

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

Arpit Jain (161314)

UNDER THE GUIDANCE OF

Dr. Geetanjali

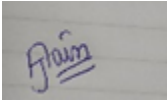


JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

May 2020

DECLARATION

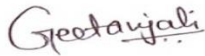
We hereby declare that the work reported in the B.Tech Project Report entitled IOS APP DEVELOPMENT submitted at **Jaypee University of Information Technology, Waknaghat, India** is an authentic record of our work carried out under the supervision of **Dr. Geetanjali**. We have not submitted this work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



Signature of the Student

Arpit Jain
161314

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



Signature of the Supervisor

Dr. Geetanjali

Date:

Head of the Department /Project Coordinator

Project Report Undertaking

I **Mr. Arpit jain**

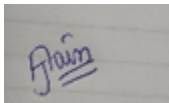
RollNo.**161314** Branch**CSE**

is doing my internship with **CODE BREW LABS** from **3rd feb** to **19th March**

As per procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.



Signature

Name Arpit Jain

Date 31-05-2020

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, Wagnaghat** 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, **MrsGurleen 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 is spent significant time in Design, Development, Branding and Online Marketing. As an entrenched driving Web Development Company in India they have aced the craft of interpreting customer thoughts and desires into top notch deliverable administrations. They have confidence in straightforwardness and organize each customer's prerequisites with a lot of duty. Additionally, their administrations are supported up with sound information and abilities of innovation, alongside savvy plans. Their worldwide customer base is assorted and runs from little new companies to huge undertakings.

Job: IOS APP DEVELOPER

iOS is Apple's portable OS that sudden spikes in demand for an iPhone, iPad, iPod Touch equipment. Apple gives devices 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 engineers is answerable for creating applications for cell phones controlled by Apple's IOS working framework.

Duties

- Design and assemble applications for the IOS stage
- Ensure the exhibition, quality, and responsiveness of uses
- Collaborate with a group to characterize, structure, and boat new highlights
- Identify and right bottlenecks and fix bugs
- Help keep up code quality, association, and automatization.

INDEX

DECLARATION	1
ACKNOWLEDGEMENT	2
ABSTRACT	3
COMPANY PROFILE	5
METHODOLOGY	5
Chapter 1 INTRODUCTION TO SWIFT	7
1.1 Features of Swift	
1.2 Swift Overview	
1.3 Swift vs Objective C	
Chapter 2 INTRODUCTION TO XCODE	11
2.1 Introduction	
2.2 General Operations	
2.3 CocoaPods	
2.4 Podfile	
Chapter 3 Application Life Cycles & Architecture	15
3.1 AppDelegate LifeCycle	
3.2 View LifeCycle	
3.3 Scene Delegate LifeCycle	
3.4 MVC Architecture	
Chapter 4 Introduction to UI Elements	22
Chapter 5 APPLICATION PROGRAMMING INTERFACE	29
Chapter 6 GRAND CENRAL DISPATCH	34
Chapter 7 Core Data and Realm	38
Chapter 8 Project	49
Conclusion	52

COMPANY PROFILE

Code Brew Labs is had practical experience in Design, Development, Branding and Online Marketing. As a settled driving **Mobile and Web Development Company** in India they have aced the craft of deciphering customer thoughts and desires into excellent deliverable administrations. They have confidence in straightforwardness and organize each customer's necessities with a lot of responsibility. In addition, their administrations are sponsored up with sound information and abilities of innovation, alongside savvy plans. Their worldwide customer base is assorted and goes from little new businesses to huge undertakings. Code Brew Labs is a firm represent considerable authority in mechanical answers for organizations - the ones that exist, and the ones that are still in their early stages. We make innovation an important resource for your business - be it a startup, a SMB or a venture - these are remembered for their growing collection. They have fiddled with Business Analytics, IoT based framework, cloud-based figuring and arrangements, alongside invasions into the universe of AR and VR.

They stretch out this aptitude to numerous industry verticals, be it Fintech, EduTech, Healthcare, Social Networking, Chat Apps, Mobile Games, Marketplaces and Food Delivery - to give some examples. Portable App Development, iPhone App improvement, Android App advancement, 2D/3D game turn of events, Mobile Strategy, Minimum Viable Product for Startups, App Monetization and Engagement, App Store Optimization, React Native are a portion of the innovations they exceed expectations in.

METHODOLOGY

Procedures for building iOS applications are methodical methods of creating programming frameworks (the how) while thinking about the various important angles (the why, what, when, who, and where) of these frameworks.

Essential Design Stages for iOS Apps:- To see how to configuration forms, you should comprehend configuration stages for iOS applications. Programming advancement 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 backers or the visionaries and moving to the improvement group and afterward to the client. There are four phases:

- **Inception:** The origin stage is the place things are being made sense of — an inexact vision; a business case; scope; a significant level, potential engineering; and elevated level evaluations of endeavors and cost.
- **Elaboration:** This is where you refine the vision, approve the center engineering, and resolve dangers. Actually, you may state that elaboration is about the goals of the dangers. This is the place a large portion of the necessities distinguishing proof is done and (with trust) reasonable undertaking the board gauges are made.
- **Construction:** The phase where you iteratively execute any outstanding highlights and plan for arrangement. The development stage is the place the undertakings are consistently including an ever increasing number of highlights. A lot of nitty gritty OO plan, 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 organization, testing, and upkeep stages occur here.

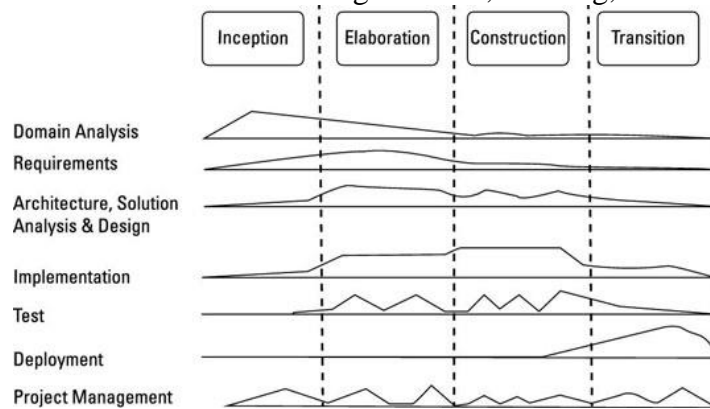


Illustration 5 :APPDESIGN

CHAPTER : 1

INTRODUCTION TO SWIFT

Quick is another programming language for iOS, macOS, watOS, and tvOS application advancement. Notwithstanding, numerous pieces of Swift will be natural from your improvement data in C and Objective-C. Swift offers its own scope of all fundamental C-Objects and C-Objects, including Int, Double and Flow object esteems, Bool for Boolean qualities, and String for content information. Quick likewise offers ground-breaking renditions of three primary assortment types, Array, Set, and Dictionary

Quick is a protected language, which implies the language causes you be clear about the sorts of qualities your code can work with. On the off chance that piece of your code requires a String, the security type keeps you from incidentally passing Int. Also, type security keeps you from inadvertently moving a String bit of code that requires a fair String. Type security causes you catch and fix blunders quicker in the advancement procedure.

Quick likewise presents a discretionary range, which treats swelling. The choices are either "there is a worth, and it's equivalent to x" or "no incentive by any means". Utilizing alternatives is equivalent to utilizing the nil file in Objective-C, yet they work for any sort, not simply classes. Not exclusively are the extraordinary strategies more secure and more exact than the nil orders in Objective-C, they are at the core of huge numbers of Swift's most remarkable highlights.

Notwithstanding the standard sorts, Swift imports propelled types not found in Objective-C, for example, tuples. Blacks enable you to make and go around value gatherings. You can utilize a tuple to restore different qualities from a capacity as a solitary incentive for correlation.

1.1 Features of Swift

Quick Supports Dynamic Libraries

Dynamic libraries are valuable code references that can be connected to the application. This component permits current Swift applications to interface rather than new Swift language types as they progress after some time. Quick's powerful libraries are stacked

straightforwardly into memory, lessen the underlying size of the application and at last increment the exhibition of the application

More secure Platform

In a serious versatile application showcase, building up a safe application ought to be a need. The plan and language structure of Swift dispenses with numerous potential sorts of blunders in Objective-C. This unbending nature implies that there will be less mishaps and instances of risky conduct. It doesn't forestall program directors from composing awful code, yet rather lessens the opportunity of committing errors. This can include 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 offers an entirely awkward code on account of its help for the preparing of content string and information. What's more, the classes are isolated into two sections; interface and usage. This lessens the quantity of documents in the task considerably, making it simpler to **manage. Automatic Reference Counting (ARC)**

Quick uses Automatic Reference Counting (ARC) to follow and deal with your application's memory use. Much of the time, this implies memory the executives "just works" in Swift, and you don't have to consider memory the board yourself. Circular segment consequently opens up the memory utilized by class examples when those cases are not, at this point required.

1.2 Swift Overview

Terminations: -

Terminations are squares of customized content that can be moved and utilized in your code. Quick terminations are the equivalent for obstructs in C and Objective-C just as in lambdas in other programming dialects. This is known as shutting over those constants and factors. Brisk download all memory the board of your catch.

Identifications:-

Posting on Swift is the primary species to their right side. Acknowledge numerous highlights that are customarily upheld just by provisos, for example, totaled properties to give extra data on the current estimation of figurings, and instances of approaches to give usefulness identified with values spoke to. they can be extended to expand their exhibition over their underlying usage; and can agree to strategies that accommodate ordinary usefulness.

Discretionary Chaining:-

Chain chainical is the way toward mentioning and calling structures, techniques, and memberships of choices that may not yet exist. On the off chance that the choice contains a worth, property, technique or call to enroll fruitful; if the discretionary is nil, property, strategy, or paid call returns. Numerous questions can be integrated, and the remainder of the cine bombs effortlessly when any steel is absent.

Conventions: -

A convention characterizes a report of techniques, structures, and different prerequisites related with a particular work or bit of work. The convention might be embraced by classification, structure, or thought so as to give continuous satisfaction of those prerequisites. Any sort that meets the prerequisites of the convention is supposed to be in consistence with that understanding.

Hereditary qualities: -

Genics is one of the most impressive highlights of Swift, and generally Swift's standard library is worked with standard code. Actually, you have been utilizing generics all through Language Guidelines, regardless of whether you don't. For instance, Swift's Array and Dictionary types are both abnormal assortments. You can make an Int-esteem list, or a String-esteem list, or a variety of any sort that can be made with Swift.

1.3 Subscriptions: -

Classes, classifications, and sections may characterize memberships, which are easy routes to getting to a gathering part's components, records or successions. You use memberships to set and return esteems for a reference without requiring distinctive arrangement and bring techniques back. For instance, you get to the components in Array structure, for example, `someArray [index]` and components in word reference structure as `some particular [key]`.

1.4 ARC: -

Swift uses Automatic Reference Counting (ARC) to follow and deal with your application's memory use. Much of the time, this implies memory the executives "just works" with Swift, and you don't need to consider memory the board yourself. Circular segment consequently liberates memory utilized by class cases when those conditions are not, at this point required.

Why swift over objective C for IOS App Development ?

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

1.5

Illustration 6:SWIFT VS OBJECTIVEC

CHAPTER 2

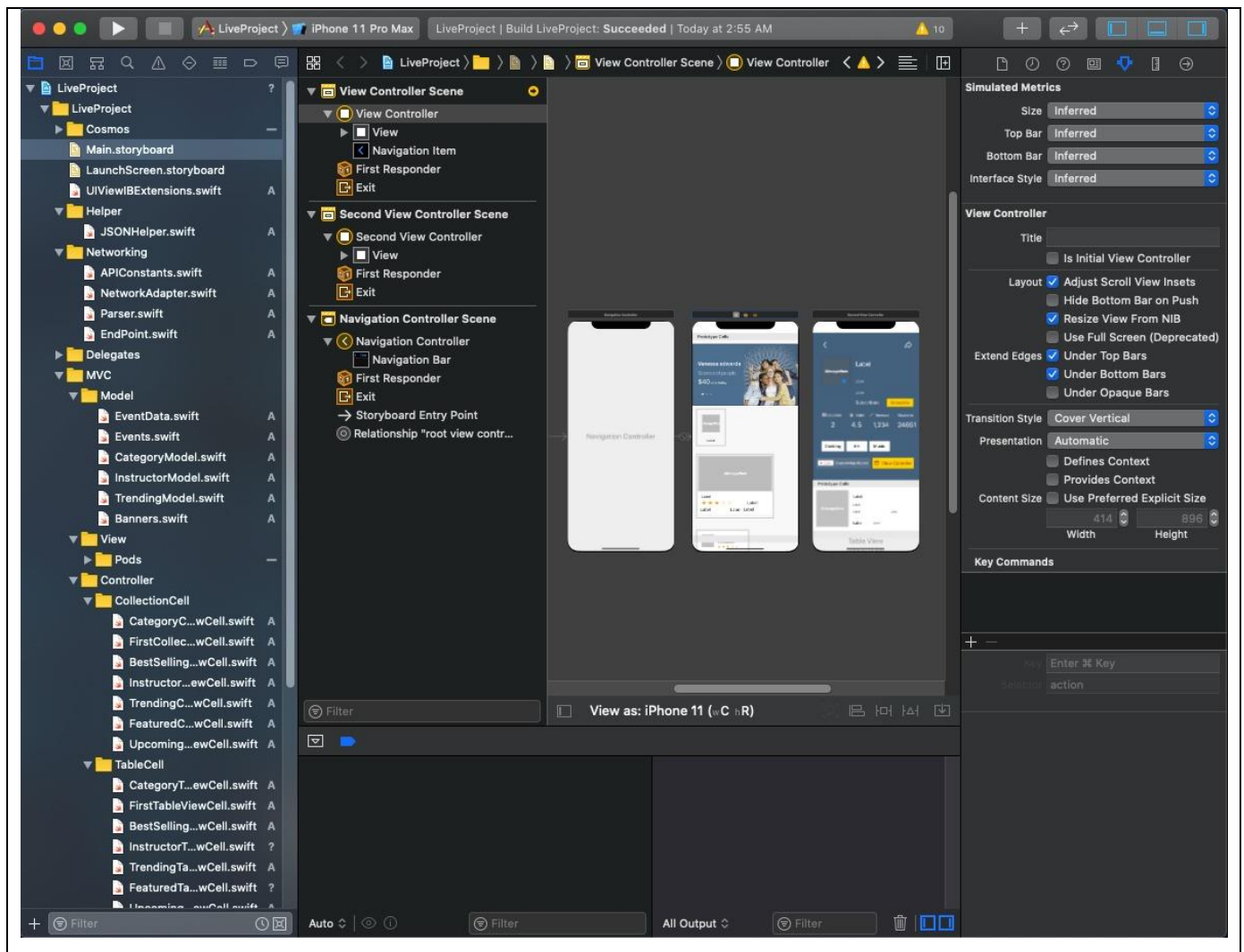
INTRODUCTION TO XCODE

2.1 Introduction

XCode is an incorporated improvement condition intended to chip away at Mac applications. It has a set-up of programming improvement devices created by apple. XCode encourages us create macOS, tvOS, iOS, and watOS programming. The most recent stable arrival of XCode is 11.3.2, which is accessible in the Mac App Store for all MacOS Mojave clients. Xcode 11 is accessible in the Mac App Store and incorporates the iOS 13 SDKs, MacOS Catalina 10.15, watchOS 6, and tvOS 13. Xcode 11 backings the advancement of gadgets running iOS 13.1. Xcode 11 backings iOS 8 gadget troubleshooting and later, tvOS 9 and later, watchOS 2 and later.

2.2 General Operations





Xcode 11 backings advancement with SwiftUI.

Xcode 11 adds support for Mac Catalyst to bring iPad applications to Mac.

You would now be able to change the presence of Xcode freely of the framework's look setting.

Xcode bolsters downloading applications from the Organizer window or order line by means of xcodebuild or xcrunaltool. Program Loader is no longer introduced with Xcode.

LaunchServices on macOS now regards the Xcode chose when we present Components, Simulator, and other Xcode designer apparatuses. For instance, on the off chance that you double tap the Instrument Track in Finder, the adaptation of Instruments in the Xcode stream is chosen. Change which Xcode is utilized with xcode - look over order line.

Itors Editors can be added to any window without requiring an Assistant Editor. Editors are

included utilizing the "Include Editor" button in the bounce bar or file> New> Editor's order. Every supervisor can autonomously show his right hand or colleague with directions in the menu Options Editor. Figure 7: XCODE

These two strategies consequently show pertinent substance when accessible. In the event that you utilize numerous editors, the display> Editor> Focus order can be utilized to briefly grow the dynamic proofreader to fill the whole window, concealing different editors. What's more, "Just Display Editor" will reset the gave editorial manager, erasing any extra perspectives empowered for that proofreader. With source control support, the Code Return button in the toolbar replaces the correlation manager. The "Show Author" order is currently accessible in the Editor proofreader menu. The SCM Log is currently situated in the Inspector.

2.3 CocoaPods: -

Cocoapods is a reliance supervisor used to launch conditions for quick and reason C extends in XCode. Like expert or frying pan, which is utilized to introduce conditions in Java. Be that as it may, trust chief is a device that oversees structures to make life simpler for engineers.

On iOS, CocoaPod incorporates outsider libraries, for example, Firebase, Alamofire, and so forth. Be that as it may, in the event that we are not utilizing Cocoods to introduce an outsider library like Firebase, we have to introduce it physically, which is an overwhelming procedure. Here, we have to introduce all the conditions of firebase and firebase, and in the event that Firebase makes changes to its SDK, at that point we have to reload it in our venture.

CocoaPods are worked with Ruby and will be introduced with the default Ruby accessible on macOS. You can utilize the Ruby Version director, anyway we suggest that you utilize standard Ruby found in macOS except if you comprehend what you're doing.

Utilize Ruby Installation will expect you to utilize sudo when entering esteems. (This is just an issue during pearl establishment.)

```
$ sudo diamond include cocoapods
```

2.4 What is a Podfile?

Podfile is an annotation that describes the price dependencies of one or more Xcode project projects. The file should simply be named Podfile. All examples in the guide are based on CocoaPods version 1.0 onwards.

How to use iPodFile?

To create a new project with CocoaPods, follow these simple steps: -

- Create a new project in Xcode as you would normally.
- Open the disease window, and \$ cd in your project information.
- Create Podfile. This can be done using the \$ pod init.
- Open your iPodfile. The first line should specify the platform and supported version.`platform: ios, '9.0'`
- To use CocoaPods you need to specify an Xcode target to connect to. So, for example when you write an iOS app, it can be the name of your app. Create a target category with the '\$ TARGET_NAME' target text and finish a few rows in the background.
- Install CocoaPod by specifying the pod '\$ PODNAME' in a single line within your target.

Question: What is Podfile.lock?

This file was created after the first pod launch, and it tracks the type of each iPod installed. For example, consider the following dependency on Podfile.

CocoaPods wi

Chapter 3

Application Life Cycles & Architecture

3.1 APPDELEGATE LIFE CYCLE:-

iOS Application Lifecycle

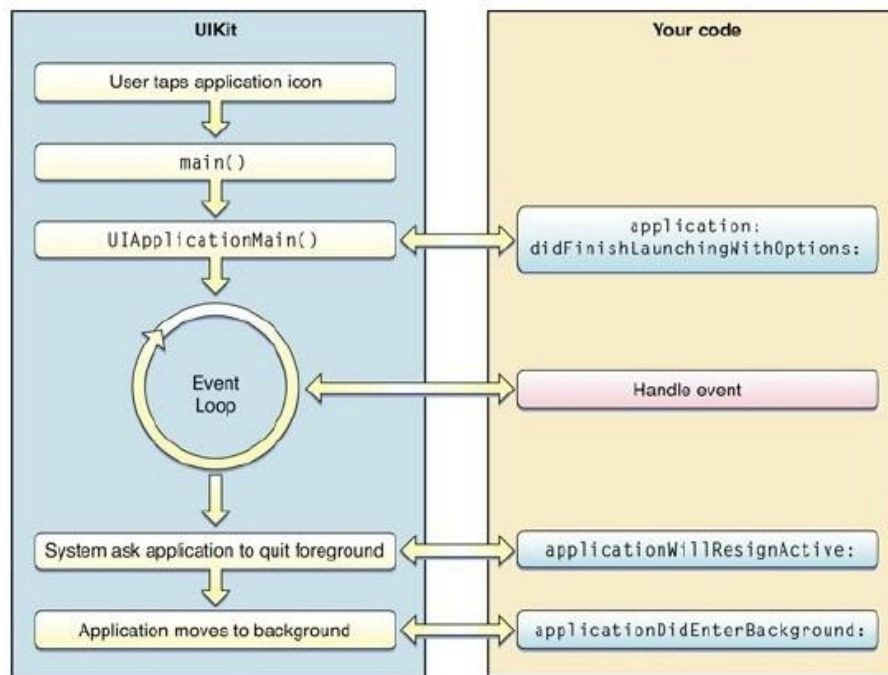


Illustration 8: APPDELEGATE LIFECYCLE

- The primary concern of section for iOS applications is **UIApplicationDelegate**. **UIApplicationDelegate** is a convention for your application that you should use to be informed of client occasions, for example, application startup, application running out of sight or previously, the application is ended, message pop-up, and so on.

- When propelling the iOS application the principal thing called the application: **willFinishLaunchingWithOptions:** - Bool. This technique is planned for the underlying arrangement of the application. Storyboards have just been transferred to this site yet reclamation has not yet occurred.
- **LAUNCH**
- **didFinishLaunchingWithOptions:** - Bool is called straightaway. This recovery strategy is considered when the application has completed the process of propelling and reestablishing the status and can play out the last execution, for example, making a UI.
- **applicationWillEnterForeground:** named after application: **didFinishLaunchingWithOptions:** or if your application is pursuing again getting a call or other program interference.
- **ApplicationDidBecomeActive:** named after the applicationWillEnterForeground: finishes the progress to the front
- **applicationWillResignActive:** is considered when the application is going to end (for instance, when the telephone gets a call or a client hits the Home catch).
- **applicationDidEnterBackground:** is considered when your application enters the space after it is latent. You have around five seconds to run whatever errands you have to interfere with things when the application is ended later or later.
- **ApplicationWillTerminate:** is considered when your application is going to be cleaned from memory. Require any last cleaning here.
- **willFinishLaunchingWithOptions:** and application: **didFinishLaunchingWithOptions:** might be created in approaches to show that the application was called to deal with message pop-up or url or something different. You have to return valid if your application can deal with the capacity or url gave.
- Knowing the strength of your application is fundamental to appropriately fire setting up your application and highlights. You don't need to utilize all the utilities: **didFinishLaunchingWithOptions**, which is typically a kitchen sink for new businesses and new companies or something to that affect.

3.2 LOOK FOR A LIFE LAMP: -

- iOS calls the UIViewController techniques as follows:
- **viewDidLoad ()** - Called when a view controller's substance see (at the highest point of its viewfinder) is made and transferred to the release board. View controller stores are ensured to have legitimate qualities at the time this technique is called. Utilize this technique to play out any extra arrangement that your view controller requires.
- **viewWillAppear ()** - Called not long before the view controller content was remembered for the application's watch list. Utilize this strategy to avoid any exercises that need to happen before the substance see is introduced on the screen. Without a name, on the grounds that the framework calls this technique, it doesn't ensure that the substance view will be noticeable.
- **viewDidAppear ()** - Named after the view controller content is remembered for the application see list. Utilize this technique to evacuate any undertakings that should be done when the view is introduced on the screen, for example, downloading information or showing liveliness.
- **ViewWillDisappear ()** - called not long before the perspective on the substance of the screen controller has been expelled from the watcher applications. Utilize this strategy to perform cleaning assignments, for example, making changes or clearing the underlying reaction position.
- **viewDidDisappear ()** - Called when the view controller's substance see was expelled from the application's viewport. Utilize this technique to play out some teardown errands. Without a name, the framework doesn't call this technique in light of the fact that the substance see is covered up or covered up. This technique is possibly called when substance sees have been expelled from the application's view bar.

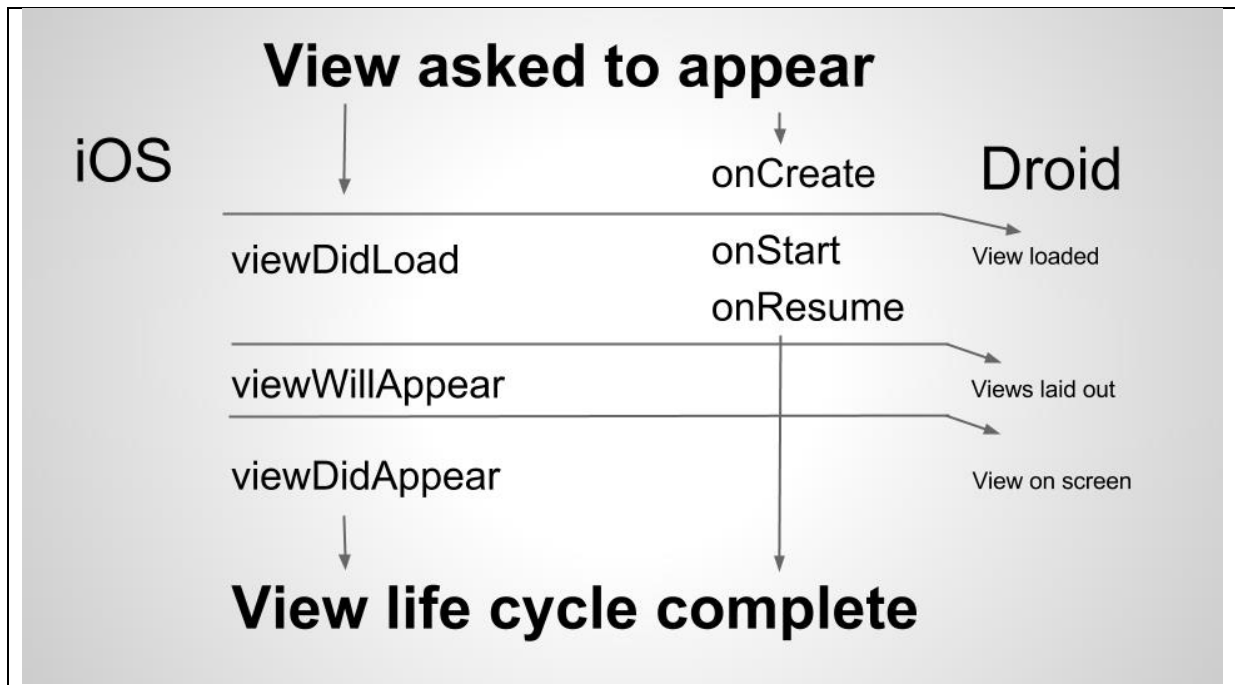


Illustration 9 :VIEW LIFECYCLE

3.3 SCENEDELEGATE LIFE CYCLE:-

The most important task of the scene team **willConnectTo** options. By the way, it's very similar to the role of the **appdidFinishLaunchingWithOptions** function in iOS 12. The function is called when a location is added to the app, so it's a good idea to prepare for that event. In the code above, we manually set the view control stack, but more about that later.

SceneDelegate also has these functions:

- **sceneDidDisconnect** : is called when the scene is separated from the app (Note that it can be reconnected later.)
- **sceneDidBecomeActive** : is called when a user first encounters an event, such as selecting it from an app switcher

- **sceneWillResignActive** : is called when a user ceases to interact with an event, for example by switching to another event
- **GroupWillEnterForeground** :is called when an event occurs before, i.e. we start or restart from the background
- **sceneDidEnterBackground** :is called when an event logs in the background, i.e. the application is limited but it is in the background

3.3 WARRANTY: -

MVC - The Model Controller Design Model is the standard design feature found in iOS applications and recommended by Apple.

MVC represents **Model-View-Controller**. It is a method of building structures or programming building design that makes making applications simpler. It isn't expected to have a particular programming language or system, yet it is an idea that you can use to make any sort of program or programming in any programming language.

For instance, in the event that you are building up an application in PHP, you can utilize systems, for example, Laravel or Codeigniter that utilization MVC design to assist you with creating applications quicker and simpler. It tends to be somewhat hard to get your head around the MVC structure at first on the off chance that you have assembled basic applications without utilizing any kind of design or system. In any case, this guide discloses how to work in MVC plan. For whatever length of time that you see how it functions, you'll simply adore working in MVC.

An application that utilizes a MVC design generally has 3 principle layers.

1. Model

Models are trademarks of your application's information. For instance, things like clients or posts will be information models. Client classification or structure may hold data connected to a particular client, for example, username, birthdate, and so forth.

The model works straightforwardly with the database. It doesn't need to manage client utilization or information handling. In certifiable mode, you will basically utilize the model to download, introduce, refresh and erase information from your database.

2. Views

Things that clients of your application can see and connect with. Review materials ought to be usable and adaptable. Things like UILabels that can show any sort of content to see.

3. Controller: -

The squares facilitate between the view and the Model. Overseers set up perspectives and information from models and update models when clients collaborate with sees.

Favorable circumstances of making MVC:

- o App advancement is quick.
- o It is simple for some designers to team up and cooperate.
- o Easy to Refresh the application.
- o It's anything but difficult to Debug as we have such huge numbers of elegantly composed measures in the app. There isn't a lot to the malicious side of development. What's more, the

negativ.

MVC architecture in iOS

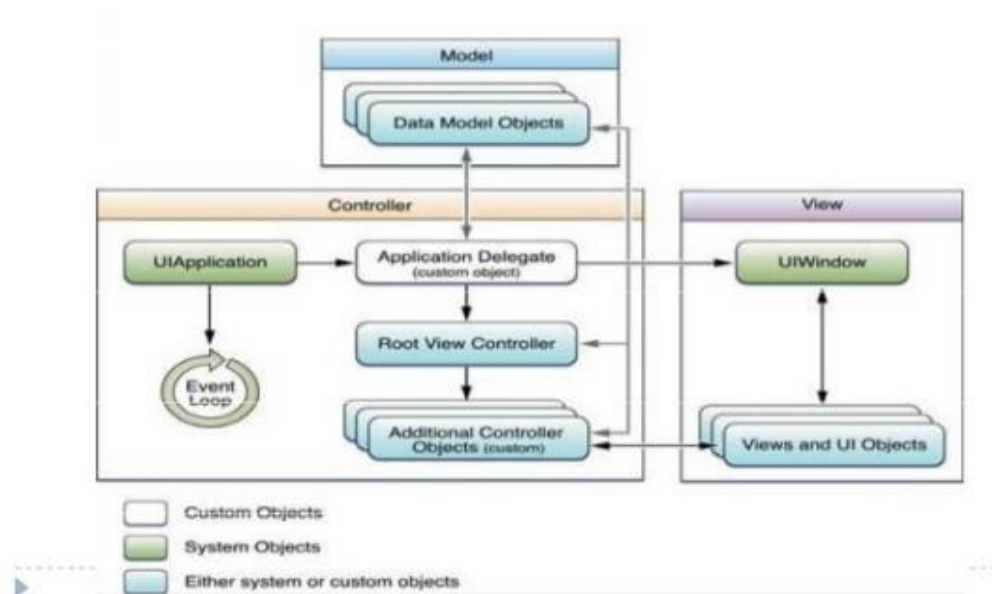


ILLUSTRATION 10: MVC ARCHITECTURE

CHAPTER :4

Introduction to UI Elements

1.1 TableView with UI Elements Used

Table view in iOS shows a segment for full print content, isolated by columns. Each column in the table contains one bit of substance for your application. For instance, the Contacts application shows the name of each contact in an alternate line, and the application's fundamental settings page shows the accessible gatherings of settings. You can organize a table to show one considerable rundown of lines, or you can isolate columns identified with classes to make exploring content simpler.

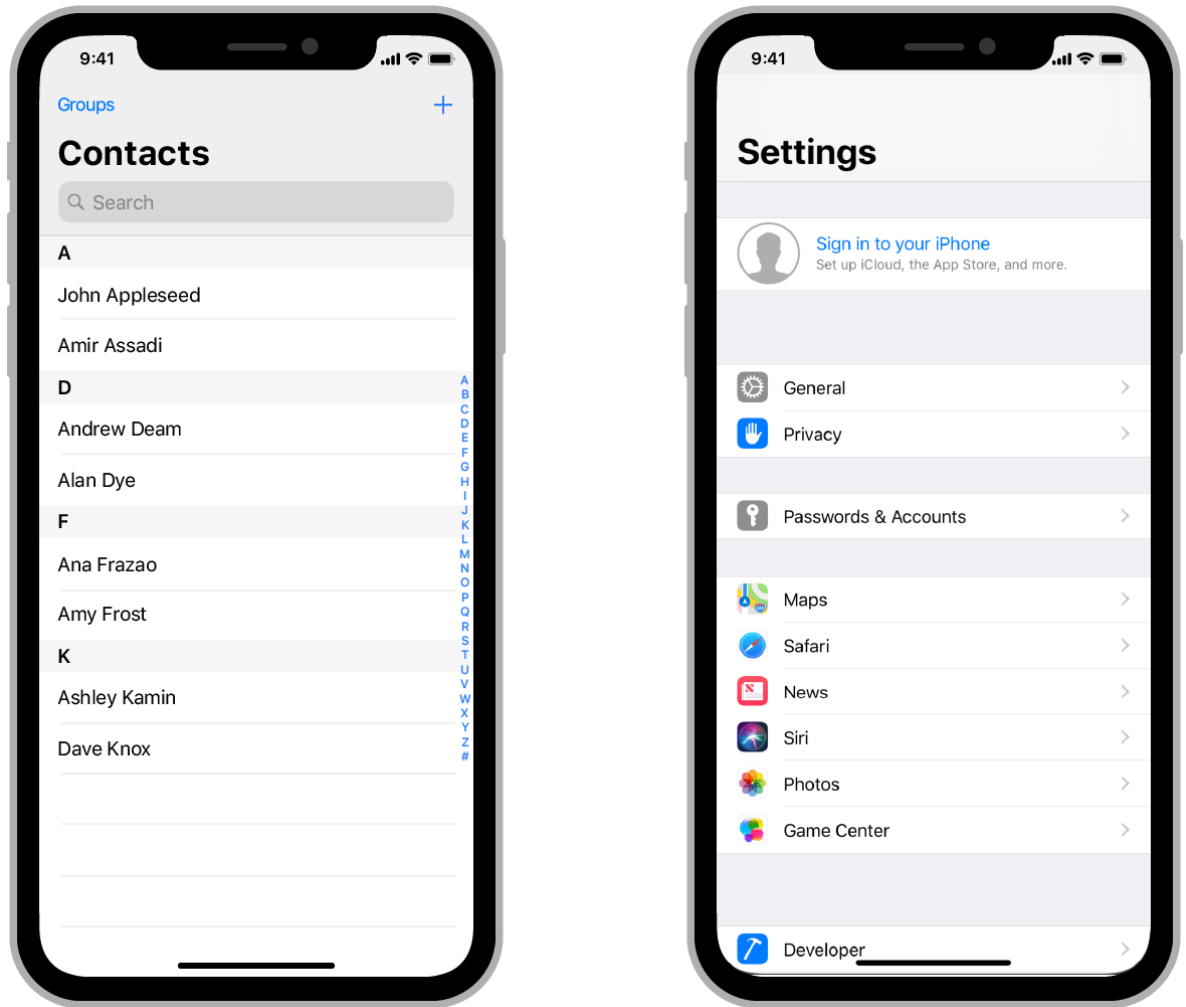


Illustration 6: Table View

The most ordinarily utilized tables are applications that have very efficient or organized information. Applications with various leveled information frequently use tables in blend with a route controller,

which explores between various degrees of the get together. For instance, the Settings application utilizes tables and route controls to alter framework settings.

UITableView controls the fundamental appearance of a table, however your application gives cells (UITableViewCell objects) that show real substance. Ordinary cell arrangement shows a basic mix of content and pictures, yet you can characterize custom cells that show whatever content you need. You can likewise give header and footer perspectives to give more subtleties to cell gatherings.

Adding Table Views to Your Contact Center

To add a table view to your model, drag the Table View Controller (UITableViewController) thing into your announcement board. Xcode makes another stage that incorporates a view controller and a table view, prepared to alter and utilize.

Table perspectives are information driven, regularly getting their information from the information source you give. The source object controls the information for your application and is answerable for building and designing table cells. On the off chance that the substance of your table don't transform, you can adjust the substance of your Storyboard document.

Giving Table Data and Cells

- **var dataSource: UITableViewDataSource?**

An item that fills in as an information hotspot for table perspectives.

- **var prefetchDataSource: UITableViewDataSourcePrefetching?**

An item that goes about as the favored information hotspot for table perspectives, accepting warnings of future cell information prerequisites.

- **convention UITableViewDataSource**

Acknowledged strategies are something you use to oversee information and give cells a table view.

- **Protocol UITableViewDataSourcePrefetching**

The convention gives the necessary information representation of the table view information, which permits you to begin chipping away at the information that might be available early.

Table View Regeneration Cells

- **func register (AnyClass ?, ofHeaderViewReuseIdentifier: String)**

The register is a classification for use in making new table headers or footer sees.

•FuncdequeueReusableHeaderFooterView (withIdentifier: String) -
>UITableViewHeaderFooterView?

Returns a functioning flash or a foot see obtained by its cursor

Overseeing Interaction with the Table

• **var load:** UITableViewDelegate?

Something that fills in as a table watcher

• The UITableViewDelegate convention

Techniques for controlling determination, setting segment headers and footers, evacuating and improving cells, and performing different activities in the table view.

Readies the presence of the Table

• var style: UITableView.Style

Table Viewing Style

. • enumUITableView.Style

Table view style properties.

• **headerView:** UIView?

The view showed over the table substance.

• **var tableFooterView:** UIView?

A view is shown beneath the table substance.

• **Back-see var:** UIView?

Area perspective on the table view.

Rundown of UI components

The particular highlights of the UI and their related usefulness are depicted beneath

UsedinTableView-

Sr.No.	UI Specific Elements
1	Text Fields ↗ It is an UI element that enables the app to get user input.
2	Input types - TextFields ↗ We can set the type of input that user can give by using the keyboard property of UITextField.
3	Buttons ↗ It is used for handling user actions.
4	Label ↗ It is used for displaying static content.
5	Toolbar ↗ It is used if we want to manipulate something based on our current view.
6	Status Bar ↗ It displays the key information of device.
7	Navigation Bar ↗ It contains the navigation buttons of a navigation controller, which is a stack of view controllers which can be pushed and popped.
8	Tab bar ↗ It is generally used to switch between various subtasks, views or models within the same view.
9	Image View ↗ It is used to display a simple image or sequence of images.
10	Scroll View ↗ It is used to display content that is more than the area of screen.
11	Table View ↗ It is used for displaying scrollable list of data in multiple rows and sections.
12	Split View ↗ It is used for displaying two panes with master pane controlling the information on detail pane.
13	Text View ↗ It is used for displaying scrollable list of text information that is optionally editable.

not

1.2 CollectionView:-

CollectionView is a part that gives a one of a kind assortment of information articles to custom structures. Presentations information in framework group.

At the point when we add a bunch view to an application interface, it's the iOS application's obligation to deal with the information related with the group see. The CollectionView object works a lot of like a table view since it recovers information from the DataSource object, which conforms to the UICollectionViewDataSource convention.

Subtleties related with the assortment see are treated as individual things that can be delegated introduced in framework structures on the iPhone screen. UICollectionView is utilized by most iOS applications, where we need information to be equitably circulated.

Assortment Views and Layout Objects

The most significant thing related with the assortment see is the format, interesting part of the UICollectionViewLayout. Auxiliary article is answerable for characterizing the association and area everything being equal and extra perspectives inside the assortment see. In spite of the fact that it characterizes their properties, the composite item doesn't have any significant bearing that data to the comparing view. Since the production of cells and extra perspectives incorporates connecting the assortment see with your information source object, the assortment see really applies the structure data to the view. Subsequently, one might say, an item is formed like another information source, giving just visual data rather than object information.

You as a rule indicate an arrangement object while making an assortment see however you can likewise change the structure of a powerful assortment. The capacity thing is put away in the collectionViewLayout property. Setting up this property straightforwardly animates the arrangement rapidly, without causing a change. In the event that you need to make changes to transforms, you should call the setCollectionViewLayout (_: pictures: end :) strategy.

On the off chance that you need to make a functioning change - activated by a touch show or contact occasions - utilize the StartInteractiveTransition (to: end :) technique to change the set item. That strategy incorporates a mid-level item whose intention is to work with your touch reference or occasion taking care of code to follow the advancement of the change. At the point when your occasion facilitating code decides the progress is finished, call the finishInteractiveTransition () or cancelInteractiveTransition () strategy to erase

1.3 Creating Cells and Supplementary Views

The information assortment source object gives all the substance of the articles and perspectives used to introduce that content. At the point when an assortment see starts to stack its substance, it asks its information source to give a perspective on each visual article.

To disentangle the way toward making your code, assortments see necessitates that you generally

decide to see, instead of unequivocally make your own code. There are two different ways of taking a gander at decent variety. What you use relies upon the kind of view mentioned:

- Use `dequeueReusableCell` (with `ReuseIdentifier`: in `:`) to recover the item cell in the bunch see.
- Use the `dequeueReusableCellSupplementaryView` (`forKind`: with `ReuseIdentifier`: for `:`) strategy to recover the extra view mentioned by the storehouse object.

Before you hit both of these ways, you should advise the compiler see how to make an intelligent view in the event that one doesn't exist. To get this, you should enlist either a class or a nib record in a bunch see. For instance, while enrolling cells, utilizing the `register(_: of CellWithIdentifier :)` or `register(_: of CellWithIdentifier :)` technique. As a feature of the enlistment procedure, you determine a reuse key that mirrors the motivation behind the view. This is a similar string you use when you split the view later.

In the wake of expelling the right perspective on your route framework, set up its substance and return it to the assortment see utilized. In the wake of acquiring basic data from the item, the assortments see utilizes it for review and showing.

For more data on utilizing information source techniques to make and plan sees, see `UICollectionViewDataSource`.

Reordering Items Interactively

Assortment sees permit you to move protests around dependent on client connection. As a rule, the request for things in an assortment see is controlled by the wellspring of your information. In the event that you bolster clients' capacity to reorder things, you can alter the touchscreen to follow client association with the assortment see thing and update the thing's area.

To begin repositioning the item's article arranges, call the `StartInteractiveMovementForItem` (at `:`) bunch see strategy. While your touch discoverer is following touch occasions, call the `updateInteractiveMovementTargetPosition` (`_`) technique to report changes to the touchpad. At the point when you're finished following the activity, call either the `endInteractiveMovement` (`:`) or the `cancelInteractiveMovement` (`:`) technique to end the connection and revive the assortment see.

During client association, the overlay see makes them less delicate to showing the present status of a thing. In the event that you don't do anything, the `height` modification methodology likewise gets things done for you, yet you can redo the design pictures on the

off chance that you need. At the point when the connection ends, it refreshes its information source and new item area.

The UICollectionViewController area gives a mechanized default show that you can use to revise protests taking into account its oversight assortment. To include this touchpad, setlayoutStandardGestureForInteractiveMovement property of the assortment see property to valid.

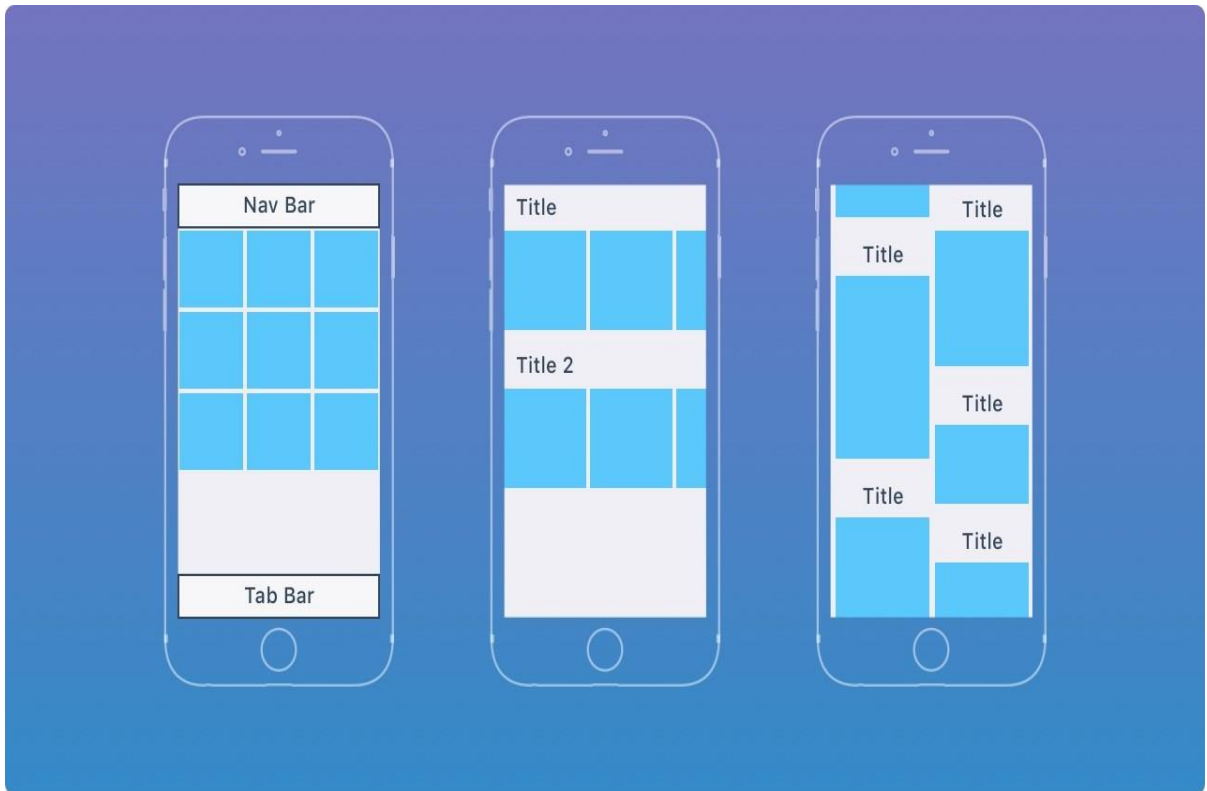


Illustration 7: CollectionView

CHAPTER 5

APPLICATION PROGRAMMING INTERFACE

2.1 Introduction

An API is an assortment of orders, capacities, conventions, and articles utilized by program clients to make programming or communicate with an outside application. It gives designers general guidelines for performing typical errands so they don't need to compose code without any preparation.

APIs are accessible for both work area and versatile applications. The Windows API, for instance, gives engineers client control and highlights, for example, windows, route bars, and discourse boxes. It likewise gives directions to getting to the record framework and performing document capacities, for example, making and erasing documents. Likewise, the Windows API incorporates correspondence orders that can be utilized to send and get information to a nearby system or Internet.

Versatile APIs, for example, the iOS API, give guidelines to discovering touchscreen inputs, for example, tapping, swiping, and pivoting. It additionally incorporates standard interface highlights, for example, a spring up console, a pursuit bar, and a toolbar, which gives route catches at the base of the screen. The iOS API additionally incorporates predefined capacities for speaking with an iOS gadget equipment, for example, a camera, amplifier, or speakers.

Practical framework APIs are incorporated into a perfect programming improvement unit. For instance, Apple's Xcode IDE permits engineers to relocate passage components into the application's interface. Provides a rundown of accessible capacities and incorporates language structure overlay of known items and orders.

While working framework APIs have a solid arrangement of highlights, different sorts of APIs are increasingly significant. For instance, a site may give a web engineer API that permits them to get to certain data from the webpage. A site API can be as basic as a lot of XML components with a couple of fundamental guidelines for recovering data.

Q.1 WHAT IS RESTAPI?

REST proposes making an information thing mentioned by the client and submits client reaction thing esteems. For instance, if a client demands a film in Bangalore at a particular time and spot, at that point you can make something on the server side.

Along these lines, here, you have a thing and you post the status of something. That is the reason REST is known as the Representational State Transfer. On the off chance that I need to characterize REST, at that point, Representational State Transfer, is the structure style and strategy for correspondence reason that is generally utilized in the improvement of different web administrations.

The REST configuration style helps in accomplishing less data transmission utilization to make the application progressively pertinent to the Internet. It is frequently seen as a 'Web language' and is altogether founded on assets.

2.2 Principles of REST API

Obviously, there are six objectives set by Drs. Handling who could clarify the advancement of the REST API in 2000. The following are six REST objectives:

Solicitations sent from the customer to the server will contain all the data expected to cause the server to comprehend the solicitations sent to the customer. This can be a piece of the URL, inquiry string parameters, body, or headers. The URL is utilized to recognize the asset and the body controls the status of the mentioning asset. At the point when the server forms the solicitation, the reaction is sent to the customer by body, status or subjects.

Server Client

Client server engineering empowers equivalent coordinated effort and separates customers from server. This improves transportability in numerous stages just as the strength of server segments.

Meeting Condition

To discover similitudes over the application, REST has four parts of the accompanying situation:

- Service recognizable proof
- Utility Benefits utilizing introductions
- Identifying messages
- Hypermedia as the status motor of the application

It is being held

So as to give better execution, applications are regularly planned not to be spared. This is finished by marking the reaction from the server

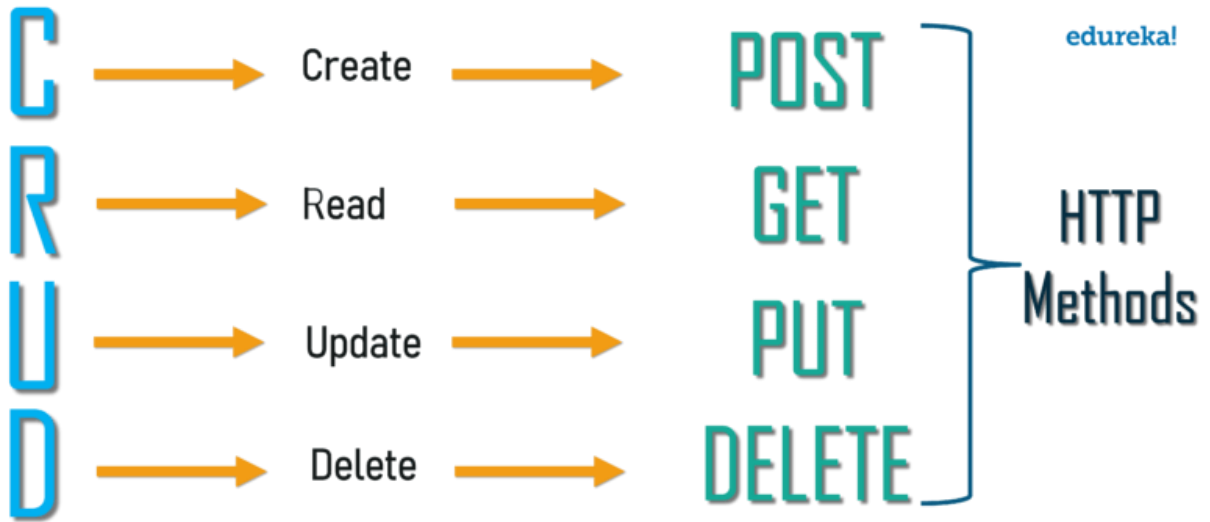


Illustration 8: RestApi Methods

- **GET** - The GET technique is utilized to remove data from a given server utilizing a given URI. While utilizing a GET demand, it should just concentrate the information and there ought to be no other information result.
- **POST** - The POST demand is utilized to make another business. It can likewise be utilized to send information to server, for instance, client information, record transfers, and so on utilizing HTML structures.
- **PUT** - Create another business or renovate a current one.
- **DELETE** - Removes every single current portrayal of the objective Source gave by the URI.

2.3 NSURLSession: -

An item that joins a gathering of system information move capacities.

The NSURLSession class and related classes give an API to downloading information from and transferring information to capacity areas demonstrated by URLs. Your application can likewise utilize this API to perform foundation downloads when your application is dormant or, on iOS, while your application is suspended. You can utilize the NSURLSessionDelegate related with the NSURLSessionTaskDelegate to help verification and get occasions, for example, redirection and employment finishing.

URL type

Errands inside a given URL meeting share a typical meeting design object, which depicts the association execution, for example, the greatest number of synchronous associations with a solitary host, regardless of whether the association can utilize the versatile system, etc.

NSURLSession has a mutual singleton meeting (with no arrangement object) for fundamental applications. It's not as customized as the occasions you plan, yet it additionally fills in as an extraordinary spot to begin in the event that you have restricted prerequisites. You enter this time by calling the common class strategy. For different kinds of meetings, you make NSURLSession with one of three sorts of setup:

- The default meeting carries on progressively like a common meeting, yet permits you to design it. You can likewise dole out a guest to the default setting for additional information.
- Ephemeral meetings are like dispensed meetings, however don't compose store, treats, or qualifications on the circle.
- Background meetings permit you to perform downloads and downloads of substance out of sight while your application is inert.

See Creating a Configuration Status object in the NSURLSessionConfiguration class for subtleties on making each kind of arrangement

- **SessionWithConfiguration:**

Makes a meeting with a predetermined meeting setup.

- **SessionWithConfiguration:**

have: Attendant: Creating a meeting with a predetermined meeting arrangement, appointment, and errand line

- NSURLSessionConfiguration

A setup object that depicts the activity and strategies of the URL meeting.

A duplicate of the current thing.

2.4 WHAT IS THE SPIRIT?

Spirit is a communication library that is inspired by the concept of secure network applications, often using signals, which give you confidence when working with your network.

How do you interact with Spirit and Alamofire?

As mentioned in the introduction to this lesson, Moya and Alamofire are strongly associated with the fact that Spirit does not make any network connection. It utilizes the power of Alamofire's battle-tested network and simply provides additional capabilities, types and concepts to advance Alamofire.

In fact, you're using Alamofire! Instead of using it directly, he uses Spirit, which uses Alamofire under the hood.

If you take a look at Podfile.lock's Starter project suggests that - Alamofire relies on Spirit:

ALAMOFIRE

Alamofire is a Swift-based communication library supported by Swifts for iOS and macOS. It provides a nice interface over the Apple's Foundation network stack that simplifies certain network communication functions.

Alamofire provides interesting request / response modes, JSON parameter and response serialization, authentication, and many other features.

In this tutorial for Alamofire, you will use Alamofire to perform basic network functions such as uploading files and requesting data from RESTful A third-party.

CHAPTER :6

GRAND CENTRAL DISPATCH

1.3 Introduction

Fantastic Central Dispatch (GCD) is a standard API for execution control. It can assist you with improving your application's responsiveness by cutting off over the top undertakings out of sight. It's a simultaneousness model that is simpler to work with than keys and strings. On the off chance that you are utilizing GCD, you don't have to deal with any numerous strings yourself. GCD innovation will deal with that, so you can concentrate on the errand itself. The GCD works with squares, which are squares of code you have to make out of sight, on the floor, in a vertical or level line by organizing either. Utilizing GCD in our application will bring greatest advantage of the considerable number of highlights referenced above without the issue of overseeing strings ourselves and we realized that GCD innovation would utilize a considerable lot of the fundamental processors found on iOS gadgets today.

GCD ideas: -

- Money synchronization: - In iOS, a procedure or application contains at least one strings. The application plan controls the strings autonomously of one another. Each string can work simultaneously, yet it is dependent upon the framework to decide when this occurs, when this occurs, and how. Single-center gadgets arrive at an agree coin in a manner called cutting time. They run one string, do a switch, and utilize another string.

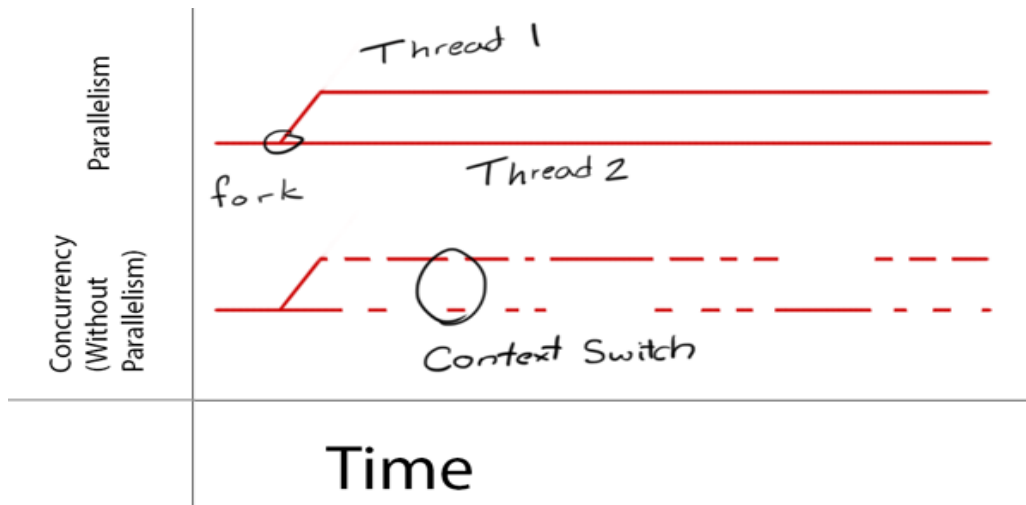


Illustration 6:Concurrency vs Parallelism

Multi-center gadgets, then again, make numerous strings simultaneously .GCD is based on the wire. Beneath the path deals with the mutual pool. With GCD you add code squares or workstations to send a line and GCD chooses which string to utilize. This at that point permits you to utilize the GCD to exploit the execution time.

Line: - As referenced prior, GCD takes a shot at a course of events that is known as a DispatchQueue class. You submit work units on the web and the GCD will do it in FIFO request (First, First Out), affirming that the principal work introduced is the first to be begun.

The Dispatch line is wire-safe which implies you can get to it from numerous strings without a moment's delay. The advantages of GCD are obvious when you see how the transmission line gives remote security in parts of your code. The way in to this is to pick the specific sort of delivery line and transportation work suitable to move your work on the web.

Quees can be sequential or successive. Dry line guarantees that just a single activity is done whenever. The GCD controls the execution time. You won't have the foggiest idea

about the measure of time between one assignment finishing and the accompanying start:

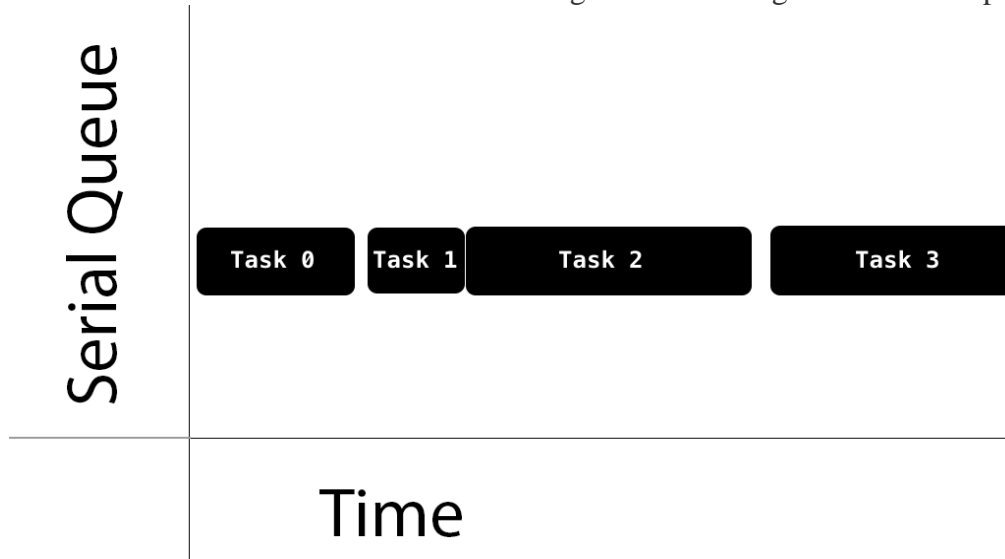


Illustration7: Serial Queue

The same line allows multiple tasks to be performed simultaneously. The line ensures that tasks start in the order you add them. Tasks can complete in any order and you have no information about when we will take the next task, or the number of tasks that will run at

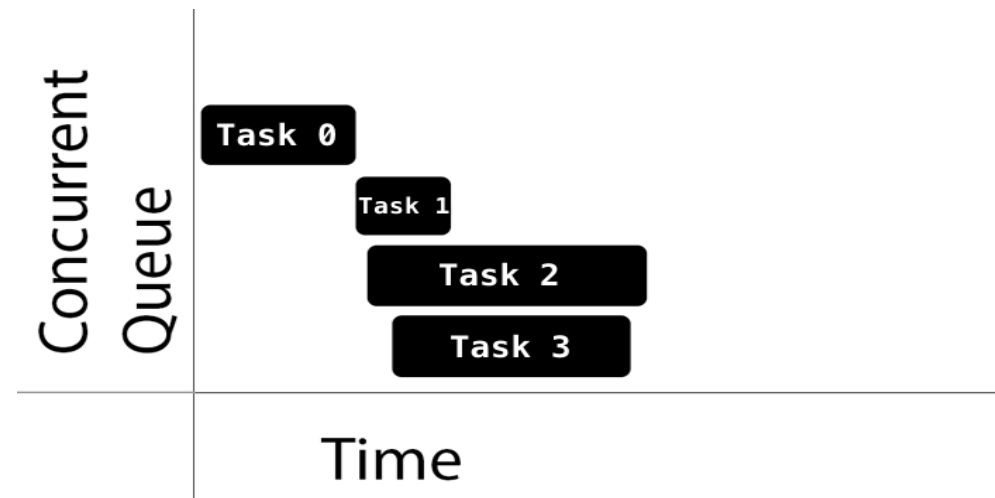


Illustration8: Concurrent Queue

Choice on when to begin work closes at GCD. At the point when one execution interim covers with another, it is dependent upon the GCD to decide if it should proceed on an alternate premise, in the event that one is found, or rather to roll out an improvement with regards to actualizing an alternate assignment.

GCD offers three kinds of lines:

1.Main Line:

- runs on the fundamental telephone and the sequential line.

2.Global Line:

- On a similar line that is imparted to the entire framework. There are four such columns with various needs: high, default, low, and foundation. A significant back line has an extremely low need and is depleted from any I/O activity to limit the effect of a flawed framework.

3.Custom line:

- In the line you make that you can make exclusively or consecutively. These online solicitations end up in one worldwide line.

When sending employments to the equivalent worldwide line, you don't indicate a particular need. Rather, you indicate the property of the administration quality class (QoS). This shows the significance of this work and aides the GCD in deciding the need that ought to be given to that task.

QoS classes are:

User Interaction: This represents assignments that must be finished rapidly to give a positive client experience. Use it for UI revive, occasion the board and negligible work that requires low dormancy. The aggregate sum of work done in this class during the term of your application ought to be negligible. This should proceed to the primary string.

User Initiation: The client starts these unfilled errands from the UI. Use when the client is sitting tight for the quick outcomes and undertakings required to proceed with the client's collaboration. They are taking a shot at the most significant worldwide chain.

Asset: This is emblematic of continuous exercises, typically with a graphical UI. Use it for estimations, I/O, organize correspondence, nonstop information takes care of and comparable capacities. This area is intended to function admirably. This will be routed to the world's driving edge line.

Background: This speaks to exercises that the client is curious about with. Use it for choice, streamlining, and different assignments that require client communication and are not tedious. This will be positioned on the planet's first most significant line.

Coordinated v/s Asynchronous

With GCD, you can present the work match up or synchronize.

The synchronization work restores the controller to the guest after finish of the activity. You can alter the work unit by synchronizing by calling `DispatchQueue.sync` (uninstall :).

The nonconcurrent task returns promptly, requesting the errand to start however not sitting tight for it to wrap up. Along these lines, the offbeat capacity doesn't keep the present string from making it to proceed with the following activity. You can program the work unit easily by calling `DispatchQueue.async` (uninstall :).

1.4 Types of Dispatch: -

There are 4 types of modes -

- 1. Broad (Quick)**
- 2. Static Dispatch**
- 3. Virtual Dispatch**
- 4. Dynamic Dispatch (Slowly)**

The compiler works to decide which routing to use, which provides an option to Inline and then down as needed.

Static Dispatch is supported by both value types and reference types. However, Dynamic Dispatch is only supported by reference types (e.g. Class). The reason for this is that, by force, or by delegation, in short, we need assets and our value models do not support inheritance. By default, Objective-C supports Power Transmission. This delivery method provides engineer flexibility in the form of Polymorphism!

CHAPTER 7

Core Data and Realm

2.1 What Is Core Data?

Center Data is a system that you use to deal with the model layer protests in your application. It gives summed up and robotized answers for normal errands related with object life cycle and article chart the executives, including tirelessness.

Center Data commonly diminishes by 50 to 70 percent the measure of code you write to help the model layer. This is essentially because of the accompanying inherent highlights that you don't need to execute, test, or upgrade:

- Change following and implicit administration of fix and re-try past essential content editing.
- Maintenance of progress spread, including keeping up the consistency of connections among objects.

- Lazy stacking of items, incompletely appeared fates (blaming), and duplicate on-compose information sharing to decrease overhead.
- Automatic approval of property estimations. Overseen objects expand the standard key-esteem coding approval techniques to guarantee that individual qualities exist in satisfactory extents, with the goal that blends of qualities bode well.
- Schema movement instruments that disentangle pattern changes and permit you to perform proficient set up mapping relocation.
- Optional incorporation with the application's controller layer to help UI synchronization.
- Grouping, sifting, and sorting out information in memory and in the UI.
- Automatic help for putting away articles in outer information stores.
- Sophisticated question accumulation. Rather than composing SQL, you can make complex inquiries by partner a NSPredicate object with a get demand.
- Version following and hopeful locking to help programmed multiwriter compromise.
- Effective coordination with the macOS and iOS instrument chains.

2.2 Creating a Managed Object Model:-

Quite a bit of Core Data's usefulness relies upon the construction you make to portray your application's substances, their properties, and the connections between them. Center Data utilizes a diagram called an oversight object model — an occasion of NSManagedObjectModel. All in all, the more extravagant the model, the better Core Data can bolster your application.

