

PROJECT REPORT ON AIRBNB APP

Submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

Mukul Gupta (161067)

UNDER THE GUIDANCE OF

Mrs Gurleen Kaur

(Head of iOS Department, Code Brew Labs)

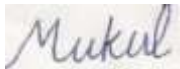


JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May 2020

DECLARATION

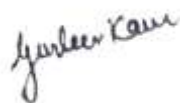
I hereby announce that the work reported in the B.Tech Project Report titled “**Airbnb app**” submitted at **Jaypee University of Information Technology, Wagnaghat, India** is an accurate record of my work carried out under the guidance of **Mrs Gurleen Kaur**. I have not submitted this work elsewhere for any other degree or diploma.



Mukul Gupta

161067

I announce that the Airbnb app project report is my unique work and has not been submitted to some other University.



Mrs Gurleen Kaur

Head of iOS Department

Code Brew Labs

Head of the Department/Project Coordinator

ACKNOWLEDGEMENT

I take this chance to express my sincere appreciation to the Director, Jaypee University of Information Technology, Solan, HP, for providing me with such a great opportunity for implementing the present work.

I would also like to thank my mentor **Dr. Shweta Pandit**, Assistant Professor (Senior Grade), Department of Electronics & Communication Engineering, for continuously guiding. This work wouldn't be possible without her guidance.

I would like to express my special thanks of gratitude towards my mentor **Mrs. Gurleen Kaur** for his guidance and valuable advice throughout the duration of my internship at Code Brew Labs, Chandigarh. Her untiring presence and ever encouraging attitude has been immensely valuable in keeping me motivated and helping me surmount all obstacles in my work. The knowledge gained from her was truly worthy.

I would also like to express my sincere thanks to **Mr. Aseem Ghavri** for his support and for providing me with the required facilities to carry out my internship.

Mukul Gupta

161067

LIST OF FIGURES

1)	Core OS layer	Figure (2.5.1)
2)	Core Service layer	Figure (2.5.2)
3)	Media layer	Figure (2.5.3)
4)	Cocoa Touch layer	Figure (2.5.4)
5)	Xcode	Figure (3.1.1)
6)	Storyboard and Interface Builder	Figure (3.3.1)
7)	Segues	Figure (3.3.2)
8)	Auto Layout Constraints	Figure (3.4.1)
9)	Use of Label, Button and Text Fields	Figure (3.5.1)
10)	Code 1	Figure (3.5.2)
11)	Code 2	Figure (3.5.3)
12)	Sliders	Figure (3.5.4)
13)	Code 3	Figure (3.5.5)
14)	Table View	Figure (3.5.6)
15)	Code 4	Figure (3.5.7)
16)	Collection View	Figure (3.5.8)
17)	Code 5	Figure (3.5.9)
18)	Sign up Page	Figure (4.2.1)
19)	Verification Page	Figure (4.2.2)
20)	Tab Bar	Figure (4.2.3)
21)	App View	Figure (4.2.4)
22)	View Controller Life Cycle	Figure (4.4.1)

LIST OF TABLES

1)	iOS Version History	Table (2.4.1)
2)	Xcode Version History	Table (3.2.1)
3)	UI Controls	Table (3.5.1)
4)	Views	Table (3.5.2)
5)	Navigation and Tab bars	Table (3.5.3)

LIST OF ACRONYMS AND ABBREVIATIONS

1	iOS	iPhone Operating System
2	OS	Operating System
3	SDK	Software Development Kit
4	MVC	Model View Controller
5	MVVM	Model View View Model
6	UI	User Interface
7	IDE	Integrated Development Environment
8	JS	Java Script
9	JSON	Java Script Object Notation
10	XML	Extensible Markup Language
11	GUI	Graphical User Interface
12	GDB	GNU Debugger
13	LLDB	Low Level Debugger
14	ARM	Acorn RISC Machine
15	GPS	Global Positioning System
16	API	Application Programming Interface
17	XIB	XML Interface Builder
18	DSP	Digital Signal Processing
19	URL	Universal Resource Locator
20	GCD	Greatest Common divisor
21	GCC	GNU Compiler Collection
22	GNU	GNU'S Not UNIX
23	LLVM	Low Level Virtual Machine
24	NIB	NextStep Interface Builder

ABSTRACT

iOS is Apple's mobile operating system. It is created and established initially for the iPhone and earlier organized on the iPod Touch and iPad as well. It is consequential from Mac OS. iOS consists of four technology coats: the Core Service coats, the Media coats, the Core Operating System coats, and the Cocoa Touch coats. iOS user interface is based upon direct manipulation using touch gestures. User interface control elements consist of buttons, switches, labels, text fields, table view, collection view, sliders. The OS includes gestures such as swipe, tap, pinch, etc. Apple does not allow the operating system to run on third-party hardware. The iOS Software Development Kit offers the whole thing we requisite to grow starting making iOS applications. It covers the code, material, and tackles we essential to mature, test, run, debug, and tune applications for iOS. XCode is an amazing tool for building iOS apps for all Apple platforms.

The Airbnb app is an iOS application. Designing and building a UI framework of this application using a navigation bar, tab bar, table view, collection view, buttons, image view, labels, etc. Implement design patterns using Model-View-Controller (MVC) design patterns. Use JavaScript Object Notation (JSON) inputs to put the data from backend and display it on the app using the Moya framework. JSON is a lightweight data-interchange format. It transmits data between a server and a web application. It provides the following services: book travel experiences, book holiday homes, and travel experiences for trips, book activities with local guides, hosts tourism-related activities.

TABLE OF CONTENTS

CAPTION	PAGE NO.
DECLARATION	ii
ACKNOWLEDGEMENT	iii
LIST OF FIGURES	iv
LIST OF TABLES	v
LIST OF ACRONYMS AND ABBREVIATIONS	vi
ABSTRACT	vii

CHAPTER-1: INTRODUCTION

- 1.1 About Company 1

CHAPTER-2: OVERVIEW OF IOS

- 2.1 About iOS 2
- 2.2 iOS History 3
- 2.3 iOS Tools 4
- 2.4 iOS Versions 5
- 2.5 iOS Architecture
 - 2.2.1 The Core OS layer 6
 - 2.2.2 The Core Service layer 7
 - 2.2.3 The Media layer 8
 - 2.2.4 The Cocoa Touch layer 9

CHAPTER-3: IOS DEVELOPMENT

- 3.1 XCode 10
- 3.2 XCode Versions 12
- 3.3 Storyboard and Interface Builder 13

• 3.4 Auto Layout Constraints	15
• 3.5 Object Library	16
• 3.6 MVC and MVVM23	
• 3.7 XCode Features	24
• 3.8 XCode Components	25
• 3.9 Action and outlets	27
• 3.10 Application delegate Lifecycle	28

CHAPTER-4: AIRBNB APP

• 4.1 About App	29
• 4.2 Interface Builder Guide	30
• 4.3 JSON, Moya Framework, Kingfisher Framework	33
• 4.4 View Controller Life Cycle	35

CHAPTER-5: SWIFT PROGRAMMING

• 5.1 About Swift	37
• 5.2 Simple Values and Control Flow	38
• 5.3 Function and Closure, Object and Class	40
• 5.4 Protocol and Generics	42

REFERENCES

PLAGIARISM REPORT

CHAPTER-1: INTRODUCTION

1.1 About Company

Code Brew Labs is an IT Solutions Company that was started in 2013 to reinvent and transmute industries with the power of forward-thinking technology. It is a transnational corporation that distributes IT Services, Product Development, and Digital Transformation. There are 200-300 members in the company. It is headquartered in Chandigarh, India, and has stretched across the globe with offices in Dubai, San Francisco, and Mexico.

Company key technologies include Mobile App Development, Web Development, Blockchain App Development, UI/UX Design Solutions, iPhone App Development, react native, game development, and Data Analytics Solutions.

It supports Startups and Enterprises with Product Development and Digital Transformation. Company work on the latest technologies like Node JS, React JS, Angular JS, etc. They worked on more than 300+ Apps across different business domains. The Company's chief field is a mobile-based application like Android Applications and iOS applications. Specialties: Mobile App Enlargement, iPhone App Enlargement, Android App Enlargement, 2D / 3D game enlargement, Mobile approach, Minimum workable Product for Startups, App Monetization and Appointment, App Store Optimization and React Native.

CHAPTER-2: OVERVIEW OF IOS

2.1 About iOS

iOS is the iPhone operating system. It is the core of iPhone, iPad touch, and iPad devices. iOS is established by Apple. It is used to grow applications for Apple products such as iPhone, iPad, Apple TV, etc. It runs on the Mac Operating System (OS) X. Apple does not permit third-party hardware to ride his operating system. Many of the technologies and tools used for development on the platform have their back ground in Mac OS X. iOS does not involve experience developing Mac OS applications. The iOS Software Development Kit (SDK) delivers everything we need to get happening building iOS applications.

The iPhone SDK contains codes, information, and tools that we require to develop, test, run, debug and tune applications for iOS. We used XCode Software to build iOS applications. XCode software delivers the editing, compilation, and debugging environment for code. XCodeembraces many objects libraries like buttons, labels, text fields, table view, collection view, navigation bar, tab bar, switches, sliders, etc. XCode also includes the initiation point for testing our applications on an iOS device. It also includes the iPhone simulators platform for testing our applications.XCode also include Core Data.

We use Swift language to builtiOS applications. IOS hascountless applications like Core Data, Cocoa touch. IOS uses many frameworks to put API data from the backend and display it in the end. It uses the Moya framework and Alomafire framework to put data from the backend. It also usesthe Kingfisher library to download the image from the internet. IOS introducesa new design pattern such as Model-View-Controller (MVC) and Model-View-View-Model (MVVC). There are many versions of iOS.

2.2 iOS History

iOS is a mobile operating system and established by Apple Inc. It was established on 29 June 2007. It all in progress way back in 2007 when Steve Jobs revealed the iPhone. The iPhone could hardly grip 3G, it required the copy and paste features, and could not run third-party apps. It was a huge step in the greatest direction. It is the second most used mobile operating system after Android.

The iOS offers the following features: Multitasking, iCloud Facility, In-App purchase, Apple Pay, Home Screen, Notification Centre, Game Centre, Bluetooth, Orientations, Camera Integration, Location Services, Maps, Notifications Centre, Accelerometer, Gyroscope, Powerful APIs, GPS, High-End Processor, Accessibility, Email, Contacts, Web pages, and messages, etc.

The Second version of iOS familiarizes the App Store. App Store permits people to download Third-Party applications and new games. Along with these features, the new phone had extended customizations to make each one single to the owner. Simple capabilities like copying and pasting became corporate. TV shows and audiobooks became presented through iTunes. IOS 5 also came to the rise of voice control, which now relates with other apps. IOS introduce new features like wireless headphones, Siri intelligence. As of June 7, 2010, Apple's App Store delivers more than 255,000 iOS applications, which had collectively been downloaded more than five billion times.

It is programmed in C, C++, and Objective C and resulting for the Mac OS. The kernel type is Hybrid kernel architecture. IOS is presently supported by 34 languages universal. The Stable release in version 5.0.1 for iPhone 4 S, and iPhone, iPhone 3G, iPhone 3GS, iPhone 4, iPod Touch, and iPad, and the unstable release is version 5.1. On October 17, 2007, Apple released a new native Software Development Kit. On March 6, 2008, Apple released the first beta, and change name with a new name for the operating system: iPhone OS.

2.3 iOS Tools

1) XCode:

XCode is an Integrated Development Environment (IDE) for Mac OS. It covering software development tools developed by Apple. It figures software for Mac OS, iOS, iPadOS, watch OS, TV OS. This software is existing in the Mac App Store free of charge. XCode supports programming languages like C, C++, java, Objective C, Objective C++, Python, ruby, swift. Xcode includes graphic user interfaces. Provides multiple iOS app templates. An Integrated Development Environment (IDE) succeeds to edit, compile, run, and debug source code. Instruments are the debugging tools and runtime performance analysis. Steps: Create a single view application and select a storyboard. Select MainStoryboard where we find single view controller. Designing and building UI framework using protocols and delegates, Model-View-View-Controller, navigation bars, tab bar, table view, collection view, buttons, image view, labels, etc. Implement design patterns using Model-View-Controller (MVC) design patterns.

2) Interface Builder:

Interface Builder is the chief part of Xcode. Interface Builder is a software development tool for Apple Mac OS operating system. It licenses Cocoa developers to create a UI interfaces for applications. The interface is deposited as a .nib file or as an XML based xib file. Use the drag and drop method for the user interface. Interface Builder marks outlet of the text field, labels, buttons, sliders, switches, table view, collection view, tab view, and navigation view. Also, make the action of buttons. Use delegates, data source in table view cell. Use delegates, data sources, delegate flow layout in the collection view. Drag and drop all UI components. Use multiple View Controllers.

3) iPhone Simulator:

The Apple iOS Simulator is a tool for mounting and prototyping our mobile app. IOS simulator can be used to test apps. It provides the skill to test your software on iPhone and iPads. There are many iPhone simulators of different resolutions.

2.4 iOS Version

Table 2.4.1: (iOS Version History)

29/06/2007	Primary OS 1.5 version was developed by Apple.
11/06/2008	The third iPhone OS 2.0 Version was developed.
17/07/2009	The third iPhone OS 3.0 Version was developed.
21/06/2010	IOS 4 was developed and delivered 1200 API topographies.
06/06/2011	IOS 5 was developed and include I Message one-to-one among double strategies joined with iCloud features.
11/06/2012	IOS 6 was developed but this type is not accessible for community usage.
19/09 2012	IOS 6 was developed and become communal.
10/06/2013	IOS 7 was developed and the feature of iOS 7 are solitary for dual iPhones.
02/06/2014	IOS 8 was developed. It is the primary method that is accessible for community beta testing.
08/06/2015	IOS 9 version was developed and delivered sustenance to promotion iOS 8 to iOS 9.
13/06/2016	Apple broadcast iOS 10 version and unconfined to the community on September 13, 2016
05/06/2017	Apple broadcast iOS 11 version and unconfined to the community on September 19, 2017
04/06/2018	Apple broadcast iOS 12 version and unconfined to the community on September 17, 2018
03/06/2019	Apple broadcast iOS 13 version and unconfined to the community on September 19, 2019

2.5 iOS Architecture

2.5.1 The Core OS Layer

The Fundamental Operating System Layer is the most recent layer of the iOS heap. This sits straight on topmost of the device hardware. The Core OS layer offers the low-level topographies that most other technologies are constructed upon. We do not use these comprehensions straight in our applications, they are mostly be used by other frameworks. This is the most important layer for the charge of handling memory-allocating and releasing memory once the application has finished with it. It also takesprecaution of file schemeresponsibilities, handling networks, and other operating system tasks. It directly interrelates with the hardware.

This layer contains low-level features:

Accelerate: This frameworks introduced in iOS 4.0. It contains an interface for handing out DSP, rectilinear algebra, and image handing out calculations. It also offers hardware optimized C-based API for performing complex. Its most collective use is in gaming apps.

Core Bluetooth: This framework allows developers to cooperate with Bluetooth Low-Energy accessories.

Security: This framework offers all the sanctuary interfaces. This framework comprisesboundaries for community and isolated keys and faith policies.

System: This framework includes the Operating System kernel. The OS kernel is accountable for memory distribution, process life cycle society, input/output, and file system admittance.

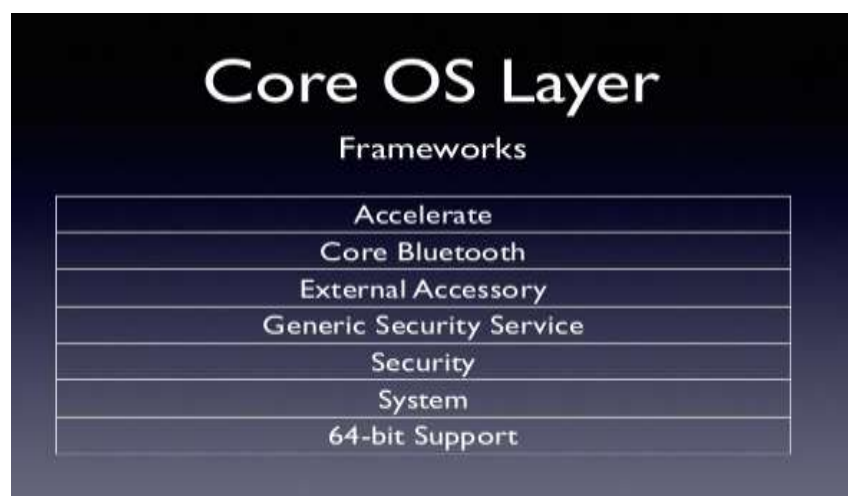


Figure 2.5.1: (Core OS layer)

2.5.2 The Core Service Layer

The Core Service coating is the second last layer of the iOS heap. The Core Service coating is the necessary schemefacilities that all solicitations use. The Core Service layer agreesinventors to admittance the services opened by apple for the iOS application growth. This layer used for retrievingrecords, low-level informationkinds, network holes. These fundamental boundaries are frequently C-based.

Some of the most important frameworks:

Address book: This agendabounces programmatic admittance to associatescatalogue of the user.

Cloud Kit: This agendabounces anintermediate for affectinginformation between our application and iCloud. Introduced in iOS 5.

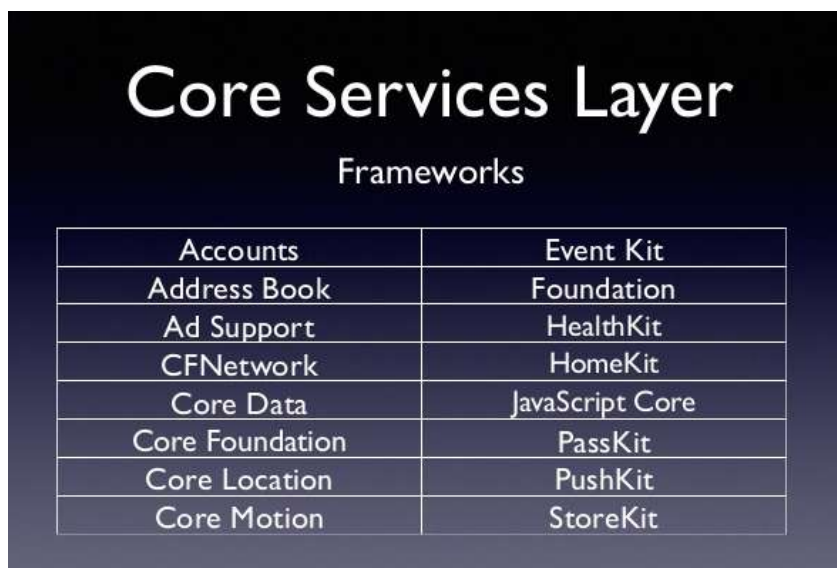
Core Data Framework: This framework gives knowledge for handling the informationPrototypical of the application.

Core Location: This agendastretcheslocality and sloganmaterial of apps.

Health kit: This framework conductfitness-related material of the user.

Social: This framework admittance the user's publicbroadcasting accounts.

Store kit: This agenda gives sustenance for the buying of facilities from iOS applications.



Core Services Layer	
Frameworks	
Accounts	Event Kit
Address Book	Foundation
Ad Support	HealthKit
CFNetwork	HomeKit
Core Data	JavaScript Core
Core Foundation	PassKit
Core Location	PushKit
Core Motion	StoreKit

Figure 2.5.2: (Core Service Layer)

2.5.3 The Media Layer

The media coating comprises graphics, a dial, and videocassette equipment. It allows generating the greatest multimedia involvement accessible on a mobile device with the frameworks.

Visual Framework:

UI Kit visual: This framework labels complex sustenance for scheming images and also cast-off for animation on the content of the views.

Core visual: This framework is the natural representation appliance for iOS applications. It also supports custom 2D vector.

Core Animation: This framework is an opening equipment that enhances the animation involvement.

Core Images: This framework bounces forward-thinking sustenance for monitoring video and stationary images.

Audio Framework:

AV Foundation: This framework supervision the soundtrack and reproduction of audio and video.

Video Framework:

AV Kit: This framework bounces interfaces for awarding videos.

AV Foundation: This framework bounces forward-thinking video playback and recording capability.

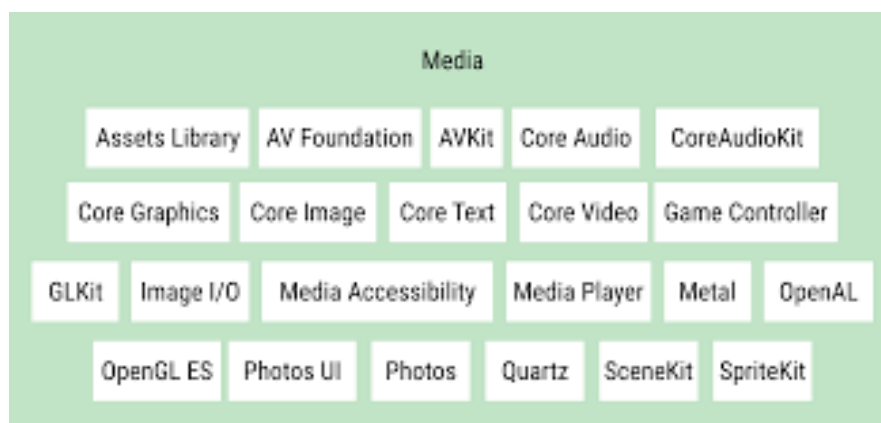


Figure 2.5.3: (Media layer)

2.5.4 The Cocoa Touch Layer

The Cocoa Touch coating is the first layer of the iOS heap. It provides the key frameworks for evolving iOS applications. This layer support multiple equipment such astouch-based input, multitasking, push notification, and many complex system services. It delivers complex features such as storyboards, Certificatesustenance, Multitasking, Production, Data Defense, Apple push notifications facility, and File sharing.

EventKit: This framework bounces view managers for presentation the customary system interfaces.

GameKit: This agendaoutfitssustenance for Game Edge which permits users to share their game-related evidence online.

iAd: This framework agreesthe user to transport banner-based announcements.

MapKit: This framework bounces a scrollable map that includes our user boundary of the application.

PushKit: This framework offersregisteringsustenance for VoIP applications.

Twitter: This framework provisions a UI for engendering tweets and sustenance for forming URLs to admittance the Twitter service.

UIKit: This framework spread over visual application in iOS. A Most significantpurpose of the UI Kit framework: Multitasking sustenance- Basic application management.

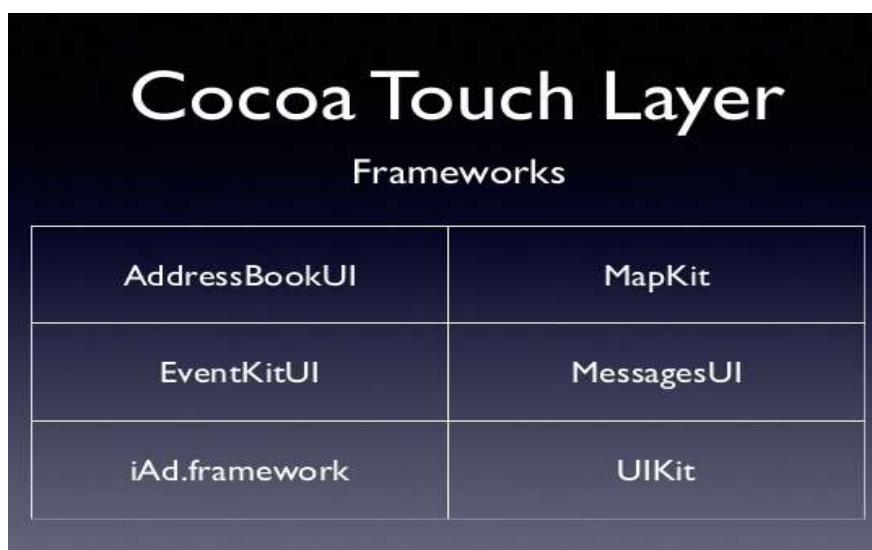


Figure 2.5.4: (Cocoa Touch layer)

CHAPTER-3: IOS DEVELOPMENT

3.1 XCode

Xcode is an integrated development environment (IDE). It delivers everything the outfits that we need to generate and grow iOS schemes and foundation files. It is developed to work on the Mac Operating system. It succeeds to edit, run, debug and execute our source code. Xcode introduces numerous features to build iOS applications easier. It is developed by Apple. It develops software for Mac OS, Tv OS, and Watch OS. Xcode provides an influential user interface for designing and building software development projects on the MAC platform. It comprises tools such as a debugger, text editor, bundle, Interface builder, Object details, etc. It provides all the elements used to design and build iOS applications. Elements include source files, images, user interface specifications, sounds, and links to supporting frameworks and libraries. It includes many UI elements such as buttons, text fields, labels, sliders, switch, tab bar view, navigation bar, table view, collection view, and so on. To create a new application, open Xcode, and start by creating new projects. A project accomplishes everything of the information related to our presentation including the source code, construct settings.

- 1) A scheme organization system for significant software products.
- 2) A combined editor for generating nib files, xib files, and storyboard.
- 3) Code-editing environments for coloring, code completion, and symbol indexing.
- 4) Documentation viewer for viewing and searching Apple documentation.
- 5) GNU Compiler Collection (GCC) compilers supportive C, C++, Objective C, Objective C++, and other programming languages.
- 6) Clang and LLVM sustenance for C, C++, and Objective C.
- 7) Sustenance for incorporated source-code management
- 8) Sustenance for management iOS development devices.

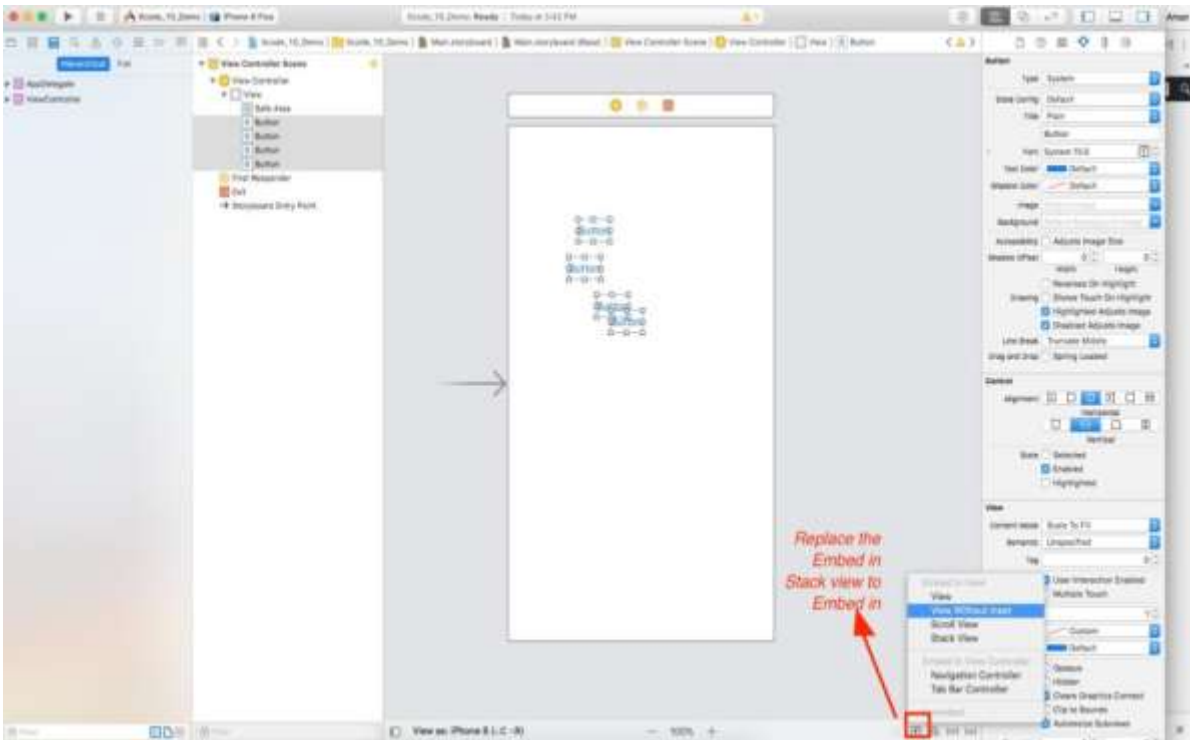


Figure 3.1.1: (Xcode)

3.2 XCode Versions

Table 3.2.1: (Xcode Version History)

SN	Release year	Operating System	Features
1.x series	2003	MacOS 10.3+	Based on the project builder. Improved debugger and better code compiler.
2.x series	2005	MacOS 10.4+	Includes the Quartz Composer, better code sense and Ant support.
3.x series	2007	MacOS 10.5+	Includes DTrace debugging tool, context-sensitive documentation, and Objective C 2.0 with garbage collection.
4.x series	2011	MacOS 10.6.8+	Includes Xcodeinsertion tools and bordermanufacturer into one application.
5.x series	2013	MacOS 10.8+	Added support for iOS 7 SDK and generating 64-bit ARM code for iOS 7.
6.x series	2014	MacOS 10.9.4+	Includes the support for all new programming language of apple i.e., Swift. Also, support for playgrounds.
7.x series	2015	MacOS 10.10.3+	Provide support for live debugging tools. Also, provide support for Swift 2.
8.x series	2016	MacOS 10.11.5+	Provide support for Swift 3
9.x series	2017	MacOS 10.12.6+	Delivermaintenance for Swift 4 and metal 2 for OS X.
10.x series	2018	MacOS 10.13.6+	Provide support for the dark mode and added new topographies in swift 4.2 and metallic 2.1
11.x series	2019	MacOS 10.14.4+	Delivermaintenance for the new topographies in swift 5.1. Also support building iPad applications.

3.3 Storyboard and Interface Builder

A Storyboard is a graphical symbol of the user interface (UI) of an iOS application. It is first hosted in iOS 5 to save time in building and developing user interfaces for the iOS applications. A storyboard is a sequence of screens. It characterizes the View Controller and the views. Transitions and connection between two storyboard screens need a segue object. The storyboard is built using a graphical editor which is provided by Xcode. We can figure and design user interfaces of the iOS application by adding user interface widgets from the media library such as buttons, views, table views, text fields, navigation bars, tab bars, collection views, labels, etc. Standard editor of the project in which the project files are edited. It contains the information about the files. Assistant Editor is mainly used to create outlets of the storyboard components (Textfield, Label, etc.)

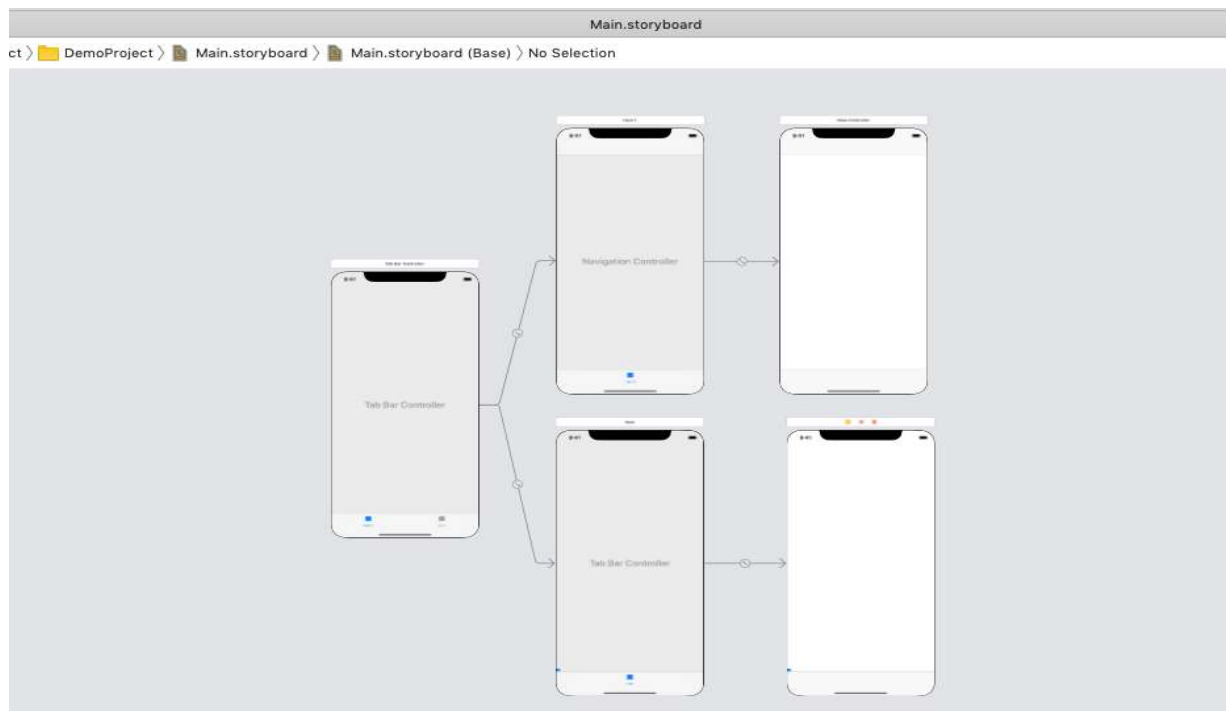


Figure 3.3.1: (Storyboard and Interface builder)

Segues: It is used to make influences between two storyboard screens. We can set the transitions like push text in the first screen to the second screen or second screen to the first screen. It is an arrow-like object. We can pass data between screens with the inbuilt method prepare For Segue: Sender.The file inspector shows the full information about the corresponding swift file opened in the standard editor. It contains the Name, Type, Location, and the interface builder document-related information.Identity Inspector is mainly used when we need to work with the storyboard. It shows information about storyboard components (View Controllers) and their corresponding Swift class files.

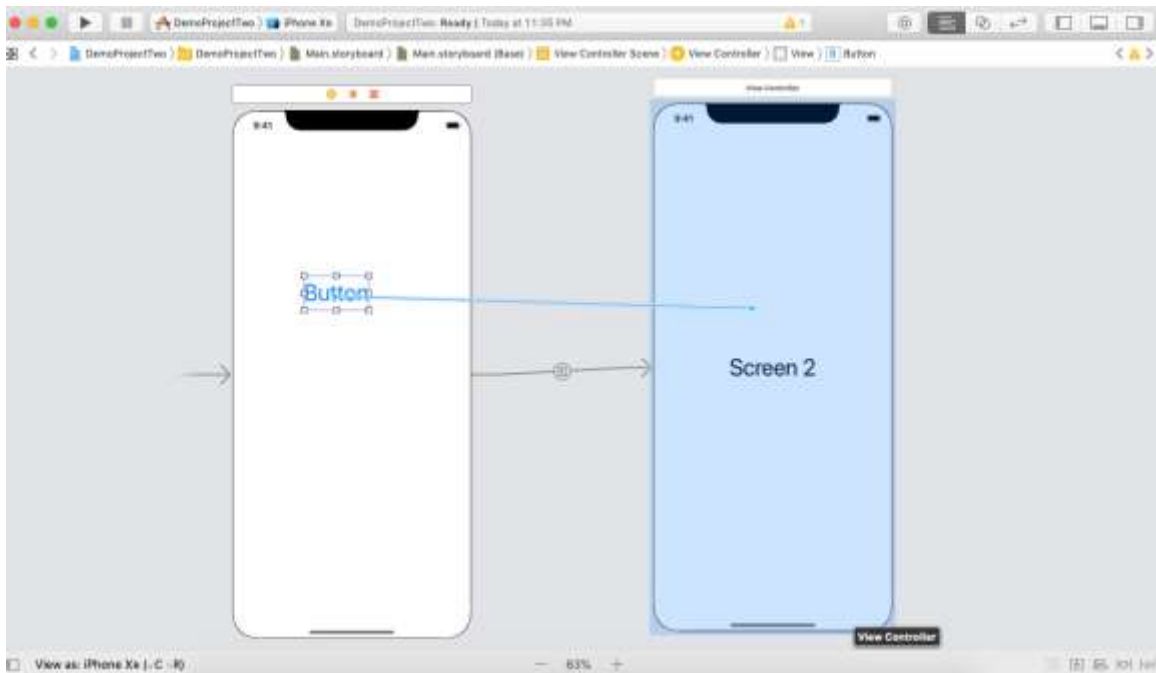


Figure 3.3.2: (Segues)

3.4 Auto Layout Constraints

Auto Layout calculates the magnitude and location of all the views in our view pyramid. For Instance, I can checks a button so that it is vertically centered with an table view and so that the button's bottom superiority always remains 16 points below the table edge. Settings User Interface view frames rules slightly than numbers. No use of the auto-resizing mask. We put some guidelines on the views in their place no stuff what bounds are available. There is a very powerful API. We almost set up rules in Xcode graphically i.e. storyboard or xib. Many things distress the size of the area available to put views: Multiple iOS versions, Screen sizes, to support autorotation, Dynamic content, Localization.

It couriers rules for the layout of elements in our interface and applies many attributes like left, right, top, bottom, leading, trailing, width, height, center X, center Y, and Baseline. Set constraints using Control-Drag method. Also, set constraints using Align, and pin menus. The size of the view depends on the content it displays. User Interface Label has an ideal height depend on the font. Also, anideal width based on the font and the text it displays. A plain user view has neither preferred height nor a preferred width. A user interface's progress view only has a height based, but no preferred width.

Layout phase:

Triggered via set needs layout.

Refresh via layout if needed

Applies frames from the previous step to each view

Use add missing constraints to set of constraints

Use Reset to suggested constraints to remove erroneous constraints.

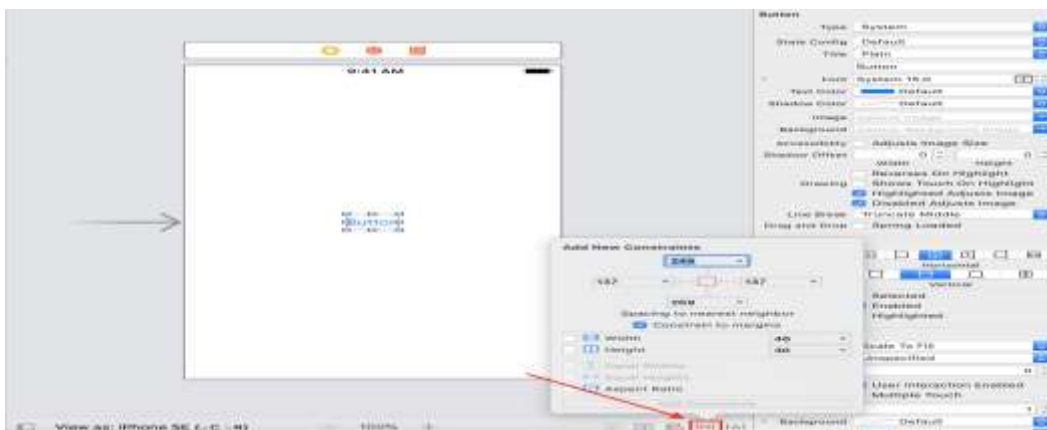


Figure 3.4.1: (Auto Layout Constraints)

3.5 Object Library

1) UI Controls

Table 3.5.1: (UI Controls)

S.No.	Components	Uses
1	Label	Signifies a class of views that display one or more lines of read-only texts.
2	Button	Controller that enables the user to interact with the application. Used to trigger the events performed by the user.
3	Text Field	Express an object which is used to display an editable transcriptzone in the interface. Used to get the manuscript-based input from the user.
4	Slider	It provides the adjoining range of values on a single scale to the user, out of which the user is computed to select a single value.
5	Switch	It provides binary choices to the user moreover on or off. It manages properties and method defines in the UI Switch class.

