

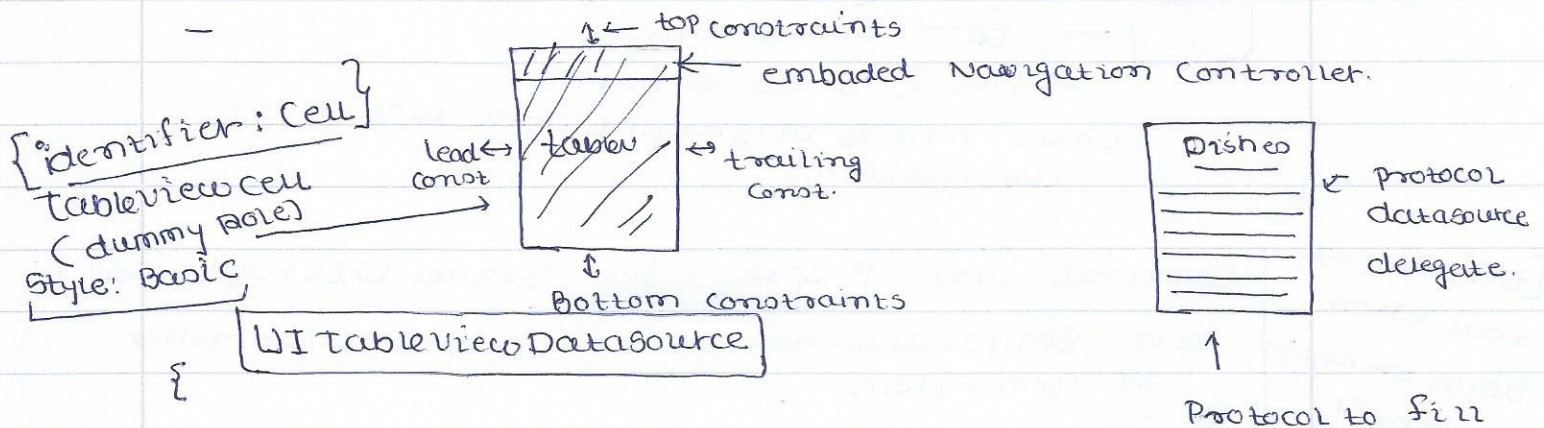
[day-5] Tableview - Collection View - View Controller

* tableview - list view, display items in list

- display in list format. (

- enabled set of option in list view.

-



@IBOutlet weak var dishesTableView: UITableView!

var items = ["Pasta", "Noodles"]

override viewDidLoad() {

super.viewDidLoad()

→ tableView.delegate = self

→ dishesTableView.dataSource = self

→ {base of all the method of data source}

}

}

func numberOfRowsInSection

→ items.count

Implement this two method.

func cellForRowAt

→ Responsible adding cell to each row.

- let rowIndex = indexPath.row

var item = items[rowIndex]

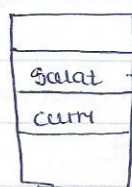
let cell = tableView.dequeueReusableCell (withIdentifier: "Cell", for: indexPath)

cell.textLabel?.text = item

return cell

}

* UITableView Delegate.



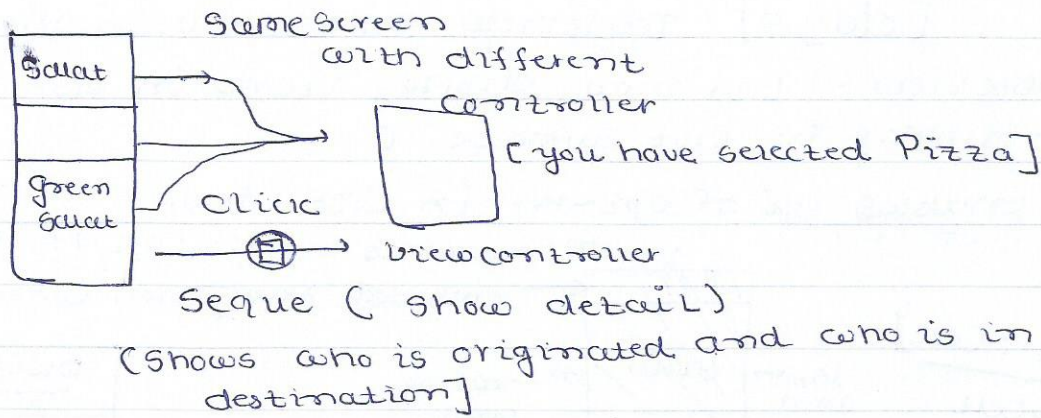
→ Select cell & display title.

