FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS
TECHNICAL UNIVERSITY OF MOLDOVA
WEB TECHNOLOGIES
LABORATORY WORK NR.3

REPORT

APRIL 16, 2018

Created by: Zaharia Gabriel
Supervised by: Plugaru Tudor

Do Some Crud

Goals

- Understand what is an ORM and how to use it;
- Get more familiar with MVC pattern;
- Understand BusinessLogic pattern;

Main Requirements

- Define one model;
- Implement basic Create, Read, Update, Delete operations for the defined model;

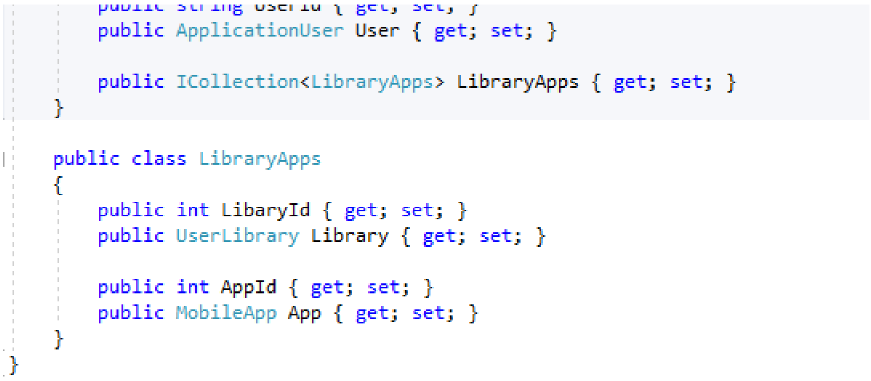
Bonus Points

- 2 or more models are defined and there are some relations between them (FK, MtM...); 2pt
- Define and ImageField or something related in a model. Display the uploaded image in detail view; 1pt
- On the list view, implement some basic filtering (search field, etc...); 1pt

IMPLEMENTATION ANALYSIS

For performing this laboratory work I chose ti work with ASP.NET Core , because it have some great features, like: Dependencies , working with local Sql Server Database and 2 modes of operating Development and Production

Defining the Model



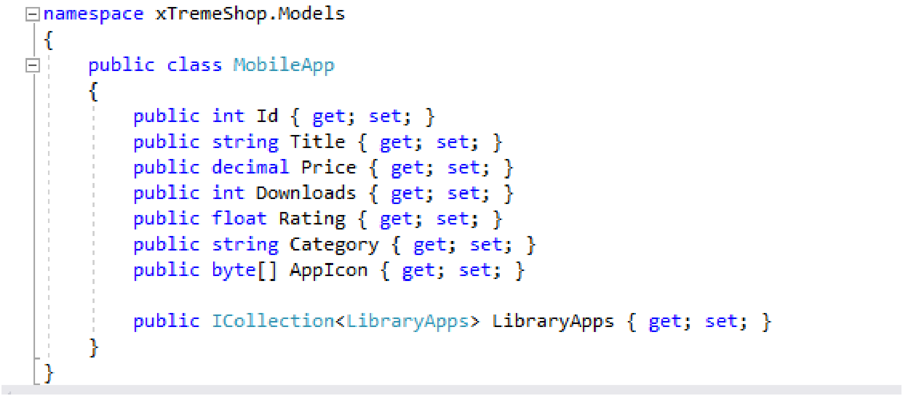
```
        public string UserId { get; set; }
        public ApplicationUser User { get; set; }

        public ICollection<LibraryApps> LibraryApps { get; set; }
    }

    public class LibraryApps
    {
        public int LibaryId { get; set; }
        public UserLibrary Library { get; set; }

        public int AppId { get; set; }
        public MobileApp App { get; set; }
    }
}
```

Figure 1: MobileApps Model

What is get and set properties ?

This code is automatically translated by the compiler to code similar to the one you posted, with this code is easier to declare properties and they are ideal if you don't want to implement custom logic inside the set or get methods, you can even use a different accessors for the set method making the property immutable.

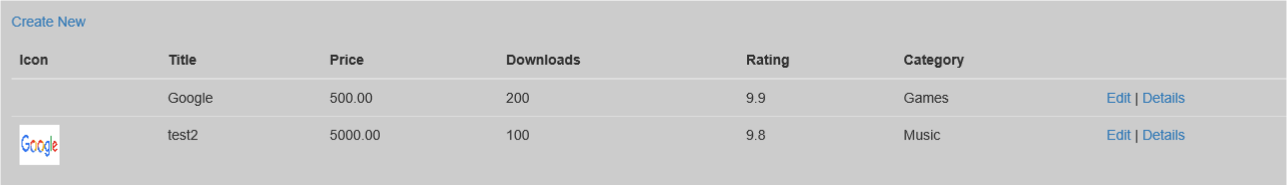
Also I have created the second Model for the Library option.



```
namespace xTremeShop.Models
{
    public class MobileApp
    {
        public int Id { get; set; }
        public string Title { get; set; }
        public decimal Price { get; set; }
        public int Downloads { get; set; }
        public float Rating { get; set; }
        public string Category { get; set; }
        public byte[] AppIcon { get; set; }

        public ICollection<LibraryApps> LibraryApps { get; set; }
    }
}
```

Figure 2: Library Model

Implementation of basic Read, Update, Create

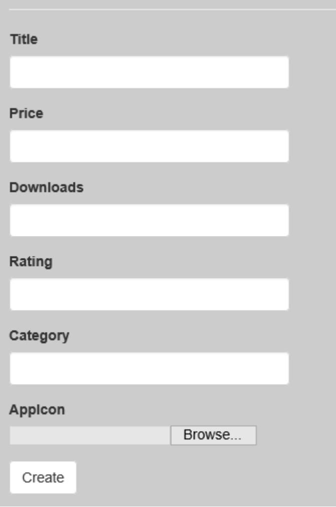


Create New

Icon	Title	Price	Downloads	Rating	Category	
	Google	500.00	200	9.9	Games	Edit | Details
Google	test2	5000.00	100	9.8	Music	Edit | Details

Figure 3: Create, Edit

Let's see the Create form of publishing the applications.



Title

Price

Downloads

Rating

Category

AppIcon

Browse...

Create

Figure 4: Create Form

Main Navbar

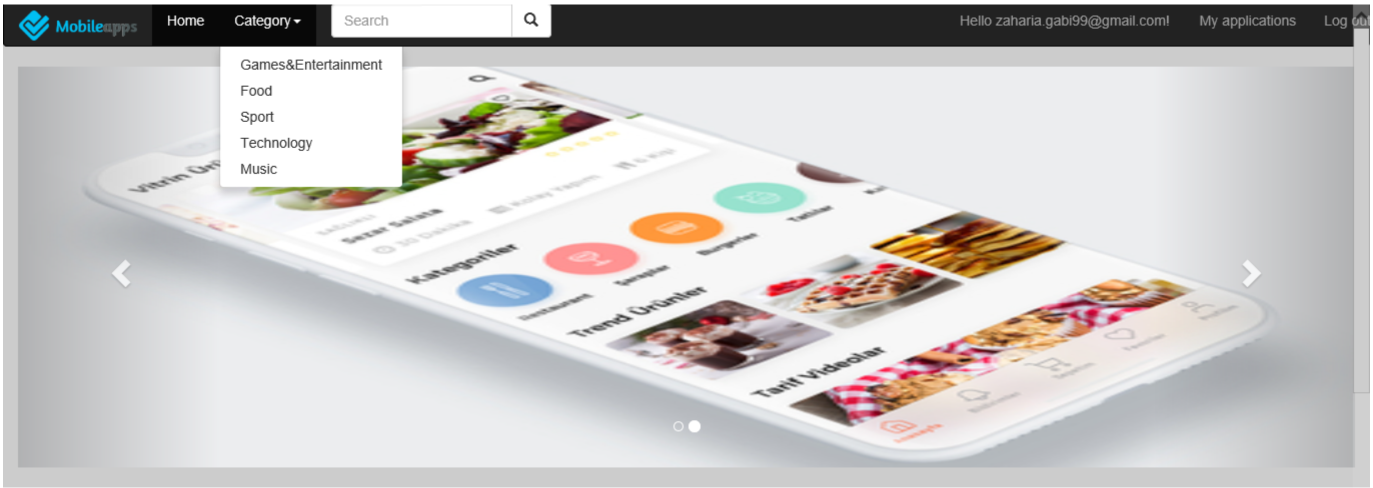


Figure 5: Main Navabr

Application features:

- Login
- See the current the user name
- Search Option
- Filter option by category, name
- Many to Many relations (Many libraries to many applications)
- Roles : Administrator/User/Host
- Possibility to edit profile
- Apps library : store the added apps

Conclusion

During this laboratory work I have expected that ASP.NET Core is more easy to use and implement and Web Service.

.NET Core supports the previously mentioned operating systems as your development workstation. Visual Studio provides an Integrated Development Environment (IDE) for Windows and macOS. You can also use Visual Studio Code, which runs on macOS, Linux, and Windows. Visual Studio Code supports .NET Core, including IntelliSense and debugging.

Connection with Database is very easy to perform and also the tables from it. Here , the administrator role has a special container , for being store and having the possibility to login with it to have the full access on the web service.

I found this laboratory work very useful , because a have learned more about .NET Core Framework and developed skills using it.

BIBLIOGRAPHY

1.Presentations by TW course

2.stackoverflow.com

3.microsoft.com