

Designing Development of a Radio App for iOS

BY

Sajan Das
ID: 162-15-8093

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science & Engineering.

Supervised By

Md. Tarek Habib
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised By

Dewan Mamun Raza
Lecturer
Department of Computer Science and Engineering
Daffodil International University




DAFFODIL INTERNATIONAL UNIVERSITY
DHAKA, BANGLADESH
MAY 2019

APPROVAL


This Project/internship titled “**Designing Development of a Radio App for iOS**”, submitted by Sajjan Das, ID No: 162-15-8093 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 03/05/2019.

BOARD OF EXAMINERS




Dr. Syed Akhter Hossain
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



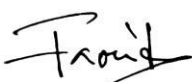
Dr. Md. Ismail Jabiullah
Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Dr. Sheak Rashed Haider Noori
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



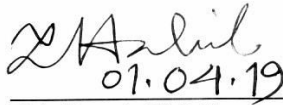
Dr. Dewan Md. Farid
Associate Professor
Department of Computer Science and Engineering
United International University

External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Md. Tarek Habib, Assistant Professor, Department of Computer Science and Engineering**, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

Supervised by:



01.04.19

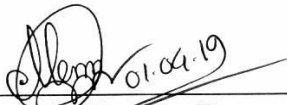
Md. Tarek Habib

Assistant Professor

Department of Computer Science and Engineering

Daffodil International University

Co-Supervised by:



01.04.19

Dewan Mamun Raza

Lecturer

Department of Computer Science and Engineering

Daffodil International University

Submitted by:



Sajan Das

ID: 162-15-8093

Department of Computer Science and Engineering

Daffodil International University

ACKNOWLEDGEMENT

First I express my heartiest thanks and gratefulness to almighty Allah for His divine blessing makes us possible to complete this project successfully.

I fell grateful to and wish our profound our indebtedness to **Md. Tarek Habib, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University, Dhaka. Deep knowledge & keen interest of our supervisor in the field of iOS Application out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior draft and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Syed Akhter Hossain, Head, Department of CSE**, for his kind help to finish my project and also to other faculty member and the staff of CSE department of Daffodil International University.

I would like to thank our entire course mate in Daffodil International University, who took part in this discuss while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

ABSTRACT

The SWIFT Radio is built only for the iPhone users. There are some uploaded songs in it. The listener can easily play all the uploaded songs consequently. Recording facilities will be available in it. Listener will be allowed to share their liked song in social media or their beloved person. There are 80 different types of radio in app store. My app is quite different from rest of the app which are available in app store. The outlook of my app will attract the user because of the graphic work. Anybody can use this app by providing their email address & password. The providing email address & password will add in the database. For further use, the user does not need to provide their new password. The previous password will do for this. This is a pilot project for iOS user.

TABLE OF CONTENT

CONTENTS	PAGE
Board of examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
CHAPTER	
CHAPTER 1: INTRODUCTION	01-02
1.1 Introduction	01
1.2 Motivation	01
1.3 Objectives	02
1.4 About My App	02
1.5 Scope of the Project	02
CHAPTER 2: BACKGROUND	03-04
2.1 Introduction	03
2.2 Related Works	03
2.3 My Project Work	03
2.4 Comparative Studies	04
2.5 Challenges	04
CHAPTER 3: REQUIREMENT SPECIFICATIONS	05-09
3.1 Business Process Model	05
3.2 Requirement Analysis	07
3.3 Functional Requirement	07
3.4 Non Functional Requirement	07
3.5 Use Case	08
3.6 Logical Data Model	08
3.7 Design Requirements	09

CHAPTER 4: DESIGN SPECIFICATION	10-21
4.1 Front -end Design	10
4.2 Back -end Design	20
4.3 Implementation Requirement	20
4.3.1 Xcode version 10	21
4.3.2 Framework	21
4.3.3 Audio Kit	21
4.3.4 RadioKit SDK	21
4.4 Interaction	21
CHAPTER 5: IMPLEMENTATION AND TESTING	22-24
5.1 Implementation of Database	22
5.2 Testing	22
5.3 Implementation of Front -end Design	22
5.4 Implementation of Xcode and Swift code	23
5.5 Testing Implementation	24
5.6 Test Results and Reports	24
CHAPTER 6: CONCLUSION AND FUTURE SCOPE	25-25
6.1 Conclusion	25
6.2 Limitations of our Application	25
6.3 Scope for further development	25
APPENDIX	26-49
REFERENCES	50-50

LIST OF FIGURES

FIGURES	PAGE NO
Figure 4.1: A Screenshot of Front-end Design of Sign up	10
Figure 4.2: A Screenshot of Home Page of Login.	11
Figure4.3 A Screenshot of Musiclist.	12
Figure 4.4: A Screenshot of Playing the music.	13
Figure 4.5: A Screenshot of Sharing System.	14
Figure 4.6: A Screenshot of Message for user.	15
Figure 4.7: A Screenshot of Features.	16
Figure 4.8: A Screenshot of Logout	17
Figure 4.9: A Screenshot of Home Page	18

LIST OF TABLE

TABLE

Table on 5.6: Test Case for Project on My Application	23
---	----

CHAPTER 1

INTRODUCTION

1.1 Introduction

Audio broadcasting and radio stations have been around for about a hundred years. However, data suggest that the majority of listeners are no longer dialing in the station frequency on their AM/FM radio. Today, the majority of people reach for their mobile devices to listen to music or news broadcasts. SWIFT is a personal radio station for the all communities. Based in SWIFT 4. SWIFT strongly focuses on bringing all types of music a unique Bangladeshi art form to everybody. Many stations have their own mobile applications. These kinds of apps not only benefit the listeners, but also benefit the stations themselves. With mobile apps listeners can check the radio stations' schedules and programs with simple swipes and taps. More listeners mean greater revenue for every radio station. As part of this iOS project, I have developed a prototype of SWIFT Radio app for iOS. An effective resource for SWIFT Radio listeners by surveying musicians and obtaining feedback about their experiences using the prototype. This document described all the process of my development.

1.2 Motivation

There is no built in radio app in iPhone. Most of my friends are using android device however I use iOS. They often turn on their radio but there was no built in radio in my device. It will inspire to develop a radio for the iOS user. This a pilot simple project on Swift Radio based on SWIFT 4. I have some vision regarding this app. In near future I want to develop it in a vast way. If this project become successful I will try to add this app in apps store.

1.3 Objectives

As we all already know that Increase acquaintance of the power and expectation of radio. Cheer up and inspire a positive use of radio.

1.4 About my App

This is basically a Radio app only for iOS user. This radio application basically for the selected community. This “Swift Radio” Application very easy to access. I hope this app will be very user friendly. Everyone can use the app in their own iPhone device. I will try my level best to develop this and I hope application will be favorable for all iPhone user. This application is easy to use and that’s the main point.

1.5 Scope of the Project

Generally this app based on iPhone users. So I figure out user will be glad for this because of better communication among all friend. Here you can access the personal radio system and music list. You can search any kind of song. Make a playlist with friends can be shared to all friend which have iPhone device.

CHAPTER 2

BACKGROUND

I mean the formal documentation by “Project Background”. In this part I personally expect to study about the related works done by others and also analysis their work to find new scopes.

2.1 Introduction

I have put a background all image which will remain the same resolution in all the iPhone device. The image will be design for iPhone 5s later to latest version of iPhone XS Max. I have used graphical animation for its better mend. I have to go through with some specific course of action to achieve this goal. First of all I need to define all requirements to develop the new system.

2.2 Related Work

When I determine to build this “Swift Radio” app as my final year project, I have seen and through Apple App-store and found no such app that is like my app, but found some similar apps. However, Which I found in apps store was not highly develop, that is the main different concept of this app. Here I listed some high index app that we have found in Apple App store

1. SiriusXM Radio.
2. RADIO.COM
3. iHeartRadio.
4. myTuner Radio.

I tried my best to develop something which really look like different. I try to bring the grain concept in here. This basic concept is developing all kinds of users for understandings. There is Firebase database in my app.

2.3 My Project Work

I demanded to build an iOS application related to develop Radio. With some extra significant features, of course which will hopeful to the user. I want my application as a perfect package. In my project I have tried to focus on simplified to use. In my project basically it is an iOS application that is why I use Xcode version 10 and Swift programming language Edition 4.1 and 4.2. In my project there are online based firebase database part. In my project I use American English font.

This application is not only for Bangladesh but also use to hold world. There may have some lacking in my app. I strongly try my level best to develop this as better as possible.

