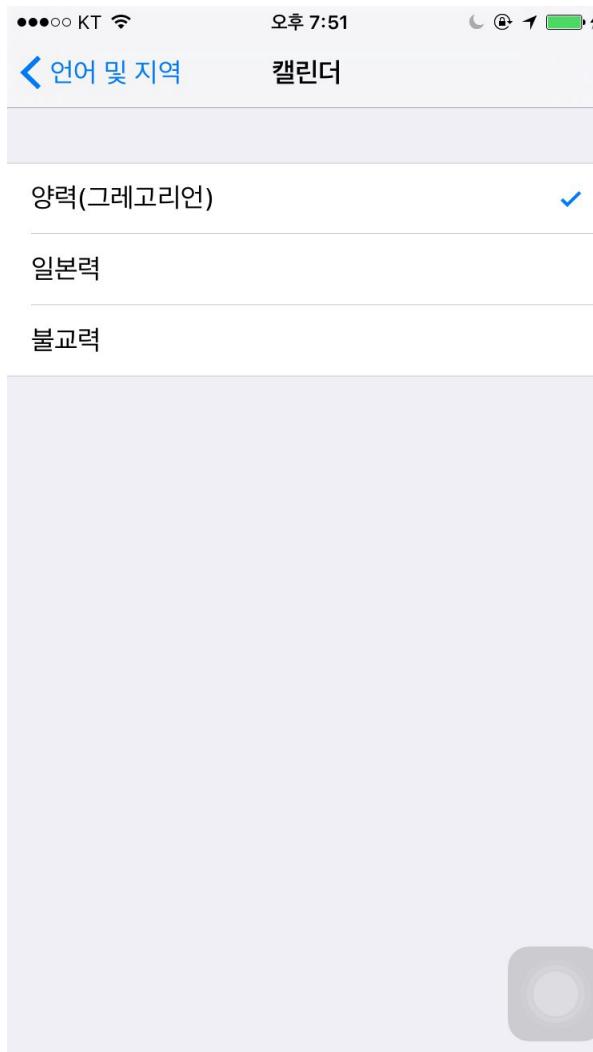
캘린더

언어 및 지역

일반

설정

Navigation 동작양식



iPhone 설정

- 일반
- 언어 및 지역
- >> 캘린더

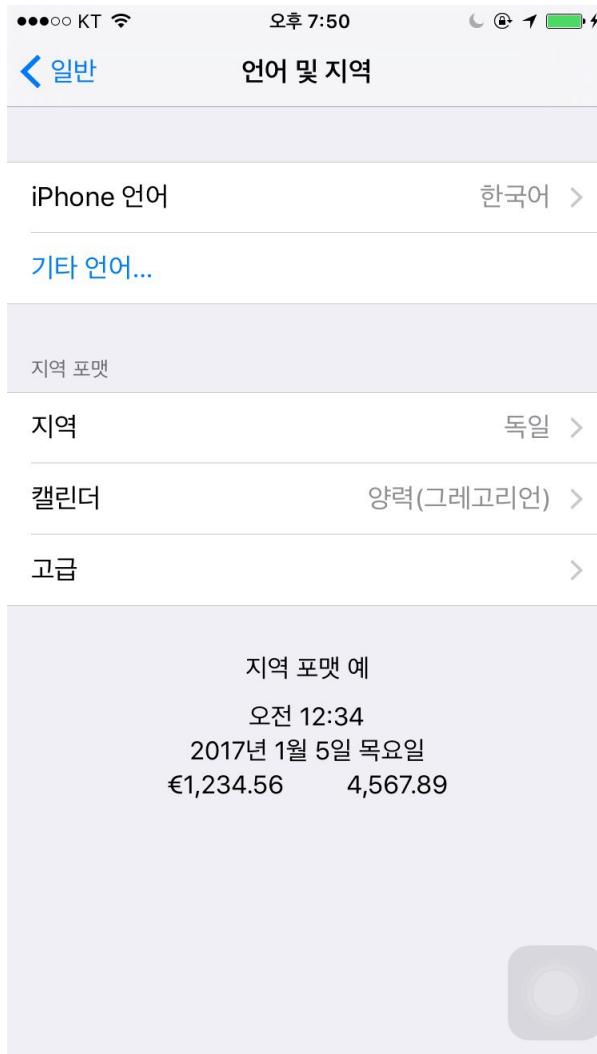
캘린더

언어 및 지역

일반

설정

Navigation 동작양식



iPhone 설정
- 일반
- 언어 및 지역 <<



캘린더

언어 및 지역

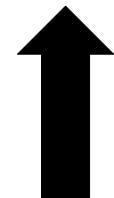
일반

설정

Navigation 동작양식



iPhone 설정
- 일반 <<



언어 및 지역

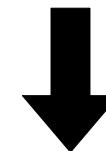
일반

설정

Navigation 동작양식



iPhone 설정
- 일반
» 자동 잠금

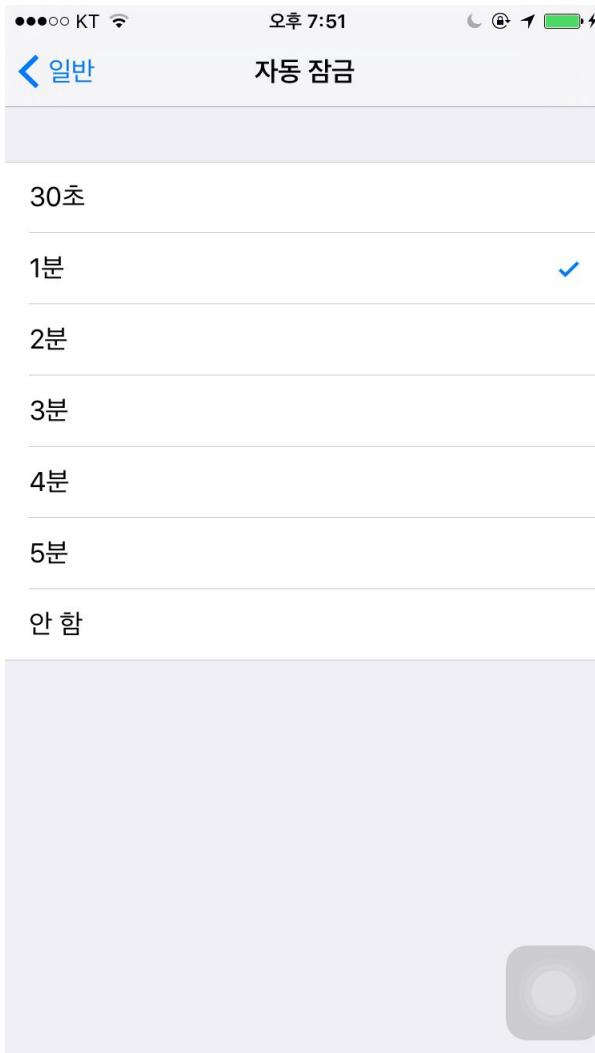


자동 장금

일반

설정

Navigation 동작양식



iPhone 설정

- 일반
- 자동 잠금

자동 장금

일반

설정

Navigation 동작양식



결론 : Stack 자료구조하고 유사함

Nav 가 아닌 다른방식 ?

