University of Zakho
COMPUTER SIENCE
Stage 4
Group :B

MOBILE APPLECATION

Xamarin MiniProject

Chat App

Supervised by Mr. yousif

By:sadiya mikail haji

Academic year/
2021/2022

Contents

Introduction to Xamarin

Xamarin was released in 2011 by a software corporation. Recently, in 2016, Microsoft acquired Xamarin. It is very popular in Cross Platform App development and provides various development tools. This is a framework for creating a C# mobile cross-platform application. There are different structures for the development of the cross-platform app. We are using JavaScript and HTML. With this framework, we can use JS libraries to build applications such as websites for a mobile device. The website is packed in a container that gives the impression of a local application. It is discrete because it provides C# and runtime language that runs on three mobile platforms. We create a mobile application with a fully native features via Xamarin. There is C # codebase in Xamarin that has access to all the features of the native SDK.

1)What is Xamarin, How, and Where is it Used?

Platforms have become indispensable for application development. And one of the most popular platforms is Xamarin. So, what is it? Read on to learn more about this cross-platform app development tool.The demand for mobile app development continues to rise in the last few years. Companies have to speed up their software development process to keep up with the rising demand. This is where tools like frameworks and platforms come in. Software development platforms help developers build applications faster and with fewer errors.One of these tools is Xamarin. The mobile app development platform is one of the most loved by developers, according to Stack Overflow's 2020 survey. So, what is it, and why do developers love using it?

2)What is Xamarin?

Xamarin is an *open-source platform* for building modern and performant applications for *iOS*, *Android*, and *Windows* with .NET. Xamarin is different because it offers a single language C# and runtime, which works on three mobile platforms (Android, iOS, and Windows). Through Xamarin, we develop a mobile app whose look and feel is entirely native. In Xamarin, we write one C# codebase which has access to all the features of native SDK.

Figure 1. Xamarin — Single Code Base for iOS, Android and Windows Applications

Xamarin applications can be written on *PC* or *Mac* and compile into native application packages, such as an .apk file on Android, or an .ipa file on iOS.

C# and XAML

- Xamarin is a platform to create multi-OS apps
- It allows you to create a single application, that can work across iOS, Android and Windows phone.
- In order to develop these applications, you use C# and XAML.
- XAML acts as the markup and data binding language for the phone app, and C# acts as the server side language.

3) How Xamarin Works

Commercial products are available from Xamarin.IOS: Xamarin.Android.

The two are developed on Mono, the open source.net platform, based on the published ECMA standard. MONO runs on all possible platforms, including Linux, Unix, FreeBSD, and mac OS, as the .NET system itself.

It completely converted the iOS SDK and Android to C# so that the developer could become acquainted with it.

Without the complexity of constantly learning syntactic languages, you can simply use the similar Code Base for both the framework.

And the UI is still the same. For the application, it has to be developed independently, then the shared codebase must be reduced.

Below are both ways the user interface can be developed. First, using the original native approach to construct the user interface. The Xamarin. Forms are a further one. These types can be used to render the user interface for different platforms simultaneously and share the code 100% if you want Native UI technology.

When the entire UI is done, the most chief move is the relation between the UI and the codebase.

Xamarin Promises:

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

4)Xamarin-Forms Architecture

Xamarin allows you to create native UI on each platform and write business logic in C# that is shared across platforms. In most cases, 80% of application code is sharable using Xamarin.

Xamarin takes care of translating or compiling all your C# code to its corresponding platform-specific code

Xamarin is built on top of **Mono**, an open-source version of the .NET Framework based on the .NET ECMA standards.

Mono has existed for almost as long as the .NET Framework itself, and runs on most platforms including Linux, Unix, FreeBSD, and macOS. The Mono execution environment automatically handles tasks such as memory allocation, garbage collection and interoperability with underlying platforms.

