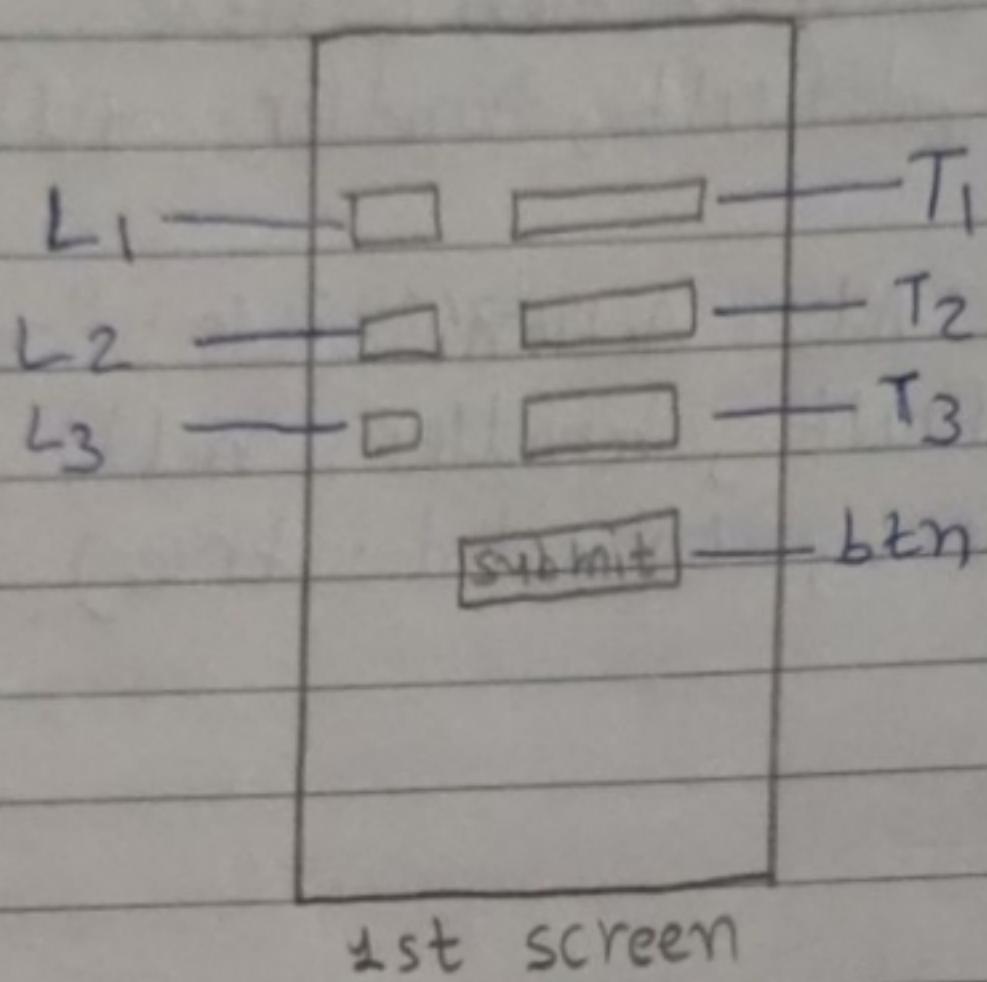


//segue (a view controller to another)

- Add three label, three text field and one btn



1st screen

- L means Label, T means textfield

Name, L₁ = Username

L₂ = User id

L₃ = Password

- Make outlet for three textfield and action for btn

- create function in viewController.swift
~~(Coding code will file) mi~~

func storeData()

{

Userdefaults.standard.set(name.text,
forkey: "username")

Textfield outlet name

Userdefaults.standard.set(id.text, forkey: "uid")

Textfield outlet name

Userdefaults.standard.set(pass.text, forkey: "pass")

same

y

code
self.Btn on outlet

func submit(sender: Any) {

let alert = UIAlertController(title: "warning",
message: "Do you want to store data?",
preferredStyle: .alert)

alert.addAction(UIAlertAction(title: "yes",
style: .default,

handler: { ACTION in self.storeData() })

(function Name)

Print("Data stored")

})

alert.addAction(UIAlertAction(title: "No",
style: .destructive, handler: nil))

