

---

# Net Music Player Mobile Application

---

**Weichen Wang**

B.Sc.(Hons) in Software Development

APRIL 16, 2018

**Final Year Project**

Advised by: Kevin O'Brien

Department of Computer Science and Applied Physics  
Galway-Mayo Institute of Technology (GMIT)



# Contents

<b>1</b>	<b>Abstract</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>6</b>
2.1	Idea Raised . . . . .	6
2.2	Music App Forecast . . . . .	7
<b>3</b>	<b>Methodology</b>	<b>9</b>
3.1	How i did my project . . . . .	9
3.1.1	Software development method . . . . .	9
3.1.2	Research methodology . . . . .	9
3.1.3	Determine the requirement of project . . . . .	10
3.1.4	Validation and testing . . . . .	11
3.2	Development Tools . . . . .	11
3.2.1	The hardware environment . . . . .	11
3.2.2	Software Environment . . . . .	11
3.2.3	Visual Studio Code Editor . . . . .	12
3.2.4	Google Firebase . . . . .	12
3.3	Platform Brief . . . . .	12
3.3.1	GitHub Profile . . . . .	12
<b>4</b>	<b>Technology Review</b>	<b>14</b>
4.1	Node.js . . . . .	14
4.1.1	Platform architecture . . . . .	15
4.2	Database . . . . .	16
4.2.1	Firebase Authentication . . . . .	17
4.3	Heroku . . . . .	21
4.4	Ionic3 . . . . .	22
4.5	Apache Cordova . . . . .	23
4.6	Angular.js . . . . .	25
4.7	JSON . . . . .	26
4.8	Package.json . . . . .	27

<b>5</b>	<b>System Design</b>	<b>29</b>
5.1	Language Choose . . . . .	29
5.1.1	Node.js Features . . . . .	29
5.1.2	JavaScript Features . . . . .	31
5.2	Heroku Cloud Hosted . . . . .	32
5.3	Architecture . . . . .	32
5.3.1	System Architecture . . . . .	32
5.3.2	Main Process . . . . .	36
5.4	System Functions Review . . . . .	38
5.5	Security System . . . . .	46
5.5.1	User Security . . . . .	46
5.6	Appearance Design . . . . .	46
5.6.1	MainPage(Welcome) Design . . . . .	46
5.6.2	Home Page Design . . . . .	47
<b>6</b>	<b>System Evaluation</b>	<b>57</b>
6.1	Advantages . . . . .	57
6.2	Limitation . . . . .	58
6.3	Future Development and Optimization . . . . .	60
<b>7</b>	<b>Conclusion</b>	<b>61</b>
<b>8</b>	<b>Appendix</b>	<b>63</b>

# Chapter 1

## Abstract

In this project, I used Node.js, Ionic3 as the framework of the system, using Google Firebase as a mobile platform for building mobile applications, providing real-time data storage and synchronization, user authentication and other functions, using Heroku as cloud platform as a service (PaaS), to provide the deployment of the application, use GitHub software version of the project to control the source code hosting services.

Music is a sound symbol that expresses people's thoughts and feelings. It is one of the carriers of people's thoughts. Music has purpose and connotation, which implies the author's life experience and thoughts. For me, I am a person who loves music very much. Everyday life is inseparable from music. I often need to use music to eliminate many of the times I studied abroad for myself. This allows me to maintain a good attitude and Mood to face a lot of daily learning. This is why I chose to develop a music player application for a mobile platform.

This project design leverages open source, cross-platform Node.js runtime environment, JavaScript programming language, Ionic3 with Angular4 framework, Google Firebase database, Heroku cloud deployment platform, and Visual Studio Code text editor developed by Microsoft to write the player. And test execution. At the same time, the detailed system design process and some interface diagrams are given. The paper also discusses the problems and solutions in the development process. The function of the music player has a complete login function, including: Login, Registration, Forgot Password, and Log Out. Page Navigation Design, Page Jump Design, permission limits for login functions, Alert pop-ups, Toast message notifications, music play, pauses, show music cover artwork. The waterfall model was used during project development. In the first phase, the project was first comprehensively and carefully analyzed, and the project schedule was accurately made to clarify the tasks in each phase. The second stage is to

code the project in modules and keep testing. The third stage is to conduct a comprehensive test and system integration test on the project.

**Key Words:** Music Player Project, Node.js, Ionic3, Angular4, Google Firebase, Heroku, GitHub.

**Authors** Weichen Wang B.Sc.(Hons) in Software Development

# Chapter 2

## Introduction

### 2.1 Idea Raised

This is a Node.js Project for the Final Year Project named Net Music Player. The project uses the Ionic3 framework, based on Google's Angular framework, corresponding to Angular4. This idea stems from my personal experience. There are many brands of music player applications in the Chinese market, with a wide range of features and functions. But when I first came to Ireland, the songs that were originally in my music player could not be played and could not be downloaded. The reason for this is due to the geographical copyright restrictions of the song itself, which led to the failure of several of my music player applications. This has always been an annoyance for me. So I came up with the idea that if I hear the song I really like, I download it locally, and then add it to my own developed music player to become a playlist that belongs only to me, and I'm listening to songs offline. So no matter where I go, I can hear the songs I want to hear anytime, anywhere.

However, for the Spotify music player, it really brought me a lot of good experience, whether it is a song resource, or an application interface, but I think there is a point that Spotify can not select specified songs to play on the mobile platform, and can not. In order to loop, think of it, you have to recharge your Spotify account so that Spotify's premium package allows you to download tracks, albums, and playlists without a data connection[1]. For me who is just a student, this is not good news.

On the other hand, for the direction of the user market, listening to music has become an indispensable entertainment activity in today's society. According to the data obtained from some surveys, it has been found that

in recent years, the popularity of smart phone users for the application of larger music players has declined significantly, while the relatively compact and simple music player applications have been downloaded and used by more and more users. IOS system software has the characteristics of beauty, simplicity and easy operation. Based on this information, developing a small IOS music player will be easy for users to like. It is precisely because of this that I decided to use Ionic3 as a framework for developing applications. Ionic's goal is also to develop native APPs. Most of the native APIs can be manipulated on Ionic because it is based on Cordova (A basic project that can manipulate native APIs using JavaScript). And Ionic is a framework for the future, in other words, it is for new mobile phones. Although this is the first time I have used the Ionic3 framework to develop mobile applications, I personally like the simplicity of the Ionic framework's native API and design, and I hope to learn how to use the Ionic framework to develop mobile application through this Final Year project, improve understanding and learning of mobile application development.

## 2.2 Music App Forecast

Digital music has gradually shifted from the PC-side war to the mobile Internet. Since it joined the cloud technology, the information function of the music app has also been perfect. Nowadays, the development of technology becomes more and more mature, and the music app has also evolved into one of the main content of mobile terminals. The ecosystem of iOS mobile music apps has expanded notably in the past five years. Many of the apps have been created by makers of desktop music production software, and most have been aimed at prosumers. But that may be changing. Regardless of whether or not it is believed that the mobile app APP has become the mainstream method of listening to songs, the demand for music app software is also very large. Everyone knows that the music software development function is of course listening to music, but the people used and the environments used are often different. Simply speaking, many people use the mobile music client to be bored, and the development of future music apps can be based on different environments to create different products. At the same time, many users of music software development are interested in music app development and related functions. In the future, developers can work hard to provide more interesting and fun functions for users to experience. There are have pictures below see figure 2.1 and figure 2.2:

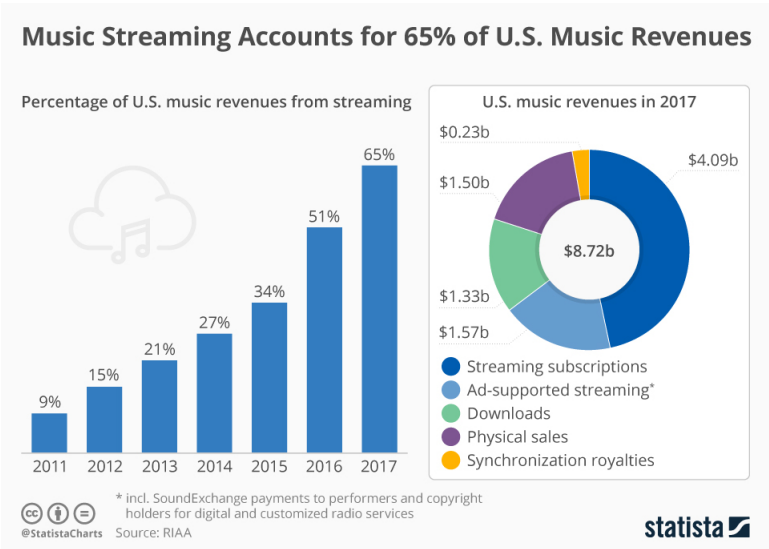


Figure 2.1: Digital music revenues in U.S

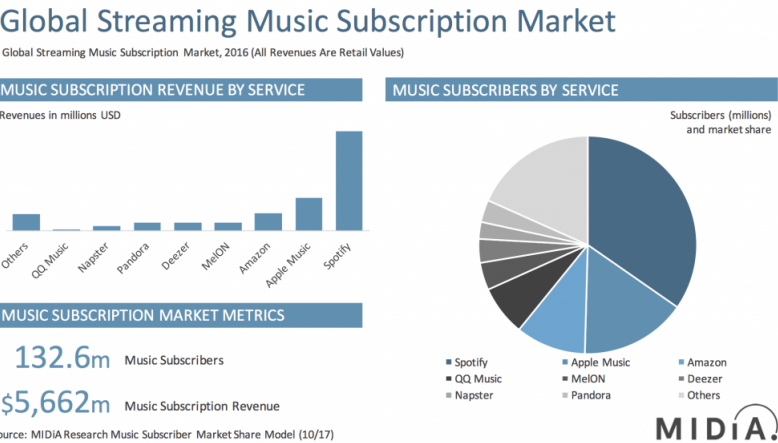


Figure 2.2: Global streaming music subscription market



# Chapter 3

## Methodology

### 3.1 How i did my project

#### 3.1.1 Software development method

This project is designed to use open source, cross-platform Node.js runtime environment, JavaScript programming language, Ionic3 with Angular4 framework, Google Firebase database, Heroku cloud deployment platform and Microsoft developed Visual Studio code text editor to write player and test execution.

#### 3.1.2 Research methodology

First of all, I did a project requirements analysis to plan the expected goals in the future development process, and then use Rapid Prototype Model to create a template, because Rapid Prototyping decreases development time by allowing corrections to a product to be made early in the process[2] , With a rapid prototype that is based on prototyping and iterative development, you can review and test your design to catch problems early on and fix them before you have invested copious amounts of time, energy and resources. Avoid the need for changes by avoiding introducing the non-essentials into the mix. But this only front-end interface design and layout. The interface design and layout of the front-end is based on the UI Component Document of Ionic's official website. The back-end function design of the project is mostly derived from the Ionic official website's native components and APIs. In general, the entire project development process is based on Ionic's own native components and APIs.

### 3.1.3 Determine the requirement of project

We all know that defining IT project requirements is a crucial task when initiating new projects. Therefore, I determined the requirements of the project based on a few things: First, a good music player needs a UI design that allows the user to feel relaxed and relieved, so the simple interface design is my first choice.

Second, a complete login registration system is required. A good database is provided to store the encrypted user's registration information, so as to effectively protect the user's account information and password.

Third, humanized page navigation design, soothing page jump will not allow users to feel uncomfortable when the page jumps back and forth. Proper guidance of user's use also enables users to maintain a comfortable and stress-free use status.

Fourth, the most basic ones allow certain music to play normally, stop, and you can also see the cover thumbnail of the song and the artist's name.

