

## Activity 1: Defining Layer Responsibilities

**Goal:** Formally define the four layers and their roles within the Movie Management System structure.

Layer	Purpose/Responsibility	Output/Artifact
<b>1. Presentation Layer (UI)</b>	Handles HTTP requests (e.g., view movie, register), manages user authentication, and renders responses (JSON/HTML). It delegates complex logic to the layer below.	<b>Controllers</b>  <i>(e.g., MovieController, SubscriptionController, AuthController)</i>
<b>2. Business Logic Layer (Service/Domain)</b>	Contains core business rules and validation logic.  <b>Key Logic:</b> Checks if a movie is Premium or Free; validates if a user has an active subscription plan before granting access.	<b>Services</b>  <i>(e.g., MovieService, SubscriptionService, NotificationService)</i>
<b>3. Persistence Layer (Data Access)</b>	Responsible for abstracting database interactions. Maps Java objects to database records and executes CRUD (Create, Read, Update, Delete) operations.	<b>Repositories</b>  <i>(e.g., MovieRepository, SubscriptionRepository, UserRepository)</i>
<b>4. Data Layer</b>	The physical storage system where the application's data resides.	<b>Database Schema</b>  <i>(MySQL Tables: Movies, Users, Subscriptions, Notifications)</i>

## Activity 2: Component Identification (Feature: View Movie Details)

**Goal:** Break down the "View Movie Details" feature into concrete components that reside in the top three layers.

### 1. Identify Components

- **Layer 1 (Presentation):**
  - **Component Name:** MovieController
  - **Responsibility:** Receives the HTTP GET request (e.g., GET /movies/{id}). It extracts the Movie ID and the current User from the session, then calls the Business Logic Layer to retrieve data.
- **Layer 2 (Business Logic):**
  - **Component Name:** MovieService
  - **Responsibility:** Implements the **access control logic**.
    - It receives the Movie ID and User info.
    - It checks: *Is the movie Premium? AND Does the user have a valid subscription?*
    - If access is allowed, it calls the Persistence Layer. If not, it throws an AccessDeniedException.
- **Layer 3 (Persistence):**
  - **Component Name:** MovieRepository
  - **Responsibility:** Constructs and executes the database query (e.g., SELECT \* FROM movies WHERE id = ?) to fetch the specific movie entity and returns it to the Service layer.

### 2. Define Interfaces

- **Interface provided by Business Logic (for Presentation Layer):**
  - IMovieService
  - **Method Signature:** public MovieDTO getMovieDetail(int movieId, User currentUser)
- **Interface provided by Persistence (for Business Logic Layer):**
  - IMovieRepository
  - **Method Signature:** public MovieEntity findById(int movieId)

## Activity 3: Component Diagram Modeling

**Goal:** Create a visual model of the Layered Architecture for the Movie Management System using a UML Component Diagram.