func.. didSelectRowAt {

let rowIndex = indexPath.row

let item = items[rowIndex]

Print ("you have select



[this method tells when segue happens.]

```

Override fun prepare (for segue: UIStoryboardSegue) {
    let vc = segue.destination as! DetailViewController
    vc.item = item
}

```

or (Alternative way)

About to jump to next screen.

→ DetailViewController → identifier == "detail-view"

→ self.add 'didSelectRowAt' method.

let vc = self.storyboard?.instantiateViewController (withIdentifier: "detail-view") as! DetailViewController

→ vc.item = item

→ self.navigationController?.pushViewController (detailViewController, animated: true)

* Delete Row



→ self.navigationItem.rightBarButtonItem = editButtonItem

(it will add edit button)

LeftBarButton & RightBarButton

```

* → → setEditing (- editing: Bool, animated: Bool) {
    tableView.setEditing (editing, animated:
    animated)
}

```

→ commitEditingStyle (method)

if editingStyle == .delete {

Print ("will delete")

* Delete Item

- Remove First from Array.
- Remove from table view.

deleteRow (indexPath, with:
• automatic)

UITableViewCell Editing
Style: enum

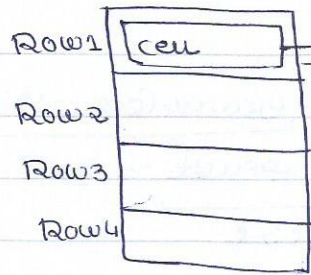
- delete
- insert
- none

* Cell Sub-title

name
price
name
price

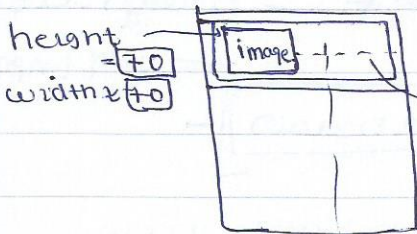
* Custom table view

DisnCell%



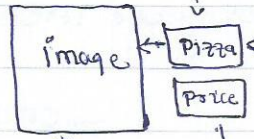
cell height = {75} → Row height = {75}

→ create UITableView cell



{x=5, y=5}

- Center vertically in container
- Leading space to container
- Fixed width
- Fixed height

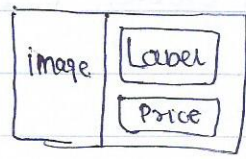


Leading to image
top like image
→ Price

drag horizontal to image
drag vertical to Price.

Current mode: Aspect Fit

Drawing Clip to Bounds.



cell → [DisnCell] [UITableView cell]

* table view allows single selection of Row

*

* Search Bar. (the search type)

→ UISearch Controller.

↳

Search Controller = UISearch Controller (searchResults
Controller: nil)

(if you want to
show in diff.
controller then
give name.)

↑ want to show search result
on a same screen.

→ Dismiss Background During Presentation = true

Defines Presentation Context = true

↳ when search controller is active nothing
can be clicked by user.

→ tableView.tableHeaderView = Search Controller, Search Bar
Search Controller. SearchResults update = self
Delegates &

* Filter Contents For Search Text (- searchText: String)
{

displays TableView. reload Data()

→ very important

}

↳ Refresh table view

- calls number of sections

→ all delegate method calls.

I - number of Rows In Section

→ indexPath.CellForRowAt

→ Commit editing style.

} → changes in
all delegates
methods

* Setup

↳ Adding datasource + Search Controller + Delegates
method

Add: