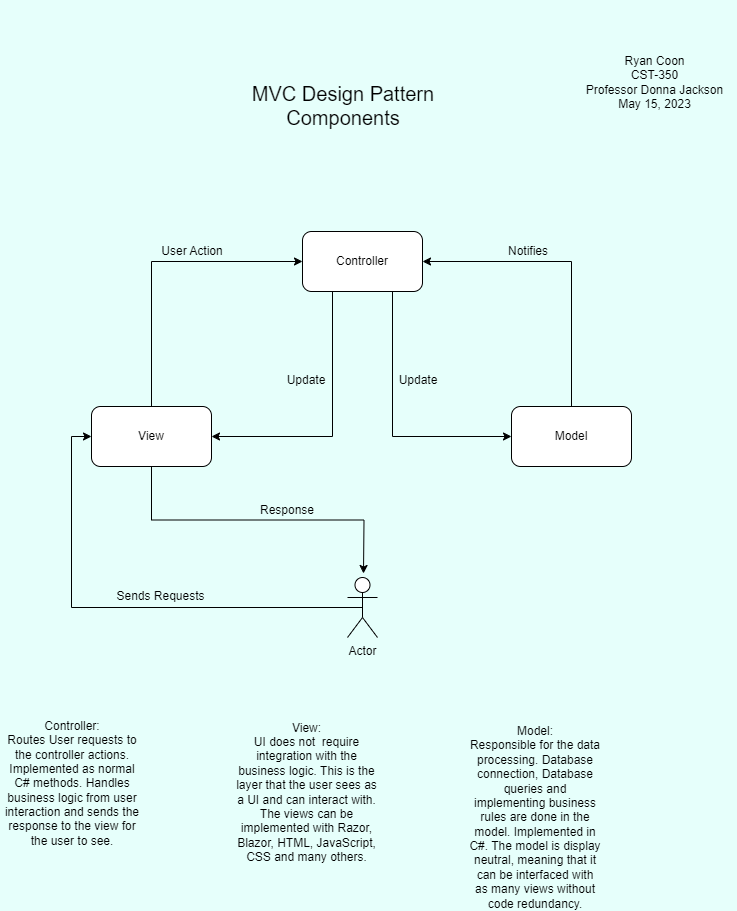
Using a drawing tool, like Draw.io or Visio, draw the different components of the MVC Design Pattern. For each component, then describe what .NET technologies that could be used, the role of the component, and how this design pattern helps solves separation of concerns. Also, describe some of the anti-patterns that could surface when using this design pattern. Comment on at least two of your classmates' posts.

ASP.NET MVC is a design approach based on separating the application into 3 objects that interact with one another in a loosely coupled manner to complete their own set of tasks. The View renders the data from the Model in response to the requests made to the Model by controlled events (Controller) made by the users interaction with the View. This helps teams work on separate parts as you can modify something in one unit without it affecting the other units. Anti-patterns can be considered bad system designs that can lead to a huge amount of maintenance making it hard to extend. Some of the anti-patterns we need to look out for are: the over-use of HTML extension models, the “Fat Controller”, placing business logic into the views, and not encapsulating action patterns into an object.



References:

Anti-patterns in ASP.NET MVC. (n.d.). https://www.matthewdresser.com/asp.net-mvc/anti-patterns

ASP.NET MVC Pattern | .NET. (n.d.). Microsoft. https://dotnet.microsoft.com/en-us/apps/aspnet/mvc

Govindaraj, A. (n.d.). What Is Anti-Pattern. https://www.c-sharpcorner.com/article/what-is-anti-pattern/

Panchal, N. (n.d.). Implementing MVC Design Pattern in .NET. https://www.c-sharpcorner.com/article/implementing-mvc-design-pattern-in-net/