WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Impact of Flutter Technology in Software Development

Naveen Kumar Gupta [a]

[a] *Assistant Professor, Department of Computer Application, Dau Dayal Institute of Vocational Education, Dr. Bhimrao Ambedkar University, Agra, India*

DOI : https://doi.org/10.55248/gengpi.5.0724.1928

A B S T R A C T

Flutter is an open-source UI toolkit developed by Google that enables the creation of natively compiled applications for mobile, web, and desktop platforms using a single codebase. Leveraging the Dart language, a comprehensive collection of widgets, and a high-performance rendering engine, Flutter offers a flexible and efficient development process. It improves cross-platform development by providing a consistent user experience, accelerating development cycles, and ensuring strong performance. Its capability to integrate with platform-specific code while maintaining a unified codebase has made it a significant force in software development, favoured for building modern, high-quality applications.

Keywords: Flutter Technology, Software Development, Programming

Introduction

Flutter is an open-source UI (User Interface) software development kit created by Google. It is used to develop applications for mobile, web, and desktop from a single codebase. Flutter allows developers to write code once and deploy it across multiple platforms, reducing development time and effort. Flutter provides a hot reload feature that allows developers to instantly see changes made to the code without restarting the application, making the development process faster and more efficient. Flutter uses a declarative UI programming model, allowing developers to create beautiful and highly customizable user interfaces using widgets. Flutter apps are compiled directly to native machine code, which ensures high performance and smooth animations. In which developers can write code once and deploy it across multiple platforms, including iOS, Android, web, and desktop. Flutter has a rich ecosystem of packages and plugins that extend its functionality, allowing developers to add features like Firebase integration, in-app purchases, and more. Overall, Flutter is a popular choice for developers looking to build cross-platform applications with a fast development cycle and native performance.

Review of Literature

Sharma, P. et al. (2024) inspects Cross-platform mobile app development is essential today, as developers face the choice of creating individual apps for different operating systems or opting for less efficient solutions to ensure portability. Flutter, an open-source SDK from Google, provides a high-performance option for developing mobile apps for both iOS and Android using a single code base. It offers Just-in-Time (JIT) and Ahead-of-Time (AOT) compilation for effective code execution and features a "hot reload" capability that enables rapid code updates and testing. With support for GPU rendering and native ARM code, Flutter is a robust solution for cross-platform app development.

Marimuthu, K. et al. (2023) evaluates the communication and information-sharing needs of educational institutions; we created a versatile app that works across mobile phones, laptops, and tablets. Utilizing Flutter and Dart with Firebase integration, the app offers features including attendance management, exam schedules, lecture notes, fee information, event updates, and online tests. It comprises both a mobile application and a web-based platform, with data synchronized via a unified Firebase server. These tools are aimed at enhancing communication efficiency within the institution and are restricted to authorized users to ensure privacy and security.

Mohammed, D. Y., & Ameen, S. Y. (2022) analyze third-party libraries are widely used to expedite software development, particularly for mobile apps. These libraries facilitate code sharing among developers, making their work reusable. This paper presents the development of a taxi service library for Android and iOS using Dart, Dio, and Retrofit. The aim is to streamline and speed up software development by offering an interface for accessing platform-specific functionalities. Flutter, an open-source SDK, allows for the creation of high-performance, reliable mobile applications from a single code base for both iOS and Android. Sharing these developments with the developer community is crucial for promoting collaboration and innovation. Additionally, Flutter provides GPU rendering and UI capabilities with native ARM code, enhancing app development efficiency.

Bhagat, S. A. et al. (2022) describe the COVID-19 pandemic has boosted the popularity of mobile applications, leading to heightened competition among development frameworks. Developers are increasingly prioritizing ease of development, and Flutter meets this need by providing a platform that facilitates app development for both iOS and Android, thereby lowering costs and simplifying the process. As an open-source SDK, Flutter enables the development of high-performance, dependable mobile applications for both operating systems. This paper outlines the benefits of using Flutter in comparison to other app development platforms.

Haider, A. (2021) evaluate Flutter, a cross-platform technology, from the perspective of users, particularly focusing on critical elements such as startup time and application size; multiple sample applications were created for this purpose, and user perception was assessed through the UEQ method. The results indicated that if application performance is a primary concern for users, Flutter might not be the optimal choice, given that its performance metrics did not always meet high expectations; however, the user perception study showed that there were no substantial differences between Flutter and native applications, whether on Android or iOS platforms. Thus, despite the performance concerns, Flutter can still be considered a viable alternative to native Android applications from the standpoint of user experience, especially when comparing user perceptions across different platforms.

Fahnbulleh, M. K., & Shuo, X. (2021) investigates Flutter, Google's cross-platform UI toolkit, focusing on its performance and user perception compared to native applications. An experiment measured CPU performance between Flutter apps and native apps built with Kotlin (Android Studio) and Swift (Xcode). Additionally, a survey evaluated user perceptions of appearance and animations, supported by a literature review. The findings reveal that Flutter can rival native apps in CPU performance but falls short in animation quality. Flutter's simplicity and reduced code requirements make it ideal for small to medium-sized applications, with potential for future enhancements in animation. Further research is suggested to confirm these results.

Tashildar, A. et al. (2020) examines Cross-platform mobile app development is essential in today's tech landscape, as developers often need to either create separate versions of an app for different operating systems or compromise on performance for portability. Flutter, an open-source SDK, offers a solution by enabling the development of high-performance, reliable apps for both iOS and Android. It features Just-in-Time (JIT) and Ahead-of-Time (AOT) compilation for efficient code execution, and a "hot reload" capability that allows developers to quickly update code and see changes instantly. With support for GPU rendering and UI using native ARM code, Flutter is a powerful option for cross-platform development.

Olsson, M. (2020) explores how Flutter, a Google-developed open-source UI toolkit, compares to native applications in terms of performance and user experience. Flutter facilitates cross-platform development using a single code base, whereas native apps created with Kotlin (Android Studio) and Swift (Xcode) are generally seen as superior in performance and mobile behavior. The study includes an experiment assessing CPU performance, a survey evaluating user perceptions of design and animations, and a review of existing literature. Findings suggest that while Flutter performs well in CPU benchmarks, it falls short in animation quality. Due to its simplicity and reduced coding needs, Flutter is ideal for small to medium-sized applications, though it has room for improvement in animation. Additional research is required to confirm and expand on these results.

Research Objectives

- To analysis the impact of flutter technology in software development.

Workflow

Flutter has revolutionized app development by facilitating efficient cross-platform development, speeding up development cycles, and increasing productivity. It enables developers to write a single codebase for multiple platforms, including iOS, android, web, and desktop, minimizing the need for separate teams for each platform. With its widget-based architecture, Flutter ensures a consistent user experience across platforms, allowing for cohesive and customizable user interfaces. Apps developed with Flutter are compiled to native ARM code, which ensures high performance, while the Skia graphics engine provides smooth animations. Flutter's growing support for web and desktop applications makes it a versatile tool that influences new development practices and tools. Its expanding ecosystem and community, along with its adoption by major companies, underscore its importance in modern app development.Flutter workflow provides a general outline of the steps involved in developing a Flutter application, but the specific details may vary depending on the project requirements, team preferences, and development methodologies. Flutter technology typically involves several stages, from project setup to deployment. Here's an overview of the typical workflow for developing a Flutter application:

Environment Setup- Install Flutter SDK: Download and install the Flutter SDK from the official website (https://flutter.dev/docs/get-started/install).Set up an Integrated Development Environment (IDE): Popular options include Android Studio, Visual Studio Code, and IntelliJ IDEA, each with Flutter plugin support.

Create a New Project: Use the `flutter create` command or IDE's built-in project creation wizard to create a new Flutter project. Specify project details such as name, organization, and Flutter SDK path.

Development: Develop the application logic and user interface using the Dart programming language and Flutter framework. Use Flutter's widget library to build the UI components, manage state, handle user input, and implement app functionality.

Testing: Create unit tests for individual functions, classes, and modules using Flutter's built-in testing framework or third-party testing libraries. Test UI components and widget behaviour using Flutter's widget testing framework. Execute tests locally or on continuous integration (CI) platforms to ensure code correctness and reliability.

Debugging and Iteration: Use Flutter's hot reload feature to instantly see changes made to the code reflected in the running app.Debug issues and errors using Flutter DevTools, a suite of tools for debugging, profiling, and inspecting Flutter apps.

Optimization: Optimize app performance: Identify and address performance bottlenecks, such as slow rendering or excessive memory usage, to improve app responsiveness and efficiency. Profile app: Use Flutter DevTools or platform-specific profiling tools to analyse app performance and identify areas for optimization.

Localization and Internationalization:- Add support for multiple languages and locales using Flutter's built-in localization and internationalization features. Define localized strings, format dates, numbers, and currencies based on the user's locale preferences.

Integration: - Integrate third-party packages: Add functionality to the app by integrating third-party packages from the pub.dev repository. Integrate platform-specific code: Use platform channels to interact with platform-specific APIs and features not directly supported by Flutter.

Deployment:- Generate platform-specific builds: Use Flutter's build commands or IDE's built-in tools to generate APK (Android) and IPA (iOS) files for distribution. Publish app: Submit the app to the Google Play Store (Android) or Apple App Store (iOS) for review and publication. Optionally, deploy the app to other platforms such as web or desktop using Flutter's experimental support for these platforms.

Maintenance and Updates:- Monitor app performance and user feedback. Regularly update the app with new features, bug fixes, and improvements based on user feedback and market trends.

Flutter Architecture with widgets-

Flutter’s architecture is designed for high performance and cross-platform compatibility. By using a combination of the Dart framework, Flutter engine, and a hierarchical widget system, Flutter enables developers to build rich and responsive user interfaces. Its integration with platform-specific code ensures that it can leverage native capabilities while maintaining a single codebase for multiple platforms.

