

August 2015

# WICN Radio IOS application

Thinh Gia Ly  
*Worcester Polytechnic Institute*

Follow this and additional works at: <https://digitalcommons.wpi.edu/iqp-all>

---

## Repository Citation

Ly, T. G. (2015). *WICN Radio IOS application*. Retrieved from <https://digitalcommons.wpi.edu/iqp-all/752>

This Unrestricted is brought to you for free and open access by the Interactive Qualifying Projects at Digital WPI. It has been accepted for inclusion in Interactive Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact [digitalwpi@wpi.edu](mailto:digitalwpi@wpi.edu).



# WPI

*WICN iOS Application*

An Interactive Qualifying Project Report  
Submitted to the Faculty of **Worcester  
Polytechnic Institute**

In partial fulfillment of the requirements for the  
Degree of Bachelor of Science

Submitted on August, 3<sup>rd</sup>, 2015

***Written by***

Thinh Ly

***Continuation of previous IQP work from:***

Anh Tran

Thanh Long Vu

Thai Trinh

***Approved by***

Professor Vincent Manzo

## Table of Contents

Abstract.....	4
Chapter 1: Introduction.....	5
Chapter 2: Background .....	6
2.1 Analysis of the project report of the Android WICN Mobile application .....	6
2.2 Analysis of Similar Applications .....	6
2.2.1 TuneIn Radio .....	6
2.2.2. Radio FM .....	9
2.2.3. iHeartRadio .....	10
2.2.4. SLACKER .....	13
Chapter 3: Methodology.....	17
3.1 Development platforms.....	17
3.2 Graphics Design.....	18
3.3 Broadcast audio stream URL.....	20
3.4 Real time track listing(Online Playlist).....	20
3.5 Drupal functionality .....	21
3.6 Application Test Method .....	21
3.7 Language and supporting libraries.....	22
3.8 Application structure .....	22
3.8 Navigation Menu.....	23
3.10 Stream Controller .....	24
3.11 Drupal Data Retrieval .....	24
Chapter 4: Data Analysis .....	26
Chapter 5: Conclusion and future development.....	27
5.1 Development Limitation .....	27
5.2 Improvement .....	27
5.3 Deployments to Apple IOS App Store.....	27
5.4 Future Improvements.....	28
Appendix.....	30
Appendix A: Screenshots of the Application .....	30
Appendix B: Survey and results.....	38

Appendix C: Application Submission to Apple Store.....	46
References .....	47

# List of Figures

- Figure 1: TuneIn Interface ..... 7
- Figure 2: TuneIn users' profile..... 8
- Figure 3: Specific Channel Information ..... 9
- Figure 4: Radio FM .....10
- Figure 5: iHeartRADIO home page .....11
- Figure 6: Heart-Shaped tab subsections .....12
- Figure 7: PERFECT FOR tab.....13
- Figure 8: Slacker Home Page .....14
- Figure 9: Stations feature .....15
- Figure 10: My Vibe feature .....16
- Figure 11: Home Page .....18
- Figure 12: General Tab .....19
- Figure 13: WICN iOS Application Full Hierarchy .....20
- Figure 14: Navigation Menu .....23
- Figure 15: Overall quality of the Application survey response .....26
- Figure 16: WICN IOS app About Us.....29
- Figure 17: WICN IOS app Schedule.....30
- Figure 18: WICN IOS app Program Highlight .....31
- Figure 19: WICN IOS app Playlists .....32
- Figure 20: WICN IOS app Podcast.....33
- Figure 21: WICN IOS app Upcoming Events.....34
- Figure 22: WICN IOS app Newsletter Signup.....35
- Figure 23: WICN IOS app Sleep/ Alarm Clock .....36
- Figure 24: WICN IOS app Donate Button.....37
- Figure 25: Survey Question 1.....39
- Figure 26: Survey Question 2.....39
- Figure 27: Survey Question 3.....40
- Figure 28: Survey Question 4.....41
- Figure 29: Survey Question 5.....42
- Figure 30: Survey Question 6.....43
- Figure 31: Survey Question 7.....44
- Figure 32: Survey Question 8.....44

## **Abstract**

Many radio stations use mobile applications as a way to expand their broadcast to listeners. The scope of this project included the assessment of the existing broadcasting streams and listeners of the Worcester Inter-Collegiate Network (WICN) radio station, including their Android OS mobile app. However, the data suggest that most of WICN's listeners use iOS rather than Android devices. As part of this project, we developed a prototype of WICN's existing Android mobile app for iOS, and surveyed feedback about the application from potential WICN listeners. In this report, we detail our procedures, which included an assessment of existing apps for radio stations and their strengths and weaknesses with regard to functionality for the broadcaster, and feature availability for the end user.

# Chapter 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. Every radio station strives to expand their listener base, and mobile applications help facilitate this expansion. WICN – Jazz+ for New England – is no exception.

WICN is a public radio station for the New England community. Based in Worcester, WICN strongly focuses on bringing Jazz music – a unique American art form – to everybody from Worcester to Central Massachusetts, MetroWest Boston, New Hampshire, Connecticut and Rhode Island, and also offers its audience National Public Radio (NPR) news. All contents of the WICN Radio Station are accessible via on-air broadcast and online via NPR Online Services.

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. They can listen to their favorite stations even when they are out of the broadcast range, which greatly expand the audience to a broader range because installing applications is not limited to any region in the whole world. More listeners mean greater revenue for every radio station.

In 2014, WICN worked with graduate students from WPI School of Business on a market research about a mobile application that would allow listeners to listen to stream WICN's broadcast and access the WICN website's contents. Inspired by the outcome of this research, two undergraduate students from WPI formed an MQP team and collaborated with the WICN staff members to build the application for the Android mobile platform. Their work gave birth to the WICN official Android application, which currently benefits both WICN and its listeners. Audience from outside of the broadcast range can listen to the online streaming, access the schedules of music and news programs and even live concerts through the application. The research from this MQP team also suggested that a much larger part of the community uses iOS devices than Android devices.

As part of this IQP project, our team developed a prototype of WICN's existing Android mobile app for iOS, and surveyed feedback about the application from potential WICN listeners. We spent the Fall semester and the D term of the Spring semester of the 2014-2015 academic year developing the iOS version of the application, and testing the viability of this application as an effective resource for WICN listeners by surveying musicians and obtaining feedback about their experiences using the prototype. This document described all the process of our development from beginning to end including initial decisions and conversations on graphic designs and features to the final product.

## **Chapter 2: Background**

In this chapter, analysis of some previous reports of projects related to WICN was conducted. Additionally, we assessed mobile apps used by many popular radio stations and detailed their corresponding features and graphic designs.

### **2.1 Analysis of the project report of the Android WICN Mobile application**

We analyzed the MQP paper written by the actual authors of the Android application for WICN to get a better understanding of what they have achieved, what could be done to the app and what we were about to do. Inside of that we found information on previous reports regarding market and demographic research on WICN Mobile app as we mentioned in the Introduction chapter, about what the audience would want to see in the app, what features they would be using, and whether the application would actually benefit WICN or not, and how it would do that.

The project GQP #105 report – WICN Mobile Application Market Research by Diana, N.; Yao-Hua Chung, K.; and Li, X. – consists of information about how an application would benefit WICN, and what the preferences of the audience are. All aspects of Edison Jimenez and Seth Crampton’s design of the app are based on that information. Part of the research also shows that most of WICN listeners use iOS rather than Android devices. 47% of the respondents use an iPhone. 47% of them use an iPad, 30% use an Android phone and only 10% use an Android tablet. This encouraged even Edison’s project team to build an iOS application so that the app would help WICN reach the broadest audience. But due to the lack of experience in working with the programming language Objective C, the team managed to build only the Android version.

### **2.2 Analysis of Similar Applications**

We analyzed the strength and weaknesses of mobile applications from some famous radio stations. By gathering information about their features and graphics design, we can create a better application.

#### **2.2.1 TuneIn Radio**

TuneIn is a private company in Palo Alto, California that has over 100,000 radio stations and four million on-demand programs and podcasts from around the world. TuneIn has both the web site and mobile app versions. It has a mobile application for both Android and iOS. Moreover, users are able to select 22 different language interfaces.

TuneIn is an incredible internet radio station with a lot of built in functions. Moreover, it also guides new users to several basic steps when they first access into the app. For example, TuneIn first directs new audience to its general category first, including Top podcast, music,



news, local network, etc. Then in each category has different subcategories. For example, in Music subsection, it has 90s, 80s, 70s, blues. In News subsection, it has BBC, CNN and a lot of other channels.

TuneIn also allows users to connect with it through google account or Facebook account, thereby allowing us to enjoy the music and radio through a social-network system. For example, we can make friends, invite friends, share, follow the channel to our network and comment on the program. On mobile app platform like android, we can also sync our contact from google and Facebook account.

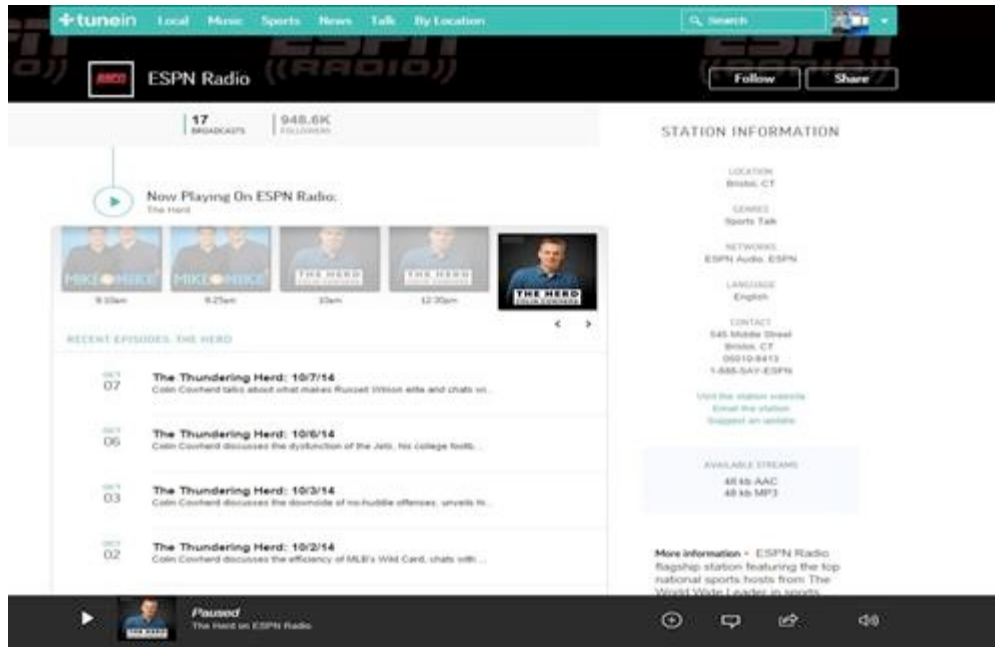


Figure 1: TuneIn Interface

Moreover, we can create our own profile with following channel and who follow us.

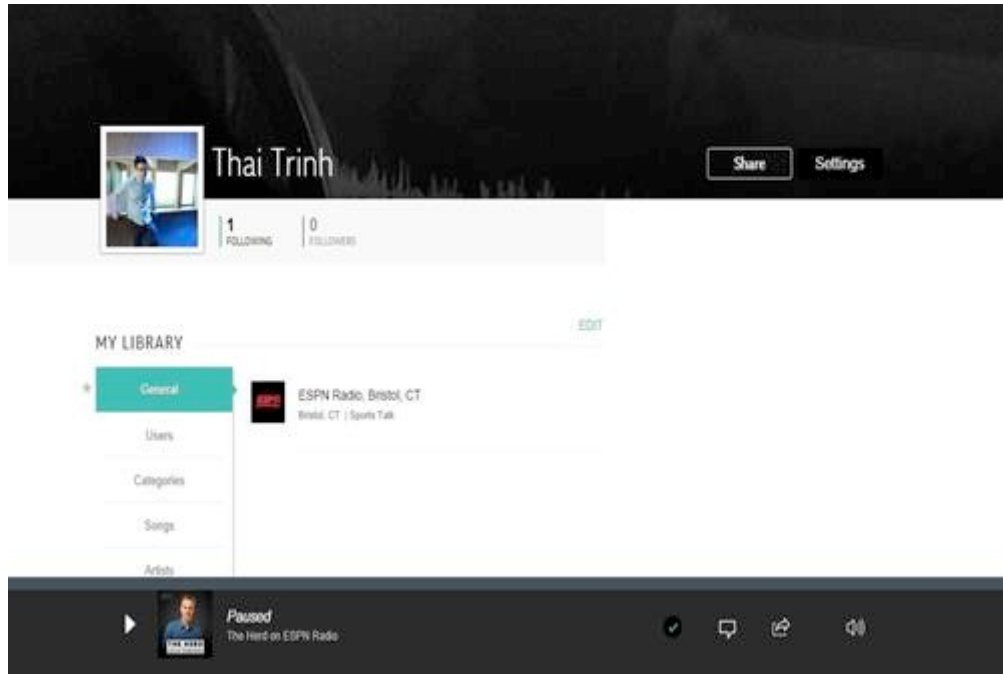


Figure 2: Tuneln users' profile

Specifically, in one channel, we are offered with a lot of built-in application of the app. For example, as we can see in below picture, the page gives us the station information, the daily schedule of its channel and also relates to update on program of other channel, which is like suggestion function on YouTube. On the mobile app we can also report station problem, set alarm for the channel, set the app to car mode and utilize the help guide for new user.

