

重庆大学本科学生毕业论文（设计）

电影院订票 APP 设计与实现



学 生：蓝兴烨

学 号：20204155

指导教师：杨广超

专 业：计算机科学与技术

重庆大学计算机学院

2024 年 6 月

Undergraduate Thesis (Design) of Chongqing University

**Design and Implementation of Cinema
Ticket Booking App**



By

LAN Xingye

Supervised by

YANG Guangchao

Computer Science and Technology

College of Computer Science

Chongqing University

June, 2024

摘 要

近年来，随着网络技术的飞速发展，电影院订票 APP 作为一种现代化的电影票务解决方案，展现了其重要的应用和推广价值。互联网的高速发展和电子商务的全面覆盖造成了人们工作和生活方式的剧烈变化，推动了行业的技术升级。电影院订票 App 便是传统影院企业跟随时代潮流而做出的决策实现，企业通过发达的计算机服务和电子商务技术一并融合到旧时的影院业务中，实现线上线下业务的有效整合和优化。

本课题围绕电影院线上购票选座展开，并以此设计管理系统。对于用户端提供查询电影和电影院信息和线上购票选座的功能，而对于管理端提供电影和电影院的管理模块。系统采用 Spring Boot 和 Uni-App 框架，并利用 IDEA、HBuilder X 等工具开发。

本文将介绍此系统开发的背景、意义和现状、系统开发的具体技术框架与工具、需求分析、系统的设计和具体实现。最后完成系统的测试和开发，确保此 App 项目能够稳定运行在各移动端设备和各电脑设备的 Web 端。

关键词：电影院；APP；线上购票选座

ABSTRACT

In recent years, with the rapid development of network technology, cinema ticket booking apps have emerged as a modern solution for movie ticketing, with significant application and promotional value. The widespread adoption of the internet and the rise of e-commerce have brought about dramatic changes in work and lifestyle, driving the modernization of traditional industry operations. Online cinema ticket booking systems are an initiative by traditional cinema enterprises to adapt to this transformation, integrating advanced internet and e-commerce technologies into traditional cinema business.

Internationally, due to its early adoption and rapid industry development leveraging internet technology, online movie ticket sales have gained significant success. Currently, in the domestic market, online ticketing services still hold enormous potential and space for development. Therefore, this paper focuses on the design and implementation of a cinema ticket booking app that allows online ticketing and seat selection.

This project revolves around online ticketing and seat selection for cinemas, serving as the basis for designing a management system. For users, it provides functionalities such as querying movie and cinema information and online ticketing with seat selection. For administrators, it offers modules for managing movies and cinemas. The system is developed using Spring Boot and Uni-App frameworks, along with tools like IDEA and HBuilder X.

The paper will cover the background and current state of app system development, specific technical frameworks and tools used in system development, requirements analysis, system design, and implementation details. Finally, thorough testing and development will ensure that this app project runs smoothly on various mobile and computer devices via the web.

Key words: Cinema;APP; Online ticket purchase and seat selection

目 录

摘要	I
ABSTRACT	II
1 绪论	1
1.1 研究背景	1
1.2 研究意义	1
1.3 研究现状分析	2
1.4 本文组织结构	3
2 相关技术介绍	4
2.1 前后端分离架构	4
2.2 Spring Boot 框架	4
2.3 MyBatis 框架	5
2.4 数据库技术	5
2.4.1 MySQL 数据库	5
2.4.2 Redis 数据库	6
2.5 Uni-App 框架	6
2.6 本章小结	7
3 需求分析	8
3.1 系统角色分析	8
3.2 功能需求分析	8
3.2.1 用户端功能需求	8
3.2.2 影院端功能需求	9
3.2.3 管理端功能需求	10
3.3 非功能需求分析	11
3.4 本章小结	12
4 系统设计	13
4.1 功能模块设计	13
4.1.1 用户端功能模块设计	13
4.1.2 影院端功能模块设计	14
4.1.3 管理端功能模块设计	14
4.2 架构设计	15
4.3 数据库设计	16

4.4 本章小结	20
5 功能实现	21
5.1 系统开发环境	21
5.2 用户端功能实现	21
5.2.1 注册模块实现	21
5.2.2 登录模块实现	23
5.2.3 个人中心模块实现	25
5.2.4 电影模块实现	28
5.2.5 电影院模块实现	30
5.2.6 场次模块实现	31
5.2.7 选座购票模块实现	34
5.3 影院端功能实现	35
5.3.1 登录模块实现	35
5.3.2 账号信息模块实现	36
5.3.3 放映厅信息管理模块实现	37
5.3.4 场次信息管理模块实现	39
5.4 管理端功能实现	41
5.4.1 登录模块实现	42
5.4.2 电影信息管理模块实现	42
5.4.3 影院信息管理模块实现	44
5.5 本章小结	46
6 总结与展望	48
6.1 总结	48
6.2 展望	48
参考文献	49
致谢	50
原创性声明和使用授权书	

1 绪论

1.1 研究背景

如今互联网技术的飞速发展人们的生活带来了许许多多的便利，在各计算机软硬件开发人员的努力下，现时社会，计算机和互联网合并成一股全新且方便日常人民生活的力量^[1]。当今电影产业蓬勃发展，数字化和网络化成为主要趋势。在这样的背景下，电影院订票 App 作为电影产业数字化转型的重要组成部分，成为了人们购票的首选方式之一。随着互联网和移动应用技术的飞速发展，用户习惯于通过手机或电脑进行各种购买和预订行为，这种便捷性和高效性逐渐渗透到了电影票购买领域。电影院订票 App 借助网络技术的普及和便利，使用户能够随时随地通过应用程序轻松预订心仪的电影场次，大大提高了购票的便利性和舒适度。

同时，消费者行为的变化也是推动电影院订票 App 发展的重要因素。现代人的生活节奏加快，对时间的利用更加高效，因此更加倾向于使用便捷、快速的方式完成各种活动，包括购买电影票。电影院订票 App 的出现满足了用户对便捷、快速购票的需求，使得电影票预订变得简单轻松，不再需要排长队或提前到电影院购票，极大地提升了用户体验。

如今，电影院订票 App 不仅仅是简单的线上购票，还涉及到座位选择、影片推荐、票务管理等方面的技术创新和业务模式创新。通过先进的技术手段和创新的业务模式，电影院订票 App 已经能够提升用户体验，提高平台竞争力，进而吸引更多用户使用。

最后，一些政策环境的支持也对电影院订票 App 的发展起到了重要推动作用。一些地区对于电影产业的数字化转型给予了政策支持，促进了电影院订票 App 的发展。政策环境的积极支持为其提供了良好的发展环境，也为相关企业和平台提供了更多发展机会和空间。

电影院订票 App 的研究背景涉及到电影产业发展、网络技术进步、消费者行为变化、技术与业务创新以及政策环境支持等多个方面，这些因素共同推动了电影院订票 App 的兴起和发展。随着社会的进步和技术的不断创新，其将在未来持续发展壮大，为用户带来更便捷、更愉快的电影观影体验。

1.2 研究意义

伴随着我国国内的计算机技术和互联网技术迅速崛起，我国的大部分企业也开始跟随潮流开始实现各行业和互联网技术的交叉发展，影视行业也不例外^[2]。传

统的影院线下售票方式有很多问题，不限于售票流程繁琐、人工售票容易出错、成本高昂等问题，同时也存在数据共享困难及其他不确定因素^[3]。通过各种计算机技术框架，能够帮助线下影院的工作人员减轻劳累的工作，并且更加有效率地操作影院的具体数据。越来越多的资本纷纷对使用计算机互联网技术来支持电影院订票业务的举动有所看好。而这种运营方式也确实能够帮助传统影院的持续发展，改善销售流程的现状，避免了高峰期（例如周末）观众买票的情况，有助于提高观众在该影院的整体体验，最后，帮助企业打造更好的企业形象。

近年来，互联网订票体系作为一种全新的互联网商务，通过持续的发展与迭代更新向各企业展示出了巨大的潜力。随着互联网技术的不断延伸，互联网订票体系不仅在城市生根发芽，在农村地区也逐渐普及。在这种趋势下，电影院订票 App 不仅为观影者提供了更便捷、更舒适的购票体验，也为电影院的发展带来了新的机遇。

综上所述，电影院订票 App 的设计与实现不仅使观影者的生活更加便利，也为电影产业的升级和转型提供了新的机遇。随着技术以难以让人估量的速度进行发展，电影院订票 App 将在不久的将来体现出更加至关重要的作用，成为电影产业数字化转型的重要支撑和推动力量。

1.3 研究现状分析

电影院订票 App 作为数字化时代电影票务的重要解决方案，受到国内外研究机构和学者的广泛关注与研究。国内的研究现状主要集中在用户体验、技术应用和市场调研等方面，以提升用户满意度和推动行业发展为主要目标。同时，国外的研究更侧重于用户行为、安全保护和市场竞争分析，致力于为行业的全面发展和创新提供宝贵的经验和理论支持。

在我国，电影行业经历了二三十余年的不断摸索，其相关的各种线下票务业务的系统也经历了许许多多的变化，而这一变化是其从零到一的重大变化，且影响力也在不断地提升直到扩展到全国。如今在国内，许多研究者和机构致力于电影院订票 App 的用户体验研究。他们通过对界面设计、交互体验和功能设置等方面的分析，致力于提升用户在 App 上的使用便利性和满意度。除此之外，国内的科研机构 and 高校也在探索移动互联网技术在电影票务领域的应用，如支付方式创新、座位选择算法优化等，以提升技术应用水平和服务质量。市场调研方面，一些市场研究机构对电影院订票 App 的市场规模、用户群体和竞争格局进行深入调查和分析，为行业发展提供了重要参考和决策支持。

而在国外，电影院订票 App 的研究更侧重于用户行为、安全保护和市场竞争分析。研究者们对用户在 App 上的行为模式和偏好进行了深入调查和分析，从而

为优化 App 的功能和服务提供了有力的数据支持。同时，他们也关注电影院订票 App 在数据安全和隐私保护方面的研究，提出了一些有效的解决方案和技术手段，以应对用户在数字化时代对数据安全和隐私保护的日益增强的关注。此外，市场竞争分析也是国外研究的重点之一，市场研究机构对电影票务行业的竞争格局、趋势和发展预测进行了全面分析，为投资者和企业提供了决策参考和战略指导。

总体而言，国内外的研究现状在不同领域展开，涵盖了用户体验、技术应用、市场调研、用户行为、安全保护和竞争分析等多个方面，为电影院订票 App 的发展提供了丰富的理论和实践支持。

1.4 本文组织结构

此小节将说明本文的组织结构，详细内容见下：

（1）第一章：绪论

着重探讨电影院订票 App 的研究背景、意义和现状分析。

（2）第二章：相关技术介绍

详细介绍了系统开发所需要的具体框架、工具和技术。

（3）第三章：需求分析

详细说明了此系统中各部分的系统角色分析、功能需求分析和非功能需求分析。

（4）第四章：系统设计

该章节根据第三章提出的需求为系统进行设计，包括各端的功能模块设计、系统架构设计和数据库设计。

（5）第五章：功能实现

该章节根据第四章提出的设计为系统进行开发实现，简略地说明了系统的开发环境和工具，并详细地说明了各端的功能实现。

（6）第六章：总结与展望

2 相关技术介绍

2.1 前后端分离架构

前后端分离架构作为一种便于软件开发的架构模式，开发人员能够在前端和后端框架中进行各自的开发，最后通过 API 进行连接。面对传统项目开发模式中的种种问题，开发者可以使用其架构来进行开发^[4]。在这样的架构下，开发者只需要在后端完成接口的实现，在前端发起请求，并完成其之间的数据传输便能实现前端和后端的独立开发^[5]。

前端和后端的独立架构模式易于开发者的维护和升级。除此之外，通过合理的接口设计和数据传输方式可以优化系统的性能和响应速度。最重要的，此架构能支持多端部署，方便扩展到不同的平台和设备。

前后端分离架构图如图 2.1 示

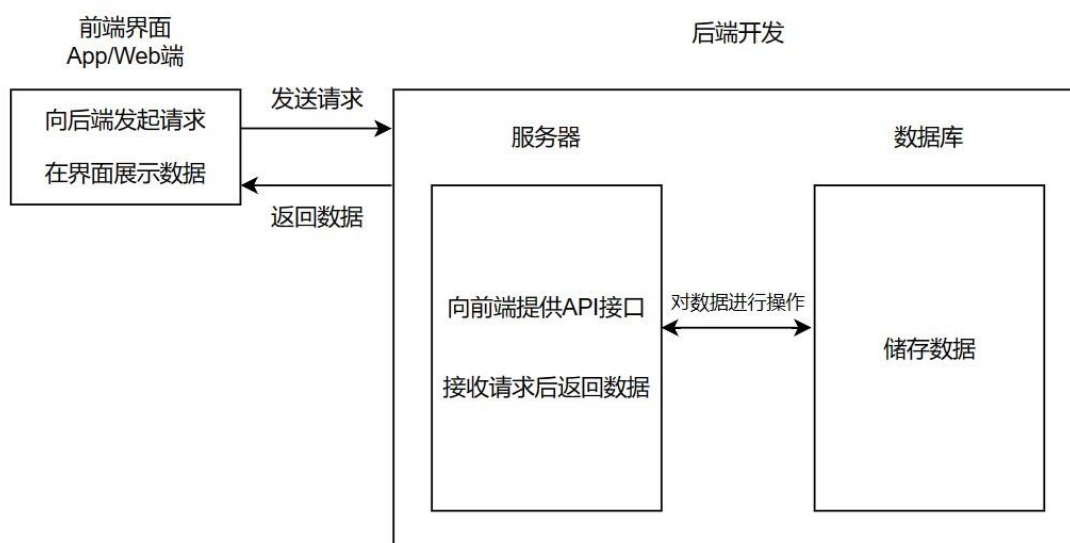


图 2.1 前后端分离架构图

2.2 Spring Boot 框架

Spring Boot 是大量开源库的集合，任何系统都可以采取 Spring Boot 框架进行构建并投入实际使用^[6]。其是一个根据 Spring 结构的应用程序的开发框架。通过使得 Spring 开发过程简易化，给予了一种方便设置和部署的形式。其框架不用开发者手动配置，可以让开发者凭借项目的具体需要，自动处理好项目所需的不同内容，在某种程度上提高了开发者的开发效率。此外 Spring Boot 默认集成了常用

