Career Finder Documentation

# Index:

1. [Summary](#_1._Summary:)
2. [Quick start, overview and links](#_2._Quick_start)
3. [Disclaimer](#_3._Disclaimer:)
4. [Under the hood, improvements](#_4._Under_the)
5. [Design notes](#_5._Design_Notes:)
6. [Troubleshooting](#_6._Troubleshooting:)

# 1. Summary:

This application uses the cross platform [Xamarin](https://developer.xamarin.com/guides/xamarin-forms/) framework (by Microsoft).  
Being cross-platform, it uses C# and XAML as a layer of abstraction over the native languages (some Java and Swift are produced during build).  
The greatest advantage is code reusability, no need to code twice in different languages.  
Of course this comes at the cost of many native functionalities being stripped from their simple use (for example the designer, yet see [Xamarin Live Player](https://developer.xamarin.com/guides/cross-platform/live/), maybe it is functional now).

The purpose of the app is to display Degrees, Jobs and People articles from the [careers with STEM](https://careerswithstem.com) website and resources.  
Jobs and Degrees (along with Uni rankings) are fetched online from CSV files, whereas People articles are directly fetched from the HTML on the website profiles pages.  
Then you can add these objects to your favourites and see them in the dedicated page.

So before making any changes to this app, I would recommend reading about Xamarin Forms, and especially having some knowledge about the .NET framework, to leverage its many packages and functionalities.

Also, I used the MVVM (ModelViewViewmodel) pattern, it is not always the case, but I did my best to respect best practices and refactor code when I could (mostly at the end, because I wasn’t skilled enough at first).  
If you don’t know what MVVM is, please see the [Guide](https://developer.xamarin.com/guides/xamarin-forms/).

I’ll often refer to the [Guide](https://developer.xamarin.com/guides/xamarin-forms/), without linking it every time.  
It’s the Xamarin website where I have found answers to many questions, and learnt how to use the framework. So, if ever in doubt, RTFM.  
Check the Xamarin Api too, and of course, Stack Overflow.

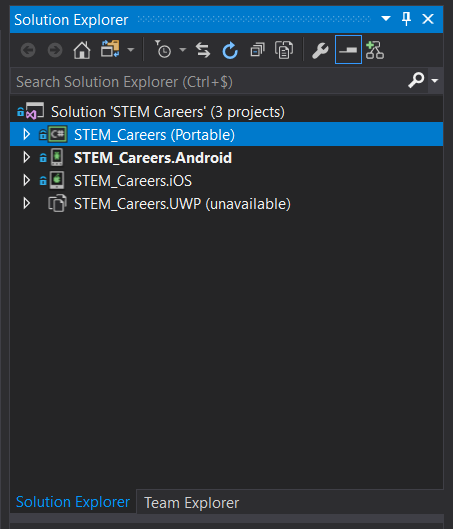
# 2. Quick start, overview and links:

