电影院售票系统

系统需求规格说明书

|  |  |
| --- | --- |
| **文件版本** | 1.2 |
| **编写日期** | 2021.6.1 |
| **发布日期** | 2021.6.5 |

**文件修改记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **修改日期** | **版本号** | **变化状态** | **修改内容** | **修改人** |
| 2021.6.1 | 1.2 | A,M | 采用用例图表示项目功能需求，模型使用规范一致的图形符号和文字描述内容。 | 常祺 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

\*变化状态: C——创建，A——增加，M——修改，D——删除

**文档审批信息**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本号** | **提交人** | **批准人** | **批准日期** | **发布日期** | **备注** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1.引言 1](#_Toc8520)

[1.1系统概述 1](#_Toc29085)

[1.2预期读者 1](#_Toc27909)

[1.3开发背景 1](#_Toc21250)

[2.系统描述 2](#_Toc25)

[2.1现状描述 2](#_Toc25757)

[2.2系统目标 2](#_Toc6377)

[2.3目标系统描述与用例图 2](#_Toc9962)

[2.4范围 4](#_Toc19179)

[2.5系统假设/约定 4](#_Toc18634)

[2.6系统架构 4](#_Toc21941)

[2.6.1 总体框架 4](#_Toc3793)

[2.6.2 子模块的设计 5](#_Toc10643)

[2.7接口与界面 6](#_Toc5624)

[2.7.1 外部接口 7](#_Toc11769)

[2.7.2 软件接口 7](#_Toc21720)

[2.7.3 通信接口 7](#_Toc16450)

[2.7.4用户界面 8](#_Toc6114)

[2.8数据库设计 9](#_Toc21911)

[2.8.1 数据库说明 9](#_Toc22573)

[2.8.2 数据库的逻辑结构 11](#_Toc16636)

[2.9软件重用方案 13](#_Toc2508)

[2.10设计关键类的服务重点 14](#_Toc26322)

[3.功能需求 15](#_Toc9681)

[3.1系统流程图 15](#_Toc12620)

[3.1.1 售票系统数据流程 15](#_Toc5478)

[3.1.2 统计系统数据流程 15](#_Toc30120)

[3.1.3查询系统流程图 16](#_Toc3743)

[3.2功能一览表与四象限分析法 16](#_Toc14389)

[3.2.1功能一览表 16](#_Toc14580)

[3.2.1 四象限分析法 16](#_Toc28002)

[3.3功能描述 18](#_Toc24202)

[3.3.1 功能描述 18](#_Toc2155)

[3.3.1 功能描述 18](#_Toc22700)

[4.验收准则和验收程序 20](#_Toc27753)

[4.1验收程序 20](#_Toc6438)

[4.2质量要求 20](#_Toc15044)

[4.3验收准则 20](#_Toc32607)

[5.环境要求 21](#_Toc4699)

[5.1.硬件要求 21](#_Toc24426)

[5.2软件要求 21](#_Toc11497)

[6.附录 22](#_Toc4700)

[6.1系统状态图 22](#_Toc29790)

# 1.引言

## 1.1系统概述

现在去电影院看电影成为了大家普遍选择的一种休息方式，但是以我们调研观察发现基本上所有的影院没有自己的售票系统，他们依靠于美团、猫眼、淘票票这些APP来进行电影票的网上售卖。据了解，甚至于有些小县城的电影院，只是有微信公众号，在一些APP上根本搜不到此影院，使得大家需要去到地方，才能购票选座。  
 基于此方面，我们的电影售票系统是以电影院为主要客户，使得电影院有独属于自己的售票系统，他们可以在自己的售票系统上进行电影信息的更新以及电影票的售卖，不必再依附于一些APP，这样就会使得他们减少一部分的成本

## 1.2预期读者

开发人员，测试人员，电影院方面的管理者，第三方测试人员。

## 1.3开发背景

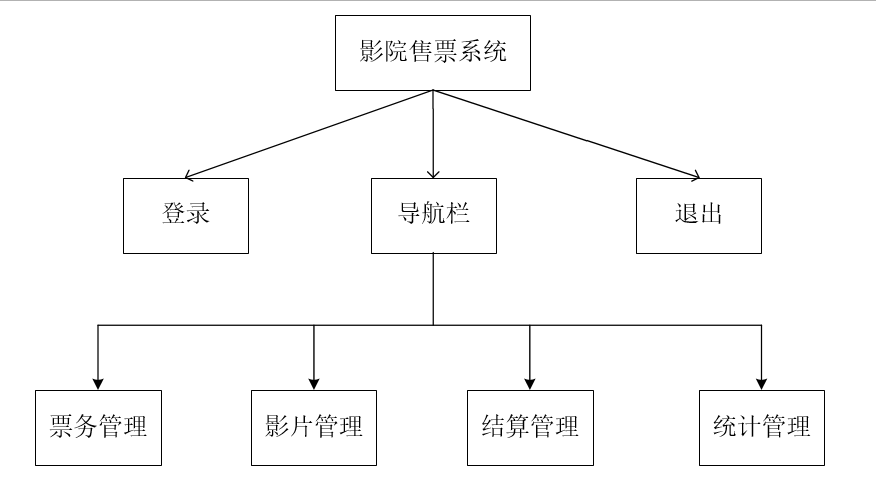
作为整个电影产业利益链和终端，影院售票系统具备影片排期、电影售票、卖品销售、报表统计、电话语音及互联网售票等功能，几乎掌握影院的全部商业秘密。一款良好的影院售票管理系统，可以满足影院集团化管控需求，提高工作效率，使复杂多样的日常经营数据流直观化、合理化地呈现出来。  
 目前我国市场上共有7家电影院售票软件商，分别是鼎新、满天星、火凤凰(即广东粤科)、中鑫汇科、火烈鸟、沃思达、以及刚刚进入市场的M1905售票系统。但是综合来看，目前的票务系统产品功能上同质化比较严重。随着互联网在电影行业的持续发力，猫眼、微票儿、格瓦拉等在线售票平台也将逐渐成为电影票销售的主要渠道。 在未来，尤其是面对影院集团的连锁化管理、市场精细化运营的强烈需求下，影片传输与影院管理也逐渐开始了虚拟化、数字化的变革，中国电影市场的票务信息化只有选择适用于新时代需求的售票系统，影院才能在电影行业的井喷发展时期保持优势，持续发展。所以针对我们的电影售票系统，由于是主要面向于电影院本身来设计的，所以我觉得在与其他行业竞争以及同行业竞争的话，竞争力还是相当可观的。

# 2.系统描述

## 2.1现状描述

现在系统处于需求分析与功能设计阶段。

## 2.2系统目标

****

**图2.2 系统目标描述图**

如图2.1是系统目标的流程图。现有系统的工作主要有以下四个方面的  
内容:

（1）票务管理:对电影票的出售、退票、预订进行管理；

（2）信息管理:对影院的影片上档、下档进行管理；

（3）结算管理:对影院的日、月销售额进行统计与查询，

（4）统计管理:对影院的影片上座率、售票情况进行统计。

## 2.3目标系统描述与用例图

本系统主要用于电影票销售，所以提供了以下几个子功能：用户注册，用户登陆，电影详细内容，生成订单，退订订单，付款，以及后台方面的电影的添加，以及电影的查询等后台功能。