2.4 Comparative Studies

As I displayed before there are some applications are common in the Apple App store which are not completely developed. Some applications are fully fixed and some has only front end design. I will give facility accurately to the user interface to connect with friends. However I placed such step to go ahead so that I can be a part of entertainment. These iOS applications have not limitations. User can connect to Wi-Fi and mobile data connect to same time with others friends. They can share their Choice full music and records. They will not be able to share this app one to others device. My application is just only for iPhone device not for others. I hope that the user interface is user friendly enough so the users will be entertain using it. I hope they will enjoy using it because it is easy to handle.

2.5 Challenges

When any developer wants to make anything different, then the developer has to face some different types of challenges, contest, competition and obstacles. No work can be completely smooth. As like this situation, to develop my project I have to face some challenges here. If I face any types of problem like scrolling, button then I am trying to sort out the problem very fast. Since there are many applications in this platform so it will be very difficult to attract user towards me. Thus designing user friendly interface, using Firebase database and performing everything in a sequence is quite difficult. In future I have to put some new content that will make my app more acceptable to user. All those things might be challenging for me.

CHAPTER 3

REQUIREMENT SPECIFICATIONS

In requirements specification section I try to create those documents those capture the complete description about how my system will perform. What to do with my app. REQUIREMENT SPECIFICATIONS is most important part in project.

3.1 Business Process Model

Business process modeling software allow you to represent your process in a digital way that can be transferred to a live automated process. The term ‘business model’ is thus used in very familiar way and for a broad range of informal descriptions to represent core aspects of including purpose, structures, operational process and policies. The main characteristics of the methodology is based on diagram as ‘Flow diagram’. Here I am trying to describe my projects business model using Data Flow Diagram. Normally, Data Flow Diagram describes form Google and how data flow is processed though a system or project. A Data Flow Diagram shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structure flowchart which focused on control flow, it’s like as diagram. Which stay in all application. Because this is the main structure for an application and for understanding.

By business model a user can realize how can it works and that’s the main point. Where it uses and where its use-case. A developer always tries to invent his/her business model by his/her own way. Because every application is unique that’s means dissimilar with each other.

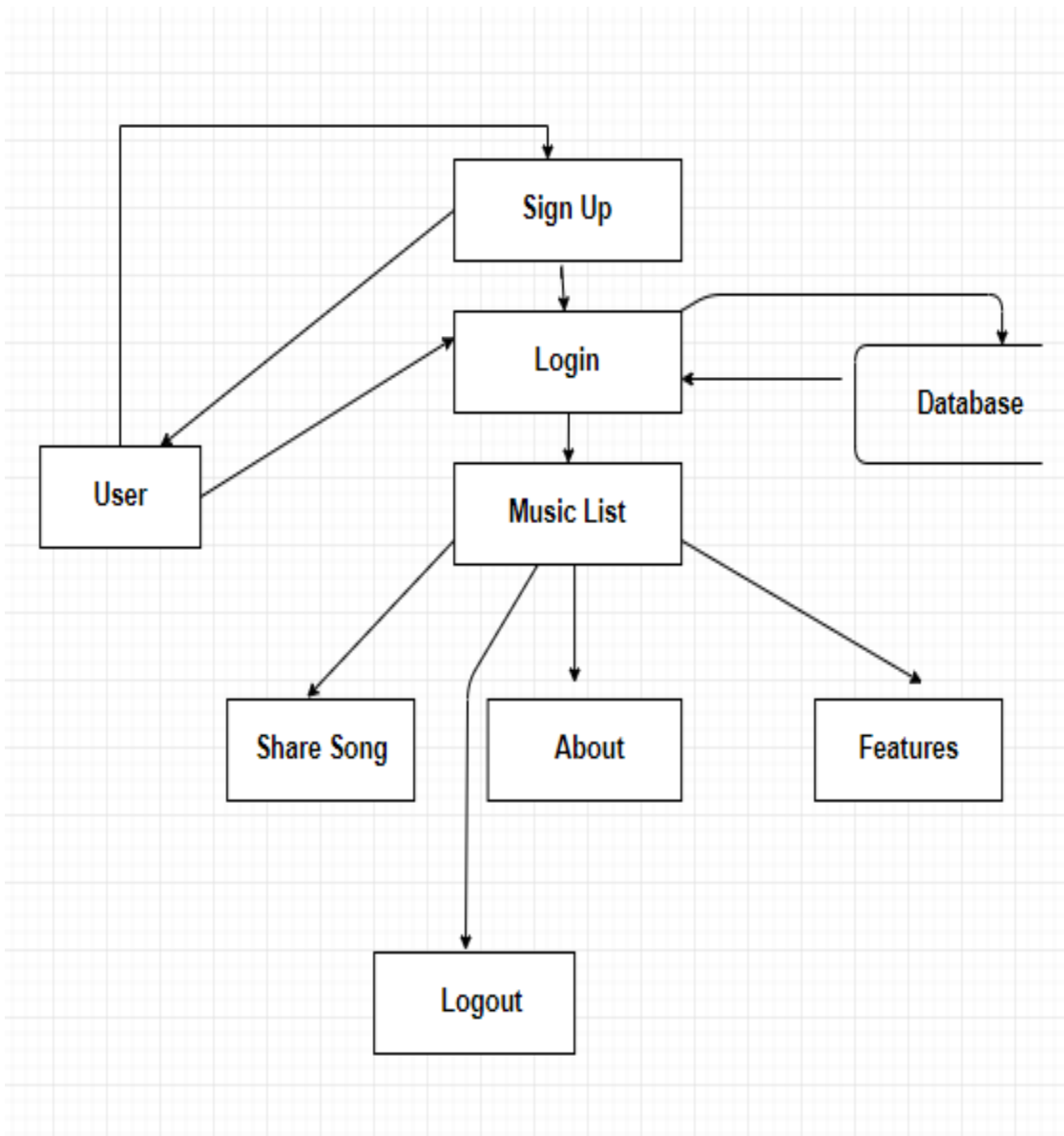


Figure 3.1 shows the Data Flow Diagram of My Swift Radio.

User submit to Email ID, Password in sign up page different time. Every time list save in database. User can see that all type music list which user input in database. Here user can include all types' songs from iTunes App.

3.2 Requirement Analysis

Requirement analysis is also called as requirement engineering which is the process of determining user expectation for a new system. These features, called requirements, must be quantifiable, relevant and detailed. Requirements analysis involves frequent communication with system users to determine specific feature expectations. Requirement collection and analysis is one of the primary conditions of application development. There are two types of requirements, one is the functional requirement and the other is non-functional requirement for deployment. Functional requirements are those activities that the application software can perform. On the other hand, Non-functional requirements define the personality of an application, as like the application is how much efficient, performance issue of the application and many more.

3.3 Functional Requirement

Functional requirement is those which are related to the technical functional of the system as like application software can perform. From the point of view of our system, it should have many functional requirements like maintaining database, add music list, share to others device and share the songs and press the share button where user can find out nearby friends, if you will connect to internet then you will send to friends at anywhere. By user can add songs.

3.4 Non-functional Requirement

Non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system in particular conditions, rather than specific behaviors as like the application is how much efficient, user friendly, performance issue of the application etc. From the point of view of my app, the application has many functional requirements as more efficient, relevant, optimize performance, memory consumer and smoother operation, loading on quickly. Applications user interface is also so user friendly and gorgeous for excellent user experience. Non-functional many build in function also use in here. This application can be used only on iOS user. Only user cannot modify the information of the application. An iOS based device with internet connection for share.

3.5 Use Case

Use case is a description of how a user will use the system to accomplish his business goals. It represents a functional or an action within the system. The two main components of a use case diagram are use cases and actors. Actor in a use case diagram is any entity that performs a role in one given system. Use-case always use for user understanding and I try my level best for efficiency.