的内嵌服务器，可以将整个项目弄成可执行的 JAR 文件或 WAR 文件，方便部署和运行。Spring Boot 作为 Spring 的 PLUS 版，其通过简化配置、快速启动、依赖管理等特性，使得开发者可以更专注于业务逻辑的实现，快速构建高效、稳定的 Spring 应用程序^[7]。

Spring Boot 框架架构图如图 2.2 示

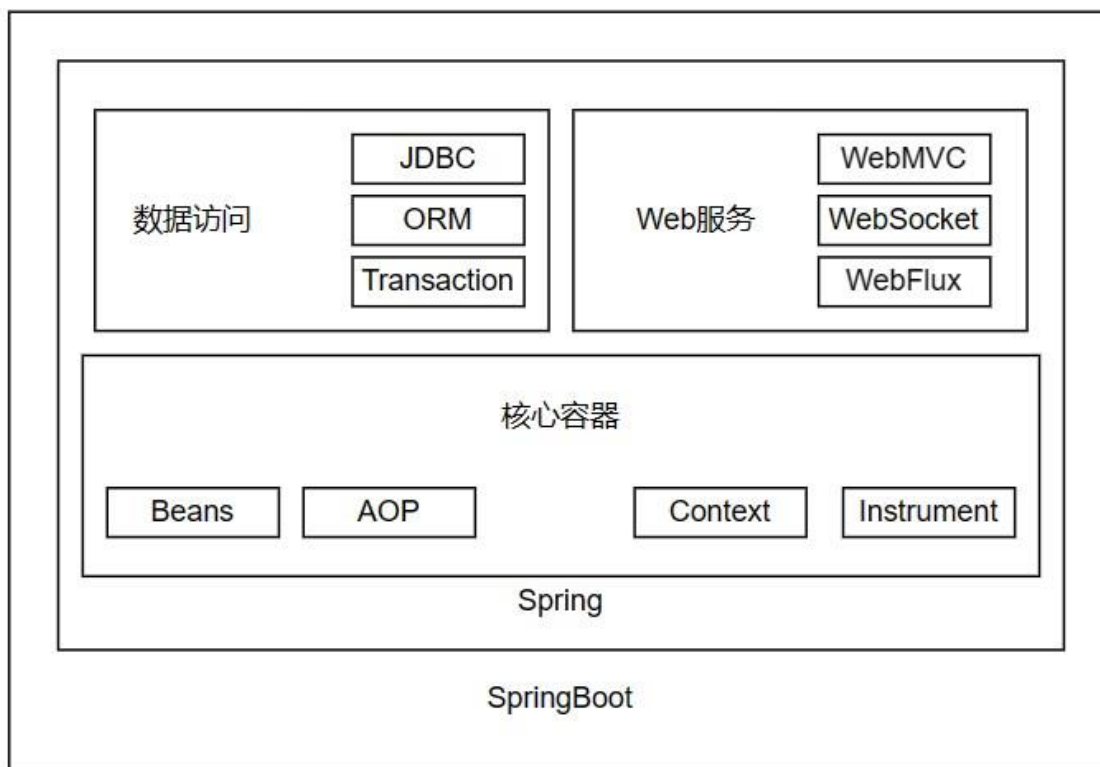


图 2.2 Spring Boot 框架架构图

2.3 MyBatis 框架

MyBatis 是一个用于简化数据库操作的持久层框架，通过 XML 或注解方式将映射接口和 Java 的 POJOs (Plain Old Java Objects, 普通的 Java 对象) 映射成数据库中的记录^[7]，实现了 SQL 与 Java 代码的分离。它支持参数绑定、结果映射、事务管理、缓存机制等功能，同时提供灵活的配置选项和扩展点，使开发者能够高效地进行数据库操作和定制开发。总体而言，MyBatis 框架在简化数据库访问、提高开发效率和代码质量方面具有显著的优势，是开发持久层的可靠选择之一。

2.4 数据库技术

2.4.1 MySQL 数据库

MySQL 是一款开源的关系型数据库管理系统，具有跨平台、高性能、易用性、

可扩展性和数据安全等特点，它广泛应用于各种软件开发项目中，包括 Web 应用程序、电子商务系统、数据分析平台等，为开发者提供了强大的数据存储和管理能力，MySQL 可以提供简单易用的管理工具和 SQL 语言接口^[9]。同时支持多个存储引擎，其存储引擎架构将查询处理以及其他系统任务和数据的存储、提取相分离，使其在使用时可以根据性能、特性，以及其他需求来选择数据存储的方式^[10]。这样使得开发者可以轻松进行数据库管理和操作，并且具备良好的性能和可靠性，能够处理大规模数据和高并发访问。

2.4.2 Redis 数据库

Redis (Remote Dictionary Server) 是一个开源的基于内存的数据结构存储系统，它支持多种数据结构，如字符串 (String)、列表 (List)、哈希表 (Hash)、集合 (Set) 和有序集合 (Sorted Set)，并提供了丰富的命令集用于对这些数据结构进行操作^[11]。用途包括缓存、会话存储、实时数据处理、消息队列等多种场景，其高性能、高并发处理能力、低延迟等特点使其成为互联网应用开发中的重要工具。因为数据的存储是存储在缓存中，Redis 能够以极高的速度读取和写入数据，这使得它在需要快速响应和处理大量请求的场景下表现出色。

在数据结构方面，Redis 提供了多种数据类型和相关命令，例如，对于字符串类型，可以进行设置、获取、追加等操作；对于列表类型，可以进行插入、删除、修剪等操作；对于哈希表类型，可以进行设置字段、获取字段、删除字段等操作；对于集合和有序集合类型，可以进行添加元素、删除元素、获取排名等操作。这些丰富的数据结构和操作使得开发者可以灵活地存储和处理数据。

2.5 Uni-App 框架

Uni-App 框架是一个基于 Vue.js 的跨平台应用开发框架，可以使用 Vue.js 的语法和组件开发方式，同时实现一次编写多端运行的能力，包括 iOS、Android、Web 以及各种小程序平台（如微信小程序、支付宝小程序、百度小程序等）。Uni-App 的主要特性是写一次代码便能够在 Web 和 App 上运行，开发者通过使用的 Vue.js 开发技术来编写应用程序，并且可以通过编译打包工具将应用发布到不同的平台上。除此之外，Uni-App 框架在性能优化方面也做了很多工作，包括减小应用体积、优化渲染性能、提升加载速度等，保证应用在不同平台上的流畅性和稳定性。

Uni-App 功能架构图如图 2.3 示



图 2.3 Uni-App 功能架构图

2.6 本章小结

本章介绍了电影院订票 App 所需的相关技术。先后涵盖了前后端分离架构、Spring Boot 框架、MyBatis 框架、相关数据库技术和 Uni-App 框架。

3 需求分析

3.1 系统角色分析

本项目主要研究电影院订票 App 的设计与实现，故系统角色分为三种。

（1）用户端角色，即通过此系统进行订票的观众。该角色是此系统的主要用户群体，他们可以查看电影信息、进行选座购票、查看个人订单和个人信息管理。

（2）影院端角色。该角色可以根据需求为自己新增或修改放映厅信息，同时，为用户端角色提供不同的放映场次，其可以根据需求设置不同电影、不同放映厅和不同时间的放映场次，并且可以查看该场次的购票情况。

（3）管理端角色。该角色可以管理电影信息，对正在上映或者待映的电影信息进行发布，并且可以对影院端角色的信息进行操作，管理影院端角色的具体信息。

以上三种角色构成了此系统的所有端。

3.2 功能需求分析

电影院订票 App 是面向用户端、影院端和管理端三种不同角色进行开发的，故应满足其各自所需的基本功能。本章将对其各自的需求进行分析，并使用用例图加以说明。

3.2.1 用户端功能需求

此系统是一个面向现实市场的 App，结合实际需求，App 需能够实现将线下电影院订票的这一行为转移到线上进行。此部分先介绍用户端的功能需求。

（1）用户账号注册需求：用户需有一个专属的账号，方确保用户可以进入 App。

（2）用户账号登录需求：用户使用其专属账号进行登录。

（3）查看电影信息的需求：用户能够查看近期正在上映或者待映的电影信息（电影的名称、海报、类型、介绍等信息），浏览电影列表。

（4）查看电影院信息的需求：用户能够查看电影院列表（根据地理位置排序），并且查看电影院的相关信息。

（5）查看场次信息的需求：用户能够查看当前场次的信息，例如当前场次的播放的电影、具体影院的位置、场次的价格、场次的时间等信息。

（6）选座购票的需求：用户可以根据喜好选择电影、电影院、场次和座位。对于用户选择座位来说应具有直观性和易用性，此系统也将为用户提供可以在座位图上直接选择座位的功能。用户可以查看选座的结果、票价和费用总计。

(7) 查看订单信息的需求：用户可以查看已购买的电影票的订单信息。

(8) 个人信息管理的需求：用户可以在个人中心查看并编辑个人资料，包括昵称、头像、联系方式等。

结合以上需求，用户端功能需求的用例图如图 3.1 示

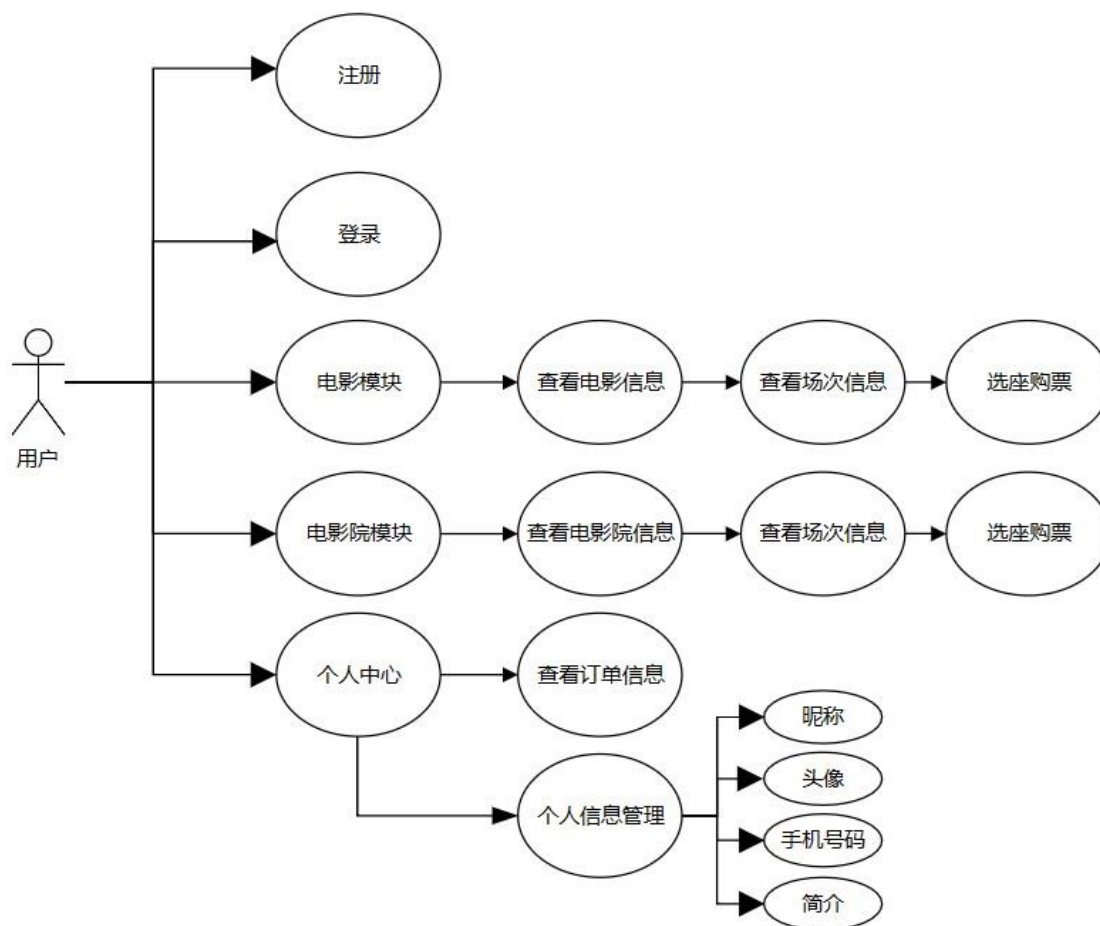


图 3.1 用户端功能需求用例图

3.2.2 影院端功能需求

此部分介绍影院端的功能需求。对于这一角色来说，影院端的功能需求旨在提供给影院工作人员一个便捷、高效的管理平台，让他们能够有效地管理放映厅信息和场次信息，为影院的运营提供服务。

(1) 影院账号登录需求：影院使用其专属账号进行登录，登录到影院管理平台。

(2) 查看当前账号的影院信息：在个人中心中查看当前账号的影院信息（影院名称、影院地址和联系电话等）。

(3) 放映厅管理的需求：影院工作人员在影院管理平台管理自己影院放映厅

的具体信息，对放映厅进行增删改查的操作，放映厅信息包括放映厅的具体座位数和放映厅名称。

（4）场次管理的需求：影院工作人员在影院管理平台根据具体的需求来对自身影院的放映场次进行操作，场次信息包括放映的电影、放映时间、具体放映厅、场次价格和场次描述等。

结合以上需求，影院端功能需求的用例图如图 3.2 示

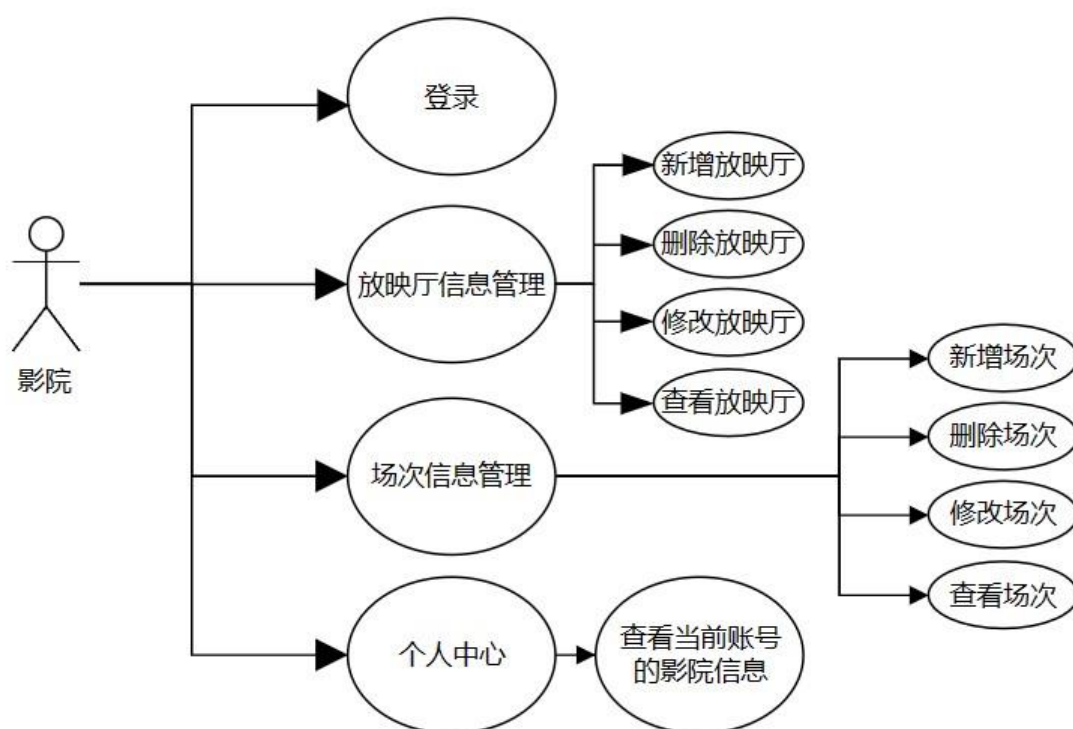


图 3.2 影院端功能需求用例图

3.2.3 管理端功能需求

最后，介绍管理端的功能需求。对于电影院订票 App 来说电影信息和影院信息是此系统中最为重要的信息，故管理端的功能需求提供给系统管理人员一个后台系统管理平台，让他们能够管理电影信息和影院信息。

（1）管理账号的登录需求：管理人员使用其专属的管理账号进行登录，登录到系统管理平台。

（2）电影管理的需求：管理人员通过平台发布电影信息，删除指定电影信息、修改电影信息和查看正在上映和待映的电影信息。

（3）影院管理的需求：管理人员通过平台为影院端新增（注册）新账号，并

对指定的影院账号进行操作。

结合以上需求，管理端功能需求的用例图如图 3.3 示

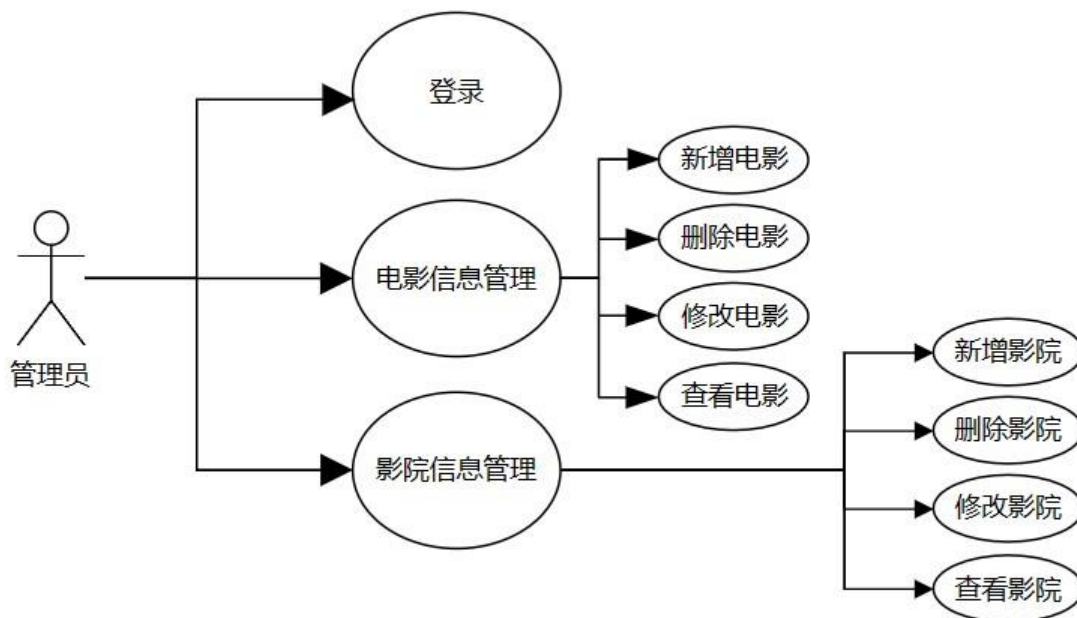


图 3.3 管理端功能需求用例图

3.3 非功能需求分析

系统的非功能需求分析主要着重考虑以下几个方面。

（1）性能需求

用户在浏览电影信息、选择座位等操作时，系统应保证快速响应，应控制在极短的时间内。此外，面对大量用户访问此系统的情况，也理应保证系统高并发的稳定性和处理能力。最重要的，系统应具备足够的容量和资源，能够应对节假日和热门电影上映等高峰期的访问和购票需求。

（2）可靠性需求

由于此系统会有大量用户访问，所以系统的可靠性十分重要，其应具备良好的稳定性，避免出现系统崩溃、故障和数据丢失等情况，保证用户购票的顺利进行。

（3）安全性需求

对于用户登录和操作时应进行有效的身份认证，防止非法用户访问和操作系统，且对系统功能和数据进行权限控制，不同角色的用户应有不同的操作权限，确保系统安全性和数据完整性，保证每个用户只能使用自己权限内的功能^[12]。系统通过要求用户在访问前验证其身份，对于此系统而言就是让用户输入账号、密码来登录这一操作，而在注册和登录这一过程中，应对用户的密码进行保护，防

止密码被盗窃或者被管理人员从数据库中查看到具体密码，本系统为防止密码明文存储将考虑使用 MD5 加密算法。除此之外，系统也应该考虑 SQL 注入等安全问题。

（4）易用性需求

系统界面设计应直观易用，操作符合用户的正常逻辑，减轻用户的繁杂理解，优化用户体验。提供用户指导和帮助功能，解答用户常见问题，提供使用说明和操作指南。

（5）可维护性需求

考虑到此系统需长期开发，代码应符合规范，易于维护和扩展，便于开发者开发和后续版本迭代。与此同时，系统也需具备监控和日志记录功能，及时发现和解决系统问题，保证系统稳定运行。

（6）兼容性需求

此系统作为面对大量用户的市场 App，应考虑到不同用户的具体需求。系统的实现应支持多种平台和设备访问，包括 PC 端、移动端和平板设备等，具备良好的跨平台兼容性。除了跨平台兼容外，也应在各常见浏览器上正常运行，提供完善的浏览器兼容性。

3.4 本章小结

本章主要对电影院订票 App 的需求进行了分析，包括系统角色分析、功能需求分析（用户端、影院端和管理端）和非功能需求分析。

4 系统设计

4.1 功能模块设计

系统功能模块划分是系统中各个功能模块的分类、规划和初步设计过程^[13]。本章将分别介绍用户端、影院端和管理端的功能模块设计，确保功能模块设计合理、清晰，有利于系统的开发、测试和维护工作。

4.1.1 用户端功能模块设计

用户端功能模块设计是电影院订票 App 中重要的部分，涵盖了用户账号管理、电影浏览、影院信息查看、选座购票、订单管理等多个方面。这些模块旨在提供用户友好的操作界面和便捷的功能流程，让用户可以轻松注册登录、浏览电影信息、查看影院详情、选择座位购票等操作。设计过程中需考虑用户体验、功能完整性和安全性，以提升用户满意度和竞争优势。

用户端的功能模块图如图 4.1 示

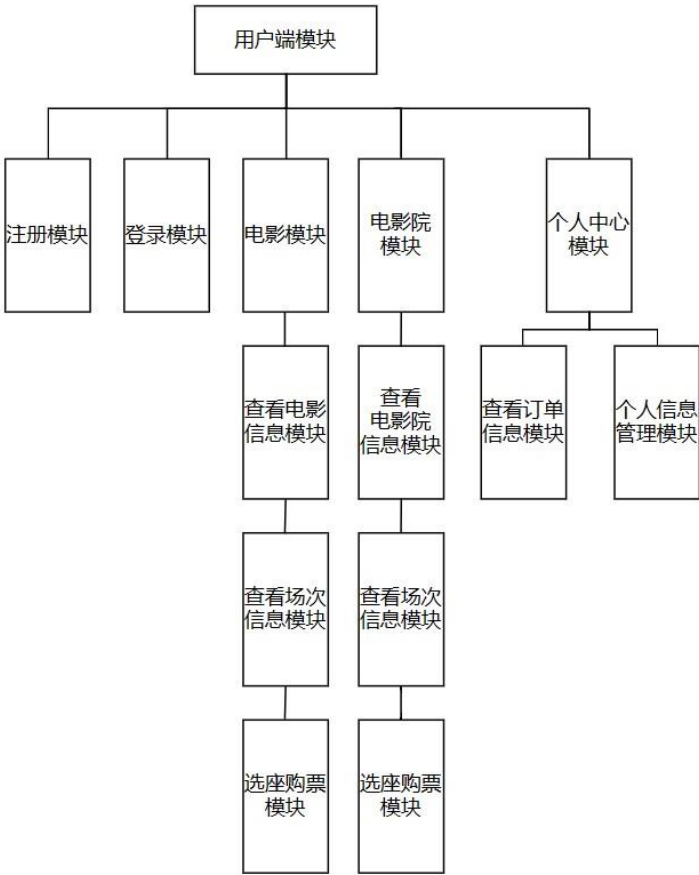


图 4.1 用户端功能模块图

4.1.2 影院端功能模块设计

影院端功能模块设计旨在为影院工作人员提供便捷高效的管理平台，包括账号登录与信息查看、放映厅管理、场次管理等功能。通过这些功能，影院工作人员可以方便地管理放映厅信息、场次安排，从而提升管理效率、优化用户体验，为影院的运营和发展提供全面的支持和服务。

影院端的功能模块图如图 4.2 示

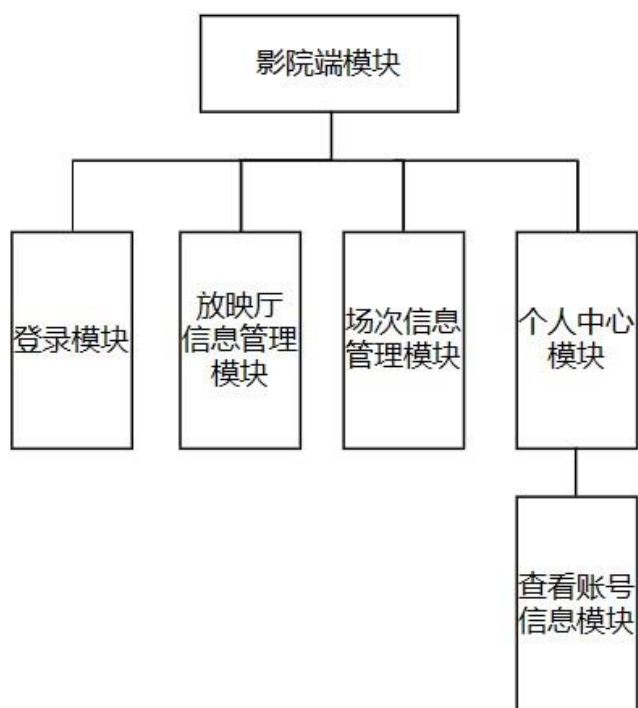


图 4.2 影院端功能模块图

4.1.3 管理端功能模块设计

此系统的管理端主要是为管理人员提供一个后台管理平台，主要是对电影信息进行管理和对影院端账号进行管理。

管理端的功能模块图如图 4.3 示

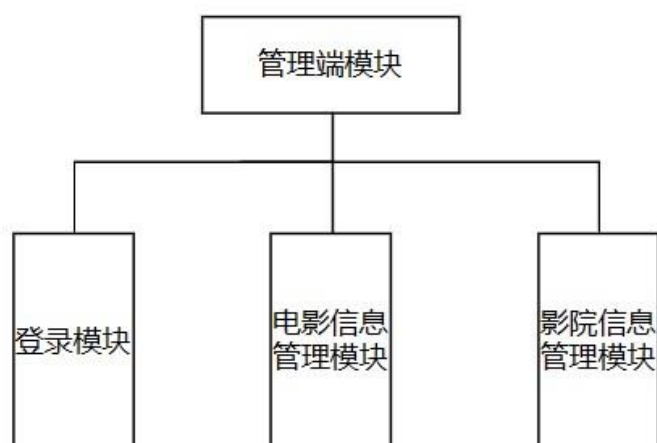


图 4.3 管理端功能模块图

4.2 架构设计

电影院订票 App 的架构设计通过了前端和后端独立开发的模式，前端通过 Uni-App 框架实现跨平台开发；而后端则好好地利用 Spring Boot 框架处理各功能逻辑并结合 MyBatis 框架简化数据库操作从而来存储数据。数据库方面选择了 MySQL 存储用户、电影和订单等信息，并引入 Redis 作为缓存提高系统性能。接口设计遵循 RESTful API 原则且前后端数据交换格式为 JSON，确保前后端数据交互顺畅。这样的架构设计能够提升系统的模块化、灵活性和可维护性，以满足用户需求并支持系统持续发展。

系统的架构图如图 4.4 示

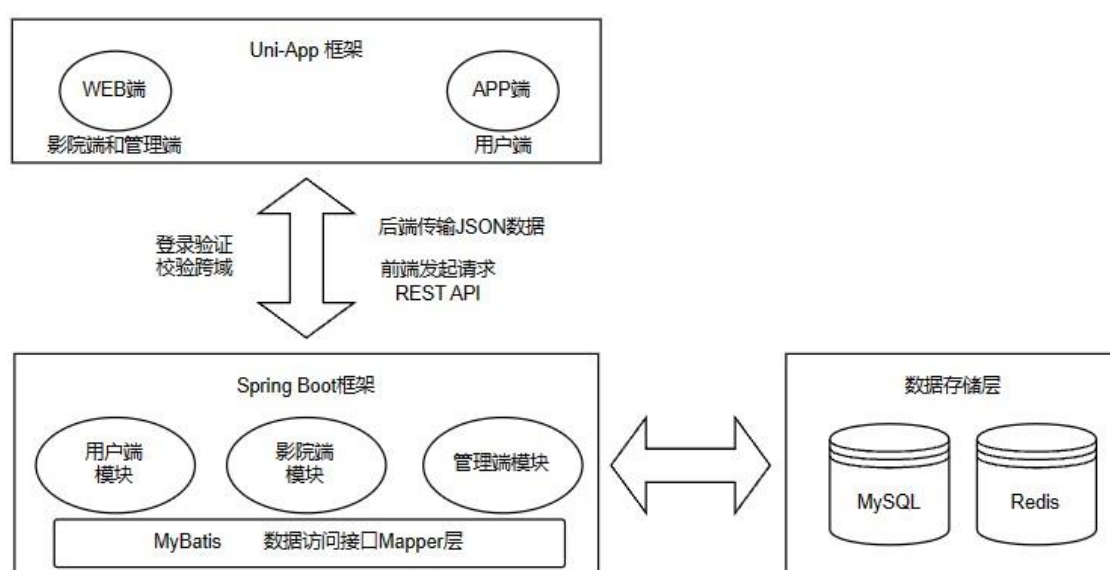


图 4.4 系统架构图

4.3 数据库设计

基于前文介绍的需求分析、功能模块设计和架构设计，后文将对此针对地设计数据库。数据库的设计，是将各角色的模糊需求通过一个统一的模型表示出来，这样的模型作为一种媒介，作为现实实体与计算机中的对象描述，最为紧要地来说明系统中各对象的各种具体属性和操作^[14]。本节将通过 E-R 图和表作为展示，描述系统各数据的结构联系，并在下文对表和其中重要数据进行说明。

系统全局的 E-R 图如图 4.5 示

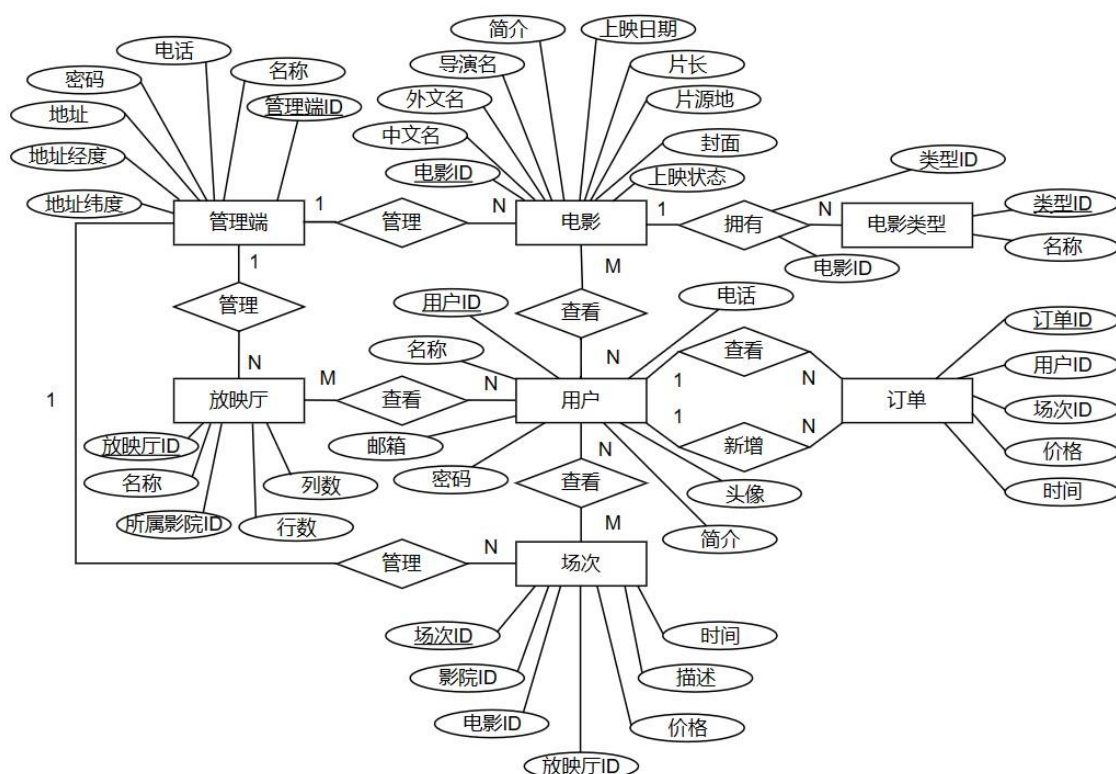


图 4.5 系统全局 E-R 图

电影院订票 App 的主要实体如上 E-R 图所示，主要有用户、管理端（管理端和影院端都是同一种账号类型）、电影、电影类型、放映厅、场次和订单。

（1）用户表数据库设计：

通过上述 E-R 图可确定数据库表的设计。在用户表中，有 7 个属性，分别是用户 ID（作为主键）、名称、邮箱（相当于账号，用户需用注册的邮箱进行登录）、MD5 加密后的密码、电话、头像（数据库中存放的是头像图片的 URL）和简介。

用户表如下表 4.1 示

表 4.1 用户表 user

字段	数据类型	数据长度	约束	说明
user_id	INT	10	主键	用户 ID
user_name	VARCHAR	255	唯一	用户名称
user_email	VARCHAR	255	非空	用户邮箱
user_password	VARCHAR	255	非空	用户密码
user_phone	VARCHAR	255	唯一	用户电话
user_url	VARCHAR	255	唯一	用户头像图片 URL
user_profile	VARCHAR	255	唯一	用户简介

（2）影院表数据库设计：

此管理端实体是实际系统中影院端和管理端共同使用的同一账号类型，系统通过辨识具体账号提供不同的功能模块。此表中，有 7 个属性，分别是影院 ID（作为主键）、名称、电话（相当于账号，后台管理人员或者影院管理人员需用专属的电话进行登录）、MD5 加密后的密码、地址、地址经度和地址纬度。

影院表如下表 4.2 示

表 4.2 影院表 cinema

字段	数据类型	数据长度	约束	说明
cinema_id	INT	10	主键	影院 ID
cinema_name	VARCHAR	255	非空	影院名称
cinema_phone	VARCHAR	255	非空	影院电话
cinema_password	VARCHAR	255	非空	影院密码
cinema_address	VARCHAR	255	非空	影院地址
cinema_x	VARCHAR	255	非空	影院地址经度
cinema_y	VARCHAR	255	非空	影院地址纬度

（3）电影表数据库设计：

在电影表中，有 10 个属性，分别是电影 ID（作为主键）、中文名、外文名，导演名、简介、上映日期、片长、片源地、封面和上映状态。

电影表如下表 4.3 示

表 4.3 电影表 movie

字段	数据类型	数据长度	约束	说明
movie_id	INT	10	主键	电影 ID
movie_name_cn	VARCHAR	255	非空	电影中文名
movie_name_en	VARCHAR	255	非空	电影外文名
movie_director	VARCHAR	255	非空	电影导演名
movie_description	VARCHAR	255	非空	电影简介
movie_release_date	DATETIME	20	非空	电影上映日期
movie_duration	VARCHAR	255	非空	电影片长
movie_country	VARCHAR	255	非空	电影片源地
movie_url	VARCHAR	255	非空	电影封面图片 URL
movie_state	VARCHAR	255	非空	电影上映状态

（4）电影类型表数据库设计：

在电影类型表中，有 2 个属性，分别是类型 ID（作为主键）和名称。除此之外，在此系统中，由于电影和电影类型是一对多的关系，所以此系统将通过连接表的方式绑定各自电影的类型信息，故下文将分别介绍电影类型表和电影类型连接表。

电影类型表如下表 4.4 示

表 4.4 电影类型表 genres

字段	数据类型	数据长度	约束	说明
genre_id	INT	10	主键	电影类型 ID
genre_name	VARCHAR	255	非空	电影类型名称

电影类型连接表如下表 4.5 示

表 4.5 电影类型连接表 movie_genres

字段	数据类型	数据长度	约束	说明
movie_id	INT	10	非空	电影 ID
genre_id	INT	10	非空	电影类型 ID

（5）放映厅表数据库设计：

在放映厅表中，有 5 个属性，分别是放映厅 ID（作为主键）、名称、所属影院 ID、放映厅座位具体行数、放映厅座位具体列数。

放映厅表如下表 4.6 示

表 4.6 放映厅表 house

字段	数据类型	数据长度	约束	说明
house_id	INT	10	主键	放映厅 ID
house_name	VARCHAR	255	非空	放映厅名称
cinema_id	INT	10	非空	所属影院 ID
house_x	VARCHAR	255	非空	行数
house_y	VARCHAR	255	非空	列数

（6）场次表数据库设计：

在场次表中，有 7 个属性，分别是场次 ID（作为主键）、影院 ID、电影 ID、放映厅 ID、场次价格、场次描述和场次开始时间。

场次表如下表 4.7 示

表 4.7 场次表 schedule

字段	数据类型	数据长度	约束	说明
schedule_id	INT	10	主键	场次 ID
movie_id	INT	10	非空	电影 ID
cinema_id	INT	10	非空	影院 ID
house_id	INT	10	非空	放映厅 ID
schedule_price	VARCHAR	255	非空	场次价格
schedule_description	VARCHAR	255	非空	场次描述
schedule_time	DATETIME	20	非空	场次时间

（7）订单表数据库设计：

在订单表中，有 5 个属性，分别是订单 ID（作为主键）、用户 ID、场次 ID、订单价格（总费用）和订单生成时间。

订单表如下表 4.8 示

表 4.8 订单表 orders

字段	数据类型	数据长度	约束	说明
orders_id	INT	10	主键	订单 ID
user_id	INT	10	非空	用户 ID
schedule_id	INT	10	非空	场次 ID
orders_price	VARCHAR	255	非空	订单费用
orders_time	DATETIME	20	非空	订单时间

4.4 本章小结

本章的系统设计中，对电影院订票 App 进行了功能模块设计、架构设计和数据库设计。通过功能模块设计，明确了用户端、影院端和管理端的各项功能需求，确保系统能够满足不同角色的操作需求。架构设计使用的是前端和后端独立开发的形式，利用了 Spring Boot 框架解决系统的各功能需求，Uni-App 框架实现前端跨平台开发，并结合了 MySQL 和 Redis 数据库，以及 RESTful API 设计，确保系统具备高效、稳定的性能和良好的用户体验。通过这些设计，为电影院订票 App 的开发奠定了坚实的基础，使其能够实现功能丰富、操作便捷的在线订票服务。

5 功能实现

5.1 系统开发环境

系统开发环境对于本系统的开发至关重要。对于电影订票 App 来说，此系统将采用 IntelliJ IDEA 作为集成开发环境（IDE），结合 Java 作为主要编程语言，同时配合 Vue.js、JavaScript 等前端技术进行系统实现。并且本系统采用 MySQL 作为主要的联系型数据库管理模型，用于存储系统所需的各种数据。除了基本的开发工具和数据库系统外，此系统还利用 Spring Boot 框架、MyBatis 框架和 Uni-App 框架，以提高开发效率和系统性能。以上的开发环境和工具选择确保了系统开发的功能实现的顺利进行。

5.2 用户端功能实现

本节将对用户端模块的实现进行介绍，并且通过具体的接口信息、时序图和界面实现图进行说明。

5.2.1 注册模块实现

该模块主要实现用户端用户的注册操作。

此模块涉及的接口如下表 5.1 示

表 5.1 注册模块接口表

接口	请求方式	请求参数	接口说明
/user/sendEmail	POST	User user	给相应用户的邮箱发送验证码
/user/register	POST	User user String code	用户注册账号

在给用户的邮箱发送验证码这一接口功能中，此系统使用 QQ 邮箱提供的 SMTP 服务，SMTP（Simple Mail Transfer Protocol）协议用于支持使用电子邮件客户端发送电子邮件^[15]，本系统中只需在 Spring Boot 中的配置文件（application.yml）中填写相应的 SMTP 设置和授权码即可使用此服务。

对于用户注册账号的接口功能，由与本系统不希望以明文的方式存储密码，故采用了 MD5 加密技术，将 MD5 工具类设置在 Spring Boot 项目中的工具类文件夹（utils）中，方便后续使用。

此模块的时序图如下图 5.1 示

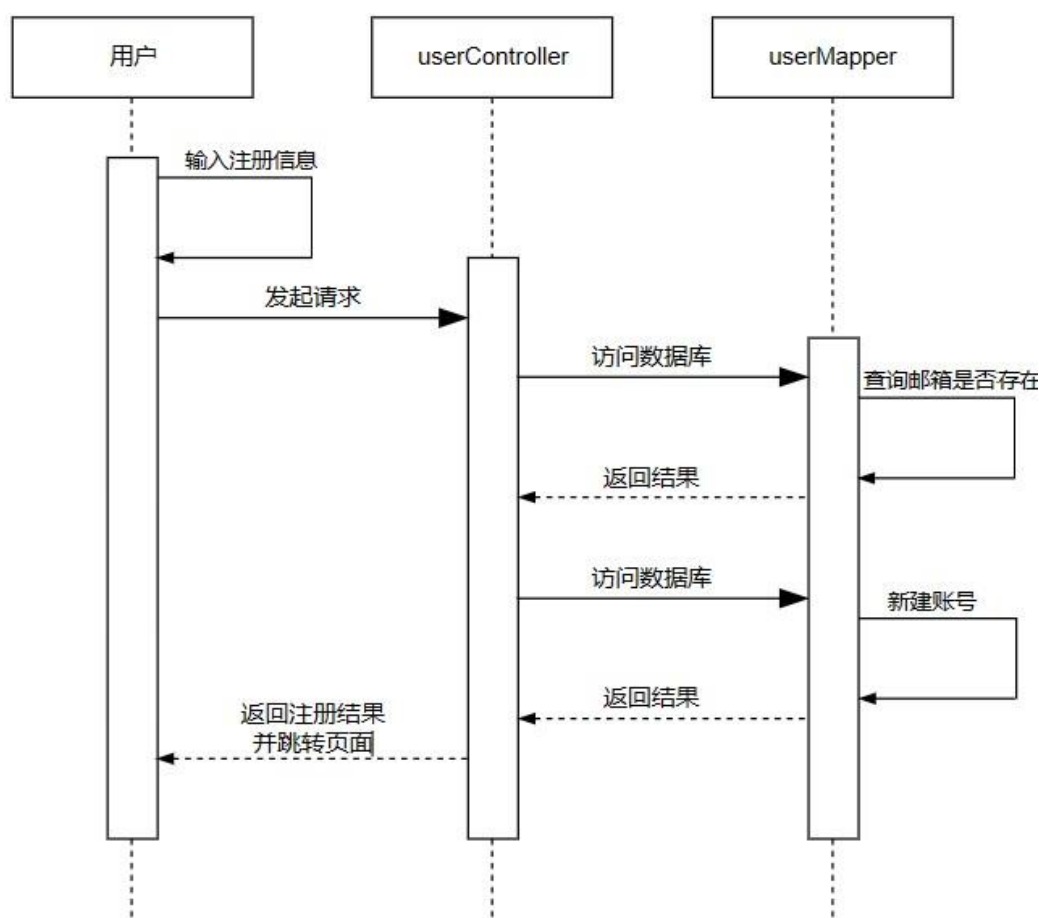


图 5.1 用户端注册模块时序图

在用户使用此系统时，用户需通过点击底部导航栏的“我的”进入登录注册界面，并选择“注册”渲染出注册界面。在此界面中需在输入框中填写具体账户的邮箱和密码，在填写邮箱后可以获取验证码，服务器将发送对应的验证码，用户在查看自己的邮箱后输入对应验证码，最后点击注册按钮注册账号并跳转到已登录界面。

此模块的界面实现图如下图 5.2 示

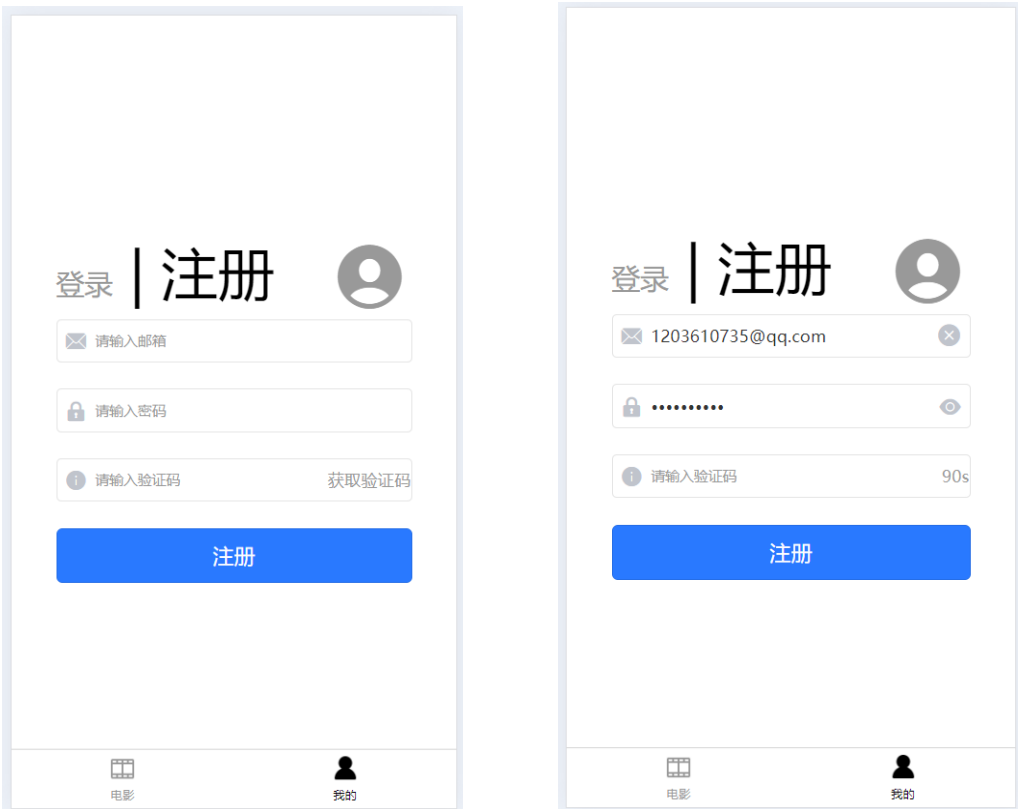


图 5.2 用户端注册模块界面实现图

5. 2. 2 登录模块实现

该模块主要实现用户端用户的登录操作。用户在使用此系统时，需通过已注册的账号和相对应的密码来进行账号的登录。

此模块涉及的接口如下表 5.2 示

表 5.2 登录模块接口表			
接口	请求方式	请求参数	接口说明
/user/login	POST	User user	用户登录账号

对于用户账号的登录校验的处理，本系统将使用 JWT 的技术。具体的实现就是，将生成的 token 存储在 Redis^[16]。在用户登录时，生成相对应的 JWT token 与该用户 ID 绑定并缓存到 Redis 中。每次用户发起请求时，服务器都相应的查询 Redie 中的 token 并判断其有效期，若有效则提供访问具体内容，若失效则让用户重新登录。

此模块的时序图如下图 5.3 示

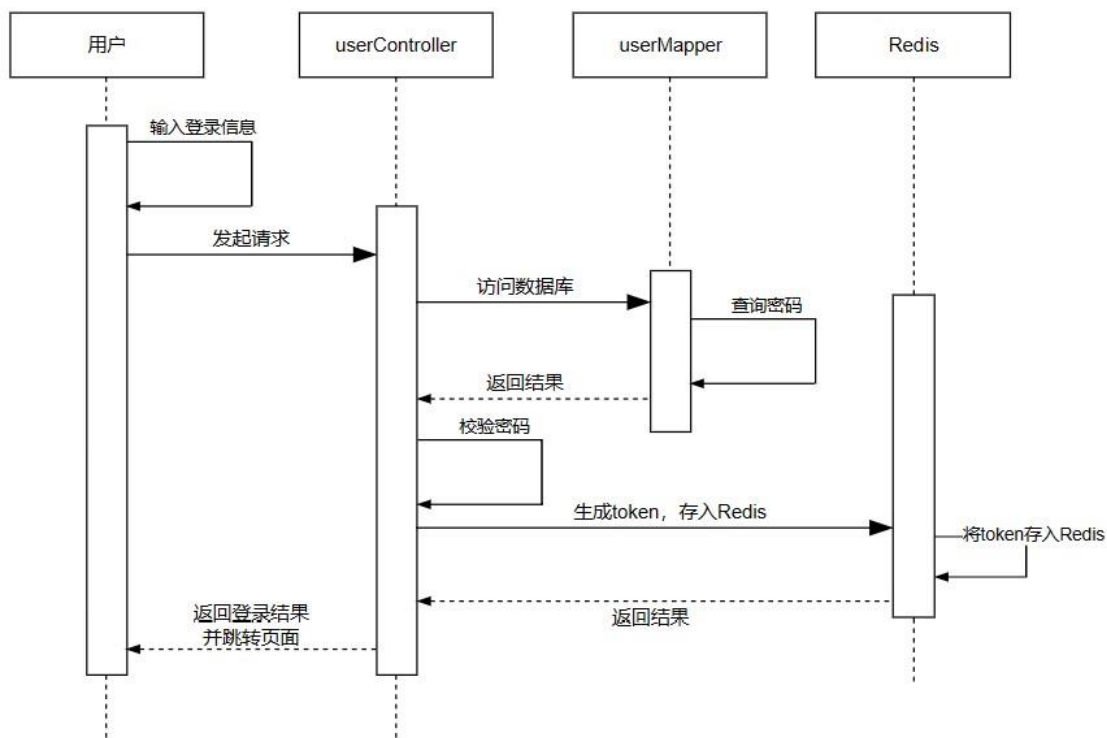


图 5.3 用户端登录模块时序图

在用户使用此系统时，用户需通过点击底部导航栏的“我的”进入登录注册界面，并点击“登录”渲染出登录界面。用户使用系统时需在输入框中填写自身用户账号的邮箱和相应密码，最后点击登录按钮进行登录并跳转界面。

此模块的界面实现图如下图 5.4 示



图 5.4 用户端登录模块界面实现图

5.2.3 个人中心模块实现

该模块主要包括查看订单信息模块和个人信息管理模块。用户在查看订单信息模块中能查看到已购票的订单信息，而在个人信息管理模块中用户能查看到账号相关信息（名称、头像、电话和简介）并进行修改的操作。

此模块涉及的接口如下表 5.3 示

表 5.3 个人中心模块接口表			
接口	请求方式	请求参数	接口说明
/orders/infoByUserId	GET	Long userId	用户查看订单列表
/user/info	GET	无	用户查看用户信息
/user/update	PUT	User user	修改用户信息
/user/updateUrl	PATCH	User user	修改用户头像 图片 URL
/upload	POST	MultipartFile file	上传文件

在上传文件的接口中，调用了阿里云的 OSS 服务。阿里云对象存储 OSS(Object Storage Service)是一款海量、安全、低成本、高可靠的云存储服务，可提供 99.999999999%（12 个 9）的数据持久性，99.995%的数据可用性，多种存储类型供选择，全面优化存储成本^[16]。系统通过此服务静态存储用户上传的头像图片并返回 URL，实现用户头像图片的绑定和展示。

用户在登录或注册后跳转到新的“我的”页面，在此页面中展示个人信息、购票记录和登出的选项，用户可以根据自身需要选择不同选项。

已登录的“我的”界面实现图如下图 5.5 示

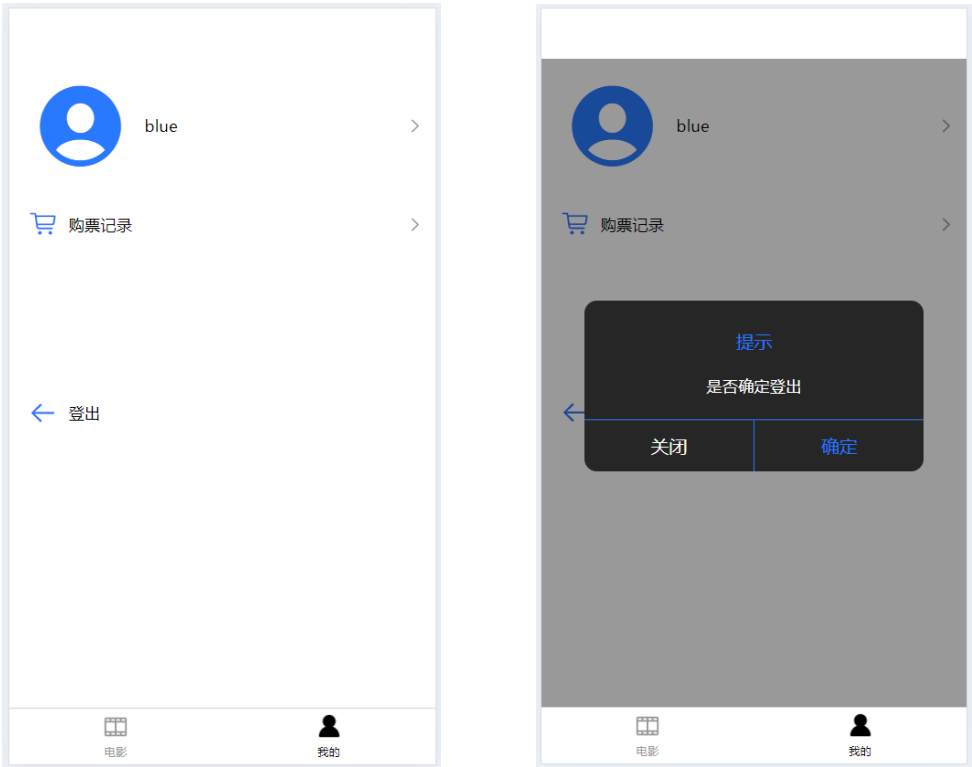


图 5.5 登录后的界面实现图

在此界面中用户点击最上方的个人信息选项将跳转到个人信息的界面。
个人信息界面实现图如下图 5.6 示



图 5.6 个人信息的界面实现图

在此界面中用户点击头像即可以更换头像，假若用户想要修改名字、电话或者简介，只要在下方面击相应的选项进入修改界面即可以进行修改。

个人信息的修改界面实现图如下图 5.7 示



图 5.7 个人信息修改的界面实现图

在“我的”界面中点击购票记录将跳转到购票记录的界面，用户可以查看自己之前购买过的电影票的相关订单的信息，有无购票记录将显示出不同的界面。

用户购票记录的界面实现图如下图 5.8 示



图 5.8 购票记录的界面实现图

5. 2. 4 电影模块实现

该模块主要实现用户查看当前正在上映和待映的电影信息列表，用户可以在此模块中浏览到电影的大致信息（中文名、外文名、导演名、上映日期、片长和封面等）。

此模块涉及的接口如下表 5.4 示

表 5.4 电影模块接口表			
接口	请求方式	请求参数	接口说明
/movie/info	GET	无	查询所有电影信息
/movie/infoMovieById	GET	Long movieId	根据 movieId 查询指定电影信息
/movie/infoGenresByMovie	GET	Long movieId	根据 movieId 查询指定电影的类型
/movie/infoMovieByState	GET	String movieState	根据上映状态查询电影信息

在此模块中由于电影有多种属性，且用户对于查询电影的操作有不同的需求，故在此模块中有各种对电影不同查询的请求。当用户浏览此系统时，“电影”作为此系统的主页将为用户提供初步的电影信息，在用户选择上方“电影”这一选项时界面将渲染出电影信息列表，同时用户可以选择“正在上映”或“待映”这两个选项，来查询不同上映状态的电影信息列表。

“正在上映”和“待映”的电影信息列表的界面实现图如下图 5.9 示



图 5.9 电影信息列表的界面实现图

在此界面中，用户可以查看各个电影的大致信息，在“正在上映”的电影信息列表中，其列表根据具体的上映日期从最近的日期往前排序；而在“待映”的电影信息列表中，其列表根据具体的上映日期从最近的日期往后排序，符合用户的查看需求。最后，用户根据自己的喜好，点击列表右边的购票按钮，进入查看指定的电影信息的界面，并在该页面点击封面处跳转至电影详细信息的界面。

指定的电影信息的界面实现图如下图 5.10 示



图 5.10 指定电影信息的界面实现图

5.2.5 电影院模块实现

该模块主要实现用户查看当前附近的电影院信息列表，用户可以在此模块中浏览到电影院的大致信息（电影院名称、电影院地址和定位距离）。

此模块涉及的接口如下表 5.5 示

表 5.5 电影院模块接口表			
接口	请求方式	请求参数	接口说明
/cinema/infoAllCinema	GET	无	查询所有电影院信息
/cinema/infoById	GET	Long cinemaId	根据 cinemaId 查询指定的电影院信息

当用户浏览此系统时，在下方导航栏选择“电影”，同时在上方选项中选择“影院”即可以渲染出当前影院信息列表（影院信息列表的排序根据当前用户的定位，从近到远进行排序），用户可以根据自身具体的需要选择喜好的影院，点击进入后将根据指定的影院选择其影院的场次。

影院信息列表的界面实现图如下图 5.11 示



图 5.11 影院信息列表的界面实现图

5.2.6 场次模块实现

该模块主要实现用户查看当前不同查询请求的场次信息列表，用户可以在此模块中查看到场次的详细信息（场次所属影院信息、场次播放的电影信息、场次放映厅的信息、场次价格、场次描述和场次的播放时间）。

此模块涉及的接口如下表 5.6 示

表 5.6 场次模块接口表

接口	请求方式	请求参数	接口说明
/schedule/infoById	GET	Long scheduleId	查询指定场次的信息
/schedule/infoByMovie	GET	Long movieId	根据 movieId 查询场次列表
/schedule/infoByCinema	GET	Long cinemaId	根据 cinemaId 查询场次列表
/schedule/infoByMovieAndCinema	GET	Long movieId Long cinemaId	根据 movieId 和 cinemaId 查询场次列表

用户在 5.2.4 小节的图 5.10 的界面中点击下方的电影场次按钮即可跳转至指定电影的场次选择（根据影院）界面。在该界面中用户可以看到该电影在哪些影院有场次，并且可以选择具体的日期来查看相对应的影院信息列表。

场次选择（根据影院）的界面实现图如下图 5.12 示



图 5.12 场次选择（根据影院）的界面实现图

在该界面中，用户可以再根据自己具体的需求选择影院，最后跳转到根据指定电影和指定影院展示的场次信息列表的界面，此界面后文会提到。同上文提到的一样，用户在 5.2.4 小节的图 5.11 的界面中选择喜欢的影院后，界面将跳转到指定影院的场次选择（根据电影）界面。在该界面中用户可以看到该影院中有哪些电影有场次，并根据用户具体日期要求选择不同日期和不同电影的场次。

场次选择（根据电影）的界面实现图如下图 5.13 示

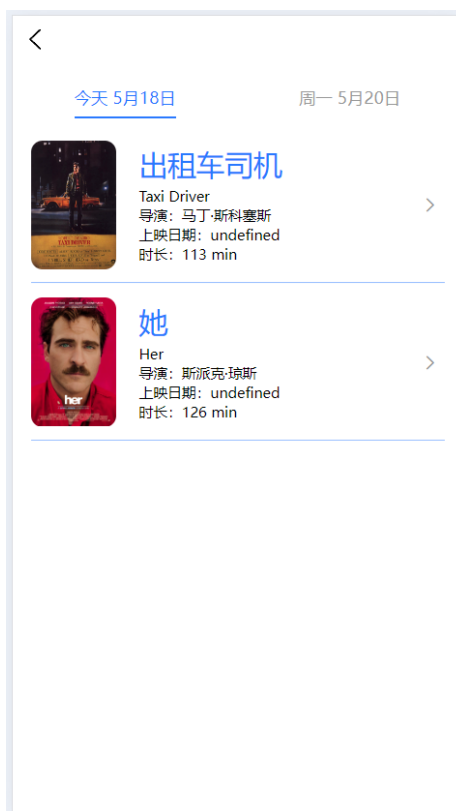


图 5.13 场次选择（根据电影）的界面实现图

用户在此界面中，可以再选择想要观看的指定电影并选择，其后跳转到上文所提到的根据指定电影和指定影院的场次信息列表的界面。在该界面中，用户可以看到该场次所属影院的相关信息和该场次播放电影的相关信息。

场次选择（根据电影和影院）的界面实现图如下图 5.14 示



图 5.14 场次选择（根据电影和影院）的界面实现图

可以看到该界面中，所属影院播放的电影的场次根据不同的场次日期来展示不同的场次信息列表，同时对于场次列表的每条信息，用户可以看到大致的场次信息（场次开始和结束时间、场次描述、场次的放映厅名称和场次的单人票价价格）。用户通过点击列表右边的购票按钮，系统将跳转至选座购票的界面；而如果用户点击上方影院的地图图标，系统将跳转到影院位置地图的界面。

影院位置地图的界面实现图如下图 5.15 示



图 5.15 影院位置地图的界面实现图

5. 2. 7 选座购票模块实现

该模块是此系统中用户端最为重要的部分，其主要实现用户查看当前场次的放映厅的座位信息、选择座位并购票等操作。

此模块涉及的接口如下表 5.7 示

表 5.7 选座购票模块接口表			
接口	请求方式	请求参数	接口说明
/house/infoById	GET	Long housed	查询场次 放映厅信息
/orders/create	POST	Orders orders	用户在购票后 创建订单
/orders/infoSeatFlag	GET	String ordersSeat Long scheduleId	查询该场次哪些 座位已经被购买

该界面作为用户端中最重要的选座购票模块的实现，应该给用户最为直观地看到场次的放映厅的具体座位信息和最为方便地进行选座购票的操作。

选座购票的界面实现图如下图 5.16 示



图 5.16 选座购票的界面实现图

可以看到在此界面中，用户可以浏览自己所选场次的放映厅的大部分信息。在最上方可以看到放映厅的名称和银幕具体位置。而在放映厅的座位展示上，用户可以通过放大或缩小查看座位的展示图，同时，可以看到的是座位被分为了三种状态分别是可选、已售和选中，其对应不同的座位图标颜色，这样有助于用户了解到放映厅中各座位的具体状态。除此之外，在界面的下半部分，用户可以看到不同座位图标的说明和具体选中座位的行数和列数，方便用户选择自己心仪的位置。最后，购票所需的总票价将显示在确认选座的按钮上，用户点击确认选座的按钮后将进行购票并生成订单，购票操作完成后用户将跳转回上一个界面。

5.3 影院端功能实现

本节将对影院端模块的实现进行介绍，并且通过具体的接口信息、时序图和界面实现图进行说明。

5.3.1 登录模块实现

由于在此系统中影院端和管理端都是使用同一种账号类型进行登录，故在此小节中介绍的登录模块实现皆为影院端和管理端的登录实现，管理端的登录模块实现在后文将不再重复提及。

该模块主要实现影院端和管理端的登录操作。不管是影院管理人员还是系统管理人员都需通过专属的账号（在此账号类型中影院端账号为影院端的电话，管理端则为自己提供的账号）来进行登录。

此模块涉及的接口如下表 5.8 示

表 5.8 登录模块接口表

接口	请求方式	请求参数	接口说明
/cinema/login	POST	Cinema cinema	影院端、管理端 登录账号

和用户端账号登录操作类似的，影院端和管理端的登录也是生成相对应的 JWT token 与该影院 ID 绑定并缓存到 Redis 中。

此模块的时序图如下图 5.17 示

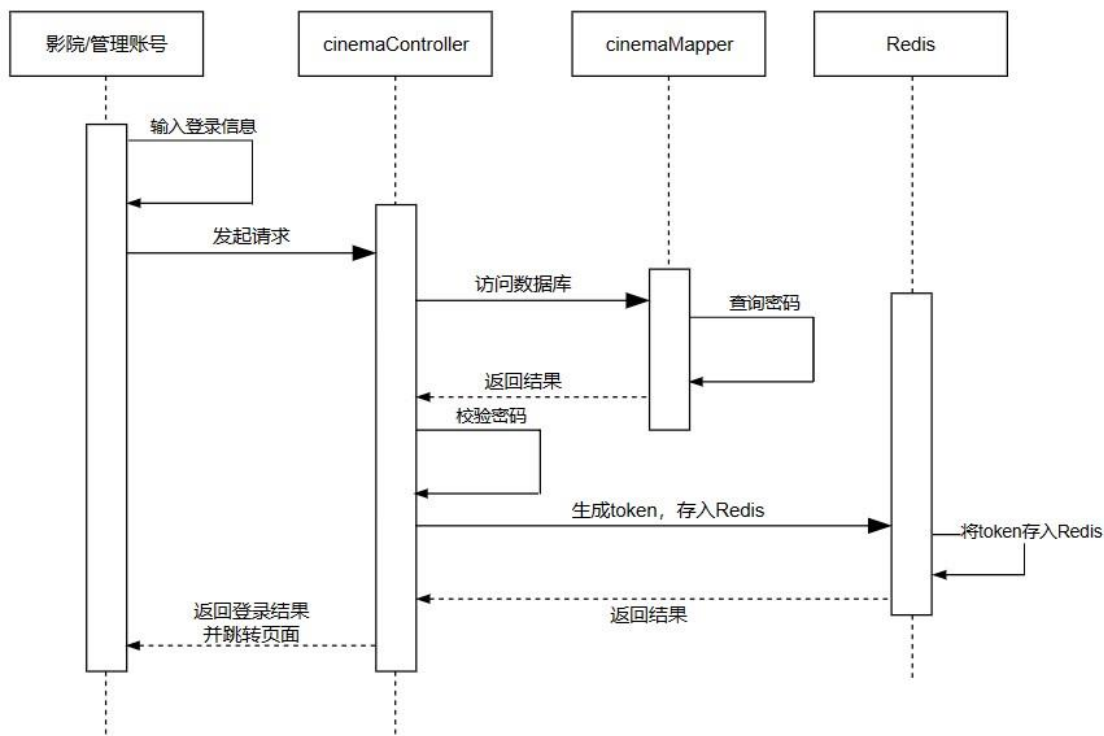


图 5.17 登录模块的时序图

对于影院端和管理端的各种管理操作都是在 WEB 端上进行，所以相关界面也应该根据 WEB 端的界面尺寸来进行设计。

影院和管理账号登录的界面实现图如下图 5.18 示



图 5.18 影院和管理账号登录的界面实现图

5. 3. 2 账号信息模块实现

该模块主要实现影院管理人员查看当前账号的影院信息（影院名称、影院地址、电话和具体位置地图）。

此模块涉及的接口如下表 5.9 示

表 5.9 账号信息模块接口表

接口	请求方式	请求参数	接口说明
/cinema/info	GET	无	查询当前账号的 电影院信息

影院端管理人员在登录后即跳转到该界面，该界面作为影院管理系统的主页，其在左边给影院端管理人员提供了导航栏分别是影院信息、放映厅信息管理、场次信息管理和登出；而在右边则展示出当前账号的影院信息。

账号信息模块的界面实现图如下图 5.19 示

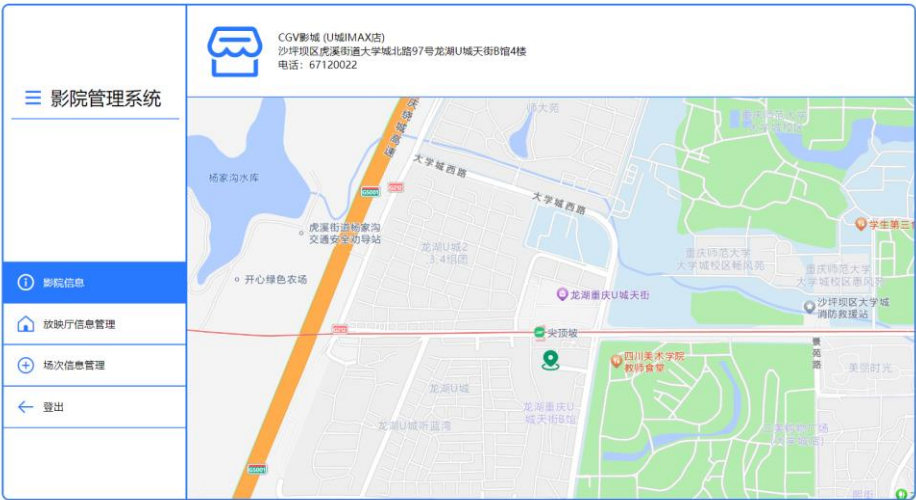


图 5.19 账号信息模块的界面实现图

5.3.3 放映厅信息管理模块实现

该模块主要实现影院管理人员对当前影院的放映厅进行管理操作。

此模块涉及的接口如下表 5.10 示

表 5.10 放映厅信息管理模块接口表

接口	请求方式	请求参数	接口说明
/house/create	POST	House house	新增放映厅
/house/infoByCinemaId	GET	Long cinemaId	查询当前影院账号的放映厅列表
/house/infoById	GET	Long houseId	查询指定放映厅信息
/house/delete	DELETE	Long houseId	删除指定放映厅
/house/update	PUT	House house	修改指定放映厅信息

此界面给影院端的管理人员展示出所有该影院的放映厅信息，同时，管理人员可以根据具体需求点击修改、删除或者新增放映厅的按钮对放映厅信息进行管理操作。

放映厅信息管理的界面实现图如下图 5.20 示



图 5.20 放映厅信息管理的界面实现图

影院端管理人员点击修改和新增放映厅按钮将跳转至新的界面；点击删除则是弹出确认框确认管理人员是否确认删除该放映厅信息，从而对放映厅信息进行管理。

修改和新增放映厅的界面实现图如下图 5.21 示

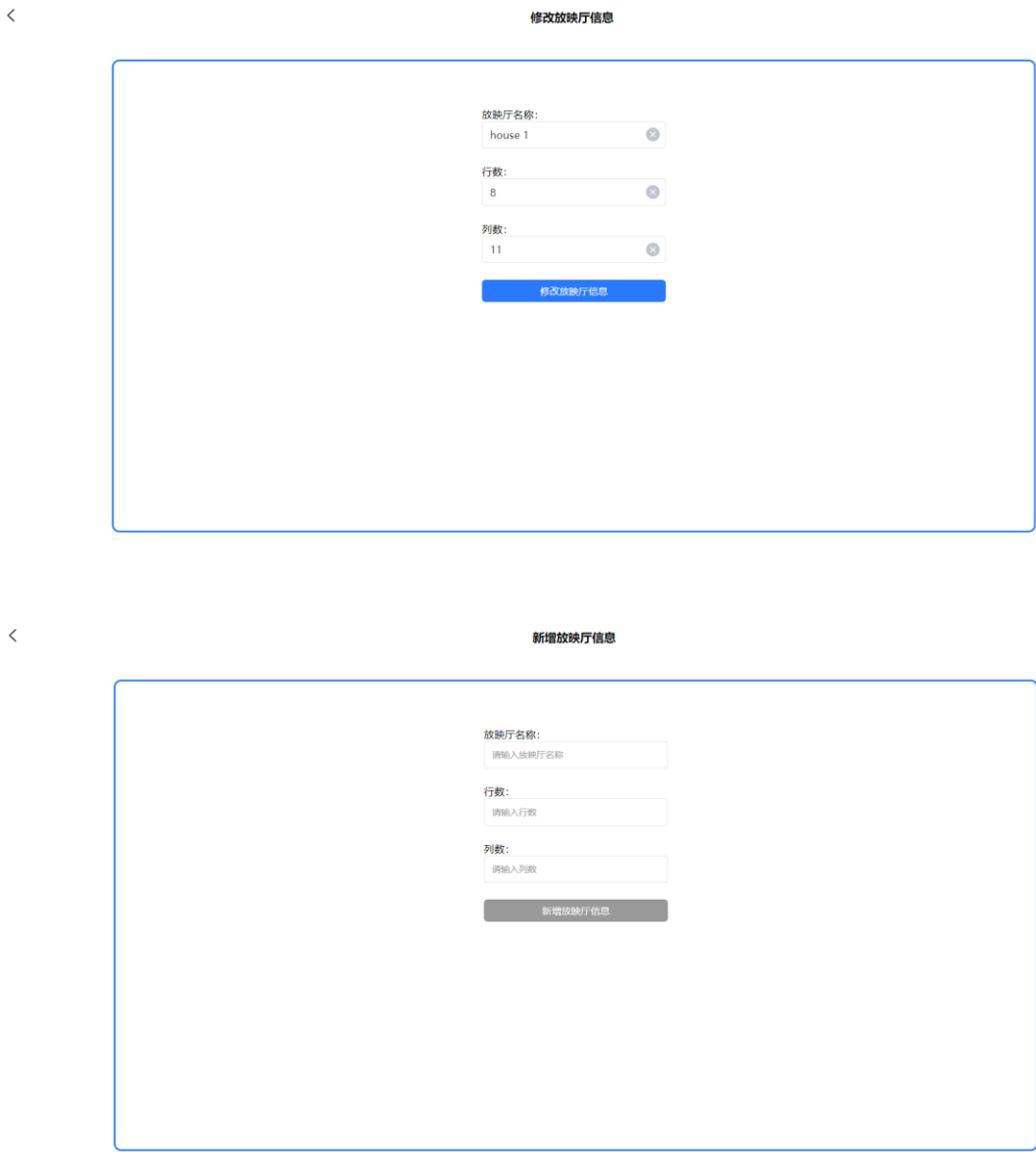


图 5.21 修改和新增放映厅的界面实现图

5.3.4 场次信息管理模块实现

该模块主要实现影院管理人员对当前影院的场次进行管理操作。
此模块涉及的接口如下表 5.11 示

表 5.11 场次信息模块接口表

接口	请求方式	请求参数	接口说明
/schedule/create	POST	Schedule schedule	新增场次
/schedule/infoById	GET	Long scheduleId	查询指定场次的信息
/schedule/infoByCinema	GET	Long scheduleId	查询当前账号的场次信息列表
/schedule/delete	DELETE	Long scheduleId	删除指定场次
/schedule/update	PUT	Schedule schedule	修改指定场次

该界面提供给影院端管理人员所有该影院的场次信息，并且，管理人员可以根据需求点击“查看”查看场次的详细信息、点击“修改”修改该场次的信息、点击“删除”删除该场次信息和点击“新增场次”为该影院新增场次。

场次信息管理的界面实现图如下图 5.22 示



图 5.22 场次信息管理的界面实现图

同放映厅信息管理类似的，影院端管理人员可以点击查看、修改和新增放映厅的按钮跳转到新的界面对场次信息进行操作，而点击删除按钮则弹出是否确认删除该场次的对话框。

查看、修改和新增场次的界面实现图如下图 5.23 示

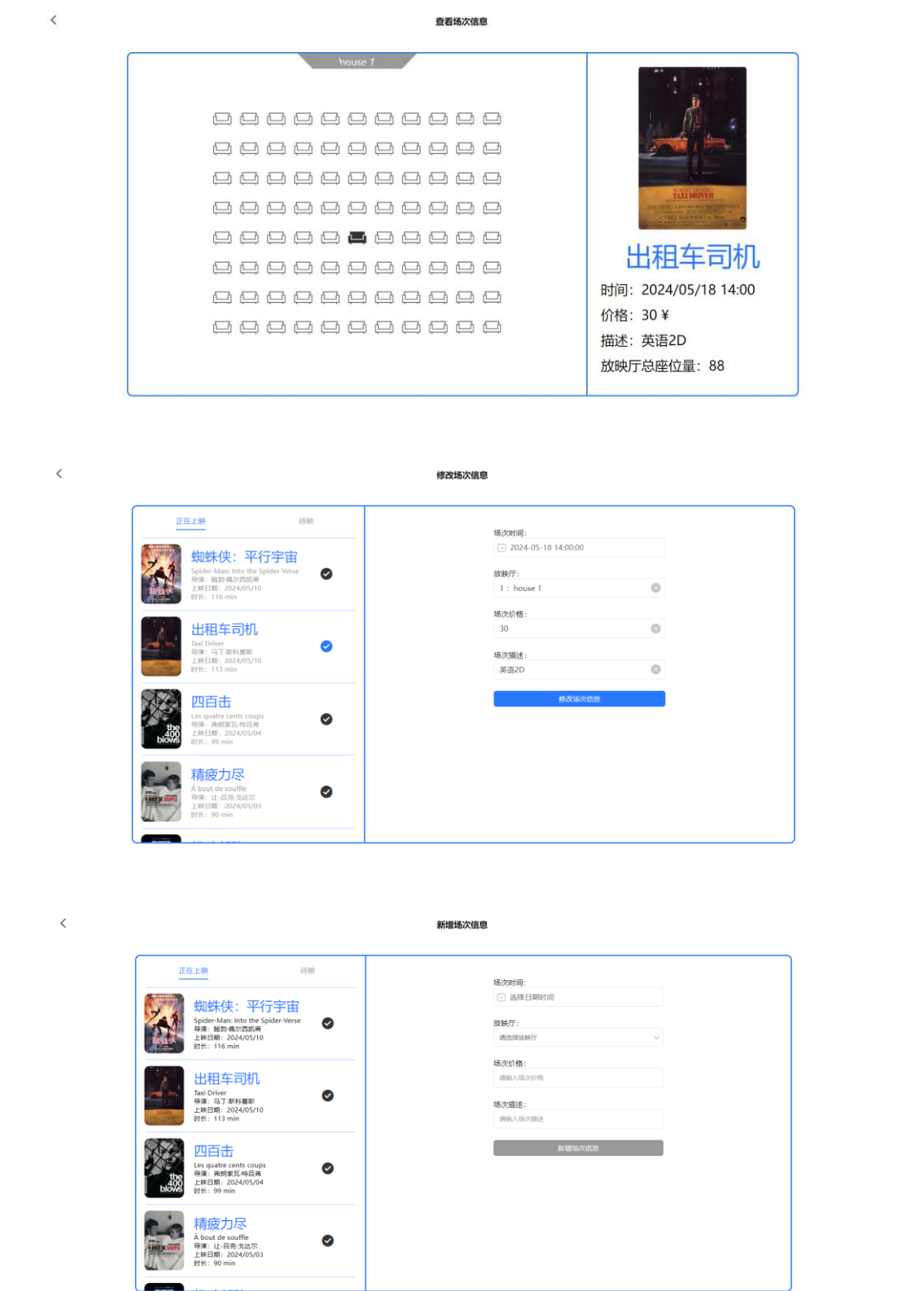


图 5.23 查看、修改和新增场次的界面实现图

5.4 管理端功能实现

本节将对管理端模块的实现进行介绍，并且通过具体的接口信息、时序图和

界面实现图进行说明。

5.4.1 登录模块实现

由于在此系统中影院端和管理端都是使用同一种账号类型进行登录，而在上文中已详细说明过该账号类型的登录模块的实现过程，故在此不重复说明，详情见 5.3.1 小节。

5.4.2 电影信息管理模块实现

该模块主要实现系统管理人员对当前电影信息进行管理操作。

此模块涉及的接口如下表 5.12 示

表 5.12 电影信息管理模块接口表			
接口	请求方式	请求参数	接口说明
/movie/create	POST	Movie movie Long[]movieType	新增电影信息
/movie/info	GET	无	查询所有的 电影信息
/movie/infoMovieById	GET	Long movieId	查询指定的 电影信息
/movie/delete	DELETE	Long movieId	删除指定的 电影信息
/movie/update	PUT	Movie movie Long[]movieType	修改指定的 电影信息

此界面给系统管理人员展示出所有的电影信息，并且让管理人员可以对电影信息进行操作。

电影信息管理的界面实现图如下图 5.24 示

<div>后台管理系统</div> <div>电影信息管理</div> <div>影院信息管理</div> <div>退出</div>	序号	电影中文名	电影外文名	时长 (min)	上映日期	设置
	1	出租车司机	Taxi Driver	113	2024/05/10	<div>查看修改删除</div>
	2	她	Her	126	2024/04/26	<div>查看修改删除</div>
	3	红猪	Porco Rosso	94	2024/05/01	<div>查看修改删除</div>
	4	搏击俱乐部	Fight Club	139	2024/05/24	<div>查看修改删除</div>
	5	一一	A One and a Two	173	2024/05/17	<div>查看修改删除</div>
	6	精疲力尽	À bout de souffle	90	2024/05/03	<div>查看修改删除</div>
	7	四百击	Les quatre cents coups	99	2024/05/04	<div>查看修改删除</div>
	8	枪火	The Mission	84	2024/05/01	<div>查看修改删除</div>
	9	机动部队	PTU	88	2024/05/01	<div>查看修改删除</div>
	10	蜘蛛侠：平行宇宙	Spider-Man: Into the Spider-Verse	116	2024/05/10	<div>查看修改删除</div>
	11	后窗	Rear Window	112	2024/05/20	<div>查看修改删除</div>
	12	假面	Persona	85	2024/05/24	<div>查看修改删除</div>
共 15 条 < 1 2 >						新增电影

图 5.24 电影信息管理的界面实现图

在该界面中系统管理人员已经可以看到各电影的大致信息（序号、电影中文名、外文名、时长和上映日期），在下方也能看到总条数了解到当前已发布了多少条电影信息并且提供了分页查询。管理人员可以通过点击查看、修改、删除和新增电影来管理电影信息。

查看、修改和新增电影的界面实现图如下图 5.25 示

<

查看电影信息



出租车司机

外文名：Taxi Driver

导演：马丁·斯科塞斯

简介： 特拉维斯（罗伯特·德尼罗 Robert De Niro 饰）从越战中退伍回来后一直在纽约以开出租车为生。战争后的他多少有些失落，加上开出租车目睹了纽约夜幕下的种种罪恶，令他变得愤世嫉俗、痛恨社会。在追求总统候选人帕兰坦竞选办公室的秘书贝西失败后，特拉维斯决定去刺杀帕兰坦 以证明自己。在准备刺杀行动的过程中，特拉维斯遇上了被逼卖淫的雏妓艾瑞斯。在企图救艾瑞斯出火坑未果后，特拉维斯手持武器闯进淫窝，将里面的一干人等通通干掉了！本以为即将收到法律的惩罚，然而媒体却将特拉维斯描述成了一个拯救少女的英雄……

上映时间：2024 年 05 月 10 日

类型：剧情 犯罪

时长：113 min

片源地：美国

电影状态：正在上映

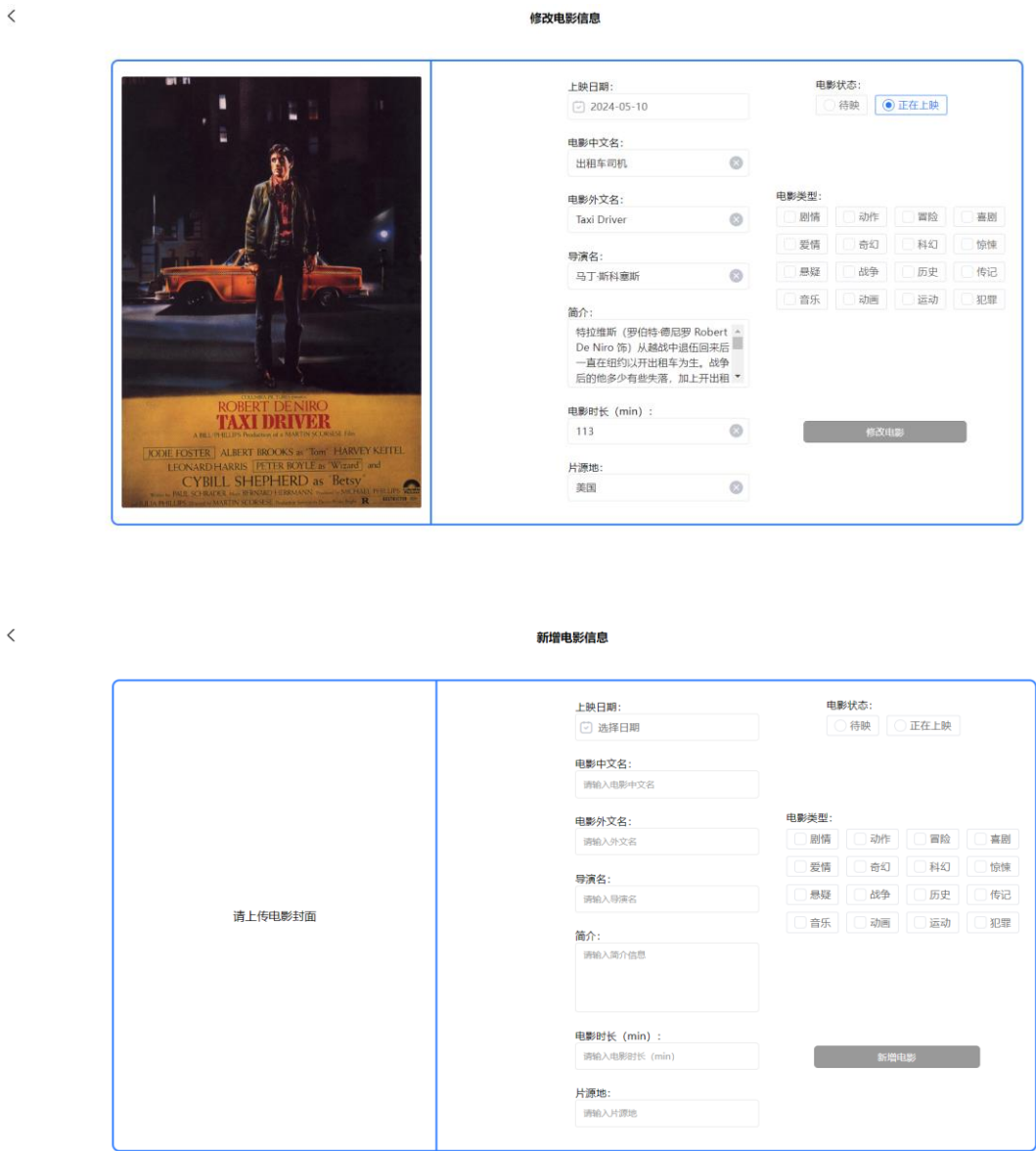


图 5.25 查看、修改和新增电影的界面实现图

5.4.3 影院信息管理模块实现

该模块主要实现系统管理人员对当前影院信息进行管理操作。
此模块涉及的接口如下表 5.13 示

表 5.13 影院信息管理模块接口表

接口	请求方式	请求参数	接口说明
/cinema/create	POST	Cinema cinema	新增影院信息
/cinema/infoAllCinema	GET	无	查询所有的影院信息
/cinema/infoById	GET	Long cinemaId	查询指定的影院信息
/cinema/delete	DELETE	Long cinemaId	删除指定的影院信息
/cinema/update	PUT	Cinema cinema	修改指定的影院信息

与上文的操作逻辑类似，该界面依旧给系统管理人员显示出了所有的影院信息并提供了相对应的查看、修改、删除和新增按钮方便管理人员对影院信息进行管理。

影院信息管理的界面实现图如下图 5.26 示



图 5.26 影院信息管理的界面实现图

查看、修改和新增影院的界面实现图如下图 5.27 示

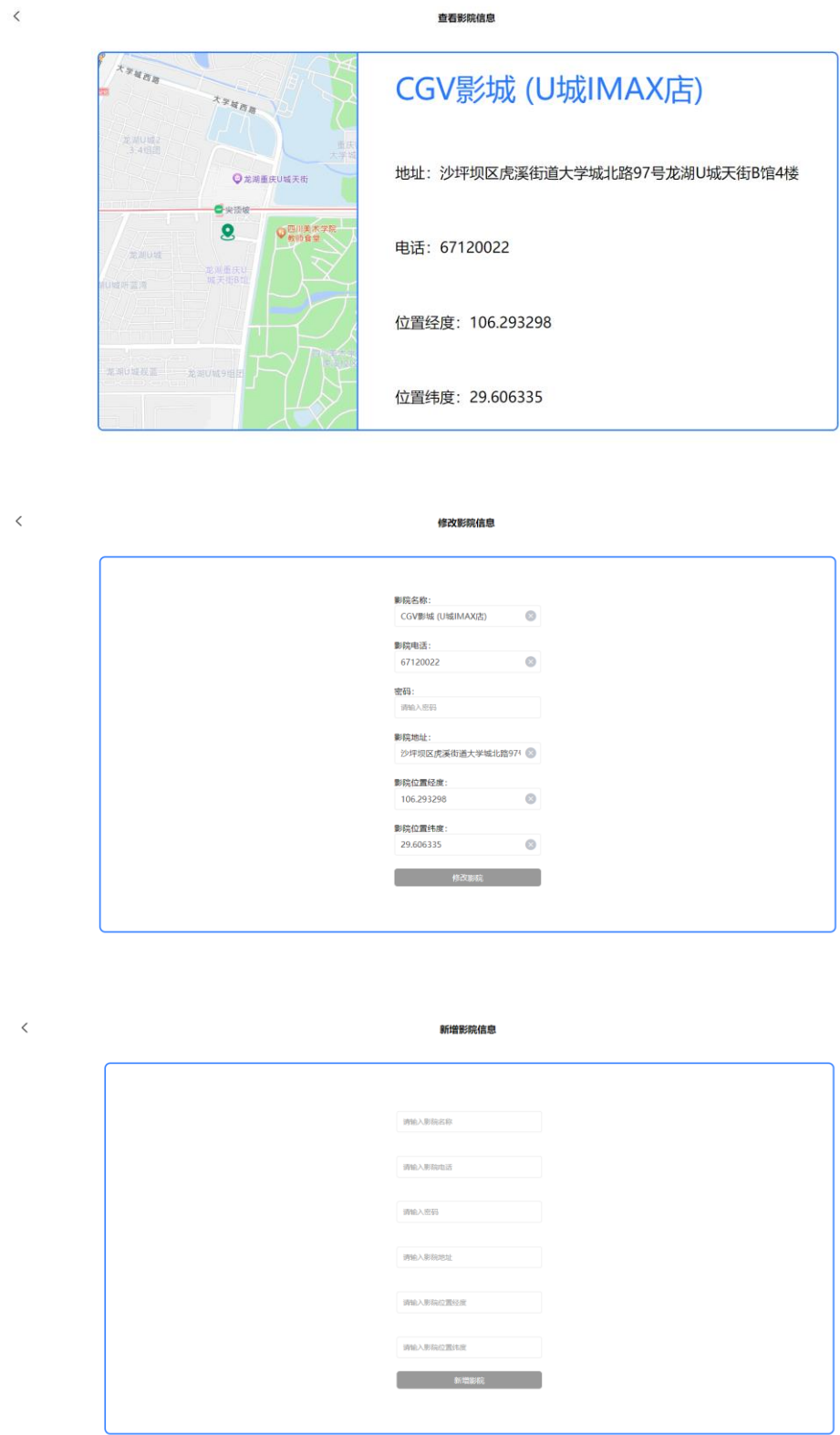


图 5.27 查看、修改和新增影院的界面实现图

5.5 本章小结

本章主要简略地说明了此系统的开发环境，前后详细地说明了用户端、影院

端和管理端各功能的实现，并且通过接口信息、时序图和界面实现图加以说明。

6 总结与展望

6.1 总结

本文主要围绕电影院订票 App 的设计与实现展开，从需求分析、系统设计到功能实现，完成了一个完整的项目流程。在系统开发过程中，虽然遇到了一些挑战，如用户需求多样性、前后端交互复杂性等，但通过不断学习与开发，最终实现了设计目标。

在功能实现方面，用户端、影院端和管理端的功能得到了充分的展现和实现。注册、登录、影片查询、选座购票等核心功能得以完善，为用户提供了便捷、快速的订票体验；同时，影院端管理功能也得到了有效设计和实现，为影院工作人员提供了高效的管理平台。

6.2 展望

展望未来，随着互联网技术的不断发展和用户需求的不断变化，电影院订票 App 还有进一步的优化和扩展空间。可以加强用户个性化推荐功能，提升用户体验；同时，加强对影院端的数据分析与反馈机制，为影院运营提供更精准的决策依据。此外，结合人工智能技术，可以实现更智能化的订票体验，进一步提升用户满意度和平台竞争力。

最后，本项目一定程度为电影院订票行业的发展和用户体验的提升做出了贡献，未来将继续努力完善和拓展，以满足用户和市场的需求。

参 考 文 献

- [1] 何波. 小型电影院售票管理系统的分析与设计[D]. 昆明: 云南大学, 2013: 1-2.
- [2] 邵新. 基于 SSH 的电影院网上订票系统的设计与实现[D]. 河北: 河北科技大学, 2016: 2-4.
- [3] 张海悦. 新一代信息技术在影院智能化升级中的应用探讨[J]. 现代电影技术, 2021, (8): 38-41.
- [4] 李林. 智慧城市信息栅格系统工程[M]. 南京: 南京东南大学出版社, 2021: 26-31.
- [5] 柏亚萌. 基于物联网 WebGIS 平台智慧管廊系统的设计与实现[D]. 北京: 北京邮电大学, 2020: 5-7.
- [6] DHALLA H K. Benchmarking the performance of RESTful applications implemented in spring boot Java and MS.Net core[J]. Journal of Computing Sciences in Colleges, 2020, (3): 149-152.
- [7] Larsson M. Hands-on Microservices with spring boot and spring cloud: Build and deploy Java microservices using spring cloud, Istio, and Kubernetes[M]. Packt Publishing Ltd, 2019: 46-52.
- [8] Ilman B, Ginanjar A. Rancang Bangun Web Service-JSON Menggunakan Kombinasi Spring dan MyBatis Framework dalam lingkungan Java Platform[J]. Jurnal Teknologika, 2019, 9(1): 68-73.
- [9] Bhagavathi C. Security Analysis of MySQL and PostgreSQL Database Management Systems[D]. University of North Texas, 2013: 76-79.
- [10] Saçak B, Bozkurt A, Wagner E. Learning design versus instructional design: A bibliometric study through data visualization approaches[J]. Education Sciences, 2022, 12(11): 752.
- [11] Redis. Introduction to Redis[EB/OL]. [2023-05-06]. <https://redis.io/docs/about/>.
- [12] 刘坤. 基于微服务架构的在线教学平台的设计与实现[D]. 陕西: 西北大学, 2022: 47.
- [13] Qin Z, Pan Y. Design of A Smart Tourism Management System through Multisource Data Visualization-Based Knowledge Discovery[J]. Electronics, 2023, 12(3): 642.
- [14] 贺适. 软件开发中数据库设计理论实践研究[J]. 电子测试, 2020.
- [15] 腾讯. 什么是 POP3/IMAP/SMTP 服务[EB/OL].
https://wx.mail.qq.com/list/readtemplate?name=app_intro.html#/agreement/authorizationCode.
- [16] 赵俊贺. 面向异构数据的生涯规划可视化平台设计与实现[D]. 北京: 中国科学院大学, 2022.
- [17] 阿里云. 什么是对象存储 OSS[EB/OL].
<https://help.aliyun.com/zh/oss/product-overview/what-is-oss?spm=a2c4g.11186623.0.0.42be1eb7KPjzxY>.

致 谢

在完成本论文的过程中，我要感谢许多人的支持和帮助。首先，我要感谢我的导师杨广超杨老师，他在我整个论文写作过程中给予了我耐心的指导和宝贵的建议，使得论文能够顺利进行并最终完成。其次，我要感谢重庆大学和身边的同学们，提供给我一个优秀的平台，在学术和技术上给予了我许多帮助和启发，使我能够不断进步和提升自己。

最后，我要感谢所有为本论文提供数据、信息和参考文献的相关机构和个人，在研究过程中，他们的贡献为论文的深入和完整性提供了重要的支撑。

原创性声明

郑重声明：所呈交的论文（设计）《_____》，
是本人在导师的指导下，独立进行研究取得的成果。除论文（设计）中已经标注引用的内容外，本论文（设计）不包含其他人或集体已经发表或撰写过的作品成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果，并承诺因本声明而产生的法律结果由本人承担。

论文（设计）作者签名：_____

日期：_____

使用授权书

本论文（设计）作者完全了解学校有关保留、使用论文（设计）的规定，同意学校保留并向国家有关部门或机构送交论文（设计）复印件和电子版，允许论文（设计）被查阅和借阅。本人授权重庆大学将本论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制方式保存和汇编本论文（设计）。

本论文（设计）属于：

保 密 ☐ 在_____年解密后适用本授权书

不保密 ☐

论文（设计）作者签名：_____

指导教师签名：_____

日期：_____

日期：_____