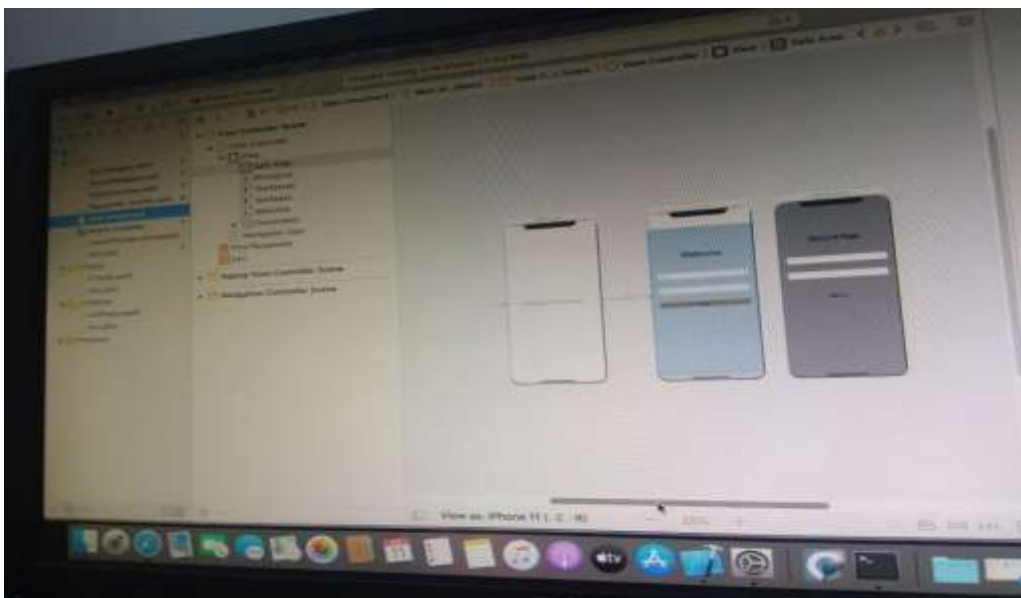


Figure 3.5.1: (Use of Label, Button and Text Fields)

```

1 //
2
3 // MARK: - IMPORTS
4 import UIKit
5
6 // MARK: - CLASS DECLARATION
7 class ViewController: UIViewController {
8
9     // MARK: - OUTLETS
10    @IBOutlet weak var textFieldEmail: UITextField!
11    @IBOutlet weak var btnNext: UIButton!
12    @IBOutlet weak var textFieldPass: UITextField!
13
14    // MARK: - PROPERTIES
15
16    // MARK: - VIEW CYCLE
17    override func viewDidLoad() {
18        super.viewDidLoad()
19    }
20
21    // MARK: - ACTIONS
22
23    // MARK: - METHODS
24    @IBAction func btnNext(_ sender: UIButton) {
25
26    }
27
28 }
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figure 3.5.2: (Code 1)

```

1 // Created by Interns on 2020-08-20 at 10:10.
2 // Copyright © 2020 Interns. All rights reserved.
3 //
4 // MARK: - IMPORTS
5 import UIKit
6
7 // MARK: - CLASS DECLARATION
8 class ViewController: UIViewController {
9
10    // MARK: - OUTLETS
11    @IBOutlet weak var textFieldEmail: UITextField!
12    @IBOutlet weak var textFieldPass: UITextField!
13    @IBOutlet weak var btnSignup: UIButton!
14
15    // MARK: - VIEW CYCLE
16    override func viewDidLoad() {
17        super.viewDidLoad()
18        // Do any additional setup after loading the view.
19    }
20
21    // MARK: - ACTIONS
22    @IBAction func btnNext(_ sender: UIButton) {
23        let signupVC = self.storyboard?.instantiateViewController(withIdentifier:
24            "SignupViewController") as! SignupViewController
25
26        signupVC.email = textFieldEmail.text!
27        signupVC.pass = textFieldPass.text!
28
29        self.navigationController?.pushViewController(signupVC, animated: true)
30    }
31
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Figure 3.5.3: (Code 2)

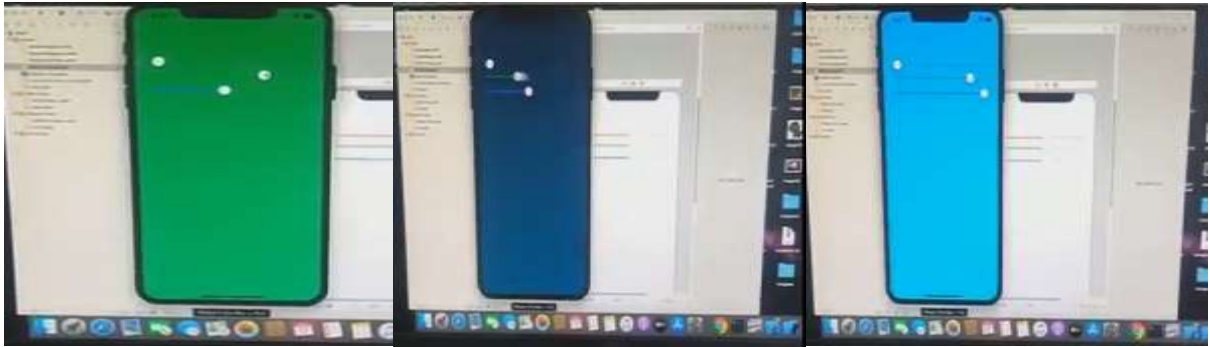


Figure 3.5.4: (Sliders)

```
ViewController.swift
slider
ViewController.swift | No Selection
1 //
2 // ViewController.swift
3 // slider
4 //
5 // Created by Interns on 17/02/20.
6 // Copyright © 2020 Interns. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var red: UISlider!
14     @IBOutlet weak var green: UISlider!
15     @IBOutlet weak var blue: UISlider!
16
17     override func viewDidLoad() {
18         super.viewDidLoad()
19         // Do any additional setup after loading the view.
20
21     }
22
23     @IBAction func updateBackgroundColor() {
24         let r = CGFloat(red.value)
25         let g = CGFloat(green.value)
26         let b = CGFloat(blue.value)
27
28         view.backgroundColor = UIColor(red: r, green: g, blue: b, alpha: 1)
29     }
30
31 }
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Figure 3.5.5: (Code 3)

2) Views

Table 3.5.2: (Views)

S.No.	Views	Uses
1	UI View	It can create and achieve the rectangular area on the screen. There are many numbers of views inside a view to creating a tree-like structure. UI View class can inherit from UI Kit.
2	Table View	It can organize the data using rows in a single column. It is used in most of the applications like contacts, Instagram, Facebook, Twitter, etc. It displays the single column containing vertically Scrolling content.
3	Collection View	Offerings the well-organized gathering of information objects in the customizable blueprints. It displays the information in the technique of a grid-iron layout. It displays both horizontal and vertical scroll. It is used in most of the application like Snapchat, Instagram, etc.
4	Scroll View	It shows the content that does not fit into the screen. It tolerates the user to drag the area of the content. It tolerates the scrolling and zooming of its contained views.
5	Image View	It can be defined as an object that can display images on the interface of the application. Exhibition the contents of various image files such as JPG or PNG.

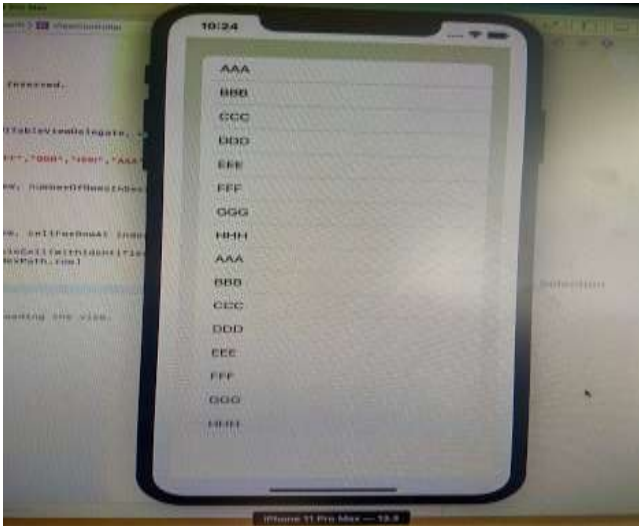


Figure 3.5.6: (Table View)

```
1 //
2 // ViewController.swift
3 // Tableview
4 //
5 // Created by Interns on 17/02/20.
6 // Copyright © 2020 Interns. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
12
13     let array1 =
14         ["AAA", "BBB", "CCC", "DDD", "EEE", "FFF", "GGG", "HHH", "AAA", "BBB", "CCC", "DDD", "EEE",
15          "FFF", "GGG", "HHH"]
16
17     func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
18         return(array1.count)
19     }
20
21     func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
22     UITableViewCell {
23         let cell = tableView.dequeueReusableCell(withIdentifier: "cell")
24         cell?.textLabel?.text = array1[indexPath.row]
25         return cell!
26     }
27
28     override func viewDidLoad() {
29         super.viewDidLoad()
30         // Do any additional setup after loading the view.
31     }
32 }
```

Figure 3.5.7: (Code 4)



Figure 3.5.8: (Collection View)

```

import Foundation
import UIKit

class Insta {

    var img : UIImage?

