# Software Development Frameworks

## JavaScript, HTML and CSS:

The main advantage of using JavaScript, HTMK and CSS is that we already know all the languages we would need, and there are plenty of tools and documentation for making apps with these languages.

### No framework

I think we could possibly make the entire UI ourselves out of HTML and CSS and add interaction through JavaScript. This approach has the advantage of all of us already knowing the languages we would need for this. The disadvantage would be that it would be slow and error prone.

### React

Using a UI development framework such as react would be a good approach for making a mobile app. To do this we would have to use JavaScript, HTML and CSS, as well as use the extra language features provided by React. React also could be built into a native app, meaning that our app could work without an internet connection. React is also the most popular JavaScript framework, meaning that the number of libraries and documentation for it is probably the largest

### Angular

Just like React, but with a different framework. Angular is less popular than react though, meaning that there would be a bit less tools and documentation we would be able to use while making our app.

## C#

While not all of us know this language, we know similar languages and the languages C# is based on. C# has plenty of good development tools for cross-platform and mobile apps, as well as the ability to be compiled for the web.

### Xamarin

While it is slightly out of date, Xamarin is still a popular choice for mobile app development with C#. Working with Xamarin would require us to rewrite a large amount of code for other platforms, as sharing our code between platforms is not as easy as with the other choices. It also is a bit limited number of platforms it supports.

### MAUI

This toolkit was released to replace Xamarin. It is cross platform by default and would only require us to write small amount of code for each additional platform we would like to support. It also has support for writing web apps, meaning that we could potentially make our apps work on mobile, desktop and web with very few changes. Just like Xamarin it provides access to useful Ui design and debugging tools, however it is a new project, meaning that it may still have some bugs, and it may be missing some key features.

### Avalonia

Avalonia is a cross-platform recreation of the WPF framework, and it provides support for a large range of devices. It is a similar project to MAUI, however as it is an older project it provides more features and supports a larger range of devices and operating systems.

## C++

C++ is a language we have learned at university, meaning that we all have basic knowledge and skills required to develop simple applications with it, however it is a complicated language, and compiling projects for multiple platforms is difficult.

### Unnamed Microsoft Toolkit

Microsoft provides their own set of tools for cross platform mobile development with Visual Studio. It allows for compiling programs for IOS and Android as well as for Windows devices, however it appears to lack IOS and browser support. This set of tools appears to be old and outdated.

## Java/Kotlin

Some of us already know Java, which is why I also decided to research it. Java is cross platform, and thus creating programs which work on multiple operating systems should be easy, however the Java ecosystem is complex, meaning that it could be difficult for all of us to learn the language and work on the project simultaneously.

### Android Development Toolkit

This is a set of mobile app development tools provided by Google, and it would give us access to the full range of features provided by the Android operating system. As the name implies, it is Android only, meaning that it does not support IOS or desktop devices, and it does not support deploying its apps to the web browser. It is designed to work with Java; however, it does also have full Kotlin support.

### Codename One

Codename One is a Cross-platform app development toolkit which supports a wide range of devices. It uses CSS for styling its UI, which it then adjusts to work on other platforms.It has support for both Android and IOS, as well as Windows and MacOs, however it doesn’t appear to be possible to deploy a project as a Web App with this toolkit.

# References

TylerMSFT (2021). *Cross-platform mobile development with C++*. [online] Microsoft.com. Available at: <https://learn.microsoft.com/en-us/cpp/cross-platform/visual-cpp-for-cross-platform-mobile-development> [Accessed 15 Oct. 2024].

‌ Avaloniaui.net. (n.d.). *Avalonia UI - Home*. [online] Available at: <https://avaloniaui.net/>.

‌ Anon, (n.d.). *Codename One: Cross-Platform App Development with Java/Kotlin*. [online] Available at: <https://www.codenameone.com/>.

davidbritch (2023). *What is .NET MAUI? - .NET MAUI*. [online] learn.microsoft.com. Available at: <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui?view=net-maui-8.0>.

Android Developers. (n.d.). *Android API reference*. [online] Available at: <https://developer.android.com/reference>.

‌ MDN Web Docs. (n.d.). *Web technology for developers*. [online] Available at: <https://developer.mozilla.org/en-US/docs/Web>.

‌

React (n.d.). *Built-in React Hooks*. [online] react.dev. Available at: <https://react.dev/reference/react>.

‌

angular.dev. (n.d.). *Angular*. [online] Available at: <https://angular.dev/>.

‌