

博影软件需求(分析)说明书(规约)

修订历史:

编写日期	SEPG	版本	说明	作者	评审时间	评审人员	批准日期	签 字 人 员
2020/11/30	Boying	1.0.01130_alpha	初稿	胡钰弦	2020/11/30	李翠琪	2020/11/30	李翠琪
2020/12/01	Boying	1.0.01201_alpha	数据建模	王浩然	2020/12/01	胡钰弦	2020/12/01	胡钰弦
2020/12/03	Boying	1.0.01203_alpha	数 据 建 模 修 改，添加功能 建模和行为建 模	胡钰弦	2020/12/03	王浩然	2020/12/03	王浩然
2020/12/05	Boying	2.0.01205_alpha	非功能需求	王浩然	2020/12/05	李翠琪	2020/12/05	李翠琪
2020/12/12	Boying	2.0.00112_beta	修改需求、完 成文档整合	胡钰弦	2020/12/12	李翠琪	2020/12/12	李翠琪

1. 引言..... 2

1.1. 背景 2

1.2. 参考资料..... 3

1.3. 假定和约束..... 3

1.4. 用户的特点..... 3

2. 功能需求 3

2.1. 系统范围 3

2.2. 系统体系结构 4

2.3. 系统总体流程 4

2.4. 需求分析..... 10

2.4.1. 功能建模..... 10

2.4.2. 数据建模.....	22
2.4.3. 行为建模.....	25
3. 非功能需求.....	34
3.1. 性能要求.....	34
3.1.1. 时间特性要求.....	34
3.2. 数据管理能力要求.....	35
3.2.1 常量约定.....	35
3.2.2 数据存储要求.....	35
3.3. 安全及保密性要求.....	35
3.4. 灵活性要求.....	35
3.5. 其他专门要求.....	35
3.5.1 灾难恢复.....	35
3.5.2 可维护性.....	36
4. 运行环境规定.....	36
4.1. 设备.....	36
5. 需求跟踪.....	36

1. 引言

1.1. 背景

新一代互联网主力人群由于缺乏群体记忆,加之互联网快速分化,个性化群体快速崛起,文化需求多元化加重。现场娱乐市场作为一个多元化市场,分类众多,且各具特色。但由于行业观演环境还有待改善和培育,因此整个市场尚待焕发生机。在市场及需求的驱动下,现场娱乐在线票务平台发展迅速,比如大麦网、摩天轮票务、体育易等等。正是基于这些软件

作为参考，制作一款具有特色的娱乐票务网站——博影。本项目的任务提出者为李翠琪，开发者为李翠琪、江一舸、王浩然、胡钰弦，该项目为有现场观看音乐剧、话剧等演出需求的用户提供演出信息与购票服务。

1.2. 参考资料

Pressman R S. Software Engineering A Practitioner's Approach[M]. Seventh Edition. McGraw- Hill Higher Education, 2017-1.

1.3. 假定和约束

发布时间：2020 年 12 月 27 日发布

开发条件：成员自备电脑，配置开发环境，能够支持产品正常运行。

经费支持：无经费支持

设计成本：自行承担设计成本

设备要求：因特网连接，有足够的带宽来充分显示各种显示；设备的 CPU 速度或 RAM 不是很大的问题。

具体使用的技术、工具和数据库：使用 Vue 前端框架 Java 后端框架和 MySQL 数据库；数据库不应该以纯文本形式存储密码，也不需要多次无效登录尝试后成为密码恢复功能或锁定。

使用限制：当使用 Firefox 和 Chrome 以外的互联网浏览器时，系统可能无法正常工作。

功能要求：系统应该能够发送验证码到用户的手机，以确保安全；该系统将使用电子票代替实物票。

1.4. 用户的特点

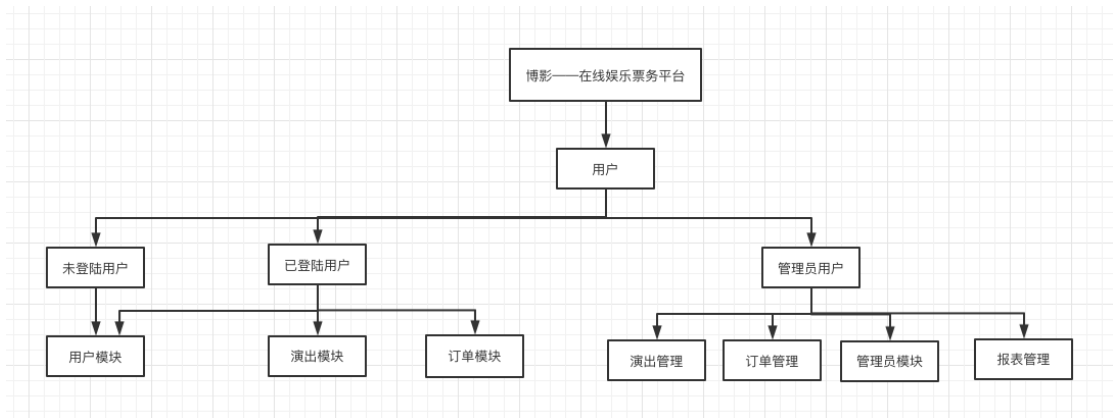
1. 喜欢看演出比如音乐剧、话剧、演唱会的用户
2. 热衷于追星的粉丝

2. 功能需求

2.1. 系统范围

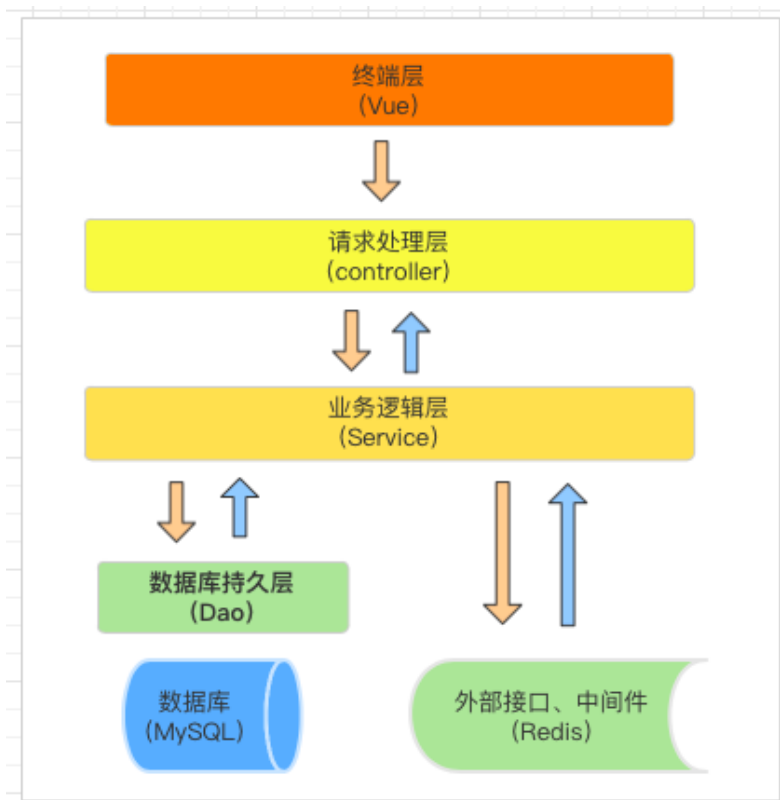
对于有线下观看话剧、音乐剧、演唱会等演出的需求的用户，我们提供了博影这个平台。使用用户能够获取最新的演出信息并购票，管理员用户能够管理后台演出的上架更新等。

