Stocks - My stocks' analysis

Report

Mestrado Integrado em Engenharia Informática e Computação

Computação Móvel

José Gomes - up201305016 Pedro Faria - up201406992

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

December 16, 2018

Contents

1 Introduction

For this second project we were tasked of designing and developing a system for an easy acquisition of stock exchange information about several companies.

The application should give the user a chance to select up to 2 companies. After these companies are chosen, the application shows, in a graphic, how the close value of stocks changed during days. These number of days are provided by the user. The creation of the graphic was made using SkiaSharp from Xamarin.Forms.

The main goal with this assignment is to introduce us tools that help develop applications for the three major smartphone operating systems: Android, iOS and Windows Phone using Xamarin.

In this project we tested our application in 3 different types of devices, being: IOS, Android and UWP(Universal Windows Platform).

2 Architecture

The architecture system that we used is very simple. Our application communicates with MarketData(from Barchart) API directly and interprets the responses in json.

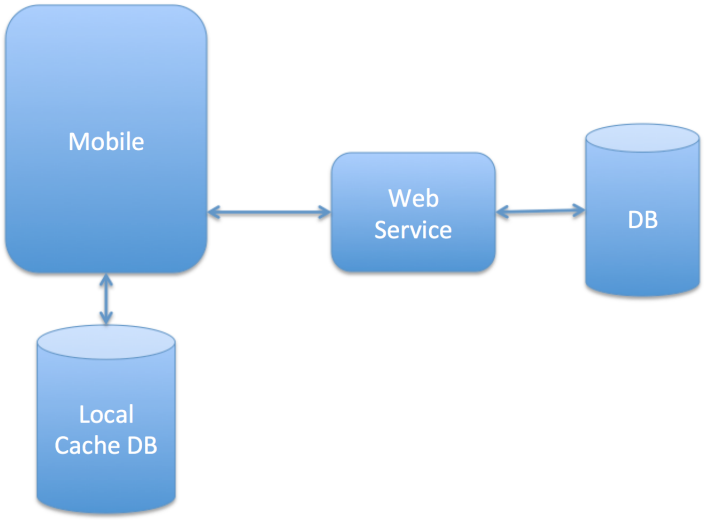

Figure 1: Arquitecture of the system

The architectural pattern we used is Model, View, View-Movel (MVVM). This pattern was present in theoretical classes and helps the developer to cleanly separate the business and presentation logic of an application from its user interface. This way, it helps maintaining a clean separation between application logic and the UI. It also helps to address numerous development issues and can make an application easier to test, maintain and evolve.

The next figure describes how we organized our code in MVVM format.

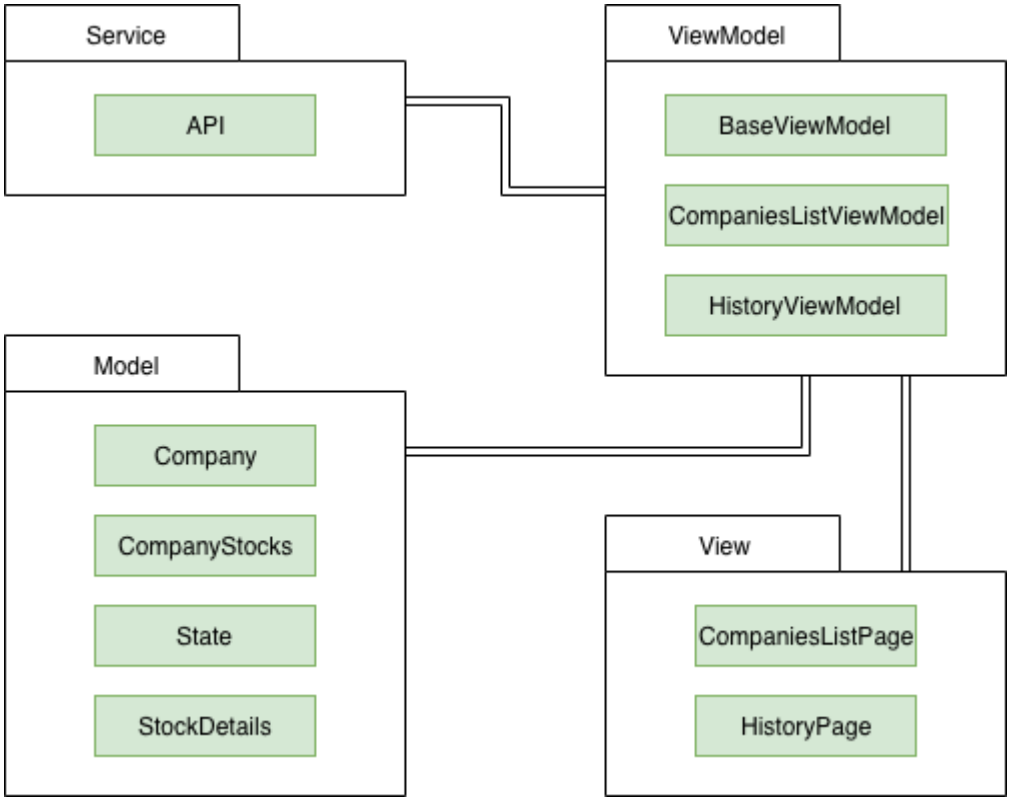

Figure 2: Software Architecture

3 Features

The following pictures show how our application can be handled by an user.

When launching the application, a list of all companies are displayed in a list-view as described in the next image.

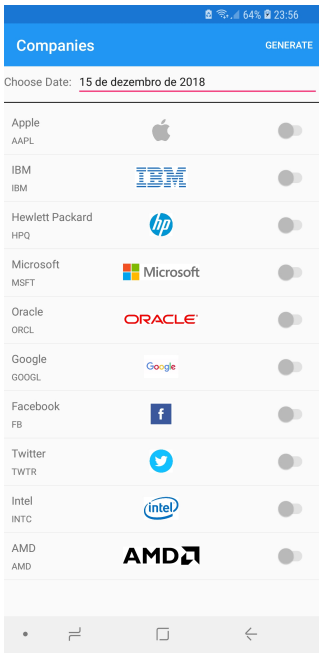

Figure 3: Main page in Android

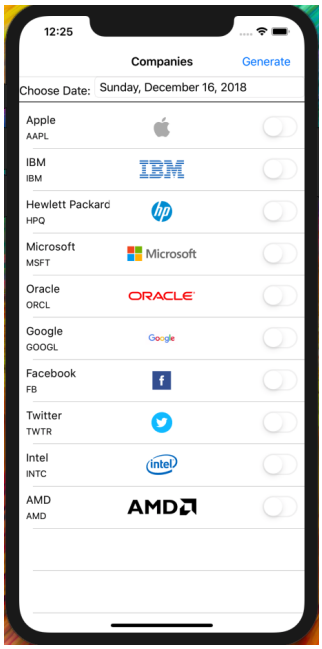

Figure 4: Main page in iOS phone

Figure 5: Main page in Windows

A possible action by the user is choosing the date to check the exchange stocks. User need to choose a minimum of 7 and maximum of 30 days from the current date. If wrong date is chosen by the user, a warning is displayed on the screen. The next image describes the calendar present to the user to choose the date.

