UNIVERSITY OF WESTMINSTER

C# Overview

Lecture 4 - .NET application development

(Architectural Design Patterns)

UNIVERSITY OF WESTMINSTER

Architectural Design Patterns

- MVC
- MVVM
- Component Architecture

MVC

- MVC (**M**odel **V**iew **C**ontroller) is a design pattern used to decouple user-interface (**view**), data (**model**), and application logic (**controller**).

- It helps achieve separation of concerns as each part has its own responsibility.

- Using the MVC pattern for websites, requests are routed to a Controller that works with the Model to perform actions and/or retrieve data and pass it to a View.

UNIVERSITY OF WESTMINSTER

MVC - Controllers

Controllers are classes that:

- Handle browser requests.
- Retrieve model data.
- Call view templates that return a response.

MVC - Models

- Models are classes that represent the data of the app.
- The model classes use validation logic, based on attributes and data annotation, to enforce business rules for that data.
- Typically, model objects retrieve and store model state in a database.

UNIVERSITY OF WESTMINSTER

MVC - Views

- Views are the components that display the app's user interface (UI).
- The Controller chooses the View to display and provides it with the Model.
- The View renders the final page, based on the data in the Model.

UNIVERSITY OF WESTMINSTER

MVC - Summary

- **M**odel **V**iew **C**ontroller

1. Controller processes user requests.
2. Model represents the data.
3. View takes the model and visualizes the data.

UNIVERSITY OF WESTMINSTER

MVC

- MVC is typically used for web development
- Visual Studio provides an MVC template for building ASP.NET applications

DEMO

Web ASP.NET MVC Application

MVVM

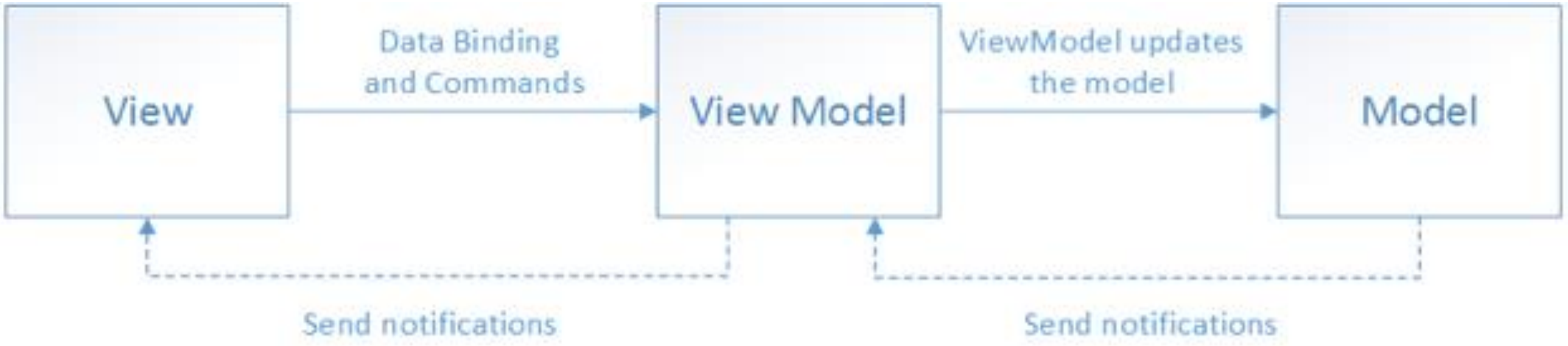
- The Model-View-ViewModel (**MVVM**) pattern helps to cleanly separate the business and presentation logic of an application from its user interface (UI).

- It makes an application easier to test, maintain, and evolve.

- It can also greatly improve code reusability and allow developers and UI designers to more easily collaborate.

MVVM

- There are three core components in the MVVM pattern: the model, the view, and the view model. Each serves a distinct purpose.



MVC vs MVVM

- MVC or MVVM? This is typically not a choice.
  - MVC is used for a “separation of concerns” on the server side.
  - MVVM is works with the reactive architecture, and this is achieved primarily due to Databinding.

- In .NET world, MVC is typically used with ASP.NET (web development), while MVVM is used for Mobile and Desktop development.

- It is possible to replace MVC with MVVM and vice versa but it usually requires significant changes and the use of 3rd party libraries.

DEMO
Mobile Cross-Platform Xamarin Application

Component Architecture

- Component architecture isn’t based on a request-reply model and page-centric architecture.
- User interactions are handled as events that aren't in the context of any particular HTTP request.
- Switching between pages (routing) is done on the client instead of the server side.

Component Architecture

- Components represent a reusable piece of UI. Each component maintains its own state and specifies its own rendering logic.

- Components support nesting which means that one component can be composed out of other components.

- It’s typically used for building client applications and some popular frameworks that are based on this architecture are: Angular, React, Blazor, etc.

Component Architecture

DEMO

Blazor Web Assembly Client Application

Resources

- Documentation
  - https://dotnet.microsoft.com/apps/aspnet/mvc
  - https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/mvvm
  - https://docs.microsoft.com/en-us/dotnet/architecture/blazor-for-web-forms-developers/architecture-comparison

- Books
  - Software Architecture with C# 9 and .NET 5: Architecting software solutions using microservices, DevOps, and design patterns for Azure, by Gabriel Baptista and Francesco Abbruzzese