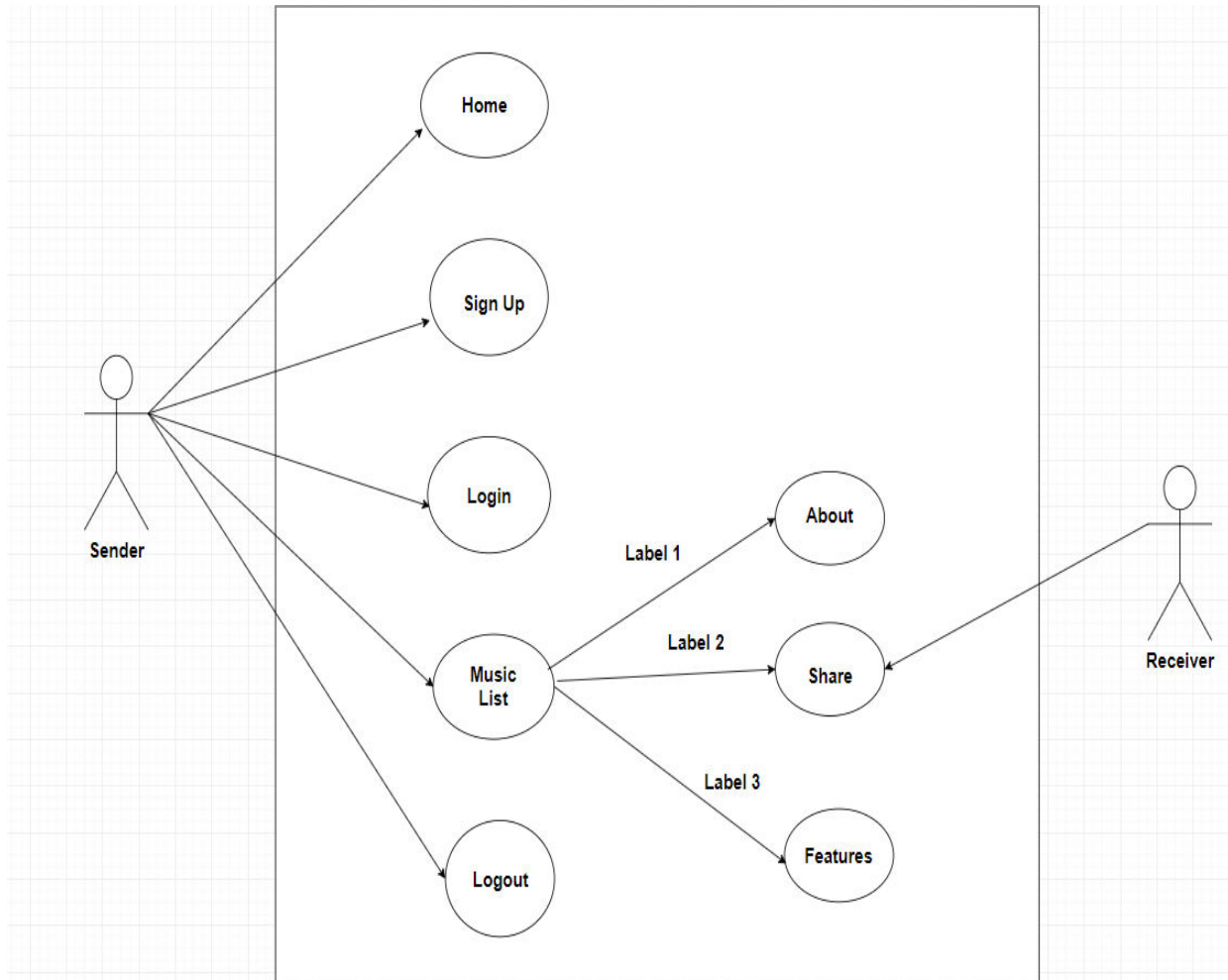


Figure 3.2: Use-case modeling of My Swift Radio

Here in figure 3.2 I show how to use this application and first homepage and then all. Option is the main part and all button include in there. If any user tries to enter in this, then press any button which he/she like most. Use case main show how to use the project or something. This use case indicates the same thing in here. In here everything is including, a user wants like this [4].

3.6 Logical Data Model

A logical information clarify is an information model of a particular issue space communicated freely of a specific Firebase database item or capacity innovation yet as far as information structures, for example, music list. I used cocoa and cocoa framework. In my application I have to use Firebase database system. My application in used user details purpose application so that's why there is database part.

3.7 Design Requirements

The design requirements for my project will dissent me from the other projects, because my project will working on my specific problem statement and the product, system, or experience that is designing. My designed my project for some specific segment that is providing nearby hospital and medicine shop and search on specific place and set alarm on specific time. I try to provide the all things to my application that are necessary for a user to operate my application easily.

Here some requirement:

1. In my application, I have use authentication for my user because of this app is a private app. So I have design registration, login section.
2. I haven't design profile in application and just display Email and password.
3. In my application I should provide a simple user interface for simplify the uses in that case I can use navigation view controller which provide many operation in one simple place.
4. I design whole description English font and all important code design in the app.
5. I can also link up my application with Google, Facebook, and twitter built in to connect social media.
6. Everyone who knows Bengali or English can use this application.
9. The interface of the application is suitable even for the blind people because user can speak with Siri.
10. An iOS based device with internet connection.

I am trying to build my app as a complex free. I think about user when I design my application. Because of this reason, my iPhone application is so user friendly.

CHAPTER 4

DESIGN SPECIFICATION

Design Specification is a statement of how a design is developed. In the section of Design Specification I try to show the design of my iPhone based app. I also discussed which I use to develop my application.

4.1 Front-end Design

The front-end design is everything involved with including design and some languages, what the user sees he/she open the app. The most important part of a project is front-end designing to keep the users interested in the application. Generally most of the users expect a simple user interface or graphical user interface from the creator. The application fails to attract of the user, if the front-end design is so intricate. The front-end design of my project is as follows

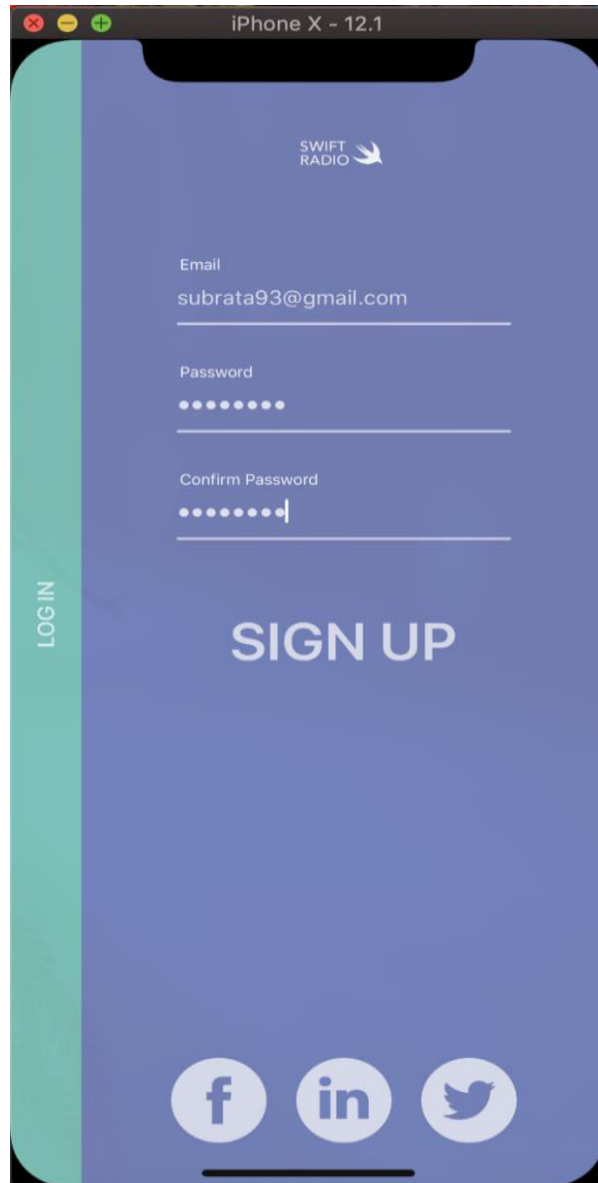


Figure 4.1: A Screenshot of Front-end Design of Sign up

Figure 4.1 shows the front page of the user interface. Users will see it every time when they open the application. I tried to design my application front-end so simple.

Hope the users will find it easily accessible and get benefited from it

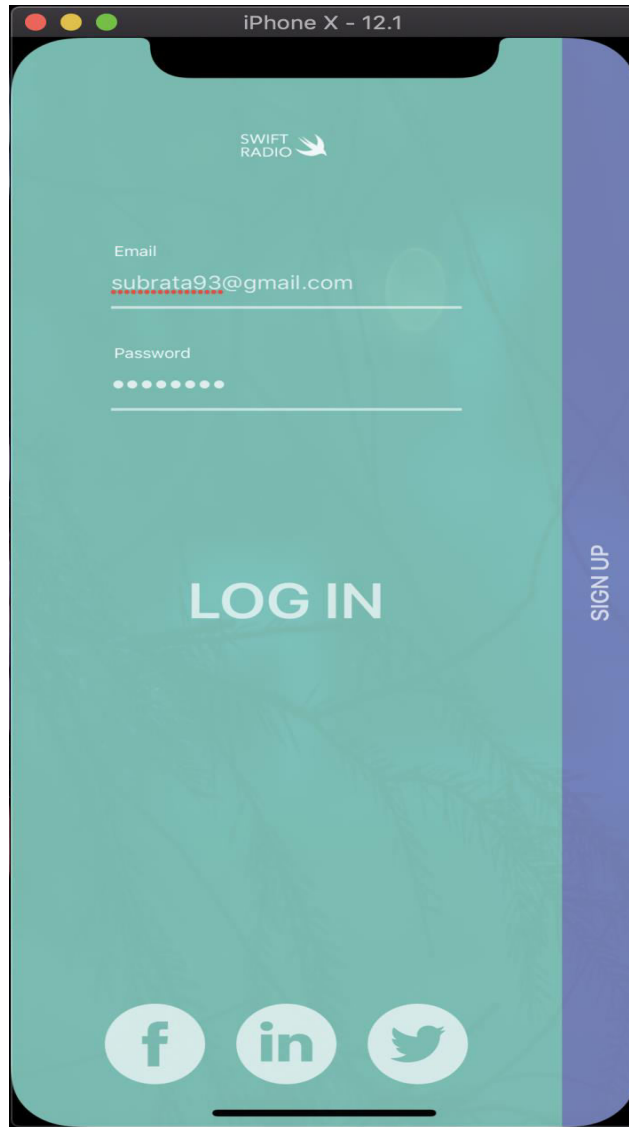


Figure 4.2: A Screenshot of Home Page of Login.

