

手机音乐播放器界面设计与后台管理

专业：计算机科学与技术 学生：黄雨龙 指导老师：周修考

摘要

音乐，作为一门可以去反应了人们生活情感的艺术，它能够令人心旷神怡；它可以作为一种听觉上的享用；它可以使得我们的心灵更加干净。我们的情操可以依赖音乐来陶冶，我们的自身的情感也可以通过音乐来抒发。在以前，人们要享受音乐，只能通过乐器的演奏才能达到这一目的。随着时代的进步，出现了留声机、录音机、CD 播放器等能够播放音乐的器件。到了如今，手机成了最流行、最为广泛使用的音乐播放器。

本设计为手机音乐播放器系统，是基于 HTML5、JavaScript 及 node.js 开发的一款手机 WebAPP，以手机音乐 WebAPP 播放器为主题设计而成，使用户可以不用下载就可以通过上网来享受音乐。用户可以通过此 APP 来进行获得推荐歌曲，达到无目的听歌，也可以通过收听个人收藏歌曲，达到有目的听歌。本文从理论和实践两个方向出发，对一个具有双重听歌功能的音乐 APP 进行分析和实现实现。论文首先对本 WebAPP 音乐播放器的课题背景、分析现状及研究课题的目的和意义进行概述，又通过对目前手机音乐播放器的生存环境来进行系统需求及可行性的分析，确定了系统的设计目标及实施方案；最后本着可用性、先进性、标准性的原则实现了系统的各个功能模块及进行系统的实现进行了较为详细的讲解。

本音乐播放器的主要功能有：用户的注册/登录、对本地音乐进行搜索、向服务器发送音乐搜索请求、上传用户的收藏记录、歌曲的上一首/下一首、歌曲的播放/暂停、对歌曲的评论/留言等。论文在撰写当中，力求通过理论和实践应用的相互结合，给出一份系统相应功能模块的阐述。在研究最后讨论了平台中的一些不足，并提出一些完善和改进的方向。

关键词：WebAPP 音乐播放器 HTML5 node.js 歌曲播放

目 录

1 绪论	3
1.1 背景及现状	3
1.2 研究目的及意义	4
1.3 研究的主要内容	4
2 相关技术基础	5
2.1 HTML5.....	5
2.2 JavaScript	6
2.3 Node.js.....	6
3 系统的需求分析	7
3.1 功能需求	7
3.2 运行及其他需求分析	9
4 系统设计	10
4.1 软件结构的设计	10
4.2 系统的客户端设计	11
4.3 系统的后台设计	12
5 系统的实现	14
5.1 用户的注册/登录设计	14
5.2 主页面	16
5.3 播放器	20
6 系统的测试	22
6.1 测试方法	22
6.2 测试结果	22
结论	24
参考文献	25

1 绪论

1.1 背景及现状

中国互联网络信息中心于 2017 年 01 月 22 日发布《中国互联网络发展状况统计报告》。报告中明确的指出：统计时间为 2016 年 12 月，我国具有 7.31 亿的上网用户，同期时期，欧洲的全部人口也才约 7.2 亿^[1]。在这 7.31 亿中，手机上网用户拥有上网用户的 95.1%。这意味着手机上网已经成为了人们生活的一个重要部分，所以 WebApp 很有发展前景。

相比于互联网的出现及发展，音乐的出现目前已经无法进行考察，早在人类还没有任何一种语言时，音乐就已经出现了，原生的音乐就是声音的强弱和高低。这已经可以表达人类的感情和想法。随着人类劳动的发展，便慢慢出现了演奏乐器及播放器件。在《音乐爱好者》一刊中，曾指出：“音乐可以消除人的工作紧张、可以有效的避免慢性疾病、可以减轻人的生活压力”^[2]。计算机科学与技术及互联网的发展，为音乐的传播提供了一个新的渠道。各种各样的音乐播放器软件纷纷如雨后的春笋，一夜之间不断的冒出，而当 WebApp 与音乐结合在一起，这就产生了全新的音乐播放器 WebApp 音乐播放器。

从微软在 1998 年发布了 windows98 时，音乐播放器开始被逐渐接受，因为计算机与互联网的大力发展，各大公司开始逐步研究出了各种各样的音乐播放器。如今 WebApp 的兴起带动了 WebApp 音乐播放器的研发，在国内，以精美为主的 UI 和多功能模块为主的原生 APP 音乐播放器而言，使用次数居多的 QQ 音乐、酷狗音乐、网易云音乐等软件开发公司也进军了 WebApp 音乐播放器的市场，从一开始的单调的 UI 及简单的模块设计，逐渐的发展到如今的成熟 WebApp，其 UI 及各种模块框架已经与原生 App 不相上下，甚至在发展前景一面，更胜原生 App。而国外，webApp 播放器的形式更为多样化，UI 从简洁到精美，模块从单一到复杂数量数不胜数。其中以美国的 Spotify、挪威的 Tidal、丹麦的 Beoplayer、俄罗斯的 AIMP3 等，与时俱进，从原生 APP 向 WebApp 研究不断深入^[2]。

在目前而言，WebApp 在工具层面上，它正在时刻的扩展着自己，虽然存在性能、适配、分辨率、离线等问题，然而我们还是有理相信这些问题能够因为时代的进步和技术的进步而被顺利的解决，作为一种处于原始阶段的开发项目，WebApp 具有很高的潜力，所以 WebApp 音乐播放器也算是一种基于音乐的受众面积广泛而进行的一种推广尝试。在 WebApp 出现后，WebApp 的一款龙头软件 Finance Time，它在三个月时间里下载量已经突破百万大关，因此我们可以认为市场对 WebApp 的接受程度在不断的得到提高，WebApp 音乐播放器可能会迎来一

个发展的春天。

对于开发难度比较大的原生 APP 来说,平台的需求也很高,这一点会使得未来 WebApp 将因其开发简单和拥有跨平台方面的能力,在技术和商业方面得到越来越广泛的应用。相信在不久后 WebApp 是有可能取代原生的 App^[3]。

1.2 研究目的及意义

早期时候,受技术限制开发出来的功能单一、界面简洁的原生 APP 音乐播放器可以很容易的满足用户的使用需求。对于目前的技术而言,功能模块越来越多的原生 APP 音乐播放器对手机的内存要求越来越高,用户还需要对原生 APP 音乐播放器开放各种隐私权限,这很大程度的影响了用户使用。因而用户越发的想要拥有一款能够保证功能模块齐全的同时又占用手机内存小可以让自己的手机去存放其他东西、对手机的权限需求小可以保证自己隐私不被泄露的软件,所以研究 WebApp 音乐播放器势在必行^[4]。

在知乎、博客、搜狐社区等大型论坛,不断有用户进行对原生 APP 音乐播放器进行解剖统计,而结论中都提及到了市面上常用的原生 APP 音乐播放器的安装包大小在 20-40MB 之间,而软件权限基本高于 15 个,在 15-35 个之间。相反的,WebApp 音乐播放器它无需安装,只要在支持 HTML5 的浏览器中打开即可使用,它对设备的硬盘内存并没有需求,而运行内存也不需要很大,更重要的是 WebApp 音乐播放器并没有需要手机的任何权限。在更新和维护方面,WebApp 可以实时更新,发布后用户刷新就可以看到最新的版本,不用通过任何商店之类的审查,实时性很高^[5]。这也是研究 WebApp 音乐播放器的意义所在了。

1.3 研究的主要内容

在音乐播放器市场上,原生 APP 占据了大份额的市场,但随着 WebApp 的兴起,播放器的形式在不断的改变。本文主要是大体的介绍本款 WebApp 音乐播放器的设计过程,通过技术实现、用户需求及系统设计等方面,设计出一款方便大家使用的 WebApp 音乐播放器。

本文的首章分析了课题的研究背景,从音乐的演变及 WebApp 的诞生到 WebApp 的现状再到研究 WebApp 的目的及意义等方面介绍了 WebApp 的发展前景。

第二章 本课题涉及到的 HTML5、JavaScript 脚本、Node.js 等技术概论。

第三章 从系统的功能需求和、系统的运行需求及其他可行性需求对系统进行分析。

第四章 进行系统设计分析,阐述如何完善系统各功能模块和系统运行环境

的搭建。

第五章 分析系统如何进行功能界面的实现及系统的运行。

第六章 如何对系统进行测试。

2 相关技术基础

本课题系统是基于 HTML5、的产品开发，使用了 HBuilder 7.5 作为开发工具。作为 Web 的核心语言，HTML5 是标准通用标记语言下的一个应用超文本标记语言（HTML）的第五次重大修改。对于移动设备来说，开发 HTML5 应用的方法，一种是整份都是 HTML5 语法，另外一种是具有 HTML5 及 JavaScript 结合。这些应用有一个很大的优势，就是可以在网页上直接修改和调试，随着科技进步，HTML5 规范开发完成时将成为主流。据 IDC 统计，全球带有支持 HTML5 的浏览的手机是高达 10 亿的，同时开发 WebApp 的程序员数量已经高达 200 万。

2.1 HTML5

HTML5 的平滑过渡是它的核心理念，有的浏览器并不支持 HTML5，这个时候它可以自发的向前兼容，使得 Web 的内容正常显示。它拥有的开放接口，能够提供很多数据与应用的接入，让浏览器内部与一些外部应用进行数据的直接相连。

相较于其它开发语言，HTML5 它的跨平台性是十分的强大的，从手机浏览器，到平板电脑浏览器，再到 PC 浏览器都有基于 WebApp 程序存在。可以说，如果想要 WebApp 程序能够运行，那么只要用户拥有能够支持 HTML5 的设备即可。

HTML5 具备着简单易用性，它是数据利用标签来进行编写的，这样的属性精简表示使得它的文本传输效率得到了很大的提升。它的样式是通过 CSS 来编写，而 CSS 的编写样式是相对于其他语言而言较为简单的，所以同样的突显出 HTML 的简单^[6]。而它的事件则是通过 JavaScript 来完成，如果需要后台，则通常使用 Node.js 完成。

HTML5 也拥着 Audio、Video、Canvas 和摄像头等多媒体功能，它能够提高用户的体验度，随着对 HTML5 的研究不断深入，它也开始带有地理位置服务功能、文件的上传功能、离线应用功能、本地数据的存储功能等^[7]。它的潜力致使它不断的成长，慢慢的将会给用户带来更多、更好的体验。

