Practical No-04

Aim:

Modeling UML Use Case Diagrams and Capturing Use Case Scenarios

Project: Movie Recommendation System

🧪 Aim of the Experiment

To model the functional requirements of a Movie Recommendation System by identifying users (actors), system operations (use cases), and their relationships. This will be represented using a UML Use Case Diagram, which helps stakeholders understand the system clearly.

📘 Introduction

A Use Case Diagram is a type of UML diagram used to represent how users interact with a system. It provides a high-level view of system functionality. Each actor has one or more goals represented as use cases, and the system facilitates achieving those goals.

For your Movie Recommendation System, this means capturing what actions users like Registered Users, Guests, and Admins can perform—like rating movies, getting recommendations, or managing movie data.

✅ Objectives

After completing this practical, you will be able to:

Identify different actors and use cases in the Movie Recommendation System.

Define the relationships (association, include, extend) between actors and use cases.

Draw a complete UML Use Case Diagram.

Understand how to capture system requirements using modeling.

📚 Theory & Explanation

🔸 Use Case Diagrams

These diagrams depict:

Actors (users or external systems interacting with your system)

Use Cases (system functionalities)

Relationships (associations between actors and use cases, includes, and extends)

🔸 Actors in the Movie Recommendation System

Registered User

Can log in, rate movies, receive personalized recommendations, search and add to watchlist.

Guest User

Can browse and search for movies, but can't rate or get personalized recommendations.

Admin

Manages the movie database, user profiles, and system performance.

Recommendation Engine (Supporting Actor/System)

Processes user data to generate recommendations using algorithms.

🔸 Use Cases (System Functionalities)

Here are the main actions or services your system will support:

Actor Use Case

Registered User Register, Login, Browse Movies, Rate Movie, View Movie Details, Add to Watchlist, Get Personalized Recommendations

Guest User Browse Movies, View Movie Details

Admin Login, Add/Remove Movies, Manage Users, View Reports

System Recommendation Engine (Suggest Movies)

🔄 Use Case Relationships

Include Relationship

Rate Movie → includes → Login

Get Recommendations → includes → Rate Movie

Extend Relationship

Add to Watchlist → extends → Browse Movies

Update Profile → extends → Login

🖼 Graphical Representation – UML Use Case Diagram

You can use any tool (Draw.io, Lucidchart, Creately, StarUML) to create the diagram. It should include:

Stick figures for actors

Ovals for use cases

Straight lines to show interactions

<<include>> and <<extend>> stereotypes for relationships

🔍 Case Study Scenario

Here’s how your Movie Recommendation System might work in practice:

A Registered User logs in, rates a few movies, and views their personalized recommendations.

A Guest User browses the catalog but cannot rate or get tailored suggestions.

An Admin logs in to upload a new movie, remove outdated titles, or check user activity logs.

The Recommendation Engine uses the data (movie ratings, preferences, watch history) to suggest movies using collaborative filtering or content-based filtering algorithms.

🧾 References

UML Distilled by Martin Fowler

Object-Oriented Analysis and Design with Applications by Grady Booch

Lucidchart UML Tool Documentation

StarUML or Draw.io for diagram creation

Project-specific logic based on your recommendation algorithm (Collaborative Filtering, Content-Based, Hybrid)

✅ Bonus: Tips for a Great Use Case Diagram

Keep it simple but complete.

Focus on what the system does (not how).

Group related use cases if needed (like all admin tasks together).

Clearly show which user can perform which action.