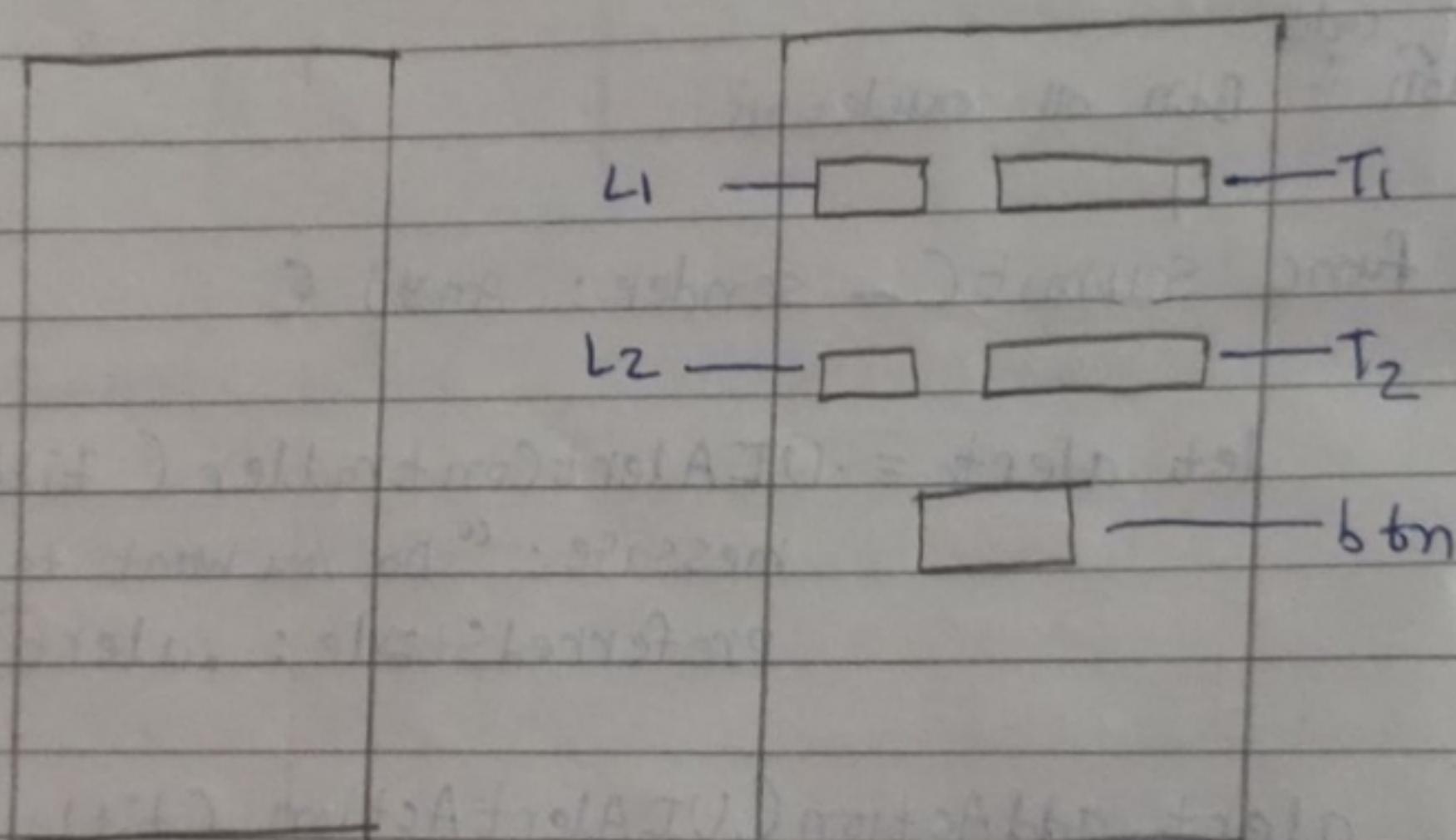
self.present(Calert, animated: true)

- Now, go to Main storyboard

add View controller

In new view controller add two label, two
textfield and one btn

New View controller



1st screen : 2nd screen

- L means Label, T means Text Field

Name, L1 = Userid

L2 = Password

- Now, create new file for new View Controller

For new file - right click on Project → new file → select cocoa touch → subclass mi select UIViewController and class mi (my class Esig Class mi file of am 8 rudi fia or null egi)

