Abstract

The mobile application has become an indispensable tool in various domains, evolving from information-based and productivity-based applications to encompass mobile games, banking, social media, and more. However, suboptimal performance can lead to a negative user experience, prompting users to uninstall the application and provide negative reviews. To address performance issues, the implementation of MVP-GetX is chosen due to the combination of the Model-View-Presenter (MVP) architecture pattern and the GetX state management, offering significant advantages in terms of performance and efficiency. Furthermore, the GetX state management has proven to prioritize performance and efficiency. The use of MVP-GetX as a mobile application development approach offers a dual advantage in improving performance efficiency and maximizing application responsiveness. Additionally, this choice is also considered due to its compatibility with the Flutter framework, allowing high-performance application development for various platforms and ensuring optimal results in delivering an exceptional user experience. This research compares two Flutter applications: one without a specific architecture pattern or state management (no-pattern), and another using the MVP pattern with GetX state management (MVP-GetX). The research focuses on measuring CPU and memory usage, which are aspects used to assess application performance, and by testing performance in different scenarios with various dataset sizes. The research results indicate that the MVP-GetX combination demonstrates consistent and efficient resource management in terms of memory usage, outperforming the no-pattern approach, especially with larger datasets. However, the CPU usage results show that both the no-pattern and MVP-GetX approaches are comparable and equally proficient in several testing scenarios across various dataset sizes.

Keywords: MVP, Flutter, Performance, GetX, Mobile App