

---

# **Software Requirements and Design Document**

**for**

## **Cinexus**

### **Cinema Managment System**

**Prepared by: Rayyan M. Zia  
& Usman Zafar**

**Random Binary Universe**

**Friday, 1st December 2023**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 Product Scope	1
1.3 Title	1
1.4 Objectives	1
1.5 Problem Statement	1
<b>2. Overall Description</b>	<b>2</b>
2.1 Product Perspective	2
2.2 Product Functions	3
2.3 List of Use Cases	3
2.4 Use Case Diagram	4
<b>3. Other Nonfunctional Requirements</b>	<b>5</b>
3.1 Performance Requirements	5
3.2 Safety Requirements	5
3.3 Security Requirements	5
3.4 Software Quality Attributes	5
3.5 Business Rules	5
3.6 Operating Environment	6
3.7 User Interfaces	6
<b>4. Domain Model</b>	<b>6</b>
<b>5. System Sequence Diagram</b>	<b>7</b>
<b>6. Sequence Diagram</b>	<b>17</b>
<b>7. Class Diagram</b>	<b>26</b>

# 1. Introduction

## 1.1 Purpose

*This Software Requirements Specification (SRS) document outlines the requirements for the Cinexus Cinema Management System, version 1.0. This document pertains to the first version of the Cinexus Cinema Management System, denoted as version 1.0. Future revisions or releases may be necessary to enhance or extend the system's functionalities based on user feedback and evolving requirements.*

## 1.2 Product Scope

*Cinexus - A Cinema Management System built to provide a unified platform for managing multiple cinemas, enabling customers to book, cancel, and view movie showtimes, as well as facilitate cinema managers in organizing and overseeing cinema-related operations.*

## 1.3 Title

*Cinexus - Cinema Management System*

## 1.4 Objectives

*The Cinexus Cinema Management System is designed to cover the following key functionalities:*

### *1- Customer Operations:*

- *Book and reserve seats for movie showtimes.*
- *Cancel reservations.*
- *View the catalog of available movies.*
- *Add and view customer reviews.*

### *2- Cinema Manager Operations:*

- *Add new cinemas to the system.*
- *Arrange movie showtimes for each cinema.*
- *Resolve customer issues and inquiries.*

### *3- Additional Customer Feature:*

- *Arrange Private Screening Events:*  
*Customers can contact cinema managers equipped with necessary supplies for organizing private screening events.*

## 1.5 Problem Statement

*The Cinexus Cinema Management System project was initiated in response to the growing complexity and inefficiencies faced by cinema-goers in deciding between prices of multiple theaters, filtering through movies and ratings and getting quick query responses. The entertainment industry has witnessed a surge in demand, leading to an increased number of cinemas, movie showtimes, and customer interactions. The existing manual systems were proving inadequate to handle the workload of modern cinema operations, resulting in scheduling conflicts, customer service challenges, and a lack of centralized management. The project aimed to streamline and automate these processes to enhance overall operational efficiency.*

*The core problem addressed by the Cinexus Cinema Management System is the need for a comprehensive solution that consolidates cinema management tasks and customer interactions into a centralized platform. The manual methods for booking, cancellations, movie cataloging, and event organization were time-consuming and error-prone. The software provides a systematic approach to cinema management, enabling cinema managers to effortlessly add new cinemas, schedule showtimes, and promptly address customer issues. For customers, the system simplifies the movie-going experience, offering features like easy seat reservations, access to movie catalogs, and the ability to organize private screening events. By automating these processes, the Cinexus system significantly reduces manual workload, minimizes errors, and ensures a seamless cinema experience for both managers and customers.*

*Feasibility studies were conducted to assess the practicality of implementing the Cinexus Cinema Management System. The technical feasibility confirmed the viability of developing the system using Java, ensuring compatibility and scalability. Economic feasibility considerations indicated a positive return on investment due to increased operational efficiency, reduced manual labor costs, and enhanced customer satisfaction. Furthermore, the system addressed the operational and organizational feasibility by streamlining workflows, providing a user-friendly interface, and aligning with the business goals of cinema operators. Overall, the Cinexus project was deemed feasible, offering a strategic and technologically sound solution to the challenges faced by cinema management in the contemporary entertainment landscape.*

## 2. Overall Description

### 2.1 Product Perspective

*The Cinexus Cinema Management System is a new, self-contained product designed to revolutionize the way multiple cinemas are managed and how customers interact with cinema services. This system is not a replacement for existing systems but rather an innovative solution to address the challenges faced by cinema operators in handling the increasing complexity of modern cinema operations. Its introduction is prompted by the need for a centralized platform that seamlessly integrates cinema management tasks, enhances customer experience, and improves overall operational efficiency. As the entertainment industry evolves, there is a demand for a sophisticated solution to handle tasks such as scheduling showtimes, customer reservations, and event management. Cinexus is developed as a standalone product to cater to these specific needs, providing a user-friendly interface for both cinema managers and customers.*

## 2.2 Product Functions

### *Customer Operations:*

- *Reserve and book seats for movie showtimes.*
- *Cancel reservations.*
- *View the catalog of available movies.*
- *Add and view customer reviews.*

### *Cinema Manager Operations:*

- *Add new cinemas to the system.*
- *Arrange movie showtimes for each cinema.*
- *Resolve customer issues and inquiries.*

### *Additional Customer Feature:*

- *Arrange Private Screening Events:*
- *Customers can contact cinema managers equipped with necessary supplies for organizing private screening events.*

## 2.3 List of Use Cases

*Login or Register Account*

*View Movies*

*Reserve Seats*

*Make Payments*

*Give Ratings, Feedbacks and Complains*

*Refund and Cancel Bookings*

*Add Cinemas*

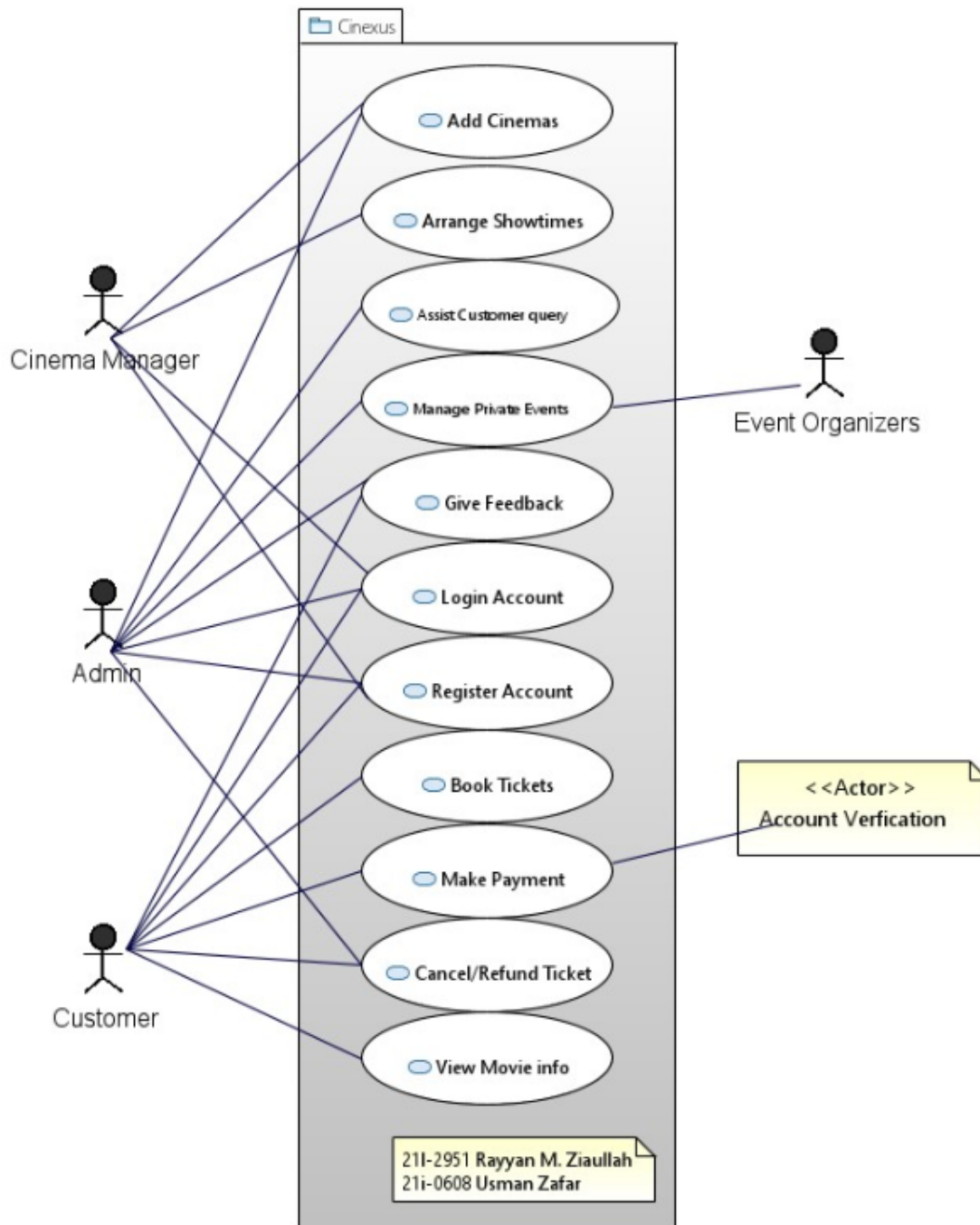
*Arrange Movie Showings*

*Assist Customer Query*

*Manage Private Events*

## 2.4 Use Case Diagram

### Use Case Diagram



### **3. Other Nonfunctional Requirements**

#### **3.1 Performance Requirements**

The system needs to be responsive to user actions, ensuring a smooth and quick user experience. It should handle a large number of users simultaneously during peak usage without significant slowdowns. The system should also be available 24/7 with minimal downtime for maintenance. Information about movies and cinemas should load swiftly, and payment transactions should be processed securely and efficiently. Regular data backups are necessary for data security, and data restoration should be quick in case of system failures.