利用方框图来说明该系统的组成：



2.2. 系统体系结构

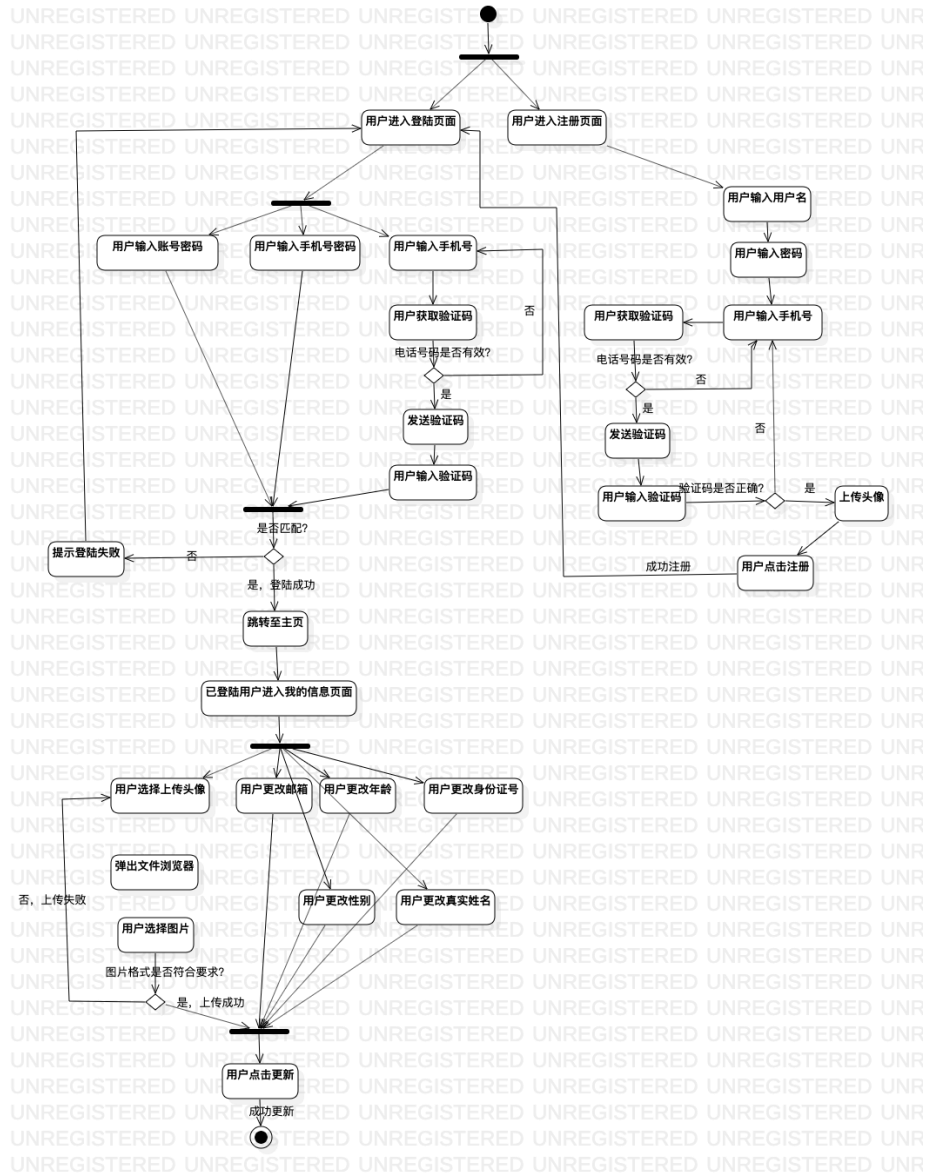
具体架构图如下：



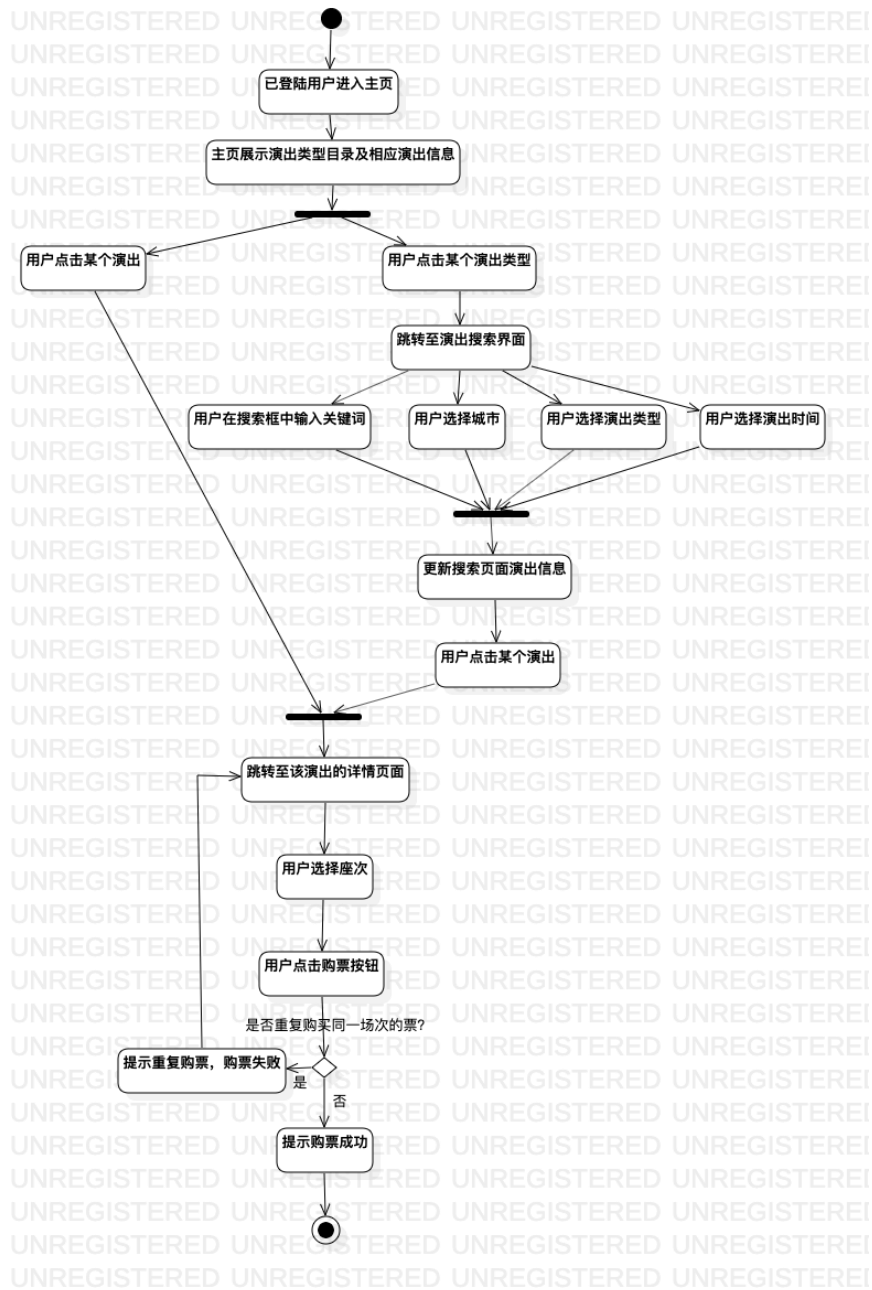
2.3. 系统总体流程

对于博影娱乐票务网站的使用用户的三大模块：用户模块、演出模块、订单模块分别构建了他们的整体流程图，如下：

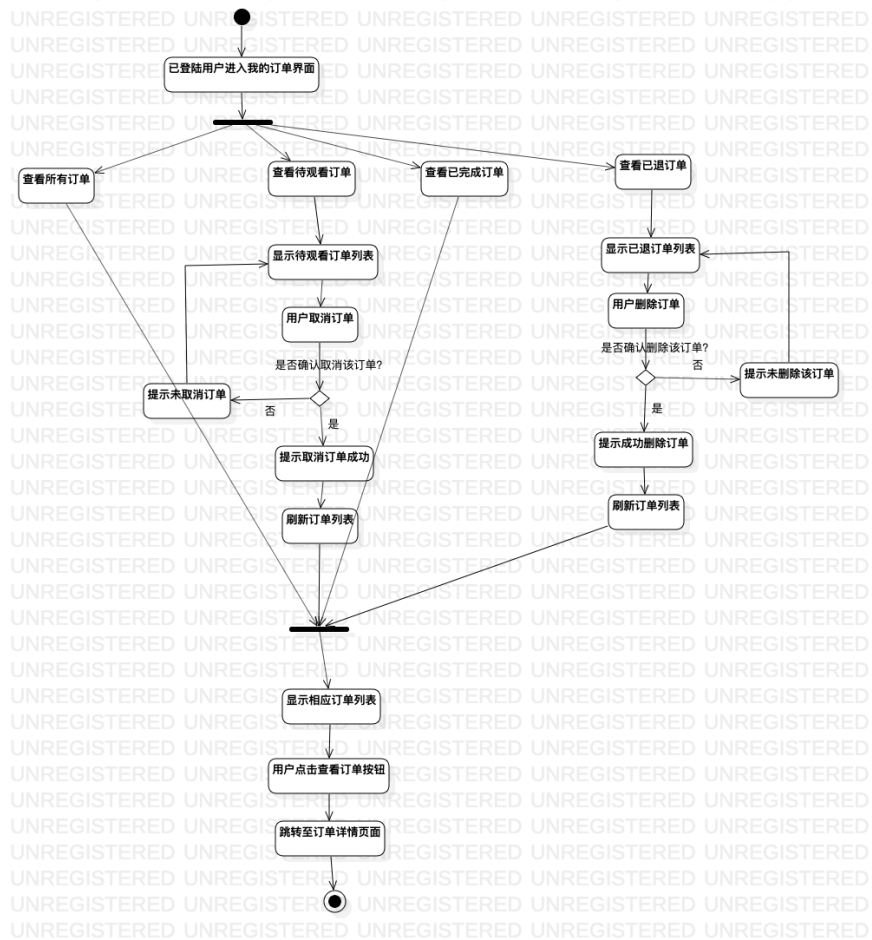
用户模块：



演出模块：

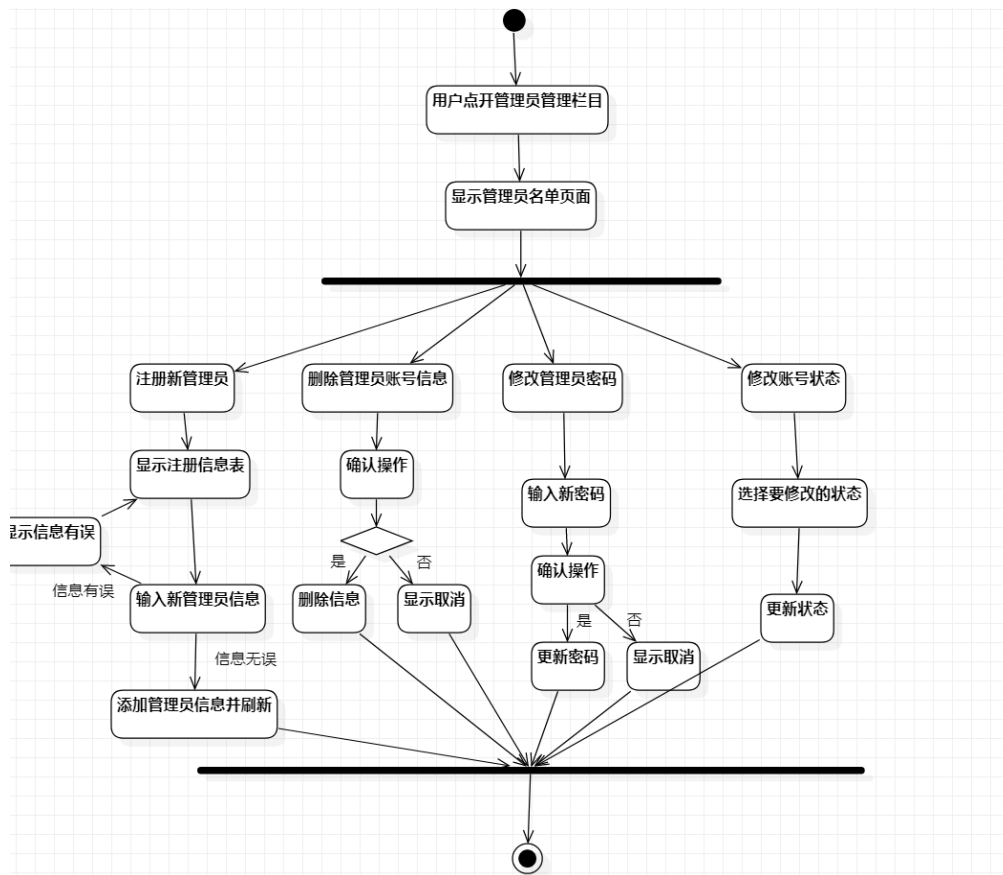


订单模块：

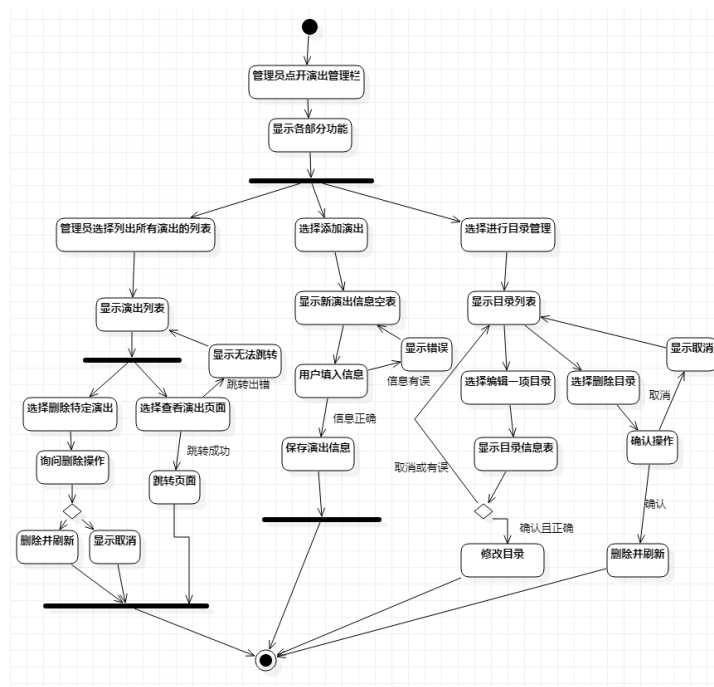


对于博影后台管理网站的管理员用户的四大模块：订单管理、演出管理、管理员模块、报表管理分别构建了他们的整体流程图，如下：

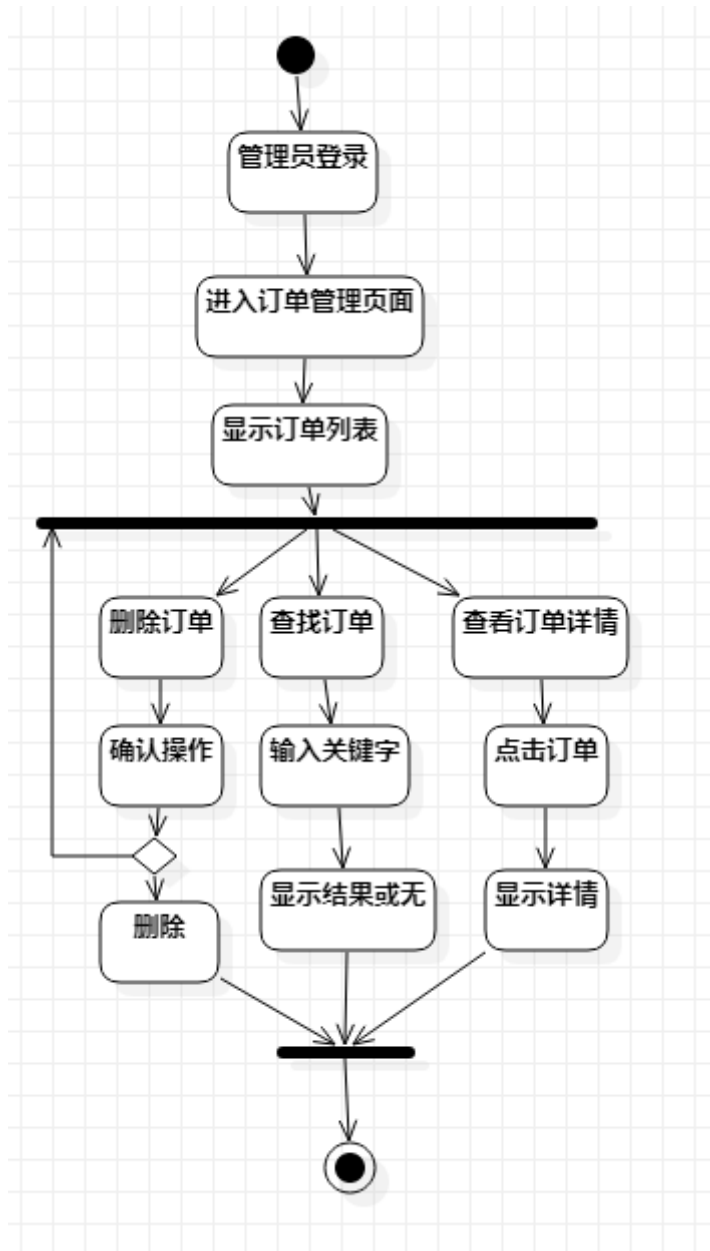
管理员管理模块：



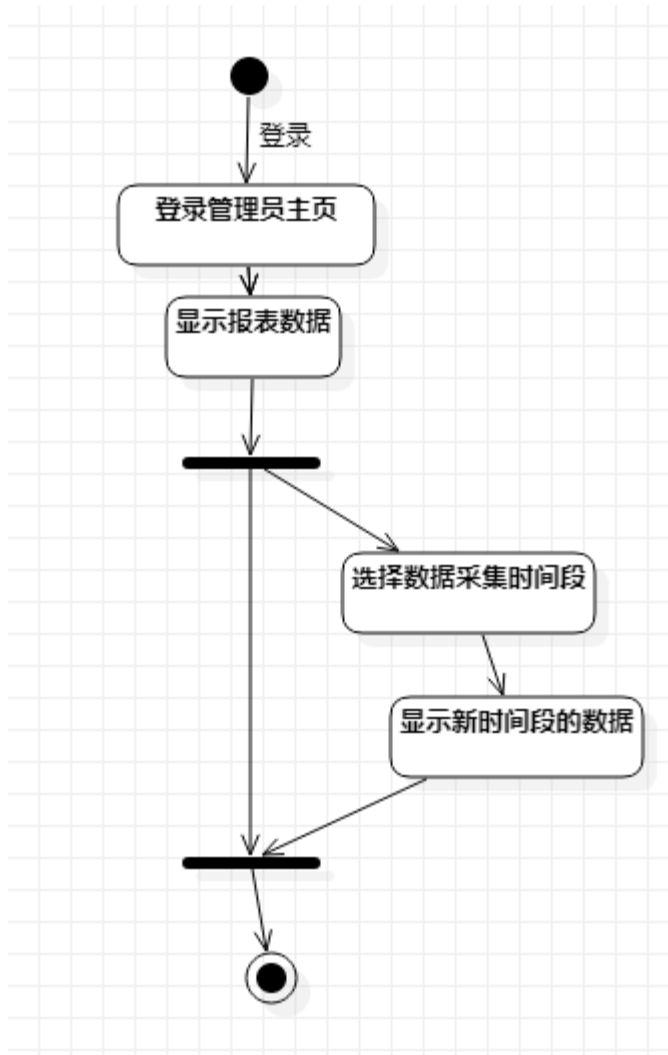
演出管理模块：



订单管理模块：



报表管理模块：



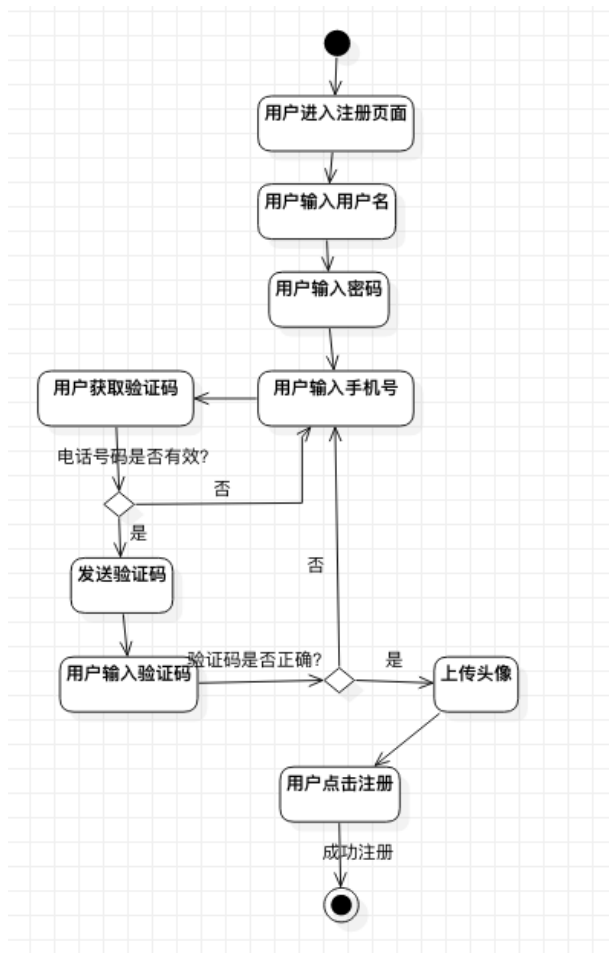
