

SPRINT 2 – DEVELOPER JOURNAL

Research the terms and concepts listed below.

- Binding Commands
- Binding Properties
- Branch
- Clone
- Design Patterns
- Fork
- Git Technology
- GitHub
- WPF: ICommand and INotifyPropertyChanged

OR

Razor Pages: state management and PageModel class

- Merge
- MVVM
- N-Tier
- Push vs Pull
- Seed Data
- Stage vs Committing
- Version Control System (VCS)
- WPF Styles or RazorPages _Layout.cshtml

NOTE: You do not need to submit definitions, but you should know what they are and may be asked to describe them in the questions below.

Answer the following questions in your own words:

1. Describe how the MVVM pattern applies to the architecture you are using in your project?
 - a It stands for Model, View, ViewModel
 - b Model = Is the source for the data used.
 - c View = This is all of the UI elements and talks with ViewModel to update its views. Doesn't contain any business logic. Should only contain logic to change views.
 - d ViewModel = This is the actual business logic. More tuned towards Filtering or validating user input. It communicates with the Model to access the data it needs.
 - i This applies to our project as we have already created a repository (to store our model data – seed data), then we will be incorporating the “View” aspect in which we create the sorting and filtering, etc., and then I am assuming sprint three will be where we incorporate the design (View) aspect of our project.
2. Discuss why design patterns have been established and why developers employ them.
 - a By having design patterns it can help projects to have a smooth build and completion. It helps teams to delegate work so it is easier for everyone to understand what is going on, easier to create more features, and it makes testing much easier.
3. Discuss how binding dependency properties work in a WPF project OR how model binding works in a Razor Pages project.
 - a You are pretty much creating like a #tag scenario for the page view, by going into the page model, and inserting the information there. You need to be connected to the database first and then you create a property (ex. Public Models.Product Product which will be what you bind the data to). Then when you go back into the View you can insert the “#” for the shortcut. (Ex. <input asp-for="@Model.Products.Product"/>. This can then be “copied and pasted” for other properties as well, just change the Model and property name.

SPRINT 2 – DEVELOPER JOURNAL

4. State the purpose of Seed Data.
 - a This is how you insert initial data into your database. It gives a starting point or baseline that you can use for testing, development, and helps to ensure people working on the application understand what data the application is meant for. You wouldn't want someone going in and adding random data that does not belong.
5. Describe the general responsibilities of the three-layer pattern: Presentation, Business, and Data Access.
 - a Presentation: This is the User Interface (UI). It is responsible for interacting with the user.
 - b Business: This is where the logic is located. It processes commands, makes the logical decisions, and then performs as programmed. It is in between the Presentation and Data Access. Example: If you wanted the application to get a list of movies released this month, the business layer would take the request, send the request to the Data access layer, and then kick it back to the business (logic) layer, where it would calculate the results, and push the list back to the presentation layer.
 - c Data access: This is where all the information is stored. It gets this information from wherever it is stored.
6. State the purpose of a Version Control System.
 - a Version Control systems like Git/GitHub for example help with quality control. It helps to manage and track code for projects. Makes sure everyone is working on the newest version and even track who is working on what for the project.
7. Discuss the use of Git and GitHub for a development team. Specifically, the concepts of cloning, branching, merging, pushing, and pulling.
 - a This makes collaboration among teammates easier and quicker. By cloning you make sure that everyone is working from the same starting point, everyone has the same information, because it is a copy of the original repository on your local machine. Branching allows teammates to work on delegated parts of the project without affecting the entire project. Merging allows you to update the entire project, while pushing and pulling allows you to push your work for others to pull and vs versa, which makes sure everyone is working on the most updated version of the branches, not the entire tree.
8. State the difference between staging and committing.
 - a Staging is where you choose while files you want to commit. This way when you commit it only commits the intended files.
9. State the difference between cloning and forking.
 - a These are doing the same thing, creating copies of the original repository, just done on different "platforms". Cloning is done when you use Git and it is a copy saved onto your local machine. Forking is an original copy done on the GitHub accounts repository.
10. Share in observations you made this sprint about yourself as a developer or your team.
 - a I have realized that as we get back into razor pages, I am remembering more from previous classes than I thought I would. I did struggle a little bit with Git, as I have never used it before, however now that I have a better understanding of it I prefer it over using GitHub through visual studio or otherwise. In addition, as our teams bonding continues over the project my imposter syndrome has decreased immensely. I am so happy that I lucked out with Nicole and Shane as teammates. We are working SO incredibly well together, and I cannot wait to see how the final product turns out with the application. It's a shame that we must switch teams for other projects!