PHENIKAA UNIVERSITY

PHENIKAA UNIVERSITY: SCHOOL OF COMPUTING

COURSE: SOFTWARE ARCHITECTURE

Lab 1: Elicitation & Modeling Requirements for Movie Ticket Booking System

Instructor:	Vu Quang Dung
Group 8:	Nguyen Gia Bao (23010383)
	Pham Xuan Bach(23010118)
	Bui Minh Duc (23010513)
	Vu Duc Hieu (23010223)
Class:	**CSE703110-1-2-25(N01)**

Hanoi, 2025 December

MỤC LỤC

1. Abstract/Summary

The Movie Ticket Booking System project aims to develop a robust movie ticket booking platform. This initial lab focused on requirements elicitation and modeling. Therefore, we documented essential Functional, Non-Functional, and Architecturally Significant Requirements (ASRs) in a structured format. The core system behavior was modeled using a Use Case Diagram (UML), clearly defining system boundaries, external actors, and key behavioral relationships.

This foundation establishes the scope and the critical architectural drivers that will be addressed in subsequent labs, particularly the design of the Layered Architecture in next Lab 2.

2. Lab Specific Section: I. Requirements Elicitation & Modeling

2.1. Software Requirements Specifications (SRS)

2.1.1. Identify Actors

- **Web Customer:** Users who access the website or application to search for movies, view screening schedules, and book tickets.
- **Cinema Manager:** The individual responsible for updating new movie listings, scheduling screenings, and managing theater auditoriums.
- **Payment Gateway:** An external system that securely processes financial transactions.

2.1.2. Functional Requirements (FRs)

	Requirements Name	Description
FR-01	Search & Browse Movies	The system shall allow customers to search for movies by title, genre, or theater, and display lists of "Now Showing" or "Coming Soon" films
FR-02	View Screening Details	Customers shall be able to view detailed information regarding showtimes, auditoriums, and available seat counts for a specific movie

FR-03	Visual Seat Selection	The system must display a real time seating chart of the auditorium, allowing customers to select available seats
FR-04	Booking & Payment (Checkout)	The system shall process ticket bookings, including ticket selection, the application of discount codes (if applicable), the purchase of concessions, and integration with online payment processing
FR-05	Booking History Management	Upon successful payment, the system must store the electronic ticket (QR Code) in the customer's personal profile to facilitate ticket verification at the theater

2.1.3. Non-Functional Requirements (NFRs)

	Requirements Name	**Description**
NFR-01	Performance	The system must respond to search queries and load seating charts within 2 seconds, even under a load of 5,000 concurrent users
NFR-02	Availability	The system must operate continuously 24/7, achieving an uptime of 99.9%, particularly during weekends and holidays
NFR-03	Security	All payment information and user passwords must be encrypted; furthermore, the system must comply with payment security standards

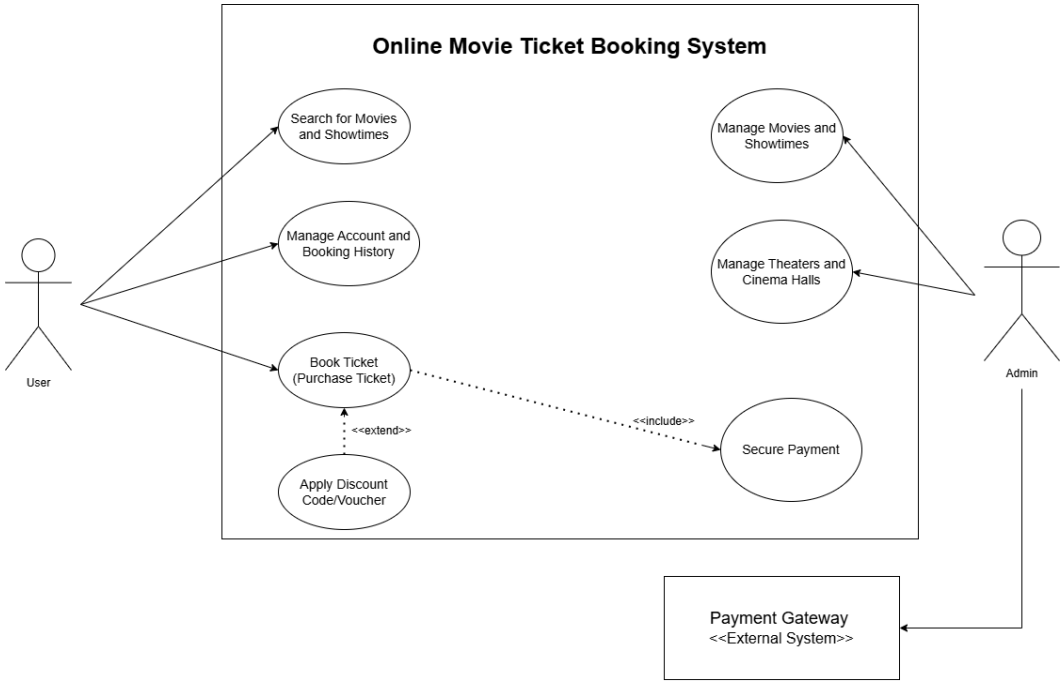
2.1.4. Identify Architecturally Significant Requirements (ASRs)