2.4. 需求分析

在这一部分，我们将进一步细化需求，如下是建模过程及模型产品。

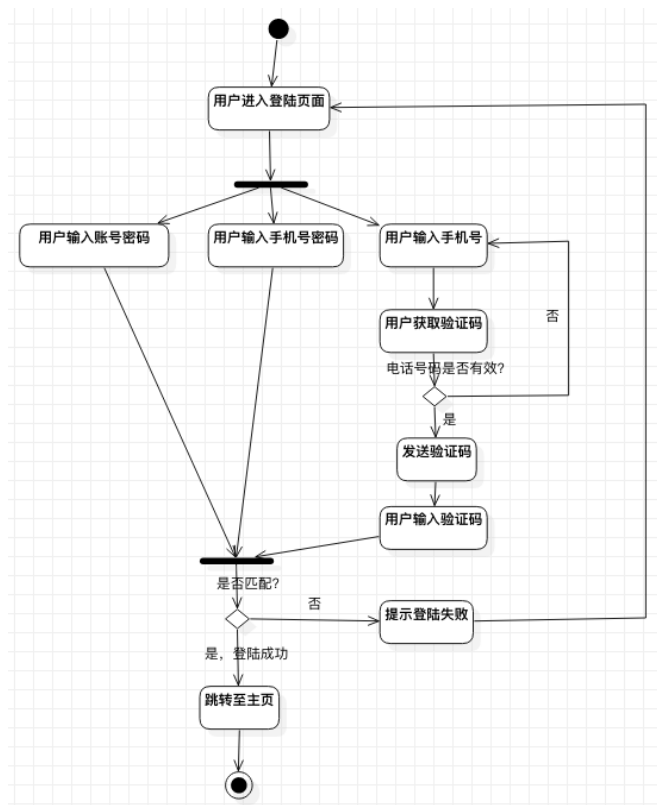
2.4.1. 功能建模

2.4.1.1 用户模块：

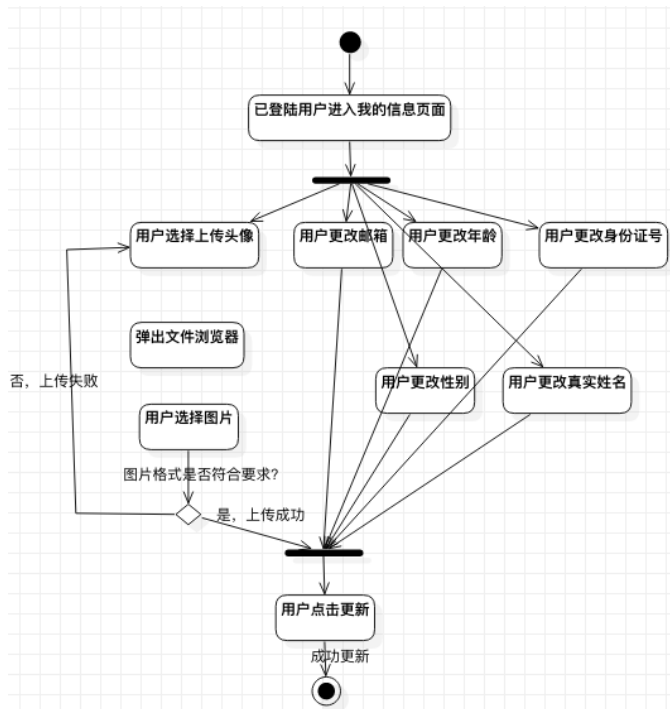
注册：



登陆：

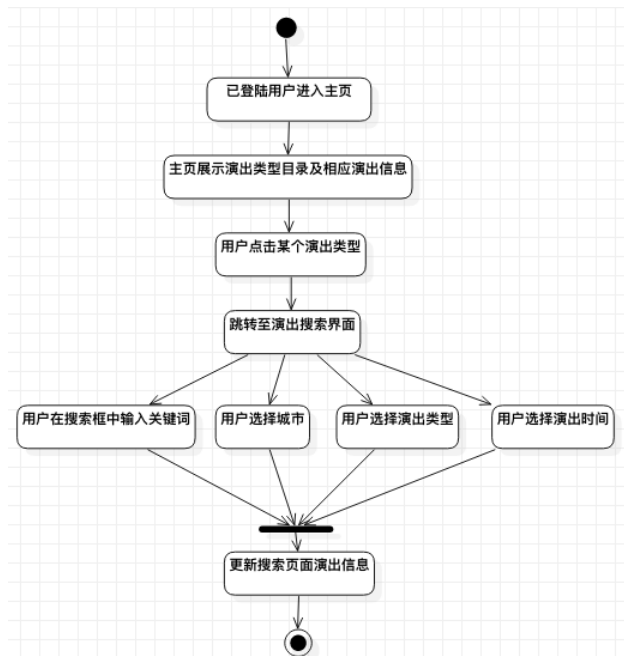


管理个人信息：

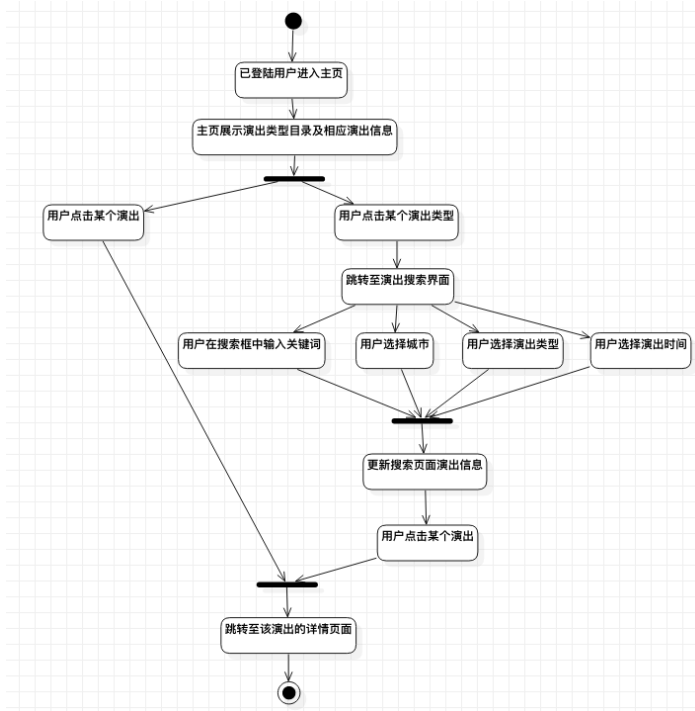


2.4.1.2 演出模块：

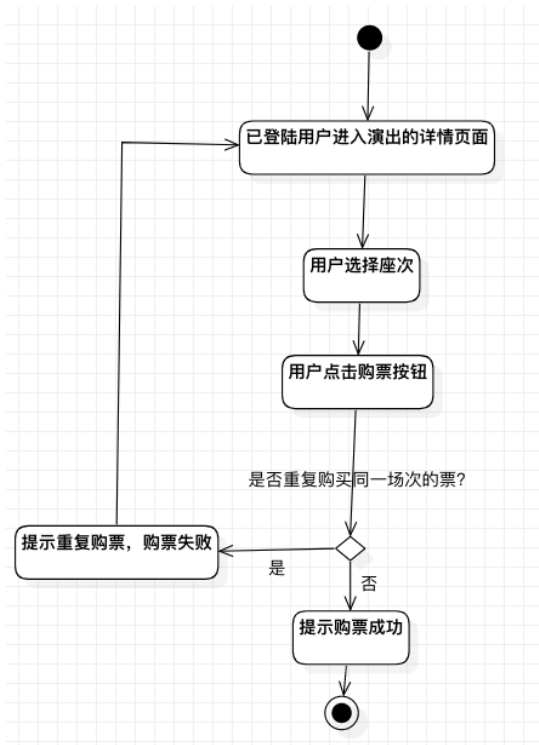
搜索演出：



查看演出详情：

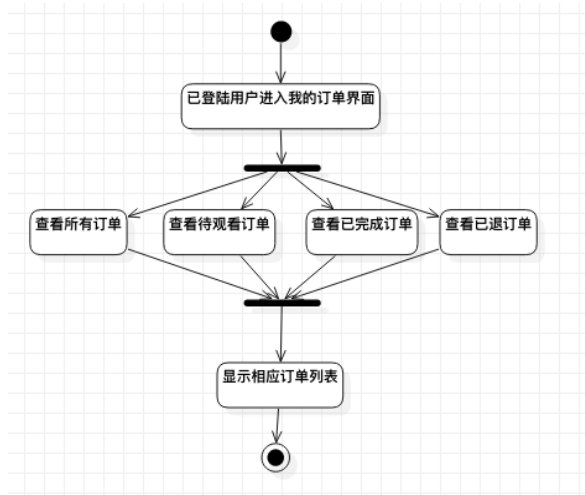


购票：

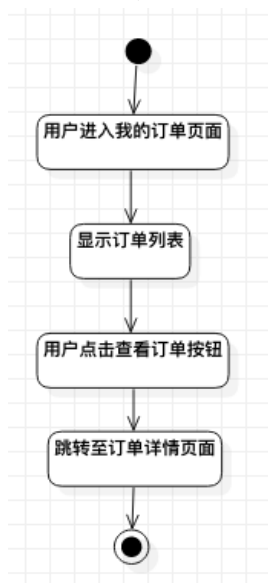


2.4.1.3 订单模块：

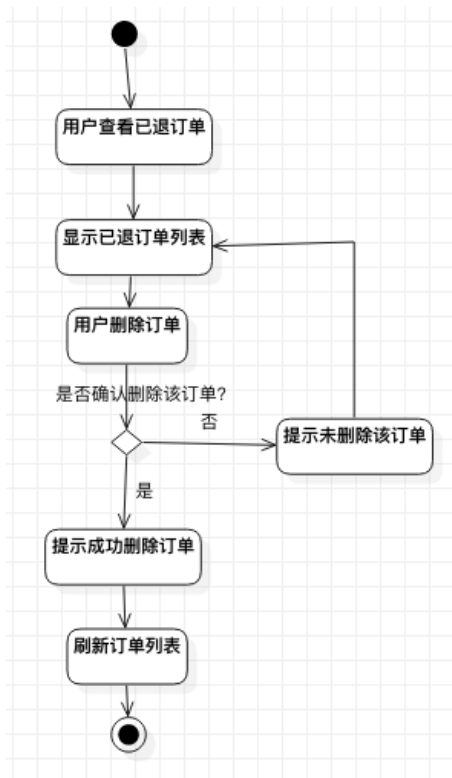
查看订单：



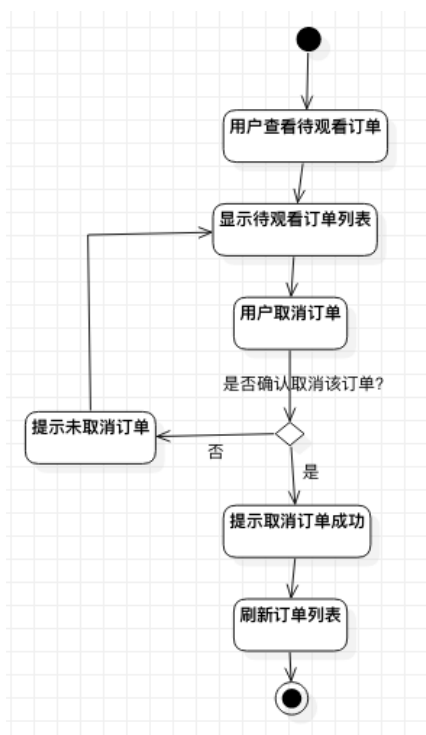
查看订单详情:



删除订单:

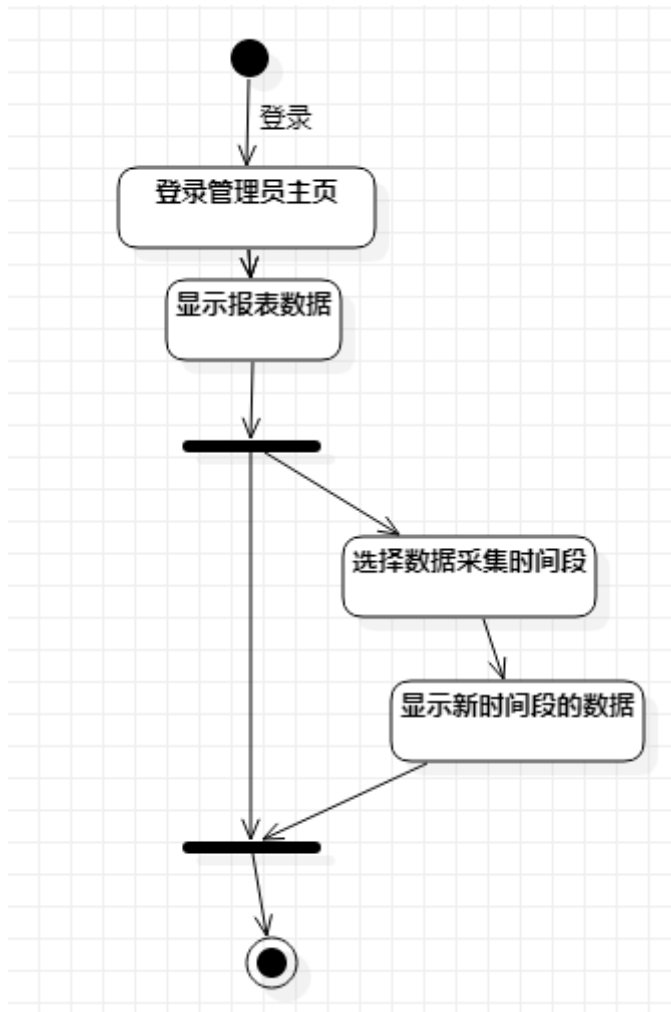


取消订单:



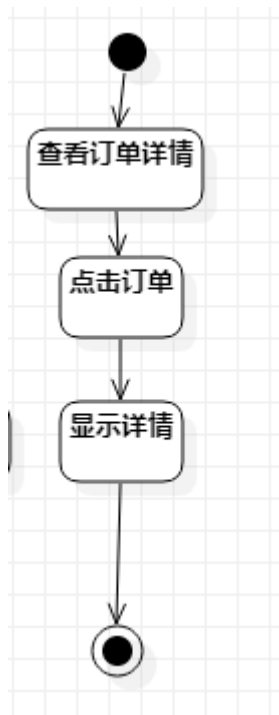
管理员模块:

报表显示:

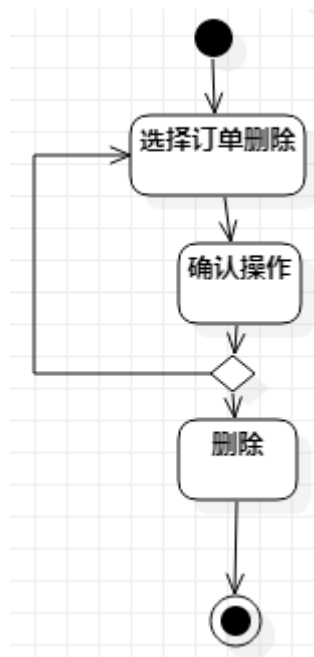


订单模块：

查看订单详情：



删除订单：

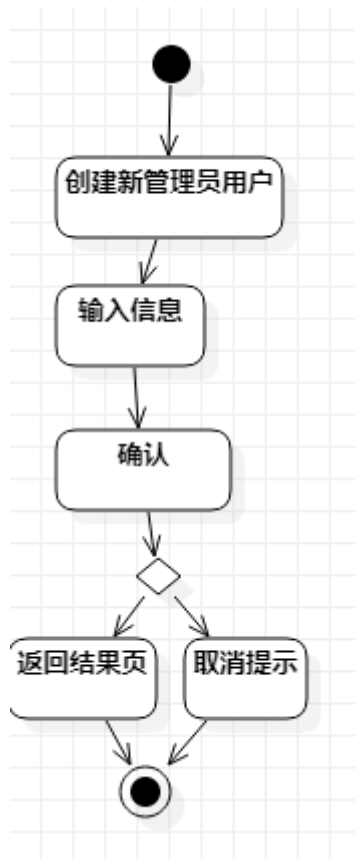


查询特定状态订单：

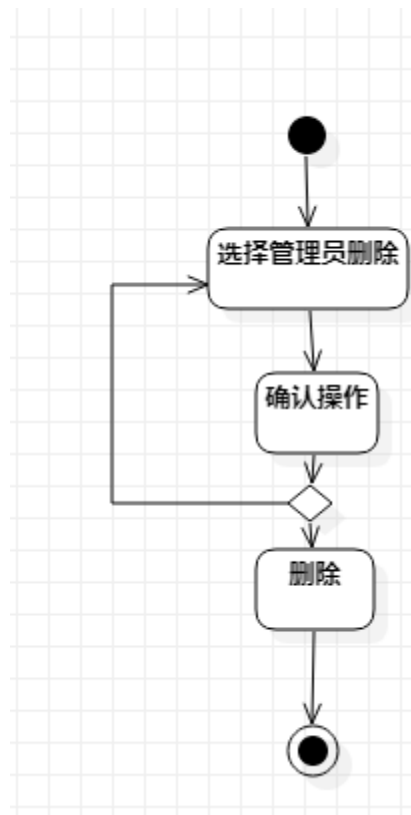


管理员管理：

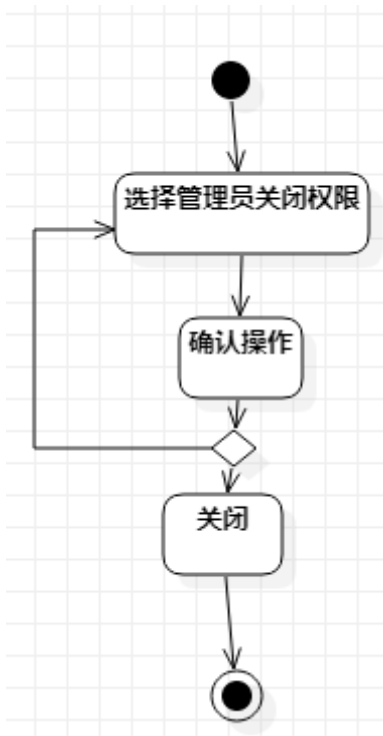
创建管理员：



删除管理员:

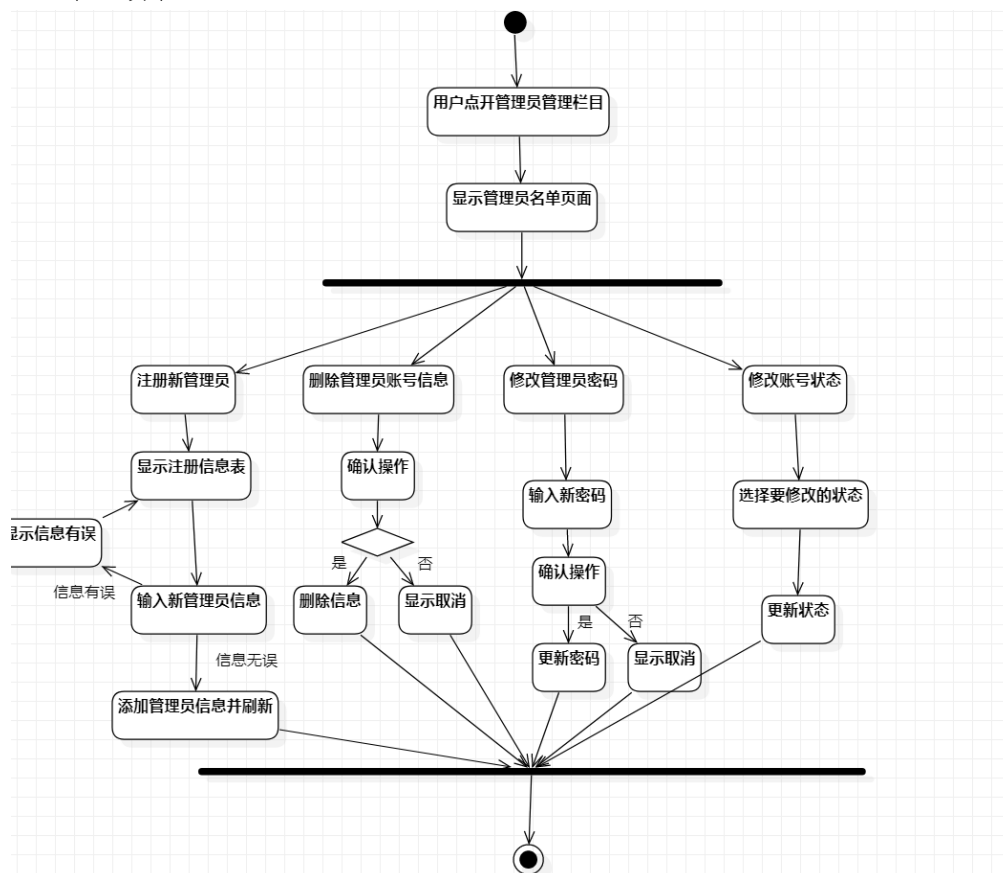


关闭权限:

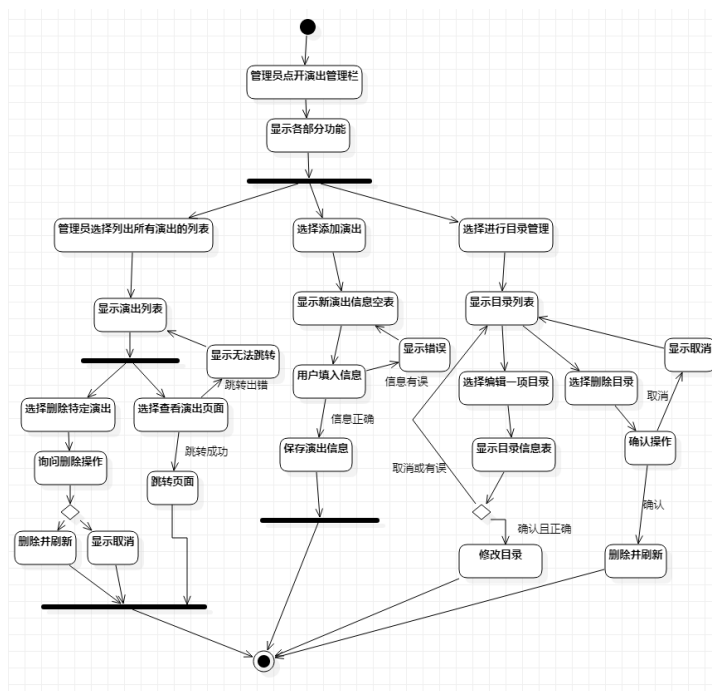


2.4.1.4 管理员部分：

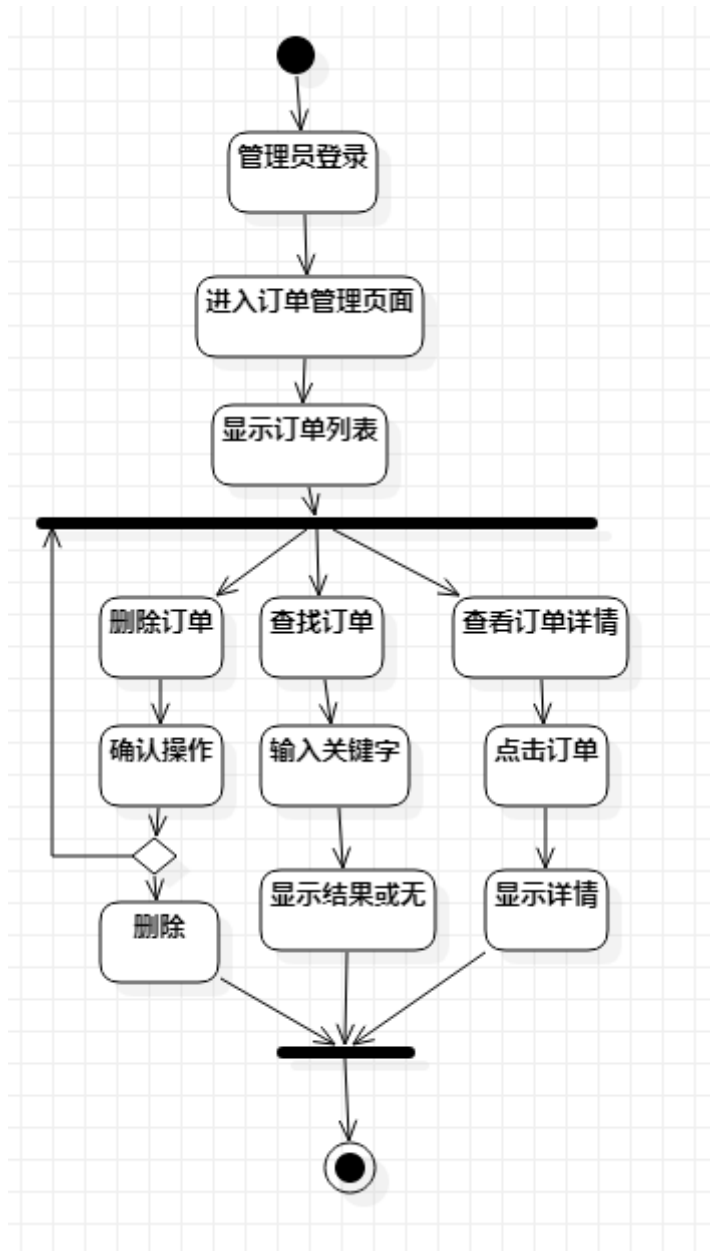
管理员管理：



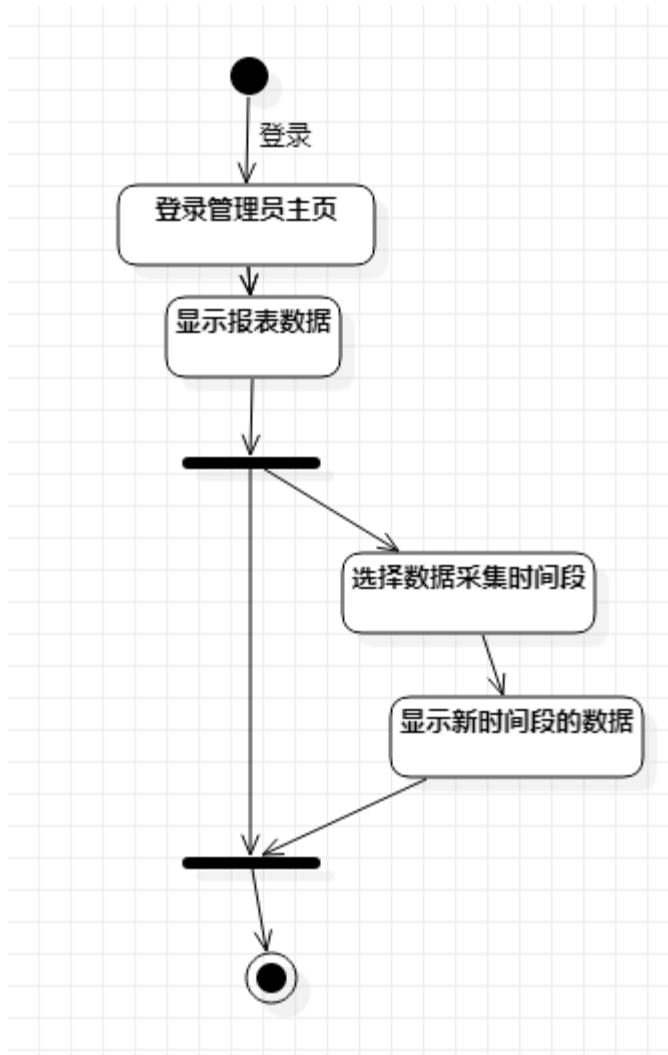
演出管理：



订单管理：



报表管理:



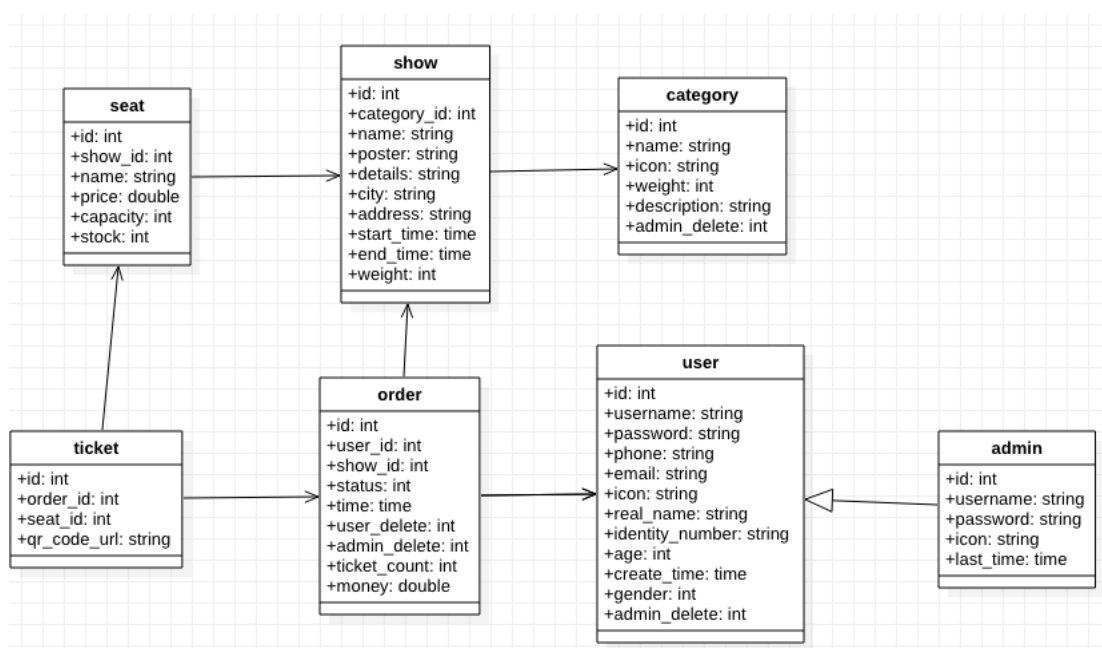
2.4.2. 数据建模

根据需求规约，以 use case 为单位，对文字描述的需求进行语法扫描，提取出有实际意义的名词/名词词组，分析筛选出以下潜在类，并且对潜在类接受或拒绝的六大特征进行进一步的分析讨论。

潜在类	一般分类	处理
用户	角色	接受
账号	事物	接受→拒绝
密码	事物	拒绝
手机号	属性	拒绝
验证码	事物	拒绝
身份证号	属性	拒绝
主页	组织单元	拒绝
页面	组织单元	拒绝

演出	事物	接受
演出目录	事物	拒绝->接受
城市	属性	拒绝
时间	属性	拒绝
演出座次	事物	接受
演出场次	事物	接受->拒绝
票	事物	接受
订单	事物	接受
订单列表	组织单元	拒绝
管理员用户	角色	接受

分析类图如下：



管理员用户继承自用户类；一个用户可以有多个订单，一个订单只能属于一个用户；一个订单可以有多张票，一张票只能属于一个订单；一个演出可以有多个订单，一个订单只能对应一个演出；一个演出只能属于一个目录，一个目录能有多个演出；一个演出可以有多个座次，一个座次只能属于一个演出；一张票可以有多个座次，一个座次只能属于一张票。

每个分析类的详细介绍如下：

2.4.2.1 用户

属性：用户 ID，账号，手机号，密码，真实姓名，身份证号，邮箱，年龄，性别，头像，创建时间，状态。

操作：输入账号或手机号和密码，进入主页，输入搜索信息，购票，查看订单。

说明：用户的基本信息，以及一系列基本活动。被已登陆用户、未登陆用户与管理员用户继承，是最基础的角色类。用户的所有属性都是持久的，它存储了注册用户的基本信息。

2.4.2.2 演出

属性：演出 ID，演出名，所属类型，海报，权重，城市，具体地址，开始日期，结束日期，详情。

说明：演出的基本信息。演出类中所有属性都是持久化的，它负责存储演出的基本信息。

2.4.2.3 演出目录

属性：演出类型 ID，目录名称，权重，图标，文字描述，状态。

说明：便于对各种各样的演出进行统一管理。

2.4.2.4 演出座次

属性：演出座次 ID，所属演出 ID，座次名称，库存，容量，定价。

说明：演出的座次信息。

2.4.2.5 票

属性：票 ID，所属订单 ID，所属座次 ID，二维码图片。

说明：演出的票信息。

2.4.2.6 订单

属性：订单 ID，所属用户 ID，所属演出 ID，订单状态，提交时间，用户删除状态，管理员删除状态，总金额，票数量。

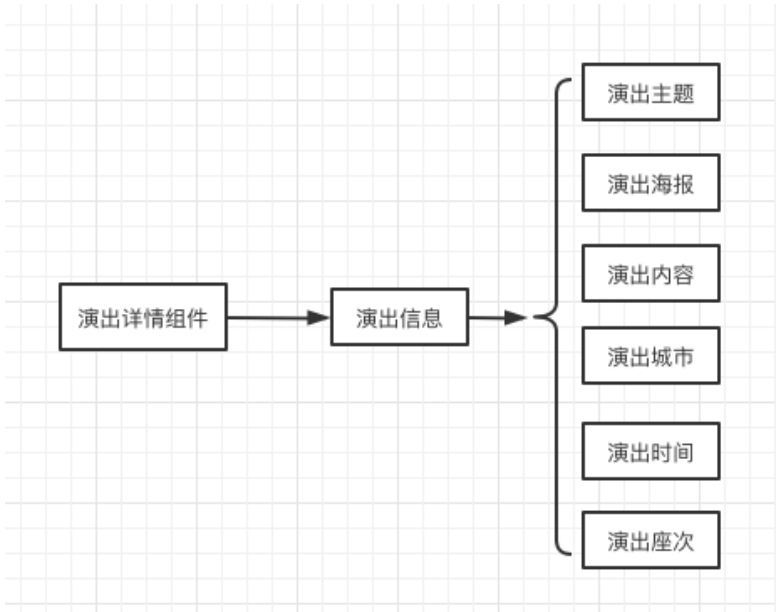
说明：用户的订单信息。

2.4.2.7 管理员用户

属性：管理员 ID，管理员账号，密码，头像，上一次登陆时间。

说明：用户的常用联系人信息。

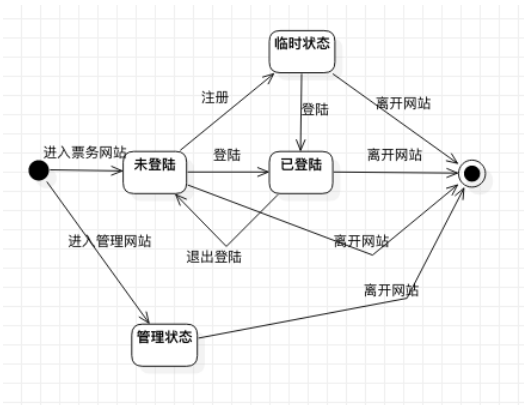
对于演出组件的内容，我们做了如下建模：



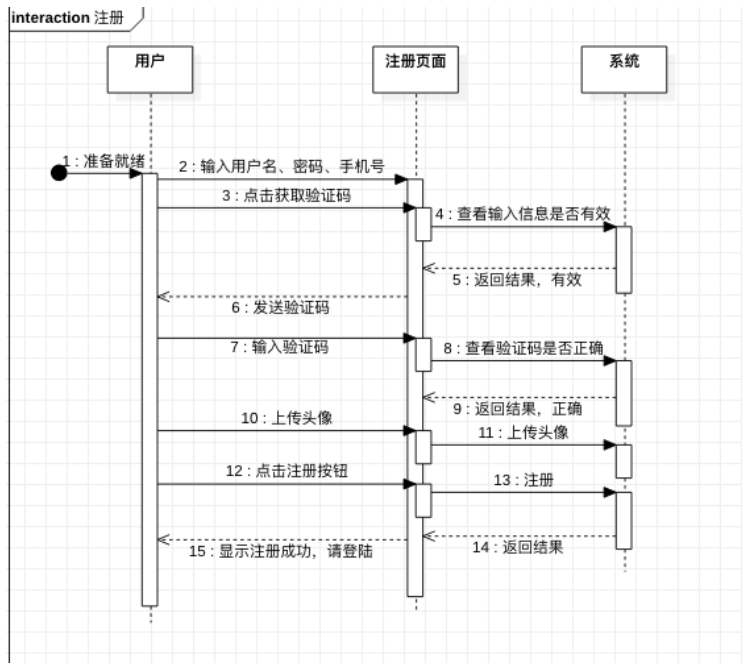
2.4.3. 行为建模

2.4.3.1 用户模块：

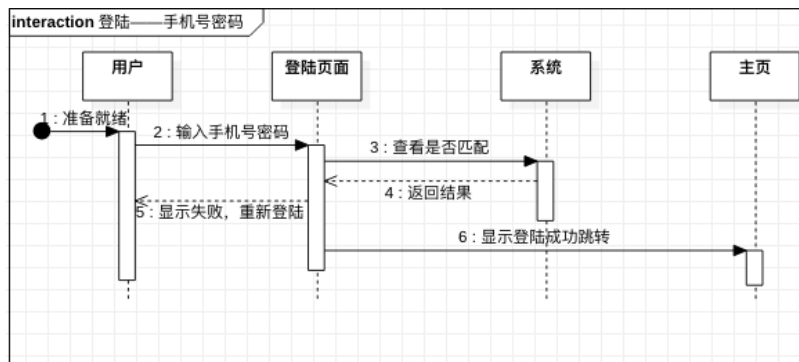
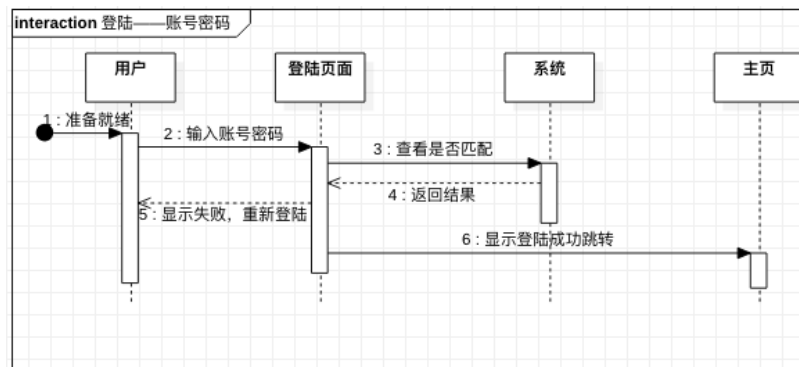
用户分析类状态图：

