Wrocław University of Science and Technology
Faculty of Information and Communication Technology

Field of study: **IST**
Speciality: **CE**

MASTER THESIS

Title

Maciej Sroczek

Supervisor
dr inż. Dariusz Konieczny

Keywords: mobile app performance, native, cross-platform, Kotlin, Swift, Flutter, React Native, Ionic

WROCŁAW 2023

ABSTRACT

There are various aspects affecting the overall perception of quality of a mobile application, with performance being one of the most significant, especially from the perspective of the user. Having that in mind, it is crucial to understand the differences between the available mobile development approaches and in which use cases they are able to provide the highest value.

The purpose of this master's thesis was to perform a comparative analysis of the performance of mobile applications built using both native and cross-platform solutions. Exemplary applications were implemented with Kotlin, Swift, Flutter, React Native, and Ionic to be used as the environment for the experiments. The experiments provided results considering the selected performance metrics, e.g., CPU, memory, and power usage. The results were interpreted in order to find benefits and/or weaknesses for each studied solution, as well as to try to define optimal scenarios for their use.

STRESZCZENIE

Na ogólne postrzeganie jakości aplikacji mobilnej wpływają różne aspekty, przy czym wydajność jest jednym z najistotniejszych, zwłaszcza z perspektywy użytkownika. Mając to na uwadze, kluczowe jest zrozumienie różnic pomiędzy dostępnymi podejściami do wytwarzania aplikacji mobilnych i w jakich przypadkach użycia są one w stanie zapewnić najwyższą wartość.

Celem niniejszej pracy magisterskiej było przeprowadzenie analizy porównawczej wydajności aplikacji mobilnych zbudowanych z wykorzystaniem rozwiązań natywnych oraz cross-platformowych. Przykładowe aplikacje zostały zaimplementowane przy użyciu Kotlin, Swift, Flutter, React Native oraz Ionic, aby posłużyć jako środowisko do przeprowadzenia eksperymentów. Eksperymenty dostarczyły wyniki uwzględniające wybrane metryki wydajności, np. zużycie procesora, pamięci i energii. Wyniki zostały zinterpretowane w celu znalezienia korzyści i/lub słabości dla każdego badanego rozwiązania, a także próby zdefiniowania optymalnych scenariuszy ich wykorzystania.

CONTENTS

1. INTRODUCTION

Over the last few years, mobile devices, such as smartphones, tablets, or even smartwatches, could be seen as a rather essential part of human lives. This is confirmed by the big and still increasing number of over 7 billion mobile users across the world [9]. Because nearly 90 percent of users spend their time using different apps, the number of mobile app downloads is very high, at over 200 billion in 2020, which has a direct impact on the expansion of the mobile app market [10]. The growth of the mentioned market results in the evolution of different implementation methods for mobile development, with native and cross-platform being the most widely used.

Native mobile development implies creating software that can only be run on a specific platform (operating system), such as Android or iOS [1]. In order to do so, platform-specific tools must be utilized. In the case of Android, the programming language Kotlin may be used, and in the case of iOS, Swift. While it can be seen as a limitation, it provides some advantages, such as being able to use different elements of the system directly and, with that, maximize the achievable performance.

Cross-platform mobile development aims to eliminate the need to implement multiple versions of the same mobile app in order to make it available for users of different platforms. This method assumes the use of a single codebase that enables building the app for various operating systems. From the perspective of a user, each of them should perform and look as if they were implemented natively [3]. Such an approach quickly became popular among developers, including successful companies such as Meta and Google [2]. Some examples of cross-platform frameworks are Flutter, React Native, and Ionic.

All of the differences between the above-mentioned implementation approaches can make them more or less applicable in various scenarios. The selection of either native or cross-platform development method as well as the specific technology is really important because it may directly affect aspects such as development time, cost, and overall end-product quality. However, most of the popular solutions are constantly being updated, which leads to the necessity of recurrent comparative analysis in order to obtain the most up-to-date state of the art. Such knowledge will then be helpful to determine in which cases different development approaches and tools should be optimally used.

1.1. THE PURPOSE OF THE THESIS

The purpose of this master's thesis is to carry out research on the performance of selected cross-platform frameworks in comparison to each other and to native development methods. A number of metrics will be selected for analysis based on a literature review and personal experience. Exemplary applications will be prepared as an environment for the experiments. The results will form the basis for defining the advantages and downsides of developing single codebase cross-platform applications. Furthermore, optimal scenarios of use will be proposed for each studied framework and native technology.

