

iOS APP DEVELOPMENT

Industrial Training Report

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF
THE DEGREE OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE ENGINEERING

By

Ketan Batra (161277)

UNDER THE GUIDANCE OF

Dr. Pardeep Kumar



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May 2020

DECLARATION

I hereby declare that the work reported in the B.Tech Project Report entitled iOS APP DEVELOPMENT submitted at **Jaypee University of Information Technology, Wanknaghat**, India is an authentic record of our work carried out under the supervision of **Dr. Pardeep Kumar**. We have not submitted this work elsewhere for any other degree or diploma.

Ketan Batra

Signature of the Student

Ketan Batra
161277

This is to certify that the above statement made by the candidates is correct to the best of my knowledge

Pardeep Kumar

Signature of the Supervisor

Dr. Pardeep Kumar

Date:

ACKNOWLEDGEMENT

It is my privilege to thank all the people who contributed to make this internship period a great and unforgettable experience of my life.

First of all I would like to thank **Jaypee University of Information Technology, Waknaghat** for providing me the golden opportunity of gaining practical industry knowledge through internship training at **CODEBREW LABS**. I take this opportunity to express my sincere gratitude and deep regards to my Mentor, **Mrs Gurleen Kaur** for her exemplary guidance, monitoring and constant encouragement throughout the course of this internship. I also thank all my team members for sharing the knowledge and experience of their respective fields which assisted me in the successfully blend in the ongoing live project.

Finally I thank each and every member of the **CodeBrew** family in technical, non-technical fields and support staff who have directly or indirectly helped me. This internship program provided me a lot of opportunities to work in different settings and exposed me to interaction with different sets of people globally. I was able to utilize and enhance different skills in me like decision making, communication, teamwork and coordination.

ABSTRACT

Code Brew Labs had practical experience in Designing, Developing, Branding and Online Marketing. As an entrenched driving Web Development Company in India they have aced the craft of deciphering customer thoughts and desires into great deliverable administrations. They put stock in straightforwardness and organize each customer's prerequisites with a lot of duty. Besides, their administrations are sponsored up with sound information and aptitudes of innovation, alongside practical plans. Their worldwide customer base is assorted and goes from little new businesses to huge undertakings.

Job: IOS APP DEVELOPER

iOS is Apple's versatile OS that sudden spikes in demand for an iPhone, iPad, iPod Touch equipment. Apple gives instruments and assets to making iOS applications and embellishments for these gadgets. As an iOS designer, you can program in local dialects, for example, Swift or Objective-C or construct cross-stage local applications utilizing React Native (JavaScript) or Xamarin (C# and F#).

Our group of iOS designers is answerable for creating applications for cell phones controlled by Apple's IOS working framework.

Obligations

- Designing and building apps for the IOS stage
- Ensure the presentation, quality, and responsive nature of uses
- Collaborate it with group to characterize, plan, and boat new highlights
- Identify and right bottlenek and fix all bug
- Help keep up code quality, association.

LIST OF FIGURES

1. Figure 1: Grintafy
2. Figure 2: Grade-up
3. Figure 3 : Sozu
4. Figure 4: Business Goals
5. Figure 5: App-Design
6. Figure 6: Swift vs Object C
7. Figure 7: X-CODE
8. Figure 8: App Delegate and View Delegate LifeCycle
9. Figure 9: View LifeCycle
10. Figure 10: Model View Controller Architecture
11. Figure 11: SignUp
12. Figure 12: Verification Page
13. Figure 13: Data transferring between pages (Protocols)

INDEX

DECLARATION	1
ACKNOWLEDGEMENT	2
ABSTRACT	3
LIST OF FIGURES	4
COMPANY PROFILE	6
METHODOLOGY	10
Chapter 1 INTRODUCTION TO SWIFT	11
1.1 Swift - Features	
1.2 Swift - Overview	
1.3 Swift vs Objective C	
Chapter 2 INTRODUCTION TO X-CODE	17
2.1 Introduction	
2.2 General Functioning	
2.3 Cocoa-Pods	
2.4 Pod files	
Chapter 3 Delegate Life Cycles & Architectures	21
3.1 App-Delegate LifeCycle	
3.2 View Delegate LifeCycle	
3.3 Scene Delegate LifeCycle	
3.4 Model View Controller Architecture	
Chapter 4 DATA Transferring mini project app	27
4.1 Coding	
4.2 Outputs	
Conclusion	35

Organization PROFILE

Code Brew Labs is represented considerable authority in Design, Development, Branding and Online Marketing. As an entrenched driving Mobile And Web Development Company in India they have aced the specialty of interpreting customer thoughts and desires into top notch deliverable administrations. They trust in effortlessness and organize each customer's necessities with a lot of duty. Additionally, their administrations are upheld up with sound information and aptitudes of innovation, alongside practical plans. Their worldwide demographic is various and ranges from little new businesses to enormous enterprises. Code Brew Labs is a firm having some expertise in mechanical answers for organizations - the ones that exist, and the ones that are still in their incipient stages. We make innovation a significant resource for your business - be it a startup, a SMB or a venture - these are remembered for their extending collection. They have fiddled with Business Analytics, IoT based framework, cloud-based registering and arrangements, alongside raids into the universe of AR and VR.

Administrations

Quickened Mobile Page:- An open source venture made to improve the presentation of website pages for cell phones. The innovation behind AMP empowers lightweight pages that heap all the more rapidly for Smartphones and tablets clients. With the assistance of this innovation, you will have the option to distribute content a lot quicker. It doesn't make a difference where your business is arranged as the task permits us to execute the framework and adjust it to all sort of crowds.

Top Mobile App Development Services for Startups and Enterprises

Code Brew Labs is a first class Mobile App Development organization in India that offers probably the best improvement administrations. Regardless of whether you are a startup or an undertaking organization, you can profit our exceptionally desired iPhone and Android App improvement administrations which will be natural, amazing and locks in.

Advanced Marketing

Advanced Marketing has become the center of the considerable number of organizations, regardless of their size. Code Brew Labs is glad to be a standout amongst other Digital Marketing Company in India, where we practice all the most recent white cap strategies so as to make a business noticeable on web.

Top Mobile Game Development Services

Code Brew Labs is one of the most encouraging and conveying Mobile App Game advancement organization in India. We have long periods of involvement with making quality and increased gaming applications.

Products:

GRINTAFY:

An extraordinary application exclusive for an foot-ball lover in Middle East that allows user to make & promotee the profile, play with friends and let others join with 1000000+ downloads taking trend of foot-ball to another level.

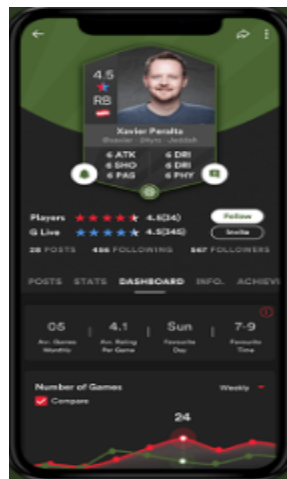


Figure 1: GRINTAFY

B) RED VAULT

A most Rewarded Application for each gamer In Middle East. Super-dazzling designs and intuitive UI let the players play and acquire more. Red vault effectively handles 90 percent of gaming business sector of GCC with 100K+ download.

C) GRADEUP :- Exam Preparation Application, Subsidiary of Times Internets.

One of the biggest and best stages for serious test readiness in India with 10M+ downloads. Learning made available and moderate!



Figure 2 :GradeUp

D) ERP SOLUTION FOR SML ISUZU:

ERP Solution to the business vehicle manufacturer.SML ISUZU is an eminent business vehicle producer, having some expertise in transports, ambulances and altered vehicles.SML Isuzu Limited is a business vehicle maker set up in 1983.SMLI delivers and sells transports, ambulances and modified vehicles.



Figure 3 : SML ISUZU

The 'BREW' Procedure Helps Exceed the Business Goals



Figure 4 : Business Goals

Briefing Of The Business Idea

Inspecting the thoughts and subtleties, we propose the most ideal arrangements that will meet your image prerequisites in reality.

- Thorough research to keep you in front of your rivals
- A structure for your altered application with explicit highlights

Examining A Perfect Architecture

Perusing our heap of most recent advances, our brewers make a completely modified arrangement, made only for your business

- An remarkable UI plan for better client experience

- Development of the center usefulness with cutting edge highlights Evolving Bug-Free Business App.

METHODOLOGY

Fundamental Design Stages for iOS Apps:- To see how to configuration forms, you should comprehend configuration stages for iOS applications. Programming improvement steps are requested in time by stages, beginning from a phase of small understanding and moving to phases of dynamically expanded information and consistency, with the exercises beginning with the patrons or the visionaries and moving to the advancement group and afterward to the client. There are four phases:

- **Inception:** The beginning stage is the place things are being made sense of — a rough vision; a business case; scope; a significant level, potential design; and elevated level appraisals of endeavors and cost.
- **Elaboration:** This is where you refine the vision, approve the center engineering, and resolve dangers. Truth be told, you may state that elaboration is about the goals of the dangers. This is the place the vast majority of the prerequisites recognizable proof is done and (with trust) practical venture management estimates are made.
- **Construction:** The phase where you iteratively execute any residual highlights and get ready for sending. The development stage is the place the undertakings are consistently including an ever increasing number of highlights. A lot of nitty gritty OO structure, usage, and testing happens here.
- **Transition:** This is where you convey a completed discharged. Here, the product is gone over to clients. By chance, you may likewise beta test a framework in this stage. Exercises from the sending, testing, and upkeep stages happen here.

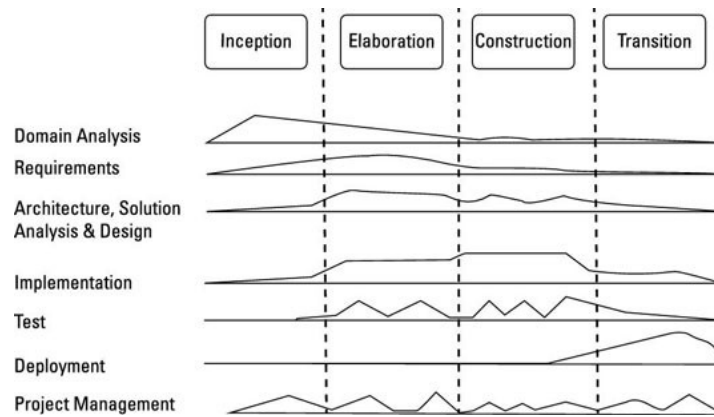


Figure 5 :APPDESIGN

CHAPTER :- 1

INTRODUCTION TO SWIFT

Swift is other programming language for iOS, macOS, watchOS, and tvOS application improvement. Regardless, various bits of this be normal from the experience of making in C-language and Objective-C language . swift gives its own types of all essential C and Objective-C types, including int for numbers, Double and Float for skimming focuses , Bool for Boolean characteristics, and String for scholarly data.

Swift is sort safest programming language, which suggests the language urges you to clear the sorts of characteristics your code can works up with . In the occasion that bit of code needs a String, type prosperity shields from passing the int coincidentally. In like way, type prosperity keeps you from unexpectedly passing an optional String to a touch of code that needs non-optional Strings. Type security causes you fixing and catching goofs as exactly on schedule possible in headway strategy.

Swift likewise presents discretionary sorts, that handles the nonattendance of the worth. Optional states that wether there are a worth, and that rise to x or there isn't an incentive by any stretch of the imagination". utilizing nil with pointers is like Utilizing optionals with pointers in Objective-C, however they work for any sort, not simply classes. Not only are optional more secure and enough revealing than nul pointer in Objective-C, they're at the center of an extensive part of Swift mosts amazing features.

Notwithstanding natural sorts, Swift presents propelled type are not there in Objective C, for example, tuple .Tuples empowers to make and go around the grouping in qualities. You could make utilizing the tuple to restore numerous qualities from the capacity as a solitary worth.

1.1 Features of Swift

Swift Supports Dynamic Libraries

Dynamic libraries are executable lumps of code that can be connected to an application. This element permits current Swift applications to connect against more up to date forms of the Swift language as it develops after some time. Dynamic libraries in Swift are legitimately transferred to the memory, eliminating the underlying size of the application and eventually expanding application execution.

More secure Platform

In the serious portable application commercial center, building up a protected application ought to be a need. Quick's sentence structure and language developments avoid the few kinds of slip-ups conceivable in Objective-C. This strength implies that there will be less crashes and instances of risky conduct. It doesn't keep developers from composing terrible code, yet rather commits it more averse to make errors. This includes an additional layer of value control during advancement.

Less Code and Less Legacy

With Objective-C, there are numerous issues that cause application crashes. Quick gives code that is less mistake inclined due to its inline support for controlling content strings and information. Moreover, classes aren't separated into two sections; the interface and the usage.

This cuts the quantity of documents in the undertaking down the middle, which makes it a lot simpler to deal with.

Programmed Reference Counting (ARC)

Quick uses(ARC) to follow & deal with the application's Main-Memory utilization. Much of the time, this implies memory the executives just work in Swift & you don't have to consider memory the executives yourself. Curve naturally opens up the memory utilized by class examples when those occasions are not, at this point required.

Swift Overview

Closures:-

Closures are free squares of value that can just pass around & use in your code. Terminations in Swift resemble squares in C language & Objective-C and to lambda in other programming dialects. Closures just catch & store the reference to any non changing value and components from settling in what they are best described also known as closing over these variable & constant. Swift handles an entirety of the memory the executives of catching for you.

Enumeration:-

Enum in the Swift are five star type in their own way . They embrace numerous highlights customarily upheld just by classes, for example, figured properties to give extra data about the list's present worth, and occurrence strategies to give usefulness identified with the qualities the list speaks to. Specifications can likewise characterize initializers to give an underlying case esteem; can be stretched out to grow their usefulness past their unique usage; and can adjust to conventions to give standard usefulness.

Optinal Chaining:-

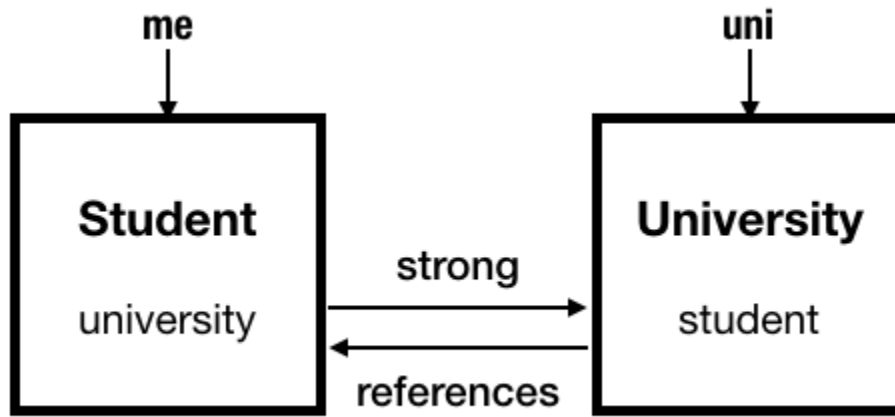
Optinal Chaining(OC) is the system for addressing & calling methods, strategies, and addendums on an optional that may starting at now be nil. If the optional have a value, the method, procedure and addendum call succeed if the optional is nul, the property, method, or addendum get back to brings nil.

Protocols :- protocol characterizes a diagram of techniques ,methods, and different prerequisites that best fits a specific errand or bit of usefulness. The convention would then be able to be embraced by a class, structure, or identification to give a real execution of those prerequisites. Any sort that fulfills the prerequisites of a convention is said to adjust to that convention.

Generics:- Generic is one of often remarkable highlights of Swift language & a great part of Swift standard libraries is worked with an conventional code. Truth be told you have been utilizing generic all through the entalect manage, regardless of whether you didn't understand it. For example Swift Array and Dictionary types are both regular varieties. You can make a display that holds Int values, or a bunch that holds String regards, or without a doubt a group for whatever other sort that can be made in Swift.

Subscripts:- Classes, structures, and lists can characterize Subscripts, which are alternate ways for getting to the part components of an assortment, rundown, or arrangement. You use addendums to set and recover values by list without requiring separate strategies for setting and recovery.

ARC:- Swift uses(ARC) to follow and deal with your application's Main- Memory use. By and large, this implies memory the board just work in Swift & you don't have to consider Main-Memory the executives yourself. Bend naturally opens up the memory utilized by class examples when those occasions are no longerneeded.



Why swift over objective C :

OBJECTIVE-C vs SWIFT COMPARISON		
Characteristics	Objective-C	Swift
Performance	<ul style="list-style-type: none"> ✗ Not fast due on runtime method lookup table and custom messaging ABI ✓ Exception: C functions 	<ul style="list-style-type: none"> ✓ High performance
Safety	<ul style="list-style-type: none"> ✗ Uses null pointers and may cause no operations 	<ul style="list-style-type: none"> ✓ Uses an approach that allows programmers to find and fix bugs quickly
Maintenance	<ul style="list-style-type: none"> ✗ Two separate files of code complicate developers' job 	<ul style="list-style-type: none"> ✓ Easy to maintain
Syntax	<ul style="list-style-type: none"> ✗ Includes a lot of @ symbols, lines, semicolons, and parentheses 	<ul style="list-style-type: none"> ✓ Resembles English
Complexity	<ul style="list-style-type: none"> ✗ Text strings is very verbose and need a lot of steps to link two pieces of information. 	<ul style="list-style-type: none"> ✓ Requires less code lines for the same operation
Community support	<ul style="list-style-type: none"> ✓ Loyal fans of 30 years 	<ul style="list-style-type: none"> ✓ A fast-growing group of supporters
Memory management	<ul style="list-style-type: none"> ✗ ARC doesn't work for procedural C code and APIs like Core Graphics 	<ul style="list-style-type: none"> ✓ Supports the ARC for all APIs
Dynamic libraries support	<ul style="list-style-type: none"> ✓ Dynamic libraries supported 	<ul style="list-style-type: none"> ✓ Dynamic libraries supported
Long-term outlook	<ul style="list-style-type: none"> ✓ Continuous support by Apple 	<ul style="list-style-type: none"> ✓ Rapidly growing language

Figure 6: SWIFT VS OBJECTIVEC

CHAPTER 2

INTRODUCTION TO X-CODE

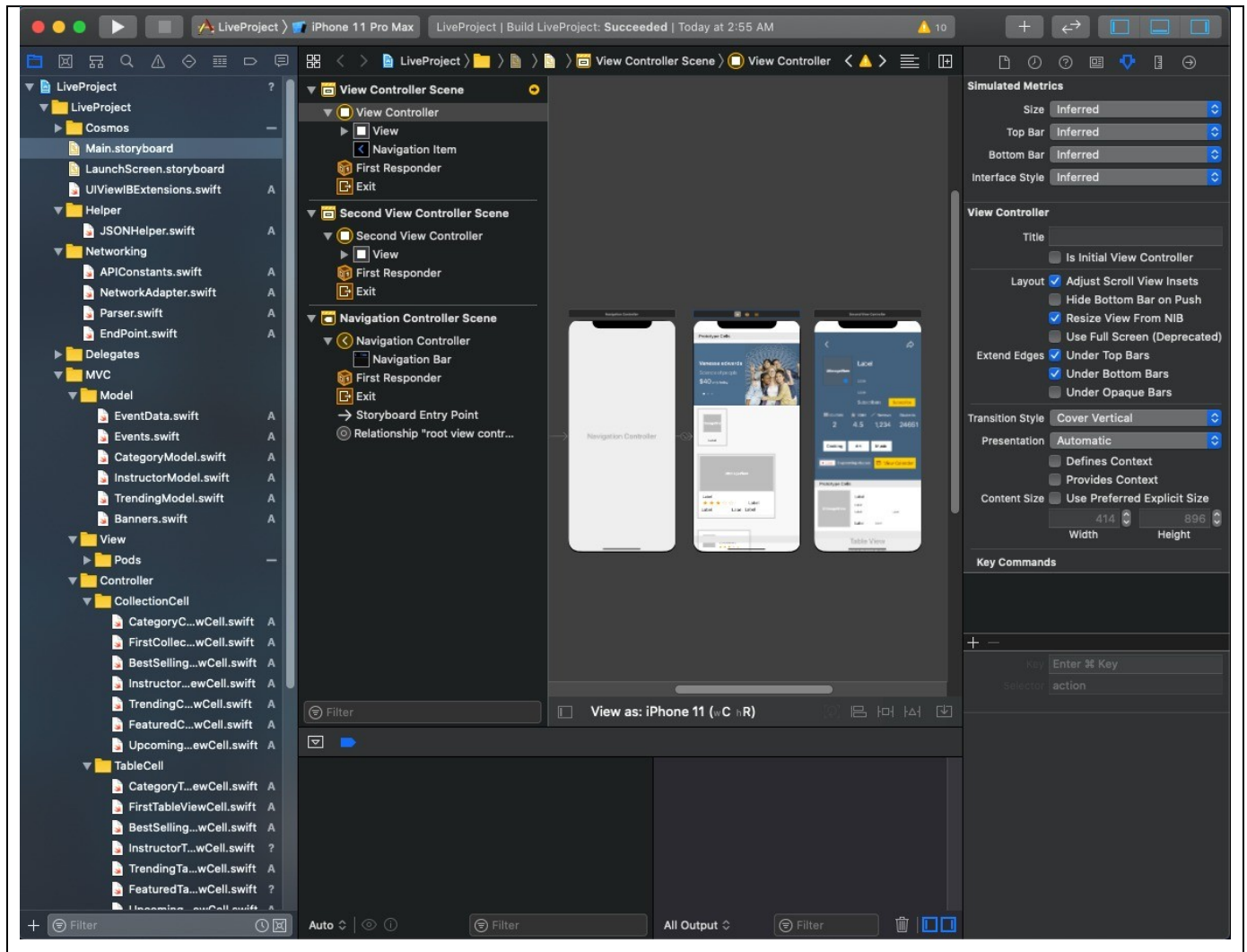
2.1 Introduction

XCode is a coordinated improvement condition created to chip away at Mac working frameworks. It contains a set-up of programming improvement devices created by apple. XCode encourages us to create programming for macOS, tvOS, iOS, and watchOS. The most recent stable arrival of XCode is 11.3.2, which is easily available on the App Store for all the clients of macOS Mojave. X-code11 is accessible in the App Store and incorporates SDKs for iOS 13, macOS Catalina 10.15. Xcode 11 backings advancement for gadgets running iOS 13.1.

2.2 General Operations



Figure 7 - Xcode



Xcode 11 backings advancement with SwiftUI.

- Xcode 11 adds support for Mac Catalyst to bring iPad applications to the Mac.
- You would now have the option to change the nearness of Xcode self-sufficiently of the system appearance setting.
- Xcode underpins transferring applications from the Coordinator window or from the request line with xcode fabricate or xcrun instrument. App Loading isn't, now included with Xcode.

2.3 CocoaPods:-

Cocoapods is a reliance supervisor used to introduce conditions for quick and target C extends in XCode. It is like the expert or frying pan, which are utilized to introduce conditions in Java. In any case, a reliance director is an apparatus that deals with the arrangement of structures to make life a couple of simpler for the designers.

In iOS, Cocoapods introduces outsider libraries like Firebase, Alamofire, and so forth. Be that as it may, in the event that we don't utilize Cocoapods to introduce an outsider library like Firebase, we have to introduce it physically, which is a monotonous procedure. Here, we have to introduce all the conditions of firebase alongside firebase, and on the off chance that Firebase makes changes to its SDK, at that point we have to redownload it into our venture.

CocoaPods is worked with Ruby and it will be installable with the default Ruby accessible on macOS. You can utilize a Ruby Version director, anyway we suggest that you utilize the standard Ruby accessible on macOS except if you recognize what you're doing.

Utilizing the default Ruby introduce will expect you to utilize sudo when introducing diamonds. (This is just an issue for the span of the jewel establishment, however.)\$ sudo gem install cocoapods

2.4 What is a Podfile?

Podfile is the specification that best describe the dependency of target of one or many X-code project. files should simply be named `Podfile.`

How to use a PodFile?

For creating a new project with cocoapods we follow these simple step:-

- Create a new project in X-code.
- Open the terminal window and type `$ cd` to change the directory in your project.
- Create podfile. Then to make it run we type `$ pod init.`

- Open your podfile the very first line would specify the platform and version supported
- In order to use CocoaPods you need to define the Xcode target to link them to. So for example if you are writing an iOS app, it would be the name of your app. Create a target section by writing `target '$TARGET_NAME' do` and an `end` a few lines after.
- Add a cocoapod by describing `pod '$PODNAME'` on one line inside your target file.

Chapter 3

Application Life Cycles & Architecture

3.1 APP-DELEGATE LIFE CYCLE:-

iOS Application Lifecycle

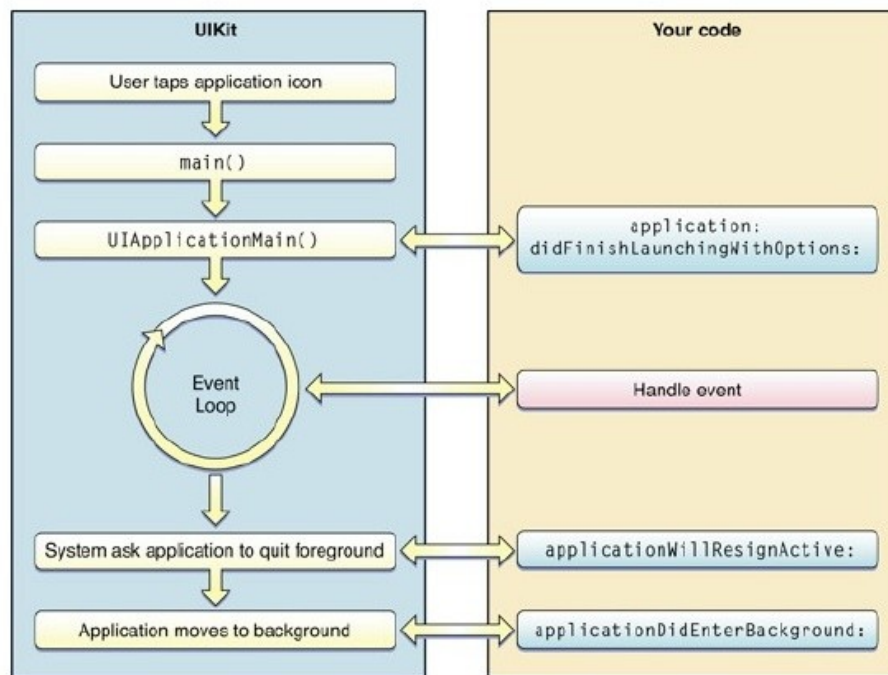


Figure 8: APPDELEGATE LIFECYCLE

Main point of entrance into iOS apps is **UIApplicationDelegate**. **UIApplicationDelegate** is a protocol that our application had to implement to get notified regarding the user's event such as application launching, application goes to background (BG) or foreground (FG), application is terminated, a push alert was opened, etc.

Whenever a iOS application is being launched the firstly thing called is **application: willFinishLaunchingWithOptions:-> Bool**. This method intended for initial app setting. Storyboard have already been loaded at this point but state restoration hasn't occurred yet.

LAUNCH

- **application: didFinishLaunchingWithOptions: -> Bool** is called next. This call-back procedure is called when the application has finished launching and restore its states.
- **applicationWillEnterForeground:** is called after **application: didFinishLaunchingWithOptions:** if application become active again even after receiving a call or any other interruptions.
- **applicationDidBecomeActive:** is called after **applicationWillEnterForeground:** to finish up the transition to the foreground.

Termination

- **applicationWillResignActive:** is called when application is about to be In-active
- **applicationDidEnterBackground:** is called when application enter a background(BG) state after becoming In-active.
- **applicationWillTerminate:** is called when your app is about to be purged from memory. Call any final cleanups here.

3.2 VIEWDELEGATE LIFE CYCLE:-

iOS call the methods of UIViewController as follows:

- **viewDidLoad()**—Called when the view controller content view is created and is load from a storyboard.
- **viewWillAppear()**—Called just before the view controller content view is added to the app's view hierarchy
- **viewDidAppear()**—Called just after the view controller content view has been added to the app's view hierarchy.
- **viewWillDisappear()**—called just before the view controller's content view is removed from the app's view hierarchy.
- **viewDidDisappear()**—Called just after the view controller's content view has been removed from the app's view hierarchy.

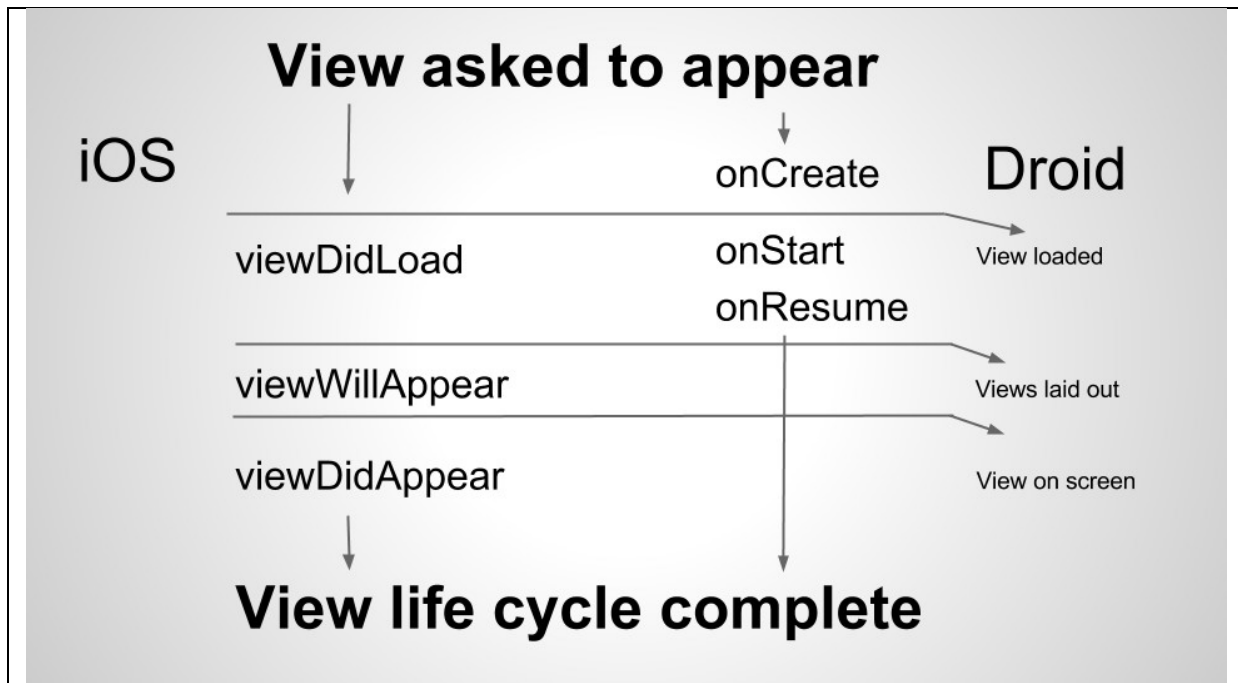


Figure 9 :VIEW LIFECYCLE

3.3 SCENEDELEGATE LIFE CYCLE:-

A *scene delegate* important function is `scene(_:willConnectTo:options:)`. In a way, it is most similar to the role of the `application(_:didFinishLaunchingWithOptions:)` function on iOS 12. The function is called when a scene is added to the app, so it's the perfect point to configure that scene. In the above code, we're manually setting up the view controller stack, but more about that later.

These are the functions of scene delegate:

- `sceneDidDisconnect(_:)` is called when the scene has been disconnected from the application . as it can reconnect again later on.
- `sceneDidBecomeActive(_:)` is called whenever the user start interacting with a scene such as Selecting it from the application switcher
- `sceneWillResignActive(_:)` is called whenever the user stop interacting to a scene, for example by switching from one scene to another.
- `sceneWillEnterForeground(_:)` is called whenever a scene enter the foreground(fg) that is starts or resumes from a background(bg) state
- `sceneDidEnterBackground(_:)` is called whenever a scene enters the background(bg) that is the app is minimized but is still present in background.

3.3 MVC ARCHITECTURE:-

The MVC — Model View Controller Architecture Design Pattern is the most widely recognized engineering found in iOS applications and is suggested by Apple.

MVC represents Model-View-Controller. It is an engineering or a product configuration design that makes making gigantic applications simple. It doesn't have a place with explicit programming language or system, however it is an idea that you can use in making any sort of utilization or programming in any programming language.

For instance, on the off chance that you are building up an application in PHP, you can utilize structures like Laravel or Codeigniter that utilizes MVC design to assist you with creating applications quick and basic. It may be somewhat difficult to get your head around the MVC structure from the start on the off chance that you have made straightforward applications without utilizing any sort of design or system. Be that as it may, this guide is to disclose you how to take a shot at MVC engineering. When you will see how it functions, you will simply cherish chipping away at MVC.

An application which receives the MVC design regularly has 3 primary layers.

1. Model :-

Model are portrayals of the application's information. For instance object like User or the post can be the models of information. A User struct or class could hold data attached to a particular client like Id, D.O.B, etc. Model works straightforwardly with the database. It doesn't need to manage UI or information handling. In genuine situation, you will just utilize model to get, supplement, refresh and erase information from your database.

2. View :- Views are the entity that clients of the application can watch and collaborate with and View items ought to be long-lasting and adaptable. Items like the UILabel that can show any sort of content are seen.

3. Controller :- Controllers intervene between the model and the view. Controller sets up sees with information from the models & update models when clients cooperate with sees.

Advantages of MVC architecture:

- o Development of the app turns out to be quick.
- o Easy for various engineers to team up and cooperate.
- o Easier to Update.
- o Easier to Debug

MVC architecture in iOS

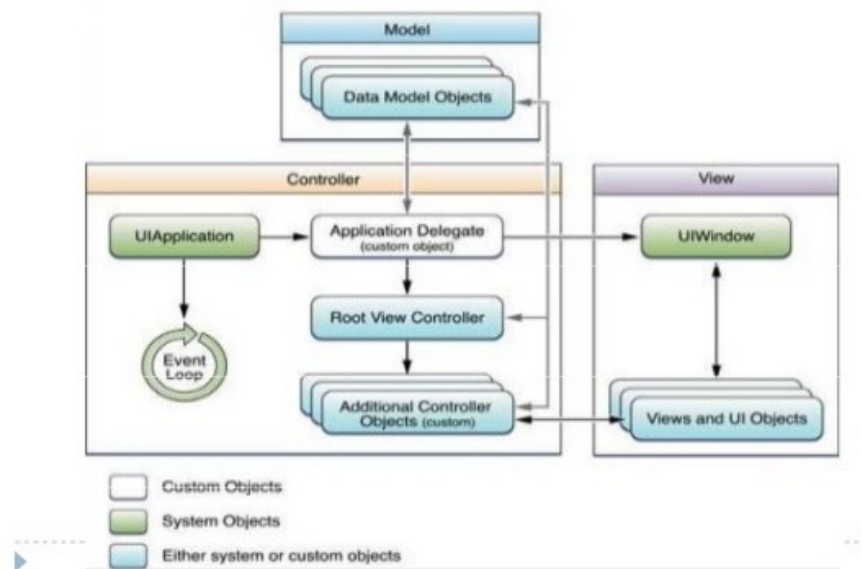


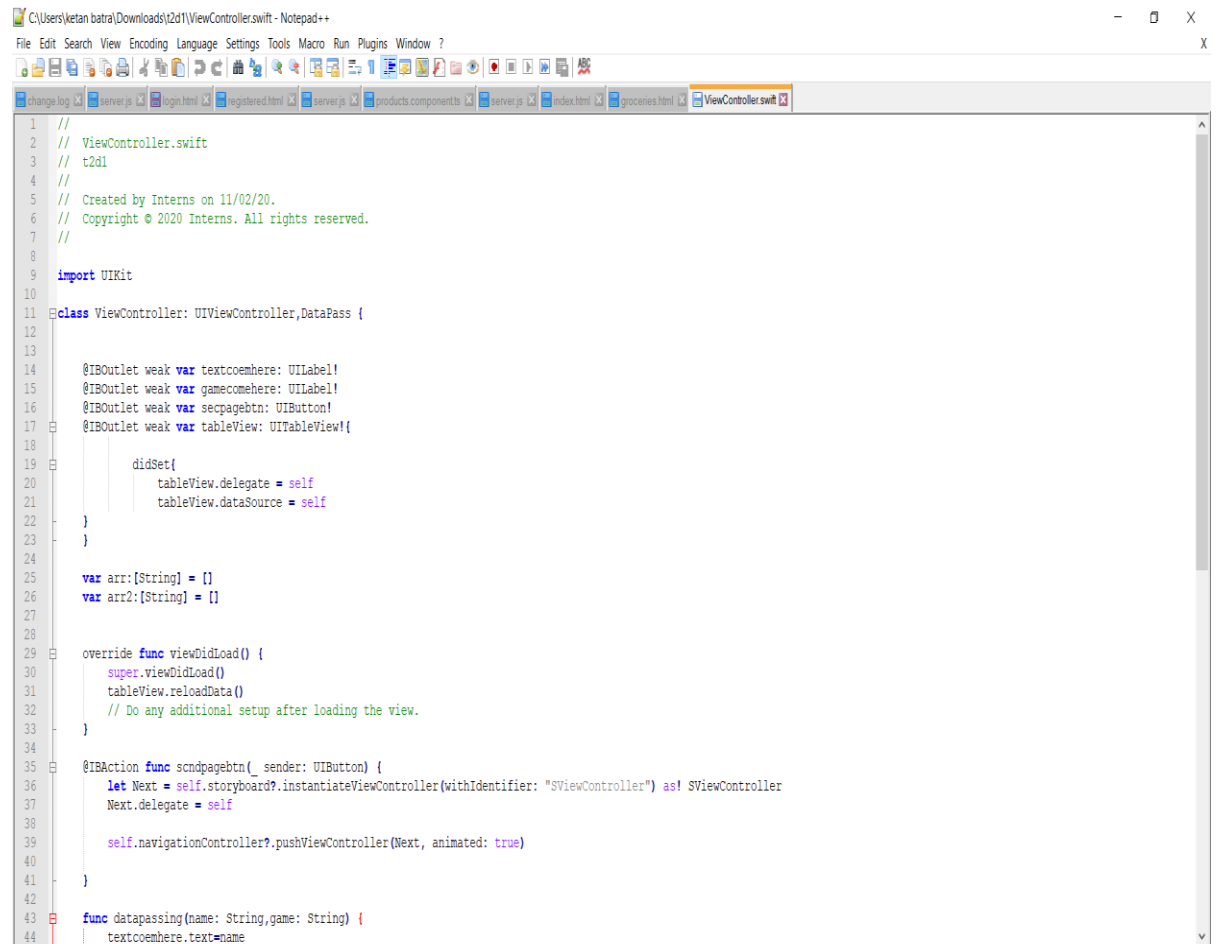
Figure 10: MVC ARCHITECTURE

CHAPTER - 4

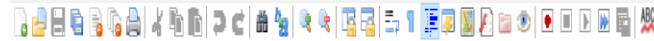
DATA Transferring mini project application

Coding:-

View controller –



```
1 //
2 // ViewController.swift
3 // t2d1
4 //
5 // Created by Interns on 11/02/20.
6 // Copyright © 2020 Interns. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController, DataPass {
12
13
14     @IBOutlet weak var textcoemhere: UILabel!
15     @IBOutlet weak var gamecomehere: UILabel!
16     @IBOutlet weak var secpagebtn: UIButton!
17     @IBOutlet weak var tableView: UITableView!
18
19     didSet{
20         tableView.delegate = self
21         tableView.dataSource = self
22     }
23 }
24
25 var arr:[String] = []
26 var arr2:[String] = []
27
28
29 override func viewDidLoad() {
30     super.viewDidLoad()
31     tableView.reloadData()
32     // Do any additional setup after loading the view.
33 }
34
35 @IBAction func scndpagebtn(_ sender: UIButton) {
36     let Next = self.storyboard?.instantiateViewController(withIdentifier: "SViewController") as! SViewController
37     Next.delegate = self
38
39     self.navigationController?.pushViewController(Next, animated: true)
40 }
41
42
43 func datapassing(name: String,game: String) {
44     textcoemhere.text=name
```



```
37     Next.delegate = self
38
39     self.navigationController?.pushViewController(Next, animated: true)
40
41 }
42
43 func datapassing(name: String,game: String) {
44     textcoemhere.text=name
45     gamecomehere.text=game
46
47
48     arr.append((textcoemhere.text ?? nil)!)
49     arr2.append((gamecomehere.text ?? nil)!)
50     //print(arr)
51     // print(arr2)
52
53 }
54
55
56 }
57
58 extension ViewController: UITableViewDelegate, UITableViewDataSource {
59
60
61
62 func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
63     return arr.count
64 }
65
66 func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
67     let Cell = tableView.dequeueReusableCell(withIdentifier: "SecondTableViewCell", for: indexPath) as! SecondTableViewCell
68     Cell.label1.text = arr[indexPath.row]
69     Cell.label2.text = arr2[indexPath.row]
70     //(tableViewCell as? TableViewCell)?.item = items[indexPath.row]
71     return Cell
72 }
73
74 func tableView(_ tableView: UITableView, heightForRowAt indexPath: IndexPath) -> CGFloat {
75     return 150
76 }
77
78 }
```

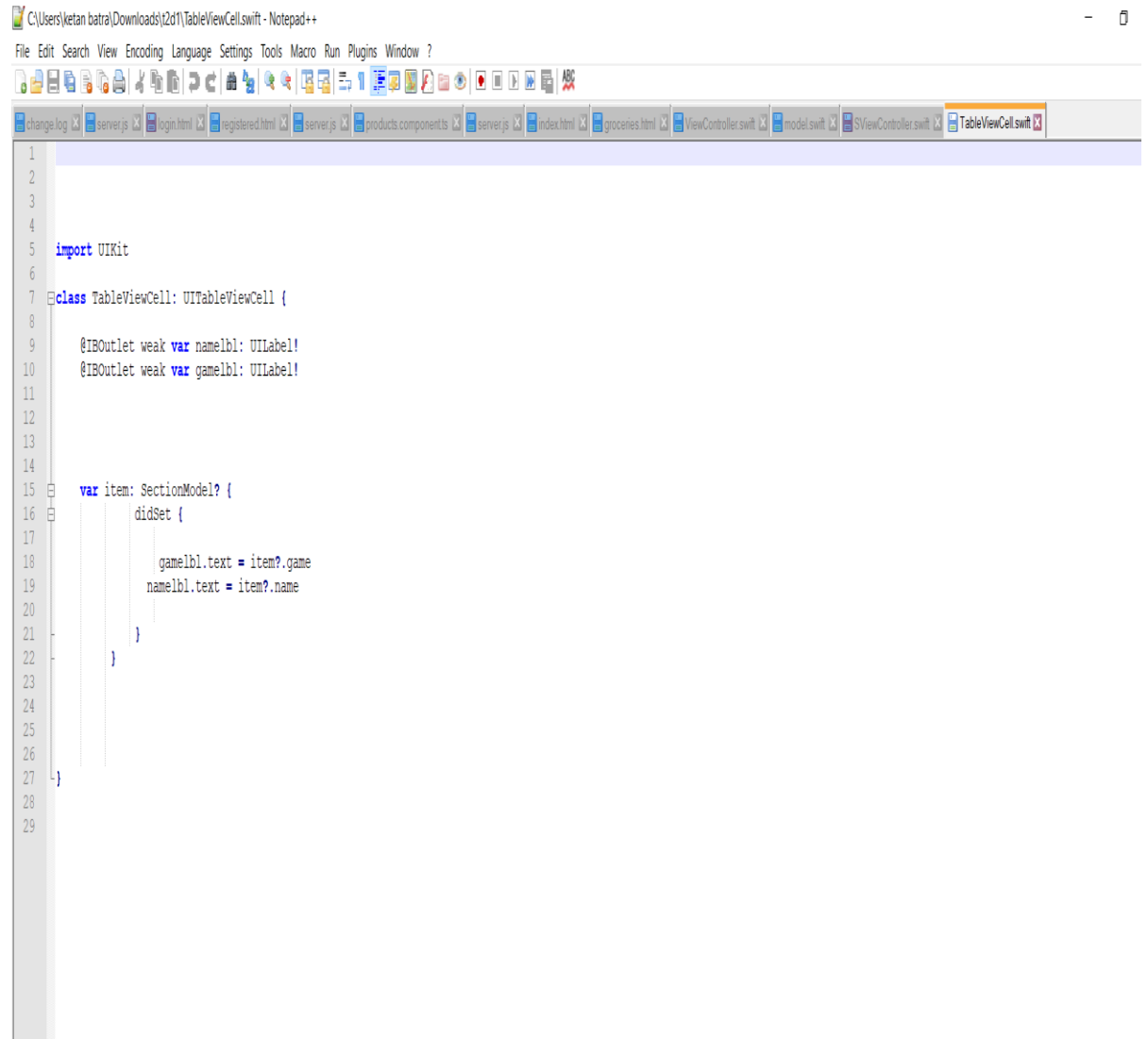
Facebook Integration :-

```
C:\Users\ketan batra\Downloads\Facebook\ViewController.swift - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log x server.js x login.html x registered.html x server.js x products.components x server.js x index.html x groceries.html x ViewController.swift x model.swift x SViewController.s
1 //
2 // ViewController.swift
3 // Facebook
4 //
5 // Created by Interns on 19/03/20.
6 // Copyright © 2020 Interns. All rights reserved.
7 //
8
9 import UIKit
10 import FBSDKCoreKit
11 import FBSDKLoginKit
12
13
14 class ViewController: UIViewController {
15
16     override func viewDidLoad() {
17         super.viewDidLoad()
18         let loginButton = FBLoginButton()
19         // Optional: Place the button in the center of your view.
20         loginButton.center = view.center
21         view.addSubview(loginButton)
22     }
23
24
25 }
26
27
```