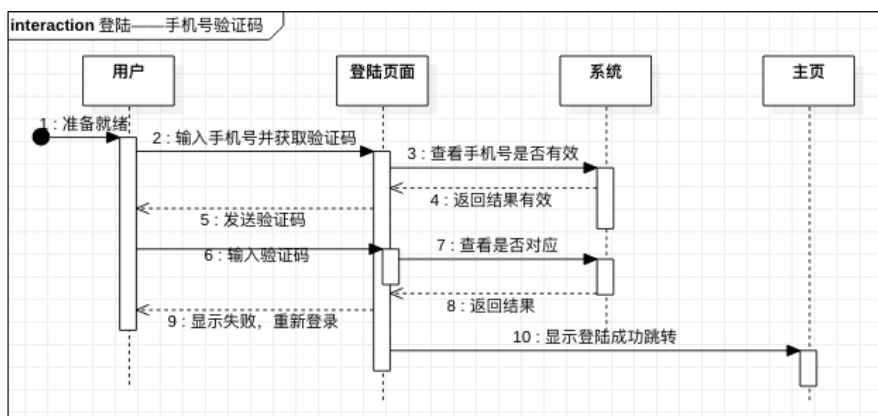


用户有登录，未登录、管理员等状态，使用不同的方法后，状态发生变化。
注册：

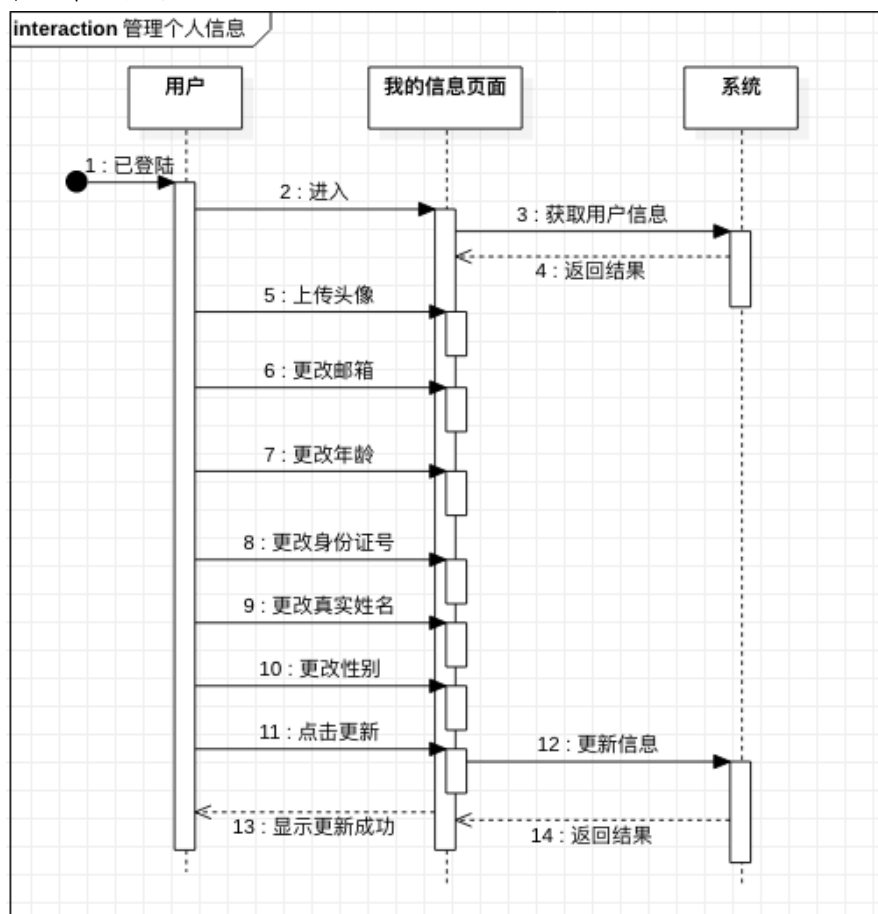


登陆:



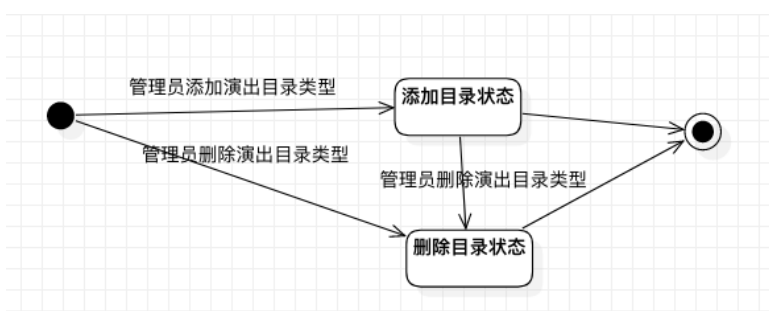


管理个人信息:

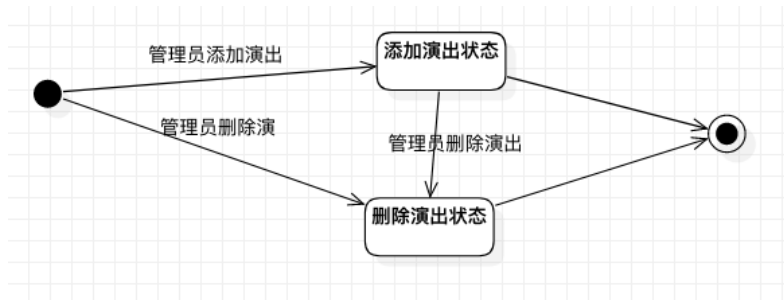


2.4.3.2 演出模块:

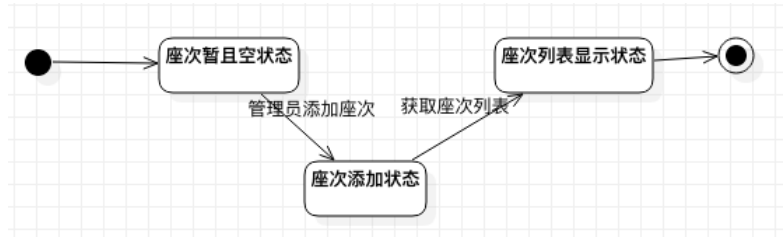
演出目录分析类状态图:



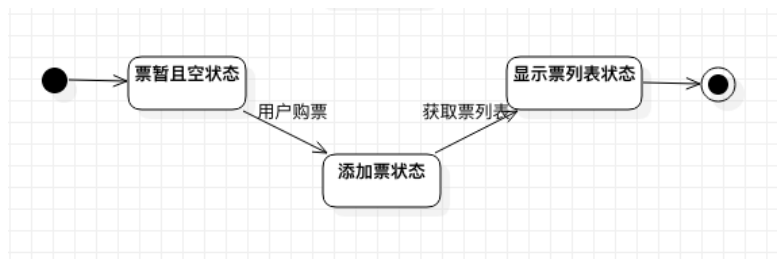
演出分析类状态图：



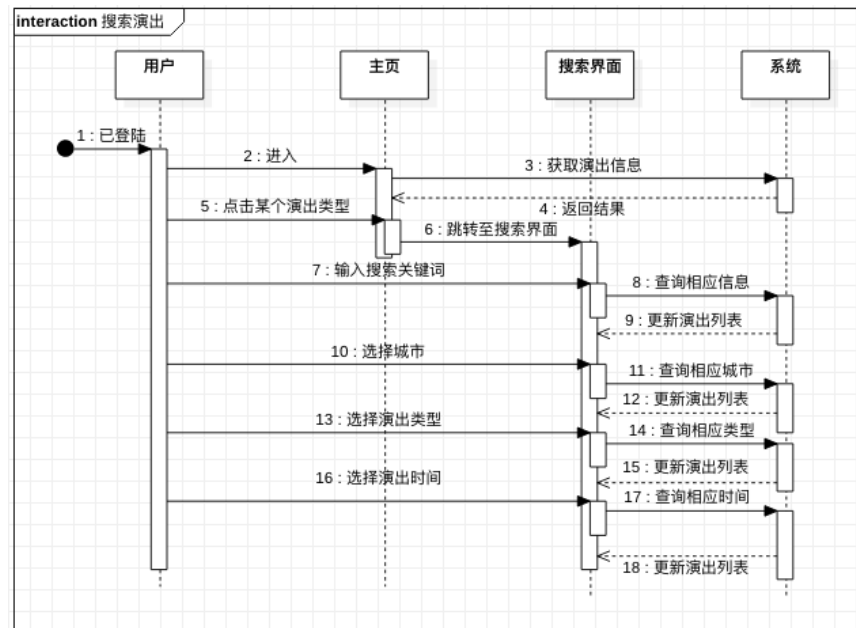
座次分析类状态图：



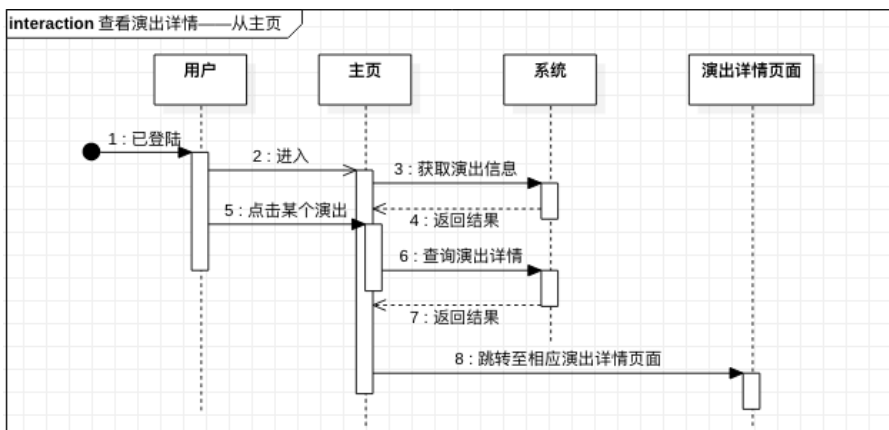
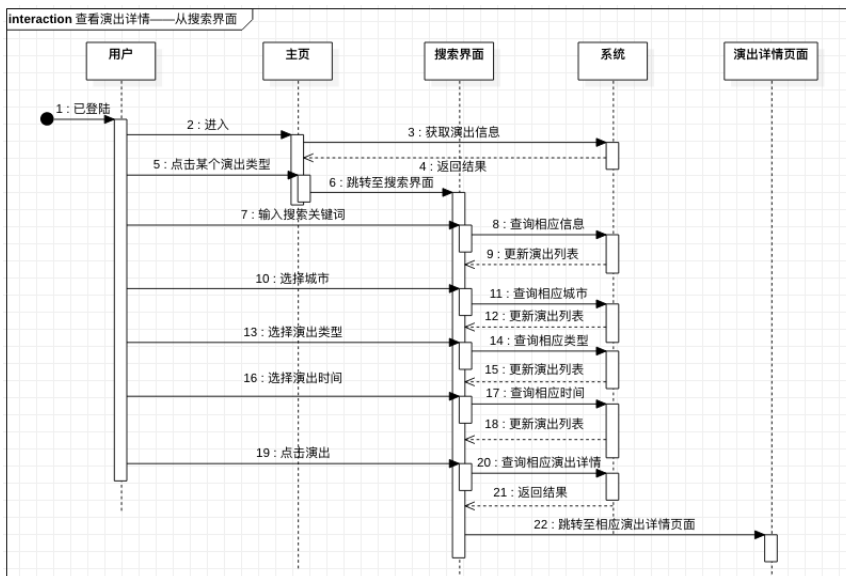
票分析类状态图：



