



湖南工程职业技术学院

毕业设计

产品设计说明书

题 目 _____ 大乐音乐网站的设计与实现 _____

毕业设计类型 ☒ 产品设计 ☐ 工艺设计 ☐ 方案设计

专 业 名 称 _____ 软件技术 _____

所 属 班 级 _____ 2018173801 _____

学 生 姓 名 _____ 梁瀛丹 _____

学 生 学 号 _____ 201817380130 _____

指 导 老 师（学校） _____ 胡瑶 _____

指 导 老 师（企业） _____ 邹金富 _____

湖南工程职业技术学院教务处 制

目录

插图索引.....	III
附表索引.....	IV
第 1 章 项目概况.....	1
1.1. 设计背景.....	1
1.2. 设计意义.....	1
1.3. 设计思路.....	2
第 2 章 需求分析.....	3
2.1. 需求描述.....	3
2.2. 用例建模.....	3
2.3. 用例描述.....	4
2.3.1. 音乐管理用例描述.....	4
2.3.2. 歌手管理用例描述.....	5
2.3.3. 音乐类型用例描述.....	6
2.3.4. 用户管理用例描述.....	7
第 3 章 系统设计.....	8
3.1. 数据库设计.....	8
3.1.1. 概念模型.....	9
3.1.2. 逻辑模型.....	10
3.1.3. 物理模型.....	10
3.1.4. 数据库脚本.....	11
3.2. 系统架构设计.....	17
3.3. 模块设计.....	18
3.3.1. 音乐管理类图.....	18
3.3.2. 音乐管理操作流程.....	19
3.3.3. 歌手管理类图.....	20
3.3.4. 歌手管理操作流程.....	20
3.3.5. 音乐类型类图.....	21
3.3.6. 音乐类型操作流程.....	22
3.3.7. 用户管理类图.....	23
3.3.8. 用户管理操作流程.....	24
第 4 章 产品实现.....	26
4.1. 音乐管理模块实现.....	26
4.1.1. 界面设计.....	26
4.1.2. 功能实现.....	27
4.2. 歌手管理模块实现.....	30
4.2.1. 界面设计.....	30
4.2.2. 功能实现.....	31
4.3. 音乐类型模块实现.....	33
4.3.1. 界面设计.....	33
4.3.2. 功能实现.....	34
4.4. 用户管理模块实现.....	34
4.4.1. 界面设计.....	34
4.4.2. 功能实现.....	36
第 5 章 产品测试.....	37

5.1. 测试方法与策略.....	37
5.2. 音乐管理模块测试.....	37
5.3. 歌手管理模块测试.....	43
5.4. 音乐类型模块测试.....	45
5.5. 用户管理模块测试.....	46
第6章 结论.....	50
6.1. 毕业设计成果特点.....	50
6.2. 设计成果的实用价值或应用前景.....	50
6.3. 不足之处或遗留未予解决的问题.....	50

插图索引

图 2-1 大乐音乐网用例模型图.....	4
图 3-1 大乐音乐网概念模型图.....	9
图 3-2 大乐音乐网逻辑模型图.....	10
图 3-3 大乐音乐网物理模型图.....	11
图 3-4 大约音乐网架构设计图.....	18
图 3-5 音乐管理类图.....	19
图 3-6 音乐管理时序图.....	19
图 3-7 歌手管理类图.....	20
图 3-8 歌手管理时序图.....	21
图 3-9 音乐类型类图.....	22
图 3-10 音乐类型时序图.....	23
图 3-11 用户信息类图.....	24
图 3-12 用户管理时序图.....	25
图 4-1 音乐信息列表界面图.....	26
图 4-2 音乐信息修改界面图.....	27
图 4-3 音乐信息添加界面图.....	27
图 4-4 音乐信息主要代码图.....	29
图 4-5 歌手管理列表界面图.....	30
图 4-6 歌手修改界面图.....	30
图 4-7 歌手添加界面图.....	31
图 4-8 歌手管理模块主要代码图.....	32
图 4-9 音乐类型列表界面图.....	33
图 4-10 类型添加界面图.....	33
图 4-11 音乐类型模块主要代码图.....	34
图 4-12 用户管理列表界面图.....	35
图 4-13 用户添加界面图.....	35
图 4-14 用户详情界面图.....	36
图 4-15 用户管理模块主要代码图.....	36

附表索引

表 2-1 音乐管理用例描述表.....	4
表 2-2 歌手管理用例描述表.....	5
表 2-3 音乐类型用例描述表.....	6
表 2-4 用户管理用例描述表.....	7
表 3-1 UserInfo 表（用户信息表）	8
表 3-2 Manage 表（管理员信息表）	8
表 3-3 Singer 表（歌手信息表）	8
表 3-4 Music 表（音乐信息表）	8
表 3-5 MusicList 表（音乐收藏表）	9
表 3-6 MusicType 表（音乐类型表）	9
表 5-1 音乐管理模块测试用例表.....	37
表 5-2 歌手管理模块测试用例表.....	43
表 5-3 音乐类型模块测试用例表.....	45
表 5-4 用户管理模块测试用例表.....	46

第 1 章 项目概况

1.1. 设计背景

随着计算机产业的迅速发展，电子计算机已广泛的应用于信息管理、文字处理、辅助设计、辅助教学及人们的日常生活中。音乐检索系统主要是针对音乐检索的一系列相关工作的管理，本系统的建立使音乐检索更加规范化、系统化，查询手段更加便捷化。同时，音乐检索系统也是实现音乐检索现代化和信息化的重要内容。

近年来，随着听音乐人数的增多，人工书写的方式已经不能满足如此庞大的数据。为了更好的适应信息时代的高效性，一个利用计算机来实现音乐检索工作的系统将必然诞生。基于这一点，设计了一个音乐检索系统，用来就音乐检索进行管理，以便在最短的时间内，高效准确的完成整个检索过程。

基于对上述的认识，收集相关资料和数据，查阅有关文献及技术参数，对音乐检索的实际需求等方面进行了调研，发现目前所采用的手工记录的方法进行管理存在以下几个方面的弊端：对象范围广，数据存储不易；填写的数据多，且大量数据重复；不易存档，成千上万的信息和堆积如山的单据对管理人员来说是一个负担，需要大量的文档资料。鉴于以上种种原因，音乐检索的管理急需一种软件来代替手工操作来管理数据资料。

计算机技术在现代管理中的应用，使计算机成为领导者和管理人员应用现代技术的重要工具。计算机辅助音乐检索活动，可以极大地增强音乐爱好者采集、处理信息的能力。计算机系统能根据管理过程的变化情况，将原始数据、资料等进行加工、保存，管理人员可以在解决具体问题需要信息资料时，随时进行检索查询，了解整个音乐检索系统的动态情况，进行动态管理，从而有效的处理音乐检索的管理工作，实现音乐检索信息管理的自动化，提高效率。

1.2. 设计意义

大乐音乐网的开发目的是使音乐检索模式转变成信息管理，为音乐检索人员提供方便条件。对音乐检索的实际情况进行调研之后，进行详细的需求分析，对现有的管理模式进行改进，开发出一套新型的管理系统，从中领悟系统开发的思想，掌握系统开发的流程和方法。音乐检索系统的实施的现实意义：减少管理音乐检索的工作人员；检索人员可以随时浏览查看，而且更加直观；改变了以前手工记录的方式，电脑录入更加方便快捷；实现了音乐检索的计算机化。

1.3. 设计思路

本系统目前的主要对象是喜欢听音乐的用户。

本系统分为六大模块，分别是登录注册模块、音乐首页模块、音乐管理模块、歌手管理模块、音乐类型模块、用户管理模块。

登录注册模块：用于用户和管理员的登录。

音乐首页模块：查询不同音乐类型、不同歌手、搜索框搜索音乐、在线听音乐、下载音乐的功能。

音乐管理模块：对音乐进行增删查改的功能。

歌手管理模块：对歌手进行增删查改的功能。

音乐类型模块：对音乐类型进行增删查的功能。

用户管理模块：对用户进行增查的功能。

设计开发流程如下：

1. 需求分析：包括需求调研，系统功能，性能分析，业务流程分析。
2. 概要设计：包括系统架构设计，功能模块设计，数据库设计。
3. 详细设计：包括方法设计、方法的封装、数据结构设计、属性使用设计。
4. 系统实现：包括编写代码，系统单元测试。
5. 系统测试：对项目功能进行系统测试。

第 2 章 需求分析

2.1. 需求描述

系统中有多种角色，每种角色拥有不同的功能；管理员具有音乐管理、音乐类型、歌手管理、用户管理全部功能。

管理员通过权限操作对系统的权限值进行增删改查，通过菜单权限分配实现各菜单的访问权限；音乐管理对音乐的基本信息进行增删改查，实现音乐的添加，并上传音乐封面图片。管理员对用户进行增查，实现用户的添加，查看用户的基本信息。音乐类型对音乐的类型进行增删改查，并对音乐所属的类型进行修改，实现音乐下音乐类型的添加。歌手管理对歌手的基本信息进行增删改查，实现歌手的添加，并上传歌手的图片。

2.2. 用例建模

本系统中主要包括一类用户：管理员，其中可以对这一类用户说明下。
根据前面的分析，可以得大乐音乐网整个项目的总体用例图如下图 2-1 所示：

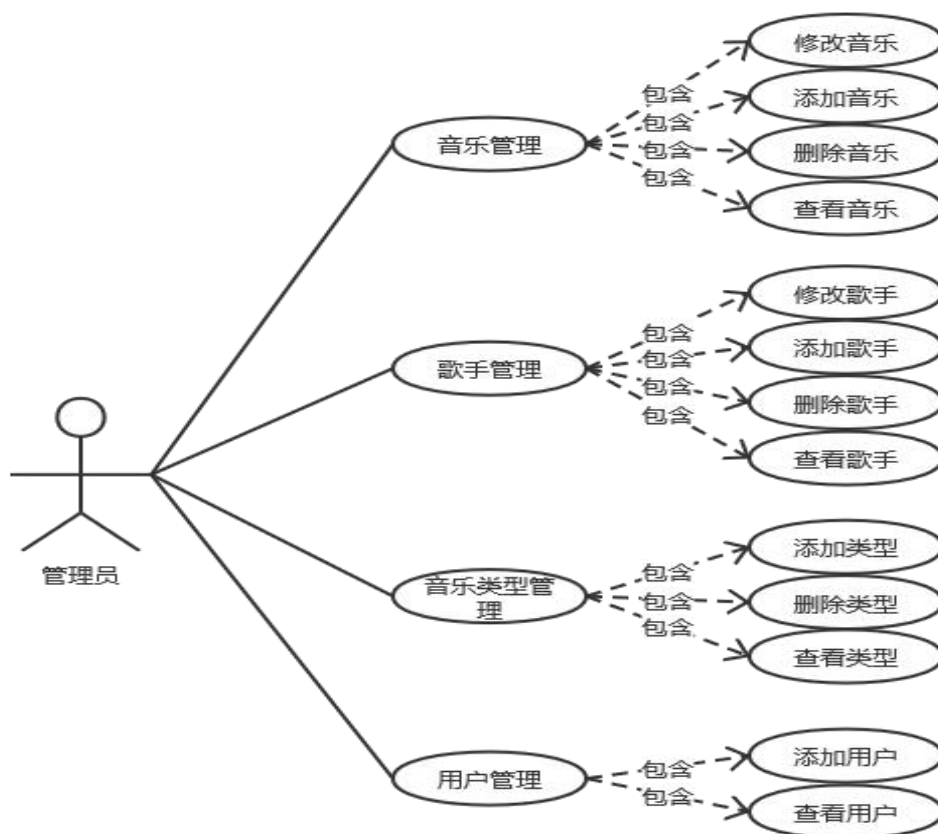


图 2-1 大乐音乐网用例模型图

2.3. 用例描述

2.3.1. 音乐管理用例描述

音乐管理：管理员登录后可以查看音乐网存在的所有音乐，可以新增或修改音乐信息。其具体的用例描述如下表 2-1 所示：

表 2-1 音乐管理用例描述表

用例编号	CASE01
用例名称	音乐管理
用例描述	系统管理员对后台数据的信息进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中新增了音乐信息或修改了音乐信息或删除了系统中已经存在的音乐或者查看音乐的详细信息。

活动步骤	<p>1-系统管理员登录系统</p> <p>2-系统显示音乐列表</p> <p>3-系统管理员点击【新增音乐】按钮，弹出添加音乐界面，系统管理员输入音乐信息，点击【保存】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的音乐列表。</p> <p>4-系统管理员在用户列表中选择了某个音乐信息，点击【修改】，弹出用户修改界面，显示音乐列表的当前信息，管理员更改音乐信息，点击【保存】，系统对输入的信息进行验证，将合法的信息保存，然后显示修改后的音乐列表。</p> <p>5-系统管理员在音乐列表搜索某个音乐信息，在输入框中输入搜索内容，点击【搜索】，弹出搜索结果，显示音乐的当前信息。</p> <p>6-系统管理员在音乐列表中选择某个音乐信息，点击【删除】，弹出删除提示框，当音乐确定删除后，系统将当前音乐信息删除，并返回到用音乐表，显示删除后的音乐列表。</p>
扩展点	无
数据格式	音乐管理（音乐 ID）

2.3.2. 歌手管理用例描述

歌手管理：管理员登录后可以查看全部歌手信息，可以新增或修改歌手信息或是删除选中的歌手信息。其具体的用例描述如下表 2-2 所示：

表 2-2 歌手管理用例描述表

用例编号	CASE02
用例名称	歌手管理
用例描述	系统管理员对歌手的信息进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中新增了歌手或修改了歌手信息或删除了系统中已经存在的歌手或查看全部歌手的信息。
活动步骤	<p>1-系统管理员登录系统</p> <p>2-系统显示歌手列表</p> <p>3-系统管理员点击【新增歌手】按钮，弹出新建歌手界面，系统管理员输入歌手信息，点击【保存】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的歌手列表。</p> <p>4-系统管理员在歌手列表中选择某个歌手，点击【修改】，弹出歌手</p>

	<p>修改界面，显示歌手的当前信息，管理员更改歌手信息，点击【保存】，系统对输入的信息进行验证，将合法的信息保存，然后显示修改后的歌手列表。</p> <p>5-系统管理员在歌手列表搜索某个歌手信息，在输入框中输入搜索内容，点击【搜索】，弹出搜索结果，显示歌手的当前信息。</p> <p>6-系统管理员在歌手列表中选择某个歌手，点击【删除】，弹出删除提示框，当歌手确定删除后，系统将当前歌手信息删除，并返回到用歌手表，显示删除后的歌手列表。</p>
扩展点	无
数据格式	歌手管理（歌手 ID）

2.3.3. 音乐类型用例描述

音乐类型：管理员登录后可以查看全部音乐类型，可以新增或修改音乐类型或是删除选中的音乐类型。其具体的用例描述如下表 2-3 所示：

表 2-3 音乐类型用例描述表

用例编号	CASE03
用例名称	歌手管理
用例描述	系统管理员对音乐的类型进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中新增了音乐类型或修改了音乐类型或删除了系统中已经存在的音乐类型或查看全部音乐的类型。
活动步骤	<p>1-系统管理员登录系统</p> <p>2-系统显示音乐类型列表</p> <p>3-系统管理员点击【新增类型】按钮，弹出新建类型界面，系统管理员输入音乐类型信息，点击【保存】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的音乐类型。</p> <p>4-系统管理员在类型列表搜索某个音乐类型，在输入框中输入搜索内容，点击【搜索】，弹出搜索结果，显示音乐类型的当前信息。</p> <p>5-系统管理员在类型列表中选择某个音乐类型，点击【删除】，弹出删除提示框，当类型确定删除后，系统将当前音乐类型删除，并返回到用类型列表，显示删除后的音乐类型。</p>

扩展点	无
数据格式	音乐类型（音乐类型 ID）

2.3.4. 用户管理用例描述

用户管理：管理员登录后可以查看全部用户信息，可以新增或查看用户信息。其具体的用例描述如下表 2-4 所示：

表 2-4 用户管理用例描述表

用例编号	CASE04
用例名称	用户管理
用例描述	系统管理员对用户的信息进行 CRUD 操作
前置条件	系统管理员成功登录系统
后置条件	系统管理员在系统中新增了用户或查看全部用户的信息。
活动步骤	<p>1-系统管理员登录系统</p> <p>2-系统显示用户列表</p> <p>3-系统管理员点击【新增类型】按钮，弹出新建用户界面，系统管理员输入用户信息，点击【保存】按钮，系统对输入的信息进行验证，将合法的信息保存，然后显示新增后的用户信息。</p> <p>4-系统管理员在类型列表搜索某个用户信息，在输入框中输入搜索内容，点击【搜索】，弹出搜索结果，显示用户的当前信息。</p> <p>5-系统管理员在用户列表中选择用户信息，点击【查看】，显示用户的当前信息，点击【关闭】按钮关闭显示窗口。</p>
扩展点	无
数据格式	用户管理（用户 ID）

第3章 系统设计

根据大乐音乐网的需求分析结果，本章主要设计了大乐音乐网的数据库设计，以数据字典的形式一一罗列出表自身的组成结构，并使用 PowerDesigner 工具设计了概念模型、逻辑模型和物理模型，客观清晰的展示了表与表之间的关系；本系统采用 MVC 架构设计模型、划分了系统的功能模块使用 UML 类图和 UML 时序图详细描述了各功能的设计思路。

3.1. 数据库设计

大乐音乐网的数据库共设计了六张数据表，分别为用户信息表、管理员信息表、歌手信息表、音乐信息表、音乐类型信息表、音乐收藏信息表。各表的具体结构如下表 3-1 到 3-6 所示：

表 3-1 User Info 表（用户信息表）

字段名	中文名	类别	长度	允许为空	主键
UserID	用户编号	int			Y
UserName	用户名	nvarchar	50		
UserPwd	用户密码	nvarchar	50		
UserPhone	用户电话	nvarchar	50		
UserEmail	用户邮箱	nvarchar	50		
IsDelete	是否存在	bit			

表 3-2 Manage 表（管理员信息表）

字段名	中文名	类别	长度	允许为空	主键
ManageID	管理员编号	int			Y
ManageAccount	管理员账号	nvarchar	50		
ManagePwd	管理员密码	nvarchar	50		
IsDelete	是否存在	bit			

表 3-3 Singer 表（歌手信息表）

字段名	中文名	类别	长度	允许为空	主键
SingerID	歌手编号	int			Y
SingerName	歌手姓名	nvarchar	50		
SingerType	歌手类型	nvarchar	50		
SingerImage	歌手图片	nvarchar	50		
IsDelete	是否存在	bit			

表 3-4 Music 表（音乐信息表）

字段名	中文名	类别	长度	允许为空	主键
MusicID	音乐编号	int			Y
SingerID	歌手编号	int			

TypeID	音乐类型编号	int			
MusicImage	音乐图片	nvarchar	300		
MusicName	音乐名称	nvarchar	50		
MusicCountry	音乐国家	nvarchar	50		
MusicDate	发布日期	date			
MusicFiles	音乐文件	nvarchar	300		
IsDelete	是否存在	bit			

表 3-5 MusicList 表（音乐收藏表）

字段名	中文名	类别	长度	允许为空	主键
ListID	音乐收藏表编号	int			Y
MusicID	音乐编号	int			
UserID	用户编号编号	int			
IsDelete	是否存在	bit			

表 3-6 MusicType 表（音乐类型表）

字段名	中文名	类别	长度	允许为空	主键
TypeID	音乐类型编号	int			Y
TypeName	音乐类型名称	nvarchar	50		
IsDelete	是否存在	bit			

大乐音乐网的用户模块和管理员模块的设计与实现主要使用到了上表的 3-3, 3-4, 3-5, 3-6。其中 3-4 表主要用于用户模块, 3-3, 3-4, 3-6 表主要用于管理员模块。

3.1.1. 概念模型

根据前文的数据字典的描述, 通过使用 PowerDesigner 工具进一步设计了数据库的概念模型结构, 展现出各表的自身结构以及各表与表之间的关联。大乐音乐网的概念模型如下图 3-1 所示:

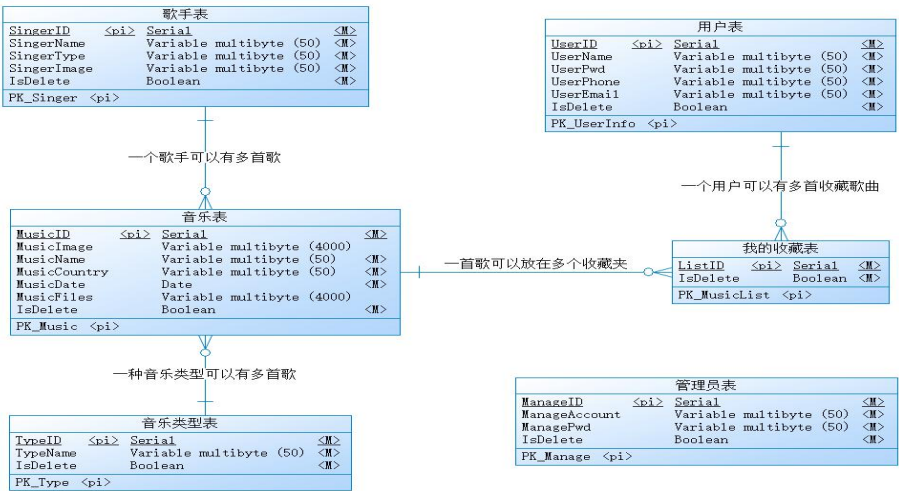


图 3-1 大乐音乐网概念模型图

3.1.2. 逻辑模型

根据前文的数据字典的描述，通过使用 PowerDesigner 工具进一步设计了数据库的概念模型结构，展现出各表的自身结构以及各表与表之间的关联。

大乐音乐网逻辑模块的的概念模型如下图 3-2 所示：

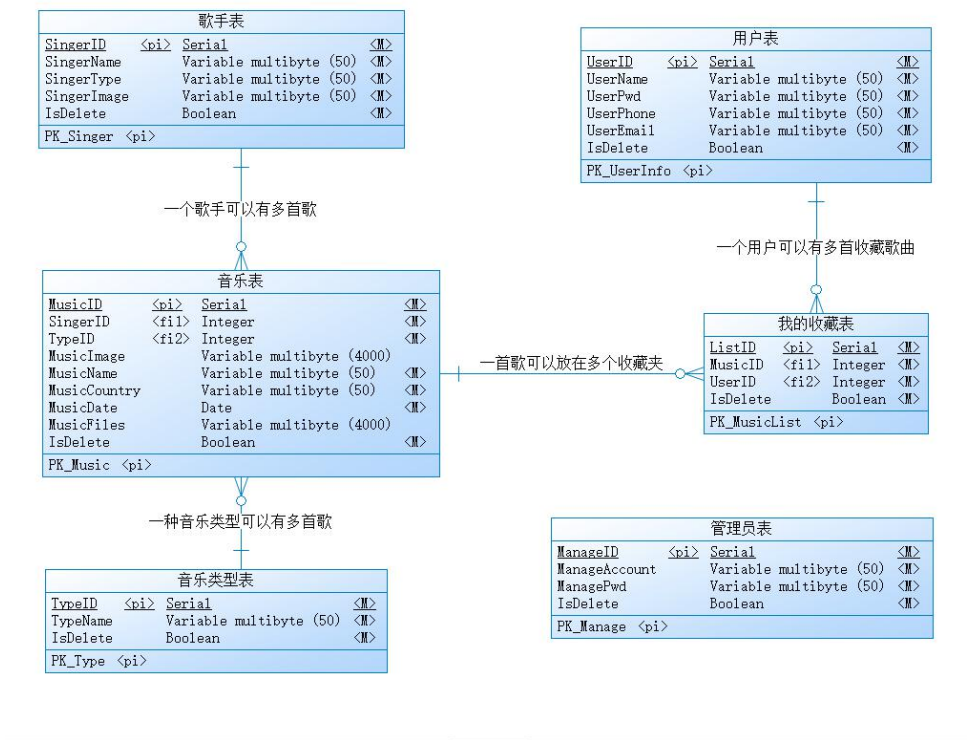


图 3-2 大乐音乐网逻辑模型图

3.1.3. 物理模型

利用 PowerDesigner 工具，在上节的逻辑模型基础上考虑各种具体的技术实现因素，进行数据库体系结构设计，真正实现数据在数据库中的存放。大乐音乐网的物理模型如下图 3-3 所示：

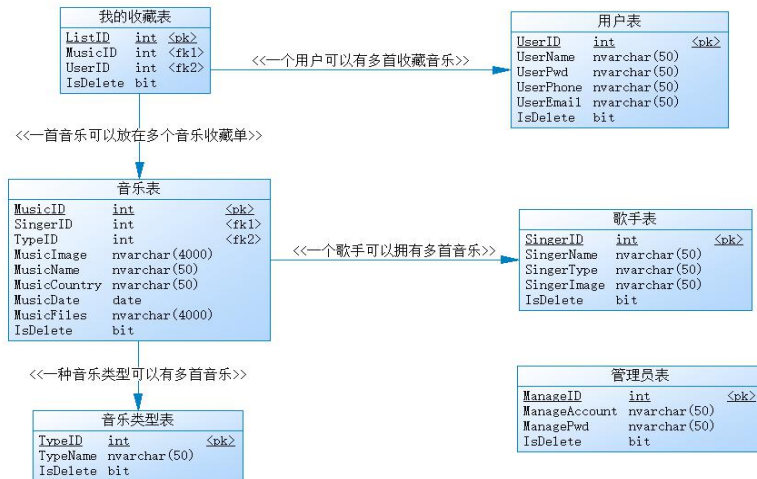


图 3-3 大乐音乐网物理模型图

3.1.4. 数据库脚本

```

USE [DaYueDB]
GO
/***** Object:  Table [dbo].[Manage]      Script Date: 2020/9/9 14:34:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Manage](
    [ManageID] [int] IDENTITY(1,1) NOT NULL,
    [ManageAccount] [nvarchar](50) NOT NULL,
    [ManagePwd] [nvarchar](50) NOT NULL,
    [IsDelete] [bit] NOT NULL,
    CONSTRAINT [PK_Manage] PRIMARY KEY CLUSTERED
(
    [ManageID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object:  Table [dbo].[Music]      Script Date: 2020/9/9 14:34:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
  
```



```

CREATE TABLE [dbo].[Music](
    [MusicID] [int] IDENTITY(1,1) NOT NULL,
    [SingerID] [int] NOT NULL,
    [TypeID] [int] NOT NULL,
    [MusicImage] [nvarchar](4000) NULL,
    [MusicName] [nvarchar](50) NOT NULL,
    [MusicCountry] [nvarchar](50) NOT NULL,
    [MusicDate] [date] NOT NULL,
    [MusicFiles] [nvarchar](4000) NULL,
    [IsDelete] [bit] NOT NULL,
    CONSTRAINT [PK_Music] PRIMARY KEY CLUSTERED
(
    [MusicID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object:  Table [dbo].[MusicList]      Script Date: 2020/9/9 14:34:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[MusicList](
    [ListID] [int] IDENTITY(1,1) NOT NULL,
    [MusicID] [int] NOT NULL,
    [UserID] [int] NOT NULL,
    [IsDelete] [bit] NOT NULL,
    CONSTRAINT [PK_MusicList] PRIMARY KEY CLUSTERED
(
    [ListID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object:  Table [dbo].[MusicType]      Script Date: 2020/9/9 14:34:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[MusicType](
    [TypeID] [int] IDENTITY(1,1) NOT NULL,
    [TypeName] [nvarchar](50) NOT NULL,
    [IsDelete] [bit] NOT NULL,
    CONSTRAINT [PK_Type] PRIMARY KEY CLUSTERED

```

```

(
    [TypeID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object:  Table [dbo].[Singer]      Script Date: 2020/9/9 14:34:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Singer](
    [SingerID] [int] IDENTITY(1,1) NOT NULL,
    [SingerName] [nvarchar](50) NOT NULL,
    [SingerType] [nvarchar](50) NOT NULL,
    [SingerImage] [nvarchar](50) NOT NULL,
    [IsDelete] [bit] NOT NULL,
    CONSTRAINT [PK_Singer] PRIMARY KEY CLUSTERED
(
    [SingerID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
/***** Object:  Table [dbo].[UserInfo]      Script Date: 2020/9/9 14:34:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[UserInfo](
    [UserID] [int] IDENTITY(1,1) NOT NULL,
    [UserName] [nvarchar](50) NOT NULL,
    [UserPwd] [nvarchar](50) NOT NULL,
    [UserPhone] [nvarchar](50) NOT NULL,
    [UserEmail] [nvarchar](50) NOT NULL,
    [IsDelete] [bit] NOT NULL,
    CONSTRAINT [PK_UserInfo] PRIMARY KEY CLUSTERED
(
    [UserID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

```

```
SET IDENTITY_INSERT [dbo].[Manage] ON
```

```
INSERT [dbo].[Manage] ([ManageID], [ManageAccount], [ManagePwd], [IsDelete]) VALUES (1, N'admin', N'lyd123', 1)
```

```
SET IDENTITY_INSERT [dbo].[Manage] OFF
```

```
SET IDENTITY_INSERT [dbo].[Music] ON
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (1, 1, 4, N'漂洋过海来看你.jpg', N'漂洋过海来看你', N'中国', CAST(0x543B0B00 AS Date), N'周深 - 漂洋过海来看你.mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (2, 5, 4, N'躲起来.jpg', N'躲起来', N'中国', CAST(0x043F0B00 AS Date), N'郁可唯 - 躲起来 (Live).mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (3, 2, 2, N'3.jpg', N'一吻天荒', N'中国', CAST(0xE2350B00 AS Date), N'胡歌 - 一吻天荒(电视剧《轩辕剑》片头曲).mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (4, 6, 4, N'咚巴拉.jpg', N'咚巴拉', N'中国', CAST(0x8A3E0B00 AS Date), N'萨顶顶 - 咚巴拉.mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (5, 7, 3, N'5.jpg', N'奇妙能力歌', N'中国', CAST(0x9D390B00 AS Date), N'陈粒 - 奇妙能力歌.mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (6, 3, 3, N'6.jpg', N'安河桥', N'中国', CAST(0x78370B00 AS Date), N'宋冬野 - 安和桥.mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (7, 4, 6, N'喜欢你.jpg', N'喜欢你', N'中国', CAST(0xFE3C0B00 AS Date), N'毛不易 - 喜欢你 (Live).mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (8, 8, 4, N'天使的指纹.jpg', N'天使的指纹', N'新加坡', CAST(0x3F2E0B00 AS Date), N'孙燕姿 - 天使的指纹.mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (9, 9, 5, N'9.jpg', N'寄明月', N'中国', CAST(0x423D0B00 AS Date), N'SING 女团 - 寄明月.mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (10, 10, 6, N'张三的歌.jpg', N'张三的歌', N'中国', CAST(0x76310B00 AS Date), N'青鸟飞鱼 - 张三的歌.mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (18, 2, 2, N'光棍.jpg', N'光棍', N'中国', CAST(0xCA340B00 AS Date), N'胡歌 - 光棍.mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (19, 15, 1, N'月夜.jpg', N'月夜', N'中国', CAST(0x46410B00 AS Date), N'双笙 _ 妖扬 - 月夜.mp3', 1)
```

```
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName],
```

```

[MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (20, 16, 2, N'你好灰姑娘.jpg', N'你好灰姑娘', N'中国', CAST(0x4E410B00 AS Date), N'胡夏 - 你好灰姑娘.mp3', 1)
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (21, 1, 7, N'9.jpg', N'飒小姐', N'中国', CAST(0x53410B00 AS Date), N'火箭少女 101 - 飒小姐 (Live).mp3', 1)
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (22, 18, 8, N'欧若拉.jpg', N'欧若拉', N'中国', CAST(0x8B410B00 AS Date), N'欧若拉 (Live) - 张韶涵.mp3', 0)
INSERT [dbo].[Music] ([MusicID], [SingerID], [TypeID], [MusicImage], [MusicName], [MusicCountry], [MusicDate], [MusicFiles], [IsDelete]) VALUES (23, 19, 9, N'躲起来.jpg', N'飒小姐', N'中国', CAST(0x8B410B00 AS Date), N'火箭少女 101 - 飒小姐 (Live).mp3', 1)
SET IDENTITY_INSERT [dbo].[Music] OFF
SET IDENTITY_INSERT [dbo].[MusicList] ON

INSERT [dbo].[MusicList] ([ListID], [MusicID], [UserID], [IsDelete]) VALUES (2, 20, 1, 0)
INSERT [dbo].[MusicList] ([ListID], [MusicID], [UserID], [IsDelete]) VALUES (6, 1, 1, 1)
INSERT [dbo].[MusicList] ([ListID], [MusicID], [UserID], [IsDelete]) VALUES (7, 19, 1, 1)
INSERT [dbo].[MusicList] ([ListID], [MusicID], [UserID], [IsDelete]) VALUES (8, 18, 1, 1)
SET IDENTITY_INSERT [dbo].[MusicList] OFF
SET IDENTITY_INSERT [dbo].[MusicType] ON

INSERT [dbo].[MusicType] ([TypeID], [TypeName], [IsDelete]) VALUES (1, N'古风', 1)
INSERT [dbo].[MusicType] ([TypeID], [TypeName], [IsDelete]) VALUES (2, N'影视', 1)
INSERT [dbo].[MusicType] ([TypeID], [TypeName], [IsDelete]) VALUES (3, N'民谣', 1)
INSERT [dbo].[MusicType] ([TypeID], [TypeName], [IsDelete]) VALUES (4, N'流行', 1)
INSERT [dbo].[MusicType] ([TypeID], [TypeName], [IsDelete]) VALUES (5, N'电子', 1)
INSERT [dbo].[MusicType] ([TypeID], [TypeName], [IsDelete]) VALUES (6, N'抒情', 1)
INSERT [dbo].[MusicType] ([TypeID], [TypeName], [IsDelete]) VALUES (7, N'乡村', 1)
INSERT [dbo].[MusicType] ([TypeID], [TypeName], [IsDelete]) VALUES (8, N'伤感', 0)
INSERT [dbo].[MusicType] ([TypeID], [TypeName], [IsDelete]) VALUES (9, N'女团', 0)
SET IDENTITY_INSERT [dbo].[MusicType] OFF
SET IDENTITY_INSERT [dbo].[Singer] ON

INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete]) VALUES (1, N'周深', N'1', N'周深.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete]) VALUES (2, N'胡歌', N'1', N'胡歌.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete]) VALUES (3, N'宋东野', N'1', N'宋东野.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete]) VALUES (4, N'毛不易', N'1', N'毛不易.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete]) VALUES (5, N'郁可唯', N'0', N'郁可唯.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete]) VALUES (6, N'萨顶顶', N'0', N'萨顶顶.jpg', 1)

```

```

INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete])
VALUES (7, N'陈粒', N'0', N'陈粒.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete])
VALUES (8, N'孙燕姿', N'0', N'孙燕姿.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete])
VALUES (9, N'SING', N'2', N'SING.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete])
VALUES (10, N'青鸟飞鱼', N'2', N'青鸟飞鱼.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete])
VALUES (14, N'花粥', N'0', N'花粥.jpg', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete])
VALUES (15, N'双笙', N'0', N'双笙.png', 1)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete])
VALUES (16, N'胡夏', N'1', N'胡夏.png', 0)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete])
VALUES (18, N'张韶涵', N'2', N'张韶涵.jpg', 0)
INSERT [dbo].[Singer] ([SingerID], [SingerName], [SingerType], [SingerImage], [IsDelete])
VALUES (19, N'火箭少女', N'2', N'火箭少女 101.jpg', 0)
SET IDENTITY_INSERT [dbo].[Singer] OFF
SET IDENTITY_INSERT [dbo].[UserInfo] ON

INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (1, N'丁禹兮', N'lyd123', N'13874457739', N'1046471532@qq.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (2, N'赵露思', N'lyd123', N'15115192260', N'1046471532@163.com', 1)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (4, N'陈星旭', N'lyd123', N'13874491394', N'132133@163.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (8, N'xiaoding', N'lyd123', N'13874457739', N'1046471532@qq.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (9, N'小一', N'lyd123', N'13815192260', N'1046471532@163.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (10, N'小二', N'lyd123', N'13815193360', N'1046471532@163.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (11, N'小丹', N'lyd123', N'13815192260', N'1046471532@163.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (12, N'小四', N'lyd123', N'13815192260', N'1046471532@163.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (13, N'小五', N'lyd123', N'15115192340', N'1046471532@163.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (14, N'小六 liu', N'lyd123', N'13874466639', N'1046471532@163.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (15, N'小七', N'lyd123', N'13815192260', N'1046471532@163.com', 0)
INSERT [dbo].[UserInfo] ([UserID], [UserName], [UserPwd], [UserPhone], [UserEmail],
[IsDelete]) VALUES (16, N'小八 ba', N'123456', N'13874465539', N'1046471532@163.com',
0)

```

```

SET IDENTITY_INSERT [dbo].[UserInfo] OFF
ALTER TABLE [dbo].[Music] WITH CHECK ADD CONSTRAINT [FK_Music_Singer]
FOREIGN KEY([SingerID])
REFERENCES [dbo].[Singer] ([SingerID])
GO
ALTER TABLE [dbo].[Music] CHECK CONSTRAINT [FK_Music_Singer]
GO
ALTER TABLE [dbo].[Music] WITH CHECK ADD CONSTRAINT [FK_Music_Type]
FOREIGN KEY([TypeID])
REFERENCES [dbo].[MusicType] ([TypeID])
GO
ALTER TABLE [dbo].[Music] CHECK CONSTRAINT [FK_Music_Type]
GO
ALTER TABLE [dbo].[MusicList] WITH CHECK ADD CONSTRAINT
[FK_MusicList_Music] FOREIGN KEY([MusicID])
REFERENCES [dbo].[Music] ([MusicID])
GO
ALTER TABLE [dbo].[MusicList] CHECK CONSTRAINT [FK_MusicList_Music]
GO
ALTER TABLE [dbo].[MusicList] WITH CHECK ADD CONSTRAINT
[FK_MusicList_UserInfo] FOREIGN KEY([UserID])
REFERENCES [dbo].[UserInfo] ([UserID])
GO
ALTER TABLE [dbo].[MusicList] CHECK CONSTRAINT [FK_MusicList_UserInfo]
GO

```

3.2. 系统架构设计

MVC 模型大乐音乐网使用 MVC 设计模式进行设计，有利于把模型数据与视图层实现有效分离，设置较为灵活的用户页面，提高用户体验效果。

MVC 设计模式包括业务模型、用户界面、控制器三层，MVC 是一种使用 MVC (Model View Controller 模型-视图-控制器) 设计创建 Web 应用程序的模式：Model (模型) 表示应用程序核心 (比如数据库记录列表)。View (视图) 显示数据 (数据库记录)。Controller (控制器) 处理输入 (写入数据库记录)。MVC 模式同时提供了对 HTML、CSS 和 JavaScript 的完全控制。Model (模型) 是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。View (视图) 是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。Controller (控制器) 是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。MVC 分层有助于管理复杂的应用程序，因为您可以在一个时间内专门关注一个方面。例如，您可以在不依赖业

业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。MVC 分层同时也简化了分组开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。MVC 模型结构如图 3-4 所示：

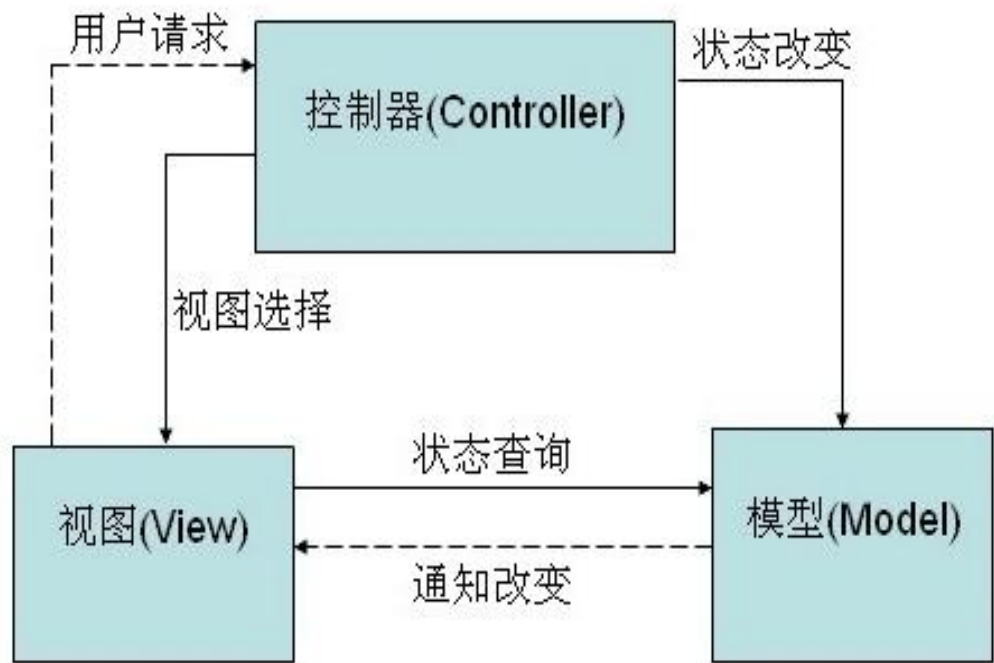


图 3-4 大约音乐网架构设计图

3.3. 模块设计

3.3.1. 音乐管理类图

音乐管理的类图如下图 3-5 所示。各类具体说明如下：

(1) ManageController 类：ManageController 类完成对音乐信息的增删查改操作，位于该项目目录 DaleMusic/Controllers/ManageController 处。方法包括音乐查询方法 (MusicManage())、音乐添加方法 (MusicCreat())、音乐修改方法 (MusicDetails())、音乐删除方法 (MusicDelete())、恢复音乐方法 (MusicIsDelete())。

(2) Music 类：Music 类音乐基本信息的类，位于该项目目录 Models/Music 处。属性包括音乐编号 (MusicID)、歌手编号 (SingerID)、音乐类型编号 (TypeID)、音乐图片 (MusicImage)、音乐名称 (MusicName)、音乐国家 (MusicCountry)、音乐发布时间 (MusicDate)、音乐文件 (MusicFiles)、是否存在 (IsDelete)。

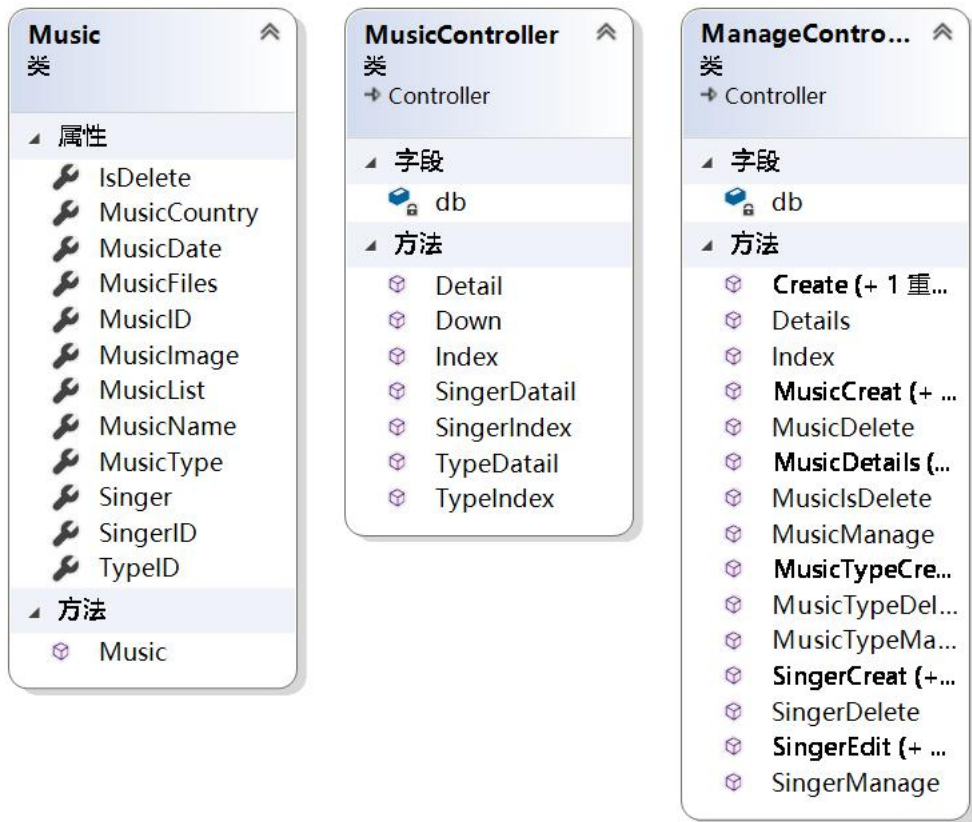


图 3-5 音乐管理类图

3.3.2. 音乐管理操作流程

音乐管理模块主要完成对音乐基本信息的增删查改功能,其操作流程如图 3-6 音乐管理时序图所示:

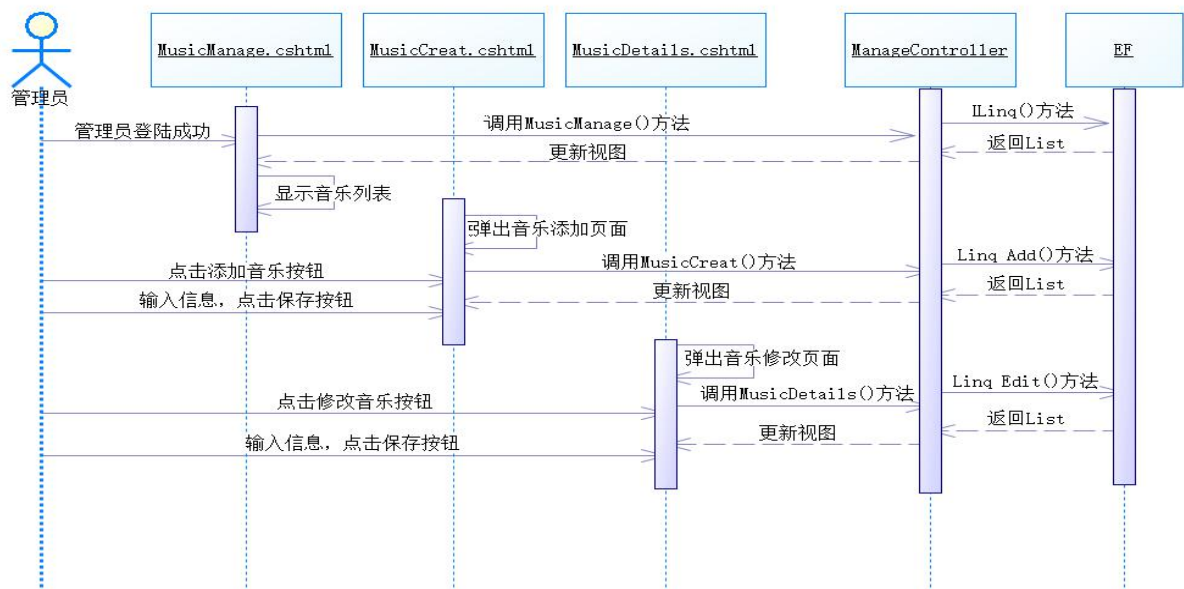


图 3-6 音乐管理时序图

3.3.3. 歌手管理类图

歌手管理的类图如下图 3-7 所示。各类具体说明如下：

(1) ManageController 类：ManageController 类完成对歌手信息的增删查改操作，位于该项目目录 DaleMusic/Controllers/ManageController 处。方法包括歌手查询方法 (SingerManage())、歌手添加方法 (SingerCreat())、歌手修改方法 (SingerEdit())、歌手删除方法 (SingerDelete())。

(2) Singer 类：Singer 类歌手基本信息的类，位于该项目目录 Models/Singer 处。属性包括音乐编号 (SingerID)、歌手名称 (SingerName)、歌手类型 (SingerType)、歌手图片 (SingerImage)、是否存在 (IsDelete)。

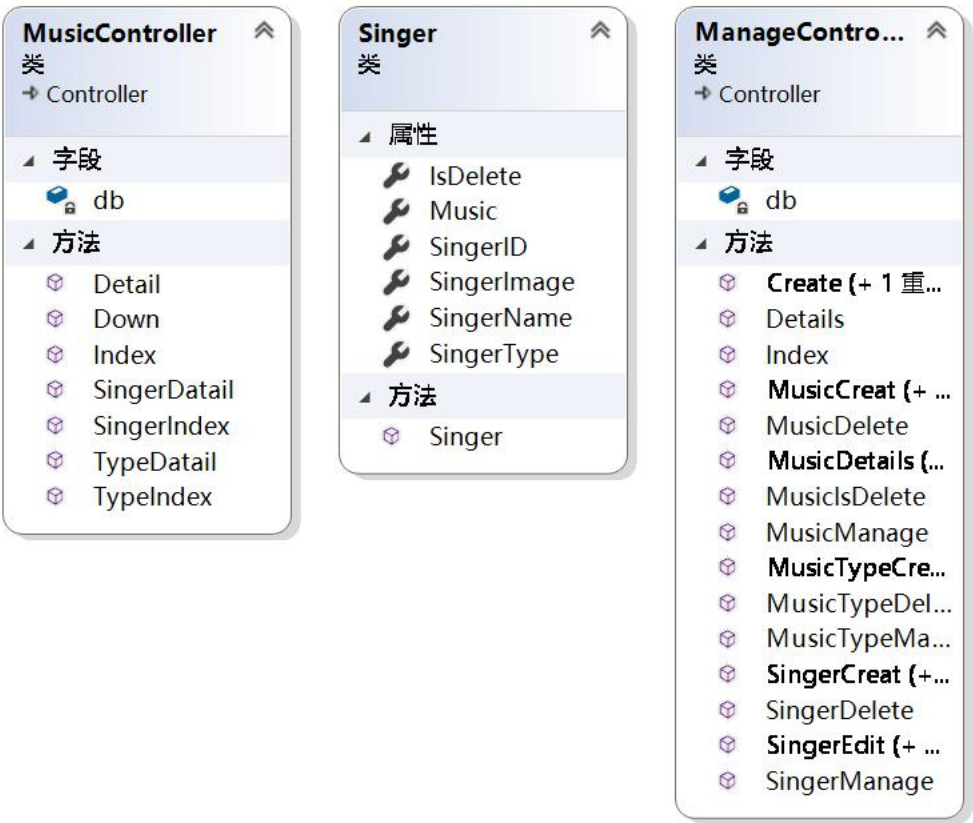


图 3-7 歌手管理类图

3.3.4. 歌手管理操作流程

歌手管理模块主要完成对歌手基本信息的增删查改功能，其操作流程如图 3-8 教歌手管理时序图所示：

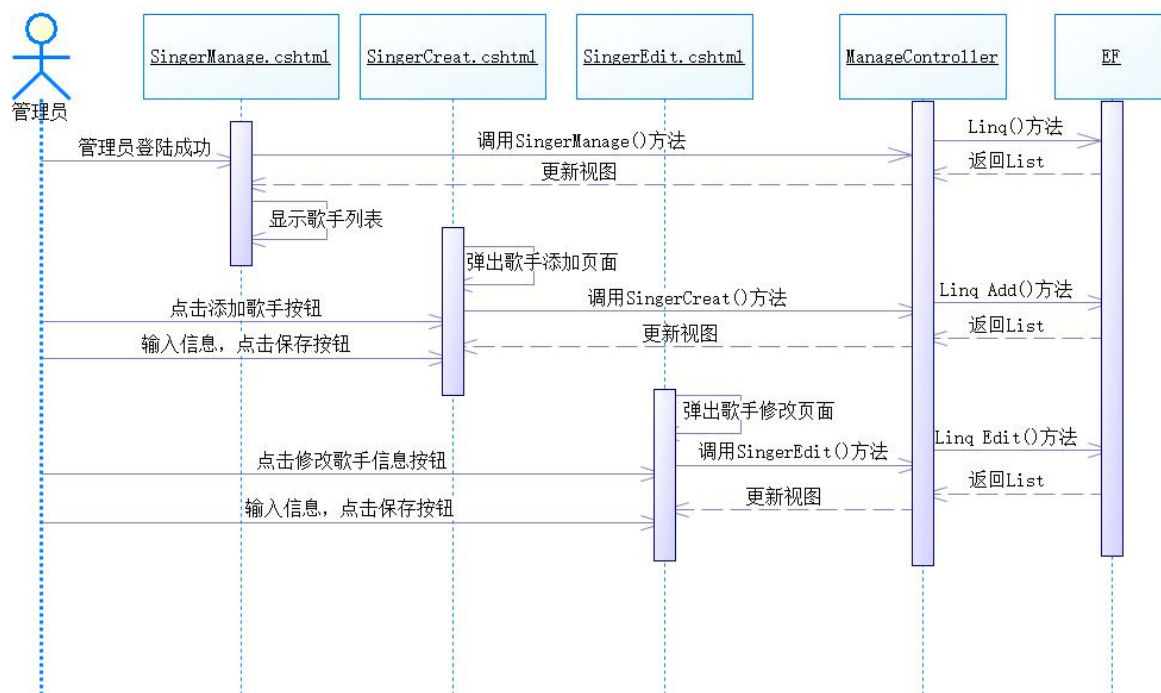


图 3-8 歌手管理时序图

3.3.5. 音乐类型类图

音乐类型的类图如下图 3-9 所示。各类具体说明如下：

(1) ManageController 类：ManageController 类完成对音乐类型信息的增删查改操作，位于该项目目录 DaleMusic/Controllers/ManageController 处。方法包括音乐类型查询方法（MusicTypeManage()）、音乐类型添加方法（MusicTypeCreat()）、音乐类型删除方法（MusicTypeDelete()）。

(2) MusicType 类：MusicType 类音乐类型基本信息的类，位于该项目目录 Models/MusicType 处。属性包括音乐类型编号（TypeID）、音乐类型名称（TypeName）、是否存在（IsDelete）。

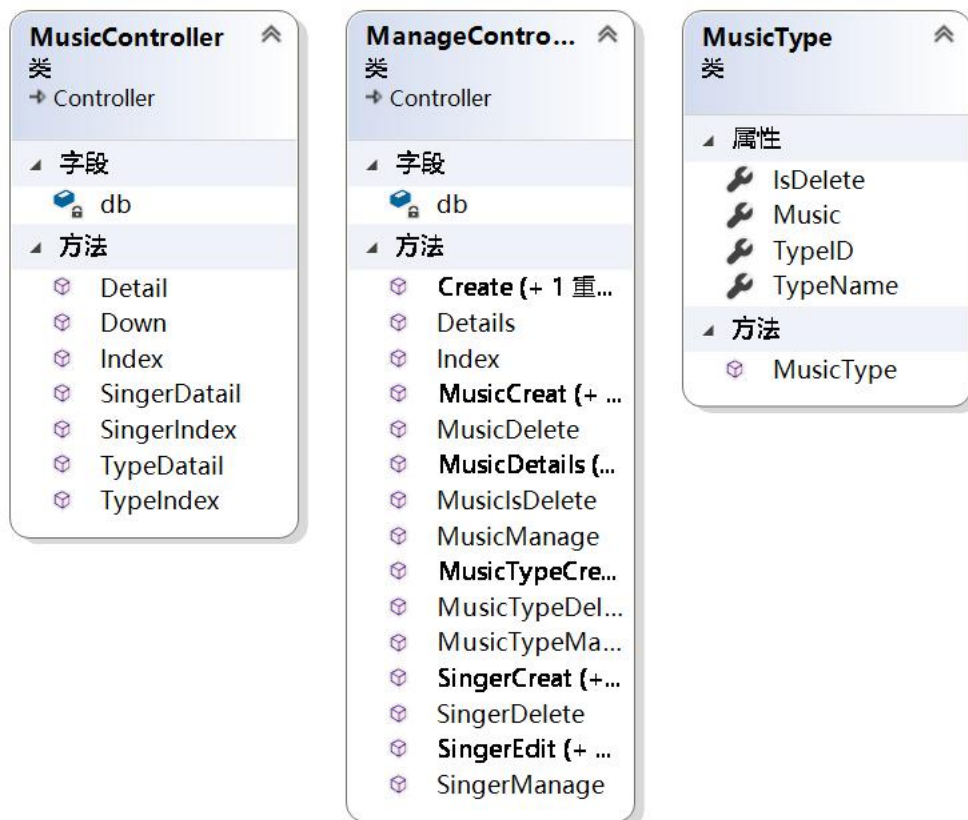


图 3-9 音乐类型类图

3.3.6. 音乐类型操作流程

音乐类型模块主要完成对音乐类型基本信息的增删查功能,其操作流程如图 3-10 音乐类型时序图所示:

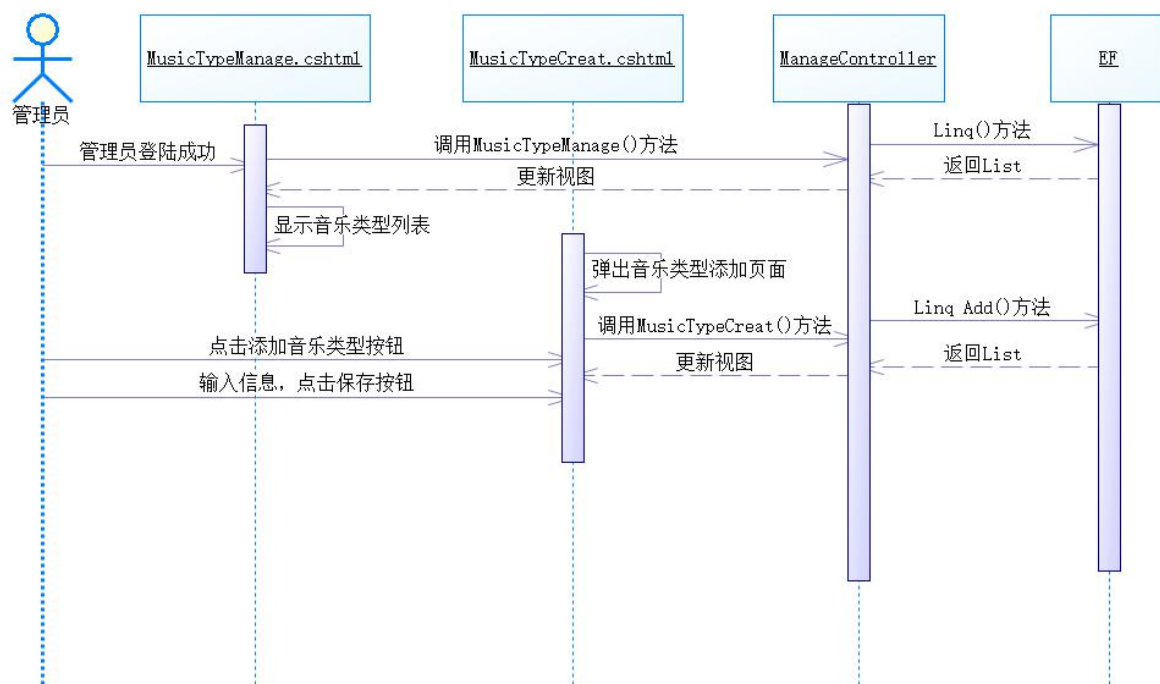


图 3-10 音乐类型时序图

3.3.7. 用户管理类图

用户管理的类图如下图 3-11 所示。各类具体说明如下：

(1) ManageController 类：ManageController 类完成对用户信息的增删查改操作，位于该项目目录 DaleMusic/Controllers/ManageController 处。方法包括用户查询方法（Index()）、用户添加方法（Create()）、音乐详情方法（Details()）。

(2) UserInfo 类：UserInfo 类用书基本信息的类，位于该项目目录 Models/UserInfo 处。属性包括用户编号（UserID）、用户名称（UserName）、用户密码（UserPwd）、用户电话（UserPhone）、用户邮箱（UserEmail）、是否存在（IsDelete）。

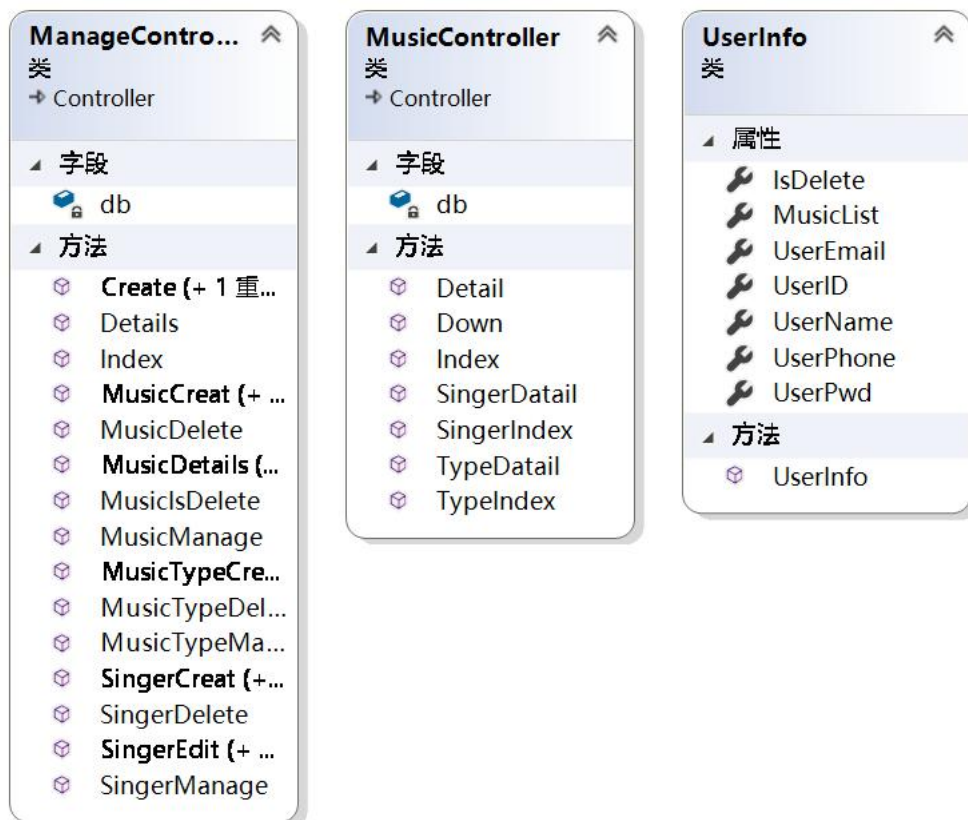


图 3-11 用户信息类图

3.3.8. 用户管理操作流程

用户管理模块主要完成对用户基本信息的增查功能,其操作流程如图 3-12 用户管理时序图所示:

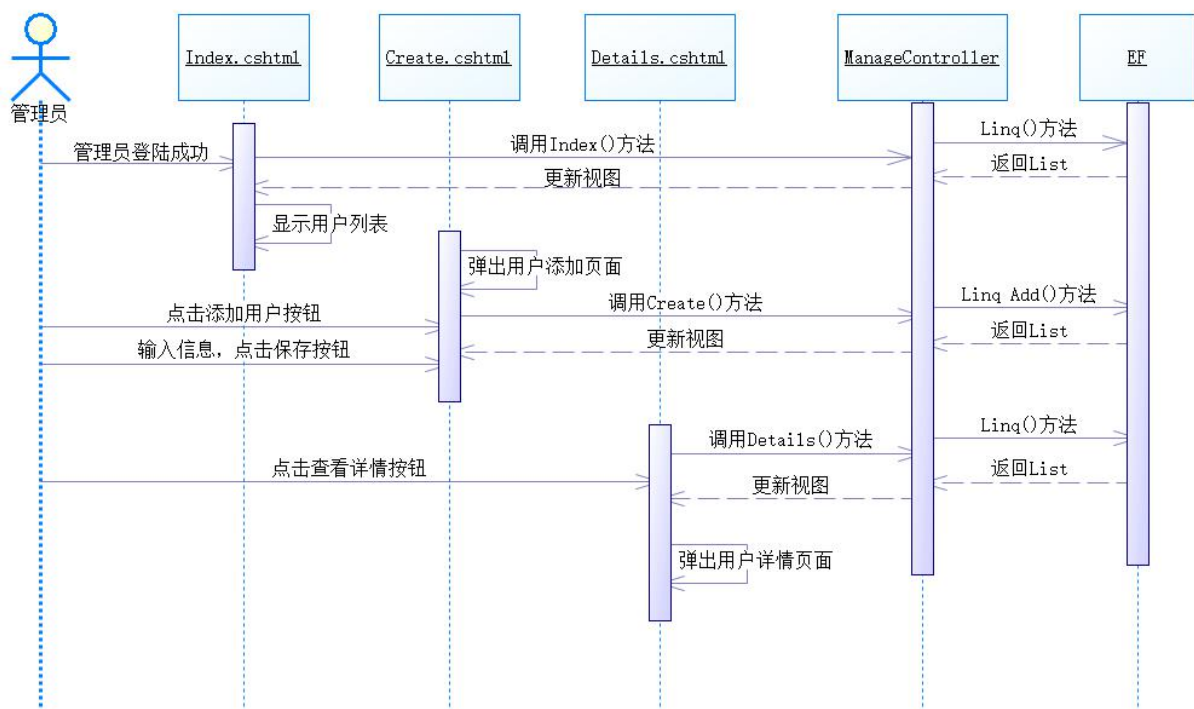


图 3-12 用户管理时序图

第 4 章 产品实现

4.1. 音乐管理模块实现

音乐管理模块包括音乐信息展示、新增音乐、修改音乐信息、删除音乐、搜索框（歌曲名）模糊查询功能。

4.1.1. 界面设计

音乐信息列表界面如图 4-1 所示, 点击修改按钮跳转到音乐信息修改页面, 音乐信息修改界面如图 4-2 所示, 点击删除按钮删除对应的音乐信息, 音乐信息添加界面如图 4-3 所示, 添加成功后弹出添加成功弹窗:

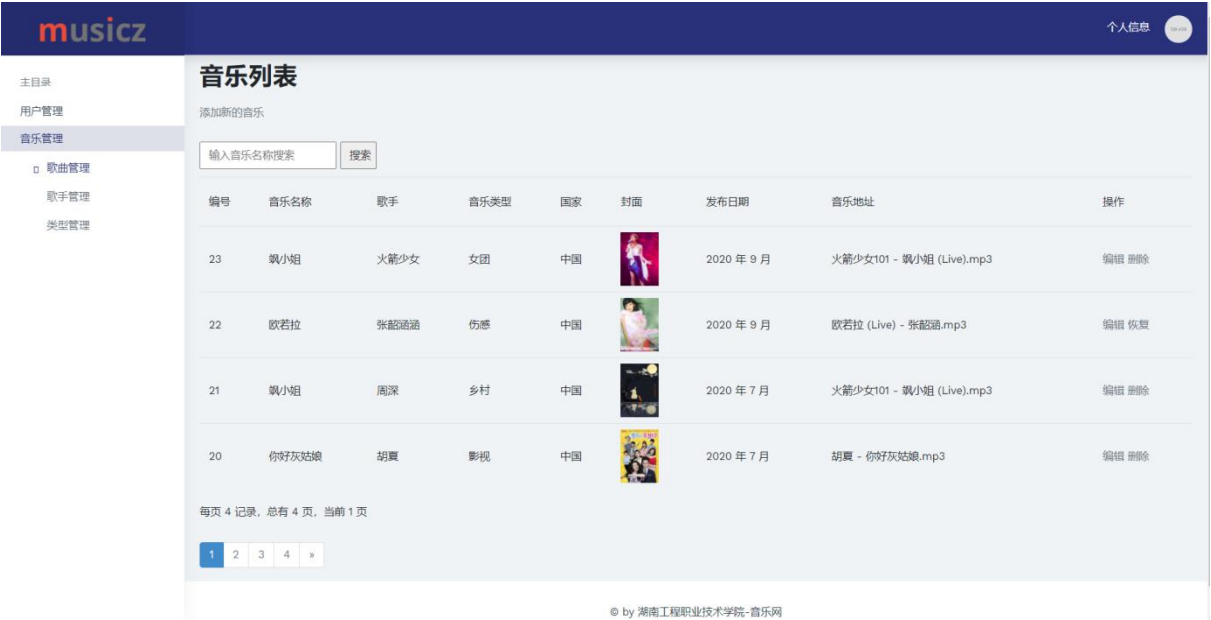


图 4-1 音乐信息列表界面图

musicz

个人信息

主目录

用户管理

音乐管理

音乐名称:

魏小姐

音乐歌手:

火箭少女

音乐类型:

女团

音乐国家:

中国

音乐封面:

选择文件

未选择任何文件

发布日期:

2020 年 9 月

音乐地址:

选择文件

未选择任何文件

提交

返回上一级

© by 湖南工程职业技术学院-音乐网

图 4-2 音乐信息修改界面图

musicz

个人信息

主目录

用户管理

音乐管理

音乐名称:

音乐名称

音乐歌手:

---请选择音乐歌手---

音乐类型:

---请选择音乐类型---

音乐国家:

音乐国家

音乐封面:

选择文件

未选择任何文件

发布日期:

年 / 月 / 日

音乐地址:

选择文件

未选择任何文件

提交

返回上一级

© by 湖南工程职业技术学院-音乐网

图 4-3 音乐信息添加界面图

4.1.2. 功能实现

音乐信息模块增删查改的主要代码


```

//音乐列表
public ActionResult MusicManage(int? page = null, string MusicName = "")
{
    List<Music> answers = db.Music.Where(p => MusicName == "" || p.MusicName.Contains(MusicName)).OrderByDescending(p => p.MusicID).ToList();

    //当前页码
    // ?? 空合并运算符，用于可为空的值类型和引用类型定义默认值。
    //如果不为空，则返回左侧操作数；否则返回右侧操作数。
    int pageNum = page ?? 1;

    //每页显示多少条
    int pageSize = 4;

    //通过ToPagedList扩展方法进行分页
    //参数：当前页、每页显示的页数
    IPagedList<Music> answerPagedList = answers.ToPagedList(pageNum, pageSize);
    return View(answerPagedList);
}

//新增音乐
public ActionResult MusicCreat()
{
    List<Music> music = db.Music.ToList();
    ViewBag.singer = db.Singer.ToList();
    ViewBag.musictype = db.MusicType.ToList();
    return View(music);
}

[HttpPost]
//判断上传文件是否为空
public ActionResult MusicCreat(Music music, HttpPostedFileBase MusicImage, HttpPostedFileBase MusicFiles)
{
    if (MusicImage != null)
    {
        //判断文件大小
        if (MusicImage.ContentLength > 0 && MusicImage.ContentLength < 4194304)
        {
            //获取上传文件路径
            string PhotofileName = Path.GetFileName(MusicImage.FileName);
            //获取文件后缀名【两种方式——截取==jpg, , , GetExtension==.jpg】
            //string suff = fileName.Substring(fileName.LastIndexOf(".")+1);
            string Photosuff = Path.GetExtension(PhotofileName);

            //判断文件格式
            if (Photosuff == ".gif" || Photosuff == ".jpg" || Photosuff == ".png")
            {
                //保存上传文件的路径
                MusicImage.SaveAs(Server.MapPath("~/Content/images/" + MusicImage.FileName));
                //获取上传文件名，用于后期拿图片
                ViewBag.img = MusicImage.FileName;
                //存入数据库字段中
                music.MusicImage = MusicImage.FileName;
            }
            else
            {
                ViewBag.Message = "文件格式不正确！";
            }
        }
        else
        {
            ViewBag.Message = "请上传文件！";
        }
    }
    if (MusicFiles != null)
    {
        //判断文件大小
        if (MusicFiles.ContentLength > 0 && MusicFiles.ContentLength < 4194304)
        {
            //获取上传文件路径
            string MusicfileName = Path.GetFileName(MusicFiles.FileName);
            //获取文件后缀名【两种方式——截取==jpg, , , GetExtension==.jpg】
            //string suff = fileName.Substring(fileName.LastIndexOf(".")+1);
            string Musicosuff = Path.GetExtension(MusicfileName);

            music.MusicFiles = "~/Content/音频/" + MusicFiles.FileName;
            //判断文件格式
            if (Musicosuff == ".mp3")
            {
                //保存上传文件的路径
                MusicFiles.SaveAs(Server.MapPath("~/Content/音频/" + MusicFiles.FileName));
                //获取上传文件名，用于后期拿图片
                ViewBag.img = MusicFiles.FileName;
                //存入数据库字段中
                music.MusicFiles = MusicFiles.FileName;
            }
            else
            {
                ViewBag.Message = "文件格式不正确！";
            }
        }
        else
        {
            ViewBag.Message = "文件大小不符合要求！";
        }
    }
    else
    {
        ViewBag.Message = "请上传文件！";
    }
    db.Music.Add(music);
    db.SaveChanges();
    return RedirectToAction("MusicManage");
}

//编辑音乐
public ActionResult MusicDetails(int? id)
{
    Music music = db.Music.Find(id);
    ViewBag.singer = db.Singer.ToList();
    ViewBag.musictype = db.MusicType.ToList();
    return View(music);
}

[HttpPost]
public ActionResult MusicDetails(Music music, HttpPostedFileBase MusicImage, HttpPostedFileBase MusicFiles)

```

```

    if (MusicImage != null)
    {
        //判断文件大小
        if (MusicImage.ContentLength > 0 && MusicImage.ContentLength < 4194304)
        {
            //获取上传文件路径
            string fileName = Path.GetFileName(MusicImage.FileName);
            //获取文件后缀名【两种方式-----截取==jpg, , GetExtension==.jpg】
            //string suff = fileName.Substring(fileName.LastIndexOf(".") + 1);
            string suff = Path.GetExtension(fileName);
            //判断文件格式
            if (suff == ".gif" || suff == ".jpg" || suff == ".png")
            {
                //保存上传文件的路径
                MusicImage.SaveAs(Server.MapPath("~/Content/images/" + MusicImage.FileName));
                //获取上传文件名,用于后期拿图片
                ViewBag.img = MusicImage.FileName;
                //存入数据库字段中
                music.MusicImage = MusicImage.FileName;
            }
            else
            {
                ViewBag.Message = "文件格式不正确! ";
            }
        }
        else
        {
            ViewBag.Message = "文件大小不符合要求! ";
        }
    }
    if (MusicFiles != null)
    {
        //判断文件大小
        if (MusicFiles.ContentLength > 0 && MusicFiles.ContentLength < 4194304)
        {
            //获取上传文件路径
            string fileName = Path.GetFileName(MusicFiles.FileName);
            //获取文件后缀名【两种方式-----截取==jpg, , GetExtension==.jpg】
            //string suff = fileName.Substring(fileName.LastIndexOf(".") + 1);
            string suff = Path.GetExtension(fileName);
            music.MusicFiles = "~/Content/音频/" + MusicFiles.FileName;
            //判断文件格式
            if (suff == ".mp3")
            {
                //保存上传文件的路径
                MusicFiles.SaveAs(Server.MapPath("~/Content/音频/" + MusicFiles.FileName));
                //获取上传文件名,用于后期拿图片
                ViewBag.img = MusicFiles.FileName;
                //存入数据库字段中
                music.MusicFiles = MusicFiles.FileName;
            }
            else
            {
                ViewBag.Message = "文件格式不正确! ";
            }
        }
        else
        {
            ViewBag.Message = "文件大小不符合要求! ";
        }
    }
    else
    {
        ViewBag.Message = "请上传文件! ";
    }
    db.Entry(music).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("MusicManage");
}

//删除音乐
public ActionResult MusicDelete(int id)
{
    db.Music.Pind(id).IsDelete = false;
    db.SaveChanges();
    return RedirectToAction("MusicManage");
}

//恢复音乐
public ActionResult MusicIsDelete(int id)
{
    db.Music.Pind(id).IsDelete = true;
    db.SaveChanges();
    return RedirectToAction("MusicManage");
}
}

```

图 4-4 音乐信息主要代码图

4.2. 歌手管理模块实现

歌手管理模块包括歌手信息展示、添加歌手、修改歌手信息、删除歌手、搜索框（歌手名）模糊查询功能。。

4.2.1. 界面设计

歌手管理列表界面如图 4-5 所示, 点击修改按钮跳转到歌手信息修改页面, 歌手信息修改界面如图 4-6 所示, 点击删除按钮删除对应的歌手信息, 歌手信息添加界面如图 4-7 所示, 添加成功后弹出添加成功弹窗:

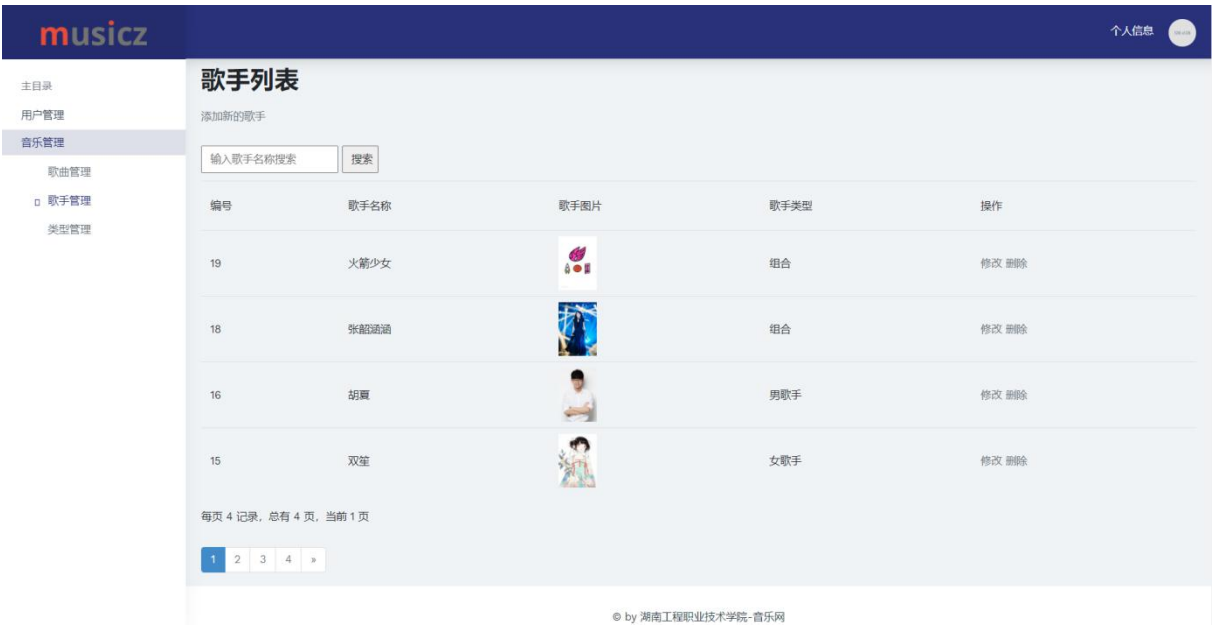


图 4-5 歌手管理列表界面图

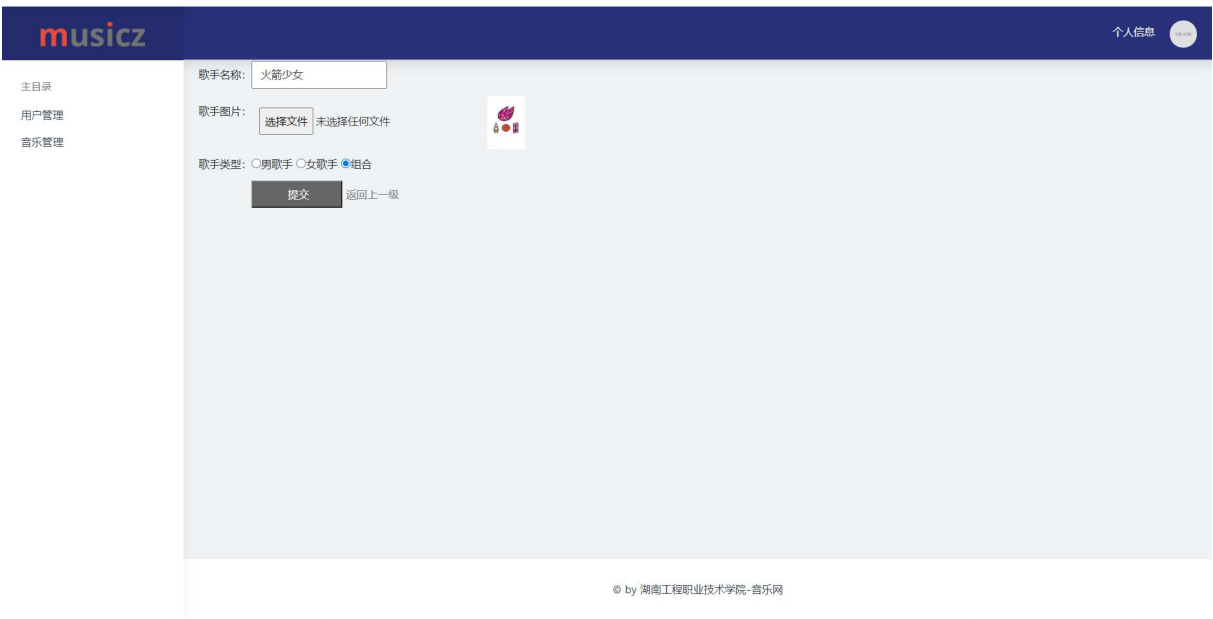


图 4-6 歌手修改界面图

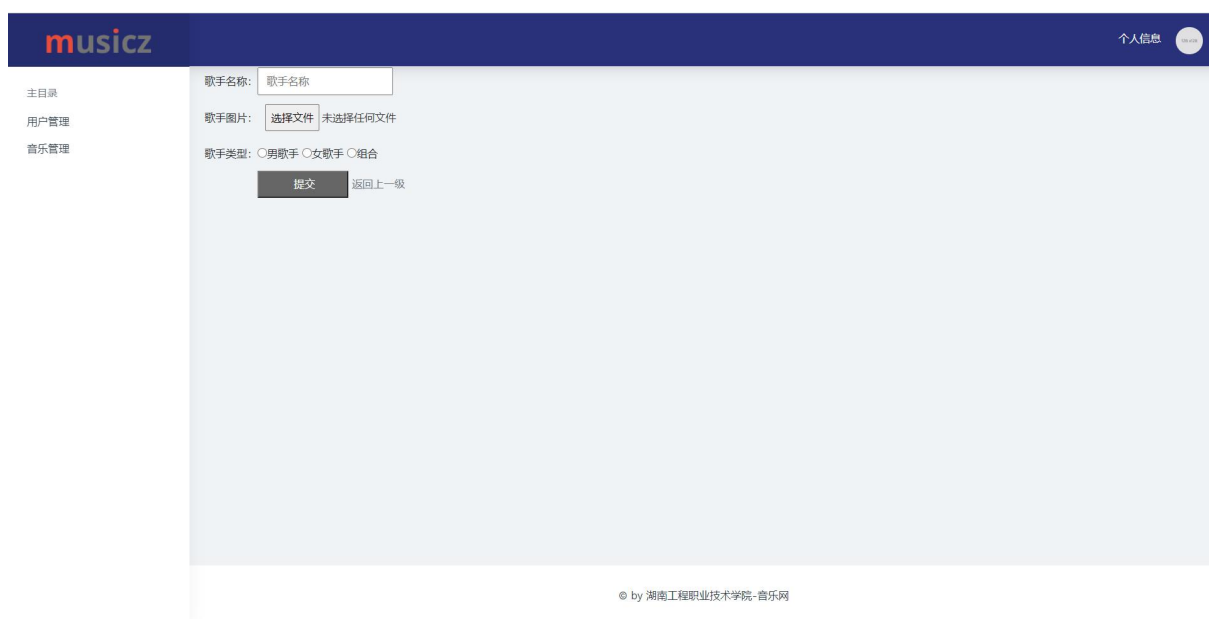


图 4-7 歌手添加界面图

4.2.2. 功能实现

歌手管理模块增删查改的主要代码

```
//歌手列表
public ActionResult SingerManage(int? page = null, string SingerName = "")
{
    List<Singer> answers = db.Singer.Where(p => SingerName == "" || p.SingerName.Contains(SingerName)).OrderByDescending(p => p.SingerID).ToList()
    //当前页码
    // ?? 空合并运算符，用于为可为空的值类型和引用类型定义默认值。
    //如果不为空，则返回左侧操作数；否则返回右侧操作数。
    int pageNum = page ?? 1;
    //每页显示多少条
    int pageSize = 4;
    //通过ToPagedList扩展方法进行分页
    //参数：当前页、每页显示的页数
    IPagedList<Singer> answerPagedList = answers.ToPagedList(pageNum, pageSize);
    return View(answerPagedList);
}

//新增歌手
public ActionResult SingerCreat()
{
    return View();
}

[HttpPost]
public ActionResult SingerCreat(Singer singer, HttpPostedFileBase file)
{
    if (file != null)
    {
        //判断文件大小
        if (file.ContentLength > 0 && file.ContentLength < 4194304)
        {
            //获取上传文件路径
            string fileName = Path.GetFileName(file.FileName);
            //获取文件后缀名【两种方式-----截取==jpg, , , GetExtension==.jpg】
            //string suff = fileName.Substring(fileName.LastIndexOf(".")+1);
            string suff = Path.GetExtension(fileName);
            //判断文件格式
            if (suff == ".gif" || suff == ".jpg" || suff == ".png")
            {
                //保存上传文件的路径
                file.SaveAs(Server.MapPath("~/Content/歌手图片/" + file.FileName));
                //获取上传文件名，用于后期拿图片
                ViewBag.img = file.FileName;
                //存入数据库字段中
                singer.SingerImage = file.FileName;
            }
            else
            {
                ViewBag.Message = "文件格式不正确！";
            }
        }
        else
        {
            ViewBag.Message = "文件大小不符合要求！";
        }
    }
    else
    {
        ViewBag.Message = "请上传文件！";
    }
}
```

```

    }

    db.Singer.Add(singer);
    db.SaveChanges();
    return RedirectToAction("SingerManage");
}

//修改歌手
public ActionResult SingerEdit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Singer singer = db.Singer.Find(id);
    if (singer == null)
    {
        return HttpNotFound();
    }
    return View(singer);
}

[HttpPost]
public ActionResult SingerEdit([Bind(Include = "SingerID, SingerName, SingerType, SingerImage")] Singer singer, HttpPostedFileBase file)
{
    if (file != null)
    {
        //判断文件大小
        if (file.ContentLength > 0 && file.ContentLength < 4194304)
        {
            //获取上传文件路径
            string fileName = Path.GetFileName(file.FileName);
            //获取文件后缀名【两种方式-----截取==jpg, , , GetExtension==.jpg】
            //string suff = fileName.Substring(fileName.LastIndexOf(".") + 1);
            string suff = Path.GetExtension(fileName);

            //判断文件格式
            if (suff == ".gif" || suff == ".jpg" || suff == ".png")
            {
                //保存上传文件的路径
                file.SaveAs(Server.MapPath("~/Content/歌手图片/" + file.FileName));
                //获取上传文件名,用于后期拿图片
                ViewBag.img = file.FileName;
                //存入数据库字段中
                singer.SingerImage = file.FileName;
            }
            else
            {
                ViewBag.Message = "文件格式不正确!";
            }
        }
        else
        {
            ViewBag.Message = "文件大小不符合要求!";
        }
    }
    else
    {
        ViewBag.Message = "请上传文件!";
    }
    singer.SingerImage = file.FileName.ToString();
    db.Entry(singer).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("SingerManage");
}

//删除歌手
public ActionResult SingerDelete(int? id)
{
    List<Singer> singer = db.Singer.Where(p => p.SingerID == id).ToList();
    if (singer.Count > 0)
    {
        return Content("<script>alert('请解除该歌手的音乐绑定,再删除该歌手');history.back(-1);</script>");
    }
    else
    {
        var user = db.Singer.Find(id);
        db.Singer.Remove(user);
        db.SaveChanges();
    }
    return RedirectToAction("SingerManage");
}

```

图 4-8 歌手管理模块主要代码图

4.3. 音乐类型模块实现

音乐类型模块包括音乐类型展示、添加音乐类型、删除音乐类型、搜索框（音乐类型名）模糊查询功能。。

4.3.1. 界面设计

音乐类型列表界面如图 4-9 所示, 点击添加按钮跳转到音乐类型添加页面，音乐类型添加界面如图 4-10 所示，添加成功后弹出添加成功弹窗：



图 4-9 音乐类型列表界面图

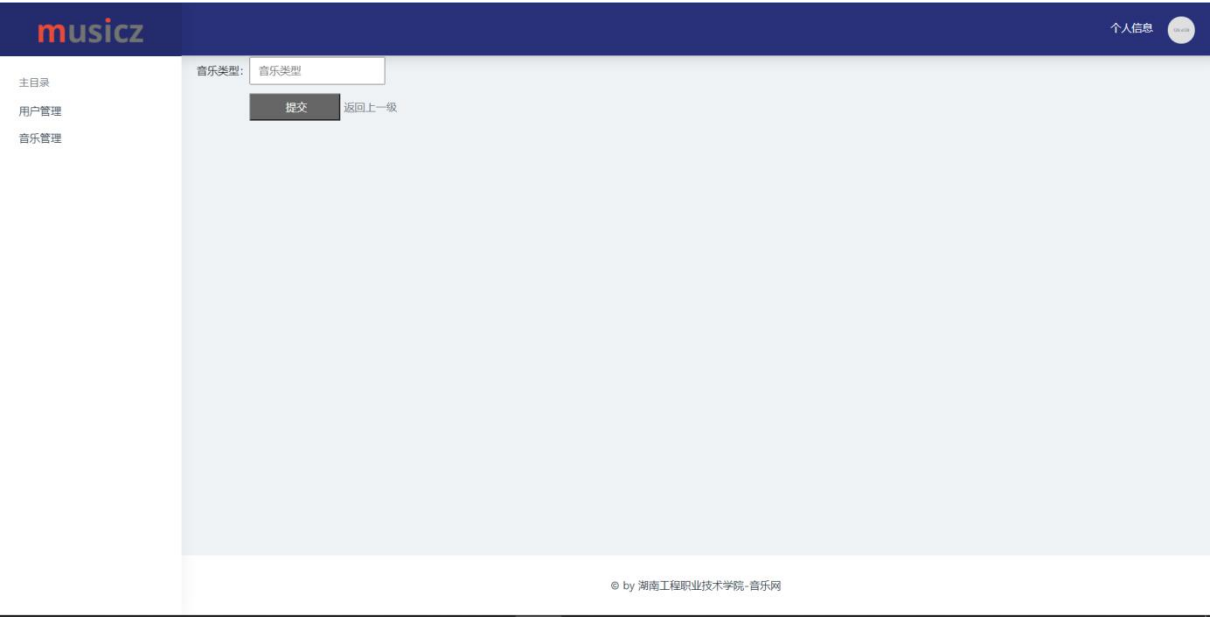


图 4-10 类型添加界面图

4.3.2. 功能实现

音乐类型模块增删查改的主要代码

```
//音乐类型列表
public ActionResult MusicTypeManage(int? page = null, string TypeName = "")
{
    List<MusicType> answers = db.MusicType.Where(p => TypeName == "" || p.TypeName.Contains(TypeName)).OrderByDescending(p => p.TypeID).ToList();

    //当前页码
    // ?? 空合并运算符，用于为可为空的值类型和引用类型定义默认值。
    //如果不为空，则返回左侧操作数；否则返回右侧操作数。
    int pageNum = page ?? 1;

    //每页显示多少条
    int pageSize = 4;

    //通过ToPagedList扩展方法进行分页
    //参数：当前页、每页显示的页数
    IPagedList<MusicType> answerPagedList = answers.ToPagedList(pageNum, pageSize);
    return View(answerPagedList);
}

//新增音乐类型
public ActionResult MusicTypeCreat()
{
    return View();
}

[HttpPost]
public ActionResult MusicTypeCreat(MusicType musicType)
{
    if (ModelState.IsValid)
    {
        db.MusicType.Add(musicType);
        db.SaveChanges();
        return RedirectToAction("MusicTypeManage");
    }
    else
    {
        return View();
    }
}

//删除音乐类型
public ActionResult MusicTypeDelete(int? id)
{
    List<MusicType> musicTypes = db.MusicType.Where(p => p.TypeID == id).ToList();
    if (musicTypes.Count > 0)
    {
        return Content("<script>alert('请解除该类型的音乐绑定，再删除该类型');history.back(-1);</script>");
    }
    else
    {
        var user = db.Singer.Find(id);
        db.Singer.Remove(user);
        db.SaveChanges();
    }
    return RedirectToAction("MusicTypeManage");
}
```

图 4-11 音乐类型模块主要代码图

4.4. 用户管理模块实现

用户管理模块包括用户信息展示、添加用户、查看用户详情、搜索框（用户名）模糊查询功能。。

4.4.1. 界面设计

用户管理列表界面如图 4-12 所示, 点击添加按钮跳转到音乐类型添加页面, 音乐类型添加界面如图 4-13 所示, 添加成功后弹出添加成功弹窗, 点击详情按钮跳转到用户信息详情页面, 用户信息详情界面如图 4-14 所示:

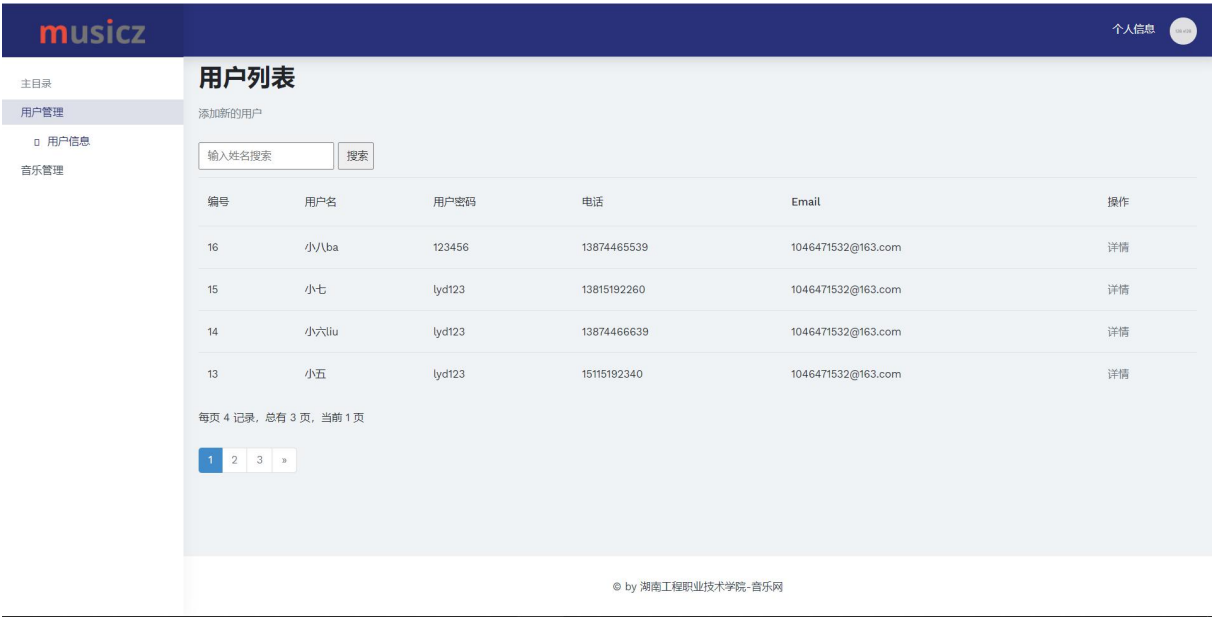


图 4-12 用户管理列表界面图

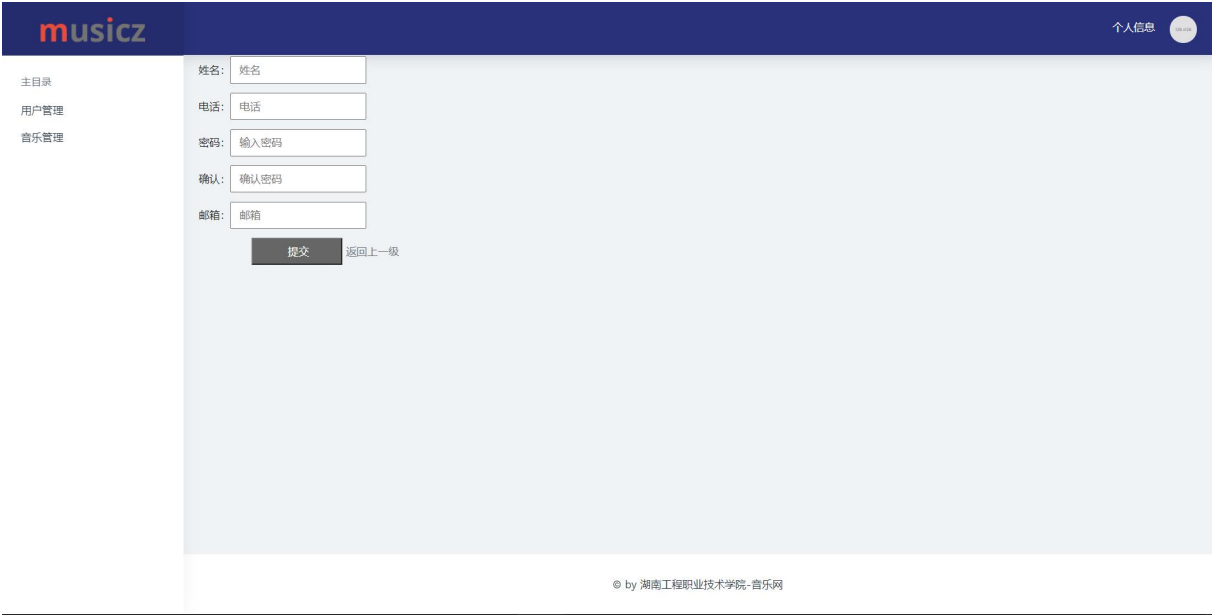


图 4-13 用户添加界面图

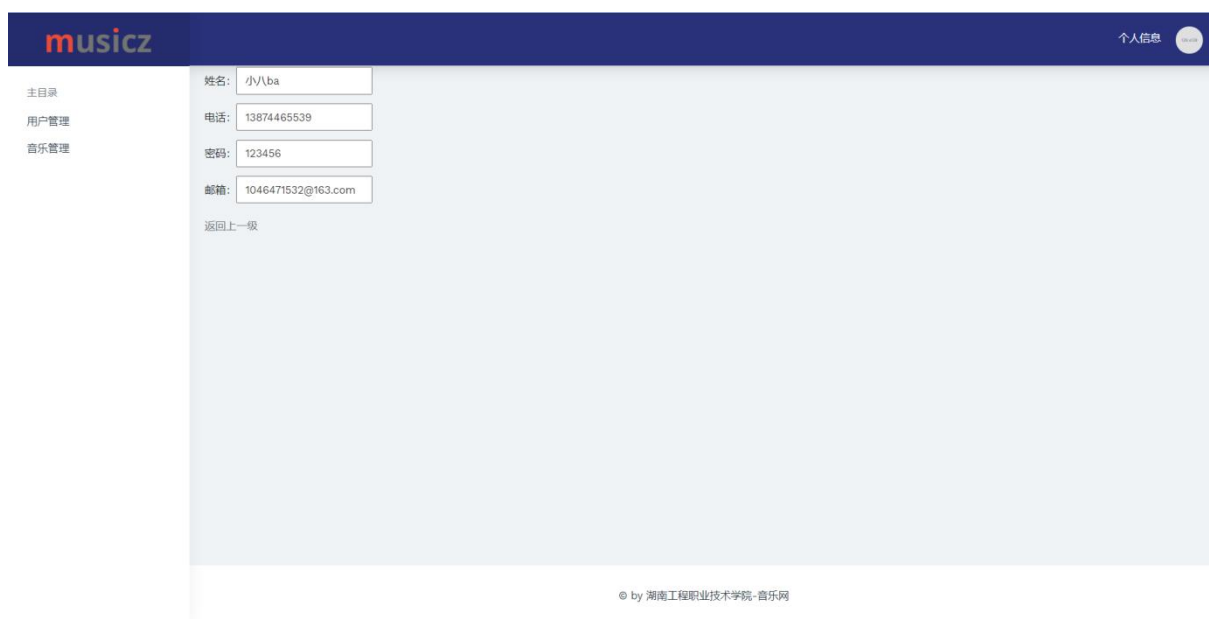


图 4-14 用户详情界面图

4.4.2. 功能实现

用户管理模块增删查改的主要代码

```
//用户列表
public ActionResult Index(int? page = null, string UserName = "")
{
    List<UserInfo> answers = db.UserInfo.Where(p => UserName == "" || p.UserName.Contains(UserName)).OrderByDescending(p => p.UserID).ToList();
    //当前页码
    // ?? 空合并运算符，用于为可为空的值类型和引用类型定义默认值。
    //如果不为空，则返回左侧操作数；否则返回右侧操作数。
    int pageNum = page ?? 1;

    //每页显示多少条
    int pageSize = 4;

    //通过ToPagedList扩展方法进行分页
    //参数：当前页、每页显示的页数
    IPagedList<UserInfo> answerPagedList = answers.ToPagedList(pageNum, pageSize);
    return View(answerPagedList);
}

//新增用户
public ActionResult Create()
{
    return View();
}

[HttpPost]
public ActionResult Create(UserInfo user)
{
    if (ModelState.IsValid)
    {
        db.UserInfo.Add(user);
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    else
    {
        return View();
    }
}

//用户详情
public ActionResult Details(int? id)
{
    UserInfo userInfo = db.UserInfo.Find(id);
    return View(userInfo);
}
```

图 4-15 用户管理模块主要代码图

第 5 章 产品测试

5.1. 测试方法与策略

本项目在所有功能编码完成且系统能完整运行的状态下进行系统测试，以确认各项需求的功能是否能正常运行，并给出测试结果。

本系统采用人工测试的方式进行黑盒测试对系统功能进行验证，使用到的测试用例设计方法包括：边界值分析、等价类划分、错误猜测法、因果图法等。

本项目的测试策略包括：

1. 功能测试

功能测试验证本系统能按照设计要求实现模块的各个功能，数据应完整、界面美观、操作方便。

2. 数据项测试

- 输入字母和数字等数据项是否能够正确显示，并添加到系统中；
- 鼠标点击、图形模式的数据项（如滑动条）是否正常工作；
- 是否能够识别非法数据；
- 提示消息是否可理解。

3. 业务测试

功能测试完成后进行业务测试，业务测试关注的要点是业务流程，及数据流从软件中的一个模块流到另一个模块的过程中的正确性。

5.2. 音乐管理模块测试

音乐管理模块的测试主要验证管理员是否能正确的实现音乐的增删改查功能，并对管理员输入的数据进行合法性验证，对输入的非法数据给出提示，并验证提示信息的可理解性。该模块的测试用例及测试执行结果如下表 5-1 所示：

表 5-1 音乐管理模块测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
1	添加音乐信息	【添加】按钮功能测试	高	管理员登录成功，打音乐管理页面	无	点击【添加】按钮	弹出“添加音乐”页面	通过

2	添加音乐信息	输入全部正确的新增数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“添加音乐信息”窗口	1、音乐名称：飒小姐 2、音乐歌手：火箭少女 101 3、音乐类型：流行 4、音乐国家：中国大陆 5、音乐封面：飒小姐.jpg 6、发布时间：2016-03-12 7、音乐地址：火箭少女 101 - 飒小姐 (Live).mp3	输入以上数据，点击【取消】按钮	不保存当前新增内容，关闭当前窗口，返回列表页	通过
3	添加音乐信息	不输入音乐名称（非空验证）数据，进行保存	高	管理员登录成功，打开“添加音乐信息”窗口	1、音乐名称： 2、音乐歌手：火箭少女 101 3、音乐类型：流行 4、音乐国家：中国大陆 5、音乐封面：飒小姐.jpg 6、发布时间：2016-03-12 7、音乐地址：火箭少女 101 - 飒小姐 (Live).mp3	输入以上数据，点击【保存】按钮	提示：音乐名称不能为空	通过
4	添加音乐信息	不输入音乐歌手（非空验证）数据，	高	管理员登录成功，打开“添加音乐信息”窗口	1、音乐名称：飒小姐 2、音乐歌手：	输入以上数据，点击【保存】按钮	提示：请选择音乐歌手	通过

		进行保存			3、音乐类型：流行 4、音乐国家：中国大陆 5、音乐封面：飒小姐.jpg 6、发布时间：2016-03-12 7、音乐地址：火箭少女 101 - 飒小姐 (Live).mp3			
5	添加音乐信息	不输入音乐类型（非空验证）数据，进行保存	高	管理员登录成功，打开“添加音乐信息”窗口	1、音乐名称：飒小姐 2、音乐歌手：火箭少女 101 3、音乐类型： 4、音乐国家：中国大陆 5、音乐封面：飒小姐.jpg 6、发布时间：2016-03-12 7、音乐地址：火箭少女 101 - 飒小姐 (Live).mp3	输入以上数据，点击【保存】按钮	提示：请选择音乐类型	通过
6	添加音乐信息	不输入音乐国家（非空验证）数据，进行保存	高	管理员登录成功，打开“添加音乐信息”窗口	1、音乐名称：飒小姐 2、音乐歌手：火箭少女 101 3、音乐类型：流行 4、音乐国家：5、音乐	输入以上数据，点击【保存】按钮	提示：请输入音乐国家	通过