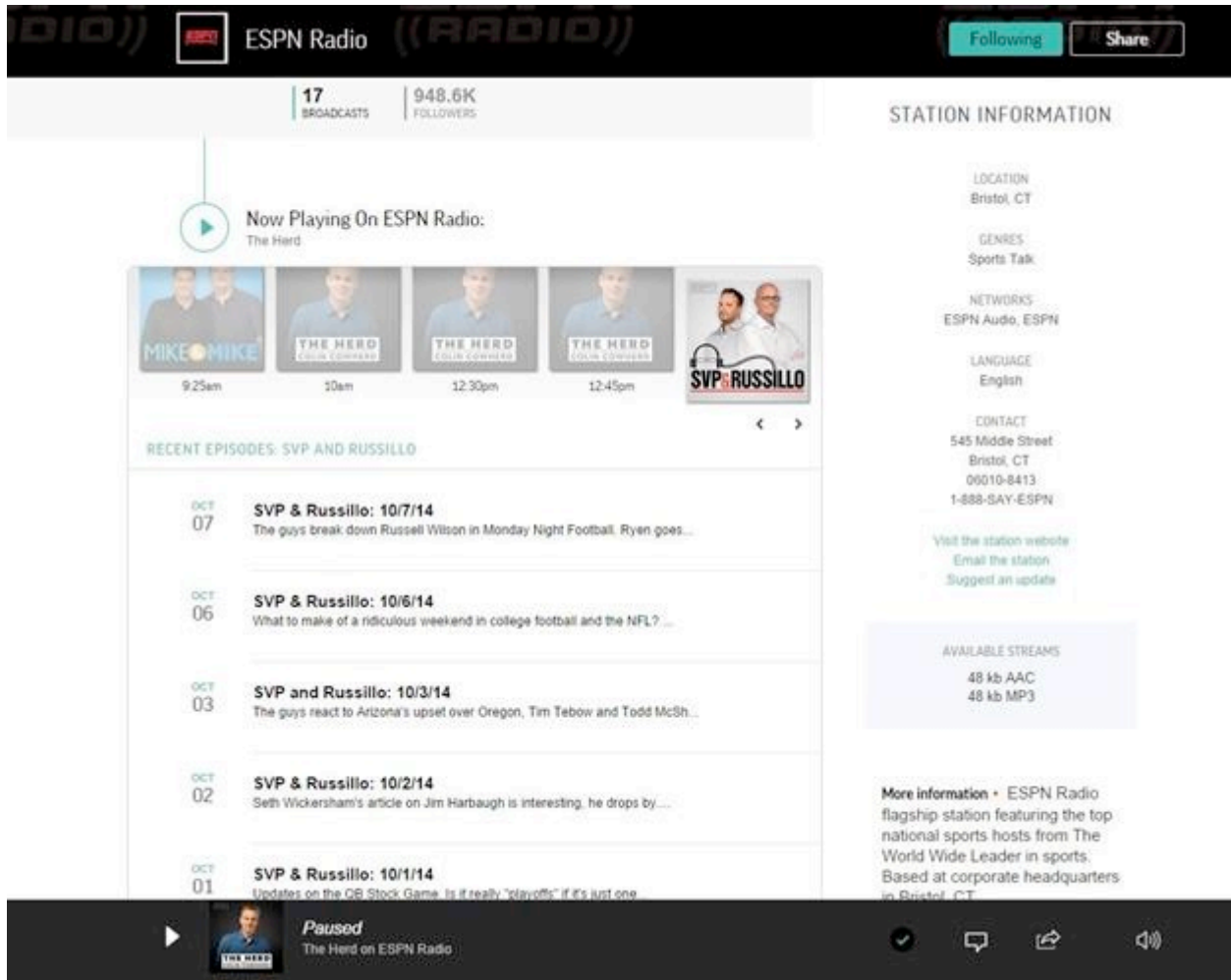


Figure 3: Specific Channel Information

## 2.2.2. Radio FM

Radio FM is a mobile app that offers several radio channels sorted by countries category. It is a simple, well organized app that fulfills most basic applications for a radio like favorite list, sharing channel to friends, sleep timer and station description. Moreover, it also records our recent listened channel as well as allows us to add more channel into our radio list. In general, it is a simple, quickly access app that is very convenient especially for usage of car or conventional radio listener, who are not in needed of those fancy and complicated feature of modern radio app.



Figure 4: Radio FM

### 2.2.3. iHeartRadio

iHeartRadio is an internet radio station that is owned by iHeartMedia, Inc. iHeartRadio functions as both a music recommender system and radio station. This product has its own internet version, mobile app version, and also on some video game consoles.

iHeartRadio requires users to utilize the app through their Facebook or Google account. This information will be used to store later record of recent channel, favorite channel and creating our own channel as well. The features of this app can be divided mostly into two sections: The heart-shaped function tab on the top left of its background and the group of three application tabs below (FOR YOU, FAVORITES and PERFECT FOR).

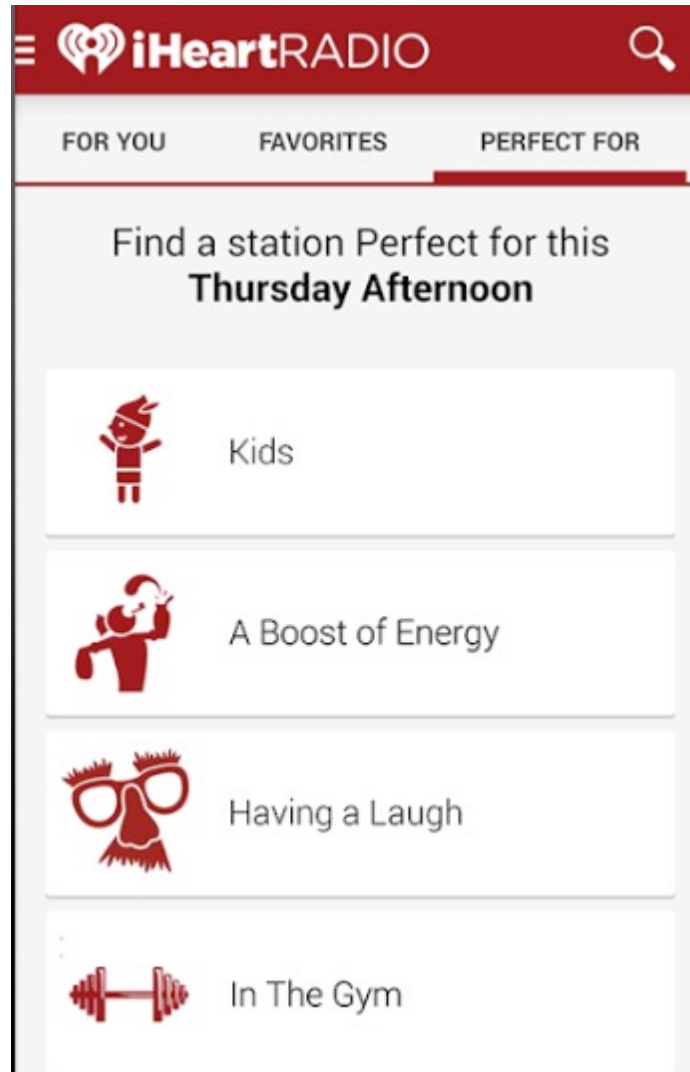


Figure 5: iHeartRADIO home page

Specifically, the heart-shaped tab includes some fundamentals options like Home, Favorites playlist, Live Radio suggestion, Create Station and Shows & Personalities. Especially, Shows & Personalities tab allows users to access on-demand episodes from the best in News, Sports, Finance, Comedy and Entertainment. On the bottom, there are some tools like alarm clock, sleep timer, setting and exiting apps.

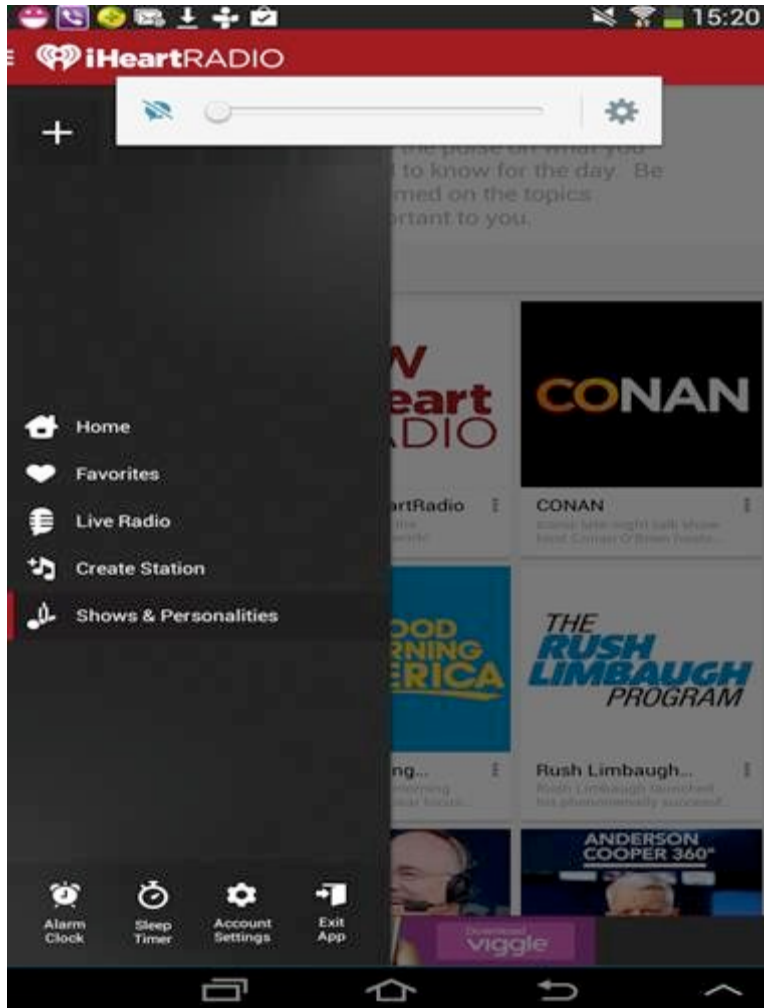


Figure 6: Heart-Shaped tab subsections

The group of three application tabs includes FOR YOU, FAVORITES and PERFECT FOR. To be specific, the FOR YOU tab has stations recommendation based on personal musical preference. The FAVORITES tab allows users to access their favorite stations from iHeartRadio home screen. The PERFECT FOR tab is a channel suggesting tools that match the time on the app to specific playlist for activities of the day. For example, in the afternoon, there will playlist for traffic jam, workout. At night, there will be playlist for sleeping or relaxing.