#### **3.2 Safety Requirements**

*User data, especially personal and payment information, must be stored and transmitted securely to prevent unauthorized access. The system must adhere to privacy regulations, allowing users to control their data and providing a clear privacy policy. Emergency procedures should be in place to handle data breaches or system failures and promptly notify affected users. Strong user authentication mechanisms should be implemented to prevent unauthorized access.*

#### **3.3 Security Requirements**

*Users must create accounts with secure passwords and verify their email addresses for identity confirmation. The system should incorporate role-based access control to restrict users to functionalities appropriate to their roles (e.g., customer, cinema owner, admin). Sensitive data should be encrypted during both storage and transmission. Regular security audits are necessary to monitor system logs for breaches, and the system must comply with relevant data protection and security regulations, following external policies.*

#### **3.4 Software Quality Attributes**

*The system should prioritize ease of use, featuring a user-friendly interface and intuitive navigation. It should operate reliably with minimal downtime or crashes, and the codebase should be well-documented and modular to facilitate future updates and maintenance. Compatibility with various web browsers and operating systems is essential. Additionally, the system's architecture should allow for easy scalability to accommodate future growth.*

#### **3.5 Business Rules**

*Business rules are operational guidelines that determine who can perform specific actions under defined conditions within the system. They include rules like user registration criteria, cinema owner verification, booking limitations, cinema listing requirements, cancellation policies, and user role privileges. These rules ensure orderly operation and compliance with policies and regulations.*

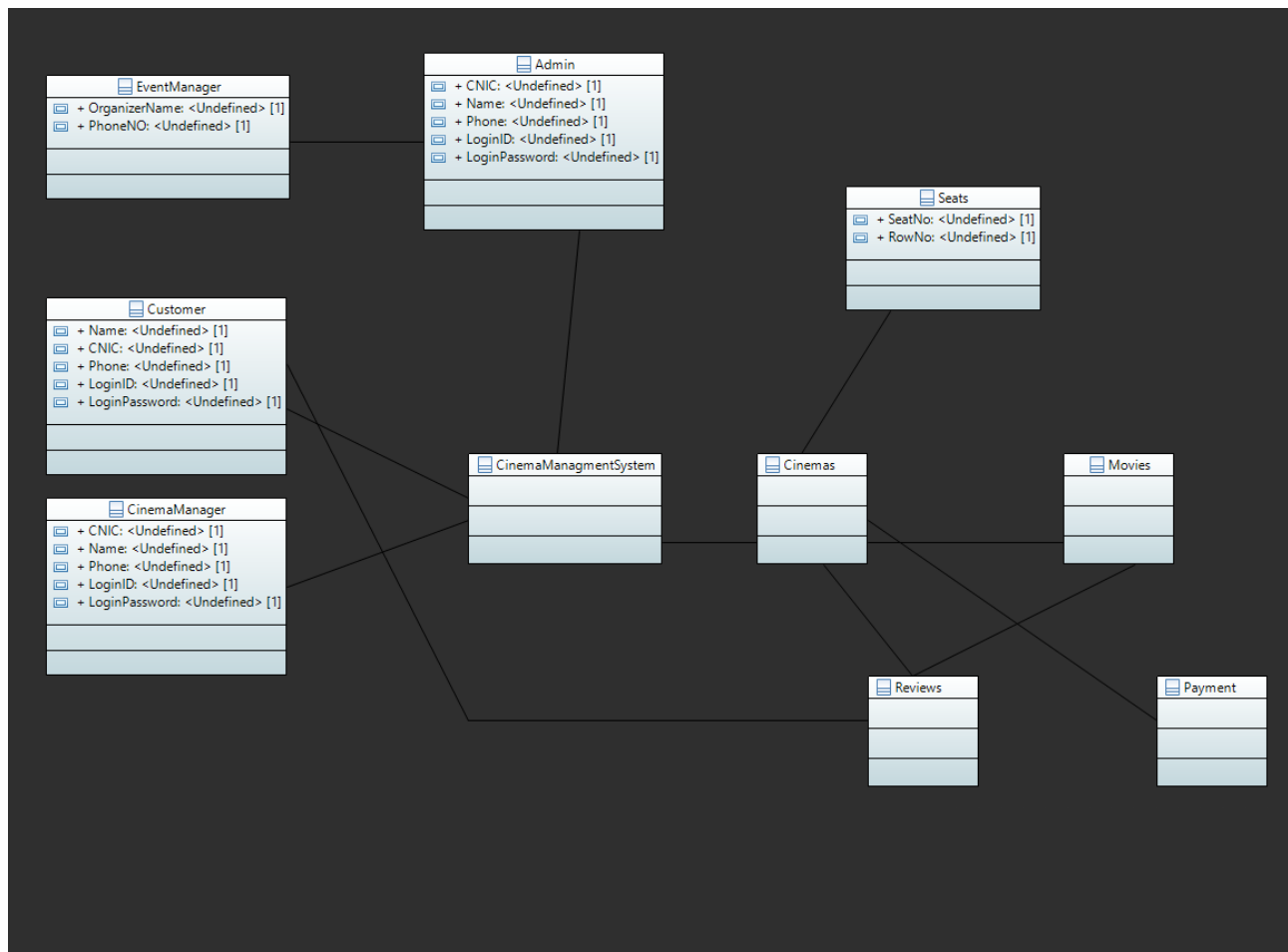
### 3.6 Operating Environment

The system will function as a web-based platform accessible from modern web browsers and mobile devices. It should be capable of running on standard web hosting servers with either Linux or Windows operating systems, supporting popular server-side scripting languages.

### 3.7 User Interfaces

The system should provide a responsive web interface optimized for both desktop and mobile devices. The design should prioritize a user-friendly experience with straightforward navigation, efficient search functionality, and an intuitive booking process. Detailed user interface design specifications should be documented separately for reference during development.

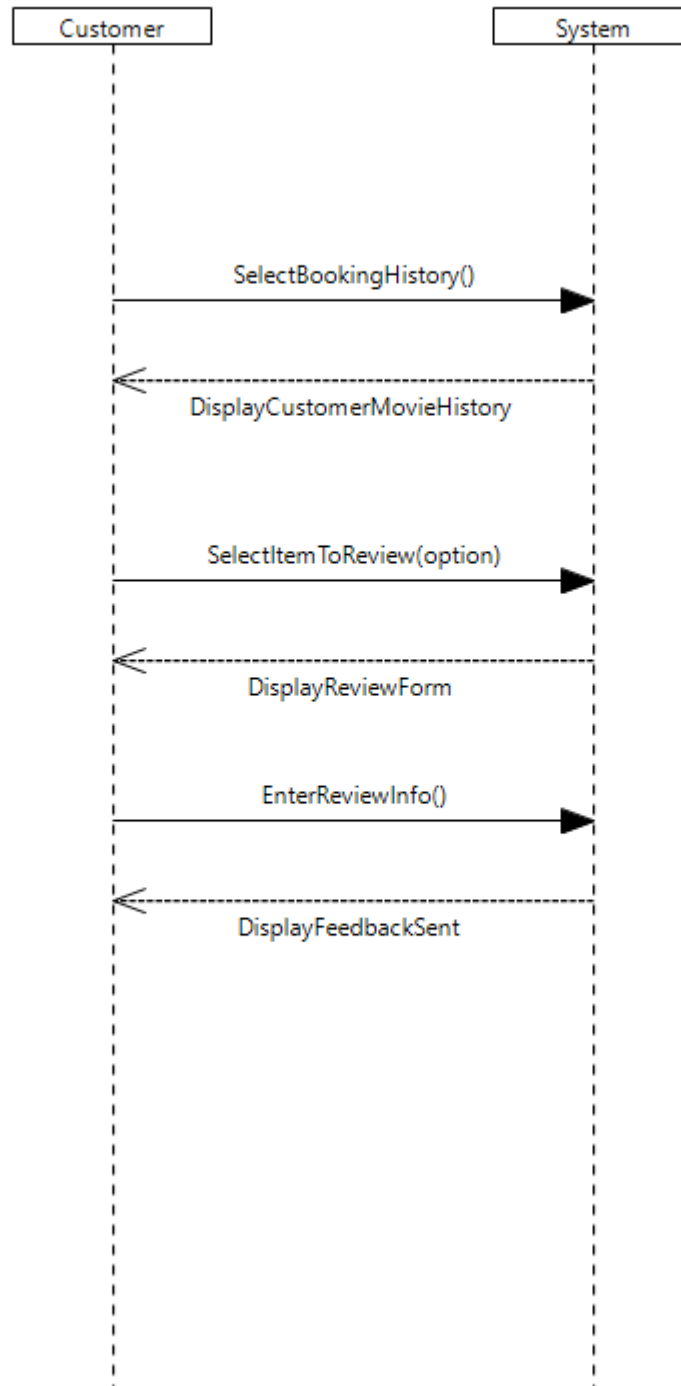
## 4. Domain Model

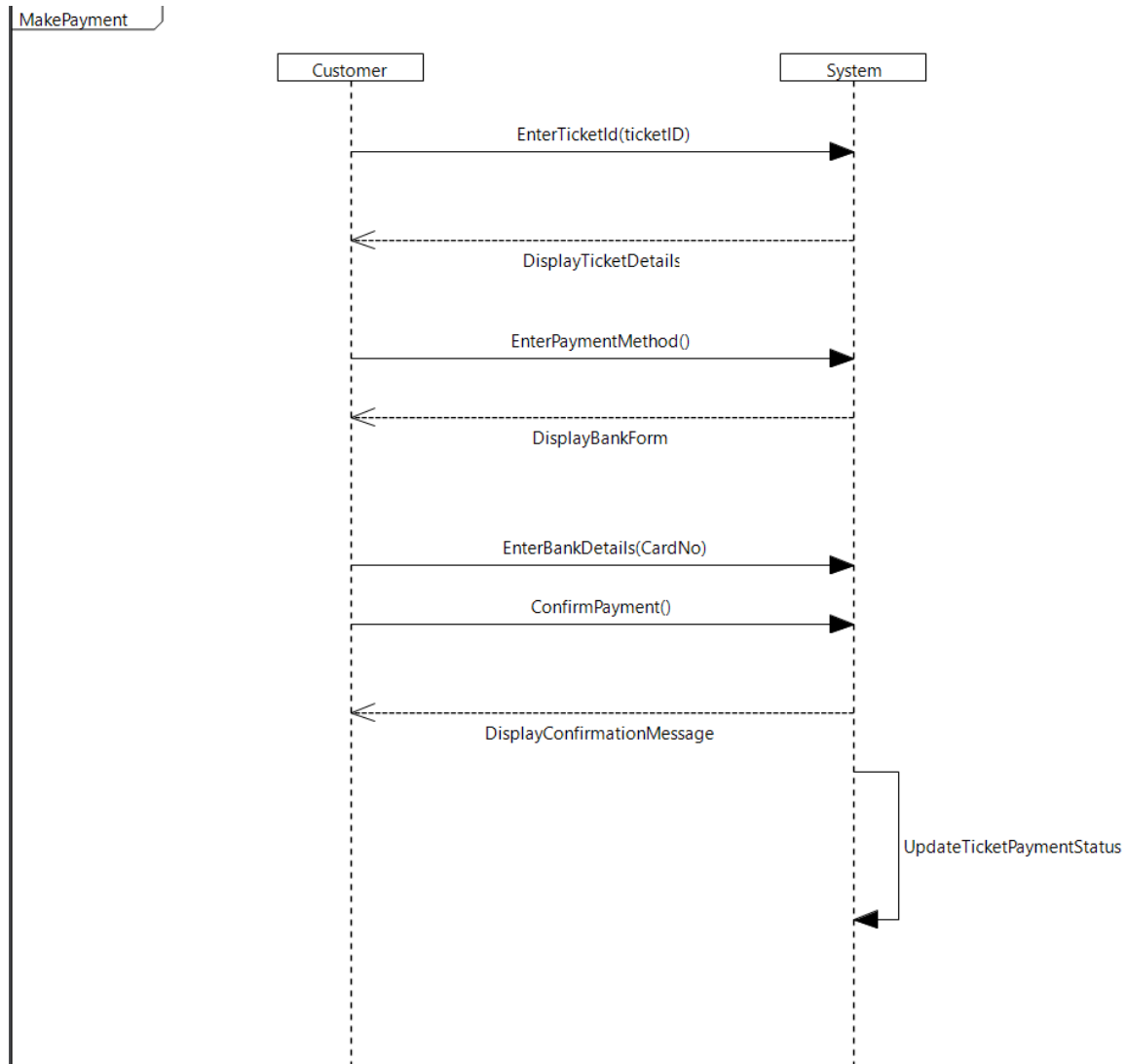


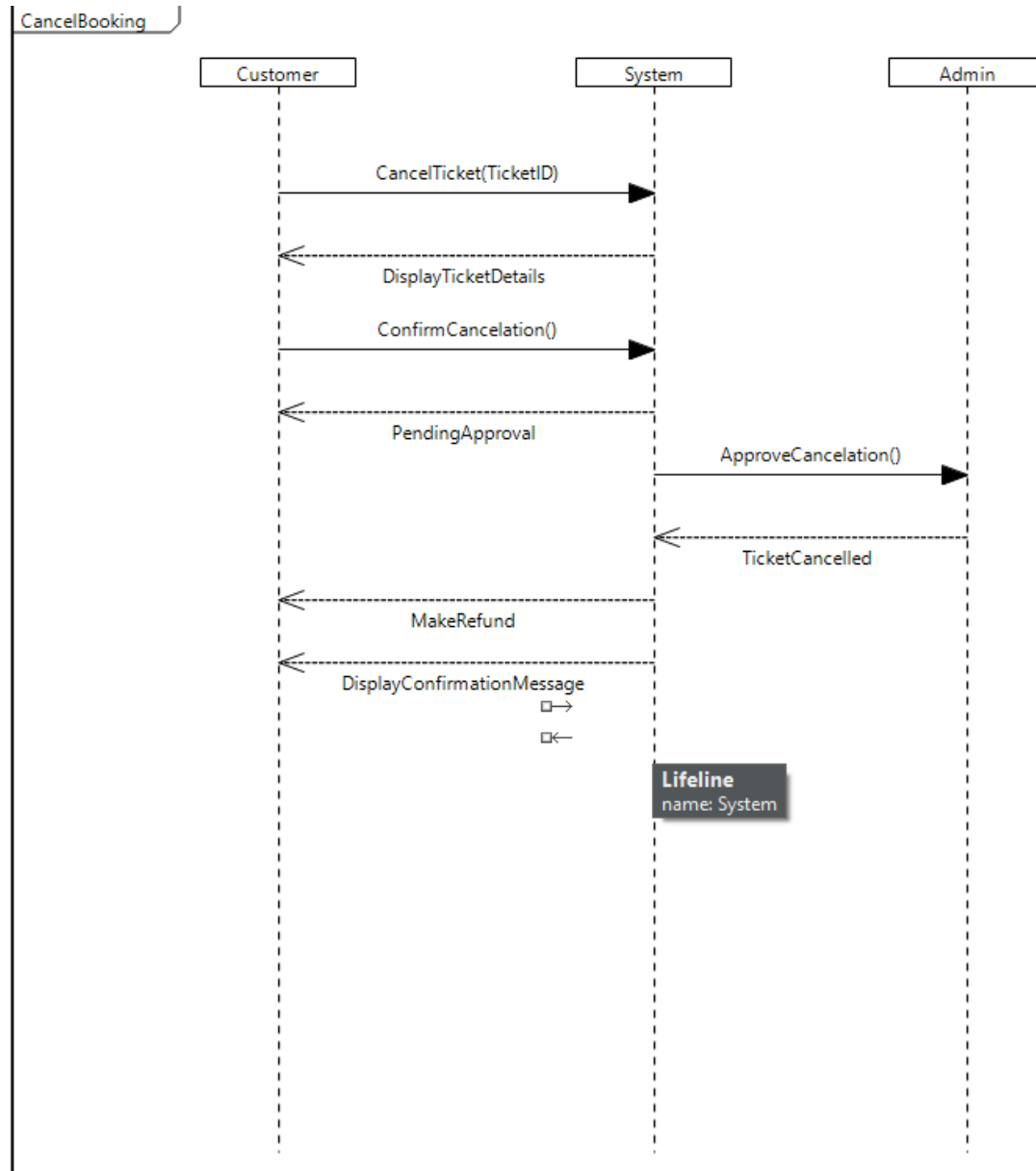


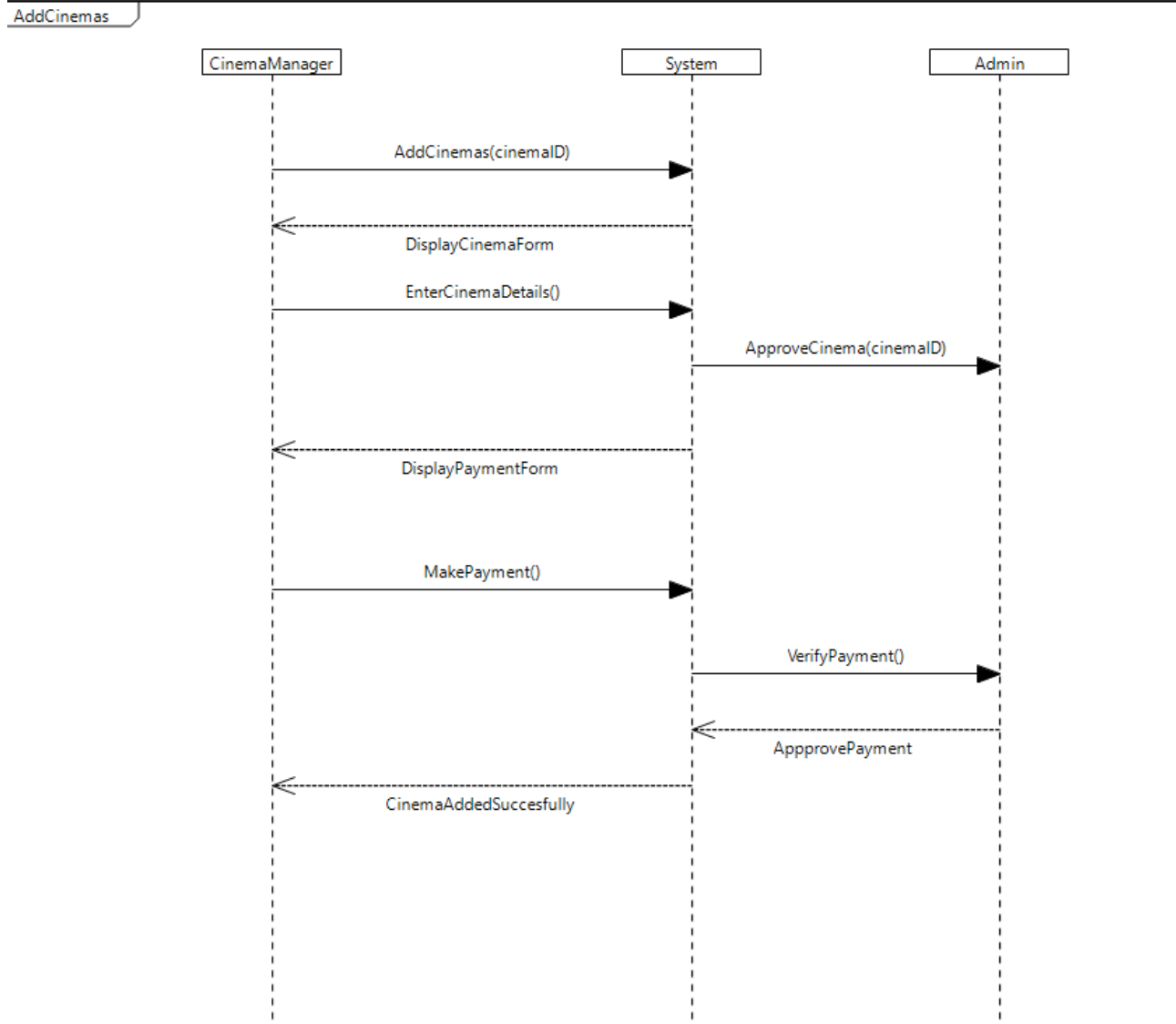
## 5. System Sequence Diagram

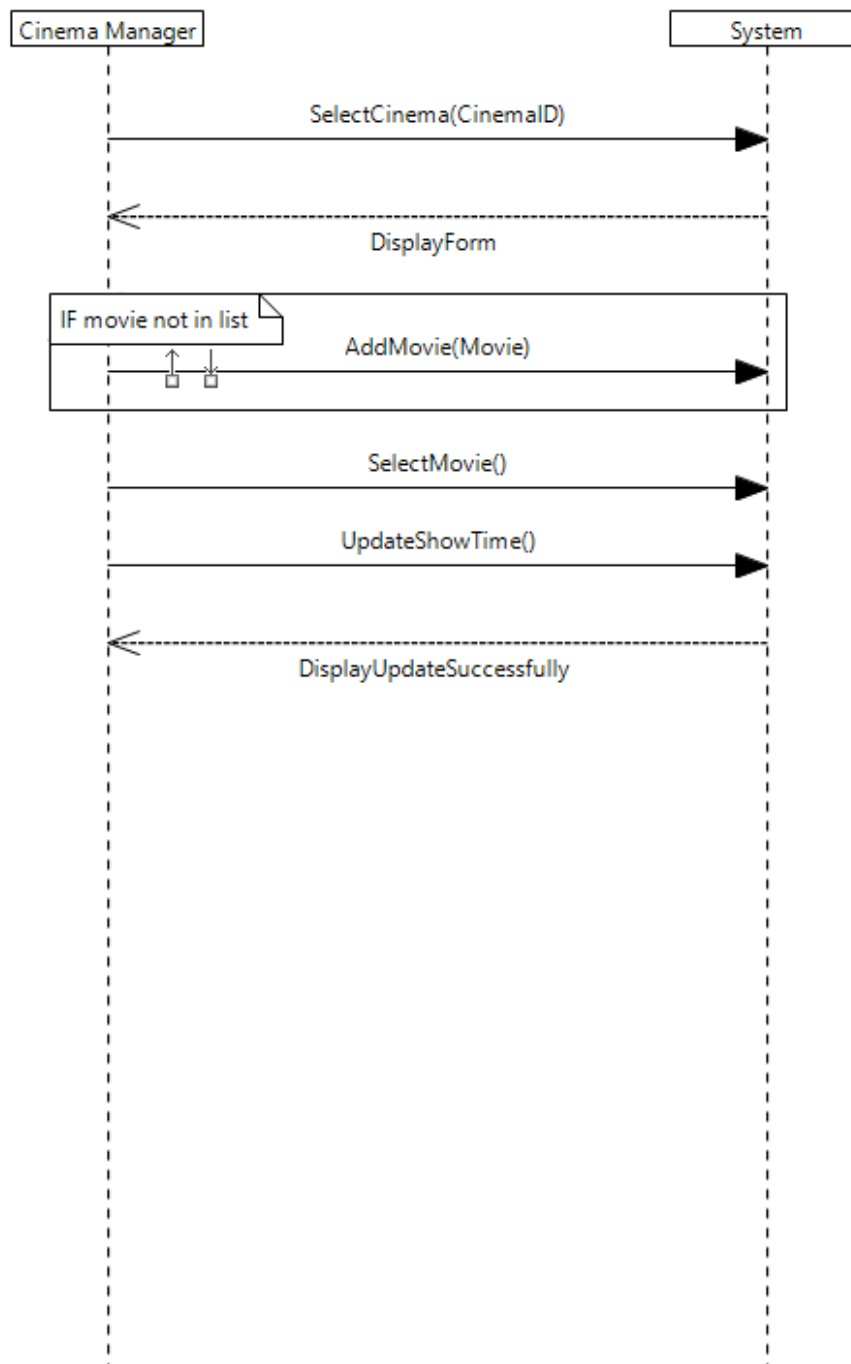
GiveFeedback

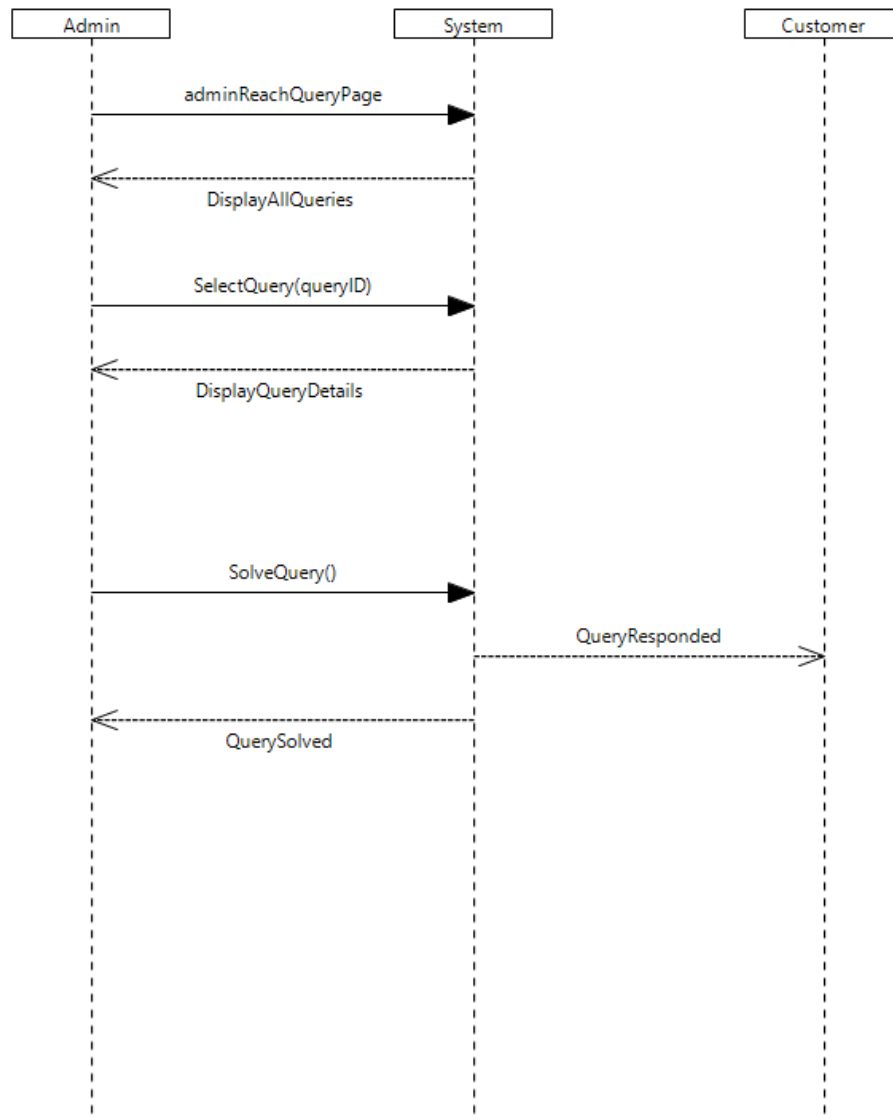




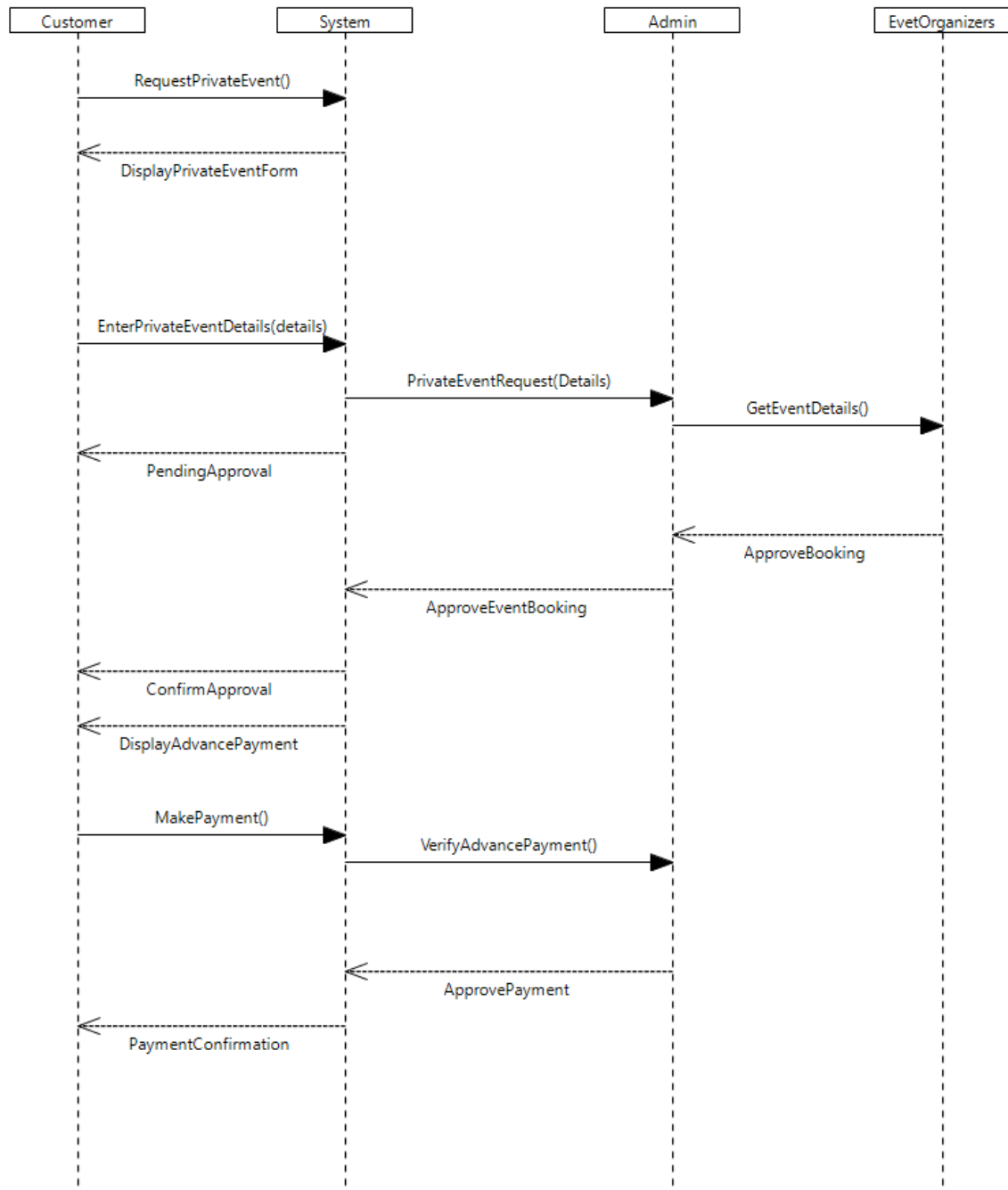


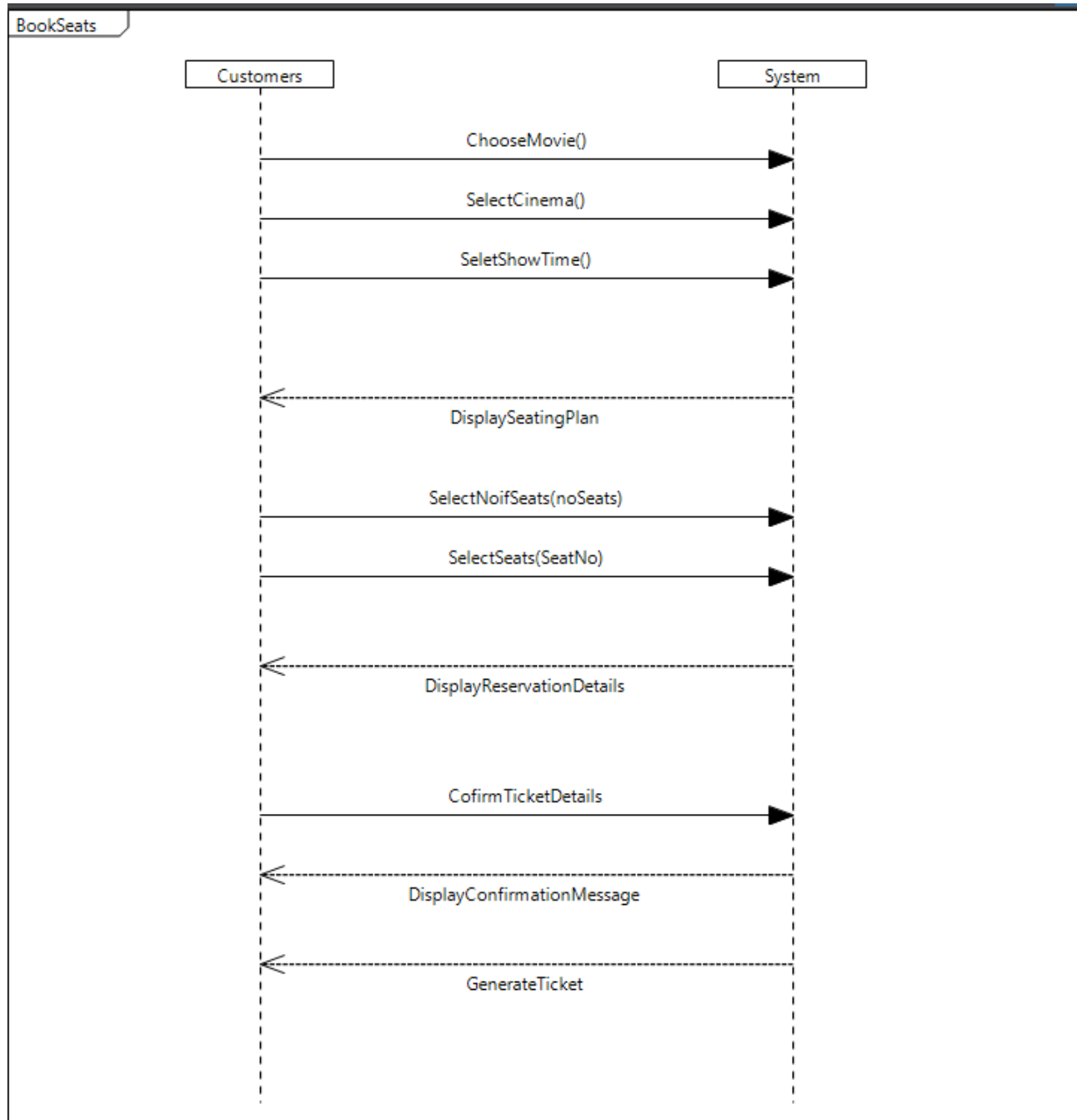


ArrangeMovieShowings

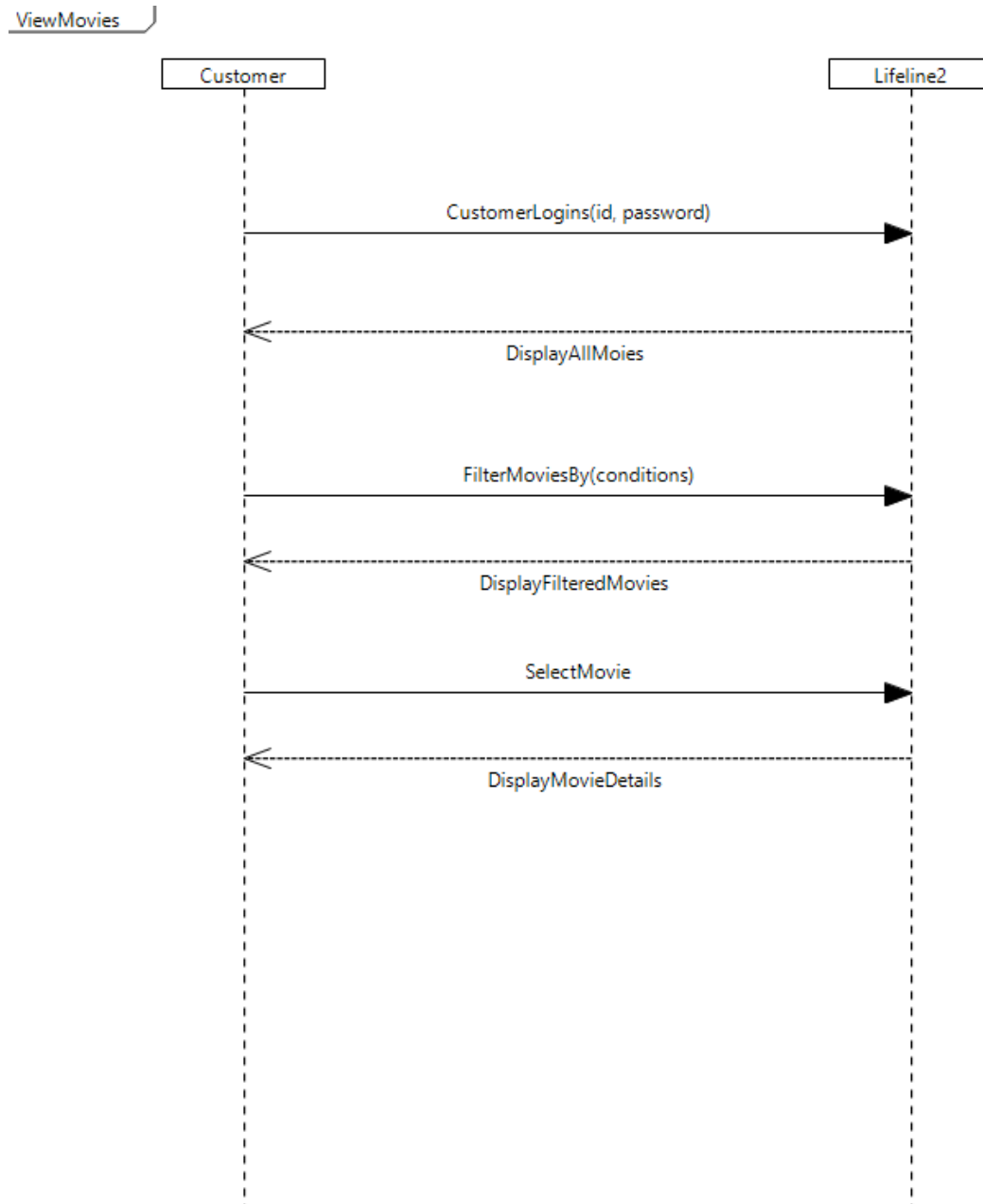
AssistCustomerQuery

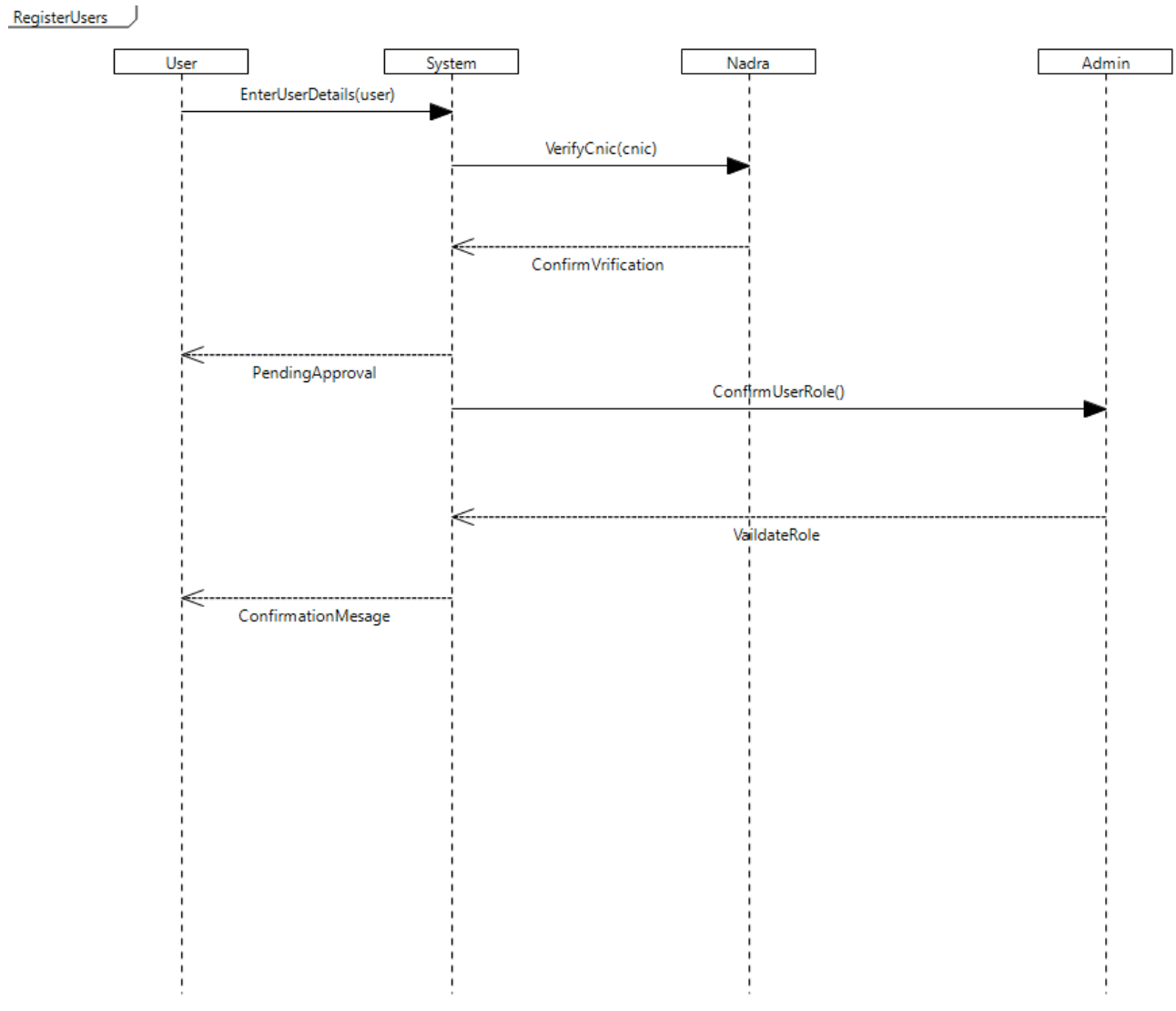
## OrganizePrivateEvent



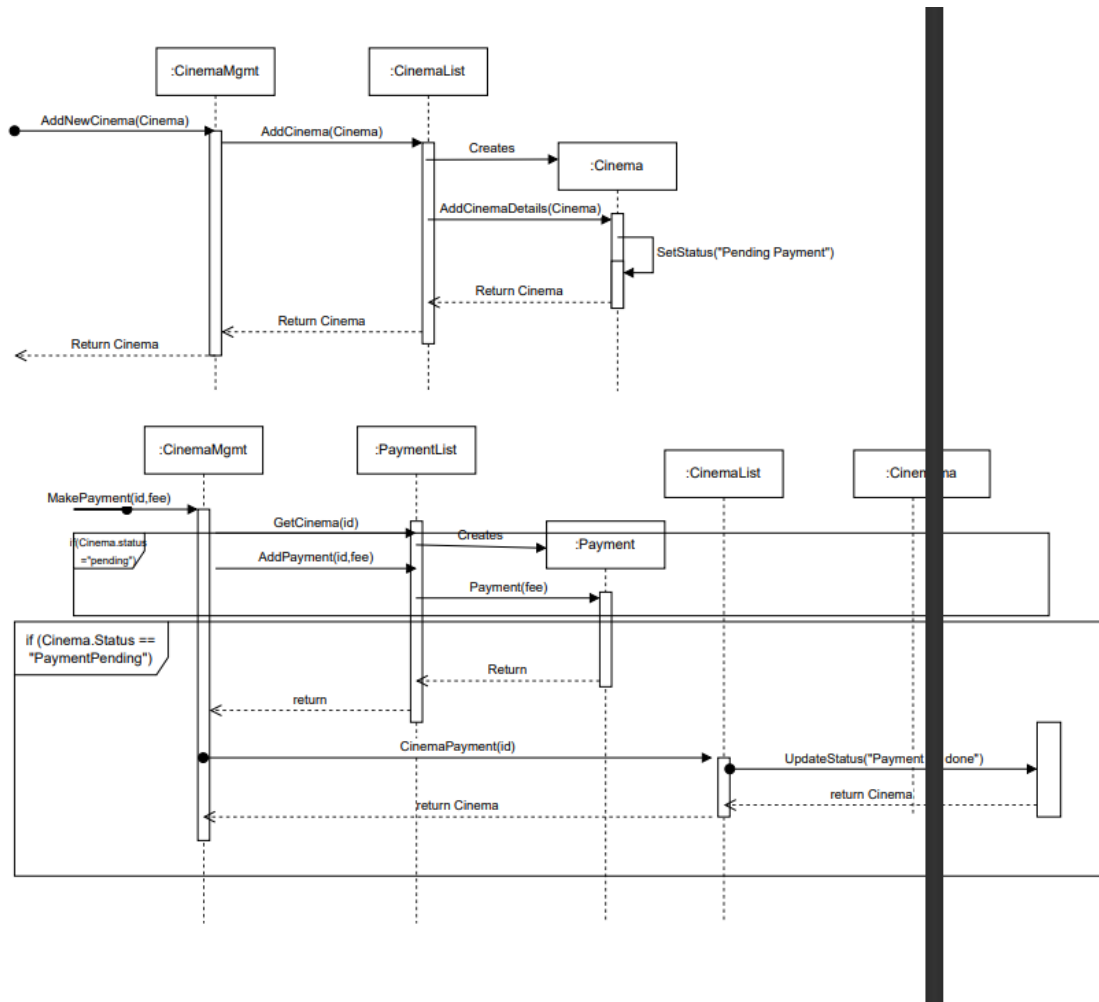


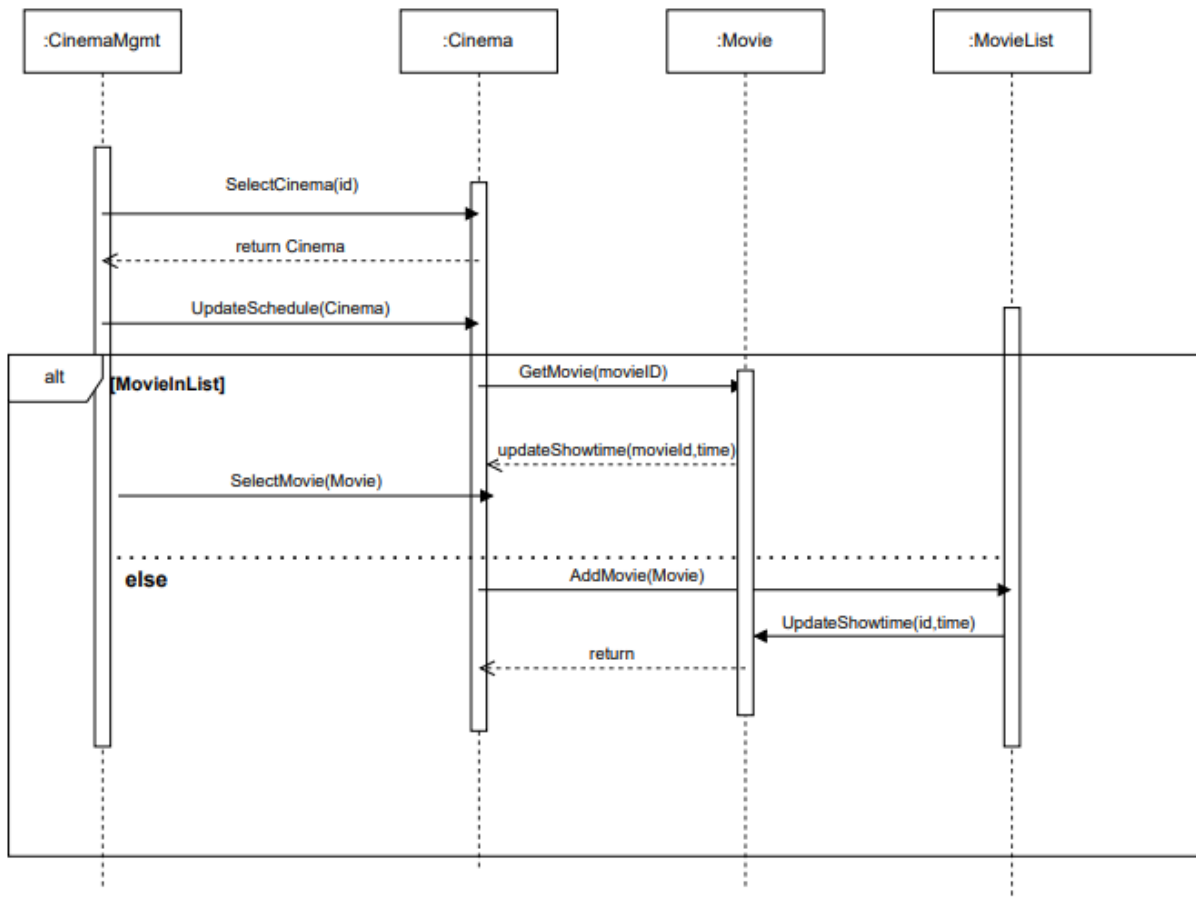


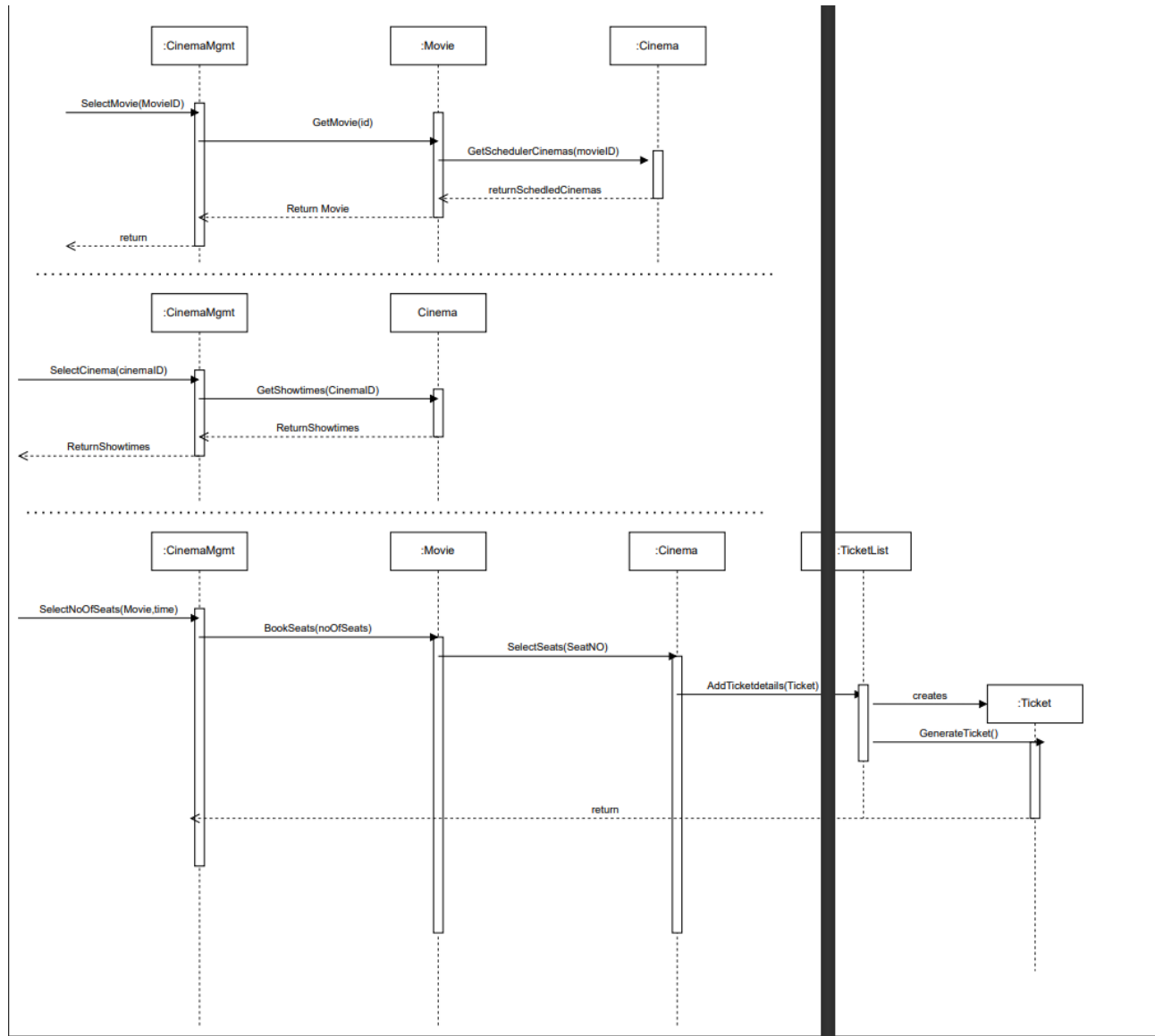


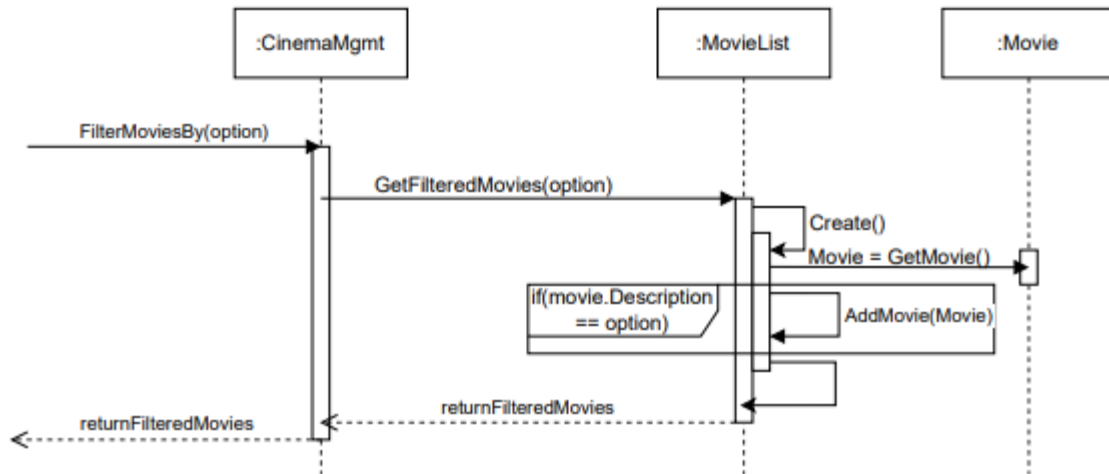


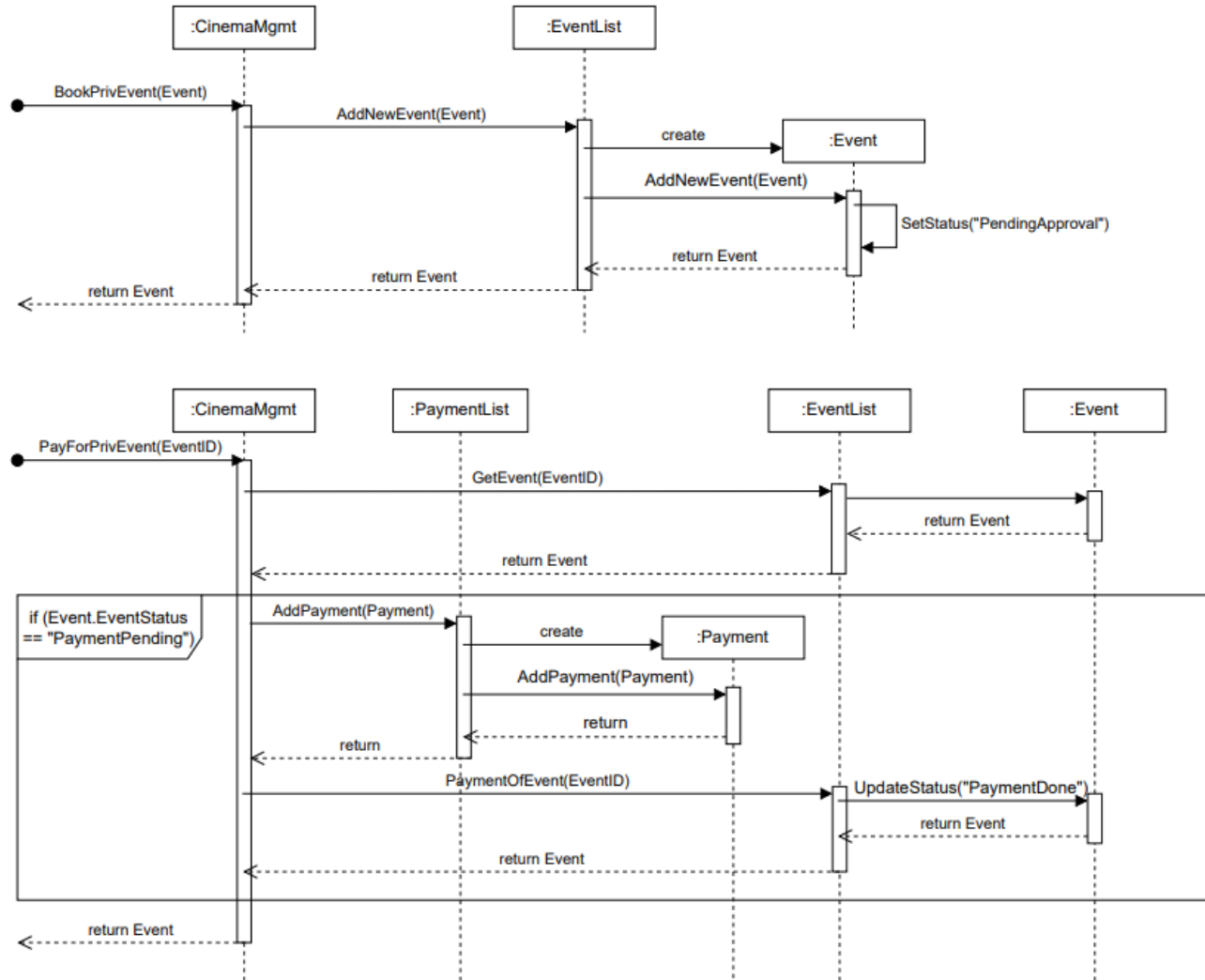
## 6. Sequence Diagram

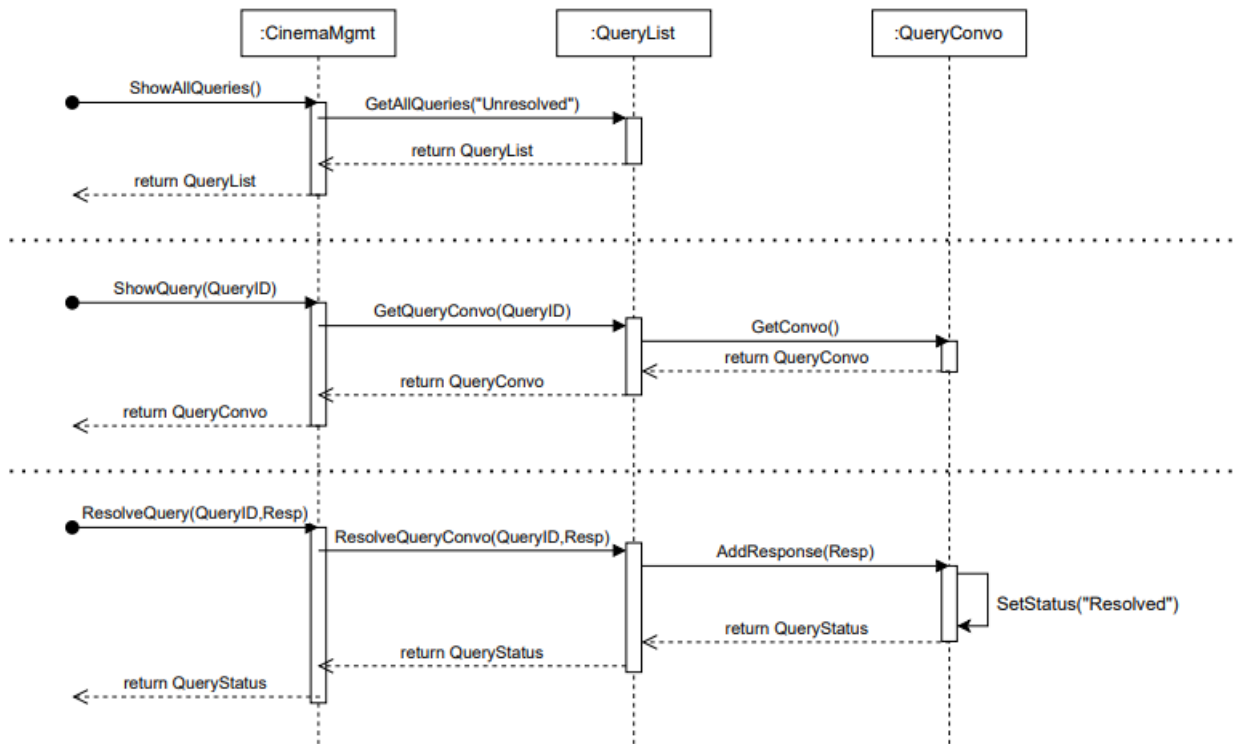




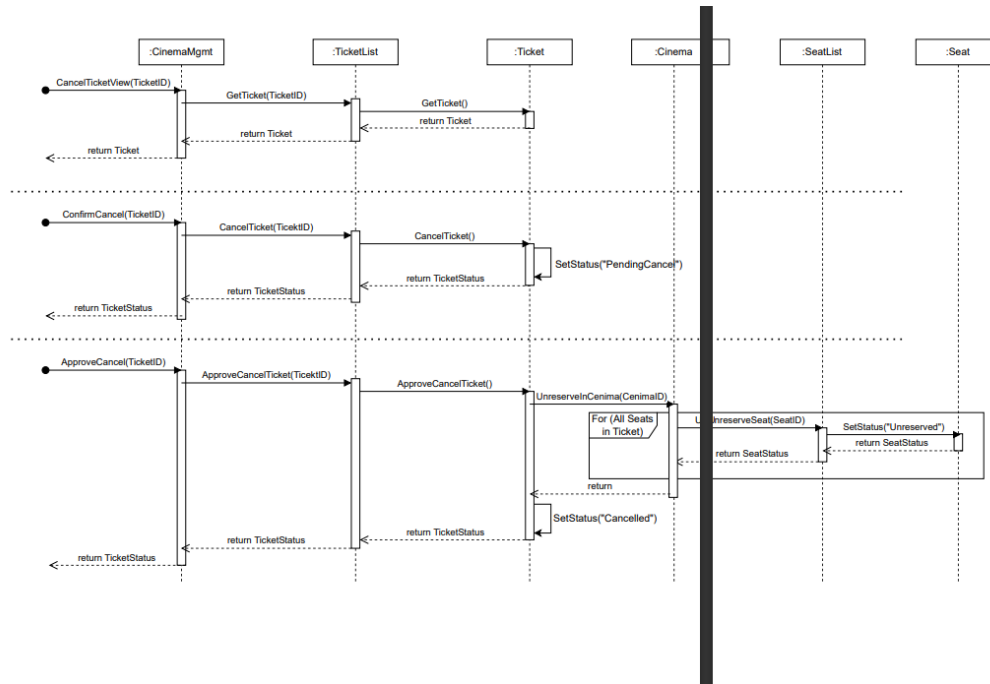


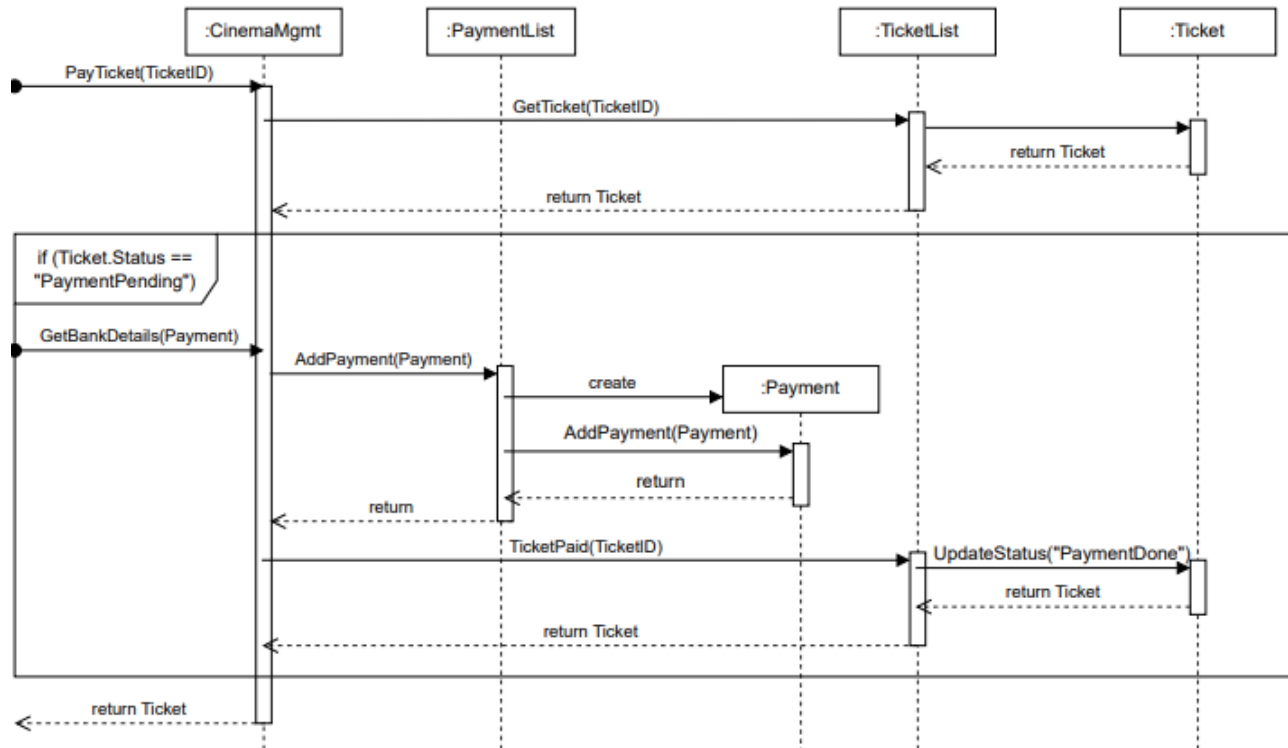


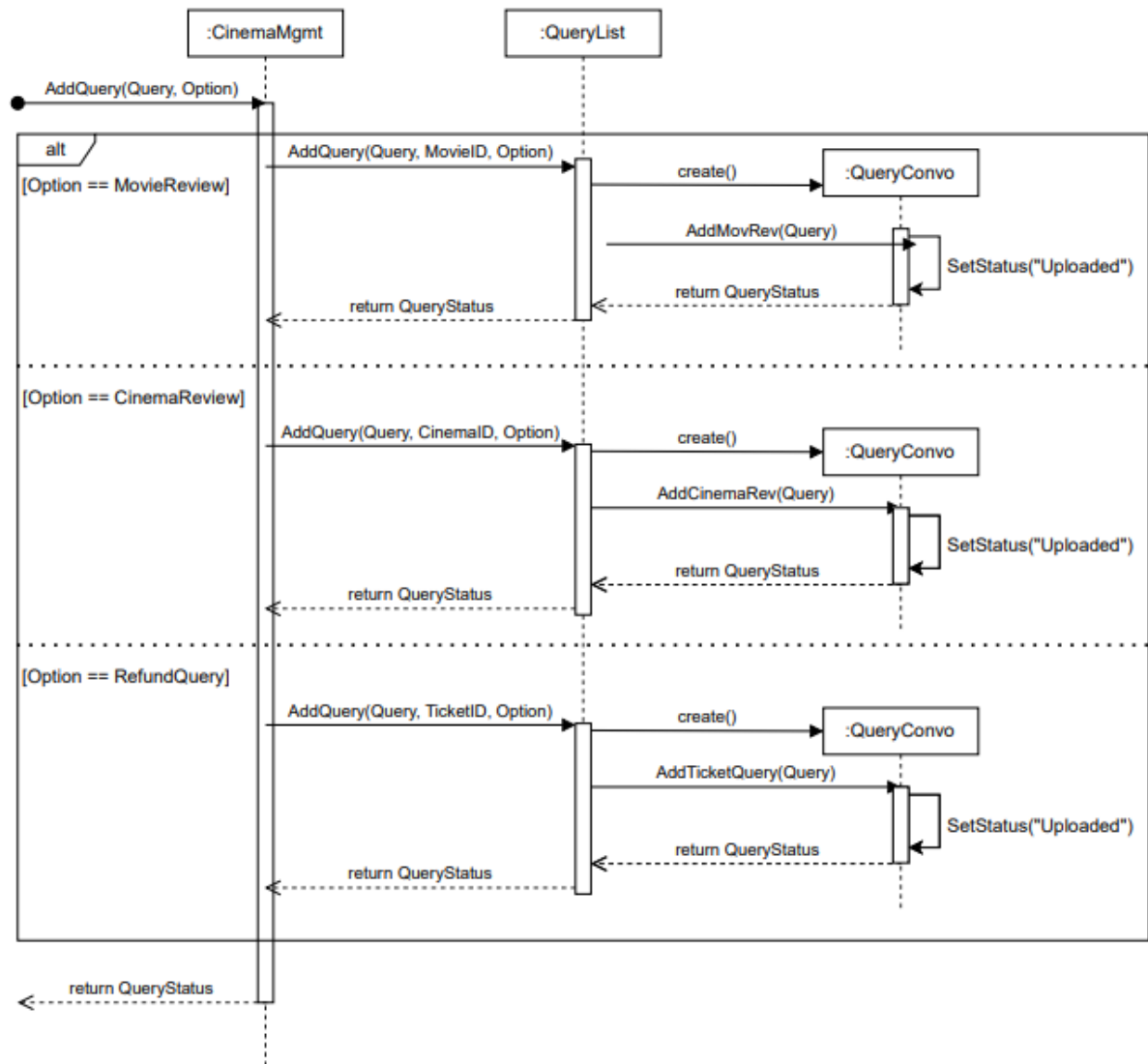












```

classDiagram
    class Seats {
        +status: string
        +Seats: int
        +SeatStatus()
        +SelectSeats()
        +Bookseats(Seats)
    }
    class SeatList {
        + Seats: List
        + GetSeat(SeatID)
        +SelectSeats()
        + Bookseats(Seats)
    }
    class Cinema {
        + status: string
        +Seats: int
        +AddCinema(cinema)
        +SelectSeats()
        +Bookseats(Seats)
    }
    class CinemaList {
        + Cinema: cinema
        + AddCinema(Cinema)
        + Cinemapayment(cinemaID)
    }
    class CinemaMgmt {
        + field: type
        + method(type): type
    }
    class EventList {
        + event: Event
        + AddNewEvent(Event)
        + Eventpayment(eventID)
    }
    class Event {
        + status: string
        + AddNewEvent(Event)
        + setStatus()
    }
    class TicketList {
        + movie: Movie
        + AddMovie(Movie)
        + ApplyFilterCondition()
    }
    class MovieList {
        + movie: Movie
        + AddMovie(Movie)
        + ApplyFilterCondition()
    }
    class Ticket {
        + seats: int
        + UpdateShowtime(time)
        + Bookseats(seats)
    }
    class Movie {
        + seats: int
        + UpdateShowtime(time)
        + Bookseats(seats)
    }
    class PaymentList {
        + fee : int
        + Addpayment(fee)
    }
    class Payment {
        + fee: int
        + payment(fee)
    }

    Seats "1" *-- "1" SeatList
    SeatList "1" *-- "1" Cinema
    Cinema "1" *-- "1" CinemaList
    CinemaList "1" *-- "1" CinemaMgmt
    CinemaMgmt "1" *-- "1" EventList
    EventList "1" *-- "1" Event
    Event "1" *-- "1" Payment
    Payment "1" *-- "1" PaymentList
    TicketList "1" *-- "1" Ticket
    MovieList "1" *-- "1" Movie
    Ticket "1" *-- "1" Movie
    Ticket "1" *-- "1" Payment
    
```