iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** I **Month of publication:** January 2022

DOI: https://doi.org/10.22214/ijraset.2022.39920

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue I Jan 2022- Available at www.ijraset.com

Review on Mobile Application Development Based on Flutter Platform

Shreya A. Bhagat [1], Sakshi G. Dudhalkar[2], Prathmesh D. Kelapure[3], Aniket S. Kokare[4], Prof. Sudesh A. Bachwani[5]
[1, 2, 3, 4, 5]*Department of Computer Engineering*
[1, 2, 3, 4]*Government College of Engineering Yavatmal, Maharashtra, India,*
[5]*Assistant Professor, Government College of Engineering Yavatmal, Maharashtra, India*
[1, 2, 3, 4]*Dr. Babasaheb Ambedkar Technological University Lonere, India*

Abstract: ***Covid-19 pandemic has made mobile applications very popular. There is a huge competition between many mobile application development frameworks available in the market. In today's world ease of development is the thing all developers are seeing for and flutter has come with the required platform to support development of applications for both iOS and Android platforms, consequently decreasing the cost and complexity of application creation across both platforms. Flutter is the cross-platform mobile application development open-source SDK tool for developing high-performance and more reliable mobile applications for iOS and Android operating systems. This paper has been undertaken to indicate the benefits of using flutter over other app development platforms.***

Keywords: Cross-Platform Mobile Application, Flutter, Dart, GetX State Management, Firebase.

I. INTRODUCTION

Mobile application development is rapidly growing. From trade, telecommunications and e-commerce to insurance, healthcare and government, organizations across industries must meet user hopes for real-time, appropriate ways to conduct transactions and access data. Nowadays, the mobile applications that unlock their value are the most popular way for people and businesses to connect to the internet. To stay relevant, responsive and effective, organizations essential to develop the mobile applications that their customers, partners and employees demand.

One of the major problems that are being faced by the developer is cross-platform development. In this, the developers have to choose the Operating System which means select either Android or iOS or both, and develop the application accordingly. If in case an application is to be developed in both Android and iOS, then they have to code in different platforms, need to learn different programming language according to the corresponding Operating System, different testing needs to be done which is a hard task as well as time-consuming and cost a lot of money.

To overcome the cross-platform problem, Google has developed a platform knows as Flutter which is an open-source mobile user interactive framework, declared in the year 2017, and is ranked 34th among the software, which is used for the development of an application [1]. It can develop native applications with a single code. In a simple word, the functionality of flutter, which is a software, is that it allows a developer to develop a mobile application of both the operating system that is Android and iOS, with only one codebase. This means that you can use one programming language and one codebase to create two different apps (for iOS and Android) [2]. The only programming language that flutter use is known as Dart. Dart is used for both back-end as well as for front-end purposes to enhance the user experience while using an application [3]. Irrespective of the fact that flutter is developed recently, it not only solves the problem of cross-platform but also provides some additional benefits which are beneficial for a developer as well as for the software industry [4,5].

II. RELATED WORK

Flutter is Google's portable UI software development kit (SDK) for crafting natively compiled mobile, web, and desktop applications. It offers a complete environment with a framework, widgets, and tools. Flutter is also open-source and free, mens you can use it in a straightforward way.

Combined, these things allow you to create apps efficiently and quickly. Flutter is a cross app development framework that faultlessly interacts with cameras, geolocation, networks, and storage. It performs better than other cross-platform development technologies like React Native and Xamarin. The architecture and engineering design of Flutter allows you to build responsive and user-friendly applications.

III. WHY FLUTTER

Flutter as a framework is very promising and right now has a big dev community. Even now we can find complex apps in the market which are based on Flutter, like Alibaba, Google Ads, Reflect, Birch Finance, Hamilton Musical, Hookle (Skuza, 2019). In the Authors opinion, this technology is a good choice for small and medium-size applications or when content and basic features require constant iteration.

The technology potential is also big as during Flutter interact conference Google introduces support for web applications (Sneath, 2019). Dart language is also the fastest-growing programming language nowadays. Its list features added during the last two years is also big and includes extension functions, null safety support [6].

IV. FEATURES OF FLUTTER

1) Flutter's hot reload feature allows it to offer faster performance. With it, the application gets compiled using the arm C/C++ library, creation it closer to machine code and enabling it to run more quickly. The Flutter team has put lots of effort into providing a wide variety of ready-to- use widgets. Most of them are incredibly customizable, saving your time like no other framework before.

2) Using Flutter, you can write code, manage, and run it across multiple platforms such as Android, IOS platforms. For the developers, this saves time, money, and effort.

3) This makes the app development process simpler and faster. Additionally, it also allows us to modify or update the code once a change is made.

4) In contrast to most cross-platform frameworks, Flutter does not rely on intermediate code representations or interpretations. The Flutter application is made directly into the machine code, which removes any performance issues associated with the interpretation process. With Flutter, you get a fully compiled release application ahead of time.

5) Each Flutter app is built using Dart programming language, which uses JIT and AOT compilation for faster startup time, faster performance, and smoother functionality. With the JIT feature, you can increase the speed of development and refresh the UI.

6) Flutter's documentation is well-organized and informative. It serves as a central repository for all written documents.

7) Flutter offers a customizable layered architecture that allows for highly customizable designs, expressive UIs, and fast rendering.

V. DART PROGRAMMING LANGUAGE

Flutter itself is not a programming language. Rather, it’s an SDK with prewritten code, consisting of ready-to-use and customizable widgets.

The programming language that’s used is Dart, is also developed by Google. By avoiding using a bridge to communicate with the native layer such as Android or iOS, Flutter minimizes performance issues and boosts app start-up time.

To develop an app using Flutter, you need developers to code in Dart. That shouldn’t be an issue, because it’s similar to Kotlin, Java, Swift, and JavaScript. Also, it’s easy to learn. According to Google, Dart is a client-optimized language for fast apps on any platform. Object-oriented like Java, C++, and Python, it compiles ahead of time to native ARM or x64 machine code, and to JavaScript bytecode for web. As a result, apps written in Dart are impossible to distinguish from native apps at the machine level.

VI. FEATURES OF DART PROGRAMMING LANGUAGE

A. Open Source

Dart is an open-source programming language, means it is freely available. It is developed by Google, approved by the ECMA standard, and comes with a BSD license.

B. Platform Independent

Dart supports all primary operating systems. They are Windows, Linux, Macintosh, etc. The Dart has its own Virtual Machine which known as Dart VM, that allows us to run the Dart code in every operating system.

C. Object-Oriented

Dart is an object-oriented programming language and supports all oops concepts such as classes, inheritance, interfaces and optional typing features. It also supports advance concepts like mixin, abstract, classes, reified generic, and robust type system.

D. Easy to Learn Language

Dart is easy language to learn, and Google developers have put in a tremendous effort in the documentation part. Developers with OOPS background can quickly plunge into programming if they know the basics because it having the Java-like syntax. Dart also allows for easy editing as they can test small sections of code even if the complete application is not ready yet. Dart is fairly easy to grasp, modern, functional, flexible and competitive. The ecosystem is simple, understanding the terminologies, the proper tools and SDKs for the language is easy, and accessing the frameworks and libraries is easier. If a developer is familiar with any programming language, not just necessarily an OOP language, they can intuitively start using Dart.

E. Comes with Good Documentation

Developers discover that Dart is a good first programming language to learn because it has very good documentation and excellent introduction. Getting started is also easy; just type the Dartpad url, and you can get sarted. More and more developers have switched to Dart, thanks to its simple syntax, excellent community support, easy features that guides developers when they are in the training process.

F. High Performance Factor

Applications run in Dart run faster than in other programming languages. And features like JIT compilation and AOT compilation add to the performance feature of Dart. JIT compilation or Just in Time compilation helps you to enable hot reloads, while AOT or Ahead of Time compilation helps with fast startup and better execution of the app.

VII. FLUTTER 2.5

The initial release of Flutter in 2017 spelled the beginning of a new era in the hybrid development of cross-platform apps released by Google. Flutter took the world by storm and, within a few years, countless companies and developers adopted Flutter as their platform and framework of choice for developing cross-platform applications. Because Flutter offers you to develop mobile (Android and iOS), desktop, and, as of the release of Flutter 2.0, even web apps without changing their codebase or rewriting the apps from scratch.

Flutter 2.5! This is a big release, with the 2nd highest states in the history of Flutter releases: 4600issues closed and 3932 PRs merged from 252 contributors with 216 reviewers. If we look back over the last year, we see a huge 21,072 PRs created by 1337 contributors, of which 15,172 of them were merged [7]. According to Statista, Flutter ranks second place in the list of ten most famous cross platform frameworks of 2020. That’s a big achievement for a technology that came into the app development world just three years earlier!

The release of the new Flutter version on March 3, 2021, finally made it possible not only to write new code using the null safety feature but also to migrate the previous Flutter apps to null safety. Null pointer or null reference was created by Tom Hoare in 1965. Later, in 2009, he would call it a billion-dollar mistake during his speech at a software development conference. Tom Hoare was designing the first comprehensive type system for object-oriented programming.

To make sure that the use of all references is safe, he decided to implement the null pointer reference as the easiest solution. As a result, the null error was born. Sound null safety eliminates bugs caused by null pointers. It makes types in your Dart code non-nullable by default.

It means that variables can’t contain null unless you say they can. Dart and Flutter became a good solution to developer productivity issues and reduced the number of bugs by adding null safety. With null safety, now Flutter developers can specify which variables can be null and all the before unnoticed errors will display up during static analysis. It means that all issues can be observed and fixed fast during edit time [8].

VIII. ADVANTAGES OF FLUTTER

A. Hot Reload

Flutter hot reloading helps save time while developing by letting the developer see the applied changes in real-time. This ability helps developers be significantly more efficient and productive. It allows the developer to pause code execution, make changes to the code, and continue the code from the same place. It allows seeing the applied changes almost instantly, without even losing the current application state. And this is exactly what makes Flutter app development several times faster due to the increased development speed.

B. Own Rendering Engine

Flutter allow you to do so much stuff with your apps that aren't available on other platforms. Obviously, it requires the framework to be pretty powerful. Flutter uses Skia for rendering itself onto a platform-provided canvas. Because of the engine, UI built in Flutter can be launched on virtually any platform. Putting it differently, you no longer have to adjust UI to transfer it to a platform, which simplifies the development process hugely.

C. Same Business Logic and User Interface.

Likely for Flutter sharing the UI and business logic on Android and iOS operating system allow developers to achieve a seamless experience regardless of the OS.

D. Getting Started

Getting started with Flutter is way too easy and the documentation is really good. Creating first application with Flutter is easier than with Android and iOS.

E. Open Source and an Engaged Community

As an open-source platform, Flutter is free to use and has a growing community contributing to its outstanding documentation and assisting with issues that developers may encounter. There are also many YouTube videos available for those who want to start learning Flutter or improve their skills in this Google's mobile UI framework.

F. One Code for 2 Platforms

Developers write just one codebase for your 2 apps covering both Android and iOS platforms. Flutter doesn't depend on the platform, because it has its own widgets and designs.

IX. ARCHITECTURE OF FLUTTER

The architecture of flutter is comprised of four components.

A. Flutter Engine

It is a runtime that is portable for high quality mobile applications. It is based on C++ programming language. The applications of flutter are developed using flutter core libraries which includes graphics and animation, network input output and file, plugin architecture, accessibility support and a dart runtime. Low level graphics can be rendered using open source Google's graphics library called Skia.

B. Foundation Library

The building blocks for development of a flutter application makes use of packages and they are contained in the foundation library. Dart is the language used to write these libraries.

C. Widgets

The core concept of the Flutter framework is the widgets. I flutter, everything is a widget. Widgets are basically user interface components used to create the user interface of the application. The application is itself a widget in flutter. The application is the top- level widget and its UI is build using one or more children (widgets), which again build using its children widgets. This composability feature helps us to create a user interface of any complexity. For example, the widget hierarchy of the hello world application (created in previous chapter) is as specified in the following diagram –

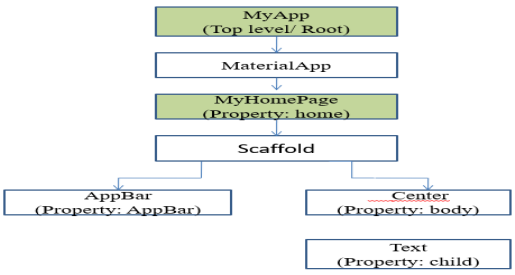


Figure1: Layers of Widgets

Here the following points are worth notable −

1) MyApp is the user created widget and it is build using the Flutter native widget, MaterialApp.
2) MaterialApp has a home property to specify the user interface of the home page, which is again a user created widget, MyHomePage.
3) MyHomePage is build using another flutter native widget, Scaffold
4) Scaffold has two properties – body and appBar
5) Body is used to specify its main user interface and appBar is used to specify its header user interface
6) Header UI is build using flutter native widget, AppBar and Body UI is build using Center widget.
7) The Center widget has a property, Child, which refers the actual content and it is build using Text widget

D. Design Specific Widgets

Flutter has widgets specific to a particular platform - Android or iOS. Android specific widgets are designed in accordance with Material design guideline by Android OS. Android specific widgets are called as Material widgets. iOS specific widgets are designed in accordance with Human Interface Guidelines by Apple and they are called as Cupertino widgets.

Some of the most used material widgets are as follows − Scaffold, AppBar, BottomNavigationBar, TabBar, TabBarView, ListTile, RaisedButton, FloatingActionButton, FlatButton, IconButton, DropdownButton, PopupMenuButton, ButtonBar, TextField, Checkbox, Radio, Switch, Slider, Date & Time Pickers, SimpleDialog, AlertDialog

Some of the most used Cupertino widgets are as follows – CupertinoButton, CupertinoPicker, CupertinoDatePicker,CupertinoTimerPicker, CupertinoNavigationBar, CupertinoTabBar, CupertinoTabScaffold, CupertinoTabView, CupertinoTextField, CupertinoDialog, CupertinoDialogAction, CupertinoFullscreenDialogTransition, CupertinoPageScaffold, CupertinoPageTransition, CupertinoActionSheet, CupertinoActivityIndicator, CupertinoAlertDialog, CupertinoPopupSurface, CupertinoSlider

1) *Gestures:* Gestures are widgets in it used for interaction as in how to respond to and how to listen to and this done by using Gesture Detector. The interactions of tapping, dragging and scaling of the child widget is included in the gesture detector which is an invisible widget. Other features for interaction can be included in the widget by using gesture detector.
2) *State Management:* The state of the flutter widget is maintained by a special widget i.e. Stateful widget. Whenever there is an internal change, stateful widget is re-rendered automatically. The distance between the old widget user interface and new widget interface is calculated to optimize the re-rendering and the essential things are only rendered those are changes.
3) *Layers:* The core concept of flutter architecture are layers which are divided into several categories based on complexity and the arrangement is made in a top down manner. The application's user interface is the topmost layer and this layer is particular to android platforms and iPhone operating system platforms. All the flutter widgets are present in the second topmost layer. Everything is rendered to the applications through the next layer which is called rendering layer.

All these layers are trailed by Gestures which are widgets in flutter architecture used for interaction as in how to respond to and how to listen to and this done by using Gesture Detector, foundation library which are the building blocks for development of a flutter application makes use of packages and they are contained in the foundation library, engine and codes that are specific to the platform.

X. FLUTTER PACKAGE

A package is a namespace that contains a group of similar types of classes, interfaces, and sub-packages. We can think of packages as similar to different folders on our computers where we might keep movies in one folder, images in another folder, software in another folder, etc. In Flutter, Dart organizes and shares a set of functionalities through a package. Flutter always supports shared packages, which is contributed by other developers to the Flutter and Dart ecosystem. The packages allow us to build the app without having to develop everything from scratch.

1) *pubspec.yaml:* It is the project's configuration file that will use a lot during working with the Flutter project. This file contains:
a) Project general settings such as name, description, and version of the project.
b) Project dependencies.
c) Project assets (e.g., images).

A. Types of Packages

According to the functionality, we can classify the package into two types:

1) *Dart Package:* It is a general package, which is written in the dart language, such as a path package. This package can be used in both the environment, either it is a web or mobile platform. It also contains some Flutter specific functionality and thus has a dependency on the Flutter framework, such as fluro package.
2) *Plugin Package:* It is a specialized Dart package that includes an API written in Dart code and depends on the Flutter framework. It can be combined with a platform-specific implementation for an underlying platform such as Android (using Java or Kotlin), and iOS (using Objective C or Swift). The example of this package is the battery and image picker plugin package.

XI. GETX STATE MANAGEMENT

GetX is a fast, stable, and light state management library in flutter. There are so many State Management libraries in flutter like MobX, BLoC, Redux, Provider, etc. GetX is also a powerful micro framework and using this, we can manage states, make routing, and can-do dependency injection.

There are three principles of GetX:

1) *Performance:* As compared to other state management libraries, GetX is best because it consumes minimum resources and provides better performance.
2) *Productivity:* GetX's syntax is easy so it is productive. It saves a lot of time for the developers and increases the speed of the app because it does not use extra resources. It uses only those resources which are currently needed and after its work is done, the resources will free automatically. If all the resources are loaded into the memory then it will not be that productive. So better to use GetX for this.
3) *Organization:* GetX code is organized as View, Logic, navigation, and dependency injection. So we don't need any more context to navigate to other screen. We can navigate to screen without using the context so we are not dependent on widget tree.

XII. GETX MANAGEMENT

1) *State Management:* There are two types of state management Simple State Manager: It uses GetBuilder. Reactive State Manager: It uses GetX and Obx.
2) *Route Management:* If we want to make Widgets like Snackbar, Bottomsheets, dialogs, etc. Then we can use GetX for it because GetX can build these widgets without using context.
3) *Dependency Management:* If we want to fetch data from other Class then with the help of GetX, we can do this in just a single line of code. Eg: Get.put().

XIII. FIREBASE

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents. Firebase supports authentication using passwords, phone numbers, Google, Facebook, Twitter, and more. The Firebase Authentication (SDK) can be used to manually integrate one or more sign in methods into an app, Data is synced across all clients in real time and remains available even when an app goes offline, Firebase Hosting provides fast hosting for a web app; content is cached into content delivery networks worldwide, the application is tested on virtual and physical devices located in Google's data centers and Notifications can be sent with firebase with no additional coding.

Firebase is a platform for mobile and web application development that provides services and tools needed to develop a successful application [9] [10]. With the help of firebase, you can improve the quality of your app by using services such as crashlytics, performance, monitoring, test lab, real time database, authencation and many more [11]. The platform is run by Google and trusted by top apps on the Playstore such as The New York Times, Shazam and Duolingo.

Firebase Realtime database shows data in JSON format. It is a cloud-hosted database and one of the advantages of using this tool of Firebase is that the data is synchronized across all clients in real-time and is even available when the app goes offline. On the other hand, Firebase Authencation, provides ways to authenticate users and thereby provide customized services [12].

XIV. FLUTTER TESTING

Testing is an activity, which is used for verifying and validating a software or application that is bug-free and meets the user requirements. It confirms that the actual result matches the expected result. It also benefits to progress the software or application in terms of efficiency, usability, and accuracy. Testing is one of the most important phases in the application development life cycle to ensure the application is of high quality.

It is the most consuming stage in the application or software development. Flutter framework offers excellent support for the automated testing of an application. Generally, to test an application completely testing categorizes into three types. They are as follows:

A. Unit Testing

It is the easiest method for testing an application or software. It tests a single function, method, or class. The goal of unit testing is to ensure the correctness of code under a variety of conditions. Generally, unit testing does not interact with user input, render on the screen, read or write data from the disk, and do not use external dependencies by default. When you use external dependencies, they are mocked out with packages like Mockito.

B. Widget Testing

Widget testing is used to tests a single widget. The goal of this testing is to guarantee that the widget's UI looks and interacts with other widgets as expected. The process of widget testing is identical to unit testing, but it is more complete than a unit test. This testing involves multiple classes and requires a test environment to find more bugs. A widget, which is being tested, can receive and responds to user actions and events and able to instantiates the child widgets.

C. Integration Testing

An integration testing involves both the testing unit testing and widget testing along with the external components of the application. It authenticates a complete app or a large part of the app. The goal of integration testing is to ensure that all the widgets and services work together as expected. It can also use to verify the app's performance. Generally, integration testing runs on devices such as Android emulator or iOS simulator.

Table I. Trade-offs between different type of testing

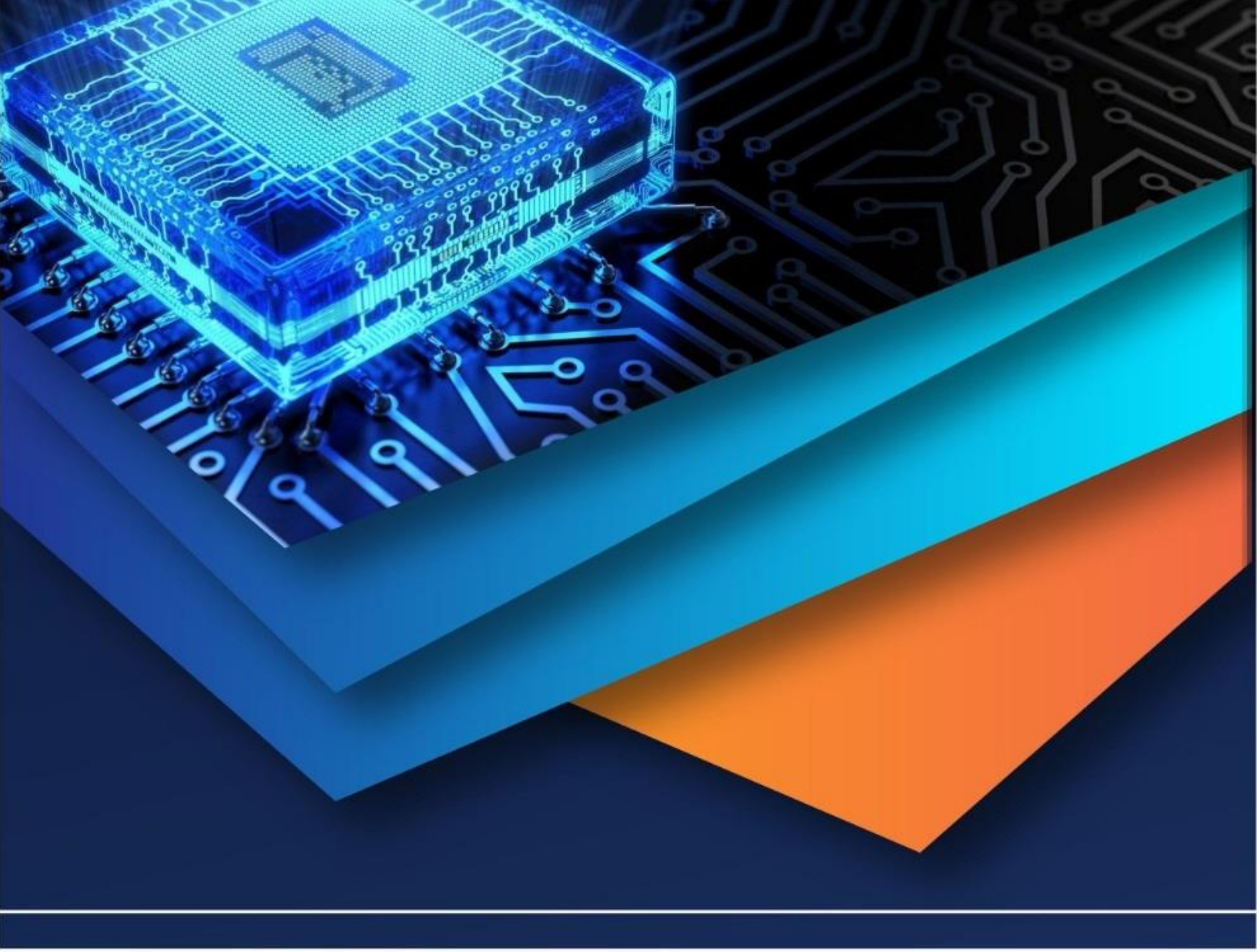
	Unit Testing	Widget Testing	Integration Testing
Confidence	Low	Higher	Highest
Maintenance	Cost	Low Higher	Highest
Dependencies	Few	More	Most
Execution Speed	Quick	Quick	Slow

XV. CONCLUSIONS

Flutter is one of the greatest innovative mobile technologies on the market right now. Flutter is basically the fastest framework to develop cross-platform mobile application Flutter has bright future and huge opportunities for developers. Flutter has become a powerful framework and can't be ignored anymore. For businesses looking to create applications on both iOS and Android, Flutter is a best choice. Flutter is a useful toolkit that enables easy ways of creating new applications. It is the best option for making the apps with amazing UI and high performance. it is a 100% promising framework, considering speed of implementation

REFERENCES

[1] Technologies, T., 2019. Why Should Android App Developers Consider Flutter? [Blog] Think Future Technologies. Available at: https://www.tftus.com/blog/why-mostly-android-developer-consider-flutter-app-development, Accessed on: Sep. 29, 2020
[2] Kumar, D., 2019. "Flutter" To Build iOS & Android Apps. [Blog] Medium. Available at: https://levelup.gitconnected.com/flutter-to-build-iosandroid-apps-f8786d6fe987, Accessed on: Sep. 26, 2020
[3] Dart dev. n.d. Dart Programming Language. [website] Available at: https://dart.dev, Accessed on: Sep. 26, 2020
[4] Martin, S., 2019. Why Flutter Has Become the Best Choice to Develop A Startup Mobile App In 2020? [Blog] Medium. Available at: https://medium.com/flutter-community/why-flutter-has-become-the-best-choice-to-develop-a-startup-mobile-app-in-2020 5785ea153b13#:~:text=Firstly%2C%20Flutter%20allows%20developers%20to,to%20work%20on%20multiple%20interfaces, Accessed on: Sep. 30, 2020
[5] Sharma, A., 2020. Kotlin Vs Flutter: Who Will Rule the Cross-Platform App Market? [blog] Appinventiv. Available at: https://appinventiv.com/blog/kotlin-vs-flutter-cross-platform-app-development, Accessed on: Sep. 29, 2020
[6] Szczepanik M, Kedziora M. State Management and Software Architecture Approaches in Cross – Platform Flutter Application. InENASE 2020(pp.407-414)
[7] Flutter Developer Tools - https://flutter.dev/learn
[8] Flutter Tutorial: https://proxify.io/articles/flutter-2-null-safety
[9] Firebase Realtime Database and Firebase Authentication: https://firebase.google.com/
[10] "Mastering Firebase for Android Development: Build Real-time, Scalable, and Cloud-enabled Android Apps with Firebase" by Ashok Kumar S[Book]
[11] "Firebase Cookbook: Over 70 Recipes to HelpYou Create Real-time Web and Mobile Appications with Firebase" by Houssem Yahiaoui [Book]
[12] A, Upadhyay A, Sabitha AS, Bansal A, White B, Cottrell L. Implementation of PingER on Android Mobile Devices Using Firebase. In2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) 2020 Jan 29 (pp. 698-703). IEEE.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY