University of Zakho

Faculty of Science

Department Of Computer

Semester 8 - Group _A

Project: Mobile Application

By

Eman amin *Barin Jamil*

Maryam youwarsh *Hadil Sadiq* *Sozdar Hashim*

Letcher name :

Yusif arshak

2021-2022

Table of Contents

Introduction to Mobile Applications

A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Apps are generally small, individual software units with limited function. This use of app software was originally popularized by Apple Inc. and its App Store, which offers thousands of applications for the iPhone, iPad, and iPod Touch. A mobile application also may be known as an app, web app, online app, iPhone app, or smartphone app. There are currently 3 platforms (Android, iOS, and Windows).

- Android (Java, Kotlin)
- iOS (Swift)
- Windows (C#)

If you ever wondered if it's unnecessary to write the same application 3 times - one for Android, the other for iOS, the third for Windows then you're in the right place.

What is Xamarin?

Xamarin allows to development of native applications for all the platforms at once, with a minimum of OS-dependent scripts and all using only C# .NET and XAML. So we can share about 90% of common code across all platforms.As the name suggests, it's a classic XML with extended syntax (eXtensible Application Markup Language). We'll use it to design the visual layout of our applications, which we'll then "bring to life" with C# scripts.

And how does it work? As I mentioned earlier, it's kinda revolutionary that Xamarin development is native and in C# .NET at the same time, even for platforms with different native languages. The principle is again very simple. When we build the project, it's compiled to the native subsystems (Java for Android, Swift for iOS) and each application then looks and behaves as if it was written in the native language of the platform. Certainly now we all see the major advantages of Xamarin. Another indisputable advantage is that if we don't know Java or Swift and want to create even an application just for Android or iOS, we can do it in C#. Xamarin offers 5 basic solution types:

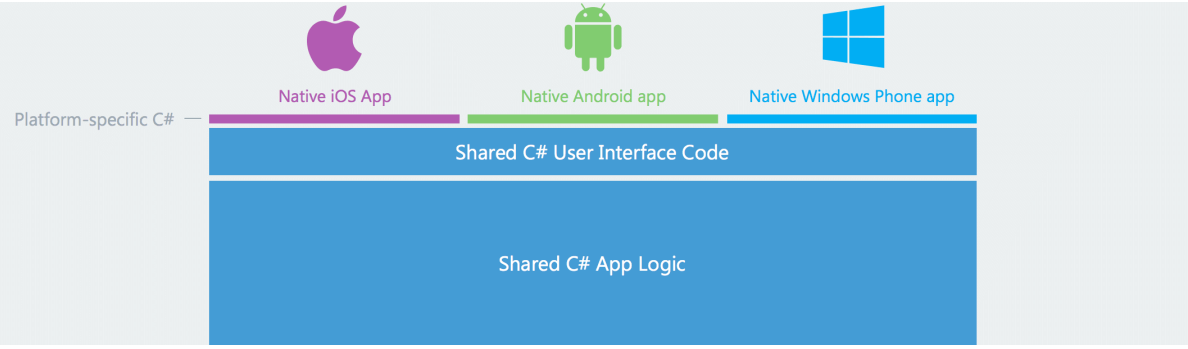
- Xamarin.Forms (development for all platforms at once)
- Android application
- iOS application
- Android Wear application
- WatchOS application

So we can choose what we need comfortably. In this course, we'll focus mainly on the development using Xamarin.Forms. I'm sure you'd be able to use the other solution types if needed

How does Xamarin work?

Cross-platform development frameworks for mobile are still interesting topics of discussion among mobile developers**.** Xamarin is one of the popular cross-platform frameworks from Microsoft. With Xamarin, you can develop mobile applications for iOS, Android, and Windows platforms with a single codebase written in C#. You can also develop desktop applications for macOS and Windows. And it’s not hybrid. Xamarin based apps are compiled native applications that have near-native performance, even for gaming applications.

