

Developing Mobile Applications for Modern Smartphones¹

PAVEL, Alexandru

Spiru Haret University

Scientific Research Center in Mathematics and Computer Science²

pavel.alexandru90@gmail.com

Abstract

Smartphones have become one of the best selling devices on the IT market in the last years, with more units than the PC. The current smartphone landscape is made out of multiple operating systems. Developers want to make their apps available to the widest available market, but that implies handling multiple platforms. Depending on the nature of the app, this can represent a big challenge. The app used here to demonstrate this, although simple, is a good fit to exemplify the problems that might occur during development.

Keywords: *mobile device, mobile operating system, Android, iOS, Windows Phone.*

ACM classification: J.7, K.4.4

1. *Introduction*

In this paper, the three most important mobile operating systems will be discussed: Android, iOS and Windows Phone. The subject of this paper was chosen because the field of mobile applications is dynamic and here to stay. Mobile devices have witnessed a fast evolution, currently out numbering personal computers and gaming consoles. Tackling the challenge of developing an app for the three by most current important mobile operating systems is a truly beneficial method for obtaining better understanding of the integrated development environments and programming languages.

¹Citation: Pavel A., *Developing Mobile Applications for Modern Smartphones*, "An. Univ. Spiru Haret, Ser. Mat.-Inform.", 12(1), pp. 25–30, 2016.

²Master student, Program: Modern Technologies in Information System Engineering.

2. *Android, iOS and Windows Phone*

The Android mobile operating system is developed by Google Inc. Although a big part of this OS is available as Open-Source via AOSP (Android Open Source Project), the preinstalled Google services (Play Store, Google Search, Google Play Services, Google Music, etc.) are Closed-Source [1].

In the 3rd trimester of 2014, Android was the most popular mobile operating system, taking up more than 80% of the Smartphone market [2].

Devices running Android cover a very wide selection of specifications, dimensions and prices. Many producers offer extremely cheap devices (even under 50 USD, in some cases). At the same time, high-end devices are available, providing excellent performance at a higher price. This freedom has allowed Android to become popular even in the low-end market.

The iOS mobile operating system is produced by Apple Inc. and is entirely Closed-Source. All devices running iOS are developed and designed by Apple and built by Foxconn (one of the largest hardware manufacturer, located in Taiwan) and other partners.

In the 3rd trimester of 2014, iOS was the second most popular mobile operating system, taking up roughly 11% of the market. The devices running iOS are available only in the high-end price range, often having price upwards of 500 USD.

The Windows Phone mobile operating system is produced by Microsoft and is entirely Closed-Source. While it can be considered a descendent of Windows Mobile, which was embedded in Pocket PC devices, this operating system has been tailored for the modern smartphone.

Windows Phone, along with other mobile operating systems, completes the remaining market percentage, making it the third most popular mobile OS, judging by the market penetration.

3. *SDK, IDE and development*

The Android platform has the great advantage of providing a SDK (Software Development Kit) for Windows, Max OS and Linux desktop operating systems. Windows Phone offers a SDK only for Windows 8 and above, and the iOS SDK is only available for recent versions of Mac OS.

In the case of iOS, we must also factor in the cost of Apple desktop or laptop computer, as MacOS can legally be operated only on Apple hardware. This is important because the high costs of Apple hardware can limit the number of new, independent, developers.

The IDEs (integrated development environment) provided in each of the SDKs are modern, complex tools, which can be described using the "easy to learn, difficult to master" ideology. Seasoned developers will find most of the necessary tools available in all SDKs.

The recommended IDE for Android is Android Studio, iOS uses XCode, while Visual Studio is used for Windows Phone. Even though these IDEs are, in large part, similar, Xcode stands apart. The iOS IDE offers a WYSIWYG (What You See Is What You Get) method of defining visual elements in the

.swift code files. This means dragging a line from the visual element you want to manipulate to the swift code file in which the manipulation is executed. Without proper exemplification, this may not be an obvious procedure for an experienced programmer but may seem natural for someone with low or no programming knowledge.

The interfaces building components available in the IDEs are very similar, using mostly the same interface elements and WYSIWYG layout system. Drag and drop is used to place visual elements, such as text fields and buttons, on a canvas representing the screen of the target device.

Each IDE implements a unique project structure, with visible differences. The most important is the programming language used. Android mostly uses Java and C++. Windows Phone uses C Sharp and C++ and iOS implements either Objective C or Swift along with C++.

The C programming language is implemented in all these SDKs for performance reasons, as certain operations require a more powerful language and interpreter. For many of the auxiliary files (such as layout control and various information about the application), Android and iOS implement XML, while Windows Phone implements XAML [3].

The available emulators could not be more different. Unfortunately, the Android emulator is the slowest one, from a performance point of view. Even when configured using Intel Hardware Accelerated Execution Manager (Intel®HAXM) technology, the performance provided is well below that offered by Windows Phone or iOS [4].

While Android can barely offer 30 frames per second in the best case, Windows Phone and iOS offer smooth emulation of the target devices, reducing the time needed to test quick changes to the code.

Unfortunately, none of the emulators can fully emulate the target devices functionality. Bluetooth, headphone connection, third party accessories are only part of the elements that cannot be emulated.

4. *The CursBNR application*

The CursBNR app was created from observing the need of quickly knowing the current exchange rate of certain currencies. The scope of the CursBNR app is to make the exchange rate easily available to the user, while offering a familiar interface and good performance. With user interfaces tailored for each of the operating systems, the application is easy to use, and is accessible even to newcomers to the modern mobile operating systems.

Once the app is launched, the user is greeted with a loading message while, in the background, the most up to date xml files are being downloaded. These files are provided by BNR (National Bank of Romania) to everyone who wants to use the data. Once downloaded onto the device's storage, the xml files are parsed and prepared to be shown on screen. During tests, this procedure took from 0.5 seconds to few seconds, depending on the device used. Afterwards, the loading screen disappears and the user can see the main currency list. This list can be scrolled vertically by sliding the finger up and down on the screen.

Upon tapping any of the currencies, the user will see another screen, showing a ten day evolution of the selected currency, both in text and graph form.

Tapping the physical back button of the device or the one present in the top left of the screen (depending on the operating system) the user will see the main currency list once again.

Long pressing on any currency will copy the current value in the clipboard, for future use or sharing in other documents.

Accessing the currency converter depends on the platform. On Android, this can be accessed by swiping the finger from left to right on any of the currencies. For iOS, there are distinct tabs at the bottom of the screen. On Windows Phone, the proprietary "pivot" interface has been implemented. It can be described as a more fluid tab interface, where the user can navigate either via the tabs at the top of the screen, or by swiping between the screens.

5. *Developing CursBNR for different platforms*

The goal of CursBNR is to make accessible the BNR currency information easier while on the move. That is the reason why the app implements simple visual elements, native to each operating system, along with good performance and low requirements compared to similar apps [4].

Regarding the user interface, it can be seen as a challenge to mold the app concept to each operating system. This is obvious when using visual elements that are not available in all operating systems, as mentioned in the previous paragraph. While the concept of a web app (the application itself being only a web browser element, and the interface and logic being implemented via web technologies) is easier to use from a developer standpoint, it can lead to poor performance and reduced user satisfaction due to the way each operating system handles web-browsing elements within applications, as well as the need to download html and css elements at each page load.

The development difficulty rises with the complexity of the project. Developing a similar application using different programming languages, some of the logic elements required some alteration, due to the way each programming language (Java, C Sharp and Swift) works. Text string manipulation needed special attention, because each programming language uses special syntax and different functions for appending and extracting elements. Because each operating system implements a different XMP parsing functions library. This means that, because the content of the XML file is extracted differently, it also means that the data structures holding the extracted information need to be different.

The IDEs, while mostly similar, have certain differences that may cause annoyances. For example, Android Studio requires additional settings and plugins in order to use a subversion based version control system, compared to Visual Studio and Xcode which support it natively. Other differences appear when comparing the debugging tools available for each SDK, as they are quite different, both in utility, as well as in the implemented user interface.

Android, being the most open platform, allows the developer to use any device for developing apps. Windows Phone and iOS require completing an unlocking process, often through a web portal, in order to be allowed to install unsigned code on the devices. Additionally, Windows Phone and iOS require either a one time payment or a subscription in order to unlock the devices and/or to be allowed to sign code.

Developer looking to monetize their apps should turn their eyes towards iOS, as this platform has the most users that are willing to pay for apps. Android and Windows Phone arrive second and third, respectively [5].

The development difficulty rises with the complexity. For CursBNR some of the logic elements required some alteration, as some elements behaved differently under each of the programming languages. Also, string manipulation is handled differently and required revising.

6. *Conclusion*

In conclusion, writing an app for three (or more) platforms at the same time, without a common base, can be hard work, consuming both time and money. In certain cases, it can even be counter productive from a returning investment point of view. The CursBNR app was first developed for Android, then for Windows Phone and lastly, for iOS, once the Swift public beta became available.

Comparing the three platforms, no clear winner can be chosen. Each one caters to the needs of some users. iOS offers a simple interface, solid performance and many applications, but high entry price and almost no ability of customising the interface. Android is highly customizable, offers fairly good performance, but system updates are slow and has a smaller number of high quality applications. Windows Phone offers great performance even on low end devices, but a very small number of quality applications and low customizability.

This paper helps to understand the concepts and intricacies of app development for each of the three operating systems.

Only a small part of the differences and similarities of the three platforms have been presented, as in the case of more complex projects, more differences can be uncovered.

Developing the CursBNR app for each of the three platforms did not pose great difficulties, but certain aspects required consulting online forums and dedicated developer documentation portals. These portals offer detailed information about each component of the operating systems and SDKs.

It also became clear that more complex projects may require extended development time and building certain custom elements to supplement missing functionality or to offer compatibility with older operating system versions.

References

1. Balky carriers and slow OEMs step aside: Google is defragging Android, <http://arstechnica.com/gadgets/2013/09/balky-carriers-and-slow-oems-step-aside-google-is-defragging-android/>.
2. Smartphone OS Market Share, Q4 2014, <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>.
3. What are XAML and XML? Why we use XAML in Silverlight and WPF, <http://www.c-sharpcorner.com/uploadfile/abhikumarvatsa/what-are-xaml-and-xml-why-we-use-xaml-in-silverlight-and-wpf/>
4. Khang Vo, *Pro iOS Apps Performance Optimization*, "Apress", 2011.
5. Pavel Alexandru, *Technici de dezvoltare a aplicațiilor mobile*, Licence degree thesis, Universitatea Spiru Haret București, Facultatea de Matematică și Informatică, România, 2013.
6. Android vs. iOS: User Differences Every Developer Should Know <https://www.comscore.com/esl/Insights/Blog/Android-vs-iOS-User-Differences-Every-Developer-Should-Know>,