University of Zakho

Department of Science

Computer Science

Academic Year 2021 - 2022


# Mobile Application

Prepared By:

1.Renas Hameed.

2.Zindan Hameed.

3.Baroj Real.

4.Kavin Nafe.

Supervised By:

MR.Yousif Garabet Arshak.

# Table of Contents

## Introduction

In this report, we will talk about the find jop application. Our purpose for making this application is to search for a job more easily or to share your own job and let other people see it. If they need it, we made this application to reach you.

To make this application, we have to choose a programming language that is good and performs well, so we chose xamarin.form open-source UI framework. because xamarin.form is a Cross-platform, so we can make the application at once and do it on multiple platforms, so we will do less work and save less time. xamarin.form is a high-performance framework made and supported by Microsoft

We used Firebase Authentication for login and Sign Up in this application, Firebase Authentication is a free platform developed by Google to create mobile and web applications.

and we have used Sqlite to store data of this app

SQLite is the most used and preferred database software for web and mobile developers. This is because it is an open and free source, and it provides quite simple and server less set up.

## What is Xamarin.Form

Xamarin.Forms is a feature of Xamarin. Xamarin is the popular mobile development framework which extends the .Net Development Platform with tools and libraries for building mobile apps. Xamarin.Forms is an open-source, cross-platform framework acquired by Microsoft for building Android, iOS, and windows app with .NET from a single shared codebase. We use Xamarin. Forms built-in Pages, layouts, and controls to build and design mobile apps from a single API that is highly extensible. Subclass any control to customize the behavior or to define our controls, layouts, pages, and cells to make our apps pixel perfect [1].


## Who Xamarin.Forms is for

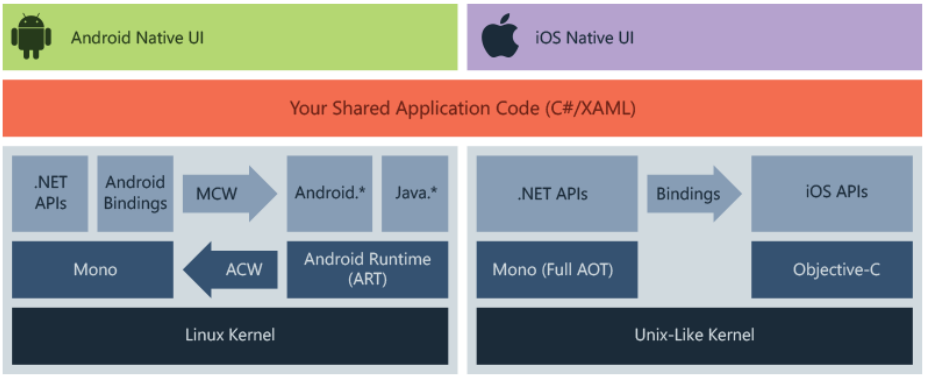
Xamarin.Forms is for developers with the following goals:

Share UI layout and design across platforms.

Share code, test and business logic across platforms.

Write cross-platform apps in C# with Visual Studio [4].

## How Xamarin works




The diagram shows the overall architecture of a cross-platform Xamarin application. 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.

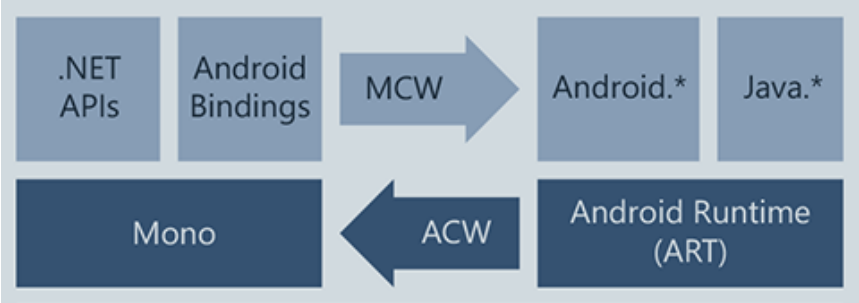
Xamarin is built on top of .NET, which automatically handles tasks such as memory allocation, garbage collection and interoperability with underlying platforms [4].

## Added features

Xamarin combines the abilities of native platforms, while adding features that include:
Complete binding for the underlying SDKs – Xamarin contains bindings for nearly the entire underlying platform SDKs in both iOS and Android. Additionally, these bindings are strongly-typed, which means that they're easy to navigate and use, and provide robust compile-time type checking and during development. Strongly-typed bindings lead to fewer runtime errors and higher-quality applications.
Objective-C, Java, C, and C++ Interop – Xamarin provides facilities for directly invoking Objective-C, Java, C, and C++ libraries, giving you the power to use a wide array of third party code. This functionality lets you use existing iOS and Android libraries written in Objective-C, Java, or C/C++. Additionally, Xamarin offers binding projects that allow you to bind native Objective-C and Java libraries using a declarative syntax.
Modern language constructs – Xamarin applications are written in C#, a modern language that includes significant improvements over Objective-C and Java such as dynamic language features, functional constructs such as lambdas, LINQ, parallel programming, generics, and more.
Robust Base Class Library (BCL) – Xamarin applications use the .NET BCL, a large collection of classes that have comprehensive and streamlined features such as powerful XML, Database, Serialization, IO, String, and Networking support, and more. Existing C# code can be compiled for use in an app, which provides access to thousands of libraries that add functionality beyond the BCL.
Modern Integrated Development Environment (IDE) – Xamarin uses Visual Studio, a modern IDE that includes features such as code auto completion, a sophisticated project and solution

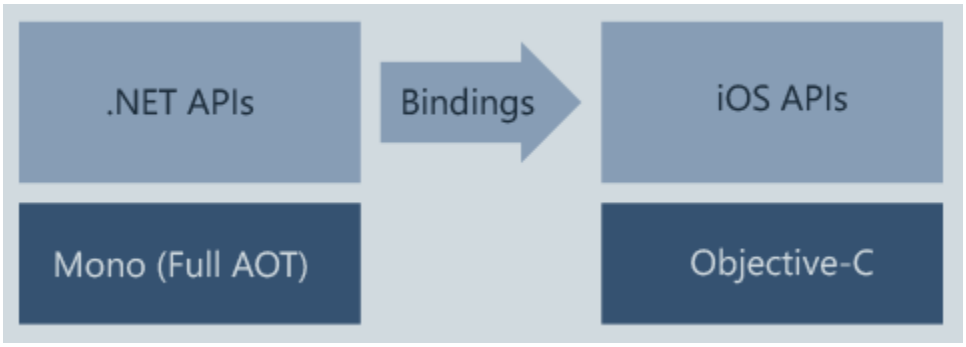
management system, a comprehensive project template library, integrated source control, and more.
Mobile cross-platform support – Xamarin offers sophisticated cross-platform support for the three major platforms of iOS, Android, and Windows. Applications can be written to share up to 90% of their code, and Xamarin.Essentials offers a unified API to access common resources across all three platforms. Shared code can significantly reduce both development costs and time to market for mobile developers [4].

## Xamarin.Android




Xamarin.Android applications compile from C# into Intermediate Language (IL) which is then Just-in-Time (JIT) compiled to a native assembly when the application launches.
Xamarin.Android applications run within the Mono execution environment, side by side with the Android Runtime (ART) virtual machine. Xamarin provides .NET bindings to the Android.*and Java.* namespaces. The Mono execution environment calls into these namespaces via Managed Callable Wrappers (MCW) and provides Android Callable Wrappers (ACW) to the ART, allowing both environments to invoke code in each other [4].

## Xamarin.iOS




Xamarin.iOS applications are fully Ahead-of-Time (AOT) compiled from C# into native ARM assembly code. Xamarin uses Selectors to expose Objective-C to managed C# and Registrars to expose managed C# code to Objective-C. Selectors and Registrars collectively are called "bindings" and allow Objective-C and C# to communicate [4].

## Xamarin.Essentials

Xamarin.Essentials is a library that provides cross-platform APIs for native device features. Like Xamarin itself, Xamarin.Essentials is an abstraction that simplifies the process of accessing native functionality. Some examples of functionality provided by Xamarin.Essentials include [4]:

- Device info
- File system
- Accelerometer
- Phone dialer
- Text-to-speech
- Screen lock

## Xamarin.Forms

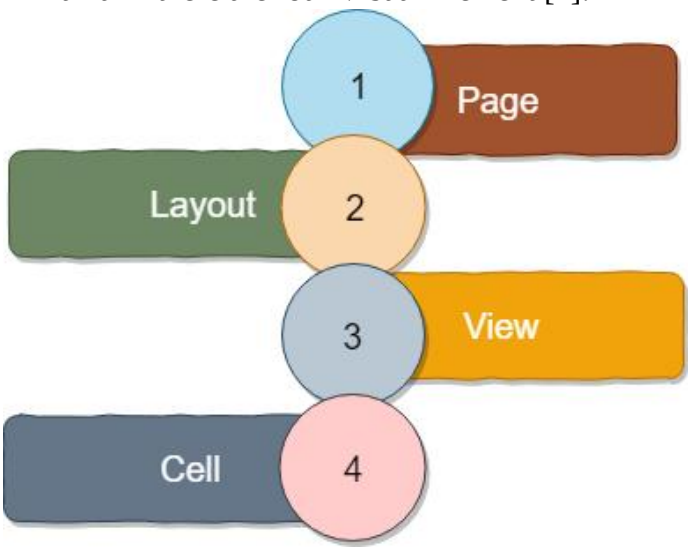
Xamarin.Forms is an open-source UI framework. Xamarin.Forms allows developers to build Xamarin.iOS, Xamarin.Android, and Windows applications from a single shared codebase. Xamarin.Forms allows developers to create user interfaces in XAML with code-behind in C#. These user interfaces are rendered as performant native controls on each platform. Some examples of features provided by Xamarin.Forms include [4]:
XAML user-interface language
Databinding
Gestures
Effects
Styling

## Xamarin.Forms Structure

In the beginning, when we open any device, any mobile phone whatever we see on the screen, the visible area is called a Page. All the visible area on the screen we consider it as a page, and we can compare that just like a ROM. And then how do we structure the things on the page, and how do we plan the things is known as Layout. The view is the actual item which we will put on the places, either we will stack them, put them on the left side, right side, etc [1].

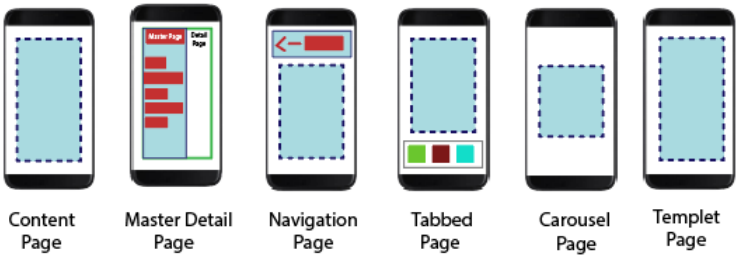
## Visual Elements

In Xamarin, elements shown on the device, the screen is called Visual Element. In a device like a Cell Phone, those are visible, or we can see are known as Visual Element.
In Xamarin there are four Visual Element [1]:



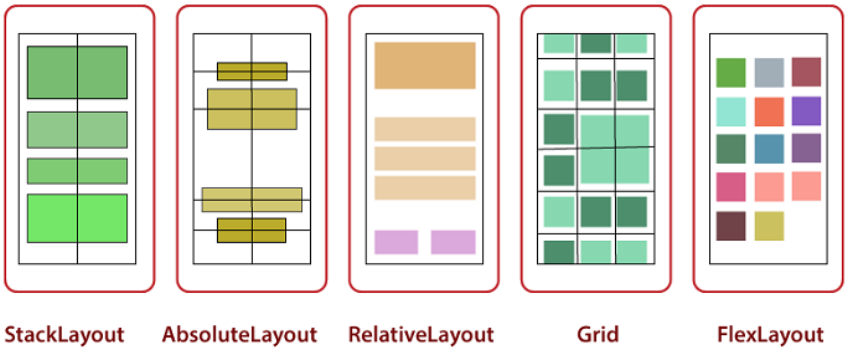
**1.Page**

In a device, from the navigation bar to the end of the screen known as a page.



