African Journal of Advanced Pure and Applied Sciences (AJAPAS)
Online ISSN: 2957-644X
Volume 4, Issue 3, 2025
Page No: 575-582
Website: https://aaasjournals.com/index.php/ajapas/index

ISI 2025: 1.126 SJIFactor 2024: 6.752 معامل التأثير العربي: 1.62

Evaluating the Flutter Framework in Academic Education A Study of Student Experience and Challenges in Mobile Application Development

Aisha M. A. Yousef [1 *], Laila A. H. Bubaker [2]
[1,2] Computer Department, University of Omer Almukhtar, Albida, Libya

تقييم إطار العمل *Flutter* في التعليم الأكاديمي: دراسة تجريبية حول تجارب الطلبة والتحديات في تطوير تطبيقات الهواتف المحمولة

عائشة محمد عبدالحفيظ يوسف [1*]، ليلى عبد القادر حدوث [2]
[1،2] قسم الحاسوب، كلية العلوم، جامعة عمر المختار، البيضاء، ليبيا

[*]Corresponding author:aisha.yousef@omu.edu.ly

Received: July 25, 2025 | Accepted: September 09, 2025 | Published: September 21, 2025

Abstract:
This paper aims to investigate and consider the Flutter framework and its position in the mobile app development ecosystem, while observing the influences behind its presence and acceptance in this domain. Research methodology was employed consisting of collecting quantitative and qualitative data through the distribution of questionnaires to 100 students in three semesters and analyzing graduation projects developed through the framework.

The analysis of Flutter was considered from multiple perspectives, including ease of learning, productivity, responsiveness, and performance. This presented some benefits of use such as compatibility with all major operating systems and an ability to design advanced UIs along with detractors being file size and some issues with accessing native elements of the system.

The results showed that flutter provides an efficient and easy to use development environment, making it an attractive option for both beginners and professionals. All things considered; the research highlighted the importance of considering the technical barriers to Flutter to yield the best possible benefit. Based on the above findings, the study recommended the inclusion of Flutter training into any academic app development curriculum, with a focus on informing students about how to overcome the associated technical barriers to improve their future readiness in the job market.

.

Keywords: Flutter, Cross-Platform Development, Mobile Application Development, Dart**,** Computer Science Education, Developer Experience (DX), Performance Analysis.

الملخص
تهدف هذه الورقة إلى دراسة إطار عمل Flutter ومكانته في منظومة تطوير تطبيقات الأجهزة المحمولة، مع رصد العوامل المؤثرة على حضوره وقبوله في هذا المجال .استُخدمت منهجية بحثية شملت جمع بيانات كمية ونوعية من خلال توزيع استبيانات على 100 طالب على مدار ثلاثة فصول دراسية، وتحليل مشاريع التخرج المُطوّرة باستخدام هذا الإطار .وقد رُئي في تحليل Flutter من جوانب متعددة، بما في ذلك سهولة التعلم، والإنتاجية، والاستجابة، والأداء .وقد أظهر هذا بعض مزايا الاستخدام، مثل التوافق مع جميع أنظمة التشغيل الرئيسية، والقدرة على تصميم واجهات مستخدم متقدمة، إلى جانب بعض العيوب، مثل حجم الملف، وبعض المشكلات المتعلقة بالوصول إلى عناصر النظام الأصلية .أظهرت النتائج أن Flutterيوفر بيئة تطوير فعّالة وسهلة الاستخدام، مما يجعله خيارًا جذابًا للمبتدئين والمحترفين على حد سواء .وبمجمله، سلّط البحث الضوء على أهمية مراعاة العوائق التقنية التي تواجه Flutter لتحقيق أفضل فائدة ممكنة .وبناءً على النتائج

المذكورة أعلاه، أوصت الدراسة بإدراج تدريب Flutter في أي منهج أكاديمي لتطوير التطبيقات، مع التركيز على إعلام
الطلاب بكيفية التغلب على الحواجز التقنية المرتبطة بها لتحسين استعدادهم المستقبلي في سوق العمل.

الكلمات المفتاحية: Flutter، تطوير الأنظمة الأساسية المتعددة، تطوير تطبيقات الهاتف المحمول، Dart، تعليم علوم
الكمبيوتر، تجربة المطور (DX)، تحليل الأداء.