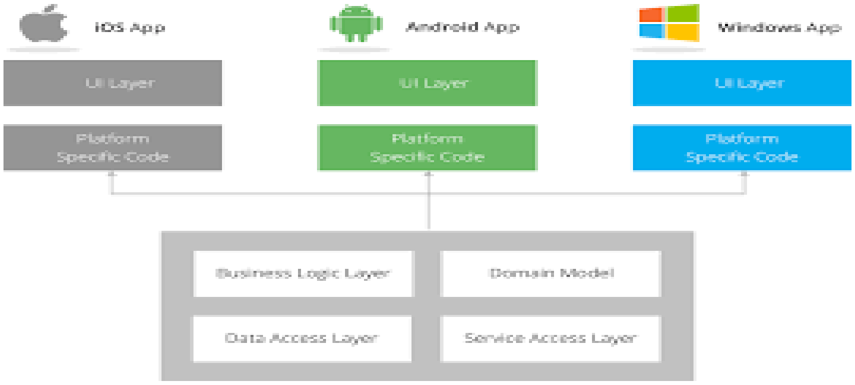


Figure 2. Xamarin Forms Architecture

Xamarin.Android Architecture

Android callable wrappers are a JNI bridge which are used any time the Android runtime needs to invoke managed code. Android callable wrappers are how virtual methods can be overridden and Java interfaces can be implemented. See the Java Integration Overview doc for more Both runtime environments run on top of the Linux kernel and expose APIs to the code that allows access to the underlying system.

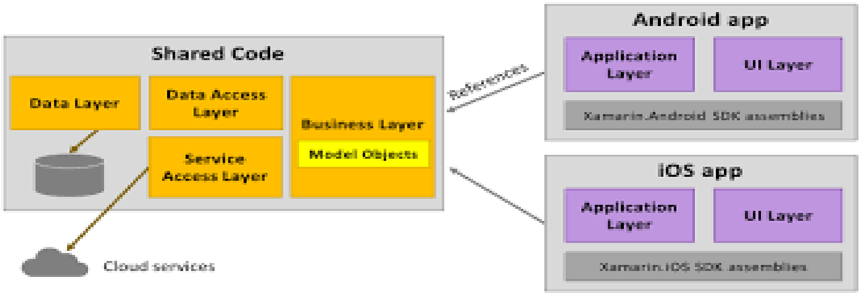
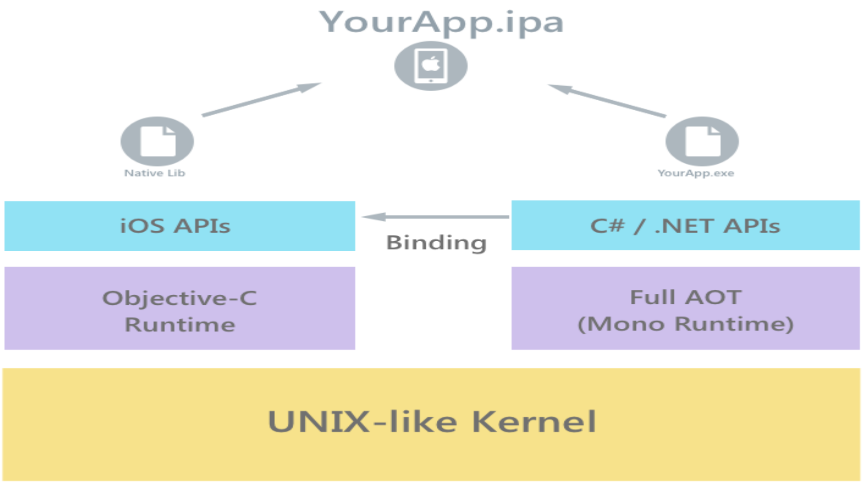


Figure 3. Diagram of Mono and ART above the kernel and below .NET/Java + bindings

Xamarin.iOS Architecture

Xamarin.iOS applications run within the Mono execution environment, and use full Ahead of Time (AOT) compilation to compile C# code to ARM assembly language. This runs side-by-side with the Objective-C Runtime. Both runtime environments run on top of a UNIX-like kernel, specifically XNU, and expose various APIs to the user code allowing developers to access the underlying native or managed system.



5) System Requirements

MacOS Requirements

- *Operating System* — macOS Mojave (10.14)
- *Development Environment* — Visual Studio for Mac
- *Xamarin.iOS* — Yes and iOS 12 SDK recommended
- *Xamarin.Android* — Yes and Android 6.0 / API level 23 recommended
- *Xamarin.Forms* — Xamarin.Forms are used to create cross platform applications like on Android, iOS or Windows mobile. Therefore, we are going to learn how to create the Xamarin.Forms solution from scratch. Xamarin.Forms solutions consist of different projects such as Portable, Android, iOS and Windows. Here, we will put our business logic inside the portable code and some platform specific code inside the Android/iOS/Windows project.
- *Xamarin.Mac* Xamarin.Mac exposes the complete macOS SDK for .NET developers to build native Mac applications using C#.

- Windows Requirements:
- *Development Environment* — Visual Studio
- *Xamarin.iOS* — Yes (with Mac)
- *Xamarin.Android — Yes*
- *Xamarin.Forms* — Android, Windows/UWP (iOS with Mac computer)
- *Xamarin.Mac* — Open project & compile only

6) Xamarin Comparison with other Cross platforms(Xamarin vs Flutter vs React NativeApp)

Speed:

App speed is always a concern for the app owners so it has to be quick and responsive. Flutter offers faster app development with faster app speed at the output. Other platforms like React Native and Xamarin also performs better in the app speed but whilst comparing with the Flutter they are slightly slower. The app speed also depends on the development process, functions, features, and other terms so it’s hard to compare on the instant but we can conclude the Flutter as a faster cross-platform app development framework.

Community Support:

Community support is built to give support and point out any issues related to the framework. While talking about community support, Xamarin lost in the competition because it has limited community support. On the other hand, React Native and Flutter have a better community for solving any issues and make it better for the users. According to the GitHub, React Native has more stars and Followers compare to the Flutter. So we can say React Native is the clear winner in terms of community supports.

Security:

Security of the source code and application is a must thing that's why security is listed in the priority action for all cross-platform developers. As mentioned earlier, React Native support third-party plugins and not have robust security supports. React Native lose the race of security, so let's talk about other cross platforms. Flutter and Xamarin based apps are robust and compatible with security access.

Customization:

As we know the React Native allows third-party plugins that simply means customization with the React Native is easy and compatible with Flux. Flutter's Hot Reload feature and Xamarin Live Reload feature empower both platform for live changes in the on-going development. Flutter and Xamarin have pre-install layout elements that simply indicates the freedom of customization with both cross-platform app development.

Usability By Developers:

The Xamarin is based on the C# that is simple and easy to learn the language. Flutter works on the Dart, an object oriented programming which is also easy to learn for newbie developers. React Native coded in Java that is most widely used in any development so working on Java is a piece of cake for every developer. In the concise, Xamarin and React Native are the easiest to access cross-platform app development framework.

Popularity:

According to Xamarin, the cross-platform tool is used by over 1.4 million developers worldwide. Moreover, Xamarin products are used by over 15,000 companies from over 120 countries in different industries such as media, transportation, finance, healthcare, and gaming. Due to its ability to write native UI code for app development, many popular brands use the Xamarin for crafting their application. Xamarin develops native codes but it is not free, so it is considered for the premium users. As most of the developers use Xamarin but still many developers rely on the Flutter and React Native.

Popular Apps that used React Native, Xamarin and Flutter

React Native Platform

- Walmart
- SoundCloud
- Bloomberg
- FaceBook
- Instagram

Xamarin Platform

- Storyo

- SuperGiant Games
- The World Bank
- APX
- Skulls of the Shogun

Flutter Platform

- Alibaba
- Hamilton Musical
- Google AdWords
- AppTree
- Google Greentea

7)Why Choose Xamarin for App Development over React Native & Flutter?

Xamarin is equipped with robust emulators that suit various mobile platforms, and this is an important reason why many business organizations are relying on this cross-platform app development framework. Xamarin provides Several options for debugging with the freedom to crosscheck from the desktop, emulator, or directly on the device. Here are some of the noteworthy features of this framework that developers are enjoying.

- Performance: The performance level of apps is highly competitive compared to hybrid or any other cross-platform development tools.

Image loading is 14% faster when compared to other platforms, and image-saving speed is super-fast as well.

- Development Speed: Xamarin comes with a library of templates that permits to use of standard interface elements. Developers enjoy when the inception to development speed is faster with Xamarin.

- Sharable Code: Developers are equipped with the feature to write C# code that can run on cross-platforms. PCL (Portable Class Libraries), Shared Projects, .Net Standard Libraries facilitate code sharing. Adding more value to this feature, the Xamarin.forms framework makes it possible to share the same code on several other platforms.

- Native UX: It is compatible with different operating systems and enables developers to achieve native look whether they develop the app for Android or iOS.

- Open Source: Xamarin is an open-source free tool that comes with Microsoft’s development environment Visual Studio. It gives you the freedom to use built-in tools for manual customization along with an option to reuse the codes.

- Resourceful: Developers can go in-depth if they want to understand how exactly the platform works. Xamarin is owned by Microsoft and the platform is focused on simplifying the cross-platform app development process.

Xamarin comes with a compilation of tools like Xamarin.mac, Xamarin.insights and Xamarin.testcloud.

It gives you full freedom to create Mac Apps, analyze apps through Insights and run

Features

What is Xamarin's strong point? It allows for sharing 90 percent of your code to major platforms. Aside from that, the cross-development platform also offers features that many developers love, such as:

1. Software Development Kits (SDK) Binding

Xamarin has the bindings for all the platform SDKs for Android and iOS. Furthermore, these bindings are easy to use and navigate. They also provide robust compile-time type checking. In short, these bindings can help in developing more error-free and higher-quality applications.

2. Wide Arrays of Third-Party Codes

The platform provides you with facilities to apply Java, Objective-C, and C++ libraries directly. Thus, allowing you to use wide arrays of third-party codes. In addition, Xamarin has project binding capabilities that let you tie Java libraries and native Objective-C by using declarative syntax.

3. Use of Modern Language Constructs

All Xamarin applications are developed in the C# programming language. It is a modern language that features more dynamic functional constructs like parallel programming, lambdas, LINQ, and more.

4. Cross-Platform Support for Mobile App Development

It offers cross-platform support for Android, Windows, and iOS. This means you can share 90% of your codes to any or all of the three platforms. Aside from this, using the tool will also allow you to access common resources through a unified API across all platforms. Thus, you can significantly reduce the development time and cost.

Who Is It For?

With these features, the open-source platform is perfect for developers who are:

- Planning to write, share, and test codes and business logic across different platforms.
- Developing cross-platform applications using C# and Visual Studio.

Provides Native Experience

Most cross-platform development tools strip their platforms of their uniqueness. Xamarin, on the other hand, highlights the platforms' strengths. It does this by converting Objective-C, Java, and .NET to C#.

Developers would then be able to build software that fits all operating systems using the shared codebase and libraries. As a result, the software provides the complete user native experience.

Covers Most Mobile Operating System

Other tools allow development for Android and iOS systems; the instrument takes it a step further. It covers operating systems of Windows Phone and Blackberry.

Reduces Cost

Like all cross-platform development tools, it eliminates the need to develop separate apps for different operating systems. So, there is no need to employ additional developers to create apps for other operating systems.

In addition, you can reduce the maintenance costs as a single team can do the troubleshooting after deployment.

Quicker Development to Market-Time and Lower Maintenance Downtime

Because the tool is cross-platform, you only need to write the code once. Then, you can share business logic across all mobile operating systems. Thus, hastening the software development process, which allows you to release your product quickly.

Moreover, downtime due to maintenance and troubleshooting will be shorter. Your team will only need to check and fix issues in one operating system and share it with the rest.

It’s an Open-Source Platform

The cross-platform mobile app development tool is part of Microsoft’s open-source .NET platform. This means that it is free and has strong community support. Over 3,700 companies are contributing to enhancing and forward the development platform.

As stated above, Xamarin offers developers and companies alike many benefits. Its features help in reducing both the software development time and cost. So, where can you find the best Xamarin developers to help you with your mobile app development projects?

Where to Find Xamarin Developers?

The first step in finding the best Xamarin developers is to detail their responsibilities first. These developers play critical roles in the design and development of cross-platform mobile services and solutions.