Now,

select 2nd screen → property card aini option → add new class file

- Now, create for two textfield and action for btn (for 2nd screen) make outlet and action in new file
- create function in new file (create function after btn action)

```
func checkData() {
    if (name.text != UserDefaults.standard.string(forKey: "uid")) &&
        Pass.text != UserDefaults.standard.string(forKey: "Pass"))
}
```

S

```
let alert = UIAlertController(title: "Congo",
    message: "Login Sucessfully", preferredStyle:
    .alert)
```

```
        alert.addAction(UIAlertAction(title: "OK",  
        style: .default, handler: nil))
```

```
self.present(alert, animated: true, completion: nil)
```

Y

else

S

```
let alert = UIAlertController(title: "Sorry",  
message: "Wrong Password",  
preferredStyle: .alert)
```

```
alert.addAction(UIAlertAction(title: "OK",  
style: .default, handler: nil))
```

```
self.present(alert, animated: true, completion:  
nil)
```

Y

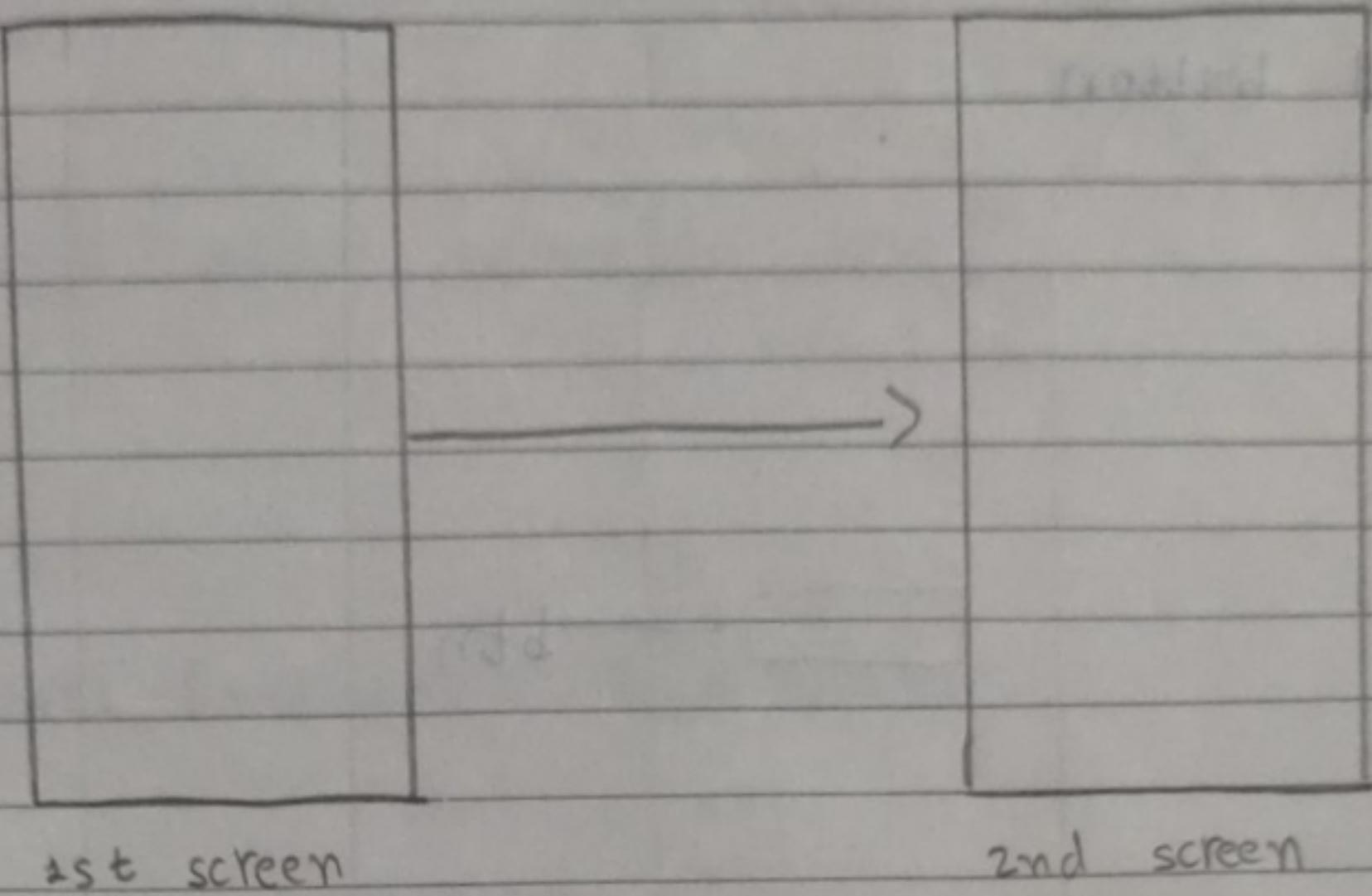
- Now call function in btn.action

func login S

check_data()

