

# **WeSee 电影售票系统**

## **——软件设计文档**

2017/06/24



# 1.引言

## 1.1 编写目的和范围

本次设计文档编写的目的是说明电影售票系统前后端的设计考虑，包括使用的开发环境，开发语言，框架，插件的选择，整体的设计架构思路，模块如何划分等内容。为系统的开发和维护提供基础。

## 1.2 术语表

序号	术语或缩略语	说明性定义
1	nodejs	Javascript 在后台运行的环境
2	Owl Carousel	强大、实用但小巧的 jQuery 幻灯片插件
3	Minified	是一个客户端 JavaScript 库，在一定范围内媲美 jQuery 和 MooTools。它的功能包括 DOM 操作，事件，动画，cookies 和 HTTP 请求。整个库大小小于 4kb。
4	iconfont	国内功能很强大且图标内容很丰富的矢量图标库
5	noUiSlider	小巧的 jQuery 区域范围滑块插件演示
6	Isotope	是一款效果非常神奇的元素分类过滤和排序布局 jQuery 插件
7	mongodb	一个基于分布式文件存储的数据库
8	express	是一个简洁而灵活的 node.js Web 应用框架, 提供了一系列强大特性帮助你创建各种 Web 应用, 和丰富的 HTTP 工具。
10	Powerdesigner	数据库建模工具
11	Umlet	UML 绘图工具

## 1.3 使用的文字处理和绘图工具

主要使用 Umlet 作为 UML 绘图工具，Powerdesigner 作为数据库的建模工具。

# 2 设计概述

## 2.1 整体架构思路

WeSee 互联网电影主要是为了方便我们能够更加便利快捷地购买电影票。该系统的核心功能是用户可以在线选座购票，查看电影信息和影院信息。所以从前端来看我们的目标主要是实现以下这些页面：电影主页，正在热映页面，即将上映页面，个人中心，电影详情

以及三个弹出窗口的实现：登陆窗口，注册窗口，订票窗口

后台的目标是数据模型的构造，数据库的搭建，框架的使用和路由的编写。其中路由编写包括：/hom，/now\_show\_page，/later\_show\_page，/personal\_center，/movie\_detail\_x (x 表示第几部电影)

数据模型包括：用户信息表：users，电影信息表：moviedatas，座位表：seatdatas，电影订单表：orderdatas

## 2.2 技术选型和理由

前端需要主要使用到 HTML5,CSS3, javascript，Jquery，Owl Carousel，iconfont，noUiSlider，Isotope 等技术。

本次要实现的是一个电影搜票系统，设计到很多图片的动画，因此可以考虑使用 Owl Carousel 来实现图片的左右切换。Owl Carousel 是一个强大、实用但小巧的 jQuery 幻灯片插件，它具有一下兼容所有浏览器，支持响应式，支持 CSS3 过度，支持触摸事件，支持 JSON 及自定义 JSON 格式，支持进度条，支持自定义事件，支持延迟加载，支持自适应高度等特点。应用它在我们主页上实现大图片的切换。

另外还有两个页面，正在上映和即将上映，这里面涉及到我们根据不同的电影类型电影进行分类，可以考虑使用 Isotope 插件，Isotope 是一款效果非常神奇的元素分类过滤和排序布局 jQuery 插件。Isotope 是 Masonry 布局的作者 David

DeSandro 的一款力作，该分类过滤和排序插件允许你以非常简单和炫酷的方式来隐藏和显示元素，以及对元素按照指定的规则进行排序。

Iconfont 是国内功能很强大且图标内容很丰富的矢量图标库，用户图标，电话图标，邮箱图标等都使用到它。

后台使用 nodejs 作为 javascript 的运行环境，express 作为开发框架，使用 jade 作为模板引擎，数据库环境需要配置 Mongodb。

npm 提供了大量的第三方模块，其中不乏许多 Web 框架。之所以选择 express 是因为它的轻量级，灵活性和简洁性。它提供一系列强大的功能，比如：模板解析、静态文件服务、中间件、路由控制等等,是目前最流行的基于 node.js 的 Web 开发框架，并且支持 Ejs、jade 等多种模板，可以快速地搭建一个具有完整功能的网站。