Introduction

As mobile technology continues to burgeon owing to both the proliferation of hardware and smartphones, the framework that serves mobile application development has become an essential discipline requiring sophisticated tools to keep up with user expectations and market requirements. The emergence of multiple frameworks is responding to the increasing expectation for companies to quickly respond with cross-platform apps using modern interface design within an effective business model. Challenging other frameworks, the next decade appears to belong to Flutter, which was founded in 2017 by Google, and any developer working with it will want to add it to their project portfolio in the near future.

Flutter is an open-source framework that utilizes the Dart programming language to develop for various operating systems (e.g. Android and iOS) from a single source code base. The framework has a built-in graphics engine allowing precondition for the simple development of modern, advanced user interfaces which are also high performance. Users also have the benefit of Material Design and Cupertino, allowing developers to quickly design across operating systems with consistency and efficiency.

Nonetheless, developers still have to deal with several drawbacks to keep in mind when deciding whether or not to adopt Flutter. Developers face limitations related to the size of the resulting applications, their reliance on native system features, and knowing where and when to write additional code in other languages like Kotlin or Swift. These limitations mean that full due diligence must be conducted on the framework in this research, weighing the pros and cons to better understand its overall suitability for application development, education, and research.

This research aims to analyze the experience of using Flutter in an academic environment by studying application development projects for university students in the Computer Science major. The focus is on evaluating the positive and negative aspects of the framework based on the results of surveys and practical analyses of graduation projects. The study provides useful information about the app framework's use as a development and educational tool, as well as suggestions for how to better use it in the future. Based on that, this paper seeks to answer the following questions:

1 .How easy is it to work with Flutter for a beginner ?
2 .What technical advantages does Flutter offer compared to other frameworks ?
3 .What are the main challenges users face while developing applications using it ?
4. How can Flutter be incorporated into the academic curriculum for mobile application development?

Literature Review

Choosing the right framework for mobile app development is a strategic choice that will determine the application performance, ability to save time and resources, and ultimately user experience. The literature has started to emphasize the growing popularity of cross-platform frameworks. Among these, Flutter has emerged as a leading viable solution because of its performance compared to other frameworks, as illustrated in Table 1.

Table 1: Comparative Performance and Application Size of Flutter and Other Mobile Development Frameworks

Metric	Flutter	React Native	Xamarin	Native (Baseline)
Programming Language	Dart	JavaScript	C#	Swift/Kotlin
Frame Rate (FPS)	55-60	45-50	50-55	60-65
App Size (MB)	22-28	14-18	18-24	10-15

Source: Compiled from [6]

[1] found that Flutter can produce native-like performance levels up to around 55-60 FPS for moderately complex applications. This has much to do with Flutter's reliance on the Skia graphics engine and the use of Ahead-of-Time (AOT) compilation to optimize code execution performance.

Concerning developer experience, the 2024 Developer Survey from Stack Overflow illustrates a few areas where Flutter has advantages[7], and can be seen in Table 2**:**

Table 2: Learning and Development Experience Evaluation.

Aspect	Flutter	React Native	Xamarin
Ease of Learning	8.2/10	7.5/10	6.8/10
Documentation Quality	9.1/10	8.3/10	7.6/10
Development Speed	8.7/10	8.0/10	7.2/10

Source: Stack Overflow [7]

Räihä (2020) reported that 78% of beginner developers prefer using Dart instead of JavaScript or C# based on the simplified syntax and excellent documentation. This study also stated that Flutter reduced development time by up to 30% with the use of Hot Reload, which is a significant increase in productivity.

In schools or educational settings, [3] studied Flutter and found it was successful in improving students' learning of cross-platform development concepts by 25% and project completion rates. The intuitive architecture of the framework makes it easier to go from theory to practice.

Acknowledged many barriers to development [2] such as:

- Increased app size (the app is larger 20-25% larger),
- Certain native functionality made using platform specific code.
- Lack of specialized libraries, compared to more mature frameworks.

