

Abstract

In Software Engineering 2 we go one step further and developing a new program with some cool technologies like logging, MVC and etc. Our App is about some photographers which their photos are stored in our app. They must first login themselves into the app and then we give them some options which their can uses such as changing their personal data's and also some from the pictures such as IPTC and EXIF, which for this program we created it as a mockup. We wanted to created an option for the user to be able to upload more pictures to the databank but we didn't have enough time.

Technologies

- | | | |
|------------------------|-----------------|------------|
| 1) ASP.Net Core | 2) Electron.Net | 3) Log4Net |
| 4) 3-Layer-Artitecture | 5) SQL Server | 6) ADO.Net |

Description

Despite what our professor asked us to do we decided (of course after he gave of a green light) to develop the project with some new technologies such as APP.NET and .NET Core 2. Probably it made our work little bit more difficult but at the end I think we learned some essential things which we can use in future when we looking for a job.

As it said in Moodle we had to use at least one Design pattern in our project and that's why we decided to use MVC design pattern in our project, fortunately .NET Core gives us this option.

What it does is basically it creates a template for us with three existence folders which are:

- 1) Model
- 2) View
- 3) Controller

Which Model is presenting our data that we are working with, in this case our data's are:

Alireza Esbati, Sina barakati

- 1) Photographers
- 2) Photos
- 3) Exif information of the pictures
- 4) Iptc Information of the picture

The View is what the user sees in the browser or any other software.

The Controller is basically the connection between the model and the view. What it does is that it takes the request from the user and provide the requested informations and then give it back to the view which will show for the user on the screen. In our project we only needed one controller cause it's not a huge program and we didn't need to create more than one, as well as view.

Each method in our home controller returns a type IActionResult which represent the result of an action method. It returns the content of a view of a required view which we already programmed. Our Home Controller contain of 9 different methods and each have their own tasks:

Name	Description
Index	Shows the start page with login option and the list of the members
Home	Home is next step after the user log itself to the website. It contains the data from the user and also some buttons to redirect user to some other controller such as Gallery. It takes the data from the Index Controller through HttpPost request.
Privacy	It is About the data privacy
Edit	This controller return a view which the user can edit his/ her data and save it to the database. The data that user gave it to the server will get to the edit controller through HttpPost attribute.
Profile	If the user make some changes in the personal data the app redirects the user into a page which informs the user that data's has been successfully changed. At the bottom we have a button called Profile which call the Home controller.

Gallery	It redirect the user into the gallery site which contains all the photos of the user. The Route attribute on the top defines the URL of the View.
EditPhoto	If the user wants to make a change to the photos he/ she has to click on Photo details, which redirect the user into the page that contains the picture itself and some information's about the IPTC and EXIF. It takes the id of the picture so it can provide the data's from database.
EditPhoto	Here the user make some changes to the IPTC or EXIF data of the photo and the controller get the information's through HttpPost attribute and make the change in the database
Error	This was by default from ASP.Net template

Before we get into the database we wanted to mention that we used the 3 layer architecture which contains: Presentation layer, Business Layer and Data access layer.

Presentation Tier- The presentation tier is the front end layer in the 3-tier system and consists of the user interface. This user interface is often a graphical one accessible through a web browser or web-based application and which displays content and information useful to an end user. This tier is often built on web technologies such as HTML5, JavaScript, CSS, or through other popular web development frameworks, and communicates with others layers through API calls. If we understood correctly all the component that we have in our PhotoGallery project count as presentation layer.

Application Tier- The application tier contains the functional business logic which drives an application's core capabilities. It's often written in Java, .NET, C#, Python, C++, etc. This one which also known as Business layer was a little bit challenging to understand exactly what it means. We did some research in the internet cause we wanted to use the MVC model in our project and that's why we didn't know where does the business layer exactly go. Some people said that we need to create another project in our solution for our business layer and data layer which we did actually and some other said that the business layer is with the model together and that was what

confused us. At the end we understood that the business layer are just for connect the Presentation layer and the data access layer but before that the more important job is that it should also check the validation of the request and the passed data through the https to the server.

Data Tier- The data tier comprises of the database/data storage system and data access layer. Examples of such systems are MySQL, Oracle, PostgreSQL, Microsoft SQL Server, MongoDB, etc. Data is accessed by the application layer via API calls. In our program we used the Microsoft SQL Server which is our favorite. In this Project which you will find it under DAL we wrote all the codes which are relative to the database. All the changes that the user do with the personal data or the pictures data or also login through the website all procced here in this project.

In our database we created 4 tables:

- 1) Members → which contains the information's of all of the members
- 2) Photos → which includes the data's of the pictures + the id of photographer to know which photo belongs to who
- 3) IPTC → information about picture (we leaved it as mockup)
- 4) EXIF → information about camera (also as mockup)

We also did something extra and it is that a person can also register to our website, unfortunately we didn't have enough time to implement the plugin which user can upload photos to the database but maybe in future.

Another component that we used was logging. Actually we noticed that ASP.Net uses by default a very simple logging but we also used one the famous nuget package for this component which also recommended from our professor and it was Log4net. It was also a little bit quite challenging for us to understand why do we need logging in our program but after some research in internet we realize how important and useful it is.

Logging is just a fancy word to define a process of writing down everything you do.

There are several kinds of logging. You may log every operation of an application, log only when errors occur, or log crytical operations done by a

user, especially if you want to have control of who's doing what for audit purposes.

Applications can log at a code level for debugging or at a user level for audits and forbidden access register.

Every operating system have routines or services running on the background with the task of taking notes of everything that is happening. This way a SysAdmin can detect security breaches, malfunctions etc.

The Business layer in our project also validate the data that user give to the forms for registering and if everything is fine then it gives it to the data access layer and it do the rest and register the user to the website.

At the end we wanted to say that we learned a lot from this project but it was just a beginning nut also a very good start for learning further more about the ASP.Net core.