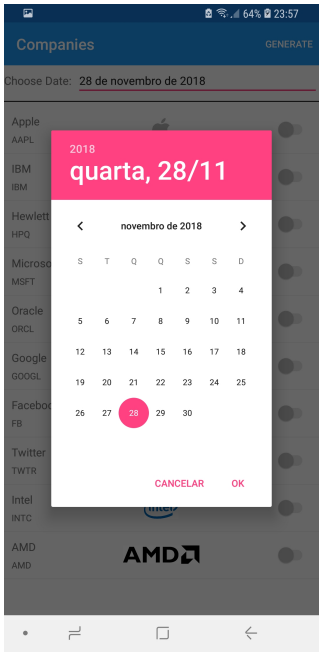

Figure 6: Date picker in Android

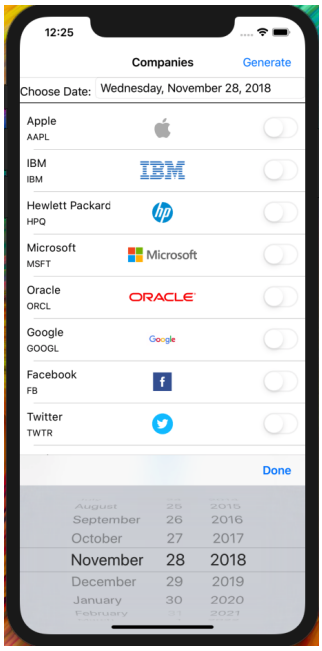

Figure 7: Date picker in iOS phone

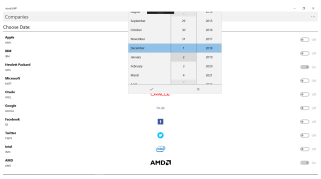

Figure 8: Date picker in Windows

Other possible action by the user can have, is to select the desire companies so that he is able to see the stocks exchanges. User can select 1 or 2 companies to compare their daily stock exchanges. If no company, or more than 2 is selected by the user, a warning is displayed on the screen. The next image describes the user selecting 2 companies.

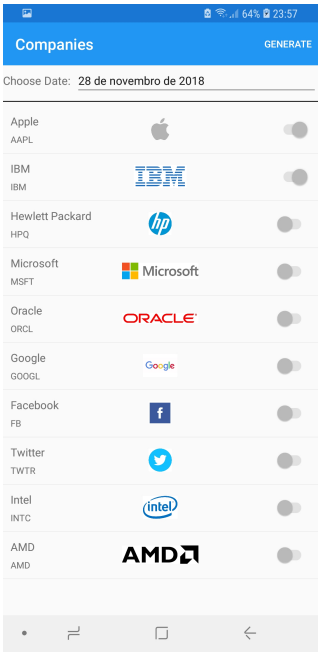

Figure 9: Companies selected in Android

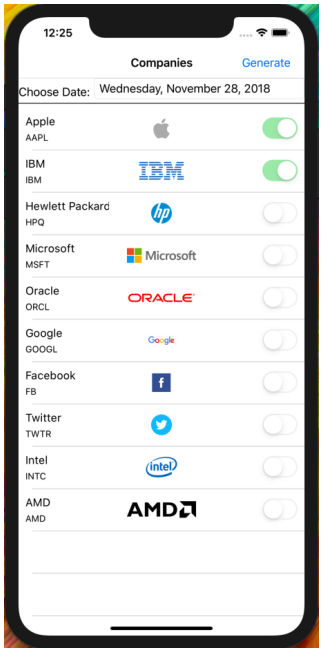

Figure 10: Companies selected in iOS phone

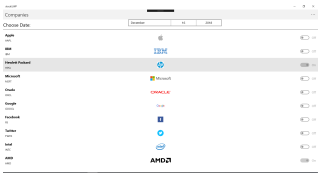

Figure 11: Companies selected in Windows

To see the stock exchange results, after selecting the desired companies and days, user need to click on "Generate" button on top right of the screen. A new view is displayed with a graphic that contains each day stock exchange as open, close, minimum and maximum values from a desired day.

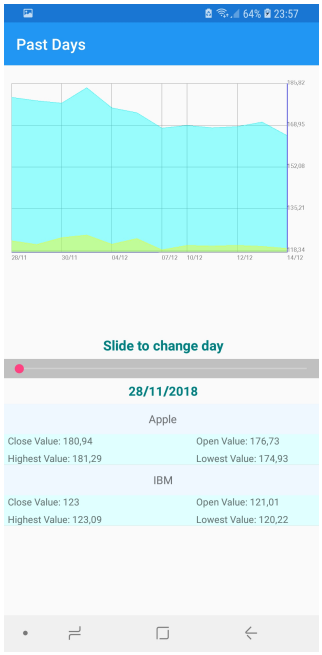

Figure 12: Graphic in Android

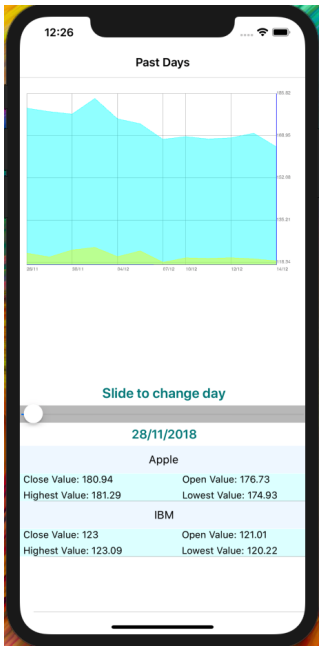

Figure 13: Graphic in iOS phone

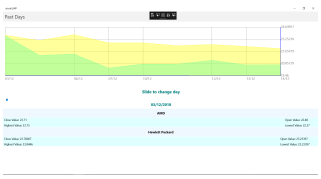

Figure 14: Graphic in Windows

One feature we desired to implement, was the ability of the user to slide trough chosen days to see the close, open, highest and lowest values of a day from the selected companies.

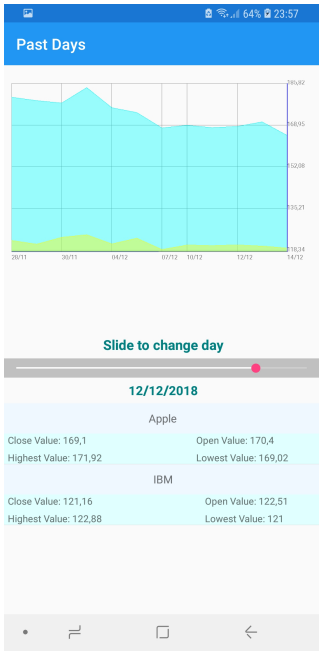

Figure 15: Stock exchange value in Android

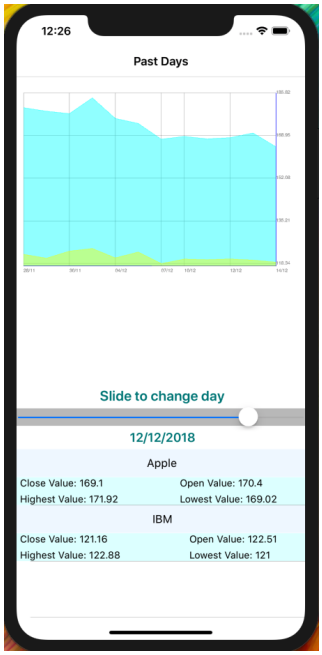

Figure 16: Stock exchange value in iOS phone

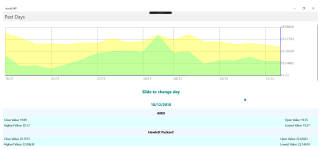

Figure 17: Stock exchange value in Windows

4 Tested Platforms

We choose to test the application on Android, iOS and UWP. We also tested on Mac platform, but unfortunately some important features (DeviceInformation and SkiaSharp) used on shared app logic wasn't compatible with it. We also had some problems of incompatibility when testing in each platform. At first, when the user select a desired company, the view-cell background color changes correctly on Android and UWP, but on iOS that didn't work due to a unknown bug. So we decided to go to a switch button approach and that did work on all platforms.

We tested in android devices and emulators with api equal or higher than 23.

For UWP, we tested in Windows 10 Fall Creators update version 10.0 and build 16299. The UWP project wasn't compatible with the *DependencyService* we were using to show some error messages.

5 Improvements

We decided to put a table below the skiasharp graph so it can provide more info to the user. The other data provided is: open value, lowest and highest value per day for every company selected. This information is displayed on a table that is altered via a slider.

6 Conclusions

Even though there were some incompatibilities between operating systems, we think we were able to achieve the expected result for this project. As we dove into these platforms we understood some nuances of working on a cross-platform application.

Nevertheless we strongly believe that we were able to achieve the main objectives of this assignment.

7 Bibliography

https://docs.microsoft.com/en-us/xamarin/
https://www.barchart.com/ondemand/free-market-data-api

8 Figure Index

List of Figures