Y

- Now create bridge between 1st screen and 2nd screen



- select 1st screen and drag to 2nd screen, select Present Modality option

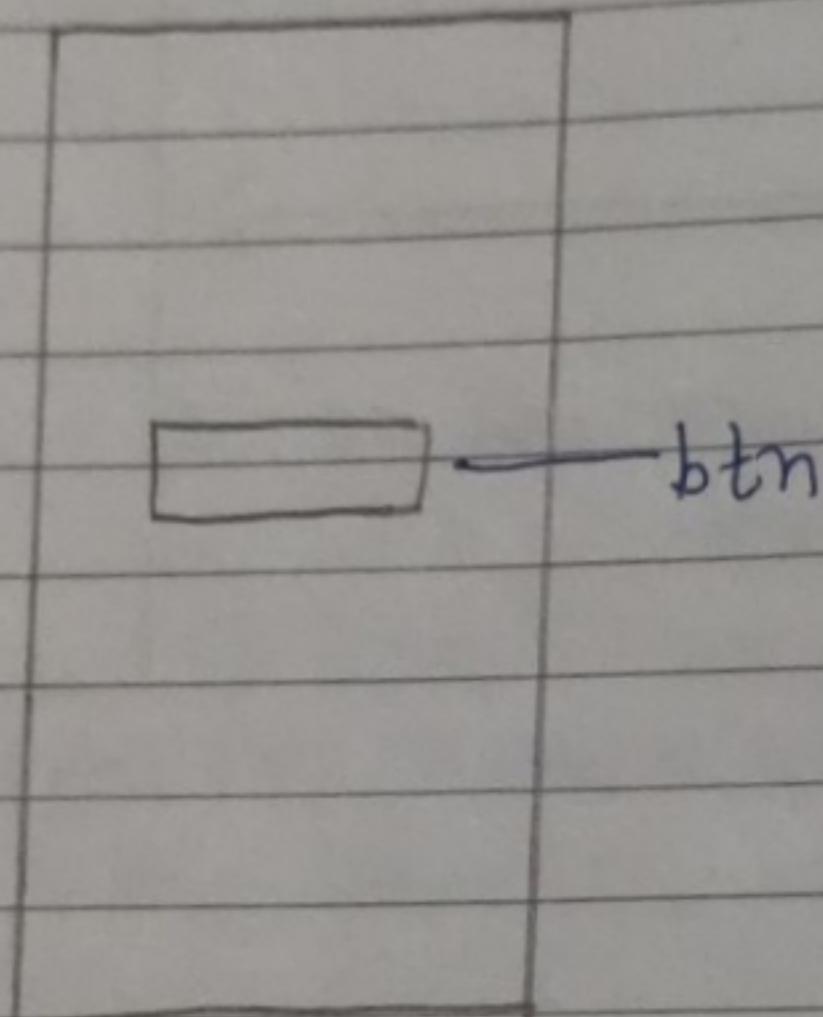
Now write this line after print statement

Cin 1st screen code file

```
        print("Data stored") // 2nd method (using  
self.performSegue(withIdentifier: "iBridge",  
sender: self)  
        // 3rd method (using Identifier name)  
    })
```

// UIAlertController

- Add button



- Now, create action for button
- Now, create function for C create function after btn action)

```
func demoC() {
```

```
let alert = UIAlertController(title: "Exit",  
message: "Are you sure?", preferredStyle:  
preferredStyle: .alert)
```

```
alert.addAction(UIAlertAction(title: "Yes",  
style: .default, handler: nil))
```

Normal blue color will

alert.addAction(UIAlertAction(title: "No",
style: .destructive, handler: nil))
Red color will
present(alert, animated: true)