Xamarin allows developers to develop native applications for Android, iOS, and Windows Phone platforms, with a single codebase, i.e. C#, and a single IDE, i.e. Visual Studio or Xamarin Studio. Thus, a single developer can develop native mobile applications without knowing Java, Kotlin, Objective-C, or Swift. What that means is, that all the C# code has to be converted to make it work on these three separate platforms.
Magic? Well, yes. Xamarin takes care of translating or compiling all your C# code to its corresponding platform-specific code.
Xamarin promises this:

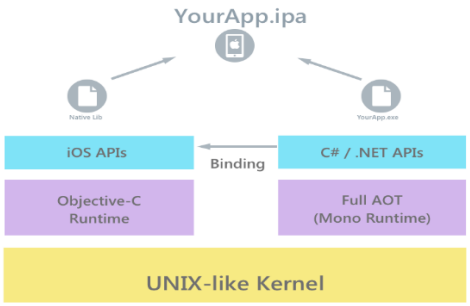


- Xamarin provides a native UI interface.
- Xamarin provides native API access.
- Xamarin provides native performance.

How exactly does Xamarin achieve this?

Xamarin and iOS

For iOS, Xamarin provides a fully compiled (AOT – Ahead Of Time) binary that directly runs on your device to provide native performance. Xamarin.iOS exposes a C#/CIL binding to the Cocoa Touch API and also provides access to ECMA CIL APIs and various other .NET APIs.



Xamarin and Android

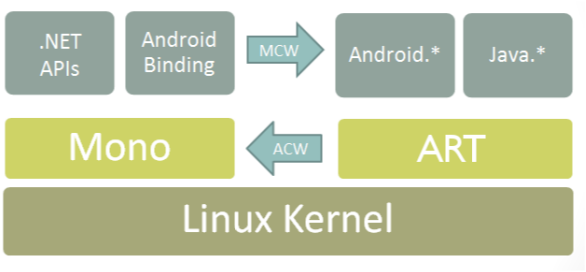


Diagram of Mono and ART above the kernel and below .NET/Java + bindings

For Android, Xamarin leverages the JIT (Just In Time) compilation to create an optimized executable. Xamarin. Android applications run within the Mono execution environment. This execution environment runs side-by-side with the Android Run Time (ART) virtual machine. Both runtime environments run on top of the Linux kernel and expose APIs to the code that allows access to the underlying system. Xamarin. Android applications also contain the Android Callable Wrappers (ACW) to allow Android to call into managed code. Managed Callable Wrappers (MCW) are used whenever managed code needs to call into Android APIs.

System Requirements

Ultimate UI for Xamarin supports the following development environments:

Develop Android and iOS mobile applications on Windows and Mac:

- Windows 10 or higher.
- Visual Studio 2017 or higher
- Xamarin for Visual Studio 4.5+
- For building iOS apps - Configuring iOS build host

IDES:

- Visual Studio 2019
- Visual Studio 2017
- Xamarin Studio
- Visual Studio for Mac

Device:

- Android 6.0 and later
- iOS 8.0 and later

Development:

- Windows 10 and later
- Mac OS 13.2 "Sierra" and later
- Android 9.0 and later

what is Flutter and which wan is better Xamarin or flutter

You can use Flutter, an open-source mobile SDK to create cross-platform mobile apps. It helps you to create mobile apps that look like native apps. You can use one codebase for developing an Android as well as an iOS app. Google introduced Flutter in 2015. The company had initially named it “Sky”, and it ran on Android. Google formally launched Flutter 1.0 in December 2018.Flutter witnessed multiple iterations of development subsequently. At the time of writing this, the last stable release came in October 2020.

Project complexity: Complex projects require you to focus more on development and testing. Excellent tooling support for testing and CI/CD helps in such projects. Use Flutter for complex projects. The speed of development required: Xamarin provides better development tools than Flutter, and it’s easier to use. If you need fast development, use Xamarin. Availability of developers: You will find Flutter developers easier than Xamarin since it’s more popular. Flutter is also more loved than Xamarin. This means that finding Flutter developers will be easier in the future. If finding developers is a crucial factor for you, then choose Flutter. Cost: Commercial-scale development using Xamarin is more expensive than using Flutter. This is due to the license cost of Microsoft Visual Studio IDE. If tooling cost is high on your priority, then choose Flutter. Both Flutter and Xamarin present robust solutions for cross-platform mobile development. Xamarin and Flutter have many advantages, and they have a few limitations too. Consider your project requirements carefully before choosing a development framework.

Advantages or Pros of Xamarin

The pros of Xamarin technology are listed below:

Performance: Xamarin apps are famous for having native-like performances.

Complete development ecosystem: Xamarin apps are build using C#, .Net, and Microsoft Visual Studio, which makes it a complete cross-platform mobile app development framework.

Shared App Logic: It is a powerful feature of Xamarin, making it the most useful cross-platform development tool. Application Logic such as the input validation, web service calls, database interactions, and backend enterprise integrations are coded once in C# and can share 75% of the code across various operating systems. It saves the time and effort of the developers.

Seamless user experience: Xamarin provides the advantage of Native UI, access to specific device features, and, most importantly, native performance. These features make the user experience amazing.

Disadvantages or Cons of Xamarin

The following are the disadvantages of Xamarin Technology:

Updates delay: If the new features or updates roll out in this technology, these changes are usually delayed until they are reflected in the Xamarin tools. This may cause issues with our app.

Heavy graphics: Xamarin is not suitable for building complex applications or mobile games with heavy graphics.

Platform-specific code: Sometimes, we might need to re-write some parts of the UI in our app in native code, such as Kotlin or Java for Android, and Swift or Objective-C for iOS.

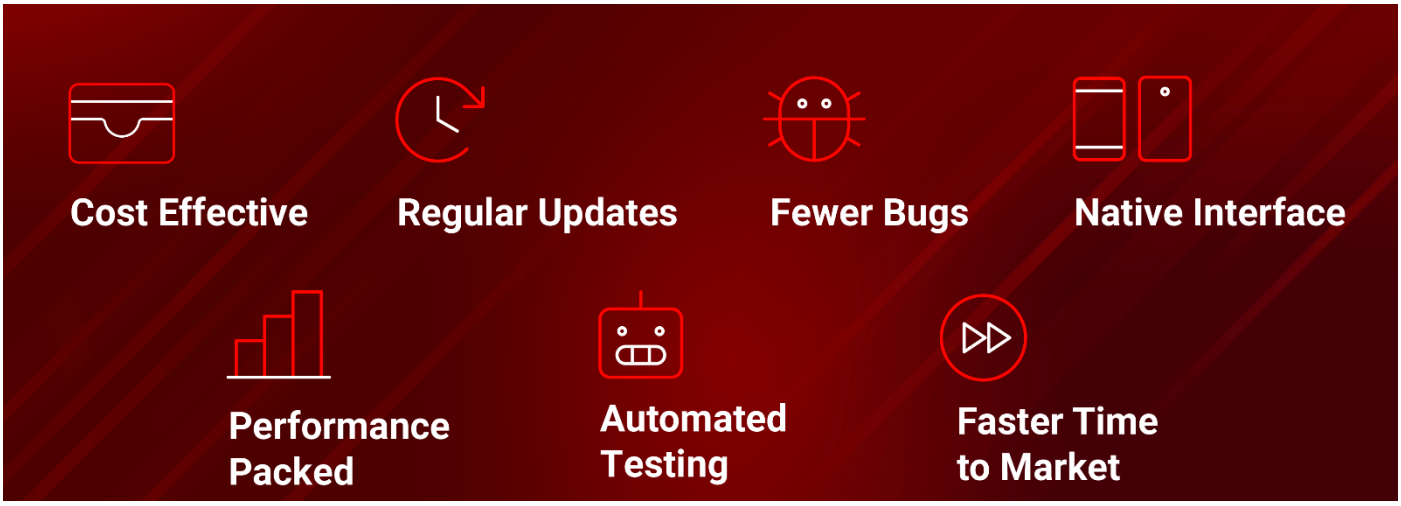
Difference between Flutter and Xamarin

The demand for mobile apps day by day increasing to a considerable extent. Due to this, the developers have searched a fast framework to build the app. Many developers have started using a cross-platform feature to make a fast app that gives many features to design e-commerce apps, interactive apps, and social apps.

In this section, we are going to compare the differences between Flutter and Xamarin, which helps us to identify which framework is best for our app idea. Before making a comparison, we will discuss in brief about these technologies.