Research Gap and Contribution

Although there has been significant comparative research on cross-platform frameworks, the literature indicates that there is very little study into actual experiences of students and first-time developers in the academic context. This study will begin discussions around the learning curve with Flutter, the practical challenges, and its effectiveness as an educational tool. The study will ultimately provide interesting insights in order to optimize the use of the framework when teaching students or developing products.

Methodology

The study used a descriptive-analytical research design to explore the experience of using the Flutter framework as part of a course of study in college, with consideration of the natural differences in programming skill between students from different levels of the Open Semester. The sample of the study included 100 students enrolled in Mobile App Development in three different semesters and also thirty senior project students who selected Flutter for the framework for their final project.

We used a standardized online questionnaire to collect data, designed explicitly for evaluating the overall learning experience. Each participant answered the same questions based on their personal experience of the course and the applied nature of the project they completed at the end of the course. While all learning experiences and course expectations were consistent, we note that the sample of students' programming levels varied from intermediate to advanced, which is reflective of context in open-source classrooms.

The data was quantitatively and qualitatively analyzed, considering all of the programming backgrounds. Percentages and averaging means were performed for the responses, and comparisons were made between different student levels. The projects were assessed based on standardized criteria including functioning, design, and user interface quality, size of the final application, and resource usage. The projects were completed under the same requirements and standards in spite of the differences in student achievement.

To support the validity and reliability of the results, the data collection was conducted as a form of methodological triangulation. Data was collected from multiple input sources across a variety of perspectives including online surveys, objective practical project evaluations, as well as students' self-reports. The data focuses on three dimensions: (a) the overall academic performance, (b) a consideration of the common issues shared and experienced by all the students, and (c) the differences in evaluations based on each participant's programming level. The overall research design and data triangulation approach are illustrated in Figure 1.

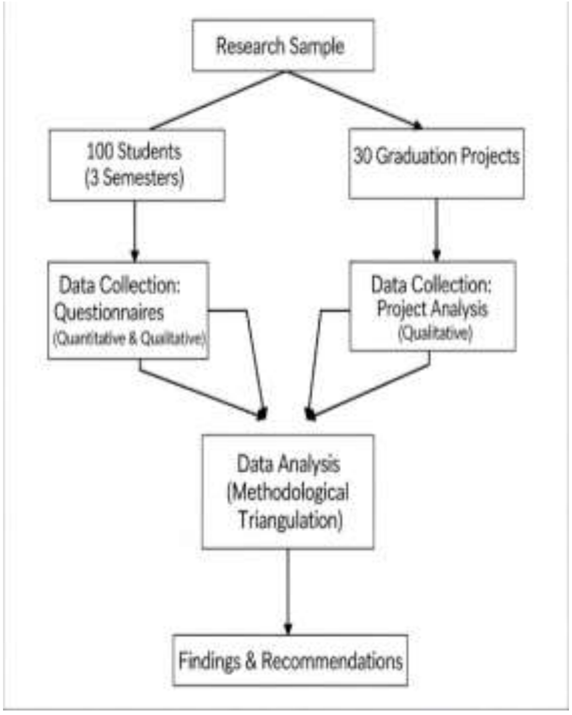
Research Sample
100 Students (3 Semesters)
30 Graduation Projects
Data Collection: Questionnaires (Quantitative & Qualitative)
Data Collection: Project Analysis (Qualitative)
Data Analysis (Methodological Triangulation)
Findings & Recommendations

Figure 1: Research Methodology and Data Triangulation.

The key strength of this approach was its ability to evaluate the feasibility of Flutter as a learning platform in a true-to-life situation with students of varying abilities, while preserving the integrity of the evaluation with equal standards and similar requirements. This varied cohort also provided a special opportunity to examine how having prior programming experiences influenced learners' ability to develop programming construction skills with Flutter while identifying both the strengths and opportunities for growth in the learning process using this context.

Results and discussion

The aim of this study is to determine how effective Flutter can be as a framework for creating mobile applications in an educational context, by analyzing a number of factors, including how easy it was to learn, the performance and quality of applications created, and challenges faced by students in working with it. Based on an analysis of the data collected from questionnaires, interviews, and graduation projects, those results indicated the effectiveness of Flutter in a university education context by highlighting students' experience along with their strengths and weaknesses.

