Documentation

MovieApp

**Student**: Andrija Nikić 2020440070

Contents

[Introduction 3](#_Toc64233984)

[Development goal 3](#_Toc64233985)

[System scope 3](#_Toc64233986)

[Product display 3](#_Toc64233987)

[Product perspective 3](#_Toc64233988)

[Product functions 3](#_Toc64233989)

[Product characteristics 4](#_Toc64233990)

[Restrictions 4](#_Toc64233991)

[Requirements specification 5](#_Toc64233992)

[User interface 5](#_Toc64233993)

[Software interface 7](#_Toc64233994)

[Hardware interface 7](#_Toc64233995)

[Communication interface 7](#_Toc64233996)

[Database 8](#_Toc64233997)

[Project constraints 8](#_Toc64233998)

[Verification of requierements 9](#_Toc64233999)

[Controller code 9](#_Toc64234000)

[Movies controller 13](#_Toc64234001)

[Rentals controller 17](#_Toc64234002)

# Introduction

The documentation provides insight into the development goals, the scope of the system as well as the presentation of the product or basic features, both functional and non-functional requirements of the application.

## Development goal

The application is a product that aims to rent movies via the Internet by various users, as well as an overview of the offered items.

## System scope

The system allows users to review offered products, as well as rent them.

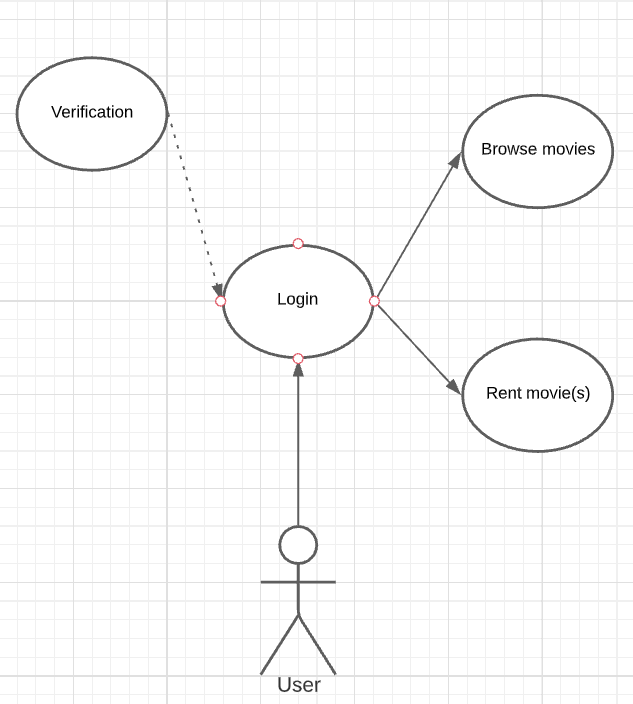
## Product display

### Product perspective

An application with which users will be able, without spending too much time, to review the offered products and rent them.

### Product functions

The user can view the product of the application as well as sort by name and genre. By clicking on a specific product, the user will be redirected to a page with additional information about the product itself. Access to the list of movies and rental form can only be accessed after logging in with proper parameters. Access to the administrator part of the application can also only be accessed after logging in with proper parameters.



Use Case 1 – User

### Product characteristics

The application involves three types of users: visitor, user and administrator. The application is not limited in terms of users who can access it. A visitor is only allowed access to the main page of the website, while a user (which a visitor can become after logging in with proper credentials) is able to browse products and rent them, as well as sort them by name and genre. It is taken into account that it will be used by users with very little experience in the field of internet browsing and e-renting, but also by those who already have previous experience. Having the correct parameters for login gives access to administrative rights that can be used to edit content.

### Restrictions

* Usability

The application should be such that users can efficiently and without any help perform the desired actions. User requests will be processed without errors and too long a wait interval.

* Performance

The application should be able to respond to user requests in the shortest possible time.

* Security

Only the administrator can access and edit the database, so certain measures have been taken to prevent unauthorized access to sensitive data.

* Reliability

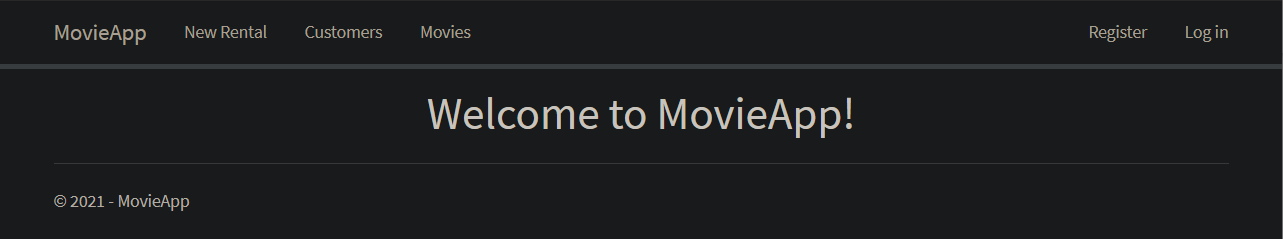
The application should function without errors and interruptions.

# Requirements specification

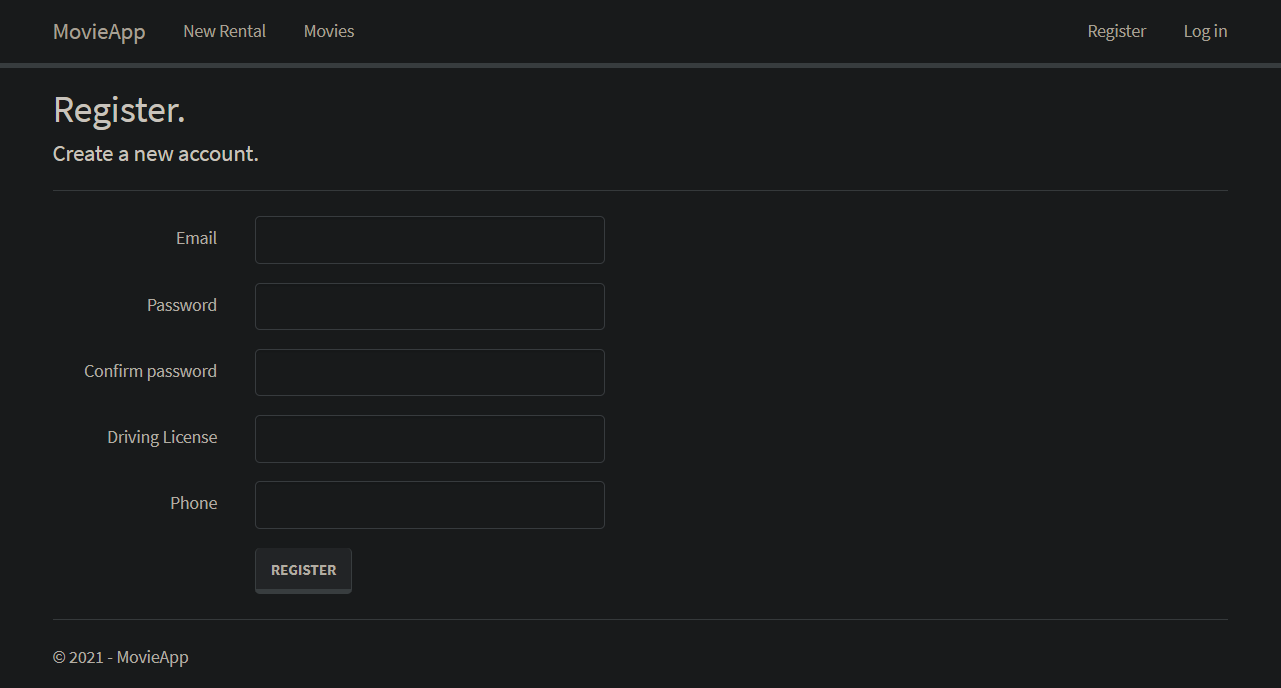
External interface

## User interface

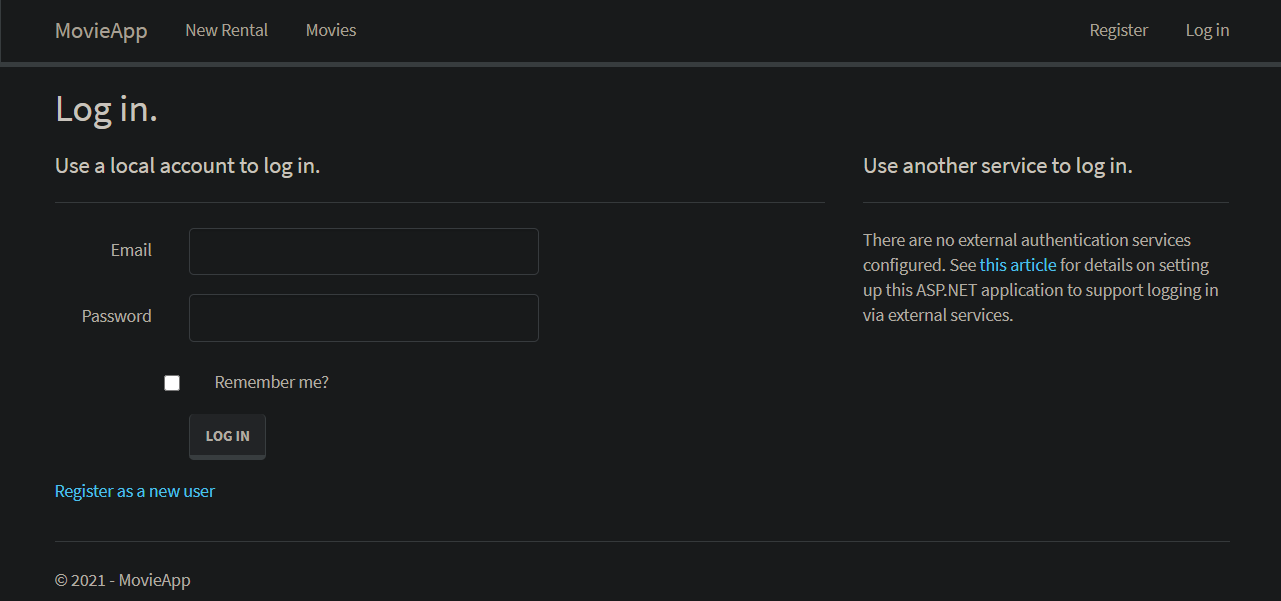
Upon successful access to the application, the user is shown the main page of the website. On the top part there is a navigation bar with links to the main page, new rental page, movies page, customers page, registration page and login page. The link to the customers page is only available to administrator accounts, whereas regular users and visitors will not be able to see and access it.



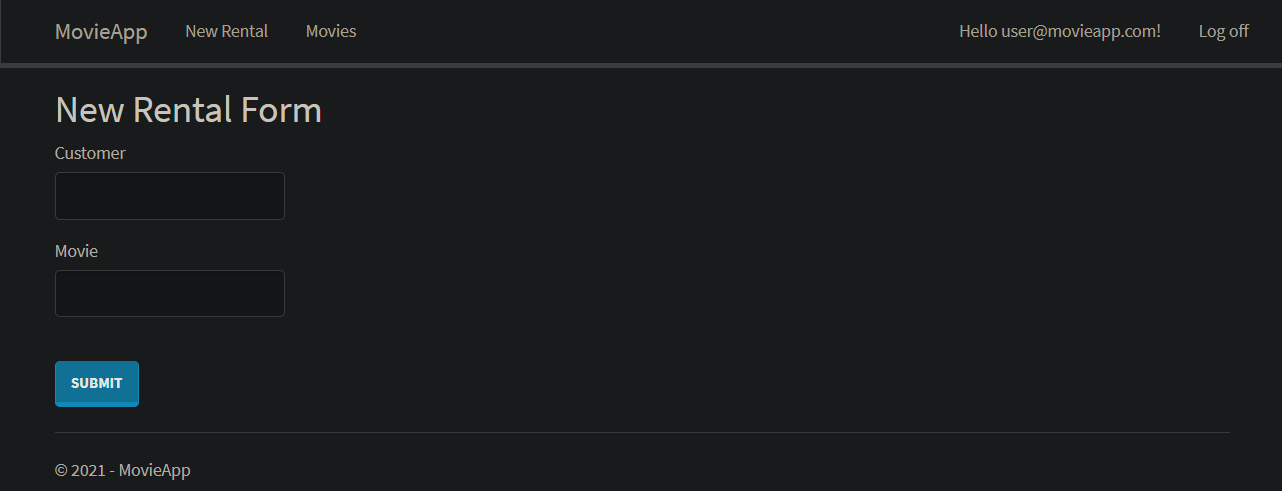
User Interface 1 - Home page



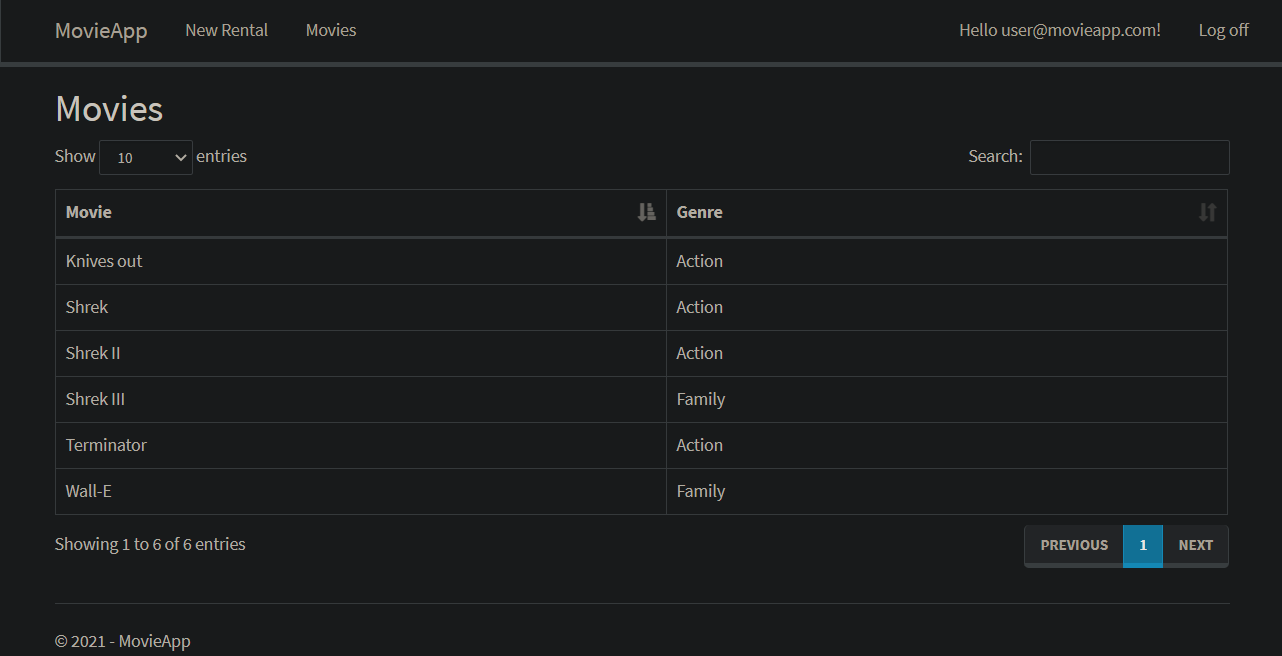
User Interface 2 - Registration page



User Interface 3 - Login page



User Interface 4 - Rental form



User Interface 5 - Movies page

## Software interface

The application can be used as long as the user’s device is using an operating system that supports some of the modern browsers such as Google Chrome and Mozilla Firefox.

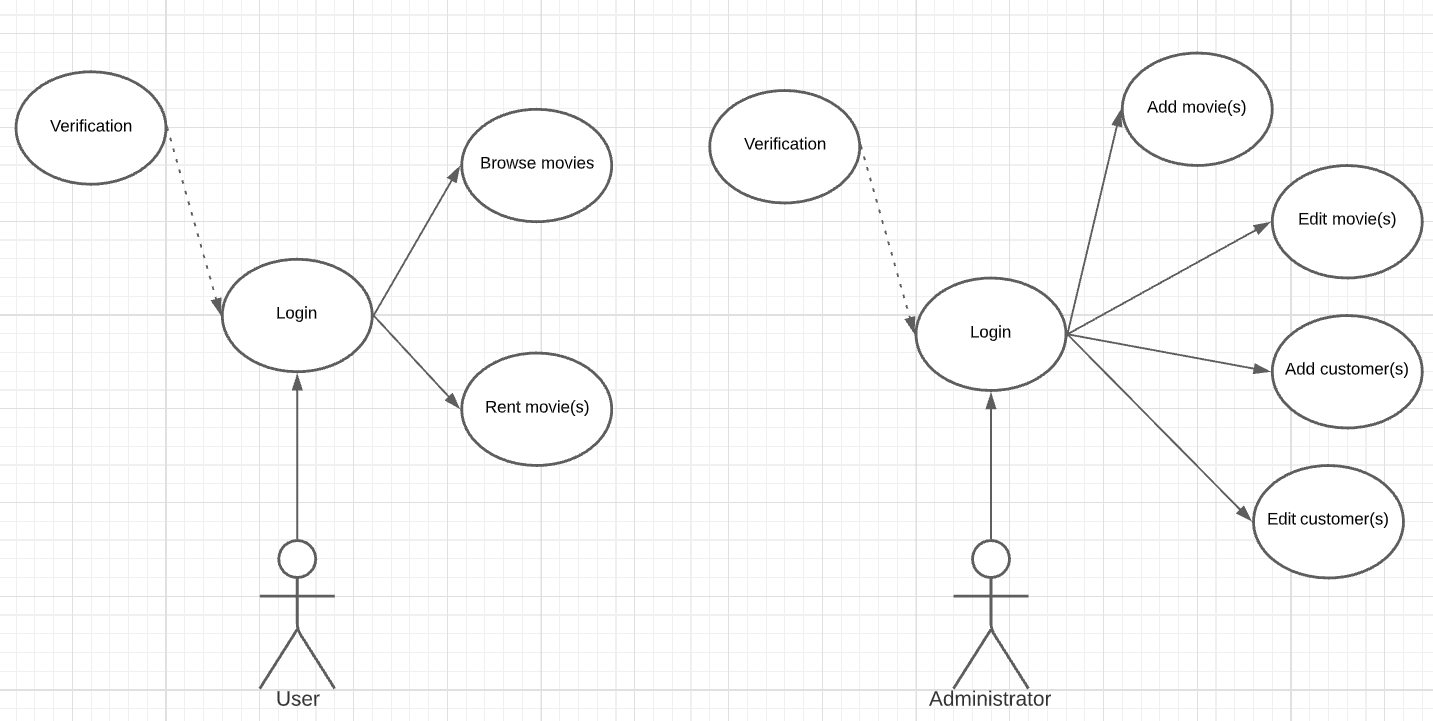
## Hardware interface

Access to the application is possible if the device used is connected to the Internet.

## Communication interface

The application uses HTTPS protocol.

**Functional requirements**



Use Case 2 - Functional Use Case

Functions that can be performed by an administrator:

* Edit movies

Upon logging into the application, the administrator can edit movie details.

* Add movies

Upon logging into the application, the administrator can add movies to the database.

* Edit customers

U Upon logging into the application, the administrator can edit customer details.

* Add customers

Upon logging into the application, the administrator can add customers to the database.

Functions that can be performed by a user:

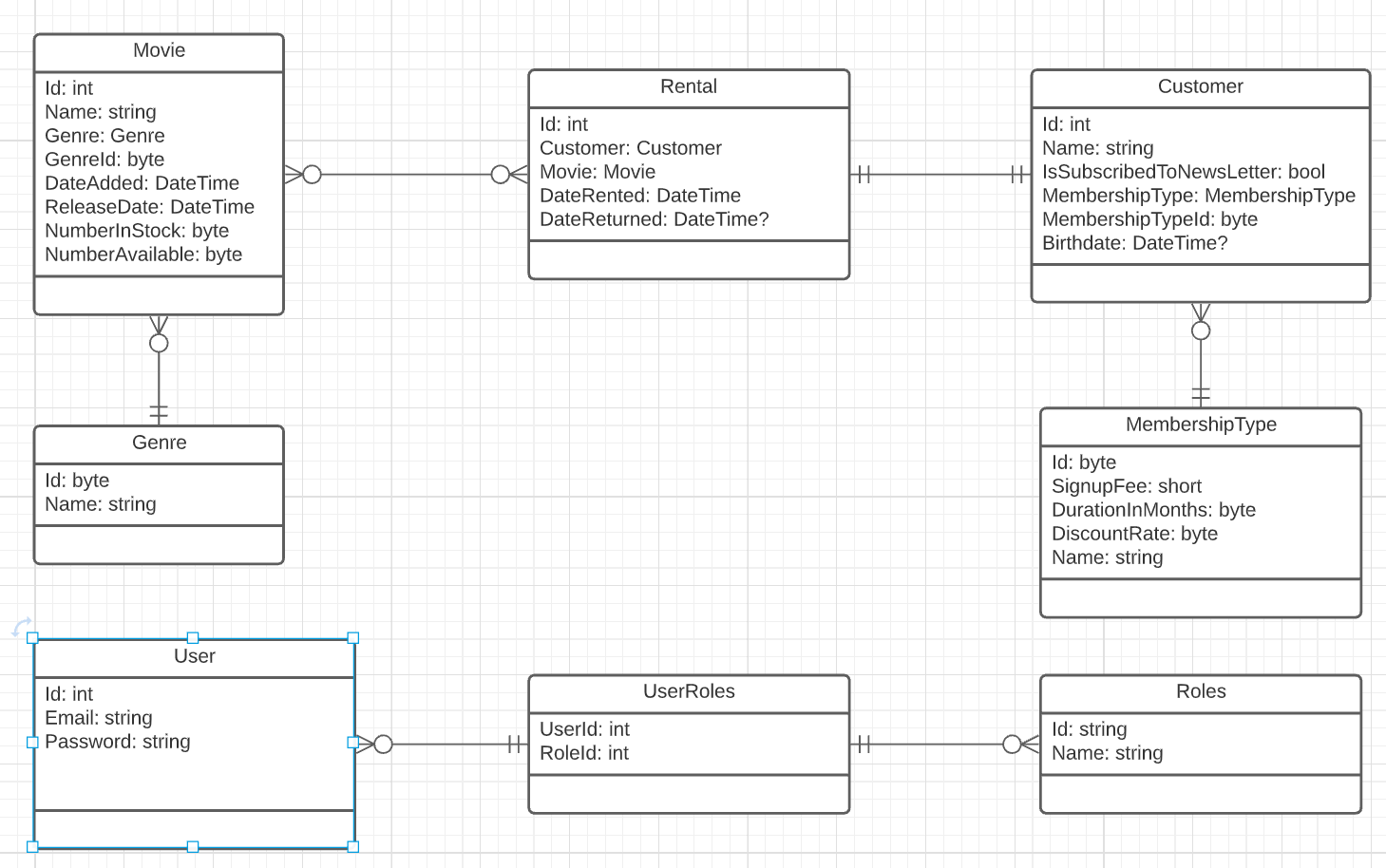
* Browse movies

Upon logging into the application, the user can browse through a list of movies, while also being able to sort them by name and genre, perform a search and choose how many will be displayed per page.

* Rent movies

Upon logging into the application, the user gains access to the rental form which enables them to rent a movie, while providing their name and the movie(s) they want to rent out.

# Database

****

Entity Relationship Diagram 1 – Database

# Project constraints

The application is to be developed by using the .NET framework alongside the C# programming language, while all the application code should be organized by the rules of the MVC architecture. The database is to be relational and should be developed using Entity framework. The application can generate a graphical user interface on the server side, using a template to generate HTML.

It is necessary to provide a certain degree of verification of the data sent by the user to the application. Layers of protection are possible, as follows: (1) HTML pattern in the data entry fields in the forms; (2) JavaScript validation of values entered into data entry fields in front-end forms;

It is necessary to write the accompanying project documentation on the development of the application, which contains (1) a database model with a detailed description of all tables, their fields and relations; (2) a diagram of the organization of application elements with an emphasis on emphasizing the MVC architecture on a specific example of a selected request, e.g. routes that your application processes; (3) a list of all activities supported through the application for all application user roles shown in the form of a Use-Case diagram; (4) a list of all controllers and their methods that perform the necessary program and business logic in order to perform all planned application activities.

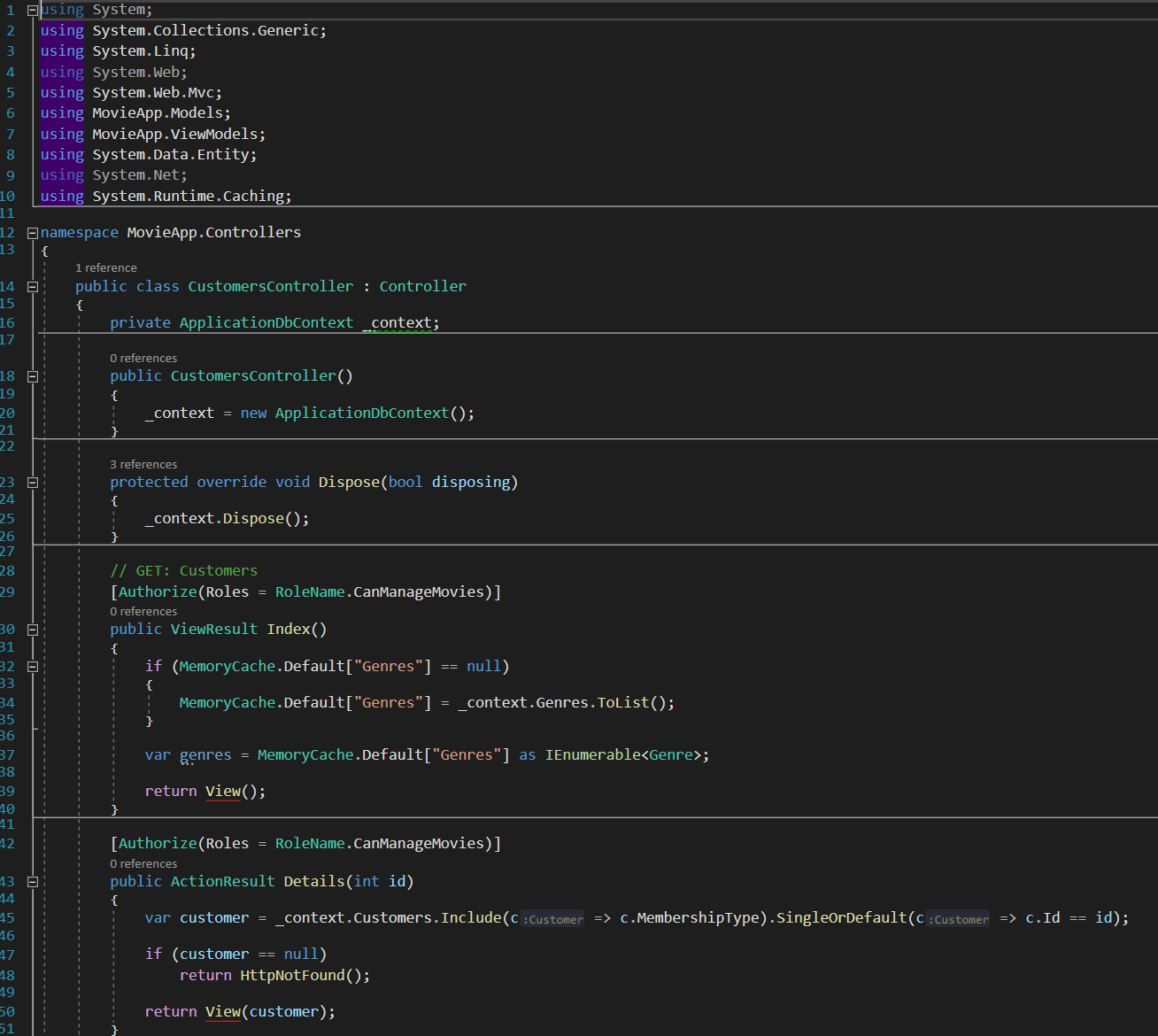
Project development must be carried out using the Git code versioning tool and the complete application code will be available in the public Git repository, e.g. on free GitHub or Bitbucket services.

Glimpse is to be used for profiling and performance surveillance.

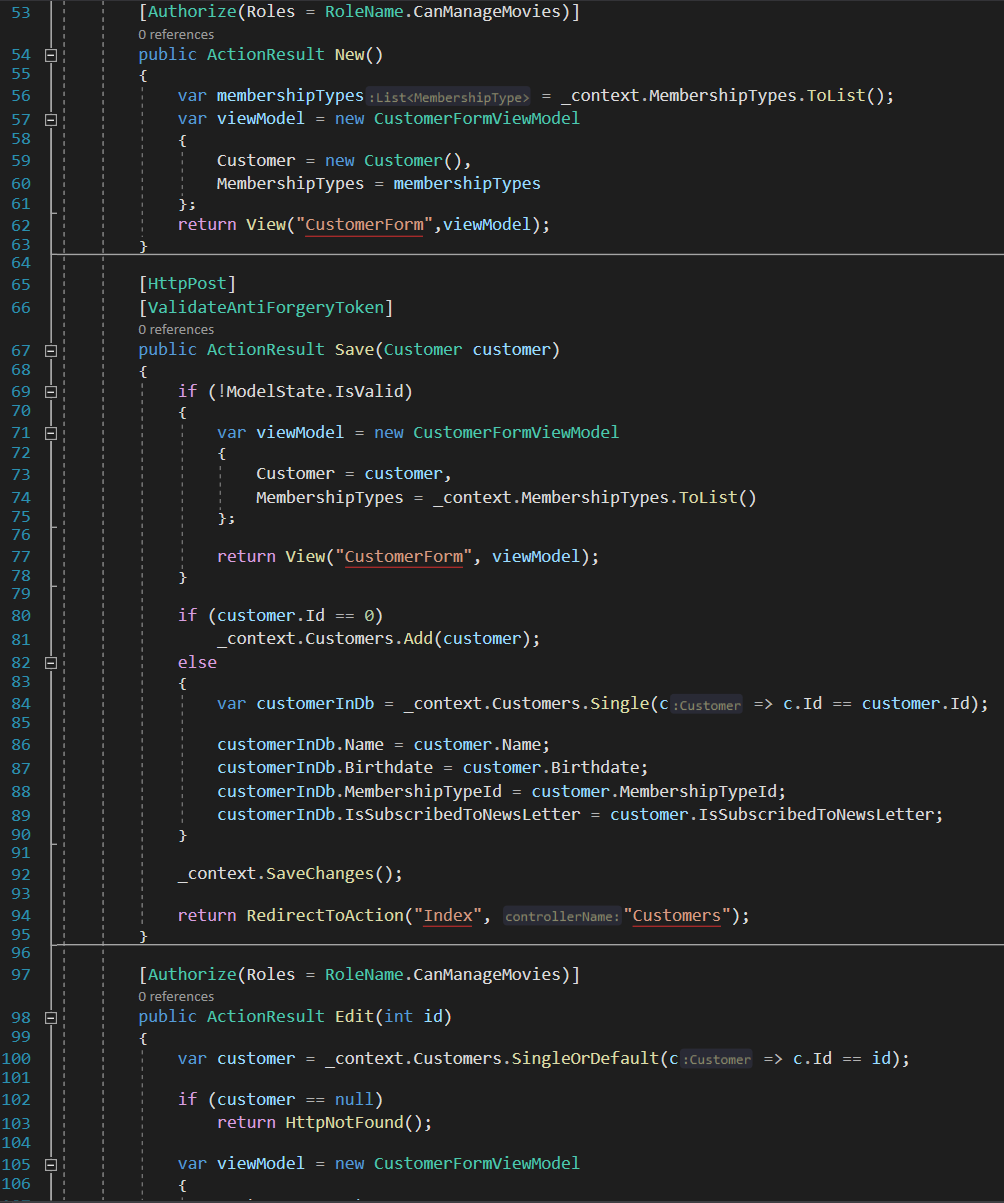
ReSharper is to be used for it’s many purposes, e.g. complying with code standards, analyzing code quality, eliminating errors and code smells etc.

# Verification of requierements

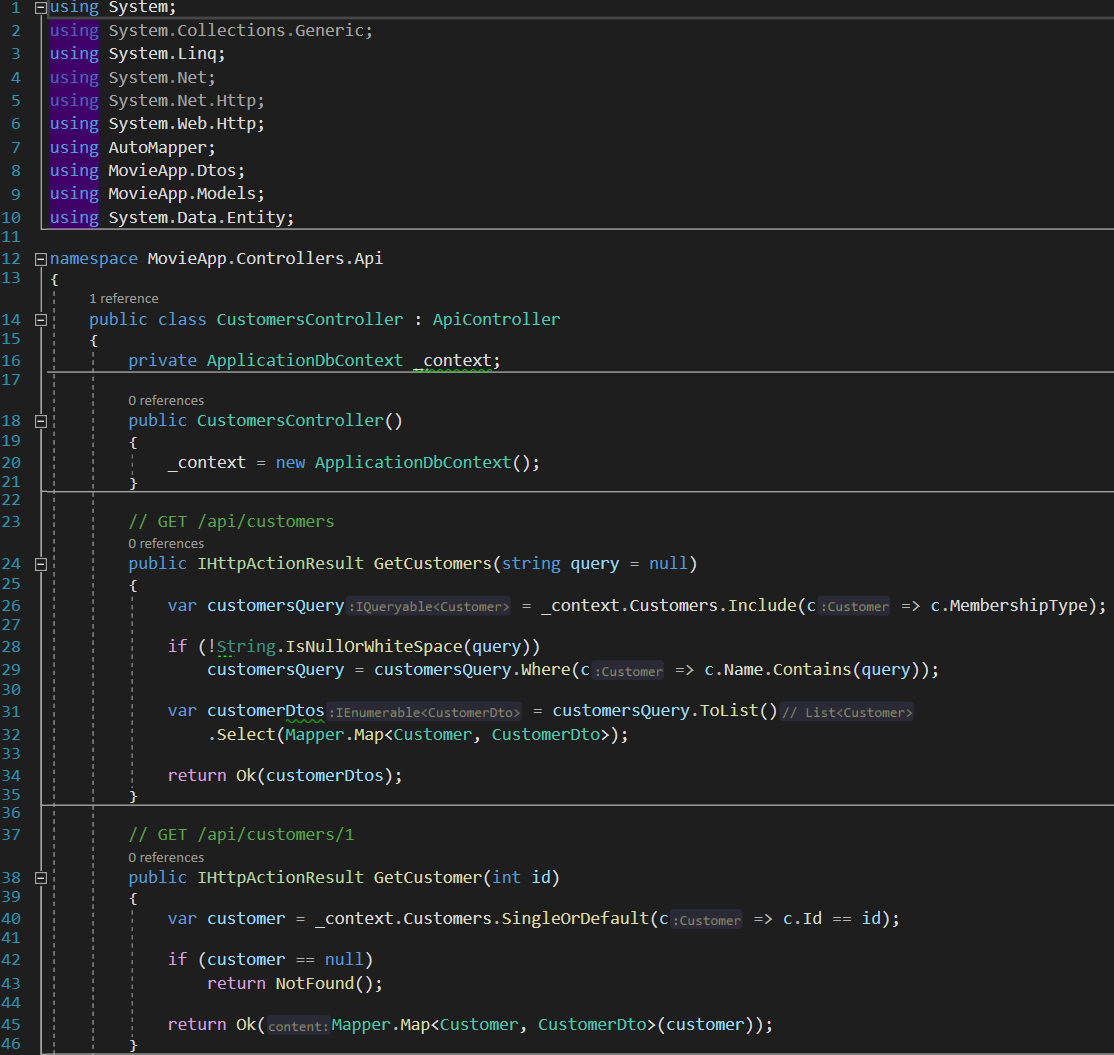
## Controller code

****

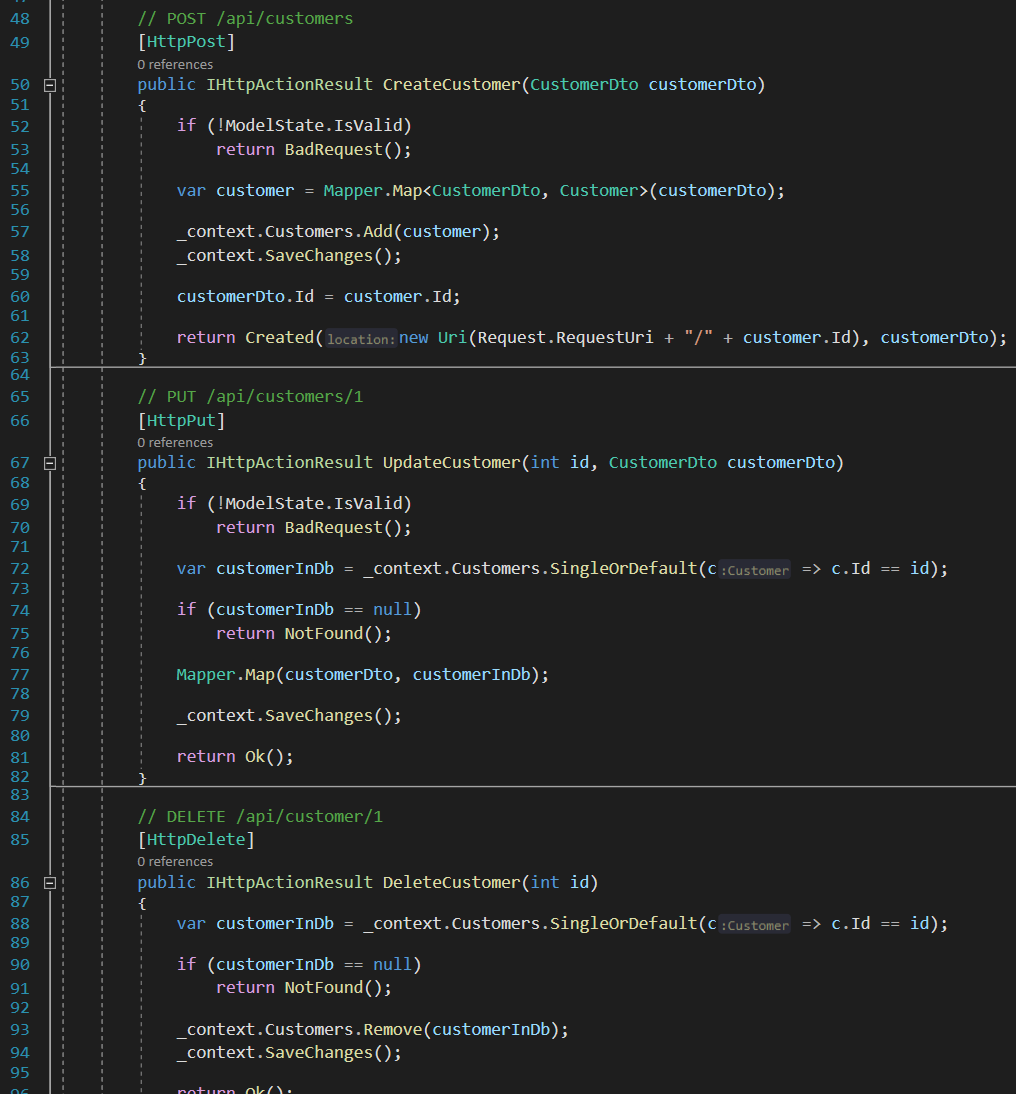
Controller code 1 - Customers controller



Controller code 2 - Customers controller

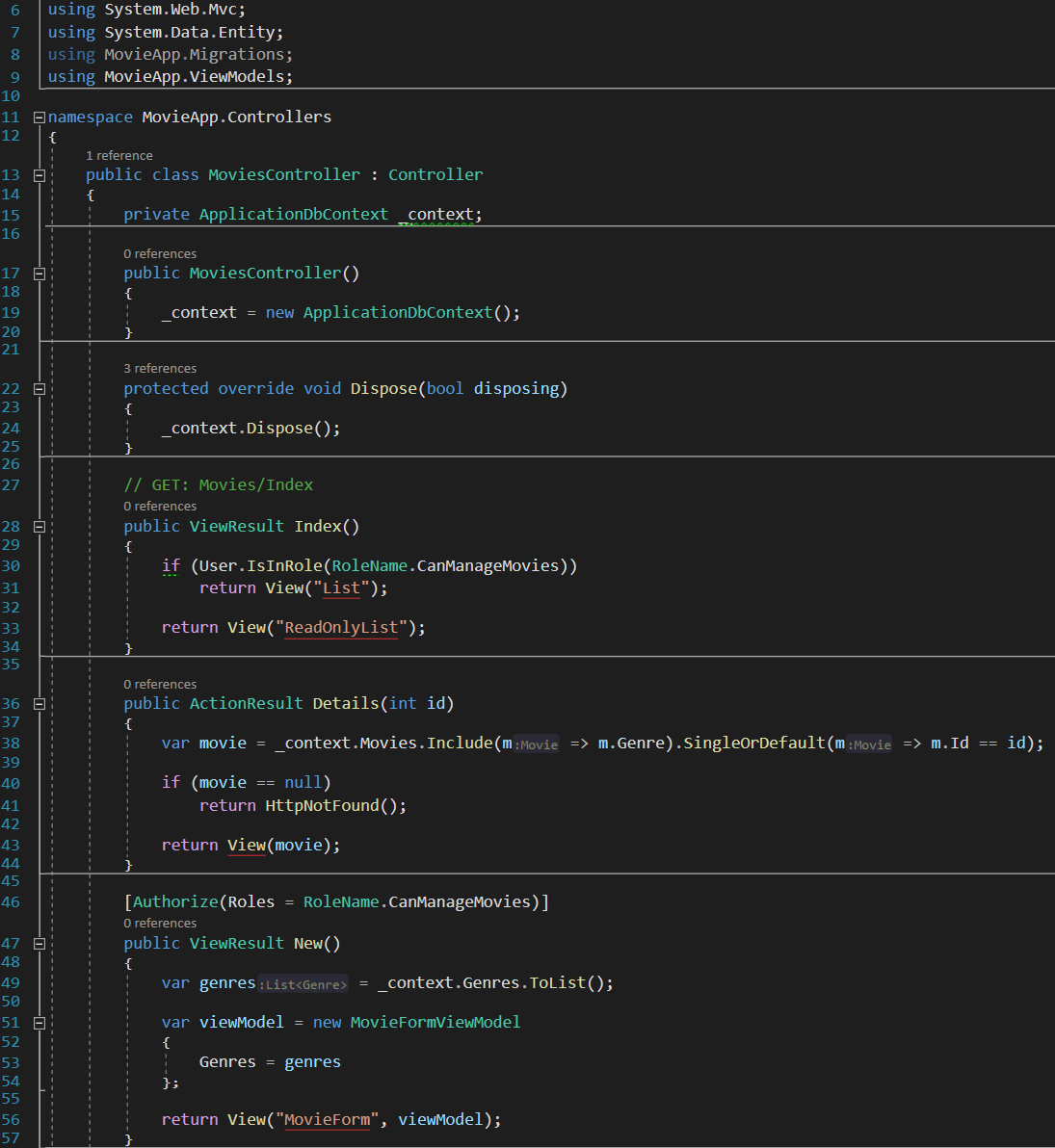


Controller code 3 - Customers API controller

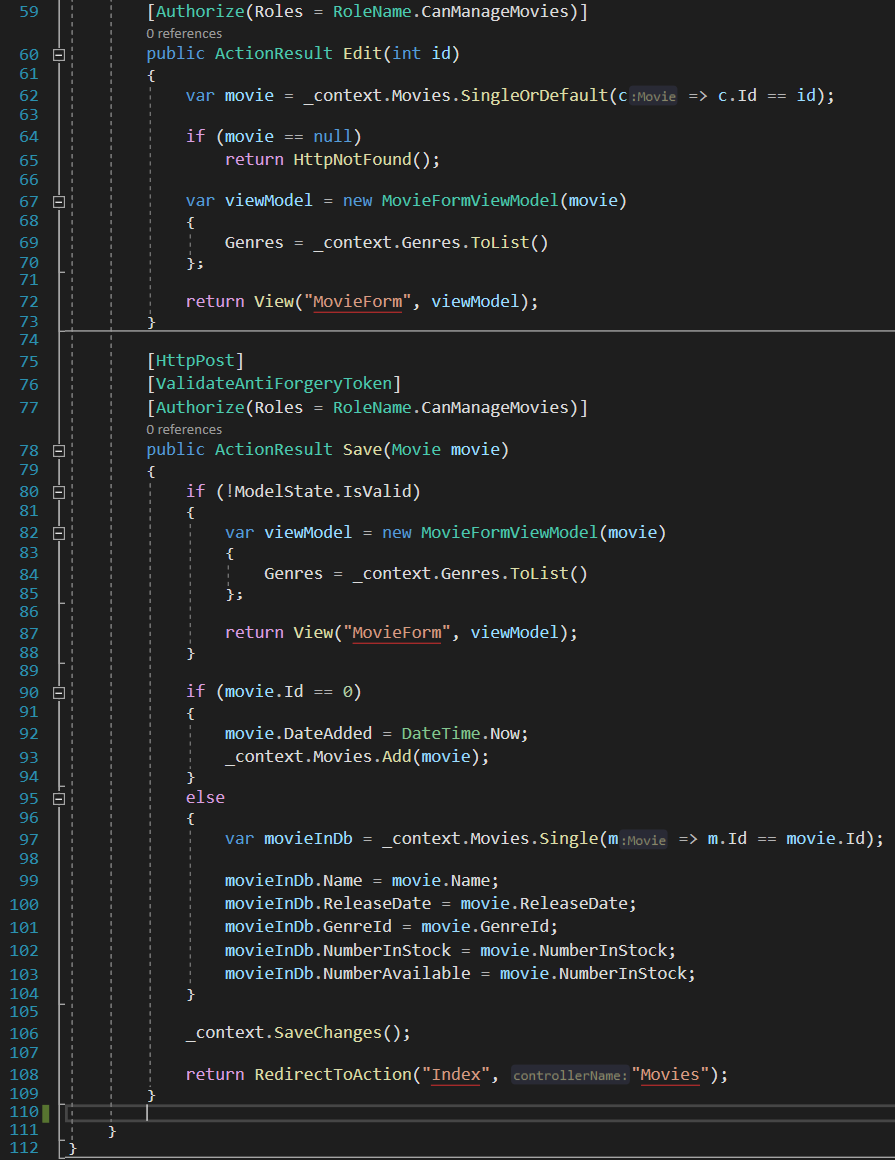


Controller code 4 - Customers API controller

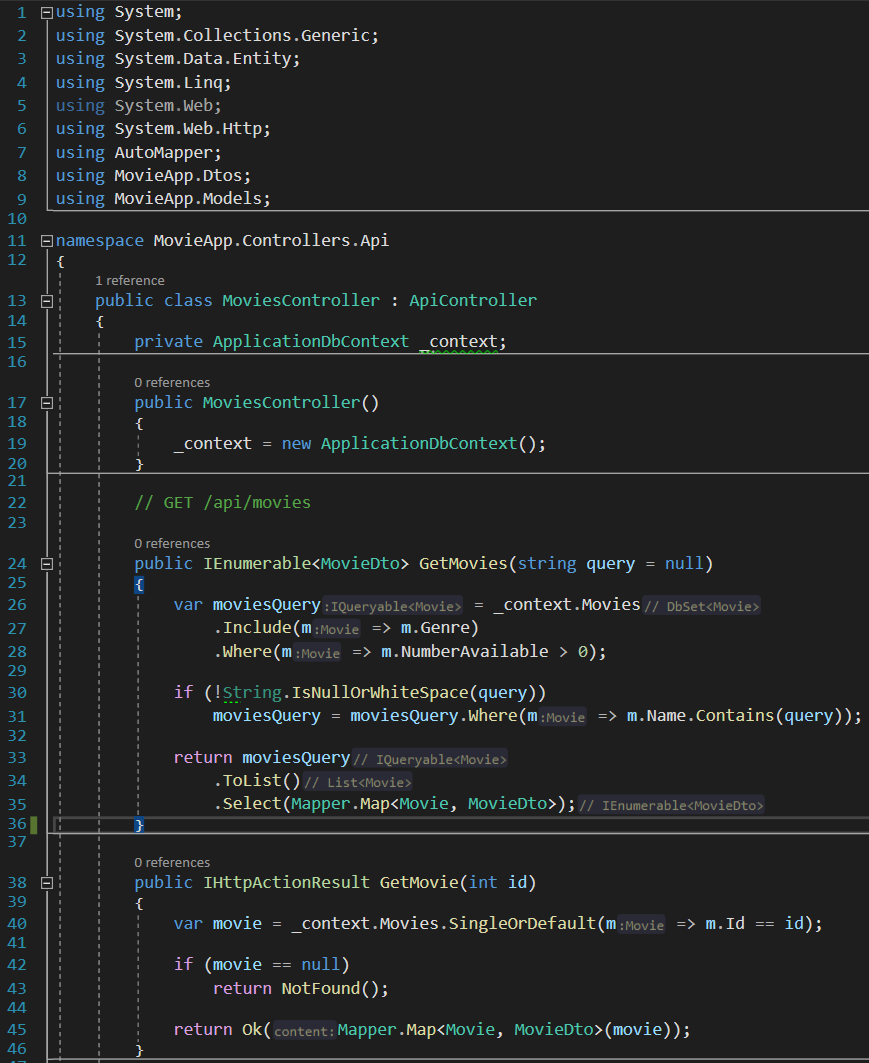
## Movies controller

****

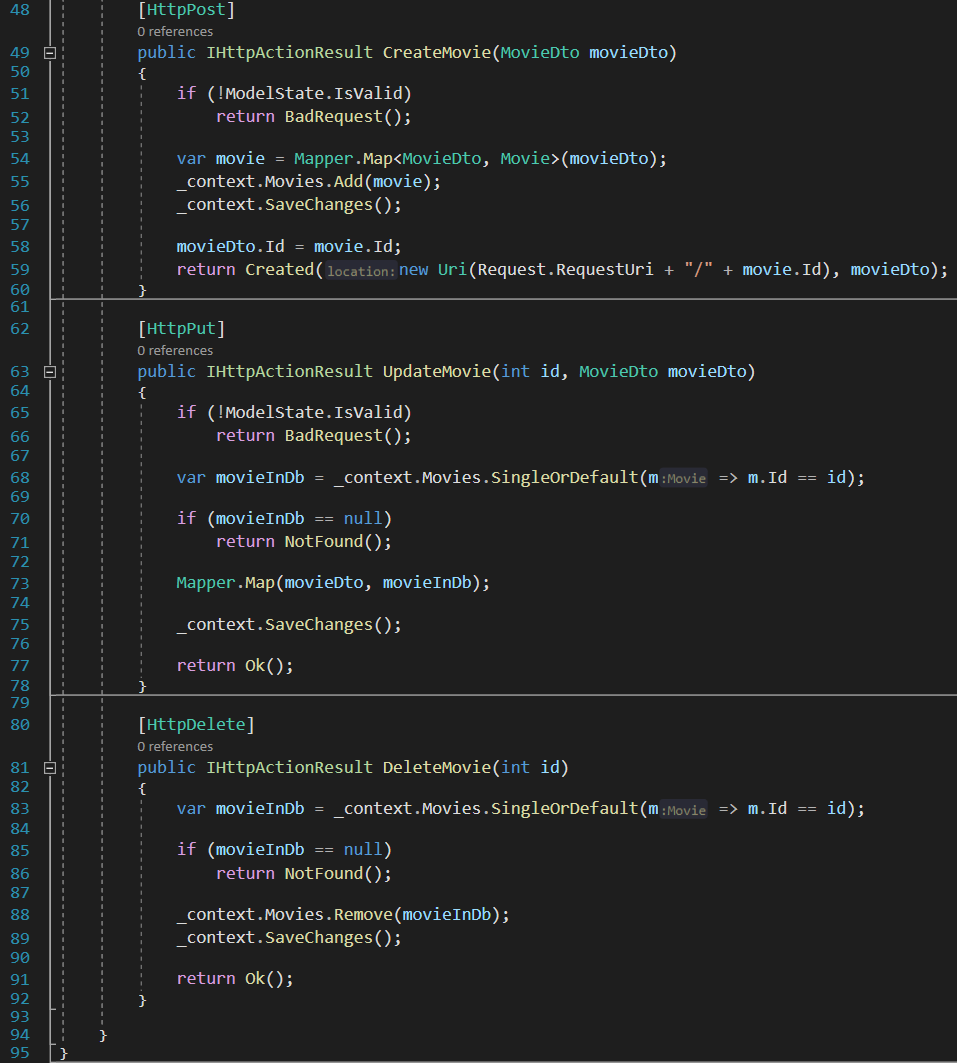
Controller code 5 - Movies controller



Controller code 6 - Movies controller

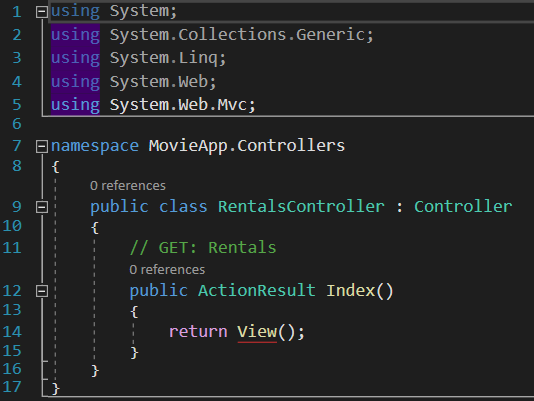


Controller code 7 - Movies API controller

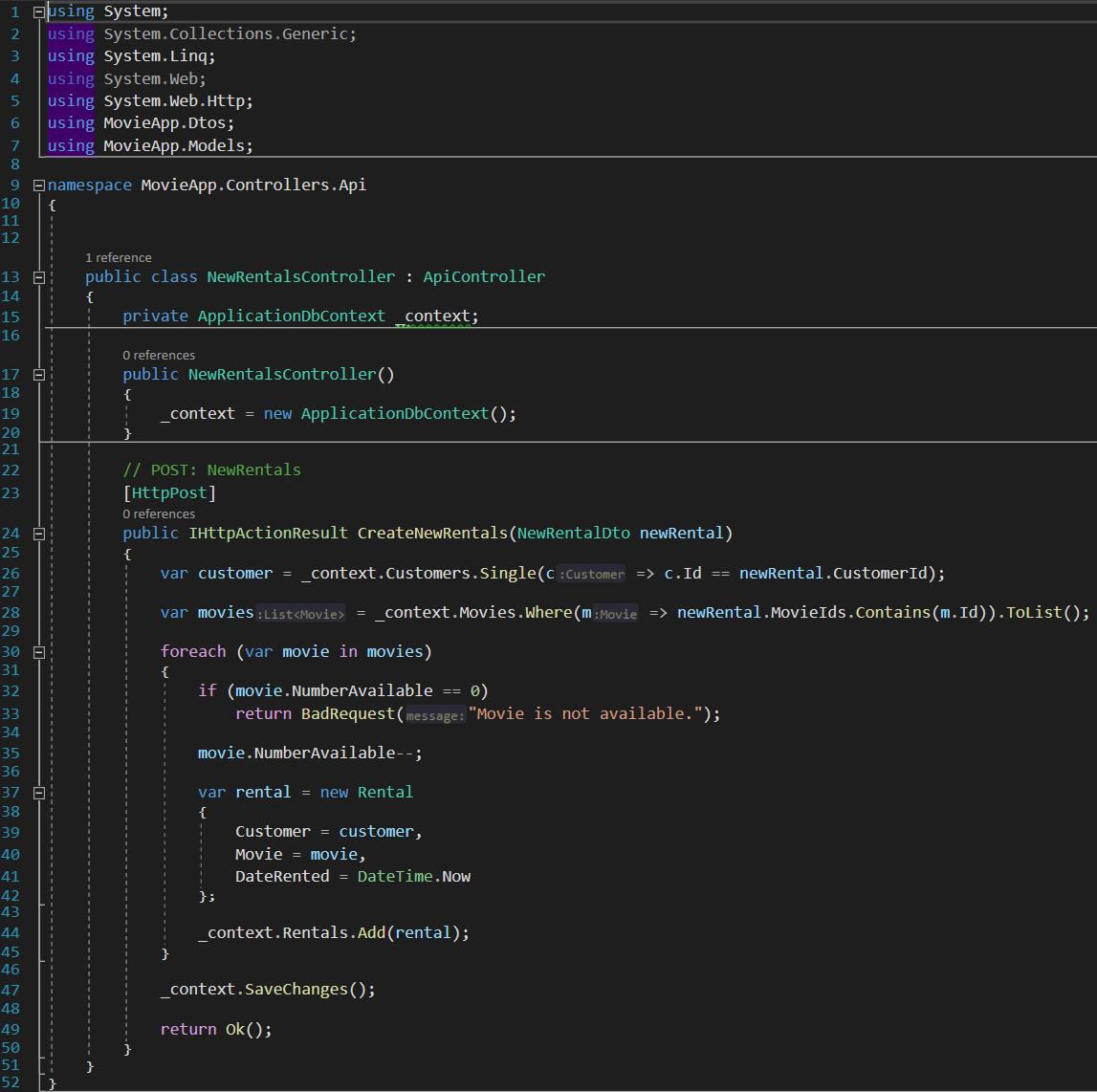


Controller code 8 - Movies API controller

## Rentals controller

****

Controller code 9 - Rentals controller

****

Controller code 10 - Rentals API controller