	Requirements Name	**Description**

ASR 1	High Scalability	**Statement:** The system must withstand traffic spikes of up to 10 times the normal load without failure during ticket sales for "blockbuster" movies. **Impact:** This requirement necessitates the adoption of a Microservices architecture to allow independent scaling of the Booking/Payment modules without affecting other modules. Additionally, Load Balancing and Caching mechanisms must be employed to reduce database load.
ASR 2	Data Consistency	**Statement:** The system must strictly prevent two customers from successfully booking the same seat at the same time. **Impact:** This requires the use of Distributed Locking mechanisms or strict Transaction management (ACID) within the database when processing seat status.
ASR 3	Interoperability	**Statement:** The system must integrate seamlessly with multiple different payment gateways and allow for the easy addition of new gateways. **Impact:** This requires designing with the Adapter Pattern or utilizing a separate Payment Service so that changing payment partners does not alter the core code of the booking system.

2.2. Modeling Artifact: UML Use Case Diagram

The UML Use Case Diagram below models the functional scope of the **Movie Ticket Booking System**.



- **Actors:** The system interacts with three primary actors:
    + **User:** Represents the typical web customer who uses the system to find and purchase tickets.
    + **Admin:** Responsible for the operational management of the cinema inventory.
    + **Payment Gateway <<External System>>:** An external system necessary for processing financial transactions.
- **System Boundary:** The main rectangle delineates the functional scope of the Online Movie Ticket Booking System.
- **User Interactions:** Interacts with the system to Search for Movies and Showtimes, Manage Account and Booking History, and Book Ticket (Purchase Ticket).

- **Admin Interactions:** Responsible for content and venue management through Manage Movies and Showtimes and Manage Theaters and Cinema Halls.
- **Payment Gateway Interactions:** Interacts with the Secure Payment Use Case to validate card information and process financial transactions requested by the system during the booking process.

3. Architectural Design

3.1. The Problem Statement

The core architectural challenge is designing a system that satisfies the high throughput requirement (ASR 1) while guaranteeing atomic operations for seat reservation (ASR 2). A traditional, monolithic architecture would easily become a bottleneck when handling the concurrent user load expected during peak ticket sales, inevitably leading to catastrophic double booking scenarios. Therefore, the architecture must prioritize scalability and transaction integrity in the core booking engine.

3.2. Impact of ASRs on Layered Architecture

- The architecture is explicitly shaped by the identified ASRs. The requirement for High Scalability (ASR 1) necessitates the adoption of a Microservices Architecture, allowing the critical Booking Service and Payment Service to be scaled horizontally and independently to handle traffic spikes up to ten times the normal load.
- Furthermore, Data Consistency (ASR 2) is paramount, driving the design to incorporate Distributed Locking mechanisms and strict ACID Transactions to prevent double booking.
- Finally, Interoperability (ASR 3) requires the isolation of third-party dependencies through the Adapter Pattern within a specialized Payment Service, ensuring the core system remains stable when integrating new payment gateways.

4. Conclusion & Reflection

The requirements elicitation phase for Movie Ticket Booking System successfully defined the project scope through detailed Functional, Non-Functional, and Architecturally Significant Requirements. The UML Use Case Diagram provides a clear visual model of user interactions and the critical path.

The identified ASRs—especially those relating to Modifiability and Scalability—directly necessitate the adoption of a structured pattern like the Layered Architecture for the subsequent design phase (next Lab 2), ensuring the architecture can sustainably support the required system behaviors.