I used a lot of native UI components in the ionic3 framework when designing the front-end UI, allowing me to quickly build a basic project model, which provided a lot of time for the development of major functions. The design of the login registration system includes registration, login, and forget password modules. The functions of these modules are based on the authentication module in the official Google Firebase documentation. In the design of page navigation design and music play, stop, and other functions are based on Ionic3 native API components and integration plug-ins.

- Meeting: Regarding project meetings, since I was a single person developing the Final Year Project, I did not have any meetings on the team project discussions. However, in the process of design and development, I will keep meeting with my project supervisors regularly. In the process of meeting, I will present to supervisors my weekly progress and achievements in the development of the project and propose ideas and plans to be implemented in the coming week. And supervisors will make appropriate suggestions and supplements on what I have stated.
- Frequency: In the process of designing and developing the project, I maintain a weekly meeting with my project supervisor to discuss the meeting. The meeting time is determined every Thursday from 12 noon to 1 pm, and the meeting place is the campus-wide room PF18.
- Structure: Regarding the project structure, the entire project is a Node.js project, using Ionic3 native framework, so the following structures exist in the project:

1. `./src/index.html`: is the main entry point for the app, though its purpose is to set up script and CSS includes and bootstrap, or start running.
2. `./src/`: Inside of the src directory we find our code. This is where most of the work for an Ionic app will take place. When we run `ionic serve`, our code inside of `src/` is transpiled into the correct Javascript version that the browser understands. `src/app/app.module.ts` is the entry point for our app.
3. `./src/app/app.html`: the main template for the app in `src/app/app.html`.

### 3.1.4 Validation and testing

For verification and testing, I used the `ionic3` terminal command to perform phased verification and testing of the project on the Visual Studio Code terminal command line, and also integrated the functionality of the project with the built-in developer console in Google Chrome. Verification and testing, view the output of data, modify the code error.

## 3.2 Development Tools

### 3.2.1 The hardware environment

- Laptop configuration:

CPU: 1.6 GHz Intel Core i5.

RAM: 8 GB 1600 MHz DDR3.

Graphic: Intel HD Graphics 6000 1536 MB.

Operation System: OS X El Capitan.

### 3.2.2 Software Environment

- Development Tool: Visual Studio Code (version 1.22.1).
- Browser: Google Chrome.
- Development Environment: Node.js.
- Database: Firebase.

### 3.2.3 Visual Studio Code Editor

- Download Link: <https://code.visualstudio.com/>

Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences. It is free and open-source.[1]

Visual Studio Code is based on Electron, a framework which is used to deploy Node.js applications for the desktop running on the Blink layout engine. Although it uses the Electron framework, the software does not use Atom and instead employs the same editor component (codenamed "Monaco") used in Visual Studio Team Services (formerly called Visual Studio Online).[1]

In the Stack Overflow 2018 Developer Survey, Visual Studio code was ranked the most popular developer environment tool, with 34.9% of 75398 respondents claiming to use it.[1]

### 3.2.4 Google Firebase

- Official Website Link: <https://firebase.google.com/>

Firebase is a mobile and web application development platform developed by Firebase, Inc. in 2011, then acquired by Google in 2014.

## 3.3 Platform Brief

### 3.3.1 GitHub Profile

GitHub (originally named Logical Awesome LLC) is a Web-based hosting service that uses git for version control. It is mainly used for computer code. It provides all of Git's distributed version control and source code management (SCM) capabilities and adds its own features. It provides access control and various collaboration features for each project, such as bug tracking, feature requests, task management, and wikis.

GitHub Personal Repository and Wiki pages You can edit these pages using the git repository. As a distributed version control system, Git does not have a conceptual main library. Each warehouse copy can be used independently. Any differences between these warehouses can be combined together.

GitHub can host a variety of gits and provide a unified web interface. However, providing different service code management websites such as Source Forge and Google Code, GitHub's unique branch project management is very simple and highly interoperable. If you want to contribute code to your project it is an easy task: first on the project site for the "fork" button, then check the code and add the changes to a separate code base, and finally through the built-in "pull request" mechanism applicable to the project director code.

# Chapter 4

## Technology Review

### 4.1 Node.js

Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML and run client-side by a JavaScript engine in the user's web browser. Node.js lets developers use JavaScript for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser.

Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

Node.js allows you to create web servers and web tools using JavaScript and a set of "modules" that handle various core functions. Provides modules for file system I/O, network (DNS, HTTP, TCP, TLS/SSL or UDP), binary data (buffer), cryptographic functions, data streaming and other core functions. The Node.js module uses APIs designed to reduce the complexity of writing server applications.

The Node.js application can run on Linux, macOS, Microsoft Windows, Non-Stop and Unix servers. Alternatively, you can write them using CoffeeScript (JavaScript substitution), Dart or TypeScript (strongly typed JavaScript), or any other language that can be compiled to JavaScript.

Node.js is primarily used to build network programs such as Web servers. The biggest difference between Node.js and PHP is that most functions in PHP block until completion (commands execute only after previous commands finish), while Node.js functions are non-blocking (commands execute concurrently or even in parallel, and use callbacks to signal completion or failure).

Working principle see figure 4.1 and figure 4.2:

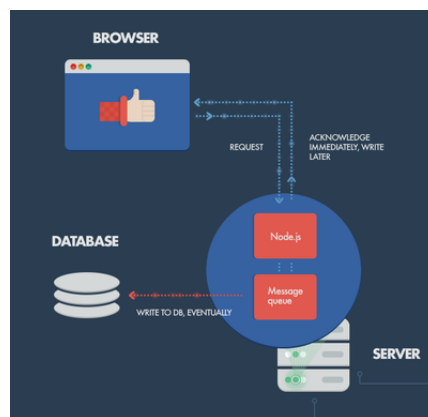


Figure 4.1: Node.js Working Principle

#### 4.1.1 Platform architecture

Node.js brings event-driven programming to the Web server to support the development of fast Web servers in JavaScript. Developers can create highly scalable servers using a simplified model of event-driven programming without the use of threads, which use callbacks to indicate task completion. Node.js combines the ease of use of scripting languages (JavaScript) with the power of Unix network programming.

Node.js was built on the Google V8 JavaScript engine since it was open-sourced under the BSD license, extremely fast, and proficient with internet fundamentals such as HTTP, DNS, TCP. Also, JavaScript was a well-known language, making Node.js immediately accessible to the entire web development community.

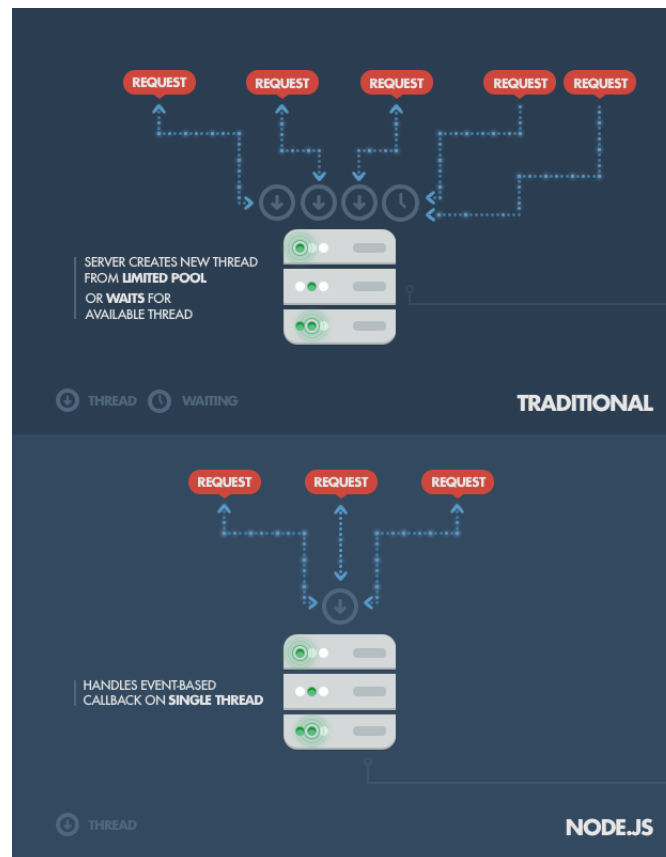


Figure 4.2: Node.js Working Principle

## 4.2 Database

Firebase is an app cloud development platform that supports Android, iOS, and webpages at the same time. It helps app developers quickly set up back-end services in the cloud, provides instant data bases, effectively shortens app development time, and helps developers focus on front-end optimization. Firebase provides the following important service features:

- **Firebase Analytics:** The powerful unlimited data analysis function is the core product of Firebase. Store and sync data between users and devices - at global scale - using a cloud-hosted, noSQL database. Cloud Firestore gives you live synchronization and offline support along with efficient data queries. Its integration with other Firebase products enables you to build truly serverless apps.[1] Currently, it is completely free to use. There are 500 preset event types, provide unlimited event



records, and can customize key events and user attributes. Visualized The Dashboard allows users to see at a glance the performance analysis of all the most important user behaviors and marketing activities. see figure 4.3:

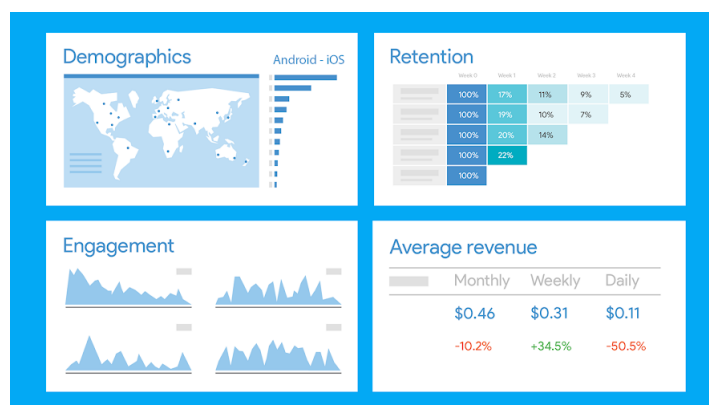


Figure 4.3: Google Analytics for Firebase

- **Develop:** Use Firebase to create high-quality apps that are secure and efficient. The back-end services provided include a Realtime database, a simple Authentication, Cloud Messaging, Hosting, and Archives. Storage, Remote Config, Android Test Lab for Android, and Crash Reporting allow developers to easily test apps on different OSs and hardware devices Under the performance, quickly find problems and ensure the quality of the app.
- **Grow:** Firebase allows developers to easily set up messaging notifications, app indexing, dynamic links, email and sms invitations (Invites), and deep integration with Google AdWords. Information allows app developers to provide more accurate advertising or communication messages to target target groups.
- **Earn:** Integrate with its own product AdMob, add advertising network, easily put advertising space, and a variety of advertising formats, as well as a more clear tracking mechanism to help app developers easily make money. see figure 4.4:

### 4.2.1 Firebase Authentication

Firebase Authentication aims to make building secure authentication systems easy [3], while improving the sign-in and onboarding experience for end users.

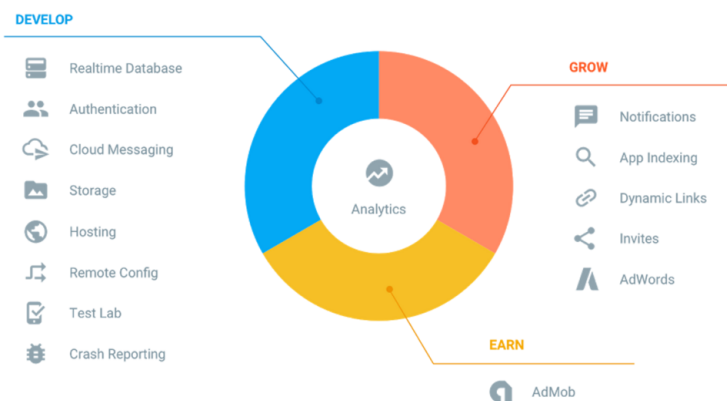


Figure 4.4: Google Firebase Three Major Functions

It provides an end-to-end identity solution, supporting email and password accounts, phone auth, and Google, Twitter, Facebook, and GitHub login, and more. see figure 4.5:



Figure 4.5: Google Firebase Authentication

- How does it work: To sign a user into your app, you first get authentication credentials from the user. These credentials can be the user's email address and password, or an OAuth token from a federated identity provider. Then, you pass these credentials to the Firebase Authentication SDK. Our backend services will then verify those credentials and return a response to the client.

After a successful sign in, you can access the user's basic profile information, and you can control the user's access to data stored in other Firebase products. You can also use the provided authentication token to verify the identity of users in your own backend services[3].

- Flexible, drop-in UI: FirebaseUI provides a customizable, open source, drop-in auth solution that handles the UI flows for signing in users. The FirebaseUI Auth component implements best practices for authentication on mobile devices and websites, which can maximize sign-in and sign-up conversion for your app[4]. see figure 4.6:

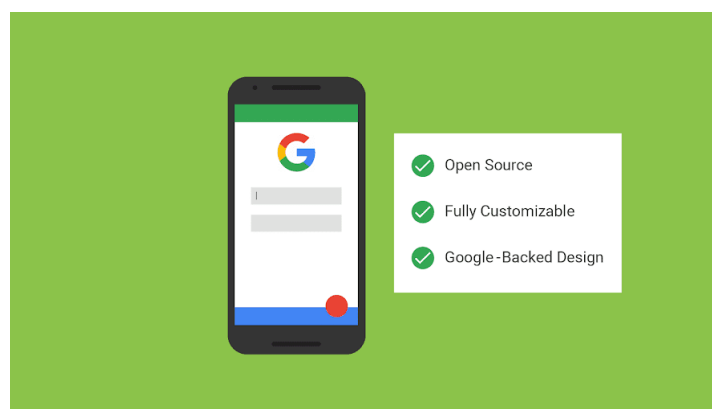


Figure 4.6: Google Firebase Authentication

- Comprehensive security: Built by the same team that developed Google Sign-in, Smart Lock and Chrome Password Manager, Firebase security and applies Google's internal expertise of managing one of the largest account databases in the world. see figure 4.7:
- Fast implementation: It can take months to set up your own auth system, and it requires an engineering team to maintain that system into in the future. Set up the entire authentication system of your app in under 10 lines of code, even handling complex cases like account merging. see figure 4.8:
- Setting Authentication Code:  
Add Firebase to your web app. Copy the code snippet below and paste it at the bottom of your HTML code before other script tags.

```
<script src="https://www.gstatic.com/firebasejs/4.12.1/firebase.js">  
</script>
```

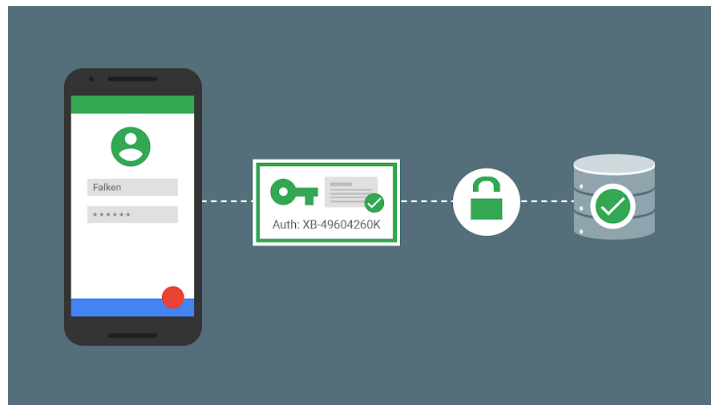


Figure 4.7: Google Firebase Authentication Security

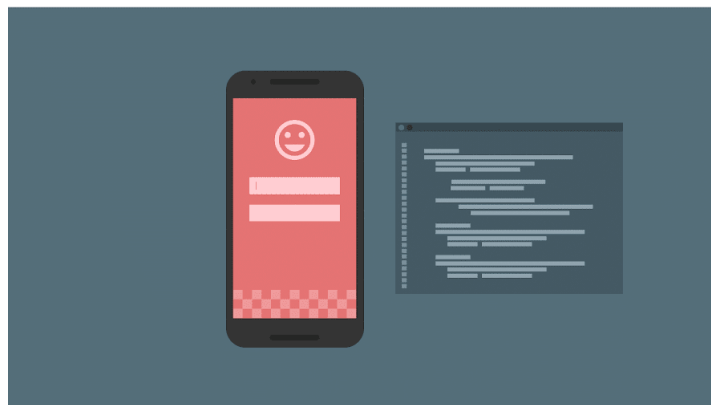


Figure 4.8: Google Firebase Authentication Fast

```

<script>
  // Initialize Firebase
  var config = {
    apiKey: "AIzaSyCWyA_drTwdJ-_N80A2vajNXfvKxyzh9ks",
    authDomain: "netmusic-c9a02.firebaseio.com",
    databaseURL: "https://netmusic-c9a02.firebaseio.com",
    projectId: "netmusic-c9a02",
    storageBucket: "netmusic-c9a02.appspot.com",
    messagingSenderId: "971450191473"
  };
  firebase.initializeApp(config);
</script>

```

This is our updated app.module.js file. see figure 4.9:  
 After you add the code, you need to go into authentication and enable

```

25 import { StatusBar } from '@ionic-native/status-bar';
26 import { SplashScreen } from '@ionic-native/splash-screen';
27 import { StreamingMedia } from '@ionic-native/streaming-media';
28 // import { Media } from '@ionic-native/media';
29 //Add firebaseAuth
30 //Ref: https://console.firebase.google.com/project/netmusic-c9a02/overview?hl=zh-cn
31 const firebaseAuth = { //firebase登陆方法配置
32   apiKey: "AIzaSyCwyA_drTwdJ-_N80A2vajNXfvKxyzh9ks",
33   authDomain: "netmusic-c9a02.firebaseio.com",
34   databaseURL: "https://netmusic-c9a02.firebaseio.com",
35   projectId: "netmusic-c9a02",
36   storageBucket: "netmusic-c9a02.appspot.com",
37   messagingSenderId: "971450191473"
38 };
39
40
41 @NgModule({ //被@NgModule修饰的class都被成为ng Module
42   declarations: [ //声明当前模块需要的指定 组件信息
43     MyApp

```

Figure 4.9: Setting Authentication Code in app.module.js file

the provider by clicking and enabling the desired feature. see figure 4.10:

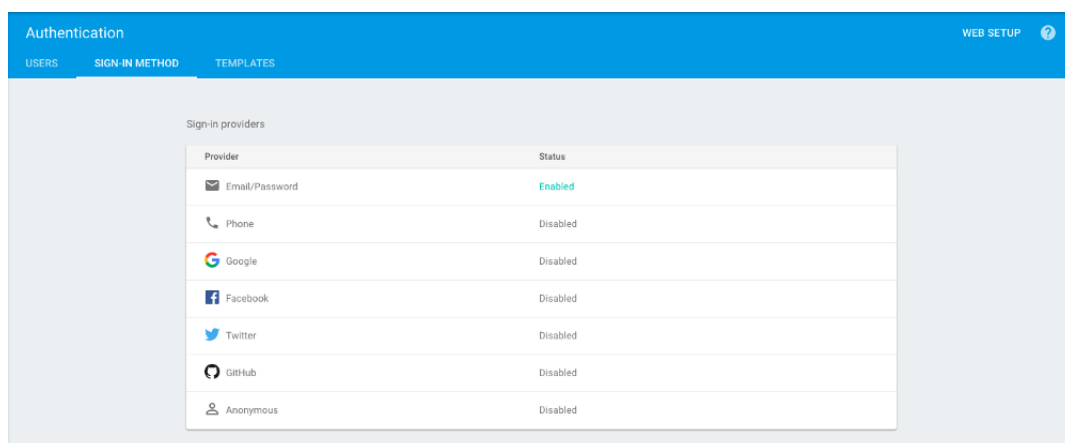


Figure 4.10: Google Firebase Authentication Option

## 4.3 Heroku

Heroku is a cloud platform as a service (PaaS) supporting several programming languages that is used as a web application deployment model. has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go.

Deploying applications: Application development on Heroku is primarily done through git. The application gets a new git remote typically named as Heroku along with its local git repository where the application was made. Hence to deploy heroku application is similar to using the git push command.

There are many other ways of deploying applications too. For example, developers can enable GitHub integration so that each new pull request is associated with its own new application, which enables all sorts of continuous integration scenarios. Dropbox Sync lets developers deploy the contents of Dropbox folders to Heroku, or the Heroku API can be used to build and release apps. Deployment then, is about moving the application from a local system to Heroku.

## 4.4 Ionic3

Ionic is a complete open-source SDK for hybrid mobile app development. The more recent releases, known as Ionic 3 or simply "Ionic", are built on Angular. Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML5, and Sass. Apps can be built with these Web technologies and then distributed through native app stores to be installed on devices by leveraging Cordova.

Services and features: Ionic provides all the functionality which can be found in native mobile development SDKs. Users can build their apps, customize them for Android or iOS, and deploy through Cordova. Ionic includes mobile components, typography, interactive paradigms, and an extensible base theme.

Besides the SDK, Ionic also provides services that developers can use to enable features, such as push notifications, A/B testing, analytics, code deploys, and automated builds.

Ionic also provides a powerful command-line interface (CLI), so developers can get started with creating a project with a simple command. The CLI also allows developers to add Cordova plugins and additional front-end packages [5], enable push notifications, generate app Icons and Splash screens, and build native binaries.

Supported platforms: Ionic is focused on building for modern Web standards and for modern mobile devices. For Android, Ionic supports Android 4.1 and up. For iOS, Ionic supports iOS 7 and up. Ionic 2 supports the Universal Windows Platform for building Windows 10 apps. Ionic Framework, pow-

ered by Angular.js, supports BlackBerry 10 apps.

Installation: Ionic is an npm module and requires Node.js.

- Install Ionic Code: `npm install -g ionic`  
First, install Node.js. Then, install the latest Ionic command-line tools in your terminal. Follow the Android and iOS platform guides to install required tools for development.
- Start an App: `ionic start myApp tabs`
- Run your App: `cd myApp`  
`ionic serve`  
Most applications can be built in browsers using ion services. When you are ready to deploy the application to a real device, you can review the deployment guide.

## 4.5 Apache Cordova

Apache Cordova is an open-source mobile development framework[6]. It allows you to use standard web technologies - HTML5, CSS3, and JavaScript for cross-platform development. Applications execute within wrappers targeted to each platform, and rely on standards-compliant API bindings to access each device's capabilities such as sensors, data, network status, etc. Plugins are an integral part of the cordova ecosystem. They provide an interface for Cordova and native components to communicate with each other and bindings to standard device APIs. This enables you to invoke native code from JavaScript.

Since the Cordova plugin cannot use Angular's dependency injection[7], it can only be invoked by accessing global variables and methods, which is very inconvenient for code management. So the Ionic development team encapsulates a series of open source Cordova plugins based on the Cordova API[7], named Ionic Native's set of plugins. In our new Ionic project, you can see several Ionic Native plug-ins that you can access by opening `package.json`.

Cordova native plug-in:

Here is an example to get a mobile photo album, enter the following code to install the imagepicker plugin:

\$ ionic cordova plugin add add cordova-plugin-telerik-imagepicker  
Afterwards[7], on the page that needs to be used, the following method can be used to call the mobile photo album:

```
showAlbum() {  
  this.platform.ready().then(() => {  
    (<any>window).imagePicker.getPictures(photo => {  
      console.log(photo[0]);  
    });  
  });  
};
```

Ionic Native:

Unlike native Cordova plug-ins, Ionic Native requires two steps to install. Take the ActionSheet plugin as an example:

\$ ionic cordova plugin add cordova-plugin-actionsheet

\$ npm install --save @ionic-native/action-sheet

The first step is identical to the previous installation of the plug-in. The second step is to install the Ionic package code.

Plug-in restrictions: Looking for good plugins is very helpful for development work, but not all plugins can be used on all platforms, and not all methods can be used. When looking for plugins at Ionic Native, you first need to look at the supported platforms. see figure 4.11:

## Supported platforms

- Android
- Browser
- iOS
- Windows
- Windows Phone 8

Figure 4.11: Supported Platforms

If you are using a Cordova plugin that does not support your project platform, you can see the prompt output in the developer console in your browser. see figure 4.12:



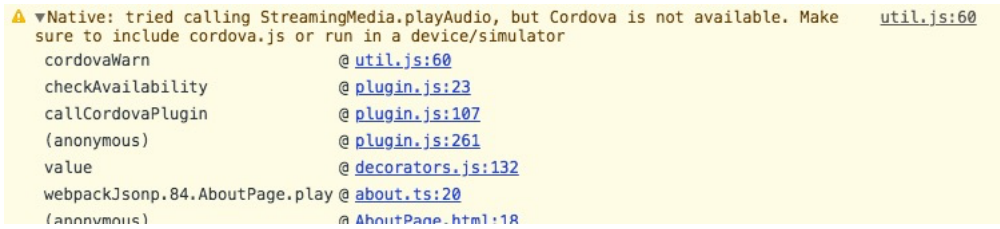


Figure 4.12: Supported Platforms Error

## 4.6 Angular.js

AngularJS (commonly referred to as "Angular.js" or "AngularJS 1.X") is a JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications[8]. The JavaScript components complement Apache Cordova[6], a framework used for developing cross-platform mobile apps. It aims to simplify both the development and the testing of such applications by providing a framework for client-side model–view–controller (MVC) and model–view–viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications.

The AngularJS framework works by first reading the HTML page[8], which has additional custom tag attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources. Some of the most commonly used directives are:

- ng-bind: Sets the text of a DOM element to the value of an expression. For example,

```
<span ng-bind="name"></span>
```

displays the value of 'name' inside the span element. Any change to the variable 'name' in the application's scope reflect instantly in the DOM.

- ng-class: Conditionally apply a class, depending on the value of a boolean expression.

- `ng-model`: Similar to `ng-bind`, but establishes a two-way data binding between the view and the scope.

Ionic and Angular:

The first thing to be clear is that Ionic is a derivative of Angular. Angular is a standalone JS library that can be used independently for developing applications just like jQuery. Ionic only extends Angular and uses Angular to implement many components that are compatible with mobile applications. And built a very good style library, is the most successful application of Angular samples. Even if Ionic is not used, Angular can be used with any style library such as Bootstrap, Foundation, etc. to get the desired page effect.

Ionic/Angular and Cordova plug-ins:

## 4.7 JSON

JavaScript Object Notation or JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser–server communication, including as a replacement for XML in some AJAX-style systems. JSON filenames use the extension `.json`.

Example of JSON file to describing a person:

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
```

```
    },  
    {  
      "type": "office",  
      "number": "646 555-4567"  
    },  
    {  
      "type": "mobile",  
      "number": "123 456-7890"  
    }  
  ],  
  "children": [],  
  "spouse": null  
}
```

## 4.8 Package.json

All npm packages contain a file, usually in the project root, called `package.json` - this file holds various metadata relevant to the project[9]. This file is used to give information to npm that allows it to identify the project as well as handle the project's dependencies. It can also contain other metadata such as a project description, the version of the project in a particular distribution, license information, even configuration data - all of which can be vital to both npm and to the end users of the package. The `package.json` file is normally located at the root directory of a Node.js project[9].

Node itself is only aware of two fields in the `package.json`:

```
{  
  "name" : "barebones",  
  "version" : "0.0.0",  
}
```

The most important of `package.json` is the name and version fields[9]. These are actually required, and without them, the software package will not be installed. The name and version together form an identifier that is assumed to be completely unique. Changes to the package should change as the version changes.

`Package.json` is a json file that describes the dependencies. When we are developing, we want to add some necessary packages to the project. Simply execute `'npm install <name> --save'` and we will add the package name and corresponding version to `package.json`[9], and then only need to be in

package. In the directory where json is located, open the command line and execute `npm install` or `cnpm install`, and it will automatically parse `package.json` and install the dependent packages mentioned in the command line. The project will run normally[9].

# Chapter 5

## System Design

### 5.1 Language Choose

#### 5.1.1 Node.js Features

Node.js uses the Google V8 JavaScript engine to execute code, and a large percentage of the basic modules are written in JavaScript. Node.js contains a built-in library to allow applications to act as a Web server without software such as Apache HTTP Server or IIS. NPM is the pre-installed package manager for the Node.js server platform. It is used to install Node.js programs from the npm registry. The package manager allows publishing and sharing of open-source Node.js libraries by the community, and simplifies installation, updating and un-installation of libraries.

##### 1. Asynchronous I/O:

All APIs of Node.js library are asynchronous i.e. non-blocking. So a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call. It is also a reason that it is very fast.

##### 2. Single Threaded:

Node maintains the single-threaded nature of JS in the browser. The biggest benefit of single-threading is that it doesn't have to worry about synchronization issues like multi-threaded programming, no deadlocks, and no thread context switching overhead. Single-threaded also has its weaknesses, mainly in three aspects: can not use multi-core CPU, the error will cause the entire application to exit, the robustness of the application is worth studying, and

a large number of calculations will occupy the CPU can not continue to call asynchronous I / O.

In order to solve the above problem, Node adopts the same idea as HTML5 Web Workers, using `child_process` to solve the problem of large amounts of computation in a single thread. By distributing the calculations to various child processes, a large number of calculations can be decomposed, and then the results are passed through the event messages between the processes.

### 3. Cross-platform:

Node is cross-platform, that is the same set of JS code can be deployed and run on Windows, Linux, OSX and other platforms. This is mainly due to Node building a platform layer architecture `libuv` between the operating system and the Node upper module system. see figure 5.1:

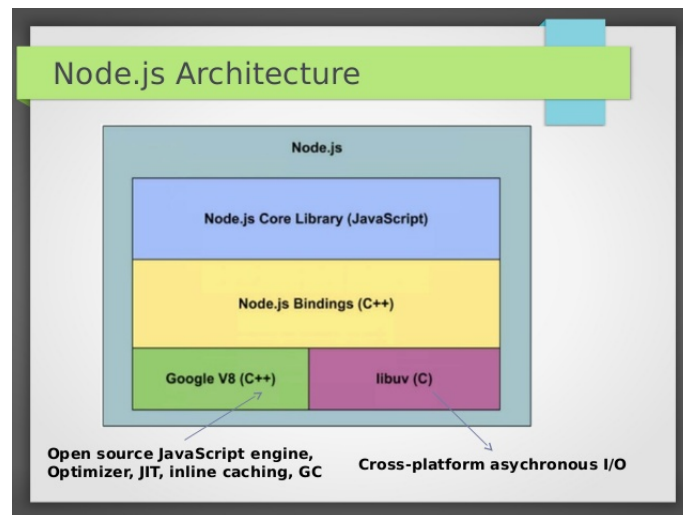


Figure 5.1: Node.js Cross-Platform Architecture

### 4. Highly Scalable:

Node.js is highly scalable because event mechanism helps the server to respond in a non-blocking way.

### 5. No Buffering:

Node.js cuts down the overall processing time while uploading audio and video files. Node.js applications never buffer any data. These applications simply output the data in chunks.

#### 6. Open source:

Node.js has an open source community which has produced many excellent modules to add additional capabilities to Node.js applications.

#### 7. License:

Node.js is released under the MIT license.

### 5.1.2 JavaScript Features

Node.js is based on JavaScript design.

#### 1. Browser Support:

When you want to access Flash content, you need to install the Flash plugin in your browser. But with JavaScript, you don't have to use any plugins at all. This is because all browsers accept JavaScript as a scripting language and provide integrated support for it. All you have to do is handle the task of relying on different browsers for the DOM (Document Object Model).

#### 2. Used on the Client and Server side:

Because JavaScript can access the browser's Document object model, you can change the structure of the web page at runtime. For this reason, JavaScript can be used to add different effects to web pages. On the other hand, JavaScript can also be used on the server side. For example, in the popular open source enterprise content management system Alfresco, JavaScript is used to create Web scripts. This makes it easy to add custom tasks to the outdoors.

#### 3. Functional Programming Language:

In JavaScript, functions can be assigned to variables just like any other data type. Not only that, the function can also accept another function as a parameter, and it can also return a function. You can also have features without a name.

4. Object-Oriented: JavaScript is an object-based and event-driven scripting language. It is also good at security capabilities. JavaScript is used to develop INTERNET client and server applications. It can be easily embedded in HTML files and used by the built-in interpreter in the browser. Implement JavaScript. Using JavaScript, the browser can respond to the needs of user

events without having to return data over the network. In this way, the user's data can be processed directly by the client application. It can also easily interact with web clients and make the web alive. For example: When you visit a website, JavaScript can immediately reply by moving the mouse or clicking a button or other component on this page or dragging a window.

## 5.2 Heroku Cloud Hosted

Heroku is a cloud platform as a service (PaaS) supporting several programming languages that is used as a web application deployment model.

Deploying Applications: Application development on Heroku is mainly done via git. When deploying your web project to Heroku, first you need to build the project in the terminal in the editor of the project code. By typing 'ionic build' in the terminal, ionic build will perform an Ionic build, which compiles web assets and prepares them for deployment, then use git command to push the project code to your github repository. Finally you just need to go into the project repository in your Heroku account to view the Deploy bar, select Github in the deployment method, and bind yours A GitHub link in the Github account where you want to deploy the project. After completing this, you can deploy your project to Heroku using automatic deployment and manual deployment below. After the deployment is complete, you will be able to view your project by clicking 'Open APP' in the top right corner of the Heroku project page.

Applications that are run on Heroku typically have a unique domain (typically "applicationname.herokuapp.com") used to route HTTP requests to the correct dyno.

## 5.3 Architecture

### 5.3.1 System Architecture

In the project I am using a powerful hybrid/hybrid HTML5 mobile development framework Ionic3. Its features are the use of standard HTML, CSS, and JavaScript to support cross-platform development (currently support: Android, IOS) native app applications. . The Ionic framework provides the highest Web and native application components for building highly interactive local and progressive Web applications. It is the only mobile application stack that allows developers to design applications for all app stores and mobile networks, using a unique code base. In addition, Ionic's platform



continuity ensures that your application looks great on every device. At the same time it provides web developers with a complete mobile toolkit and everything you need to create a complete mobile app in minutes. You can think of Ionic as a front-end UI framework that handles all the appearance and UI interactions your application needs. It is like a "local bootstrap" but supports a wide range of common local mobile components, smooth animations, and incredible designs. Some features of the Ionic framework development are shown below:

Ionic apps are made of high-level building blocks called components. Components allow you to quickly construct an interface for your app. Ionic comes with a number of components, including modals, popups, and cards. When creating a component, a component folder is created. In addition to the component we created, there is also a `components.module.ts` file. This file mainly helps us to put all our components together and then put it. This file is imported into the imports and providers inside `app.module.ts`. see figure 5.2:

```
declarations: [  
  MyApp,  
  HomePage,  
  OtherPage  
],  
imports: [  
  BrowserModule,  
  IonicModule.forRoot(MyApp),  
  ComponentsModule  
],  
bootstrap: [IonicApp],  
entryComponents: [  
  MyApp,  
  HomePage,  
  OtherPage  
],  
providers: [  
  StatusBar,  
  SplashScreen,  
  ComponentsModule,  
  {provide: ErrorHandler, useClass: IonicErrorHandler}  
]  
)
```

Figure 5.2: Ionic Component

I have used many of the components of the ionic framework in my project, The `AngularFireModule.initializeApp(firebaseAuth)` in the picture implements the initialization of the application and then imports the authentication function code. show below: see figure 5.3, figure 5.4 and figure 5.5:

Putting it in `app.module.ts` means that you can use this component on all pages. If you only use it on a page, import it into the `module.ts` file on the specified page.

```
1,
imports: [      //导入其他模块, 就是要使用其他模块的功能, 必须要导入。封装好的。
  BrowserModule,
  IonicModule.forRoot(MyApp),
  // Add 2 imports
  AngularFireModule.initializeApp(firebaseAuth),
  AngularFireAuthModule
],
```

Figure 5.3: Ionic Component in project

```
providers: [ //服务提供者, 主要用来定义服务(service)。
  StatusBar,
  SplashScreen, //启动画面
  LoadingController,
  StreamingMedia,
  App,|
  {provide: ErrorHandler, useClass: IonicErrorHandler}
],
```

Figure 5.4: Ionic Component in project

In the project I used a lot of native UI components provided by the Ionic framework, as shown below:

1. Alert:

Alerts are a great way to offer the user the ability to choose a specific action or list of actions. They also can provide the user with important information, or require them to make a decision (or multiple decisions).

From a UI perspective, Alerts can be thought of as a type of “floating” modal that covers only a portion of the screen. This means Alerts should only be used for quick actions like password verification, small app notifications, or quick options. More in depth user flows should be reserved for full screen Modals. see figure 5.6:

2. Button:

Buttons are an essential way to interact with and navigate through an app, and should clearly communicate what action will occur after the user taps them. Buttons can consist of text and/or an icon, and can be enhanced with a wide variety of attributes. see figure 5.7:

3. Card: Cards are a great way to display important pieces of content, and are quickly emerging as a core design pattern for apps. They are a great way to contain and organize information, while also setting up predictable expectations for the user. With so much content to display at once, and often so little screen real estate, cards have fast become the design pattern of choice for many companies, including the likes of Google, Twitter, and Spotify.

```

31 const firebaseAuth = { //firebase登录方法配置
32   apiKey: "AIzaSyCwYyA_drTwdJ-_N80A2vajNXfvKxyzh9ks",
33   authDomain: "netmusic-c9a02.firebaseio.com",
34   databaseURL: "https://netmusic-c9a02.firebaseio.com",
35   projectId: "netmusic-c9a02",
36   storageBucket: "netmusic-c9a02.appspot.com",
37   messagingSenderId: "971450191473"
38 };

```

Figure 5.5: Ionic Component of Firebase Authentication

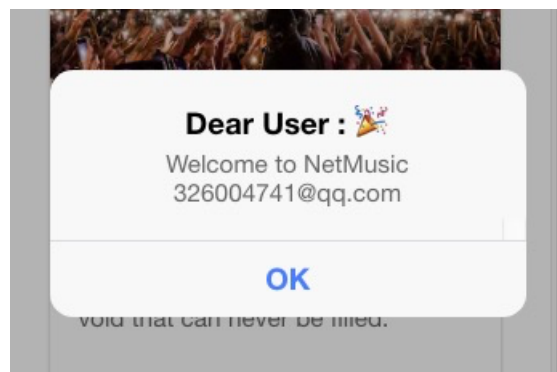


Figure 5.6: Ionic Component for Alert

For mobile experiences, Cards make it easy to display the same information visually across many different screen sizes. They allow for more control, are flexible, and can even be animated. Cards are usually placed on top of one another, but they can also be used like a "page" and swiped between, left and right.

3.1 Images In Cards: Images often vary in size, so it is important that they adopt a consistent style throughout your app. Images can easily be added to cards. Adding an image to a card will give the image a constant width, and a variable height. Lists, headers, and other card components can easily be combined with image cards. see figure 5.8:

4. Icons: Ionic comes with the same 700+ Ionicons icons we've all come to know and love. see figure 5.9:

5. Inputs: Inputs are essential for collecting and handling user input in a secure way. They should follow styling and interaction guidelines for each platform, so that they are intuitive for users to interact with. Ionic uses Angular 2's form library, which can be thought of as two dependent pieces, Controls, and Control Groups.

Each input field in a form has a Control, a function that binds to the value

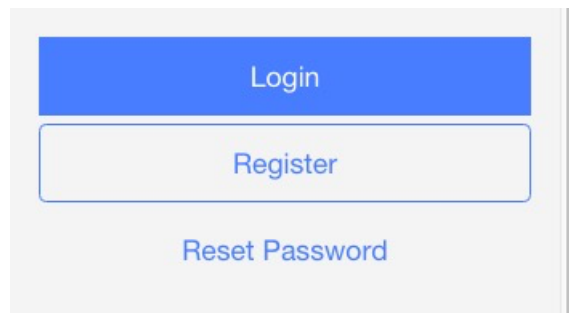


Figure 5.7: Ionic Component for Button

in the field, and performs validation. A Control Group is a collection of Controls. Control Groups handle form submission, and provide a high level API that can be used to determine whether the entire form is valid.

6. Loading: The Loading component is an overlay that prevents user interaction while indicating activity. By default, it shows a spinner based on the mode. Dynamic content can be passed and displayed with the spinner. The spinner can be hidden or customized to use several predefined options. The loading indicator is presented on top of other content even during navigation.

7. Toast: Toast is a subtle notification that appears on top of an app's content. Typically, Toasts are displayed for a short duration of time then automatically dismiss.

### 5.3.2 Main Process

The key process is:

1. Welcome Page: The homepage provides three function buttons: Login, Register and ForgotPassword. After the three buttons are clicked, they will be navigated to the exclusive pages of the three functions. Among these three pages, button functions for returning navigation are added, and when you click the button by mistake, it is convenient to return to the previous page.
2. Registration: After the user enters the correct e-mail address and password of no less than six digits, he can click the register button to register. During the registration process, the 'createUserWithEmailAndPassword' method in Firebase is called and the email and password you enter are stored in the firebase authentication section[3].
3. Login page: After the user enters the registered mailbox and face, he can click the login button to log in. After the success, pop-up window will appear

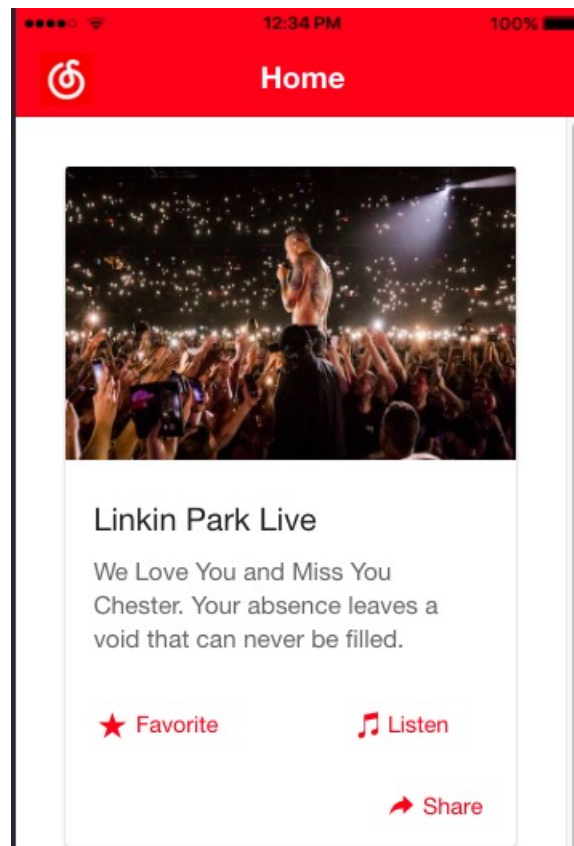


Figure 5.8: Ionic Component for Card

and the user's account mailbox will be displayed in the pop-up window, and then navigate to `LoggedInPage(Home)`.

4. **Forgotten Password:** In the input box, the user enters the email account that needs to change the password. After clicking the button, the user will receive the email to change the password in the registered email, click on the email link, and enter the new password that the user wants to change. , you can successfully use the new password and sync to Firebase at the same time.

5. **Home:** List the songs you may like, click Play and you can jump to the playlist to play.

6. **PlayList:** Show songs and artist and song names in the song list, and click play and stop to control the music. see figure 5.10 below:

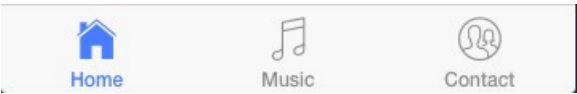


Figure 5.9: Ionic Component for Icons

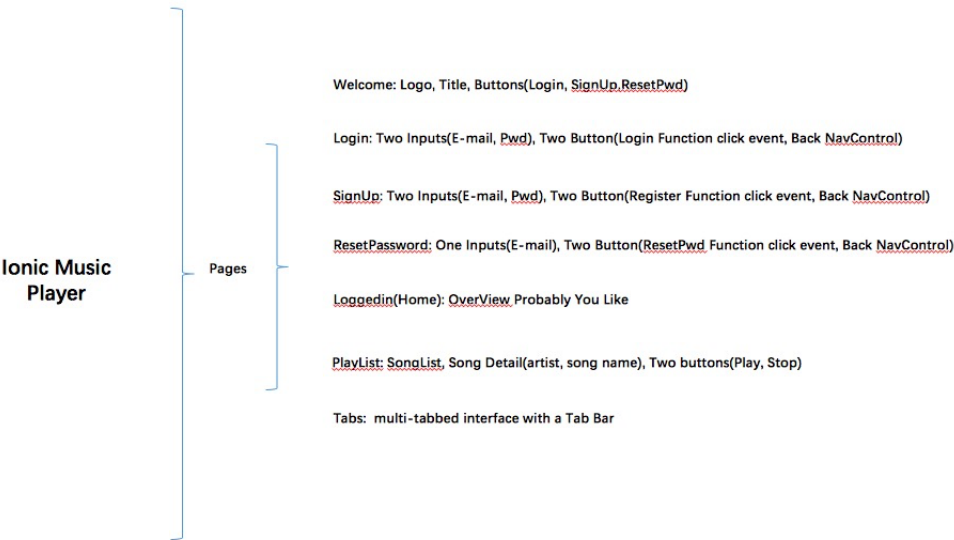


Figure 5.10: Project Pages

## 5.4 System Functions Review

According to the functional requirements of the system, the music player system is divided into front-end management and back-office management. Front-end management includes: page navigation design, page design, button type design, pop-up window design, waiting for loading design, Toast reminder design, browsing song list, song title and artist details, recommended song section. Back-end management includes user management module, song list management, song function management.see figure 5.11 below:

Front-end as follows:

Page Navigation:

All three pages in the login system can be returned to the main interface by returning navigation buttons.

Page Design:

The project has a total of seven pages:

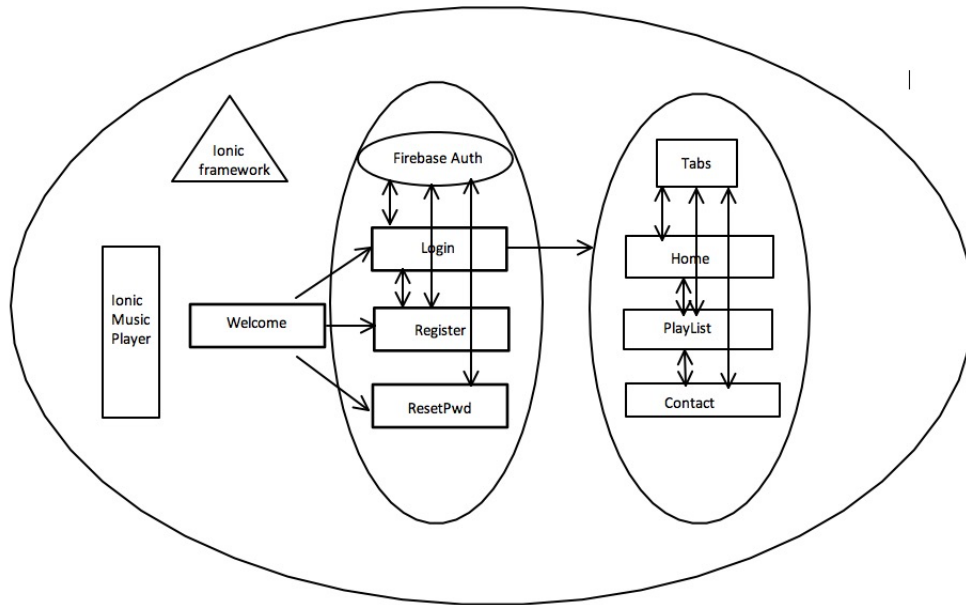


Figure 5.11: System Design

Welcome,  
Login,  
Register,  
ForgotPassword,  
Home,  
PlayList,  
Contact.

Pop-up window design(Alert):

The project sets Alert on the following pages:

Login,  
Register,  
Home.

Loading design:

The Project sets Loading on the following pages:

Login,  
Register,  
Forgot Password,  
Home.

Toast Notification:

The Project sets Toast Notification on the following pages:

Forgot Password.

Back-end as follows: User Management:

For users, you can register with the correct email address and a password that is not less than six digits. You can log in successfully after you register. If the user forgets the password or wants to change the new password, he or she can enter the email address of the account directly. The email authentication function verified by the Firebase user will receive an email to change the password notification in the email address entered by the user. Enter the new password that the user wants to modify. Click OK to successfully change the password. The user can use the new password to log in to the account. At the same time, the user's new password will also be synchronized to the Firebase authentication module[3], which facilitates unified management of the user management system.

For Admin: For administrators, the registration information of registered users can be accepted in the Firebase authentication module[3], including the registered email address, password, registration time, latest login time, and user's UID code. At the same time, the administrator also has the authority to perform certain specific operations on the registered account, for example, deactivate the specified account, delete the specified account, and reset the account password. Such operations increase the administrator's back-end high-intensity management of accounts, and can also operate user accounts in the background to meet user needs when the user cannot perform account management operations. Another point is that when some users use the application to perform illegal or forced changes to the program system files and other undesirable operations, the administrator can immediately find out the specified account, or delete the specified account information. In this way, illegal users are forced to exit the application immediately, and it is also convenient for the administrator to protect the application. About User management image see figure 5.12 below:

(1) Login Function:

Create a form that allows existing users to sign in using their email address and password. When a user completes the form, call the `signInWithEmailAndPassword` method:

The method for implementing the login function is shown in the figure 5.13 below:

(2) Sign up new users Function:



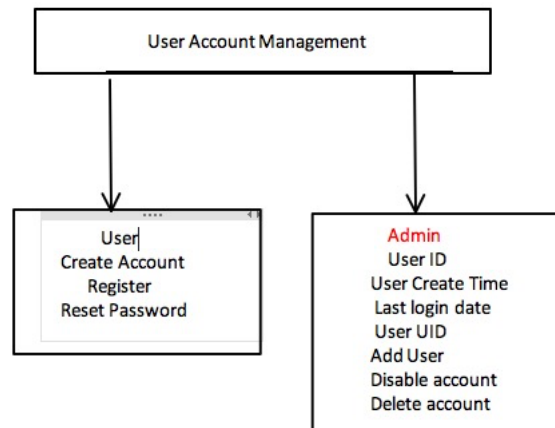


Figure 5.12: System User Management

Create a form that allows new users to register with your app using their email address and a password. When a user completes the form, validate the email address and password provided by the user, then pass them to the `createUserWithEmailAndPassword` method: shown in the figure 5.14 below:

(3) Send a password reset email Function:

You can send a password reset email to a user with the `sendPasswordResetEmail` method. For example: shown in the figure 5.15 below:

(4) Signs out the current user Function:

You can use the `SignOut` method to log out of user accounts. For example: shown in the figure 5.16 below:

(5) Play Music Function:

Ionic native plug-in Streaming Media is used to implement music play function. This plugin allows you to stream audio and video in a fullscreen, native player on iOS and Android. Supported platforms have Amazon Fire OS, iOS, Android. For example: shown in the figure 5.17 below:

(6) Stop Music Function:

Ionic native plug-in Streaming Media is used to implement music stop function. This plugin allows you to stream audio and video in a fullscreen, native player on iOS and Android. Supported platforms have Amazon Fire OS, iOS, Android. For example: shown in the figure 5.18 below:

```

login() { // Add login function 方法
  // firebase.auth.Auth.signInWithEmailAndPassword(email: string, password: string): Promise<any>
  // this.uname.value => username(email) 当前用户的帐号, this.password.value => 当前用户的密码, 直接用this.引用
  this.fire.auth.signInWithEmailAndPassword(this.uname.value, this.password.value)
  .then( data => {
    console.log('got some data', this.fire.auth.currentUser); //在console传出当前账号用户的信息.
    this.alert('Welcome to NetMusic ' + this.fire.auth.currentUser.email); //套用当前文件的alert()函数
    this.navCtrl.setRoot( TabsPage ); // 定向导航到LoggedInPage,且不可返回. 如想要返回: 改用navCtrl.push()
    // 设置当前导航堆栈的根目录. 可以把LoggedInPage更改为播放器主界面.
    // setRoot(): 只是view之间的切换, 不带有任何返回button之类的东西. 不能返回的
    // user is logged in
    let loader = this.loadingCtrl.create({
      content: "Please wait...",
      duration: 500
    });
    loader.present();
  })
  .catch( error => {
    console.log('got an error', error); // if 登录信息不匹配, then 在这里报错, 并弹出alert信息.error 信息自动生成
    this.alert(error.message); // email error: message: "The email address is badly formatted."
    // password error: message: "The password is invalid or the user does not
  })
  // 登录成功后在console.log中显示 would sign in with + 登录的帐号和密码.
  console.log('Would sign in with ', this.uname.value, this.password.value);
}

```

Figure 5.13: Login Function

## (7) Play Video Function:

Ionic native plug-in Streaming Media is used to implement video play function. This plugin allows you to stream audio and video in a fullscreen, native player on iOS and Android. Supported platforms have Amazon Fire OS, iOS, Android. For example: shown in the figure 5.19 below:

```
signup(){ //registerUser()
  //Api connections
  //定向导航到TabPage, 要记住上面的import
  // this.navCtrl.push(LoginPage);
  //写注册 登陆方法用邮箱
  this.fire.auth.createUserWithEmailAndPassword(this.uname.value, this.password.value)
    .then(data => {
      console.log('got data', this.fire.auth.currentUser);
      this.alert('Registered!');
      this.navCtrl.push(LoginPage);
      let loader = this.loadingCtrl.create({
        content: "Please wait...",
        duration: 500
      });
      loader.present();
    })
    .catch(error => {
      console.log('got an error', error);
      this.alert(error.message);
    });
  console.log('Would register user with ',this.uname.value, this.password.value);
}
```

Figure 5.14: Register Function

```
resetpassword(){
  this.fire.auth.sendPasswordResetEmail(this.email)
    .then(date => {
      console.log('send email');
      let toast = this.toastCtrl.create({
        message: 'Congratulations! Please check your email to change your new password.',
        duration: 3000,
        position: this.position
      });
      toast.present(toast);
    })
    .catch(error => {
      console.log('got an error', error);
      this.alert(error.message);
    });
}
```

Figure 5.15: Reset Password Function

```
logout() {  
  // push():是异步的, 带有过度效果, 可以返回(有返回按钮)  
  // setRoot():是一个view之间的切换, 所以不带有返回button之类的东西,  
  // this.nav.setRoot(page,params,opt);  
  // 1、page 组件: 想放进导航的 组件名称  
  // 2、params 参数: 像下一个页面传递参数 Object类型  
  // 3、过度完成。返回一个promise  
  window.localStorage.removeItem(this.email);  
  // this.navCtrl.setRoot(WelcomePage);  
  this.app.getRootNav().setRoot(WelcomePage);  
  console.log(this.fire.auth.currentUser);  
  //window.localStorage.removeItem('email');  
}
```

Figure 5.16: Sign Out Function

```
play(){  
  let options: StreamingAudioOptions = {  
    successCallback: () => {console.log('Finished Audio')},  
    errorCallback: (e) => {console.log('Error: ', e)},  
    initFullscreen: false, // IOS only!  
  };  
  this.streamingMedia.playAudio('file:///Users/weichenwang/Music/%E7%BD%91%E6%98%93%E4%BA%91%E9%9F%B3%E4%B9%90/
```

Figure 5.17: Play Music Function

```
stopAudio(){  
  this.streamingMedia.stopAudio();  
}
```

Figure 5.18: Play Music Function

```
startVideo(){  
  let options: StreamingVideoOptions = {  
    successCallback: () => { console.log('Finished Video')},  
    errorCallback: (e) => {console.log('Error: ', e)},  
    orientation: 'portrait'  
  };  
  this.streamingMedia.playVideo('http://www.sample-videos.com/video/mp4/720/big_buck_bunny_720p_30mb.mp4', options);  
}
```

Figure 5.19: Play Video Function

## 5.5 Security System

### 5.5.1 User Security

When the user is registering an account, he/she needs to enter the correct email address to be able to receive emails from the Firebase authentication[3] background when the password is changed, and modify the current user's password through the link in the email. At the same time when registering, the user also needs to input a password which cannot be lower than six digits, otherwise the system will pop up Alert to prompt you with an error message. In addition, regardless of whether the user is registering or logging in, when the password is entered, the entered password is encrypted and overwritten so that when the password is entered, the last character or number currently displayed can be seen, and three seconds after the input is stopped. A character or number is also overwritten by encryption. About User security system have some images below:

Login method does not meet system requirements see figure 5.20 and figure 5.21 below:

Register method does not meet system requirements see figure 5.22 below:

## 5.6 Appearance Design

Following section will introduce the interface design of my music play:

### 5.6.1 MainPage(Welcome) Design

This Section used to describe the Main page design. The main page contains three function buttons, which are Login, Register and Reset Password.

(1) see MainPage figure 5.23 below:

(2) see Login figure 5.24 below:

(3) see Register figure 5.25 below:

(4) see Reset Password figure 5.26 below:

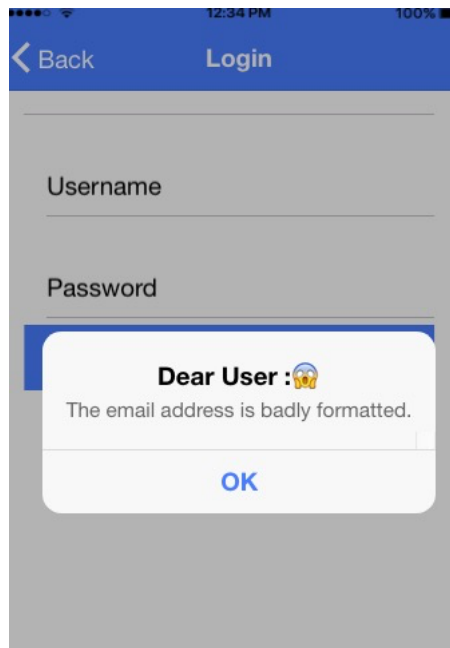


Figure 5.20: User Login Security System

The application's Logo is also designed on the main page, as well as Title.

### 5.6.2 Home Page Design

This Section used to describe Home page design. The home page mainly contains a Card UI component. In the Card component, a picture of a recommended song, a song name, a song introduction, adding to a favorite, playing music, and sharing are displayed. The Logout function button is also designed at the bottom of HomePage. At the same time, Tabs tabs are designed at the bottom of the application. The contents of the tabs are: Home, Music and Contact, which can facilitate the conversion of the main page after entering the program.

- (1) Screen shot of Home page see image 5.27 below:
- (2) Screen shot of PlayList Page see image 5.28 below:
- (3) Screen shot of Contact Page see image 5.29 below:

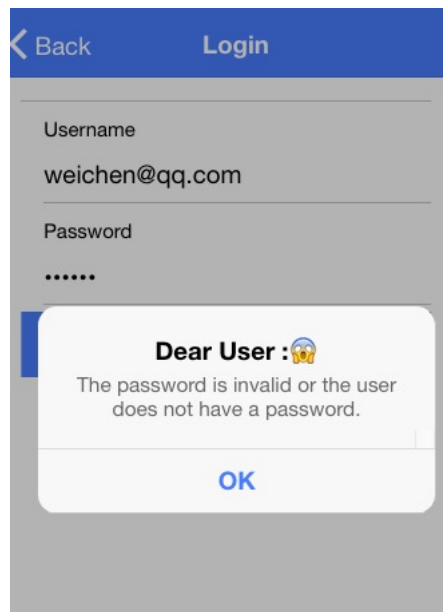


Figure 5.21: User Login Security System

For the Contact page, the main implementation is the playback of music audio and video playback. The method used to achieve this is the ionic native Cordova plug-in StreamingMedia.



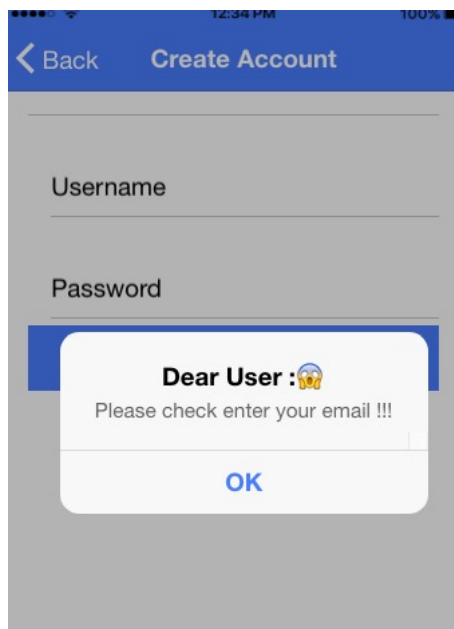


Figure 5.22: User Register Security System

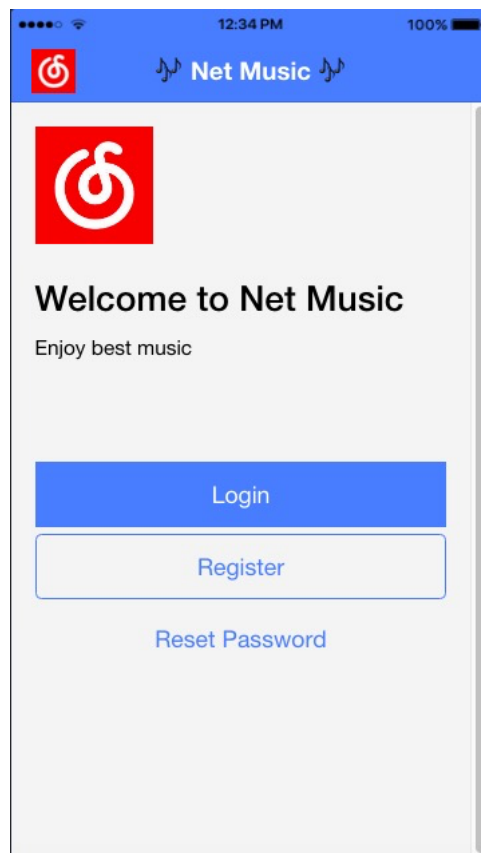


Figure 5.23: Main Page

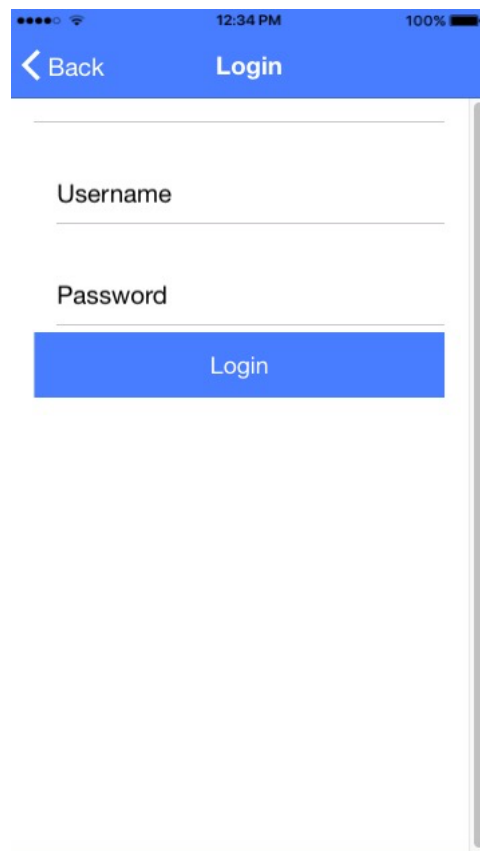
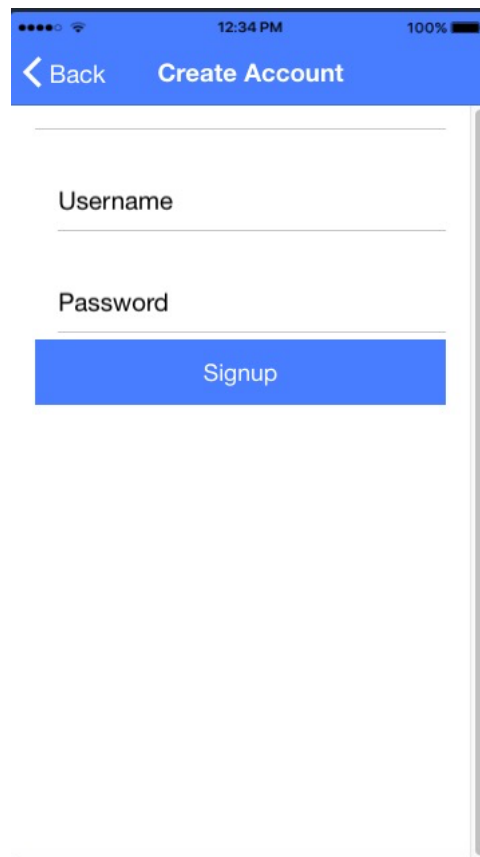


Figure 5.24: Login Page



A mobile application interface for creating a new account. The screen has a white background with a blue header bar. The header bar contains a back arrow and the text 'Back' on the left, and 'Create Account' on the right. Below the header, there are two text input fields: the first is labeled 'Username' and the second is labeled 'Password'. Both fields have a light gray border. Below the password field is a blue rectangular button with the text 'Signup' in white. The status bar at the top of the screen shows signal strength, Wi-Fi, the time '12:34 PM', and '100%' battery.

Figure 5.25: Register Page

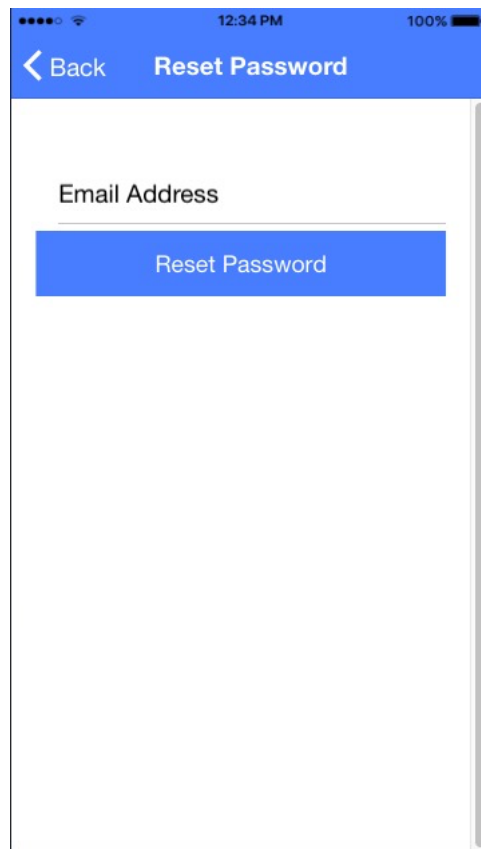


Figure 5.26: Reset Password Page

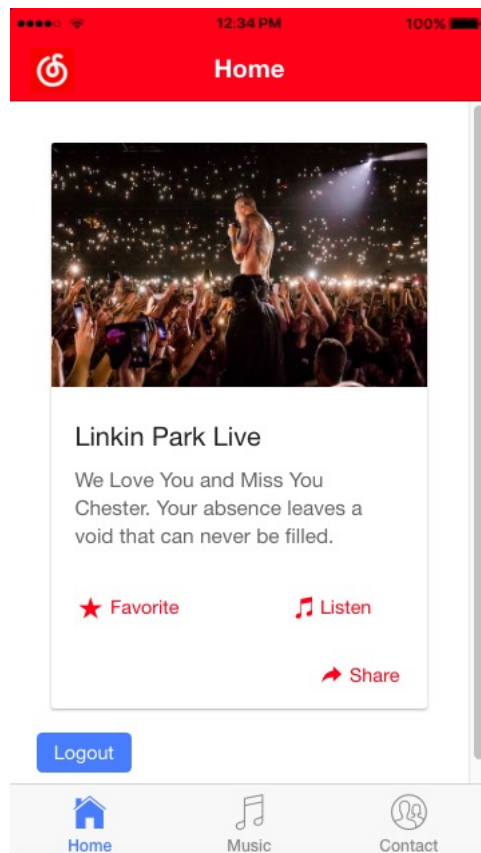


Figure 5.27: Home Page

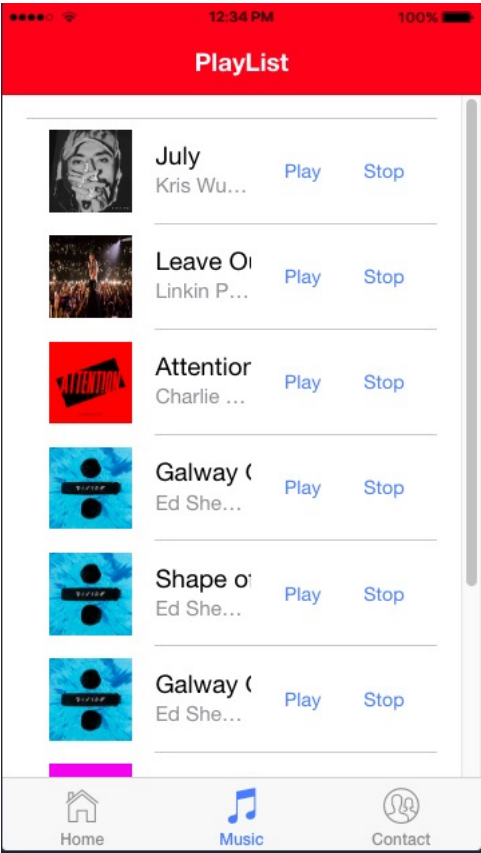


Figure 5.28: PlayList Page

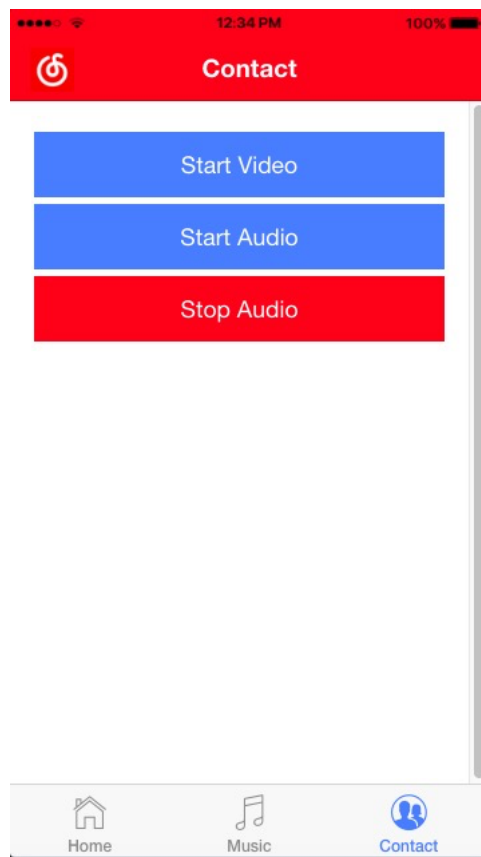


Figure 5.29: Contact Page



# Chapter 6

## System Evaluation

### 6.1 Advantages

This is Node.js Project which has the fastest growing ecosystem with well over 110,000 free JavaScript modules. Tapping into this resource can help you to develop apps faster and more efficiently[10].

Use the same platform to develop your Server and Mobile apps. Many times your mobile app will have a corresponding browser based application for desktop access. Having one platform and one codebase for both the browser and mobile applications can make the development process far more efficient.

The Fast Suite: Node.js runs on the V8 engine developed by Google that uses JavaScript into native machine code and operates at a super speed. Node.js spares all the trouble involved with forming separate threads and instead uses a single thread, that is, the event loop that takes care of all the asynchronous I/O operations. Major actions in web applications including reading or writing to the database, network connections or file system can be performed quickly with this suite.

Rise In Productivity: All the Node.js programs are crafted using JavaScript which implies that the silos that were present between the frontend and the backend developers are split entirely thereby, increasing the effectiveness of the web development process at different enterprises. Organizations can combine the frontend and backend teams together into one unit for smooth functioning by getting rid of the inter-silo dependencies.

The Hosting Deal: This project passed Github's successful deployment of

the project to Heroku. Less time on repetitive processes means that developers can spend more time and attention on creating quality code, which inherently reduces bugs and issues. A regular release cadence allows product teams to loop in user feedback along the way. This helps further hone the app's feature set and user experience, and increases customer satisfaction.

The project uses the Ionic framework as a system framework because of the cross-platform compatibility of Ionic[11]. Ionic framework brings the functionalities of platform-specific optimized CSS equivalent on various mobile Operating Systems. Ionic drastically reduces the need for code rewriting with its mobile-optimized HTML, JS, and CSS components. Moreover, ionic can easily be integrated into AngularJs making the code structure more manageable[8].

Also benefits from Ionic's clean user interface, Ionic framework has numerous built-in CSS and JS components that cover most of the essential things which are needed for mobile application development. Some of these components include navigation, buttons, tabs, sliding menu, sliding boxes, form inputs, and many more. Ionic's default style is highly intuitive, smooth and you can customize it as per your need by simply adding predefined CSS classes.

In the project I also designed a secure user management system using Google Firebase authentication. Firebase Authentication provides a more engaging experience across platforms and apps by providing back-end services, an easy-to-use SDK, and an off-the-shelf interface library to authenticate users logging in to your app. Firebase Authentication supports authentication via passwords, popular federated identity providers such as Google, Facebook, Twitter, and more, making it easy for users to access your content and sign in to your app quickly and securely.

## 6.2 Limitation

Due to the short-term development of a brand-new project, the design of the entire music player is not perfect enough, and the overall code design also has many defects. But I think this music player will continue to maintain its advantages and features and will be better under the joint development of Node.js platform and Ionic's native framework.

Inefficient code:

First, due to the low level of proficiency in Node.JS and Ionic's native frame-

work, it takes a lot of time during the development process to view official developer documentation and develop instructional videos, greatly reducing the time required for effective development projects. And in order to have a thorough understanding of the functional code of each part, it also takes a lot of time to search for an explanation about some of the code function blocks. Caused part of the code can not be perfect and clear logical thinking.

