

吉首大学张家界学院

**本科生课程设计**

|  |
| --- |
| 软件工程导论 |
| 电影订票系统 |
| 宁锭、肖方仁、易荣添、汪意、谭皓宇、阳乐臻 |
| 理工农学院 |
| 21级软件工程2班 |
| 田颐 |
| 2023年6月15日 |

课程名称：

课题名称：

组 员：

学 院：

年 级：

指导教师：

填写时间：

目录

**[1. 可行性研究 2](#_Toc24016)**

[1.1 技术可行性 2](#_Toc10984)

[1.2 经济可行性 2](#_Toc7860)

[1.3 社会可行性 2](#_Toc9865)

**[2. 需求分析 3](#_Toc18446)**

[2.1 用户需求 3](#_Toc20577)

[2.2 功能需求 3](#_Toc26870)

[2.3 非功能需求 3](#_Toc25083)

**[3. 设计阶段 3](#_Toc26708)**

[3.1 数据库设计 3](#_Toc5120)

[3.2 系统架构设计 3](#_Toc23556)

**[4. 实现阶段 4](#_Toc19793)**

[4.1 开发工具 4](#_Toc24080)

[4.2 代码组织 4](#_Toc31694)

[4.3 实现顺序 4](#_Toc30929)

**[5. 测试阶段 4](#_Toc27778)**

[5.1 测试环境 5](#_Toc18673)

[5.2 测试类型 5](#_Toc26126)

**[6. 维护阶段 5](#_Toc11430)**

**[7. 结语 5](#_Toc8070)**

**1. 可行性研究**

**1.1 技术可行性**

该系统应基于网络进行开发，用户可以通过电脑或手机等设备访问网页来进行电影订票。考虑到技术的可行性，本系统应选择成熟可靠的技术方案来进行开发。对于前端，可以选择使用 HTML、CSS 和 JavaScript 进行开发；对于后端，可以采用 node.js 进行开发，同时应该采用一些现有的开源组件来加快开发进度和提高代码质量。此外，对于数据存储，考虑到应用场景，使用更加轻量级的 NoSQL 数据库是更合适的选择。

**1.2 经济可行性**

该系统的开发和运营成本需要投入一定的资金，需要通过收取用户购票的手续费来实现盈利。市场上已经有多家电影订票平台，因此需考虑市场占有率和盈利能力是否足够。需要进行详细的市场分析，确定财务收支计划，并评估项目的投资回报率和成本效益。

**1.3 社会可行性**

该系统可以为用户提供方便快捷的电影票务服务，符合广大观众的需求，具有广泛的社会意义。同时，该系统也有助于电影行业的数字化转型，提高行业的效率和竞争力。

**2. 需求分析**

**2.1 用户需求**

该系统需要支持用户浏览电影信息和影院信息，并能够在线选购电影票。用户可以在系统中选择座位和支付电影票的价格，并可以在线查看已购买的电影票和对已购买的电影票进行取消操作。需要注意的是，用户在系统中进行的所有操作应该具有良好的用户体验。

**2.2 功能需求**

该系统需要实现的功能包括用户注册、用户登录、浏览电影信息、选座位、购票、支付票价、取消订单、查看订单、查看电影和影院信息等。

**2.3 非功能需求**

系统应该考虑到用户的数据安全性和隐私保护。另外，系统应该具有高可用性，防止系统崩溃或宕机等情况发生。

**3. 设计阶段**

**3.1 数据库设计**

该系统需要设计用户表、电影信息表、影院信息表、订单表以及座位表。用户表需要包含用户基本信息、登录凭据以及购票历史信息。电影信息表和影院信息表需要记录电影及影院的基本信息和场次，订单表需要记录用户选购的电影票信息，座位表需要记录每个场次的座位信息。

在数据库设计中应注意到表之间的关系和索引的设计，这对于提高系统的性能和响应速度有重要影响。

**3.2 系统架构设计**

本系统采用分布式架构，将各个功能模块分离成不同的服务，通过 API 接口来实现各个模块之间的通信。

此外，应该采用前后端分离的设计模式，将前端代码和后端代码分别部署在不同的服务器上，以提高系统的运行效率和稳定性。

**4. 实现阶段**

**4.1 开发工具**

开发工具包括代码编辑器、数据库管理工具和版本控制工具等。在本次开发中，可以选择使用 Visual Studio Code 作为代码编辑器，使用 Robo 3T 作为数据库管理工具，使用 Git 进行版本控制。

**4.2 代码组织**

在代码组织上，可以采用前后端分离的方式进行开发。前端代码可以采用模块化的方式进行组织，使用类似 React 或 Vue.js 等前端框架进行开发，以实现更好的可维护性和代码复用性。

后端代码可以采用 MVC 设计模式进行组织。其中，Model 层负责数据的存储和查询，View 层负责页面的渲染和用户交互，Controller 层负责业务逻辑的处理。同时，为了提高代码的可扩展性和可维护性，需要采用一些开源组件来加快开发进度和提高代码质量。

**4.3 实现顺序**

在实现阶段，可以按照以下步骤进行开发：

（1）设计数据模型，创建数据库表；

（2）实现用户注册与登录功能、浏览电影信息和影院信息功能；

（3）实现选座位、购票、支付票价、取消订单、查看订单等功能；

（4）制作 API 接口，实现模块之间的通信；

（5）进行整体测试，修复存在的问题。

**5. 测试阶段**

**5.1 测试环境**

在测试阶段，可以采用 Docker 容器，模拟真实环境进行测试。设置一个与生产环境相同的测试环境可以更加准确地发现问题。

**5.2 测试类型**

测试类型包括单元测试、集成测试、系统测试和安全性测试。其中，单元测试主要针对代码的功能进行测试，集成测试主要针对模块的集成和通信进行测试，系统测试主要测试整个系统的功能和性能，安全性测试主要测试系统的安全性。

**6. 维护阶段**

在系统上线后，需要进行定期的维护和更新。维护内容包括修复已知的漏洞，更新系统和数据库的版本等。另外，还需要处理用户反馈的问题和建议。需要建立反馈机制，及时响应用户反馈，并根据反馈意见进行完善和更新。

维护阶段的任务不仅仅是问题修复，更重要的是系统的优化。可通过对代码、数据库、网络等方面的优化，提高系统的性能和可扩展性，从而保持系统持续运行的稳定性和安全性。

**7. 结语**

本文档介绍了一个电影订票系统的设计和实现过程，从可行性研究、需求分析、设计、实现、测试和维护六个方面进行了详细的阐述。希望本文档可以为电影订票系统的开发提供指引，并帮助读者更好的理解软件工程实践中的各个环节。