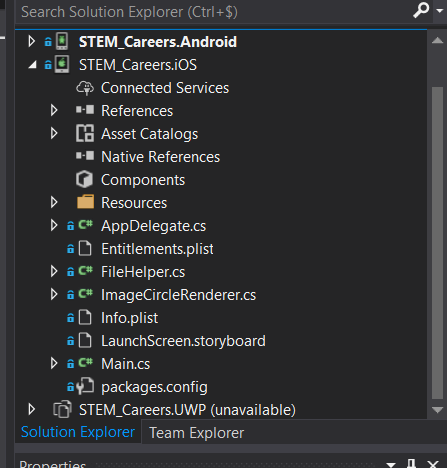
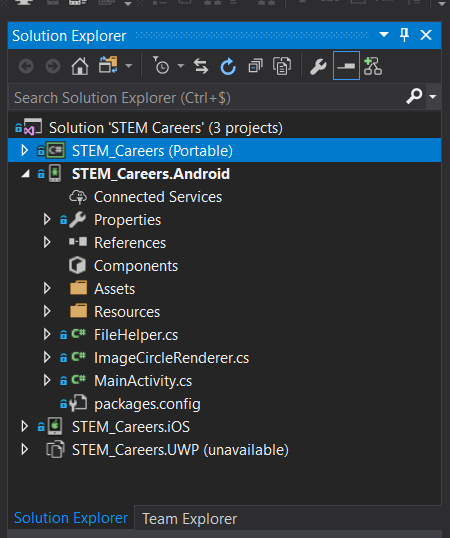
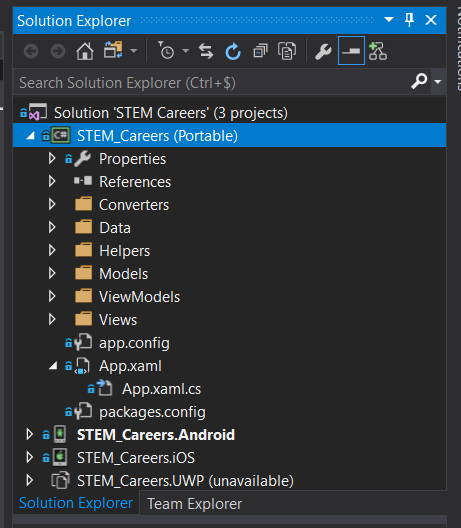
So, to get started, you’ll need a good machine with [Visual Studio](https://www.visualstudio.com/) installed with Xamarin onboard, the Xamarin.Forms version is 2.5.0.91635 and using a performant Mac might prove easier because Apple doesn’t let Windows emulate iDevices.  
I used a PC with VS 2017 and emulated on an iMac over ssh. See the Guide.

The project is located on the [refractionintern@gmail.com](mailto:refractionintern@gmail.com) GitHub at this [link](https://github.com/RefractionMedia/STEM-Careers).

The google drive folder associated with the app is: <https://drive.google.com/drive/folders/0B1IaMDQyEubcaWdpOTRRdEo5U1E>

If link has changed, it should be “0. DIGITAL ESSENTIALS”





The solution “STEM Careers” is composed of three (or four if you count UWP, which I disabled) projects with the root namespace of STEM\_Careers:

* (PCL) The Portable Class Library, which is where most of the coding is done, it uses the MVVM pattern (mostly) to handle data bindings between pages and the underlying data  
  When we want to implement native features (such as File access) we use an interface in the PCL, and implement it in each platform project.
* .Android and .IOS projects are for their respective platforms, it’s where we code natively if needed, put in resources (such as images) and configure Release and Debug for the platform

Each project has a Debug and Release configuration, Debug is straightforward, Release is a bit trickier:

* Android is built then archived Ad-Hoc with the signing key located in the google drive (be very careful not to lose it). Then new versions are published via [Google Play Console](https://play.google.com/apps/publish/) with the [refractionmedia.aus@gmail.com](mailto:refractionmedia.aus@gmail.com) account
* IOS is built on “Release”, ”iPhone” (not simulator) config, then the. ipa file must be retrieved from the /bin/Release/iPhone folder of the IOS project  
  IOS build requires a dev & prod licence available on the apple dev site, as [catchpoleheather@gmail.com](mailto:catchpoleheather@gmail.com)

Each Page is built following the MVVM pattern, so you have the Page.Xaml page which will structure the data with a ScrollView, ListView, StackLayout, Buttons, Labels etc… Then the code behind the page, in the Page.Xaml.cs file, where we usually have the “button\_clicked” event handler and so on.  
Finally, we have a ViewModel that we use to instantiate, load and bind data (like degrees, that are fetched from the database, and linked to corresponding universities as the page loads).