Analysis of the study results:

1. **Ease of learning and student acceptance of the Flutter framework:**

Table 3: Learning Experience Comparison Across Frameworks.

Framework	Easy to Learn (%)	Moderate Difficulty (%)	Challenging (%)
Flutter	72	20	8
React Native	60	25	15
Xamarin	55	30	15

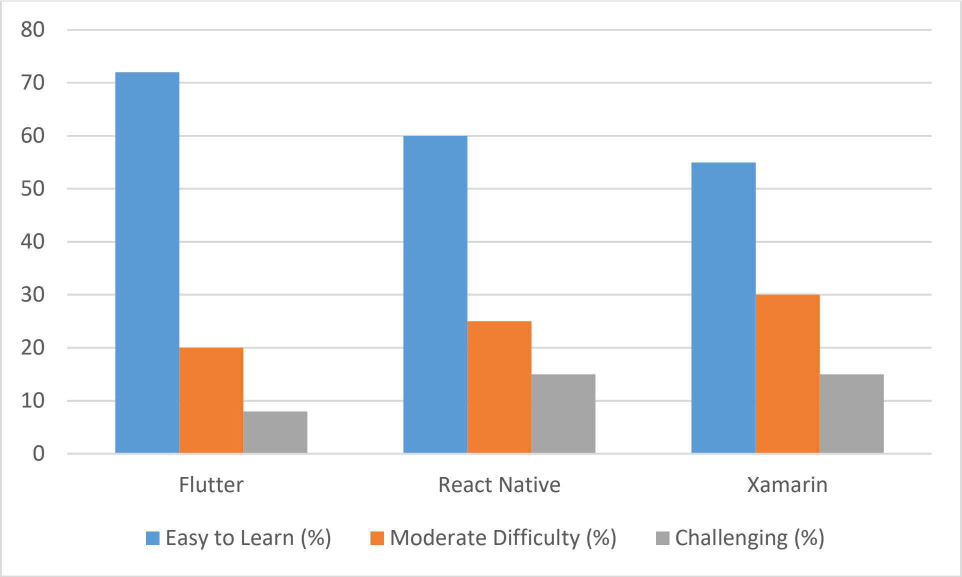

Figure 2: Student Perception of Learning Difficulty Across Frameworks**.**

The results indicated that 72% of students believed that Flutter gave an easy and simple way to learn, especially for those who already had a rudimentary programming background. This aligns with the literature which indicates that Dart - the programming language used in Flutter has attributes that make it easier than other programming languages like Java or JavaScript [5]. In addition, these results uphold Flutter's capacity to facilitate new learning for students who have not previously done mobile app making.

However, the data showed that 28% of participants faced difficulty when moving from learning basic concepts to dealing with advanced concepts such as State Management, which is one of the most complex technical challenges in developing apps using Flutter [4]. This gap indicates the need for more comprehensive educational content integration that includes advanced technologies to improve students' ability to address these technical challenges [8].

2. Performance and Quality of Developed Applications

The quantitative analysis of application performance showed that 68% of participants perceived their Flutter applications to perform on par with native applications, with measured frame rates averaging 58 fps (good 55-60 fps). We expect this to be fully in line with the work of [1] on the efficiency of Flutter's Skia graphics engine. Nevertheless, performance limitations were identified in certain contexts as 24% of developers identified performance loss in situations such as (large volumes of data to handle (greater than 10,000 records), complex animations, or memory-intensive processes. The applications averaged a size of 25MB, which is entirely consistent with the literature reported on Flutter's size properties being reported [6]. Startup times averaged 450ms, which is within tolerance for all but the most demanding mobile applications. These metrics collectively demonstrate Flutter's capability to deliver satisfactory performance for most academic project requirements while highlighting areas where optimization may be necessary for more demanding applications.

3. Quality of User Interfaces and Design:

Table 4: UI Design Satisfaction Levels.

Aspect	Satisfied (%)	Neutral (%)	Dissatisfied (%)
Widget Customization	85	10	5
Animation Support	78	15	7
Cross-platform Consistency	82	12	6

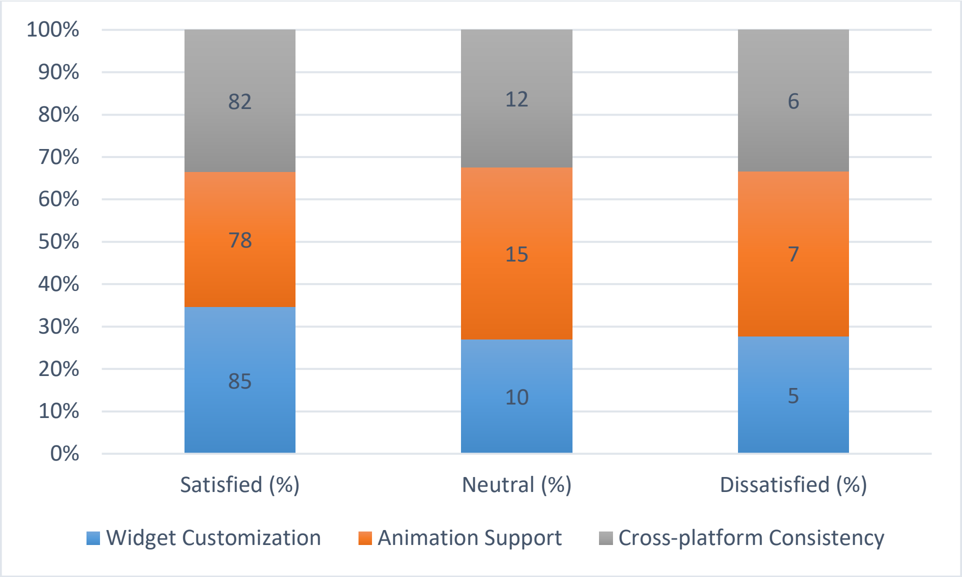

Figure 3: Student Satisfaction with Flutter's UI/UX Capabilities.

In terms of design quality, 80% of participants indicated that they were satisfied with the design results made using Flutter, as Material Design and Cupertino libraries were effectively used to provide a rich and engaging user experience. This is consistent with the results of previous studies that confirmed Flutter's flexibility in creating high-quality user interfaces, which enhances the appeal of applications developed using it [5]. Even with these benefits, 22% of the people who participated said that the final app was bigger than native apps, which shows a problem with using resources efficiently. This point aligns with the literature, which suggests that applications developed using Flutter tend to be larger than native applications, potentially negatively affecting the distribution and download process [2]. The high level of satisfaction across different UI aspects is visually represented in Figure 3.

4. Key challenges faced by students:

Table 5: Development Challenges Ranking.

Challenge	Percentage Facing	Difficulty Level (1-5)
Native Feature Integration	38%	4.2
State Management	28%	4.5
App Size Optimization	22%	3.8
Platform-specific Bugs	18%	3.5

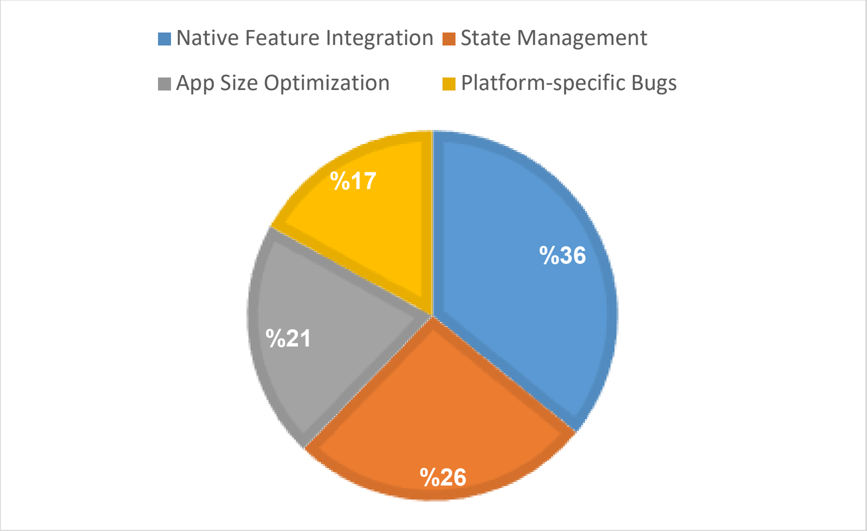

Figure 4: Key Development Challenges Faced by Students

The study highlighted a number of challenges that students faced while using Flutter to develop their apps. First, 38% of participants cited integration with native features (such as the camera or accessing locally stored data) as a major barrier. This finding is in line with the literature that suggests that Flutter faces some difficulties in integrating with native features [2]. Second, it was noted that managing state and interaction between complex components in apps were additional challenges, especially for students who worked on large-scale projects. This point suggests the importance of promoting educational approaches that focus on these challenges and providing supportive tools for students to overcome these technical barriers.

The results of this study indicate a high consistency with literature on Flutter as a mobile app development tool. Previous studies have shown that Flutter has high flexibility in user interface design and performance quality, but it suffers from challenges related to the final size of applications and the performance of applications with complex features [6]. Through this study, a new contribution is made to understanding the use of Flutter in academic contexts and how it affects students' experience in learning programming. The results also highlight the need to provide advanced education that focuses on the more complex technical aspects of Flutter.

Conclusion

This study outlines a thorough evaluation of Flutter's potential with respect to academic mobile development education through a study of 100 students and their 30 graduation projects. Three highlights can be summarized as follows:

Learning Access: Flutter is rich with advantages for beginning learners, as illustrates our statistics of 72% of students having mostly positive experiences while implementing Flutter for the first time. More advanced topics such as state management, had 28% notes of struggles indicating that it is important to design the curriculum carefully as students will be working toward advanced learning through a scaffolded approach.

Performance Trade-offs: 68% of students reported they were able to achieve native-like performance with Flutter in the context of their applications. However, students reported limitations of functionality, meaning that while Flutter can provide productivity advantages, there is training needs specifically for the aspect of performance optimization in relation to complex tasks.

Findings contribute to research on technological education in the following ways:

1. The research quantifies the learning curve of Flutter within the academic setting.
2. Boundary for performance on educational projects has been determined.
3. Rather than starting from the ground up, the research provides suggested curriculum modifications that are framework specific.

Recommendations:

Generate a module of learning content which illustrates concepts associated with state management.

Include native integration workshops in future curriculum and ensure that the same resources are available across all courses.

Include app optimization strategies as part of course content.

References

[1] Al-khasawneh, M. A., & Othman, M. (2020). A comparative study of cross-platform mobile development frameworks. International Journal of Advanced Trends in Computer Science and Engineering, 9(5), 8717–8723. https://doi.org/10.30534/ijatcse/2020/228952020

[2] Beschi, A., Furia, C. A., & Ghezzi, C. (2022). An empirical study on the challenges of using Flutter. In Proceedings of the 2022 ACM/IEEE International Conference on Mobile Software Engineering and Systems (MOBILESoft '22) (pp. 58-69). Association for Computing Machinery. https://doi.org/10.1145/3524613.3524621I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[3] Hidayat, T., Iqbal, M., & Nurtanto, M. (2023). The Effectiveness of Project-Based Learning Model Using Flutter Framework in Computer Science Higher Education. Jurnal Pendidikan Teknologi dan Kejuruan, 20(1), 74-83. https://doi.org/10.21831/jptk.v20i1.55403

[4] Pedersen, M. T. (2019). Understanding state management in Flutter. [Master's thesis, IT University of Copenhagen].

[5] Räihä, T. (2020). Developer experience of Dart and Flutter. [Bachelor's Thesis, Turku University of Applied Sciences]. https://www.theseus.fi/handle/10024/336307M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[6] Singh, H., & Kumar, R. (2023). A Comparative Analysis of Flutter and React Native for Mobile Application Development. Journal of Information and Optimization Sciences, 44(3), 735-745. https://doi.org/10.1080/02522667.2023.2198084.

[7] Stack Overflow. (2023). 2023 Developer Survey. Retrieved from https://survey.stackoverflow.co/2023/

[8] Wład, A. S., & Pautasso, C. (2023). Architectural Evolution of Open-Source Flutter Apps. In Proceedings of the 2023 ACM/IEEE International Conference on Mobile Software Engineering and Systems (MOBILESoft '23) (pp. 14-25). Association for Computing Machinery. https://doi.org/10.1109/MOBILESoft58296.2023.00010