					封面：飒小姐.jpg 6、发布时间： 2016-03-12 7、音乐地址：火箭少女 101 - 飒小姐 (Live).mp3			
7	添加音乐信息	不输入音乐封面（非空验证）数据，进行保存	高	管理员登录成功，打开“添加音乐信息”窗口	1、音乐名称：飒小姐 2、音乐歌手：火箭少女 101 3、音乐类型：流行 4、音乐国家：中国大陆 5、音乐封面： 6、发布时间： 2016-03-12 7、音乐地址：火箭少女 101 - 飒小姐 (Live).mp3	输入以上数据，点击【保存】按钮	提示：请选择音乐封面	通过
8	添加音乐信息	不输入发布时间（非空验证）数据，进行保存	高	管理员登录成功，打开“添加音乐信息”窗口	1、音乐名称：飒小姐 2、音乐歌手：火箭少女 101 3、音乐类型：流行 4、音乐国家：中国大陆 5、音乐封面：飒小姐.jpg 6、发布时间： 7、音乐地	输入以上数据，点击【保存】按钮	提示：请输入发布时间	通过

					址：火箭少女 101 - 飒小姐 (Live).mp3			
9	添加音乐信息	不输入音乐地址（非空验证）数据，进行保存	高	管理员登录成功，打开“添加音乐信息”窗口	1、音乐名称：飒小姐 2、音乐歌手：火箭少女 101 3、音乐类型：流行 4、音乐国家：中国大陆 5、音乐封面：飒小姐.jpg 6、发布时间：2016-03-12 7、音乐地址：	输入以上数据，点击【保存】按钮	提示：请输入音乐地址	通过
10	修改音乐信息	【修改】按钮功能测试	高	管理员登录成功，打开音乐列表页面	无	点击【修改】按钮	弹出“修改音乐”页面，显示带入的音乐信息	通过
11	修改音乐信息	“修改音乐信息”窗口文字显示及 UI 排版正确性验证	中	管理员登录成功，打开“修改音乐信息”窗口	无	查看页面显示	页面文字显示及 UI 排版正确	通过
12	修改音乐信息	输入全部正确的修改数据，进行【保存】按钮功能测试	高	管理员登录成功，打开“修改音乐信息”窗口	1、音乐名称：飒小姐 2、音乐歌手：火箭少女 101 3、音乐类型：流行 4、音乐国家：中国大陆 5、音乐封面：飒小姐.jpg	输入以上数据，点击【保存】按钮	保存当前修改内容，关闭当前窗口，回到列表页	通过

					6、发布时间： 2016-03-12 7、音乐地址：火箭少女 101 - 飒小姐 (Live).mp3			
13	修改音乐信息	修改音乐信息记录正确性验证	高	管理员登录成功，打开音乐信息页面	无	查看音乐信息列表页数据显示	列表页相应内容随之更新	通过
14	修改音乐信息	输入全部正确的修改数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“修改音乐信息”窗口	1、音乐名称：飒小姐 2、音乐歌手：火箭少女 101 3、音乐类型：流行 4、音乐国家：中国大陆 5、音乐封面：飒小姐.jpg 6、发布时间：2016-03-12 7、音乐地址：火箭少女 101 - 飒小姐 (Live).mp3	输入以上数据，点击【取消】按钮	不保存当前编辑内容，关闭当前窗口，返回列表页，列表页相应内容前后不变	通过
15	修改音乐信息	修改音乐信息正确性验证	高	管理员登录成功，打开音乐管理页面	无	查看音乐信息页数据显示	列表页相应内容随之更新	通过
16	删除音乐信息	【删除】按钮功能测试	高	管理员登录成功，打开音乐管理页面	无	点击【删除】按钮	提示：确认删除该首歌吗？	通过
17	删除音乐信息	删除弹出框【确认】按钮测试	高	管理员登录成功，点击【删除】按钮，弹出删除	无	点击【确认】按钮	删除成功	通过

				提示框				
18	删除音乐信息	删除弹出框【取消】按钮测试	高	管理员登录成功，点击【删除】按钮，弹出删除提示框	无	点击【取消】按钮	取消删除	通过

5.3. 歌手管理模块测试

歌手管理模块的测试主要验证管理员是否能正确的实现歌手信息的增删改查功能，并对管理员输入的数据进行合法性验证，对输入的非法数据给出提示，并验证提示信息可理解性。该模块的测试用例及测试执行结果如下表 5-2 所示：

表 5-2 歌手管理模块测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
1	添加歌手信息	【添加】按钮功能测试	高	管理员登录成功，打开歌手管理页面	无	点击【添加】按钮	弹出“添加歌手”页面	通过
2	添加歌手信息	输入全部正确的新增数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“添加歌手信息”窗口	1、歌手名称：郁可唯 2、歌手图片：郁可唯.jpg 3、歌手类型：女歌手	输入以上数据，点击【取消】按钮	不保存当前新增内容，关闭当前窗口，回到列表页	通过
3	添加歌手信息	不输入歌手名称（非空验证）数据，进行保存	高	管理员登录成功，打开“添加歌手信息”窗口	1、歌手名称： 2、歌手图片：郁可唯.jpg 3、歌手类型：女歌手	输入以上数据，点击【保存】按钮	提示：歌手名称不能为空	通过
4	添加歌手信息	不输入歌手图片（非空验证）数据，进行保存	高	管理员登录成功，打开“添加歌手信息”窗口	1、歌手名称：郁可唯 2、歌手图片： 3、歌手类	输入以上数据，点击【保存】按钮	提示：请选择歌手图片	通过

					型：女歌手			
5	添加歌手信息	不输入歌手类型（非空验证）数据，进行保存	高	管理员登录成功，打开“添加歌手信息”窗口	1、歌手名称：郁可唯 2、歌手图片：郁可唯.jpg 3、歌手类型：	输入以上数据，点击【保存】按钮	提示：请选择歌手类型	通过
6	修改歌手信息	【修改】按钮功能测试	高	管理员登录成功，打开歌手管理页面	无	点击【修改】按钮	弹出“修改歌手”页面，显示带入的歌手信息	通过
7	修改歌手信息	“修改歌手信息”窗口文字显示及UI排版正确性验证	中	管理员登录成功，打开“修改歌手信息”窗口	无	查看页面显示	页面文字显示及UI排版正确	通过
8	修改歌手信息	输入全部正确的修改数据，进行【保存】按钮功能测试	高	管理员登录成功，打开“修改歌手信息”窗口	1、歌手名称：郁可唯 2、歌手图片：郁可唯.jpg 3、歌手类型：女歌手	输入以上数据，点击【保存】按钮	保存当前修改内容，关闭当前窗口，返回列表页	通过
9	修改歌手信息	修改歌手信息记录正确性验证	高	管理员登录成功，打开歌手管理页面	无	查看歌手信息列表页数据显示	列表页相应内容随之更新	通过
10	修改歌手信息	输入全部正确的修改数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“修改歌手信息”窗口	1、歌手名称：郁可唯 2、歌手图片：郁可唯.jpg 3、歌手类型：女歌手	输入以上数据，点击【取消】按钮	不保存当前编辑内容，关闭当前窗口，返回列表页，列表页相应内容前后不变	通过

11	修改歌手信息	修改歌手信息记录正确性验证	高	管理员登录成功，打开歌手管理页面	无	查看歌手信息页数据显示	列表页相应内容随之更新	通过
12	删除歌手信息	【删除】按钮功能测试	高	管理员登录成功，打开歌手管理页面	无	点击【删除】按钮	提示：确认删除该歌手吗？	通过
13	删除歌手信息	删除弹出框【确认】按钮测试	高	管理员登录成功，点击【删除】按钮，弹出删除提示框	无	点击【确认】按钮	删除成功	通过
14	删除歌手信息	删除弹出框【取消】按钮测试	高	管理员登录成功，点击【删除】按钮，弹出删除提示框	无	点击【取消】按钮	取消删除	通过

5.4. 音乐类型模块测试

音乐类型模块的测试主要验证管理员是否能正确的实现音乐类型的增删查功能，并对管理员输入的数据进行合法性验证，对输入的非法数据给出提示，并验证提示信息的可理解性。该模块的测试用例及测试执行结果如下表 5-3 所示：

表 5-3 音乐类型模块测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
1	添加音乐类型信息	【添加】按钮功能测试	高	管理员登录成功，打开音乐类型管理页面	无	点击【添加】按钮	弹出“添加音乐类型”页面	通过
2	添加音乐类型信息	输入全部正确的新增数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“添加音乐类型信息”窗口	1、音乐类型名称：伤感	输入以上数据，点击【取消】按钮	不保存当前新增内容，关闭当前窗口，回到列表页	通过

3	添加音乐类型信息	不输入音乐类型名称（非空验证）数据，进行保存	高	管理员登录成功，打开“添加音乐类型信息”窗口	1、音乐类型名称：	输入以上数据，点击【保存】按钮	提示：音乐类型不能为空	通过
4	删除音乐类型信息	【删除】按钮功能测试	高	管理员登录成功，打开音乐类型页面	无	点击【删除】按钮	提示：确认删除该音乐类型吗？	通过
5	删除音乐类型信息	删除弹出框【确认】按钮测试	高	管理员登录成功，点击【删除】按钮，弹出删除提示框	无	点击【确认】按钮	删除成功	通过
6	删除音乐类型信息	删除弹出框【取消】按钮测试	高	管理员登录成功，点击【删除】按钮，弹出删除提示框	无	点击【取消】按钮	取消删除	通过

5.5. 用户管理模块测试

用户管理模块的测试主要验证管理员是否能正确的实现用户信息的增查功能，并对管理员输入的数据进行合法性验证，对输入的非法数据给出提示，并验证提示信息的可理解性。该模块的测试用例及测试执行结果如下表 5-4 所示：

表 5-4 用户管理模块测试用例表

用例编号	测试项目	测试标题	重要级别	预置条件	输入	执行步骤	预期输出	测试结果
1	添加用户信息	【添加】按钮功能测试	高	管理员登录成功，打开用户管理页面	无	点击【添加】按钮	弹出“添加用户”页面	通过
2	添加用户信息	输入全部正确的新增数据，进行【取消】按钮功能测试	高	管理员登录成功，打开“添加用户信息”窗口	1、姓名：粉红的猪 2、电话：12251515212 3、密码：1yd123 4、确认密	输入以上数据，点击【取消】按钮	不保存当前新增内容，关闭当前窗口，回到列表页	通过

					码: lyd123 5、邮箱: 12345@163. com			
3	添加 用户 信息	不输入姓名（非空验证）数据，进行保存	高	管理员登录成功，打开“添加用户信息”窗口	1、姓名: 2、电话: 1225151521 2 3、密码: lyd123 4、确认密码: lyd123 5、邮箱: 12345@163. com	输入以上数据，点击【保存】按钮	提示：姓名不能为空	通过
4	添加 用户 信息	不输入电话（非空验证）数据，进行保存	高	管理员登录成功，打开“添加用户信息”窗口	1、姓名: 粉红的猪 2、电话: 3、密码: lyd123 4、确认密码: lyd123 5、邮箱: 12345@163. com	输入以上数据，点击【保存】按钮	提示：电话不能为空	通过
5	添加 用户 信息	输入错误的电话格式(判断)数据，进行保存	高	管理员登录成功，打开“添加用户信息”窗口	1、姓名: 粉红的猪 2、电话: 12345 3、密码: lyd123 4、确认密码: lyd123 5、邮箱: 12345@163. com	输入以上数据，点击【保存】按钮	提示：电话格式不正确。	通过
6	添加 用户 信息	不输入密码（非空验证）数据，进行保存	高	管理员登录成功，打开“添加用户信息”窗口	1、姓名: 粉红的猪 2、电话: 1225151521 2 3、密码: 4、确认密码: 5、邮箱: 12345@163.	输入以上数据，点击【保存】按钮	提示：密码不能为空	通过

					com			
7	添加用户信息	输入错误的密码格式(判断)数据, 进行保存	高	管理员登录成功, 打开“添加用户信息”窗口	1、姓名: 粉红的猪 2、电话: 12251515212 3、密码: 123 4、确认密码: 123 5、邮箱: 12345@163.com	输入以上数据, 点击【保存】按钮	提示: 密码不能少于 6 位	通过
8	添加用户信息	不输入确认密码 (非空验证)数据, 进行保存	高	管理员登录成功, 打开“添加用户信息”窗口	1、姓名: 粉红的猪 2、电话: 12251515212 3、密码: 1yd123 4、确认密码: 5、邮箱: 12345@163.com	输入以上数据, 点击【保存】按钮	提示: 确认密码与第一次输入不同, 请重新输入	通过
9	添加用户信息	不输入邮箱 (非空验证)数据, 进行保存	高	管理员登录成功, 打开“添加用户信息”窗口	1、姓名: 粉红的猪 2、电话: 12251515212 3、密码: 1yd123 4、确认密码: 5、邮箱:	输入以上数据, 点击【保存】按钮	提示: 邮箱不能为空	通过
10	添加用户信息	输入错误的邮箱格式(判断)数据, 进行保存	高	管理员登录成功, 打开“添加用户信息”窗口	1、姓名: 粉红的猪 2、电话: 12251515212 3、密码: 1yd123 4、确认密码: 1yd123	输入以上数据, 点击【保存】按钮	提示: 邮箱格式不正确	通过

					5、邮箱： 12345			
11	查看 用户 信息	【详情】 按钮功能 测试	高	管理员登录 成功，打开 用户管理页 面	无	点击【详 情】按钮	弹出“用 户详情” 页面，显 示用户具 体信息	通过

第 6 章 结论

6.1. 毕业设计成果特点

大乐音乐网管理员拥有音乐管理、歌手管理、音乐类型、用户管理的增删查改操作。通过此管理系统的开发，将软件技术专业授课阶段学习的前沿理论知识 C#、MVC、面向对象、数据库等与实践联系，融会贯通，不但提高了理论水平，还提高了项目开发的能力。项目采用 ASP.NET MVC 进行搭建，Web 前端采用 Html+CSS、Bootstrap 等流行的技术，结合 Javascript 和 jQuery 实现动态特效。Web 后端采用 MVC+EF 框架实现业务逻辑的处理和数据库访问。

6.2. 设计成果的实用价值或应用前景

随着互联网消费时代的到来，人们物质生活不断的提高的同时，也致使工作压力大，心情低落，影响工作效率，但一首好的音乐能够沁人心脾，能够让人精神振奋，缓解疲惫的身心，因此需要一款优秀的音乐网站将用户喜欢的歌曲分门别类的管理，可以方便用户欣赏自己喜爱的音乐，大乐音乐网就是深受用户喜欢的音乐网站。

6.3. 不足之处或遗留未予解决的问题

大乐音乐网-音乐信息、歌手信息、用户信息模块的设计与实现是基于 B/S 管理模式的结构平台和面向对象的设计思想，并利用一些与 MVC 相关的技术，对于大乐音乐网的音乐信息、歌手信息以及用户信息模块要求的功能已全部完整的实现。尽管如此，本系统在面向众多用户的使用过程中，仍遇到了某些潜在的问题，但大体上还是完成了大乐音乐网的功能需求。

本系统的不足之处：

1. 音乐信息中的首页必须通过登录才可以显示，不然不会显示。
2. 图片不能批量上传以及修改时不能显示出来。
3. 数据库的恢复和备份没有完全实现自动化。
4. 界面不够美观，需要调整样式及色彩搭配。

以上这些问题，将在今后的工作中进行完善。同时我们会在使用该系统的过程中，不断总结反馈的信息，希望该系统能在未来广泛推广使用，服务更多的人群。