

Industrial Practitioner Perspective of Mobile Applications Programming Languages and Systems

Amira T. Mahmoud^{1*}, Ahmad A. Muhammad², Ahmed H. Yousef³, Hala H.Zayed⁴, Walaa Medhat⁵, Sahar Selim⁶
School of Information Technology and Computer Science, Nile University, Giza, Egypt^{1,2,4,5,6}
Center for Informatics Science, Nile University, Giza, Egypt^{1,6}
Faculty of Engineering, Egypt University of Informatics, Cairo, Egypt³
Faculty of Engineering, Ain Shams University, Cairo, Egypt³
Faculty of Computers and AI, Benha University, Benha, Egypt^{4,5}

Abstract—The growth of mobile application development industry made it crucial for researchers to study the industry practices of choosing mobile applications programming languages, systems, and tools. With the increased attention of cross-platform mobile applications development from both researchers and industry, this paper aims at answering the question of whether most of the industries are using cross-platform development or native development. The paper collects feedback about industry's most used mobile development systems. In addition, it provides a map of the different technologies used by mobile applications development companies according to some of the demographics like company size and location. An online questionnaire is carried out to collect the data. The survey targeted both amateur and professional mobile developers. A total of 85 participants participated in answering the survey. Qualitative analysis using descriptive statistics is done on the results of the survey. Although the results show that there is an industrial trend towards using the cross-platform languages, native development is still used by the well-established companies. More than 50% of the participants are found to be aware of the performance issues of the cross-platform development languages and frameworks. Comparison with findings of related work is discussed which raises more research questions and draws out future research in this field.

Keywords—Android; cross-platform; development; iOS; mobile applications, questionnaire

I. INTRODUCTION

Mobile applications are becoming essential in human lives. During the corona pandemic, many countries had totally shut down to protect their people from the crisis. Mobile applications played a very important role in providing people the ability to do all their daily tasks through their mobiles. Health organizations and governments have developed numerous mobile applications for managing the pandemic [1]. Hundreds of applications emerged to aid people in their work, education, shopping, entertainment, and more. In fact, the second quarter of 2020 became the largest for mobile app usage, with new downloads reaching billions. Mobile applications are found in two main stores: Apple store for iOS platform and Google play store for Android platform. Recently, Huawei released a new platform called Harmony OS for Huawei mobile phones and devices. This release was a response for the Huawei ban which happened in May 2019 [2].

Development of mobile applications for the different platforms is done in one of two ways: either the applications are developed natively for each platform, or the application is developed once on a cross-platform framework. Native development means to develop the application in java or Kotlin for Android and in swift or objective C for iOS. However, for the past few years, swift is being used more than objective C and is considered the official language for iOS platform. Regarding Android in 2018, Google announced that they support Kotlin for Android [3]. Native development might take much time, money, and effort from mobile application vendors. However, native development provides the flexibility for developers to handle platform-specific functionalities [4] like push notifications, camera access, and GPS.

On the other hand, Cross-platform development is a way adopted by several frameworks that depend on variant methodologies to save the time and effort of developing the application specifically for each platform. Cross-platform frameworks have their own challenges and limitations. One of the most important challenges is the dependency of these frameworks on languages that are different from the native languages. This implies developers to learn these new languages and frameworks. Many researchers evaluated these frameworks and compared them to each other and to the native development [5]–[10]. Most of this research figured out that cross-platform development has limitations regarding the performance of the product applications. These limitations include memory, speed, user experience, and security. However, the inventors of these frameworks are enhancing and improving their tools every day. Many libraries are emerging to aid developers and increase the flexibility of developing the applications.

Researchers are continuing to introduce new cross-platform techniques. They depend on the research findings that the existing cross-platform frameworks are not closing the gap between developers and end-users' needs regarding applications' development and performance. Therefore, it is essential to make studies that aim at investigating the industry and practitioners' feedback on using cross-platform frameworks. In addition, researchers also need to know how industry practitioners use Kotlin, compared to Java, after six years of 2017 announcement of Google. On the other hand, there is a need to investigate whether there are developers for the new platform released by Huawei company, or the platform

is not being used yet. This feedback will lead researchers to industrial needs regarding the cross-platform and native development techniques and their practical usage.

The aim of this empirical study is to answer three research questions that are believed will aid researchers in this field. The questions are presented in section three. These questions are answered by conducting an online questionnaire among junior and professional mobile applications practitioners. Analysis of the survey results will be used to draw a conclusion on the practitioners' perspective regarding mobile application development techniques, which is the most used, and what are the most performance issues they face. It will also guard practitioners and researchers to decrease the gap between them and use the implications presented in this article in their future work.

The rest of this paper is organized as follows: Section two will present the literature of mobile applications development. Section three will present the research questions and our methodology for creating the questionnaire to answer these questions. In addition, the methodology of analyzing the data is explained. In the fourth Section, the results of the conducted questionnaire are presented in the form of descriptive statistics and graphs. Section five includes discussion of the results and implications for practitioners and researchers. Section six then compares our findings to the related work findings. Finally, a conclusion that summarizes the results and analysis is presented in Section seven.

II. RELATED WORK

The related work and literature of this topic are summarized in two main subsections. Subsection one will highlight the existing tools and approaches. Subsection two will present the survey papers that are done on this topic while subsection three will summarize the practitioner studies done. These studies are the most related to our work.

A. Approaches and Tools

Originally, mobile application development was done using the native development language for each platform. The mobile application was developed once for each platform. Languages used for the native development are java or Kotlin for Android and Objective C or swift for iOS. Some years ago, new approaches appeared to develop the application once for multiple platforms. These approaches are called cross-platform development approaches. These approaches are categorized into: 1. Web-based, 2. Hybrid, 3. Interpreted, 4. Compiled approach, 5. Model driven [11][12].

The web-based approach simply relies on the web languages that are already supported by all platforms. The Hybrid approach hybrids web and native code to reach the native UI of mobile applications. Continuous communication between the web view and native components represents the overhead of this approach. The interpreted approach depends on having a layer that interprets the JavaScript code and bridges the JavaScript engine with the native engine to be able to render the native components. This bridging also represents an overhead for the applications developed by this approach. Compiled approach depends on compiling the source code of an application to another code. Flutter which is considered the

most used cross-platform framework nowadays uses the compiler-based approach. One subcategory lies under the compilation approach is called trans-compilation. Trans-compilation involves compiling source code from one high-level language to another high-level language [13]. Tools introduced in [14]–[19] used this approach to translate source code of mobile applications from java for Android to swift for iOS and vice versa. The limitation of these tools lies in their dependence on having corresponding code for each functionality used in the mobile application. This mapping concept might be successful for small applications, but it is not tested yet on conversion of real or complicated applications. The model driven Development (MDD) approach depend on generation of user interface code and business logic from models and templates of the application [20]. This approach is limited by the features and abilities provided by the models used and the experience of developers in using the models. MD2 is one of the tools which depend on the concept of model driven development [21].

