



Demo Banking App

Prepared by: Matthew Joughin

19 May 2017

SUMMARY

Objective

Create a demo banking app that runs on all available platforms supported by Xamarin.Forms

Goals

Prove that it can be done and run on iOS, Android, MacOS, and UWP (Universal Windows Platform), all sharing the same common code and UI's, so that features can be rapidly added to all platforms by a much smaller team of developers than used presently.

Solution

Used latest Xamarin.Forms beta package to create the project accordingly (so that we gained MacOS and UWP which are in beta) and execute on this goal

Project Outline

Created the solution and added functionality for basic starting use case for a banking app:

- Logging in
- Viewing and adding of Beneficiaries

A backend was not added, only a mock one, as it wasn't necessary to prove that one Xamarin.Forms app could run on all the target platforms.

Getting the code:

Clone from GitHub:

<https://github.com/mrwcjoughin/matthewjoughin-bankingapp-xamarin.git>

Running the code:

On Windows:

- You will need Visual Studio 2017 (any edition) or higher, with the mobile features ticked/installed - you can get this from Microsoft's website
- This will allow you to run the UWP natively
- This will allow you to run the Android project through the emulator or if you have a physical device attached
- This will allow you to run the iOS project if you have an iPhone connected and a Mac machine on the network to act as the build machine

On MacOS:

- You will need the latest version of XCode installed - you can get this from Apple's website or from the Apple MacOS App Store
- You will need the latest version of Xamarin or Visual Studio for Mac installed - you need to choose the Alpha update channel if you want the access to run the MacOS project - as it's currently in beta as of the date of this readme

On either platform, whichever IDE you are in, you need to set the startup project to which ever platform project you wish to run to run that platform

Testing the Demo:

Login with any username or password - as long as each is at least 5 characters

APPLICATION FLOW AND NAVIGATION

By using Xamarin.Forms the application gets the benefit of the inherit flow and navigation of the target platform.

While conceptually the flow is as follows:

- Login page
- Once logged in the user is presented with a dashboard, on which there are 2 options:
 - Beneficiaries
 - This shows existing Beneficiaries
 - It also allows adding of Beneficiaries - which presents the user with a screen to fill out the Beneficiary details and save it
 - Logout
 - This allows the user to logout
 - It asks for conformation

However way the user sees it will differ as you can see from the Platforms section, as Xamarin.Forms takes advantage of the native platforms controls and positioning of navigation controls in particular to make the user feel at home in their chosen platform.

COMPARISON TO PTTRNS APPLICATION

PayPal (<https://pttrns.com/search?q=paypal>) was my inspiration to do a banking app.

However, instead of cloning PayPal's app from start to finish, I wanted to demonstrate that an app could be written for all current consumer platforms with one shared code base - both for code behind the view and the view it's self.

This is something that is seldom done in the real world, and I wanted to show that Xamarin.Forms has come so far, that it is now actually possible for a production application.

The biggest challenge now that the UI is taken care of by Xamarin.Forms, and the behind the view code is handled in the PCL project, is the nuget packages. Namely: finding packages for the inherit needs of the app that support all the platforms. This proved a **big** challenge, but through perseverance I managed to get everything I needed.

The acid test I used was to do AES Encryption. That is critical for a banking app, and I wanted to prove that it could be done in a Xamarin.Forms app without any compromise. What I found was that strangely enough Microsoft provided the best nuget package for this, but that this package supported all the required platforms except their own - UWP! Why this is I have no idea - I have even tweeted the head of the .net platform and Xamarin to find out - but such as it is it doesn't. So I found a nuget package that did support UWP, which is PCLCrypto, but found it didn't support MacOS yet. So I created an interface in the PCL project, and I created the classes that implemented it in the platform specific projects, and for all except UWP I used Microsoft's package, and for UWP I used PCLCrypto. Each platform registers it on startup up using the DependencyService class provided by Xamarin.Forms and hey presto we have working encryption for all platforms without the PCL project having to worry about how its done on each platform.

PLATFORMS

iOS

Login:

Carrier

5:05 PM

Login



Email Address

Password

Login

Beneficiaries:

Carrier

5:12 PM

Beneficiaries

Add

John Doe
FNB, Cheque, 34523453

Jane Doe
FNB, Savings, 45346343

Add Beneficiary:

Carrier 5:12 PM

< Beneficiaries New Beneficiary Save

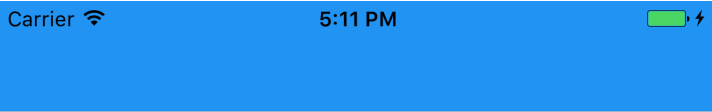
Name

Bank

Account Type

Account No

Logout:



Are you sure you want to Logout ?

[Yes](#)

Android

Login:

11:19

Login

SuperBank

the best bank. bar none.

Email Address

Password

Login

Beneficiaries:

11:20

Beneficiaries

ADD

BENEFICIARIES

LOGOUT

John Doe

FNB, Cheque, 34523453

Jane Doe

FNB, Savings, 45346343

Add Beneficiary:

11:21

←

New Beneficiary

SAVE

BENEFICIARIES

LOGOUT

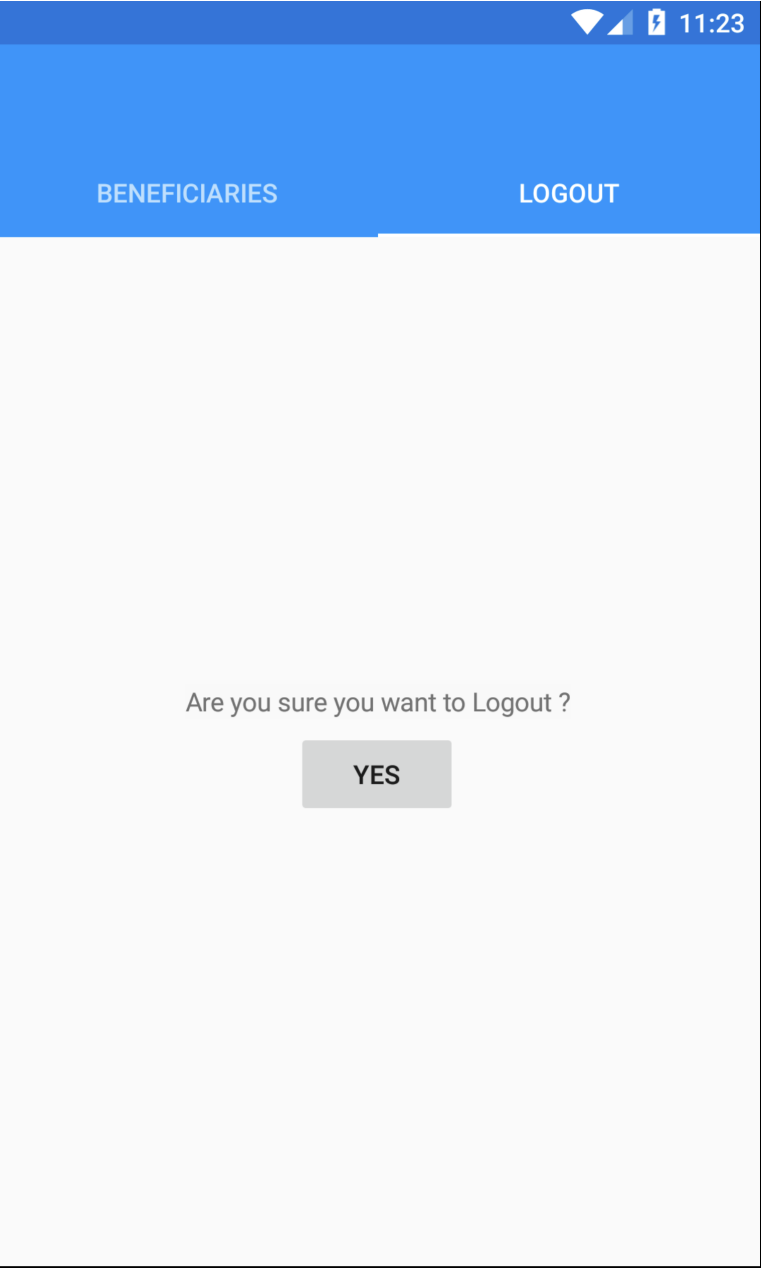
Name

Bank

Account Type

Account No

Logout:



MacOS

Login:

Login



Email Address
Password

Login

Beneficiaries:

Beneficiaries

Add

Beneficiaries

Logout

John Doe

FNB, Cheque, 34523453

Jane Doe

FNB, Savings, 45346343

Add Beneficiary:

New Beneficiary

Save

BeneficiariesLogout

Name

Bank

Account Type

Account No

Logout:

BeneficiariesLogout

Are you sure you want to Logout ?

Yes

UWP (Universal Windows Platform)

Login:

UWP

SuperBank

the best bank. bar none.

Email Address

Password

Login

Beneficiaries:

UWP

Beneficiaries

BeneficiariesLogout

John Doe
FNB, Cheque, 34523453
Jane Doe
FNB, Savings, 45346343

Add Beneficiary:

UWP

New Beneficiary

Beneficiaries

Logout

Name

Bank

Account Type

Account No

Logout:

UWP

Beneficiaries

Logout

Are you sure you want to Logout ?

Yes