Figure 4.2 show us, this is the homepage user like most. Press sign up button and enter this. In home page only sign up button, also used Navigation view controller because the connection between the sign up and login button. User can click login button and open the music list



Figure 4.3: A Screenshot of Music list.

I have to include for demo five songs in the music list. User can include song from iTunes app. In future I will add different category and level of songs in it.

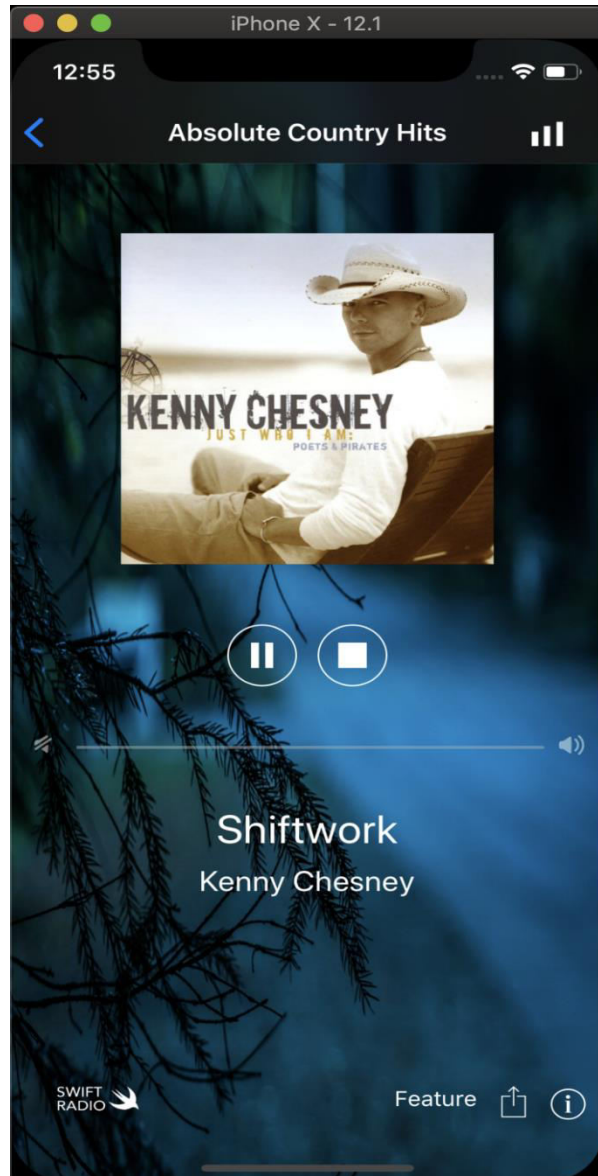


Figure 4.4: A Screenshot of Playing the music.

Here user can select the specific song and press on button then playing the song. If user press pause button by this time that song will pause again if the user press the stop button the song will become stop. Again the user press the play button the song will be playing.

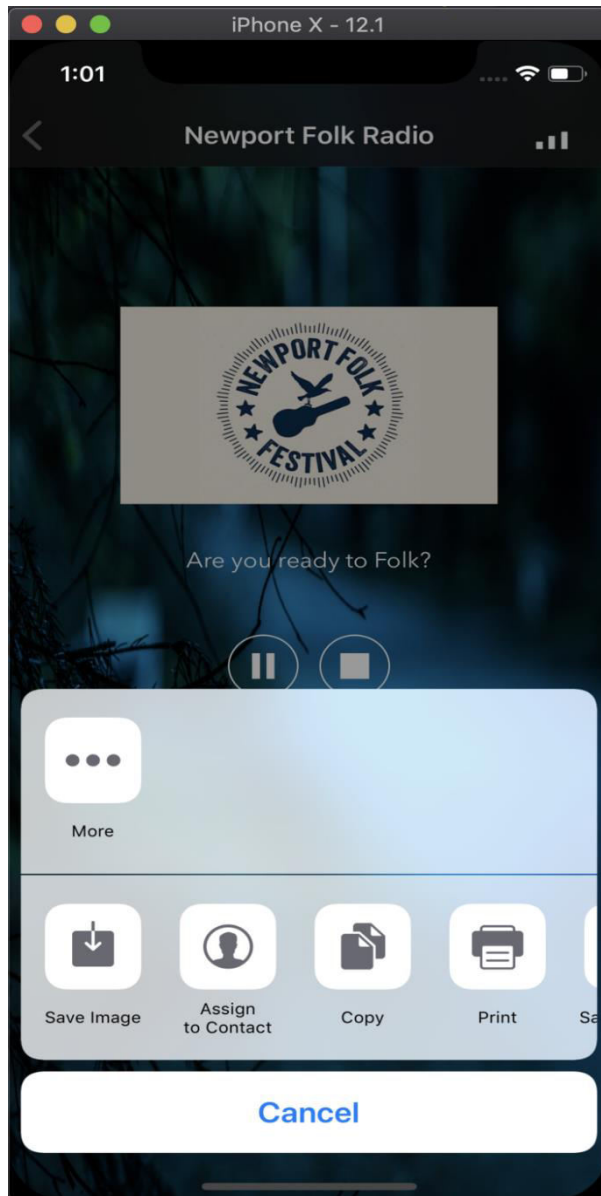


Figure 4.5: A Screenshot of Sharing System.

User can share the song with their friends to this way. Here song will share by the Air drop, Swift Radio app. If the user remain out to network range then the user can share the song by using Air drop.

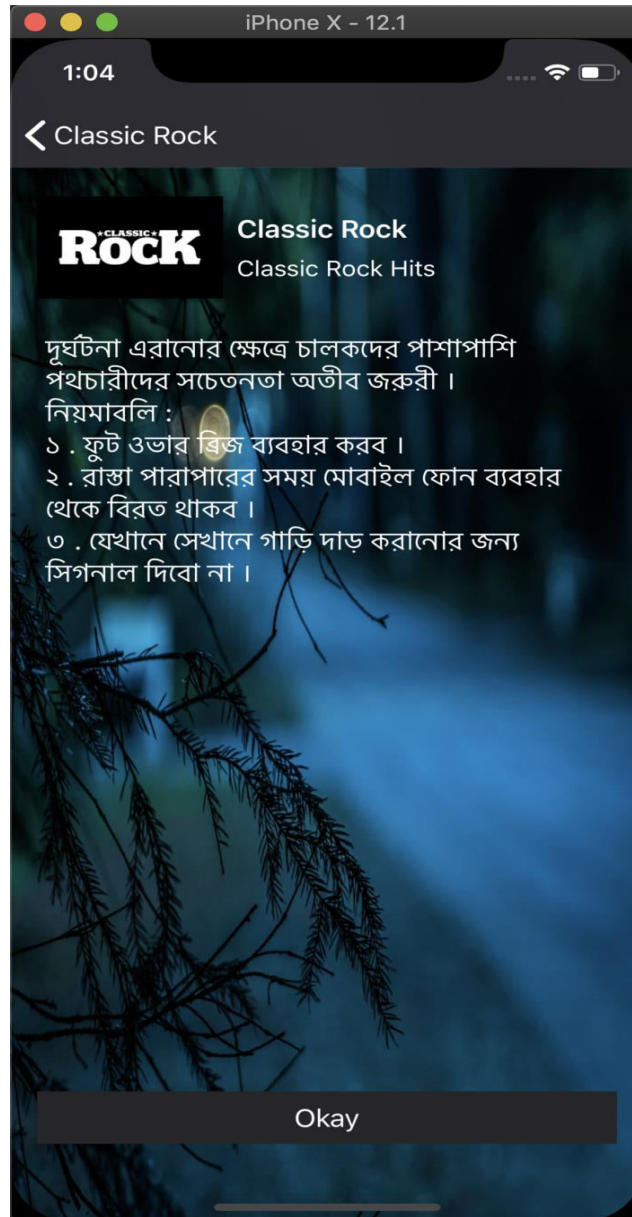


Figure 4.6: A Screenshot of Message for user.

