**电影选票系统需求分析说明书**

[一． 引言 1](#_Toc316041091)

[1. 编写目的 1](#_Toc316041092)

[2. 背景 1](#_Toc316041093)

[3. 定义 2](#_Toc316041094)

[二． 任务概述 2](#_Toc316041095)

[1．目标 2](#_Toc316041096)

[2. 用户的特点 2](#_Toc316041097)

[3. 系统功能示例 2](#_Toc316041098)

[三． 需求细则 2](#_Toc316041099)

[1. 对功能的规定 2](#_Toc316041100)

[2. 对性能的规定 5](#_Toc316041101)

[3. 对排版的规定 5](#_Toc316041102)

[4. 对可维护性的规定 5](#_Toc316041103)

[5. 对个性的规定 6](#_Toc316041104)

[6. 对项目过程的规定 6](#_Toc316041105)

# 引言

## 1.编写目的

通过与多位软件使用者进行全面深入地探讨和分析，并完成《电影选票系统》市场的前期调查后，提出了这份软件需求分析说明书。

此需求分析说明书对《电影选票系统》软件做了全面细致的用户需求分析，明确所要开发的系统应具有的功能、性能与界面，使系统分析人员及软件开发人员能清楚地了解用户的需求，并在此基础上进一步提出概要设计说明书和完成后续设计与开发工作。

本说明书的预期读者为客户、业务或需求分析人员、测试人员、用户文档编写者、项目管理人员。

## 2. 背景

## 3. 定义

# 任务概述

## 1．目标

## 2. 用户的特点

## 3. 系统功能示例

**需求：**

1. 电影场次管理

对于电影场次进行增、删、查、改

生成影票

1. 影票管理

订单的增、查

1. 使用数据库（mysql或sqlite）保存用户信息、场次信息、影票信息

用户表，电影表，影院表，场次表，影票表，评论表

**技术要求：**

1. 影院类
2. 电影类
3. 场次类
4. 影票类
5. 用户类
6. 评论类
7. 应用MVC模式

购票流程：

1. 用户注册登录，浏览电影列表，选择电影进入购票界面，购买影票，通过影票来提交电影评论。
2. 浏览电影的界面应当包括电影的列表，电影的预览图，电影相关的文字介绍，所有用户对该电影的评论。
3. 购买影票界面应当提供对场次的搜索工具，搜索完成后显示相关的场次列表，通过选择相关场次显示该场次的座位信息。
4. 评论界面应当显示当前用户的购票记录，如果该记录有评论过则显示评论，如果没有则显示文本框让用户输入。

# 需求细则

## 对功能的规定

需求是整个工程的开始，对需求的深刻了解可以提高开发过程中的设计与理解，此需求要求侧重点为：功能模块、设计及代码规范、产品优化及健壮性。分基本功能、主要功能和提高功能，其中，基本功能为必须完成功能，如缺失则不能参加答辩；主要功能指难度适中，适合所有学员的功能，要求全部实现；提高功能适于能力较强的学员完成，通过这类功能，可以对自己的技术能力和综合水平有显著提高。

1. 实体类基本功能

属性：基本

* 1. 用户实体类

描述：包含用户相关内容的相应多种类，应均有各种属性及基本方法

* 1. 管理员实体类

描述：包含管理员相关内容的相应多种类，应均有各种属性及基本方法

* 1. 电影信息实体类

描述：包含电影信息相关内容的相应多种类，应均有各种属性及基本方法

1. 数据库

属性：基本

描述：用数据库合理存储操作过程中产生的需要保存的数据，应与实体类内容相对应。

1. MVC模式

属性：主要

描述：使用MVC模式对C控制、V视图、M模型层进行封装和处理

1. 管理员主体功能

属性：主要

描述：管理员可以在登录后对电影场次进行增删改查等操作，可以对影票订单进行增加和查看等操作。

1. 用户主体功能

属性：主要

描述：用户可以在注册登录后，浏览电影信息（包含预览图、信息及评论），并查找场次进行选座购票，购票之后可以进行评论。

1. Swing

属性：提高

描述：使用Swing制作用户交互界面，实现购票系统的窗口化管理

1. 推荐及自选个性功能

属性：提升

描述：完成推荐功能或自定义个性功能，为应用增添个性功能以提高应用的趣味性、安全性等。

推荐：提供预告片演示、抽奖免单、VIP用户等

## 对性能的规定

本软件在设计方面本着方便、实用及娱乐性高的宗旨，性能方面应遵循如下原则：

* 流畅性： 软件运行应该流畅，不会出现卡机；
* 美观性： 软件界面应力求美观，赏心悦目；
* 稳定性： 软件运行过程中不会出现BUG；
* 边界条件：主界面应该全屏；
* 资源损耗：设计尽量节约资源；不用的资源应及时回收（如服务结束，线程应退出）；

## 3. 对可维护性的规定

对可维护性的最终要求：别人能够轻松上手你的代码。

* 结构清晰：封装良好；继承关系清晰；代码块清晰；
* 变量命名规范，变量名应该具有自明性；
* 注释充分：变量、方法、代码功能块、一些复杂算法……等都需要清晰明了地说明；

## 4. 对个性的规定

把项目做出个性出来。下列各项中有和比人不同之处、或很有创意，即可认为有个性。独立设计的软件，一般都会出现一些个性。参考、抄袭不会出现个性。

* 设计思路：包括软件的整体架构、功能块的设计思路等等；
* 功能实现：从用户的角度，使用上发现与众不同的地方；
* 其它方面；

## 5. 对项目过程的规定

本着紧张但不急躁、不参考、不拷贝的原则进行。

* 紧张但不慌张

项目周期只有两周，这还包括项目答辩时间。所以项目时间比较紧张，但不能慌张。要有自己明确的设计思路，一步步沿着思路走下去，以此来巩固自己所学，锻炼自己的独立工作能力。

* 能自己做，绝不参考别人

自己还没有做，还没有想，就去看比人的，这样尽管功能做出来了，但却没有什么意义，真正面试时还是不会。作者和读者，天壤之别。

如果自己实在无法搞定，一个问题卡了快一天了，则可以咨询别人一下想法，再行编码；尽量不直接看别人代码。

* 不拷贝

**一旦发现拷贝，取消答辩资格。答辩时发现，答辩成绩减半。**

copy别人的代码，甚至直接运行别人的代码，以此作为自己的项目进展，这是严禁的。**严禁运行效果出来了，却不知道是哪些代码造成的，严禁明明是自己写的代码，但却不知道为什么这么写。**

# 四．运行环境

## 1. 设备

PC

## 2.支持软件

Eclipse