iOS alapú szoftverfejlesztés - Labor 06

A laborsegédletet összeállította: Kelényi Imre - imre.kelenyi@aut.bme.hu és Kántor Tibor - kantor.tibor@bmeautsoft.hu

A labor témája:

* Dinamikus Table View
* Statikus Table View
* Dinamikus cellamagasság
* Table View szerkesztése: cellák törlése, mozgatása, új cellák felvétele
* Image Picker, képek kiválasztása galériából

A labor célja egy utazásokat nyilvántartó alkalmazás megírása és ezzel együtt a Table View használatának begyakorlása.

A laborhoz tartozó nagyobb kódrészletek a következő url-en érhetők el "copy-paste barát" formában:

https://gist.github.com/DjCantor/4ee9d62fdbc5475d4077

# iTravel

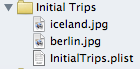
## Kezdeti adatok betöltése

Hozzunk létre egy "Single View" application-t "iTravel" névvel iPhone-ra.

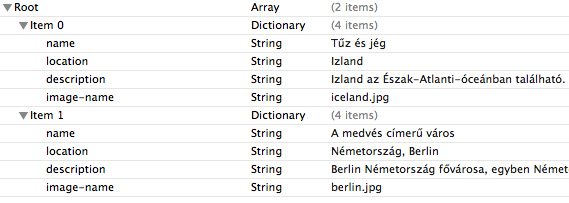
Hozzunk létre egy új, "TripsDataManager" nevű osztályt, melyben az utazások adatait fogjuk tárolni.

Az egyes utak adatai egy-egy Dictionary-ben vannak eltárolva. Hogy mindig legyen pár adatunk a teszteléshez, egy Property List-ből betöltünk pár kezdeti értéket. Töltsük le a következő ZIP fájlt, majd a benne lévő könyvtárat ("InitialTrips") adjuk hozzá a projekthez:

**https://dl.dropbox.com/u/152439/aut-ios/iTravel-InitialTrips.zip**



A Property List-et megnyitva megnézhetjük a kezdeti adatokat:



Az osztálynak egyetlen property-je lesz, mely egy NSMutableArray-ben tárolja az utak adatait, vegyük fel:

var trips : [AnyObject]

Írjuk meg az alap inicializálót, mely betölti a kezdeti utakat a Property List-ből:

override init(){

let filePath = NSBundle.mainBundle().pathForResource("InitialTrips", ofType: "plist")

trips = NSArray(contentsOfFile: filePath!)! as [AnyObject]

super.init()

}

Hozzunk létre egy property-t TripsDataManager-ből az Application Delegate-ben és példányosítsuk is!

let tripsDataMgr = TripsDataManager()

Hozzunk létre egy, az App Delegate elérését megkönnyítő osztály metódust is:

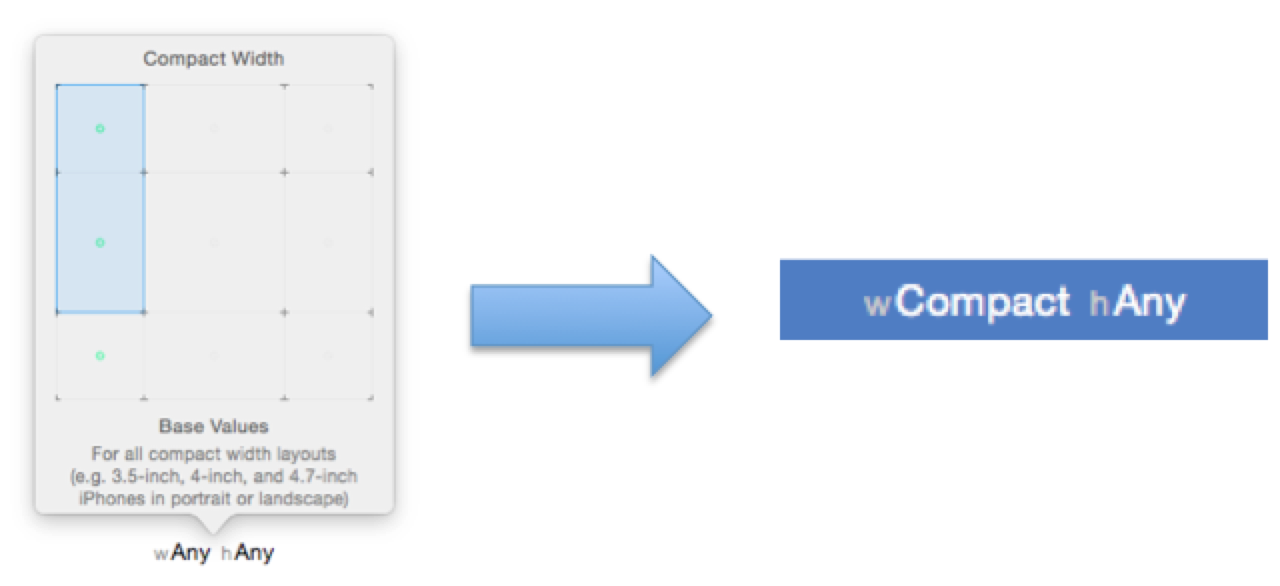
class func sharedAppDelegate() -> AppDelegate {

return UIApplication.sharedApplication().delegate as! AppDelegate

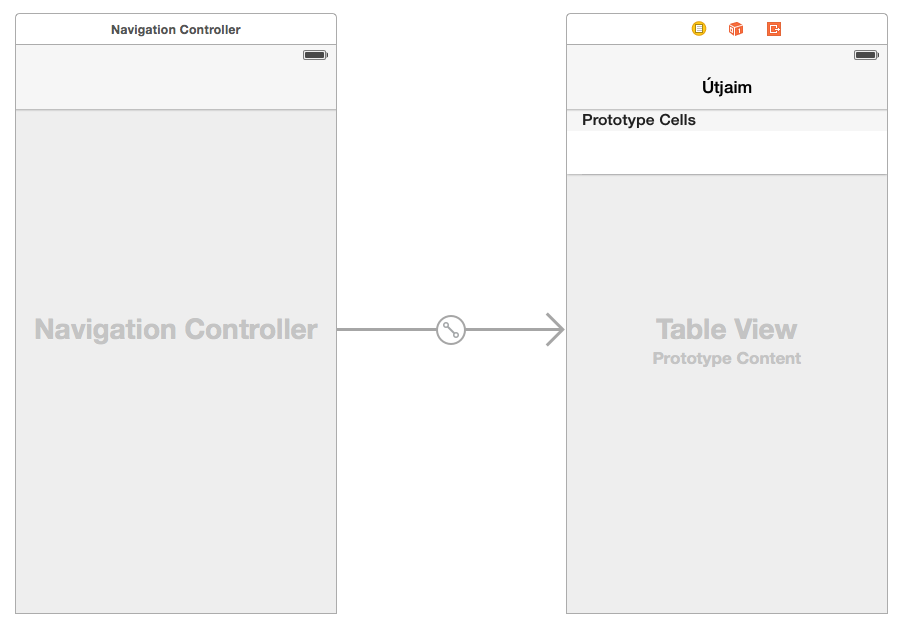
}

## Utak listájának megjelenítése: dinamikus Table View

Térjünk át a Storyboard-hoz! Mivel az alkalmazást iPhonera szánjuk így váltsunk át wCompact | hAny méretosztályba:

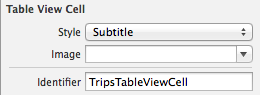


Rakjunk be egy Table View Controller-t! Ágyazzuk be ezt a Table View Controller-t egy Navigation Controllerbe és állítsuk be a hozzá tartozó Navigation Item Title property-jét (pl. "Útjaim"-ra):



Töröljük le a forrásfájlok közül a legenerált "ViewController"-t (Move to Trash), majd hozzunk létre egy új, TableViewController-ből származó osztályt, "TripsViewController" névvel. Állítsuk be a storyboardban a Table View Controller osztályát az újonnan definiált osztályra!

Állítsuk át a Table View Controller-ben lévő Table View Cell stílusát "Subtitle"-re és adjunk meg neki egy azonosítót: "TripsTableViewCell".



ITTripsViewController-hez adjunk hozzá egy tagváltozót, melyen keresztül elérhetjük az utak adatait:

var tripsDataManager : TripsDataManager?

Ezt inicializáljuk a viewDidLoad metódusban:

tripsDataManager = AppDelegate.sharedAppDelegate().tripsDataMgr

Valósítsuk meg a Table View Data Source metódusait:

override func numberOfSectionsInTableView(tableView: UITableView) -> Int {

return 1

}

override func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {

return tripsDataManager!.trips.count

}

override func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -> UITableViewCell {

let cell: UITableViewCell = tableView.dequeueReusableCellWithIdentifier("TripsTableViewCell") as UITableViewCell

let tripData = tripsDataManager!.trips[indexPath.row] as [String:AnyObject]

cell.textLabel?.text = tripData["name"] as? String

cell.detailTextLabel?.text = tripData["location"] as? String

if let tripImageName : String = tripData["image-name"] as? String {

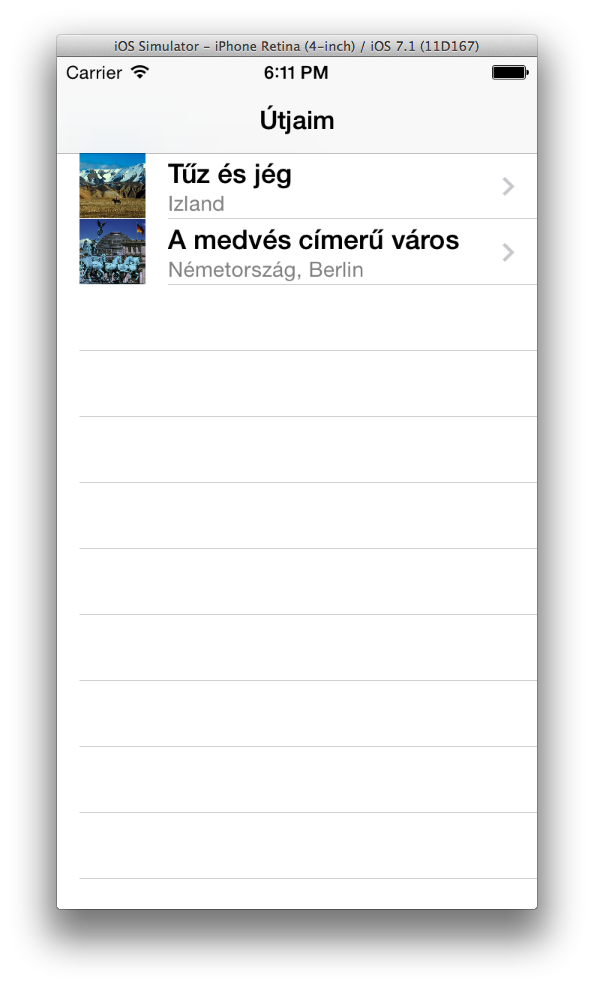
cell.imageView?.image = UIImage(named: tripImageName)

}

return cell

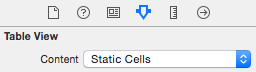
}

https://gist.github.com/DjCantor/4ee9d62fdbc5475d4077#file-cellforrowatindexpath



## Utak részletezése: statikus Table View

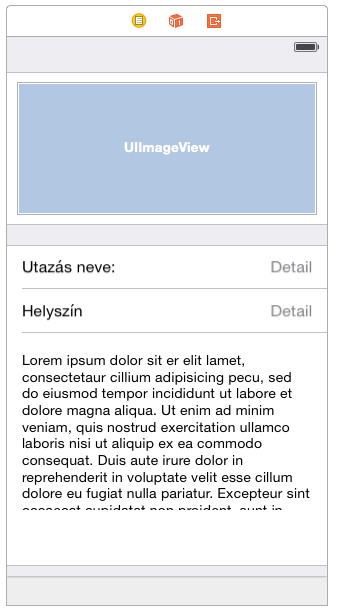
Hozzunk létre egy új Table View Controller-t majd állítsuk át a benne foglalt Table View "Content" property-jét "Static Cells"-re:



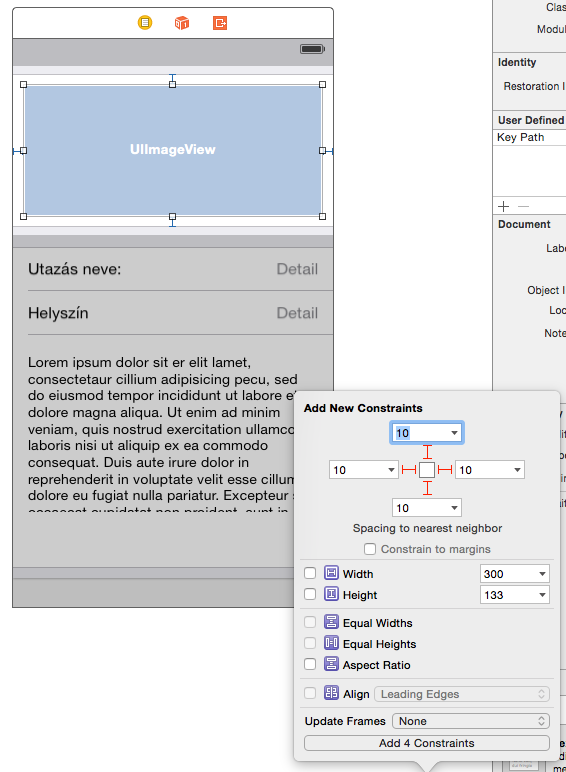
Állítsuk be a Sections-t 2-re, a Style-t pedig "Grouped"-ra:



Ezek után módosítsuk a Table View-t oly módon, hogy az első szekcióban 1 "Custom", a másodikban pedig 2 "Right Detail" és 1 "Custom" stílusú cella legyen (az egyes cellák Style property-jét kell állítgatni). Az első szekció cellájába adjunk hozzá egy Image View-t, a második szekció utolsó cellájához pedig egy Text View-t. Módosítsuk a feliratokat, hogy a következő képhez hasonló felületet kapjunk (a Text View cellájának magassága lehet nagyobb is):



Bár ha nem adunk meg a nézetben egyetlen auto layout kényszert, akkor is működni fog, az elegáns megoldás, ha mint az Image View-t, mind a Text View-t rögzítjük a szülő nézeteikhez néhány kényszer megadásával (pl. 10 távolság a szülő összes oldalától):



Az Image View "Mode" property-ját állítsuk "Aspect Fit"-re:



A Text View-nak kapcsoljuk ki az Editable property-jét és állítsuk átlátszóra a háttérszínét:





A 4 cellát kiválasztva a "Selection" property értékét állítsuk "None"-ra, hogy ne lehessen kiválasztani a cellákat:



Hozzunk létre egy új, UITableViewController-ből leszármazó osztályt, "TripDetailViewController" névvel és állítsuk be ezt a storyboard-ban létrehozott View Controller osztályául.

Vegyünk fel Outlet-eket a 4, adatmegjelenítésre szolgálló nézethez (minden cellából azt a nézetet válasszuk ki, mely az egyedi adat megjelenítésére szolgál, és ezeket kössük be Outletekre):

@IBOutlet weak var tripImageView: UIImageView!

@IBOutlet weak var tripNameLabel: UILabel!

@IBOutlet weak var tripLocationLabel: UILabel!

@IBOutlet weak var tripDescriptionTextView: UITextView!

Vegyünk fel egy trip nevű, Dicrionary típusú property-t TripDetailViewController-be, ez fogja tárolni az éppen kiválasztott út adatait:

var trip : [String:AnyObject]?

Definiáljuk fel a viewWillAppear: metódust, melyben beállítjuk a nézetek kezdeti értékét a trip property-ben eltárolt utazás alapján:

override func viewWillAppear(animated: Bool) {

super.viewWillAppear(animated)

if trip != nil {

if let tripImageName = trip!["image-name"] as? String{

tripImageView.image = UIImage(named: tripImageName)

}

tripNameLabel.text = trip!["name"] as? String

tripLocationLabel.text = trip!["location"] as? String

tripDescriptionTextView.text = trip!["description"] as? String

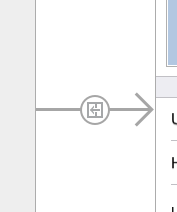
navigationItem.title = trip!["name"] as? String

