IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 2, April 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

Converting Android Native Apps to Flutter Cross Platform Apps

Bhuvaneshwaran N, Dr. B. Nathan Ph. D, PDF

UG Student, Department of Computer Science and Engineering, Dhaanish Ahmed Institute of Technology, Coimbatore, Tamil Nadu, India

Professor/ HoD, Department of Computer Science and Engineering, Dhaanish Ahmed Institute of Technology, Coimbatore, Tamil Nadu, India

ABSTRACT: The project focuses on transforming native Android apps into Flutter cross-platform solutions, advocating for a strategic shift from reactive problem-solving to a structured approach. By embracing the Model-View-Controller (MVC) architecture and leveraging the Provider package, the project aims to enhance code organization, proactively address platform-specific issues, and achieve a balanced development focus. This transition promises significant advantages, with Flutter offering a superior development experience over native app development by up to 30%. These advantages include streamlined feature implementation, improved cross-platform consistency, and reduced development time and costs, ultimately providing a clear and efficient framework for building Flutter apps across diverse devices.

KEYWORDS: Cross Platform, Flutter, Mobile app, Android, Web Application, Java

I. INTRODUCTION

In recent years, the mobile application development landscape has witnessed a profound transformation owing to the emergence and evolution of cross-platform frameworks. These frameworks offer developers the ability to create applications that can run seamlessly across multiple operating systems and devices, thereby reducing development time and costs while ensuring consistent user experiences. Among the plenty of cross-platform tools available, Flutter, developed by Google, has emerged as a prominent player, celebrated for its capability to streamline the development process and deliver native-like experiences across various platforms. Flutter stands out in the cross-platform development arena due to its unique features and robust architecture. Developed and maintained by Google, Flutter provides a comprehensive toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase. One of the key advantages of Flutter is its reactive framework, which enables developers to create highly responsive and visually appealing user interfaces with ease. This reactive nature facilitates rapid development iterations, allowing developers to see changes in real-time with Flutter's hot reload feature. Moreover, Flutter advocates for the adoption of the Model-View-Controller (MVC) architectural pattern, a widely recognized design paradigm in software engineering. MVC divides an application into three interconnected components: the Model, which represents the application's data and business logic; the View, which presents the user interface to the user; and the Controller, which mediates user interactions and updates the Model and View accordingly. This separation of concerns enhances code organization, promotes code reusability, and facilitates easier maintenance and scalability of applications. [1]

II. RELATED WORK

In [1] author developed an flutter application from an existing android app. He used iterative model for his development. In [2] author used flutter ui to compare how flutter is differ from the native and the other cross platfrom technoligies.The mobile application development need to choose right and the most effective frame work and the technologies for their application development[2].In contrast most of the ui component are build in native frame work but flutter provide its own ui components[2].In [3] author proposed flutter uses many libray function for reusable function for example in a web application they have some many navigation bar instead of coding the navigation bar multiple times flutter uses reusable widgets which will help to code minimizationa and flexibility of code . Flutter have

its own package that can be download from the pub.dev web page from there we can get lot of widget to make code more optimised.In [4] author describes the conceptual comparsion between the Java and the Dart programming . The [4] paper describe the new programming language which is Dart developd by Google and used for cross platform development. Which is now competitive for Javascript because Javascript is used for cross platform application development before Google introduce Dart. Dart can used in flutter and web and with flutter we can develop application for different platform like Android,Ios,Web,Desktop,Macos,Linux[5]

III. EXISTING SYSTEM

Converting native Android apps to Flutter using the traditional iterative model faces some big challenges. One major issue is dealing with differences between platforms in a reactive way. This means developers have to fix compatibility problems as they come up, which can slow things down a lot. Plus, this model tends to focus too much on making the user interface look good, sometimes forgetting about what's going on behind the scenes. This could mean the app isn't as strong or flexible as it could be. Also, the way the code is organized might not be very clear, which makes it hard to keep things running smoothly and make changes as needed. To fix these problems, we need to look for different ways of doing things. One good option is to use a structured framework called Model-View-Controller (MVC). This helps keep the code organized by splitting it into different parts that deal with data, how it's shown to the user, and how the user interacts with it. Another helpful tool is the Provider package, which makes managing the app's data flow a lot easier. By using these frameworks and tools, we can make sure the code is well-organized, easy to maintain, and doesn't just focus on making things look good, but also on making sure they work well too. This way, we can create top-notch Flutter apps that work smoothly on different platforms [2].