    init(_ primg : UIImage?) {
        img = _primg
    }

    class func getfeed() -> [Insta] {

        return [Insta.init(1),
                Insta.init(2),
                Insta.init(3),
                Insta.init(4),
                Insta.init(5),
                Insta.init(6),
                Insta.init(7),
                Insta.init(8),
                Insta.init(9),
                Insta.init(10),
                Insta.init(11),
                Insta.init(12),
                Insta.init(13),
                Insta.init(14),
                Insta.init(15),
                Insta.init(16),
                Insta.init(17),
                Insta.init(18)]
    }
}

extension ViewController : UICollectionViewDelegate, UICollectionViewDataSource,
UICollectionViewDelegateFlowLayout {

    func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection:
    Int) -> Int {

        return 18
    }

    func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath:
    IndexPath) -> UICollectionViewCell {

        let cell = collectionView.dequeueReusableCell(withReuseIdentifier:
        "SampleCell", for: indexPath)

        return cell
    }

    func collectionView(_ collectionView: UICollectionView, layout:
    UICollectionViewLayout, sizeForItemAt indexPath:
    IndexPath) -> CGSize {

        let width = (UIScreen.main.bounds.width-52)/3
        return CGSize.init(width: width, height: width)
    }

    func collectionView(_ collectionView: UICollectionView, layout:
    UICollectionViewLayout, insetForSectionAt section: Int)
    -> UIEdgeInsets {

        return UIEdgeInsets.init(top: 8, left: 8, bottom: 8, right: 8)
    }
}

```

Figure 3.5.9: (Code 5)

3) Navigation and Tab bars:

Table 3.5.3: (Navigation and Tab bars)

S.No.	Bars	Uses
1	Navigation Bar	It performs on the bottom of the screen. It is not set in stone. It can modify the layout and button order. It disappear entirely and use widgets to navigate our phone.
2	Tab Bar	It is used to constitute and display one or more bar button items. It choice between different subtasks, views or models in the application.

3.6 MVC and MVVM

MVC Stands for Model, View, and Controller.

Model: It grips the data of the application. It is used to analyze response and request with the server's API.

View: It represents what the user sees in the device. It is preserved as the face of the application. It does not contain any domain-specific logic.

Controller: It acts as a negotiator between the Model and the View via the delegate pattern. They should not be any straight relation between the View and the Model. The controller updates the view based on any changes in the Model. User updates or enters any information in the view.

MVVM stands for Model, View, and View-Model.

Model: It holds the data of the application. Responsible for data passing.

View: It displays the model data on the screen counting the controls and the elements. It is highly reusable.

View-Model: Transforms the model data into the values and passes it to the view. It covers business logic. All the business logic, including data manipulation, take place at View-Model.

3.7 Xcode Features

1) Standard Editor:

It contains all the useful material about the project files. This editor is used to edit project files.

2) Assistant Editor:

It is mainly used to generate actions and outlets of the storyboard components i.e. Text field, labels, button, etc.

3) Project Navigator:

It shows the file construction of the project. It is used to circumnavigate throughout the whole project.

4) File Inspector:

It displays all information about the swift file released in the standard editor. It contains the type, location, Name, and the interface builder certificate-related information.

5) Identity Inspector:

It is mainly used when we want to work on the storyboards. It also shows material about storyboard components and their equivalent swift class files.

6) Attribute Inspector:

It gives some attributes to the corresponding UI View in the main storyboard like content mode, interaction, tags, background color, font size, font color, etc.

7) Size Inspector:

It provides information about the size constraints given to the view while designing using a storyboard.

8) Connection Inspector:

It shows the information about the connections of the storyboard UI View to the Swift class files.

3.8Xcode Components

1) UIViewController:

A View Controller is a container in the storyboard on which we can draw the content views to develop iOS applications.

2) TableViewController:

In which we have additional a table view to the UIViewController class. In which we have to select a prototype cell which represents the number of section in the table view.

3) CollectionViewCntroller:

This content view used to represent the collection of items where the UICollectionViewCell represents the number of rows in which we have added the items.

4) UIView:

It is the rectangular area on the screen in which we have added any number of views inside this rectangular area to create a hierarchical structure of the UIViews.

5) TableView:

It can be defined as the view which can arrange the data using rows in a single column. We can horizontally scroll the TableView. It is used in many applications like contacts in the mobile, WhatsApp, Facebook, Instagram, etc.

6) CollectionView:

It offerings the well-orderedgathering of data items in the customizable arrangements. It displays the data in the formula of a grid-iron layout. We can horizontally and vertically scroll the collectionView. It also used in many applications like Instagram, Snapchat, Facebook, etc.

7) TabBar:

A tabbar is used to display and organize one or more bar button items in a tabbar for selecting between different views, modes, and subtasks in the iOS application.

8) Navigator Bar:

It contains the navigation items typically bar button items. It is mainly used to navigate within the hierarchy of screens.

9) ActivityIndicator View:

It is a type of content view that can be used to show that a task is in progress. It is mainly used in-network calls.

10) ImageView:

It can be defined as an object that can display the image on the interface or the content View of the iOS applications.

11) WebView:

It can be defined as an object which can display the interactive web content and load HTML strings.

12) MapView:

It is the object that can display an embeddable map interface. It also shows the user current location on the map.

13) Label:

It represents a hierarchy of views that display one or more lines of read-only texts. We can set the font, color, size of the label in the iOS application.

14) Text Fields:

It used to display an editable typescriptzone in the interface. It also used to get the text-based input data from the user. We can change the color, font, size, margin of the text fields.

15) Button:

It enables the user to interact with the application. It is used to trigger the events performed by the user. We can give any color to button also change the font, size of the button.

3.9 Action and Outlets:

We can make the outlets of many storyboard components like text field, labels, buttons, slider, etc. We also make the Action of many storyboard components like buttons, slider, etc. Action is represented by `IBAction` where `ib` is interface builder. Outlets are represented by `IBOutlet` where `ib` is interface builder.

Steps involved:

- 1) Select our iPhone application
- 2) Select the `ViewController` file
- 3) Select the UI components from the Xcode library.
- 4) Drag and Drop the UI components to our content view in the main storyboard.
- 5) Right click on the UI components and select, hold and the drag the new referencing outlet.
- 6) Drop in the `ViewController` in between the curly braces.
- 7) Type the name for the outlets. Click connect and the `IBOutlet` will be connected.
- 8) Similarly, to add an action, right click on the UI components and drag the new referencing action.
- 9) Drop it in the `ViewController` file and name it.
- 10) Click Connect and the `IBAction` will be connected.

Syntax of outlet:

```
@IBOutlet weak var labelTxt: UILabel!
```

```
@IBOutlet weak var buttonOutlet: UIButton!
```

Syntax of action:

```
@IBAction func buttonAction(sender: AnyObject) {  
labelTxt.text = "Project Report"  
}
```

3.10 Application delegate Life Cycle

1) Application: willFinishLaunchingWithOptions:

This method is for the preliminary application Setup. Storyboards have even now been burdened at this opinion.

2) Application: didFinishLaunchingWithOptions:

This method is called when the application has finished launching and reconstructed the state and do eventual initialization such as generating the user interface.

3) Application: willEnterForeground:

It is called after application: didFinishLaunchingWithOptions become active again after receiving system interruption.

4) Application: didBecomeActive:

This method is called when the application enters the foreground and is active.

5) Application: willResignActive:

This method is called when the application is changes its state to inactive.

6) Application: didEnterBackground:

This method is called when the app enters a background state after becoming inactive. The application gets finished after that.

7) Application: willTerminated:

It is called when the app is about to be eliminated from memory. Calls any final cleanups here.

CHAPTER-4: AIRBNB APP

4.1 About app

Airbnb is an iOS application. This app has many landscapes like travel experiences, book holiday homes, for travelers, and hosts. Find new places and travel adventures to go far away and near to you. Also, admittance to rental holiday homes, new involvements, and places to visit all round the world. Book the whole thing for our trips, or start receiving money by sharing our extra space.

Book Travel involvements: Go on local trips directed by internal professionals. It's for manifold days. Explore the alternative music scene, hunt for animals in the forest.

Book Long weekend homes: Take for over 1 thousand holiday home rent payment across 10+ nations, Search by value, services, neighborhood, and more.

Standard Destinations: Involvement the beautiful visions and find local monitors for the perfect holidays. Discover popular cities.

For the traveler: Book long weekend home rentals and travel involvements for our next journey, family holiday, or work trips. Save our much loved rental homes, involvements, and spaces. It invites friends and family to strategy the journeys. Enhance events and experiences.

For the host: Share your additional space or chief involvements that make your city abundant. Inform your catalogue and calendar obtainability.

4.2 Interface Builder Guide

Developing and Building User interface of Airbnb app using navigation bar, tab bar, table view, collection view, labels, text field, button, etc.

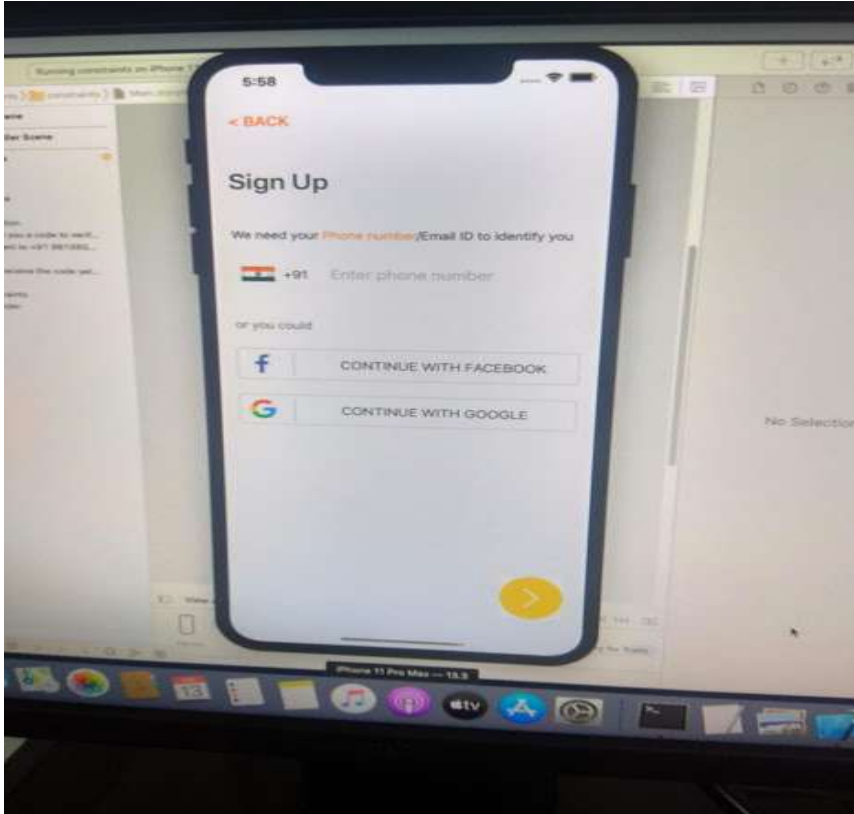


Figure 4.2.1: (Sign Up Page)

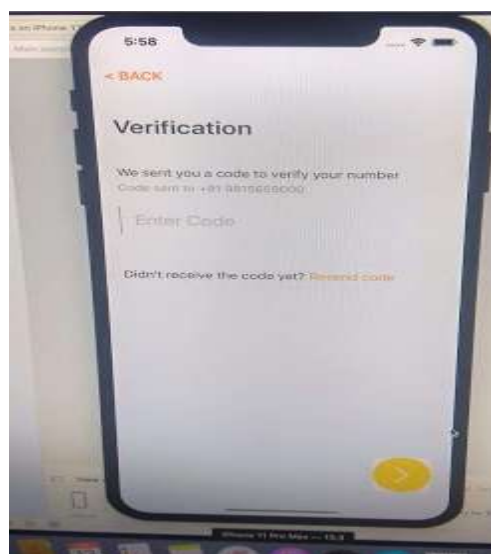


Figure 4.2.2: (Verification Page)

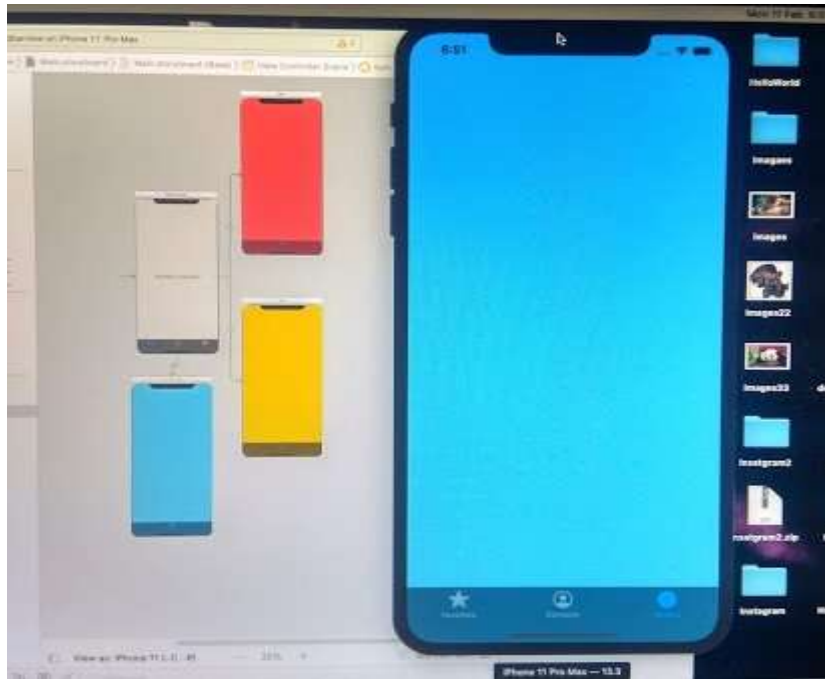
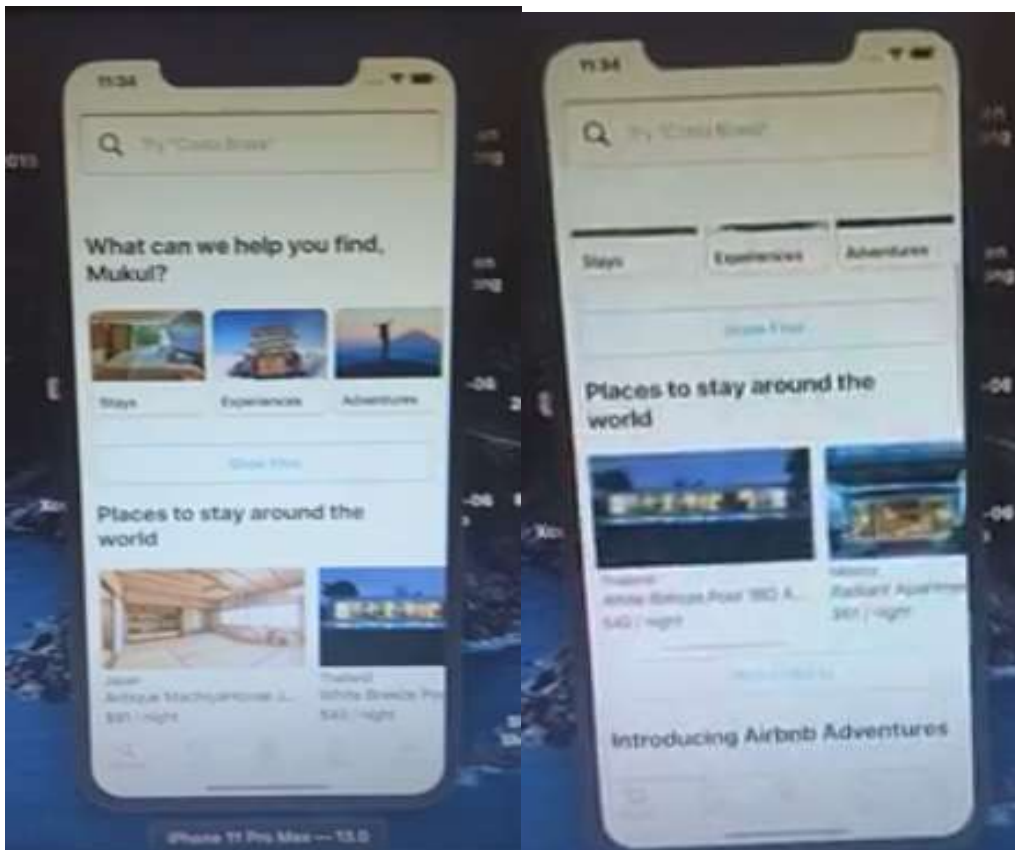


Figure 4.2.3: (Tab Bar)

App View:



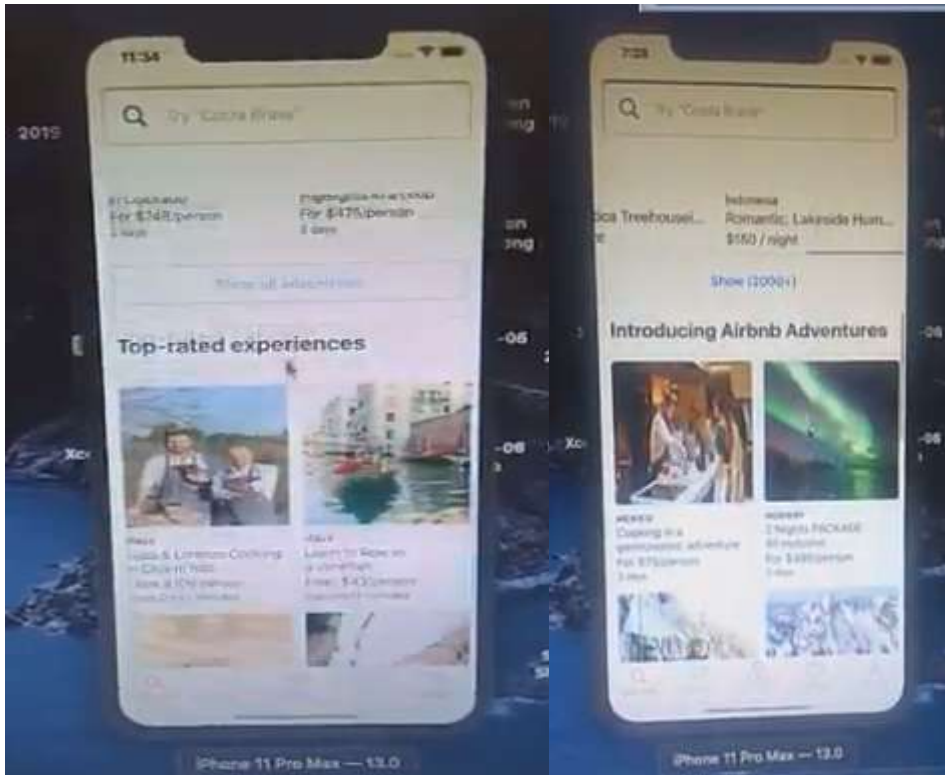


Figure 4.2.4: (App View)

4.3 JSON, Moya and Kingfisher framework

JSON:

JSON stands for JavaScript Object Notation. It is a trivial data-interchange format. It is easy for the community to write and read. Also, it is peaceful for a machine to generate and analyze. It is based on the collection of the JavaScript programming language. It is frequently used for generating, transmitting, and serializing structured data over a network connection. It communicates data between a server and web applications. In this app, we used JSON to generate data from the backend.

For Example:

```
{  
  "firstName": "ABC",  
  "lastName": "XYZ",  
  "isAlive": true,  
  "age": 32,  
  "address": {  
    "streetAddress": "PQR Street",  
    "city": "India",  
    "state": "HP",  
    "postalCode": "173001"  
  }  
}
```

Moya Framework:

Moya is a networking abstraction library shaped by the concept of encapsulating network appeals in a type-safe way. Set up an API manager. Clean code. Focus on our application's core features in its place of networking. Easy to remain network responses for unit tests. Use a popular swift networking library. Experience with Rest API. We request to the Airbnb database API to get data. It was erected by OrtaTherox and Ash Furrow. There are many structures of Moya: it checking compile-time for correct API endpoint accesses.

It perfect tradition of different endpoints with associated Enumeration ideals. It test stumps as first-class nations so unit testing is super easy.

Kingfisher Framework:

Kingfisher is a transparent and unmixed Swift implemented library for caching and downloading images from the web. The whole thing done through kingfisher goes asynchronously. It will never concern about blocking our UI thread. Multiple-layer cache. Downloaded images will be cache in both disk and memory. Boost our app dramatically. Cache management can set the maximum duration or size the cache could take. The cache will also be cleaned automatically. Kingfisher uses NSURLSession and technology of GCD. It makes a strong swift framework. It also provides easy APIs to use. Cancellable processing task. Independent components. It improves user interface performance. A category uses UI Image View for setting image from an URL directly. It requires iOS 11.0+ or Xcode 13.4.

4.4 View Controller Life Cycle

1) loadView:

This technique is called when we load our nib file and attach it to the view. This technique is achieved by the View Controller. The view controller calls this technique when our current view is nil. It takes a content view and sets this view to the view controller's view which is also known as a super view.

2) viewDidLoad:

This technique is called when our user interface view is ready to be displayed. This technique is called after load view. This technique is loaded once in view controller life cycle. This technique permits us to initialize the properties of the view and view controller's objects. We can perform certain tasks in this technique like user interface, network calls.

3) viewWillAppear:

This technique is called every time before the view is visible to and before any animations like the user tapping the back button, closing a modal dialog, when the view controller's tab is selected in a tab bar controller's, or a variety of other reasons are configured. We can override this technique to achieve custom tasks related to displaying the view such as to hide fields and disable any action before the view becomes visible.

4) viewDidAppear:

This technique is called when after the view is present on the screen. Also, called after the views are added to the super view or view hierarchy. This technique performs certain tasks like save data to core data or start animations or starting playing videos or starting playing a sound or to start collecting data from the network.

5) viewWillDisappear:

This technique is called before the view is removed from the view hierarchy. The view is still on view hierarchy but not removed yet. Any unloaded animations have not been configured. Add code to handle timers, hide the keyboards, canceling network requests, or any change in the user interface view. This technique is used to achieve cleanup tasks.

6) viewDidDisappear:

This technique is termed after the view controller's view has been detached from the view hierarchy. This technique is used to do additional activities. This technique is called when the system does not call this activity. The contact view has become hidden. Use this technique to stop listening for notifications or device sensors.

7) Deinit:

We called deinit function to clean resources memory. This technique is termed before the view controller is uninvolved from the memory. We usually dominate this technique to clean resources that the view controller has allocated.

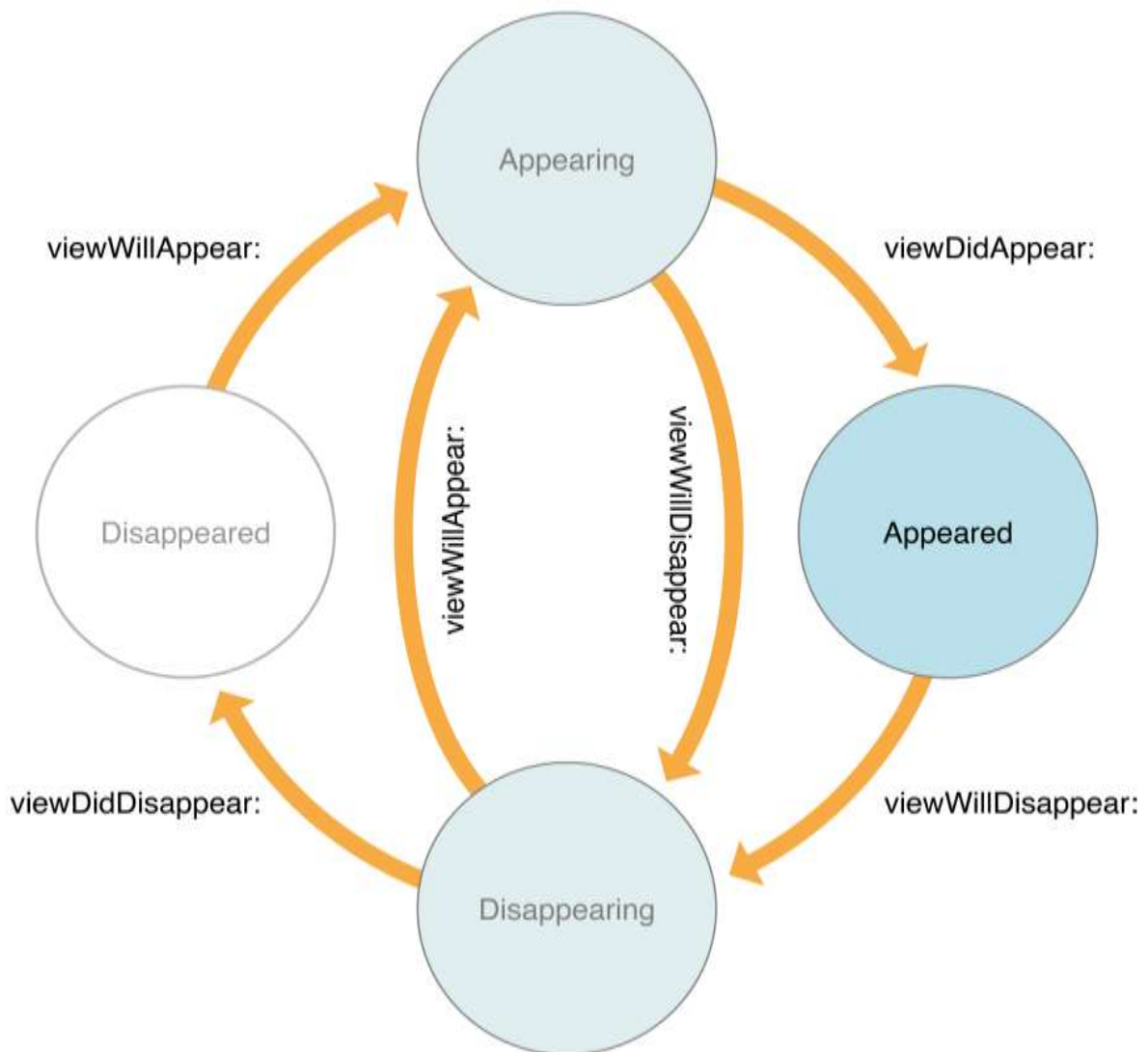


Figure 4.4.1: View Controller Life Cycle

CHAPTER-5: SWIFT PROGRAMMING

5.1 About Swift

Swift is a programming language. It was developed by Apple. It is used for the development of iOS and OS X design to replace Objective C. Its file extension is .swift. It is a fantastic way to write software. It develops software for iPhones, phones, servers, or anything else that runs code. Swift is a safe, fast, and interactive programming language that combines modern languages. It is used for iOS, Mac OS, Watch OS, and TV OS application development. Many parts of the Swift programming are familiar with C or Objective C. Swift Compiler is enhanced for presentation and the verbal is enhanced for growth.

Swift provides some versions comprising `int` for integers, `double` and `float` for fluctuating-opinion standards, `bool` for boolean standards, and `string` for word-based data. It also offers influential varieties such as `Set`, `Array`, and `Dictionary`. These three versions are known as primary collection types. Swift also familiarizes forward-thinking categories such as `tuple`. `Tuples` permit us to generate and license around alignments of value.

Swift also presents selective categories. `Option` types holder the nonappearance of a value. `Optional` says moreover there is a value or there are no values at all. Swift is a category-innocuous language. It assists us to be strong about the categories of value. We can declare constants with the `let` keyword and variable with the `var` keyword. It also provides type annotation. It declares a constant or variable to be clear around the kind of values the constant or variable can stock. It also provides category aliases. It defines a substitute name for an prevailing type. It defines type aliases with `typealiases` keyword.

5.2 Simple Value and Control Flow

Simple Value:

Simple value in swift includes constants and variables. We can announce constants with let keywords and the variable with the var keywords. We can't modify the value of the constants before declare it. We can modify the value of variable anytime or anywhere in the code.

For Example:

```
let name = "Project report" #Constants
```

```
var x = 10 #variable
```

Control Flow:

Control Flow is the order function calls, statements, and instructions in which the program's code execute. There are five control flows used in swift such as if, switch, for-in, while, and repeat-while. Use if and switch statement to make conditionals. Use for-in, while, and repeat-while to make loops. Switches sustenance any generous of data and operations. Use for-in to iterate over items. Use while to repeat a block of code until a condition changes.

If Statement:

```
Let x = 10  
If (x == 10) {  
print ("Hello")  
} else {  
print ("hi") }  
}
```

Switch Statement:

```
Swich x {  
Case 10:  
print (" hi ")  
default: return 0 }  
}
```


for-in loop:

```
varnum = [1,2,3,4,5]
```

```
fori in num {
```

```
  print (i)
```

```
}
```

While loop:

```
leti = 1
```

```
  While i< 10 {
```

```
    i *= 5
```

```
  }
```

```
  print (i)
```

Repeat-while loop:

```
var x = 2
```

```
repeat {
```

```
  x *= 2
```

```
  } while x < 100
```

```
  Print (x)
```

5.3 Function and Closure, Object and Class

Function:

Functions are independent portions of code that perform an explicit tasks. Function Statement communicates the compiler about a function's designation, arrival type, and limitations. Function definition provides the actual body of the function. Function is defined by the func keyword.

For example:

```
func sayHello () -> String {  
    return "Hello:  
        }  
    Print ( sayHello() )
```

Closures:

Closures are independent chunks of functionality that can be approved everywhere and used in our code. It is similar to lambdas in other programming language.

Syntax:

```
{ ( parameters) -> return type in  
    Statements  
}
```

Object and class:

It is the basic concept of object orientated programming. A class is a user defined blueprint and prototype for which objects are created.

Syntax:

```
class hello {  
    var hello : Int?  
    init (hello : Int){  
        self.hello = hello }  
}
```

Enumerations:

It describes a communal category for a group of associated values and allows us to work with those value in a category-safe way. Use enum keyword to create an enumerations.

Syntax:

```
enum {  
  case one, two  
  case jack  
}  
Switch self {  
  case .jack :  
    return "Jack"  
}
```

5.3 Protocol and Generic

Protocol:

A protocol defines a blueprint of technique, properties, and other requirements that perform particular task or piece of functionality. Protocol or delegate is used for back tracking. It is used for transfer data from one view controller to other view controller or second view controller to first view controller.

Syntax:

```
Protocol protocol_name {  
    #protocol definition  
}
```

Generic:

Generic code empowers us to transcribestretchy, recyclablefunctions and categories that can work with any category, theme. Generic can avoids duplication of code. It can use any data type in a single code.

Syntax:

```
funcabc<T>(a, b){  
    let temp = a  
        a = b  
        b = temp  
}
```

REFERENCES

- 1) <https://www.apple.com/in/ios/ios-13/>
- 2) <https://swift.org/documentation/#the-swift-programming-language>
- 3) <https://developer.apple.com/documentation/>
- 4) <https://www.javatpoint.com/ios-development-using-swift>
- 5) <https://www.tutorialspoint.com/ios/index.htm>

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date:14-07-2020.....

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

Name: MUKUL GUPTA Department: Electronics and Communication Enrolment No: 161067

Contact No.7018495891E-mail.anchalmukul@gmail.com

Name of the Supervisor: Dr. Shweta Pandit

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): PROJECT REPORT ON AIRBNB APP

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 = 55
- Total No. of Preliminary pages = 9
- Total No. of pages accommodate bibliography/references = 1

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at **6 (%)**. Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(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	

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