

*Online Movies Content Management System*  
*Dr. Mark Austin – System Engineering*  
*ENPM 643*

Submitted By  
Aditya Badhwar

# Presentation Agenda

---

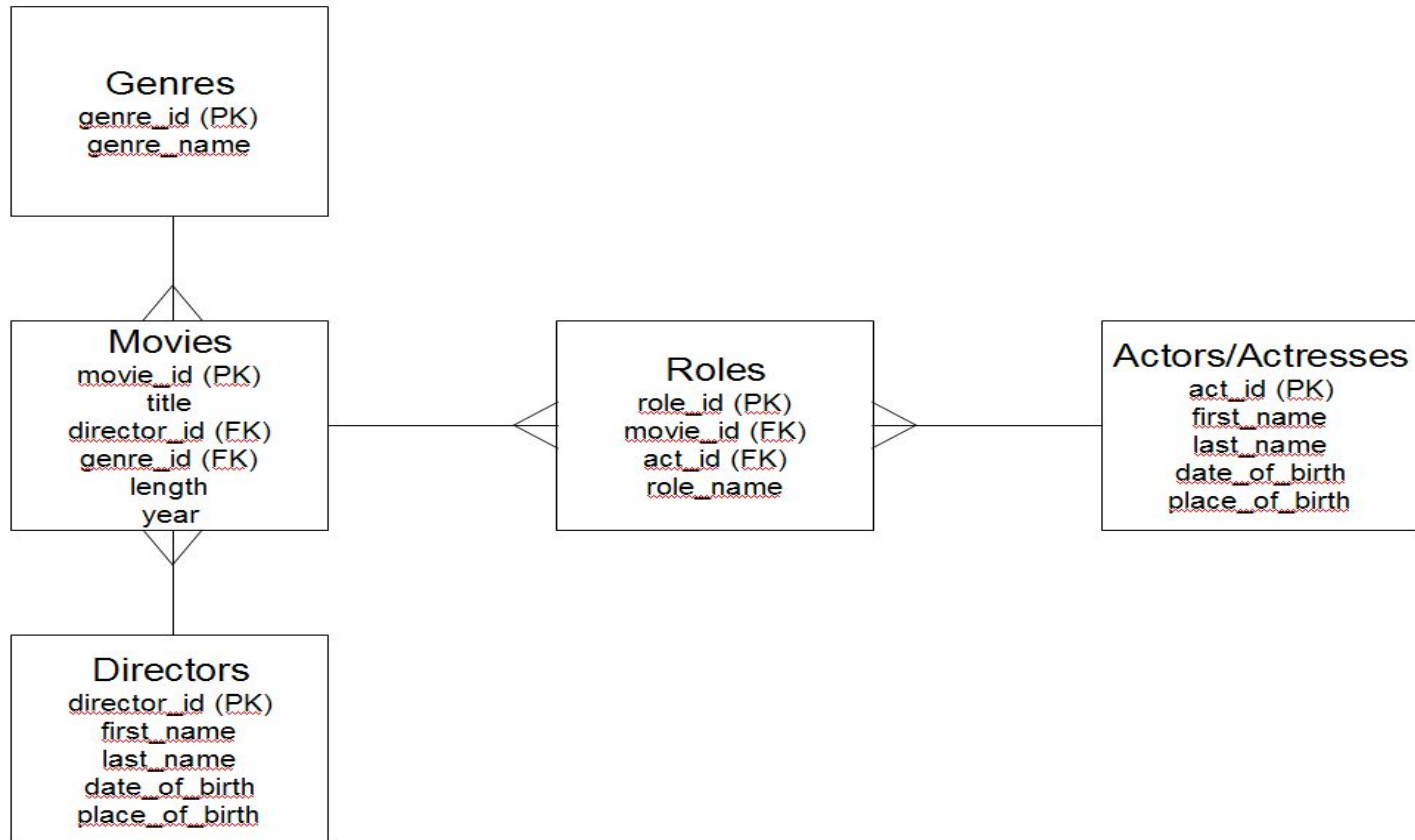
- ☐ Introduction
- ☐ SDLC Model/Agile Unified Process
- ☐ Goals
- ☐ UML Diagrams
  - Communication Diagram
  - Sequence Diagram
  - Use Case Diagram
- ☐ Validation and Verification
- ☐ Software Testing (Dynamic/Static)
- ☐ Software Validation tools (Load Runner / Win Runner)

# Introduction

---

- The Online Movies content management system is a Web based system used for storing,controlling,publishing media industry specific data. The idea behind a CMS is to make these files available inter-office, as well as over the web
  - =>The ability to control content through a web interface, irrespective of the web browser
  - =>Identification of all key users and roles specific to users
  - =>The ability to track changes made by users
  - =>Record the changes and undo them if desired

# ER Diagram



# SDLC Models

---

- Waterfall
- Agile (Agile unified process)
- RUP (Rational unified process)
- Iterative
- RAD (Rapid application development)
- Spiral
- XP (Extreme programming)

# Agile Unified Process (AUP)

---

**Agile software development** is a conceptual framework for software engineering that promotes development iterations throughout the life-cycle of the project

- Minimize risk by developing software in short amounts of time
- Software developed during one unit of time is referred to as an iteration
- Each iteration is an entire software project: including planning, requirements analysis, design, coding, testing, and documentation
- An iteration may not add enough functionality to warrant releasing the product to market but the goal is to have an available release (without bugs) at the end of each iteration

# The Agile Manifesto

---

- ❖ Customer satisfaction by rapid, continuous delivery of useful software
- ❖ Working software is delivered frequently (weeks rather than months)
- ❖ Working software is the principal measure of progress
- ❖ Even late changes in requirements are welcomed
- ❖ Close, daily cooperation between business people and developers
- ❖ Face-to-face conversation is the best form of communication
- ❖ Projects are built around motivated individuals, who should be trusted
- ❖ Continuous attention to technical excellence and good design
- ❖ Simplicity
- ❖ Self-organizing teams

# AUP comprises four phases

---

## 1. **Inception**

Identify the initial scope of the project, a potential architecture for the system, and obtain initial project funding and stakeholder acceptance.

## 2. **Elaboration**

Prove the architecture of the system.

## 3. **Construction**

Build working software on a regular, incremental basis which meets the highest-priority needs of project stakeholders.

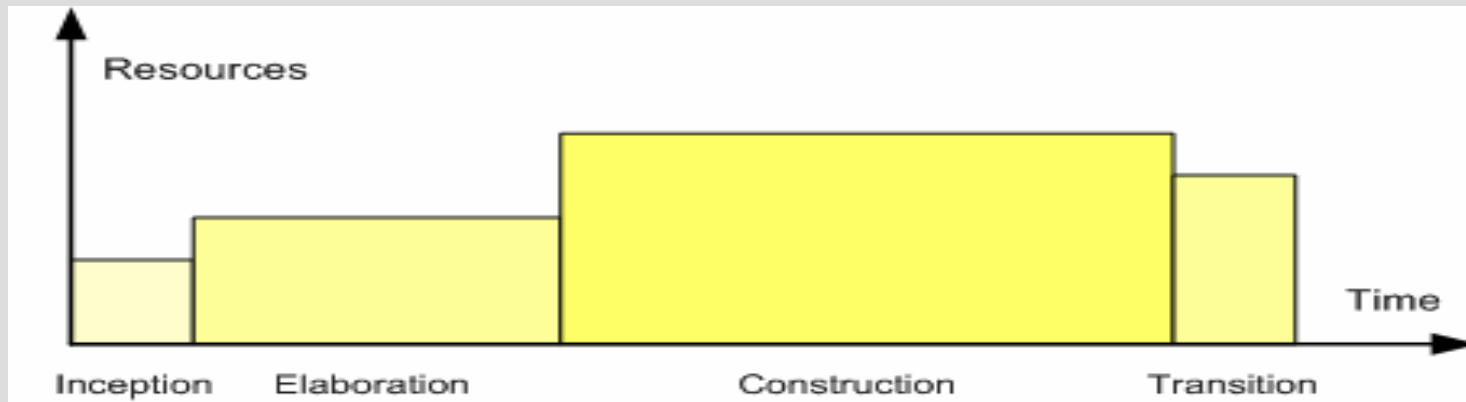
## 4. **Transition**

Validate and deploy the system into the production environment.

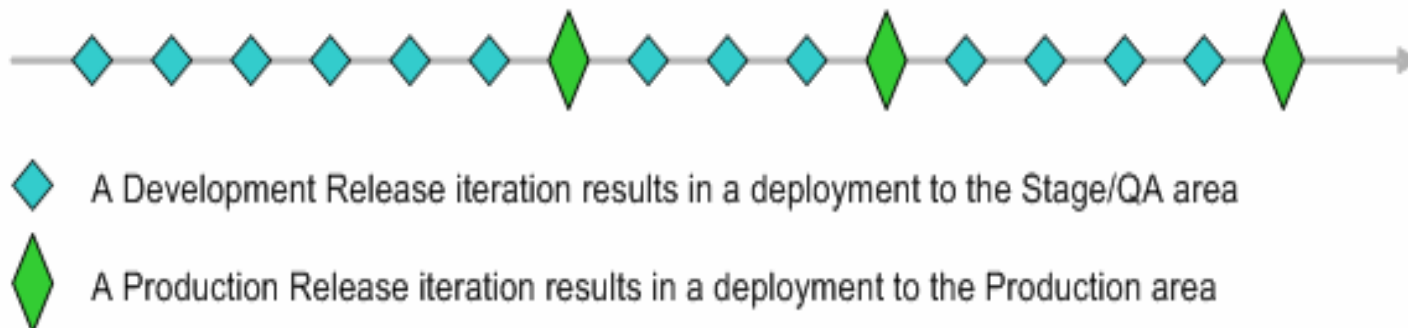


# AUP Model

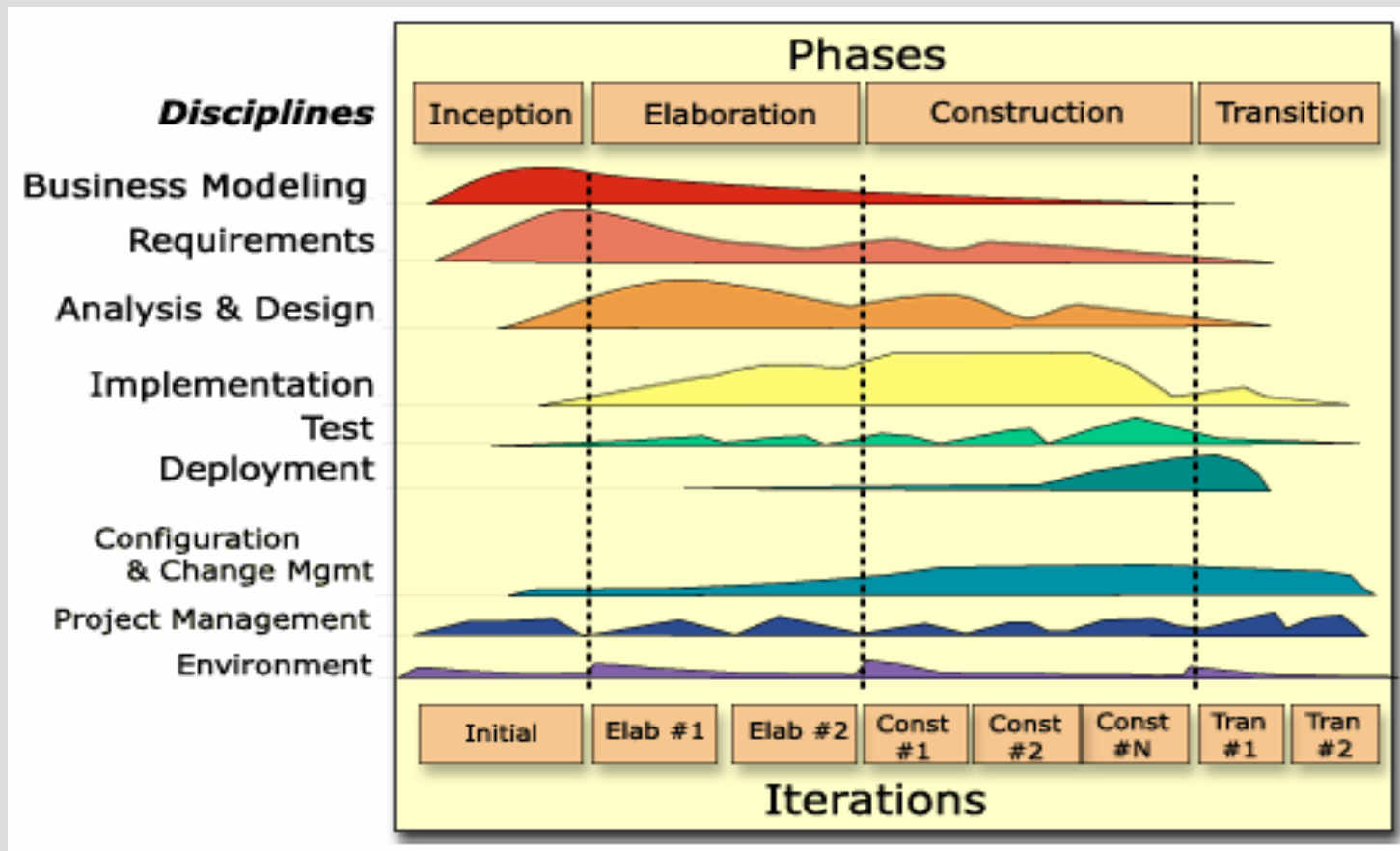
---



## Project Timeline



# AUP Characteristics

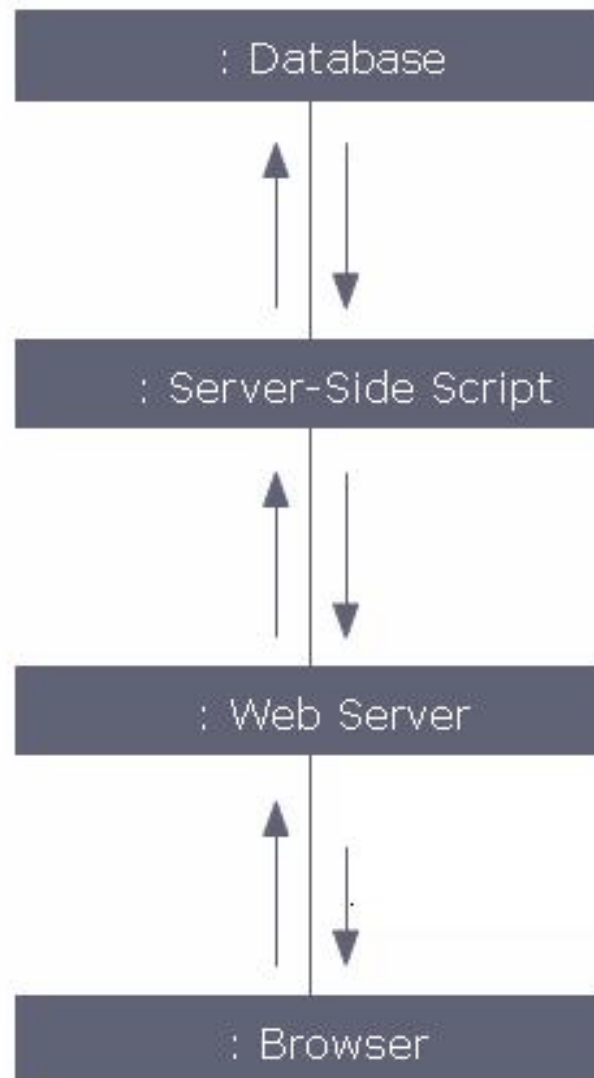


# Goals

---

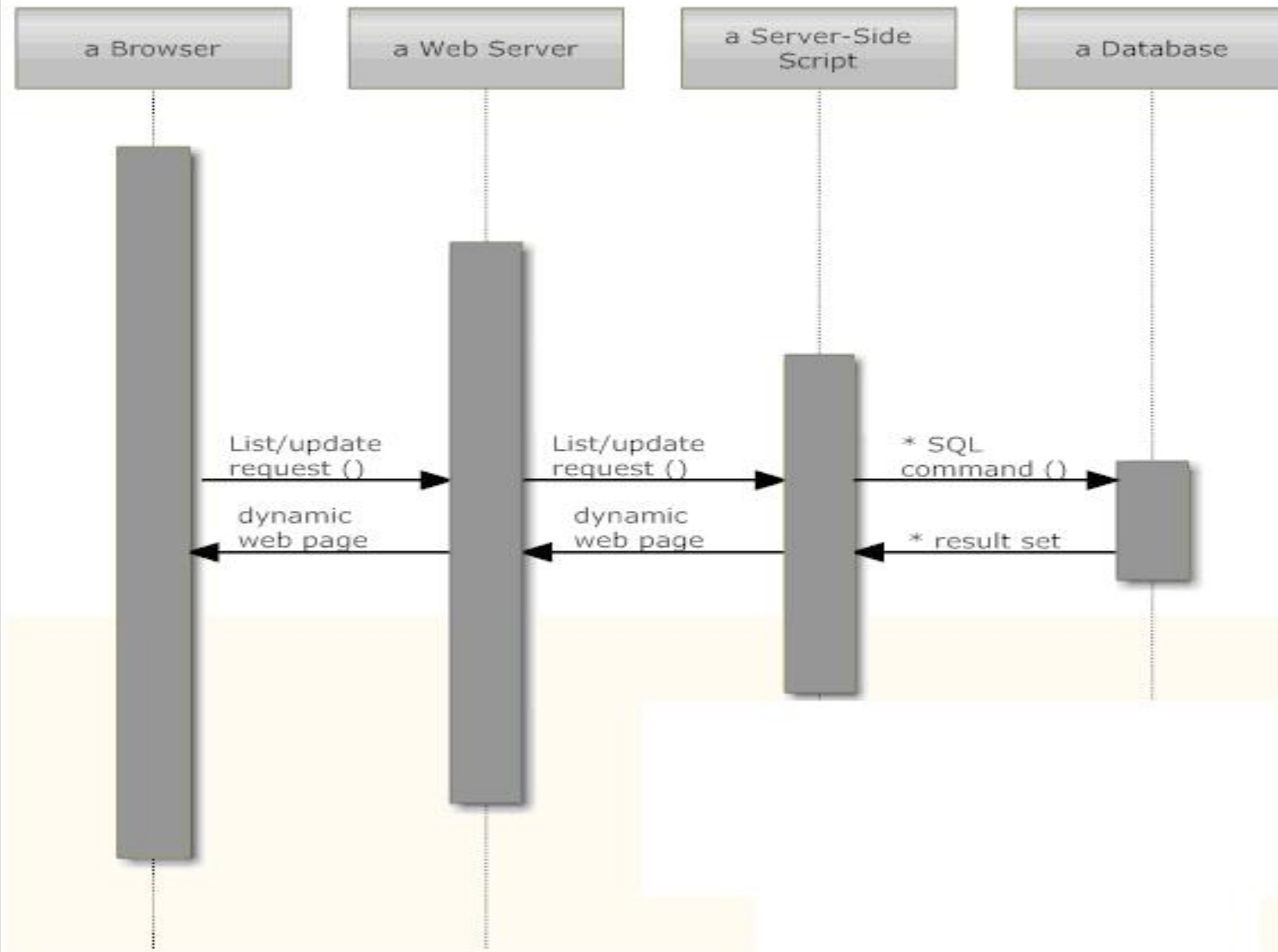
- Goal1:The system shall permit the users to read and display the data
- Goal2:The System shall authenticate the user and set the access rights to the application
- Goal3:The system shall permit the administrator to add, delete and update records
- Goal4:The system shall permit the staff to have view, sort functionality for all modules
- Goal5:The system shall permit external users to view movie records and there roles

## UML Communication Diagram

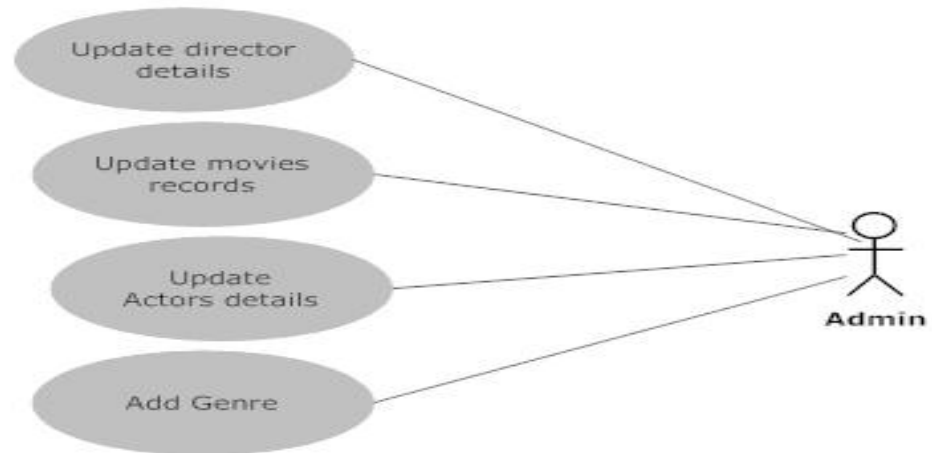


# UML Sequence Diagram

## PHP Server page

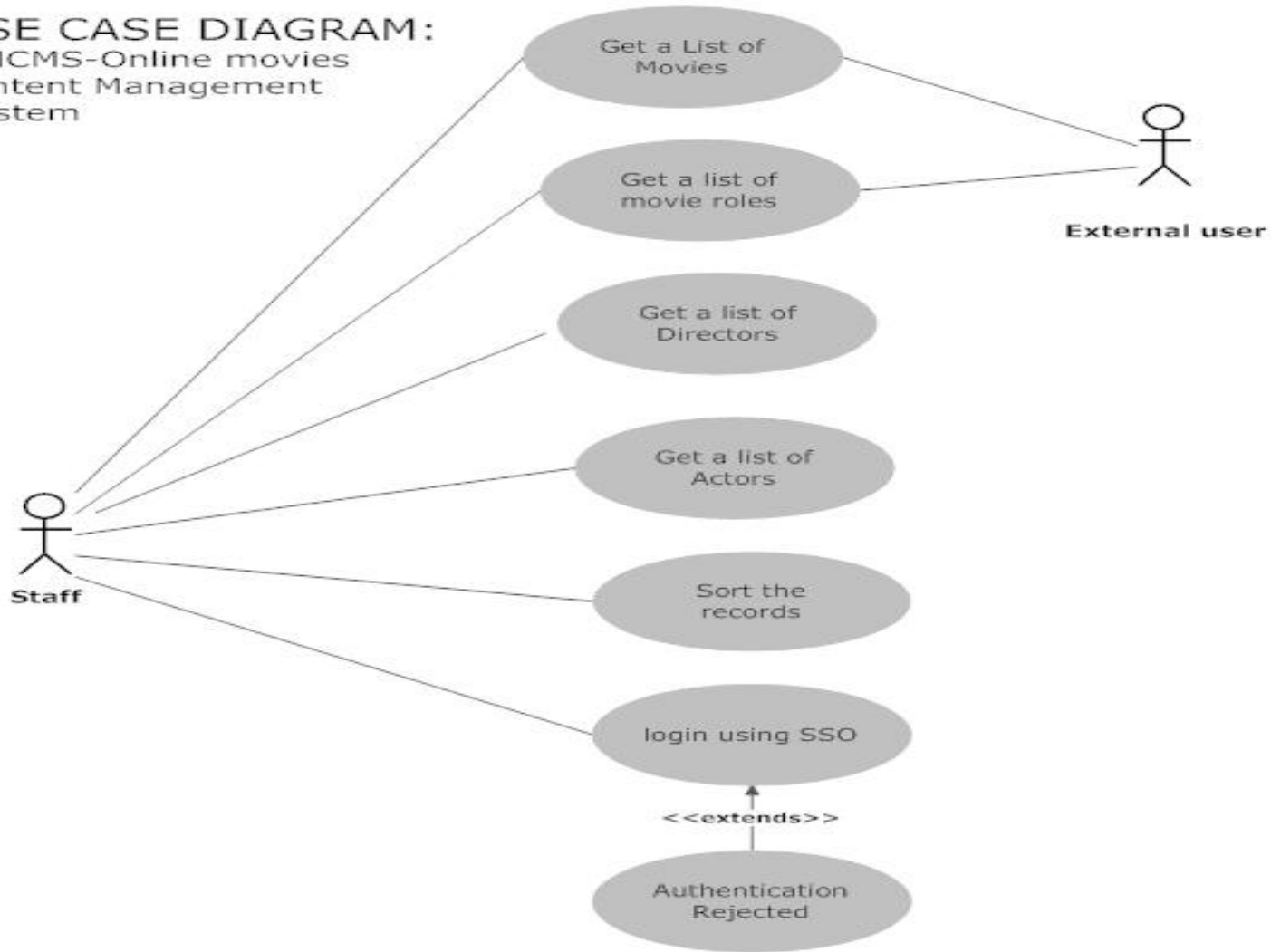


**USE CASE DIAGRAM:**  
OMCMS-Online movies  
content Management  
System



# USE CASE DIAGRAM:

OMCMS-Online movies  
content Management  
System

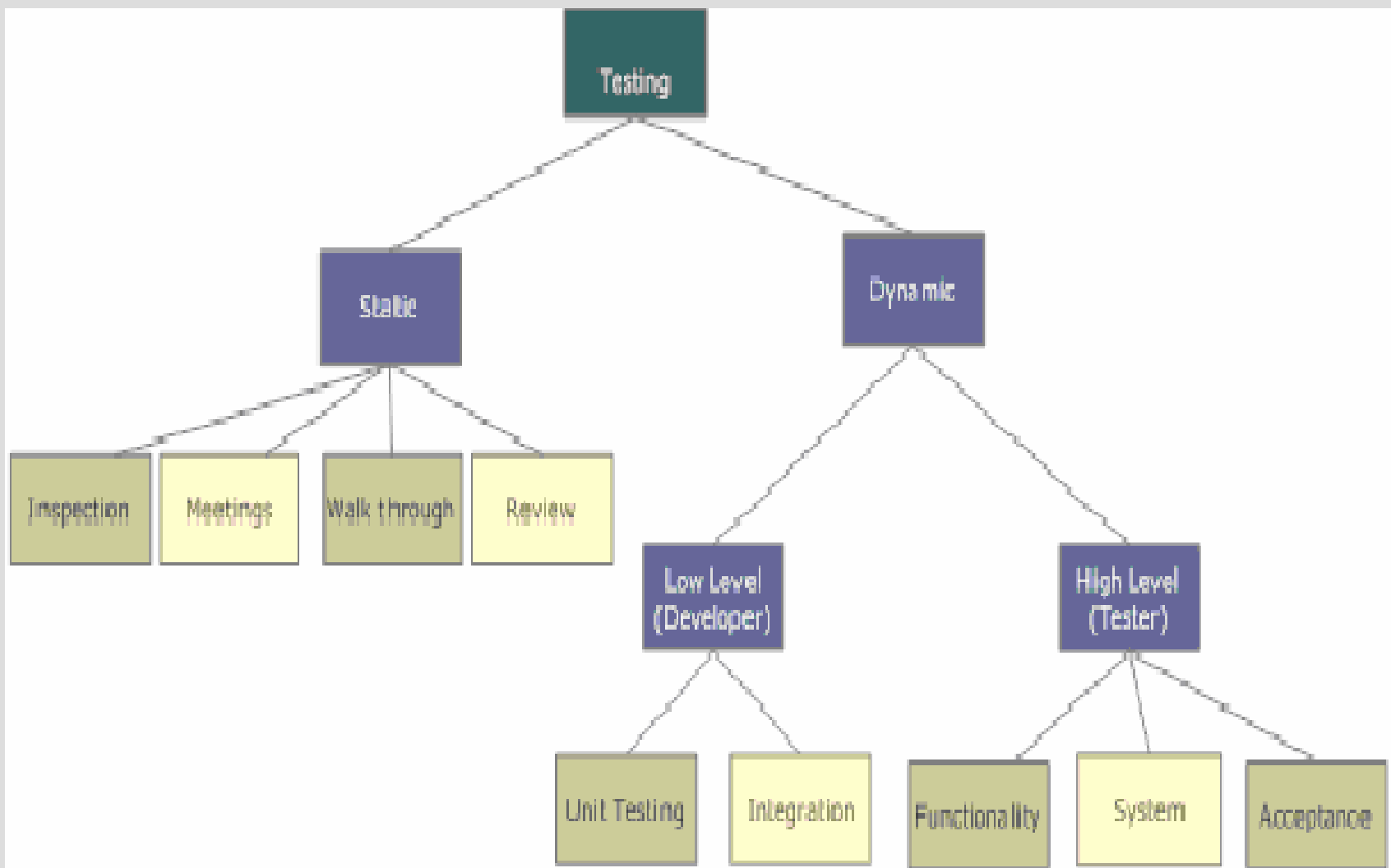


# Validation and Verification

---

- ❑ **Verification** ensures that the final product satisfies or matches the original design (low-level checking) — i.e., you built the product right. This is done through **static testing**.
- ❑ **Validation** checks that the product design satisfies or fits the intended usage (high-level checking) — i.e., you built the right product. This is done through **dynamic testing** and other forms of review





Static Testing	Dynamic Testing
<ul style="list-style-type: none"> <li>1) Done by a Senior</li> <li>2) Verification of the software is done "Are we Building the product Right"</li> <li>3) It is Process oriented</li> <li>4) Quality assurance is given</li> </ul>	<ul style="list-style-type: none"> <li>1) Done by testers and Developers</li> <li>2) Validation of Software is done "Are We Building the Right product"</li> <li>3) It is Product Oriented</li> <li>4) Quality Control</li> </ul>

# Dynamic Testing

---

- ❑ 1. White Box Testing
- ❑ 2. Black Box
- ❑ 3. Gray Box Testing

# White Box Testing

---

- ❑ Check that all the conditions have been executed at least once
- ❑ Check all the loops are working to their boundaries
- ❑ Check all the Boolean values are on their true/false side
- ❑ Check the internal data structure of the software

# Black Box Testing

---

- ❑ Check the overall functionality of the application
- ❑ Check the external databases and interfaces of the application (Links in WebPages)
- ❑ Also known as Behavioral testing, as it checks the behavior of the application on the basis of SRS

# Win Runner

---

- ❑ Automated Testing
- ❑ Used for Functionality and Regression testing
- ❑ Works on record and playback method
- ❑ Generates a TSL(Test script language) on the basis of recorded actions
- ❑ TSL can be changed as per test cases and run to view results

```
WinRunner - [C:\Users\Guest\director]
File Edit Create Run Debug Tools Settings Window Help
Verify

# Mozilla Firefox Start Page - Mozilla Firefox
set_window ("Mozilla Firefox Start Page - Mozilla Firefox", 2);
obj_mouse_click ("MozillaWindowClass_0", 977, 44, LEFT);

# Shell_TrayWnd
set_window ("Shell_TrayWnd", 1);
toolbar_button_press ("ToolbarWindow32_1", "WinRunner - [Noname2*]"); # Button Number 2;

# Task Switching
win_activate ("Task Switching");

# University of Maryland Central Authentication Service - Mozilla Firefox
win_activate ("University of Maryland Central Authentication Service - Mozilla Firefox");
set_window ("University of Maryland Central Authentication Service - Mozilla Firefox", 1);
obj_type ("MozillaWindowClass_3", "abadhwar<kTab>A@12ditya<kReturn>");

# Confirm
set_window ("Confirm", 1);
obj_type ("MozillaWindowClass", "<kReturn>");

# Welcome to the Movies Databases - Mozilla Firefox
set_window ("Welcome to the Movies Databases - Mozilla Firefox", 10);
obj_mouse_click ("MozillaWindowClass_4", 408, 75, LEFT);
obj_mouse_click ("MozillaWindowClass_4", 15, 398, LEFT);
obj_mouse_click ("MozillaWindowClass_4", 871, 74, LEFT);
obj_mouse_click ("MozillaWindowClass_4", 122, 171, LEFT);
obj_type ("MozillaWindowClass_3", "sanjay<kTab>bansali<kTab>1944~-11~-23<kTab>punjab");
obj_mouse_click ("MozillaWindowClass_4", 53, 321, LEFT);

# Security Warning
set_window ("Security Warning", 2);
obj_mouse_click ("MozillaWindowClass", 255, 122, LEFT);

# IMDb Movies Menu - Mozilla Firefox
set_window ("IMDb Movies Menu - Mozilla Firefox", 2);
obj_mouse_click ("MozillaWindowClass_4", 409, 81, LEFT);
obj_mouse_click ("MozillaWindowClass_4", 15, 400, LEFT);
obj_mouse_click ("MozillaWindowClass_4", 27, 415, LEFT);
```

# Load Testing using LoadRunner

---

- ❑ Load-The Degree to verify how a application performs under limited number of users for a specific period of time
- ❑ Stress-Exceeding beyond the maximum load bearing capacity
- ❑ Load runner uses vusers (Virtual users) to test a application



# Project Report

---

- ☐ Verification Traceability Matrix-Linking the requirements to tests performed
- ☐ Assembly of System architecture
- ☐ Detailed Requirements  
Level 1,Level 2,Level3

# References

---

- ❑ Lecture Notes for ENSE 623 by Dr. Mark Austin
- ❑ System Analysis and Design with UML by Alan Dennis
- ❑ PHP and MYSQL by Larry Ullman
- ❑ [www.wikipedia.org](http://www.wikipedia.org)
- ❑ [https://h10078.www1.hp.com/cda/hpms/display/main/hpms\\_content.jsp](https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp)

# Questions?

---