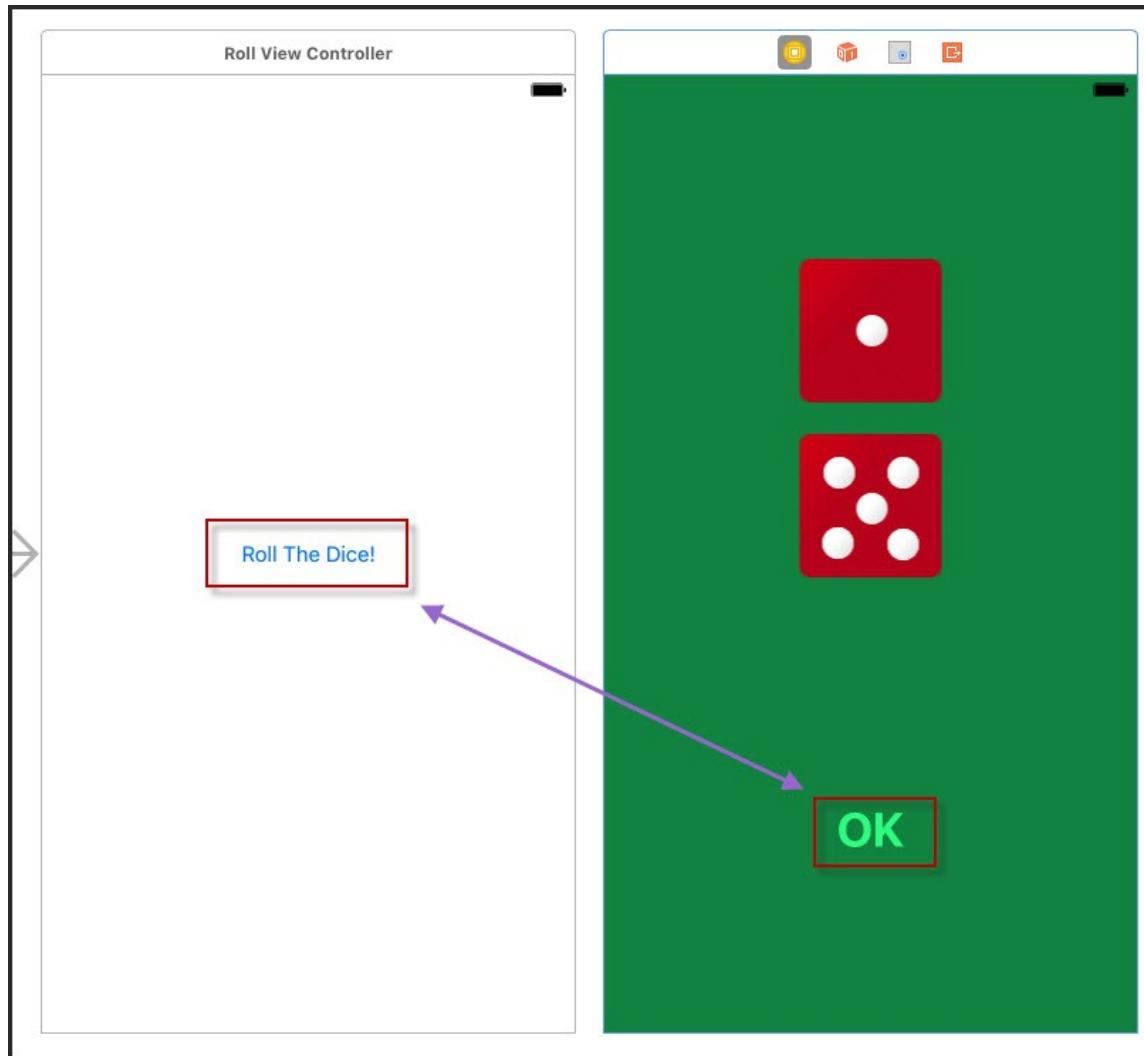
화면전환에는 다음과 같이 여러가지 방법이 있음

Code

Segue & Code

Segue Only

Nav 가 아닌 다른방식 ?



- RollView 컨트롤러
- DiceView 컨트롤러

Nav 가 아닌 다른방식 ?

Code Only

- RollViewController

```
// MARK: Actions

@IBAction func rollTheDice() {
    var controller: DiceViewController

    controller = self.storyboard?.instantiateViewController(withIdentifier: "DiceViewController") as!
        DiceViewController

    controller.firstValue = self.randomDiceValue()
    controller.secondValue = self.randomDiceValue()

    present(controller, animated: true, completion: nil)
}
```

Nav 가 아닌 다른방식 ?

Code Only

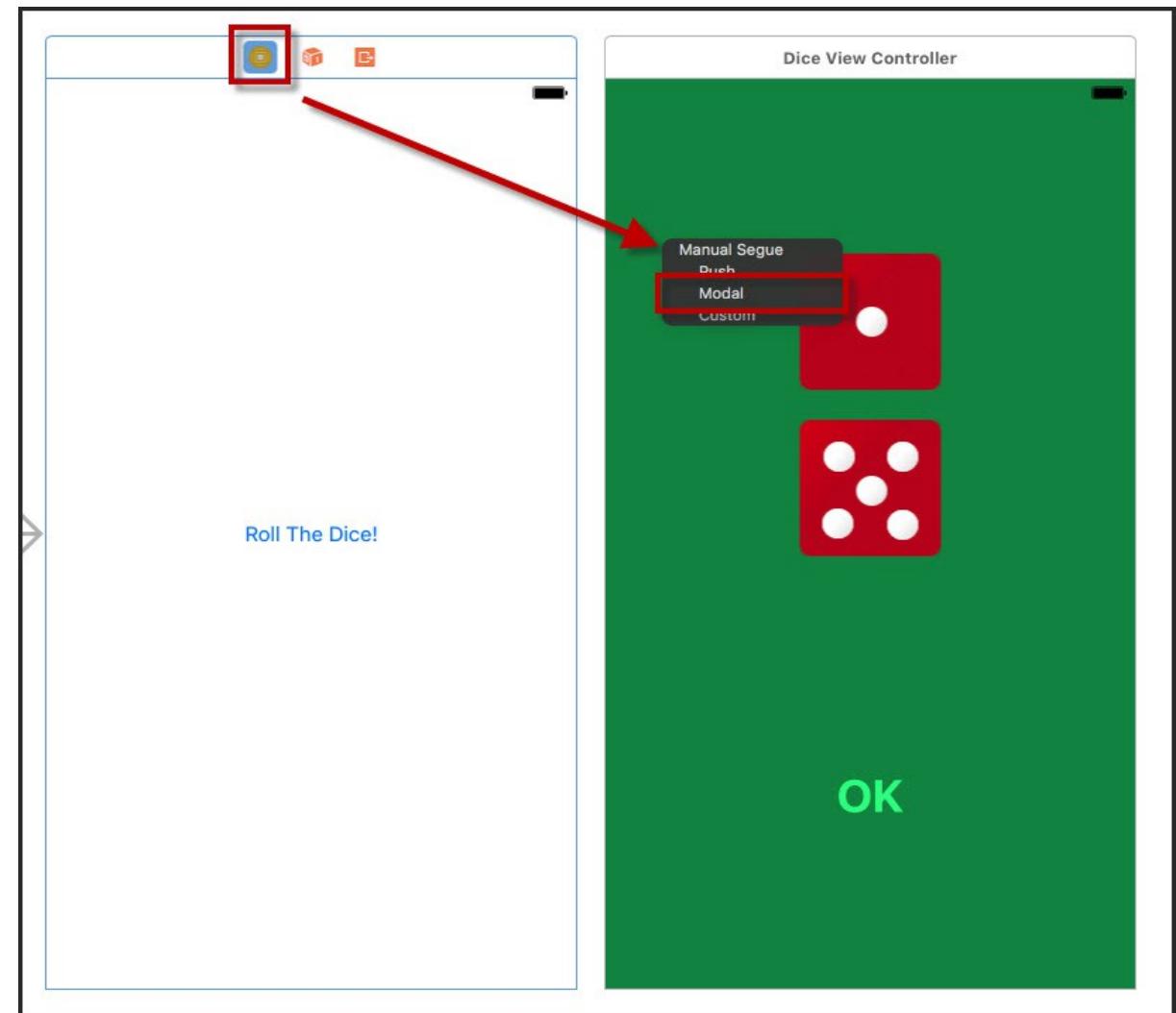
- DiceViewController

```
/**  
 *   dismiss this view controller  
 */  
func dismiss() {  
    self.dismiss(animated: true, completion: nil)  
}
```

Nav 가 아닌 다른방식 ?