1.2. THE SCOPE OF THE THESIS

To begin with, a problem analysis will be performed, which will result in defining the specifications for the experiments to be carried out. Conducted experiments will provide data for further analysis, which will be organized into groups based on the experiment environments, studied platforms, and frameworks. The results will be interpreted in the context of quality and possible optimal use-cases for implementing mobile applications using the selected frameworks and native methods. All of the research must be documented.

1.3. THE STRUCTURE OF THE THESIS

The thesis has been divided into seven chapters. The first chapter aims to provide a brief introduction to the topic. The second chapter contains the literature review, which helps to present the relevancy of the subject matter as well as provide knowledge necessary for the further work. In the third chapter, the research method is mostly defined based on the literature review. The fourth chapter concerns the implementation of testing environments and the realization of prepared experiments. In the fifth chapter, the results from performed experiments are visualized and described. The sixth chapter contains the discussion that emerged from the experiment results and the conclusions drawn. Finally, in the last chapter, the complete work is summarized and key takeaways are featured. Additionally, limitations are explained, and suggestions for future work are proposed. The dissertation closes with a bibliography as well as lists of figures and tables.

2. LITERATURE REVIEW

2.1. RELATED WORK

2.2. MOBILE DEVELOPMENT RELEVANCY

2.3. DEVELOPMENT APPROACHES

The definition of mobile development can be more or less general. It can be seen as a broad process of implementing a mobile application starting with planning and designing and finishing with testing and releasing. A more software-oriented definition is that mobile development simply refers to implementing an application for mobile devices by coding it using a selected technology stack [4].

Mobile development can become a complex task considering the variety of devices and platforms available on the market. There are many different approaches available and in order to choose one over another the mobile application requirements should be taken into account as well as target platforms and devices, development and time costs [11].

In this chapter, there are presented selected popular approaches to mobile application development. Each of them is described mainly in the context of architecture, technology stack and tools, platforms supported and possible advantages or disadvantages.

2.3.1. Native mobile development

Since almost a decade, the mobile operating system market has been dominated by Android and iOS, reaching 99,3% in March 2023 [6]. For this reason, in the context of this thesis only the above mentioned operating systems are being taken into account.

Native mobile development encompasses building mobile applications that can only be implemented using a platform-specific programming language and deployed to a single operating system [12]. Such an approach brings with it the necessity for creating and maintaining multiple codebases and with that possibly multiple development teams [5]. The number of distinct codebases does is not simply equal to the number of target platforms as different versions of a single platform may require to be implemented independently.

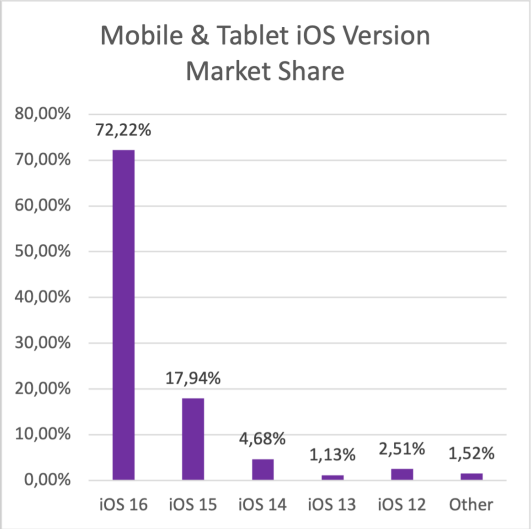

Fig. 2.1. iOS version market share [8]

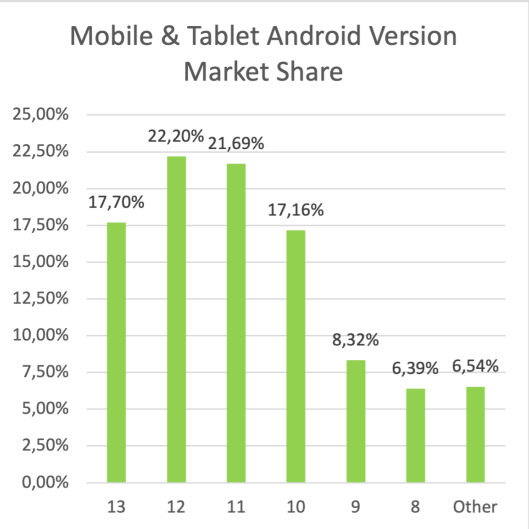

Fig. 2.2. Android version market share [7]

As can be seen in the Figure 2.1, in case of iOS, almost 90% of devices are running either the most or second-most recent major version of the operating system. Therefore, when targeting the Apple's system, probably a single codebase would be enough.

However, in case of Android, there is a high level of market fragmentation, as nearly 20% of smarthones or tablets are running older versions released as far as in 2015 (Figure 2.2). Because there are limitations such as deprecation of code commands and API bahavior changes between distant versions, multiple codebases may be chosen to be maintained separately per a single mobile application.

2.3.1.1. Android

Include Material!!!

2.3.1.2. iOS

Include Cupertino!!!

2.3.2. Web development

2.3.3. Cross-platform mobile development

2.3.3.1. Flutter

2.3.3.2. React Native

2.3.3.3. Ionic

2.3.3.4. Comparison

tutaj tabelka z frameworkami I w kolumnach rozne elementy, np., supported platforms, itd.?

2.4. EVALUATION OF CROSS-PLATFORM FRAMEWORKS

2.5. PERFORMANCE MEASUREMENT

2.5.1. Mobile development

2.5.2. Web development

3. RESEARCH METHOD

3.1. PERFORMANCE METRICS

3.1.1. Mobile environment

3.1.2. Web environment

3.2. RESEARCH SCENARIOS

3.3. TESTING TOOLS

3.3.1. Mobile environment

3.3.2. Web environment

3.4. TESTING DEVICES

4. IMPLEMENTATION OF SAMPLE APPLICATIONS

4.1. APP 1???

5. RESEARCH RESULTS

5.1. MOBILE ENVIRONMENT

5.1.1. App 1???

5.2. WEB ENVIRONMENT

5.2.1. App 1???

6. DISCUSSION

6.1. MOBILE ENVIRONMENT

6.2. WEB ENVIRONMENT

7. SUMMARY

7.1. CONTRIBUTION

7.2. LIMITATIONS

7.3. SUGGESTIONS FOR FUTURE WORK

Flutter Impeller

BIBLIOGRAPHY

[1] Competition Markets Authority, *Mobile ecosystems. Market study final report.* Updated 4 Aug. 2022.

[2] Kotlin, *The six most popular cross-platform app development frameworks*, `https://kotlinlang.org/docs/cross-platform-frameworks.html`. Accessed 18 Apr. 2023.

[3] Lachgar, M., Hanine, M., Benouda, H., Ommane, Y., *Decision framework for cross-platform mobile development frameworks using an integrated multi-criteria decision-making methodology*, International Journal of Mobile Computing and Multimedia Communications 13(1). 2022.

[4] Microsoft, *What is mobile application development?*, `https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-mobile-app-development/#definition`. Accessed 22 Apr. 2023.

[5] Mota, D., Martinho, R., *An approach to assess the performance of mobile applications: A case study of multiplatform development frameworks*, Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering - Volume 1: ENASE,. 2021.

[6] Statcounter, *Mobile operating system market share worldwide*, `https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202303-202303-bar`. Accessed 23 Apr. 2023.

[7] Statcounter, *Mobile tablet android version market share worldwide*, `https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide/#monthly-202303-202303-bar`. Accessed 23 Apr. 2023.

[8] Statcounter, *Mobile tablet ios version market share worldwide*, `https://gs.statcounter.com/ios-version-market-share/mobile-tablet/worldwide/#monthly-202303-202303-bar`. Accessed 23 Apr. 2023.

[9] Statista, *Forecast number of mobile users worldwide 2020-2025*, `https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/`. Accessed 18 Apr. 2023.

[10] TechJury, *55+ jaw dropping app usage statistics in 2023*, `https://techjury.net/blog/app-usage-statistics/#gref`. Accessed 18 Apr. 2023.

[11] Velvetech, *5 key mobile development approaches*, `https://www.velvetech.com/blog/5-key-mobile-development-approaches/`. Accessed 22 Apr. 2023.

[12] Vilček, T., Jakopec, T., *Comparative analysis of tools for development of native and hybrid mobile applications*, 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). 2017.

LIST OF FIGURES

LIST OF TABLES