

{ day-4 } [Storyboard - segue - stack view]

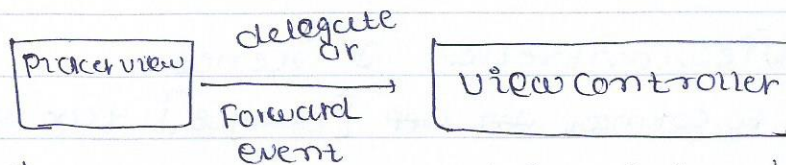
①

* PickerView

owner → ask for a Bill to waiter
waiter → delegate

Sunnyvale
Cupertino
Santa Clara

View Controller get notified → for [clicking on view]



(dummy element)

→ everything is controlled by

↳ dataSource = self (source of data. view controller)
↑ who will provide the data to the

dataSource

Class UIViewController: UIViewController, UIPickerView

@IBOutlet

↳ acts as a delegate

DataSource {

↑ need to implement the protocol then

only view controller can provide data to dataSource

→ numberOfComponents (in pickerView: UIPickerView) -
{ // one column

Column	Column

}

- func pickerView () → Int { }

* dataSource & delegates & view controller. (view controller works as a delegates)

→ numberOfComponents → 1
→ TitleForRow → "Hello World"
{ Selected title [Row]
For a Array }
→ didSelectRow: Int → selected Row from selected components.

index1	index2

[Component: 1]

```

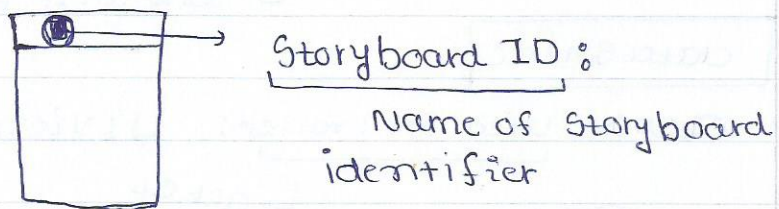
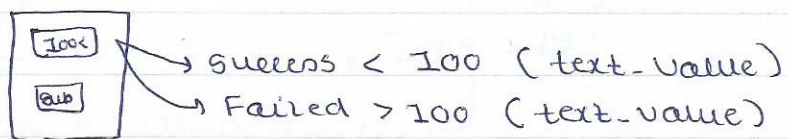
Let dateFormatter = DateFormatter()
dateFormatter.dateFormat = "dd mm yyyy hh:mm:ss a"
                                [am or pm]

```

* Multiple View Controller

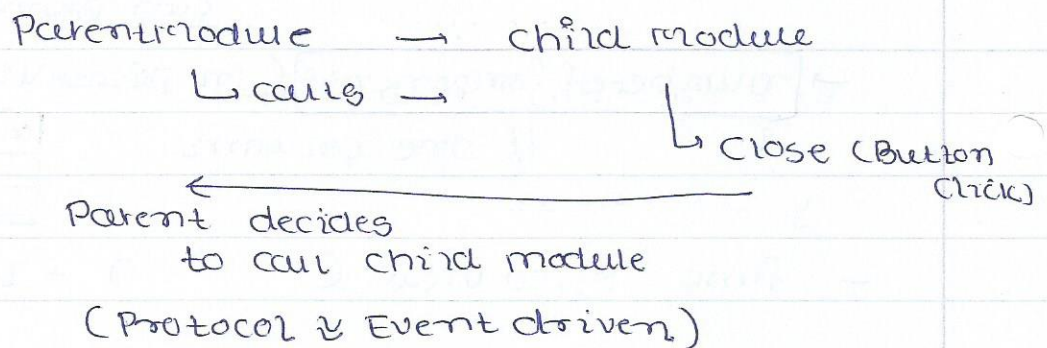
ViewController & Scenes.

Go to connect an app [Images] two different Screen.



Self. storyboard!. instantiate ViewController (with Identifier: "failure-view-controller")

* Model →



→ Can Child Controller defines Protocol & Parent Controller implements them

* Child: Successview Controller: {

```

var delegate: Successview Controller Delegate!
@IBAction func onClose (_ sender: Any) {
    if delegate != nil {
        delegate
    }
}

```


3

* Navigation View Controller :

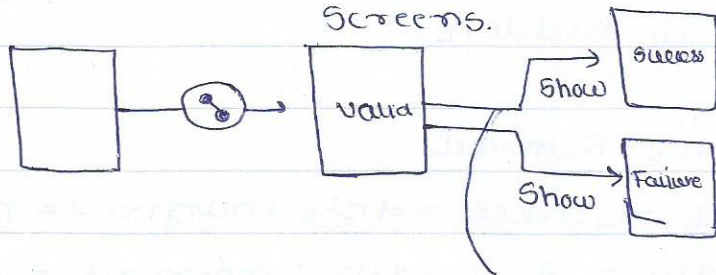
↳ Push navigation Controller - it maintains the history
it will create { viewController.java }

↓
Navigation
Controller

Self. navigationController? pushViewController (vc!,
Animated: true)

Self. navigationController.