Everyone who live in the society has their responsibility to the society. From that aspect I also think that I need to show some responsibility by doing some public awareness. User can return the pervious page by pressing Okay button.

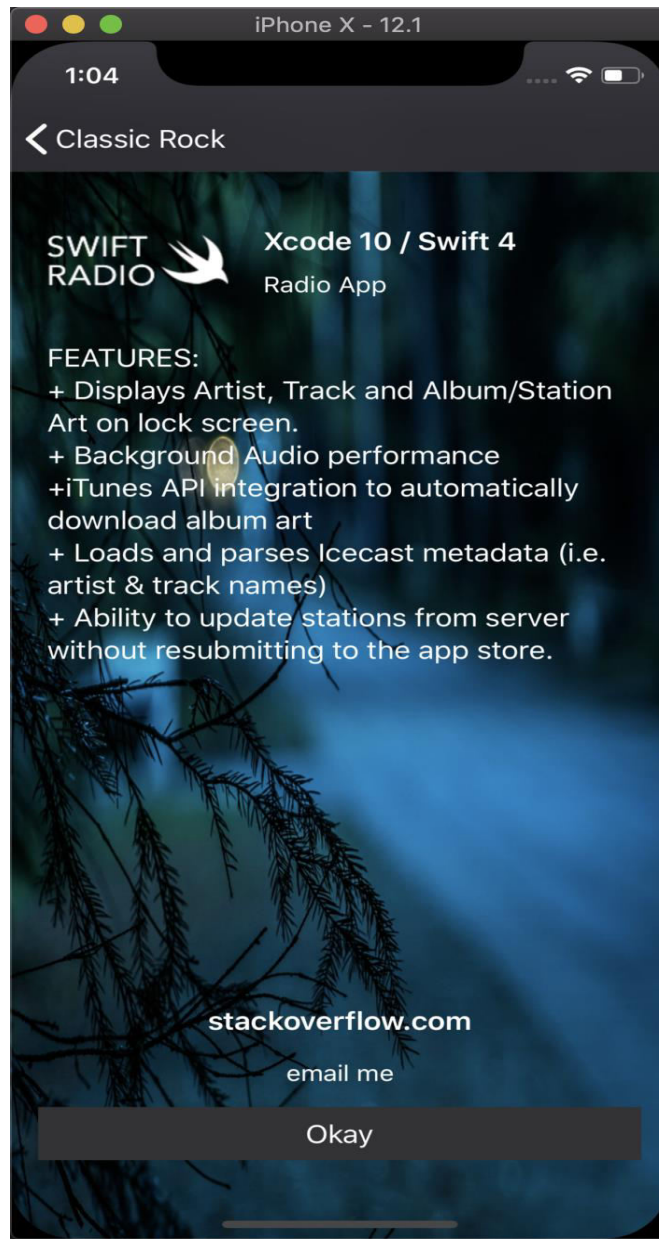


Figure 4.7: A Screenshot of Features

I got help from the stackoverflow.com website. It is open for all and very helpful for all the programmer like me.

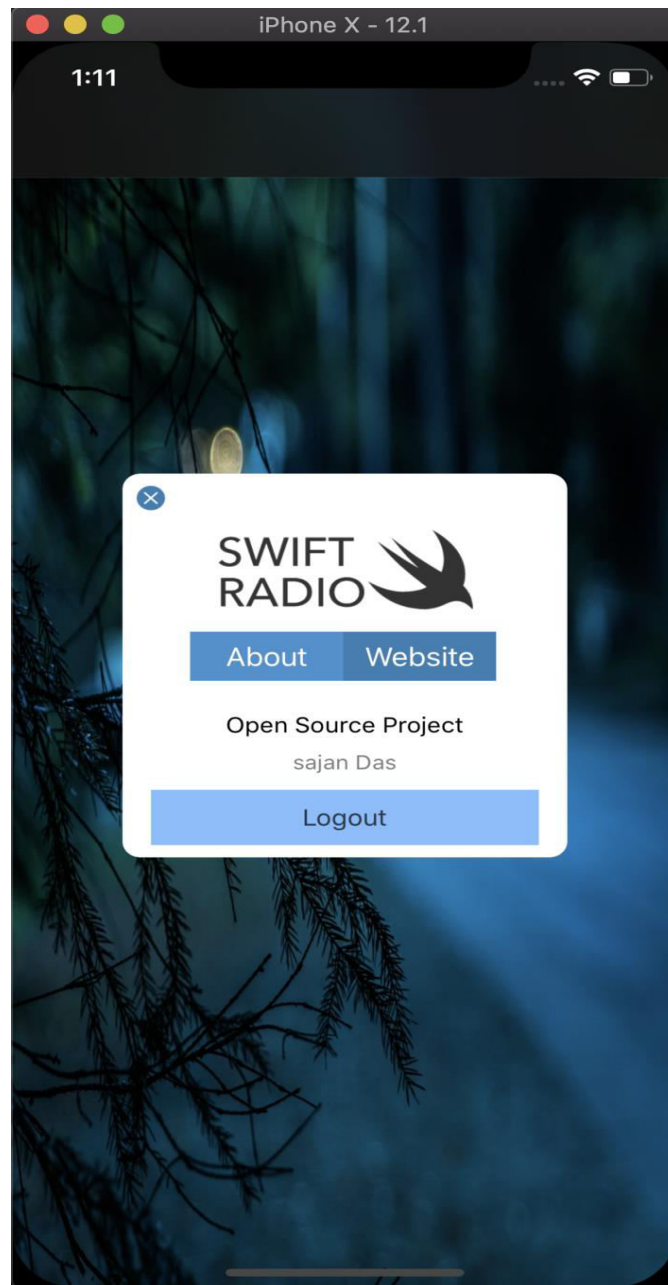


Figure 4.8: A Screenshot of Logout

Figure 4.8 show us, it is open source project, if anybody shows any curiosity on this project then the user can easily excess all the information by pressing the button Website. Here user can sign off by pressing the Logout button.

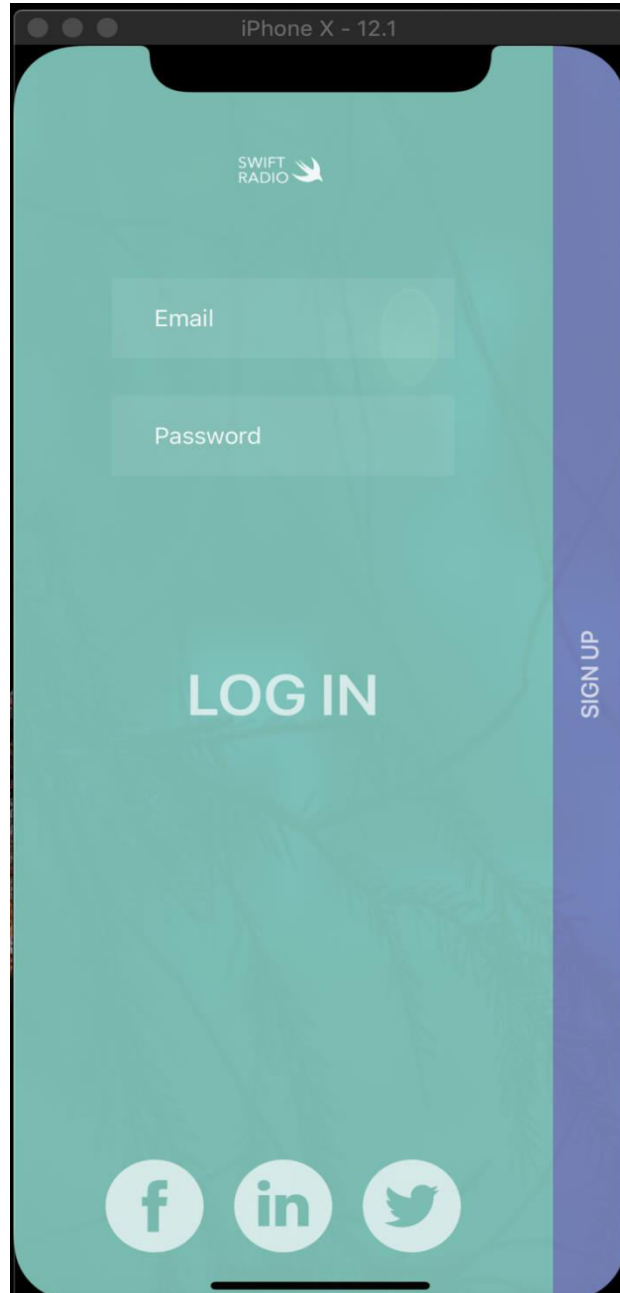


Figure 4.9: A Screenshot of Home Page

Figure 4.9, Shows us that, the startup window will show again after press the button Logout.

