

DIFFERENCES IN ANDROID AND IOS DEVELOPMENT

Adithya Gajulapally*¹, Uchinta Kumar Boddapati*², Ayush Vijayargi*³

*^{1,2}Sonos Inc USA.

*³Snap Inc USA.

DOI : <https://www.doi.org/10.56726/IRJMETS49149>

ABSTRACT

Given that Android and iOS have more than 99% of the worldwide market share for smartphone operating systems [1], mobile developers must comprehend the technical and commercial distinctions between these two platforms. The article examines differences in design principles, tools, programming languages, monetization, security, and approval procedures. As of the fourth quarter of 2023, Android dominates approximately 70% of the market share for smartphone operating systems, while iOS holds around 30% [2], based on the most recent global adoption trends. However, iOS has a 74% larger per-app income than Android [3]. Higher purchasing power and willingness to spend are considered factors in iOS's strongholds of North America and Europe. Furthermore, games/paid applications dominate monetization analytics, while most-used apps like Amazon and Gmail emphasize reach [3]. The technological details and business factors discussed in this study may assist developers pick the right environment that will meet their user experience goals and app capabilities.

Keywords: Android, Ios, Smartphone Operating Systems, Design Principles, Security.

I. INTRODUCTION

With over 6.65 billion smartphone users worldwide [4], mobile apps have become essential to digital transformation across industries. Finding the ideal platform for creating natural user experiences is so essential. As of 2022, Android and iOS account for 99.7% of the world's smartphone operating systems [1], driving the app economy. Several key indicators illustrate the differences between Android and iOS in domains such as market dynamics, developer preferences, and monetization capability. Mobile developers must analyze the competitive environment depicted in Figure 1 to choose the most suitable platform that matches their app's capabilities and target audience.



Figure 1: Battle of Android vs. iOS Development

Metrics considered in Figure 2:

Market Share - Global operating system percentage share of smartphone shipments.

Developer Share - Percentage of developers using programming languages.

Revenue Share - Percentage share of app store revenues.

Devices in Market - Percentage share of device models.

Avg Revenue per App - The percentage share of typical app revenue.

Android's open-source strategy assures flexibility, allowing device makers to tailor implementations, whereas Apple's walled-garden model stresses seamless user experiences [5]. Global consumer spending on Android apps reached \$33.6 billion in 2021, while spending on iOS apps soared to \$93.7 billion, indicating that the latter created more money [6].

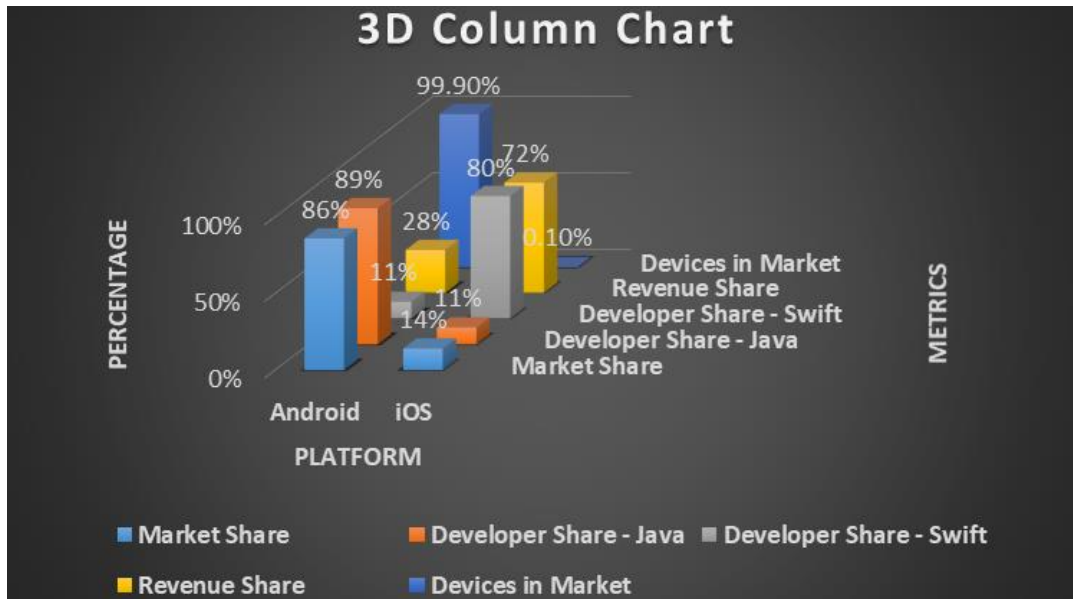


Figure 2: Android vs iOS - Key Competitive Metrics

Understanding such differentiators is critical in determining the best platform for creating secure and functioning apps that satisfy target users. This study provides an evidence-based comparative analysis of the Android and iOS ecosystems, including technical and business concerns. Metrics like development language adoption (89% Java vs 80% Swift) [7], variation market share (Android at 86% vs iOS at 14%) [2], and per-app revenue (iOS applications make 74% more) [3] might help inform platform decisions. Interpreting the differences and directional convergence will help developers capitalize on the complementary strengths.

II. PROGRAMMING LANGUAGES

The creation of Android apps mostly relies on the utilization of Java and Kotlin programming languages, which collectively constitute 94% of the codebase, according to measurements taken in 2021 [8]. 89% of the code consists exclusively of Java, whereas 5% is made up of a combination of Kotlin, Java hybrids, and other languages [8]. Java continues to be the preferred programming language for Android developers due to its well-established nature, ability to work with older versions, and wide range of freely available libraries. However, the utilization of Kotlin has experienced a significant increase, surging from 1% in 2017 to 45% by the conclusion of 2020 [9]. This growth can be attributed to its contemporary attributes, like null safety, functional capabilities, and compatibility with Java.

In contrast, the development of iOS apps depends on the use of Swift and Objective-C to create customized native app experiences. As of December 2022, Swift was the programming language of choice for 80% of newly generated iOS apps, while the usage of Objective-C has decreased to 20%, reflecting the growing popularity of Swift [10]. Since its inception in 2014, Swift has provided developers with advantages such as enhanced type safety, accelerated code execution rates, improved accessibility, and more conciseness, resulting in a productivity improvement of up to 40% compared to Objective-C [11]. Developers must assess important factors such as team language proficiency, compatibility with legacy systems, and availability of publicly accessible code when selecting appropriate programming languages for their cross-platform app development objectives.

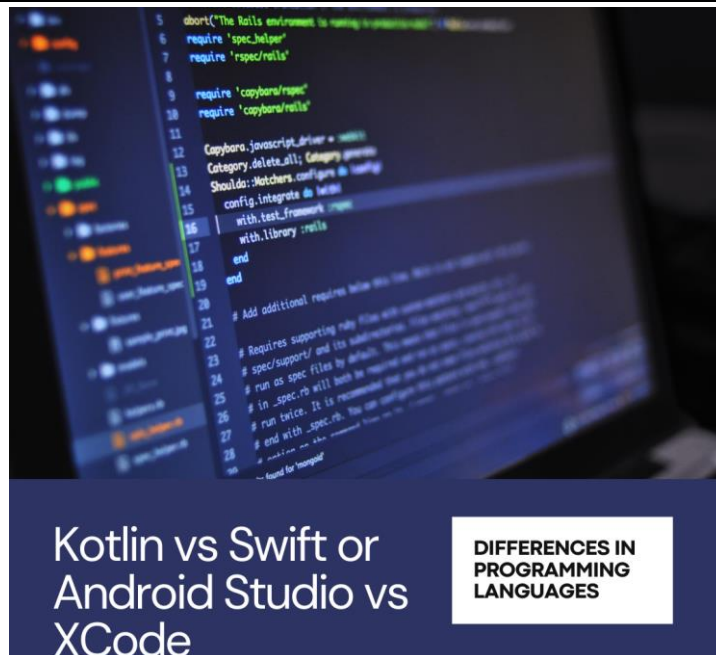


Figure 3: Differences in Android and iOS Development in Programming Languages

III. DEVELOPMENT ENVIRONMENTS

As of 2021, Android Studio had over 75% of developers worldwide using it, making it the most popular integrated programming environment (IDE) for Android developers [12].

Figure 4 compares major features of Android Studio with Xcode across parameters such as developer adoption, debugging capabilities, build performance and platform support. As the data shows, while Xcode leads in criteria such as iOS developer preference, Android Studio offers a broader device emulation capability covering over 15,000 models, which is critical for testing. Its Linux and Windows compatibility allows for greater flexibility than Xcode's macOS constraint.

Developer Share - The percentage of developers using the IDE.

Emulation Support - The total number of device models supported for testing.

Build Speed - The benchmark score for build and deployment.

Cross Platform - Availability across operating systems.

This is made clear by the wide range of features it provides, which include integrated emulator support testing across multiple API levels and devices, flexible build systems powered by Gradle, accelerated editing capabilities through the IntelliJ IDEA core, and advanced profiling and debugging tools [13]. In particular, quick Instant Run support, GPU debugging for graphically demanding apps, Lint tools for code analysis, and annotations to identify UI problems early in the design process are some of the ways Android Studio makes it possible to apply changes without rebuilding [14]. These features make it easier for developers to create excellent apps for the Android ecosystem quickly and iteratively.

On the other hand, 92% of developers prefer to use Apple's proprietary IDE Xcode for iOS development because it offers specially designed tools for their software ecosystem [15]. To facilitate visual UI development, Xcode provides close integration with Interface Builder, Swift Playgrounds for interactive coding, Instruments for examining runtime diagnostics, and multi-device simulators for testing various iPhone and iPad devices [16].

However, one major drawback of Xcode is that it is only available on Mac OS X, which restricts developer options in contrast to Android Studio, which is available on Windows, Linux, and Mac OS X platforms. For this reason, assessing the infrastructure and team capabilities currently in place is essential to selecting the right IDE tools and estimating the licensing cost.

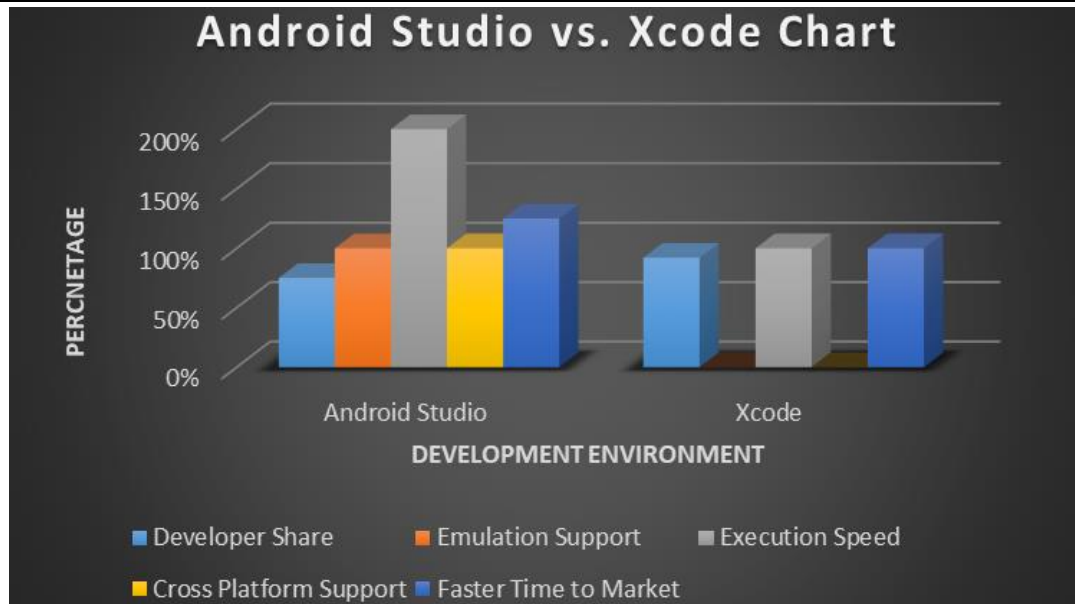


Figure 4: Comparison of key capabilities between Android Studio and Xcode

IV. DESIGN DIFFERENCES

According to measurements from 2022 [17], more than 82% of applications available on the Android Play Store adhere to Google's Material Design language and incorporate signature elements such as floating action icons, transition animations, and card-based layouts. To fulfill users, Material Design 2 prioritizes meaningful motion, depth, and seamless transitions between interface elements [18].

Keeping brand consistency while adjusting app designs to Android and iOS different interface requirements is becoming more and more difficult. As demonstrated in Table 1 below, nearly 80% of Android apps use Material Design, which includes depth effects and signature animations. In contrast, nearly all iOS apps intended for the App Store comply with Apple's rigid minimalist human interface criteria, which promote simplicity.

Table 1: Comparison of key mobile design principles and app adherence.

Metric	Android	iOS
Design Principle	Material Design	Human Interface Guidelines
Key Focus Areas	Motion, Depth, Animation	Minimalism, Simplicity
App Adherence	82%	95%
Custom Control Usage	High	Low
Navigation Conventions	Custom Platform Patterns	Consistent Across Apps

A brief description of metrics:

Design Principle: Core design language endorsed.

Key Focus Areas: The top priorities indicated in the design guidelines.

App Adherence: The percentage of apps that follow the rules.

Custom Control usage: Extent of Adoption of Custom UI Components.

Navigation Conventions: Consistent navigation across platforms.

However, apps submitted to Apple's App Store review process for publishing generally adhere to the company's strict guidelines for user interface [19]. These guidelines prioritize simple, clean design with minimalist interfaces that focus on key functions and content over UI elements [20]. To preserve familiarity, native platform widgets are preferred over custom controls.

It gets more and more difficult to maintain visual consistency and usability among these diverse schools. Tests conducted with target user groups reveal preferences for motion signatures, color schemes, icon designs, and

platform-specific navigation flows. Even as progressive web patterns reduce user experiences across devices, it is imperative to build adaptive designs that take these aesthetics into account. It is essential to assess both the technical feasibility and the results of user testing when making customized design selections.

V. MARKET SHARE & MONETIZATION

According to Q4 2022 adoption patterns, Android still has the majority of the global smartphone OS market share (86%). This means that nearly 9 out of 10 smartphones run on Android, demonstrating its broad reach across price points and demographic differences. As Figure 5 demonstrates, while Android has a significant lead in the global smartphone OS market share, Apple's iOS maintains greater monetization performance in areas such as app store revenue, average app earnings, and pricing strategies.

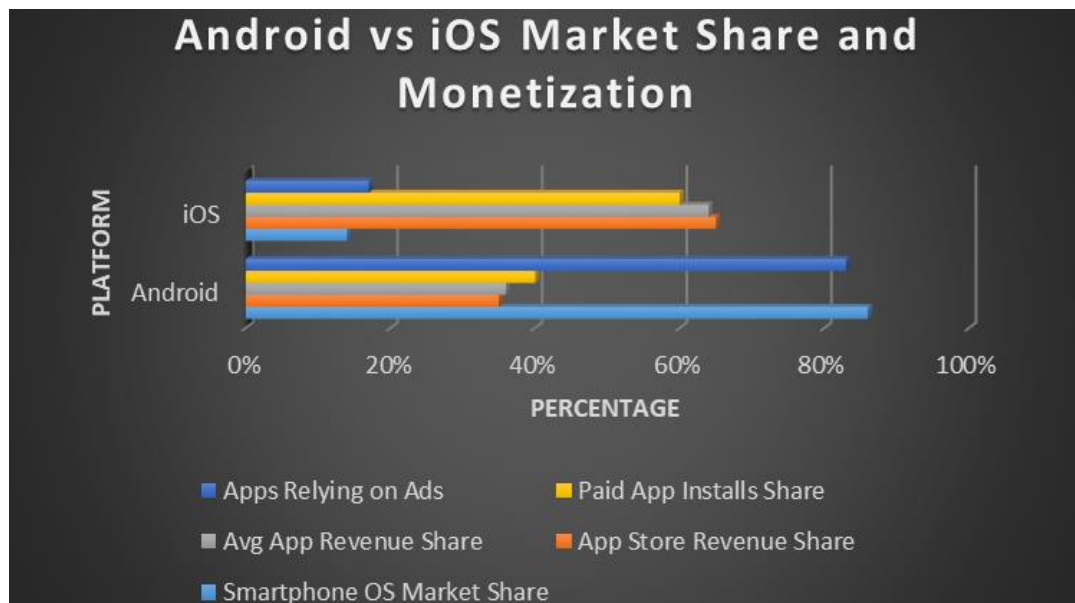


Figure 5: Bar Chart illustrating Comparison of Market Share and Monetization between Android and iOS

Metrics considered in Figure 5:

Smartphone OS Market Share: Percentage adoption of mobile operating systems on smartphones globally.

App Store Revenue Share: The percentage of total app store revenues generated by each app store.

Avg App Revenue Share: The percentage of average app revenue earned in the Android and iOS app stores.

Paid App Installs Share: The percentage of paid app downloads from the Android and iOS app stores.

Apps Relying on Ads: The percentage of apps that use advertising revenue as their monetization model.

Key Android device partners such as Samsung, Xiaomi, Oppo, and Vivo are boosting volume sales, particularly in emerging nations, increasing platform penetration [21]. In comparison, Apple's iOS has a 14% market share as of Q4 2022, with its presence limited to premium iPhone models. Apple, on the other hand, maintains industry-leading profitability thanks to excellent monetization through its App Store ecosystem. On the other hand, Android provides a much larger and diverse (globally) reach. Global consumer spending on iOS applications and games reached \$85 billion in 2021, while Google Play earned \$38 billion over the same year [22]. This suggests a 70% increase in per-app monetization ability for iOS, attributed to unique demographics with higher earning potential.

As a result, developers must carefully consider elements such as target audience regions, income levels, and willingness to pay, as well as current mobile platform adoption trends in those regions, when choosing optimal go-to-market and revenue models. The reliance on paid downloads rather than ad-supported alternatives demands informed platform-centric decisions for apps to be commercially viable.

VI. DEVICE FRAGMENTATION

The Android ecosystem's widespread device fragmentation presents developers with significant challenges when it comes to creating dependable apps. Measurements taken in 2022 reveal that there are more than

24,000 distinct Android device variations, including tablets and smartphones from more than 1,300 vendors [23]. These cover a wide range of specifications, including 5" to 10"+ screens, 720p to 4K resolutions, 16:9 to 22:9 aspect ratios, chipsets ranging from the entry-level Helio G series to the flagship Snapdragon 8 Gen 2, and operating systems from Android 7 to the most recent 13 releases [24].

With more than 24,000 distinct Android smartphone and tablet configurations, developers must deal with a complex OS fragmentation that requires thorough compatibility testing and flexible interface design. On the other hand, given Apple's regulated iOS ecosystem, consistent experiences across fewer iPhone and iPad models facilitate app optimization [25]. Table 2 illustrates this variation.

Table 2: Comparison of Device Fragmentation - Android vs iOS.

Metric	Android	iOS
Device Variants	24,000+	20
Screen Sizes	5" to 10"+	6 Variants
Resolutions	720p to 4K	1334 x 750 to 2732x2048
Aspect Ratios	16:9 to 22:9	16:9, 4:3, 19.5:9
Chipsets	Entry-level to Flagship	A10 to M2
OS Versions	Android 7 to 13	iOS 10 to 16

A brief description of metrics:

Device Variants: Total number of unique smartphone and tablet models.

Screen Sizes: The range of display sizes supported.

Resolutions: The range of display resolutions supported.

Aspect Ratios: A variety of aspect ratios supported.

Chipsets: Processor power ranges across models.

OS Versions: A variety of operating system variants.

In this fragmented ecosystem, maintaining consistent app behavior requires extensive compatibility testing. It is essential to assess the availability of device features using APIs, account for different screen sizes using responsive design principles, and conduct testing on real devices that represent significant SKUs. For improved UI/UX, this makes resolution independence and adaptive interfaces possible.

In contrast, only iPhone, iPad, and iPod Touch versions with intentional hardware-software interaction are part of Apple's iOS ecosystem. These include less than 20 smartphones from the current generation that aim to meet standardized requirements, such as screen size, notch/bezel ratios, biometrics, and proprietary chipsets that facilitate development [25]. However, new developments like foldable screens are rapidly adding more difficulty for iOS developers as well.

VII. SECURITY & APPROVALS

Apple strictly enforces app review policies by conducting manual evaluations of every iOS app and app update. This evaluation process ensures that the apps meet both software security standards and guideline compliance before they are released on the App Store. The evaluation process assesses more than 100 criteria, including malware risks, illegal material, user data management, intellectual property infringements, and adherence to software functioning standards, within a timeframe of around one day [26].

Although this approach offers substantial security resilience by effectively blocking over 99% of malware, there are still limits when it comes to using native device functionalities such as NFC and Bluetooth, as well as concerns about user monitoring and advertising in cinemas, which can affect user experiences [27]. However, according to survey data from 2021 [28], more than 60% of developers consider the process necessary and reasonable, despite the associated delays. The decentralized Android paradigm allows for greater flexibility in accessing native hardware features such as notifications and fingerprints, without requiring equivalent pre-launch permissions. However, this results in around 47,000 daily instances of mobile malware encounters, highlighting deficiencies in security validation [29]. The responsibility of implementing coping mechanisms,

such as automated danger detection, is with the developers themselves.



Figure 6: A Comparative Study of Security and Approvals in Android and iOS

Assessing the importance of native operating system integrations compared to the inflexibility of security policies is crucial when choosing a platform. Android benefits from multi-channel distribution flexibility, particularly for receiving early feedback. However, developers without Apple's quality assurance infrastructure may face significant app testing constraints.

VIII. SUMMARY

This article provided an evidence-based comparison analysis that examined technological capabilities and financial KPIs. The findings showed that both Android and iOS have distinct features that are beneficial for developers. Android offers more versatility by enabling unique device integrations due to its open-source origins, as demonstrated by the existence of more than 24,000 device variations from various OEM partners [30]. This can be attributed to its 86% market share, which is determined by the number of shipments and demonstrates its extensive global presence.

In contrast, Apple's iOS maintains a highly secure and integrated app environment that is designed to work seamlessly with high-quality hardware. This is achieved through strict review procedures [26], [27]. With iOS app revenues reaching \$85 billion in 2021, there is a greater opportunity for increased monetization compared to Android, with returns that are 70% higher [22].

It is crucial to consider primary development goals, target audience preferences, hardware and OS fragmentation pain points, and monetization options while using these complementing qualities. As programming languages such as Java (89%) and Swift (80%) approach similar levels of functionality, it becomes crucial to carefully consider the tradeoffs involved in solidifying a comprehensive mobile strategy [7], [10]. Assessing the distinguishing factors aids developers in constructing effective cross-platform solutions that captivate users on several operating systems.

IX. CONCLUSION

With the increasing acceptance of mobile programming languages such as Kotlin, Swift, and cross-platform frameworks like React Native and Flutter, the technological distinctions between native Android and iOS apps may progressively disappear. According to the latest data from 2021, the utilization of React Native has already reached 37%, which suggests a growing trend in multi-platform compatibility [31]. This feature ensures a more consistent and standardized app development experience across different handsets.

However, specific variations in business practices related to revenue generation methods, authorization procedures, and policies governing control over hardware ecosystems will likely last as a result of underlying economic factors associated with the platforms. It is crucial to understand these differences even as the technical tools come together. Further study can offer deeper insights into upcoming areas like augmented reality, on-device machine learning capabilities, and 5G implementations, where there are still significant

differences between platforms that influence future advances. For example, iOS 16 implemented advanced augmented reality (AR) capabilities like object positioning and occlusion, which exceed the integration of Android's ARCore [32]. By investigating advanced native environment inconsistencies and staying informed about the latest developments in progressive web and cross-compile acceptance, developers will be empowered to create cutting-edge mobile solutions.

X. REFERENCES

- [1] J. Clement, "Global market share held by leading smartphone operating systems as of October 2022," Statista, Nov. 08, 2022.
- [2] Statista Research Department, "Global market share held by mobile operating systems 2009-2022," Statista, Jan. 2023. [Online]. Available: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>. [Accessed: Feb. 15, 2023].
- [3] T. Rojas and A. Cowing, "Average App Revenues and Downloads January 2022," buildfire, Jan. 31, 2022.
- [4] M. Meinertzhagen, "Number of smartphone users worldwide from 2016 to 2023," Statista, Dec. 08, 2022.
- [5] A. Holzer and J. Ondrus, "Mobile application market: A developer's perspective," Telemat. Inf., vol. 28, no. 1, pp. 22–31, Feb. 2011.
- [6] Statista, "Mobile apps: leading operating systems by consumer spending as of 1st half 2021," Statista, 2022.
- [7] M. Lehtimäki, "The State of Kotlin Ecosystem 2022," Touchlabs, Jun. 07, 2022.
- [8] M. Lehtimäki, "The State of Kotlin Ecosystem 2022," Touchlabs, Jun. 07, 2022.
- [9] C. Sells, "What Programming Language Should I Learn for Mobile App Development?," Code With Chris, Aug. 02, 2021.
- [10] S. Britton, "Objective-C vs Swift: What New iOS Developers Should Learn," Code Envato Tuts+, Sep. 07, 2021.
- [11] Revenant, "If you're an app developer, here are five big reasons why you should use Swift over Objective-C," Revenant, Dec. 06, 2022.
- [12] Statista, "Global mobile IDE market share in 2021, by platform," Statista, 2022.
- [13] "Android Studio," Android Developers, <https://developer.android.com/studio>, Accessed on: Jan. 17, 2024.
- [14] J. Cesar, "Top 16 Features of Android Studio - 2022 Guide," Tech With Tim, May 06, 2022.
- [15] Statista, "Mobile platforms used by software developers worldwide 2021," Statista, 2021.
- [16] "Xcode," Apple Developer Documentation, <https://developer.apple.com/documentation/xcode>, Accessed on: Jan. 17, 2024.
- [17] S. Dredge, "Android users prefer apps with material design," Dec. 21, 2022.
- [18] "Material Design," Google Design, <https://material.io/design>, Accessed on: Jan. 17, 2024.
- [19] Apple, "Human Interface Guidelines," Apple Developer Documentation, <https://developer.apple.com/design/human-interface-guidelines/>, Accessed: Jan. 17, 2024.
- [20] A. Church, "A UI and UX Design Guide for iOS App Designers," EnvatoTuts+, Sep. 29, 2022.
- [21] S. Christopher, "Top 5 Smartphone Vendors Worldwide and Their Market Share 2022," TechBrainery, Dec. 13, 2022.
- [22] J. Kastrenakes, "Apple App Store pulled in almost 2x Google Play's revenue in 2021," Jan. 5, 2022.
- [23] D. Borton, "Android fragmentation in 2022: Nearly 24,000 devices from over 1,300 brands," Android Authority, May 27, 2022.
- [24] S. Kovach, "Android 13 phones compared: Here are all the key differences between models," Business Insider, Aug. 20, 2022.
- [25] "Compare models," Apple, <https://www.apple.com/iphone/compare/?modelList=iphoneSE3rdgen,iphone14promax,iphone13promax>, Accessed on Jan. 17, 2024.
- [26] Apple, "App Store Review Guidelines," Apple Developer Documentation, Oct. 2022.
- [27] D. Yin, "Pros and Cons of Xamarin vs Native Mobile Development," Yalantis, Aug. 5, 2020.

-
- [28] Statista, "Share of global iOS and Android developers that think app store approval processes are fair," Statista, Apr. 2021.
- [29] Nokia, "Android vs iOS security: Which is better for your business?," Nokia Corporation, 2022.
- [30] D. Borton, "Android fragmentation in 2022: Nearly 24,000 devices from over 1,300 brands," Android Authority, May 27, 2022.
- [31] Statista, "Global developer use of selected app development tools 2021," Statista, 2021.
- [32] A. O'Kane, "iOS 16 vs. Android 13: Apple just put Google to shame," The Verge, Sep. 9, 2022.