* Segue: → Create connection between two
Screens.



Segue creates link and after
clicking on validate button. it will
jump to success screen.

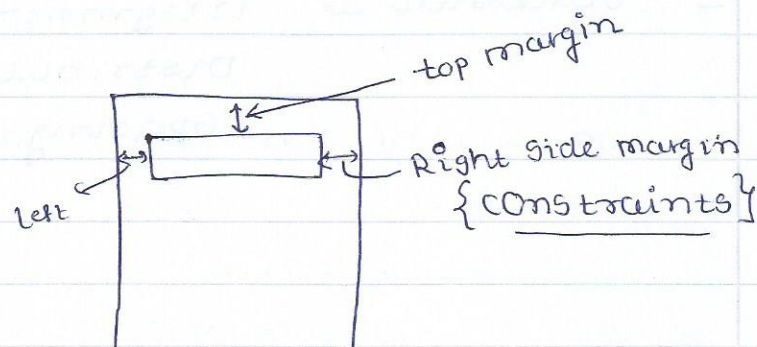
* ViewController: UIViewController {

Override func prepare (for segue: UIStoryboardSegue,
sender: Any!) {

↳ let result

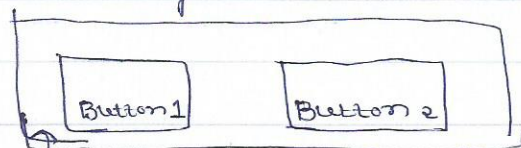
}

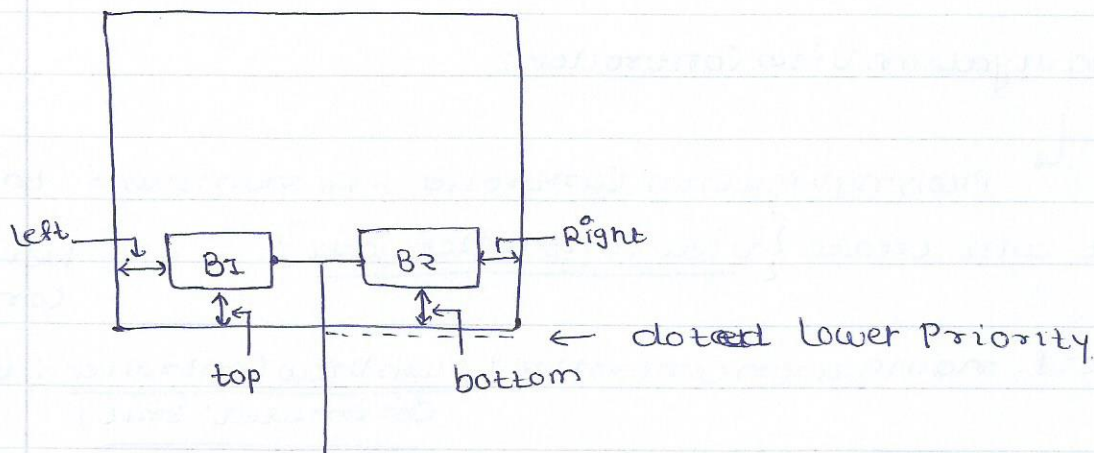
* Autolayout



* Size classes ~ two layout for different screen

* Priority : →





From B1 to B2 — horizontally spacing

B1 & B2 \approx should be equal height

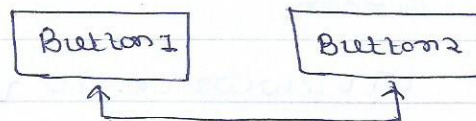
* Expanding B1 Button B1 to screen when Button2 is hidden.

→ Button2 Removed.

Button1 → takes == left margin == right margin = 20

B1 width = B2 width constraint = disabled.

* Stackview. & embeded Stackview



Select both and select embeded Stack view

→ Left space = bottom space = trail space.

→ Stackview ⇒ Alignment → Fill

Distribution → Fill Equality

Spacing → 10