IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A Comparative Study of Flutter with other Cross-Platform Mobile Application Development.

[1]Rutuja Kurale, [2]Kumkum Bala

[1]Student, [2]A Assistant Professor

[1]Department Of Computer Engineering,[1]Bharati Vidyapeeth's College of Engineering, Lavale, Pune, India

Abstract: As a Growing demand in Mobile app development ends up in a rise in alternatives for developing Mobile Apps. There are many approaches for constructing mobile apps like Native apps, Cross-platform apps, Hybrid apps, and Progressive Web app Development. In today's world, the easy and fast way to App development is that the thing all developers are trying to find and flutter has come up with the desired platform to support the event of applications for both Android and iOS. One of the key benefits cross-platform mobile application development frameworks provide is a means of writing applications once and deploying them on multiple platforms. Now, there are multiple cross-platform mobile application frameworks in the market. Although comparing each one of them is not the goal of this study, that is why we prefer flutter over other platforms available in the market. Flutter is an open-source SDK that mainly focuses on creating high-performance, high-fidelity mobile apps for iOS and Android. This study performs to point the advantages of developing an app with Flutter over other cross-platform development platforms.

Keywords: iOS, Android, Hybrid apps, native apps, cross-platform development, SDK, Flutter

I. Introduction

In Today's era , Every individual uses smartphones irrespective of its operating system. As per Market share of mobile operating systems worldwide 2012-2021(Published by S. O'Dea, Jun 29, 2021), Google's Android and Apple's iOS jointly possess over 99 percent of the worldwide market share. While Developing any app developer can't ignore any of the mobile operating systems, in order that developer should develop multiple apps for multiple operating systems. These two platforms share immense dissimilarities which often necessitate different skillsets for development.

II. Ease of Use

Cross-platform development comes here to resolve this issue. The most natural way of developing an application for a particular platform would be by using the platform's native technologies. Besides providing flawless feels and appearance, the applying runs faster. However, because the number and sort of mobile platforms grow, targeting several platforms using native technologies would want platform-specific competence, and also the development process would be costly. Now, there are multiple cross-platform mobile application frameworks in the market. Although comparing each one of them is not the goal of this study, in that why we prefer flutter over other platforms available in the market.

III. Mobile Application Development Types

1) Native Applications: Native mobile app development involves developing an app for a selected targeted platform, can run on only that platform, that's coded in an exceedingly specific programming language, like Objective C for iOS or Java for Android operating systems, and users access them from dedicated app stores (such because the App Store or Google Play). Native mobile apps provide quick performance and a high degree of reliability. Both Apple and Google provide their development tools, interface elements, and SDK. Most companies will invest in native mobile app development because of the myriad of benefits offered as compared to other varieties of apps like Hybrid or Web.

2) Web Applications : Web applications include applications accessed via the internet over a network and developed using browser-supported languages (e.g.HTML, CSS, JavaScript, TypeScript). Therefore shortly, any website that includes a client-side component that permits users to perform an operation is called a web app. Most web apps work online. Examples include Google Maps, Google Sheets, Uber, etc., web apps function only when you have internet connectivity. A web app uses in real-time in a web browser with CPU resources. Web application development is that the process of using client-side and server-side programming to develop an application obtainable over a web browser. The frameworks like tools and libraries that developers use to make developing in a particular language easier and more efficient for web app development.

Angular is a front-end framework supported by Google. It's the most perfect solution as it has the most bundled functionalities, Compared to other front-end frameworks.Node.js is a runtime that enables you to write server-side code using JavaScript. As JavaScript is much popular, Node.js is also broadly used and there are many good developers working with Node.js. It's usually used for real-time web applications.

3) Hybrid Applications: The defining characteristic of hybrid apps is that it combines native parts along with conventional Internet technologies such as HTML, CSS, and JavaScript. Loosely speaking, it is a cross between native apps and web apps. Because of their less development timeframe, lower cost and ability to be scaled to a distinction of platforms and operating systems Hybrid apps become so popular. Most preferred by developers, When developers want to develop an app for various platforms at a time. The centre of a hybrid-mobile application is just an app that's written by JavaScript, HTML, CSS. In 2017, Google introduced the concept of Progressive Web Applications (PWAs), which permit these varieties of applications to adopt more app-like features like standard app icons, push notifications, offline capabilities, and more.Here are some key point Differences in different App development platforms:

Feature	**Native Apps**	**Web Apps**	**Hybrid Apps**
Code Reusability	No	Yes	Yes
Device Access	Full	Limited	Full
Cross-platform support	No	Yes	Yes
Performance	Medium	Medium to High	High
Speed of App	High	Medium	Medium
Distribute through App Store	Yes	No	Yes
UX	High	Medium	Medium
Development Time & Cost	High	Low	Low

Fig: Comparisons in different App development platform.

Through the above chart, we will conclude that hybrid solutions offer a balance between the flexibleness of web apps, and also the functionality of native apps, without forgetting the flexibility to figure across multiple device types and platforms, whilst also leveraging the capabilities of the mobile device hardware.

IV. What is Cross-platform App Development?

Cross-platform app development has become an outstanding alternative to traditional, native development. There are many causes behind its growing popularity. Firstly, it permits you to launch your app on various platforms concurrently. This means you can reach a broader audience than in the case of building an app for a particular platform. Secondly, it allows a faster time-to-market, as the code doesn't need to be written from scratch separately for web, iOS, and Android. At lastly, this interprets to well-known savings in the development and maintenance phase, that why you need one development team to include all platforms. Cross-platform software may run on multiple platforms or as few as two. Some frameworks for cross-platform development are Xamarin, Phonegap, Ionic, and React Native, Codename One, Kivy, Qt, Flutter, NativeScript.

V. Types of Cross-Platform Application Development

1) React Native : React Native is an open-source framework by Facebook for building cross-platform applications that support different mobile platforms or devices. If you had a well-known knowledge of JavaScript, you can easily work on React Native which is based on ReactJS, a JavaScript library for building user interfaces. It uses JavaScript XML (JSX).

It is also extremely cost-effective with optimal performance. React Native uses native components from the API compared to other frameworks. It implies that it relates to a native app for that particular platform. React has since matured with the contributions from different organizations and developers' communities to be able to support mobile application development using React Native. By using React Native one can build a mobile application that has native feels and views, and access platform-specific UI elements from JavaScript code. Unlike hybrid mobile applications, which mainly use WebView to package application code into platform-specific code, React Native compiles JavaScript application code into platform-specific executable code that may utilize native APIs and UI components. The component is the main building block in a React Native application. The component is that the declarative definition of some views within the UI. React Native is suitable for testing and reutilization of code because of its modular components. The most popular apps like Facebook, Instagram, Pinterest, Skype, UberEats are built with React Native development.

Benefits -

1) Increase development productivity and faster time to market.

2) Hot Reload - This is the most unique feature of React Native is Hot Reload. It has a "Hot Reload" option at the moment you save changes that you made in your code at that moment you can see the output. It supports the developer to refresh the app during the development stage itself. This speeds up the development of applications instead of the regular compilation process. It also helps in live tracking all the changes made new by the developer.

3) Great community support - React native allows for third-party plugins. The framework allows developers to combine third-party plugins. The platform is very adaptable with customization offerings.

4) First-class Typescript support - TS has matured significantly and is analogous to the type systems in Swift/Java.

5) Developer-friendly - Developers admire React's clear component lifecycle and declarative UI. If one encounters any problems related to React Native development the community experts will provide help easily. Developers also can share their experiences and create portfolios of their work which helps them to improve and write better codes.

6) Open source - Developers can contribute fixes and fork React Native, and even port to new platforms.

7) Code reusability - This is probably the most important benefit of React Native. The developer can write a single codebase that can be reused between different mobile platforms. The 90 percent of code written in this framework is usually reused.

2) Xamarin : Xamarin is a famous cross-platform development framework applied by developers to develop native-like and performant apps. It came into use in 2011 and was later obtained by Microsoft in 2016. Xamarin is an Open source, Free, Cross-platform App Development Framework of Microsoft for developing Android and iOS apps with C# and .Net. It permits one to create apps that easily run across multiple platforms. Xamarin s famous for providing strong back-end support for native app-like experiences in mobile apps. Xamarin seems like an abstraction layer that supports the communication of shared code for the platforms.

Benefits –

1)Faster development: It uses a single tech stack and shareable codebase that reduces the development time. Developers need to make only small changes to release apps across various platforms.

2)Native user experience: Xamarin supports native APIs and toolkits that provide native-like app design and performance.

3) Unique technological stack: It creates apps for multiple mobile platforms using one language that doesn't need switching between environments since everything can be done in Visual Studio.

4)Cost-Effective: Develop, test, and deploy apps for different mobile platforms without employing multiple teams. Testing and deployment can also be controlled by the same team, adding more flexibility to the budget.

4) Ease in maintenance: Update changes within the source file and that they reflect accordingly across the various apps.

3) Flutter - A "tool" that permits you to build native cross-platform (iOS, Android) apps with one programming language and codebase. Flutter is originally developed by Google that it is a free, open-source framework. Flutter is a cross-platform software development kit(SDK) to build mobile applications for important app development platforms like iOS, Linux, Mac, Windows, and Android. Although flutter is a new tool or framework as it was released in May 2017, it become hybrid or cross-platform developer's first choice because of its ease in development and deployment of applications and wide range of UI toolkits for building not only native likes app but also for beautiful UI based applications. Dart language which is object oriented provides various benefits and is based on C/C++ and Java, also a product of Google used to code Flutter apps.

Flutter app development has ready-made widgets to create applications. Flutter's widget technique permits for a consistent object model and easy-to-use developing process compared to other common approaches like layouts, views, or controllers. Amongst the rich set of widgets, Flutter gives are the Material Design library and Cupertino widgets for developing UIs and intense motion APIs. These widgets provide available and customizable functionalities for building up native user interfaces for platforms like iOS.

Benefits -

1) Like React Native, Flutter also provides hot reload, the ability to keep the app running and fast reflect any code changes edited at runtime, without losing state on the emulators or any hardware for iOS and Android.

2) code reusability: Developers write just one codebase for multiple platforms, including both Android and iOS platforms. Flutter can run on multiple platforms with one codebase.

3) Fast Development: It uses a single tech stack and shareable codebase that reduces the development time. Developers need to make only small changes to release apps across various platforms, because of a strong set of fully customizable widgets to develop native-like interfaces in few moments.

4) Powerful and Flexible UI: The layered architecture provides full customization, which results in amazingly fast rendering and strong and flexible designs that give us a focus on native end-user experiences.

5) Native Performance: When it comes to performance, Flutter tops as it is much quicker than React Native. Flutter's widgets incorporate all-important platform changes like scrolling, navigation, icons, and fonts, and are compiled to native ARM machine language using Dart's native compilers.

4)Ionic - Ionic Framework is that the latest open-source mobile app development framework (SDK), used for building cross-platform mobile apps with JavaScript, CSS, HTML5, and Angular. Ionic practices standardized web technologies like JavaScript, HTML, and CSS. Those common with web development can create apps and fix them in an app store. These hybrid apps have many advantages like platform support, indelible development, and integration of 3rd party plugins.

Benefits -

1)Optimize Performance: Ionic mobile app development is basically performance-oriented. It provides pre-processors, essential tools, and services for distributing code and other options for optimizing the performance of the appliance.

2) Push and Update: For Ionic apps, you'll jump to the resubmission of your app changes on the app store. Just push app supplements & innovative features of your app.

3) It's pleasure to create and operate on Ionic apps: Application progress depends on the outlining and UI feel and technology change and delivery perfection.

4) Beautiful Designs: Ionic is concentrated mainly on the design and feel, and UI interaction of your app. There are great UI interactions, gestures, animations, and other things.

VI. Comparison Between Cross-platform Platforms

	Xamarin	IONIC	React Native	Flutter
Programming language used	C# and .Net	HTML5, CSS and JavaScript	JavaScript and React.JS	Dart – Object Oriented, Easy-to-learn
Identical to Native applications	Close	Moderate	Very close	Highly close
Native Performance	Good	Better	Great	Great
GUI	UI using Xamarin, Android/iOS or Xamarin Forms.	UI using HTML and CSS.	UI using Native UI controllers	Uses exclusive widgets and produce amazing UI
Hot Reload	Yes	Yes	Yes	Yes
Supported Platforms	Android 4.0.3+, iOS 8+, Window 10	Android 4.4+, iOS 8+, Windows 10	Android 4.1+, iOS 8+	iOS 8+, Android jelly beans +
Pricing	Open-Source + Paid	Open-Source + paid	Open-Source	Open-Source
Popularity	54,000 stars on Github (May 2021)	45,500 stars on Github (May 2021)	95,300 stars on Github (May 2021)	120,000 Stars on Github (May 2021)
Popular Apps	Olo, the World Bank, Storyo.	JustWatch, Pacifica, Nationwide and many more.	Facebook, Instagram, Airbnb, UberEats, Pinterest, Skype, Tesla	Hamilton

Fig : Comparison Between Cross-platform Platforms

Each and Every tool and framework has its benefits and drawbacks. With Flutter and React Native giving the most effective native-like experience, Xamarin offering two separate app development approaches, and Ionic offering code quality, the ultimate choice should be made consistent with your project requirements [8].

VII. Why Flutter is Better?

1) Flutter Architecture :

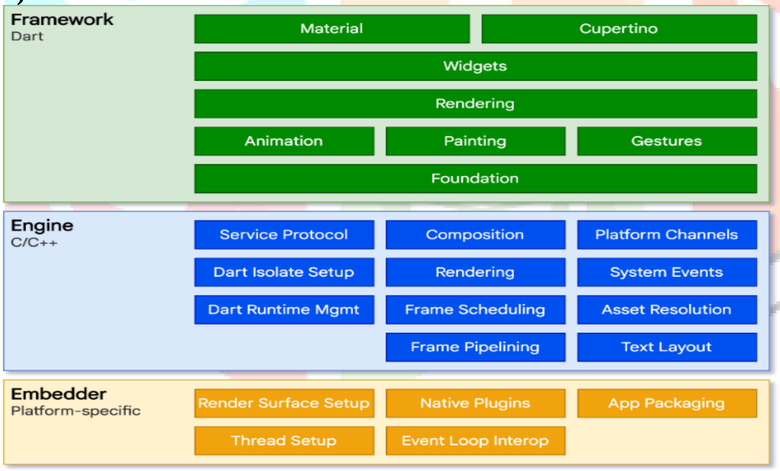

Fig Flutter Architecture [11]

Flutter is a cross-platform UI toolkit that's intended to provide code reuse over operating systems like iOS and Android, while also allowing applications to interface directly with underlying platform service. The core idea of the Flutter framework is In Flutter, Everything is a widget and all about widgets only. Widgets are fundamentally UI components used to build the UI of the application. In Flutter, the application is itself a widget. The application is the parent widget and its UI is developed using children (widgets), which is again built using its children widgets. This composability specialty helps us to create a user interface of any complexity.

2) Dart: Dart language is object-oriented, easy to learn, provides various benefits, and is based on C/C++ and Java, also a product of Google used to code Flutter apps. Google developed and maintained the dart programming language. It is widely used inside Google and it has been verified to have the skill to develop tremendous web applications, such as AdWords. Flutter application refreshes the view tree on each new frame even when few other systems use reactive views. This behavior results in a drawback that a lot of objects, which could survive for a singular frame, will be created.

Looking at the Google trends chart for Flutter is dominating over the other frameworks with their amazing Graphical User Interface, High performance, and rich of widgets that provide native-like experience in comparison to React Native, there is a variation beginning in trends for searching which means that more people are taking interest in Flutter development. That's why Flutter is not used much when gazing at the Stackoverflow survey of 2019 where Flutter is barely utilized by 3.4% of all users and not even on the list for the professional developers use. Dart language is liked by 1.9% of the users. Flutter and Dart both are yet very highly ranked in the "Most loved" categories as 3rd and 12th.

In the Stackoverflow survey of 2020 [9], Flutters users statistics have grown from 3.4% to 7.2% in the overall statistics. Flutter takes the 3rd position because of the most loved framework/library/tool for 2020 [10]. The Flutter 2.0 release for developers has proposed new opportunities within the fields of not only mobile development but also in web and desktop.

Things that all make Google Flutter a Highly Promising Cross-Platform Tool -

1. Fast Development - Flutter gives a Hot Reload feature, similar to React Native, which allows developers to make changes in the code and see how it appears in the app right away (in milliseconds) in the app. So, using Flutter, developers can simply and instantly add features, fix bugs, and experiment with the app's look.
2. Easy to learn - Flutter uses the Dart language, unlike React Native that uses Javascript. That's why, developers would must learn Dart to use Flutter, but the relief comes within the easy learning. Any developer with experience in Object-oriented programming (OOP) languages, such as C++ and Java, can easily learn Dart and start using it.
3. Custom Designs - The attraction of Flutter is that it allows you to add new widgets and customize the existing ones as per your wants and needs to delight users. So, you can easily try a different look till you are fully satisfied with your design.
4. Reactive Programming Architecture - Similar to React Native, Flutter's architecture is based on reactive programming. Flutter apps react to user input by changing variables/properties and the re-rendering of the UI takes place based on the new state.
5. Compatibility with Older Devices - Flutter apps have the identical look even on old versions of Android and iOS. There's no requirement for compact libraries as the widgets and the renderer for those widgets are part of your app.
6. High Performance and Speed - Flutter apps are compiled into native code for multiple platforms, which helps improve overall performance and eliminates issues that arise due to the requirement for a JavaScript bridge. Since Dart doesn't depend upon JavaScript bridge, it greatly reduces the app start up times yet.

VIII. Conclusion :

In the circumstances of developing a mobile application for several platforms and/or different form factors, We should identify what mobile application type we target, and which approach to use. The study through a comparison of mobile cross-platform development approaches made us able to understand each approach, and so we can say that the Flutter is a beneficial toolkit that provides easy ways of building new applications. It has become more and more popular recently. The basic results in this report indicate flutter has a slight advantage as compared to other cross-platform development platforms but moreover certain tests still require to be carried out to come to a final result. Appearance-wise, Flutter and native appear to modify little to a bulk of users. It is capable to mimic the native looks to a certain point. To settle the discoveries and ideas of Flutter, it is a tool with an assuring feature if the community maintains to grow in the direction that it is right now. The path to draw when to settle on Flutter over two separate native builds and other cross-platforms may be chosen at the event of smaller to medium applications which are more flexible. Considering that Flutter's strong side is being a cross-platform solution, Flutter still performs well on one application base if compared to native applications. Flutter might not beat native for developing applications at this time but the results show good potential for the future although further studies have to be performed in these areas to conclude safer solutions.

IX. References:

[1] Beginning App Development with Flutter: Create Cross-Platform Mobile Apps by Rap Payne
[2] Wenhau Wu. March 2018 Thesis React Native vs Flutter ,Cross-Platform Mobile Application Framework
[3] Cross-platform mobile development approaches,Publisher: IEEE
[4] Analysis of Cross-Platform Mobile App Development Tools, Publisher: IEEE

[5]https://fluttercompletereference.com/preview/flutter_reference.pdf
[6] Flutter Complete Reference: Create beautiful, fast and native apps for any device, by Alberto Miola (Author), Felix Angelov (Editor), Matej Rešetár (Editor), Rémi Rousselet (Editor)

[7]https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/
[8] https://www.elitechsystems.com/comparison-react-native-vs-xamarin-vs-ionic-vs-flutter/
[9]https://insights.stackoverflow.com/survey/2020#technology-most-loved-dreaded-and-wanted-languages-loved
[10]https://insights.stackoverflow.com/survey/2020#technology-most-loved-dreaded-and-wanted-languages-loved
[11]https://flutter.dev/docs/resources/architectural-overview