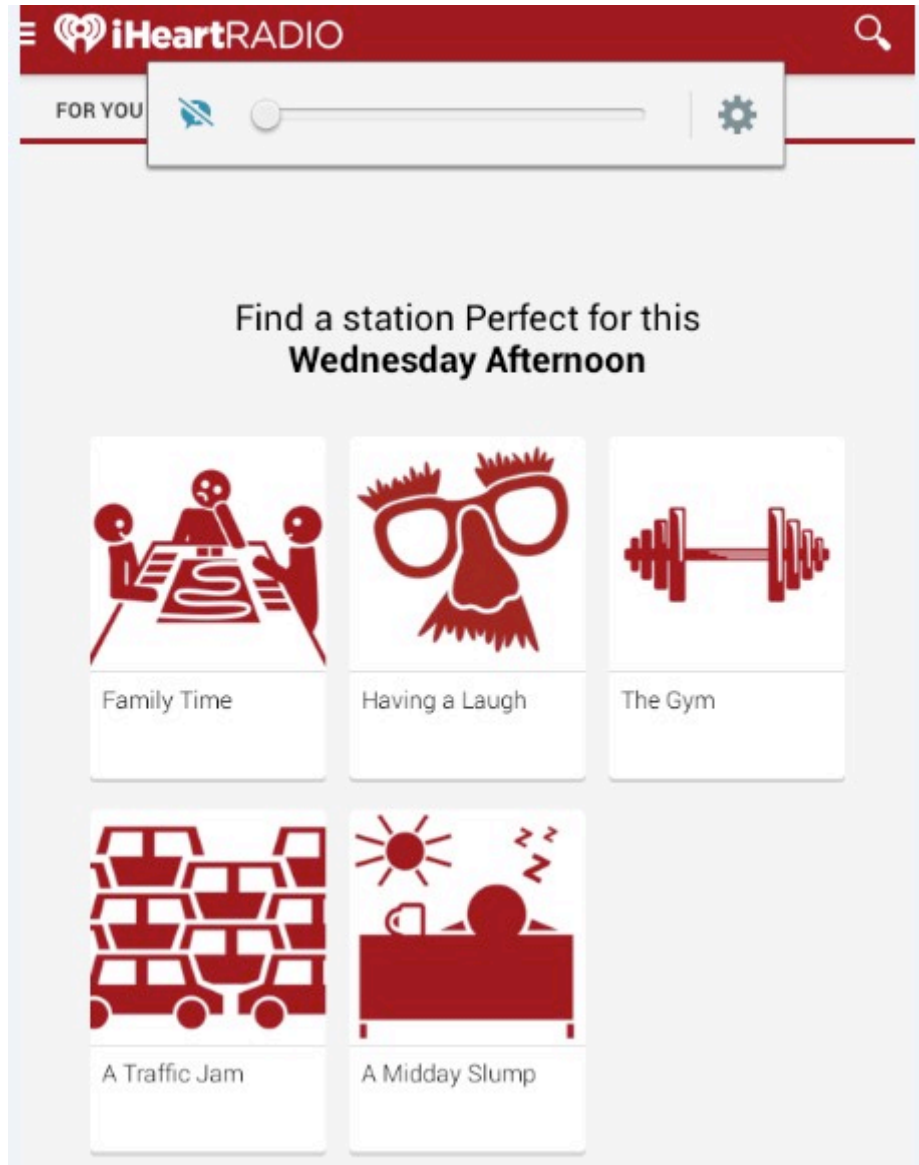


Figure 7: PERFECT FOR tab

#### 2.2.4. SLACKER

Slacker Radio is an internet radio station available in the US and Canada. Listeners can access to this music service through its website, mobile app and some other devices.

This radio app features a very special function that allows Slacker users to develop stations by rating favorite songs and banning artists or songs. Based on this, the station is then adjusted to play specific kind of music to satisfy those rating. Moreover, if users find their favorite bands and artists, Slacker will then create a station with respect to those finding result and similar musicians, allowing users to discover other artists that has the same genre and style to their favorite one.

Now, we are going to consider the specific function on the Slacker mobile app. In general, it has all basic features like the setting tab, browse tool and advertisement. The setting tab consists of account logging, upgrading, help section, setting button and exit button. Next, on the home back ground Slackers has a screen on the upper half that is a running background with suggested albums or playlist. Moreover, on the bottom half, it divides into two most important features, which are Stations and My Vibe. Right under those 2 columns are the icons of the playlists users just listened.

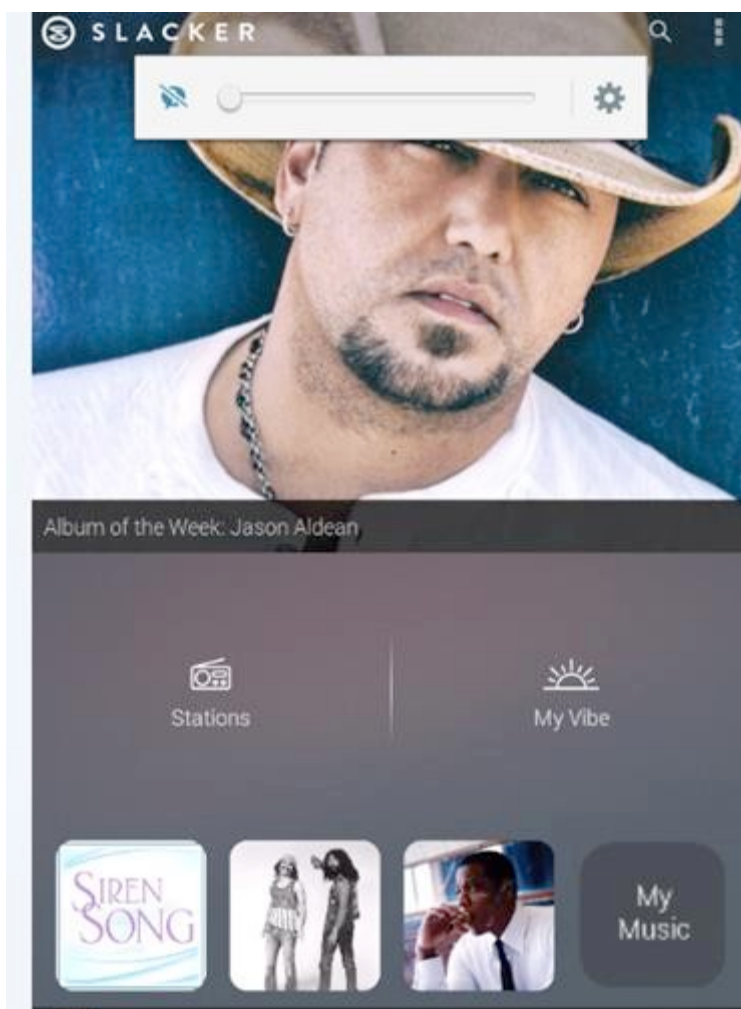


Figure 8: Slacker Home Page

Now, we will take a look at what is inside the Stations button. As figure below, it features with some genre and categories: Hits, top stations, ESPN, ABC, etc. In each playlist, there is option tools that offering the following options: View info, Bookmark, share (both the station and the song), download. Furthermore, while listening to playlist, we can favorite it as well as ban it too, which will be recorded into account history.



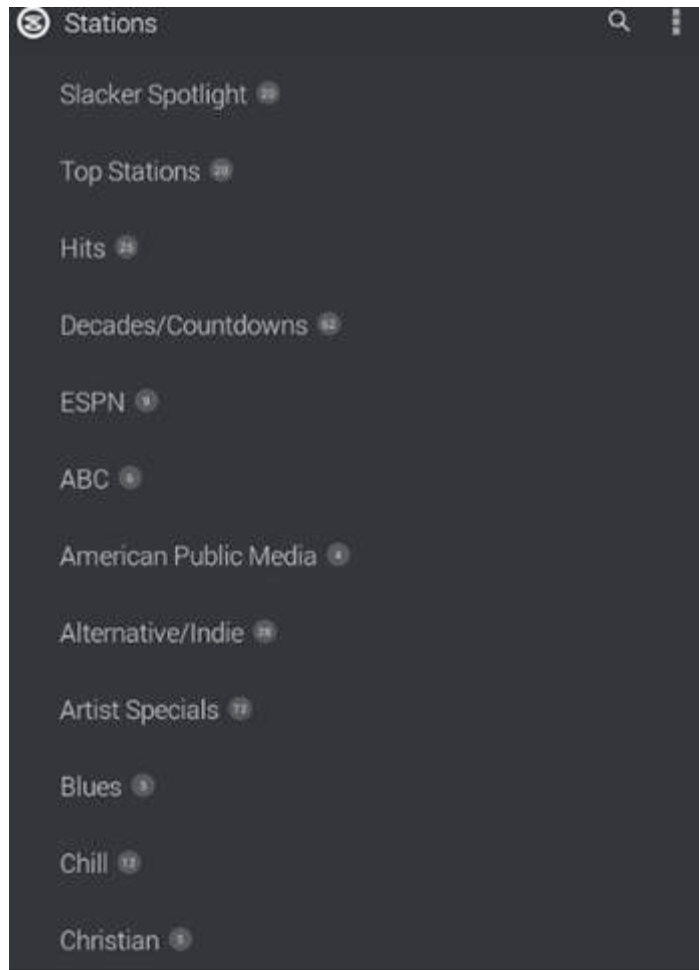


Figure 9: Stations feature

My vibe features music themes based on the time that updated in the machine with some suggested playlist. For example, in the figure below, based on the time on the app, it is Wednesday Afternoon. Slacker provides users with following subcategories: Deep dive, music discovery, going to sleep, insomnia, gaming. Each of those themes has its own corresponding type of music.

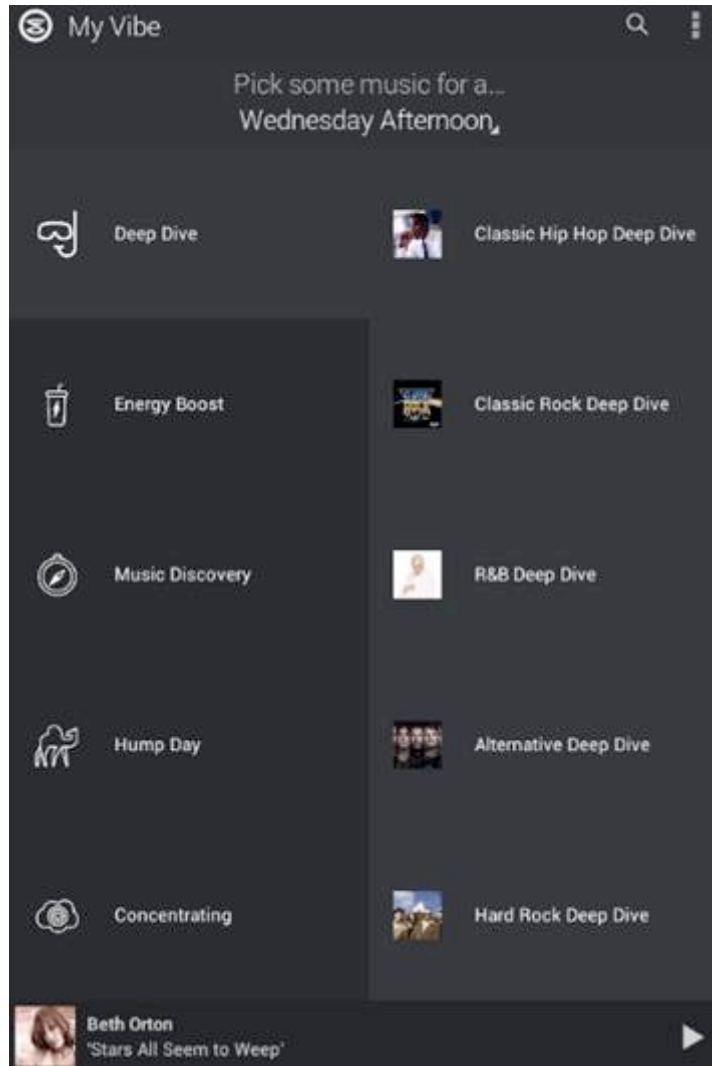


Figure 10: My Vibe feature

## Chapter 3: Methodology

The objectives of our IQP include assessing WICN's current broadcasting options for their intended audience, and developing an iOS mobile app prototype that could potential help them reach a wider audience. In this chapter, we described the creation of the prototype as well as the mechanism employed for obtaining data from a target audience via survey regarding the app's validity.

There are plenty of tools and language that can be used to code iOS, but we chose Xcode application and Objective-C language. Xcode is an application for Macbooks, and its basic function is to develop applications for Apple devices. It provides many tools and library for making an iOS mobile application. One of its advantages is that Xcode have a tool called Interface Builder, which works almost like a usual drawing application, which helped us a lot in the graphical design of the app.

The Objective-C language is a high-level coding language. It works almost like Java combine with C++, but with simpler syntax. Its method, library, helper function are all highly-leveled, so that they can easily be understood and that helps a lot in coding any complex mobile app.

Our project can be divided into 3 parts. The first part is radio streaming. Our app needs to access the streaming URL on the WICN website, and play that streaming inside the app. For this part, we used a library called AVAudioPlayer to get the URL and play the streaming from that URL. We also have 2 simple buttons for play and stop the streaming. We got reference for this part from some tutorial video on YouTube.