- Now call function in btn action

func submit {

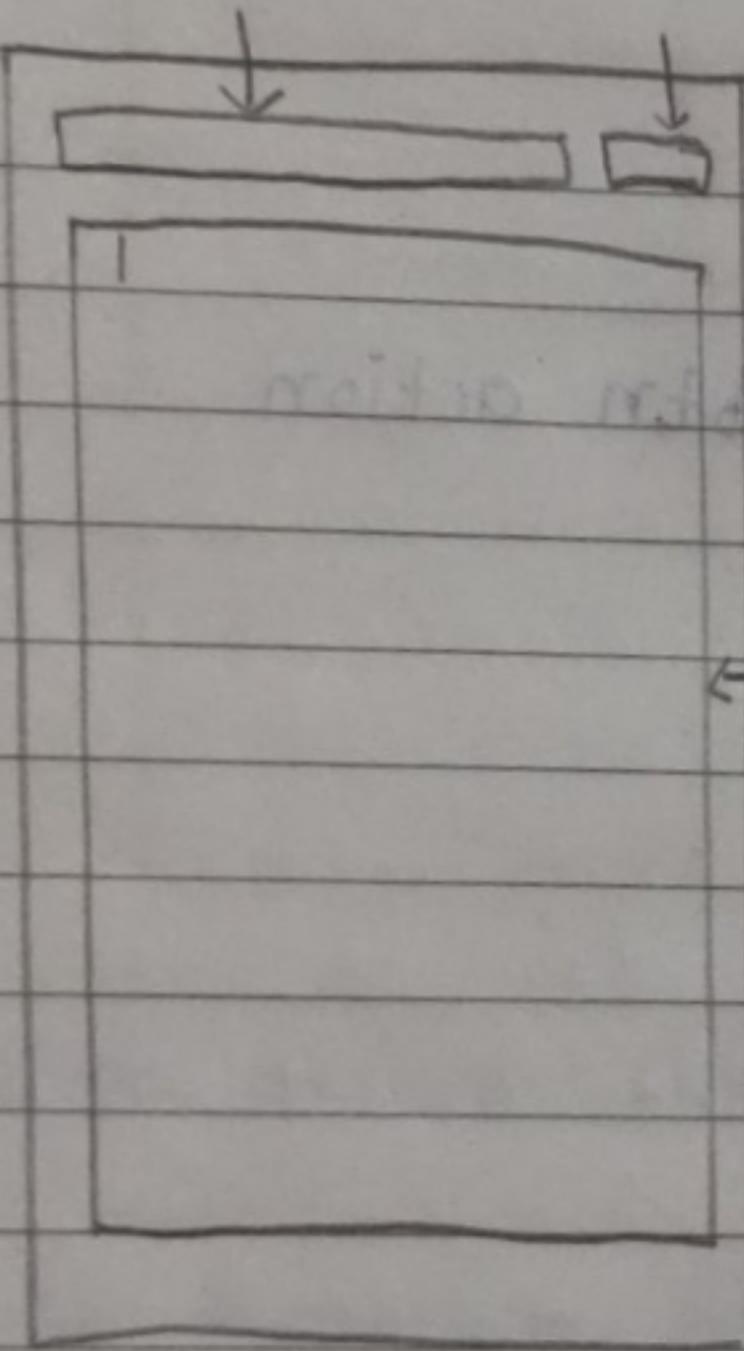
demo()

//UI WebView

- Add one textfield, one button and webView

textfield

btn

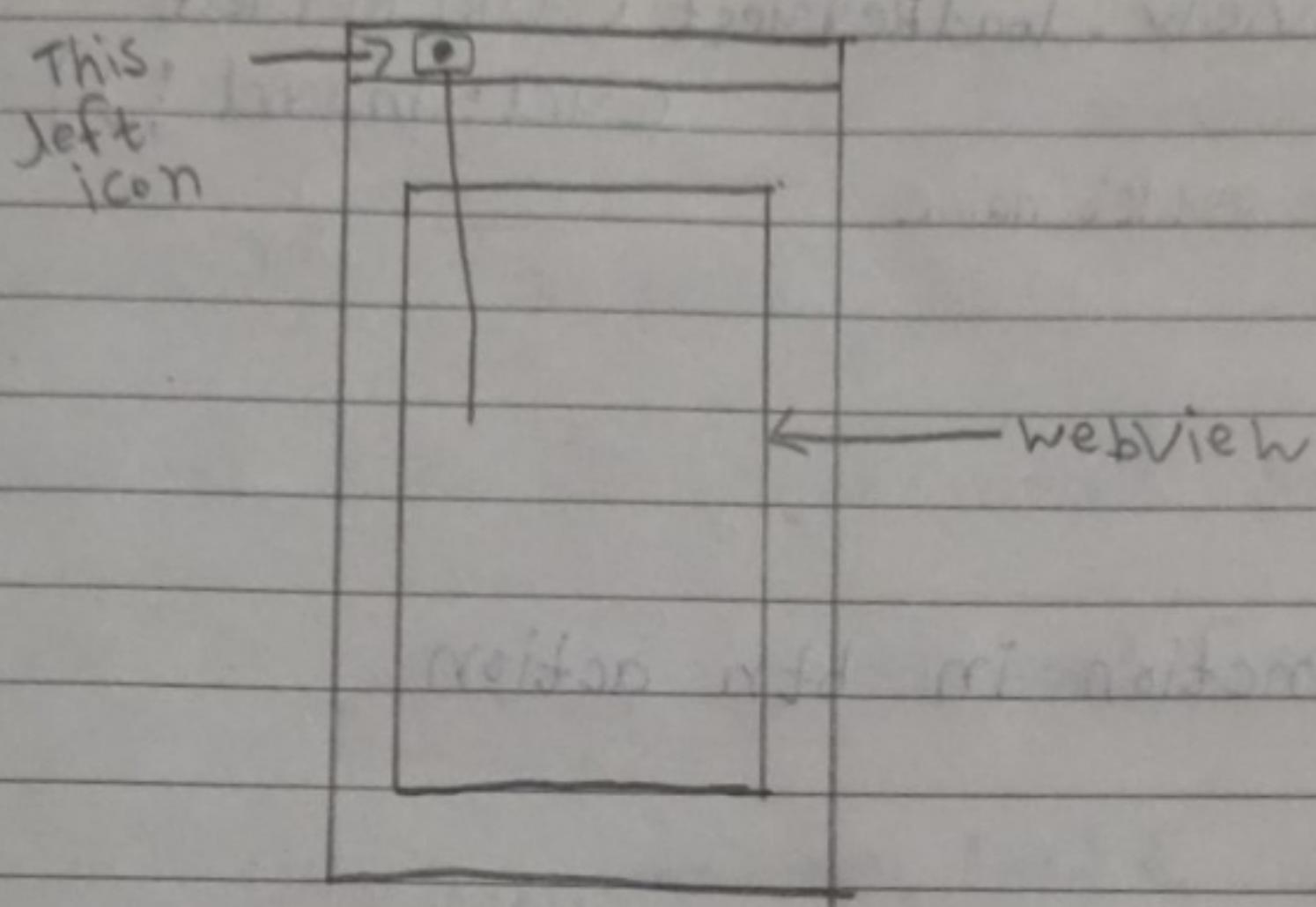


webView

- Now create outlet for textField and webView
and action for btn
- Now create function & create function after
btn action)

Phone open URL (web address):

- Select webView drag and drop on screen yellow icon, select delegate option



- Now create outlet for textField and webView, and action for btn
- Now add Delegate method in class
(class and line mi, saai)
class ViewController: , UIWebViewDelegate
- Now create function (Create function after btn action)

argument
↓
func open_URL (webaddress : String)

argument
↓
let myurl = URL (string : webaddress)

mywebView.loadRequest (URLRequest
curl: myurl !)
|
webView outlet name |
Var

y

- Now call function in btn action

func go
argument | textfield
open_URL (webaddress : myText.text !)
| outlet

y

- If set app open url and webView curl sua load saig sua di suon code

- Code write in viewDidLoad

super.viewDidLoad()

open_URL (webaddress : "https://www.
↑ func | google.com")
argument

y

- In webView add Indicator View add size & code.
- Add Indicator View in main storyboard
- Now create outlet for Indicator View
- ~~After outlet~~
- Create outlet for code start scroll
type webView didstart load (select this function)

func webView did start load {

outlet name ~~myIndicatorView~~ myIndicatorView.isHidden = false
myIndicatorView.isAnimating

- Now type webView didFinish (select this function)

func webView didFinish {

myIndicatorView.isHidden = true

view.isUserInteractionEnabled = true

2

- webview did start click function all of view. isUserInteractionEnabled = false

// Picker View