

Karmasis Research

Tugan Atila

tuganatila@gmail.com

1- .Net and Java Differences

Both .NET and Java are popular and powerful programming languages that are widely used for developing a variety of applications. But they have one key difference which is:

Platform independence: Java is known for its platform independence. It can run on any operating system, while .NET is primarily designed to work on Windows. This doesn't mean .NET will work on Windows only because of the Mono framework which allows .NET to run on Linux MacOS and mobile. Yet again .NET is not a platform independent as Java. There are couple other differences to mention:

Language support: Java supports multiple languages including Java, Groovy, and Kotlin, while .NET primarily supports C#, F#, and Visual Basic.

And also Both .NET and Java have great *Community support*, *Performance* and *Development tools* just .NET focuses on the Windows applications while Java focuses on all platforms.

2- WPF

WPF stands for Windows Presentation Foundation, and it is basically used by developers to create highly customizable, scalable, and interactive graphical user interfaces on Windows for building desktop applications. WPF is a part of the .NET framework and was introduced with .NET 3.0.

WPF uses XAML to define the user interface, which allows developers to separate the UI design from the application logic.

Some of the key features of WPF include support for vector graphics, 3D rendering, animation, data binding, styling, and templates. With WPF, developers can create desktop applications with modern and visually appealing interfaces that can run on any Windows machine.

3- .NET versioning

.NET versioning is the version numbers of different releases of the .NET Framework. Version numbers are used to identify and track changes and updates to the framework, and to help ensure that applications built with different versions of the framework are compatible with each other.

In addition to the version number, each release of the .NET Framework also includes a set of compatible runtime libraries, class libraries, and other components that developers can use to build and run their applications.

Each release of the .NET Framework has a unique version number, consisting of four parts: major version number, minor version number, build number, and revision number. The major version number indicates a significant release that includes major new features or changes, while the minor version number indicates a smaller release with more incremental changes.

4- S.O.L.I.D

The SOLID principles are a set of five design principles in object-oriented programming that aim to make software systems more maintainable, flexible, and easier to understand. "SOLID" stands for:

Single Responsibility Principle (SRP): A class should have only one responsibility. This principle helps to ensure that classes are focused and do not become too complex.

Open/Closed Principle (OCP): A class should be open for extension but closed for modification. This principle helps to create code that is flexible and easy to extend without changing the existing code.

Liskov Substitution Principle (LSP): Any object of a superclass should be able to be replaced with an object of a subclass without breaking the program.

Interface Segregation Principle (ISP): Clients should not be forced to depend on interfaces they do not use. This principle helps to create smaller, more focused interfaces that are easier to implement and maintain.

Dependency Inversion Principle (DIP): High-level modules should not depend on low-level modules. Both should depend on abstractions. Abstractions should not depend on details. This principle helps to write code that is loosely coupled and modular.

5- .NET Console Chat app

First of all, to do this task we need to choose a programming language such as C#, Visual Basic, F# to use the .NET framework. It is hard for me to decide between Visual Basic and C# since VB is made purely for .NET coding and I am familiar with C#. We will see which one is suitable in the process but C# is often favoured by developers so we will be going with it for now.

We also need to choose a communication protocol/architecture. This is the main choice we have to make since the whole programming area will depend on this option and there is quite a variety of available options to choose. Some of the most commonly used chat communication protocols:

WebSocket: This protocol enables real-time two-way communication between a client and a server. I am already familiar with this option. It is widely used for chat applications and other real-time applications.

SignalR: This is a framework built on top of the WebSocket protocol that enables real-time communication between a client and a server. And it is quite the popular option to choose for real time chat system coding on .NET. It is also widely used for chat applications and other real-time applications.

XMPP (Extensible Messaging and Presence Protocol): This protocol is widely used for instant messaging and online presence information which is over-engineered for our goal. It is an open standard protocol that allows users to communicate with each other through messaging clients and chat applications.

IRC (Internet Relay Chat): This protocol is not what we need but it is widely used for group communication in real-time. It enables users to join channels and communicate with other users in those channels.

MQTT (Message Queuing Telemetry Transport): This protocol is designed for lightweight messaging and is widely used for Internet of Things (IoT) applications. It enables devices to communicate with each other in real-time, with low overhead.

After looking at communication protocols, the XMPP and IRC are not suitable for our project. Which leaves us with WebSocket, SignalR and MQTT. Since MQTT is for IOT devices communicating even though we could use this one we will not be using it to choose a more suitable, easy to code protocol.

We will be using **C#** and **SignalR** because of the fact that they are more popular in the community which means more resources to learn. I'm already familiar with WebSocket but I prefer to learn another option.

Extra - Elasticsearch

Elasticsearch is a powerful and flexible open-source search and analytics engine. Elasticsearch is commonly used for full-text search, **log analysis**, business analytics, and real-time data processing. It supports a wide range of use cases, including e-commerce search, website search, social media monitoring, and **security analytics**.

