# .NET Core Azure

# Sprint-1

## Film Management System

## Document Revision History

| Date | Revision No. | Author | Summary of Changes |
|------|--------------|--------|--------------------|
|      |              |        |                    |
|      |              |        |                    |
|      |              |        |                    |
|      |              |        |                    |
|      |              |        |                    |

# Table of Contents

**Capgemini Public**

# INTRODUCTION

This document outlines a project for the .NET Line of Technology (LOT). The project is to develop Film Management System. This document contains the requirements, work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the .NET LOT.

## SETUP CHECKLIST

### Minimum System Requirements

- Intel Pentium 4 and Windows 2010
- Memory 4 GB
- Internet Explorer 11.0 or higher / Chrome
- SQL Server 2014 or 2016 client and access to SQL Server 2014 or 2016 server
- Visual Studio 2019
- Visual Studio Code
- Git

## INSTRUCTIONS

- The code modules in the mini project should follow all the coding standards.

# PROBLEM STATEMENT

## OBJECTIVE

Need to develop Film Management System.

**Abstract of the project:**

**Film Management System (FMS)** is an application, which helps to manage an actor details like – personal and professional info of actor. Personal info like First Name, Last Name etc. And professional info like –Films, Film Category, Film Rating etc.

FMS aims to manage complete life cycle of film viz add (create a film – where all info will be captured for the first time), edit (modify film info) – like roles, films, etc. Remove (delete film) film data from system. Along with add, edit and remove – system should display all actor summaries (View) and must support search facility.

**Phase 1:** Create Client Application in Angular and Services using ASP.NET Core Web API.

**Macro level Operations/offerings:**

1. Create Film (capture film info)

2. Modify Film info

3. Remove Film

4. View all Film (Film summary)

5. Search Film by name

6. Search Film by Actor

7. Search Film by Category

8. Search Film by Language

9. Search Film by Rating

**MODULE LIST and MODULE DETAILS**

**CREATE FILM**

Following info need to capture

- Film title (Must be Unique)

- Film description

- Release year

- Rental duration

- Rental rate

- Length

- Replacement cost

- Rating

**SEARCH FILM**

User should be able to search a film by film title, release year, rating User may use any one, two or all the three parameters to search a film.

**MODIFY FILM**

Search (film title, release year, rating) film and modify the info of film. System should show existing data/info of film and should support modify of one, more or all info.

**REMOVE FILM**

Search (film title, release year, rating) and remove the film. System should ask for confirmation and on confirmation the data will be removed.

**FILM SUMMARY (VIEW)**

System should show (display) film list in a tabular format (one row for each film, and columns for film info). It is not required to show all the data of film in table; only important info likes – film title, release year, rating. project is needs to show in table.

**Film Report**

1. Search Film by name

2. Search Film by Actor

3. Search Film by Category

4. Search Film by Language

5. Search Film by Rating

**Constrains**

- Proper validation is required (specially film title, release year, rating) System must check uniqueness on film title, release year, rating.

- System must show appropriate massage on all activity (whether activity is successful or failure)

- User must have proper menu to select the activity (create, modify, search, view, remove) that user want to perform.

Create the following database tables with following fields: [make your assumptions in case you require few more fields]

**Table Name – FILM**

| Field name | Constrains | Data type |
|---|---|---|
| Film  id | Primary key (system generated) | Number |
| Description | | Text |
| Title | | Text |
| Release year | | Date |
| Language id | Foreign Key | Number |
| Original language id | Foreign Key | Number |
| Rental Duration | | Date |
| Length | | Number |
| Replacement cost | | Number |
| Rating | | Number |
| Special Features | | Text |
| Actor id | Foreign Key | Number |
| Category id | Foreign Key | Number |

**Table Name – ACTOR**

| Field name | Constrains | Data type |
|---|---|---|
| Actor id | Primary key (system generated) | Number |
| First Name | | Text |
| Last Name | | Text |
| Film id | Foreign Key | Number |

**Table Name – CATEGORY**

| Field name | Constrains | Data type |
|---|---|---|
| Category id | Primary key (system generated) | Number |
| Name | | Text |
| Film id | Foreign Key | Number |

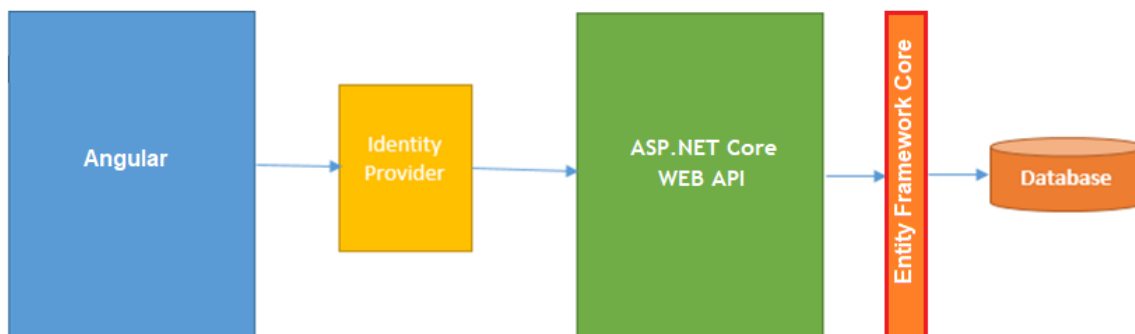| Table Name – LANGUAGE | | |
|---|---|---|
| **Field name** | **Constrains** | **Data type** |
| Language id | Primary key (system generated) | Number |
| Name | | Text |
| Film id | Foreign Key | Number |

## PROJECT STRUCTURE

**Description:** Create below sample reference project structure, which will help to reuse most of module for Web application. You are allowed to bring your own project structure and project to achieve the requirement.

In this sprint you have to use

- Angular UI as the presentation layer
- ASP.NET Core WEB API to create services
- Entity Framework Core to interact with the database.

The project should have minimum target framework – .Net Core 3.1 for Core WEB API Services and Angular 10 for UI Design



Design guidelines

- All the exceptions/errors to be captured and user friendly message to be displayed on the Common Error page.

## IMPLEMENTATION

### SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop the Film Management System by building the functionality incrementally in each of the course modules of .NET LOT.

| Sr. No | Course | Duration (in PDs) | Functionality to be built |
|--------|--------|-------------------|---------------------------|
| 1 | Angular ASP.NET Core Web API Entity Framework Core SQL Server | 5 | Developing Presentation components (Angular), Business components (ASP.NET Core WEB API) and Data access components (Entity Framework Core) |