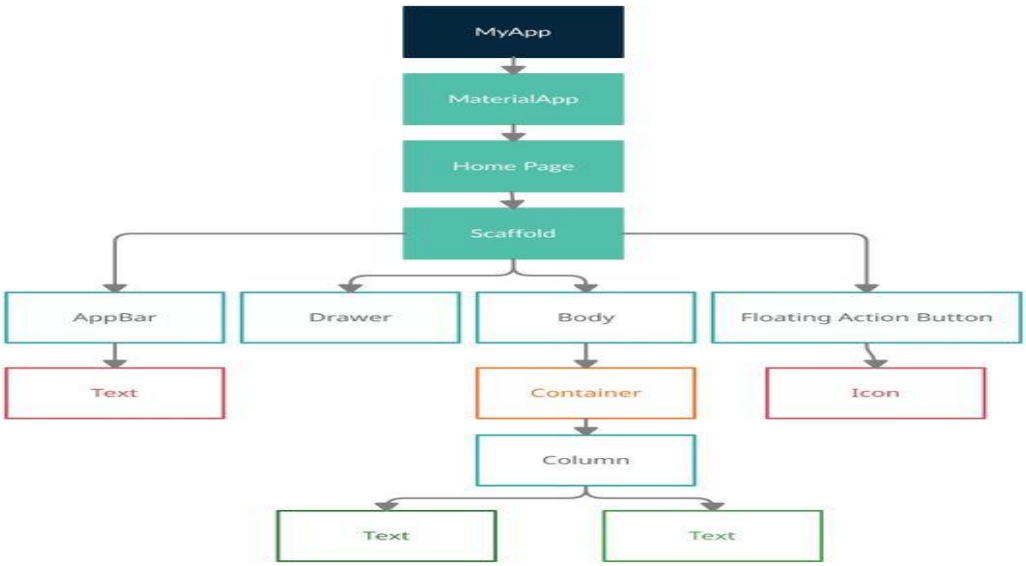

Fig. 1- Flutter App Architecture with Widgets

Objective of Flutter Technology-

Flutter’s main objective is to deliver a contemporary, efficient, and cross-platform framework for developing high-quality applications for mobile, web, and desktop. It allows code to be reused across various platforms, providing swift and native performance through direct machine code compilation. Flutter features a diverse collection of customizable widgets and a versatile UI framework for creating attractive and user-friendly interfaces. Its notable features include hot reload for fast development, a dynamic community and ecosystem for support, and scalability for projects of any scale. Ultimately, Flutter seeks to advance innovation and excellence in software development by enhancing efficiency, performance, and developer productivity.

Conclusion

Flutter is a high-performance framework that supports rapid development and offers a wealth of features, making it an excellent choice for future app development. It guarantees a consistent user experience across multiple platforms and benefits from strong community support and backing from Google. Flutter's adaptability and continuous improvements make it a top choice for building modern, responsive, and visually appealing applications,

ensuring its lasting value in the developer's toolkit. Flutter's architecture is designed to be both comprehensive and efficient, enabling developers to build high-performance, cross-platform applications.

References

Fahnbulleh, M. K., & Shuobo, X. (2021). Examining the Usage of Flutter to Design a Yacht in 3D. *International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)*, *8*(1), 1-11.

Tashildar, A., Shah, N., Gala, R., Giri, T., & Chavhan, P. (2020). Application development using flutter. *International Research Journal of Modernization in Engineering Technology and Science*, *2*(8), 1262-1266.

Olsson, M. (2020). A Comparison of Performance and Looks between Flutter and Native Applications: When to prefer Flutter over native in mobile application development.

Haider, A. (2021). Evaluation of cross-platform technology Flutter from the user's perspective.

Mohammed, D. Y., & Ameen, S. Y. (2022). Developing Cross-Platform Library Using Flutter. *European Journal of Engineering and Technology Research. https://doi. org/10.24018/ejeng*, *2*.

Bhagat, S. A. (2022). Review on Mobile Application Development Based on Flutter Platform. *International Journal for Research in Applied Science and Engineering Technology*, *10*(1), 803-809.

Marimuthu, K., Panneerselvam, A., Selvaraj, S., Venkatesan, L. P., & Sivaganesan, V. (2023). Android based college app using flutter dart. *Green Intelligent Systems and Applications*, *3*(2), 69-85.

Sharma, P., Shrivastava, V., Pandey, A., & Pathak, V. App Development using Flutter Technology.

Boukhary, S., & Colmenares, E. (2019, December). A clean approach to flutter development through the flutter clean architecture package. In *2019 international conference on computational science and computational intelligence (CSCI)* (pp. 1115-1120). IEEE.

Payne, R., & Payne, R. (2019). Developing in flutter. *Beginning App Development with Flutter: Create Cross-Platform Mobile Apps*, 9-27.

Mamoun, R., Nasor, M., & Abulikailik, S. H. (2021, February). Design and development of mobile healthcare application prototype using flutter. In *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)* (pp. 1-6). IEEE.

Sharma, S., Khare, S., Unival, V., & Verma, S. (2022, October). Hybrid Development in Flutter and its Widgits. In *2022 International Conference on Cyber Resilience (ICCR)* (pp. 1-4). IEEE.

Fentaw, A. E. (2020). *Cross platform mobile application development: a comparison study of React Native Vs Flutter* (Master's thesis).

Kuzmin, N., Ignatiev, K., & Grafov, D. (2020). Experience of developing a mobile application using flutter. In *Information Science and Applications: ICISA 2019* (pp. 571-575). Springer Singapore.