Segue & Code

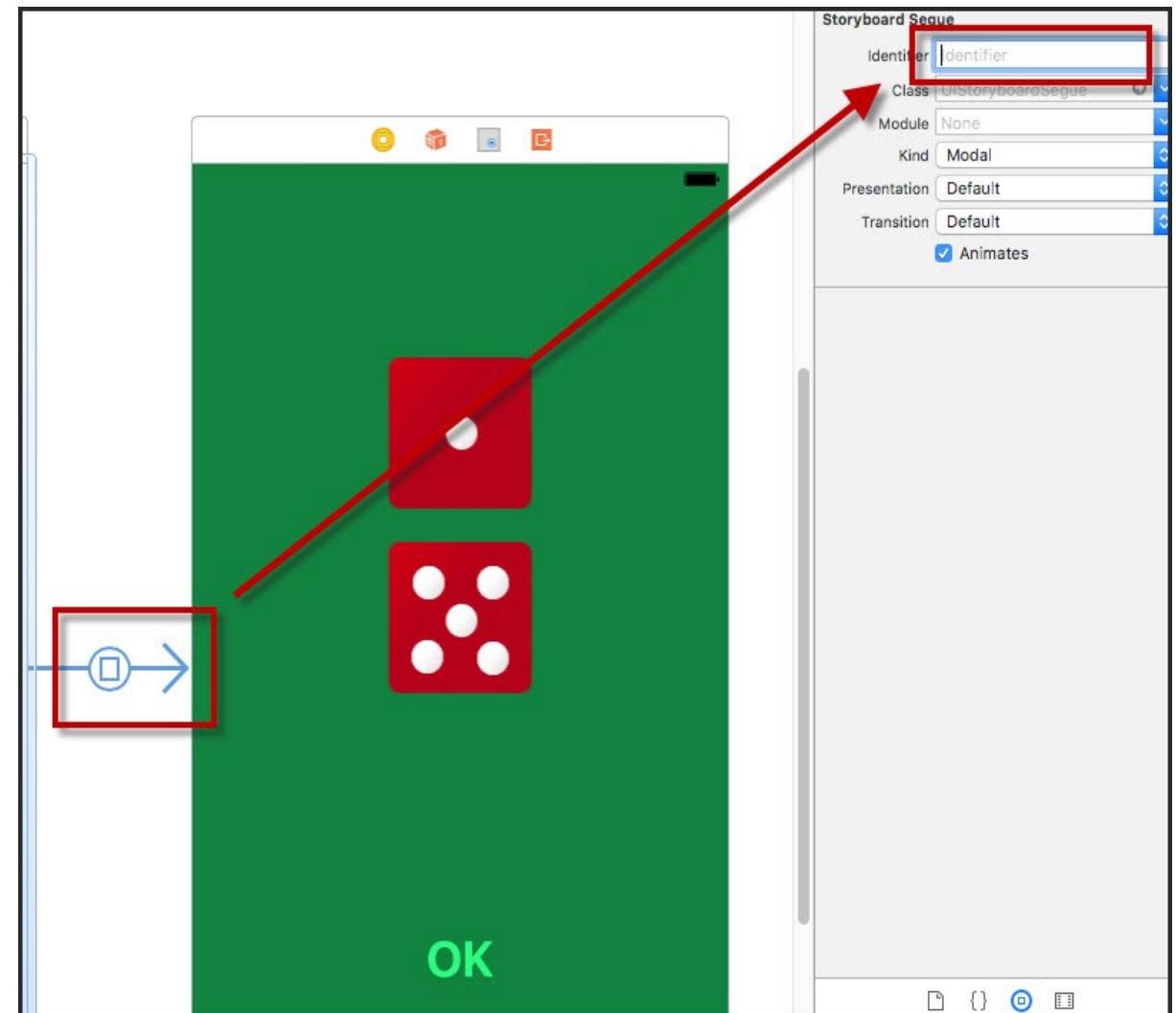
세그웨이 생성



Nav 가 아닌 다른방식 ?

Segue & Code

Identifier 지정



Nav 가 아닌 다른방식 ?

Segue & Code

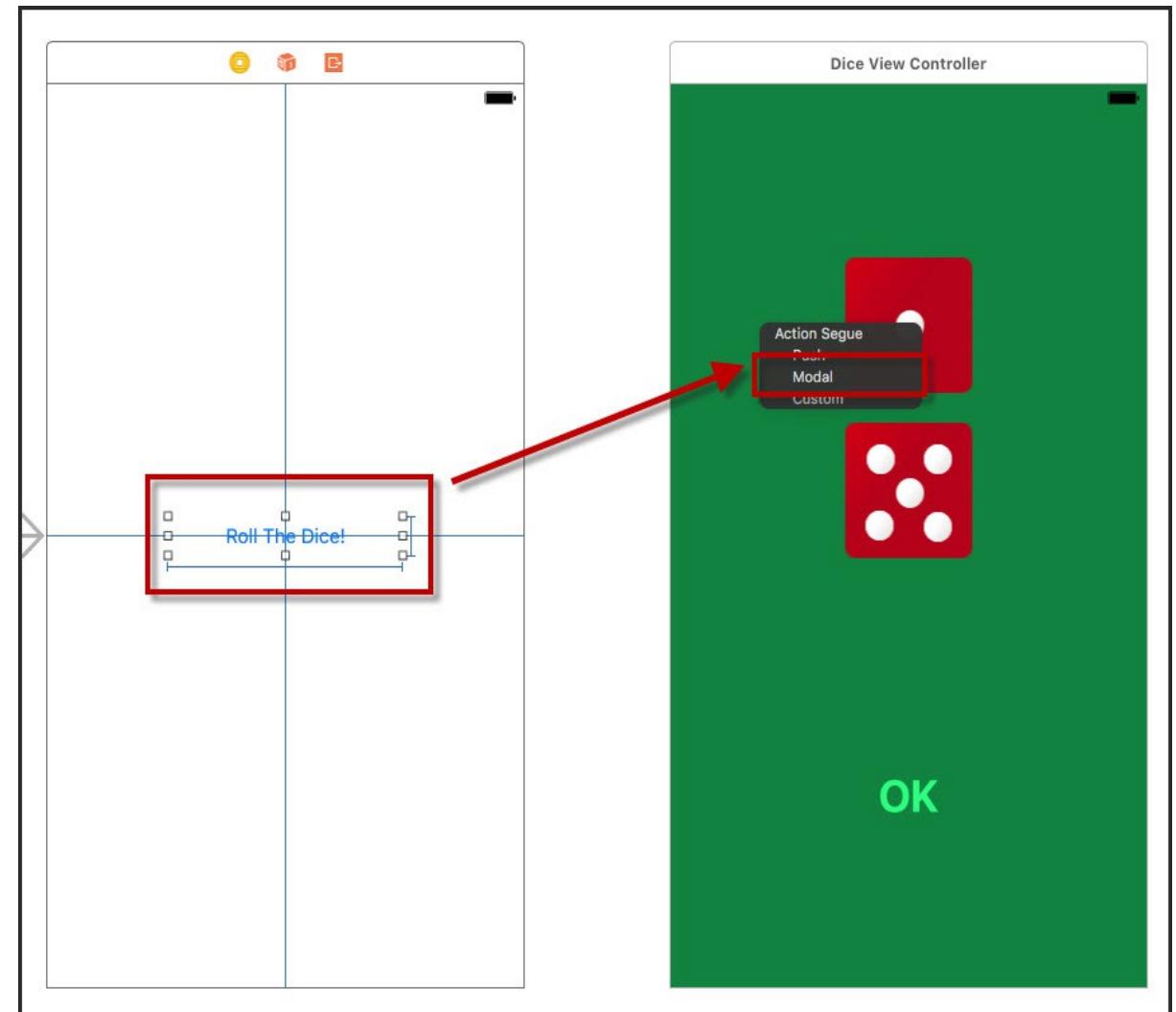
버튼 액션 함수에 performSegue 구문 추가
- 방금 지정한 identifier 인자로 넘김

```
@IBAction func rollTheDice(){
    performSegue(withIdentifier: "rollDice", sender: self)
}
```

Nav 가 아닌 다른방식 ?

Segue Only

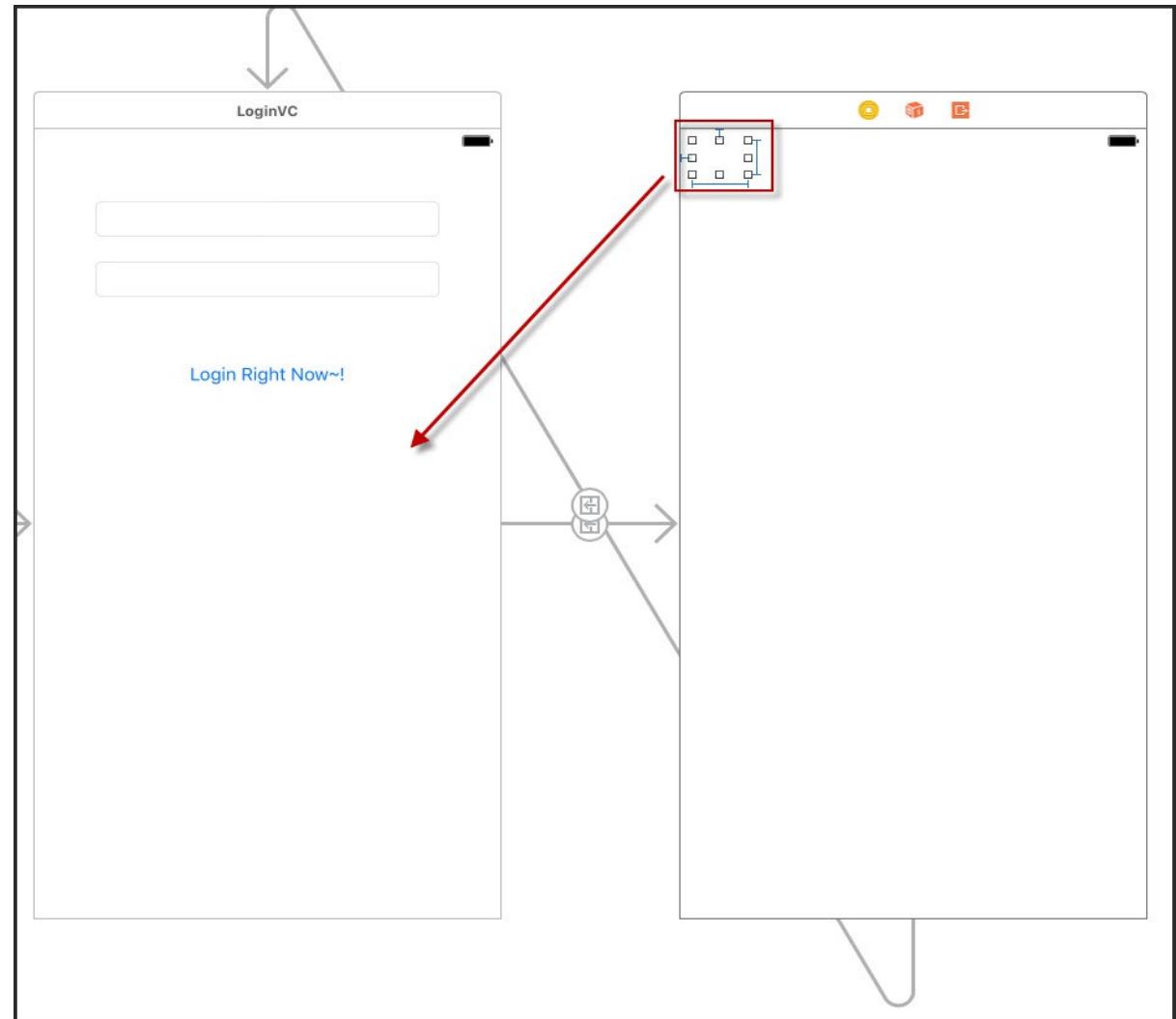
버튼에 직접 Action
Segue 추가



Nav 가 아닌 다른방식 ?

