

实验六: UML时序图

实验内容:

- 1.分析用例所涉及到的类或对象.
- 2.根据用例中的类或对象的关系和行为顺序绘制顺序图.
- 3.提交实验报告.

实验步骤:

1.分析用例所涉及到的类或对象的关系和行为.

时序图概述: 时序图是一种展示参与者之间互动情况的建模工具。它显示了对象之间的交互顺序, 展现了消息的传递以及对象之间的时序关系。时序图中包括如下元素: 角色, 对象, 生命线, 激活期和消息。

1、角色 (Actor)

系统角色, 可以是人或者其他系统, 子系统。

2、对象(Object)

对象代表时序图中的对象在交互中所扮演的角色, 位于时序图顶部和对象代表。

对象实例类角色。对象一般包含以下三种命名方式: 第一种方式包含对象名和类名。第二种方式只显示类名不显示对象名, 即为一个匿名对象。第三种方式只显示对象名不显示类名。

3、生命线(Lifeline)

生命线代表时序图中的对象在一段时期内的存在。时序图中每个对象和底部中心都有一条垂直的虚线, 这就是对象的生命线, 对象间 的消息存在于两条虚线间。

4、激活期(Activation)

激活期代表时序图中的对象执行一项操作的时期, 在时序图中每条生命线上的窄的矩形代表活动期。它可以被理解成C语言语义中一对花括号“{ }”中的内容。激活期实例

5、消息(Message)

消息是定义交互和协作中交换信息的类, 用于对实体间的通信内容

分析用例所涉及到的类或对象的关系和行为:

用户公共模块:

- 类/对象: 用户、用户公共模块控制器
- 关系和行为: 用户类包括登录、注册、个人信息操作等行为

普通用户专有模块:

- 类/对象: 普通用户、普通用户专有模块控制器、音乐、专辑、歌手、歌单集合
- 关系和行为: 普通用户可以创建包含多首歌曲的歌单、查看关注的歌手集合、查看自己收藏的专辑集合、查看下载的音乐集合

歌手专有模块:

- 类/对象: 歌手、歌手专有模块控制器、音乐、专辑
- 关系和行为: 歌手可以查看自己创作的音乐、专辑集合, 拥有的粉丝集合

播放器模块:

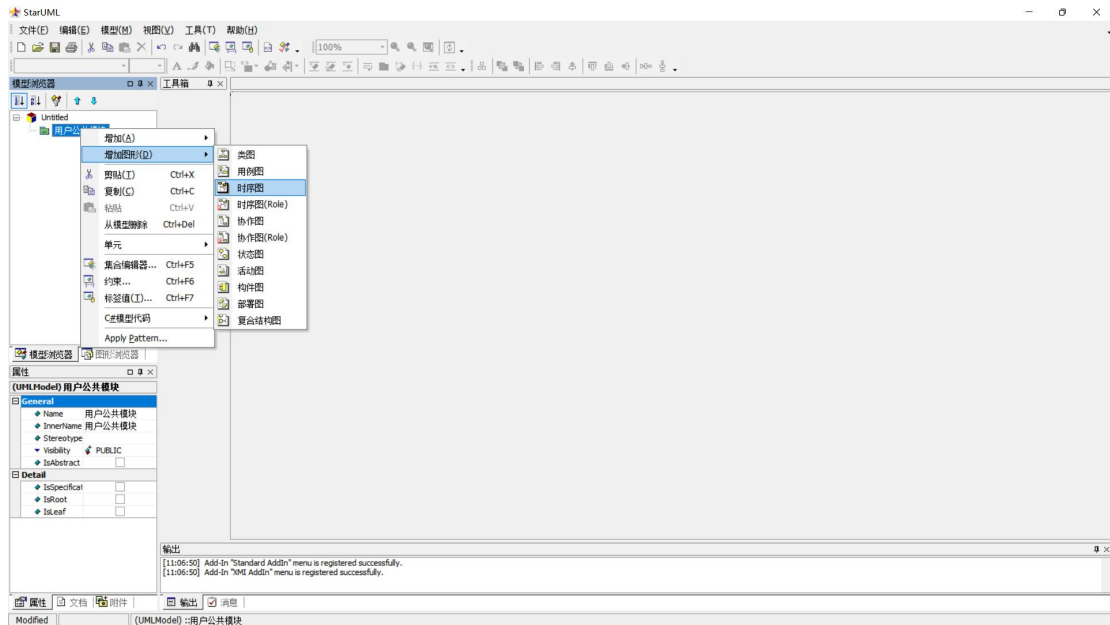
- 类/对象: 用户、搜索模块控制器、音乐
- 关系和行为: 用户可以播放音乐、查看当前播放音乐的详细信息、选择收藏音乐或下载音乐

音乐搜索模块:

- 类/对象: 用户、音乐搜索模块控制器、音乐、歌手
- 关系和行为: 用户根据歌名、歌手或歌词进行音乐搜索

2.使用StarUML绘制UML时序图(顺序图)。

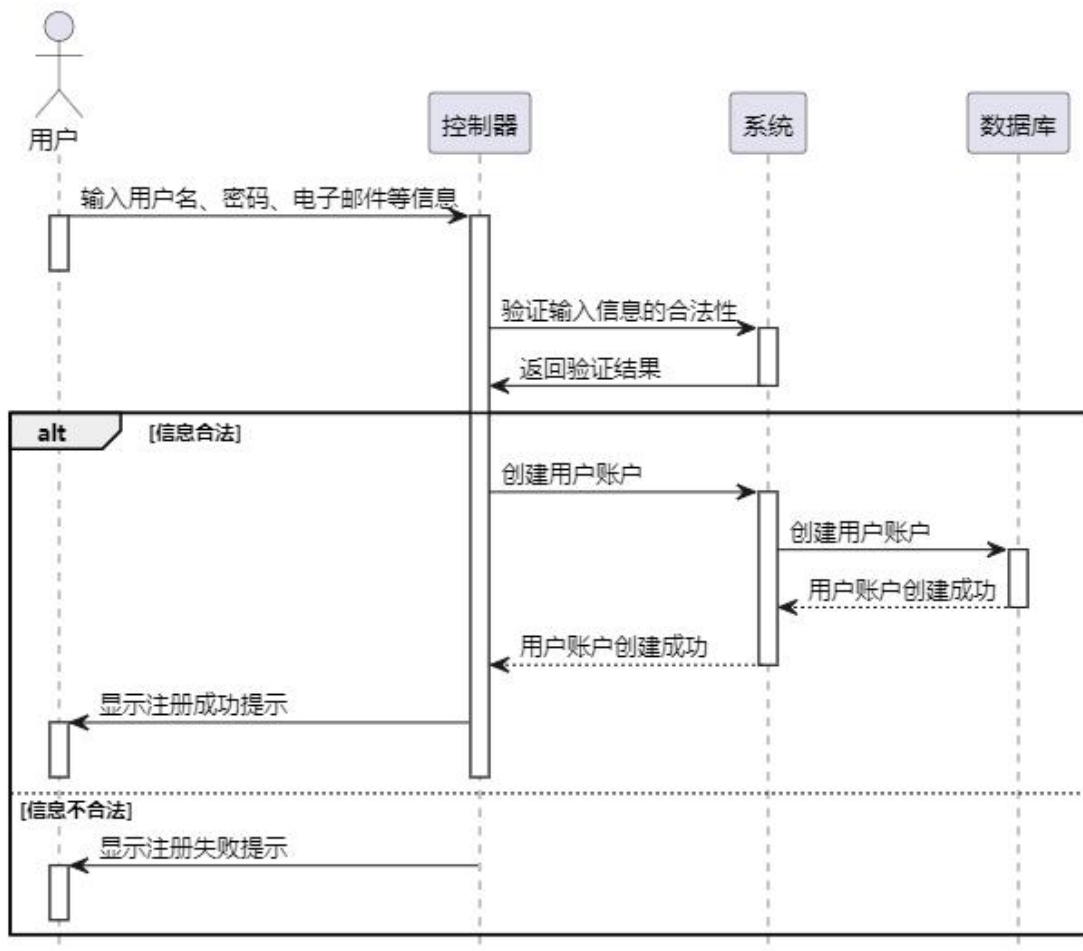
在如图的界面中选择时序图。



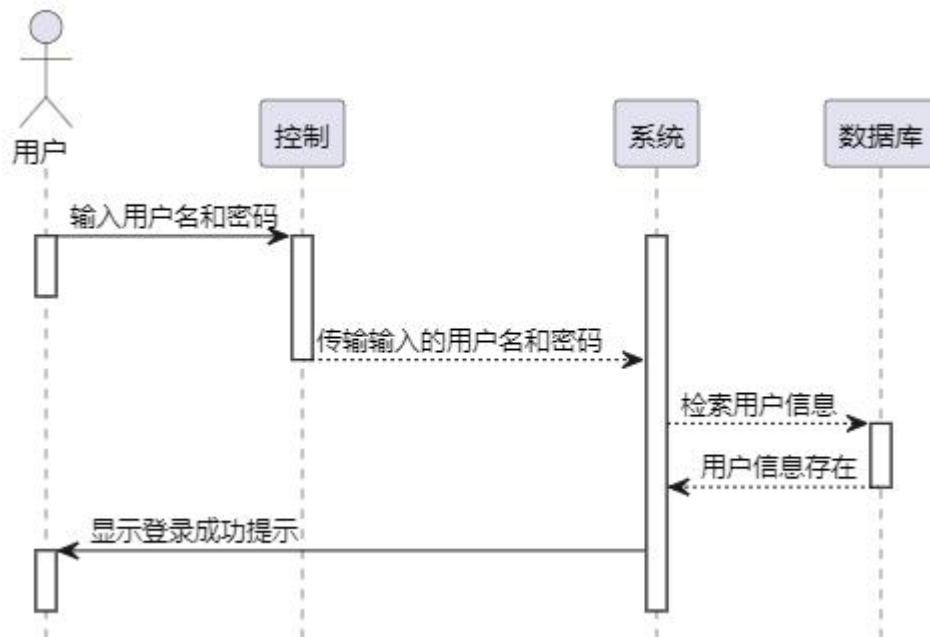
系统时序图

(1) 用户公共模块

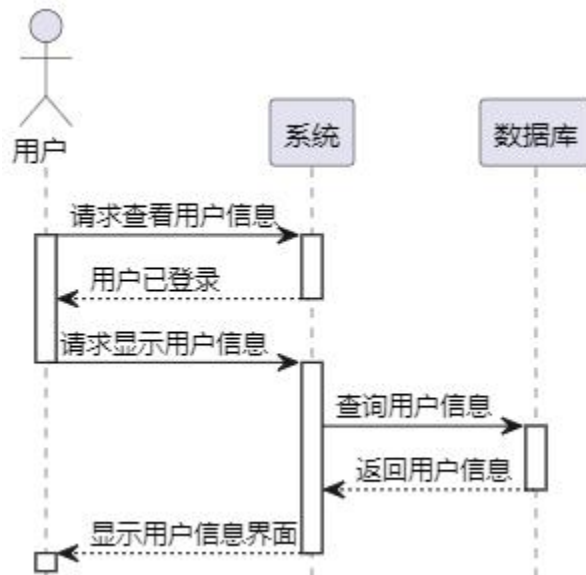
用户注册时序图：



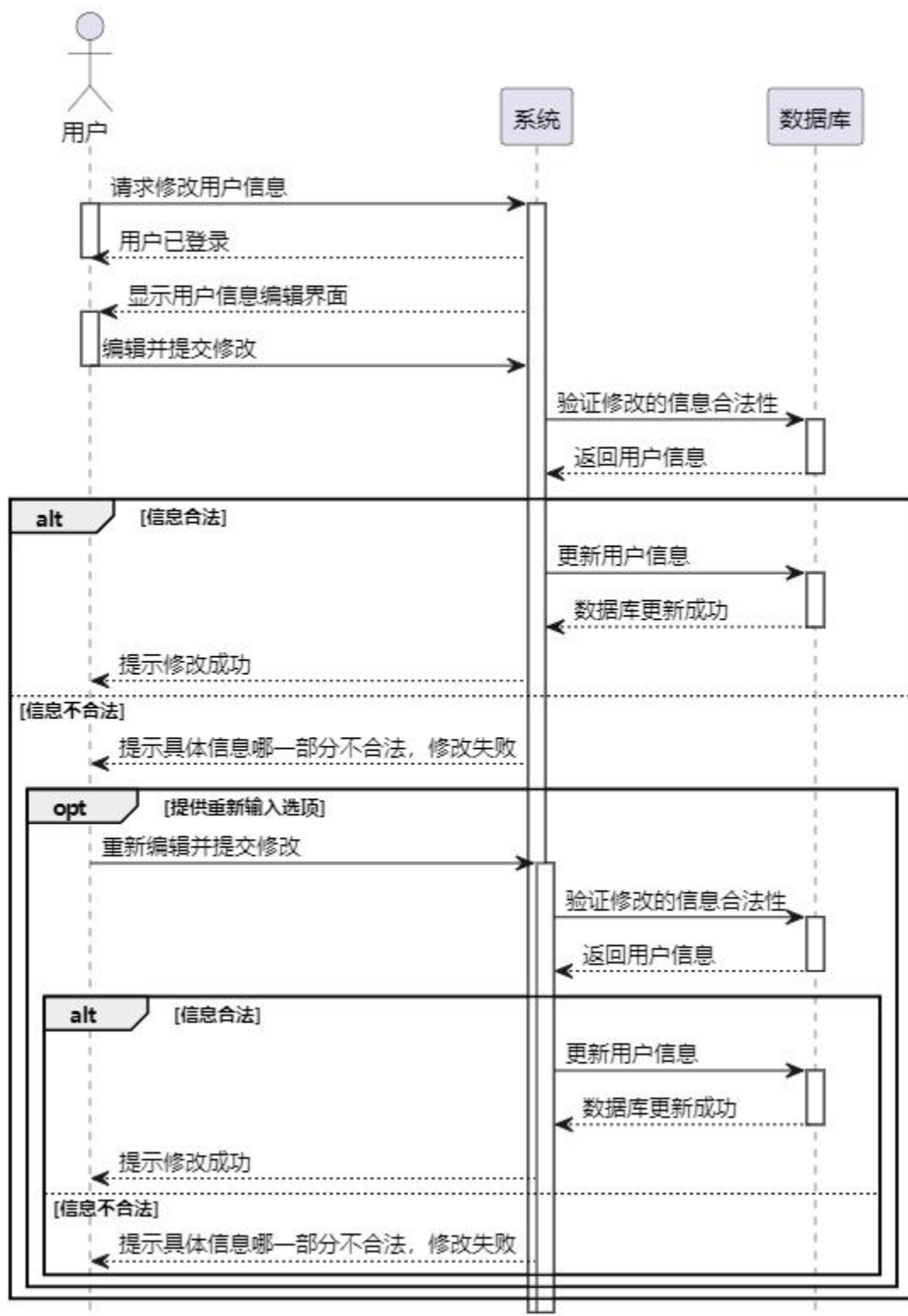
用户登录时序图：



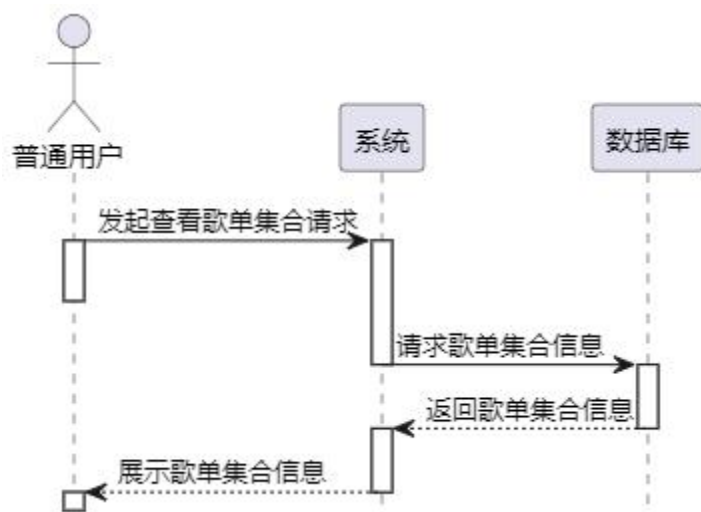
查看用户信息：



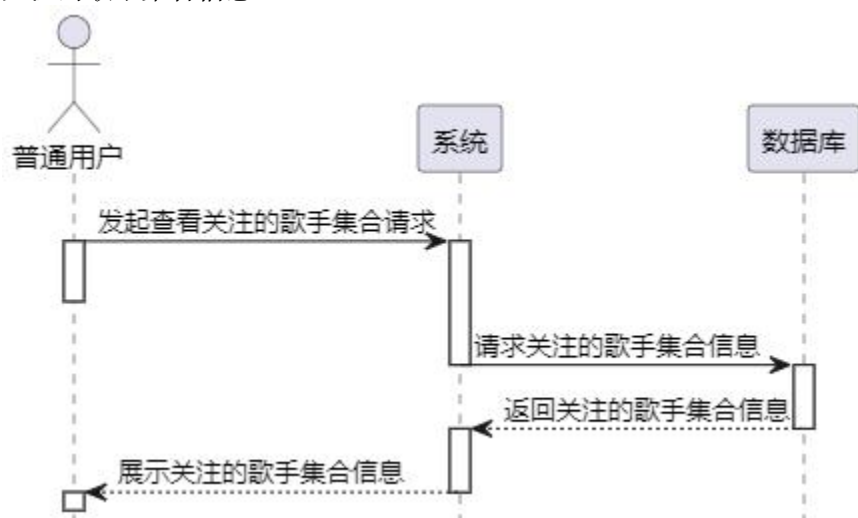
修改用户信息：



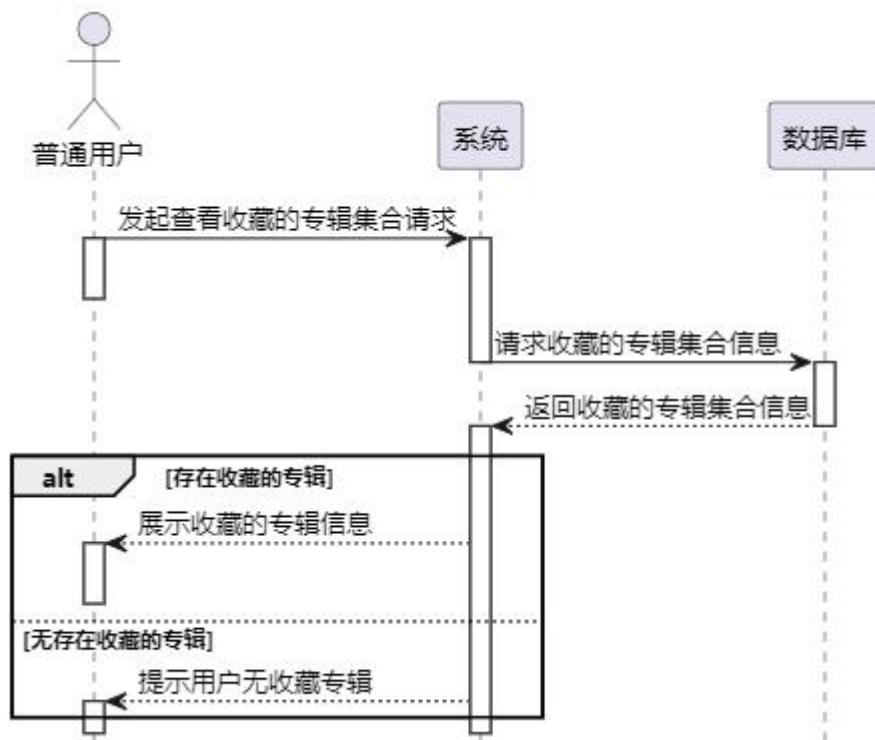
(2) 普通用户专有模块
查看创建的歌单集合:



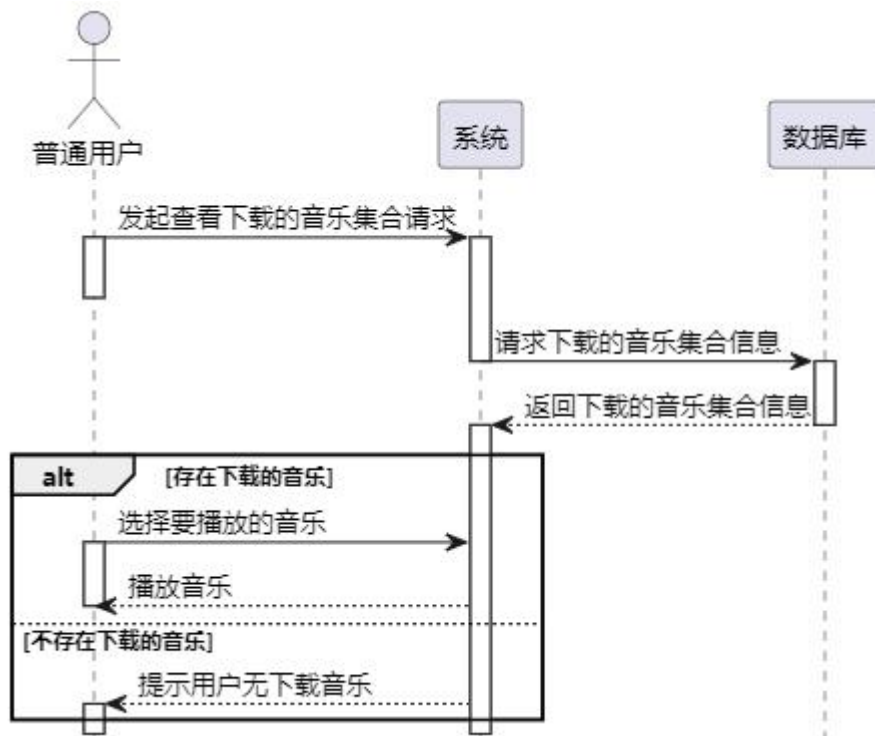
查看关注的歌手集合信息：



查看收藏的专辑集合：

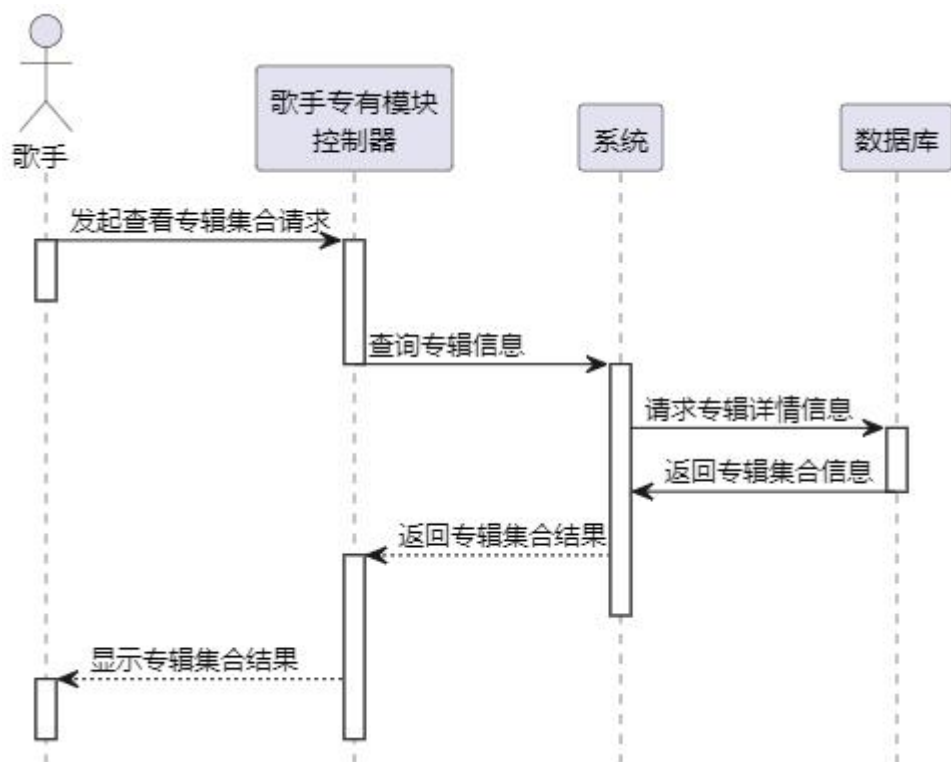


查看下载的音乐集合:

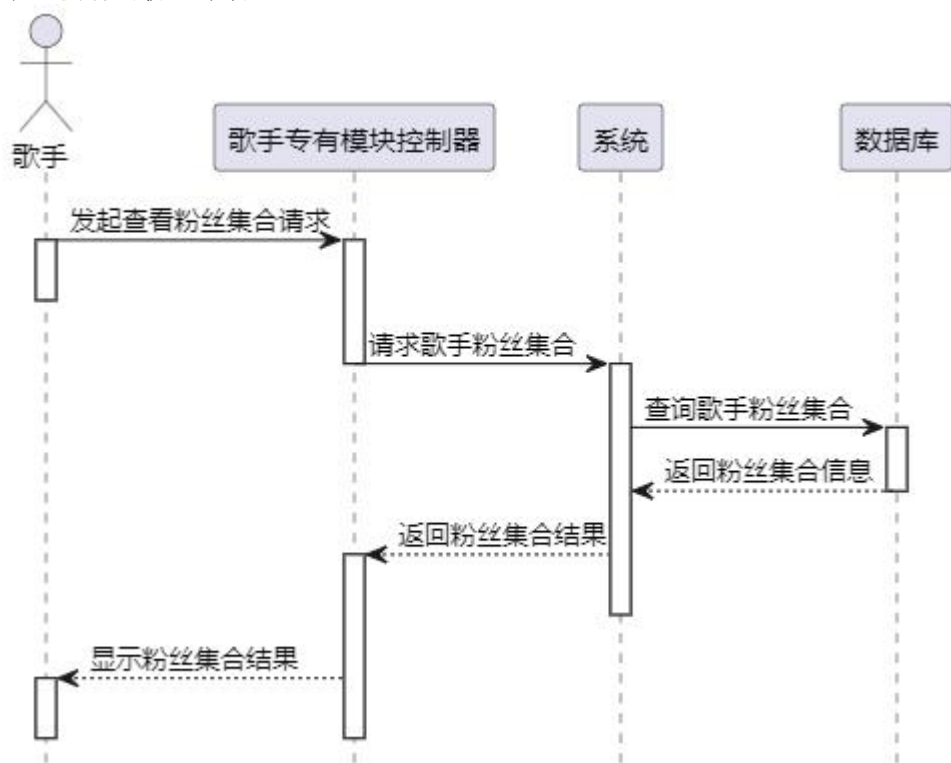


(3) 歌手专有模块

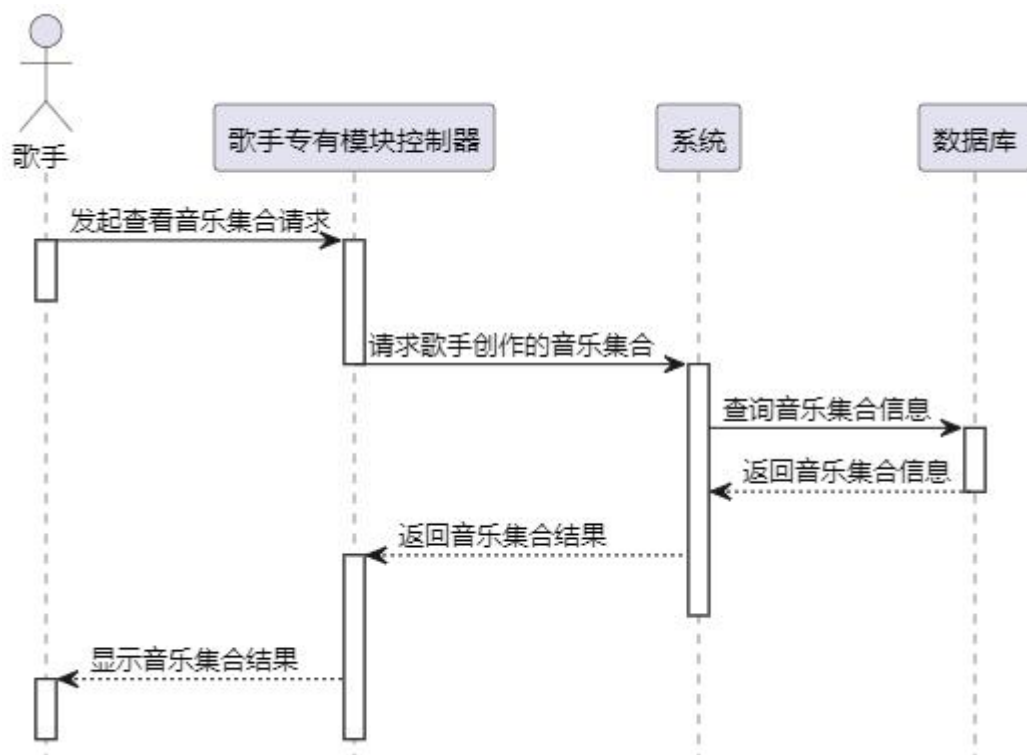
查看歌手创作的专辑集合:



查看歌手拥有的粉丝集合:

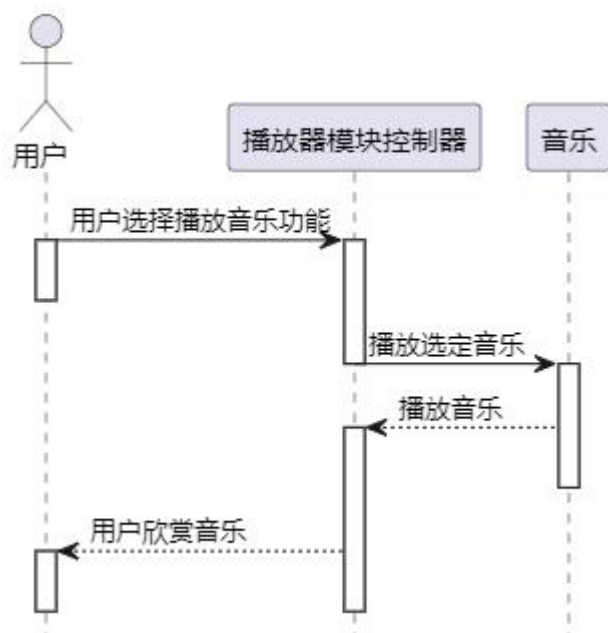


查看歌手创作的音乐集合:

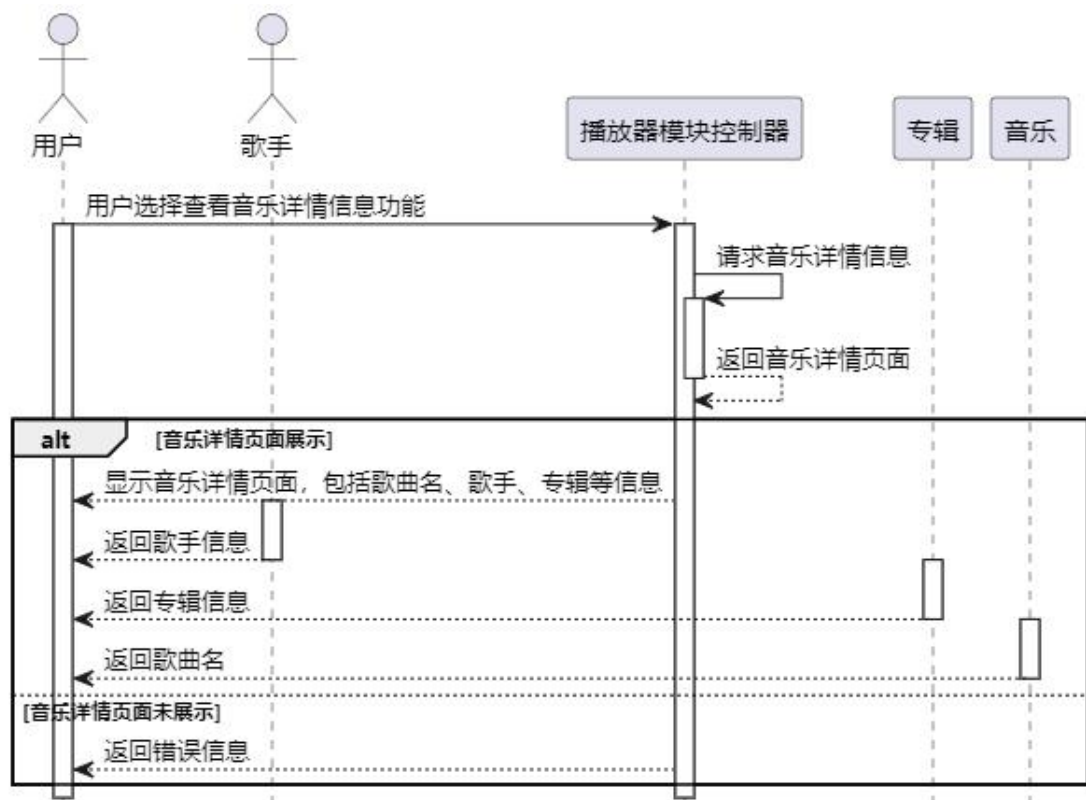


(4) 播放器模块

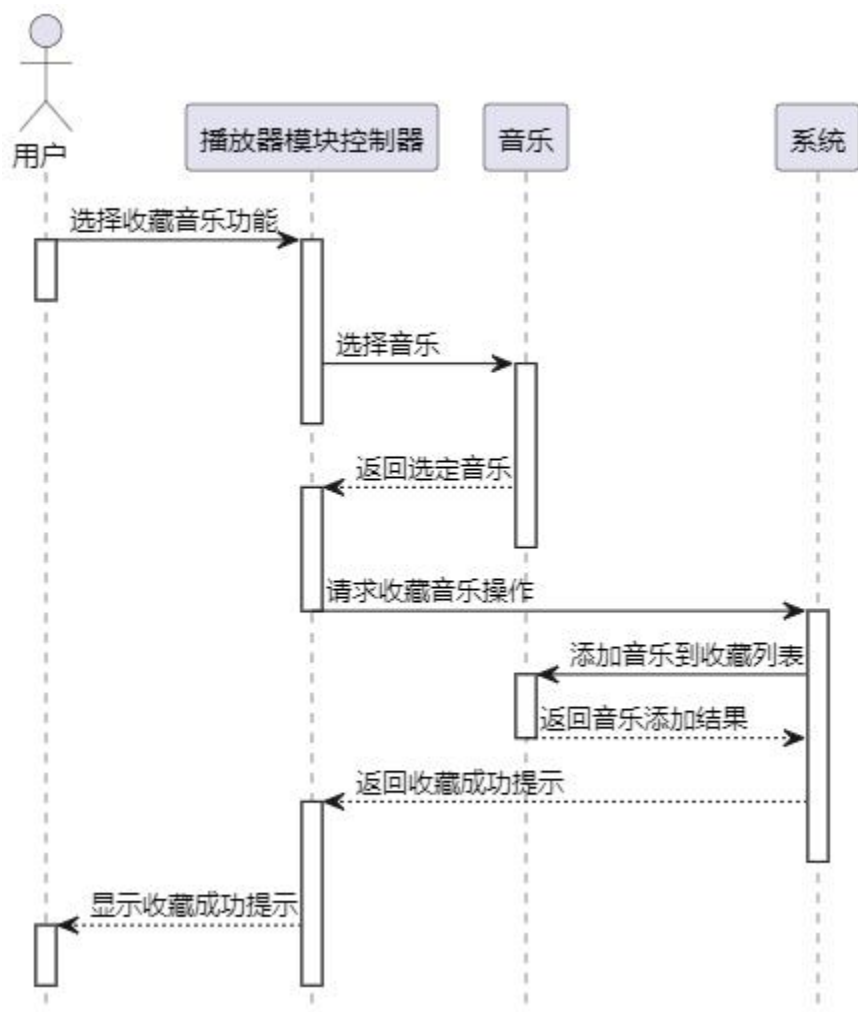
播放音乐时序图:



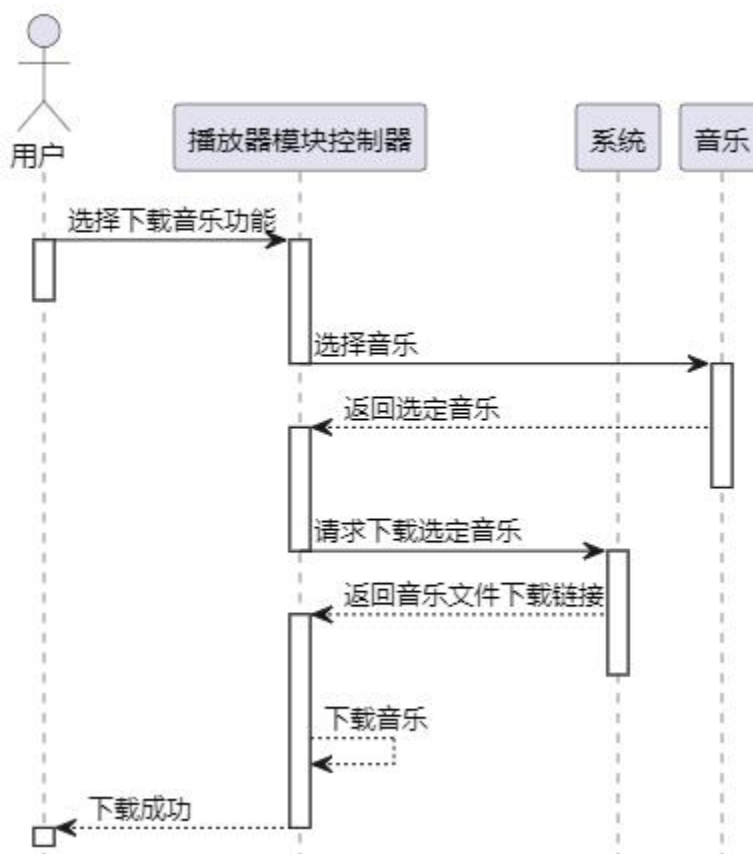
查看音乐详情信息:



收藏音乐时序图:

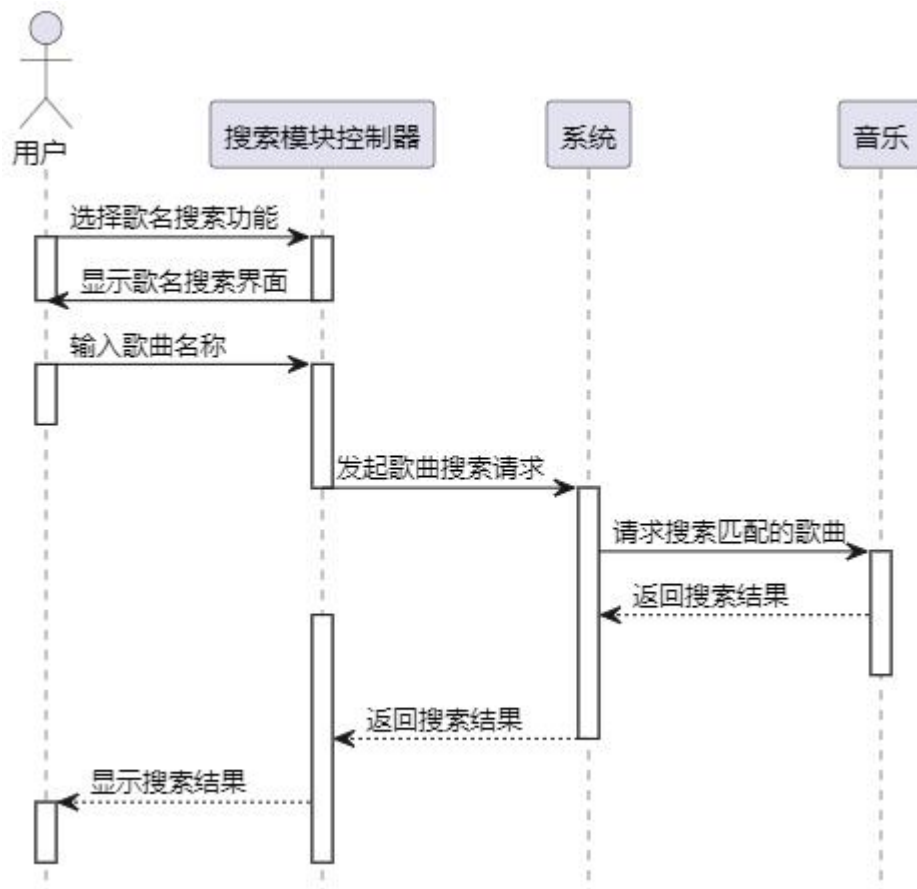


下载音乐时序图：

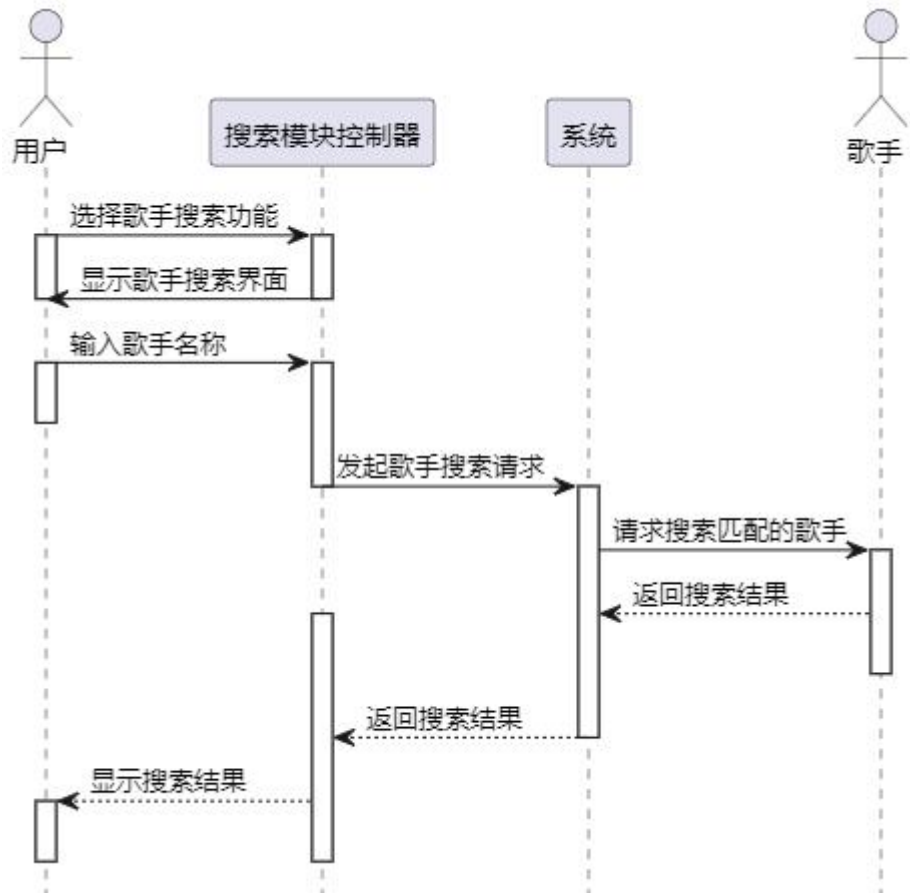


(5) 音乐搜索模块

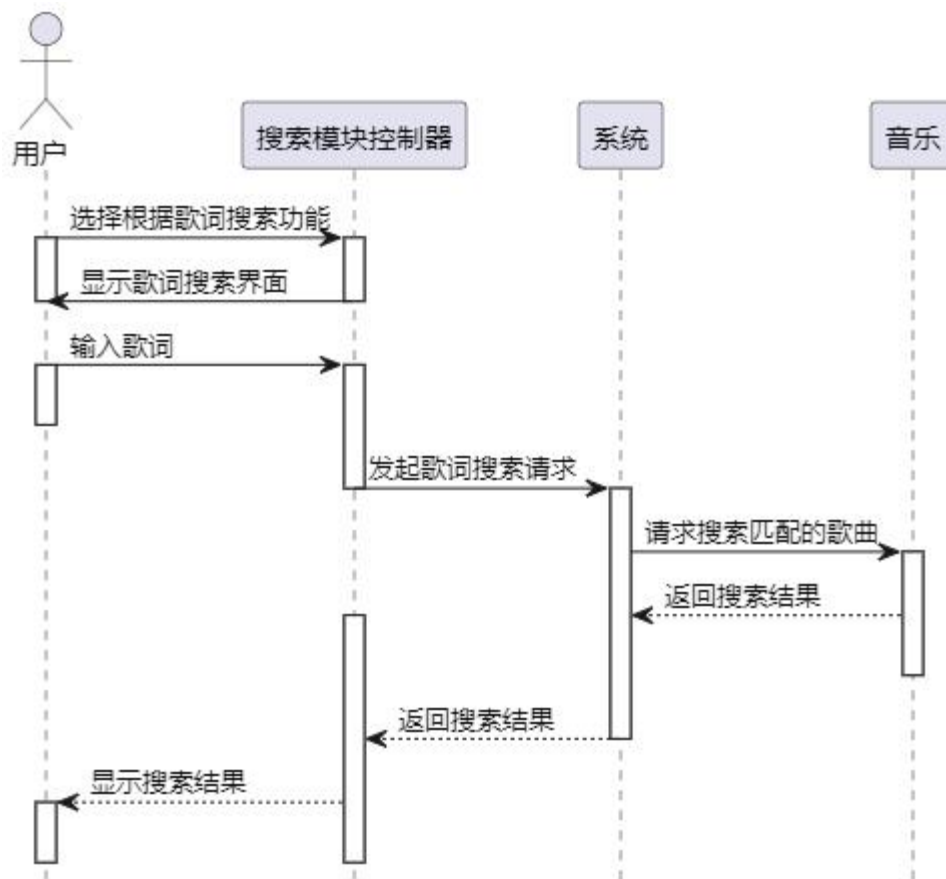
根据歌名搜索音乐时序图：



根据歌手名搜索音乐时序图：



根据歌词搜索音乐时序图：



3、实验分析与讨论

在本次实验中，通过分析用例所涉及到的类或对象，以及用例中的类或对象涉及的关系和行为顺序，并使用StarUML建模工具创建了相应的UML时序图。通过时序图展示各个模块之间的交互过程，并深入分析了用户公共模块、普通用户专有模块、歌手专有模块、播放器模块和音乐搜索模块之间的信息交换与流程。