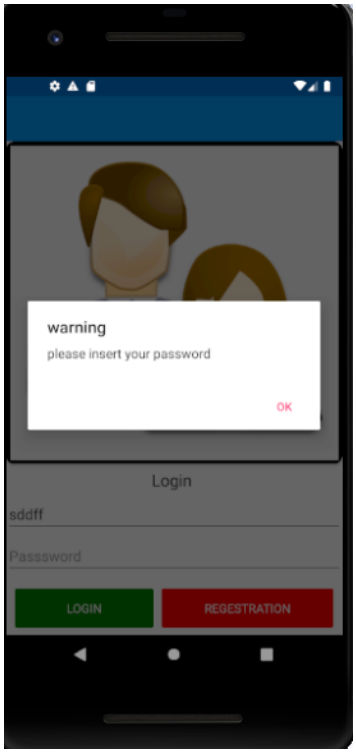
Why use Xamarin?

Xamarin builds native and cross-platform application to meet your business needs.

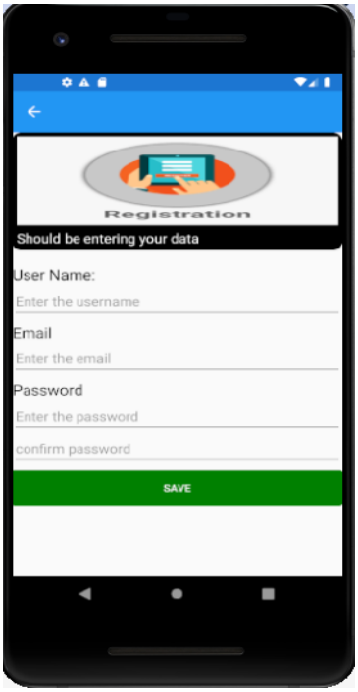


Mini project

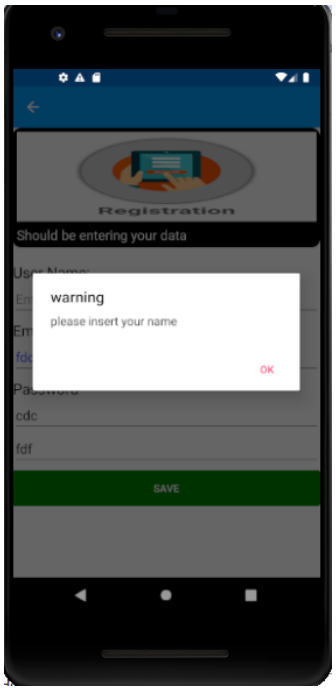
We have created a small application for my mobile. This is an app that works on Android and iOS platforms. When we open the app, it will show us the app login page. In this app, you can't do anything without logging in, unlike most apps. They have some services that work without logging in, but in our app it doesn't. On this page we used scroll-view, farm, image, label, input, login button, register button and on login page it asks you to enter Gmail and password you can login to the app and in case you forgot to enter password or account it will show you this message You are asked to enter it



And we have a button registration in case you do not have an account, you must create a new account for yourself. It will ask on this page to enter a user name, password, and configuration password.

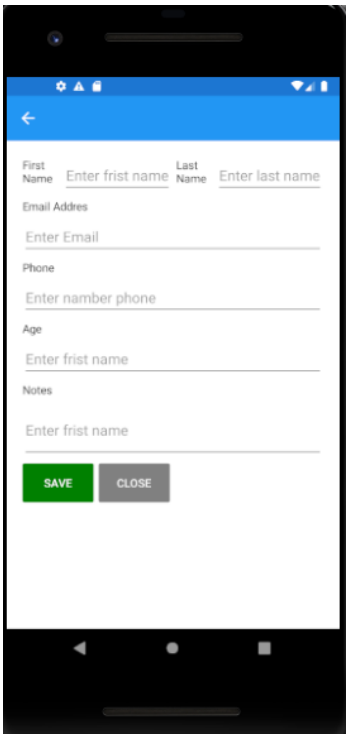


If you do not enter what is asked of you and press the save button, this message will display.

