

**VIETNAM GENERAL CONFEDERATION OF LABOR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY**



**KIỀU THÀNH PHÁT – 521H0125
HUỖNH TUẦN ANH – 521H0003
NGUYỄN HOÀNG PHÚC – 521H0509**

MICROSOFT MAUI

**CROSS-PLATFORM MOBILE
APPLICATION DEVELOPMENT**

Advised by
Msc.IT MAI VĂN MẠNH

HO CHI MINH CITY, YEAR 2024

**VIETNAM GENERAL CONFEDERATION OF LABOR
TON DUC THANG UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY**



**KIỀU THÀNH PHÁT – 521H0125
HUỖNH TUẦN ANH – 521H0003
NGUYỄN HOÀNG PHÚC – 521H0509**

MICROSOFT MAUI

**CROSS-PLATFORM MOBILE APPLICATION
DEVELOPMENT**

Advised by
Msc.IT MAI VĂN MẠNH

HO CHI MINH CITY, YEAR 2024

ACKNOWLEDGMENT

We would like to sincerely thank lecturer Mai Van Manh for wholeheartedly supporting us in the learning process as well as teaching in class, helping us acquire much necessary knowledge for the subject and the foundation in many forms. The lively and inspiring teaching style helped us a lot in the learning process.

Ho Chi Minh City, day 12, month 4, year 2024

Representative

(Signature and full name)



Kiều Thành Phát



Huỳnh Tuấn Anh



Nguyễn Hoàng Phúc

**THIS PROJECT WAS COMPLETED AT TON
DUC THANG UNIVERSITY**

I hereby declare that this thesis was carried out by myself under the guidance and supervision of lecturer Mai Văn Mạnh; and that the work and the results contained in it are original and have not been submitted anywhere for any previous purposes. The data and figures presented in this thesis are for analysis, comments, and evaluations from various resources by my own work and have been duly acknowledged in the reference part.

In addition, other comments, reviews and data used by other authors, and organizations have been acknowledged, and explicitly cited.

I will take full responsibility for any fraud detected in my thesis Ton Duc Thang University is unrelated to any copyright infringement caused on my work(if any).

Ho Chi Minh City, day 12, month 4, year 2024

Representative

(Signature and full name)



Kiều Thành Phát



Huỳnh Tuấn Anh



Nguyễn Hoàng Phúc

CONFIRMATION AND ASSESSMENT SECTION

Instructor confirmation section

Ho Chi Minh December, 2023

(Sign and write full name)

Evaluation section for grading instructor

Ho Chi Minh December 2023

(Sign and write full name)

SUMMARY

This report, under the guidance of lecturer Mai Van Manh from the Faculty of Information Technology at Ton Duc Thang University, delves into .NET MAUI and cross-platform mobile application development technologies. Completed at 8:30 AM on May 10th, 2024, this report owes its successful completion to lecturer Mai Van Manh's invaluable lessons on . mobile development and related frameworks.

The primary focus of this report is on .NET MAUI, a framework for building cross-platform mobile applications using C# and XAML. .NET MAUI facilitates the creation of engaging applications across platforms like Android, iOS, macOS and Windows through its model-view-viewmodel architecture and unified API. Particularly applicable for businesses requiring consistent experiences across devices, .NET MAUI provides a robust foundation for crafting intuitive user interfaces reusable on multiple operating systems.

This report also examines complementary technologies like C#, XAML and backend controlled using nodejs api storing data on the mongodb database. C# serves as the main programming language while XAML enables declarative user interface definition. Together with .NET MAUI, they empower developers to efficiently write cross-platform mobile applications.

It's essential to note that while this report endeavors to provide comprehensive insights, there may be potential errors present. The author welcomes constructive feedback and suggestions from teachers, viewing them as opportunities for refining knowledge on these dynamic technologies. Continuous learning is intrinsic to this field, and any advice received will be appreciated for enhancing the quality of future work.

CONTENTS

LIST OF FIGURES	7
LIST OF TABLE	8
ABBREVIATIONS	9
1 INTRODUCTION	11
1.1 Overview .NET MAUI	11
1.1 Reason of Report	12
1.2 Overview of Report	13
2 FUNDAMENTAL THEORY	14
2.1 NET MAUI	14
2.1.1 How .NET MAUI works	14
2.1.2 Development environment	15
2.2 C# & .NET	17
2.2.1 What is C# & .NET	17
2.2.2 How to Work with C# & .NET	18
2.2.3 Meaning of C# & .NET with .NET MAUI	19
2.3 XAML	20
2.3.1 What is XAML	20
2.3.2 How does XAML work ?	21
2.3.3 What makes XAML special ?	21
2.3.4 Meaning of XAML with .NET MAUI	23
3 COMPARION	26
3.1 .NET MAUI and Xamarin	26
3.1.1 Platform Support	26
3.1.2 Performance	27
3.1.3 Conclude	28
3.2 React Native and .NET MAUI	29
3.2.1 Platform Support	29
3.2.2 Performance	31
3.2.3 Conclude	32
4 IMPLEMENTATION EXAMPLE	34
4.1 Project Introduction	34
4.2 System Scope	34
4.3 Functional Requirements	35
4.4 Use Cases Diagram	36

4.5	Activity Diagram	38
4.6	Entry Relation Diagram	43
4.7	Demonstration	45
5	CONCLUSIONS	49
5.1	Result Achieved	49
5.2	Drawback	49
5.3	Future Improvement	49
	WORK ASSIGNMENT AND MEMBERS EVALUATION	50
	REFERENCE	53

LIST OF FIGURES

Figure 1-1 NET MAUI overall architecture.....	11
Figure 1-2 NET MAUI weather application	12
Figure 2-1 flowchart of cross platform development with NET MAUI	14
Figure 2-2 diagram of .NET base class library (BCL)	15
Figure 2-3 diagram of .NET MAUI JIT(just in time) compiled	16
Figure 2-4 .NET and C# icons	18
Figure 2-5 .NET and C# overall architecture.....	19
Figure 2-6 XAML icon	20
Figure 2-7 flowchart describing XAML run time compiler.....	21
Figure 2-8 content properties example demonstration	22
Figure 2-9 XAML and NET MAUI icons.....	24
Figure 3-1 Xamarin icon	26
Figure 3-2 Diagram showing Xamarin and Xamarin traditional architecture.....	27
Figure 3-3 Google Pixel result	28
Figure 3-4 Android One result	28
Figure 3-5 React Native icon	29
Figure 3-6 Diagram showing React Native architecture.....	30
Figure 3-7 OnePlus Nord2 and Samsung S23 performance analysis	31
Figure 4-1 note management use cases diagram.....	38
Figure 4-2 login activity diagram.....	39
Figure 4-3 sign up activity diagram	40
Figure 4-4 CRUD note activity diagram.....	42
Figure 4-5 note management ERD diagram	43
Figure 4-6 sign in interface	45
Figure 4-7 sign up interface	46
Figure 4-8 CRUD note management interface	47
Figure 4-9 create note interface	48

LIST OF TABLE

Table 1 interface description table	48
Table 2 task assignment table	51
Table 3 overall progress table	52

ABBREVIATIONS

.NET MAUI	.NET Multi-platform App U
BDT	Broadband Digital Terminal
ĐH TĐT	Trường Đại học Tôn Đức Thắng
FFT	Fast Fourier Transform
MIMO	Multi-Input Multi-Output
UI	User Interface
API	Application Programming Interface
IL	Intermediate Language
JIT	Just In Time
AOT	Ahead Of Time
CLR	Common Language Runtime
FCL	Framework Class Library
WPF	Windows Presentation Foundation
ASP.NET	Active Server Pages .NET
CRUD	Create, Read, Update, Delete
BCL	Base Class Library
XAML	Extensible Application Markup Language
OS	Operating System
IDE	Integrated Development Environment
NPM	Node Package Manager
CR	Carriage Return
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
URI	Uniform Resource Identifier
JSON	JavaScript Object Notation
SQL	Structured Query Language

HTTP	Hypertext Transfer Protocol
FTP	File Transfer Protocol
DNS	Domain Name System
RAM	Random Access Memory
SDK	Software Development Kit
CPU	Central Processing Unit
GPU	Graphics Processing Unit
URL	Uniform Resource Locator
CL	Command Line
CLI	Command Line Interface
POC	Proof of Concept
R&D	Research and Development

1 INTRODUCTION

1.1 Overview .NET MAUI

‘.NET MAUI’, short for .NET Multi-platform App UI, a technology developed by Microsoft, allows users to customize and run cross-platform applications on completely different environments. By writing an application with a source code, you can use it to run on platforms such as Android, MacOS, IOS, Windows.

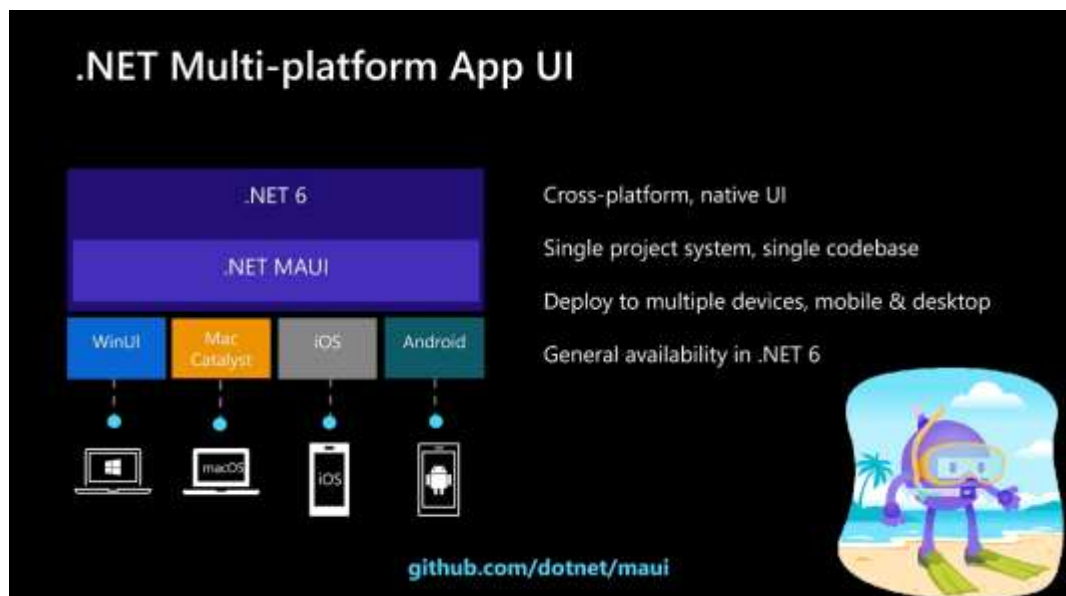


Figure 1-1 NET MAUI overall architecture

‘.NET MAUI’ is a great improvement completed and developed by Microsoft based on ‘Xamarin.Forms’, helping to improve performance and increase scalability, helping users feel more comfortable and familiar when using an application. Use on many different platforms, avoid wasting time learning the rules of use as well as how the application works on different platforms.

Because application development using XAML and C# is so familiar to someone who is a fan of .NET, I think this is such a great technology that was born for you, because the way we We often write source code for ASP.Net, Winform,... They all belong to the .NET technology platform, so using .NET MAUI will also

make it easier for you to perform development and debugging operations. It's easier to fail the application the way you used to, it enables sharing of user interface and design, sharing of code, testing and business logic across platforms. If in the Java language we often have the famous saying "write once and run anywhere", that same sentence is once again reproduced in .NET MAUI technology, allowing developers to more easily application deployment because it unifies Android, iOS, macOS, and Windows APIs.



Figure 1-2 NET MAUI weather application

1.1 Reason of Report

In the context of today's technological development, application development has become easier than ever and the convenience that .NET MAUI has brought helps us create an ecosystem in a simple way. easier and lower cost.

That is the main reason I made this technology research report, and it is also a part that marks our exploration, learning and improvement of technological knowledge, demonstrating a serious attitude in learning and implementation.

We hope this report can contribute in some way to helping developers gather a small piece of knowledge and bring it to you.

1.2 Overview of Report

In this report, we have researched .NET MAUI technology according to the following processes:

- ❖ Deep dive into .NET MAUI technology: General introduction, how it works, its architecture, the platforms it supports (Including development and testing environments), how to deploy.
- ❖ Introduction Languages/frameworks related to .NET MAUI.
- ❖ Comparison between .NET MAUI and other framework platforms.
- ❖ Deploy the .NET MAUI sample application.
- ❖ Conclude.

2 FUNDAMENTAL THEORY

2.1 NET MAUI

2.1.1 How .NET MAUI works

Initially when writing the application, a common source code will be used for the platforms, compiled through .NET MAUI corresponding to the separate platforms that need to be launched.

.NET MAUI applications are created framework for applications depending on the platform, on Android, .NET for Android and similar for iOS and Mac. Particularly on windows it is called WinUI3. All of the above platforms access the same .NET Base Class Library (BCL). Here the BCL will abstract the details of the platforms from the original code.

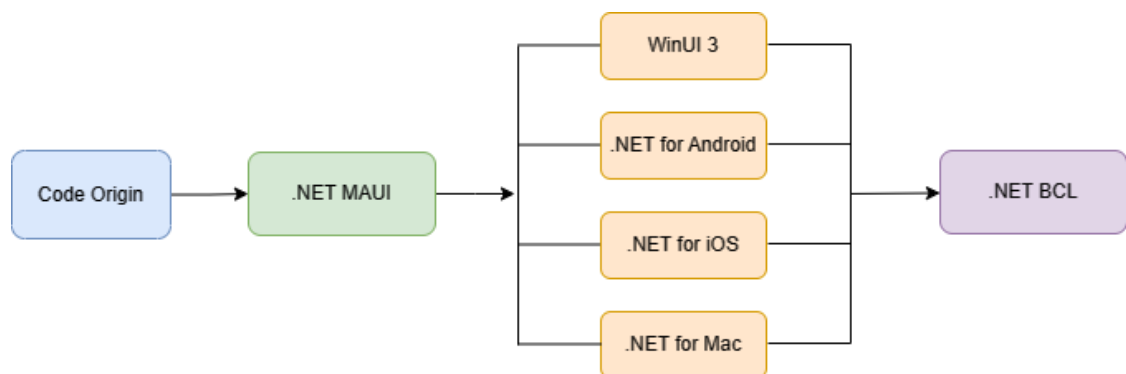


Figure 2-1 flowchart of cross platform development with NET MAUI

Thanks to that, BCL allows applications running on different platforms to share common business logic, will have separate interface formations as well as ways to interact with interfaces on each platform, creating perfect compatibility. In each version, so we can have an application that can run on multiple platforms creating synchronization in an application development ecosystem.

Depending on the environment where the .NET runtime application runs, the code will be executed in each different environment, for example with Android, iOS and macOS, an implementation of the .NET runtime is called Mono Runtime. As for Windows, it will be different from the remaining platforms, it will run on an

execution environment called .NET CoreCLR.

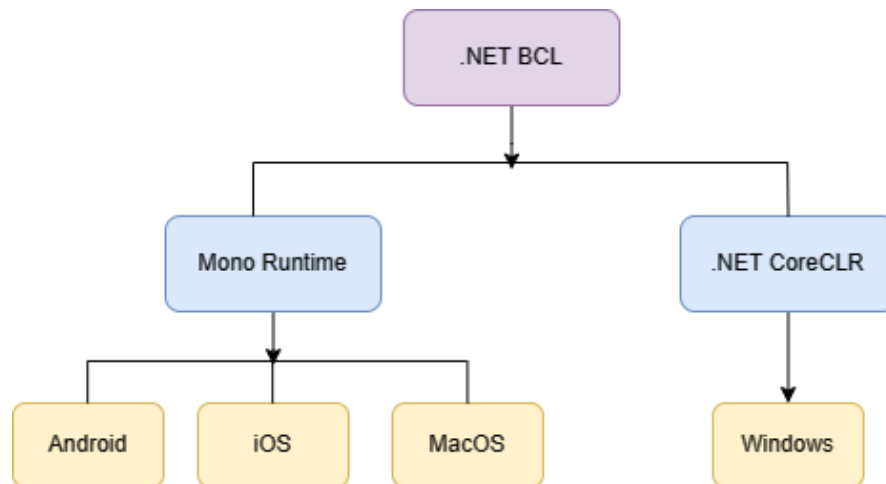


Figure 2-2 diagram of .NET base classes library (BCL)

When writing an application with .NET MAUI, the project's code will interact with the .NET MAUI API. This API provides tools and capabilities for building user interfaces, handling events, and more. When wanting to use platform specific features that .NET MAUI does not support directly. In this case, it is possible to write code that directly interacts with the platform's API. For example, you may want to use a specific Android API to take advantage of an Android-only feature such as requesting permission to change the system's default application or system file interaction, etc.

.NET MAUI acts as an intermediary layer between the code and the platform's API. When writing code that interacts with the .NET MAUI API, .NET MAUI will convert that code into commands that match the API of the platform the application is running on. This helps your application run on multiple platforms without having to write separate code for each platform.

Of course, it is also possible to define separate functions for each platform, but this requires a high level of complexity and understanding in writing advanced .NET MAUI applications.

2.1.2 Development environment

.NET MAUI has separate support for each different platform

❖ Develop for android applications

In Android development, when writing an application in C#, the code will be compiled into an intermediate language called IL (Intermediate Language). IL is a machine independent language, it does not depend on any particular platform.

When the application is launched, the IL will be recompiled using the JIT (Just In Time) Compiler. The JIT compiler will compile the IL to the native assembly of the platform the application is running on (namely Android). This native machine code is then launched directly on the device's hardware. This process helps optimize the performance of the app because the native code is generated at the time the app launches and is optimized for the specific platform it is compatible with.

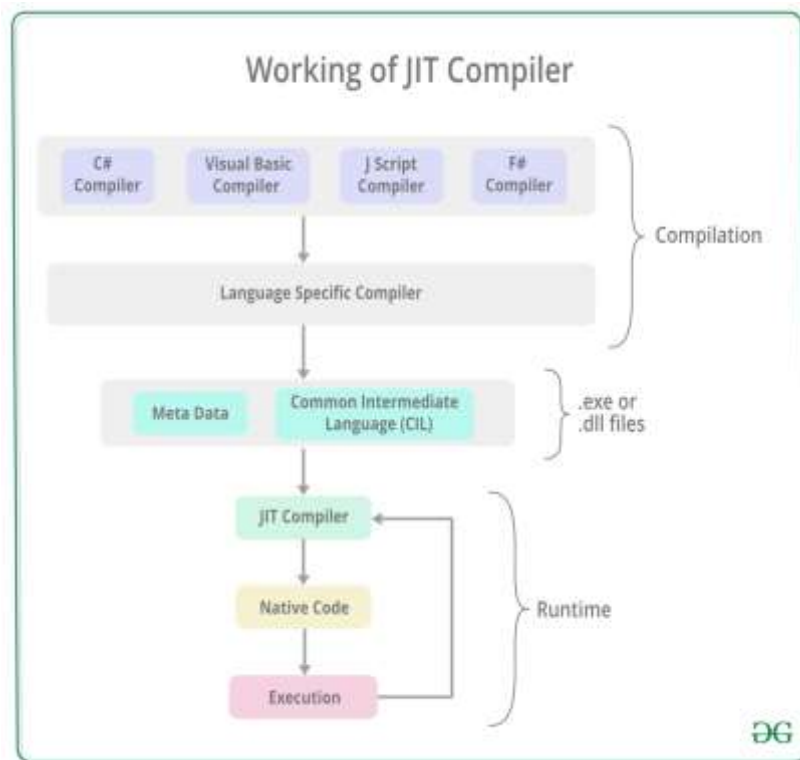


Figure 2-3 diagram of .NET MAUI JIT(just in time) compiled

❖ Develop for iOS applications

When you write an app in C#, instead of compiling to IL (Intermediate Language) and then using the JIT (Just-In-Time) Compiler like on Android, your iOS app will be completely compiled ahead of time. (Ahead-Of-Time) from C# to ARM native machine code. This AOT compilation process takes place when

building the application, before the application is run on the device. This ARM code is then run directly on the device's hardware.

The effect of AOT compilation is that it increases application performance because there is no need to compile the code at runtime, it runs directly..

On the other hands:

- The created application cannot run on non-ARM architecture devices.
- Dynamic code generation is not allowed. Some libraries like `System.Reflection.Emit` cannot work, or `System.Runtime.Remoting` library is not supported and there are some rules imposed on C#.
- Not all generic initialization can be determined at compile time.

❖ Develop for macOS applications

When developing macos applications using .NET MAUI will use Mac Catalyst. Mac Catalyst is an Apple technology that converts iOS apps built with UIKit into macOS desktop apps. When you build a .NET MAUI app for macOS, it uses Mac Catalyst.

When an app is converted to a macOS app via Mac Catalyst, it may require some extended APIs from AppKit (another Apple framework built for the macOS interface) and cross-platform APIs. Supported on macOS. These APIs have been added to ensure the application works properly on the macOS platform, optimizing and taking advantage of features and compatibility with the operating system.

❖ Develop for Windows applications

Using the Windows UI 3 (WinUI 3) library allows developers to build .NET MAUI applications for windows that look beautiful and are compatible with Windows.

2.2 C# & .NET

2.2.1 What is C# & .NET

As part of Microsoft's .NET strategy, the company developed C#, a contemporary object-oriented programming language. High-level abstractions of difficult low-level tasks are provided by C#, which is meant to be straightforward, strong, type-safe, and easy to use. It is one of the main languages used in Windows application development.



Figure 2-4 .NET and C# icons

Microsoft created the .NET software development framework and ecosystem to make it simple to engineer desktop and online applications. It is a powerful framework that comes with an enormous class library called the Framework Class Library (FCL) and supports a number of programming languages, including C#, F#, and VB.NET. The Common Language Runtime (CLR) is the runtime part of .NET that manages memory, threads, program execution, and other system functions.

2.2.2 How to Work with C# & .NET

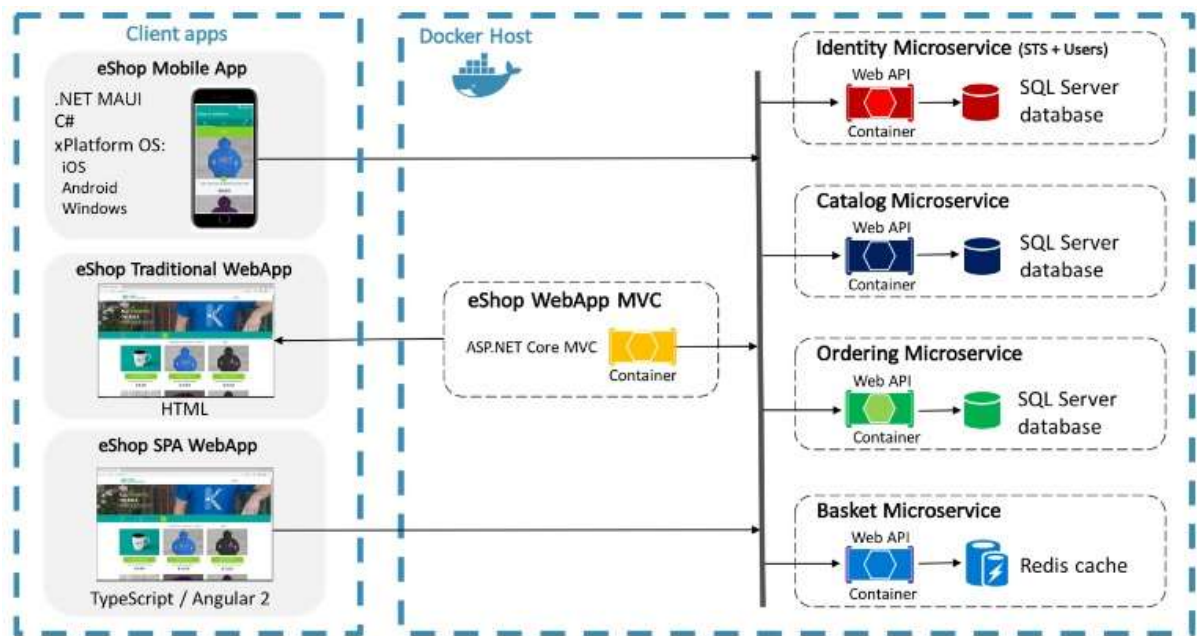


Figure 2-5 .NET and C# overall architecture

The usual steps to begin working with C# and .NET are as follows:

- **Configure the Development Environment:** Install an Integrated Development Environment (IDE) that has all the tools, libraries, and compilers required for .NET development, such as Visual Studio.
- **Draft the code:** Write programs with C#. Console programs, GUI apps with Windows Forms or WPF, web applications with ASP.NET, and mobile applications with Xamarin can all be made.
- **Compile the Code:** The .NET compiler generates Intermediate Language (IL) code by combining the C# code. At runtime, the CLR compiles the IL code into native machine code.
- **Open the application:** On the target system, the CLR runs the application and takes care of memory management, security, and other system functions.

2.2.3 Meaning of C# & .NET with .NET MAUI

The Xamarin platform has evolved into .NET MAUI (Multi-platform App UI). Forms toolkit that enables programmers to use a single codebase to construct native cross-platform applications. In order to create UIs that can run on Windows, macOS, iOS, and Android, .NET MAUI expands the .NET platform by adding new

frameworks and tools.

By utilizing the .NET environment to construct adaptable programs that can operate on nearly any desktop or mobile platform, you can design applications that are versatile when you utilize C# and .NET with .NET MAUI. This unified approach streamlines development processes, permits cross-platform sharing of business logic, and preserves the flexibility to interact with platform-specific features and tailor user experiences to the conventions and behaviors of each platform.

The use of C# and .NET in conjunction with .NET MAUI fundamentally expands the functionality and reach of .NET applications by offering a contemporary, cross-platform framework that seamlessly meshes with the current .NET environment and facilitates the development of apps more quickly.

Useful Examples: Suppose a retail business uses .NET MAUI to make shopping on Android and iOS smartphones and Windows and macOS computers smooth. Adapting to platform-specific interfaces and user interactions, the unified codebase supports features like product catalogs, checkout procedures, and notifications.

2.3 XAML

2.3.1 What is XAML



figure 2-6 XAML icon

Extensible Application Markup Language (XAML) is a declarative markup language. Declarative meaning to describe what you want to achieve without specifying how. Markup is a way to describe how the language is type being a tag based annotation So, XAML is a tag based markup language that lets developers implement the 'what' instead of the 'how'. XAML work by mapping the instantiation of an instance object into it .NET type define in the assemblies, this direct tie to the baking system is uncommon for other interpreted lanagauge. XAML also enable a workflow flow where different developer can work on different part of the application like the UI or logic

2.3.2 How does XAML work ?

XAML is a declarative markup language that focuses on the what to implement more than the how to implement. the XAML processor is assign to implement the how part for the developer it work by:

1. Interpreted the XAML file by it platform specifit XAML processor in .NET MAUI cass usually a specific set of backing types defined in assemblies
2. Then the code is transform internally to code that create the UI
3. Bases on the Model-View-ViewModel The View interface code and ViewModel code are combine together into a partial class with will compile by into XAML is compiled directly into intermediate language (IL) with the XAML compiler

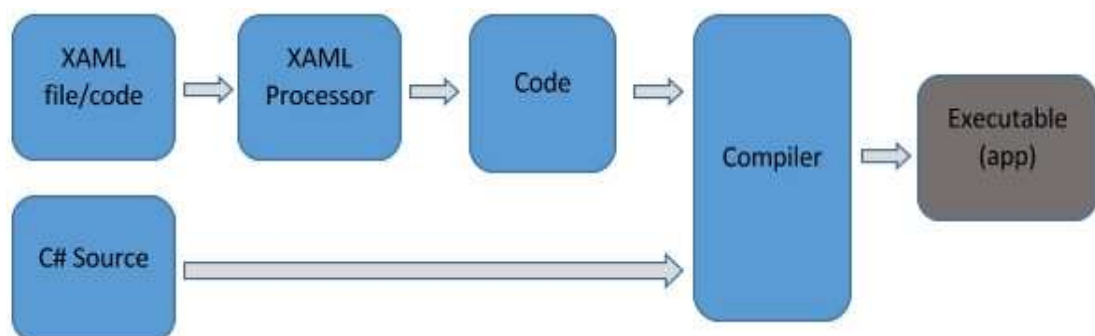


Figure 2-7 flowchart describing XAML run time compiler

2.3.3 What makes XAML special ?

XAML Content Properties

The `ContentPropertyAttribute` allows for the direct nesting of child elements or text to be automatically assigned to a content properties attribute of its parent as long as the parent class specifies the `ContentPropertyAttribute`. This made using the class more convenience eliminating the need to specify properties like `content = "click"`, streamlining the parent-child relationship in the markup

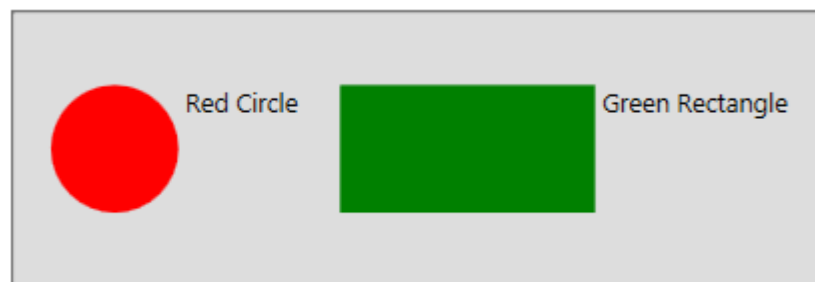


Figure 2-8 content properties example demonstration

Here we can say that the red circle and green rectangle is the content properties of `StackPanel` if we don't use the `ContentPropertyAttribute` it will make us specify only one element to be the contents child instead of multiple

Markup extensions

`MarkupExtension` class in .NET, is inherited by all the foundational extension classes in XAML. This can include those used in .NET MAUI, WPF and other technology

Markup extension purpose is to be used to reference different type of object that already exist or object that will be initialized at run time, an example of this is the binding markup extension `{Binding}` help you assign method or variable of a XAML partial class `ViewModel` logic code to it View code

In .NET MAUI markup extension help define properties for attribute syntax where it is not possible, letting you create complex object right on the attribute syntax without using element syntax

The Markup extensions defer the value of properties to a resource lookup. When the program compiles the properties which were set by the markup extension will not be initialized when XAML code is compiled, instead the actual value is

determined by the markup extension including looking up binding data in the ViewModel code or reacting to a dynamic event. This operation allow for UI flexibility, because the properties value can change bases on the event and condition at runtime

Attached properties

In XAML there is a feature that enables certain properties or events to be attached to an element even though the element is not defined in the attached properties classes , this is call attached properties and attached events are similar to that of global variables in a program that can be call in any class of the code.

For an element to support attached properties or events ,it need to support the backing property store ,this is an underline data structure in the .NET framework where you store the properties instead of the element itself, allowing for properties storage event thought the attached properties is not define the element class

The syntax for attach properties is `owner.nameProperty`, this resembles the property element syntax but the owner is the element properties that have a different type from the nameProperty object owner. The owner also provide the assessor like getter and setter for interaction with the backing property store where the value of attached properties are stored

The attached properties most common used is to enable the child element to influence the layout and behavior of it parent element for example :

2.3.4 Meaning of XAML with .NET MAUI

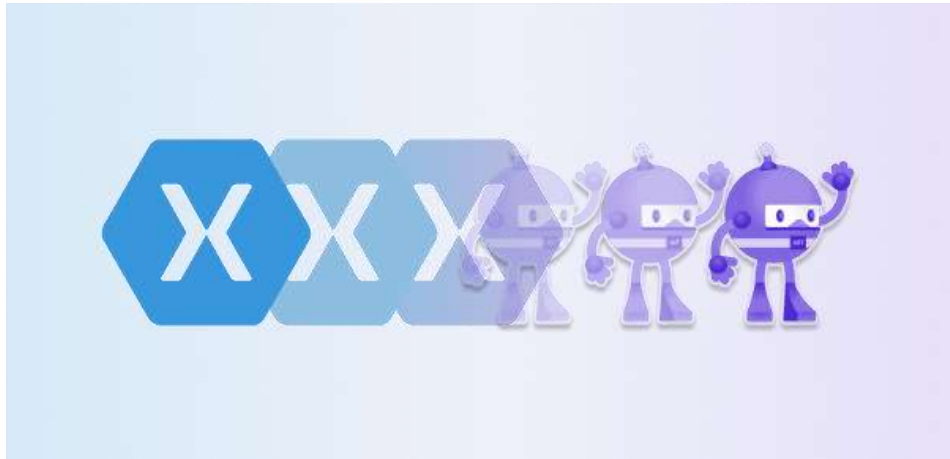


Figure 2-9 XAML and NET MAUI icons

XAML enables the developer to define user interface code by markup similar to languages such as HTML and CSS. Model-View-ViewModel is also a structure that is best implemented in XAML allowing the developer to link their viewmodel code logic with their view UI element via data bindings. XAML is recommended to be the standard for designing interfaces in NET MAUI due to its simplicity for better code readability without sacrificing the implementation logic. In an XAML file the developer can define the user interface contained in NET MAUI that include the layout, pages and custom class. There are two ways to run an XAML file; one is to embed the file, meaning to store it in the package directly without the need for converting into IL code; this is for faster debugging and helps with hot reload. The second way is for the XAML file to be converted into IL code which will help reduce the latency and improve overall performance.

Advantage that XAML bring

XAML is often more maintainable and readable when compare with other UI coding language

XAML parent child hierarchy meaning that an element contains another element within itself. This structure, representing the user interface objects, makes the layout organization to the user interface much more easier to understand

Disadvantage that XAML bring

XAML is a markup language, which means that it does not contain logical code

so it must rely on the ViewModel code to handle business logic. Because of this, there is also no loop in XAML so it has to use tags like ListView and CollectionView.

XAML does not contain any conditional statement or processing like other programming language, so it used mostly data binding and binding converters for transforming and changing the data as it pass from the source to the target of the binding

XAML cannot initialize a new class if the class is not of type parameterless constructor, but there may be some operation to bypass this

XAML can also cannot call any specify method that is not in it ViewModel code but this can sometime be bypass sometime

3 COMPARION

3.1 .NET MAUI and Xamarin

3.1.1 Platform Support



Figure 3-1 Xamarin icon

Xamarin was born in mid-2011 and was developed with the purpose of deploying applications on multiple platforms of Common Language Infrastructure (CLI) and Common Language Specifications (Commonly known as Microsoft .NET). It is built based on C# language, programmers can use Xamarin to program Windows, Android and iOS applications on the same code base. Xamarin.Forms is an extended variant of Xamarin.

Xamarin Forms vs Xamarin Native



Figure 3-2 Diagram showing Xamarin and Xamarin traditional architecture

User Interface of Xamarin.Forms allows reuse of UI code across different platforms, but often suffers from performance and customization limitations. With Xamarin, advanced knowledge of developers may be required to write platform-specific code to optimize user experience or to take advantage of platform-specific APIs.

.NET MAUI is the latest development from Xamarin.Forms. Provides the ability to develop cross-platform applications from Android, iOS, MacOS to Windows using only one common source code. As an improvement of Xamarin.Forms, .NET MAUI also has the features that Xamarin.Forms has, and is more improved than before.

3.1.2 Performance

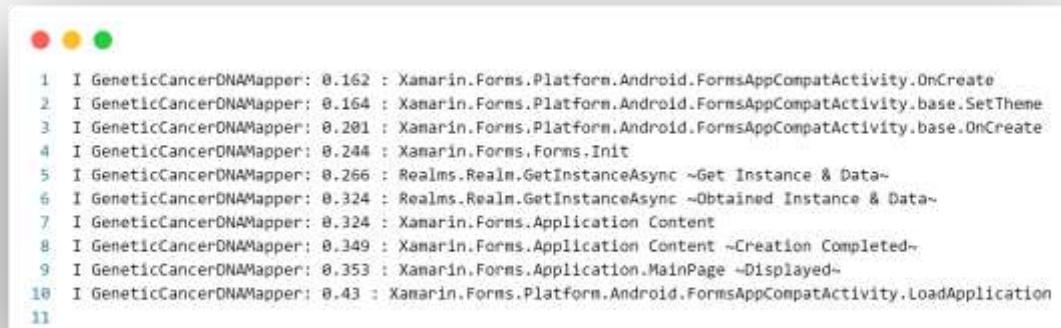
❖ Xamarin

Xamarin is considered to have slow application launch times and slower application response speeds compared to other frameworks.

To prove its delay, here is the research of a DevOps programmer on stackoverflow with over 73k reputation, account name "SushiHangover". He performed a real experiment to demonstrate the speed of Xamarin in real time, the

Xamarin version he used was "Xamarin.Forms v2.4.0.269-pre2", the project simply used mix of coded and XAML based Pages, Controls, etc...

He tested on a "Google Pixel (Oreo-based)" device and the result was that the app is usable in ~430ms.



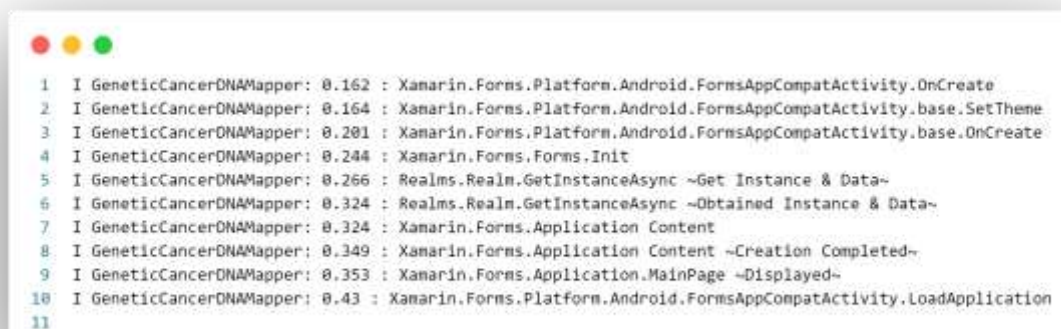
```

1 I GeneticCancerDNAMapper: 0.162 : Xamarin.Forms.Platform.Android.FormsAppCompatActivity.OnCreate
2 I GeneticCancerDNAMapper: 0.164 : Xamarin.Forms.Platform.Android.FormsAppCompatActivity.base.SetTheme
3 I GeneticCancerDNAMapper: 0.201 : Xamarin.Forms.Platform.Android.FormsAppCompatActivity.base.OnCreate
4 I GeneticCancerDNAMapper: 0.244 : Xamarin.Forms.Forms.Init
5 I GeneticCancerDNAMapper: 0.266 : Realms.Realm.GetInstanceAsync ~Get Instance & Data~
6 I GeneticCancerDNAMapper: 0.324 : Realms.Realm.GetInstanceAsync ~Obtained Instance & Data~
7 I GeneticCancerDNAMapper: 0.324 : Xamarin.Forms.Application Content
8 I GeneticCancerDNAMapper: 0.349 : Xamarin.Forms.Application Content ~Creation Completed~
9 I GeneticCancerDNAMapper: 0.353 : Xamarin.Forms.Application.MainPage ~Displayed~
10 I GeneticCancerDNAMapper: 0.43 : Xamarin.Forms.Platform.Android.FormsAppCompatActivity.LoadApplication
11

```

Figure 3-3 Google Pixel result

But with a 512MB "Android One" device, the application takes nearly 4.5 seconds to launch.



```

1 I GeneticCancerDNAMapper: 0.162 : Xamarin.Forms.Platform.Android.FormsAppCompatActivity.OnCreate
2 I GeneticCancerDNAMapper: 0.164 : Xamarin.Forms.Platform.Android.FormsAppCompatActivity.base.SetTheme
3 I GeneticCancerDNAMapper: 0.201 : Xamarin.Forms.Platform.Android.FormsAppCompatActivity.base.OnCreate
4 I GeneticCancerDNAMapper: 0.244 : Xamarin.Forms.Forms.Init
5 I GeneticCancerDNAMapper: 0.266 : Realms.Realm.GetInstanceAsync ~Get Instance & Data~
6 I GeneticCancerDNAMapper: 0.324 : Realms.Realm.GetInstanceAsync ~Obtained Instance & Data~
7 I GeneticCancerDNAMapper: 0.324 : Xamarin.Forms.Application Content
8 I GeneticCancerDNAMapper: 0.349 : Xamarin.Forms.Application Content ~Creation Completed~
9 I GeneticCancerDNAMapper: 0.353 : Xamarin.Forms.Application.MainPage ~Displayed~
10 I GeneticCancerDNAMapper: 0.43 : Xamarin.Forms.Platform.Android.FormsAppCompatActivity.LoadApplication
11

```

Figure 3-4 Android One result

❖ .NET MAUI

3.1.3 Conclude

.NET Multi-Platform app UI (.NET MAUI) designed for building cross platform application, with NET MAUI you can write in one code base and run on any operating system such as IOS , Android , macOS , Windows. NET MAUI which was rebuild from the is the next evolution of Xamarin.Forms with the

controlling UI build from scratch , if you have used or experience with the Xamarin.Forms than adjusting to NET MAUI will be easier , this does not mean that there will not be any different. Using the NET MAUI you will only need to build one project and that project will use the resources and API from different platform specific operating systems. The aim of NET MAUI is for you to build the application in a single code bases

NET MAUI and Xamarin shared the same development environment Visual Studio , but with Xamarin needing an extra extension the installation of Xamarin.Forms and Xamarin Studio must be included . Net MAUI is also integrated with the .NET 6 development framework enabling you to utilize additional library and feature for your enterprise development application code this is oppose to Xamarin which is still stuck on .NET Core so it is recommended for dedicated developer to move to .NET MAUI

3.2 React Native and .NET MAUI

3.2.1 Platform Support



figure 3-5 React Nateive icon

React Native a popular javascript framework which was made and developed

by Facebook in the year 2015, React Native have made a significant impact on the mobile app development industry reshaping the event landscape from its initial release to known. React Native power come from It ability to be run javascript code on mobile devices, for excellent Ui and UX experience. It achieves this by implementing the javascript library on different code bases like that of android, IOS and different OS systems. But also keeping the integrity of a single code base. There is many way to implement React Native in your code the most common one is to build the app from the ground up, or integrate API or UI using

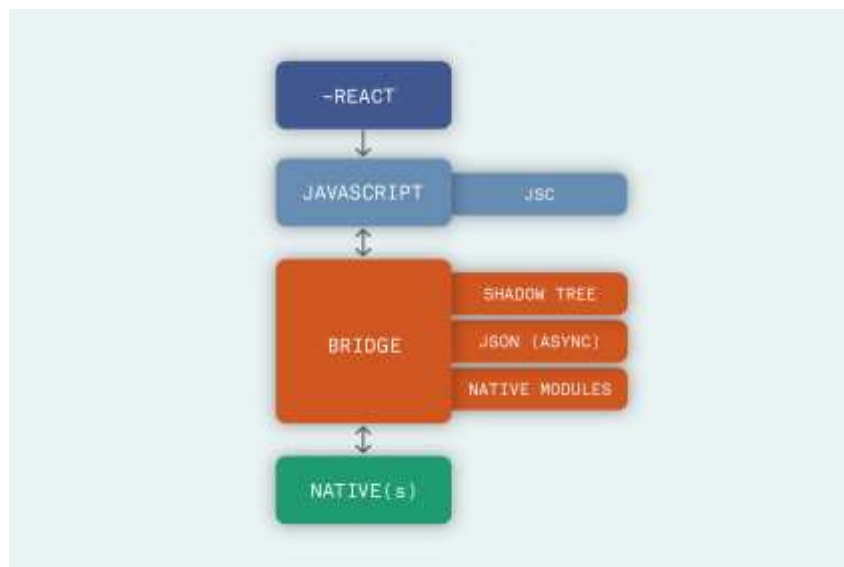


Figure 3-6 Diagram showing React Native architecture

The community surrounding React Native has greatly improved its wealth of third-party libraries and plugins. It has good interaction with various code editors, especially Visual Studio Code. Combined with tools like Expo and well-maintained documentation from Facebook, it has helped streamline the development experience for all sorts of mobile applications. The primary target of React Native is on platforms like IOS and Android, but there is also a consensus to explain into different platforms. React Native although not as extensive as NET MAUI offer a more specialized experience

There are several key features of the React Native application :

Principle on write once deploy anywhere design : enable developer to have a single consistent code bases and deploy on multiple different application like IOS , Android and Window

Main Language: The main language of React Native is JavaScript , a high level language enabling developer to easily read code

Support for Different Library: React Native enable the ease of used between different library and API allowing for flexible adjustment for UI and UX

NPM Installation: NPM install allow for a non complicated approach for dependency configuration

Hot Reload just like with NET MAUI , React Native also enable hot reloading of UI element so the time took for UI changes become near zero to none

Flipper by Default: Flipper is a specialize development tools to help with the debugging of the application for React Native , it help with live debugging and monitor of performance

3.2.2 Performance

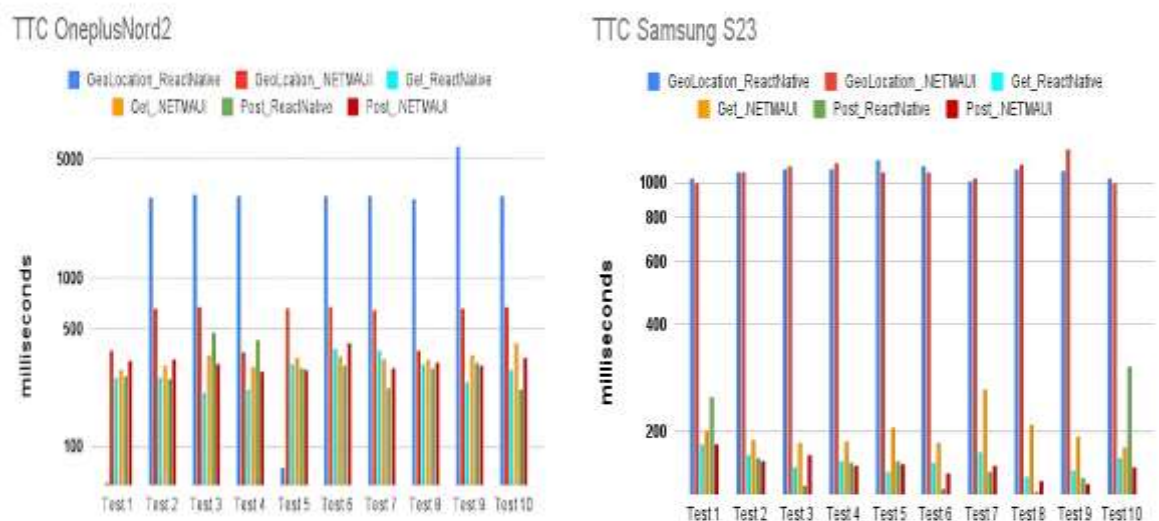


Figure 3-7 One plusNord2 and Samsung s23 performance analysis

The Picture shows the difference in performance for 3 API request tasks for NET MAUI and React Native. The task include getting the location of the user, get data from the Mongoddb database , this research is conducted in the document Evaluating the Performance of React Native and .NET MAUI for Cross-Platform Mobile Development : A Comparative Study by Xuan Dung Tran in Stiv Abdulwahed university Geolocation:

OnePlus Nord 2 phone, in NET MAUI has a smaller range (368ms to 678ms) and lower mean TTC (565ms) than React Native platform (61ms to 5966 ms, mean 2072ms) , making it a faster . On the Samsung S23 phone , NET MAUI has a higher mean TTC (1103ms) and standard deviation (76ms) than that of React Native (with mean 1080ms and standard deviation 42ms), indicating slower consistency

Get List: On both devices OnePlus Nord phone and Samsung S23 phone, .NET MAUI has a higher mean TTC (OnePlus Nord 2: 334ms and Samsung S23: 201ms) than that of React Native (OnePlus Nord 2: 285ms and Samsung S23: 165ms). However, with a lower standard deviance on the OnePlus Nord 2 for .NET MAUI (33ms) compared to that of React Native (55.7ms), making it have a more consistent performance. On the Samsung S23, the standard deviation is higher for .NET MAUI (23ms) than React Native (10.3ms).

Post Item: On the OnePlus Nord 2 phone, .NET MAUI has a higher mean TTC (319 ms) than that of React Native (306ms). On the Samsung S23 phone, .NET MAUI has a low mean TTC (160ms) and standard deviation (12ms) than that of React Native (mean 177ms, standard deviation 56.1ms), making it a faster and more consistent options

3.2.3 Conclude

In concussion React Native principle of “write one and deploy anywhere” coupled with its incorporation of javascript library and framework , make it a good choice for developers who want to maintain a single code base while deploying the

application on multiple platforms like IOS , Android and Windows . Even though React Native has a slower performance when compared to NET MAUI it still provides a consistent performance . React Native is a consistent player in the mobile application development and is a compelling choice for excellent UI and UX developemnt

4 IMPLEMENTATION EXAMPLE

4.1 Project Introduction

This essay will examine the creation of a modern and effective to-do list application with .NET MAUI. Our objective is to develop a user-friendly tool that enables people to efficiently manage their chores. Let's examine the salient characteristics and advantages of our to-do list application based on DotnetMaui:

Cross-Platform Compatibility:

We can create cross-platform apps with a single shared codebase that work on Windows, macOS, iOS, Android, and iPadOS thanks to .NET MAUI.

UI Design:

Our software has an eye-catching and user-friendly UI.

With a few taps, users can quickly add, edit, and remove jobs.

Every task has a clear timestamp that adds organization and context.

CRUD Operations:

For tasks, we've integrated Create, Read, Update, and Delete (CRUD) capabilities. Users have the ability to add new tasks, mark them as finished, change already-entered information, and remove tasks that have been completed.

4.2 System Scope

The system being discussed is a DotnetMaui-developed Todolist application. Users may efficiently create and manage tasks with the help of this program, which functions as an organizational tool.

Main features include:

- Task creation: The floating action button allows users to add new tasks.
- Task management: All tasks have the option to be changed or deleted, shown by the pencil and trash can icons.

- **User interaction:** Task names and content previews are easily visible in the app's user-friendly layout.
- **Registration and Login:** To create and manage user accounts, the program offers login and registration features. By clicking the link " Don't have account? SIGN UP" new users can create an account. Users must provide details like "Email" and "Password" on the registration page. By filling up the appropriate boxes with their email address and password, registered users can access the application by clicking the "SIGN IN" button. Both security and user experience are improved by this feature

By providing users with an easy-to-use yet effective method of tracking their everyday activities, this system seeks to increase productivity. It meets the needs of people looking for a productive way to use DotnetMaui's capabilities to manage their personal duties across many platforms with its simple design and necessary features.

4.3 Functional Requirements

User Login:

The system supports user login functionality. The system will check the user's credentials during login. If the credentials are correct, the user will be redirected to their account. If not, an error message will be displayed.

User Sign Up

The system supports user registration functionality. The registration process requires the user to input their name, email, password, and confirmation of the password. If any credential such as password and confirm password do not match or if the user already exists, an error message will be displayed.

User Note Security:

The system ensures that each user has exclusive access to their own notes within their account. To enforce this security measure, the system implements password encryption for each user's password. This design ensures that users

cannot access notes belonging to others.

User Note Management:

The system provides users with the ability to perform CRUD (Create, Read, Update, Delete) operations on their notes. Users can modify the content of a note, delete a note, or add a new note to the database.

Note Requirements:

Each note requires a title and content. The system automatically generates a unique ID for each note to prevent repetition of IDs. This ensures that each note can be uniquely identified within the system.

Main Technology

4.4 Use Cases Diagram

Use Case 1: User Login

Actors: User

Description: User log in to the system.

Preconditions: The system is accessible, and the user has valid credentials.

Postconditions: User gain access to the system.

Main Flow:

- User initiate the login process.
- The system prompts for email and password.
- User provide valid credentials.
- The system verifies the credentials.
- If authentication is successful, user are granted access; otherwise, an error message is displayed.

Use Case 2: User Sign Up

Actors: User

Description: User register for an account .

Preconditions: The system is accessible.

Postconditions: User have register their account successfully and can now use the login function.

Main Flow:

- User initiate the sign up process.
- The system prompts for name email ,password and confirm password.
- User provide valid credentials.

- The system verifies the credentials.
- If authentication is successful, user are redirected to the login page; otherwise, an error message is displayed.

Use Case 3: Note Management

Actors: Users

Description: Users are responsible for managing note information, including updating, deleting, adding new note,

Preconditions:

The User is authenticated and logged into the Note Management application.

Postconditions: The note data in the system is appropriately modified, added based on the actions performed by the User.

Main Flow:

Update Note Information:

- Selects the note to be updated.
- Modifies the relevant note information (e.g., Note Content , Note Title).
- Submits the updated information.
- The system validates and updates the certificate details.

Delete Note:

- Choose the note to be deleted.
- Confirms the deletion.
- The system removes the selected note from the database.

Add New Note:

- Fills in the required information for the new note (e.g., Note Content, Issuing , Note Title).
- Submit the form.
- The system validates the information and adds the new note to the system

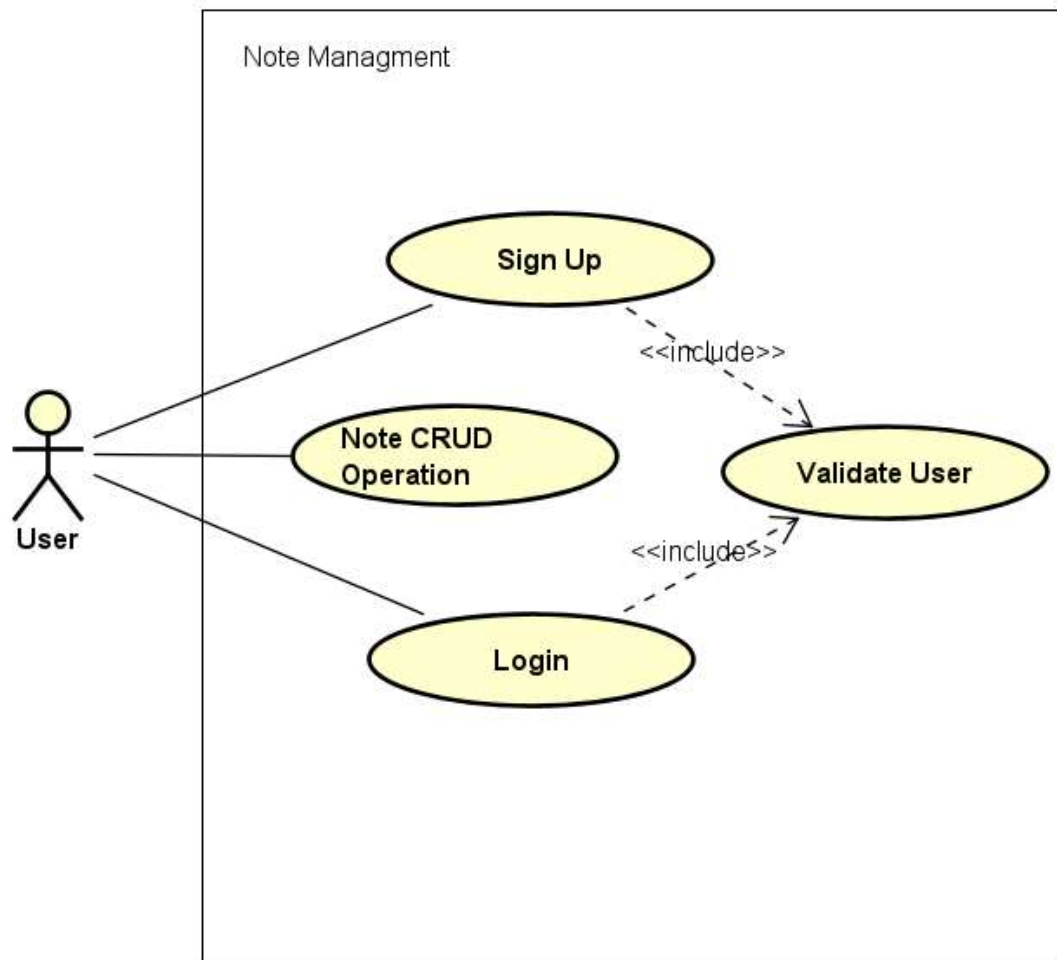


Figure 4-1 note management use cases diagram

4.5 Activity Diagram

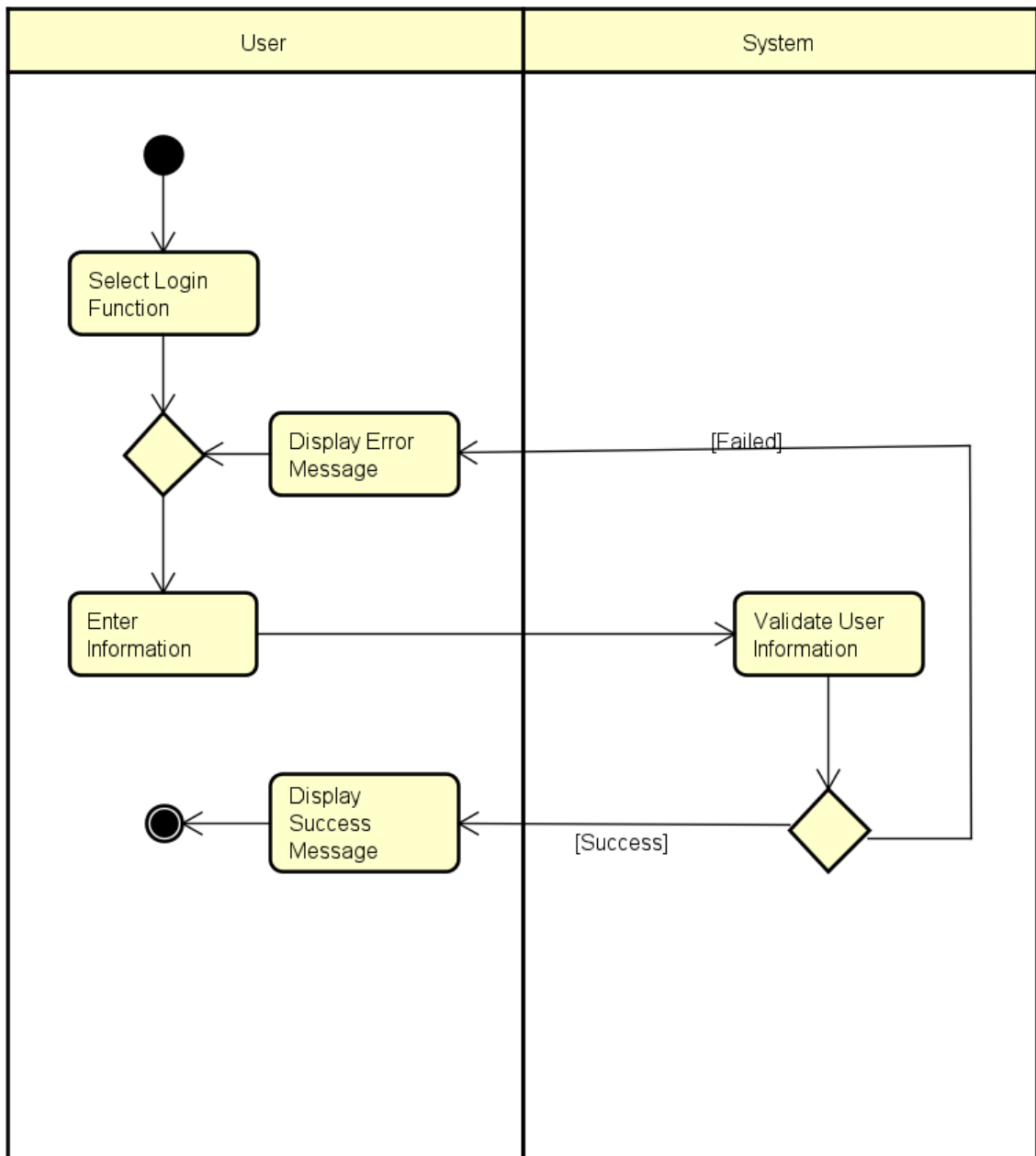


Figure 4-2 login activity diagram

1. **Select Login Function:** This is the first step where the user initiates the login process by selecting the login function.
2. **Enter Information:** In this step, the user is prompted to enter their login information, typically a username and password.
3. **Validate User Information:** Once the user has entered their information, the system takes over. The entered information is validated. This could

involve checking if the username exists in the system and if the entered password matches the one on record for that username.

4. **Display Error Message** (after [Failed] validation): If the validation fails, the system displays an error message to the user. The exact nature of the error might vary, but it typically indicates that either the username doesn't exist or the entered password is incorrect.
5. **Display Success Message** (after [Success] validation): If the validation is successful, the system displays a success message, indicating that the login process has been completed successfully.

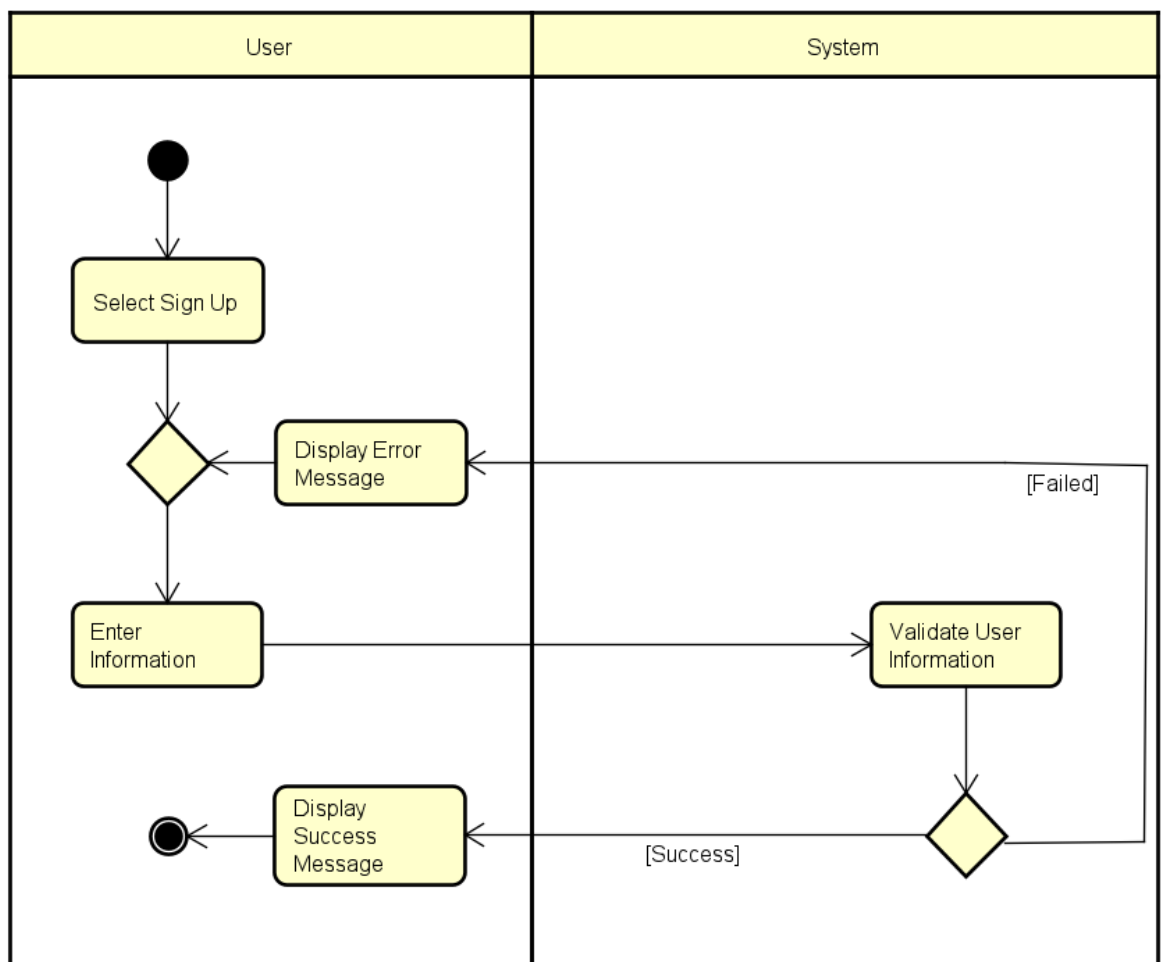


Figure 4-3 sign up activity diagram

1. **Select Sign Up:** This is the first step where the user initiates the sign-up process by selecting the sign-up function.
2. **Enter Information:** In this step, the user is prompted to enter their sign-up

information, typically a username, email, and password.

3. **Validate User Information:** Once the user has entered their information, the system takes over. The entered information is validated. This could involve checking if the username or email already exists in the system and if the entered password meets the system's security requirements.
4. **Display Error Message** (after [Failed] validation): If the validation fails, the system displays an error message to the user. The exact nature of the error might vary, but it typically indicates that either the username or email already exists or the entered password does not meet the system's security requirements.
5. **Display Success Message** (after [Success] validation): If the validation is successful, the system displays a success message, indicating that the sign-up process has been completed successfully.

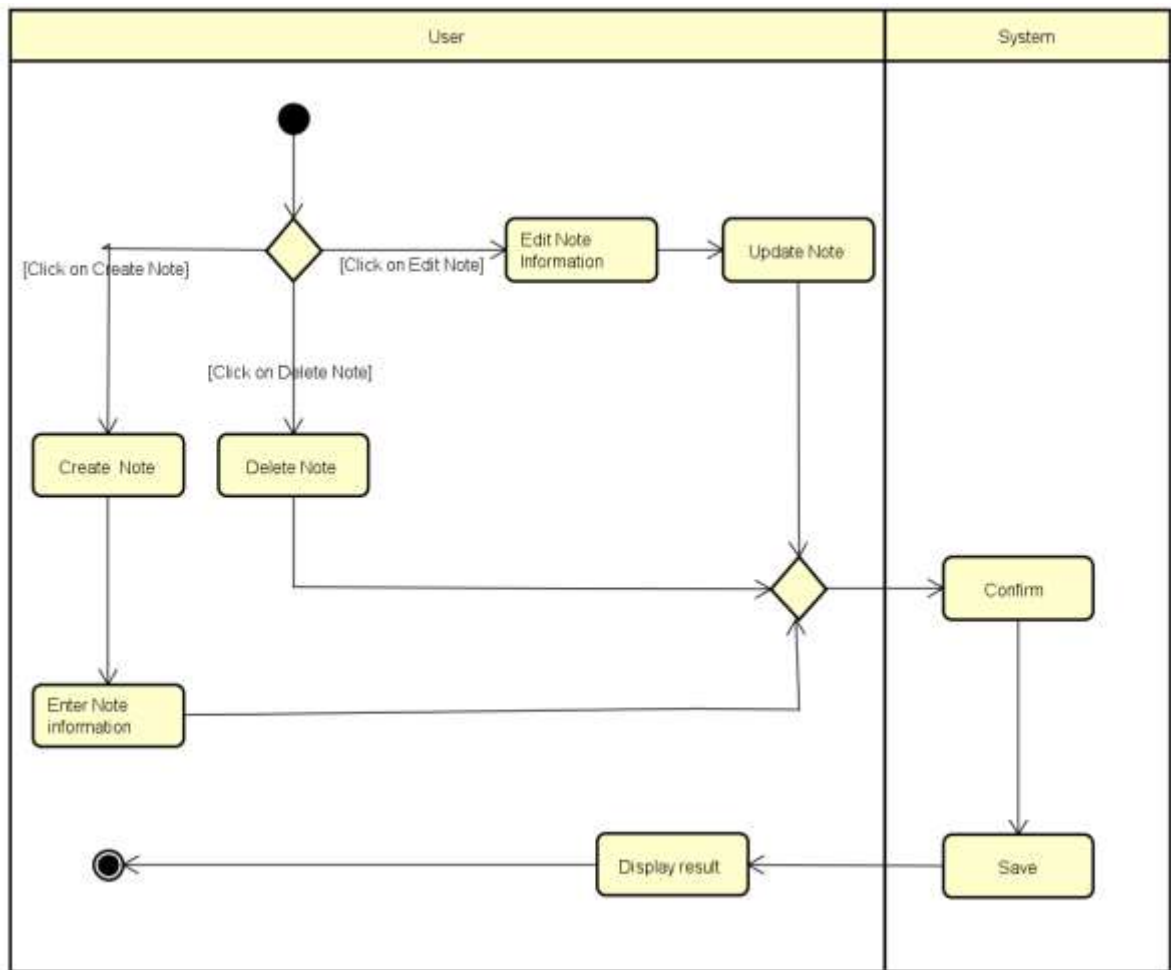


Figure 4-4 CRUD note activity diagram

1. **Click on Create Note:** This is the first step where the user initiates the process of creating a note by clicking on the “Create Note” button.
2. **Enter Note Information:** In this step, the user is prompted to enter the information for the note they wish to create.
3. **Create Note:** Once the user has entered their information, the system takes over. The entered information is used to create a new note in the system.
4. **Click on Edit Note:** If the user wishes to edit the note, they can click on the “Edit Note” button.
5. **Enter Information:** The user is then prompted to enter the new information for the note.
6. **Update Note:** The system updates the note with the new information

provided by the user.

7. **Click on Delete Note:** If the user wishes to delete the note, they can click on the “Delete Note” button.
8. **Delete Note:** The system deletes the note from the system.
9. **Display Result:** The system displays the result of the user’s action, whether it’s the creation, editing, or deletion of a note.
10. **Save:** The system saves the changes made by the user.

4.6 Entry Relation Diagram

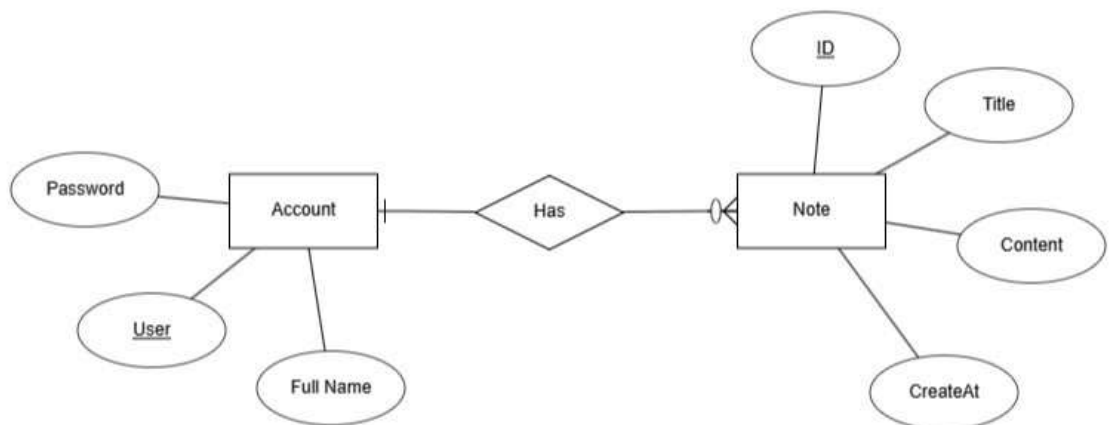


Figure 4-5 note mangement ERD diagram

Note Table:

Attributes:

- ID: Primary Key, uniquely identifies each student.
- Title : Store the title of the note
- Content: Store the Content of the note
- CreateAt : Store the date that the note was created

Reasoning:

The ID enable a unique identifier for each note in the database enabling data integrity

Create at is a date indicating the creation date of the note

Account Table:

Attributes:

- Password : a hash password to verify the user credential .
- User: Stores the email address that is used for the user system name.
- FullName: Stores the name of the user .

• Reasoning:

The reason why there is no ID in the user table is because the system identify the user and their note by the bear token so an ID is not require

Password is hash to ensure data security

4.7 Demonstration

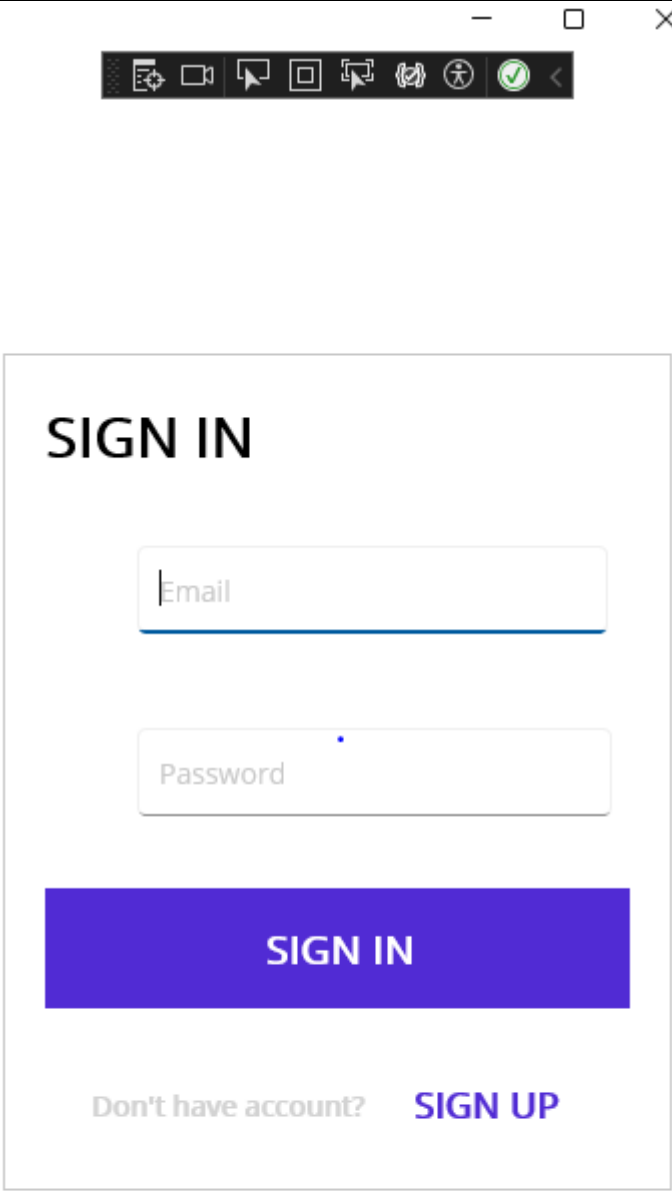
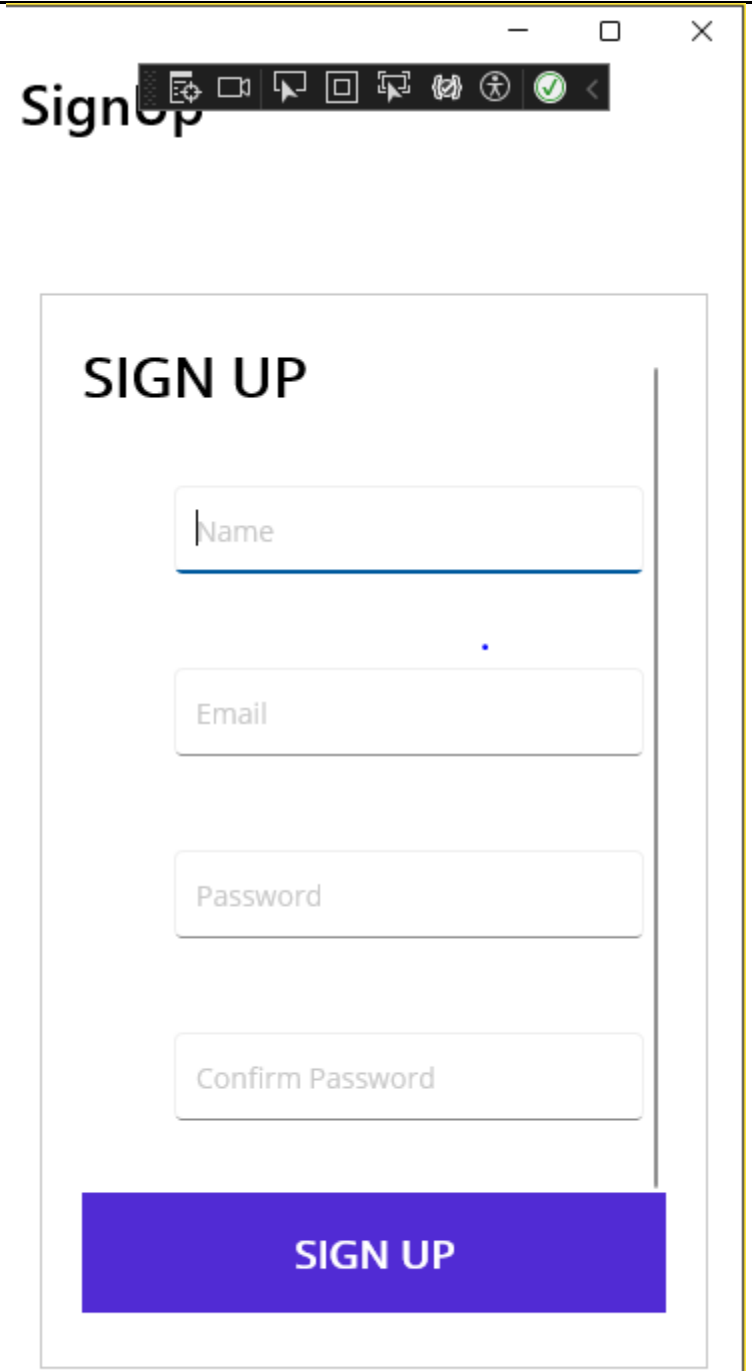
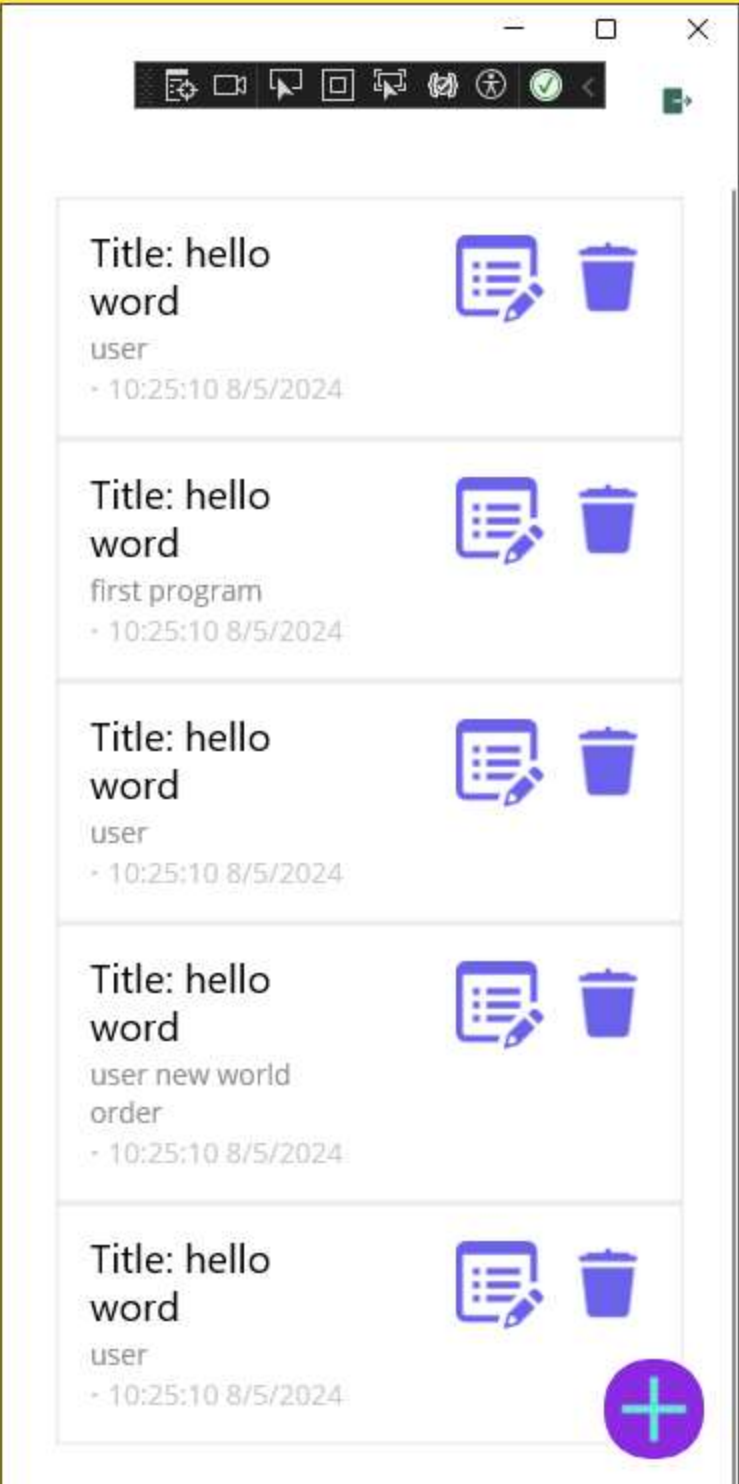
	Specific Description	UI
1	<p>Login Interface:</p> <p>This interface serves as the authentication gateway, facilitating user access to the system through the provision of individualized credentials. To gain entry, users are required to enter both their designated email and password</p>	 <p>The screenshot shows a web browser window with a toolbar at the top containing icons for back, forward, search, and other functions. The main content area displays a 'SIGN IN' form. The form has a title 'SIGN IN' in bold black text. Below the title are two input fields: 'Email' and 'Password'. The 'Email' field is a simple text box. The 'Password' field is a text box with a blue dot indicating a password character. Below the input fields is a large blue button with the text 'SIGN IN' in white. At the bottom of the form, there is a link 'Don't have account? SIGN UP' in blue text.</p>

Figure 4-6 sign in interface

<p>2</p>	<p>Sign Up Interface :</p> <p>Here the user can sign up for an account , this is require if the user currently do not have an account .</p> <p>When sign up successfully the user will be redirected back to the login page</p>	 <p>The screenshot shows a web browser window with a title bar containing standard window controls (minimize, maximize, close) and a toolbar with various icons. The main content area displays a 'Sign Up' form. The form is titled 'SIGN UP' in large, bold, black letters. Below the title, there are four input fields: 'Name', 'Email', 'Password', and 'Confirm Password'. Each field has a light gray border and a small blue dot at the end of the text. Below the input fields is a large, solid blue button with the text 'SIGN UP' in white, bold, uppercase letters. The background of the form is white, and the entire form is enclosed in a thin gray border.</p> <p><i>Figure 4-7 sign up interface</i></p>
----------	--	--

<p>3</p>	<p>Note Managment Interface: In this section, various CURD operations can be used to modify note information , user can also logout of the system</p>	<div data-bbox="774 268 1516 1747"><p>The screenshot displays a web application window titled 'Note Management'. It features a top navigation bar with icons for home, list, add, edit, delete, and user profile. The main content area shows a list of five notes, each with a title, content, and a timestamp. The notes are: 1. Title: hello word, user, 10:25:10 8/5/2024; 2. Title: hello word, first program, 10:25:10 8/5/2024; 3. Title: hello word, user, 10:25:10 8/5/2024; 4. Title: hello word, user new world order, 10:25:10 8/5/2024; 5. Title: hello word, user, 10:25:10 8/5/2024. Each note has a blue edit icon and a blue delete icon. A large blue plus icon is at the bottom right of the list.</p></div> <p>Figure 4-8 CRUD note mangement interface</p>
----------	---	--

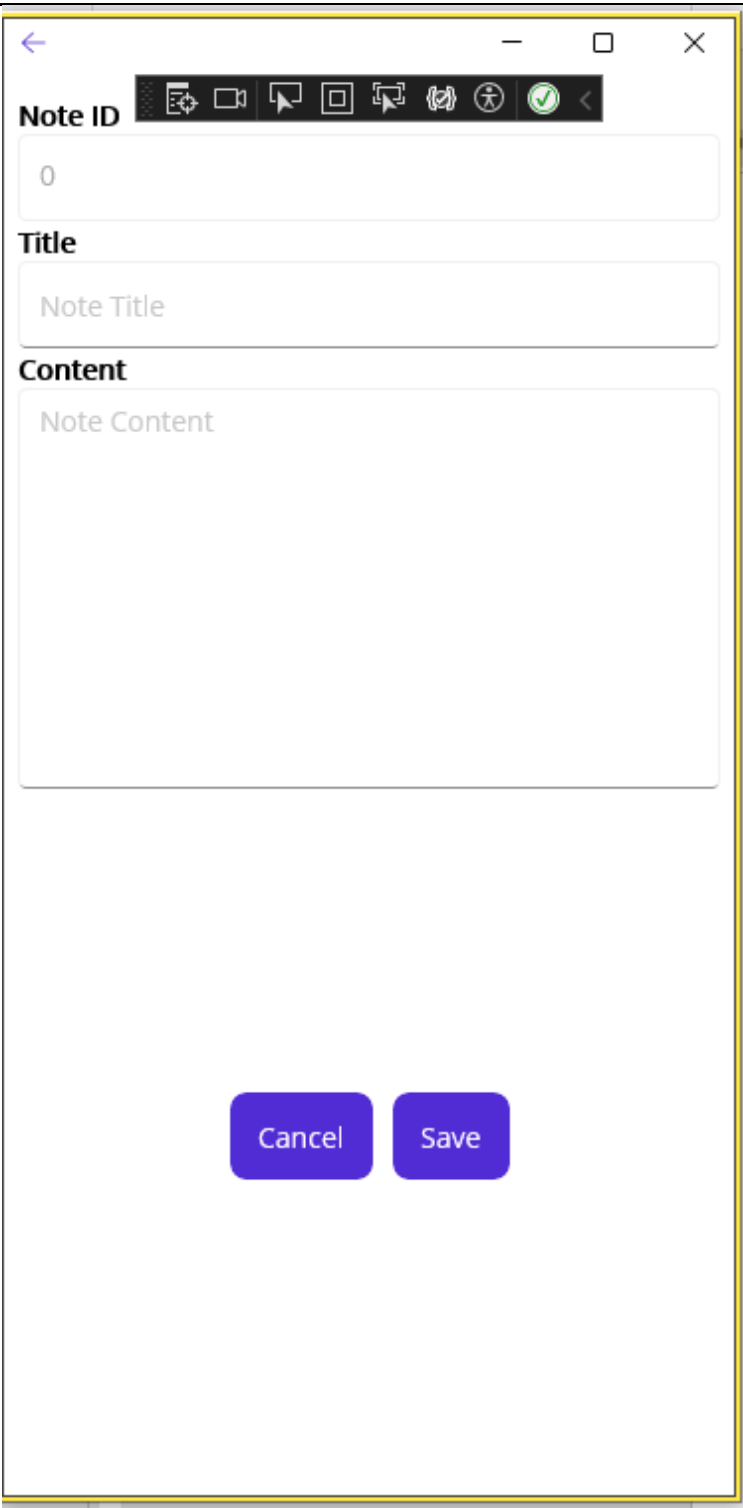
4	<p>Add Note Interface :</p> <p>In this section , if the user edit the note , note information such as title and content are fill and the note ID will be the edited note id number , but if the user create a note instead then both the title and content will be empty with the note ID being 0</p>	 <p>The screenshot shows a mobile application interface for adding or editing a note. At the top, there is a navigation bar with a back arrow on the left and standard mobile window controls (minimize, maximize, close) on the right. Below the navigation bar is a toolbar containing several icons: a document with a plus sign, a video camera, a speech bubble, a square, a rectangle, a circle, a line, a plus sign, a green checkmark, and a back arrow. The main content area has three input fields: 'Note ID' with the value '0', 'Title' with the placeholder 'Note Title', and 'Content' with the placeholder 'Note Content'. At the bottom of the interface are two large, rounded rectangular buttons: 'Cancel' and 'Save'.</p>
---	---	---

Figure 4-9 create note interface

Table 1 interface description table

5 CONCLUSIONS

5.1 Result Achieved

The cross-platform application was successfully achieved using .NET MAUI . The note management application allows the user to create , edit and delete notes on multiple platform including but not limited to Android , IOS Window and MacOS all in a single code bases. It have created a clean and easy to use UI with additional key features like authentication , note CRUD operation and platform compatibility. By building this project we have gain a large amount of knowledge about .NET MAUI and .NET architecture as a whole and how to build cross platform application with a shared codebase

5.2 Drawback

The main drawback of the application is it is limited in use other than creation and management of user notes . This could potentially discourage user experience and user attention when compare to other note taking app on the market. Offline note storage and synchronization have not been support so the application require a internet connectivity

5.3 Future Improvement

To improve the user experience the note taking application should be given additional features like filtering the note , tag on whether a note is urging or not and arrange different UI that meet user specification. Offline note creation with synchronization on app restart would provide a better user experience. Enabling more storage on google drive or a cloud bases service allowing user to access the note when they are on a different device . Finally , publishing the app to Android and IOS platform would allow real world testing in a larger scale

WORK ASSIGNMENT AND MEMBERS EVALUATION

Task Assignment

Name	The content of work
NGUYỄN HOÀNG PHÚC	<ul style="list-style-type: none"> • Design the main screen for displaying the list of note • Design the task detail screen for viewing, editing, and deleting note. • Compare React Native and .NET MAUI in terms of platform support and performance. • Evaluate the members based on their work. • Write about XAML, how it works, what makes • Create use cases diagram. • Create activity diagram. • Demonstrate the application
HUỖNH TUẤN ANH	<ul style="list-style-type: none"> • Implement functionality to create, read, update, and delete tasks. • Test the application on different platforms to ensure compatibility and functionality. • Introduce the project. • Define the system scope. • List the functional requirements. • Write about C# & .NET, how they work, and their meaning with .NET MAUI

KIỀU THÀNH PHÁT	<ul style="list-style-type: none"> • Create a login page for returning users to sign in. • Create a registration page for new users to sign up. • Write an overview of .NET MAUI. • Explain the reason for the report. • Provide an overview of the report • Write about .NET MAUI and how it works. • Assign work to team members. • Compare .NET MAUI and Xamarin in terms of platform support and performance. • Create entry relation diagram
-----------------	--

Table 2 task assignment table

Member evaluation

Full name	Assigned tasks (100%)	Completion level
NGUYỄN HOÀNG PHÚC	33.3%	On schedule as assigned
HUỖNH TUẤN ANH	33.4%	On schedule as assigned
KIỀU THÀNH PHÁT	33.3%	On schedule as assigned

Table 3 overall progress table

REFERENCE

1. Matt Goldman. (2023). .NET MAUI in Action
2. Jesse Liberty (2023) NET MAUI for C# Developers
3. Xuan Dung Tran and Stiv Abdulwahed (2023) Evaluating the Performance of React Native and .NET MAUI for Cross-Platform Mobile Development: A Comparative Study