|  |  |
| --- | --- |
|  |  |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** | Devi K V |  |  | | **Role** |  |  |  | | **Signature** |  |  |  | | **Date** |  |  |  | |
|  |

Table of Contents

[1.0 Introduction 3](#_Toc23440118)

[1.1 Purpose of this document 3](#_Toc23440119)

[1.2 Definitions & Acronyms 3](#_Toc23440120)

[1.3 Project Overview 3](#_Toc23440121)

[1.4 Scope 3](#_Toc23440122)

[1.5 Target Audience 3](#_Toc23440123)

[1.6 Hardware and Software Requirement 3](#_Toc23440124)

[1.6.1 Hardware Requirements 3](#_Toc23440125)

[1.6.2 Software Requirements 4](#_Toc23440126)

[2.0 Functional Requirements 4](#_Toc23440127)

[2.1 Functional Requirements 4](#_Toc23440128)

[2.3 System Architecture Diagram 4](#_Toc23440130)

[3.0 Design Specification 5](#_Toc23440131)

[3.1 Pre-requisite Steps 5](#_Toc23440132)

[3.2 UI Design 5](#_Toc23440133)

[3.3 Data Design 5](#_Toc23440134)

[3.4 Component Design for identified Use cases 5](#_Toc23440135)

[3.5 General Design Constraints 6](#_Toc23440136)

[3.6 Requirements to Design Traceability 6](#_Toc23440137)

[4.0 Submission 6](#_Toc23440138)

[4.1 Code submission instructions 6](#_Toc23440139)

[5.0 Change Log 6](#_Toc23440140)

# Introduction

## Purpose of this document

The purpose of the Document is to elaborate the functional requirements and technical design specification of the problem statement to be solved.

## Definitions & Acronyms

|  |  |
| --- | --- |
| Definition / Acronym | Description |
|  |  |
|  |  |
|  |  |
|  |  |

## Project Overview

* Building a Model using ADO.NET , WEB API

## FUNCTIONALITIES:

1. XYZ COMPANY ‘S MOVIES LIST TO BE DISPLAYED WITH ADO.NET ,WEB API , MS SQL SERVER
2. To Create a simple application for XYZ Movie production House.
   * Allows to add 2 movies Information
   * Displays existing Movies Information
   * Display a movie details by movie ID
   * Delete a Movie By ID
   * Update Existing Movie Director name by Movie ID

## Scope

* C#.NET,WEB API
* ADO.NET
* SQL SERVER.

## Target Audience

Beginners

## Hardware and Software Requirement

### Hardware Requirements

|  |  |  |
| --- | --- | --- |
| # | Item | Specification/Version |
|  | Processor | Intel® coreTM i5 or higher |
|  | RAM | 8 GB |
|  | HARD DISK | OS partitation 200GB |

### Software Requirements

|  |  |  |
| --- | --- | --- |
| # | Item | Specification/Version |
|  | Operating System | Windows 10 |
|  | Application Software | Visual Studio2019 or higher |
|  | Database Server | MS Sql Server 2014 Expression Edition |

# Functional Requirements

## Functional Requirements

*<List the high level use cases or behaviors and actors of the system. This requirement # and name must be unique and must be referred elsewhere in the document. >*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Req. # | Req. Name | Req. Description | Actors / Users | Comments |
|  |  |  |  |  |

System Architecture Diagram

* Understanding Requirements and Database Design.
* Creating Solution and Adding projects to It.
* Creating The business Objects.
* Creating Data Access Layer in
* Creating Business Logic Layer in C#.Net.
* Creating Presentation Logic Layer in PostMan
* Designing Models,BUSINESS LOGIC
* Business Rules Validations.
* Securing Your App.

# Design Specification

## Pre-requisite Steps

A Web API APPLICATION ALLOWS A USER TO MANAGE MOVIES I OR DISPLAY MOVIES DETAILS,CRUD Operations

## POSTMAN --UI

STEPS AS FOLLOWS:

* Create New Web Api  : MoviesRepoController.cs
* Add 2 classes(Movie.cs,MovieRepository.cs) to Models folder in Solution Explorer
* Add 1 interface(IMovie.cs)
* Configure Ado.NET Tech in MovieRepository.cs by implementing methods of IMovie.cs
* MoviesRepoController is connected to MovieRepository.cs
* POSTMAN (UI)

## Component Design for identified Use cases

## Component Design for identified Class Diagrams:

|  |
| --- |
| Web.CONFIG:    </entityFramework>    <connectionStrings>      <add name="SampleDB" connectionString="data source=DESKTOP-A8KSJON;initial catalog=sample;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework" providerName="System.Data.SqlClient" />    </connectionStrings>  </configuration> |

PROJECT-STRUCTURE

SCENARIO-1

|  |  |
| --- | --- |
| Type | Name |
| Class | SampleDB |
| Class | Movie |
| Class | MovieRepository |
| interface | IMovies |
| ApiController | MoviesRepoController |
|  |  |
|  |  |

CLASS METHOD DESCRIPTION:

SCENARIO-2

|  |  |  |
| --- | --- | --- |
| Class | Method-name/Properites | Description |
| Movie | MovieId | Int |
|  | MovieName | String |
|  | Actors | String |
|  | Director | String |
|  | Release\_Date | DateTime |
|  | Producer | String |
|  |  |  |

SCENARIO-3

|  |  |  |
| --- | --- | --- |
| Class/interface | Method/properties | Description |
| IMovie.cs | List<Movie>  GetMoviesList(); | Returns all existing movie details |
|  | void newMovie(Movie); | a)Takes input for all properies defined in Movie class of  B)to add  2 movies information. |
|  | Movie GetMovieById(int id); | Takes movie id as parameter returns movie record. |
|  | Void UpdateMovieById(int id); | To update movie details |
|  | Void DeleteMovieById(int id) | To delete movie |

SCENARIO-4

|  |  |  |
| --- | --- | --- |
| Class | Method/Propertiess | Description |
| * MovieRepository .cs | List<Movie>  GetMoviesList(){  // Ado.NET code here  } | Returns all existing movie details |
|  | void newMovie(Movie)  {  // Ado.NET code here  } | a)Takes input for all properies defined in Movy class of DBSet class  from user. B)to add  2 movies information. |
|  | Movie GetMovieById(int id)  {  //ADO.NET code here;  } | Display Details by id |
|  | Void UpdateMovieById(int id); | To update movie by movieid as parameter |
|  | Void DeleteMovieById(int id) | To delete a movie as movieid parameter |

MovieRepository .cs -------Component names:

|  |  |
| --- | --- |
| Type | Name |
| Class | MovieRepository.cs |
| Class | Movie.cs |
| Configuration file | Web.config |
| interface | IMovie.cs |
| ApiController | MoviesRepoController |

* **FOLLOWING STEPS WILL EXPLAIN WHAT EACH CLASS IS WILL DO FOR PROJECT**

|  |  |
| --- | --- |
| *Component Name* | MovieRepository .cs |
| *Component Definition* | *ADO.NET Code* |
| *Type* | *Class :IMovie { }* |
| *MethodName* | public List<Movie> GetMoviesList()  { |
| *Access modifers* | *public* |
| *Return type* | *List<Movie>* |
| *Interactions* | *IMovies.cs* |
| *Resources* | *Web.config* |

|  |  |
| --- | --- |
| *Component Name* | *Movie.cs* |
| *Component Definition* |  |
| *Type* | *Class* |
| *String names* |  |
| *Access modifers* | *public partial* |
| *Properties* | *int Movie\_Id*  *string Movie\_Name*  *datetime Release\_date*  *string Actors*  *string Director*  *string Producer* |
| *Interactions* |  |
| *Resources* | *---* |

STEP4:

|  |  |
| --- | --- |
| *Component Name* | MoviesRepoController *.cs* |
| *Component Definition* | *Execution* |
| *Type* | *ApiController* |
| *MethodName* | *Get(),getById(int),post(),put(),delete()* |
| *Access modifers* | *public* |
| *Return type* | HttpResponseMessage (Post,GetById,put(),Delete() |
| *Interactions* | *MoviesRepository.cs* |
| *Resources* | *----------------------* |

## General Design Constraints

## Requirements to Design Traceability

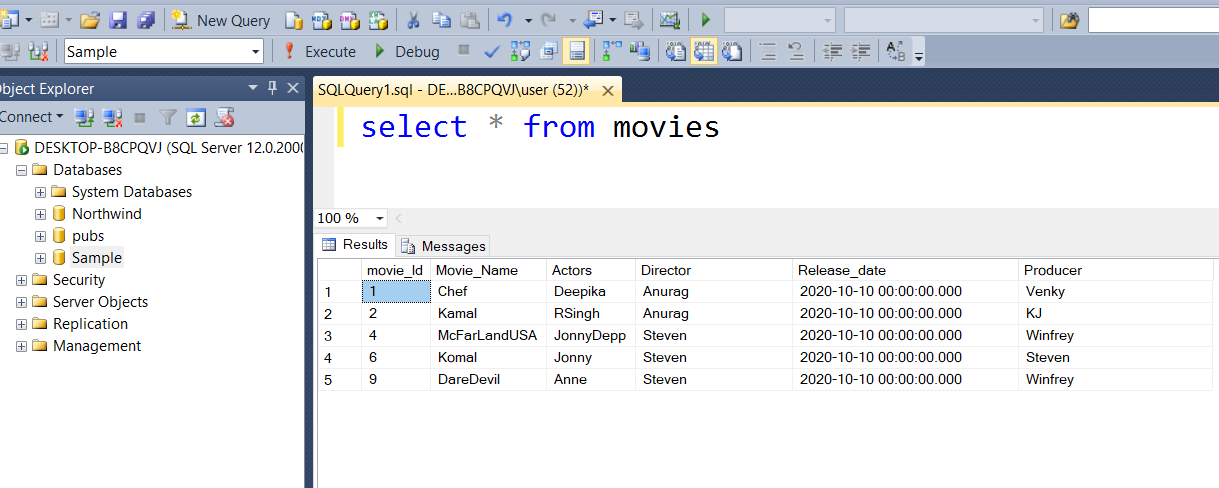
# Submission

## Code submission instructions

* **\*Note:  
  Don't change the given code Skeleton.  
  Do not include new namespaces.**

## Data Design

Database Scenario:



Add above db , into table after creating table, and connect with ado.net

Identifying The Objects(Tables)

|  |  |
| --- | --- |
| Database Name | Table name |
| Sample | Movies |

CODE-SKELETON: