

{day-6} Universal App, Size classes & Camera.

* Move Rows -

indexPath.row != 0 &&
indexPath.row != 2

	edit
1 -	
2 -	
3 -	
4 -	

← Shuffling Rows

canMoveRowAt

→ Bool {

Return true

→ it will be able to move

}

moveRowAt

← indexPath to

-- destinationIndexPath: {

→ move the row & update an error.

← update Local Array

let itemToMove = dishes[_]
dishes.remove(at: sourceIndexPath)

dishes.insert(itemToMove, at: destination)

}

→ Pizza & Pasta - cannot move

canMoveRowAt {

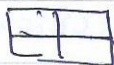
return indexPath.row != 0 && indexPath.row != 2

}

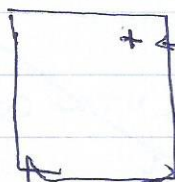
* Insert New Item in Project.

→ {
 → setEditing
 - committed editing
} # should be removed

→ Add Button "Top Right"



[Automatic]



Bar Button (type of Add Button)

Action

@IBAction addItem {

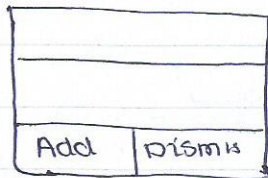
@IBAction func addNewItem() { Print("Adding new item")

let alertController = UIAlertController()

alertController.addAction {

viewController.addAction("Add")

viewController.addAction("Dismiss")



let dishNameTextField = viewController

let dishPriceTextField =

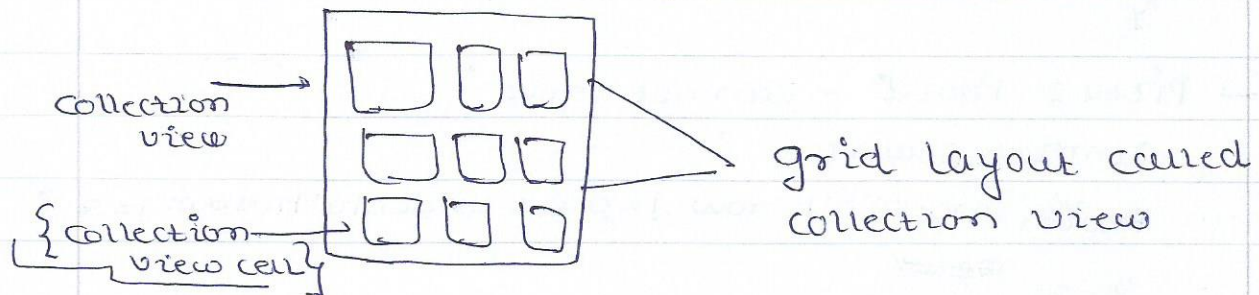
dishes.append(dish(...))
dishTableView.insertRows()

or

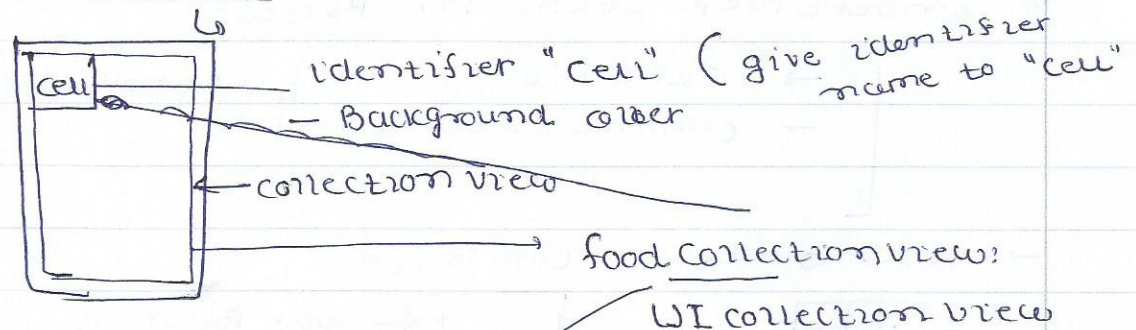
{willRefreshEntire
tableView}

{ dishes.insert(newDish, at: 0) }
dishTableView.reloadData() }

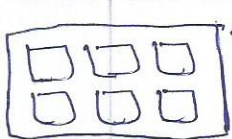
* Collection Views (Add Collection View)



- Flatiron.com



viewController: UIView, UIViewController, UITableViewDataSource
func viewDidLoad() {
 view.dataSource = self }
func numberOfItemsInSection() { return 10 }
func cellForItemAt {
 let cell = collectionView.dequeueReusableCell(withIdentifier: "cell")



← dynamically

Add UI Collection View Delegate Flow Layout {

← (gives padding)

let insets = UIEdgeInsets (top: 10, left: 10, bottom: 10)

FlowLayout

add two method of UI Collection Delegate Flow-

(Func insetForSectionAt) → return insets.