}

}

https://gist.github.com/DjCantor/4ee9d62fdbc5475d4077#file-viewwillappear

A storyboard.ban kössük be egy Push segue-el a Trips View Controller-ben lévő Table View prototípus celláját a Detail View Controller-re. Nevezzük el a segue-t "TripDetailSegue"-nek:





Ezek után definiáljuk felül TripsViewController-ben a prepareForSegue: metódust, hogy átadhassa az éppen kiválasztott cellához tartozó utazás adatait a megjelenő Trip Detail View Controller-nek:

override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {

if segue.identifier == "TripDetailSegue" {

let vc = segue.destinationViewController as! TripDetailsViewController

let row = tableView.indexPathForSelectedRow?.row

vc.trip = tripsDataManager!.trips[row!] as? [String:AnyObject]

}

}

Teszteljük az alkalmazást!

## Cellák törlése és mozgatása

Ahhoz, hogy a Table View celláit törölni vagy mozgatni tudjuk, a Table View-t Edit módba kell kapcsolni. Szerencsére ennek elvégzésére van egy beépített "Edit gomb" a Table View Controller-ben, ezt csak hozzá kell adnunk például a Navigation Bar-hoz. Adjuk a következő sort TripsViewController viewDidLoad: metódusához:

navigationItem.rightBarButtonItem = editButtonItem()

Először is engedélyeznünk kell, hogy az egyes soros szerkeszthetők legyenek:

override func tableView(tableView: UITableView, canEditRowAtIndexPath indexPath: NSIndexPath) -> Bool {

return true

}

Majd pedig el kell végeznünk a törlést:

override func tableView(tableView: UITableView, commitEditingStyle editingStyle: UITableViewCellEditingStyle, forRowAtIndexPath indexPath: NSIndexPath) {

if editingStyle == .Delete{

tripsDataManager!.trips.removeAtIndex(indexPath.row)

tableView.deleteRowsAtIndexPaths([indexPath], withRowAnimation: .Fade)

}

}

https://gist.github.com/DjCantor/4ee9d62fdbc5475d4077#file-commiteditingstyle

A cellák mozgatásának engedélyezése:

override func tableView(tableView: UITableView, canMoveRowAtIndexPath indexPath: NSIndexPath) -> Bool {

return true

}

A mozgatás folyamata, mely során az adatforrásként használt tömbben felcserélünk két elemet:

override func tableView(tableView: UITableView, moveRowAtIndexPath sourceIndexPath: NSIndexPath, toIndexPath destinationIndexPath: NSIndexPath) {

let tripToMove = tripsDataManager!.trips[sourceIndexPath.row] as [String:String]

tripsDataManager!.trips.removeAtIndex(sourceIndexPath.row)

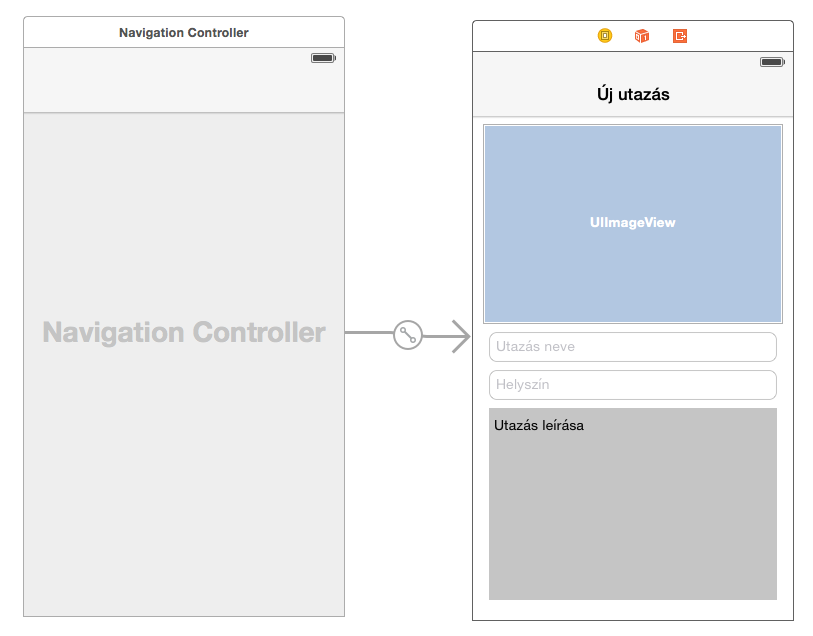
tripsDataManager!.trips.insert(tripToMove, atIndex: destinationIndexPath.row)

}

https://gist.github.com/DjCantor/4ee9d62fdbc5475d4077#file-moverowatindexpath

## Új cella hozzáadása

Vegyünk fel egy új View Controller-t a storyboard-ba és ágyazzuk be rögtön egy szintén újonnan létrehozott Navigation Controllerbe. A cél ezután az, hogy a következő felületet állítsuk össze:



* Vegyünk fel egy Image View-t és állítsuk be a háttérszínét ("Background" attribútum) valamilyen élénk színre
* Vegyünk fel két Text Field-et, és állítsunk be bennünk "Placeholder"-t ("Utazás neve" és "Helyszín")
* Vegyünk fel egy Text View-t és állítsuk át a háttérszínét, valamint a kezdeti szöveget pl. "Utazás leírása"-ra

Az idő szűkössége miatt nem definiálink auto layout kényszereket, de ne felejtsüjk, hogy így az alklamzás nem fog jól mutatni más kijelzőméreteken vagy ha elforgatjuk. Egy éles alkalmazásban ez nem elfogadható.

Húzzunk be 1-1 Bar Buttonemet a view ocntroller tetején lévő navigation barra, majd állítsuk ezek "Identifier" attributumát "Cancel"-re, illetve "Save"-re:





Hozzunk létre egy új, "EditTripViewController" nevű, UIViewController-ből származó osztályt és illesszük be (írjuk felül) a .h és a .m tartalmát a következő kódrészletekkel:

https://gist.github.com/DjCantor/4ee9d62fdbc5475d4077#file-edittripviewcontroller-swift

Végezzük el a következő módosításokat:

* Kössük be a kódban definiált Outletekre az Image View-t a 2 Text Field-et és a Text View-t
* Kössük be az alsó Text Field "Editing Did Begin" és "Editing Did End" eseményeit a "textFieldEditingDidBegin:" és a " textFieldEditingDidEnd:" metódusokra, továbbá a DidEndOnExit metódusra is!

Adjunk hozzá a Trips View Controller navigation bar-jához egy Bar Button Item-et, majd ennek "Identifier" attribútumát állítsuk "Add"-ra:



Kössünk rá az új gombra egy Modal segue-el az új navigation controllerrel (ami a szerkesztő view controllert tartalmazza). A segue-t nevezzük el "AddTripSegue"-nek, majd teszteljük az alkalmazást!

Amikor az egyelőre üres képre rákattint a felhasználó, meghívódik egy Tap gesztus kezelő metódus EditTripViewController-ben (handeImageViewTap:), itt szeretnénk megjeleníteni egy képkiválasztó rendszer view controller-t.

func handleImageViewTap(sender: UITapGestureRecognizer){

let pickerController = UIImagePickerController()

pickerController.delegate = self

pickerController.sourceType = .SavedPhotosAlbum

presentViewController(pickerController, animated: true, completion: nil)

}

Majd valósítsuk meg a "didFinishPickingMediaWithInfo:" metódust:

func imagePickerController(picker: UIImagePickerController, didFinishPickingMediaWithInfo info: [String : AnyObject]) {

let selectedImage = info[UIImagePickerControllerOriginalImage] as UIImage

imageView.image = selectedImage

dismissViewControllerAnimated(true, completion: nil)

}

A szerkesztő nézetből való kilépéshez 1-1 unwind segue-t fogunk használni. Ehhez először fel kell vennünk 1-1 speciális akció metódust abban a view controllerben, ahova "vissza szeretnénk térni" a szerkesztő nézet bezárásakor. Ez esetünk a Trips View Controller, úgyhogy ehhez adjuk a következő két metódust:

@IBAction func editTripViewControllerDidSave(unwindSegue:UIStoryboardSegue){

let vc = unwindSegue.sourceViewController as EditTripViewController

let trip : [String:AnyObject] = ["name" : vc.nameTextField.text, "location" : vc.locationTextField.text, "description" : vc.descriptionTextView.text, "image":vc.imageView.image!]

tripsDataManager?.trips.append(trip)

tableView.reloadData()

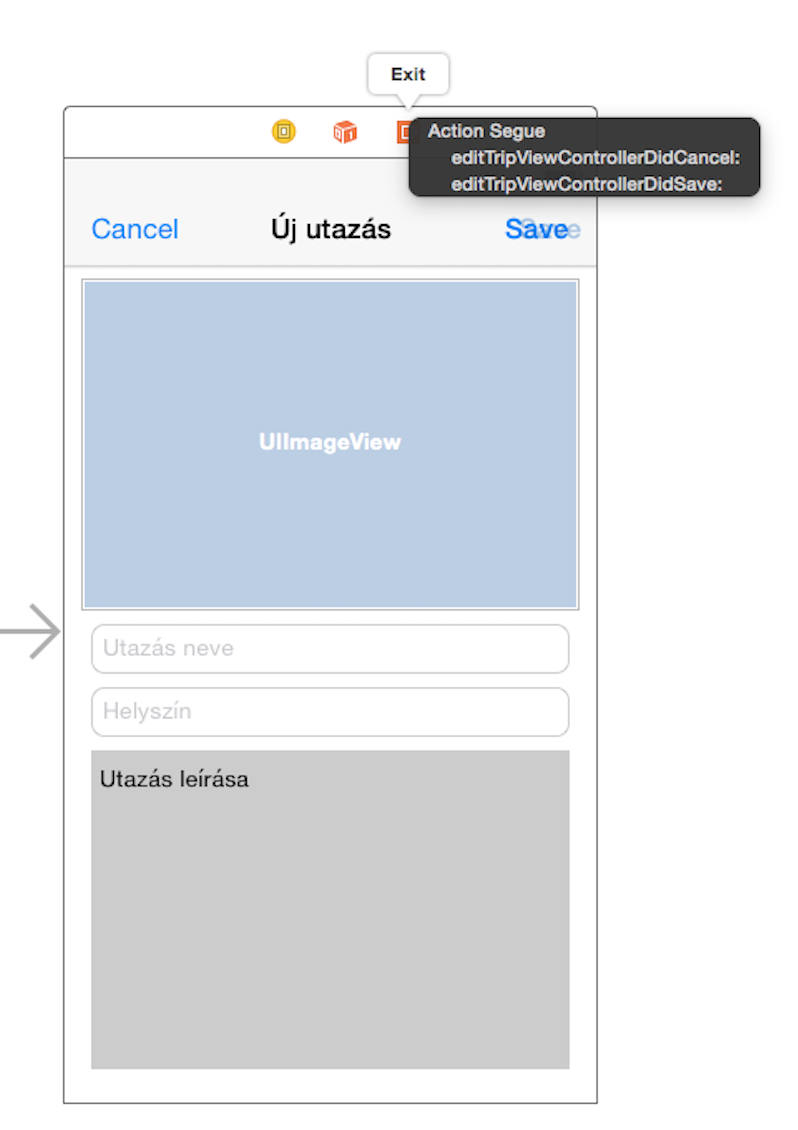
}

@IBAction func editTripViewControllerDidCancel(unwindSegue:UIStoryboardSegue){

}

https://gist.github.com/DjCantor/4ee9d62fdbc5475d4077#file-didsave-didcancel

A storyboardban kössük be a "Cancel" és a "Save" gombokat a kis Exit jelre és itt válasszuk ki Trips View Controller megfelelő metódusait:



Módosítsuk cellForRowAtIndexPath:-t, hogy az "image" kulcs esetén már egy kész képet olvasson ki az utazás adatait tároló dictionary-ból:

...

if let tripImageName : String = tripData["image-name"] as? String {

cell.imageView?.image = UIImage(named: tripImageName)

}

if let tripImage : UIImage = tripData["image"] as? UIImage {

cell.imageView?.image = tripImage

}

...

Hasonló módon ITTripDetailViewController-ben is módosítani kell a viewWillAppear: metódust:

...

if let tripImageName = trip!["image-name"] as? String{

tripImageView.image = UIImage(named: tripImageName)

}

if let tripImage : UIImage = trip!["image"] as? UIImage {

tripImageView.image = tripImage

}

...