4.2 Back-end Design

The back-end is the main portion of an app. The front-end is the visual representation of the back-end. Back-end is the most significant part. So the structure, security, content management of it is very much important that is why I have to use security system in this app no one can assess this app without your permission. The back-end is also called the server side design. It controls the behavior of the site upon an action is taken. The back-end has generally three parts such as server, firebase database and application. I use firebase database part in here. Because of this app is an assistance type application. So I am connect firebase database in our application. My application has authentication part so that's why there are registration part. In future I will develop my application. As the application might contain of Swift Radio, details cannot be written here.

4.3 Implementation Requirement

I used different types of implementation components, attributes and cocoa framework to build our application, Make this application easily accessible and useful to user has become possible. I implement the code of the Xcode version 10 and swift code in my project. The special thing is in my application it is written by American English font.

Implementation of my project needs the followings:

4.3.1 Xcode Version 10

Xcode 10 includes everything you need to create amazing apps for all Apple platforms. Now Xcode and Instruments look great in the new Dark Mode on macOS Mojave. The source code editor lets you transform or refactor code more easily, see source control changes alongside the related line, and quickly get details on upstream code differences. You can build your own instrument with custom visualization and data analysis. Swift compiles software more quickly, helps you deliver faster apps, and generates even smaller binaries. Test suites complete many times faster, working with a team is simpler and more secure, and much more [1].

4.3.2 Framework UIKit

Construct and manage a graphical, event-driven user interface for iOS. The UIKit framework provides the required infrastructure for your iOS apps. It provides the window and view architecture for implementing your interface, the event handling infrastructure for delivering Multi-Touch and other types of input to your app, and the main run loop needed to manage interactions among the user, the system, and your app.

4.3.3 Audio Kit

AudioKit is an audio synthesis, processing, and analysis platform for iOS, macOS, and tvOS. This document serves as a one-page introduction to AudioKit, but we have much more information available on the AudioKit.

4.3.4 Radiokit SDK

You can use this swift code as a front end for a more robust streaming backend. Brain Stormont, creator of Radiokit, has created a branch with the professional Radiokit SDK already integrated [2].

4.4 Interaction

Includes full Xcode project to jumpstart development. I keep my system facile and easier for better performance and experience for the user.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation of Database

The “Swift Radio” is app based on media I successfully add to my app. First open the app and sign up the button then it will be open and fulfill the login requirement. I have to link with Pod library function therefor connect to Firebase database. User can sing up their own information purpose security. Obviously it is secure for user. I have a database part just only for manage user.

5.2 TESTING

Testing is a significant part in any app or any software. Build the app and then developer can debug and after debugging a developer can find the bug. The “Swift Radio” app is work fine with iPhone device which iPhone version 5s later to latest version of iPhone XS Max. Test-driven development is a first class workflow within Xcode. The Test navigator makes it incredibly easy to jump to any test in project, individual test etc.

5.3 Implementation of Front-end Design

It is a good challenge to develop a fantastic front-end design which will be attracted to user. It is very nice work to balance the design with iPhone any display size. I have to use animation on it which will be user get impress. So, a creator needs to check many times of their iOS application by building or running within an iOS device. I tried to think as a simple and easier in user interface design for creating user attraction and interactive design to my iPhone application. I also tried design for making out looking and creating attraction to the user. On the other hand, the user can enjoy to using a very simple and easier user interface. So it was really a good challenge to me when I was designed my iOS application user interface. But most challenging part is to make iOS application. This app only for iOS user. I attached my application front-end implemented design. This is the home page.

5.4 Implementation of Xcode and Swift Code

Using Xcode version 10 for my application I have to design my app on Main.Storyboard and swift code has to coding in AppDelegate.swift, RadioPlayer.swift and SwiftRadio-setting.swift.for database I have to use POD. POD is very important in swift code. It is a library function for database to connect app. I use Xcode platform for my application connection with internet and design. In Xcode has personal language such as Objectives C and swift are supported. I have to use object, class, and some method in swift code. That means work was done in swift code. In Xcode has library and I got for design which was I need whenever I used UIButton, UIImage, UILabel, UITextField, UITextView, Navigator Control , standard Editor, Assistant edition, Debug Area, Inspector for my application [3].

Figure 6.1 of appendix section i use the screenshot of the LoginViewcontroller.swift Swift code

Figure 6.2 of appendix section i use the screenshot of the AppDelegate.swift Swift code.

Figure 6.3 of the appendix section, it shows screenshot of RadioPlayer.swift swift code.

5.5 Testing Implementation

Table 5.6: Test Case for Project on Swift Radio

Test Case	Test Input	Expected Output	Actual Output	Result	Tested On
1.Install Application	Tested on Various – 1. iPhone 5s (12.2) 2. iPhone 6(12.2) 3. iPhone 6+ (12.2) 4. iPhone 7 (12.2) 5. iPhone 7 + (12.2) 6. iPhone 8 (12.2) 7. iPhone 8 +(12.2) 8. iPhone X (12.2) 9. iPhone XS(12.2) 10. iPhone XR(12.2) 11. iPhone XS Max(12.2)	Successfully install all those versions.	Install successful	Passed	12/3/2019

5.6 Test Results and Reports

I develop my app, based on feedback. I tested my apps our 3 friends. They gave advise to me that how i can make my system more user friendly and developed. My friend gives an opportunity to count the result of testing quickly. My app has not published in Apple App-store yet. I have no developer account. If I upload this app in App-store therefor I have to pay USD 99 to Apple. I cannot afford the amount of money right now. In future I will upload this app and develop more features. Test result are so important for any kind of application.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

I tried so hard and got so far but in end of my project iOS based mobile application has been successfully implemented. I tested the application with many types of iPhones, for implementing my application. I saw that my app worked properly and it gave the required data form my app. Authentication is needed for this application. I tried my best to complete all requirement for my application. The design is very user friendly. The user interface is not looking like complex view it is so simple. User can softly handle my application. I hope that people will use my application and get proper service.

6.2 Limitations of our Application

My application has some limitation i will overcome this limitation in future

1. User cannot modify this app.
2. User don't have any profile.
3. User cannot use negative way.
4. Can only be use 24/7 app.

6.3 Scope for further development

Here I will try to fix that as soon as possible. I will develop this app with more feature If people response good. If any user suggest to me some important think about app. I will try make change according to that suggestion.

APPENDIX

Swift Code:

```
//  
  
// NowPlayingViewController.swift  
  
// Swift Radio  
  
//  
  
import UIKit  
  
import MediaPlayer  
  
//*****  
  
// NowPlayingViewControllerDelegate  
  
//*****  
  
protocol NowPlayingViewControllerDelegate: class {  
  
    func didPressPlayingButton()  
  
    func didPressStopButton()  
  
    func didPressNextButton()  
  
    func didPressPreviousButton()  
  
}  
  
//*****  
  
// NowPlayingViewController  
  
//*****
```

```

class NowPlayingViewController: UIViewController {

    weak var delegate: NowPlayingViewControllerDelegate?

    // MARK: - IB UI

    @IBOutlet weak var albumHeightConstraint: NSLayoutConstraint!
    @IBOutlet weak var albumImageView: SpringImageView!
    @IBOutlet weak var artistLabel: UILabel!
    @IBOutlet weak var playingButton: UIButton!
    @IBOutlet weak var songLabel: SpringLabel!
    @IBOutlet weak var stationDescLabel: UILabel!
    @IBOutlet weak var volumeParentView: UIView!
    @IBOutlet weak var previousButton: UIButton!
    @IBOutlet weak var nextButton: UIButton!

    // MARK: - Properties

    var currentStation: RadioStation!
    var currentTrack: Track!

    var newStation = true
    var nowPlayingImageView: UIImageView!

    let radioPlayer = FRadioPlayer.shared

```

```

var mpVolumeSlider: UISlider?

//*****

// MARK: - ViewDidLoad

//*****

override func viewDidLoad() {
    super.viewDidLoad()

    // Create Now Playing BarItem
    createNowPlayingAnimation()

    // Set AlbumArtwork Constraints
    optimizeForDeviceSize()

    // Set View Title
    self.title = currentStation.name

    // Set UI
    albumImageView.image = currentTrack.artworkImage
    stationDescLabel.text = currentStation.desc
    stationDescLabel.isHidden = currentTrack.artworkLoaded

    // Check for station change

```

```

newStation ? stationDidChange() : playerStateDidChange(radioPlayer.state, animate: false)

// Setup volumeSlider
setupVolumeSlider()

// Hide / Show Next/Previous buttons
previousButton.isHidden = hideNextPreviousButtons
nextButton.isHidden = hideNextPreviousButtons
}

//*****
// MARK: - Setup
//*****

func setupVolumeSlider() {
    // Note: This slider implementation uses a MPVolumeView
    // The volume slider only works in devices, not the simulator.
    for subview in MPVolumeView().subviews {
        guard let volumeSlider = subview as? UISlider else { continue }
        mpVolumeSlider = volumeSlider
    }

    guard let mpVolumeSlider = mpVolumeSlider else { return }

    volumeParentView.addSubview(mpVolumeSlider)
}

```

```
mpVolumeSlider.translatesAutoresizingMaskIntoConstraints = false
mpVolumeSlider.leftAnchor.constraint(equalTo: volumeParentView.leftAnchor).isActive = true
mpVolumeSlider.rightAnchor.constraint(equalTo: volumeParentView.rightAnchor).isActive = true
mpVolumeSlider.centerYAnchor.constraint(equalTo: volumeParentView.centerYAnchor).isActive =
true
```

```
mpVolumeSlider.setThumbImage( imageLiteral(resourceName: "slider-ball"), for: .normal)
}
```

```
func stationDidChange() {
    radioPlayer.radioURL = URL(string: currentStation.streamURL)
    title = currentStation.name
}
```

```
//*****
// MARK: - Player Controls (Play/Pause/Volume)
//*****
```

```
// Actions
```

```
@IBAction func playingPressed(_ sender: Any) {
    delegate?.didPressPlayingButton()
}
```

```
@IBAction func stopPressed(_ sender: Any) {
```

```

    delegate?.didPressStopButton()
}

@IBAction func nextPressed(_ sender: Any) {
    delegate?.didPressNextButton()
}

@IBAction func previousPressed(_ sender: Any) {
    delegate?.didPressPreviousButton()
}

//*****
// MARK: - Load station/track
//*****

func load(station: RadioStation?, track: Track?, isNewStation: Bool = true) {
    guard let station = station else { return }

    currentStation = station
    currentTrack = track
    newStation = isNewStation
}

func updateTrackMetadata(with track: Track?) {
    guard let track = track else { return }

```



```

currentTrack.artist = track.artist

currentTrack.title = track.title

updateLabels()
}

// Update track with new artwork
func updateTrackArtwork(with track: Track?) {
    guard let track = track else { return }

    // Update track struct
    currentTrack.artworkImage = track.artworkImage
    currentTrack.artworkLoaded = track.artworkLoaded

    albumImageView.image = currentTrack.artworkImage

    if track.artworkLoaded {
        // Animate artwork
        albumImageView.animation = "wobble"
        albumImageView.duration = 2
        albumImageView.animate()
        stationDescLabel.isHidden = true
    } else {
        stationDescLabel.isHidden = false
    }
}

```

```

    }

    // Force app to update display
    view.setNeedsDisplay()
}

private func isPlayingDidChange(_ isPlaying: Bool) {
    playingButton.isSelected = isPlaying
    startNowPlayingAnimation(isPlaying)
}

func playbackStateDidChange(_ playbackState: FRadioPlaybackState, animate: Bool) {

    let message: String?

    switch playbackState {
    case .paused:
        message = "Station Paused..."
    case .playing:
        message = nil
    case .stopped:
        message = "Station Stopped..."
    }

    updateLabels(with: message, animate: animate)
}

```

```

    isPlayingDidChange(radioPlayer.isPlaying)
}

func playerStateDidChange(_ state: FRadioPlayerState, animate: Bool) {

    let message: String?

    switch state {
    case .loading:
        message = "Loading Station ..."
    case .urlNotSet:
        message = "Station URL not valide"
    case .readyToPlay, .loadingFinished:
        playbackStateDidChange(radioPlayer.playbackState, animate: animate)
        return
    case .error:
        message = "Error Playing"
    }

    updateLabels(with: message, animate: animate)
}

//*****
// MARK: - UI Helper Methods
//*****

```

```

func optimizeForDeviceSize() {

    // Adjust album size to fit iPhone 4s, 6s & 6s+,7,7+,8,8+,x
    let deviceHeight = self.view.bounds.height

    if deviceHeight == 480 {
        albumHeightConstraint.constant = 106
        view.updateConstraints()
    } else if deviceHeight == 667 {
        albumHeightConstraint.constant = 230
        view.updateConstraints()
    } else if deviceHeight > 667 {
        albumHeightConstraint.constant = 260
        view.updateConstraints()
    }
}

func updateLabels(with statusMessage: String? = nil, animate: Bool = true) {

    guard let statusMessage = statusMessage else {
        // Radio is (hopefully) streaming properly
        songLabel.text = currentTrack.title
        artistLabel.text = currentTrack.artist
        shouldAnimateSongLabel(animate)
    }
}

```

```

        return
    }

    // There's a an interruption or pause in the audio queue

    // Update UI only when it's not already updated
    guard songLabel.text != statusMessage else { return }

    songLabel.text = statusMessage
    artistLabel.text = currentStation.name

    if animate {
        songLabel.animation = "flash"
        songLabel.repeatCount = 3
        songLabel.animate()
    }
}

// Animations

func shouldAnimateSongLabel(_ animate: Bool) {
    // Animate if the Track has album metadata
    guard animate, currentTrack.title != currentStation.name else { return }

    // songLabel animation

```

```
songLabel.animation = "zoomIn"  
songLabel.duration = 1.5  
songLabel.damping = 1  
songLabel.animate()  
}
```

```
func createNowPlayingAnimation() {  
  
    // Setup UIImageView  
    nowPlayingImageView = UIImageView(image: UIImage(named: "NowPlayingBars-3"))  
    nowPlayingImageView.autoresizingMask = []  
    nowPlayingImageView.contentMode = UIViewContentMode.center  
  
    // Create Animation  
    nowPlayingImageView.animationImages = AnimationFrames.createFrames()  
    nowPlayingImageView.animationDuration = 0.7  
  
    // Create Top BarButton  
    let barButton = UIButton(type: .custom)  
    barButton.frame = CGRect(x: 0, y: 0, width: 40, height: 40)  
    barButton.addSubview(nowPlayingImageView)  
    nowPlayingImageView.center = barButton.center  
  
    let barItem = UIBarButtonItem(customView: barButton)  
    self.navigationItem.rightBarButtonItem = barItem  
}
```

```

}

func startNowPlayingAnimation(_ animate: Bool) {
    animate ? nowPlayingImageView.startAnimating() : nowPlayingImageView.stopAnimating()
}

//*****
// MARK: - Segue
//*****

override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    guard segue.identifier == "InfoDetail", let infoController = segue.destination as?
InfoDetailViewController else { return }

    infoController.currentStation = currentStation

    guard segue.identifier == "feature", let _ = segue.destination as? AboutViewController else { return }
}

@IBAction func infoButtonPressed(_ sender: UIButton) {
    performSegue(withIdentifier: "InfoDetail", sender: self)
}

@IBAction func featureButtonPressed(_ sender: Any) {
    performSegue(withIdentifier: "feature", sender: self)
}

```

```

}

@IBAction func shareButtonPressed(_ sender: UIButton) {

    let songToShare = "I'm listening to \(currentTrack.title) on \(currentStation.name) via Swift Radio
Pro"

    let activityViewController = UIActivityViewController(activityItems: [songToShare,
currentTrack.artworkImage!], applicationActivities: nil)

    activityViewController.completionWithItemsHandler = {(activityType: UIActivityType?,
completed: Bool, returnedItems:[Any]?, error: Error?) in

        if completed {

            // do something on completion if you want

        }

    }

    present(activityViewController, animated: true, completion: nil)

}
}

```

Figure 6.1: A Screenshot of Java Code of Swift Radio.


```

//
// RadioPlayer.swift
// SwiftRadio

import UIKit

//*****
// RadioPlayerDelegate: Sends FRadioPlayer and Station/Track events
//*****

protocol RadioPlayerDelegate: class {
    func playerStateDidChange(_ playerState: FRadioPlayerState)
    func playbackStateDidChange(_ playbackState: FRadioPlaybackState)
    func trackDidUpdate(_ track: Track?)
    func trackArtworkDidUpdate(_ track: Track?)
}

//*****
// RadioPlayer: App Radio Player
//*****

class RadioPlayer {

    weak var delegate: RadioPlayerDelegate?