An oversight object model permits Core Data to outline records in a tenacious store to oversight objects that you use in your application. The model is an assortment of substance portrayal objects (occasions of NSEntityDescription). An element depiction portrays a substance (which you can consider as a table in a database) regarding its name, the name of the class used to speak to the element in your application, and what properties (traits and connections) it has.

Making an Entity and Its Properties

At the point when you start another venture in Xcode and open the format determination exchange, select the Use Core Data checkbox. A source record for the Core Data model is

made as a feature of the format. That source record will have the augmentation .xcdatamodeld. Select that document in the guide zone to show the Core Data model proofreader

To create an entity

Snap Add Entity.

Another untitled element shows up in the Entities list in the guide region.

- Select the new untitled substance.
- In the Entity sheet of the Data Model reviewer, enter the name of the element, and press Return.

To make qualities and connections for the element

- With the new substance chose, click the Add button (+) at the base of the proper area.

Another untitled quality or relationship (conventionally alluded to as a property) is included the Attributes or Relationships segment of the manager region.

- Select the new untitled property.

The property settings are shown in the Relationship sheet or Attribute sheet of the Data Model examiner.

- Give the property a name, and press Return.

The property or relationship data shows up in the proofreader territory.

Initializing the Core Data Stack

The Core Data stack is a collection of framework objects that are accessed as part of the initialization of Core Data and that mediate between the objects in your application and external data stores. The Core Data stack handles all of the interactions

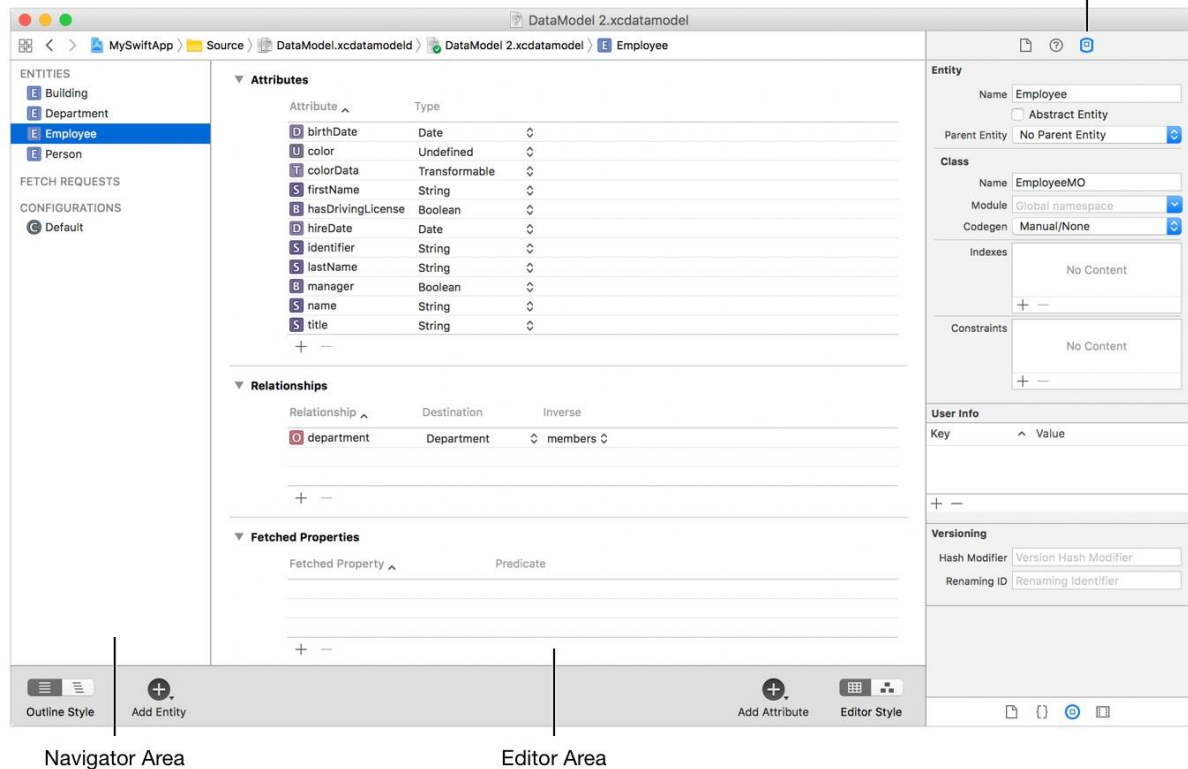


Illustration8:Employee entity in the Xcode Data Model editor

2.3 Realm

Domain is an information library (and stage) for iOS, Android and the web. It's a convincing path for SQLite and Core Data - instruments that are regularly utilized in iOS improvement that work.

Domain has three distinct items:

- Realm Database is a nearby information source association (open source)
- Realm Platform is a disconnected cloud information administration.
- Realm Studio is an administration instrument for Realm Database and Realm Platform

Domain is a database, so you can utilize it to store and recover information questions in your iOS application, in an efficient way. Subtleties are continuous during application dispatch, so it merits keeping over the long haul of the thing.

A couple of models:

- You can utilize Realm to make a store of news information, so that your application client can peruse the most recent disconnected and disconnected news.

- You can utilize it to store game information in your iOS application, and transfer information with the goal that the client can keep playing the latest relevant point of interest
- You can utilize Realm to store content notes in a database, with their properties, for example, the title of the book and when it was made

It is anything but difficult to contrast a database and a spreadsheet, for example, the one beneath. Spreadsheet lines compare to each document, called articles, and spreadsheet sections depicting the properties of those items.

Saving Data In Realm Database

Composing information in the Realm database is clear:

1. Discover the reference in the Realm model
2. Enter the things you need to spare in the review
3. Allocate exchanges to compose database

How about we talk about the standards appeared here. We have talked about information models in the past segment. The idea of the Realm has another significant component, the Realm itself.

In fact, Realm is a case of a Realm Mobile Database compartment. Truth be told, it is a .domain document that stores the files. It's anything but difficult to see Realm as a "database".

The application has a default Realm, and you can make extra Realms as you see fit. Domains can likewise adjust with the cloud-based Realm Platform. We will utilize the default, nearby Realm.

Before perusing from or keeping in touch with a database, you should discover a reference in the Realm setting. Here it is:

```
attempt {  
  
let domain = attempt Realm() }  
  
get {  
  
print(error.localizedDescription)  
  
}
```

In the above bit, the letm code = attempt Realm () code incorporates the default Realm and allots an ordinary principle. We additionally utilize the Swift blunder to get any mistakes that may happen.

- Installing a Realm can evacuate blunders when assets are squeezed, to log out. iPhone is out of memory or capacity
- During startup, just the primary (Realm) Realm will toss blunders, assuming any, on the grounds that ensuing access to that Realm has been debilitated and will consistently be fruitful
- Therefore, it is significant that you plan the application consequently you can recoup from the mistakes tossed from any unique Realm () calll, while additionally discovering blunders in your way by attempting! or then again attempt?

In the Realm, any progressions to the database must be made to the call made recorded as a hard copy. This incorporates recently made or included things, thing updates, and erasures.

The transcript of a thing in the whole library changes to the information, so it tends to be composed successively to the Realm in one go. Keeping in touch with a database is a blocking, synchronizing activity, so it functions admirably to make all the progressions

without a moment's delay. You ought to likewise intend to lessen the quantity of composing exchanges, for example Smooth as could be expected under the circumstances.

Benefits of the Realm: -

Simple support. Speedy test. Keep a long time of improvement time. Domain Mobile Database is the best option to SQLite and Core Data. Because of its zero-duplicate plan, Realm Mobile Database is a lot quicker than ORM, and is regularly quicker

than crude SQLite. Start in minutes, not hours. Domain is the best decision for your versatile information. Cutting edge reaction for versatile applications that work

Domain Mobile Platform conveys programmed and consistent continuous information synchronization with an amazing administration occasion among server and gadgets. You don't have to consider organize code once more.

Domain Mobile Database is cross-stage, supporting iOS and Android, and Realm Object Server is perfect for running on your servers or your preferred cloud. Trusted by Fortune 500 thruways, new businesses, and # 1 evaluated application accomplishments, Realm is based on applications that are utilized by a huge number of individuals consistently

.Advantages of Realm:-

Simple support. Brisk test. Keep a long time of advancement time. Domain Mobile Database is the best option to SQLite and Core Data. Because of its zero-duplicate plan, Realm Mobile Database is a lot quicker than ORM, and is regularly quicker than crude SQLite. Start in minutes, not hours. Domain is the best decision for your versatile information.

Cutting edge reaction for portable applications that work

Domain Mobile Platform conveys programmed and consistent ongoing information synchronization with a ground-breaking the board occasion among server and gadgets. You don't have to consider organize code once more.

Domain Mobile Database is cross-stage, supporting iOS and Android, and Realm Object Server is perfect for running on your servers or your preferred cloud. Trusted by Fortune 500 roadways, new companies, and # 1 evaluated application accomplishments, Realm is based on applications that are utilized by a huge number of individuals consistently

2.4 CoreData vs Realm

The task of choosing the best database solutions when developing a mobile application is much more important than it might seem at first glance. And its importance is caused, among other things, by business objectives either.

The fact is that the app operation is associated with the performance of certain data manipulations, such as saving, filtering, loading, searching, etc. And you should admit, all these processes are crucial for both the developer and the entrepreneur, as the program effectiveness depends on them.

If we're talking about creating iOS applications, then developers can use the built-in Core Data framework to perform the above data operations. However, CoreData is only one of the possible online database solutions. Today, there are a number of third-party frameworks allowing you to do the same. Among them is a very popular Realm database.

Comparison between Core Data and Realm

1.Database usage

Working with Realm is much easier than with CoreData.

The fact is that in order to use CoreData, you need a deep understanding of the API. In the case of Realm, everything is rather simpler.

CoreData manages objects explicitly in a ManagedObjectContext which you must save when making any changes. On the other hand, Realm saves all changes in the recording blocks themselves, and it does so immediately. Therefore, in case of app failure (or exiting the simulator), you can check the mobile database right away and see how it looks.

When using CoreData, stopping the emulator without first saving the context results in the loss of the current state of the database. Of course, when working with CoreData, you can save the context with each change, but this only adds to your work.

2.Creation of the scheme

Working with CoreData, you build a model in Xcode that generates the classes used to create data in objects. And when dealing with Realm, you just need to inherit from Object and mark the properties you want to save as dynamic @objc. And there are a number of functions that you can override to help set up a way to save your objects, etc. (i.e., everything is quite simple).

Though, CoreData also prevails over Realm in some ways. As you remember, the Apple framework offers a visual representation of the model in XCode. And many find the process of working with one-to-many and many-to-many relations in the Xcode model editor easier than using the LinkingObjects () function in Realm.

3. The app volume

Using CoreData has virtually no effect on our application (in this respect). However, when working with the Realm database, we must be prepared that the app volume will grow by approximately 13 megabytes (since the framework is third-party).

We've mentioned such a problem before if you recall.

4. Cross-Platform Support

CoreData isn't a cross-platform mobile database (which is so obvious: after all, we're talking about Apple's default database).

Speaking of the second system, it has cross-platform support. In addition to the iOS Realm database, there is also a Realm database for Android and OSX. And don't change the logic when working on different platforms, just use the same models in Java, Swift, and Objective-C.

5. Work speed

Realm uses its own engine, simple and fast. Thanks to its zero-copy design, Realm is much faster than ORM, and often faster than SQLite either.

6. Mobile database security

We guess you understand that security in mobile database systems is a demand of the highest importance, and it should be given special attention. Working with data is always a delicate moment, user information is often personal, so its distribution is undesirable, to put it mildly. Therefore, the mobile application database you selected must guarantee data security.

CHAPTER-8

PROJECT:DESIGNING SINGLE VIEW COURSE APP ALONG WITH GOOGLE AND FACEBOOK INTEGRATION FOR SIGN IN.

Course App:- It consists of various free and paid online courses, specializations, certificates with topics to help you advance your career, Master new skills with a wide range of Courses . As a part of our curriculum we were asked to design the app screens with Google and Facebook SDK integrations

Technology Used:-

Xcode was used to make the applications .To implement UI for the app all the basic UI Elements available in the Xcode such as UIView, UITextField, Label, ImageView were used .

To Build this application TableView and CollectionView along with Moya Various Table Cells include Collection View were used so as to implement auto layout to configure app for different screen size.

Features:-

- Refine:-Use search filter to refine your search and discover the exact course you were looking for.
- NearByInstructors:- Show all the nearby instructors available in your Region and which courses are available with them.
- View Courses:- A great feature to divide the courses into trending,featured and also displays the live courses in hand.
- Facebook ,Google Integration For Sign In.

This project played a key role in my understanding of how an App design Process takes place and how the frontend and backend integrate together in forming a visual marvel .The entire UI Design has been formulated by our team of designers and encompassed on the screen by our group of iOS developers with the ApiSupport from our NodeJS developers.

Facebook SDK for iOS:-



The Facebook SDK for iOS is the most effortless approach to incorporate your iOS application with Facebook. It empowers:

- Facebook Analytics - Understand how individuals are utilizing your item.
- Facebook Login - Authenticate individuals with their Facebook qualifications.
- Share and Send discourses - Enable sharing substance from your application to Facebook.
- App Events - Log occasions in your application.
- Graph API - Read and write to Graph API.

Step 1: Set Up Your Development Environment

In the following list, select the appropriate SDK for the language you will use in this app. If your app uses Swift, select **SDK: Cocoapods**. If your app uses Objective-C, select **SDK: FB SDK**.

Step 2: Enroll and Configure Your App with FacebookAdd Your Bundle Identifier and empower Single Sign-On for your App.

The bundle identifier (Bundle ID) should appear in the box below. If the box is empty, find your bundle identifier in your Xcode Project's iOS Application Target and paste it into the box below

Step 3: Configure Your Project

Arrange the data property list record (info.plist) with a XML piece that contains information about your application.

- Right-click info.plist, and pick Open As Source Code.
- Copy and glue the accompanying XML bit into the body of your record (<dict>...</dict>).
- Stage 4: Connect App Delegate and Scene Delegate
- Add the accompanying to your AppDelegate class. This resets the SDK when your

application dispatches, and rents the SDK handle results from the inalienable Facebook application when you play out a Login or Share action:

```
func application(
    _ application: UIApplication,
    didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
) -> Bool {

    AppDelegate.shared.application(
        application,
        didFinishLaunchingWithOptions: launchOptions
```

Illustration 10:FBSDK Integration

Google SDK for iOS:-

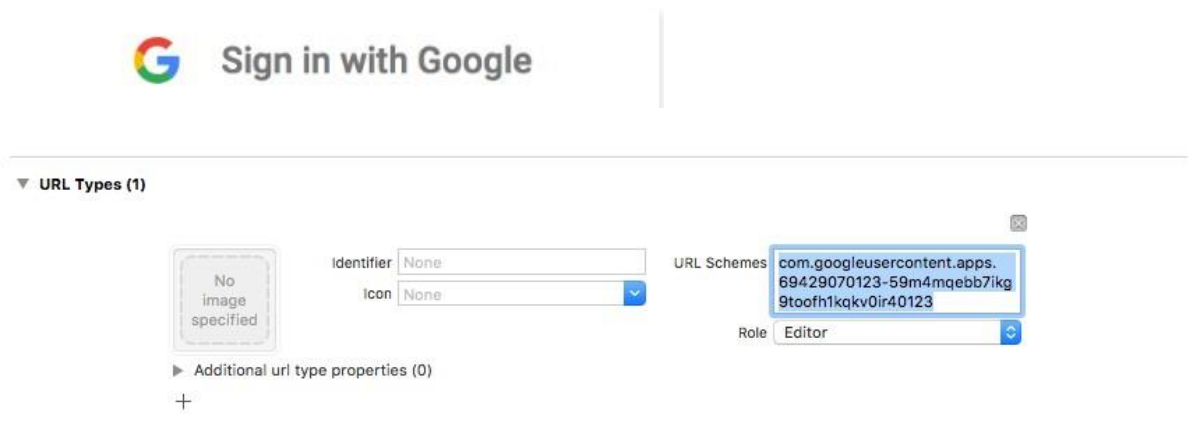


Illustration 11:Google SDK Integration

ViewControllers:-

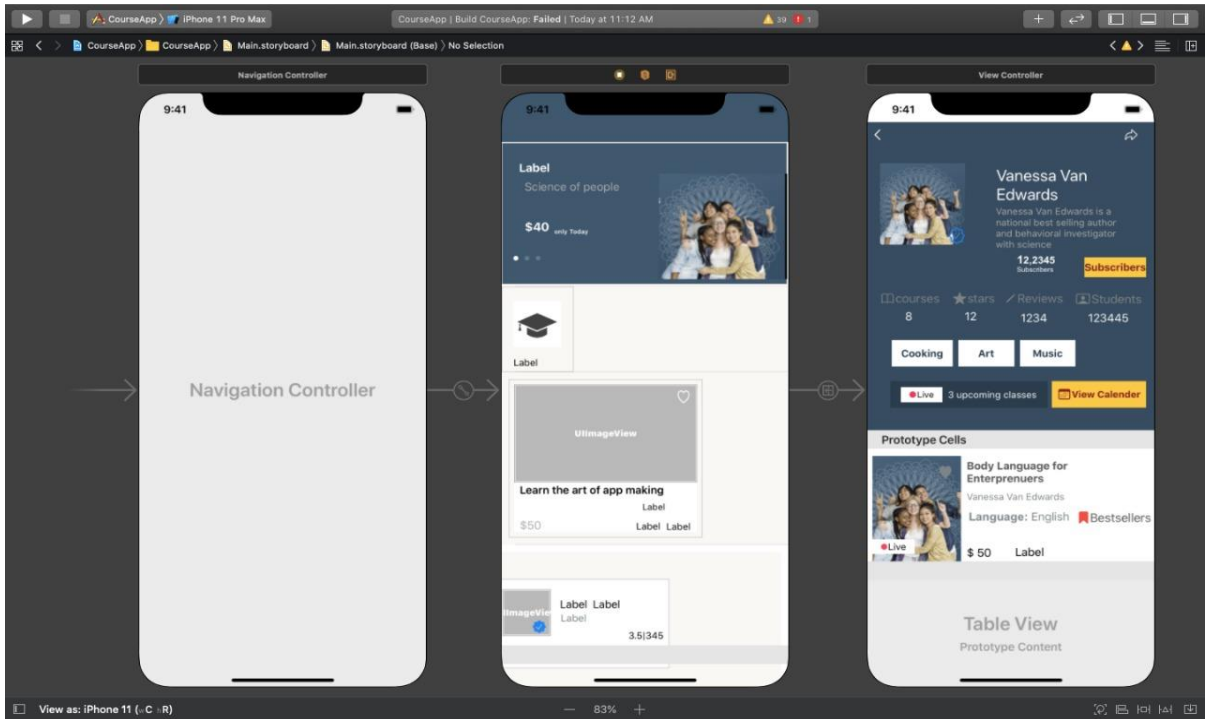
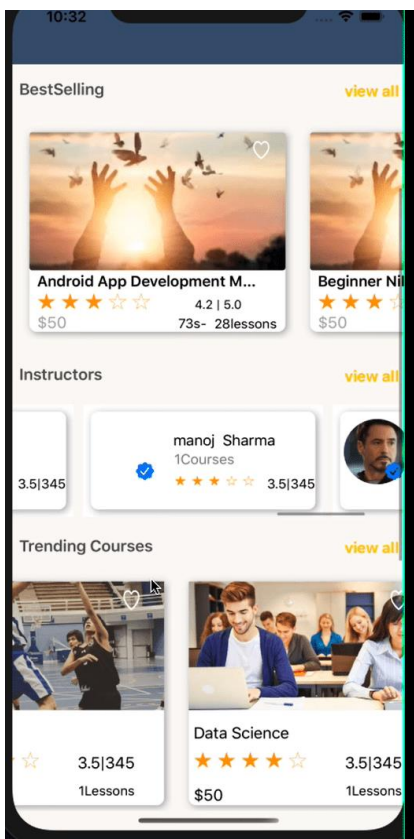


Illustration12:ViewController



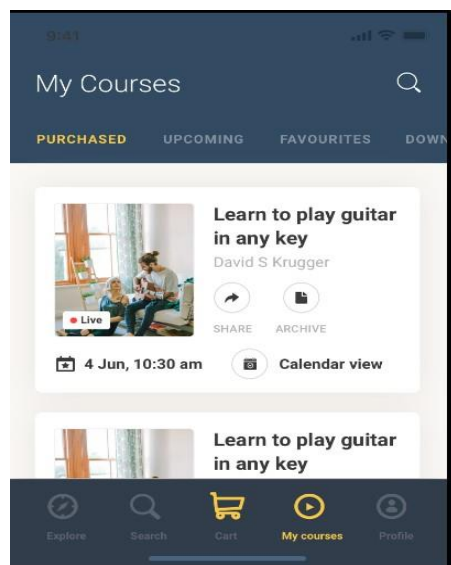


Illustration13:Views of Course app

CONCLUSION

- Had a great experience in learning about the various technologies which can be used to make our iOS apps even more dynamic and user interactive.
- Learnt how the backend is an essential support for an app and how various API models are formed and mapped in our apps
- Hands on experience to analyze/debug the issues or escalate issues to the concerned teams.
- Experience to handle industry related tools and slowly gaining expertise on integrating them to fine tune a creative app.
- Hands on experience to analyze/debug the code in Xcode and VisualStudio
- Hands-on experience to analyze/debug the issues or escalate issues to concerned teams.
- Experience of working on a Real time App Suzo which has a market base and learnt about the app's transition from initial to its deployment stage.
- Implemented the MVC Architecture for my project and designed the UI alongwith fetching data for the design using Moya.
- Hands on experience to analyze/debug the code in Xcode and VisualStudio
- Hands-on experience to analyze/debug the issues or escalate issues to concerned teams.

REFERENCES

- <https://www.code-brew.com/>
- <https://swift.org/documentation/>
- <https://developer.apple.com/xcode/>
- <https://cocoapods.org/>
- <https://medium.com/@hacknicity/how-to-switch-your-ios-app-and-scene-delegate>
- <https://developer.apple.com/documentation/objectivec>
- <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>