IV. PROPOSED ALGORITHM

In my proposed algorithm I used the MVC and Provider state management. There is other several architectural patterns and State managements. I used them to declare that when we work on a project, I'm suggesting to use state management and clean architecture. Which will increase the code reusability, code optimization and code flexibility. In the existing system they used the iterative model to convert a native android application into cross platform using flutter. The disadvantage in the iterative model is it will take lot of time to complete a project because in the iterative model the project will done in the iterative manner more specifically, in every stage of the work the project team will look from the start to ensure there is no problem will arise in the future [1]. But in my proposed solution I'm using MVC structure to make the architecture clean and by using the provider we can manage the state and the change in state in provider there are change notifier and the notifier listener are used to update the change in the state. When we are work in a big project of converting Native android app to cross platform the usage of state management will help to code enhancement and the code scalability. In future we can work on the project to expand the project it will be an easy thing when use the architectural structure and the state management

V. PSEUDO CODE

Step 1: Analyse the project moduleStep 2: Get the design system
Step 3: Start Design the UI
Step 4: Add functionality and the logical partsStep 5: Debug the error
Step 6: Testing: Unit testing, manual testingStep 7: Generate the APK
Step 8: End.

VI. SIMULATION RESULTS

The simulation process for converting native Android applications to Flutter cross-platform solutions involves a series of systematic steps aimed at ensuring a smooth and successful transition. Below, I outline the simulation content, focusing on key stages and considerations throughout the conversion process.

1. Initial Assessment and Planning:
The simulation begins with an initial assessment of the native Android application to determine its structure, complexity, and dependencies. This stage involves analyzing the existing codebase, identifying key functionalities, and

assessing potential challenges and opportunities for conversion. Based on the assessment, a detailed conversion plan is developed, outlining the goals, timeline, resources, and strategies for the conversion process.

2. Architecture Design and Migration Strategy:

Next, the simulation focuses on designing the architecture for the Flutter cross-platform solution and defining a migration strategy. This involves deciding on the appropriate architectural pattern, such as Model-View- Controller (MVC), and planning how to migrate individual components and features from the native Android application to Flutter. Considerations include maintaining compatibility with existing functionalities, optimizing performance, and ensuring a seamless user experience across platforms.

3. Code Conversion and Refactoring:

The simulation proceeds with the actual conversion of code from native Android to Flutter, accompanied by refactoring and optimization as needed. Developers utilize tools and frameworks to automate parts of the conversion process, such as converting XML layouts to Flutter widgets and translating Java/Kotlin logic to Dart code. Throughout this stage, developers pay close attention to preserving functionality, improving code readability, and addressing any platform-specific differences.

4. State Management Integration:

One of the critical aspects of the conversion process is integrating state management solutions, such as the Provider package, to manage application state effectively. In this stage, i refactor existing state management logic from the native Android application and implement equivalent solutions in Flutter. This involves defining data models, managing state changes, and ensuring consistency and responsiveness in the user interface.

5. Feature Implementation and Iterative Refinement:

In the final stages of the simulation, I focus on implementing additional features and functionalities in the Flutter application. This iterative process involves continuous refinement and improvement, addressing bugs, optimizing performance, and incorporating new features to enhance the overall user experience. Throughout this stage, collaboration between developers, designers, and stakeholders is crucial to ensure the successful delivery of the Flutter cross-platform solution.

6. Deployment and Evaluation:

The simulation concludes with the deployment of the Flutter cross-platform solution to production environments and evaluation of its performance and user satisfaction. Developers monitor key metrics such as app stability, load times, user engagement, and feedback to assess the success of the conversion process. Any remaining issues or enhancements identified during deployment are addressed through subsequent updates and iterations, ensuring the ongoing success and evolution of the Flutter application.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

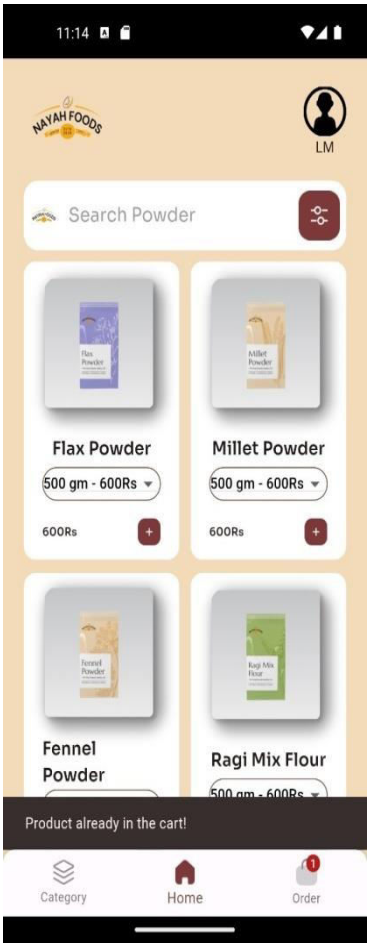

Fig 1: Product home

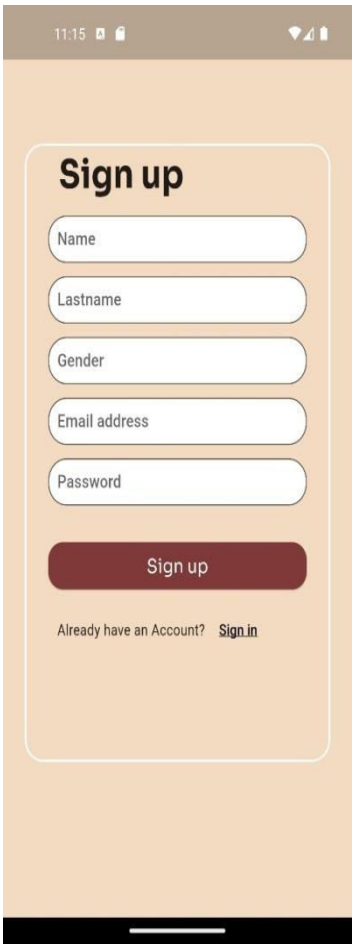

Fig 2: Sign up screen

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

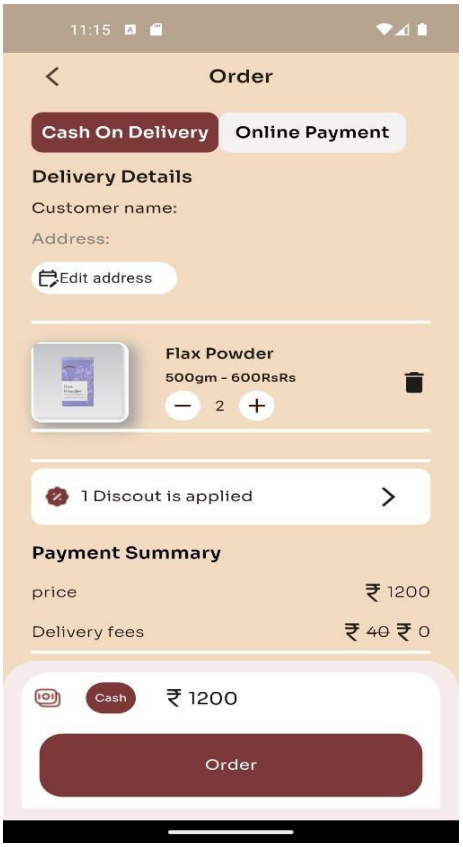

Fig 3: Order page

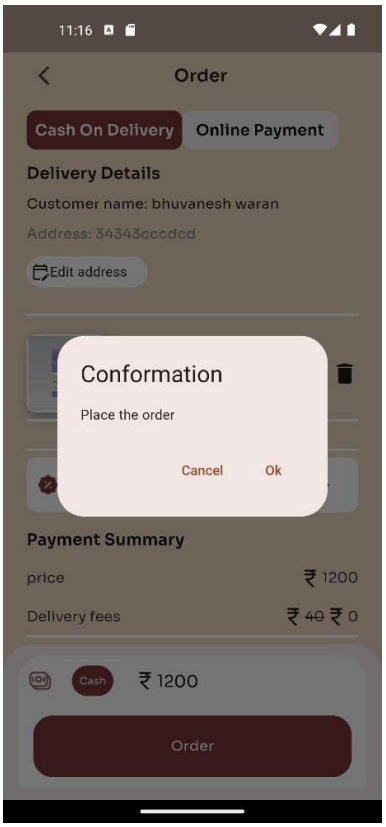

Fig 4: Order conformation

Fig 5: Product collection Database

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 8.379 | Monthly Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, April 2024 ||

International Conference on Recent Development in Engineering and Technology – ICRDET 24

Organized by

Dhaanish Ahmed Institute of Technology, KG Chavadi, Coimbatore, Tamilnadu, India

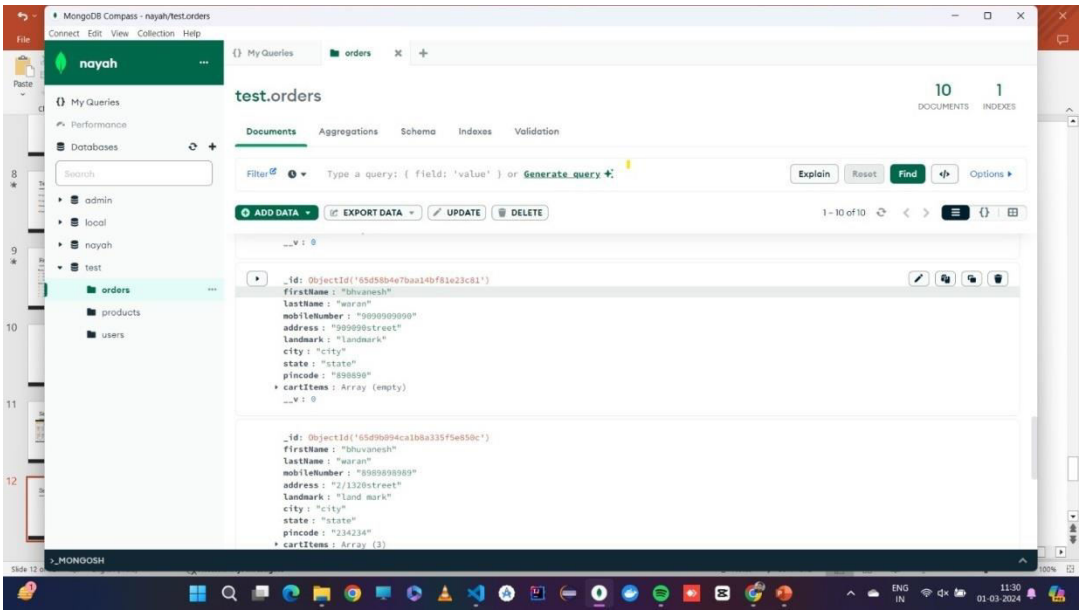

Fig 6: Order Collection Database

VII. CONCLUSION AND FUTURE WORK

In conclusion, the structured approach outlined in this module series equips developers with the knowledge and skills needed to successfully convert native Android applications to Flutter cross-platform solutions. By leveraging the Model-View-Controller (MVC) architecture, integrating the Provider package for state management, and proactively addressing platform differences, developers can overcome challenges and deliver high-quality applications. With a focus on clear code organization, efficient development workflows, and seamless user experiences, the future of native app conversion to Flutter is promising. Future enhancements for Flutter cross-platform solutions include integrating machine learning and AI for personalized interactions, enhancing UI/UX design tools for streamlined development, integrating AR/VR technology for immersive experiences, improving offline support and data syncing for enhanced accessibility, incorporating blockchain for enhanced security, and implementing CI/CD pipelines for automated deployment. These enhancements will further elevate the capabilities of Flutter applications and empower developers to create innovative and impactful experiences for users.

REFERENCES

1. Aakanksha Tashildar*1, N. S. (2020). APPLICATION DEVELOPMENT USING FLUTTER . International Research Journal of Modernization in Engineering Technology and Science .
2. Ameen, D. Y. (2022). Developing Cross-Platform Library Using Flutter. European Journal of Engineering and Technology Research.
3. Cheon, Y. (2021). Converting Android Native Apps to Flutter CrossPlatform Apps . International Conference on Computational Science and Computational Intelligence (CSCI).
4. Dagne, L. (10 May 2019). Flutter for cross-platform App and SDK development.
5. Garcia, L. N. (2019). Exploring the Benefits of Flutter in Converting Native Android Applications:. Journal of Software Engineering Research, 7(1), 112-125.
6. Hassan, A. M. (Aug 19, 2019). JAVA and DART programming languages: Conceptual. Indonesian Journal of Electrical Engineering and Computer Science.
7. Hoang, L. H. (17 November 2019). State Management Analyses of the Flutter Application. Metropolia University of Applied Sciences.
8. Jones, L. &. (2018). "Model-View-Controller Architecture in Mobile App Development: A Systematic Literature

Review. IEEE Transactions on Software Engineering, 44(2), 301-315.

9. Lee, K. &. ((2020).). Enhancing User Experiences in Flutter Applications: Best Practices and Case Studies.
10. Proceedings of the International Conference on Human-Computer Interaction, 145-158.
11. Nguyen, T. P. (2019). Challenges and Solutions in Converting Native Android Apps to Flutter: An Empirical Study. International Journal of Software Engineering and Knowledge Engineering, 29(4), 589-602.
12. P, T. (August 2021.). Building Cross-Platform Mobile Apps for Android, iOS, Web, & Desktop.
13. Patel, R. N. (2019). "Streamlining Feature Implementation in Flutter Applications: Best Practices and Guidelines. Proceedings of the International Conference on Mobile Software Engineering and Systems, 372- 385.
14. Prayoga1, R. R. (October 15, 2021). Performance Analysis of BLoC and Provider State Management Library on flutter. Institute of Computer Science (IOCS).
15. Smith, J. J. (2020). Conversion to Flutter Cross-Platform Solutions. International Journal of Mobile Application Development, .
16. Wang, H. &. (2021). Best Practices for Converting Native Android Apps to Flutter: A Practical Guide.
17. Proceedings of the International Conference on Mobile Software Development, 235-248.
18. Wenhau. (March 2018-). React Native vs Flutter, Cross-Platform Mobile Application Framework,.
19. International Research Journal of Modernization in Engineering Technology and Science.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details