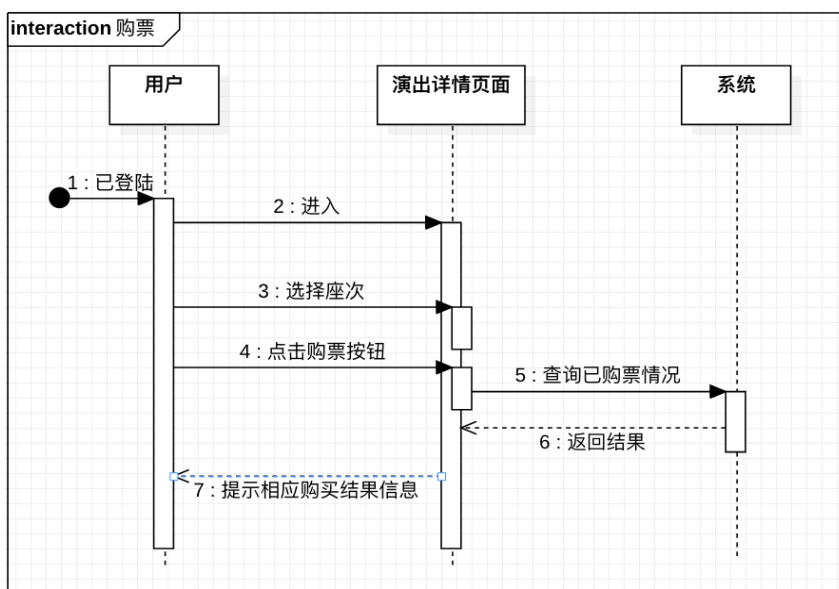
搜索演出：



查看演出详情：

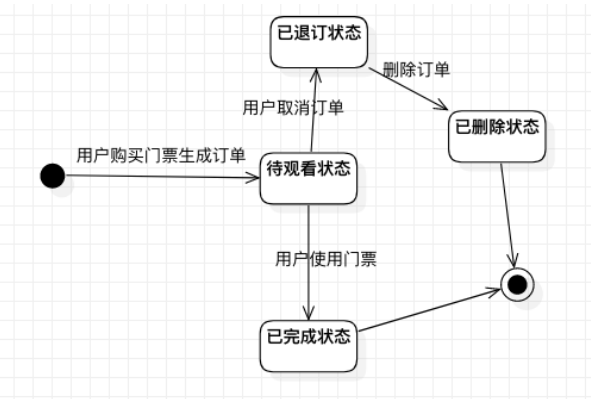


购票:

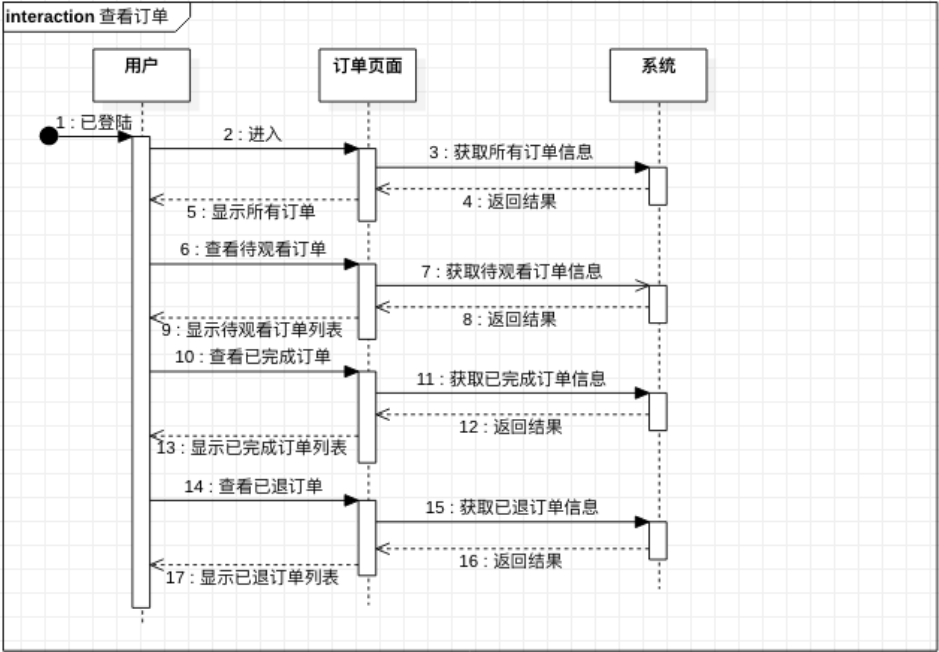


2.4.3.3 订单模块：

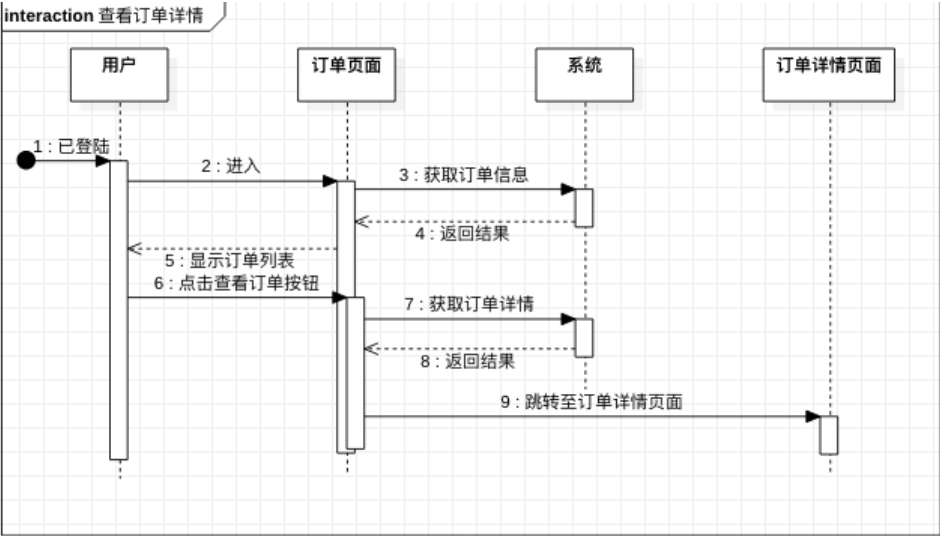
订单分析类状态图：



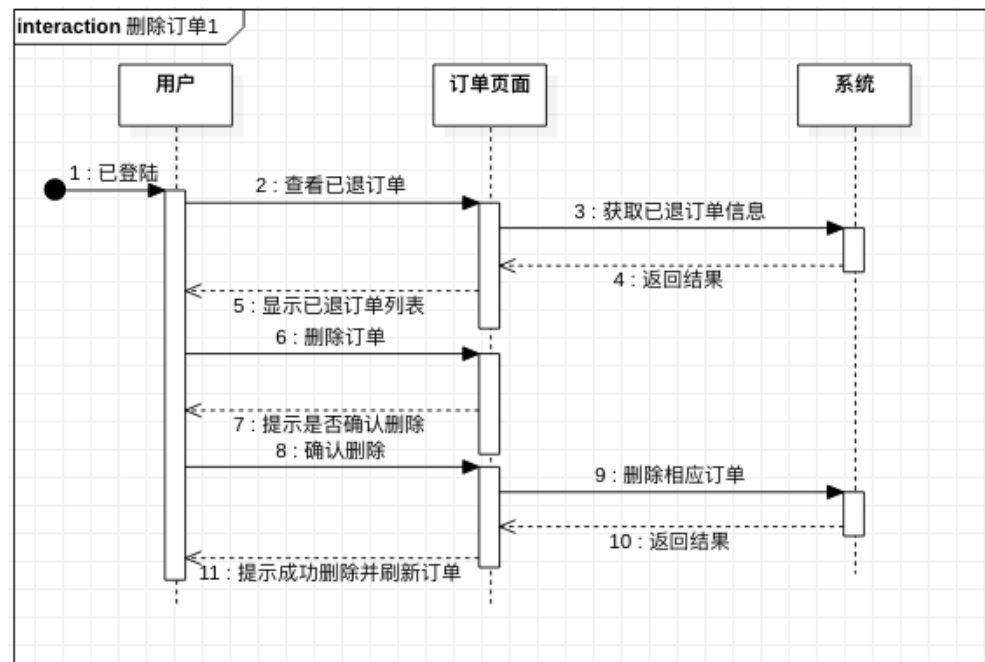
查看订单：



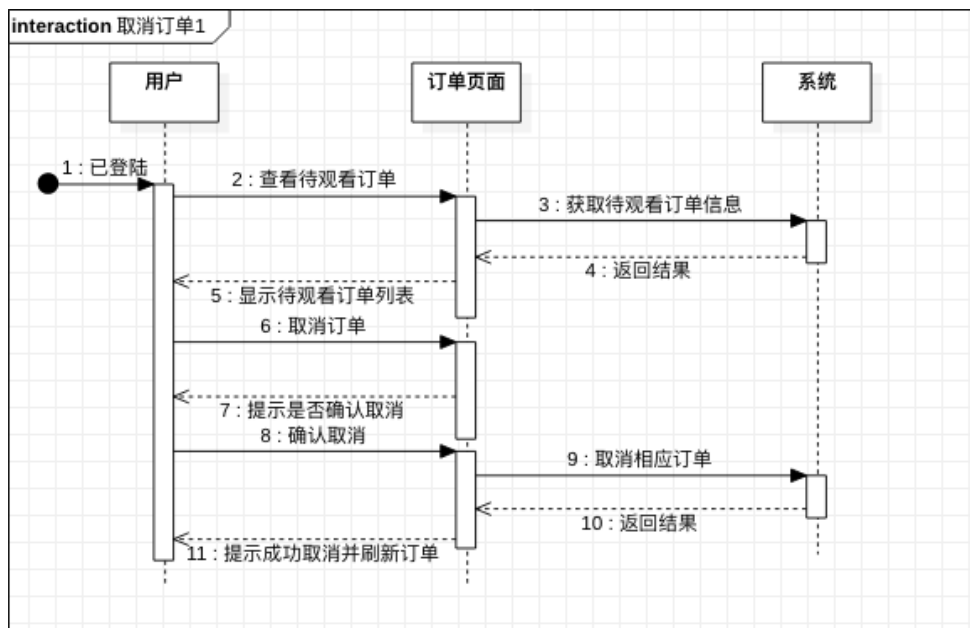
查看订单详情：



删除订单：



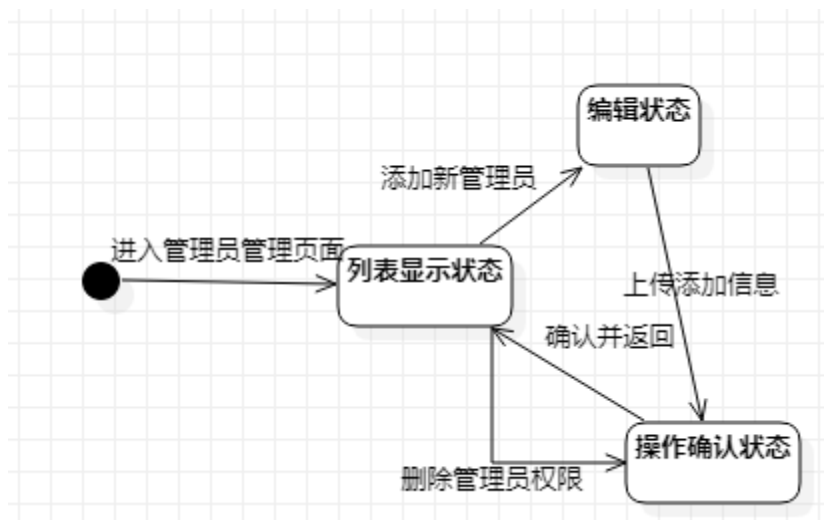
取消订单：



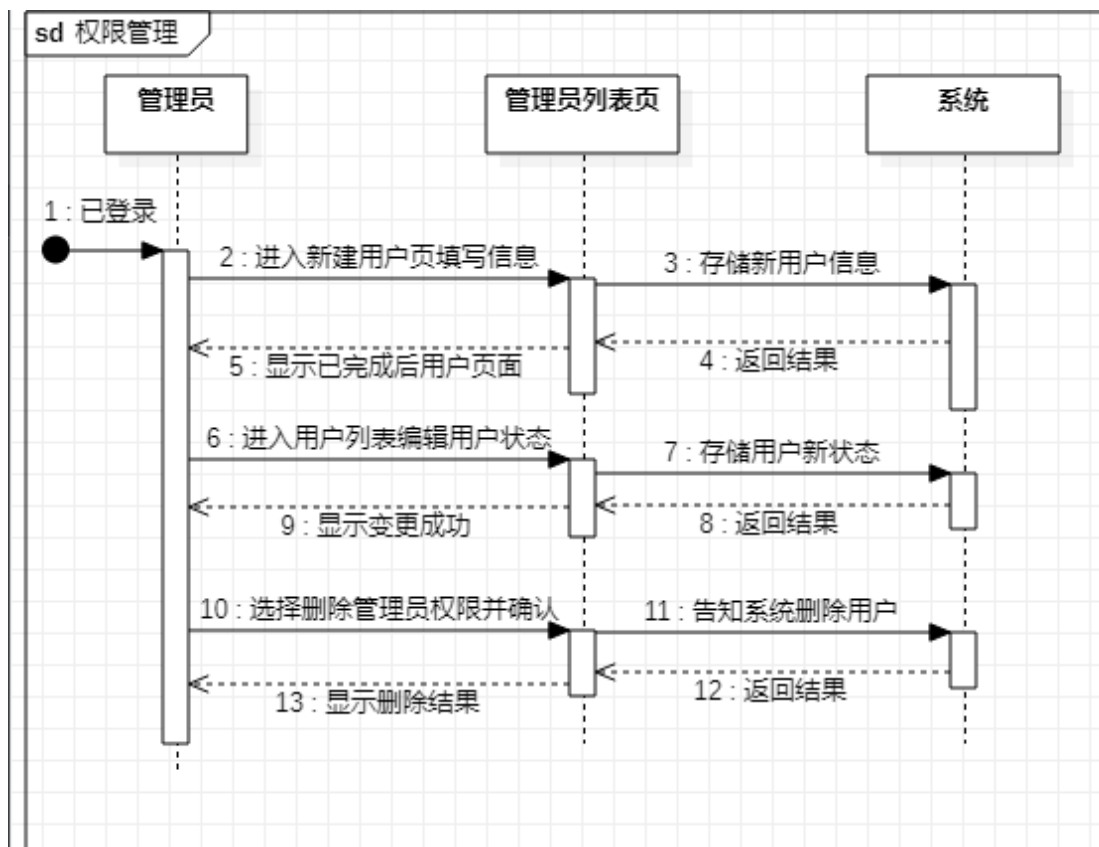
2.4.3.4 管理员部分

权限管理：

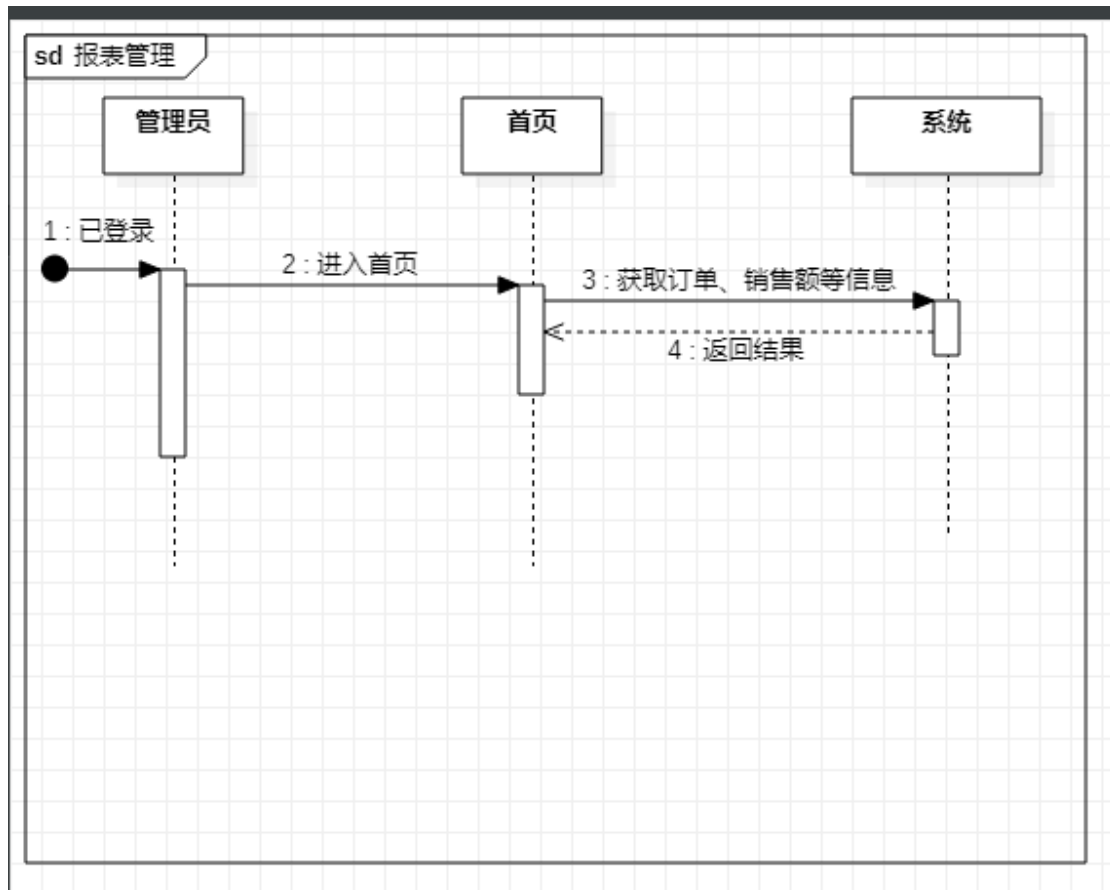
状态图：



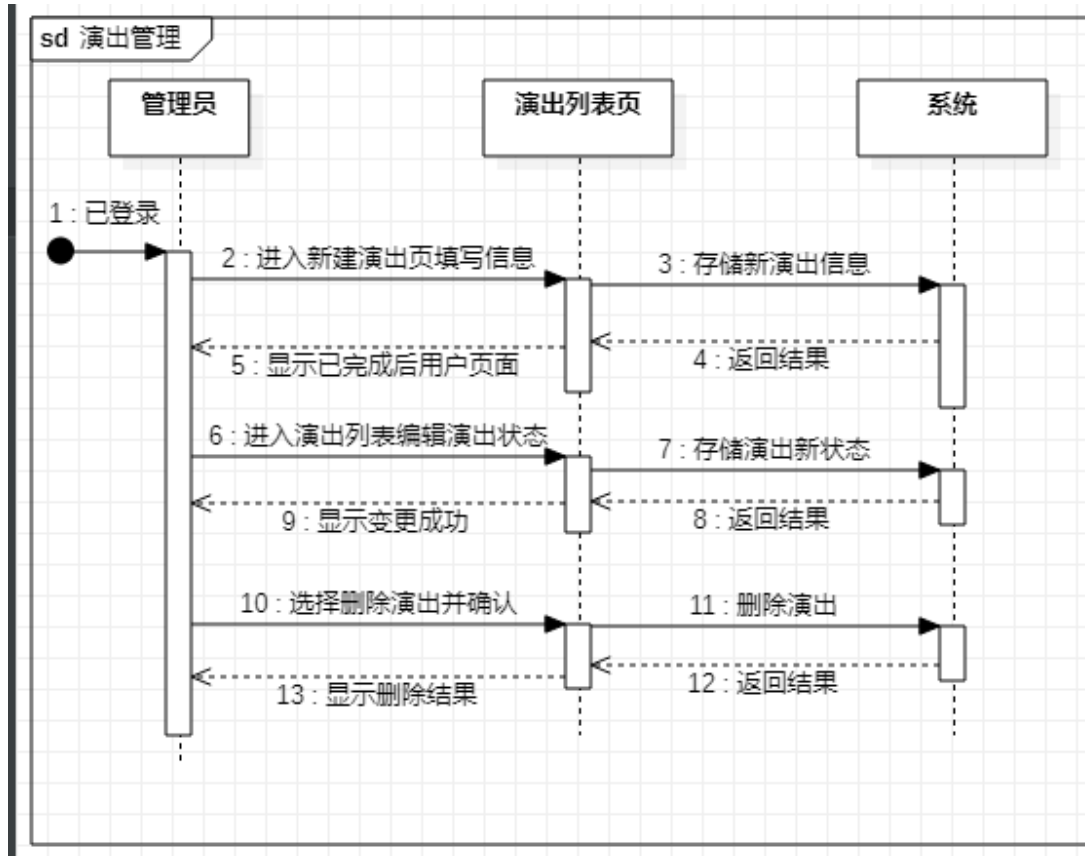
流程图：



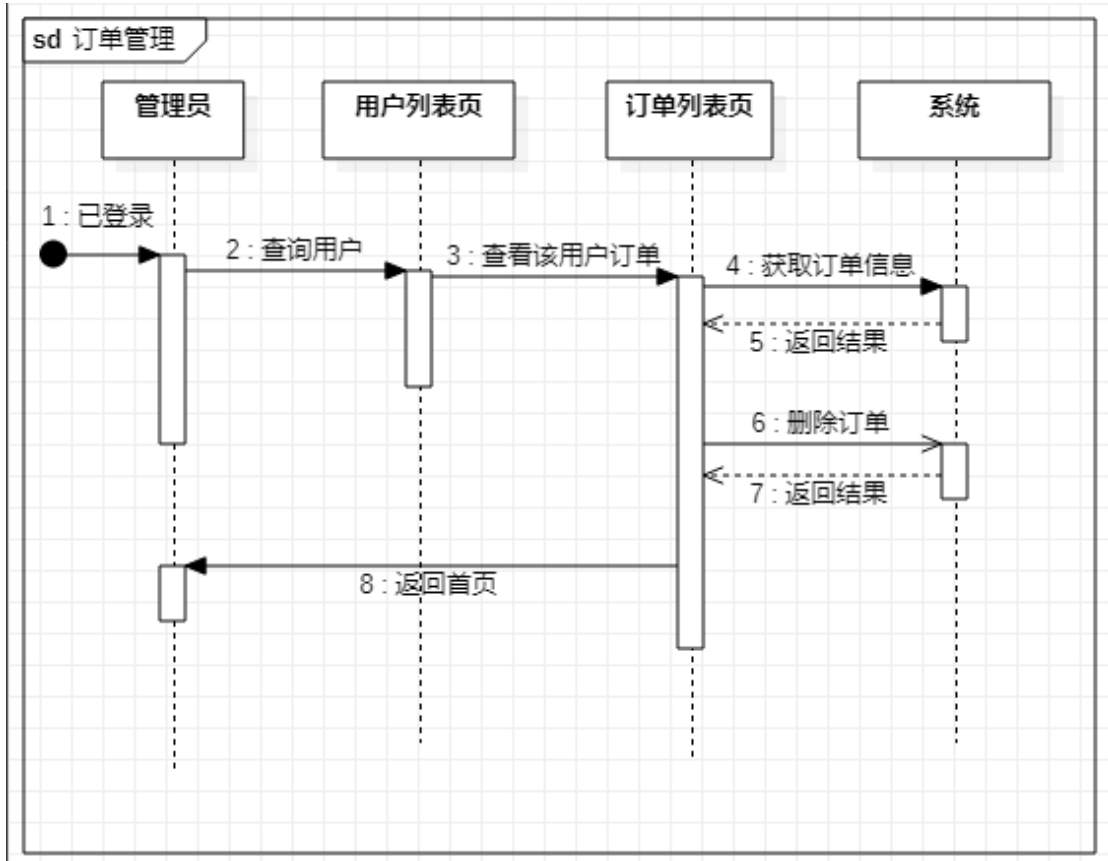
报表管理：



演出管理:



订单管理：



3. 非功能需求

3.1. 性能要求

3.1.1. 时间特性要求

1. 响应时间：

在网络状况良好的情况下，对于页面的加载，响应时间不超过 1s。在并发用户少于 100 人时，各种异步访问加载的响应时间也应小于 1s。

2. 更新处理时间

数据库的更新涉及网络数据传输，与网络传输速度有关。但正常情况下，数据库更新时间不应超过 1s，网络状况较差的情况下，不应超过 4s。数据库也是异步更新，不影响主进程。

3. 数据转换时间

对于文字的加载，数据转换较快，不应大于 1s。对于图片的加载，数据转换速度一般，在网络情况较好时，不应超过 2s。对于网页的加载就会更加耗时，在网络情况较好时，不应超过 3s。

3.2. 数据管理能力要求

3.2.1 常量约定

1. 主页界面演出 ≤ 100 条
2. 订单界面订单 ≤ 100 条
3. 演出详情、订单详情无常量约定

3.2.2 数据存储要求

本应用采用 MySQL 数据库进行数据的存储，对于应对比较复杂的查询具有优势，但是负载能力不强，为了应对高负载状态下的运行，本系统数据存储要求如下：

数据库采用主从复制，读写分离的方式应对读频率远高于写频率的数据单独存储情况；对于系统中高并发、访问频繁的数据，采用缓存数据库对 MySQL 进行隔离，防止其崩溃；数据库的备份采用单服务器集中备份的方法，保证数据的安全性和可靠性。

3.3. 安全及保密性要求

本应用存储的数据与用户的一些真实信息，如手机号、身份证号、真实姓名等有关。所以用户在使用网页时，一定要注意保护隐私，不要随意泄漏私密信息，保护自身安全。对于订单信息则存储在云端数据库。

3.4. 灵活性要求

1. 使用 Git 进行版本控制，采取前后端完全分离的政策，前端一个库，后端一个库，文档一个库，分工明确，灵活度高。
2. 软件采用 MVC 模式进行设计，界面和逻辑分离，代码维护、扩展变得容易，大大提高了灵活性。
3. 项目采用原型开发，迅速策划并迭代建模，灵活性好。

3.5. 其他专门要求

3.5.1 灾难恢复

本网站为服务器-客户端模式的网站，数据一部分保存在客户端（本地），但大多重要的信息都保存在服务端。当客户端发生异常闪退或者突然断电等事件，重新进入网页时，可以向服务端重新请求这些数据，完成灾难恢复。

3.5.2 可维护性

可维护性是指在不影响系统其他部分的情况下修改现有系统功能中问题或缺陷的能力。开发人员创建和设计系统架构时,为了提高系统的可维护性,必须考虑以下几个方面的要素:低耦合、高内聚系统文档记录。本系统将采用严格的软件工程的规范进行开发,并采用良好的设计模式保证系统各模块之间的低耦合及模块之间的高内聚。

本系统的所有代码将会被详细注释,对于系统所有代码,我们会生成详尽的技术文档。对于系统开发过程可能出现的报错,我们将以文档的方式详细罗列报错码及对应的报错信息。

4. 运行环境规定

4.1. 设备

在项目调试的过程当中,我们在各自 PC 机上正常运行项目。

5. 需求跟踪

每周至少一次 review 讨论分析项目近况,并且按照需求规约逐一核实需求并更新。