Functional implementation is not rigorous:

Due to short development times, some features in the music player have not been completed. After the user logs in to the Home page, he cannot add the songs pushed from the home page to the favorite list and cannot share it. And for the song pushed by the homepage, only the push song is currently set as a song that already exists in the playlist. In this way, when clicking Listening, the jump to the playlist page and play the specified song can be realized.

For the Contact page, I only temporarily implemented the video file playback, audio playback, and stop audio features, but did not improve the MV for the specified song. And these playback files can only be local files and cannot support online URL play.

For the playlist page, I just implemented an orderly arrangement of the music, displaying the song title as well as the artist. It has not been perfected that playing a song is a detailed information page that can enter a specific song, and can show the lyrics and progress bar of the song. In addition, I only implemented the playback and stop of music. For the time being, I haven't realized more control over the music. For example, the previous song, the next song, the loop play, the disorderly play, and so on.

For the database design of the user management system, due to the time, I failed to add more personal information about the user to each user's database, for example, the user's nickname, gender, personal picture, personal introduction, and so on. The structure of the database is incomplete, and only the user's account and password information can be stored. I think this is a bad performance for a good music player user management system database.

## 6.3 Future Development and Optimization

Optimize project code:

First of all, I will gradually optimize the inefficient code mentioned above, and look at the official documents and teaching videos to better understand the design structure and function blocks of the code. Gradually optimize the user management system database, playlist design, and player function design. Use more efficient and concise code design to complete the optimization.

Solving the limitations of playing only local resources can realize the function of playing online songs and videos. The player's playback function, such as the previous one, the next, reaches the user's basic experience with the music player.

This chapter introduces the specific deficiencies of the project and the deficiencies in the use of the framework, as well as plans for the future of the project to make changes and improve, and to explain the development of the framework and the final expected results in the future.

# Chapter 7

## Conclusion

After more than two months of unremitting efforts, I finally completed my final project design. The application is titled "Net Music Player"

First of all, this is an IOS music player based on Node.js and Ionic framework.

Second, the project was deployed on the Heroku cloud service platform via GitHub. And you can open and test the project directly in Heroku's project repository.

Third, the database of the project uses the Google Firebase database, which stores the user's registered account information in the Firebase database and facilitates the administrator's management of user accounts.

Fourth, the UI design and background function design of the entire project are based on Ionic's native development components.

At the time, when I first learned that there was no limit to the programming language used in the final year of the project design, my heart was very excited and excited, but I was lost in confusion because I knew my strength and level. I am currently It's just a student who has only been exposed to software development for a year, and I don't know much about many programming languages for real software development. When I chose to use the Ionic3 framework to develop my project, to be honest, I didn't know how the Ionic framework works at the time, and the reason for choosing it was simply to see people around it use it to develop applications and feel the whole thing. The application is very simple and lightweight, so Ionic was chosen. But when I personally started the development project, I found nothing about the ionic framework. I never learned to use the Ionic framework to de-

velop Web or mobile applications. The initial design of the project was also based on his own assessment of the design of various music players over the years. But when deciding to stick to the Ionic framework for development, I spent a lot of time to understand and learn about the Ionic framework, which led me to gradually discover that the Ionic framework can quickly build the basic model interface needed for the entire application. Allows developers to save a lot of time to focus on developing the function block code behind the application. This is also the most direct advantage of the Ionic framework. I found that I slowly began to like it.

When I was deploying the project through Github to the Heroku cloud services platform, I got help from my classmates. He taught me about the relationship between Heroku, Github and Ionic, and taught me that I understood to deploy the project. Before going to Heroku, you need to create an Ionic project on the Visual Studio Code terminal so that the project will be built on the cloud server. Then upload the project code to the repository via Github, and finally connect to the project through Github and Heroku to complete the final deployment of the project. I am very grateful to my classmates for letting me deeply understand the working relationship within the project deployment.

Due to time constraints and lack of practical experience, the graduation system has not been perfected.

Overall, although the graduation design is over, there are many loopholes in the code development of software development. I believe that the teacher's eyes also discovered that there are many naive and inadequate areas in the project. But this time graduate design has achieved great results. For me, it is a test that reflects the current level of learning. Let me once again have a newer, higher, better understanding of software development, but also let me have a new understanding of future career. I think that the results of the project are not the most important ones. The most important thing to sum up is the experience and knowledge gained from the development process and the good refreshment of the self.

# Chapter 8

## Appendix

Github Link:

<https://github.com/w326004741/Final-Project-Year4>

Heroku Deploy link:

<https://wtest1.herokuapp.com/>

Learning Resources Link:

Nodejs basic information, such as how to set the environment and data types:

<http://www.tutorialspoint.com/nodejs/>

Some basic information of the Ionic framework, such as UI components, Ionic native components, Icons:

<https://ionicframework.com/docs/>

Firebase Authentication Tutorial:

<https://firebase.google.com/docs/auth/?authuser=0>

Ionic Native Streaming Media:

<https://ionicframework.com/docs/native/streaming-media/>

Deploy Project to Heroku:

<https://stackoverflow.com/questions/43701033/ionic-pwa-deploy/43701116>

Rapid Prototype Model:

<https://www.justinmind.com/blog/rapid-prototyping-isnt-helpful-or-is-it/>

Firebase Introduction:

<https://firebase.google.com/products/?hl=zh-cn>

<https://tw.alphacamp.co/2016/07/22/firebase/>

Cordova:

[https://en.wikipedia.org/wiki/Apache\\_Cordova](https://en.wikipedia.org/wiki/Apache_Cordova)

<https://cordova.apache.org/docs/en/latest/guide/overview/index.html#plugins>

<https://www.jianshu.com/p/e02172c53296>

Youtube Video Tutorial:

Login System: <https://www.youtube.com/watch?v=PNzE1MX5iRU>

Streaming Meida: <https://www.youtube.com/watch?v=S8NsG7jOgcc>



# Bibliography

- [1] A. Ma, “The 5 best music apps that won’t tear through all your data,” 06/24/2015.
- [2] E. G. ADISESHIAH, “Rapid prototyping isn’t helpful. or is it?,” November 17, 2016.
- [3] Firebase, “Firebase authentication,” April 13, 2018.
- [4] Firebase, “Firebase products,” April 13, 2018.
- [5] Snorlax, “How ionic uses cordova plugin,” 02/08/2018.
- [6] Wikipedia, “Apache cordova,” 3 April 2018.
- [7] Cordova, “Cordova plugin,” 2018.
- [8] Wikipedia, “Angularjs,” 10 April 2018.
- [9] N. Reed, “What is the file ‘package.json’?,” Aug 26 2011.
- [10] P. Pal, “12 benefits of using node.js for web applications,” Nov 21, 2016.
- [11] SIVA, “Advantages of ionic framework app development,” DECEMBER 1, 2016.