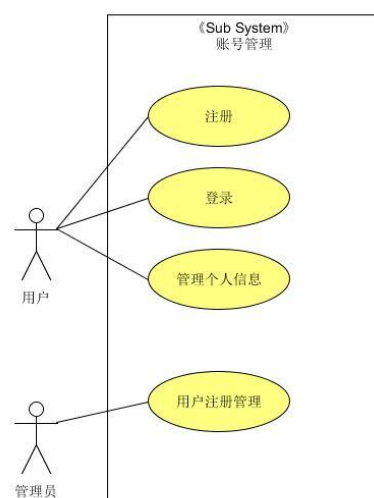
## 2.3 条件与限制

本次系统的开发应该存在许多局限性。首先是数据来源的限制，开发电影搜票系统做出来的只是一个雏形，存在数据库里面的数据没有同步到现在日新月异的市场；另外就是开发资源的局限，无法进行有效的实用性测试。

# 3.模块设计

## 3.1 登陆注册

### 3.1.1 用例图



### 3.1.2 实现图

WE SEE电影售票系统用户登录

用户名:

账号/邮箱

密 码:

密码

登录

W e See

电影售票系统

WE SEE电影售票系统用户注册

用户名 :

6~15位英文字母、数字或下划线用户名

密 码 :

6~18位英文字母、数字或下划线密码

重复密码:

再次输入密码

邮 箱 :

输入电子邮箱, 注意邮箱格式

电 话 :

11位数字电话号码, 不能以0开头

注册

W e See

电影售票系统

### 3.1.3 功能设计说明

登陆和注册主要是使用弹出窗口的方式来实现，没用单独的页面。

对于登陆，用户点击后弹出登陆窗口，输出用户名和密码，数据通过 ajax 的方式 post 到后台进行验证，如果用户名和密码不正确则进行提示，如果输入正确那么窗口消失在页面的右上角登陆处给出用户名，表明用户已经登陆成功。后台用 session 维护输出，保证用户在不同页面之间的切换始终保持登陆状态。

对于注册，用户点击注册按钮后弹出注册框，输入用户名，密码，手机，邮箱进行注册，在前端进行格式验证。使用正则表达式。验证的主要内容是

用户名：6~15 位英文字母、数字或下划线

密码：两次输入密码一致

邮箱：正确的邮箱格式

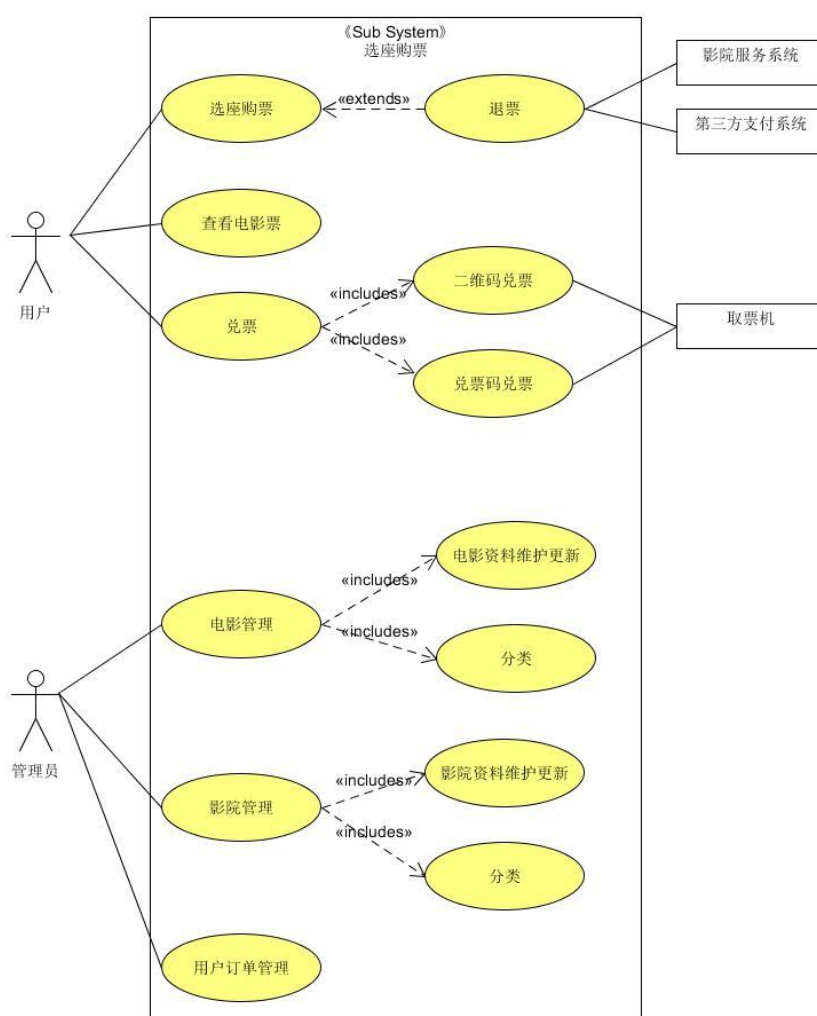
电话：11 位数字电话号码,不能以 0 开头

如果 HTML5 的 pattern 新特性，那么需要设计为 form 并且有一个提交 submit 按钮，但是由于是弹出窗口而不是单独的注册页面，所以使用 ajax 与后台进行数据交互，所以在前端验证需要写脚本，后台检查注册用户名，邮箱，电话是否重复。

这里涉及到 user 数据表，在下面的数据库设计说明。

## 3.2 订票

### 3.2.1 用例图



3.2.2 实现图



### 3.2.3 功能设计说明

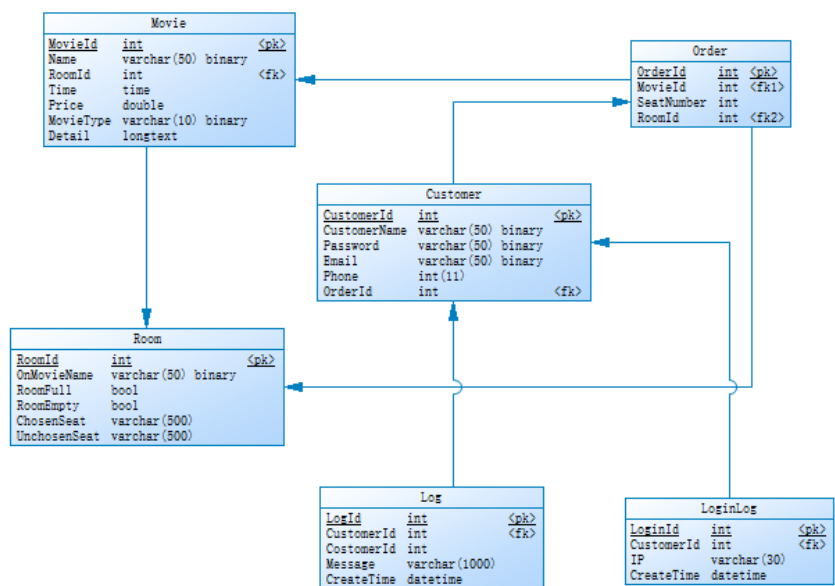
首先是通过 `getSeatData` 向后台请求座位表信息，座位表通过 01 字符串来传送，1 表示已经售出的座位，0 表示可选座位，解析呈现在页面上。用户点击购买后，第一步进行用户登陆判断，如果未登录，弹出登陆窗口，如果登陆好则进行票数判断，如果非 0，弹出购票成功，将购票信息发送到后台。

这里涉及到 2 个数据表，`seatdatas` 座位表和 `orderdatas` 订单表，在下面数据库说明。

## 4 数据库设计

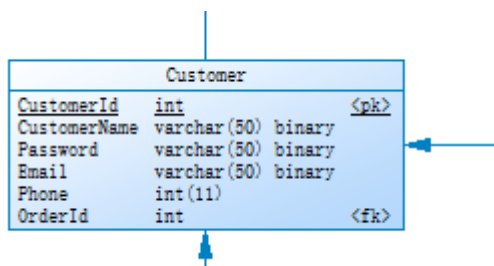
本数据库系统由 `Moviedatas`（电影信息详情表），`Orderdatas`（用户订单详情表），`Users`（用户信息管理表），`Seatdatas`（电影放映房间表），`Log`（日志表），`LoginLog`（用户登录记录表）共六个部分组成。数据库系统整体的物理模型如下图所示：





## 4.1 用户信息管理

### 4.1.1 物理模型：



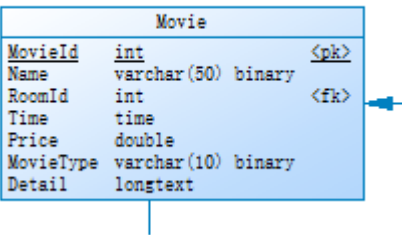
### 4.1.2 数据字典设计

编号	1	表名	Customer
表描述	记录电影售票系统的用户信息		
字段	类型		备注
<b>CustomerId</b>	uniqueidentifier not null primary key		用户标识
<b>CustomerName</b>	varchar(50) not null		用户名称
<b>Password</b>	varchar(50) not null		用户密码

<b>Email</b>	varchar(50) not null	用户 Email 地址
<b>Phone</b>	Int(11) not null	用户联系电话
<b>OrderId</b>	int foreign key	用户的订单标识

## 4.2 电影信息管理

### 4.2.1 物理模型：

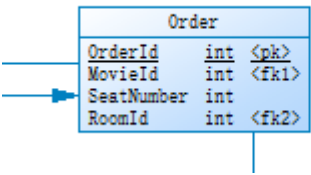


### 4.2.2 数据字典设计

编号	2	表名	Movie
表描述	记录当前热映电影的相关信息		
字段	类型	备注	
<b>MovieId</b>	uniqueidentifier not null primary key	电影标识	
<b>Name</b>	varchar(50) not null	电影名称	
<b>RoomId</b>	int not null foreign key	电影上映对应的房间号码	
<b>Price</b>	double not null	电影的价格	
<b>Time</b>	time	电影上映的时间	
<b>MovieType</b>	Varchar(10) not null	电影类型（2D/3D/IMAX）	
<b>Detail</b>	longtext	电影的其他详情	

## 4.3 订单信息

4.3.1 物理模型：

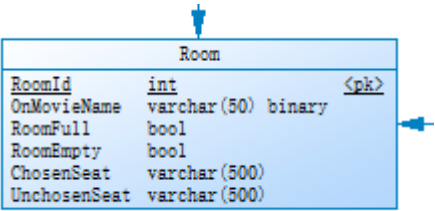


4.3.2 数据字典设计

编号	3	表名	Order
表描述	记录用户的订单详情		
字段	类型	备注	
<b>OrderId</b>	uniqueidentifier not null primary key	订单标识	
<b>MovieId</b>	int not null foreign key	用户订单的电影标识	
<b>SeatNumber</b>	int not null	用户选择的座位号码	
<b>RoomID</b>	int not null foreign key	房间号码	

4.4 电影房间信息管理

4.4.1 物理模型：



4.4.2 数据字典设计

编号	4	表名	Room
表描述	记录电影院房间的信息		

字段	类型	备注
<b>RoomId</b>	uniqueidentifier not null primary key	房间号码
<b>OnMovieName</b>	varchar(50) not null	该房间正在上映的电影名称
<b>RoomFull</b>	Bool not null	房间是否满座
<b>RoomEmpty</b>	Bool not null	房间是否为空
<b>ChosenSeat</b>	Varchar(500)	记录已经选择的座位信息
<b>UnchosenSeat</b>	Varchar(500)	未被选择的座位信息