```

```

let player = FRadioPlayer.shared

var station: RadioStation? {
    didSet { resetTrack(with: station) }
}

private(set) var track: Track?

init() {
    player.delegate = self
}

func resetRadioPlayer() {
    station = nil
    track = nil
    player.radioURL = nil
}

//*****
// MARK: - Track loading/updates
//*****

// Update the track with an artist name and track name
func updateTrackMetadata(artistName: String, trackName: String) {

```

```

if track == nil {
    track = Track(title: trackName, artist: artistName)
} else {
    track?.title = trackName
    track?.artist = artistName
}

delegate?.trackDidUpdate(track)
}

// Update the track artwork with a UIImage
func updateTrackArtwork(with image: UIImage, artworkLoaded: Bool) {
    track?.artworkImage = image
    track?.artworkLoaded = artworkLoaded
    delegate?.trackArtworkDidUpdate(track)
}

// Reset the track metadata and artwork to use the current station infos
func resetTrack(with station: RadioStation?) {
    guard let station = station else { track = nil; return }
    updateTrackMetadata(artistName: station.desc, trackName: station.name)
    resetArtwork(with: station)
}

// Reset the track Artwork to current station image

```

```

func resetArtwork(with station: RadioStation?) {
    guard let station = station else { track = nil; return }
    getStationImage(from: station) { image in
        self.updateTrackArtwork(with: image, artworkLoaded: false)
    }
}

//*****
// MARK: - Private helpers
//*****

private func getStationImage(from station: RadioStation, completionHandler: @escaping (_ image:
UIImage) -> ()) {

    if station.imageURL.range(of: "http") != nil {
        // load current station image from network
        ImageLoader.sharedLoader.imageForUrl(urlString: station.imageURL) { (image, urlString) in
            completionHandler(image ?? imageLiteral(resourceName: "albumArt"))
        }
    } else {
        // load local station image
        let image = UIImage(named: station.imageURL) ?? imageLiteral(resourceName: "albumArt")
        completionHandler(image)
    }
}
}
}

```

```

extension RadioPlayer: FRadioPlayerDelegate {

    func radioPlayer(_ player: FRadioPlayer, playerStateDidChange state: FRadioPlayerState) {
        delegate?.playerStateDidChange(state)
    }

    func radioPlayer(_ player: FRadioPlayer, playbackStateDidChange state: FRadioPlaybackState) {
        delegate?.playbackStateDidChange(state)
    }

    func radioPlayer(_ player: FRadioPlayer, metadataDidChange artistName: String?, trackName: String?)
    {
        guard
            let artistName = artistName, !artistName.isEmpty,
            let trackName = trackName, !trackName.isEmpty else {
                resetTrack(with: station)
                return
            }

        updateTrackMetadata(artistName: artistName, trackName: trackName)
    }

    func radioPlayer(_ player: FRadioPlayer, artworkDidChange artworkURL: URL?) {
        guard let artworkURL = artworkURL else { resetArtwork(with: station); return }
    }
}

```

```

    ImageLoader.sharedLoader.imageForUrl(urlString: artworkURL.absoluteString) { (image, urlString)
in
    guard let image = image else { self.resetArtwork(with: self.station); return }

    self.updateTrackArtwork(with: image, artworkLoaded: true)

    }
}
}

```

```

//
// AppDelegate.swift
// Swift Radio

```

```

import UIKit
import Firebase

```

```

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

```

```

    var window: UIWindow?

    weak var stationsViewController: StationsViewController?

```

```

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

```

```

// configure firebase
FirebaseApp.configure()

// MPNowPlayingInfoCenter
UIApplication.shared.beginReceivingRemoteControlEvents()

// Make status bar white
UINavigationBar.appearance().barStyle = .black

// FRadioPlayer config
FRadioPlayer.shared.isAutoPlay = true
FRadioPlayer.shared.enableArtwork = true
FRadioPlayer.shared.artworkSize = 600

// Get weak ref of StationsViewController
if let navigationController = window?.rootViewController as? UINavigationController {
    stationsViewController = navigationController.viewControllers.first as? StationsViewController
}

return true
}

func applicationWillResignActive(_ application: UIApplication) {

```

```
// Sent when the application is about to move from active to inactive state. This can occur for
certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the
user quits the application and it begins the transition to the background state.
```

```
// Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame
rates. Games should use this method to pause the game.
```

```
}
```

```
func applicationDidEnterBackground(_ application: UIApplication) {
```

```
// Use this method to release shared resources, save user data, invalidate timers, and store enough
application state information to restore your application to its current state in case it is terminated later.
```

```
// If your application supports background execution, this method is called instead of
applicationWillTerminate: when the user quits.
```

```
}
```

```
func applicationWillEnterForeground(_ application: UIApplication) {
```

```
// Called as part of the transition from the background to the inactive state; here you can undo
many of the changes made on entering the background.
```

```
}
```

```
func applicationDidBecomeActive(_ application: UIApplication) {
```

```
// Restart any tasks that were paused (or not yet started) while the application was inactive. If the
application was previously in the background, optionally refresh the user interface.
```



```

}

func applicationWillTerminate(_ application: UIApplication) {

    // Called when the application is about to terminate. Save data if appropriate. See also
applicationDidEnterBackground:.

    // Saves changes in the application's managed object context before the application terminates.

    UIApplication.shared.endReceivingRemoteControlEvents()

}

// MARK: - Remote Controls

override func remoteControlReceived(with event: UIEvent?) {

    super.remoteControlReceived(with: event)

    guard let event = event, event.type == UIEventType.remoteControl else { return }

    switch event.subtype {

    case .remoteControlPlay:

        FRadioPlayer.shared.play()

    case .remoteControlPause:

        FRadioPlayer.shared.pause()

    case .remoteControlTogglePlayPause:

        FRadioPlayer.shared.togglePlaying()

    case .remoteControlNextTrack:

```

```
        stationsViewController?.didPressNextButton()
    case .remoteControlPreviousTrack:
        stationsViewController?.didPressPreviousButton()
    default:
        break
    }
}
}
```

Figure 6.3: A Screenshot of Swift Code of Swift Radio.

REFERENCES

- [1]. 1.9.19: master branch migrated to Xcode 10/Swift 4.2 by @fethica.
- [2]. 1.21.18 Update: Swift Radio App gets a major update with **Version 2** by @fethica -
- Release Note.
- [3]. 10.6.17 Update: The AVPlayer branch migrated to Xcode 9/Swift 4
by @joemcmahon. Branch here: AVPlayer Branch [Accessed: 19-DEC- 2018].
- [4]. About Use-Case”,<http://www.nawandihalabja.com/airport-use-case-diagram/airport-use-case-diagram-best-of-47-awesome-free-use-case-template/>. [Accessed: 18-Oct-2018].
- [5]. <https://objectivec2swift.com/#/converter/code/>[Accessed: 19-NOV- 2018].
- [6]. “Logical Data Model”, https://en.wikipedia.org/wiki/Logical_data_model.
[Accessed: 19-NOV- 2018].
- [7]. <https://docs.swift.org/swift-book/LanguageGuide/TheBasics.html>
[Accessed: 19-JUN- 2018].
- [8]. <https://developer.apple.com/xcode/ide/>[Accessed: 19-MAY- 2018].
- [9]. <https://stackoverflow.com/>[Accessed: 19-NOV- 2018].

Plagiarism Report:

Plagiarism Report

ORIGINALITY REPORT



PRIMARY SOURCES

Submitted to Daffodil International University

1 Student Paper **10%**

2 swiftexample.info

Internet Source **4%**

3 developer.apple.com

Internet Source **2%**

Submitted to Napier University

4

Student Paper

1%

5

en.wikipedia.org

Internet Source

1%

www.antiessays.com

6

Internet Source

1%

7

audiokit.io

8

Internet Source

1%

Submitted to University of Northampton

9

Student Paper

1%

Submitted to B.S. Abdur Rahman University

Student Paper 1%

Submitted to INTI University College

10 Student Paper <1%

11 kissflow.com

Internet Source

<1%

12 Submitted to South Birmingham College

12 Student Paper

<1%

13 ada.csse.rose-hulman.edu

Internet Source

<1%

Submitted to University of Central England in Birmingham 14

Student Paper

<1%

15 documents.mx

Internet Source

<1%

16

Submitted to South Bank University

Student Paper

<1%

17

Submitted to University of Leeds

Student Paper

<1%

18

Submitted to Softwarica College of IT & E-

Commerce

Student Paper

<1%

19

pusher.com

Internet Source

<1%

20

shodhganga.inflibnet.ac.in

Internet Source

<1%

21

Gary Bennett, Brad Lees. "Swift 3 for Absolute
Beginners", Springer Nature, 2016

Publication

<1%

22

<1%

Beginning iPhone Development with Swift,

2014.

Publication

Exclude quotes Off Exclude matches Off

Exclude bibliography Off