**אפיון הפרוייקט**

סעיף 1:

יש ליצור טבלאות מתאימות בdb עבור שמירת נתוני הauthors וה books

* Author
* Name -string
* Age - int
* Image -string
* book
* BookName - string
* Price - decimals
* Author - number

דגשים:

* יש ליצור מפתח ראשי של id לכל טבלה - המפתח יקודם בצורה אוטומטית בכל הוספת רשומה.
* יש ליצור את ה Name של הAuthor עם הגדרה של ערך ייחודי
* יש ליצור את ה BookName של Book עם הגדרה של ערך ייחודי

סעיף 2:

יש ליצור את שכבת ה DAL על ידי קישור פתוח כך שיוכל לגשת ל - DB מסעיף 1

סעיף 3:

יש ליצור dll בשם **BO** המכיל את המחלקות הבאות:

* מחלקה המייצגת Author
* Name - (string - min 3 max 20)
* Age - (int between 18-120)
* Image - (string - min 5 chars)
* מחלקה המייצגת book
* BookName - (string - min 2 max 15)
* Price - (decimal - between 30 -200)
* Author - (Author class object)

סעיף 4:

יש ליצור dll בשם BLL המכיל קישור לDAL (מסעיף 2) ולBO (מסעיף 3 ),

בתוך הBLL יש ליצור שני מחלקות:

* AuthorManager - with crud to Author table (use EF from DAL)
* BookManager - with crud to Book table (use EF from DAL)

סעיף 5:

יש ליצור פרוייקט של consol-app המכיל קישור לBLL (מסעיף 4) ולBO (מסעיף 3),

בתוכו יש ליצור:

**the following actions:**

* Function without parameters - will return array of Author (the class is defined in the BO),

will call the AuthorManager in the BLL

The BLL will create an object of the EF from the DAL, and pass all the authors back

* Function - with id parameter (int) - will return an Aouthor (the class is defined in the BO),

will call the AuthorManager in the BLL

The BLL will create an object of the EF from the DAL, and  pass the author with the required id back

* Function - with an Aouthor parameter (the class is defined in the BO) - will return a boolean value.

will call the AuthorManger in the BLL

The BLL will create an object of the EF from the DAL

The BLL will try to add the author to the db, and return to the web api a boolean value that indicates if the action has completed successfully or not.

* Function -  with id parameter (int) and an Aouthor parameter (the class is defined in the BO) -  will return a boolean value

will call the AuthorManger in the BLL

The BLL will create an object of the EF from the DAL

The BLL will try to edit the author in the db, and then return to the web api a boolean value that indicates if the action has completed successfully or not.

* Function - with id parameter (int) - will return a boolean value

Will call the AuthorManger in the BLL

The BLL will create an object of the EF from the DAL

The BLL will try to delete the author from the db, and then return to the web api a boolean value that indicates if the action has completed successfully

**BookController - with the following actions:**

* Function - without parameters - will return array of Book (the class is defined in the BO),

will call the BookManager in the BLL

The BLL will create an object of the EF from the DAL, and pass all the books to the web-api

* Function - with id parameter (int) - will return an Book (the class is defined in the BO),

will call the BookManager in the BLL

The BLL will create an object of the EF from the DAL, and  pass the book with the required id to the web-api

* Function -  with an Book parameter (the class is defined in the BO) -  will return a boolean value.

The web-api will call the BookManger in the BLL,

The BLL will create an object of the EF from the DAL

The BLL will try to add the book to the db, and then return to the web api a boolean value that indicates if the action has completed successfully

* put -  with id parameter (int) and an Book parameter (the class is defined in the BO) -  will return a boolean value

will call the BookManger in the BLL

The BLL will create an object of the EF from the DAL

The BLL will try to edit the book in the db, and then return to the web api a boolean value that indicates if the action has completed successfully

* Function - with id parameter (int) - will return a boolean value

will call the BookManger in the BLL

The BLL will create an object of the EF from the DAL

The BLL will try to delete the book from the db, and then return to the web api a boolean value that indicates if the action has completed successfully

*create DB*

USE master

GO

CREATE DATABASE BookStore

GO

USE BookStore

GO

CREATE TABLE [dbo].[Authors] (

[AuthorID] INT IDENTITY (1, 1) NOT NULL,

[AuthorAge] INT NOT NULL,

[AuthorName] NVARCHAR (20) NOT NULL UNIQUE,

[AuthorImage] NVARCHAR (MAX) NOT NULL,

CONSTRAINT [PK\_Author] PRIMARY KEY ([AuthorID])

);

CREATE TABLE [dbo].[Books] (

[BookID] INT IDENTITY (1, 1) NOT NULL,

[BookName] NVARCHAR (15) NOT NULL UNIQUE,

[BookPrice] DECIMAL NOT NULL,

[AuthorID] INT NOT NULL,

CONSTRAINT [PK\_Books] PRIMARY KEY ([BookID]),

CONSTRAINT [FK\_Books\_ToTable] FOREIGN KEY ([AuthorID]) REFERENCES [dbo].[Authors]([AuthorID])

);