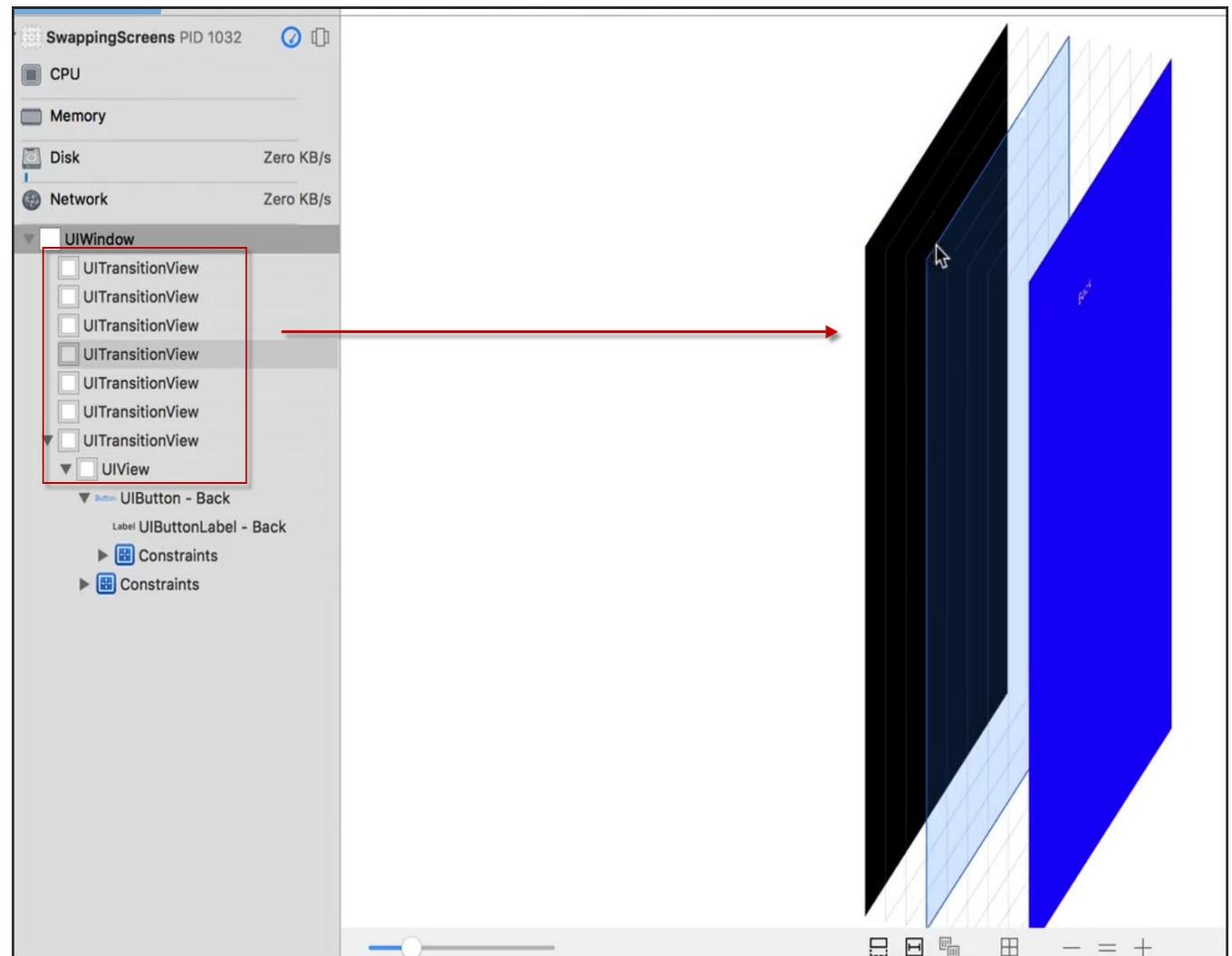
문제

다음과 같이 양쪽의 View Controller 에 각각 존재하는 버튼들을 이용해 서로 반대방향으로 뷰를 전환할 수 있을 때 발생할 수 있는 문제점은 무엇일까 ?



Nav 가 아닌 다른방식 ?

Debug View
Hierarchy



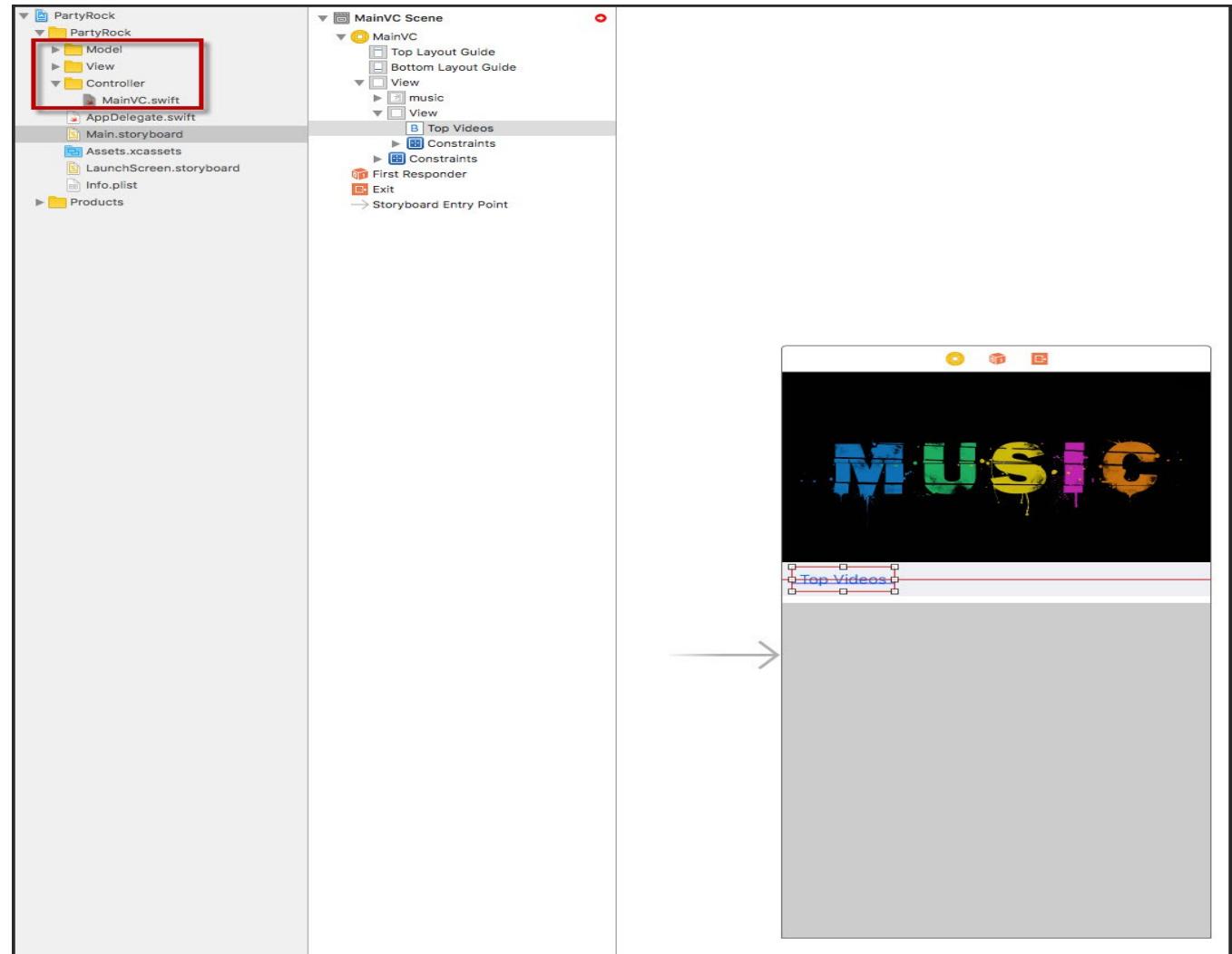
프로젝트

3. Music App

- Autolayout
- MVC 패턴
- 화면 전환

프로젝트

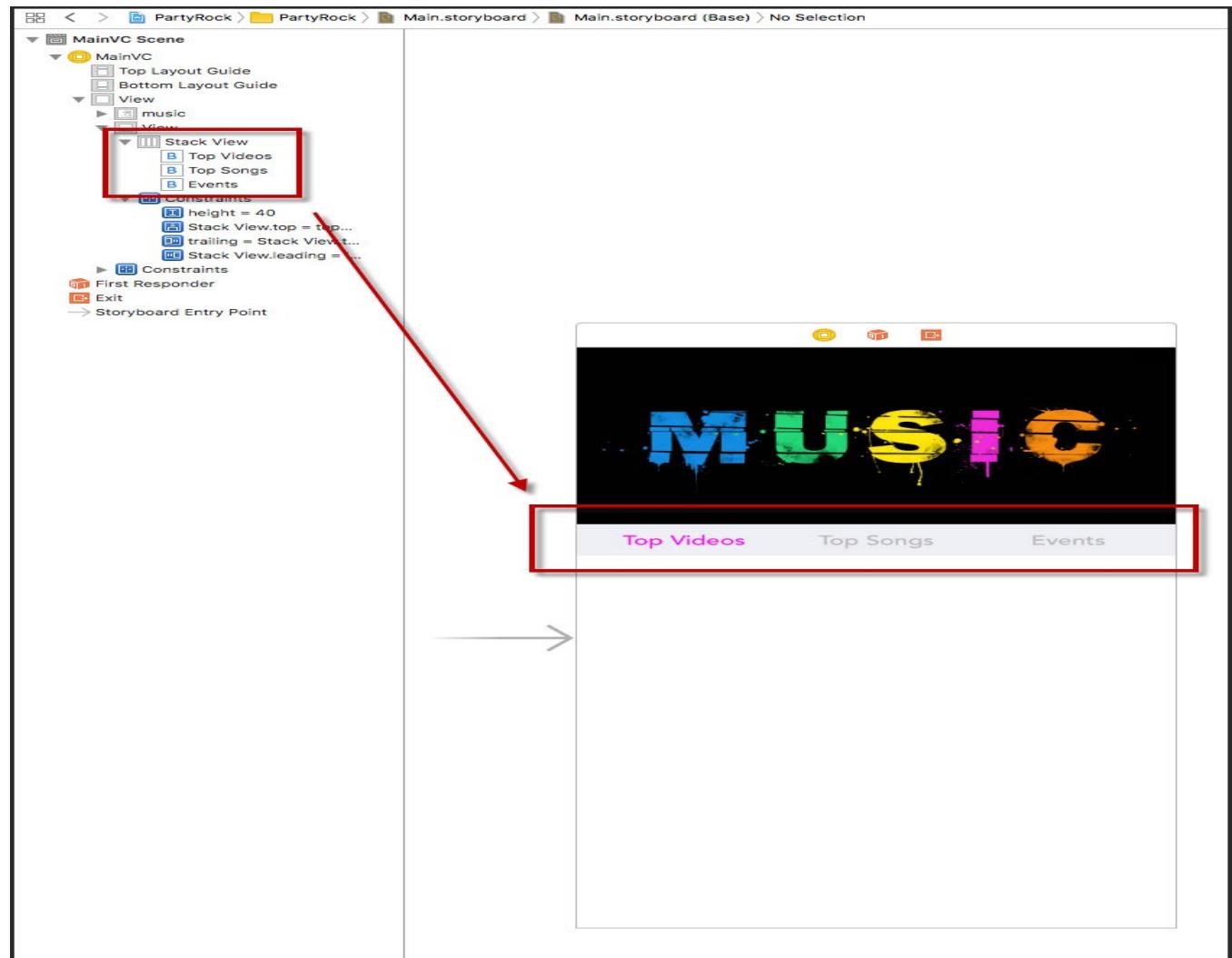
MVC 구조 개설



프로젝트

AutoLayout

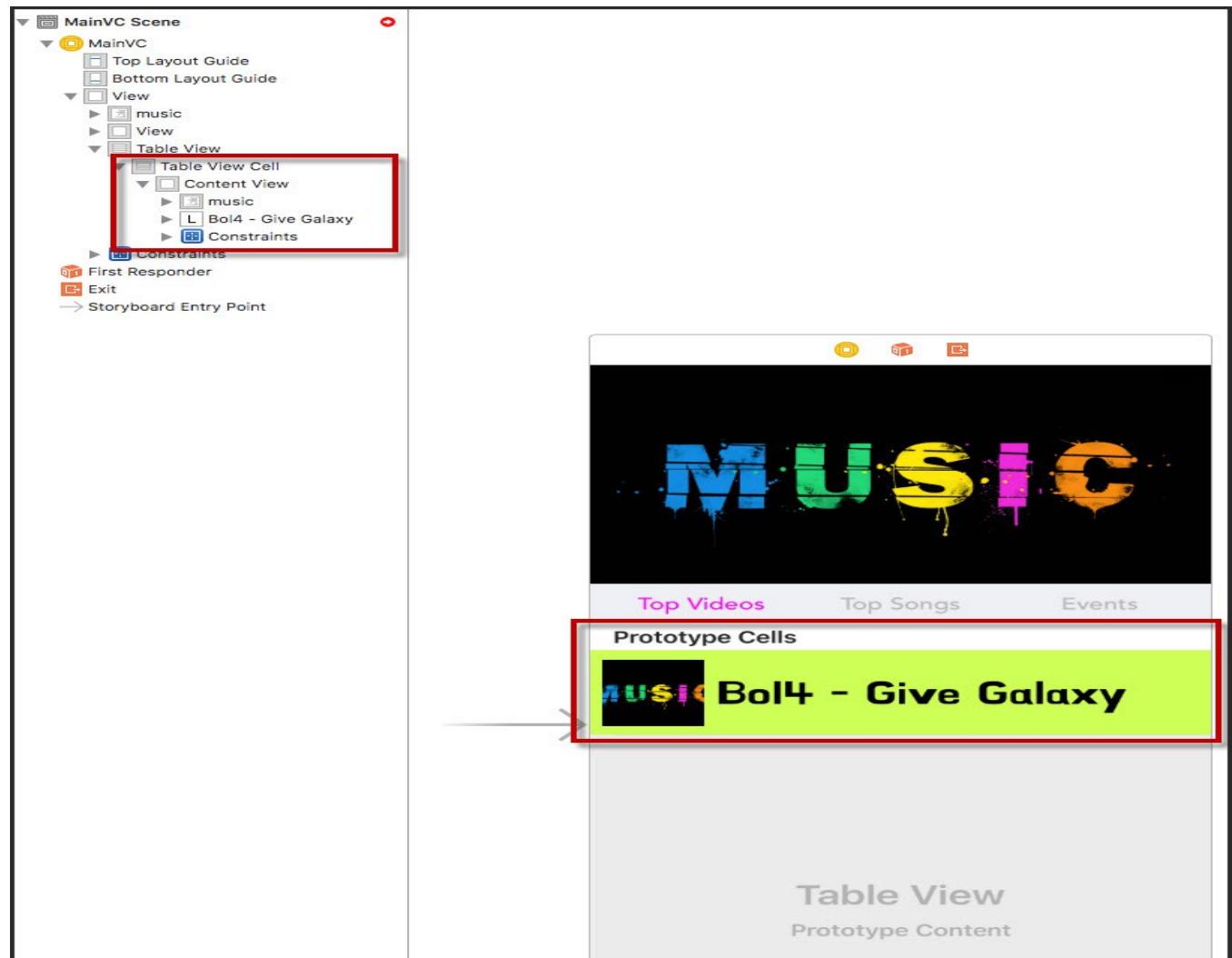
- 기본 배치를 위한 Constraint 설정
- Stack View 사용으로 버튼 정렬



프로젝트

AutoLayout

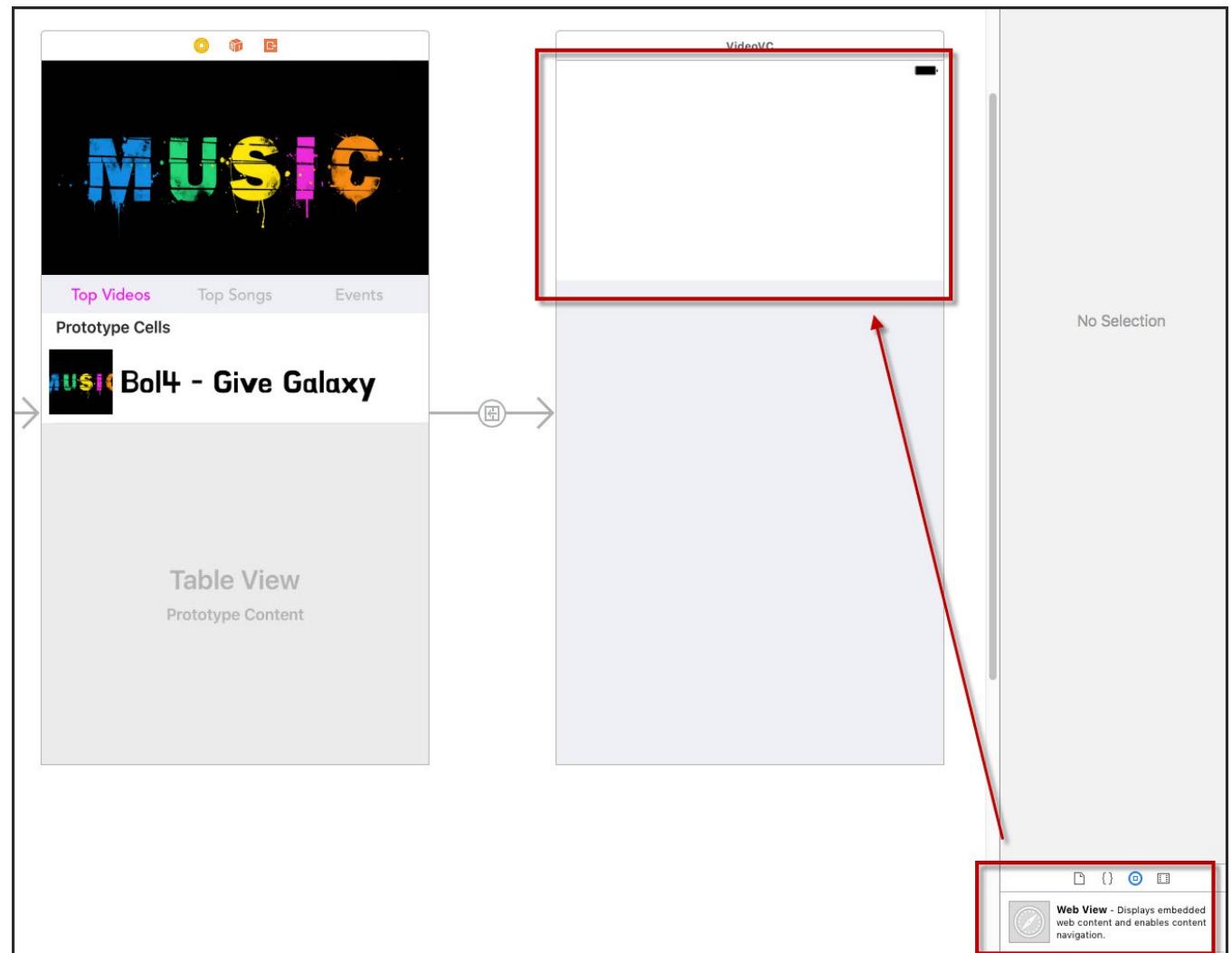
- TableView Cell
내용 추가



프로젝트

AutoLayout

- 세그웨이 추가
- Youtube 동영상을 가져올 WebView



프로젝트

MVC의 View

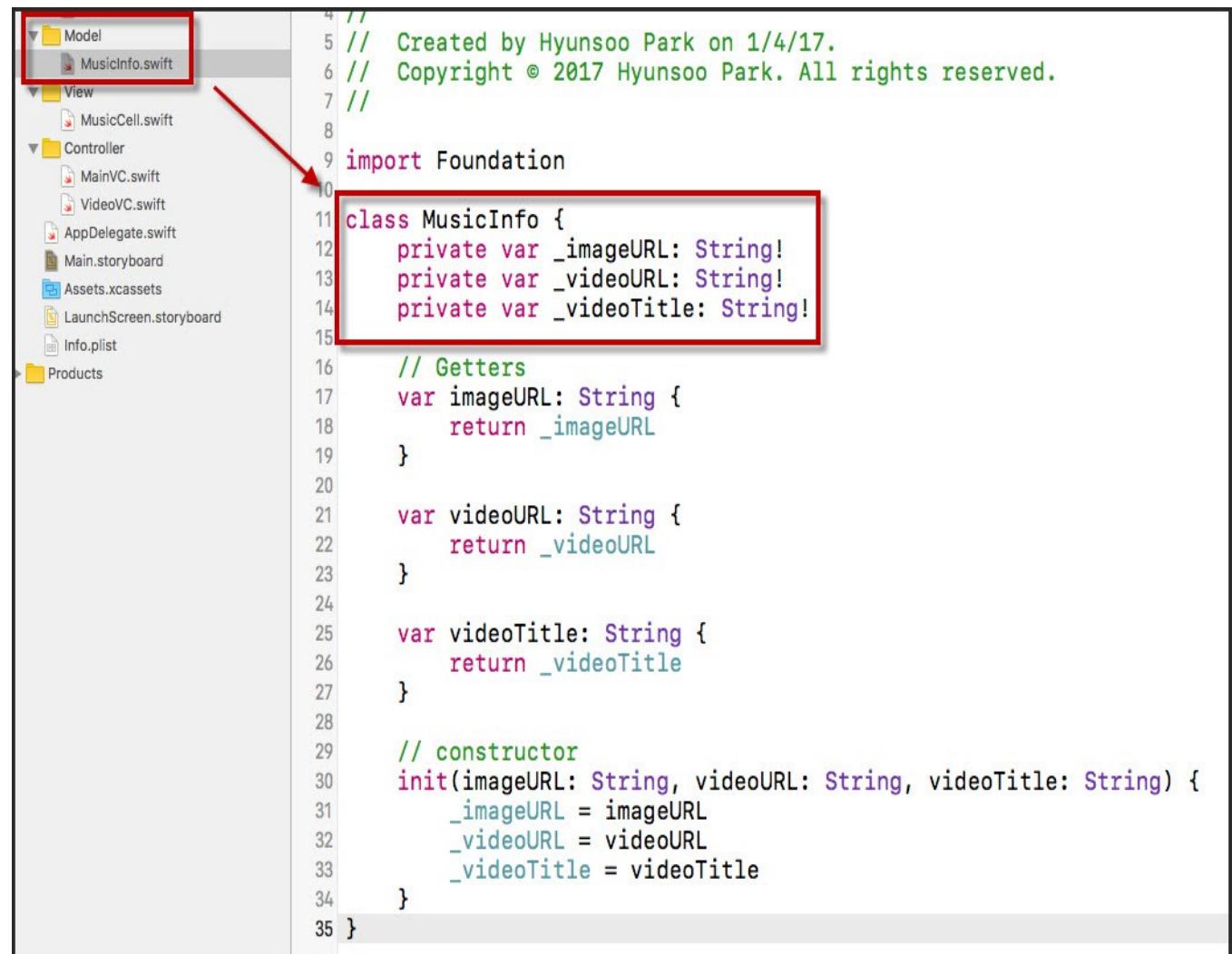
- Table 용 Cell 추가
 - » image
 - » title



프로젝트

MVC의 Model

- 곡 한 개체를 구성
할 데이터 Model
 - » imageURL
 - » videoURL
 - » title



The screenshot shows the Xcode interface. On the left is the Project Navigator with a tree view of files. A red box highlights the 'Model' folder, which contains 'MusicInfo.swift'. An arrow points from this highlighted area to the code editor on the right. The code editor displays the 'MusicInfo.swift' file, also with a red box highlighting its contents. The code defines a class 'MusicInfo' with private instance variables for imageURL, videoURL, and videoTitle, and corresponding public getters. It also includes a constructor that initializes these variables.

```
4 // Created by Hyunsoo Park on 1/4/17.  
5 // Copyright © 2017 Hyunsoo Park. All rights reserved.  
6 //  
7  
8  
9 import Foundation  
10  
11 class MusicInfo {  
12     private var _imageURL: String!  
13     private var _videoURL: String!  
14     private var _videoTitle: String!  
15  
16     // Getters  
17     var imageURL: String {  
18         return _imageURL  
19     }  
20  
21     var videoURL: String {  
22         return _videoURL  
23     }  
24  
25     var videoTitle: String {  
26         return _videoTitle  
27     }  
28  
29     // constructor  
30     init(imageURL: String, videoURL: String, videoTitle: String) {  
31         _imageURL = imageURL  
32         _videoURL = videoURL  
33         _videoTitle = videoTitle  
34     }  
35 }
```

프로젝트

Controller

- UITableView에서
delegate 함으로써
Table을 만들어둔
View의 Cell로 채워
나감

The screenshot shows a Xcode interface. On the left, a storyboard preview displays a table view with a single visible cell. The cell contains the text "Bo14 - Give Galaxy". On the right, the corresponding Swift code for the MainVC controller is shown:

```
3 // PartyRock
4 //
5 // Created by Hyunsoo Park on 1/4/17.
6 // Copyright © 2017 ed.
7 //
8 import UIKit
9
10 class MainVC: UIViewController, UITableViewDelegate, UITableViewDataSource {
11     @IBOutlet weak var tableView: UITableView!
12
13     var musicList = [MusicInfo]()
14
15     override func viewDidLoad() {
16         super.viewDidLoad()
17
18         tableView.delegate = self
19         tableView.dataSource = self
20     }
21
22     // there is new table cell for u, how do you wanna recycle it?
23     // Asks the data source for a cell to insert in a particular location of the table
24     func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
25
26         // Create a new reusable cell for us
27         if let cell = tableView.dequeueReusableCell(withIdentifier: "MusicCell", for: indexPath) as? MusicCell {
28             let musicInfo = musicList[indexPath.row]
29             cell.updateUI(musicInfo: musicInfo)
30             return cell
31         } else {
32             return UITableViewCell()
33         }
34     }
35
36     //tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int
37     func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
38         return musicList.count
39     }
40
41
42
43
44 }
```

Annotations explain specific parts of the code:

- UITableViewController의 기능을 대신 사용 (Line 10)
- UITableView 객체를 위한 데이터 모델을 재사용하기 위한 프로토콜 (Line 11)
- 테이블에서 특정 인덱스에 재활용되는 셀 생성 (Line 26)
- 테이블에 몇개의 셀이 존재하는가 (Line 39)

프로젝트

Data

- MockData 생성

```
override func viewDidLoad() {
    super.viewDidLoad()

    let b1 = MusicInfo(imageURL: "https://i.ytimg.com/vi/9U8uA702xrE/hqdefault.j
        Lo", videoURL: "<iframe width=\"560\" height=\"315\" src=\"https://www.y
        V] 우주를 줄게 - 볼빨간사춘기\")

    let b2 = MusicInfo(imageURL: "https://i.ytimg.com/vi/1eRtWW3xbleE/hqdefault.j
        custom=true&w=196&h=110&stc=true&jpg444=true&jpgq=90&sp=68&sigh=I0ohNAQH
        www.youtube.com/embed/1eRtWW3xbleE\" frameborder=\"0\" allowfullscreen></i

    let b3 = MusicInfo(imageURL: "https://i.ytimg.com/vi/pLuETPoCKRo/hqdefault.j
        custom=true&w=196&h=110&stc=true&jpg444=true&jpgq=90&sp=68&sigh=4UEqhiK7
        www.youtube.com/embed/pLuETPoCKRo\" frameborder=\"0\" allowfullscreen></i

    let b4 = MusicInfo(imageURL: "https://i.ytimg.com/vi/MfYPKZl7W1w/hqdefault.j
        custom=true&w=168&h=94&stc=true&jpg444=true&jpgq=90&sp=68&sigh=Eh7KkhdQH
        www.youtube.com/embed/MfYPKZl7W1w\" frameborder=\"0\" allowfullscreen></i

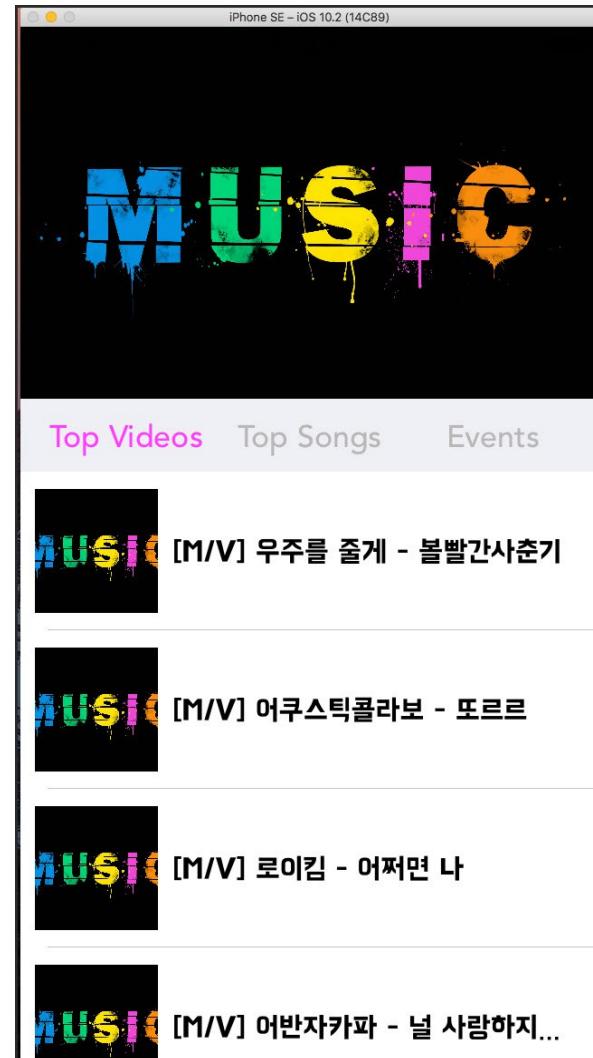
    let b5 = MusicInfo(imageURL: "https://i.ytimg.com/vi/JzociagwnBw/hqdefault.j
        Q", videoURL: "<iframe width=\"560\" height=\"315\" src=\"https://www.y
        1집 - 안아줘\")

    musicList.append(b1)
    musicList.append(b2)
    musicList.append(b3)
    musicList.append(b4)
    musicList.append(b5)
```

프로젝트

구동 후 화면

- image 업데이트 필요



프로젝트

Update image

- 비동기 스레드를 사용해 웹에 있는 이미지를 불러옴
- 이미지를 불러오면 메인 UI 스레드로 이미지를 업데이트

The screenshot shows the Xcode interface. On the left is the project navigator with files like View, MusicCell.swift, Controller, MainVC.swift, VideoVC.swift, AppDelegate.swift, Main.storyboard, Assets.xcassets, LaunchScreen.storyboard, Info.plist, and Products. The main area shows a Swift code editor for MusicCell.swift.

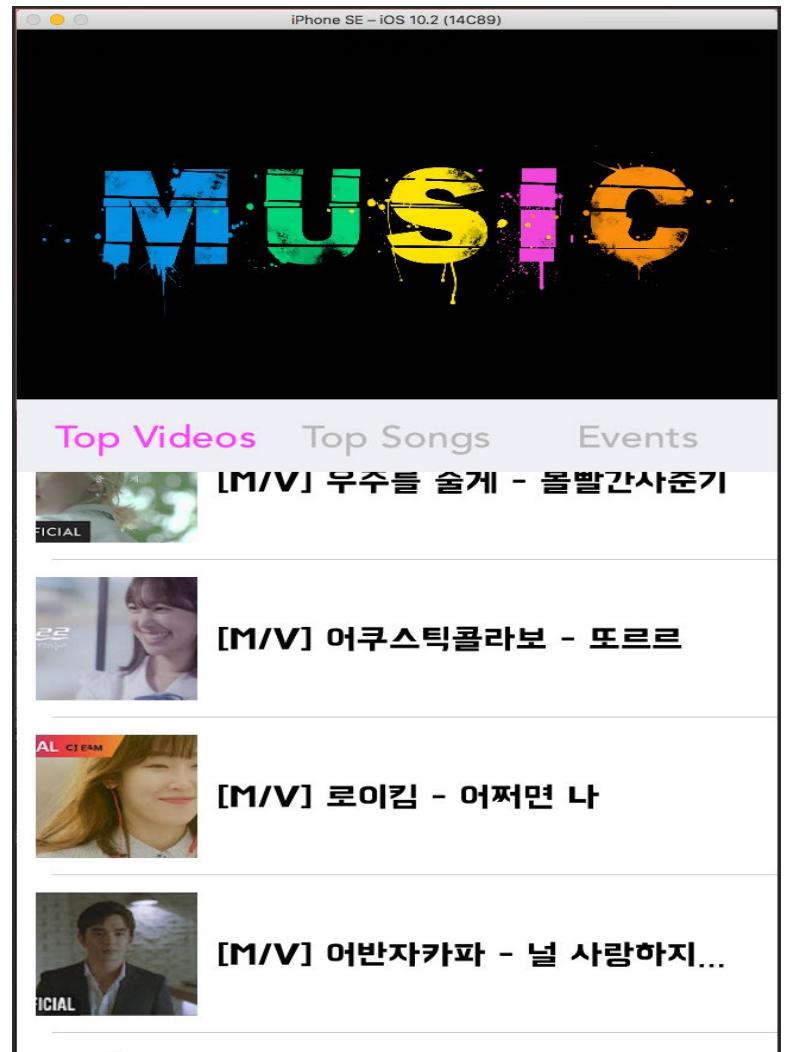
```
// Copyright © 2017 hyunsoo Park. All rights reserved.  
//  
import UIKit  
  
class MusicCell: UITableViewCell {  
  
    @IBOutlet weak var videoPreviewImage: UIImageView!  
    @IBOutlet weak var videoTitle: UILabel!  
  
    override func awakeFromNib() {  
        super.awakeFromNib()  
    }  
  
    func updateUI(musicInfo: MusicInfo) {  
        videoTitle.text = musicInfo.videoTitle  
    }  
  
    let url = URL(string: musicInfo.imageURL)!  
  
    // gives u a certain async thread that can load images asynchronously  
    DispatchQueue.global().async {  
        do {  
            let data = try Data(contentsOf: url)  
            DispatchQueue.global().sync { // for updating user interface  
                self.videoPreviewImage.image = UIImage(data: data)  
            }  
        } catch {  
        }  
    }  
}
```

A callout box highlights the line `let url = URL(string: musicInfo.imageURL)!` with the text "url을 통해 웹에서 이미지를 비동기적으로 불러옴". Another callout box highlights the DispatchQueue.async block with the text "비동기 스레드로는 UI를 업데이트하지 못함 => 메인 UI 스레드로 전환하여 프리뷰 이미지를 업데이트".

프로젝트

구동 후 화면

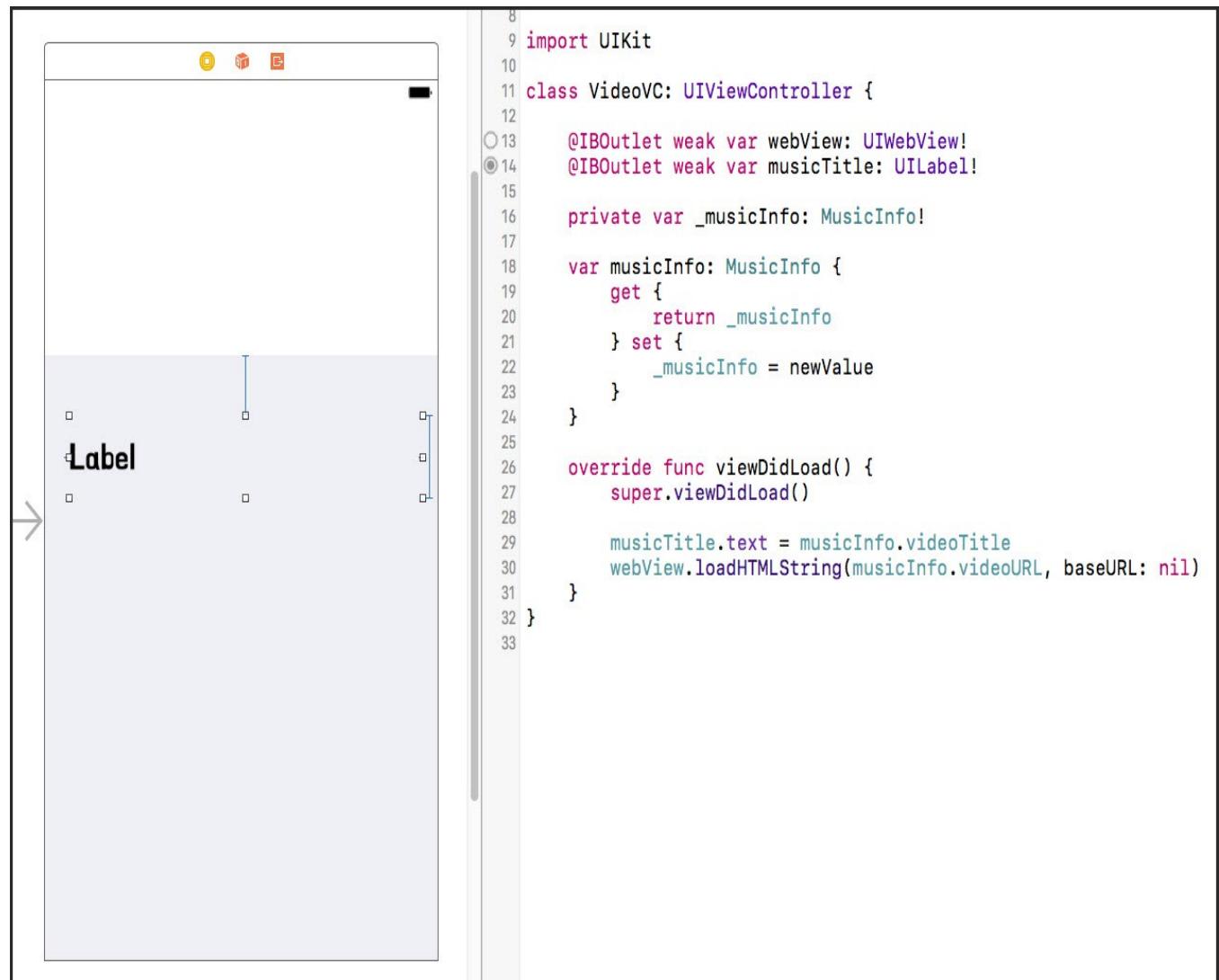
- 이미지 업데이트



프로젝트

음악 가져오기

- WebView 에 HTML 불러옴
- Label 에 곡 제목을 불러옴

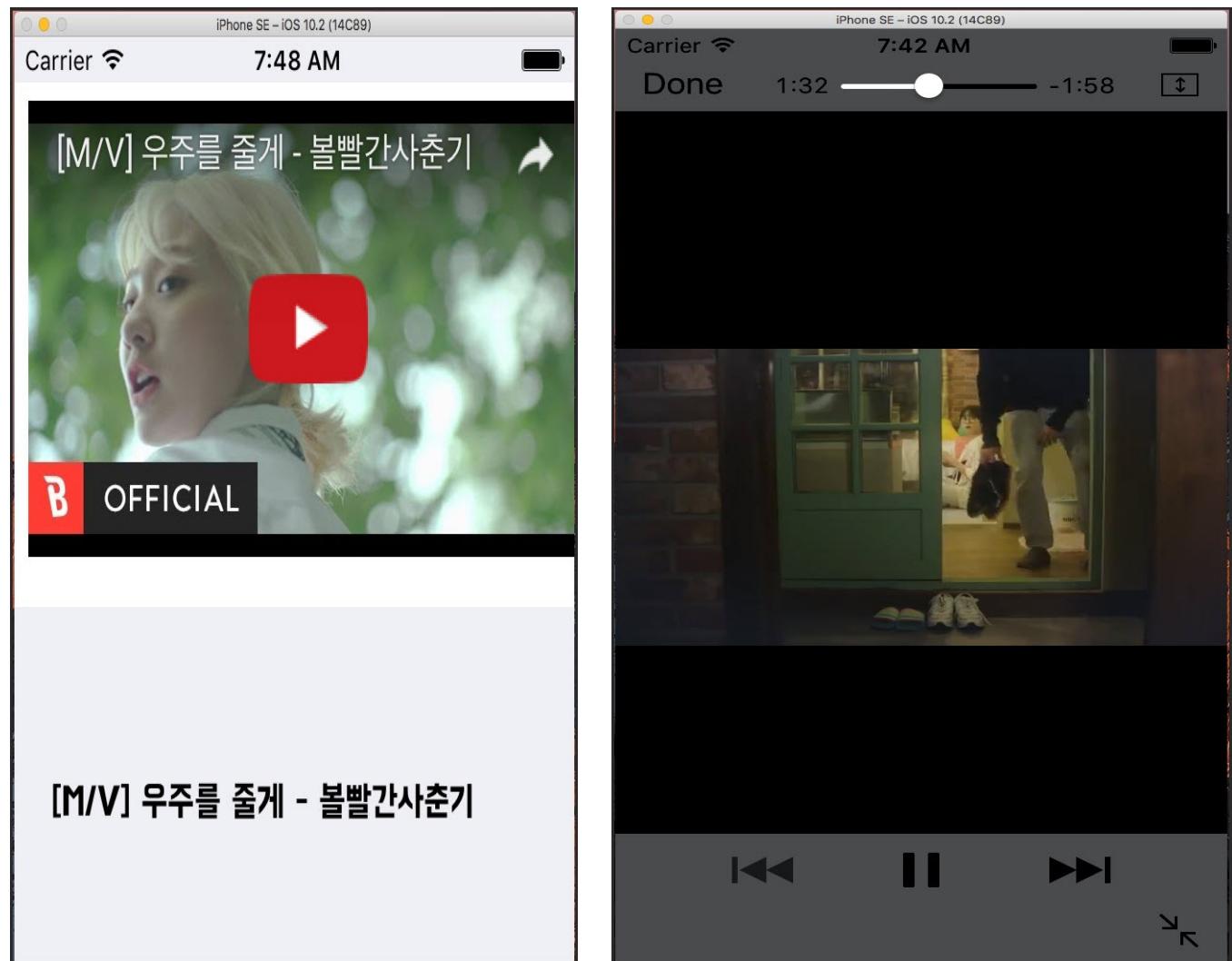


The image shows a screenshot of the Xcode IDE. On the left, a storyboard preview window displays a single view controller with a light gray background. Inside this view, there is a single label component with the text "Label". On the right side of the interface, the Swift code for the view controller is visible:

```
8 import UIKit
9
10 class VideoVC: UIViewController {
11
12
13 @IBOutlet weak var webView: UIWebView!
14 @IBOutlet weak var musicTitle: UILabel!
15
16 private var _musicInfo: MusicInfo!
17
18 var musicInfo: MusicInfo {
19     get {
20         return _musicInfo
21     } set {
22         _musicInfo = newValue
23     }
24 }
25
26 override func viewDidLoad() {
27     super.viewDidLoad()
28
29     musicTitle.text = musicInfo.videoTitle
30     webView.loadHTMLString(musicInfo.videoURL, baseURL: nil)
31 }
32 }
33 }
```

프로젝트

구동 후 화면



감사합니다
(ios 박현수)