



Research Paper

Development of Apps Industry using Flutter: A review

¹Prabhav Shukla ²Neha Tyagi ³Deepanshu ⁴Medhavi Agarwal ⁵Shruti Jain
Students 1-4, MCA, Ajay Kumar Garg Engineering College, Ghaziabad Uttar Pradesh (201015)
Assistant Professor-5, MCA Ghaziabad. AKGEC, Ghaziabad

Abstract: Flutter, an innovative open-source UI toolkit created by Google, has significantly transformed the landscape of cross-platform app development. It enables developers to create high-quality native interfaces for Android, iOS, web, and desktop from a single codebase. Leveraging the Dart programming language, Flutter provides a powerful suite of features such as hot reload for expedited development, expressive and adaptable widgets for complex UI design, and a reactive framework for smooth data synchronization. Flutter's primary strength lies in its capacity to offer consistent, visually engaging experiences across various platforms, thereby negating the need for separate development teams and accelerating time-to-market. Its layered architecture grants developers the flexibility to fine-tune every detail of the app's look and behavior, ensuring brand consistency and user satisfaction. Additionally, Flutter's extensive ecosystem includes a vast array of packages and plugins, which simplify the integration of third-party services and APIs, thereby speeding up development processes and enhancing app functionality. In summary, Flutter stands out as an exceptional solution for contemporary app development, delivering unmatched efficiency, performance, and scalability. Its cross-platform strengths, vibrant community, and detailed documentation make it the go-to framework for creating immersive and responsive applications in today's digital era. In Flutter and Dart, the keywords ``const``, ``final``, and ``static`` serve distinct roles concerning variables and class members.

Received 07 June, 2024; Revised 19 June, 2024; Accepted 21 June, 2024 © The author(s) 2024.
Published with open access at www.questjournals.org

I. Introduction

In today's fast-paced technological world, mobile applications have become indispensable tools for managing daily tasks, enabling communication, and providing entertainment on a global scale. As smartphones and tablets become ubiquitous, both businesses and individuals are increasingly utilizing mobile platforms to engage with their audiences and deliver services effectively. Consequently, mobile application development has become a key area of innovation, significantly influencing how users interact with technology.

Among the various frameworks and platforms available for mobile app development, Flutter stands out for its flexibility, performance, and ease of use. Flutter, an open-source UI software development kit from Google, allows developers to create natively compiled applications for mobile, web, and desktop platforms from a single codebase. Its extensive library of widgets, powerful tools, and cross-platform capabilities make Flutter an attractive option for developing high-quality, visually appealing applications that perform well on a wide range of devices and operating systems.

This introduction aims to provide a gateway into the world of mobile app development with Flutter. We will explore the details of Flutter development, highlight its unique features, and examine how it revolutionizes the process of creating engaging mobile experiences. From learning the basics of Flutter to mastering advanced techniques, this guide will arm you

with the knowledge and skills needed to confidently start your own mobile app development projects. Whether you are an experienced developer looking to expand your skill set or a beginner eager to dive into mobile app creation, this guide will serve as your roadmap, guiding you through the specifics of Flutter development. It will enable you to bring your app ideas to

life with precision and efficiency. Join us on this journey into the realm of mobile app development with Flutter, where innovation is limitless and creativity thrives.

II. Literature Review:

In today's digital age, a multitude of mobile applications are in use, presenting developers with significant challenges. One of the foremost issues is the decision to develop for Android, iOS, or both. Typically, applications are developed for both operating systems, necessitating different codebases: Java for Android and Swift for iOS. This dual-language requirement can be cumbersome and time-consuming, as developers must master and apply two distinct programming languages to create essentially the same application for different platforms—a process known as cross-platform development.

Google's Flutter framework addresses these cross-platform challenges by enabling developers to use a single codebase for both Android and iOS applications. Flutter uses the Dart programming language, which allows developers to write code once and deploy it across multiple platforms without modification. This approach significantly streamlines the development process, saving time and reducing the complexity associated with maintaining separate codebases.

When Dart code is written and deployed in the respective Integrated Development Environments (IDEs) like Android Studio for Android and X Code for iOS, the IDEs handle platform-specific adjustments automatically. This seamless deployment ensures that the application runs smoothly on both operating systems.

A practical experiment demonstrates Flutter's efficiency: a simple "Hello World" application was developed using Java for Android, Swift for

iOS, and Dart for Flutter. The Java and Swift versions required multiple files and exhibited considerable complexity.

The experiment's results underscore Flutter's capability to solve the cross-platform development dilemma.

III. RESEARCH GOAL

Ultimately the object of this thesis is to gain a comprehensive understanding of the overall process involved in application development. This paper attempts to determine whether the app has proven useful in addressing an age-old challenge.

It even gives you a documented application that can be used as a reference by people who are planning to develop their own Flutter application. The hypothesis at the core of it is that Fostlings will prove to be an efficient solution in combating this persistent problem: people posting misinformation about car parts all over the web. It will save you time wasted searching your way through countless unreliable sources and will allow you to make better decisions when making a big purchase. In this thesis, as we will see in the following chapters, I tackle these challenges and make significant steps towards a substantially improved experience during car-building.

IV. Flutter

In this chapter, I will introduce the cross-platform framework called Flutter which would improve the understanding of effective coding practices. Flutter is a high-performance framework for building applications with Dart programming language developed by Google. (1) This is because it offers a highly customizable, easy-to-learn set of widgets that generate cool-looking applications.

Widgets are the principal building blocks of a Flutter app. This is where the composability of Flutter lets you bring out that excellent interface with a breeze. We'll discuss the benefits of Flutter in more detail later in this chapter.

4.1 Dart Coding Language:

Dart is an object-oriented programming language developed by Google which supports classes and interfaces. A class example: The above code shows an example of a class in Dart, which looks pretty much like classes from other object-oriented programming languages.

```
void main()
{
  var obj=A(); obj.cusset="Tanya";

  print(obj.cusset);
}
class A
{
```

```
var name;
void setcusset(var name)
{
  this.name=name;
}
String get cusset
{
  return name;
}
}
```

As we discussed earlier, Flutter is fundamentally based on widgets. The key widgets that are used to create the visual aspect of a Flutter application include state management widgets, platform-specific widgets, layout widgets, and foundational widgets.

State management widgets play a crucial role as they manage the lifecycle of a Flutter application by tracking user interactions and data updates. Platform-specific widgets cater to the requirements of different platforms, allowing developers to tailor the application for either Android or iOS.

Layout widgets can accommodate either a single child or multiple children. Single child widgets, such as Container, Center, and Align, are used to position individual elements. In contrast, multiple child widgets like Row, Column, and ListView,

GridView, and Expanded are employed to systematically arrange several elements.

State Management:

State management is crucial for handling state management is essential for managing an application's lifecycle. It encompasses ephemeral state, which denotes the temporary state visible to the user and is controlled by stateful widgets, as well as app state, which remains persistent throughout the application's runtime, retaining user data and session information.

Advantages of Using Flutter:

Flutter offers a range of benefits, making it a preferred choice for developers. One of its key advantages is the ability to create high-performance applications. This is complemented by its extensive customization options and strong support from Google. The hot reload feature significantly accelerates the development process by allowing immediate application of code changes. Dart, the language used with Flutter, provides a rich library of software packages, aiding developers in efficiently realizing their creative visions. Moreover, Flutter's single codebase approach shortens development time, enabling faster deployment to both Android and iOS platforms compared to traditional native development methods.

V. DEPLOYMENT

This chapter is intended for individuals aiming to release their applications on the Apple App Store. If this does not match your goals, you may proceed to the next section. It provides a broad overview to guide you through the app store submission process, offering high-level insights instead of an in-depth tutorial. The primary objective is to help users steer clear of common mistakes during the app publishing process.

Deployment:

To start the process of uploading an application, you'll need to have an Apple Developer account. This account comes with an annual fee of \$100.

After setting up the account, you can refer to [Flutter's iOS deployment guide](<https://docs.flutter.dev/deployment/ios>) for comprehensive instructions on how to prepare your application using Xcode. Should you encounter any problems with Xcode, it is recommended to use Flutter Doctor. This tool provides a summary of any missing elements or features that aren't working correctly. Sometimes, fixing issues might require uninstalling and reinstalling CocoaPods.

Feedback Management:

It is essential to carefully consider user feedback gathered through application reviews. According to the study by William Martin and colleagues titled "A Survey of App Store Analysis for Software Engineering," they found that free applications are more likely to remain in the top charts consistently.

Regular updates with new features are crucial for ongoing success, especially in smaller app categories. Keeping a close watch on feedback and making necessary updates to the app are key practices for ensuring its long-term viability.

Quick Note: Before completing the publication process, make sure that all selected devices and versions are correctly set up in Xcode. Additionally, if the "Debug" red banner is still visible within the application, it cannot be published. To fix this issue, add the following code snippet to the main.dart page.

Permissions:

When developers incorporate photo or camera features into their applications, handling permissions becomes vital. Various techniques are available to tackle the issues related to accessing photos or cameras within an app. However, it's imperative to secure the user's consent to access these features, ensuring compliance with privacy regulations.

VI. WIDGET ARCHITECTURE

This chapter delves into the dynamics of Flutter's widgets. The accompanying diagrams illustrate

widget trees, showcasing how each page incorporates different widgets. While the diagrams present the pages independently, they are, in reality, interconnected and refer to one another.

This section offers a concise guide to comprehending widget composition in Flutter, emphasizing the distinction between single and multiple widgets. It also highlights that all widgets can alter or update the application's state.

Login:

The Login Page Widget sets up the Scaffold, comprising two child components: the AppBar and SingleChildScrollView. SingleChildScrollView, in turn, employs Column, encompassing Padding, FlatButton, and Container. The widget tree for the login page is illustrated in Figure 1.

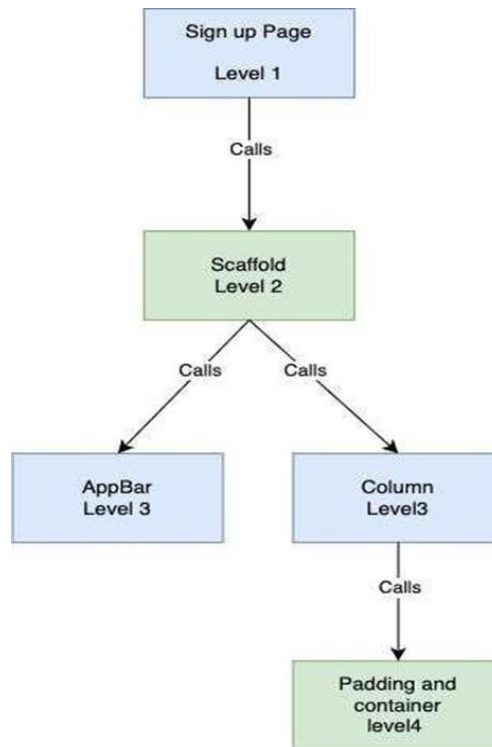


Fig1: Login

SignUp:

The registration process is facilitated by the Sign-Up widget, which is composed of two integral elements: the App Bar and the Column. Within the Column, additional components, namely Padding and Container, are invoked. The hierarchical structure of widgets defining the sign-up page is depicted in Figure 2.

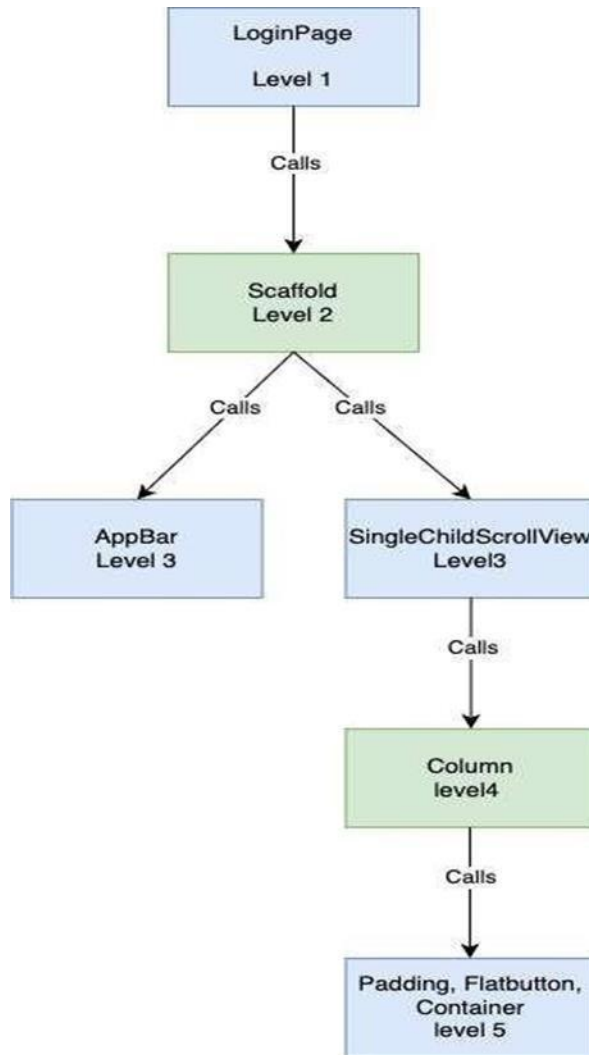


Fig2:SignUp

Specific Builds page:

The page dedicated to specific builds is referred to as Scaffold, which comprises two primary components: an App Bar and a SingleChildScrollView. The SingleChildScrollView, in turn, invokes a Column, which then includes Image.network and Padding. The widget hierarchy for the specific build page is illustrated in Figure 5 below.

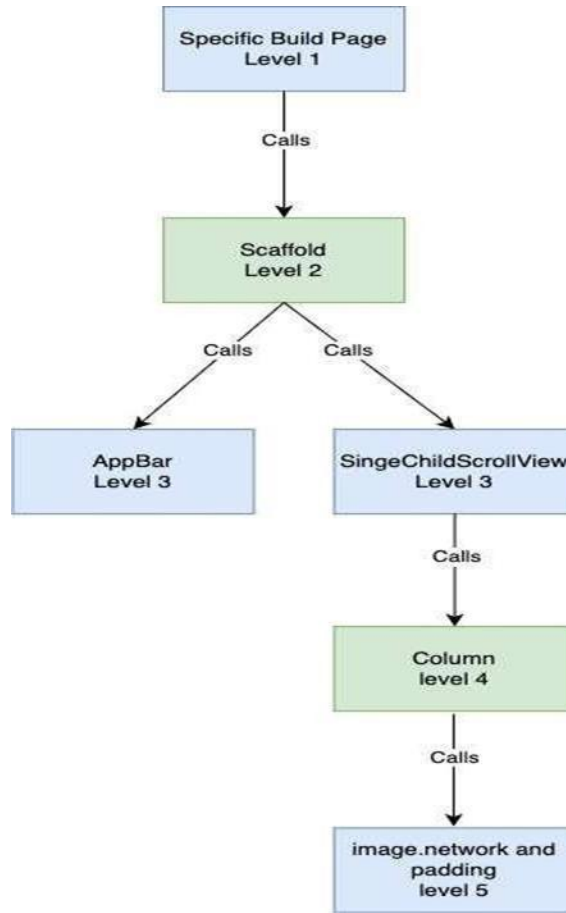


Fig5:Specific Build

Specific Discussion Page:

For the Discussion Page, the main structure is called Scaffold, which consists of two children: AppBar and Column. Within the Column, there's a sequence of widget calls: Expanded calls Container, Container calls Row, and Row calls Expanded.

In this hierarchy, expanded further calls Container, which then calls Box Decoration and RichText. Below, Figure 6 illustrates the widget tree for the Discussion Page.

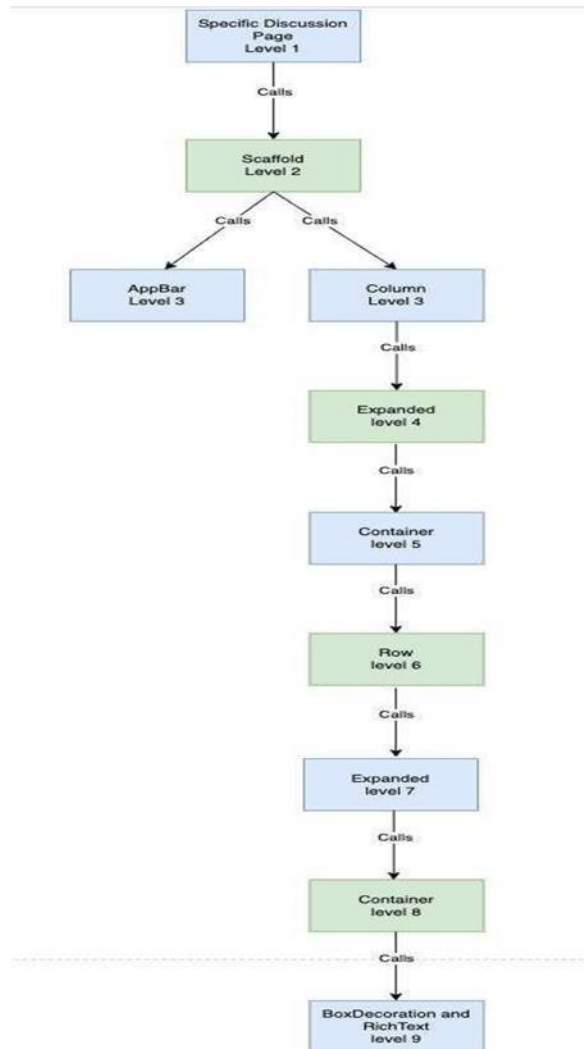


Fig6:Specific Discussion

The surge in traffic results in a significant number of service calls, which may pose challenges for some users. Hence, it is crucial to understand how to efficiently address these issues, as users anticipate resolutions rather than explanations to maintain a smooth experience on the application.

Flutter has demonstrated remarkable benefits in this aspect by obviating the necessity to develop distinct programs for various platforms. The straightforwardness of Dart has facilitated the development of the envisioned application since the inception of this thesis.

Setting a timeframe is not feasible since the publication necessitates a fully operational application beforehand.

REFERENCES

- [1]. MobOSphere Worldwide.(n.d.).Retrieved May 3,2021,from <https://gs.statcounter.com/os-market-Share/mobile/worldwide>
- [2]. MobOSMarkUS21<https://gs.statcounter.com/os-market-Share/mobile/united-states-of-America>
- [3]. JoorMeskru2013 <https://doi.org/10.1109/esem.2013.9>
- [4]. Vallon, R., et al. (2015). Agile and Lean Process Model for Mobile App Development: Austrian Industry Case Study. *Journal of Software*, 10(11), 1245–1264.<https://doi.org/10.17706/jsw.10.11.1245-1264>.
- [5]. Peek,S. (2020,July9).UnderstandingMobile App Development. Retrieved May 11, 2021, from <https://www.businessnewsdaily.com/5155-mobile-app-development.html>
- [6]. Janani. (2021, February 11). Different types of mobile application development:Essential knowledge for every business owner. Retrieved May 11, 2021, from <https://www.zuantechnologies.com/blog/different-types-mobile-application-development-everybusiness-owner-must-know/>
- [7]. "Zohud and Zein (2021) explore cross- platform mobile app dev in the industry via multiple case studies in the Intl. *Journal of Computing*." <https://doi.org/10.47839/ijc.20.1.2091> DesignRush.(2020,November
- [8]. Crafting an innovative mobile app selection guide, Valdellon (2020) explores diverse app categories, aiding users in informed choices. <https://clevertap.com/blog/types-of-mobile-apps/>
- [9]. Crafted with innovation, this definitive handbook demystifies hybrid app development, delivering unparalleled insights and actionable strategies for success. <https://www.designrush.com/trends/hybrid-mobile-app-development>
- [10]. Analyze industry growth, market share, and size for mobile apps till 2026 using the sourced report from Allied Market Research.

- <https://www.alliedmarketresearch.com/Mobile-application-market>
- [11]. Flutter Guidebook: Google's official resource hub is your go-to for everything Flutter and Dart, offering a treasure trove of guides, tutorials, and API references to empower developers at every skill level. [Explore the Flutter Guidebook] (<https://flutter.dev/docs>)
 - [12]. "Essentials of Flutter" by Ed Freitas*: Dive into the world of Flutter with this compact yet comprehensive book by Ed Freitas. From initial setup to crafting intricate UIs and seamless backend integrations, this resource covers it all. [Unlock the Essentials of Flutter] (<https://www.syncfusion.com/ebooks/flutter-succinctly>)
 - [13]. "Dart Demystified": Unravel the intricacies of Dart with this detailed specification, a must-have for any Flutter aficionado. Delve into the language's syntax, semantics, and rich feature set to supercharge your development journey. [Discover Dart Demystified] (<https://dart.dev/guides/language/spec>).
 - [14]. "Flutter Unleashed" by Eric Windmill: Embark on a practical journey with Flutter through this hands-on guide by Eric Windmill. From concept to creation, master the art of building real-world mobile apps using Flutter and Dart. [Get Started with Flutter Unleashed] (<https://www.manning.com/books/flutter-in-action>)
 - [15]. "Mastering Flutter Foundations" by Rap Payne, Seth Ladd, and Eric Windmill: Lay the groundwork for your Flutter adventure with this beginner-friendly masterpiece. From UI craftsmanship to seamless navigation, this book is your roadmap to app development success. [Begin Your Journey with Mastering Flutter Foundations] (<https://www.apress.com/gp/book/9781484251808>)
 - [16]. "Flutter Unleashed" by Eric Windmill: Embark on a practical journey with Flutter through this hands-on guide by Eric Windmill. From concept to creation, master the art of building real-world mobile apps using Flutter and Dart. [Get Started with Flutter Unleashed] (<https://www.manning.com/books/flutter-in-action>)
 - [17]. "Mastering Flutter Foundations" by Rap Payne, Seth Ladd, and Eric Windmill: Lay the groundwork for your Flutter adventure with this beginner-friendly masterpiece. From UI craftsmanship to seamless navigation, this book is your roadmap to app development success. [Begin Your Journey with Mastering Flutter Foundations] (<https://www.apress.com/gp/book/9781484251808>)
 - [18]. Jabangwe, R., Edison, H., & Duc, A. N. (2018). Exploring software engineering methodologies for mobile application development: A comprehensive literature review. *Journal of Systems and Software*, 145, 98–111. <https://doi.org/10.1016/j.jss.2018.08.028>
 - [19]. Busch, Z. (2019, June 26). The six phases of the mobile app development process. Retrieved May 11, 2021, from <https://learn.g2.com/mobile-development-lifecycle>.
 - [20]. Wikipedia contributors. (2021, April 23). Overview of website wireframing. Retrieved May 11, 2021, from https://en.wikipedia.org/wiki/Website_wireframe
 - [21]. Blair, I. (n.d.). A guide to the eight stages of the mobile app development process. Retrieved May 11, 2021, from <https://buildfire.com/understanding-mobile-app-development-lifecycle/>
 - [22]. Mobile Backend-as-a-Service - Four essential BaaS platforms to consider. (2019, July 26). Retrieved May 11, 2021, from <https://waracle.com/blog/mbaas/how-to-choose-the-right-backends-a-service-baas-platform/>



Prabhav Shukla is pursuing his MCA degree from Ajay Kumar Garg Engineering College, Ghaziabad. He obtained his B.Sc. from the University of Delhi, in 2020. His research interests are in the fields of Mobile App Development Industry using Flutter.



Neha Tyagi is pursuing her MCA degree from Ajay Kumar Garg Engineering College, Ghaziabad. She obtained her BCA from Chaudhary Charan Singh University, Meerut in 2021. Her research interests include the Mobile App Development Industry using Flutter.



Deepanshu is pursuing an MCA degree as a Student of AKGEC, Ghaziabad. He obtained his B.Sc. from Chaudhary Charan Singh University, Meerut in 2021. His research interests include the Mobile App Development Industry using Flutter.



Medhavi Agarwal is pursuing her MCA degree from Ajay Kumar Garg Engineering College, Ghaziabad. She obtained her BCA from Chaudhary Charan Singh University, Meerut in 2022. Her research interests include the Mobile App Development Industry using Flutter.



Ms. Shrutijain

Assistant Professor is working in the MCA department at Ajay Kumar Garg Engineering College, Ghaziabad. She has completed her MCA from Guru Gobind Singh Indraprastha University, Delhi.