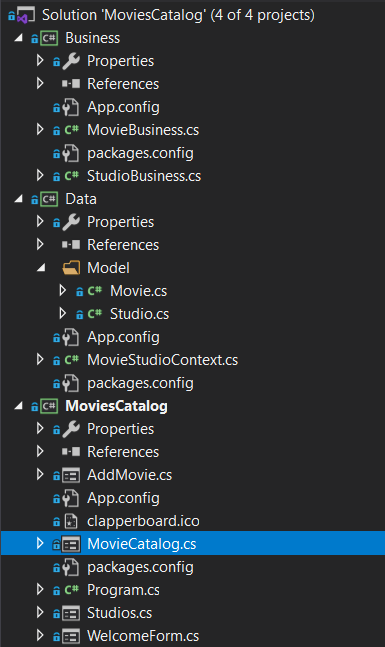
# Каталог за филми – MVC приложение

## MoviesCatalog – Windows Forms Application

MoviesCatalog е Model-View-Controller приложение от типа Windows Forms, реализирано с EntityFramework и CRUD операции, което представлява каталог/библиотека за филми и техните студиа.

## Структура на проекта

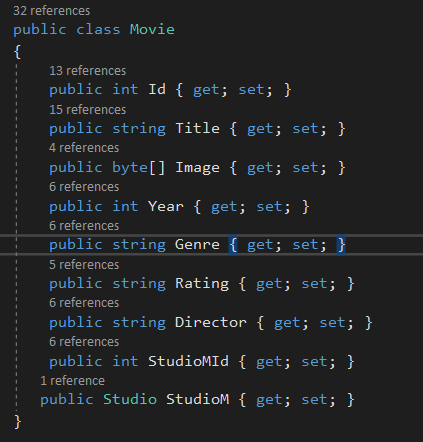
Използвана е трислойна структура, представена от три проекта – Data, Business и MoviesCatalog, които играят ролята на слой за данни, бизнес слой и презентационен.

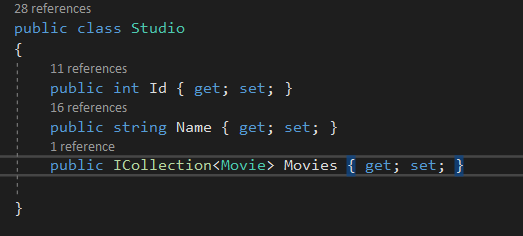


## Слой за данни

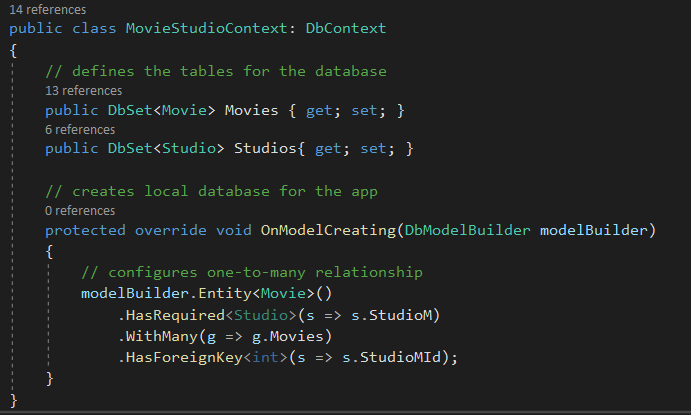
Слоят за данни се състои от папка с **модел** и **контекст**.

Моделът тук е съставен от два класа - **Movie.cs** и **Studio.cs,** намиращи се в **Model** папката,като в тях описваме свойствата им, които всъщност съответстват на колоните от таблиците в базата данни, и връзката между тях.





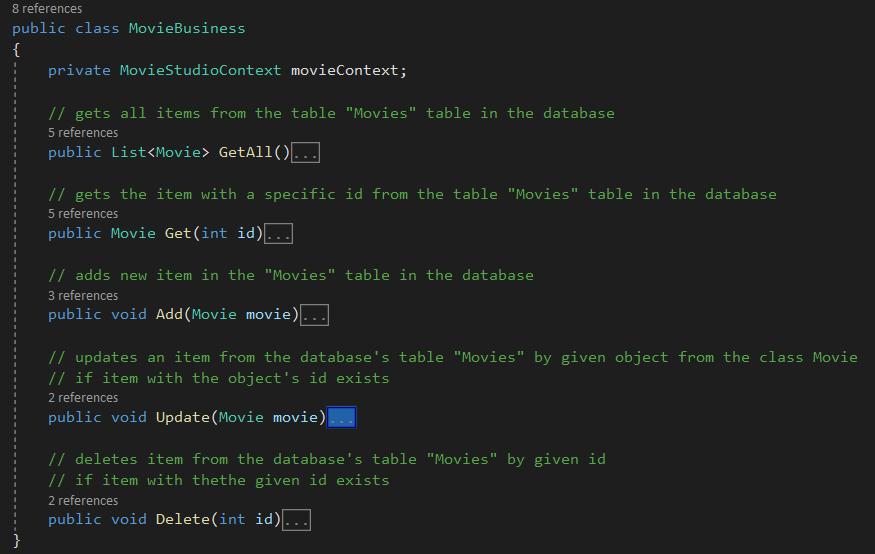
В самата папка **Data** се намира и класът **MovieStudioContext**. Той наследява **DbContext**.

В него се дефинират таблиците за базата данни, свързани помежду си с one-to-many връзка, и чрез метода OnModelCreating, тя се създава автоматично при първото стартиране на приложението от дадено устройство.

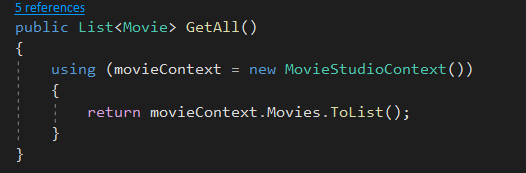
## Бизнес слой

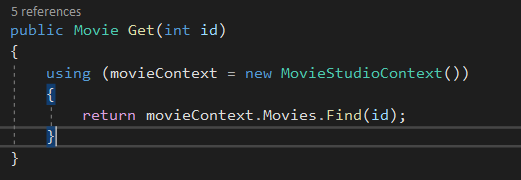
Бизнес слоят се състои от два класа – MovieBusiness.cs и StudioBusiness.cs , в които се намира бизнес логиката на приложението.

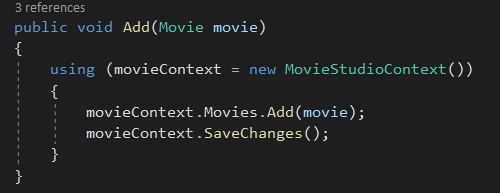
И в двата класа имаме поле от тип **MovieStudioContext**, което ще използваме в методите, които както следват са GetAll() – връща списък от всички филми/студиа от таблиците, Get() – връща филм/студио по дадено id, Add() – добавя нов филм/студио в базата данни, Update() – обновява филм/студио и Delete() – изтрива филм/студио от базата данни.

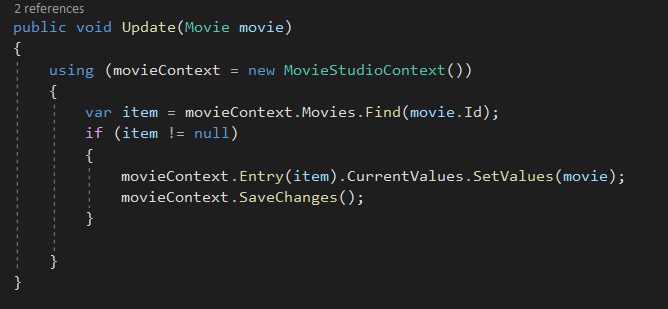
Класът MovieBusiness изглежда по следния начин:  


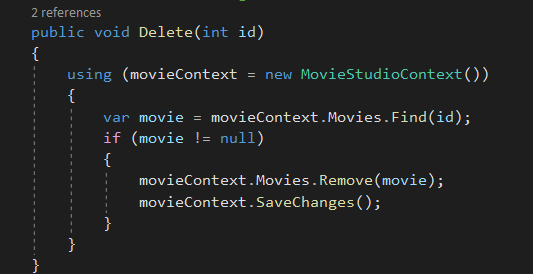
А методите както следва са:









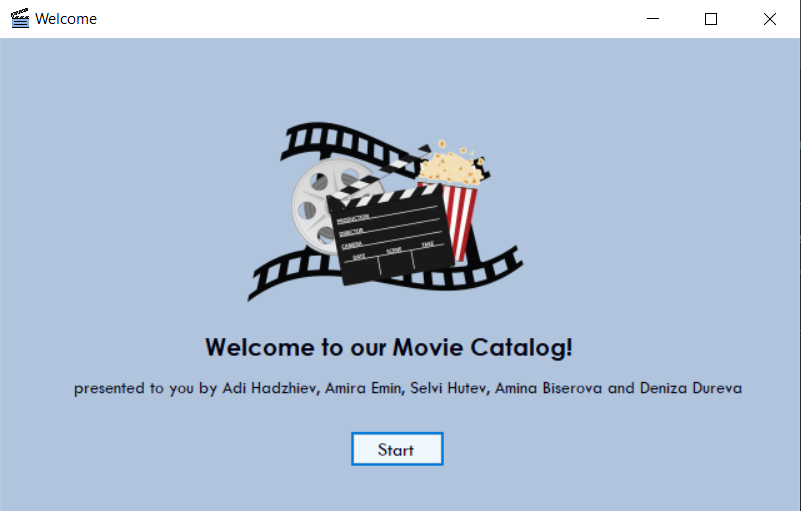


По аналогичен начин е създаден и класът StudioBusiness, но са използвани таблицата Studios и моделът Studio .

## Презентационен слой

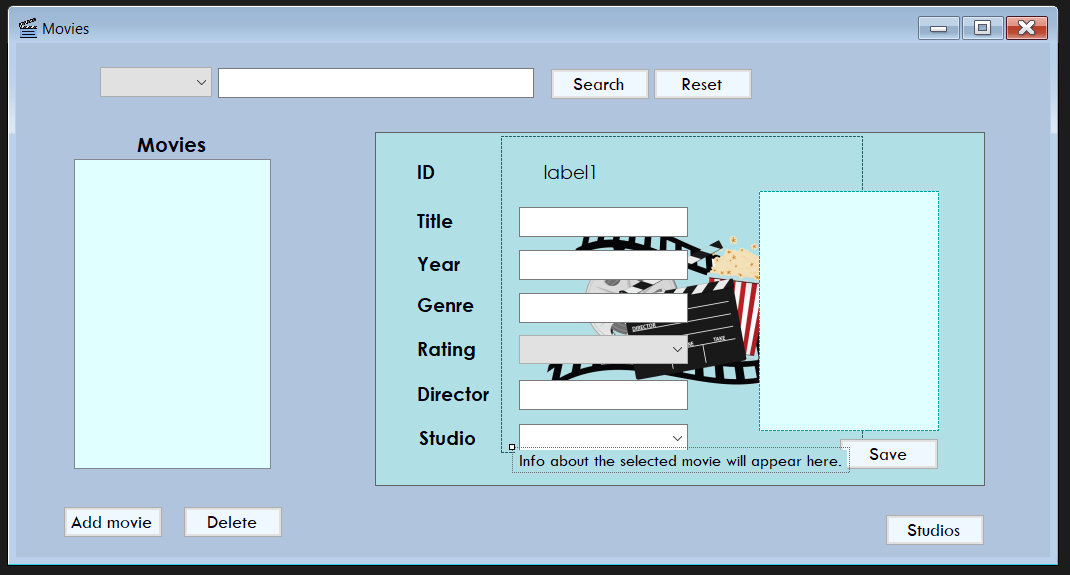
Презентационният слой е съставен от четири форми – **WelcomeForm** – тази, която се позказва при стартиране на приложението, **MovieCatalog** – главна форма за управление на филмите (предметите от класа Movie)**, Studios** – главна форма за управление на студиата (предметите от класа Studio) и **AddMovie** – форма за добавяне на нов филм**.**

**Така изглежда външният вид (WelcomeForm.cs[Design]) на формата WelcomeCatalog.**

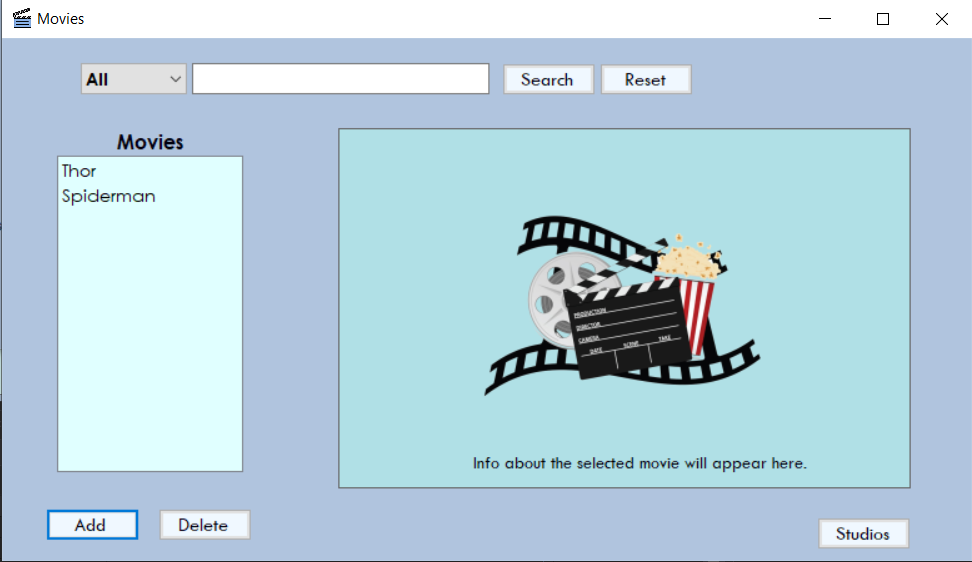
****

В тази форма имаме един бутон, който при натискане извиква главната форма MovieCatalog.cs.

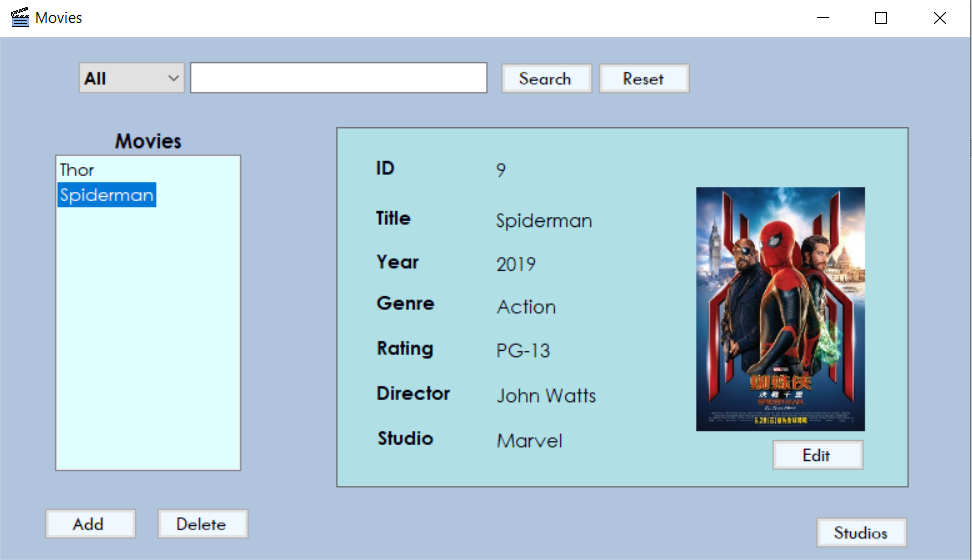
**Така изглежда външният вид (MovieCatalog.cs[Design]) на формата MovieCatalog в нестартирано състояние.**



А така, когато програмата е стартирана, в начално състояние:



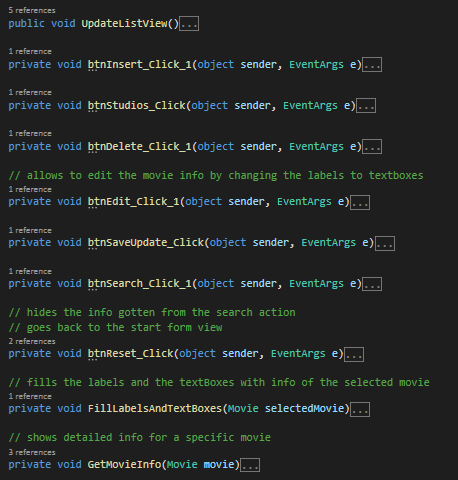
И по следния начин при избран филм:

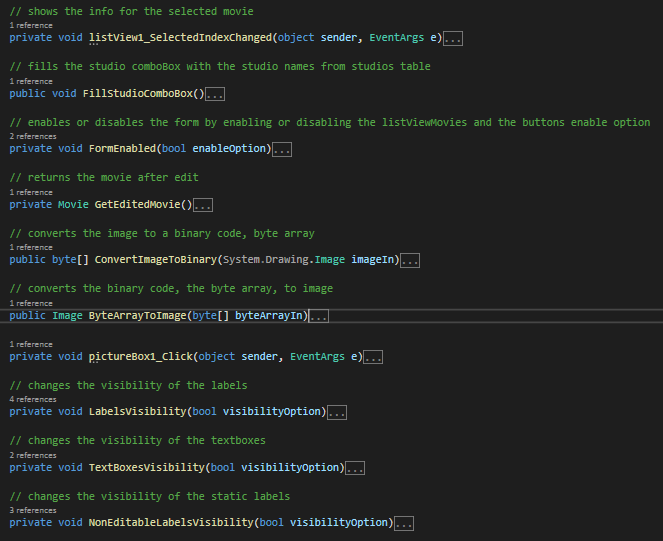


Формата MovieCatalog разделяме на няколко части:

* Част със списък от филми, която включва инструмент от типа **listView**, който показва наличните филми, и **два бутона**, които са съответно *Add* – извиква формата AddMovie и бутон *Delete*, който изтрива от базата данни избраният филм от listView-a.
* Част с информация за филма, която е съставена от няколко **статични лейбъла**, които остават непроменени в хода на програмата, а именно тези с имената на свойствата на филма, **променливи лейбъли**, които представят информацията от базата данни за избрания филм, които в последствие могат да променят съдържанието си при бъдещо редактиране на информацията, инструмент от типа **pictureBox**, съдържащ изображението от полето Image на филма и бутон *Edit*, който след натискане изчезва, заедно с променливите лейбъли и се появява бутона *Save*, заедно с възможни за редактиране *textВox*-и, а останалата част от формата става неактивна. Тя се активира отново при натискане на бутона *Save*.
* Част за търсене на филм, включваща един **textBox**, който съдържа търсената ключова дума, **listBox**, който съдържа списък с възможните категории за търсене и **два бутона** – *Search*, който при натискане търси в базата данни филм по избраната категория и ключова дума, като при намерени такива, филтрира *listView*-а и показва информацията за първия от него, а при несъвпадение се показва съобщение, и бутон Reset, който при натискане възобновява първоначалното състояние на формата.
* **Бутон Studios**, който при натискане извиква формата *Studios*.

Тези части се реализират със следните методи:

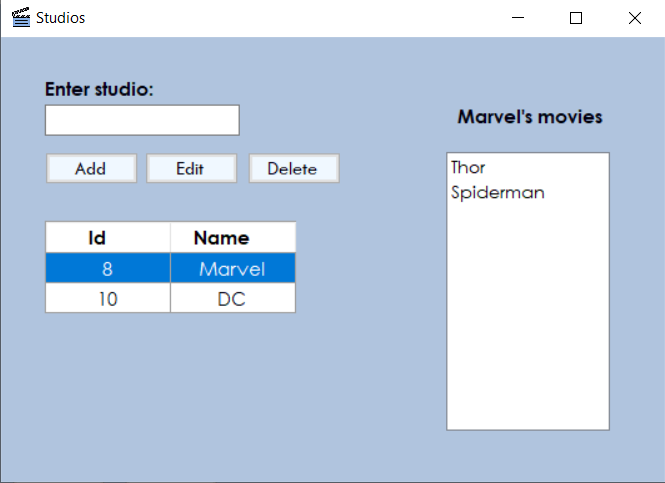




, като преди това са дефинирани следните полета:



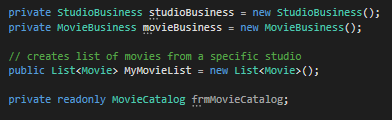
**Така изглежда външният вид (Studios.cs[Design]) на формата Studios стартирано състояние.**

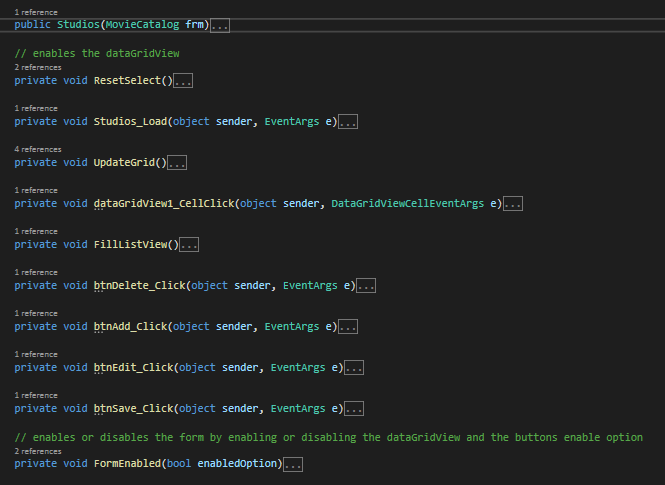


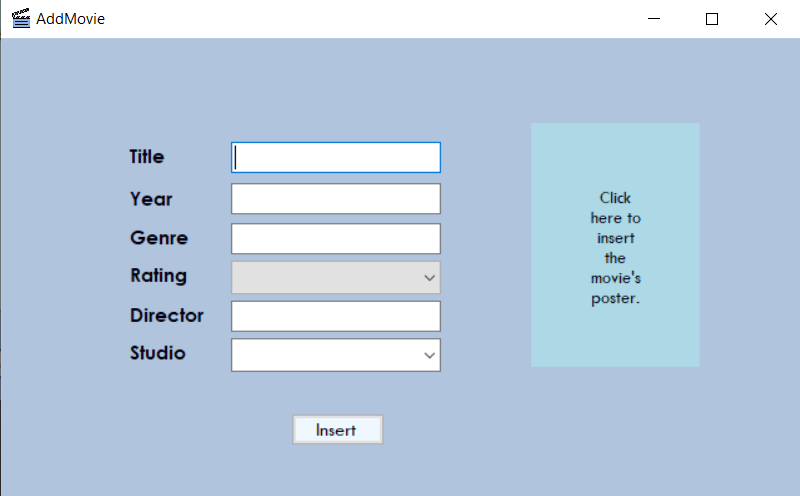
Формата Studios също разделяме на няколко части, които както следва са:

* Част със списък от наличните студиа в базата данни, представена чрез инструмента DataGridView.
* Част със списък на филмите, свързани със съответното избрано студио от dataGridView-a, включваща инструмент от типа listView.
* Част за добавяне, редактиране или изтриване на студио, съставена от няколко бутона –Add, който добавя ново студио с написаното съответно в textbox-a име в базата данни, Delete, който изтрива от базата данни избраното студио в dataGridView, ако то няма свързани към него филми, а ако има съответно, съобщение с грешка се появява, и бутон Edit – който при натискане дава възможност да се редактира името на избраното в dataGridView студио, като заедно с това самият бутон изчезва и на негово място се появява бутон Save и останалата част от формата става неактивна. При натискане на Save бутона, формата отново се активира и в базата данни се запазват промените настъпили при редактирането, като dataGridView-ът се обновява.

Работата на тези части е реализирана чрез следните полета и методи:

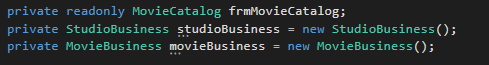


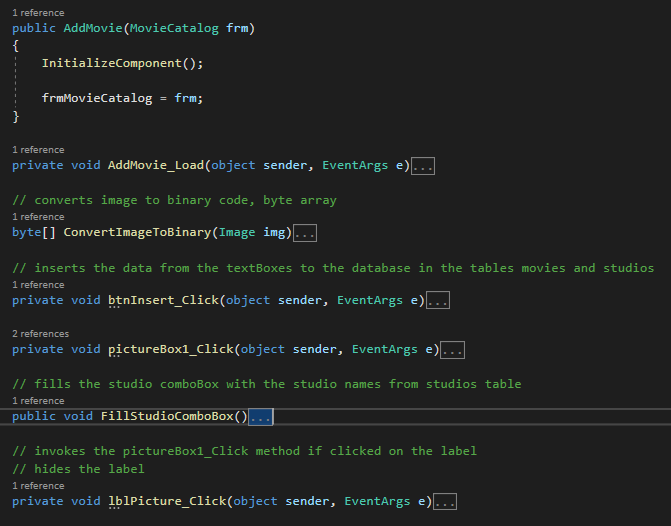


**Така изглежда външният вид (AddMovie.cs[Design]) на формата AddMovie.** 

Формата AddMovie е съставена от няколко лейбъла, които представят имената на свойствата на филма, няколко полета от типа textBox, в които ще въвеждаме информацията за филма, който искаме да добавим, статичен listBox, съответстващ на лейбъла Rating, съдържащ всички възможни видове рейтинг, от които да се избира, comboBox, съответстващ на лейбъла Studio, чрез който има възможност да се избере някое от наличните в базата данни студио или да се въведе ново такова, pictureBox, който при натискане отваря диалогов прозорец за избор на изображение и бутон Insert, който при натискане ако всички въведени данни са в правилен вид, добавя нов филм в базата данни, обновява формата MovieCatalog и затваря AddMovie формата.

Всичките тези операции се реализират чрез следните полета и методи:





Изготвил: Ади Хаджиев, Амира Емин, Амина Бисерова, Селви Хутев и Дениза Дурева