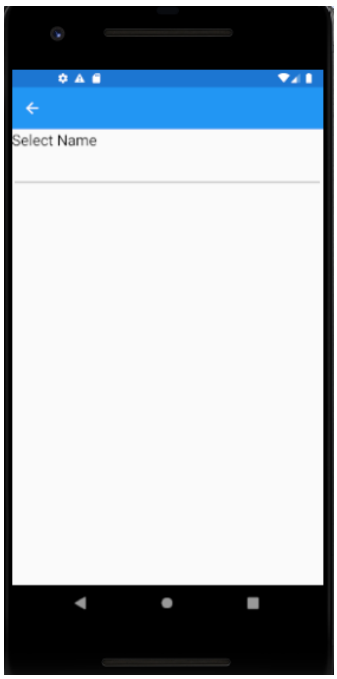


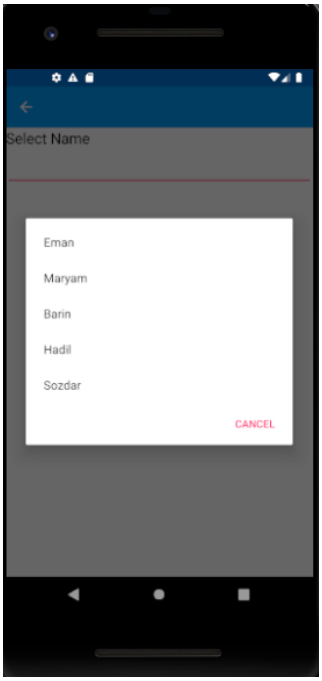
On this page we used button, label, scroll, after we created an account and logged in, it will show us the home page,

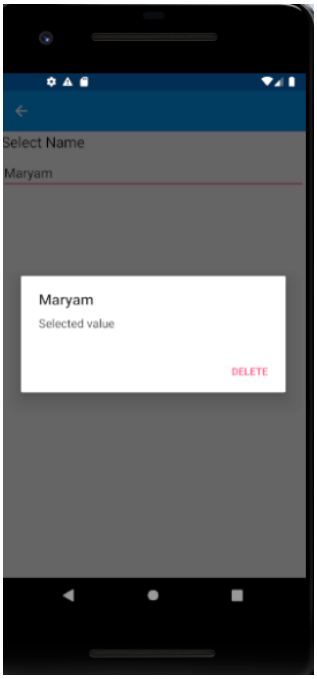
and on this page, we have three buttons, one for registering a new employee, one for deleting, and one for displaying an employee. Each of these buttons displays a new page when click on button operation. Pressing the add employee button will take you to the page to register information such as the first name. last name and phone and age and notes.



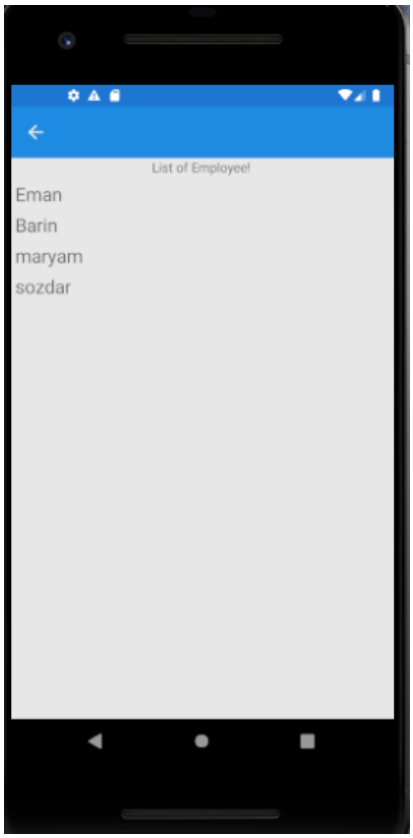
and we have the save button and the close button. And the delete button will take you to the delete employee page. You can choose a name to be delete and we used the poker properties.







Finally, employee view page will display employee names



Reference

https://www.ictdemy.com/csharp/xamarin/introduction-to-xamarin

https://www.kodehero.in/how-does-xamarin-work/

https://medium.com/globant/xamarin-forms-shell-navigation-d1179bfa2e4a

https://medium.com/@devathon_/flutter-vs-xamarin-in-2020-13fecdc8534a