Mobile Computing Assignment #2
Multiplatform Design and Development

Cláudia Marinho
up201404493

Tiago Silva
up201402841

December 16th, 2018

1 Introduction

For this assignment, we were tasked with the design and development of a multiplatform application that allows users to see and compare NASDAQ companies' stock price and evolution.

In this application, the user should be able to select one or two companies from a list of common NASDAQ companies and see a graphic showing the evolution of the close session quote of the selected companies for the last 7 or 30 days, being the user the one to choose the range of days to be shown in the graphic.

Below, we will talk about the main features, the architecture, interface, testing and use cases of the application we developed in the context of this assignment.

2 Features

The main features of the multiplatform application we developed for this assignment include:

- Provides a list of ten common NASDAQ companies with some relevant information about them, such as name, symbol and logo;
- Allows users to identify the current stock prices of the NASDAQ companies mentioned before;
- Selection of one or two companies from the aforementioned list to generate a evolution graphic;
- Selection of the range of days to be shown on the graphic (between 7 or 30);

- Display a graphic that shows the evolution of the stock prices over time with appropriate scaling and labeling:
- Allows users to compare the stock price evolution of two companies, by showing their graphics superimposed with different colors.

3 Architecture

We developed the application using **Xamarin** and the interface of the application was designed with **Xamarin.Forms**. We selected **Windows Universal Platform (UWP)** and **Android** as the testing platforms, since we didn't have access to any iOS or macOS devices. The overall application architecture is illustrated below:

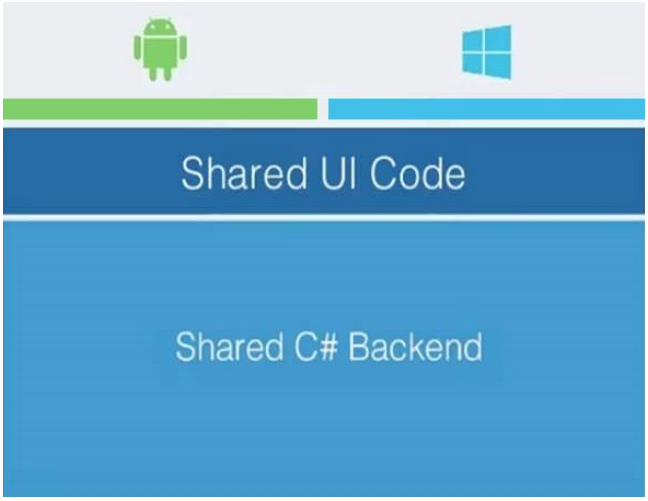

Figure 1: Overall Application Architecture.

To generate the graphics for the companies, we used the **SkiaSharp** library, the universal 2D graphics API for Xamarin.Forms.

Information about the NASDAQ companies is obtained using an external web service, supplied by barchart's market data API with a free subscription. To use this API, firstly we had to register in the website and then use our API key to make API calls. In the application, we used this service directly through two API calls, listed below:

- GET https://marketdata.websol.barchart.com/getHistory.json?apikey=<api_key>&symbol=<company_symbol>&type=daily&startDate=

<start_date>

- GET https://marketdata.websol.barchart.com/getQuote.json?apikey=<api_key>&symbols=<company_symbols>

The first API call, **getHistory** is used to get information about the stock price evolution of a certain company. The purpose of this information is to generate the evolution graphics for the companies selected by the user. The information that needs to be provided in the url to make this call are the *<api_key>*, the key provided by the service to make API calls, *<company_symbol>*, the symbol that identifies a company and the *<start_date>*, the start date of the historical data query.

The second API call, **getQuote** is used to get the current quotes of the companies, which are shown to the user. The information that needs to be provided in the url to make this call are the *<api_key>*, the key provided by the service to make API calls and *<company_symbol>*, the symbols that identify the companies we're querying for.

4 Interface

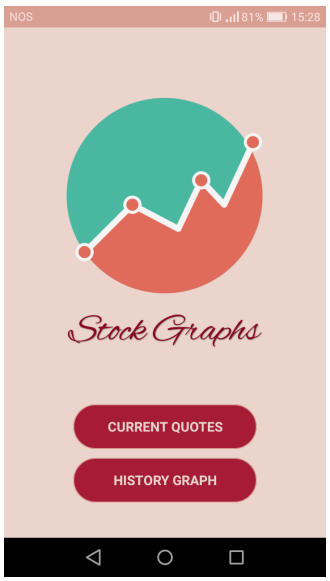

Figure 2: Main Screen on Android

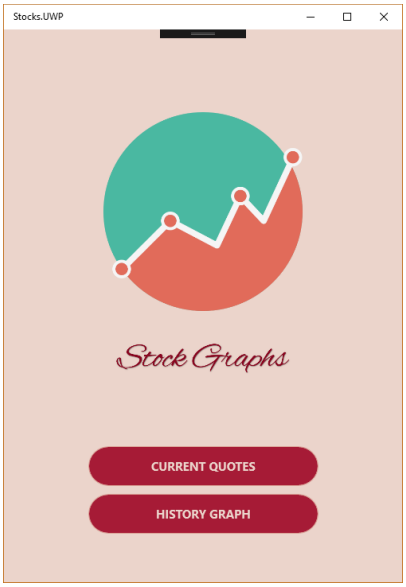

Figure 3: Main Screen on UWP

The first screen the user faces when he opens the application is the Main Screen (**figures 2 & 3**), which shows the application logo and two buttons that redirect to the main features of the application. The first button, *Current Quotes*, allows the user to see the current quotes of ten common NASDAQ companies and the second button, *History Graph*, allows the user to select one or two companies and see their stock history graphs.

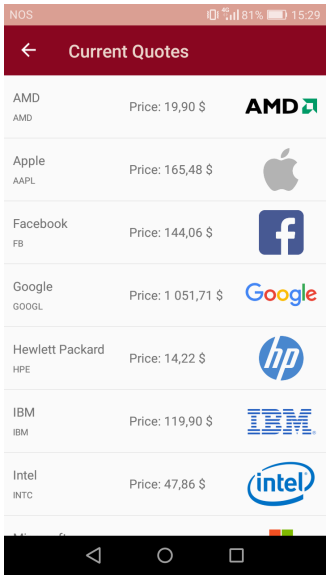

Figure 4: Current Quotes on Android

Figure 5: Current Quotes on UWP

On the Current Quotes Screen (**figures 4 & 5**), the user is shown a list of ten companies and the current quotes for each of them is shown in the middle with the label *"Price"*. Besides this, some information about each company is also shown such as the company name, symbol and logo. It's also important to mention that the user must have Internet Connection to see this screen, otherwise an error message will be shown.

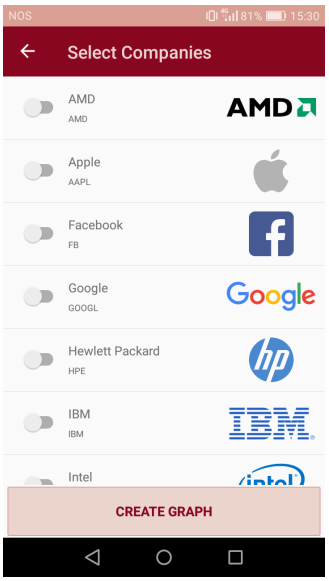

Figure 6: Companies List on Android

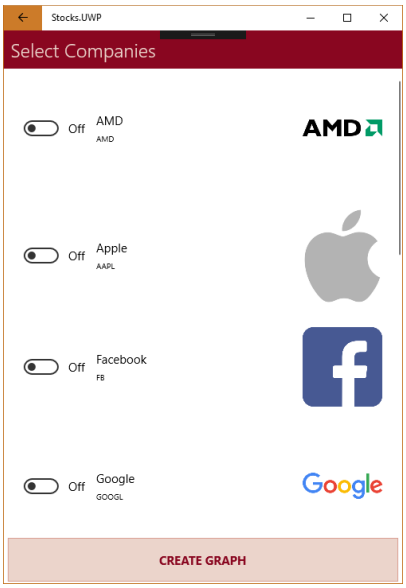

Figure 7: Companies List on UWP

As for the Companies List Screen (**figures 6 & 7**), each company is assigned to a switch that can either be on or off, being that if a switch is on, then the corresponding company is selected. From this list, the user can select one or two companies and then click on the button with the label "Create Graph" to generate the evolution graph for the selected company or companies. If the user tries to click on the mentioned button with no companies selected or with more than 2 companies selected, an error message is shown.

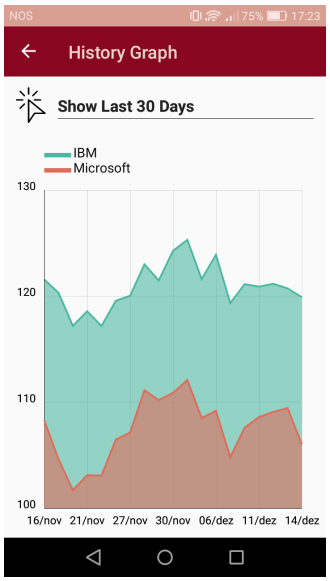

Figure 8: Evolution Graph on Android

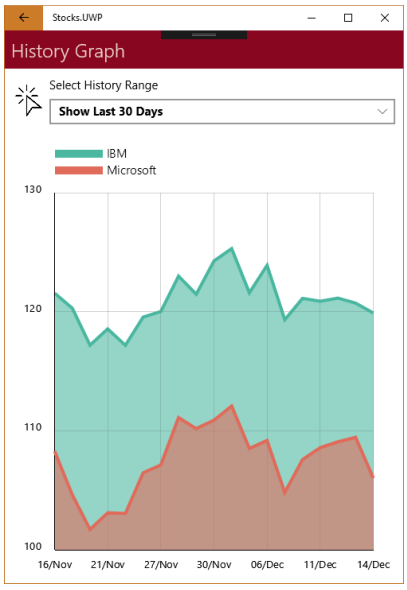

Figure 9: Evolution Graph on UWP

Lastly, on the History Graph Screen (example shown on **figures 8 & 9**), a picker to select the history range is displayed above the graphic. The picker allows the user to choose the stock history range to be displayed on the graphic, either for the last 7 days or for the last 30 days. It's also important to mention that the user must have Internet Connection to see this screen, otherwise an error message will be shown.

Labels for the colors shown in the graphic are also displayed above it so that the user is able to identify which company the evolution graph that is being shown belongs to. As for the axes, on the *X axis* of the graphic, the dates corresponding to the stock values shown on the graphic are displayed and on the *Y axis*, the scale of these values is shown. The graphic is also displayed in a grid-like format to improve readability.

The scale for the stock values is chosen taking into account the overall minimum and maximum stock values for the companies obtained through the API. After determining these values, their range is calculated and this will determine the increase unit in the scale. This was done with the goal of not showing too much information on *Y axis*, while still showing the required information to interpret the graphic.

The example shown on the **figures 8 & 9** is the evaluation graph for the companies *IBM* and *Microsoft* for the last 30 days. An example showing *AMD*'s

graphic evolution for the last 7 days is shown below:

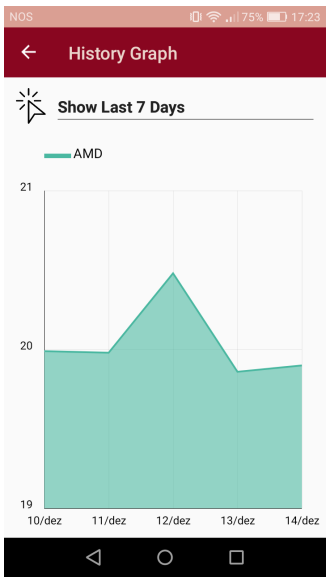

Figure 10: Evolution Graph on Android

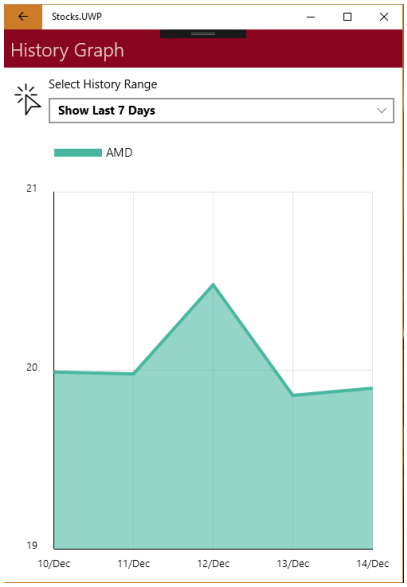

Figure 11: Evolution Graph on UWP

5 Testing

As it was mentioned before, the testing of our application was done both in Android and UWP. This proved to be a challenge, since when we were designing interfaces in Xamarin.Forms, we had to take into account both target platforms and test both applications. As such, in order for the application to look good in both platforms, we had to make some parameters platform-specific, such as the row height in the list views, margins and text sizes in the SkiaSharp graphics.

The SkiaSharp graphics were the ones to cause most of the trouble we had regarding this subject matter. For example, while the text and strokes were being rendered too big and thick in the UWP application, these were too small in the Android application. However, after some adjustments, we got the graphics to render nicely in both platforms.

We also had to do some platform-specific adjustments to match with the application theme, like the color of the status bar in Android and the color of a selected item in a list view (it was bright orange before).

6 Use Cases

The application supports the following use cases shown below:

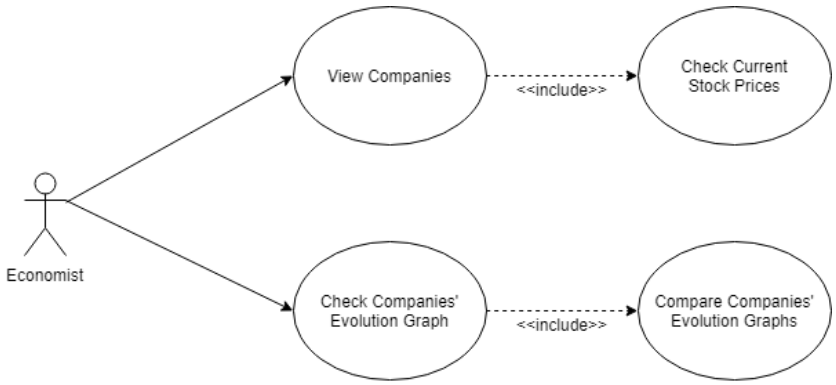

Figure 12: Application's Use Cases.

References

[1] `https://www.barchart.com/ondemand/free-market-data-api`.

[2] `https://www.nasdaq.com/screening/company-list.aspx`

[3] Mobile Computing's Slides