International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Quick-Cart: A Cross-Platform Application

Prof. Roshani Parate[1], Prerna Divekar[2], Sakshi Gome[3], Anushka Gunjal[4]
[1] Professor, Department of Computer Engineering, Zeal College of Engineering and Research, Pune, Maharashtra, India
[2,3,4] Final Year Student, Department of Computer Engineering, Zeal College of Engineering and Research, Pune, Maharashtra, India

ABSTRACT
This paper presents the development and implementation of "Quick-cart," a cross-platform food delivery application built using Flutter and Firebase. The application aims to provide a seamless user experience across both iOS and Android platforms, leveraging Flutter for unified code management and Firebase for real-time database management, authentication, and cloud functions. The study focuses on the design, architecture, and performance of the application, highlighting the advantages of using Flutter and Firebase in modern mobile application development. Through extensive testing and user feedback, Quick-cart demonstrates its potential in the competitive food delivery market.

Keywords: Cross-Platform, Flutter, Firebase, Mobile Application, Food Delivery

I. INTRODUCTION

With the rapid growth of mobile technology, there is an increasing demand for cross-platform applications that offer consistent performance and user experience on both iOS and Android devices. This paper introduces "Quick-cart," a food delivery application developed using Flutter, Google's UI toolkit for building natively compiled applications, and Firebase, Google's mobile platform that provides backend services. The aim is to explore the efficiency, ease of development, and performance benefits provided by these technologies.
The food delivery market has seen significant growth in recent years, driven by changes in consumer behaviour and advancements in technology. Traditional approaches to mobile application development often involve maintaining separate codebases for iOS and Android, which can be time-consuming and costly. Cross-platform frameworks like Flutter offer a solution by enabling developers to write a single codebase that runs on multiple platforms without compromising performance or user experience.

II. LITERATURE SURVEY

The development of cross-platform mobile applications has gained substantial traction due to the demand for efficient and cost-effective solutions that cater to both iOS and Android platforms. Flutter, introduced by Google, stands out for its high performance and expressive UI design, allowing developers to create visually appealing applications with a single codebase. Its widget-based architecture ensures rapid development and consistent user experience across platforms.

Firebase, also by Google, complements Flutter with a comprehensive suite of backend services, including real-time database management, authentication, analytics, and cloud functions. The integration of Flutter and Firebase has been shown to enhance development speed, improve application performance, and increase user satisfaction. This literature review explores the advantages of using these technologies in tandem, highlighting their impact on modern mobile application development.

From our review of the literature, we have identified several gaps and opportunities for research in this field.

1. Cross-Platform Development
Cross-platform development enables developers to write code once and run it on multiple platforms, reducing development time and cost. Previous studies have shown the advantages of using frameworks like Flutter and React Native in developing such applications. Cross-platform tools often use a single programming language and a unified development environment, streamlining the development process.

2. Flutter

Flutter, introduced by Google, has gained popularity for its ability to deliver high-performance apps with a single codebase. Its widget-based architecture allows for fast and expressive UI development. Flutter's hot-reload feature enables developers to see changes in real time, significantly speeding up the development process. Research indicates that Flutter's performance is comparable to native applications, making it a viable option for production-level apps.

3. Firebase

Firebase offers a comprehensive suite of tools for app development, including real-time databases, authentication, analytics, and cloud functions. Its integration with Flutter simplifies the backend development process. Firebase Realtime Database provides a scalable, low-latency solution for syncing data across clients in real time. Firebase Authentication supports multiple authentication methods, including email/password, phone number, and social logins, enhancing the security and user experience of the application.

III. OBJECTIVES

The primary objectives of this study are:

1. To develop a functional food delivery application using Flutter and Firebase.
2. To evaluate the performance and user experience of the application on both iOS and Android platforms.
3. To identify the benefits and challenges associated with using Flutter and Firebase for cross-platform development.

IV. METHODOLOGY

4.1 System Design

The architecture of Quick-cart is divided into three main components: the user interface, backend services, and database management. Flutter is used for the front-end development, while Firebase handles the backend services and database. The system design focuses on modularity and scalability, ensuring that the application can handle increased user load and new feature integrations without significant redesign.

4.2 Development Tools

Flutter: For building the cross-platform application.
Firebase: For real-time database, authentication, and cloud functions.
Dart: The programming language used in Flutter.

4.3 Features of Quick-cart

User Authentication: Secure sign-up and login using Firebase Authentication.
Restaurant Listings: Dynamic listing of restaurants based on cuisine, ratings, etc.
Real-Time Order Tracking: Users can track their orders in real-time, from preparation to delivery if restaurant allows.
Payment Integration: Secure and seamless payment gateway integration supporting multiple payment methods from various restaurants ease.
Push Notifications: Notifications for order status updates and promotional offers.

4.4 Implementation

4.4.1 User Interface

The user interface of Quick-cart is designed using Flutter's rich set of pre-designed widgets, ensuring a responsive and intuitive experience. The application employs a clean and modern design, with easy navigation and accessibility features. Flutter's widget tree structure allows for efficient UI component reuse and customization.

4.4.2 Backend Integration

Firebase Realtime Database is used to store and sync user data, order details, and restaurant information. Firebase Cloud Functions are utilized to handle complex backend logic, such as calculating delivery times and processing payments. The integration of Firebase Firestore ensures scalable and flexible data storage, supporting complex queries and real-time data synchronization.

4.4.3 Real-Time Features

Real-time data synchronization is a critical feature for Quick-cart. Firebase’s real-time database allows instant updates to the user interface when data changes, providing a smooth and responsive user experience. The application leverages Firebase Cloud Messaging to deliver push notifications, ensuring users receive timely updates about their orders and promotional offers.

4.4.4 Security Measures

Security is paramount in handling user data and payment information. Quick-cart employs Firebase Authentication to ensure secure user login and registration processes. Data transmission is secured using HTTPS, and sensitive information is encrypted before storage. Regular security audits and compliance with data protection regulations are integral to maintaining user trust.

V. LIMITATIONS

While Quick-cart leverages the benefits of Flutter and Firebase, several limitations need to be acknowledged:

1. Performance Overheads
Despite Flutter's high performance, there can be instances where performance overheads arise due to the abstraction layer it introduces. Complex animations and heavy graphical operations may not perform as efficiently as native applications, particularly on older or lower-end devices.

2. Platform-Specific Features
Although Flutter provides a wide range of widgets and plugins, certain platform-specific features and functionalities may not be fully supported or might require additional effort to implement. This can lead to increased development time when integrating native modules or functionalities unique to either iOS or Android.

3. Learning Curve
Developers transitioning from native development to Flutter might face a steep learning curve. Flutter uses Dart, a language that may be unfamiliar to developers accustomed to Java, Kotlin, Swift, or Objective-C. Additionally, understanding Flutter's unique widget-based architecture and reactive programming model requires time and practice.

4. Limited Third-Party Libraries
While Flutter's ecosystem is rapidly growing, it still lags behind more mature platforms in terms of available third-party libraries and plugins. This can limit the ease with which developers can integrate certain functionalities and may necessitate creating custom solutions, which can be time-consuming.

5. Dependency on Google Services
Both Flutter and Firebase are Google products, which means that Quick-cart is heavily dependent on Google's ecosystem. Any changes, deprecations, or service disruptions from Google could directly impact the application's functionality and availability.

6. Firebase Data Management
Firebase Realtime Database, while powerful, may present challenges with complex queries and data relationships. Developers may need to carefully design their database structure to avoid performance bottlenecks and ensure efficient data retrieval and synchronization.

VI. CONCLUSION

In conclusion, the development of our "Cross-platform application for food delivery" application is an important step in the reform of the food industry. Our work aims to provide comprehensive information, user-oriented and technological solutions to problems in this area, focusing on features such as various cuisine options, bulk ordering-booking capabilities, real time notification system, real-time chatting interface with restaurant, order history, real-time reviews and rating system, search abilities. Also, by solving the limitations of existing food delivery applications and improving the user experience, our application has the potential to create new business services. It will deliver a seamless, personalized and visually immersive food buying experience that meets users' needs and preferences. This project not only meets the needs of today's fast-paced digital world, but also paves the way for future innovations in food delivery services.
Quick-cart showcases the potential of using Flutter and Firebase for developing high-performance, cross-platform mobile applications. The unified development approach not only accelerates the development process but also ensures a consistent user experience across different devices.
Future work will focus on adding more advanced features, such as AI-based restaurant recommendations and enhanced delivery route optimization, various payment options provided by restaurants, changing language of the application on user need, adding voice recognition criteria for search and improving the scalability of the application to handle a larger user base.

VII. REFERENCES

[1] Lidya Chitra Laoh, Timothy Adithia Pongantung, Carolin Mulalinda "Android Application Food Delivery Services" in 2020from IEEE Xplore.
[2] D. Jyothishman, "Consumer perception towards 'online food ordering and delivery service': an empirical study", Journal ofManagement, vol.5, no 5, September-October 2018, pp 155-163.
[3] M. Gupta, "A study on impact of online food delivery app on restaurant business special reference to Zomato and Swiggy",International Journal of Research and Analytical Reviews, vol. 6, no 1, 2019, pp 889-893.
[4] Cristina-Edina Domokos, Barna Séra, Károly Simon, Lajos Kovács, Tas-Béla Szakács "Netfood: A Software System forFood Ordering and Delivery" in sep-2018 from IEEE.
[5] Yakob Utama Chandra, Cadelina Cassandra "Stimulus Factors of Order Online Food Delivery" in Aug-2019 from IEEE.
[6] Andhika, Yonky Pernando, Ihsan Verdian, Yodi, M. Reza Pradana "Vege Application! Using Mobile Application to BuyVegetarian Food" in 2019 IEEE
[7] Zahita Cahyani, Rahmat Nurcahyo, Farizal, "Popularity Analysis of Mobile Food Ordering Apps In Indonesia" in June04,2020 from IEEE.
[8] S. Jadhav, "Food ordering mobile applications – a new wave in food entrepreneurship", International Journal of InnovativeTechnology and Exploring Engineering, vol 8, 2018, pp. 302-305.
[9] C. Reddy, and G. Aradhya, "Driving Forces for the Success of Food Ordering and Delivery Apps: A Descriptive Study",International Journal of Engineering and Management Research, 10, 2020.
[10] Ali Abdalah Alalwan, "Mobile Food Ordering Apps: An Empirical Study of The Factors Affecting Customer E-Satisfaction and Continued Intention to Reuse," International Journal of Information Management, vol. 50, February 2020.