COMPARISON OF FLUTTER AND OTHER MOBILE DEVELOPMENT FRAMEWORKS

SAHIL HUSSAIN SIDDIQUI, SOUMYA JAISWAL, SHARAD SHARMA, KULYASH
“Department of Computer Science and Engineering, Noida Institute of Engineering and Technology’
Knowledge Park 2, Greater Noida, Utter Pradesh, India

siddiquisahil02@gmail.com
sharad.sharma1999@gmail.com
soumya.jais1406@gmail.com
kulyashdahiya77@gmail.com

Abstract: This study was done to show the benefits of using Flutter over other application development platforms. It deals with working on different platforms and their role in application development. In today's world, ease of development is the thing every developer is looking for, and Flutter has built the foundation needed to support app development for both Android and iOS. Developers are forced to build the same app multiple times for different operating systems (operating systems) or settle for a similar low-quality solution at the trade-off of native speed and accuracy for native accuracy. mobile body. Flutter is an open source SDK for developing more reliable and high-performance mobile applications for operating systems like iOS and Android.

Keywords: Flutter, iOS, Android, Cross-platform Development

1.INTRODUCTION

To make it available to most users, a mobile app needs to be familiar with two independent platforms, Android and iOS. These two platforms have major differences that often require different skills to develop. For example, Java or Kotlin for Android and ObjectC or Swift for iOS. As a result, 4, developers and businesses in general struggled to cope with the

complexities involved in developing cross-platform applications.
Flutter-like cross-platform framework has been discussed and implemented by different companies many times before. However, that is not enough to meet the requirements of industrial development. Although the predecessor to was ineffective, Flutter, supported by Google, is gaining attention and developers are also finding it easier to use**.** Flutter apps can also work the same on both platforms’ platform, thereby reducing the cost and complexity of building apps on iOS and Android. Flutter was built completely from scratch and around August 2017, only Google used it for commercial projects.

2.EASE TO USE

Flutter is a cross-platform framework aimed at developing, high-performance mobile applications. In addition to working on, floating apps on Android and iOS also work on Fuchsia. Flutter is special because it relies on device OEM widgets rather than using web views. Flutter uses a high-performance rendering engine to render each element of view using its own rendering engine. This gives the opportunity to build apps as good as native apps can be.

3. FLUTTER ARCHITECTURE

With architecture, the C or C++ engine code involves compiling with Android's NDK and LLVM for iOS respectively, and during compilation, the Dart code is compiled to native code. Flutter's hot reload feature is known as state hot reload and is a major factor in speeding up the development cycle. Flutter supported it through out’s development. Hot-State Reloading is accomplished by pushing updated source code to a running Dart Virtual (Dart VM) machine without altering the internal structure of the device. application. Therefore, transitions and actions will be well preserved after hot reload.

3.1. DART

In Flutter, every application is written in Dart. Google has developed and maintained a programming language called Dart. It is widely used in Google and has been proven to be capable of developing huge web applications, such as AdWords. Flutter app renews the view tree with every new frame even if some other system is using responsive view. This behaviour has the disadvantage that multiple objects, which can exist in a single frame, will be created. Since Dart is a modern programming language, it is optimised to handle this situation at memory level using "Gene Generation Garbage Collection".

3.2 Flutter/Dart UI management

Flutter uses widgets as the main concept in the code. Widgets are the nicknames for each of the components built into Flutter. It can mean a box or a text called a widget. A notable part of the widgets is that they are made by Flutter developers to look native and developers can fully customise them to their liking

3.3 Flutter compiling

The way Flutter works when running on Android is to compile Dart code written by native developers with the help of AOT compilation. These, along with the x86 libraries generated when darts are compiled, are injected into the executor and built as APKs. Flutter works similarly on iOS systems, but instead the Dart code compiles to an ARM library, and is then placed as a runner and built as app.

4. COMPARISON OF FLUTTER WITH OTHER DEVELOPMENT PLATFORMS

4.1 Native Mobile Applications

The meaning of native in the field of mobile applications refers to applications that are designed to run on a particular platform or operating system. There are many languages that can be used to create native mobile apps and some examples out of:

Kotlin, Java and Swift. Mainly, developers have focused on creating native apps since customizations are enabled for rooted devices like camera access etc.

4.2 Native implementation of UI

One thing that ties into native mobile apps is smoother animations and easier integration with mobile technology. Native apps inherit the look and feel of the targeted platforms and apply it to them interface, this is known as native UI. This allows native apps to work and look more like the “native “system to provide users with an experience more tailored to the mobile platform operating system.

4.3 Cross-platform mobile application development

Cross platforming refers to a product or software that can be used on a platform other than the one on which it was developed. In application development, the premise of cross-platform is to create and maintain a single codebase, this is the attractive part of using it because it saves development time compared to the original is limited to one platform per codebase. Examples of cross-platform frameworks/tools are Flutter, React Native, and Ionic.

4.4 Use and popularity

In an article written by Mehdi Satei 2019, there is a section where the author discusses programming languages and popular frameworks in the industry. Satei presents statistics from 2019, where Dart ranks first among the most searched languages while Flutter is the 3 most searched frameworks. Looking at Google's trend chart for Flutter vs. React Native, there's an initial difference in search trends which means more people are interested in the development more Flutter's development.

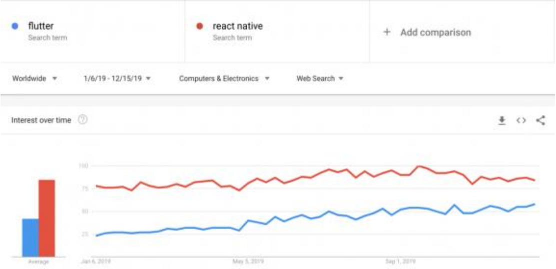


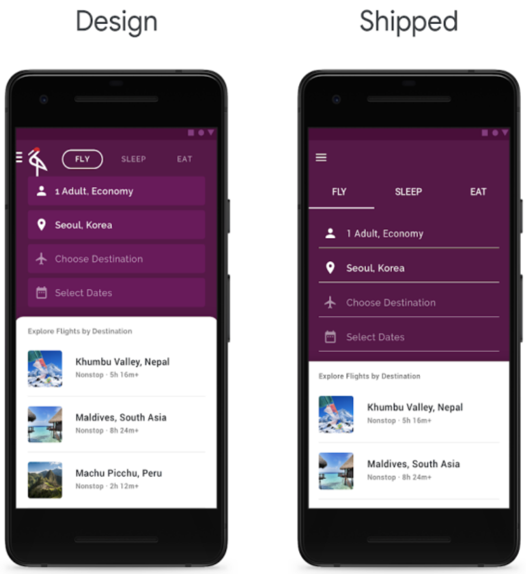
Figure 1: Trends in Flutter vs React Native

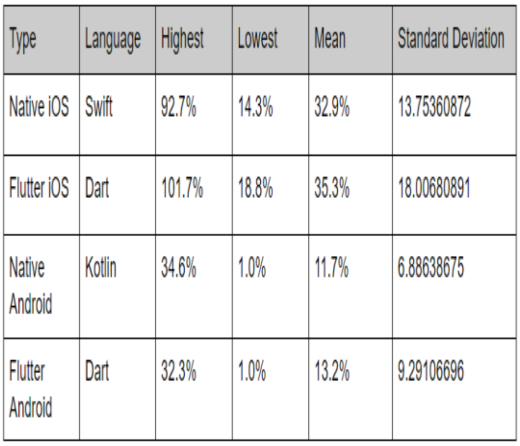
However, Flutter is not widely used considering the 2019 Stackoverflow survey, where Flutter is only used by 3.4% of all users and not even on the list of for developers professional. 1.9% of users like Dart, while for professional developers, Dart is not on the list. However, Flutter and Dart ranked highly in the "Most Liked" category at 3rd and 12th respectively.

In the 2020 Stackoverflow survey, Flutters usage stats grew from 3.4% up 7.2% in the overall stats, but remains off the list as before in the professional stats. Flutter remains at 3rd as most popular framework/library/tool for 2020.Dart is still off the professional Dart usage list, but has grown from 1.9% to 4.0% as of 2019. For 2020 According to Stackoverflow survey, it has become 7th most popular programming language

5. RESULTS OF BASIC APPLICATION

Looking at the results as a whole, Flutter wins the majority of most categories in the development area. There are however some differences that are interesting to take note of when comparing Flutter to native builds.





Type	Language	Highest	Lowest	Mean	Standard Deviation
Native iOS	Swift	92.7%	14.3%	32.9%	13.75360872
Flutter iOS	Dart	101.7%	18.8%	35.3%	18.00680891
Native Android	Kotlin	34.6%	1.0%	11.7%	6.88638675
Flutter Android	Dart	32.3%	1.0%	13.2%	9.29106696

Figure 4: Development time of each code base

Figure 3: Application lines of code and file count

5.1Code Size

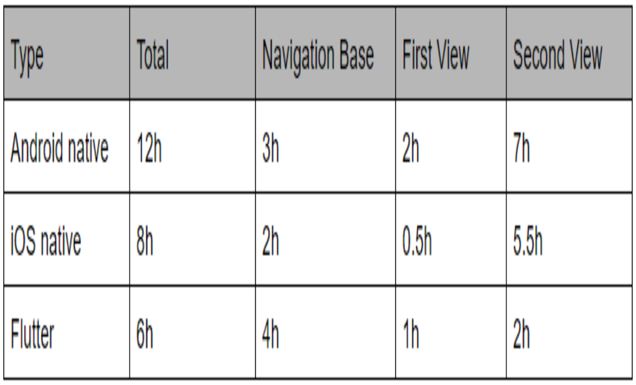
As shown in Figure 4, Flutter requires the fewest lines of code and files to build your application. The native iOS project file size and app size were small, but there were significantly more lines of code than the other builds. Native Android created the most files and required less than lines of code than native iOS.

5.2 Development Time

Android natives had the longest development time of 12 hours, followed by iOS natives with 8 hours, and finally Flutter with the shortest of 6 hours. If you are developing both native applications, you can take advantage of the use of drag and drop to accelerate your development. This is useful until you need to connect the deleted element to code This was more difficult than using the code directly to generate the layout. Layouts are always visible, even if you don't create them, which makes development easier. An equivalent development feature of Flutter is the hot reload feature. This allows you to build your application and reload it based on the newly added features.

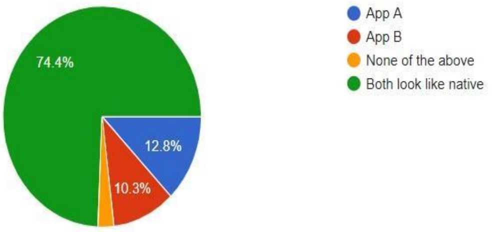
5.3 Flutter run time CPU performance comparison

Flutter iOS has higher top CPU performance than native iOS, and native Android has top CPU performance better than Flutter Android. But overall, Flutter has better average CPU performance on both OSs, outperforming native CPU performance on both iOS and Android.



Type	Total	Navigation Base	First View	Second View
Android native	12h	3h	2h	7h
iOS native	8h	2h	0.5h	5.5h
Flutter	6h	4h	1h	2h

5.4 Application Looks



This diagram is a survey conducted which clearly shows that most people believe that both the applications look the same.

6. CONCLUSION

Flutter is a useful tool kit that enables easy ways of creating new applications. It has gotten more and more popular recently. The basic results in this report indicates flutter has a slight edge as compared to native application development platforms but further more conclusive tests still need to be carried out to come to a final conclusion. Appearance wise, Flutter and native seem to differentiate little to a majority of users. It is able to mimic the native looks to a certain point.

To conclude the answers and ideas of Flutter, it is a tool with a promising feature if the community continues to grow in the direction that it is right now. The line to be drawn when to choose Flutter over two separate native builds, can be chosen at the development 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
a single application base if compared to native applications. Flutter may not beat native for developing applications at this point but the results show good potential for the future although further studies needs to be done in these areas to conclude safer answer

7. REFERENCES

[1] Wenhau Wu. March 2018 Thesis React Native vs Flutter, Cross Platform Mobile Application Framework.

[2] M. Satei, “Ott video oriented mobile applications development using cross platform UI frameworks,” KTH Royal Institute of
Technology, School of electrical engineering and computer science, diva2:1343759: Diva, 2018, p. 90.

[3] C. Software. (2019). What is flapping? Here's everything you need to know online. Available: https: // Medium. com /

@ concise software / The flutter here is everything you need to know faed 3836253f (visited March 5, 2020)

[4] ME Joorabchi、A Mesbah and P Kruchten、 "Real Challenges in Mobile App Development"、University of British Columbia
、University of British Columbia
、2013、p。 10。 [online]. Ver füg bar: http: //citeseerx.ist.psu. edu / viewdoc / download? doi = 10.1.1.724.2463 & rep = rep1 & type = pdf.

[5] C. Griffith. (2019), cross-platform application development is [online]. Available: https://ionicframework.com/resources/articles/whatiscrossplatformappdevelopment

[6] Google. (2019). Google Flatter SDK Publications, Online. Available: https: //flutter.dev/docs/development/tools/sdk/releases

[7] W. Danielsson, "React Native Application Development", Diva, diva2: 998793: Faculty of Computer Science LIU, 2016
[8] Google. (2020). Material IO Components, Online. Available: https: //material.io/components

[9] Stackoverflow, (2020). Stackoverflow Poll 2020, Online. Available: https: //insights.stackoverflow.com/survey/2020

[10] G. Flutter. (2020). Google flutter animations, [Online].