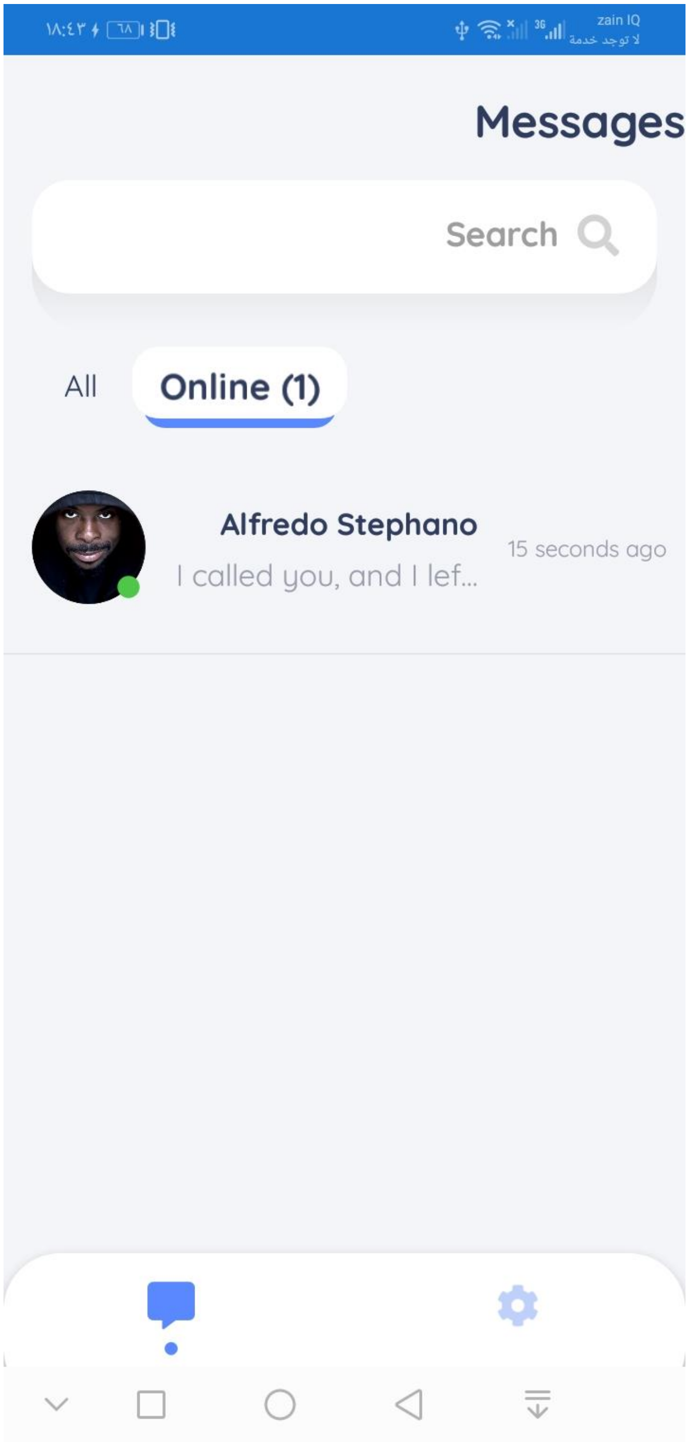
Therefore, they must have experience in mobile development for iOS, Android, and Windows. Plus, they must also have excellent communication skills to work with your team seamlessly. However, with the rising need for IT professionals in the US, you will encounter numerous competitions in recruiting the best Xamarin developers.

Luckily, another option available for you is to hire offshore software developers. And, that is where we at Full Scale come in. All of our Xamarin developers have undergone a rigorous recruitment and onboarding process.

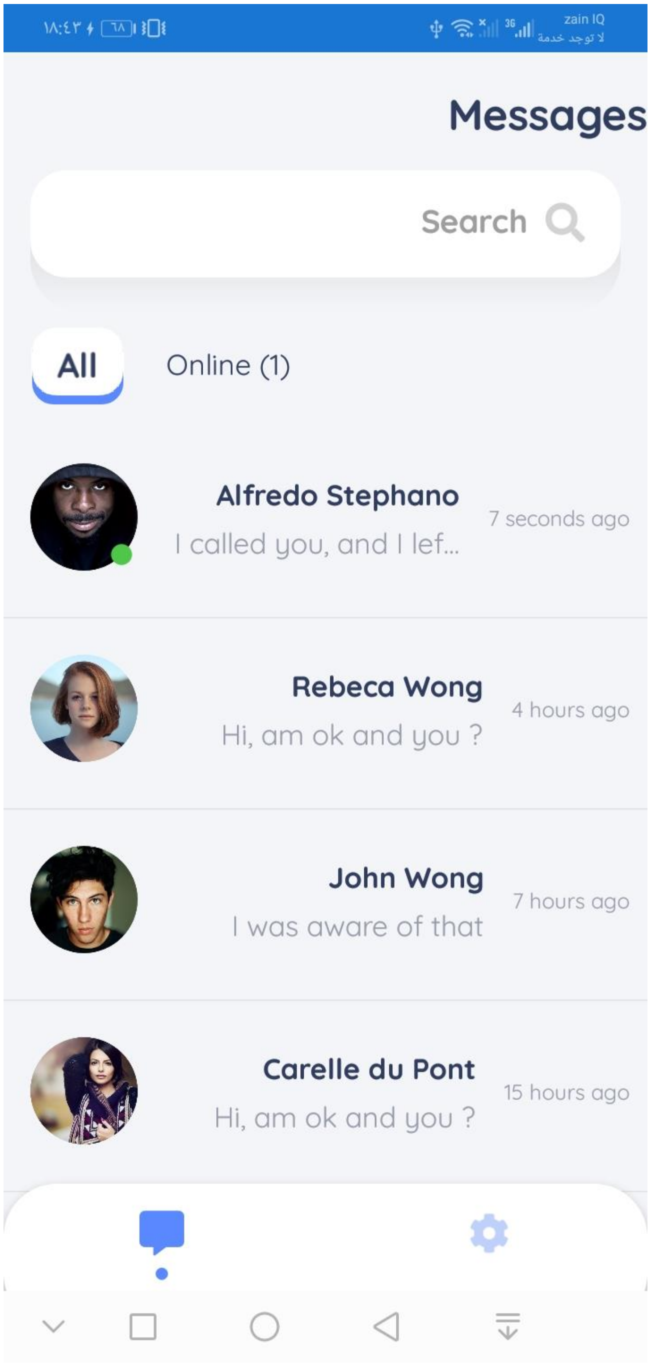
Aside from that, these developers and other experts are continuously training to be up-to-date with the latest technologies and techniques. Hence, our experienced and professional developers, programmers, testers, and other specialists are top of their fields.

With us, you can build your dream software development team to fulfill your mobile application projects

Project Picture



Alfredo Stephano
I called you, and I lef...
15 seconds ago

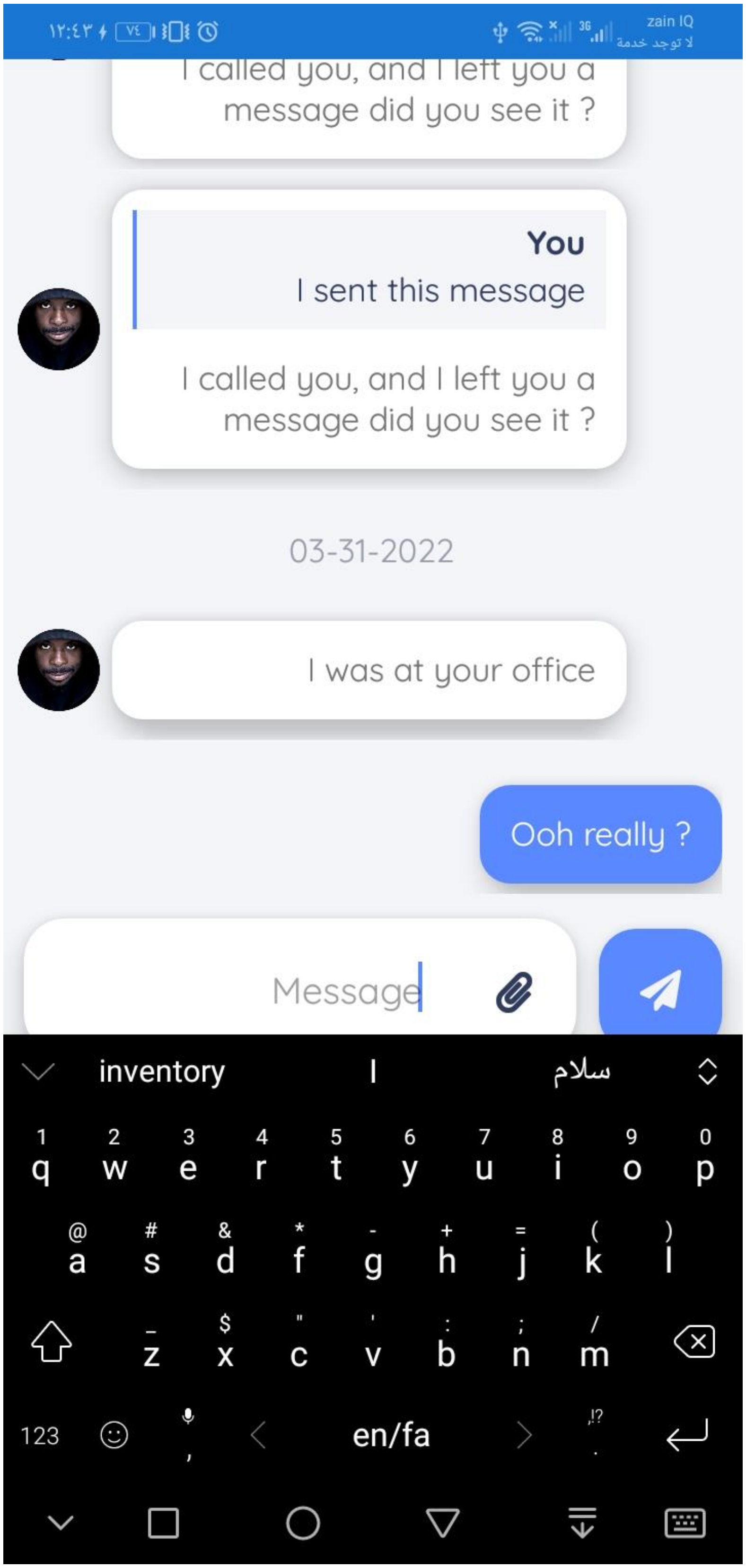


Alfredo Stephano
I called you, and I lef...
7 seconds ago

Rebeca Wong
Hi, am ok and you ?
4 hours ago

John Wong
I was aware of that
7 hours ago

Carelle du Pont
Hi, am ok and you ?
15 hours ago



I called you, and I left you a message did you see it ?

You
I sent this message

I called you, and I left you a message did you see it ?

03-31-2022

I was at your office

Ooh really ?

7.CONCLUSION

In this mini project I made about chat app that mean sending a message from one to another such as whatsapp and viber, and showing the name off all person that you want send them messages, ,for messages ,you will send a message to the corresponding person,and he will reply to you with a random message stored in the database.

4. REFERENCES

Textbooks:-

i. ^ "Xamarin delivers tool for building native Mac OS X apps with C#". December 13, 2012. Archived from the original on April 7, 2014. Retrieved April 1, 2014.

ii. ^ "Xamarin for Android". Archived from the original on April 23, 2014. Retrieved April 1, 2014.

iii. ^ "Xamarin for iOS". Archived from the original on March 30, 2014. Retrieved April 1, 2014.

iv. ^ Peter Bright (February 20, 2013). "Xamarin 2.0 reviewed: iOS development comes to Visual Studio". Archived from the original on April 14, 2014. Retrieved April 1, 2014.

v. ^ Mikael Ricknäs (June 25, 2013). "Xamarin tool aims to show the ease with which .NET apps can become mobile". Archived from the original on April 7, 2014. Retrieved April 1, 2014.

vi. ^ "Announcing Xamarin 3".

vii. ^ "Windows Platform Features - Xamarin". docs.microsoft.com.

viii. ^ Krill, Paul (January 14, 2020). "Microsoft enables native mobile development with Blazor". InfoWorld. Retrieved February 6, 2020.

ix. ^ Billson, Alex (July 15, 2018). "Cross Platform Mobile Apps with .NET and Uno". Hacker Noon. Retrieved January 20, 2019.

x. ^ "Xamarin Updates From Microsoft Build 2020". Xamarin Blog. May 19, 2020. Retrieved May 28, 2020.

xi. ^ "Introducing .NET Multi-platform App UI". .NET Blog. May 19, 2020. Retrieved June 4, 2021.

xii. ^ "dotnet/maui". GitHub. Retrieved May 28, 2020.

xiii. ^ "Xamarin Test Cloud". Archived from the original on April 7, 2014. Retrieved April 1, 2014.