2.2 JavaScript

作为一种解释型的脚本语言的 JavaScript，它的解释器是浏览器的一个部分，在客户端上被广泛的应用，为此得名 JavaScript 引擎^[8]。因为它是在运行中对代码进行逐行的解释而非完整编译后在进行解释，所以显得较为“异类”。

JavaScript 作为一种直译式的脚本语言，它需要的仅仅是浏览器的支持，而不是操作系统的支持^[9]。为此，在编写一个 JavaScript 完成之后就可以把它带到拥有支持 JavaScript 脚本语言的浏览器的机子上使用，幸运的是，JavaScript 被现在绝大数的浏览器所支持。

作为一种采用事件驱动的脚本语言，JavaScript 不需要经过 Web 服务器就能够对用户的输入和操作做出响应。当用户在访问一个网页时，JavaScript 可以直接的对鼠标的点击移动事件和键盘敲击事件进行响应。

JavaScript 采用的是弱类型的变量类型^[10]，对使用的对象和数据类型没有严格的要求，并且作为一种脚本语言它是基于对象的，可以创建新的对象，也可以使用着现有的对象，是基于 Java 基本语句和控制的脚本语言。

2.3 Node.js

Node.js 是基于 JavaScript 的一个开发平台，具有着单线程的特性，它的任务是按照顺序执行的，并且每次只执行一个，只有等队列前面的任务执行完毕之后，才会开始执行后面的任务^[11]。在 JavaScript 引擎中，JavaScript 代码的解释和执行线程只有一个，虽然还存在着处理 DOM 请求、AJAX 请求、读写文件、定时器等附属的工作线程，但 Node.js 的一个端口能够使用的主线程也只有一个。通常利用开创多个端口调用多个主线程来提高效率。

Node.js 通常来做后台具备事件驱动、同步/异步编程，也可以来设计网络的服务。他对于可扩展性高的服务器来说是一种“神器”，因为它的架构中拥有着“事件循环（event loop）”，所以会使得 Node.js 在编写时变得更为简单，能够让程序员在服务器的搭建编码上，简单而轻易的避开编写的“地雷”，让服务器的安全性得到提高，因此很多前端程序员在需要编写后台时，也常常使用 Node.js。

Node.js 的 Node 社区用户，通过不断的创新打造了许许多多的模块（扩展库），这些模块中，很多是具备连接数据库功能，或者是连接其它软件的驱动功能。有的社区用户通过自身高超的技术和能力，做出了 Node.js 的实用性软件，通过 node.js 使用时可以更为高效的解决问题。而它本身的数据密集型分布式部署环境下的实时应用系统的完美解决方案则为它的轻量高效在上一个档次。

Node.js 的一个特点是单线程，它无法同时进行多个任务，所以我们避免阻塞模式的 IO 处理方法便是不断的增加一些新的主线程，同时开通 Node.js 非阻塞模式的 IO 处理。这种处理方式会给 Node.js 带来在相对低系统资源耗用下的高性能与出众的负载能力，非常适合用作依赖其它 IO 资源的中间层服务^[12]。

3 系统的需求分析

3.1 功能需求

音乐播放器的最基本功能是播放音乐，而用户使用音乐播放器是为了满足听歌目的，这就需要到用户在音乐播放器的个人信息数据。为此我们通过不同的方面来描述本系统的控制需求，并且使用文字和例图来描述本系统的功能需求。本系统主要由两个部分构成，一个是播放器的登录/注册后台数据需求，一个是播放器的基本控制需求。

3.1.1 系统的注册/登录

在系统开启的时候，用户会直接进入到了首页，此时用户的浏览器会获取到页面的信息。在首页上面有一个登录页面的跳转链接，用户点击此链接，页面便会自动跳转到登录页面。

在登录页面，用户可以在两个输入框中输入帐号和密码，点击提交后页面可以将用户所输入的帐号和密码发送到服务器，服务器进行数据匹配，判断用户输入的账户存在与否之后判断用户输入的密码是是否与帐号所匹配。登录页面需要有一个注册页面的链接，当用户点击此链接时，页面将跳转到注册页面。

在注册页面，用户可以在三个输入框中输入帐号、密码、重复密码，点击提交后页面将用户两次所输入的密码进行匹配，判断是否相同，如果相同就将帐号和密码发送到服务器，服务器进行数据匹配，判断帐号是否存在，之后存储用户的账户和密码。从下图 3-1 可以看出播放器的注册/登录中信息发送和调用过程。