The second part would be the graphic design. We used Photoshop to develop the design, based much on the Android version, with some small modification. Then we implemented these graphical designs into our app. We got reference from YouTube tutorial videos, and a source file SWRevealViewController developed by author Joan Lluch.

The final part is to getting all of the web content into the app. When finished, the app functioned almost like the WICN website, with the same news, the same displayed links. To do this, we developed a web service, which accesses the source of the WICN websites, gets all news and displayed links (updated every day), and displays them on our app.

While working on the app, we debuged our code by an application called iOS simulator. This is a built-in app, go with Xcode. This app makes a simulation of IOS environment, so that we can run our mobile app on this simulated environment and test its functionality.

### 3.1 Development platforms

In this project, our IQP team developed an iOS application for the WICN radio station. The design of the app was based mostly on the Android version of it, which was developed by the MQP team of Edison Jimenez and Seth Crampton. Since the majority of WICN listeners used iOS devices, we wanted to make the app even better in this iOS version. Therefore, we had discussions with the WICN staff about the design and the features that should be added. While developing this app, we met with the WICN staff every one or two weeks to show them the progress that we made in order to make sure everything was on the right track.

### 3.2 Graphics Design

After researching some popular radio station mobile apps and the Android version of WICN mobile app, we came to the conclusion that most of them share a relatively same design with a top menu button that directs to many sub branches. Therefore we decided to implement our interface in this manner. To be specific, this is our home screen.

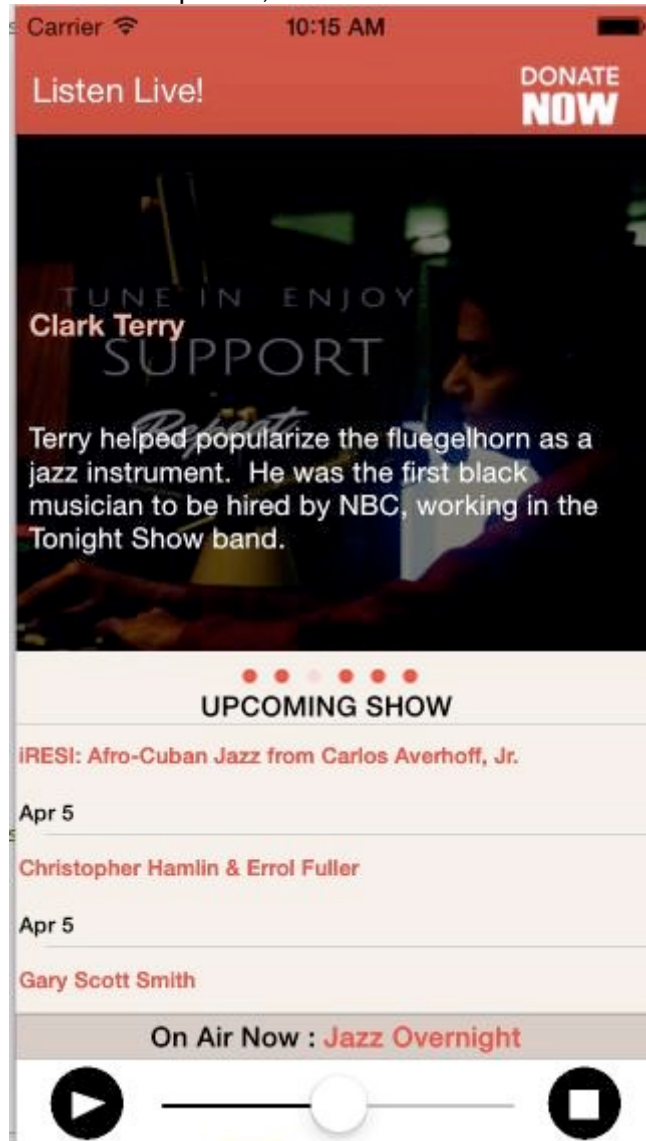


Figure 11: Home Page

As we can see, there are four sections in this Home Page design. First is Listen Live! button which directs users to the app subcategories. The Donate button is available in every tab. The news tab is at the middle and the stream player is at the bottom of the screen. The stream player feature, which the Android version does not have, is our most upgraded tool. When users click on Listen Live!, they are directed to a large table that contains all sub sections of the device, which includes Home, About Us, Schedule, Program Highlight, Playlists, Podcasts, Upcoming Events, Newsletter Signup, Sleep/Alarm Clock and Facebook.

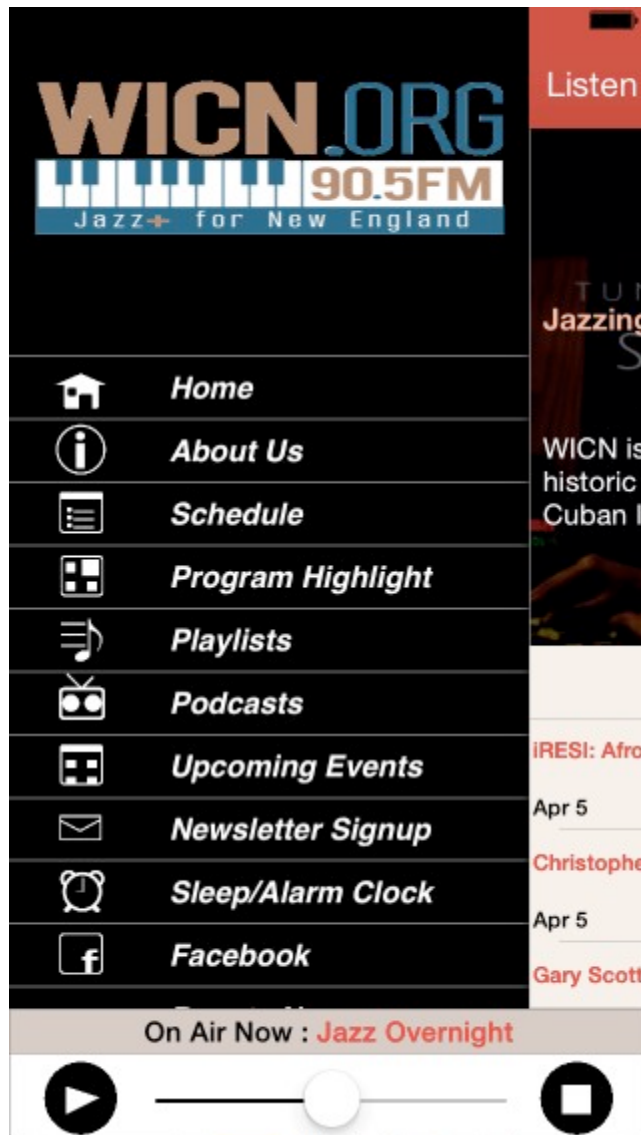


Figure 12: General Tab



Figure 13: WICN iOS Application Full Hierarchy

For more detailed pictures of each subsection, please refer to Appendix A.

### 3.3 Broadcast audio stream URL

Broadcasting audio stream inside our application required access to WICN 's online broadcast stream. There are two online stream files on WICN website to open and listen online, each for different media players.(one stream file is in .mp3 format and the other is in .pls format). To get the actual online broadcast link, we downloaded these online stream files and opened them in a text editor. This revealed the actual radio link to use on our app.

### 3.4 Real time track listing(Online Playlist)

For our project, we wanted to implement additional features that were not available in the Android app, including a playlist for currently broadcasting songs. A real time playlist is a very important feature that many other radio mobile application has. It gives users the ability to know exactly which song is playing at a specific time, or the exact time that a specific song is going to be played. However, WICN does not have a list of these current songs that are playing at a specific time except for live shows which WICN run in their Performance Hall. Songs played

in these live shows are kept track by a program called Spintron, and can be accessed through WICN website.

WICN received an additional broadcast directly from JazzWorks and there is a playlist associated with this broadcast in their webpages. However, this playlist only displays a list of songs that are played in a time duration, but not with any specific timestamps associated with each song.

For these reasons, the playlist feature is not able to satisfy real-time songs listing.

### **3.5 Drupal functionality**

In our IOS app, we redisplayed the content of the WICN website on the mobile platform. This required us to access and get the data from different sections of the site : such as “Home”, “About Us”, “Event”, “Podcast”, “Program”, etc. Each section of our app is divided into smaller sub-sections. These sub-sections contain the data and also control the way this data is displayed on the app. The sub-sections that we need to work with included the content slider in the “Home” section, the “On Air Now” title bar and the “Take Note” upcoming events schedule.

To retrieve this kind of data, one method that we could use was to parse the HTML file downloaded from the site. By parsing these HTML files, which mean to investigate the structure and format of the files, we could get access to the sub-sections and retrieve data from them. For each section of the file we could get a different HTML file. These HTML files basically contain all the links and texts that are displayed and embedded in the section, and also control the way the site is displayed. However, this approach had disadvantages. Usually, the HTML files are really large, and most of the information they contain are not necessary. Finding the correct sub-sections and retrieving correct data require a lot of analyzing and coding. Furthermore, these HTML files are really prone to changes. Since we had to get data from the HTML files, our code would have to strictly follow these files’ format and structures, so if there was going to be any changes in the format of these files the application would not be able to recognize the information we need to retrieve anymore, and we would have to recode the program.

Therefore, we chose a different approach. We were informed that WICN uses Drupal Content Management System, an open-source content management framework, to manage their website. Drupal provides features like menu-management, page layout customization, system administration and so on, and is used as administrative views for a massive amount of websites all around the world. Since the MQP team of Edison Jimenez and Seth Crampton had already installed several Drupal modules to this system to access the different sections of the WICN website, we reused these modules in our app. These modules are similar to the HTML files that we mentioned above, however they can be divided into smaller sections. They contain several pieces, each is a JSON-format file that manages a small section of the site, and all can be accessed through URLs provided by WICN. These files are much smaller than the HTML files from the website, therefore we could easily retrieve the contents in our application.

### **3.6 Application Test Method**

During the development of the application, we tested it both on our own iOS devices and on the iOS Simulator built-in with Xcode. The iOS Simulator allowed us to test the application on different versions of iOS and also different device sizes. The testing process from different

perspectives gave us a better understanding on how to deal with displaying the app on different screen resolutions and on audio quality.

After the development process, thanks to the help of our advisor, Professor Manzo, and Professor Falco from WPI Music Department, we were able to distribute a survey for our app. The survey is directed to thirty WPI Music students who are in the Jazz band or enroll in related courses for their feedback on the application. The survey took three minutes to complete. Since the WICN Radio Station needed time to get the application on the Appstore, we created a short video demonstrating the appearance and functionality of all the features inside the app, uploaded it on YouTube and attached it in the email along with the survey that was sent to students. The links to the survey and YouTube video can be found in Appendix B

### **3.7 Language and supporting libraries**

We developed our application using the integrated development environment (IDE) Xcode. Xcode contains a suite of software development tools developed by Apple for building softwares for the OS X and iOS platforms.

Our application is written in Objective-C language. Objective-C is a popular programming language, with built-in libraries specifically made for developing applications on the iOS platform. In addition to these built-in libraries, we also used 3 external libraries: ViewPagerController library, developed by MIT developers, STKStreamingKit for streaming podcasts, developed by Thong Nguyen and SWRevealViewController for Main menu, developed by Joan Lluch. All of these libraries are open-sources with copyright permission included in the project files and will be included later in this report.

To manage our source code, we used a Github account which is shared between all members of our IQP team. With the Github account, every member can easily access the project at any time, and if one person accidentally update the Github project to a version that is not working, we can revert back to the versions that other members have in their computers without any problem.

### **3.8 Application structure**

We divided our program into different pages, and each page is managed by an Objective-C class. Each class contains the attributes, declarations and methods for every action that can be performed in the page that the class manages, such as viewing the Donate web page, playing the radio stream etc. The class also determines how these pages are going to be displayed in different devices with different screen resolutions.

We created a XIB file in our project to design the application's views. In the XIB file, we have the design of each page in the app, each of these designs is linked to its according class. All of the pages are connected together through the Menu View. The Menu View is a table with multiple cells, with each cell leads to a specific page. When users tap on a cell in the Menu View, it will load the page linked with it.

One of the first challenges that came up when we developed this application was streaming the radio concurrently with other activities in the application. An obvious solution that came to our mind is concurrency. It is a technique in which the program can do multiple jobs at the same time by managing each job by a separate task. This approach is very popular in mobile application development. However in our case, we wanted to be able to turn on and off



the radio stream from every page in the application. That means we would have to include the code to retrieve the stream and also manage concurrency in every Objective-C class that we were going to use, which is very complicated.

Therefore, we came up with a different approach to solve this problem. We used an Objective-C library called ContainerController, which enabled us to add 2 different child containers into one parent view. We used one child for the radio stream player, and the other child for all other views in the application (e.g Home, About Us, Podcast...). These two containers are separated, meaning that if we wanted to switch to a different page (e.g from About Us to Podcast...), the action would not affect the streaming radio since it is bounded only in its container. The approach is a little bit slower in performance than concurrency, but easier for us to implement and to manage.

### 3.8 Navigation Menu

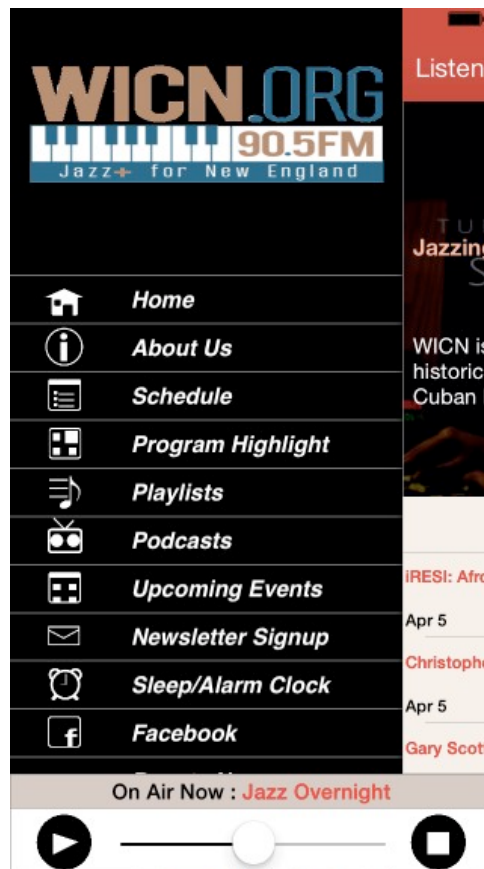


Figure 14: Navigation Menu

Above is the picture of the Navigation Menu in our application. We developed this Menu using an external library called SWRevealViewController. This public source library is frequently used to develop Navigation Menu in IOS applications. By using this library, we only needed to add a few lines of code to each Objective-C class to make sure we could open the Navigation Menu from each view (e.g About Us, Podcast...) and link all of these views into the Navigation Menu view in the XIB file, to make the Navigation Menu works.

In the top left corner of every page, there is a Menu button. Touching this button will fetch the Navigation Menu. Touching any cell in this Navigation Menu will navigate our application to the corresponding page.

We designed the cell icons in the Navigation Menu ourselves using Photoshop CS6. We drew these icons in medium scale, then moved them to the folder of our application. To add them to the menu, we created a place-holder in each cell that sits right in front of the cell's name. When the cells are loaded, their corresponding icons will also be loaded into those place-holders with constant sizing and spacing shared among all cells.

### **3.10 Stream Controller**

In our application, we had to fetch the radio streaming and the podcast audio files directly from the WICN website. The main difference between the radio streaming and the podcast files is that the radio streaming is non-stop. That means to play the radio we need to fetch data from the stream continuously and play them synchronously with the actual radio stream broadcast from WICN website. However, podcast audio files are not streaming audio files. We used the AVAudioPlayer class, which is a built-in Objective-C class, to stream the radio, but we can't do the same thing to podcast files. In order to play a podcast audio file we would have to download the whole file and save it to memory first, which could prove to be very time consuming even with strong Internet connections, and would eventually result in slow performance.

Therefore, we downloaded an open-source library called STKStreamingKit to play these podcast audio files. This library allowed us to "stream" a podcast file in the same way we did with the radio streaming, which means we could download the file partially and play these downloaded parts while downloading the others instead of having to download the entire podcast file at once and then play it later. This gave us a much better performance since now podcast could be streamed similarly to the radio streaming.

One more challenge that came up when dealing with podcast audio is that we fetched the links of the podcast audios in one Objective-C class, but we play them in a different class. This requires sending data between Objective-C classes. Therefore, we implemented a Notification object. When user touches a link in the Podcast View, this Notification object will signal a notification. This notification contains the audio link, and will be received by the class that controls the audio player. Then this link will be played by the STKStreamingKit object. Since the sending and receiving processes are executed almost simultaneously, they will not affect the application performance.

### **3.11 Drupal Data Retrieval**

Since we already had the Drupal links managing different views and sub views in the WICN website, we only needed to fetch data from these links and implement these data into our code. To get the best performance, we have an Objective-C class, in which we get all the data we need in the start of our program, then we save these data into permanent files in our storage. Whenever we need to reload a view, we don't need to fetch the data again from the web, instead use the data saved in permanent files in storage. These files will be refreshed with

new data after a specific time, to guarantee that the application will always be updated with the newest data from the WICN site.

## Chapter 4: Data Analysis

As previously mentioned, we sent survey along with the link to the application description YouTube video to WPI Music Students thanks to Professor Falco of WPI Music Department. We began collecting data between April 1st, 2015 and April 20th, 2015. Although the number of respondents was less than our expectation, we get positive feedback about the application and determine what users might and might not want to see in our application.

Of 17 responses we received, there is only one person is not satisfied with our work. Ten respondents think that this application exceeds their expectations, and three respondents think that it is exceptional.

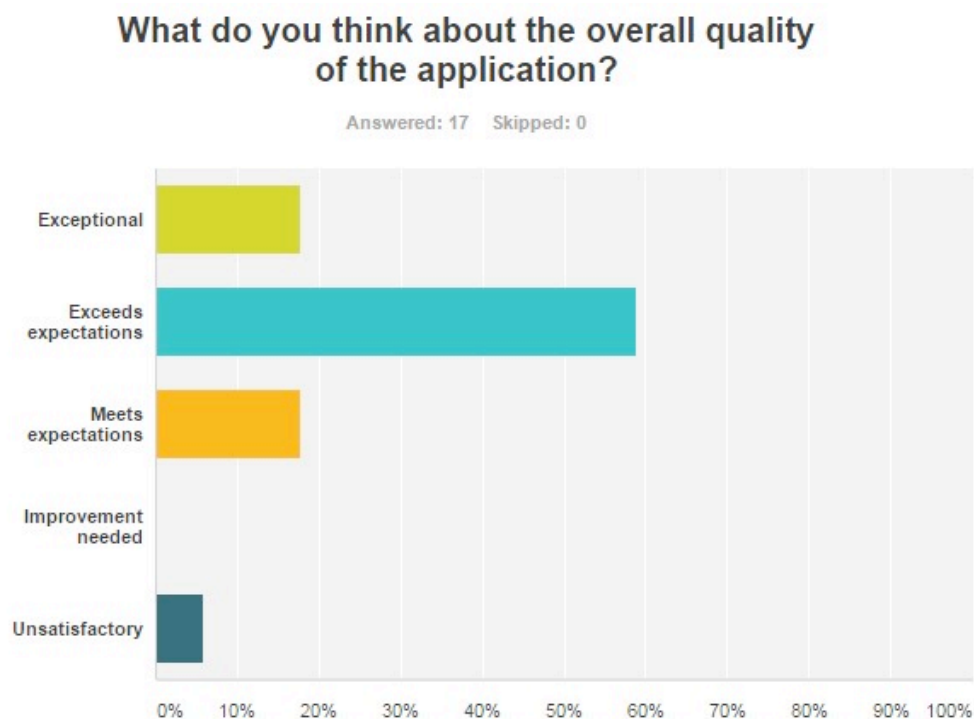


Figure 15: Overall quality of the Application survey response

When we asked if there is any feature that is unnecessary, only one respondent said: “Donate Button”. In addition, the “Donate Button” is the least important feature in the application based on the survey responses (52.94%). The rest of the respondents are satisfied with all the features of the application. We also got the feedback that the application looks user friendly with a clean, clear layout.

Overall, the team proud with the results of surveys, mostly indicate positive feedback about our application. However, we noticed that the Donate Button appeared on every page is distracting.

The survey and its results can be found in Appendix B.

## **Chapter 5: Conclusion and future development**

This project offered our project team a great amount of new programming aspects and extensive research. In a nutshell, all team members were satisfied with the final version of the application. The application has all the information that we wanted to deliver to users, while it is very easy to use and has no known bug or error.

### **5.1 Development Limitation**

In the beginning, we wanted to design our application to work in different versions of iOS devices, with the same display and the same performance, and we also wanted to include a real-time playlist, which could show the users what song are playing and what song will be play next on the broadcast.

However, we cannot include a real-time playlist, simply because WICN does not keep track of what song are playing on the broadcast. Instead, we include a playlist that is only the copy of 2 different playlists that could be found on WICN's sites, which are not actual real-time playlists.

Although we can run our application on 4 different versions of iPhone (4, 5, 6 and 6 plus), on iPhone 6 plus our application have a small but remarkable issue on display. In iOS application development, instead of the application automatically resized when being run on different screen size, we have to do all of this scaling job manually. Therefore, the size and the relative position of different sections on the view are not the same to our initial design.

To address and fix these limitations will consume a lot of time, and one of them is not feasible because of the way WICN managed their sites and online stream, so we decided to submit our product and will address these issues at a later time to continue with another part of this IQP project.

### **5.2 Improvement**

We had several improvements to the Android version of the application. We included the Play and Stop button in every different view in the app, so that users do not need to go back to the Home View to start or stop the radio. We also used these same buttons for managing podcast audio, so that instead of staying on the Podcast View while a podcast audio is played in the Android application, user can switch to different view on the application while still listening to podcasts.

One most important improvement in our application is that, in the Android application, whenever the user clicks on a link, the phone will switch to the browser application and open this link on the browser. The users will then have to go back into the app to continue using it. However, in our iOS version, we implemented the web content inside the app, so that users do not need to navigate to the phone's browser to open a link. Instead, this link will be opened right inside the app.

### **5.3 Deployment to Apple IOS App Store**

The previous version of this app by the 2014 MQP group was successfully deployed to the Android operating system. In June 2015, after the submission of this paper by our IQP team in D Term 2015, our app was submitted and approved by Apple to be distributed for free on the iOS app store. The process for submission and approval of apps is sometimes tedious and altogether much more heavily scrutinized than Android app store submission. See Appendix C for more information on the process we followed that led to app store approval for our app.

## **5.4 Future Improvements**

Alongside with improvement mentioned in Section 6.1, we want to better improve our application performance. Sometimes our application has slow response time when switching between views, or when trying to play the radio due to the network connection. We hope this issue can be fixed in the future.

# Appendix

## Appendix A: Screenshots of the Application

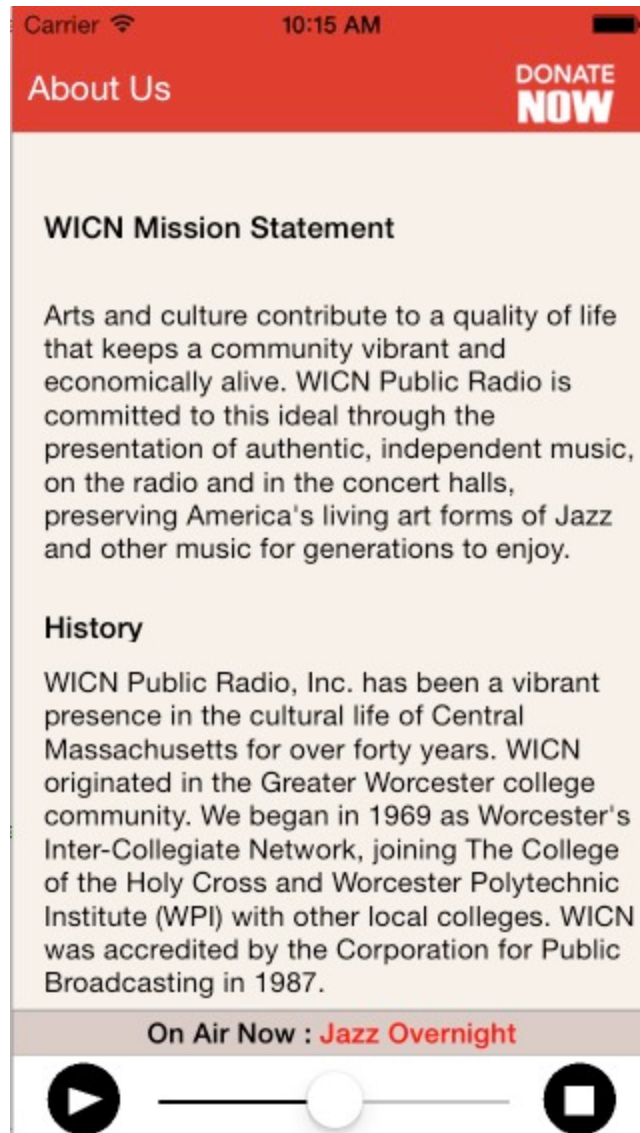


Figure 16: WICN IOS app About Us

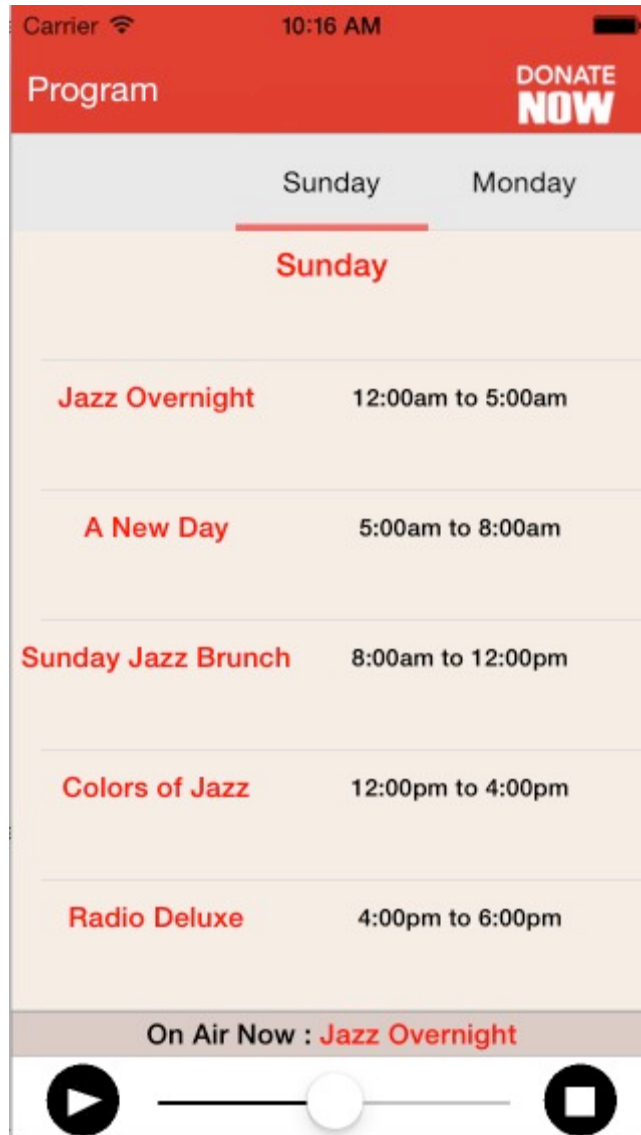


Figure 17: WICN IOS app Schedule





Figure 18. WICN iOS app Program Highlight

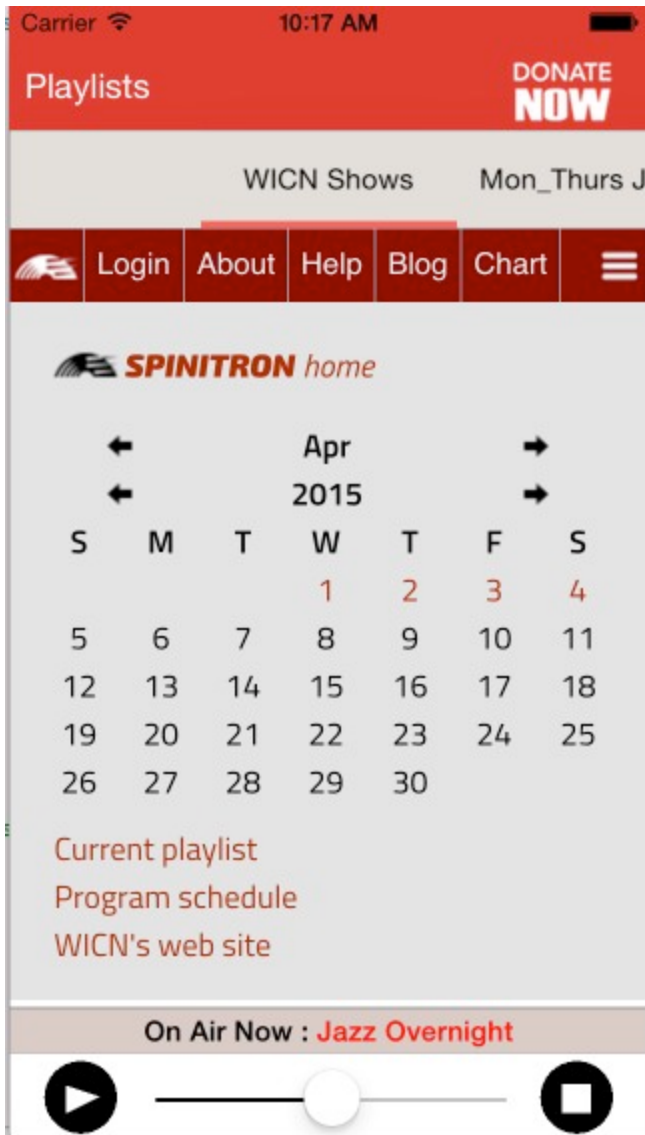


Figure 19: WICN iOS app Playlists

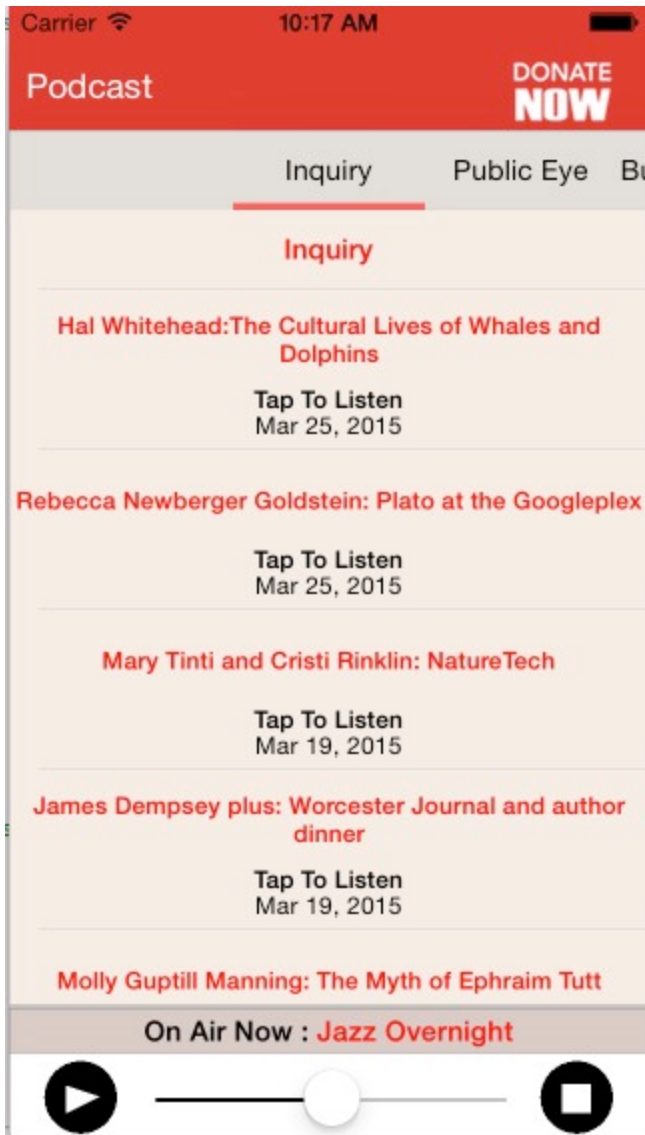


Figure 20: WICN IOS app Podcast

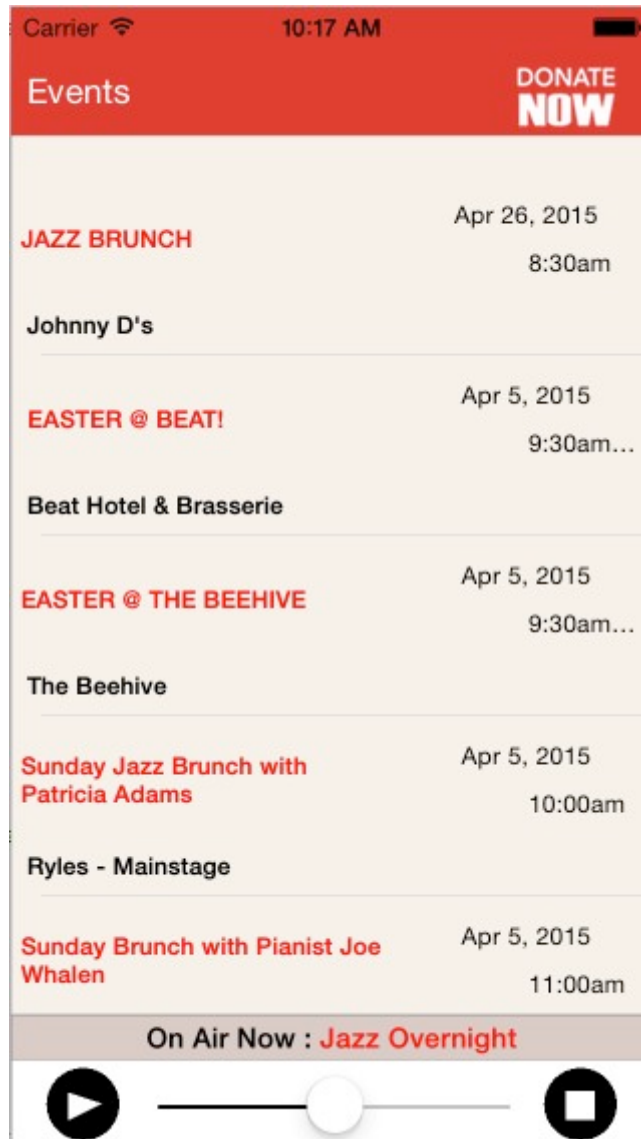


Figure 21: WICN iOS app Upcoming Events

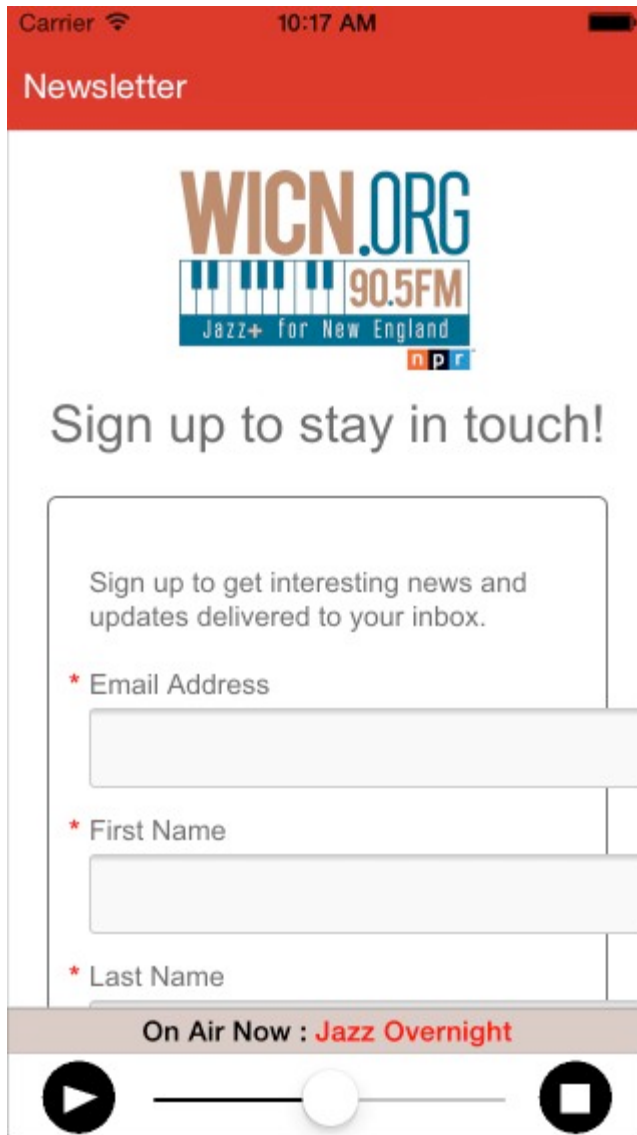


Figure 22: WICN IOS app Newsletter Signup

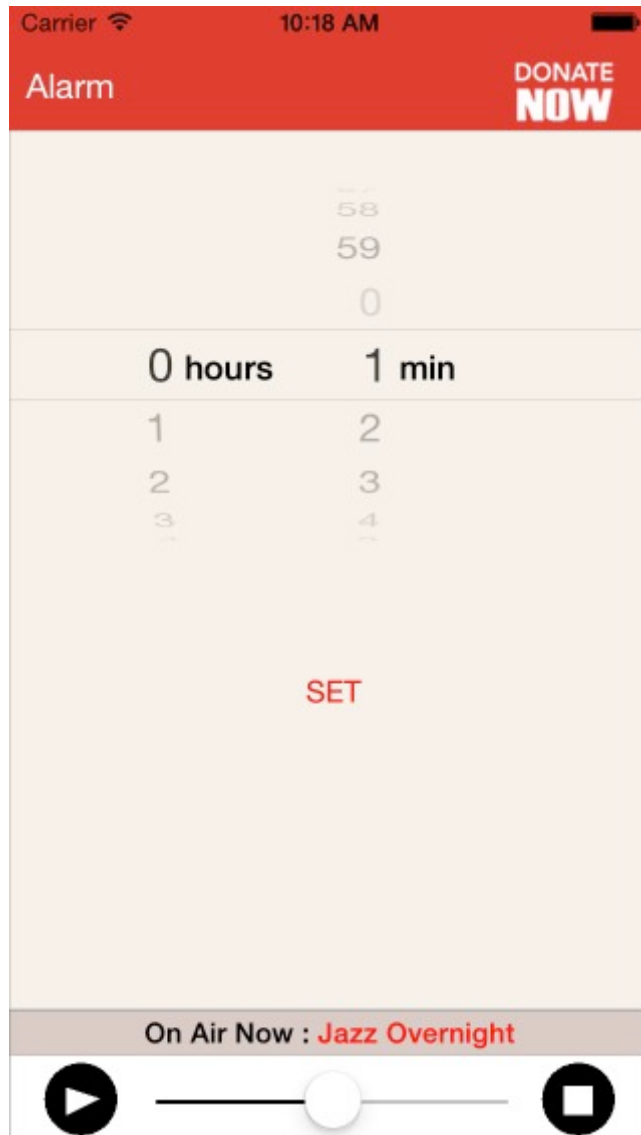


Figure 23: WICN IOS app Sleep/ Alarm Clock

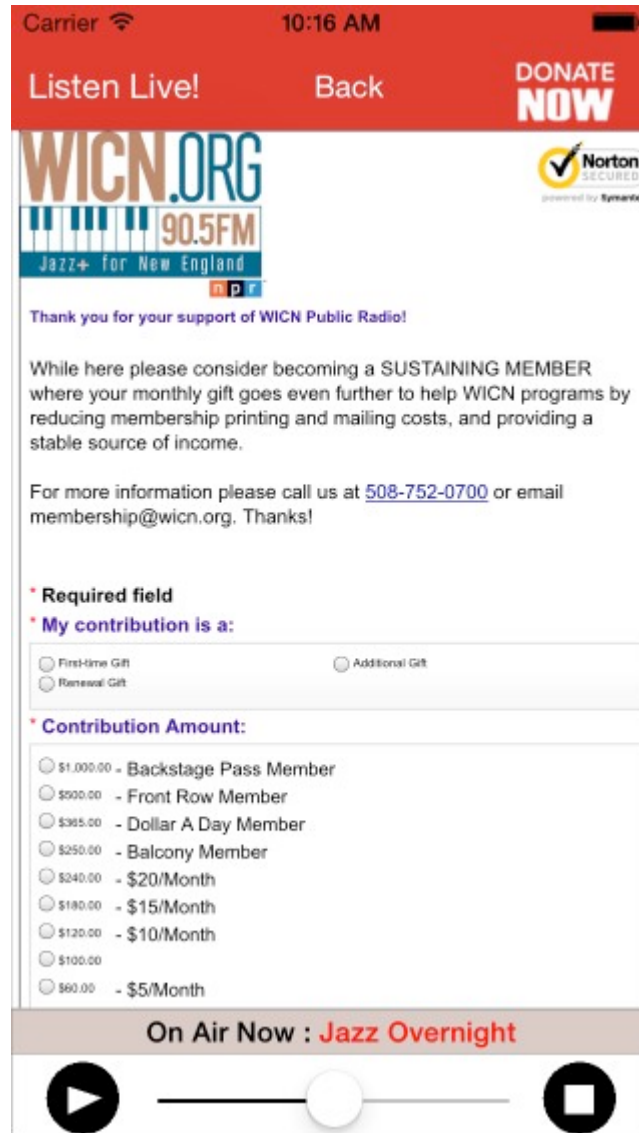


Figure 24: WICN IOS app Donate Button

## Appendix B: Survey and results

The link to the video is below:

<https://www.youtube.com/watch?v=vY0Yd-YcXgw&feature=youtu.be>

The link to the survey is below:

<https://www.surveymonkey.com/s/6898SHB>

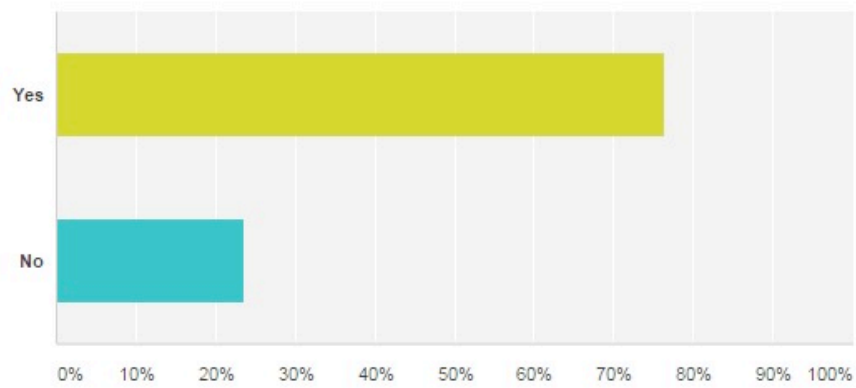
1. What is your age?
  - a. Under 21
  - b. 21-29
  - c. 30-39
  - d. 40-49

- e. 50-59
- f. 60 or older
- 2. Do you own an Iphone?
  - a. Yes
  - b. No
- 3. How often do you listen to WICN Radio?
  - a. Not at all
  - b. Less than 1 hour per week
  - c. About 3 hours a week
  - d. More than 5 hours
- 4. If you use smartphone to listen to music, what applications do you use?
  - a. Pandora Radio
  - b. iHeartRadio
  - c. Spotify
  - d. SoundCloud
  - e. Shazam
  - f. No, I do not use my phone to listen to music
  - g. Other (please specify)
- 5. Rate the importance of the following features (1 is the most important)
  - a. Live streaming
  - b. Alarm Clocks
  - c. Schedule upcoming events
  - d. Playlists of music
  - e. Facebook site of the radio station
  - f. News about WICN featured in the home page
  - g. Donate button
- 6. What do you think about the overall quality of the application?
  - a. Exceptional
  - b. Exceeds expectations
  - c. Meets expectations
  - d. Improvement needed
  - e. Unsatisfactory
- 7. Are there any features of the application that you felt were unnecessary? If so, please describe them.
- 8. Do you have any additional questions or comments?



## Do you own an Iphone?

Answered: 17 Skipped: 0



Answer Choices	Responses	Count
Yes	76.47%	13
No	23.53%	4
Total		17

Figure 25: Survey Question 1

## What is your age?

Answered: 17 Skipped: 0

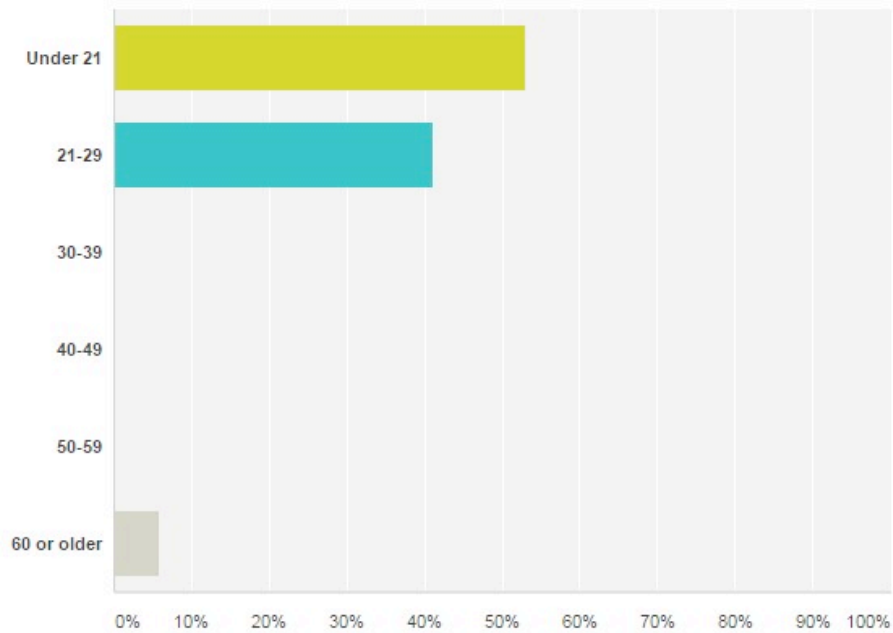
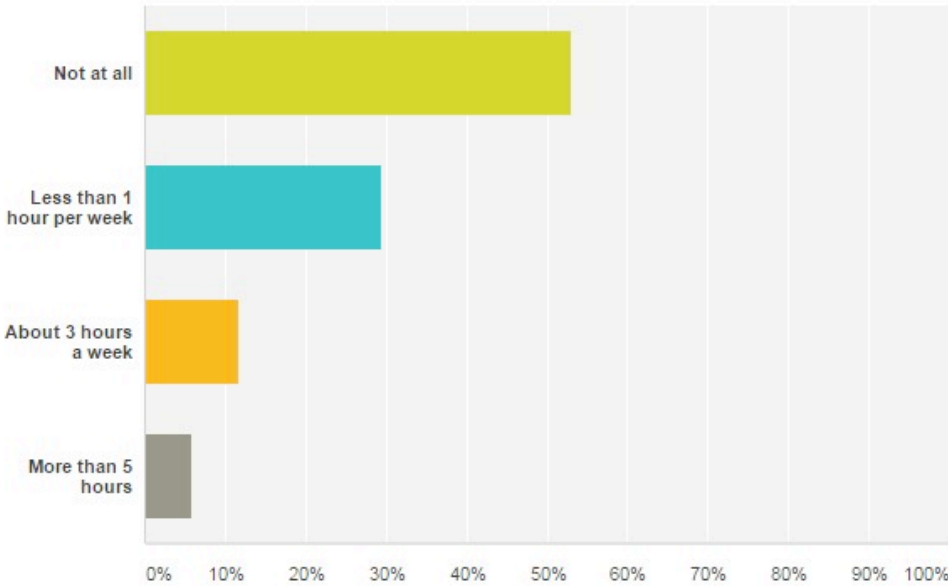


Figure 26: Survey Question 2

### How often do you listen to WICN Radio?

Answered: 17 Skipped: 0

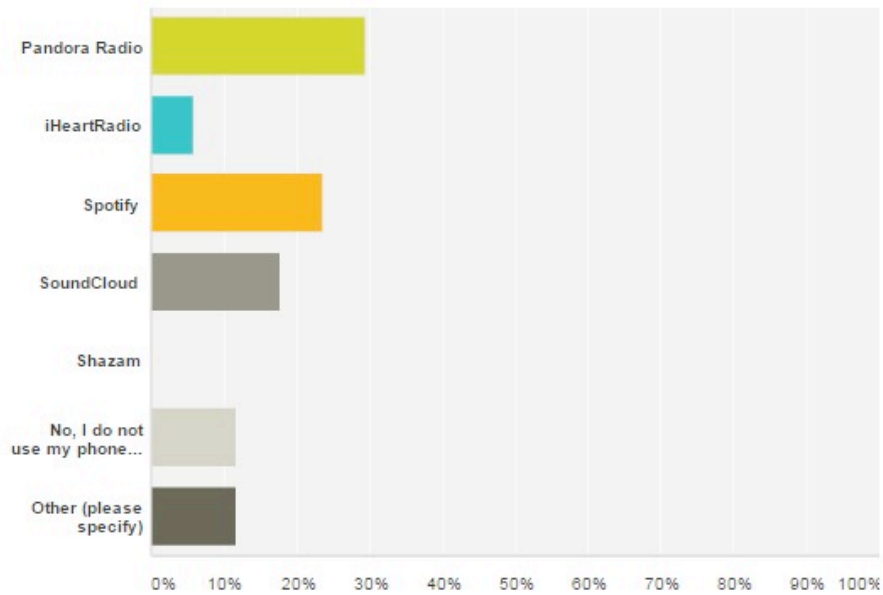


Answer Choices	Responses
Not at all	52.94% 9
Less than 1 hour per week	29.41% 5
About 3 hours a week	11.76% 2
More than 5 hours	5.88% 1
Total	17