CSV links:

* [Jobs](https://docs.google.com/spreadsheets/d/1HYXlESJyWopYZNnQmEt33t91MRxTr1_YZwxRpz6hVyE/edit#gid=1539181322)
* [Job Descriptions](https://docs.google.com/spreadsheets/d/1HYXlESJyWopYZNnQmEt33t91MRxTr1_YZwxRpz6hVyE/edit#gid=1539181322)
* [Degrees](https://docs.google.com/spreadsheets/d/1ezKQ2L5QBhzvFPjC6g9Qzt0APV0aAtNrmgnlGD5U1zA/edit#gid=975969105)
* [University rankings](https://docs.google.com/spreadsheets/d/1gGDH0QdwM18-je5p-TOaj158ke-uCSgS5-C-CfYMYuQ/edit#gid=0)

# 3. Disclaimer:

This documentation is aimed as a quick start for any dev taking over the project, but you’ll have to get reading the Guide and comments in the code files.

Proguard is not enabled in the Android project because of conflicting jdk version that caused an error in my dev environment, I would recommend reactivating it once you’ve migrated to your machine and are able to run the project.

# 4. Under the hood, improvements:

Here is the app process, roughly:

* App.cs starts, checks if database is initialized  
  true 🡪 goes to Homescreen  
  false 🡪 goes to LoadingScreen, which makes the user wait while loading profiles and CSVs
* LoadingScreen calls the Database, which calls CsvCenter for the CSVs, each CSV is parsed and data stored asynchronously.
* Once all is loaded 🡪 goes to Homescreen
* Every page basically loads data from the sqlite database, except the PeoplePage which checks for updates, see the code for comments

Many improvements can be made:

1. Authentication and email subscription with user accounts: <https://developer.xamarin.com/guides/xamarin-forms/cloud-services/authentication/oauth/>
2. An updating system for the databases. And a background loading of data, instead of the start-up launch screen that loads everything at once. Which can fail, result in incomplete profile database, or app crashing at startup (very rare bug though)
3. The lists always come back to top (annoying when scrolling a long list) because otherwise favourites star would not be updated. Consider using a common ViewModel to better handle Property changes

# 5. Design Notes:

Overall, design is handled with styles in the App.xaml file ResourceDictionary.  
If you change one, all buttons, or label of some type will change in the app.

Also, images are stored in the asset catalogue for IOS, and in several resource directories for Android.  
Then the image source is in Xaml pages, Xamarin links it to platform automatically.

# 6. Troubleshooting:

If some Picker data is wrong or duplicate, check the Degree finder CSV for any misspelling.  
The picker data (Fields, Xs, States) is retrieved from the database degree table:

* Every row has already been added to the DB Degree Table via the CSV file
* Three lists are created: Fields, States, and Xs
* For each degree, we check if the corresponding list already contains the data  
  NB: Fields and Xs columns contain multiple elements, separated by a comma, so **every element has to be correctly spelled and separated by a comma**, white spaces, and extra spaces are deleted so don’t worry about that.
* At the end we have three lists containing the relevant data.

If favourites don’t appear, it may be that Xamarin doesn’t work properly on that given architecture.  
You will have to test on many devices to pinpoint which don’t and do work.  
Try changing options in the Properties files (of Android or IOS project), otherwise just delete support for the given architecture (also through Properties).

Sometimes, when starting the app and navigating in one of the lists, an unfavourited item will have a full star, the cause for this is unknown to me. I would start with DataTemplate bugs, or file path issues…

Jobs not displaying? The way jobs are acquired from the csv is a little dodgy.  
First line has all the Xs. We put them into a list.  
Then we skip lines until we read something in the column 0: “SCIENCE”, we register that as the current field, and will keep checking every column (moduo 2) for a job name, and salary, then move on to next row.  
When we find column 0 of a row has something new, it becomes our new field, and we repeat, until nothing is found.  
So it’s pretty fragile in case of any modification. Check the code for any details. But it would be good practice to find a more structured way to do it (like the Degree Finder).

App not launching? Debug it. And check any misspellings in the code, Xamarin is a pretty fragile framework, it hasn’t matured yet and sometimes you will get errors because you wrote true instead of True in a Xaml page. So be patient, and rigorous.

Good luck,   
Edgar Munro.