Second View Controller:-

```
*C:\Users\ketan batra\Downloads\2d1\SViewController.swift - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
change.log server.js login.html registered.html server.js products.components server.js index.html groceries.html ViewController.swift model.swift SViewController.swift
1
2 import UIKit
3 protocol DataPass{
4     func datapassing(name:String,game:String)
5 }
6
7
8 class SViewController: UIViewController {
9
10     var delegate:DataPass!
11
12     var items : [SectionModel]?
13
14     @IBOutlet weak var tosendtext: UITextField!
15     @IBOutlet weak var tosendgame: UITextField!
16     @IBOutlet weak var backbtn: UIButton!
17
18     @IBOutlet weak var tableView: UITableView!{
19
20         didSet{
21             tableView.delegate = self
22             tableView.dataSource = self
23         }
24     }
25
26
27
28
29     override func viewDidLoad() {
30         super.viewDidLoad()
31         items = SectionModel.getSectionInitially()
32         tableView.reloadData()
33
34     }
35
36
37     @IBAction func homesend(_ sender: Any) {
38
39         _ = self.storyboard?.instantiateViewController(withIdentifier: "ViewController") as! ViewController
40         delegate.datapassing(name: tosendtext.text!,game: tosendgame.text!)
41         self.navigationController?.pushViewController(animated: true)
42
43     }
44 }
```


Table View Cell :-



```
1
2
3
4
5 import UIKit
6
7 class TableViewCell: UITableViewCell {
8
9     @IBOutlet weak var nameLabel: UILabel!
10    @IBOutlet weak var gameLabel: UILabel!
11
12
13
14
15    var item: SectionModel? {
16        didSet {
17
18            gameLabel.text = item?.game
19            nameLabel.text = item?.name
20
21        }
22    }
23
24
25
26
27 }
28
29
```

Data transferring on story board :-

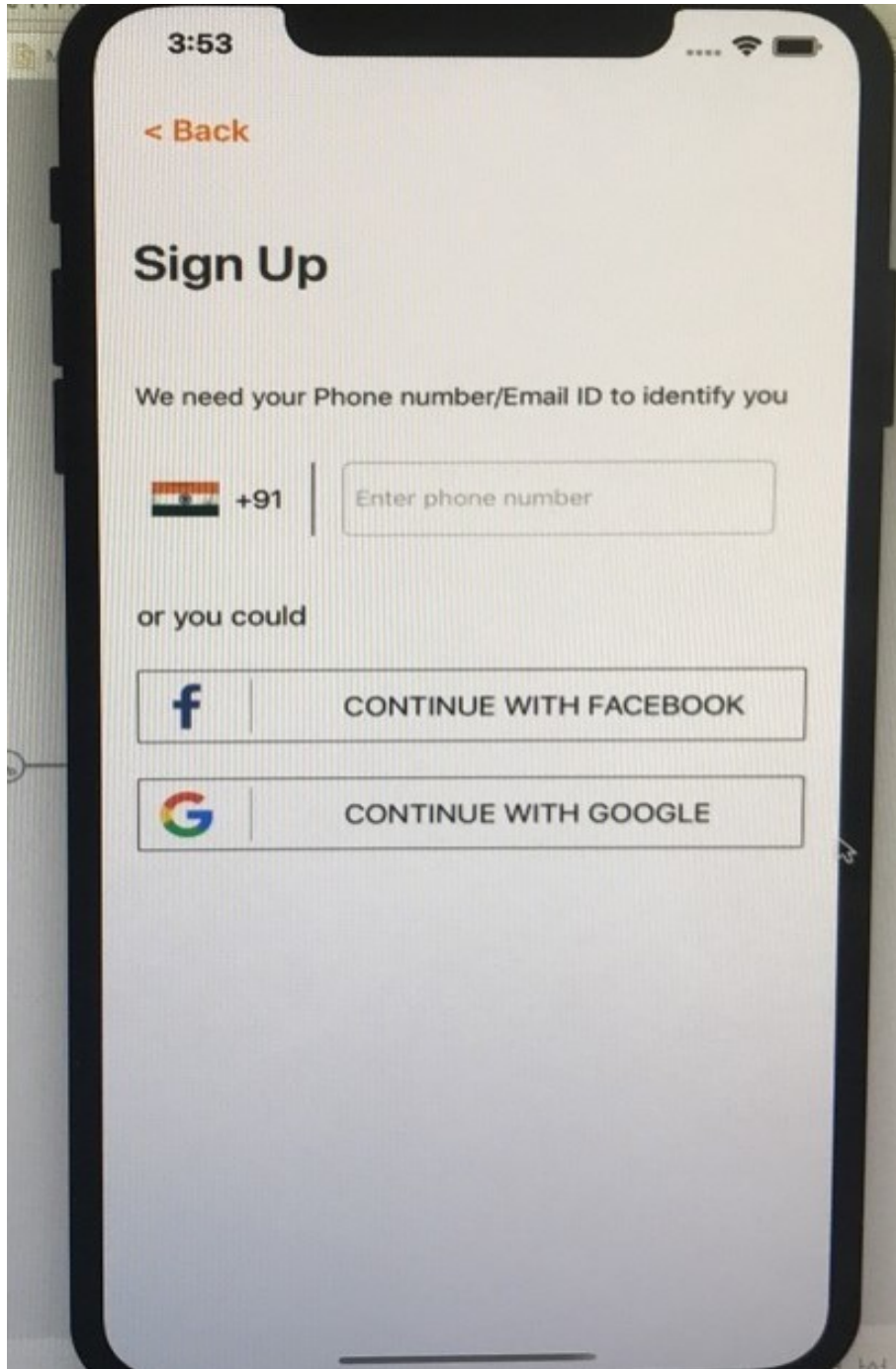


Figure 11 - Signup

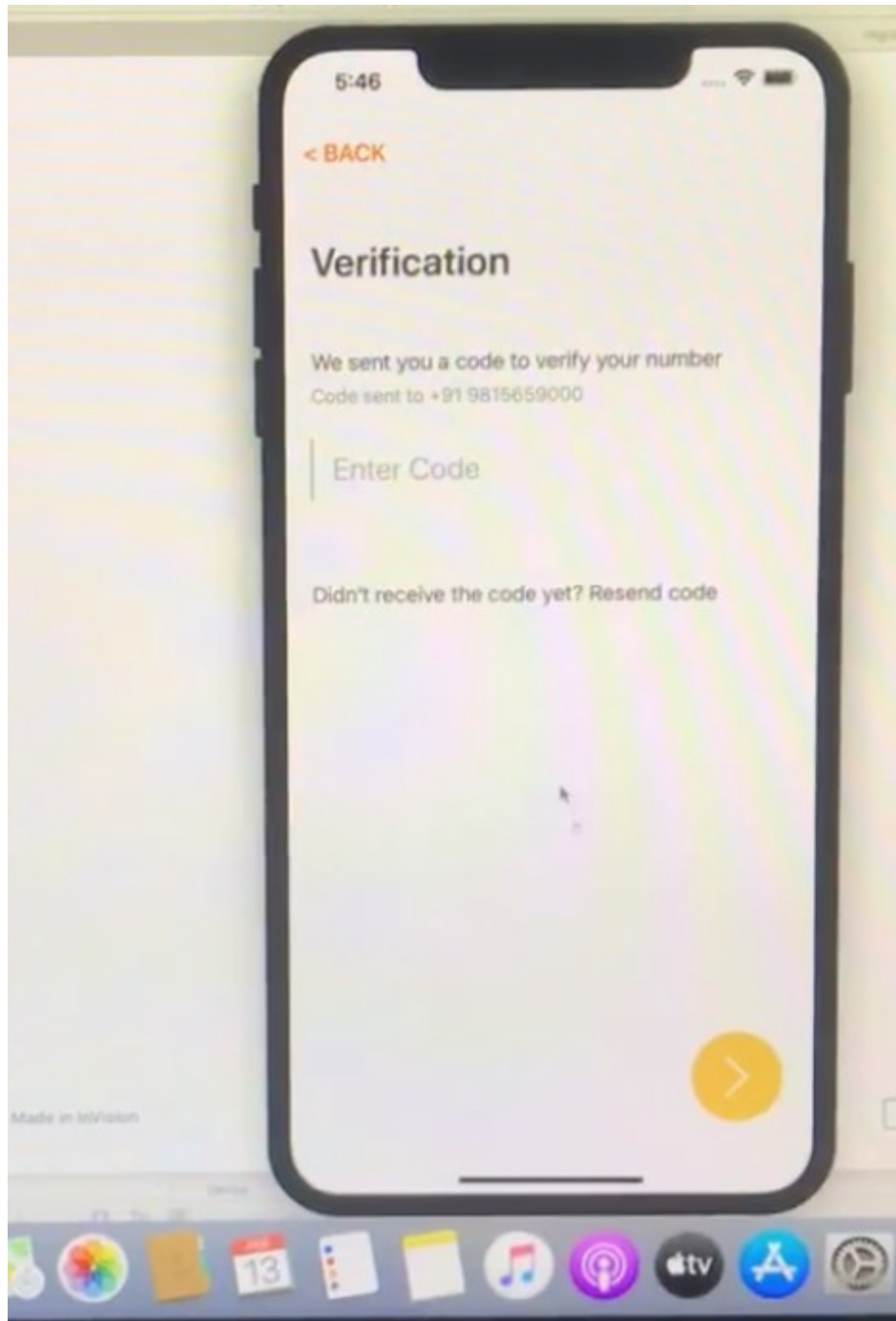


Figure 12 – verification page

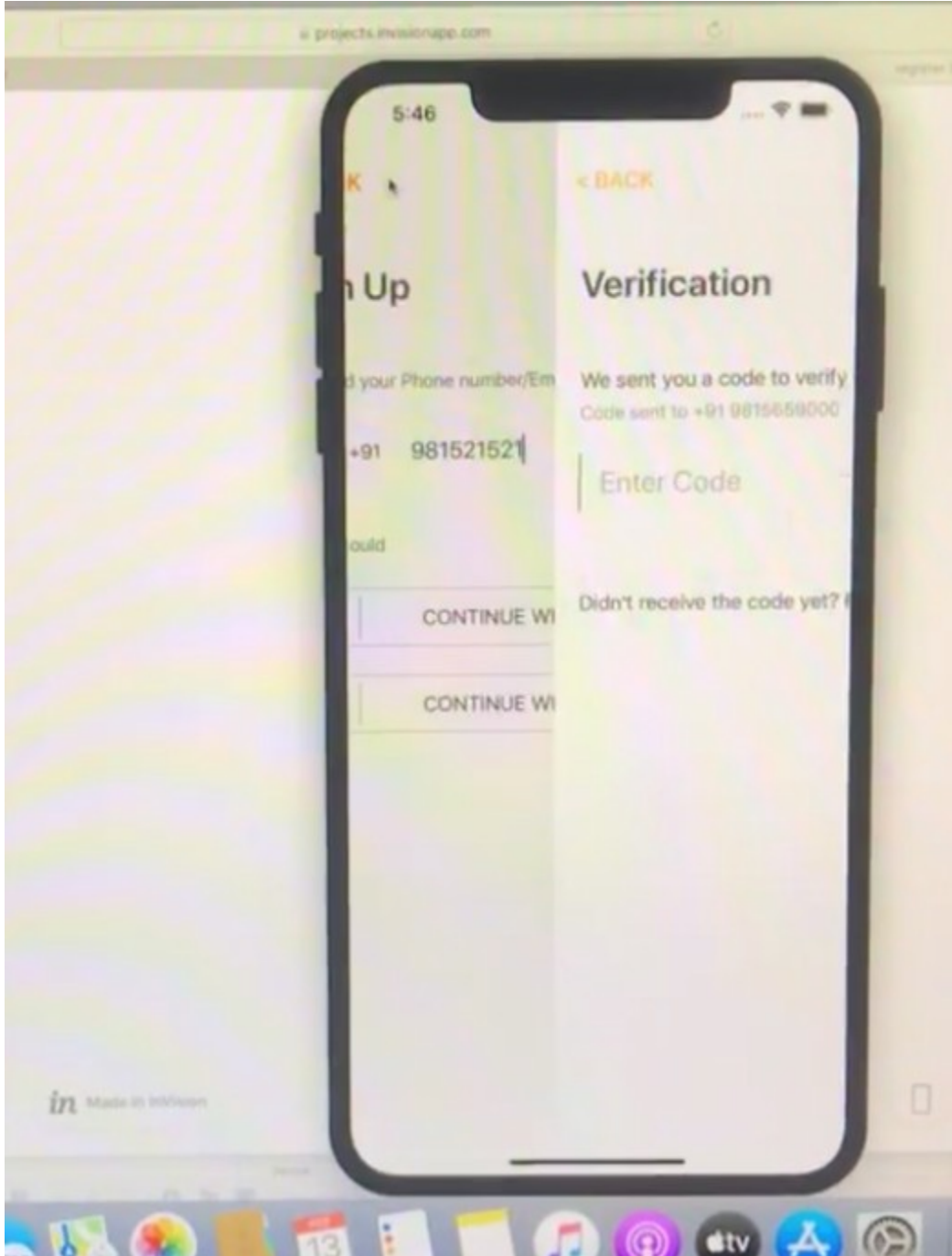


Figure 13 : page transferring using protocols

CONCLUSION

Learnt Swift language which is universally useful, multi-worldview, gathered programming language created by Apple Inc. for iOS, iPadOS, macOS, watchOS, tvOS, Linux, and z/OS.

Having a best experience in learning about the new technologies that are used to make iOS applications.

Learnt how API's models formation is done and being mapped.

Hands on experience in debugging the issues.

Experience to handle industry related tools and slowly gaining expertise on integrating them to fine tune a creative app.

Learnt using Xcode and VisualStudio

Experience of working on a Real time Application AIR BNB which has a market base

Implemented the MVC Architecture for my project and designed the UI alongwith fetching data for the design using Moya.

Experience of working on projects such as the coures app and Kashkol and learnt about the apps's transition from initial to its deployment stage.

Learnt how to integrate facebook login and apple login within an app as User friendly options

REFERENCES

- <https://swift.org/documentation/>
- <https://www.code-brew.com/>
- <https://developer.apple.com/xcode/>
- <https://medium.com/@hacknicity/how-to-switch-your-ios-app-and-scene-delegate>
- <https://developer.apple.com/documentation/objectivec>

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date:28/08/2020.....

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: KETAN BATRA

Department: CSE

Enrolment No : 161277

Contact No. - 9815659424

E-mail. - ketanbatra97@gmail.com

Name of the Supervisor: DR. PARDEEP KUMAR

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): iOS APP DEVELOPMENT

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

- Total No. of Pages = 37
- Total No. of Preliminary pages = 3
- Total No. of pages accommodate bibliography/references = 1

Ketan Batra

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at9.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

Pardeep Kumar

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Abstract & Chapters Details	
	<ul style="list-style-type: none">• All Preliminary Pages• Bibliography/ Images/Quotes• 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Page counts	
			File Size	

Please send your complete Thesis/Report in (PDF) & DOC (Word File) through your Supervisor/Guide at plagcheck.juit@gmail.com

Checked by
Name & Signature

Librarian

.....

Please send your complete Thesis/Report in (PDF) & DOC (Word File) through your Supervisor/Guide at plagcheck.juit@gmail.com