

计算机科学系实验报告

课程名称	UML 与可视化建模		班级	14 计 1	
实验名称	电影售票系统		教导教师	曾少宁	
姓名	谢丹燕	学号	1414080901139	日期	2016.3.3

一、实验目的

掌握基于 UML 2.0 的建模概念与方法，掌握各种 UML 图的概念与画法，其中包括用例图、活动图、类图、顺序图、组件图和状态图等。

二、实验设备与环境

操作系统：Windows 7；建模工具：StarUML。

四、实验要求

1. 实验及实验报告以增量方式完成，每次作业都在上一次作业的基础上完成，作业提交网站不提供报告下载，所以请同学们自行保管好自己的实验报告；

2. 请将实验报告中“占位符”信息替换为自己的实验相关信息；

3. 请认真撰写实验体会，**实验课结束时**立即上传实验报告：

<http://zeng.shaoning.net/uml/>。

五、实验内容、程序清单及运行结果

<电影售票系统>

功能：

- 订购电影票
- 退票

• 实验一：需求建模 - 用例模型

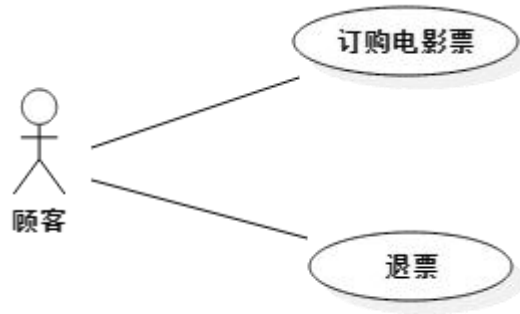


图 1：电影售票系统用例图

注：用例规约内容及项目可自行增加。

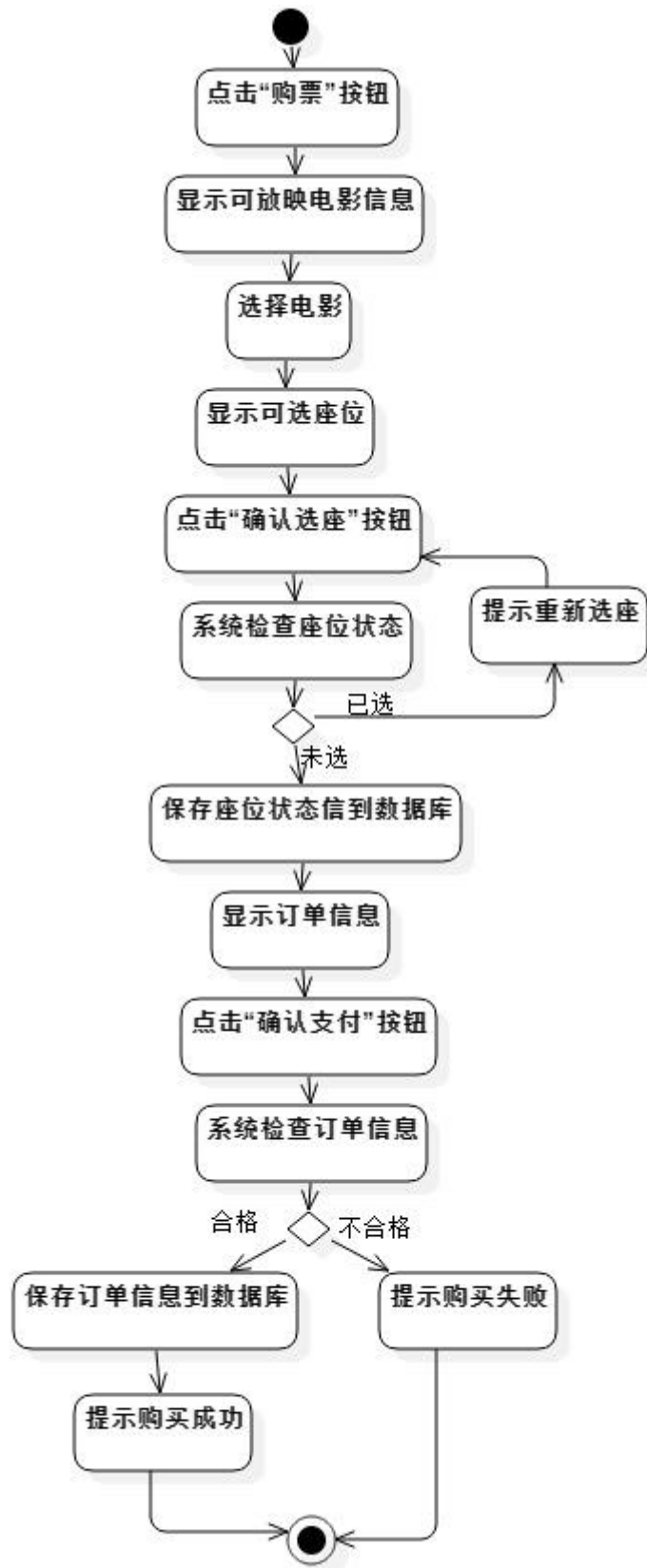
用例名称：	订购电影票
用例描述：	为顾客提供订购电影票的服务
前置条件：	顾客已登录账号
基本流程：	<ol style="list-style-type: none"> 1. 顾客打开电影售票页面，点击“购票”按钮 2. 系统显示可放映电影信息（名称，图片，简介，放映日期，价格） 3. 顾客选择电影 4. 系统显示所有可选座位 5. 顾客选择座位，点击“确认选座”按钮 6. 系统检查座位状态信息合格，更新座位状态到数据库，并显示订单信息。 7. 顾客确认好订单信息后，点击“确认支付”按钮； 8. 系统检查订单信息合格后，保存订单信息到数据库，并显示购买成功
扩展流程：	<ol style="list-style-type: none"> 5.1 系统检查座位状态信息不合格，为已选，提示重新选座 7.1 系统检查后发现该订单信息已存在，提示购买失败
后置条件：	

用例编号:	UC002
用例名称:	退票
用例描述:	为已经购买过电影票的顾客提供退票服务
前置条件:	顾客已登录账号
基本流程:	<ol style="list-style-type: none"> 1. 顾客点击“我的订单”按钮; 2. 系统转到个人中心页面, 显示所有顾客订单; 3. 顾客选择好要删除的订单, 点击“退票”按钮; 4. 系统检查该订单信息合格后, 从数据库中删除该订单, 并更新座位状态, 提示退票成功
扩展流程:	4.1. 系统检查该订单信息不合格, 提示“该订单已超期, 无法退票”的信息;
后置条件:	

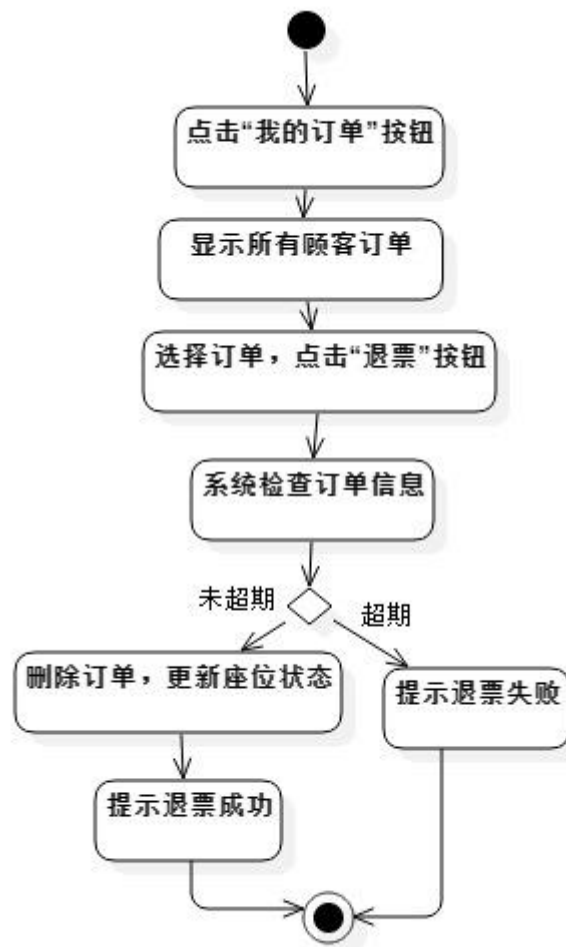
• **实验二：过程建模 - 活动模型**

使用活动图描述系统的业务过程。

方法：将用例规约中的基本流程与扩展流程抽象为过程步骤（Action），画出对应的活动图。



用例“订购电影票”活动图



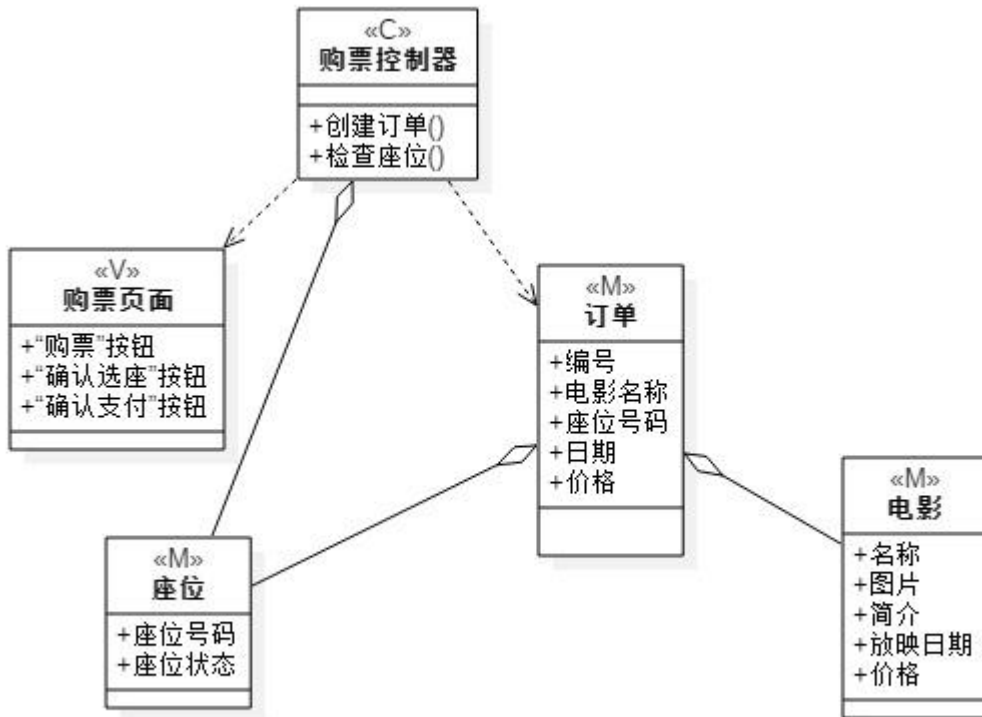
用例“退电影票”活动图

- **实验三：逻辑建模 - 类模型**

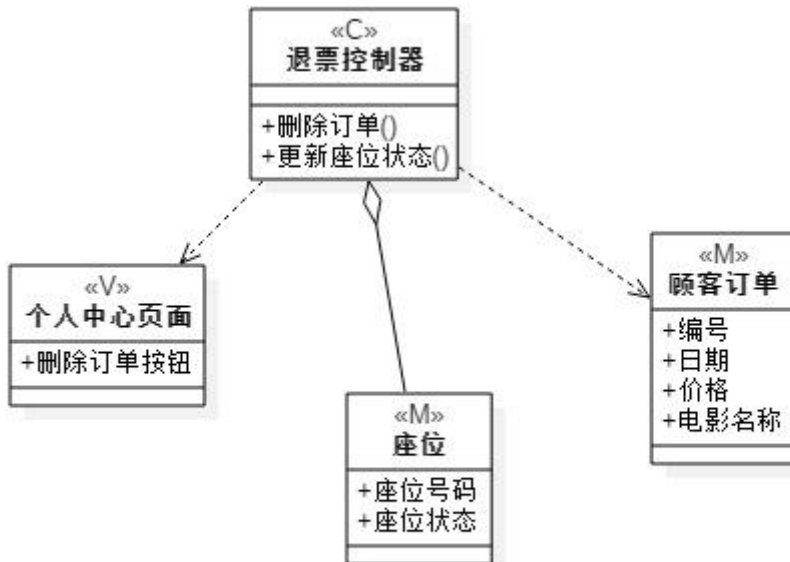
基于 MVC 设计模式找出实现用例的类。

方法：分别找出实现用例的模型（Model）、视图（View）和控制器（Controller）类，确定类之间的关系及其关键属性，画出类图。

参考：讲义 P26 页。



用例“订购电影票”类图



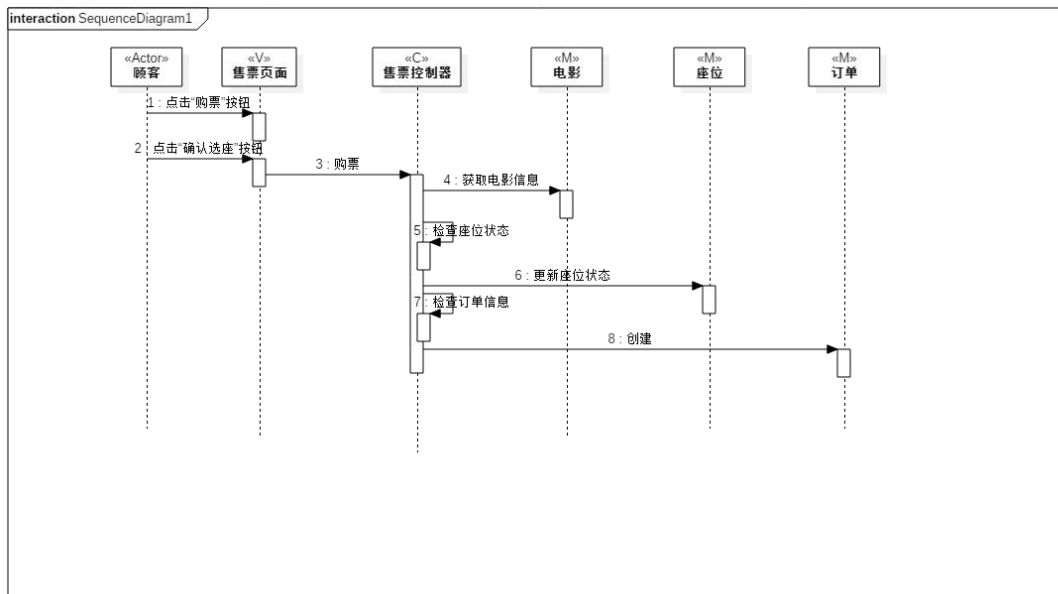
用例“退电影票”类图

• 实验四：交互建模 - 顺序模型

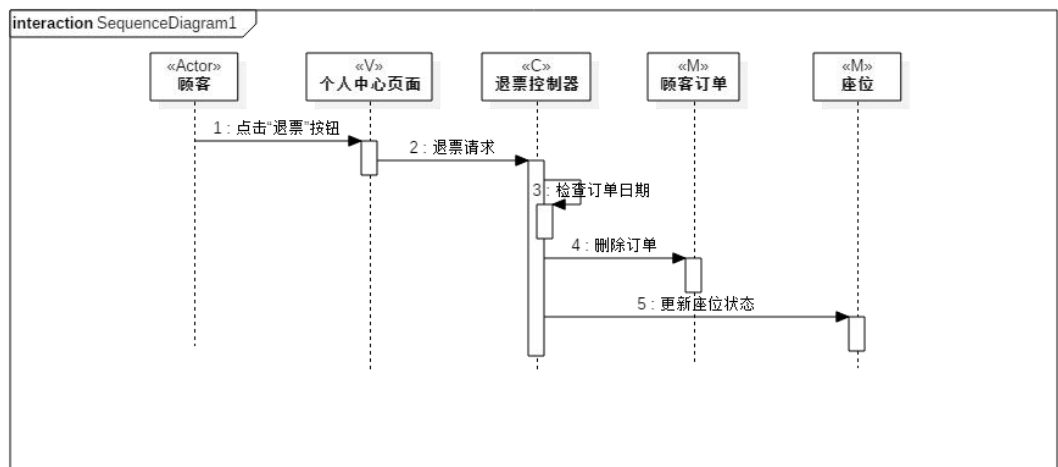
创建各个类（MVC 及 Actor）的对象，并描述对象之间的交互。

方法：分别创建参与者（Actor）、界面类（View）、控制器类（Controller）和模型类（Model）的对象，描述各个对象之间的消息及其顺序，画出顺序图。

参考：讲义 P33 页 8.7.2。



用例“订购电影票”时序图



用例“退电影票”时序图

• 实验五：状态建模 - 状态模型

对系统中最重要的对象进行状态建模。

方法：选择一种对象，定义该对象的状态，描述状态之间的切换及条件，画出状态图。

参考：讲义 P9 和 P10 页。



“订单”状态图

六、实验体会

实验一：

对于选题，我一开始思考了几个案例，然后对选哪个案例比较有把握，我纠结很久，最后经过两个的案例更改，最终选择“电影售票系统”这个案例。然后在描述该案例的用例的过程中，经历了多次更改，每一次的审视都会发现上一次的漏洞和错误。这主要归结于一开始我对 uml 的用例规约不熟悉，然后在经过每一次的更改才慢慢掌握。

实验二：

对于画用例的活动图，我本以为难度不大，只要把之前的用例规约内容一个个套进去就行了。然而真正在画活动图的过程中，我还遇到一些困难，通过活动图，我发现我之前写的用例规约还是有一些不完善的地方，于是又得重新检查用例规约并进行修改。虽然过程有些繁琐，但是通过不断的检查和纠正，让我对案例的整体结构有了更清楚的了解。

实验三：

对于类图，我一开始是很没有把握的，感觉难度很大，不知从何下手。只能先把大概的类图画出来，之后在审视大概的类图中，我又找到了进一步的扩充，产生新的类图。但是在描述几个类图的属性时，思维有些混乱，因为属性多又冗杂，所以对属性做增删改时是比较困难的一步。之后是描述几个类图之间的联系，做这一步时，结合课上的笔记，很快就完成了。对于最终画出来的类图，我还是挺满意的。

实验四：

这次实验看似简单，但是做的时候难度有点大，通过这次实验，我认识到了之前的实验结果中有很多不足，为了能够实现案例的时序图，必须不断修改前面的用例规约，随之，活动图和类图也做了很多更改，总之这次实验，做的更改幅度有些大。

实验五：

这次实验比较简单，主要根据类图创建状态图，针对之前创建的类图，我认为在我的这个电影售票系统案例中，订单是最关键的一部分，因此这次我用“订单”这个类来进行建模。一开始由于我弄不清订单的状态，画出来的状态图看起来很繁杂，为了去繁就简，于是重新修改了订单的状态，修改后的订单状态相对之前的看起来比较符合状态图的规约。