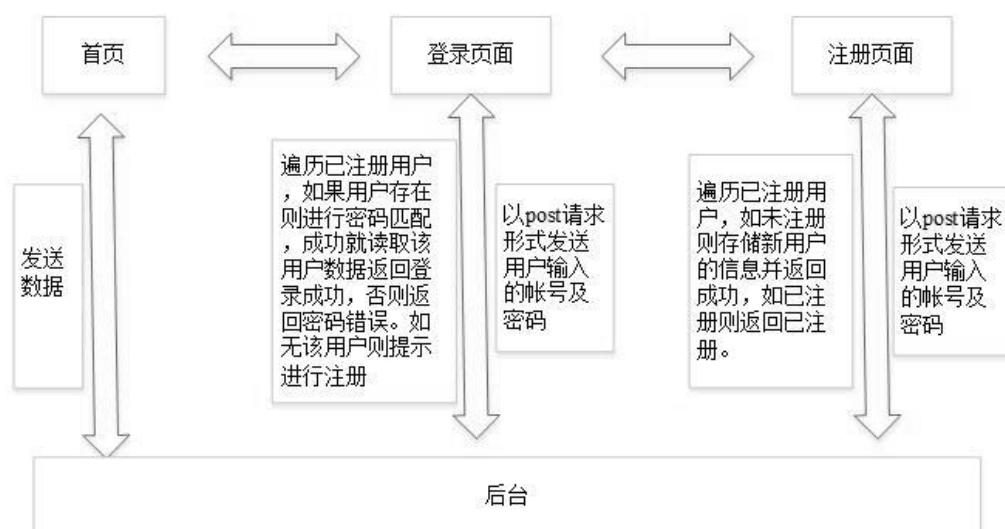


图 3-1 系统注册/登录的时序图

3.1.2 播放器的基本控制需求

播放器启动时，会自动播放上次退出时所播放的歌曲。如果用户想要进行更换歌曲，则需选择首页歌单、推荐歌单或者个人收藏歌单中的一首歌曲，对该歌曲进行单击。点击过后，页面会将用户所点击到的这首歌的信息发送到服务器，服务器到后台进行 mp3 数据提取，之后返回给服务器，服务器再将数据发送到播放器，最终播放器将播放此歌曲，并同时将该歌曲的总进度及已播放进度显示出来。

当用户想要在歌曲播放时进行暂停，则需点击“暂停”按钮。此时页面触发 JavaScript 的用户点击事件，从而使得播放器暂停，并且更改播放器控制面板上的播放图标。

当用户想要在歌曲暂停时进行播放，则需点击“播放”按钮。此时页面触发 JavaScript 的用户点击事件，从而使得播放器播放，并且更改播放器控制面板上的暂停图标。

在播放过程中点击“上一首”或“下一首”按钮，触发 JavaScript 的用户点击事件，对播放的歌曲按照当前的歌单和播放的方式进行上一首或下一首的播放。在暂停过程中点击“上一首”或“下一首”按钮同样可以切换歌曲，放开按钮后会使得播放器进行播放。

最后一个按钮则是播放方式按钮，页面的 JavaScript 通过用户的点击事件来为用户选择播放方式。默认为随机播放，通过点击次数的增加来改变方式，拥有单曲循环、歌单循环及随机播放三种方式。详情见图 3-2。



图 3-2 播放器的基本控制时序图

3.2 运行及其他需求分析

本系统的硬件运行条件：PC 机的内存大于 500MB、硬盘大于 500M、具有支持 HTML5 及 JavaScript 的浏览器。

本系统的软件运行条件：Window XP 操作系统或更高版本、HTML5 语言；Node.js、JavaScript 脚本。

本系统对浏览器的依赖性很高, 拥有很高的实时更新性、可维护性及高效率。它十分便于用于用户理解和操作。因为是基于 Web 的 WebApp, 所以在运行中是需要网络的。开发和维护过程中, 在多种矛盾和性能需求间作权衡, 并在经费、时间、技术、资源等限制条件下, 使得上述各方面最大限度的得到了满足。

4 系统设计

4.1 软件结构的设计

通过了我们系统的需求分析，我们开始进入系统的架构。本系统不仅需要完善的功能，还需要有便于用户操作的界面。因为本系统是基于 HTML5 所开发的 WebApp，服务器是由 Node.js 所做，而交互则是用 JavaScript 进行编写，为了实现方便用户操作的目标，做出如图 4-1 所示播放器系统整体流程图。

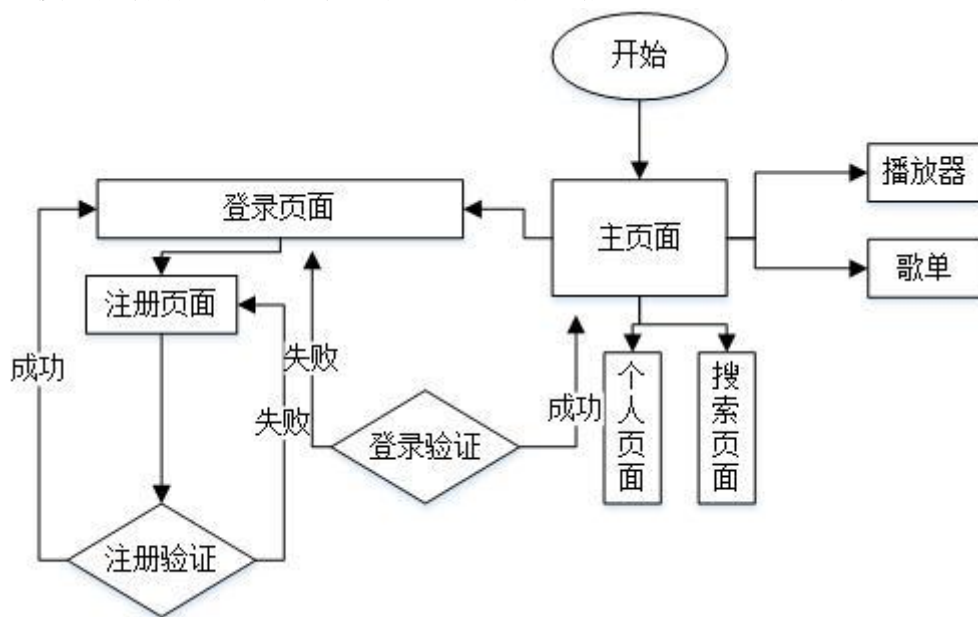


图 4-1 播放器系统流程图

所有的页面都通过 HTML5 的标签搭建，配合 CSS 样式进行精美的 UI 调整。在登录和注册页面，数据的传输与获取是采用了 Node.js 的 post 请求编写出来的。post 请求可以将用户在两个页面的输入框输入的内容发送到 Node.js 所搭建的服务器中，在服务器中，登录页面和注册页面的 post 请求所对应的接口进行数据接收。

对于 Node.js 来说，post 请求是目前主流的请求编写方式。主要是程序员进行接口的命名，然后在页面和服务器的程序文件中进行数据发送和数据接收的代码编写。利用这种方法，WebApp 的各个页面的各项数据可以进行互相传递、也可以将数据发送到服务器中。

登录页面和注册页面的数据可以发送到服务器，进行一定的操作后存储到后台，那么系统的注册登录页面就完成了。

在个人页面中，都由页面的 post 或者 get 请求从服务器的各自接口找寻到

后台中用户的信息文件，并且提取出用户的信息，显示在页面中，供用户查看。

在歌单页面中，通过用户点击歌曲触发 JavaScript 事件，将带有歌曲名字的标签的内容通过 post 请求发送到服务器。而接口在接收到这一些数据后，对后台进行访问，提取歌名 mp3 文件数据，发送到播放器，同时刷新页面

搜索页面的 post 请求节后在接收到用户在输入框中输入的数据后，将其发送到服务器的接口中。接口提取信息，进行模糊匹配，之后显示在输入框的提示栏中，并且在之后用户点击搜索后显示在搜索结果位置。

4.2 系统的客户端设计

本系统的页面主要有：登录、注册、主页面、各类歌单、个人页面、搜索页面、个人收藏页面、评论/留言界面。详细跳转链接如下：

本系统拥有一个主页面，上面为导航栏，第一个为主页面链接、第二个为用户信息界面链接、第三个为个人歌单页面链接、第四个为搜索页面链接、第五个为登录页面链接。而下面为音乐播放器及控制面板。详情如图 4-2 所示：



图 4-2 客户端首页

4.3 系统的后台设计

对于一个系统来说，存放数据的后台是十分重要的存在，本系统采用的是 Node.js 搭建的后台。在服务器中搭建了处理 post 请求、get 请求、处理上传文件、处理 cookie 和配置存储文件的构造函数。对于本系统而言，后台主要存放的是用户的帐号密码、用户的歌曲收藏记录、用户的留言内容、用户的个人信息及页面歌单的歌曲信息。数据流程图如图 4-3 所示：



图 4-3 数据流程图

4.3.1 用户信息存储

在注册页面，页面的 post 请求会将数据以 JSON 形式发送到服务器，服务器通过对应的接口接收数据，之后把数据转换为字符串的形式，将用户的注册信息进行遍历，如果不存在就可以将用户的帐号密码通过接口的写入函数，写入到后台 user 文件夹，以 txt 文件形式保存。

登录是服务器接收 JSON 数据之后，转换为字符串数据，然后匹配后台 user 文件夹里边存放用户帐号密码的 txt 文件的内容，成功匹配则提取用户的个人信息，返回登录成功的提示。

在个人页面中所填写的用户资料，是通过服务器将获取到的数据存储在后 user 文件夹中与用户账户名相对应的专属 txt 文件中，方便浏览和服务器提取。

用户的留言和评论则是通过页面 post 请求发送 JSON 数据，然后服务器对应的接口进行数据接收转换为字符串，之后将数据写入到后台的 questions 文件夹中，以当前时间加上随机数作为命名。在留言板页面上进行 post 请求，服务器到后台 questions 文件夹中提取所有存放数据的 txt，通过 JavaScript 脚本处理显示出来。

4.3.2 音乐信息的存储

用户的收藏歌曲记录是在用户点击歌曲收藏按钮后，页面会将用户所点击的特殊标签的内容（歌曲信息）提取出来，然后将歌曲信息发送到服务器，而后会在后台遍历文件夹，找寻或创建用户的个人文件夹，将歌曲信息存储到 txt 文件中，保存在用户的个人文件夹中。在歌曲收藏页面，后台会将用户个人文件夹中的歌曲信息发送到页面进行显示。

歌单页面在用户点击歌曲的时候 JavaScript 脚本会将特定的标签的内容（歌曲信息）提取出来，组成一个数组，然后通过 post 请求发送到服务器，然后写入到后台 newgedan 文件夹中的 musicst.txt 文件中，它会覆盖上一个 txt 文件，以便于播放器到提取歌曲信息，提供用户上一首/下一首切换。

最后便是歌曲文件，歌曲文件放置在服务器 www/mp3 文件下，文件的格式也大多为 mp3 格式，播放器通过提取歌曲名称、歌手名称等数据来匹配用户所点击播放的歌曲。

这样子的数据库，管理员可以通过修改所有数据存放的文件内容来修改一些

信息，所以在早期用户少量时较为方便，但遇到大量用户时便变得较为繁琐。

5 系统的实现

因为本系统为基于 HTML5 的 WebApp 开发，所以只要支持 HTML5 的浏览器中，打开本系统的首页网页，既可以链接到本系统的首页页面开始运行本系统。

5.1 用户的注册/登录设计

5.1.1 用户的登录

当用户运行本系统时，在主页面点击登录按钮，即可跳转到登录页面如图 5-1 所示。此时用户如果已经注册过本系统的账户，可以在登录界面输入用户名及密码，点击登录，服务器会将用户输入的帐号和密码发送至后台，Node.js 通过遍历后台存储用户信息的文件，与从服务器获取的数据进行匹配，如果用户名匹配上则进行密码匹配，当两者都匹配成功时返回一个 code 数据，提示用户“登录成功”，并提取用户账户存放到 cookie，跳转到主页面。如果密码匹配失败则返回一个 code 数据，提示用户“密码错误”，如果账户匹配失败则返回一个 code 数据，提示用户“用户不存在”，并提示用户进行注册。如果用户没有注册过帐号，则通过点击“注册帐号”跳转到注册页面进行注册。



图 5-1 登录页面

5.1.2 用户的注册

注册页面可以返回上一页，也可以直接返回主页面。它 UI 设计上拥有三个输入框和一个提交按钮如图 5-2 所示。当用户输入完账户和两次密码后按下“注册”按钮。此时页面的 JavaScript 将会进行两个密码的比较，如果相同则将输入发送至服务器，如果不同则提示用户进行重新输入。当服务器获取到注册页面提交的数据时，先遍历后台存储用户信息的文件，进行用户名匹配，如果匹配成功，则返回数据“账户已存在”，并提示重新注册。如果匹配失败，则返回数据“注册成功”并返回登录界面。



The image shows a mobile app registration page titled '帐号注册' (Account Registration). It features a back arrow on the top left and a home icon on the top right. The form consists of three input fields: '请填写昵称' (Please enter nickname), '请填写密码' (Please enter password), and '请再次填写密码' (Please enter password again). Below these is a blue '注册' (Register) button. At the bottom, there is a link that says '注册即同意《注册协议》' (Registration means you agree to the 'Registration Agreement').

图 5-2 注册页面

5.2 主页面

5.2.1 导航页面

主页面的导航栏主主要是五个链接，分别为：主页面、个人信息、个人收藏、搜索、登录。底部拥有一个音乐播放器，上面放置着音乐播放器的控制按钮，通过 JavaScript 可以让音乐播放器进行播放、暂停、上一首、下一首的控制，也可以改变歌单歌曲的播放方式。在播放器的底部头像内置了一个音乐播放页面，点击时会出现音乐播放的界面。而主页面的中间部分是一个显示区域，在这个区域首先显示的是推荐页面，供无目的听歌的用户查看目前各种歌曲的排行及本系统的推荐，而之后关于本系统的除注册、登录页面外，其他页面都将在这个区域进行跳转显示。界面如图 5-3 所示。



图 5-3 主页面

5.2.2 个人页面

本系统的个人页面如 5-4 所示，它分为个人信息页面和个人收藏歌单页面。当用户登录后，个人页面将向服务器发送一个获取用户信息的请求，而服务器则在后台提取用户个人信息，并且返回个人页面，显示用户的个人信息，包括用户的昵称、注册 IP、注册时间、头像等。如果用户在未登录情况下查看该页面则提示“系统异常，请重新登录”。个人收藏歌单页面里面有一个链接，当用户在不目的听歌的时候，可以将自己所喜欢的歌曲在播放界面进行点击收藏，收藏时系统会用 post 请求的方式将歌曲信息发送至服务器，服务器将其写入用户个人歌单文件中，然后通过个人收藏歌单页面的 post 请求将用户收藏的歌曲读取出来，方便用户的查看和重听。



图 5-4 个人页面

5.2.3 搜索页面

搜索页面如图 5-5 所示，它拥有一个输入框和一个提交按钮，当用户在输入框中输入文字，此时搜索页面的实时 post 请求会把用户所输入的文字发送至服务器，服务器再把一窜文字与后台收录的 mp3 歌曲进行一个正则运算的匹配，将含有用户输入的文字的歌曲显示在输入框的提示语。当用户按下搜索按钮时，搜索页面会将搜索框的内容送至服务器，服务器再把这一窜文字与后台收录的 mp3 歌曲进行一个正则运算的匹配，将含有用户输入的文字的歌曲显示在搜索结果下方。点击搜索出来的歌曲，搜索页面会将歌名发送至服务器，服务器则向播放器发送 mp3 歌曲进行播放。



图 5-5 搜索页面

5.2.4 留言/评论页面

当用户在播放页面点击评论按钮时，界面上将显示目前本系统所有用户的留言及评论。在本页面有一个留言按钮，点击之后页面会通过服务器向后台询问用户是否登录，如果没登陆则跳转到登录页面，如果已经登录会跳转到留言页面。用户可以在输入框中输入内容，输入完毕之后点击提交。此时服务器会将用户的输入写入后台的留言文件夹中，提示用户“留言成功”并返回留言页面，如果碰巧遇到不可抗拒因素导致写入失败，则提示用户“写入失败”。用户在留言页面也可以通过点击他人留言来进行对其他用户的回复，点击过后跳转到回复他人的页面，在输入框中输入内容，输入完毕之后点击提交。此时服务器会将用户的输入写入后台其他用户的留言文件中，提示用户“评论成功”并返回留言页面，如果碰巧遇到不可抗拒因素导致写入失败，则提示用户“写入失败”。界面详情如图 5-6 所示。