(Func sizeForItemAt) → screen width = self.view.frame.width.

func minimumLineSpacingForSectionAt {

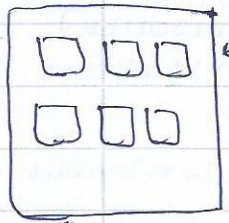
return insets.left

}

let screenWidth = self.view.frame.width
let paddingSpace = (itemPerRow + 1) * inset.left

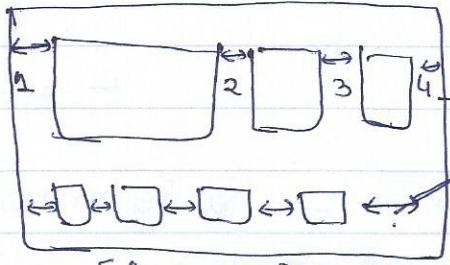
let paddingSpace = (itemsPerRow + 1) * insets.left

let availableWidth = screenWidth - paddingSpace.



← dynamically calculate height & width.

→ give 3 cell on each row.



total padding space + 10
(itemPerRow + 1)

↑ always one more

(item + 1) * inset.left

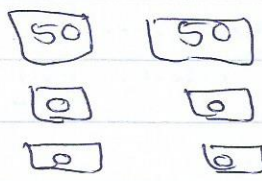
→ let availableWidth = screenWidth

* Collection View Flow Layout: has minimum spacing

→ Remove them

(minimum 3 items Per row)

Item Size
Leader
Footer



(Set the space)
=

442 523

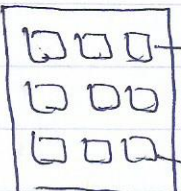
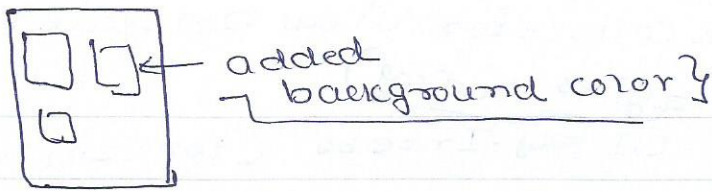


image WRI.
cell attach cell
imageview cell

content mode: Aspect Fit
: Unspecified

→ Cell → Food cell extends UI Collection View cell.

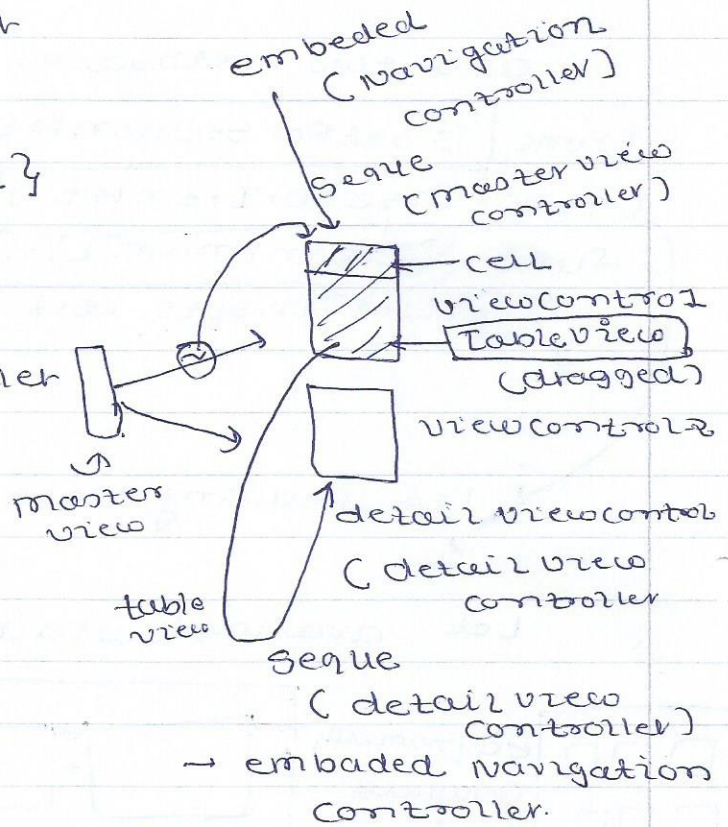
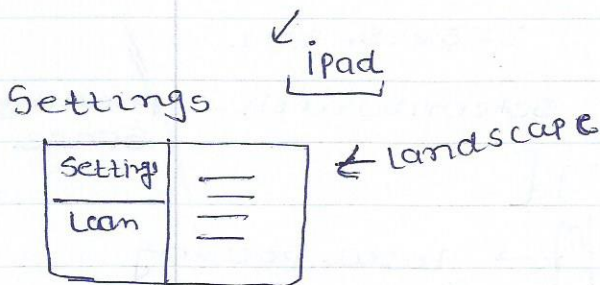
{ UI Collection View cell }



* Splitview Controller

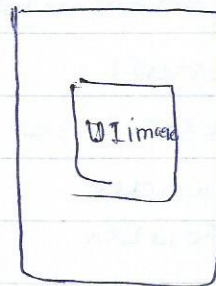
1	images {controller 1}
2	
3	
4	

* Splitview Controller (Navigation Controller)



→ Splitview → is initial view controller (Landscape mode)

→ Add: MasterviewController (Add master scene)
DetailviewController (Add Detail scene)



- Align Center x to:
- Align Center y to:
- Width Equals: 250
- Height Equals: 270

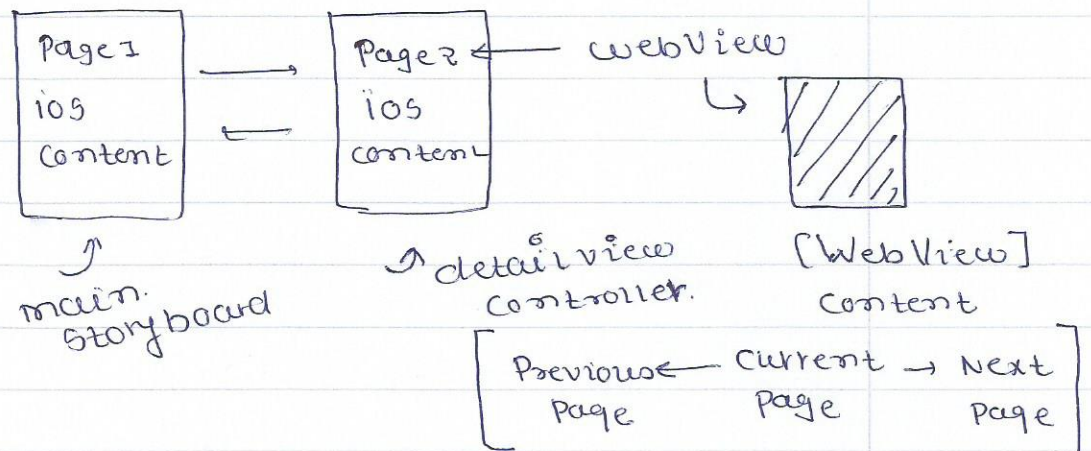


align → add constraints (add constraints)

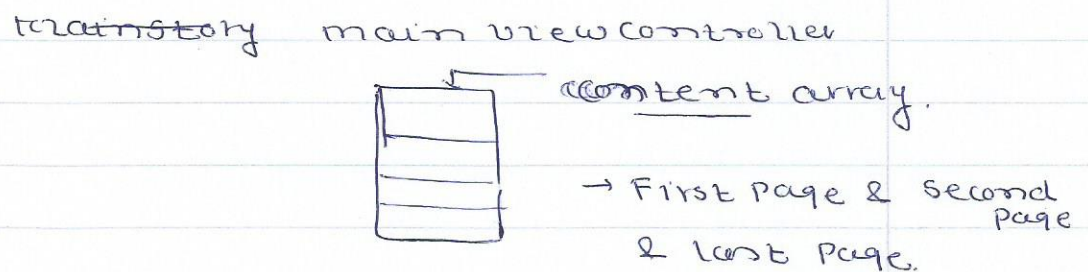
* Pageview Controller

Page1 → Page2 → Page3 (pager controller)

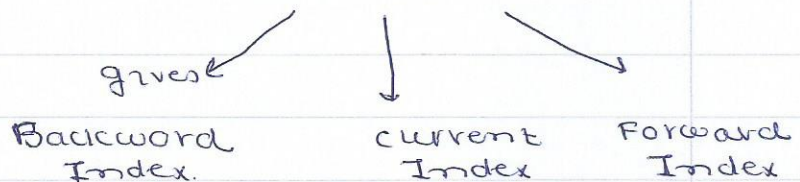
* Pageview Controller.



→ when you are in last page, you are not allowed to go to further.



→ Swipe forwarder → Pageviewcontroller



→ one Function

↳ give index

↳ By providing index gives Page