（1）用户注册：新用户可以通过注册账号来登陆。

（2）用户登陆：登陆后才能进行购买，查询订单。

（3）电影详细内容：点击进入后可以查看电影的详细信息。包括：时长，价格，版权，上映的时间等。

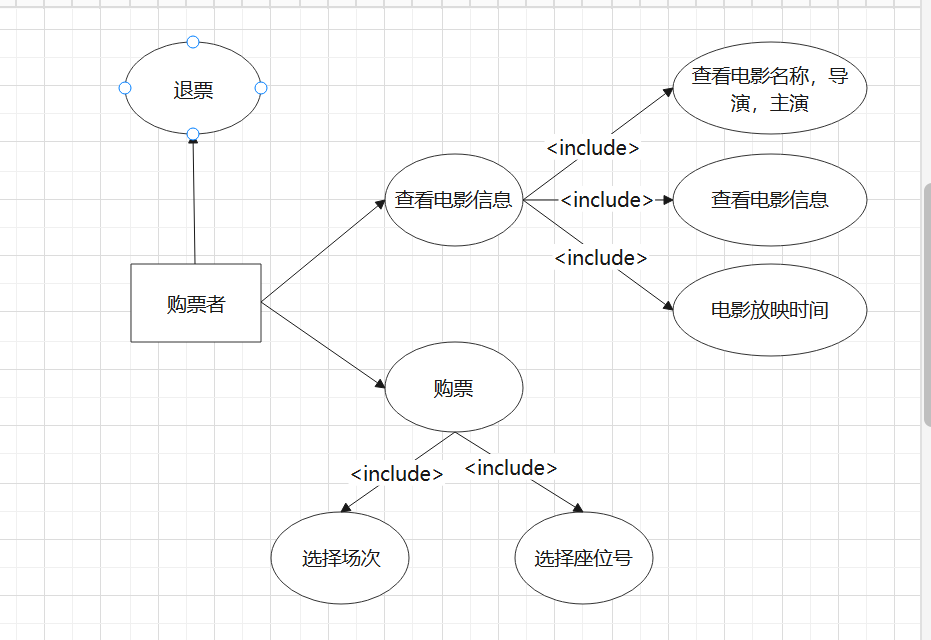
（4）生成订单：该项功能提供用户选择需要购买的影片生成订单，及取消订单等操作。

（5）退订功能：可以选择是否取消已经生成的电影订单。

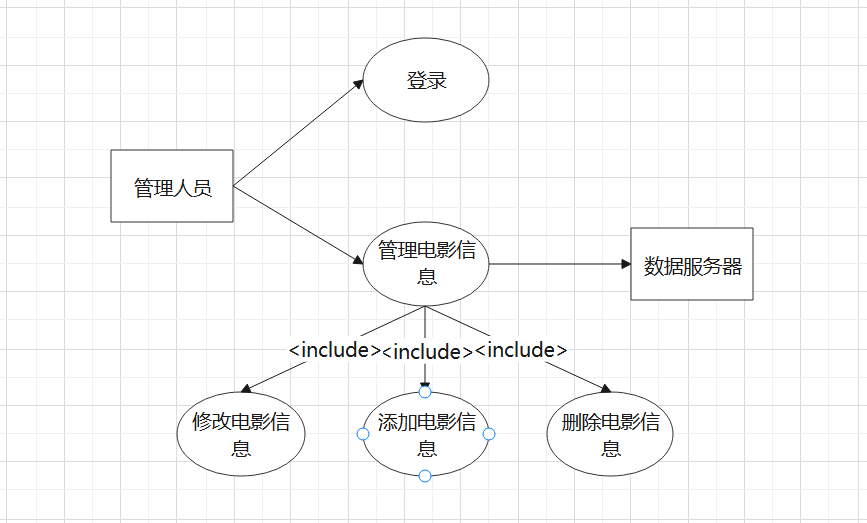
（6）付款：此页面用户可以选择付款方式，并最终确定订单的生成。

（7）后台电影的添加：该功能只有管理员有权操作，管理员可以通过影片销售排行来增加电影的内容，信息等操作。

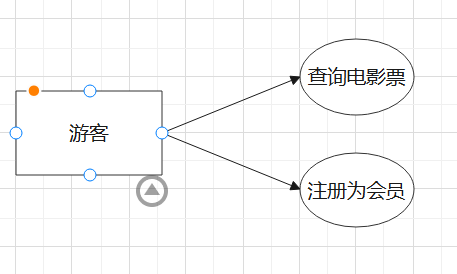
具体的用例图如下：



**图2.3.1 购票者的用例图**



**图2.3.2 影院管理者的用例图**



**图2.3.3 游客的用例图**

## 2.4范围

我们项目的范围是进行软件的设计和编码，通过前期设计的数据流图经过用户用户同意后进行编码制作这个项目。

## 2.5系统假设/约定

（1）界面美观，用户的购票流程容易。

（2）管理电影起来操作要比较简单，不要太过于复杂。

## 2.6系统架构

### 2.6.1 总体框架

新视界电影院售票管理系统采用B/S结构的软件设计模式，运用Tomcat服务器，基于JSP技术，Servlet技术，MYSQL数据库来实现高校科研管理系统的科研项目管理的主要功能。

B/S整体的结构如下:

第一层是浏览器，即客户端，只有简单的输入输出功能，处理极少部分的事务逻辑。由于客户不需要安装客户端，只要有浏览器就能上网浏览，所以它面向的是大范围的用户，所以界面设计得比较简单通用。

第二层是WEB服务器，扮演着信息传送的角色。当用户想要访问数据库时，就会首先向WEB服务器发送请求，WEB服务器统一请求后会向数据库服务器发送访问数据库的请求，这个请求是以SQL语句实现的。

第三层是数据库服务器，他扮演着重要的角色，因为它存放着大量的数据。当数据库服务器收到了WEB服务器的请求后，会对SQL语句进行处理，并将返回的结果发送给WEB服务器，接下来，WEB服务器将收到的数据结果转换为HTML文本形式发送给浏览器，也就是我们打开浏览器看到的界面**。**

### 2.6.2 子模块的设计

实质上，电影票销售系统系统的综合性相对较强，复杂程度相对较高，可对现有软件进行充分利用，进行系统设计与规划。编写一个完整的电影销售系统，需要涉及网页界面、处理程序、MySQL 后台数据库系统等，在网站页面上显示电影简介、电影海报、电影评分等诸多内容。数据库方面需要存储用户信息，电影信息以及排片信息等。

系统按照功能主要分为注册用户、管理员两个模块，用户具有用户注册、用户登录、查看电影、付款、退订的功能，管理员具有退订、编辑电影简介内容、后台电影添加删除、查看用户信息的功能。如图2.6.1所示



**图2.6.1系统模块功能图**

系统总体结构可分为系统前台和系统后台两个功能模块。前台功能实现以下功能，用户注册、用户登录、电影信息、电影搜索、付款、个人设置。系统后台功能实现以下功能，用户管理、排片管理和系统设置。系统总体功能结构如图1-3所示：

****

**图1-3系统总体功能结构图**

## 2.7接口与界面

本系统软件的接口设计遵循六大基本原则：分别是：单一职责原则、依赖倒置原则、接口隔离原则、迪米特法则、里氏替换原则、开闭原则。这6个原则是建立稳定的，灵活和健壮设计的基础，但设计的时候还要根据实际情况考虑。

（1）单一职责原则(SRP)：该原则要求我们在设计类或者接口的时候。尤其在设计接口的时候把职责分清楚，通常一个职责不是单一的方法，是一类方法的组合。

（2）依赖倒置原则：为高层模块不应该依赖底层模块，他们都应该依赖抽象。抽象不能依赖细节。细节应该依赖抽象，这个原则要求设计的时候尽量用抽象（抽象类或者接口）把各个模块独立开来，实现解耦，使各模块相对独立。简单来说就是要用面向接口设计。

（3）接口隔离原则(ISP)：接口分两种，分别为类实例接口和类接口。这个原则依赖建立在最小的接口之上，依赖自己需要的接口。

（4）迪米特法则(LoD)：也称为最少知识原则。就是一个对象应该对其他对象有最少的了解。这个法则的本质就是解耦，解耦是有限度的不能为了解耦而解耦。

（5）里氏替换原则：（所有引用基类的地方，都能透明地使用其子类的对象。）简单来说就是父类出现的地方，替换为子类不会产出异常。里氏替换原则的好处就是增强程序的健壮性，保持程序的兼容性。

（6）开闭原则：软件的实体如类，模块和函数应该对扩展开发，对修改关闭。这个原则要求我们设计的系统扩展性好，因为需要不会一直不变的，我们需要应对的永远是变化。

### 2.7.1 外部接口

（1）用户界面：在界面设计上，应做到简单明了，易于操作，并且要注意到界面的布局，应突出的显示重要以及出错信息。外观上也要做到合理化，考虑到用户多对Windows风格较熟悉，所以该系统尽量向这一方向靠拢。

（2）软件与硬件接口：本系统设有人机操作界面，考虑到操作简单，易于管理方面，主要硬件与接口设备为pc、鼠标、键盘。而软件接口主要以Windows平台为基本平台。

### 2.7.2 软件接口

我们用到了数据库这种的软件接口。从计算机系统来讲，计算机，数据库资源、第三方主体客户、开发商共用一套数据模式，开发商利用系统完成授权，借助规范化的计算机运行原理，进一步完成数据的交换工作，数据库模式属于软件开发商定制的数据模式，包括Oracle以及sqslrver，在文件交换、程序函数数据库开发的难度较低，灵活性较高，但是在特定应用的过程中，数据库应用的空间较为狭窄，开发商对其进行个性化的定制使用。

数据库应用的流程为系统一、数据库、系统二、数据库、系统一。

### 2.7.3 通信接口

B/S模式的中文意思为“浏览器/服务器”模式，又称B/S结构。它是对C/S（Client/Server客户端/服务器）模式应用的扩展。其特点就是使用浏览器取代了原来的客户端程序，用户的工作界面都是通过浏览器来实现的。

在B/S体系结构系统中，用户通过浏览器向网络的服务器发出请求，服务器对请求进行处理，将结果返回到用户浏览器上。B/S结构简化了客户机的工作，服务器将担负更多的工作，数据库的访问和应用程序的执行都是在服务器上完成，如数据请求、加工、结果返回以及动态网页生成等工作全部由WEB服务器完成，相对于C/S结构，B/S在应用程序的部署、升级、维护时，只需要在服务器上进行配置就可以了，这种方式提高了系统的安全性，同时也降低了维护成本。

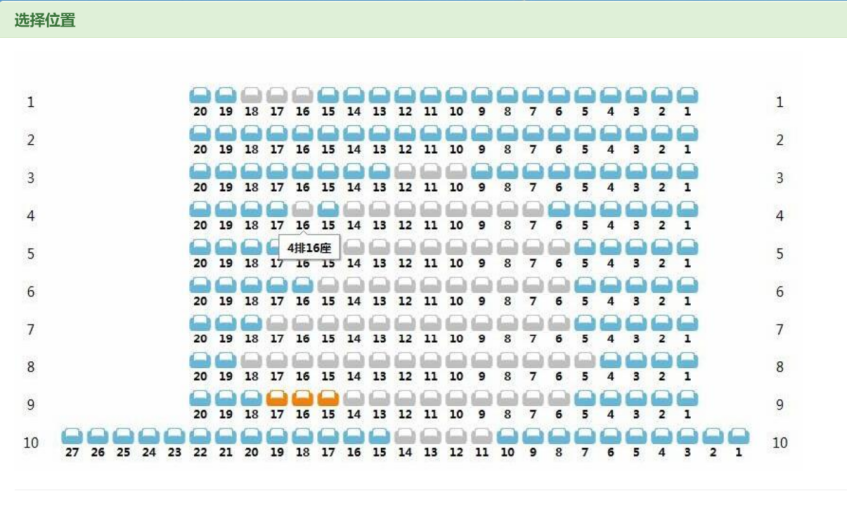
### 2.7.4用户界面



**图2.7.1 管理界面**

****

**图2.7.2主页面**



**图2.7.3选座页面**



**图2.7.4网页支付界面**

## 2.8数据库设计

数据库的设计关系到整个应用系统的运行效率，数据库设计得好，不仅有利于日常数据的维护更新，而且可以提高系统的运行效率，缩短数据查询响应周期，增加网站的流量。合理的数据库设计可以使围绕它支持的Web页面的Java代码简单化，易于实现，并且可以提高数据存储的效率，保证数据的完整一致。电影销售系统采用MySQL作为后台数据库开发工具。

### 2.8.1 数据库说明

概念模型用于信息世界的建模，与具体的DBMS无关。为了把现实世界中的具体事物抽象、组织为某一DBMS支持的数据模型。人们常常首先将现实世界抽象为信息世界，然后再将信息世界转换为机器世界。也就是说，首先把现实世界中的客观对象抽象为某一种信息结构，这种信息结构并不依赖于具体的计算机系统和具体的DBMS，而是概念级的模型，然后再把模型转换为计算机上某一个DBMS支持的数据模型。实际上，概念模型是现实世界到机器世界的一个中间层次。

信息世界中包含的基本概念有实体和联系。

（1）实体

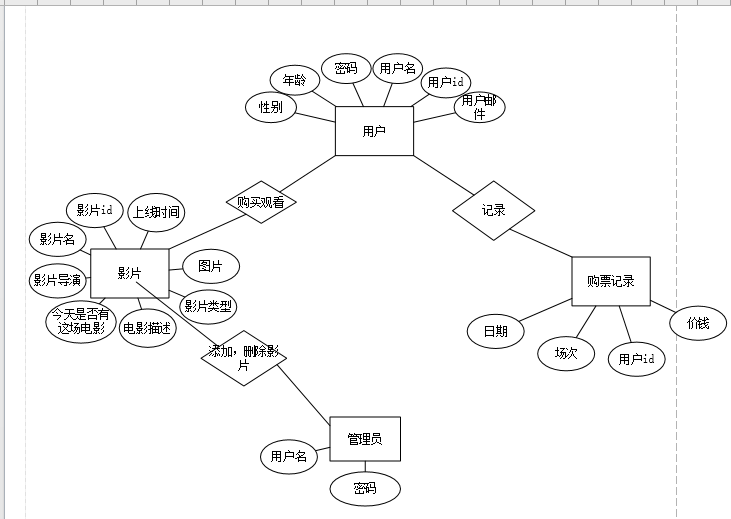
客观存在并可相互区别的事物称为实体。实体可以是具体的人、事、物，也可以是抽象的概念或联系。例如，一个学生、一门课、一个供应商、一个部门、一本书、一位读者等都是实体。

（2）联系

在现实世界中，事物内部以及事物之间是有联系的，这些联系在信息世界中反映为实体内部的联系和实体之间的联系。实体内部的联系通常是组成实体的各属性之间的联系。两个实体型之间的联系可分为3类，一对一联系，(1:1)；一对多联系(1:n)；多对多联系(m:n)。

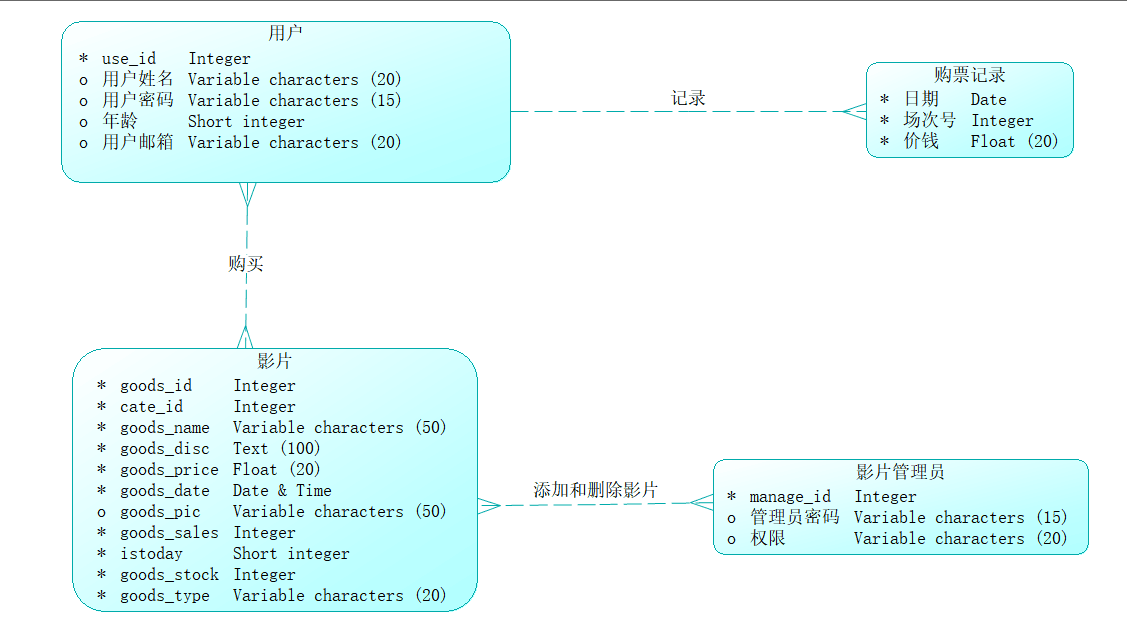
概念模型是对信息世界建模，所以概念模型应该能够方便、准确地表示信息世界中的常用概念。概念模型的表示方法很多，其中最为常用的是P.P.S.Chen于1976年提出的实体，联系方法(Entity-Relationship Approach)简记为E-R表示法)。该方法用E-R图来描述现实世界的概念模型，称为实体-联系模型，简称E-R模型。

根据数据流程分析，绘制校园二手商品交易系统的全局E-R模型如图2-1所示。



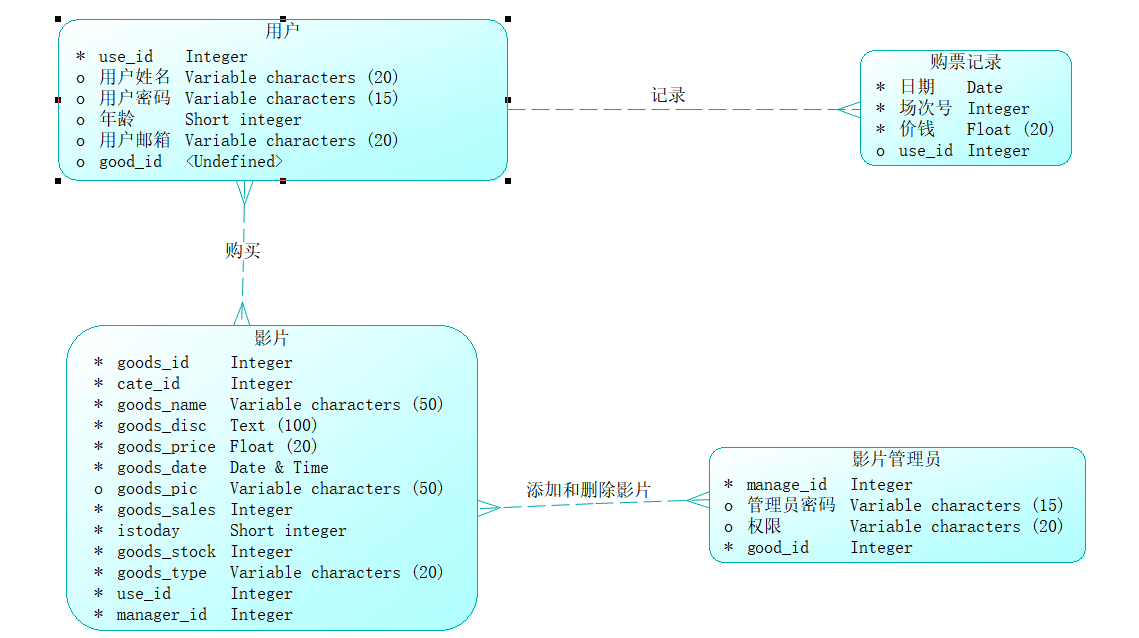
**图2.8.1系统全局E-R图**

1.概念模型



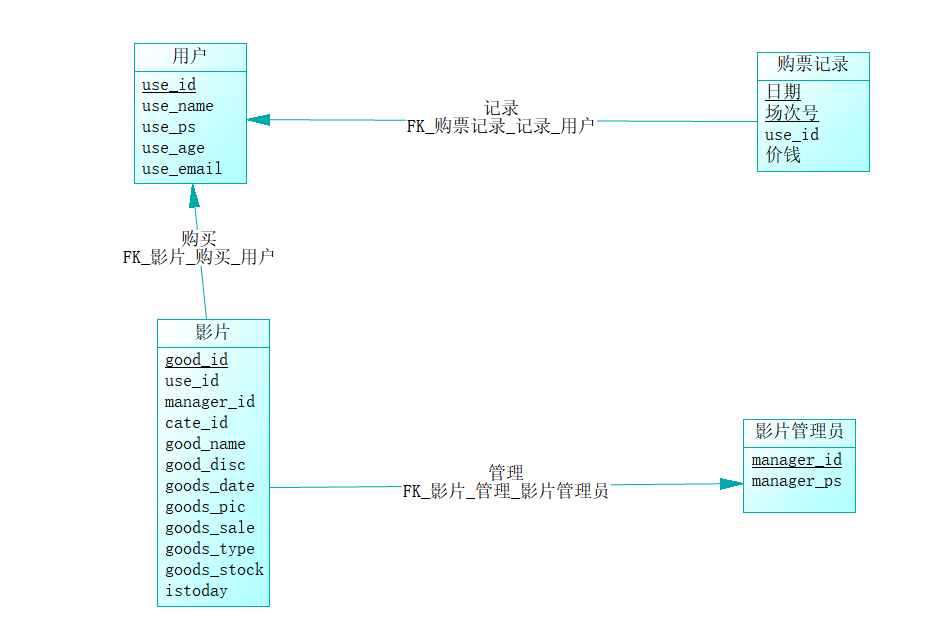
**图 2.8.2 概念模型图**

2.逻辑模型



**图 2.8.3 逻辑模型图**

3.物理模型



**图 2.8.4 物理模型图**

### 2.8.2 数据库的逻辑结构

数据库逻辑设计主要是把数据库概念设计时设计好的基本E-R图转换为与选用DBMS产品所支持的数据模型相符合的逻辑结构。它包括数据项、记录及记录间的联系、安全性和一致性约束等等。导出的逻辑结构是否与概念模式一致，从功能和性能上是否满足用户的要求，要进行模式评价。

本系统数据库名称为db\_secondhandmarket，数据库中包括：

1）种类表（category）

2）影片表（goods）

3）影片内容表（goodsdetail）

4）影片种类表（goodsdetailtype）

5）图片表（pic）

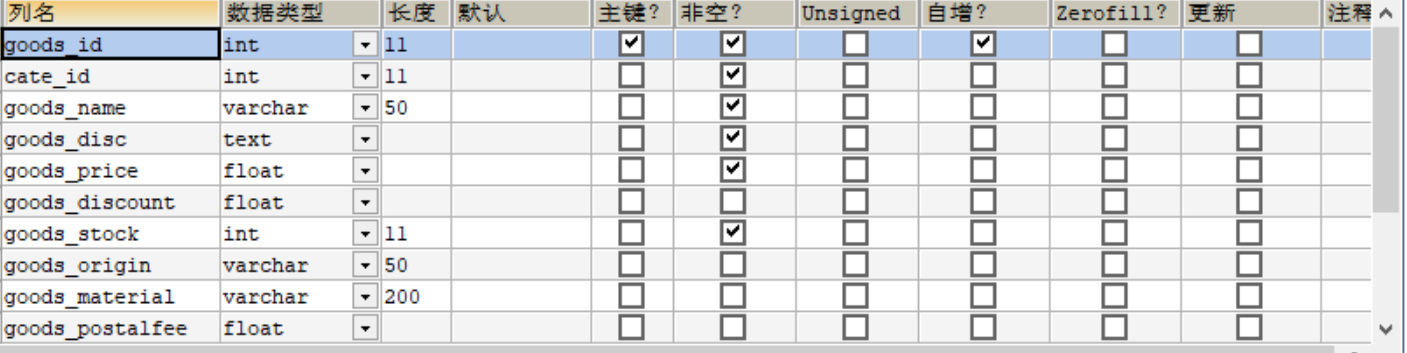
6）用户表（user）

各表数据结构如下：

（1）种类表（category）用来记录是管理员还是用户



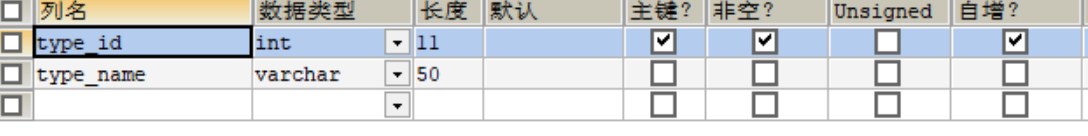
（2）影片表（goods）用于记录影片



（3）影片内容表（goodsdetail）用于记录影片细节



（4）影片种类表（goodsdetailtype）用于记录影片种类



（5）图片表（pic）用于记录图片



（6）用户表（user）



## 2.9软件重用方案

从广义的角度来说，本系统在以下三个层次进行了重用，知识的重用，方法和标准的重用以及软件成分的重用。

知识的重用是如软件工程等相关知识的重用，应用相同的知识设计规划、建设不同的软件产品。方法和标准的重用是比如本产品是利用面向对象方法设计又或者是采用国家标准局制定的软件开发文档规范。

软件成分的重用进一步我们划分为三个级别，即代码重用，设计结果重用和分析结果重用。而本软件产品的重用主要在源代码级，这与我们的开发工具有很大的关系，我们选用的开发工具是eclipse和IDE，采用spring MVC框架。利用该框架可以快速开发出一个程序的框架，是一个灵活性较强的框架。

我们的重用设计是建立在MVC框架之上。我们分析可能重用部分的实例有两种方式，一种是继承类库中的构件要用到的基本功能的类。主要是一些界面元素，如菜单活框、列表框，这些构件在很多模块中都要用到，且处理逻辑大致相同，并进行扩充其处理逻辑，如增加输入合法性检测等，这样我们在使用这些经过扩充的构件时不必每个都去重复那些处理逻辑，大大提高了效率，而且，这些构件都是经过测试或其他人使用过的，质量也有保证。

另一种方式是我们的构件在类库找不到相似的类，我们将从头创建自己的类，但为了将自己创建的类，如用户权限处理排序打印预览等，也纳入MVC的框架，我们自己创建的类都继承框架的一个抽象类，这样做的好处是把自己创建的类纳入Controller类的层次化管理，且这也是为实现虚拟函数和动态联编方便。我们也为每个件的功能和接口建立了文档，供应用开发中使用。这样我们就在MVC的框架下，建立了我们自己的一个重用构件库。最后是在应用中重用构件库中的构件。当用户需求变化时，我们能用构件库快速重构我们的应用。

使用重用明显会将开发效率提高一个层次，它不仅提高了开发效率，还保证了应用的风格和质量。特别是对我们这种新手较多的团队开发，采用重用的方法，让有经验的成员负责整个应用的框架，让新手使用重用来创建应用，这非常有利于提高效率，在保证质量的同时也为新手提供一个循序渐进的学习机会，有利于新手的成长。当然使用重用容易，但自己建立重用构件库也是需要代价的，是一个需时间积累的过程，也有一定风险。我们在重用中也遇到了一些问题，一个比较突出的问题是我们的重用主要是在源代码级，因此我们的开发必须在MVC的框架下。

我们的重用主要是在源代码级，通过类的继承来实现。其实可重用的范围是很大的，如设计的重用，测试用列的重用，可运行的代码的重用等。我们想将来扩大重用的粒度，在框架基础上，进一步根据我们单位的软硬件环境定制出一个适用于我们系统的框架，如加进注册功能，数据库连接的功能等。这样，我们可直接重用这个框架，这可以极大地提高软件的开发和维护的效率。最后，我们还设想，将来应有专门的人员管理重用，把重用的维护和应用开发分开，责任明确这样可以更进一步地做好重用工作。

## 2.10设计关键类的服务重点

本系统主要设计了三个大的关键类:servers , user, goods,即管理员类，用户类，商品类。基于这三个父类的基础上，继承更行，查询，修改等子类。

GoodsServers类主要负责对电影的上架，下架以及排片的工作；

HomeServers类只要负责用户的注册信息入库以及增删改查用户的个人信息。

Category类负责定义电影的分类；

Goods类负责定义电影的分类、ID、名字、图片、评分等信息；

Goodsdetail类时Goods的子类，负责记录电影的内容和名字；

Goodsdetailtype是Goodsdetail的子类，负责电影具体分类的工作；

Pic类用于记录图片信息；

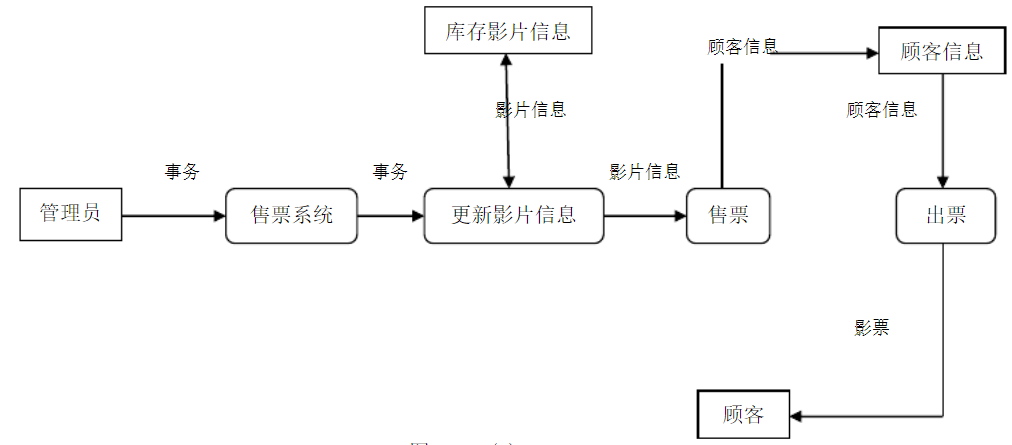
User类负责记录一些用户的个人信息，对用户的属性信息操作服务。

以上为本系统设计关键类的对应重点服务，遵循软件工程中软件设计的模块化原则，力求尽量达到“高内聚，低耦合”的程度，降低类的复杂度，且使方法属性单一，逻辑关系简单。这样可以提高类的可读性，降低功能模块之间的关联性。

# 3.功能需求

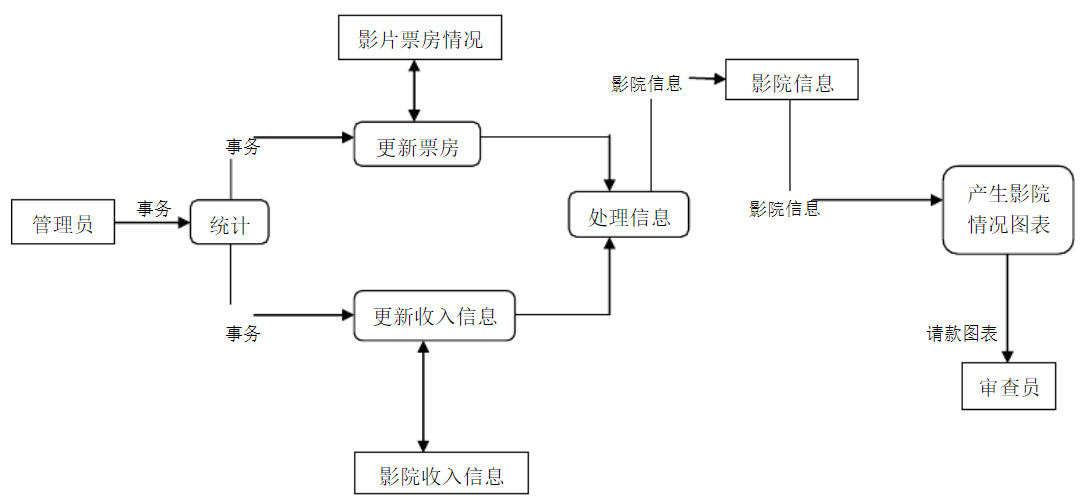
## 3.1系统流程图

### 3.1.1 售票系统数据流程



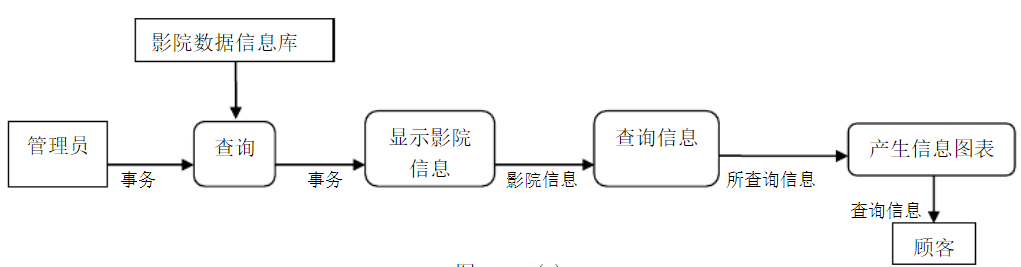
**图3.1.1 售票系统数据流程图**

### 3.1.2 统计系统数据流程



**图3.1.2 统计系统数据流程图**

### 3.1.3查询系统流程图



**图3.1.3 查询系统数据流程图**

## 3.2功能一览表与四象限分析法

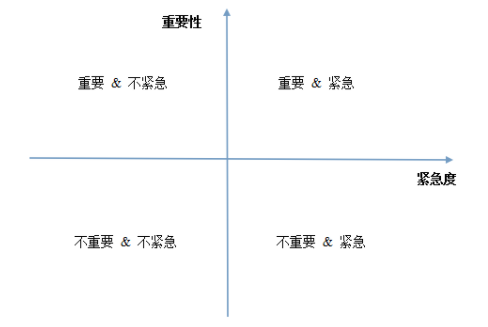
### 3.2.1功能一览表

**表3.1 功能一览表**

|  |  |  |  |
| --- | --- | --- | --- |
| 功能编号 | 功能 | 对应业务 | 优先级 |
| 01 | 购票 | 购票 | 1 |
| 02 | 退票 | 退票 | 1 |
| 03 | 影片的管理 | 影片的管理 | 1 |
| 04 | 用户注册与登录 | 注册用户 | 2 |
| 05 | 票房的统计 | 统计 | 2 |

### 3.2.1 四象限分析法

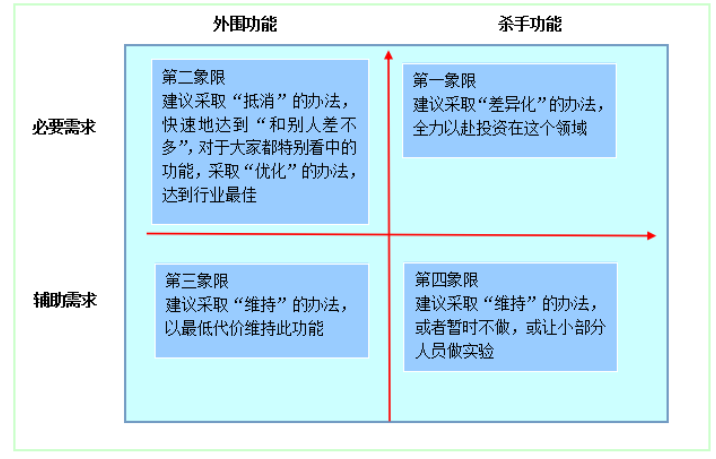
很多事情的处理可以按照不同性质进行多个维度的划分，这样复杂的事情可以通过清晰的逻辑划分而简单化。比如：工作繁琐的事情可以按照事件的重要性和紧急度进行这两个维度的划分，如图所示。



**图3.2.1 四象限基本分析原理图**

根据重要性和紧急度两个维度可以将事情划分为四个象限：第一象限的事情是首先要进行处理的，第二象限的事情是应该在第一象限处理后进行处理的，第三象限的事情可以在重要的事情处理完成后，在闲暇的时间内进行处理，第四象限的事情是不在预期范围内突然发生的事情，比如，在你处理第二象限的事情时，来了一个电话，也许不重要，但是你必须要接这个电话。

在软件工程进行需求分析的时候，四象限分析法可以帮助你清晰定位软件的功能特点和将来要进行研发的创新点，如图所示：



**图3.2.2 四象限基本分析图**

****杀手功能****：针对某一应用需求我们的竞争对手和用户已经决定了的必要满足的需求，我们的创新功能也在这一部分。

****外围功能****：辅助性的功能，比如良好的用户体验界面等。

****必要需求****：能够准确解决用户主要功能的需求，比如词典软件要保证短语释义的准确性。

****辅助需求****：非必要需求，比如换肤等功能。

针对不同的功能，有五种解决方法：

1.  **维持**——以最低成本维持此功能

2.  **抵消**——快速地达到“足够好”、“和竞争对手差不多”

3.  **优化**——花大力气做到并保持行业最好

4.  **差异化**——产生同类产品比不了的功能或优势

5.  **不做**——砍掉一个功能，有时我们不需要做所有的功能

## 3.3功能描述

### 3.3.1 功能描述

功能一:购票

流程图：见上图3.1.1.

前提条件：进行用户的注册。

输入：无。

功能二:影片的管理

前提条件：必须是影院的工作人员。

输入：电影的编号和一些基本信息。

功能三:用户注册与登录

前提条件：无。

输入：用户，密码，电话号码，电子邮箱。

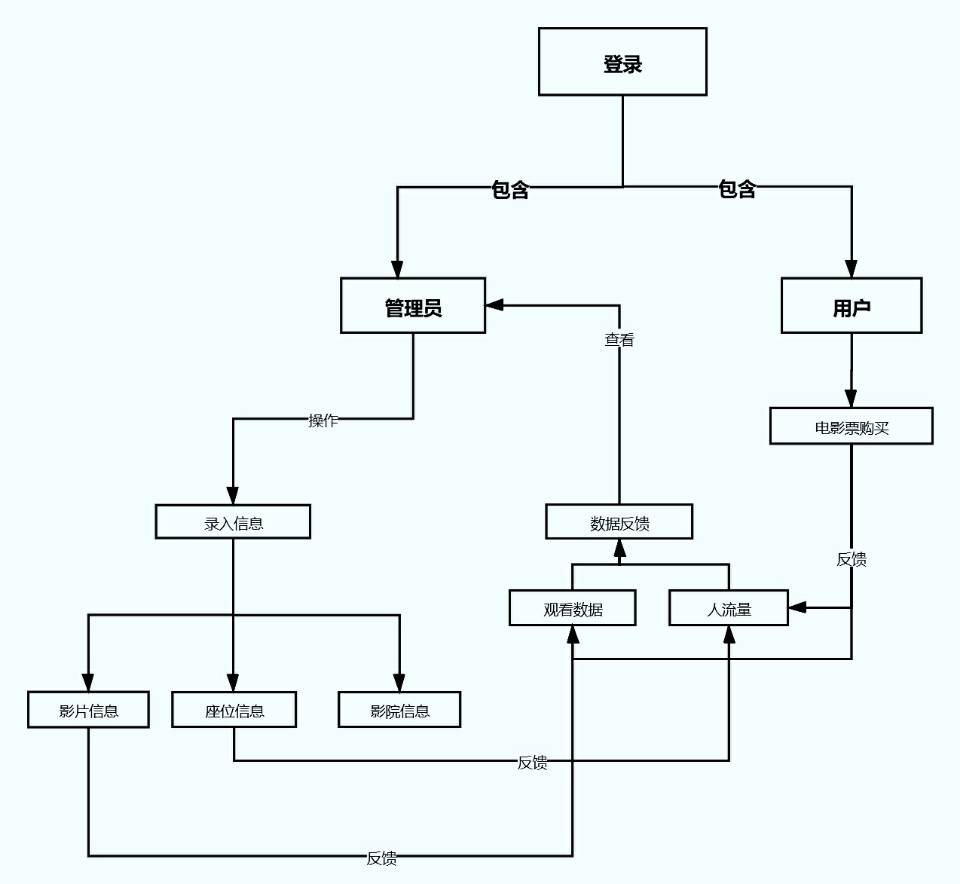
功能四:退票

前提条件：购买了需要退票的影片。

输入：影片名，座位号。

### 3.3.1 功能描述

具体的uml模型如下：



**图3.3.1 UML图**

# 4.验收准则和验收程序

## 4.1验收程序

测试工作主要包括功能测试和性能测试两部分。然后将已完成的系统从开发环境迁移至发布环境。有计划的发布功能和数据直至全部开放进行商务运作。我们将记录并转移一切客户必须掌握和了解的技术与规范方面的知识,保证客户懂得如何运作及维护系统。其次还要进行业务流程测试、容错测试、安全性测试、权限测试和易用性测试等一系列的小的细节测试；通过这些测试统来完善系统的一些小的漏洞，使我们的系统更加完美。最后用户和我们将在一个有限的范围内对系统进行试运行,系统试运行一段时间后,系统将投入正式运行。

## 4.2质量要求

质量要求分为以下3点：

（1）用户需求是度量软件质量的基础。

（2）指定的标准定义了一组指导软件开发的准则。

（3）没有显式描述的隐含需求(如期望软件是容易维护的)。

综合上面几点，验收时除了要满足基本的购票，退票，统计等基本功能外，我们设计的软件要充分的考虑数据的量级和系统的一些易维护性等一些细节的问题。

使我们设计的系统更加完善。

## 4.3验收准则

用软件工程项目的错误与严重性等级对应表来进行验收，这张表将错误分为5级分别如下所示：

（1）一级错误描述：这一级别的错误一般包括以下内容:没有实现或错误地实现重要的功能﹔业务流程存在重大隐患﹔软件在操作过程中由于软件自身的原因自动退出系统或出现死机的情况﹔软件在操作过程中由于软件自身的原因对系统或数据造成破坏﹔在现有的软﹑硬建设环境下不能实现应有的功能﹔特殊软件在操作过程中可能危及系统和人身安全等。

（2）二级错误描述：这一级别的错误一般包括以下内容:没有实现基本的功能并且不存在替代办法﹔没有实现重要功能中的部分功能﹐并且不存在替代办法﹔业务流程衔接错误﹔密钥以明文方式存储﹔没有留痕功能﹔用户的权限分配不合理﹔在现有的环境下﹐不能实现部分功能且没有替代方案﹔没有满足系统的性能要求。

（3）三级错误的描述：这一级的错误是与第2级别的错误相对应的﹐而第3级错误则存在替代方法﹔对误操作或错误操作没有提示，导致非法数据进入数据库。

（4）四级错误的描述：这一级别的错误通常为易用性方面的错误。比如界面不友好、前后风格不一;中英文混杂﹔查询结果输出不直观等。

（5）五级错误的描述：通常为文档方面的错误﹐如安装手册﹑操作手册﹑维护手册中的描述错误。其次﹐对发现的每一个错误都要确定相应的严重性等级﹐如表2中的说明。全部改正方可﹔如错误的级别和数量在合同可接受的范围外﹐用户方认为软件不可验收﹐要求开发方在规定的时间内全面整改软件,提交给软件评测中心再次进行完整的验收测试。

其中一般的验收准则为以下几点：

（1）测试用例不通过数的比例<1.5 %;

（2）不存在错误等级为1的错误﹔

（3）不存在错误等级为2的错误﹔

（4）错误等级为3的错误数量小于5;

（5）所有提交的错误已经更正;

# 5.环境要求

## 5.1.硬件要求

（1）CPU：4核及以上。

（2）内存：8GB以上。

（3）硬盘：258GB以上。

（4）外设：打印机。

（5）网络：万维网。

（6）处理器要求:600 MHz Pentium IIIb）

（7）内存要求:256 MB

（8）硬盘要求:系统驱动器上需要900 MB 以上的可用空间，安装驱动器上需要3.3 千兆字节(GB)的可用空间。

（8）Super VGA (1024x768)或更高分辨率的显示器（颜色设置为 256色）e）能够运行Windows操作系统的PC机，有外联的输入设备以及打印机

## 5.2软件要求

（1）操作系统：windows操作系统；

（2）浏览器：最好用火狐或谷歌浏览器；

（3）网络环境：要求网络环境比较好；

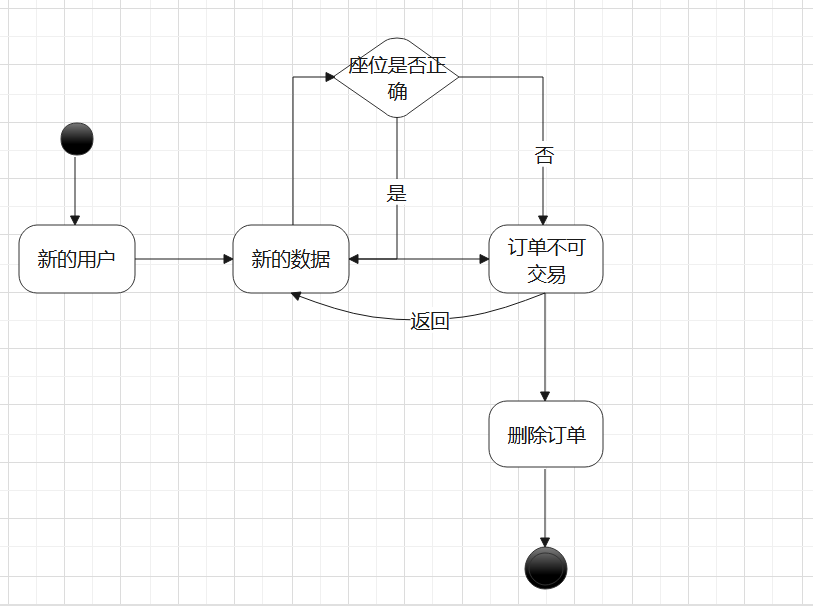
（4）工具：Tomcot，jdbc包，spring+MVC框架；

（5）服务器：WEB服务器；

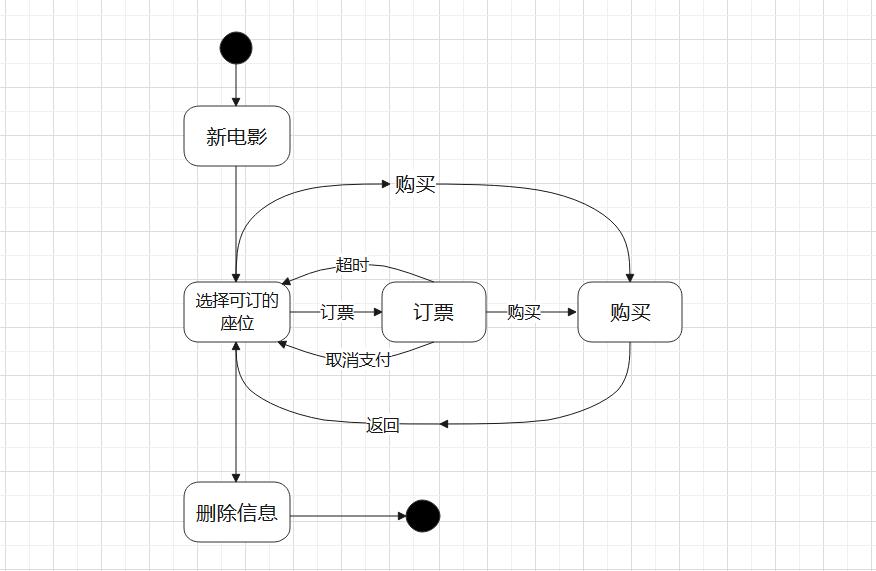
# 6.附录

## 6.1系统状态图

图6.1.1和6.1.2为系统售票和订票的状态图,比较详细的说明了售票和售票的过程。



**图6.2.1售票状态图**



**图6.2.2 订票状态图**