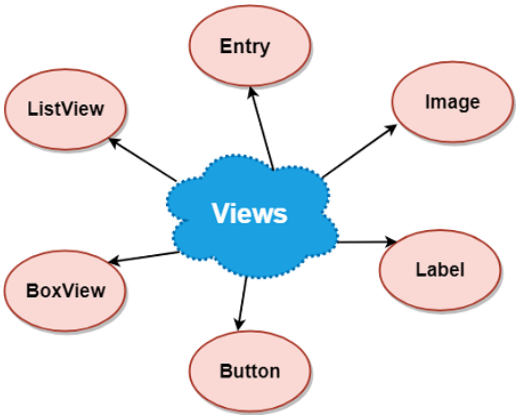

**2.Layout**

The child element in the page known as Layout



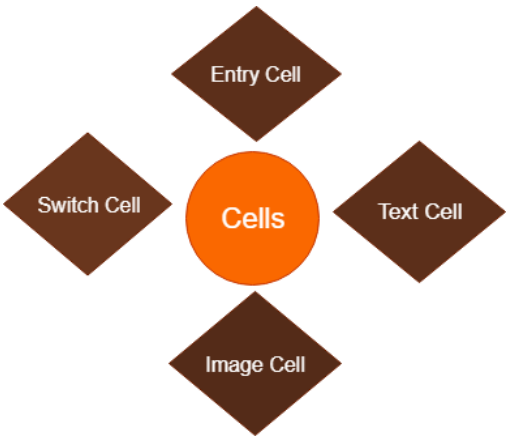

**3.View**

The layout contains lots of elements are known as Views.




**4.Cell**

The child element of Views is known as Cell.




## Xamarin.Form Local Database

SQLite as a local database in **xamarin**.Forms are perfectly fit for a simple reason as it is readily available on both Android and iOS. This all means that you can use this technology whenever you need to write a xamarin.Forms app. (newline) this is a zero-configured trading software or engine owned by SQLite. This means that is a complete mechanism available to web or application developers in which they can store their data structurally, as well as access free and opensource code. (newline) So, with SQLite, you will have the ability to put database features or functionality to your Xamarin.Forms app to retrieve and store any type of data easily.

One of the most basic needs is the ability to store and retrieve information. This can be implemented and set up using different tools on different platforms. this is another data storage and retrieval tool known for its unique features. These features include power, free and support for all operating systems such as Windows, Linux, Mac, Android and iOS. this data storage and retrieval tool is programmed in C language, and it is also a software that has a Public Domain license, which means that this tool is not owned by any organization or individual. It is not limited and everyone can use it without any restrictions [2].

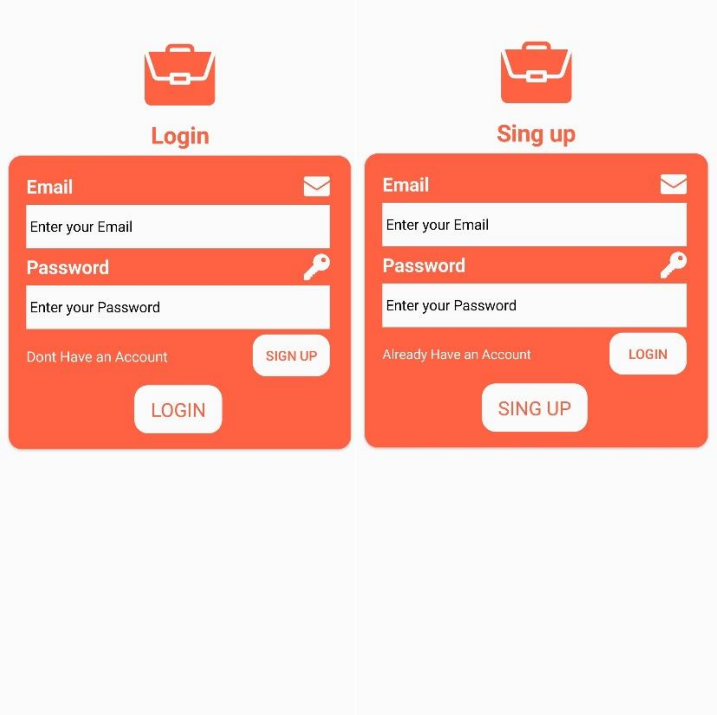
## Firebase Authentication

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices.