图 5-6 评论/留言页面

5.3 播放器

本系统为 WebApp 播放器，最为主要的是播放器，本系统的播放器分为两个部分。一个是在所有页面都会显示的底部音乐播放器控制面板，上面带有歌手的头像、歌曲名、歌手名、时间、进度等显示，也带有一套基本控制按钮：播放方式、上一首、暂停、播放、下一首。播放器通过 post 请求向服务器获取用户目前的歌单或者所点击的歌曲，然后利用这一套基本控制按钮通过 JavaScript 来对播放的歌曲进行操作，其中播放、暂停是通过 HTML5 本身自带的 audio 标签来进行，而上一首、下一首则是通过 JavaScript 更改一个全局变量来控制。播放方式默认为随机播放，这是因为在 JavaScript 中控制播放方式的变量初始值为 2，它的含义是当歌曲的播放进度等于歌曲的总长度时进行一个随机数控制的随机播放，用户可以通过点击的次数来更改播放方式，当这个变量取余为 0 时，播放方式为歌单循环播放，当这个变量取余为 1 时播放方式为单曲循环，当这个变量取余为 2 时播放方式为随机播放。

第二个部分是在用户点击歌手头像的时候，界面将会显示播放页面。页面上拥有歌手、歌名、带有 animal 旋转动画的头像、音量控制按钮、收藏按钮及评

论按钮。用户点击音量控制按钮时，JavaScript 会控制着 audio 标签改变音量大小。界面如图 5-7 所示。

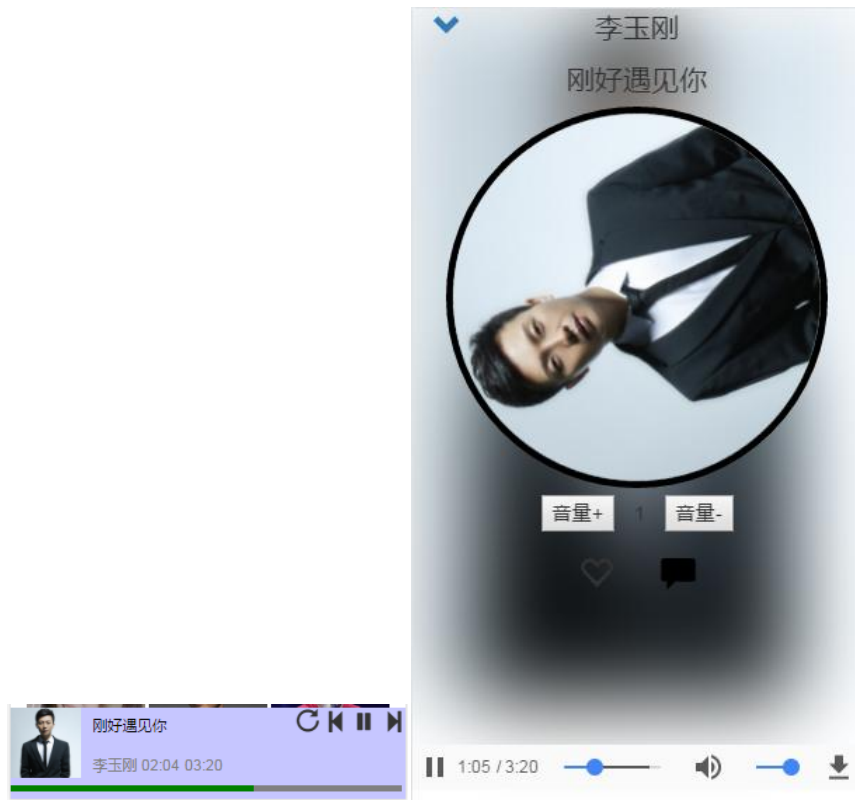


图 5-7 播放页面

6 系统的测试

6.1 测试方法

系统设计搭建完成后都需要进行测试，本系统也不例外。在测试中，我们通过邀请了同一个局域网中的 10 个不同主机 IP 的用户来进行测试。这 10 位用户首先进行了 UI 测试，测试他们对系统的界面满意程度及操作友好度感觉。之后这 10 位用户开始了动态系统测试，测试了系统的登录、注册、播放、搜索等功能。

6.2 测试结果

本系统的相关测试如下表 6-1 所示：

表 6-1 各项最终测试结果

用户 IP	登 录 /次	登 录 成 功 /次	注 册 /次	注 册 成 功 /次	播 放 /首	播 放 成 功 /首	搜 索 /次	搜 索 成 功 /次	UI 评分	操作 评分
192.168.1.126	5	5	2	2	12	12	5	5	4	3
192.168.1.129	12	12	4	4	10	10	10	10	7	6
192.168.1.127	4	4	4	4	20	20	2	2	7	7
192.168.1.145	5	5	3	3	5	5	11	11	8	7
192.168.1.136	10	10	6	6	11	11	8	8	6	5
192.168.1.142	7	7	2	2	3	3	13	13	6	6
192.168.1.112	9	9	1	1	7	7	13	13	9	8
192.168.1.123	15	15	15	15	15	15	15	15	7	2
192.168.1.131	8	8	5	5	20	20	13	13	3	8
192.168.1.144	9	9	3	3	13	13	3	3	5	6

(1) 10 位不同 IP 的各主机用户访问本系统，通过点击链接跳转到不同页面，给出个人对本系统的界面 UI 评价，然后点击系统的各种按钮按键，给出个人的操作体验评价，总体来说，用户们对于本系统的 UI 设计较为满意，但在操

作体验上较为中肯。

（2）功能的动态系统测试

1) 10 位不同 IP 的各主机用户访问本系统，进行登录/注册，每个测试用户进行了不同次数的登录/注册，经历了文件无法写入、文件命名错误等问题，通过观察 post 请求的数据发送和后台中用户信息文件的生成，修改个别错误的地方，达到每个用户都能登录为止。

2) 10 位不同 IP 的各主机用户访问本系统，5 位在登录的状态下，5 位在没有登录的状态下，通过点击不同的歌曲，测试是否能够播放及暂停、播放；点击上一首、下一首按钮，看是否可以切换歌曲；通过收听歌曲是否与用户所点击的歌曲一样及查看控制台所给的每项数据，在测试过程中修改 JavaScript 事件的编写，完善了播放功能。

3) 10 位不同 IP 的各主机用户访问本系统，5 位在登录的状态下，5 位在没有登录的状态下，通过在输入框中输入上搜索的关键字眼，后台收录的歌曲含有关键字眼的歌曲，在按下搜索按钮后大多数搜索没办法显示结果。通过测试及对代码的检查，发现是技术不够成熟，无法设计出完美的正则匹配。最后的更正方法是设计了输入框提示栏，当关键字眼输入后，含有关键字眼的所有歌曲将显示在提示栏中，通过点击来让其显示在搜索结果内。

结论

本章是对本系统的一个总结与展望。在经过漫长的一段开发时间，WebApp 音乐播放器诞生了，大体上是完成了任务书上的预定功能需求，但是由于经验的不够和技术的不足，本系统仍然存在各方面的不足。

在后台上，本系统使用 Node.js 搭建，对于数据的处理也使用 Node.js 替代了管理数据的“神器”，MySQL 数据库，这导致当遇到大量的访问产生大量用户数据时管理员的工作将变得十分复杂。在页面的搭建上，标签种类使用的过多，导致后续的后台数据获取变得较为麻烦。在 CSS 样式的编写中，对于标签的类名和 ID 没有进行统一的规划命名，导致 CSS 代码变得很冗长。

虽然有缺点，但是也有闪光点。在 CSS 编写时，做了自适应屏幕，使得 UI 界面较为漂亮。搭建的标签也因为使用的种类较多使得代码较为一目了然。因为后台使用的是 Node.js 搭建的，所以对于新产品，使用用户较少的本系统而言，后台的数据一目了然。并且本系统是基于 HTML5 的 WebApp，在功能模块上算是较为完善和多功能的，能够满足用户的有目的及无目的听歌。

在历时三个月完成了本系统及论文的撰写，在这一段时间内通过对系统进行分析、系统构建和设计。从一开始设计系统的懵懵懂懂、磕磕绊绊，到系统完成后的从容测试、尽力完善，这是一个成长的过程。

在导师的指导下，凭借着个人的能力独自完成本系统，当完成这个稚嫩的系统时，一种自豪油然而生。自己很清楚本系统的不足与闪光点，因为是一个较为简单的 WebApp 音乐播放器，不求它能被所有用户所接受，但是能够被部分接触过的用户所使用，那么它的价值就得到了体现。

未来，或许能够通过我的努力，用更加成熟的技术，更加完善的逻辑去完善、更新本系统，让它越来越能够被用户接收这是一个不小的目标。前方的路还很长，我希望我能和我所设计的本系统越走越远。

参考文献

- [1] 中国互联网络信息中心.中国互联网络发展状况统计报告[J].国家图书馆学刊,2016,(02):76.
- [2] 陈勇.WebApp 现状分析及展望[J].通信与信息技术,2012,(04):77-78.
- [3] 王晓晴.《音乐爱好者》杂志研究[D].南京:武汉音乐学院音乐系,2006.
- [4] 李璠,杨涛,任可任.HTML5 应用场景与未来展望[J].金融电子化,2016,(01):70-72.
- [5] 袁南星.基于 HTML5+JavaScript 的移动用户环境自适应[J].电脑与电信,2015,(12):44-46.
- [6] 杨俊,李艳梅.JavaScript 面向对象编程探析[J].办公自动化,2010,(08):22-24.
- [7] 王金龙,宋斌,丁锐.Node.js:一种新的 Web 应用构建技术[J].现代电子技术,2015,(06):70-73.
- [8] 蒋一鸣.基于简约风格的移动音乐 APP 界面设计研究[J].美与时代(上),2015,(03):111-113.
- [9] 李慧云.HTML5 技术与应用模式研究[J].期刊论文.2012,(46):330-325.
- [10] 徐雯.UI 设计与产品形象的关系研究[J].艺术科技,2016,(07):276-278.
- [11] 钟强.Node.JS 平台下 Web 前端架构的研究[J].无线互联科技,2013,(12):24-36.
- [12] Seonghoon Kang,Sukyoung Ryu.Formal specification of a JavaScript module system[J].ACM SIGPLAN Notices,2012,47(10).