Figure 27: Survey Question 3

### If you use smartphone to listen to music, what applications do you use?

Answered: 17 Skipped: 0



Answer Choices	Responses
▼ Pandora Radio	29.41% 5
▼ iHeartRadio	5.88% 1
▼ Spotify	23.53% 4
▼ SoundCloud	17.65% 3
▼ Shazam	0.00% 0
▼ No, I do not use my phone to listen to music	11.76% 2
▼ Other (please specify)	<a href="#">Responses</a> 11.76% 2
Total	17

Figure 28: Survey Question 4

### Rate the important of the following features (1 is the most important)

Answered: 17 Skipped: 0

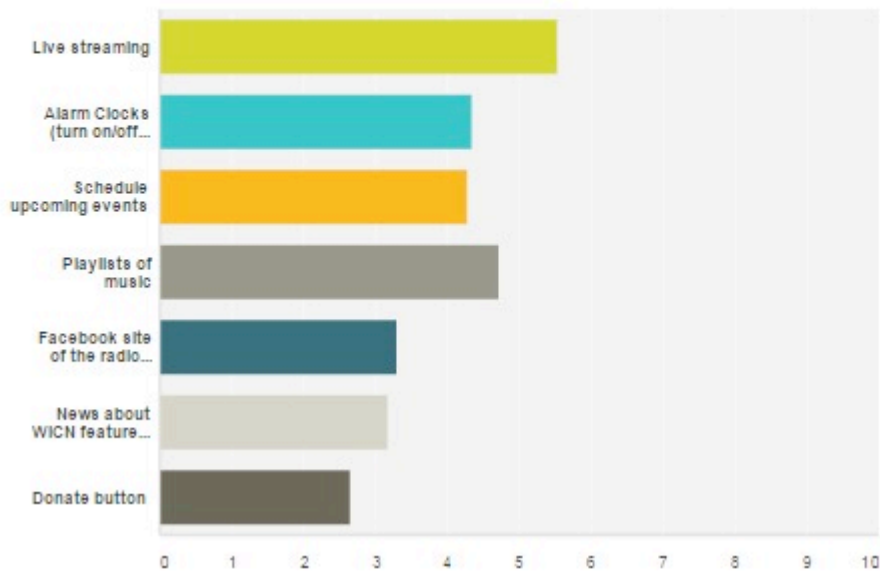
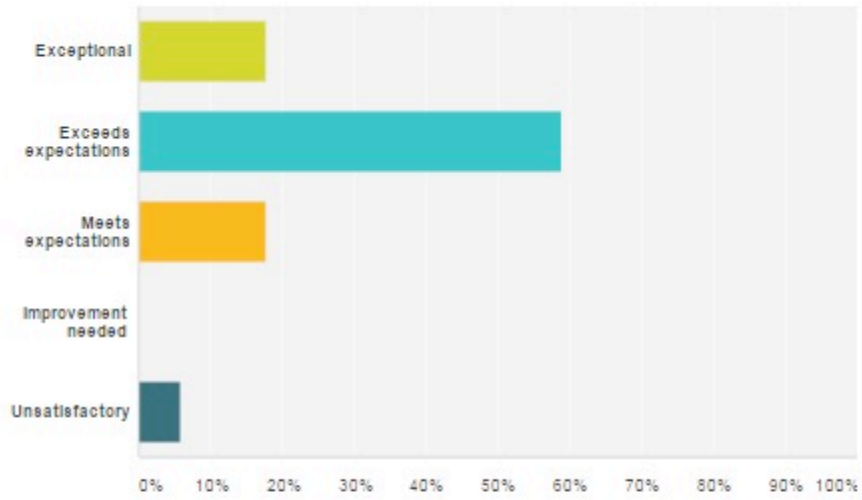


Figure 29: Survey Question 5

### What do you think about the overall quality of the application?

Answered: 17 Skipped: 0



Answer Choices	Responses
Exceptional	17.65% 3
Exceeds expectations	58.82% 10
Meets expectations	17.65% 3
Improvement needed	0.00% 0
Unsatisfactory	5.88% 1
Total	17

Figure 30: Survey Question 6

## Are there any features of the application that you felt were unnecessary? If so, please describe them.

Answered: 5 Skipped: 12

The screenshot displays the survey results for Question 7. At the top, there are three tabs: 'Responses (5)', 'Text Analysis', and 'My Categories'. Below the tabs is a yellow banner for a 'PRO FEATURE' with an 'Upgrade' button and a 'Learn more' link. Underneath is a search bar with 'Search responses' and a magnifying glass icon. The main content area shows 'Showing 5 responses' and a list of five individual responses, each with a timestamp and a 'View respondent's answers' link.

Response	Timestamp	Action
Not at all. Incredible application!!!	4/23/2015 11:44 PM	<a href="#">View respondent's answers</a>
no	4/23/2015 1:02 PM	<a href="#">View respondent's answers</a>
Donate Button	4/23/2015 1:00 PM	<a href="#">View respondent's answers</a>
play and stop buttons looked too wide. If I had the app I wouldn't know that that is what they were.	3/30/2015 11:07 AM	<a href="#">View respondent's answers</a>
HAVE ONLY SEEN VIDEO OF APPLICATION.	3/30/2015 10:27 AM	<a href="#">View respondent's answers</a>

Figure 31: Survey Question 7

## Do you have any additional questions or comments?

Answered: 4 Skipped: 13

The screenshot displays the survey results for Question 8. It features the same layout as Figure 31, including the 'Responses (4)' tab, the 'PRO FEATURE' banner, and the search bar. The main content area shows 'Showing 4 responses' and a list of four individual responses, each with a timestamp and a 'View respondent's answers' link.

Response	Timestamp	Action
ón	4/23/2015 1:02 PM	<a href="#">View respondent's answers</a>
Great work	4/23/2015 1:00 PM	<a href="#">View respondent's answers</a>
Looks very user friendly with a clean, clear layout.	3/30/2015 12:41 PM	<a href="#">View respondent's answers</a>
Need an android app too	3/30/2015 10:34 AM	<a href="#">View respondent's answers</a>

Figure 32: Survey Question 8

## **Appendix C : Application Submission to Apple IOS App Store**

After we had developed and reviewed our application, we were able to submit our application to App Store. The submission process can be broken down to four steps:

### **Step 1: Register for an Apple IOS Developer account and Certificates**

An Apple IOS Developer account can be registered on the on the Apple Developer site. After the account is registered, we can create two certificates. One of the certificates allows us to test the application on actual devices and the other to distribute.

Once the certificates are obtained, we can create two provisioning profiles (also on Apple Developer site). One is for application development, and the other one is for application distribution. For this step, App ID is required, which can be regards as a combination of our Development team ID and the name of our app. We also needs to choose the distribution certificate which we created on previous step, to include in our provisioning profile.

### **Step 2 : Creating Xcode archive**

After setting up certificates and provisioning profiles, a Xcode archive can be established. Xcode archive is a bundle, including all the code file of the application and all of it symbol file. This archive would be used to test the application and also submitted to the App Store.

### **Step 3 : Creating Itune Connect account and submitting app**

To submit and manage our application on App Store, an Itune Connect account is required. Itune Connect is a suite of web-based tools for managing content sold on the IOS App Store, Itune Store, Mac App Store and IBook Store. It manages applications, agreements, tax and banking information.. etc, that relate to our application.

Once we own an Itune Connect account, we can submit the application to Itune Connect. This step can be done by submitting the Xcode archive created in the previous step to Itune Connect using Xcode

We also need to create an Itune Connect App Record. This record includes any information that would appear on the store for the application, and all information needed to manage the app through distribution process, such as App ID, App version number, developer information, contact information, SKU... We need to provide these information to Itune Connect to create a record for our application.

After creating the App Record for our archive, we are able to submit the application to the App Store review process. This process usually took 2-4 weeks, during this time the Store managers would check for bugs and inconsistency in the application, and after this process, our application could be published on App Store.

### **Step 4 : Debugging**

After submitting our app to the App Store, the Review Process would send us response about bugs or violations to the App Store rule that our product may contain. Each time we received the response, we needed fix the bug and resubmit our product. Our application, after several modification, has been approved and is now on IOS App Store.

## References

- Apple Developer Library. (n.d.). Retrieved March 6, 2015, from <https://developer.apple.com/library/>
- Apple Inc. (n.d.). Concurrency Programming Guide. Retrieved from <https://developer.apple.com/library/ios/documentation/General/Conceptual/ConcurrencyProgrammingGuide/Introduction/Introduction.html>
- Apple Inc. (n.d.). Xcode [download].
- azamsharp. (n.d.). *Learning iOS Development Part 20 (UIView Containment)* [Video file].  
Retrieved from <https://www.youtube.com/watch?v=DzedmcFlm1c&spfreload=10>
- Buytaert, D. (n.d.). Drupal - Open Source CMS. Retrieved April 26, 2015, from <https://www.drupal.org/>
- Cengiz, I. (n.d.). ICViewPager. Retrieved from <https://github.com/iltercengiz/ICViewPager>
- Davidson, J. (n.d.). *SlideOutMenu* [zip]. Retrieved from <https://www.dropbox.com/s/uvhij5nhpb3tzm3/SlideOutMenu.zip>
- Davidson, J. (2014, July 14). *How to Make a Slide Out Menu with Objective-C in Xcode* [Video file]. Retrieved from <https://www.youtube.com/watch?v=5SUV1YY2yxQ>
- Diana, N, Yao-Hua Chung, K., & Li, X. (n.d.). *WICN Mobile Application Market Research* (GQP No. 105).
- Fulton, C. (2013, October 9). NSURLSession Tutorial. Retrieved from <http://www.raywenderlich.com/51127/nsurlsession-tutorial>
- How to access native Facebook app and Call app from an IOS app. (n.d.). Retrieved from [http://wiki.akosma.com/IPhone\\_URL\\_Schemes#Facebook](http://wiki.akosma.com/IPhone_URL_Schemes#Facebook)
- iHeartMedia Management Services, Inc. (n.d.). iHeartRadio [Mobile application software].



- Jimenez, E., & Crampton, S. (2014, May). *Smartphone radio application*. Retrieved from [https://www.wpi.edu/Pubs/E-project/Available/E-project-050614-113359/unrestricted/Smartphone\\_Radio\\_Application.pdf](https://www.wpi.edu/Pubs/E-project/Available/E-project-050614-113359/unrestricted/Smartphone_Radio_Application.pdf)
- Lluch, J. (n.d.). SWRevealViewController. Retrieved from <https://github.com/John-Lluch/SWRevealViewController>
- Nguyen, T. (n.d.). StreamingKit. Retrieved from <https://github.com/tumtumtum/StreamingKit>
- passing data from one view controller to another view controller. (n.d.). Retrieved from <http://stackoverflow.com/questions/19603034/passing-data-from-one-view-controller-to-another-view-controller>
- play audio from internet using AVAudioPlayer. (n.d.). Retrieved from <http://stackoverflow.com/questions/3635792/play-audio-from-internet-using-avaudioplayer>
- RadioFM. (n.d.). Radio FM [Mobile application software].
- Slacker Inc. (n.d.). Slacker Radio [Mobile application software].
- Theodoropoulos, G. (2014, March 9). Working With Background Fetch in iOS 7. Retrieved from <http://www.appcoda.com/ios7-background-fetch-programming/>
- TuneIn Inc. (n.d.). TuneIn Radio [Mobile application software].
- WICN Public Radio. (n.d.). WICN Radio [Mobile application software]. Retrieved from <https://play.google.com/store/apps/details?id=org.wicn.android&hl=en>