Each of the mentioned approaches has its own advantages and its limitations and drawbacks. It is now totally dependent on the applications features and requirements to select the suitable development approach. However, it is not common to see a developer or even a team of developers work with a different approach for every project depending on its requirements. This raised our research questions about the most used tools and techniques, whether more than one approach is used by the same vendor, and whether developers are aware of the performance issues of the used frameworks.

B. Literature Studies

This section summarizes literature studies related to our work, identifying similarities and differences between our work and each study. A systematic study was conducted to create a classification scheme for existing research in cross-platform mobile app development. The study aimed to identify research gaps and challenges in the field by mapping 30 studies. The primary research question focused on identifying the contributions of each of the 30 included studies [22]. Most of our literature is included in that study.

Bjorn-Hansen et al. [23], conducted an empirical study using an online questionnaire to survey 101 industry practitioners. Their study aimed to identify the most used cross-platform frameworks by developers and the issues they face. Their findings revealed that PhoneGap, Ionic, and React Native were the most used frameworks, but cross-platform solutions still faced performance and user experience issues compared to native solutions. While some of our survey questions overlapped with theirs, the main difference is that our study targets both developers and management and provides more detailed respondent demographics through in-depth questions on job position and experience. Our study thus extends some of the limitations of their work.

Ahmed [24] conducted a qualitative study by using two-phased research approach. The first phase is using systematic literature review to identify nine challenges that are found in literature. Then in the second phase, they interviewed 34 participants from industry to validate the literature by identifying 13 challenges and issues of web, native and hybrid

mobile development with nine of them already extracted from the Systematic Literature Review (SLR). The main difference with our study is that this study used an interview to gather qualitative data regarding issues facing mobile application development while our research, an online survey is used to gather quantitative data on both cross-platform and native mobile development.

Francese et al. used a qualitative study [25] in 2017 to gather qualitative data related to mobile application development and management. They invited four mobile application managers to discuss differences between mobile applications and web applications. The interview outcome was used to plan and create the survey questionnaire. The online survey was sent to 510 developers using their LinkedIn profile, but only 82% responded. The results showed that junior developers are the ones that mainly develop applications and there is a huge issue regarding testing mobile applications.

Puvvala [26], conducted a survey to investigate mobile development, and used a Delphi study with 11 senior developers to identify the top four factors influencing platform choice: development costs, ease of coding, support, and expected return. They then used these factors to create a survey on their impact on developers' platform choices, finding that availability of SDKs was the most significant factor. However, their study differs from ours in that it focused specifically on the factors influencing platform choice for application development.

Biørn-Hansen et al. investigated the approaches of Android mobile applications development through exploring applications on Google Play Store. A dataset called Androzo[27] that has 661,705 apps was used to detect the framework used for developing each of these applications. The investigation aimed to answer three research questions about the distribution, usage of cross-platform development frameworks on the Google Play Store and how the usage of deployed apps changed within the last decade. The findings showed that only 15% of the total dataset were cross platform applications and Cordova was the most used cross-platform framework.

Previous studies have not kept up with the fast-paced advancements in the mobile application development industry, as they only focused on either cross-platform frameworks or native development and did not consider both. Additionally, they lacked information on respondent demographics and did not provide in-depth questions for developers. This paper aims to address these gaps by exploring the relationship between respondent demographics and their responses. Our survey methodology, including the questions, target audience, and response validation process, will be detailed in the following section. The results of our study can also be compared to previous surveys in this field.

III. METHODOLOGY

This study used the empirical research specific steps that were stated in [28] as a reference to formulate the literature review, research questions, survey questions and analysis method. Both the survey guidelines in the review done by JS Molleri et al. [29] and the check list provided in [30] are used

to help us in evaluating our survey questions and methodology. The research questions are formulated after reviewing previous work and figuring the limitations of previous empirical studies on mobile applications development in order to ensure that answering these research questions will decrease the gap between researchers and practitioners. An online questionnaire using Pollfish website [31] is used for gathering the data. The questionnaire is shared with LinkedIn users who are working in the field of mobile application development. The next subsections will present our research questions, the survey questions, and how the survey questions are answering the research questions. Afterwards, the methodology of publishing our survey and how we selected the participants is discussed. Finally, the statistical methodologies that we used to analyze the results are explained.

A. Research Questions

RQ1: Which mobile applications development techniques do practitioners use and which platform they are developing for?

Researchers need to know the percentage of cross-platform users among mobile applications vendors and developers. Are the majority using cross-platform development? or the majority are using native development? And why? It is also important to know whether most companies are developing applications for more than one platform, or the majority is developing for only one platform. Does Huawei new platform have companies and/or developers?

RQ1.1: Which native languages are used the most?

For years, java for Android has been the most used language for developing Android applications. A few years ago, Kotlin language started to be used for Android development. Google announced in 2017 that Kotlin will be the official language for Android. We need to know the impact on practitioners and how much did they switch to Kotlin language after this announcement. A similar story happened between Objective-C language and swift for iOS development. We need to know if swift has taken over objective-C or if there are still some developers using Objective-C language.

RQ1.2: Which Cross-platform framework is used the most?

Former surveys and research were done to answer the question of the most popular and most used cross-platform framework. However, this information is changing rapidly due to the continuous evolution of existing frameworks. In addition, new frameworks are appearing, and researchers may not even be aware of their existence. Therefore, it is very important to investigate which frameworks are being used by developers and which are the most used.

RQ2: Are practitioners realizing that cross-platform development is not closing the gap with native development in terms of performance?

In the research field of mobile applications development, it is well known for researchers that cross-platform tools available do not provide the same performance and user experience as natively developed applications. Research comparing the cross-platform frameworks is done on certain applications and case studies. However, we need the

practitioners' point of view regarding this matter. This question is to investigate the performance and the issues of the cross-platform frameworks that face the practitioners and whether native development is the best way to get a high-performing application.

RQ3: Do developers know about trans-compiler-based solutions? Are developers interested in having a trans-compiler-based conversion tool?

Trans-compiler-based solutions for cross-platform development have been recently introduced by researchers [14]–[19]. These solutions are using compilers to translate the source code of mobile applications from one language to another to be available for more than one platform at a time. Researchers need to know practitioners' feedback about these solutions and whether they are a point of interest or not.

B. Questionnaire Design

In this study, questionnaires are considered because they are the fastest and easiest way to gather information for our research purpose. With wide use of professional social media like Linked in, it is easy to spread an online questionnaire among developers across the globe to obtain very useful qualitative data and analyze this data. However, this research is limited by the questionnaire drawbacks that does not allow us of know why or on what basis each participant chose a certain answer. This work can be extended later by making Interviews to focus on that point.

The questionnaire is created by Pollfish website [32] which was chosen over survey monkey and google forms to overcome the limitations regarding the number of participants and the limitations of forcing users to have a google account which may prevent some participants from taking the survey. While formulating our survey questions, we have considered making the questions clear and unbiased. We have also focused on designing questions that would answer the research questions stated previously.

The survey starts by five demographic questions that ask about age, gender, job status, job position and nationality. All these questions are predefined by Pollfish website except the job position (SQ1) and nationality (Q1) questions; we had to add them manually. The job position question was defined as a screening question since it is used to form the logic of the rest of the survey questions.

Then the following four questions (Q2,Q3,Q6,Q7) or (Q4,Q5,Q6,Q7) are profession questions that would help us draw a more rigid conclusion and deeper analysis of the results and exclude any non-valid responses. It will also help us identify correlations or dependencies between variables. Depending on the role of the participant defined in SQ1, the questions flow will differ as shown in Fig. 1. For managers and developers, we ask about the company they are working for and its location (Q2,Q3). For students, we ask about the university/college they are from and its location (Q4,Q5) while freelancers are directly navigated to Q6. The years of experience are also identified in the profession part for all participants and the number of mobile applications developed by the participant (Q6,Q7).

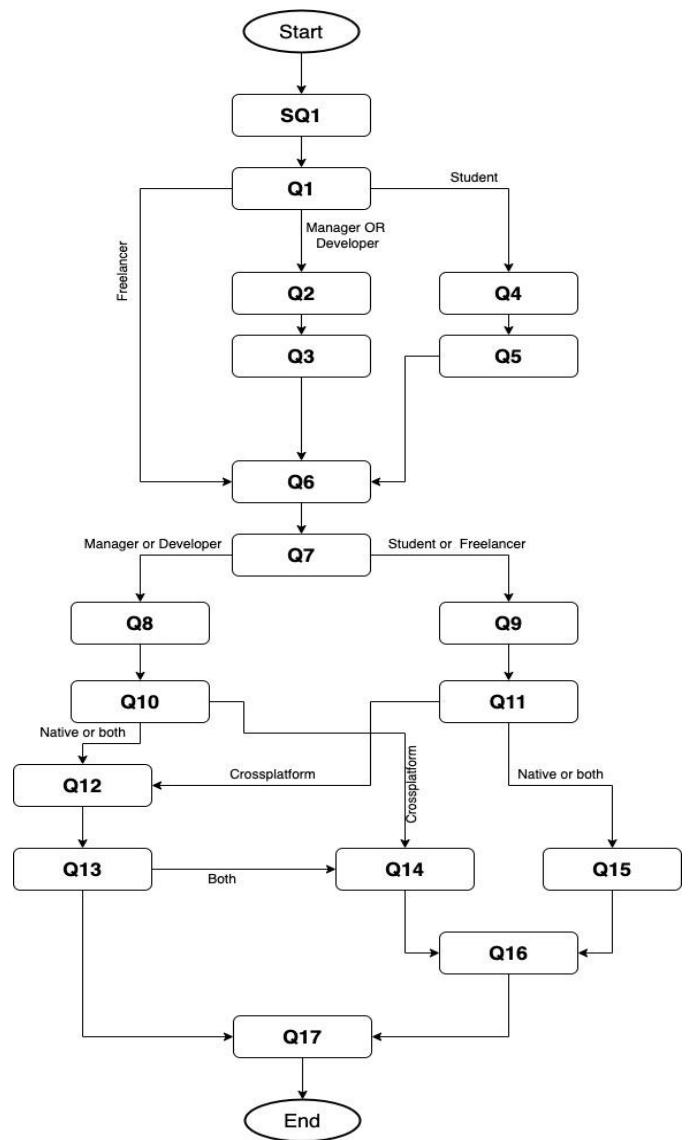


Fig. 1. Survey questions logical flow.

The personal information part is followed by technical questions. The technical questions also have two paths according to the role of the participant. In case the respondent is a student or freelancer, then the question is asked in a personal way “do you use”. If the participant is a manager or developer, then the question is asked targeting the company he/she is working for, like “does your company use”. First technical question (Q8 or Q9) is asking about the target platform whether it is Android, iOS, both platforms, or other platform. The participant can write an alternative. Then a direct question targets the approach used whether it native or cross-platform development (Q10 or Q11). For the native development, another question is asked targeting the language used in developing (Q12,Q13). Choices will be Kotlin or java for Android and swift or objective C for iOS. Then (Q14 and Q15) are targeting the most used cross-platform framework between developers and within companies. The choices of frameworks are added according to monitoring and collecting the most popular frameworks from related work and Statista

website [33]. Some frameworks have been well known for years, and others emerged recently. These two questions use Likert scale from 1 to 5, 1 corresponds for always using the framework and 5 corresponds to never. The question about the framework is followed by a question about the performance issue experienced while using the cross-platform framework (Q16). Finally, a question (Q17) about the familiarity with the Trans-compiler-based approaches is asked with a Likert scale answer between extremely familiar to not familiar at all.

C. Publishing the Questionnaire

Before publishing the questionnaire, we conducted a pilot test to check the availability of the website, the time taken and how clear the questions are. The pilot test was made among five participants of different experiences. The questionnaire took two minutes on average. The questions were clear and understandable. However, some questions needed proper branching. For example, the cross-platform frameworks question should appear only to the participants who chose cross-platform development and should not appear to the participants who chose native development. Similarly, the questions about the native language should not appear to those who chose cross-platform only. The pilot test helped us modify the logic of the questions and branching them correctly. Fig. 1 shows the flow of the questions and the different paths according to the selected answers.

We have created two different links for the survey, one for the Facebook community and the other one for LinkedIn. After publishing on Facebook communities and mobile applications development pages, we got only six participants. On the other hand, the link shared on LinkedIn showed 85 participants. Therefore, we decided to depend on LinkedIn participants. It proved to be a more professional social media through which we can communicate with real practitioners and get accurate answers to our questionnaire. The six responses we got from Facebook were excluded as we couldn't guarantee the level of professionalism the respondents had.

We used the search engine LinkedIn to search for people working in mobile applications development. We used the keywords "mobile application" with applying no filters first. Then, we applied different filters with it. For example, we used "manager" and "senior" keywords in the job title filter. We also used the location filter to reach developers in different countries like USA, Canada, England, India, Pakistan, Germany, Saudi Arabia, Emirates, and many other countries. We applied the filter of each country separately so we can send the survey to multiple people in each country. We have sent the survey link to more than 350 people through LinkedIn.

D. Analysis Methodology

The first step before conducting the analysis is filtering the responses and excluding any responses that were not valid. Two of the responses had answers that showed the participants were not answering the survey correctly. For example, one of the respondents chose all cross-platform frameworks as always used and in the native language question chose other and wrote the name of a cross-platform framework. This showed that either the participant was not reading the survey questions or had very little experience. Therefore, we excluded such

responses to guarantee that the answers and analysis are correct.

Since most of our questions were close ended, we have used descriptive statistics for qualitative analysis to report the survey results. Any provided open-ended text by the participants was used to draw more intuitions and verify the implications of the study. This will be mentioned in discussion sections.

We used a filter to categorize the participants into two main categories according to their role in the mobile applications development industry: Managers or developers and freelancers or students. For some questions, we used this filter to know the results of each category separately.

A filter by the company name was used to identify which method each company uses in developing mobile applications. Companies' names are not revealed in the results to keep the confidentiality and privacy of the respondents' answers. Companies' names were only used to retrieve the company size information from LinkedIn and validate the location information too which is already provided by the participant. Thus, we could categorize for different company sizes and locations whether they are using cross-platform or native development or both, which cross-platform is used by each company category, and which native language is used. There is a difference between the number of managers and developers and number of companies for two main reasons: 1) Not all employed participants provided their company name 2) Some participants were unemployed at the time they answered the survey; therefore, they didn't provide a company name.

Another important filter we used was the cross-platform framework filter. We filtered the answers to frameworks issues question for every framework separately. Thus, for each framework, we could know the most reported issues from its users. All the used filters and the resulting statistics according to each filter used are analyzed and presented by graphs and charts in the results section.

IV. RESULTS

From the results of the survey, we can infer much different information. Before presenting the results and the different filters applied, a general analysis is done on the demographics of the participants like their age, nationality, gender, and years of experience. The general analysis is followed by subsections presenting the results of each question with applying different filters on the results. The technical analysis is presented in three subsections. Each section represents the answer of one of the research questions.

A. Demographics Analysis

A total of 85 participants answered the survey. Most of the participants' nationality is Egyptian with some Indians, Pakistanis, and Belgians. However, they are working in different countries: Egypt, USA, Saudi Arabia, India, Netherlands, Romania, and Germany. The number of participants employed for companies is 71. We had nine participants who didn't provide their company name. However, some of them provided the company location without the

name. Therefore, the companies' location statistics are done for 62 companies as shown in Fig. 2.

Among the 62 companies, 10 of them have unknown names, thus their size is categorized as unknown. The size of the companies was retrieved from Linked in website. Therefore, the size categories are like the categorization in Linked in as shown in Fig. 3. The company size is represented by the number of employees. Most of the companies' sizes are small to moderate sized companies, 27 companies are of size 50 and below. The rest of the 52 companies are almost normally distributed among the rest of the size categories. We have nine companies from 51 to 200 employees. Another nine are from 201 to 1000 employees, and ten companies are of size 1000+.

Among the participants 91% are males the remaining are females. 32% of the participants' age ranges are between 18 to 24, 62% ranges between 25 to 34 and around 5% are above 35. These percentages were also reflected in the years of experience question. Most participants have 2-5 years of experience in the field. Regarding the employment status of the participants, 62% are employed for wages, while the rest are self-employed or out of work currently or students or working in military.

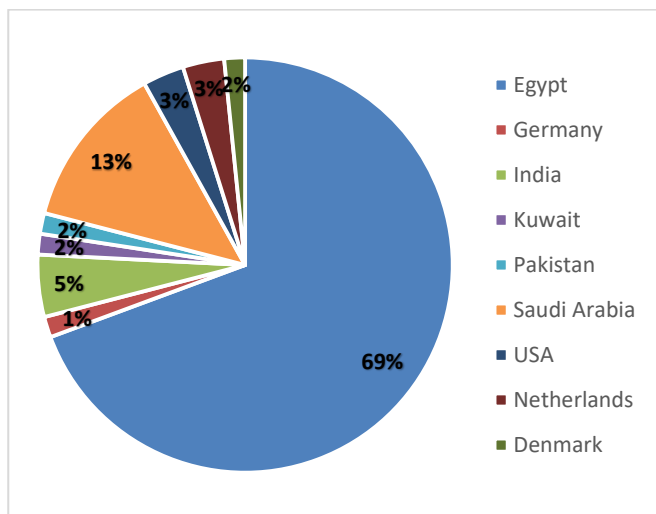


Fig. 2. Participants' companies' locations.

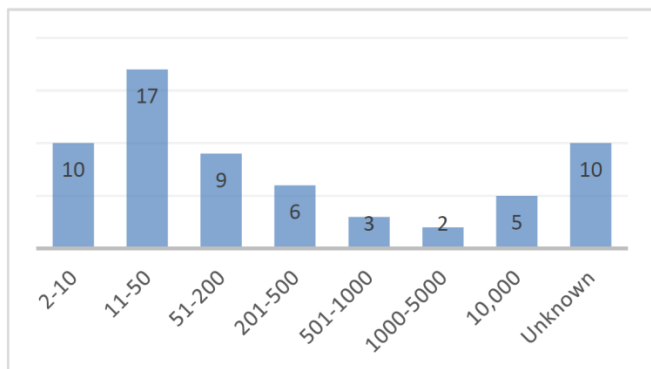


Fig. 3. Companies' size vs number of participants from each category.

We categorized the participants according to their job position into two main categories: Managers or developers who are considered and freelancers or students. This categorization differentiates between participants whose answers represent the companies they are working for and participants who are self-employed. Around 85% of the participants work as managers and developers in the field of mobile applications. 15% are freelancers and students. In the following subsection, we will present statistics answering our research questions. In the statistics of most of the questions, we will use the filter of job position to differentiate between both categories

B. Technical Questions Analysis

The technical questions' part is represented by six questions in the survey for each participant. The analysis of the answers to these questions will answer our previously mentioned research questions. Results of analyzing the participants' answers are presented as follows:

RQ1: Which mobile applications development techniques do practitioners use and which platform they are developing for?

One of the main research questions that we were eager to answer was whether the industry now is going towards cross-platform development or returning to the native development. In the questionnaire, we provided three options for this question: cross-platform, Native or both. The results revealed that most of the participants chose the "both" option. Cross-platform development came in the next level then the native development. Fig. 4 shows the statistics of the approach used by both participants' categories. The statistics showed that most of the freelancers and students are using both approaches together. However, if we observe the companies' representative participants alone, we could notice that companies are either depending on cross-platform frameworks alone or on both cross-platform and native together. Few companies are using the native approach alone. This made us use the company size filter to analyze which companies are using which approach. Table I shows the company size vs. the used approach. From examining the data in the table, we can infer that 38.7% are using cross-platform development alone, 38.7% are using both native and cross-platform development, while 22.58% are using native development alone. Since Egyptian companies have a big share among the companies, we applied the same analysis but with excluding the Egyptian companies. We got nearly the same statistics for the non-Egyptian companies.

The second part of the first RQ asks about the platform which the respondent develops applications for. "Android", "iOS", "both platforms", or "other" are the four options for the platform question in the survey. The results in showed that 73% of participants are developing for both platforms Android and iOS. However, if we examined the freelancer's percentage separately, we could notice that the majority develop applications for Android platform. No participants chose the "other" option which shows that windows phone is obsolete now and that new platforms, like Harmony OS by Huawei, are still not very popular.

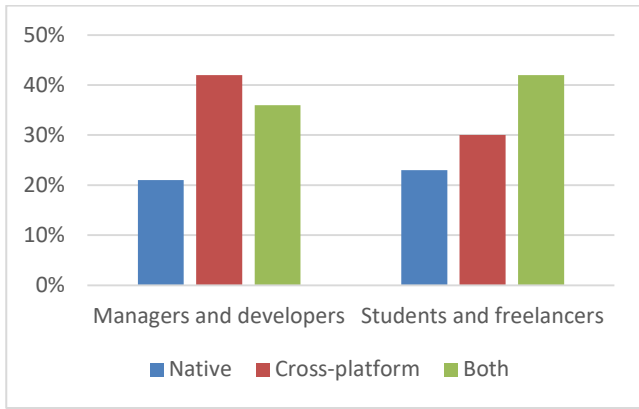


Fig. 4. Percentage of participants using native, Cross-platform or both approaches according to their category.

TABLE I. COMPANY SIZE VS DEVELOPMENT APPROACH

Company size vs used approach	Both	Cross-platform	Native
10,000+	5		
1000-5000	1		1
501-1000	1	1	1
201-500		2	4
51-200	4	5	
11-50	5	6	6
2-10	3	6	1
Unknown	5	4	1
Grand Total	24	24	14

RQ1.1: Which native languages are used the most?

For native development, Java and Kotlin are the languages used to develop Android applications. Swift and Objective C are used for iOS applications. Our aim is to know if there is any language that is preferred among developers over the other. The results showed that 47% of participants chose Kotlin for Android while 39% chose java and the rest use both. On the other hand, 92% chose swift for iOS. When we applied the companies filter, Java and Kotlin for Android are considered evenly used among practitioners. However, swift is used much more than Objective C. only one company is using Objective-C. This company is a very large sized and old governmental company. When we removed Egyptian companies from the analysis, we got the same statistics for both Android and iOS languages. Fig. 5 shows the company size vs. the used native language for 37 companies. These 37 companies are either using native development alone or native development with cross-platform development. The small-sized companies are using Kotlin and java evenly. The medium-sized companies are using Kotlin more than java, while the big companies are using java more than Kotlin.

RQ1.2: Which Cross-platform framework is used the most?

For the cross-platform development selectors, they were asked about the framework they are using. The options of this question included ten different frameworks. The most used framework according to the results is flutter, followed by React

Native, then comes Xamarin, Ionic, JQuery mobile, NativeScript, and Swiftic frameworks. The previous studies showed that React Native was on top [23], [34]. However, now flutter took the first place, and it seems to be very popular among both freelancing developers and companies. Fig. 6 shows the pie chart for seven frameworks. By applying analysis on the companies using cross-platform frameworks, we got similar ranking for the frameworks. When we tried to filter and exclude the small-sized companies then the large-sized companies alone, we got the same result. In addition, excluding Egyptian companies also resulted in the same statistics.

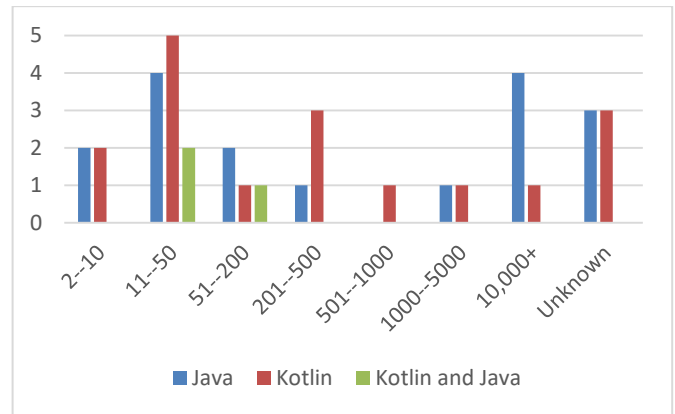


Fig. 5. Company size vs. number of companies using different native languages for android development.

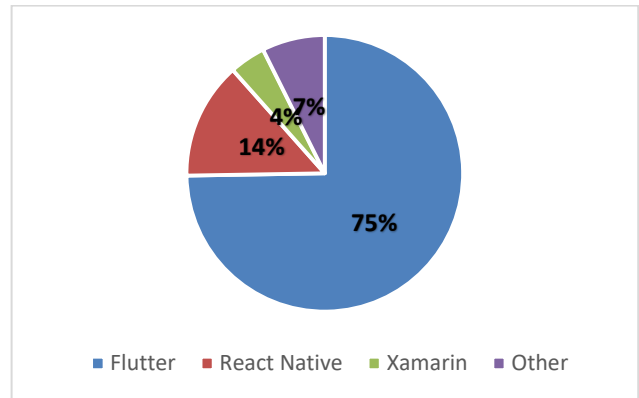


Fig. 6. Statistics of the most used cross-platform frameworks.

RQ2: Are practitioners realizing that cross-platform development is not closing the gap with native development in terms of performance?

To investigate the issues facing the developers using cross-platform frameworks, we asked the developers who are using cross-platform frameworks about the issues they face. This question allowed the participants to choose more than one answer among six different issues: Memory, speed, development effort, security, user experience and testability. An “other” option is also provided to let participants provide any other issues they see. In general, the most reported issue is Memory then comes the Speed and development effort then user experience and security, the least reported issue is testability. By filtering the results on each cross-platform

framework separately, we could see the issues related to each framework. For example, for flutter, the same order of issues is noticed, and this is normal since flutter is the most used framework. For React Native users, security then speed, and development effort were the most reported issues. However, for jQuery and Xamarin, memory, speed and security were the most reported in addition to user experience for Xamarin framework. Fig. 7 shows the statistics of the issues for the four most used frameworks according to the survey results. On the other hand, 12.5% of the participants chose the “other” option only without providing any extra issue. Few of them wrote and they cannot see any issues while using the cross-platform frameworks.

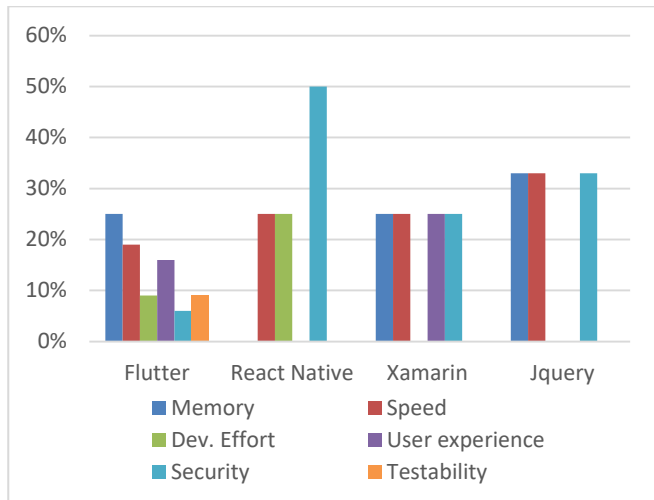


Fig. 7. Reported performance issues by each cross-platform framework.

RQ3: Do developers know about trans-compiler-based solutions? Are developers interested in having a trans-compiler-based conversion tool?

The final question of the survey asked how familiar participants are with the trans-compiler-based cross-platform solutions. 58% of the participants are not familiar with these solutions and 14% are moderately familiar, while 18% are extremely familiar. This shows that the trans-compiler-based solutions are never really tested by practitioners, and they are still under the research umbrella only.

V. DISCUSSION

The most important findings from our survey are discussed as follows: Most companies and practitioners are developing for both Android and iOS platforms. Very few participants chose only one platform. This shows that although developing the applications for more than one platform is a tedious task, and although iOS users are much less than Android users, mobile application vendors are developing applications for both platforms either natively or using cross-platform frameworks. No participant wrote other platform which shows that windows phones are now obsolete and new Operating systems like Harmony OS by Huawei are not very popular.

Only 25% chose native development solely. This means that the industry now is using a mix of cross-platform development and native development together. Almost 75% of

the survey participants are equally split between the “both” option and cross-platform when they were asked about the technology used in development. These statistics are almost similar regardless of the role of participants and regardless of the company size or location for professional participants. This shows that cross-platform frameworks proved to be a good solution but sometimes they are not enough for implementation of sophisticated functionalities on the different platforms. One of the survey participants who was using flutter framework wrote a comment stating that some functionalities must be written using native code then connected to flutter. In addition, cross-platform frameworks have performance issues regarding memory, speed, security, and user experience.

Our study also shows that Flutter is the most used right now. The numbers show that Flutter is becoming very popular among both freelancers and developers working in companies. However, the participants are aware that flutter has its own issues. The most reported issues by Flutter users were memory and speed.

Kotlin language is competing now with java for Android development. More than half of the Android developers who participated in the survey chose Kotlin as the native language they use for development. This was predicted as an impact to the announcement made by Google stating that they will use Kotlin as the official language for Android [35]. We encourage future researchers to investigate more the capabilities of Kotlin language and to support Kotlin language in their proposed development tools for mobile applications. On the other hand, swift language proved to be the most dominant for iOS development, only one participant chose objective C language. The participant who chose objective C is working for a very big and old company which means that objective C might still be used in old companies that do not want to change their structure of operation.

Regarding the familiarity of practitioners with the trans-compiler-based solutions, the results showed that practitioners are almost unaware of these solutions. They are dealing with the current cross-platform frameworks now as their current solution. However, the shift that has occurred from React Native, Phone gap, Ionic and Xamarin to flutter over the recent few years shows that whenever a better solution appears for developers and practitioners, they will easily move to it. These findings also encourage researchers to test their solutions by real practitioners and spread them to industry. This will help them introduce new solutions and tools that might close the gap between current solutions and performance issues.

VI. COMPARISON WITH RELATED WORK

To compare our work with the related works mentioned in the introduction, we focused on five main points based on our findings compared to their findings. These five main points are: What mobile platform do practitioners like to develop applications using? What development approach do practitioners prefer to use, native or cross-platform? Which cross-platform framework is the most used in the industry? What native language do practitioners use the most? What are the performance issues that are facing cross-platform mobile frameworks?

Puvalla et al. [36] discussed the factors that affect developers in choosing a development platform. They concluded that although Android has more apps than iOS, developers are switching from Android apps to iOS due to the monetization potential of the iOS apps. Therefore, iOS developers are paid more than Android developers. The findings of the survey show that the factor that affects the developer's choice of the platform the most is the availability of SDKs to support the platform. But our results show that most practitioners and companies prefer to build applications for both platforms rather than one specific platform. Our results are synonymous with the prediction made by Francesca et al. [37] in 2017, which asked a futuristic question, "In the next 5 years, on which native platform will your company actively develop?". They stated that regardless of what platform the respondents were using at that time, the target platform in the next five years would be both Android and iOS.

Regarding the question of whether practitioners prefer native or cross-platform development approaches. In 2017, Ahmed et al. findings show that most of their participants prefer using native development over cross-platform development. This is different to our findings as our result shows that only 25% are using native development alone while 37.5% are cross-platform and the remaining 37.5% chose both technologies. These differences are related to technological enhancements to the cross-platforms over the years. Besides the technical improvement, cross-platform frameworks are easier to use by developers since development community are getting bigger and giving more support to developers [23].

Bjorn-Hansen et al. [23] conducted an empirical study by targeting 101 industry practitioners. The reason is to find out which cross-platform frameworks are used the most by developers as a hobby and professionally, and what issues are facing cross-platform frameworks. Their findings show that PhoneGap, React-native, and Ionic are the most used apps both as a hobby and professionally. The findings of their research are similar to the findings of Francesca [38], which shows that PhoneGap is the most used. But the result differs from ours because in our study we found that flutter is the most used cross-platform development framework both in companies and by practitioners. This is due to the time that their survey was conducted in 2018 [23], so new technology now dominates the market. The same research group in 2022 [39] investigated approaches to Android mobile application development through applications on the Google Play store. The findings also show that Cordova is the most popular cross-platform framework. Their findings also contradict our findings. However, only 15% of their total dataset are cross-platform applications and the rest of the 85% are native applications, and the applications that are mentioned in the dataset are not recent, which is why they are not developed using flutter, one of the most recent frameworks.

In terms of the native language that the developers are using, Bjorn-Hansen et al. [23] predicted that Kotlin is the most used android development nowadays, not Java. This prediction is proven in our survey, as seen in the results section, that Kotlin is the most used android language by practitioners and companies. On the other hand, Swift is the dominating iOS programming language.

Lastly, regarding the issues that are facing cross-platform mobile frameworks, Bjorn-Hansen et al. [23] state that cross-platform solutions are still facing performance and user experience issues when compared to native solutions. Our findings suggest that memory and speed are the issues that affect Cross-platform frame works, most specifically flutter, which is the most dominant cross-platform framework in our survey.

VII. CONCLUSION AND FUTURE WORK

This paper investigates the most used development technique in the industry and to gather feedback on mobile application development. It seeks to answer the question of whether most industries use cross-platform mobile development or native mobile development. Furthermore, it investigates whether developers and industry have a perspective that is consistent with academic research by collecting data through an online questionnaire. The survey was aimed at both novice and experienced mobile developers. A total of 85 people responded to the survey, which enables us to get perspectives from the industry.

The implications that can be drawn from our findings are stated as follows: (1) Most companies and practitioners develop for both Android and iOS platform with only few choosing one platform. (2) Developers are moving towards cross-platform development. However, they are still using native development besides the cross-platform frameworks. This is due to the flaws and issues that still exist in these frameworks despite the noticeable technological enhancements. (3) Flutter is now the most used framework in the industry and the most reported issues by Flutter users were memory and speed. (4) In terms of which native language is used in the industry the most, Kotlin language currently leads the android development. However, Swift is still the main iOS programming language. (5) Most of the people in the industry are unaware of trans-compiler-based solution.

The findings presented and discussed in this article may all be significant and extensive topics for future research, both individually and collectively. This future work encourages researchers to: (1) Focus on expanding and generalizing the survey so that it will reach a large scale of people by covering more locations and adding interviews. (2) Investigate more the capabilities of Kotlin language and to support Kotlin language in their future work (3) to test their solutions by real practitioners and spread them to industry especially the Trans-compiler-based solutions. (4) To introduce a new way to decrease the issues and flaws facing cross-platform solutions.

REFERENCES

- [1] Z. A.-I. J. of A. T. and and undefined 2021, "A review of mobile applications developed by academics for COVID-19," researchgate.net, 2021, doi: 10.19101/IJATEE.2021.874058.
- [2] J. Lee and G. Gereffi, "Innovation, upgrading, and governance in cross-sectoral global value chains: The case of smartphones," *Industrial and Corporate Change*, vol. 30, no. 1, 2021, doi: 10.1093/icc/dtaa062.
- [3] B. P. D. Putranto, R. Saptoto, O. C. Jakaria, and W. Andriyani, "A Comparative Study of Java and Kotlin for Android Mobile Application Development," in 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2020, 2020. doi: 10.1109/ISRITI51436.2020.9315483.

- [4] M. Latif, Y. Lakhri, E. H. Nfaoui, and N. Es-Sbai, "Cross platform approach for mobile application development: A survey," in 2016 International Conference on Information Technology for Organizations Development (IT4OD), IEEE, Mar. 2016, pp. 1–5. doi: 10.1109/IT4OD.2016.7479278.
- [5] D. You, M. H.-Wellington, and N. Zealand, "A Comparative Study of Cross-platform Mobile Application Development," citrenz.ac.nz.
- [6] P. Nawrocki, K. Wrona, M. Marczak, and B. Sniezynski, "A Comparison of Native and Cross-Platform Frameworks for Mobile Applications," *Computer (Long Beach Calif)*, vol. 54, no. 3, pp. 18–27, Mar. 2021, doi: 10.1109/MC.2020.2983893.
- [7] L. P. Barros, F. Medeiros, E. Moraes, and A. Feitosa Júnior, "Analyzing the Performance of Apps Developed by using Cross-Platform and Native Technologies," doi: 10.18293/SEKE2020-122.
- [8] M. Caulo, R. Francese, G. S.-C. on P., and undefined 2021, "Implications on the Migration from Ionic to Android," Springer, vol. 13126 LNCS, pp. 3–19, 2021, doi: 10.1007/978-3-030-91452-3_1.
- [9] M. Isitan and M. Koklu, "Comparison and Evaluation of Cross Platform Mobile Application Development Tools," *International Journal of Applied Mathematics, Electronics and Computers*, vol. 8, no. 4, pp. 273–281, 2020, doi: 10.18100/ijamec.832673.
- [10] Bjørn-Hansen, C. Rieger, T.-M. Grønli, T. A. Majchrzak, and G. Ghinea, "An empirical investigation of performance overhead in cross-platform mobile development frameworks," *Empirical Software Engineering* 2020 25:4, vol. 25, no. 4, pp. 2997–3040, Jun. 2020, doi: 10.1007/S10664-020-09827-6.
- [11] Bjørn-Hansen, T. M. Grønli, and G. Ghinea, "A survey and taxonomy of core concepts and research challenges in cross-platform mobile development," *ACM Comput Surv*, vol. 51, no. 5, Jan. 2019, doi: 10.1145/3241739.
- [12] W. S. El-Kassas, B. A. Abdullah, A. H. Yousef, and A. M. Wahba, "Taxonomy of Cross-Platform Mobile Applications Development Approaches," *Ain Shams Engineering Journal*, vol. 8, no. 2, pp. 163–190, Jun. 2017, doi: 10.1016/J.ASEJ.2015.08.004.
- [13] Korchi, M. K. Khachouch, Y. Lakhri, and A. Moumen, "Classification of existing mobile cross-platform approaches," 2nd International Conference on Electrical, Communication and Computer Engineering, ICECCE 2020, Jun. 2020, doi: 10.1109/ICECCE49384.2020.9179222.
- [14] R. B. Hamza, D. I. Salama, M. I. Kamel, and A. H. Yousef, "TCAIOSC: Application Code Conversion," in NILES 2019 - Novel Intelligent and Leading Emerging Sciences Conference, Institute of Electrical and Electronics Engineers Inc., Oct. 2019, pp. 230–234. doi: 10.1109/NILES.2019.8909207.
- [15] D. I. Salama, R. B. Hamza, M. I. Kamel, A. A. Muhammad, and A. H. Yousef, "TCAIOSC: Trans-Compiler Based Android to iOS Converter," *Advances in Intelligent Systems and Computing*, vol. 1058, pp. 842–851, 2020, doi: 10.1007/978-3-030-31129-2_77/TABLES/5.
- [16] A. Muhammad, A. T. Mahmoud, S. S. Elkalyouby, R. B. Hamza, and A. H. Yousef, "Trans-Compiler based Mobile Applications code converter: swift to java," in 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), IEEE, Oct. 2020, pp. 247–252. doi: 10.1109/NILES50944.2020.9257928.
- [17] A. Muhammad, A. M. Soliman, S. Selim, and A. H. Yousef, "Generic Library Mapping Approach for Trans-Compilation," pp. 62–68, Jun. 2021, doi: 10.1109/MIUCC52538.2021.9447641.
- [18] S. S. El-Kaliouby, S. Selim, and A. H. Yousef, "Native Mobile Applications UI Code Conversion," *Proceedings - 2021 16th International Conference on Computer Engineering and Systems, ICCES 2021*, 2021, doi: 10.1109/ICCES54031.2021.9686093.
- [19] R. Barakat, M. B. A. Radwan, W. M. Medhat, and A. H. Yousef, "Trans-Compiler-Based Database Code Conversion Model for Native Platforms and Languages," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13761 LNAI, pp. 162–175, 2023, doi: 10.1007/978-3-031-21595-7_12/TABLES/7.
- [20] M. Usman, M. Z. Iqbal, and M. U. Khan, "A product-line model-driven engineering approach for generating feature-based mobile applications," *Journal of Systems and Software*, vol. 123, pp. 1–32, Jan. 2017, doi: 10.1016/J.JSS.2016.09.049.
- [21] H. Heitkötter, T. A. Majchrzak, and H. Kuchen, "Cross-platform model-driven development of mobile applications with MD 2," in *Proceedings of the ACM Symposium on Applied Computing*, New York, New York, USA: ACM Press, 2013, pp. 526–533. doi: 10.1145/2480362.2480464.
- [22] T. Zohud and S. Zein, "A systematic mapping study of cross-platform mobile apps," *Journal of Computer Science*, vol. 15, no. 4. Science Publications, pp. 519–536, 2019. doi: 10.3844/jcsp.2019.519.536.
- [23] Bjørn-Hansen, T. M. Grønli, G. Ghinea, and S. Alounh, "An Empirical Study of Cross-Platform Mobile Development in Industry," *Wirel Commun Mob Comput*, vol. 2019, 2019, doi: 10.1155/2019/5743892.
- [24] Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, and S. Ge, "An Empirical Study of Investigating Mobile Applications Development Challenges," *IEEE Access*, vol. 6, pp. 17711–17728, 2018, doi: 10.1109/ACCESS.2018.2818724.
- [25] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, "Mobile App Development and Management: Results from a Qualitative Investigation," *Proceedings - 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2017*, pp. 133–143, 2017, doi: 10.1109/MOBILESoft.2017.33.
- [26] Puvvala, A. Dutta, R. Roy, and P. Seetharaman, "Mobile application developers' platform choice model," *Proceedings of the Annual Hawaii International Conference on System Sciences*, vol. 2016-March, pp. 5721–5730, 2016, doi: 10.1109/HICSS.2016.707.
- [27] "AndroZoo: Collecting Millions of Android Apps for the Research Community | IEEE Conference Publication | IEEE Xplore." <https://ieeexplore.ieee.org/document/7832927> (accessed Oct. 11, 2022).
- [28] R. Malhotra, "Empirical research in software engineering: concepts, analysis, and applications," p. 472.
- [29] J. S. Molléri, K. Petersen, and E. Mendes, "Survey Guidelines in Software Engineering: An Annotated Review," *International Symposium on Empirical Software Engineering and Measurement*, vol. 08-09-September-2016, Sep. 2016, doi: 10.1145/2961111.2962619.
- [30] W. Hui, S. M. Lui, and W. K. Lau, "A reporting guideline for IS survey research," *Decis Support Syst*, vol. 126, p. 113136, Nov. 2019, doi: 10.1016/J.DSS.2019.113136.
- [31] "About Pollfish." <https://www.pollfish.com/about/> (accessed Feb. 02, 2022).
- [32] "Real Consumer Insights | Pollfish Survey Tools." https://www.pollfish.com/?utm_source=google&utm_medium=cpc&utm_campaign=gsem_Brand_sc-Ret&utm_adgroup1=pollfish&utm_term=pollfish_survey&utm_sitelink=%7Bstielink%7D&utm_device=c&utm_create=449133244745&gclid=Cj0KCQjw4PKTBhD8ARIsAHChzRjL1d2Jxcx71KUx8T2ogaHzKiMchLP7Ei-1gClI4Jy85MTS-oy_vjYaAnF5EALw_wcB (accessed May 12, 2022).
- [33] "Cross-platform mobile frameworks used by global developers 2021 | Statista." <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/> (accessed May 12, 2022).
- [34] Ahmad, K. Li, C. Feng, S. M. Asim, A. Yousif, and S. Ge, "An Empirical Study of Investigating Mobile Applications Development Challenges," *IEEE Access*, vol. 6, pp. 17711–17728, Mar. 2018, doi: 10.1109/ACCESS.2018.2818724.
- [35] Mazuera-Rozo et al., "Taxonomy of security weaknesses in Java and Kotlin Android apps," *Journal of Systems and Software*, vol. 187, p. 111233, 2022, doi: 10.1016/j.jss.2022.111233.
- [36] Puvvala, A. Dutta, R. Roy, and P. Seetharaman, "Mobile application developers' platform choice model," *Proceedings of the Annual Hawaii International Conference on System Sciences*, vol. 2016-March, pp. 5721–5730, 2016, doi: 10.1109/HICSS.2016.707.
- [37] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, "Mobile App Development and Management: Results from a Qualitative Investigation," *Proceedings - 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2017*, pp. 133–143, 2017, doi: 10.1109/MOBILESoft.2017.33.

- [38] R. Francese, C. Gravino, M. Risi, G. Scanniello, and G. Tortora, "Mobile App Development and Management: Results from a Qualitative Investigation," Proceedings - 2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems, MOBILESoft 2017, pp. 133–143, 2017, doi: 10.1109/MOBILESoft.2017.33.
- [39] Biørn-Hansen, T.-M. Grønli, T. A. Majchrzak, H. Kaindl, and G. Ghinea, "The Use of Cross-Platform Frameworks for Google Play Store Apps," Proceedings of the 55th Hawaii International Conference on System Sciences, Jan. 2022, doi: 10.24251/HICSS.2022.934.