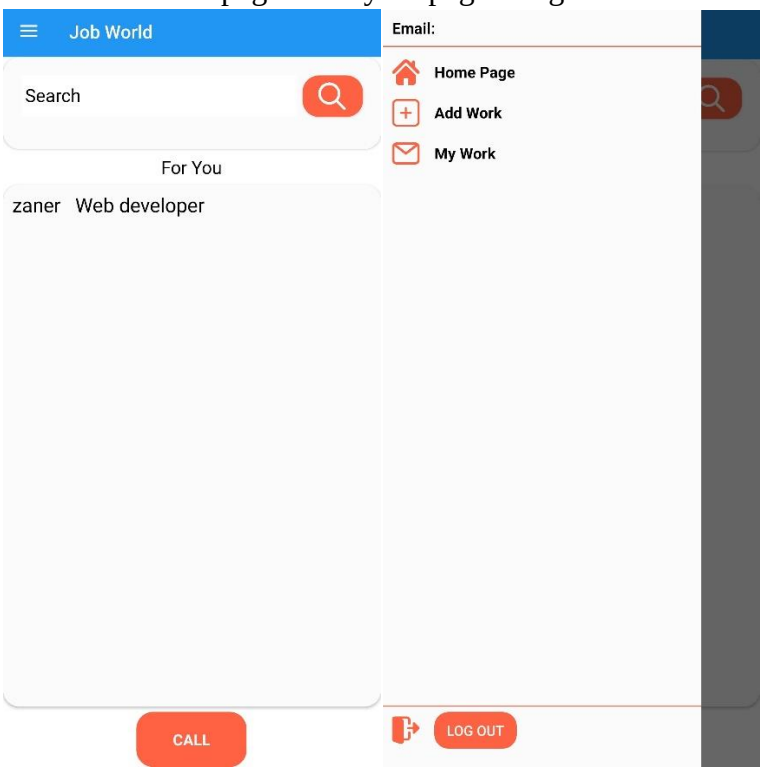
Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more [3].

## UI design

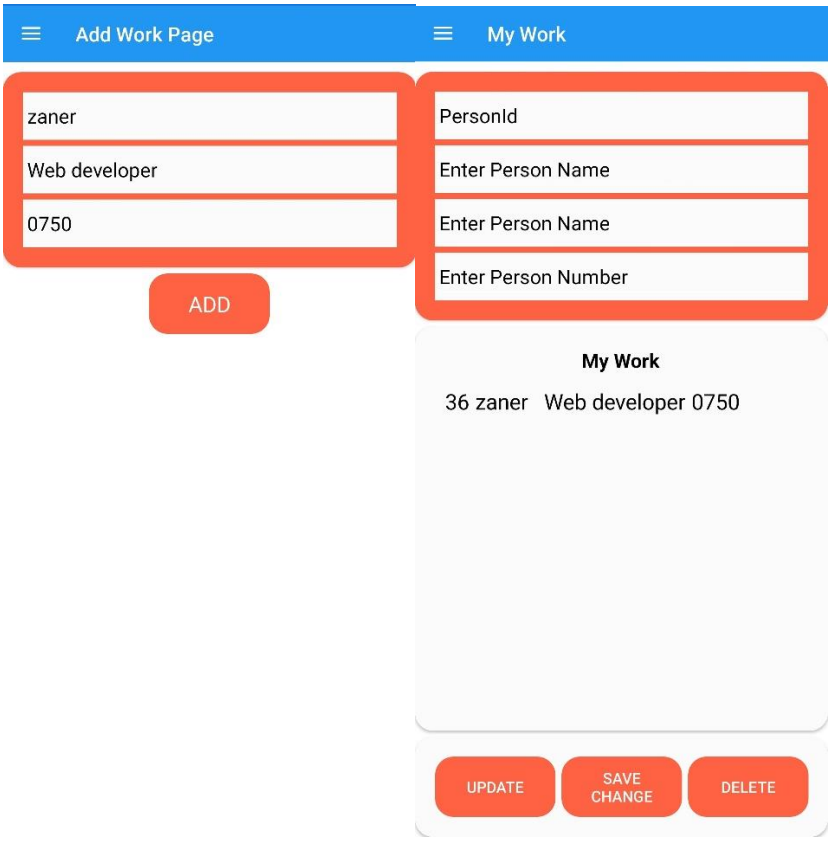
Login and Sign-in design of this application.




Here is the main page and flyout page design.



Here we has add post and we want change and delete post.




## Conclusion

In this report, we have mentioned what the application did the job and talked about how the customer has used how the client has used it, and we used the C# language Xamarin.form framework, and then Xamarin.form and then we used to mention the firebase and sqlite we used to do.

reference


1. https://www.javatpoint.com/xamarin-forms.
2. https://www.dotnek.com/Blog/Apps/sqlitenet-as-a-local-database-in-xamarinhow-i#:~:text=SQLite%20as%20a%20local%20database%20in%20xamarin.Forms%20are,zero-configured%20trading%20software%20or%20engine%20owned%20by%20SQLite.
3. https://firebase.google.com/docs/auth/.
4. https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin.