

DEVELOPMENT OF A CROSS-PLATFORM APPLICATION WITH THE NOSQL FIREBASE DATABASE AND THE IONIC FRAMEWORK

Olena Fedusenko¹, Iryna Domanetska², Yaroslav Khrolenko³, Georgy Gaina⁴

¹ Taras Shevchenko National University of Kyiv/Department of intellectual technologies, Kyiv, Ukraine, fedusenko@knu.ua

² Taras Shevchenko National University of Kyiv/Department of intellectual technologies, Kyiv, Ukraine, domanetska@knu.ua

³ Institute for Information Recording of NAS of Ukraine/Department for Intelligent Technologies of Decision-Making Support, Kyiv, Ukraine, yaroskhr@gmail.com

⁴ Taras Shevchenko National University of Kyiv/Department of intellectual technologies, Kyiv, Ukraine, ggaina@knu.ua

ABSTRACT

The article addresses the development of a cross-platform application using the Ionic hybrid web development framework for the client-side and the NoSQL Firebase database for implementing the server-side. The modern market often requires IT specialists to quickly develop high-quality solutions designed to solve specific business problems. Relevant tasks include developing a client-side application that works across multiple devices simultaneously and minimizing the server-side by leveraging cloud services that provide Database as a Service (DBaaS). Hybrid web development allows for the simultaneous creation of web and mobile applications, such as for Android and iOS, with the Ionic framework being one of the most popular solutions for hybrid development. This framework enables the development of cross-platform applications using web development tools, namely HTML, CSS, and JavaScript, and integrates with three of the most common web frameworks: Angular, React, and Vue. Additionally, it includes its own set of native visual components. Firebase is a popular tool for quickly building applications without managing infrastructure. It provides a scalable cloud-based NoSQL database for real-time data storage and synchronization. Built on Google's infrastructure, Firebase automatically scales. One of the key challenges when using NoSQL databases is ensuring data integrity during data deletion and editing operations. Using a cross-platform application developed by the authors to manage a small coffee shop's website as an example, this article proposes algorithms to ensure data integrity when using the Firebase NoSQL database. During the development of the application, the RxJS library was used to support real-time functionality, utilizing the Observer pattern to simplify the handling and composition of asynchronous or callback code.

KEYWORDS: Cross-platform application, Ionic framework, NoSQL database Firebase, Angular framework, data integrity.

INTRODUCTION

With the rapid growth of smartphones, the number of mobile applications running on Android and iOS platforms is increasing significantly every year. According to statistics from the web traffic analysis tool StatCounter, the distribution of users by operating systems in the last year was as follows: 44.59% of Internet users use Android, 26.81% use Windows, 18.47% use iOS, 5.66% use OS X, and 3.35% use other platforms.

The diversity of leading mobile platforms in terms of user interfaces, programming languages, and ecosystems makes cross-platform technologies a popular development direction. These technologies assist in creating mobile applications that can run on target platforms (usually Android and iOS) with minimal changes to the code. [2].

An analysis of the works of Y.Chang, S.Oh [3], Muhammad Shoaib Farooq, Shamyia Riaz, Atif Alvi, Asghar Ali, Ibtesam U Rehman [4], Oluwatofunmi Adetunji, Chigozirim Ajaegbu[5] and others has shown that the main trend in modern IT development is moving away from single-platform cross-platform application development (e.g., for Android or iOS) and moving towards simultaneous multi-platform development.

The analysis revealed that cross-platform application development is highly relevant. During the development of a cross-platform application for a small coffee shop, the authors faced the challenge of selecting tools that allow for quick development of both the client and server parts of the application.

Developing hybrid web applications allows for the simultaneous creation of applications for Android, iOS, and the web. One of the most popular tools for such development is the Ionic framework. In addition to the client side, such applications must also include a server side, which can be complex. One effective solution for the server side is Firebase, a NoSQL and cloud platform offering a wide range of services, tools, and features available through Firebase SDK (Software Development Kit) and API (Application Programming Interface), allowing developers to create, deploy, and effectively manage their applications [6].

The purpose of this article is to analyze the process of developing a cross-platform application using the Ionic framework and Firebase, using an application for a small coffee shop as an example.

1. THEORETICAL FOUNDATIONS OF THE STUDY

Modern approaches to cross-platform development can be classified as follows:

1. Progressive Web Apps (PWA): Applications designed to work on any platform that uses a standards-compliant browser, including desktop and mobile devices [7].
2. Native App: Developed individually for each target platform.
3. Hybrid web App: Developed based on frameworks with web interfaces and common components, with a single codebase that runs anywhere.
4. Hybrid App: This approach combines frameworks with a native user interface and shared code, as well as frameworks with a common codebase and native code.

Based on the research of Biørn-Hansen, A., Rieger, C., Grønli TM [2], Oluwatofunmi Adetunji, Chigozirim Ajaegbu, Nzechukwu Otuneme, Olawale J. Omotosho [5] and Dhananjaysinh Jhala [7], a comparative analysis of these approaches to cross-platform development is presented in Table 1

Table 1.

Results of a comparative analysis of frameworks for cross-platform development

Feature	PWA	Native App	Hybrid web App	Hybrid App
Installable	Yes	Yes	Yes	Yes
Offline access	Yes	Yes	Yes	Yes
Testable before installation	Yes	No	Yes	Yes
Cross-platform availability	Limited	No	Yes	Yes
Hardware and Platform API access	Limited	Yes	Yes	Yes

Source: Own elaboration

Based on the results of the comparative analysis, we can conclude that hybrid web applications and hybrid-native applications have the widest opportunities.

2. RESEARCH RESULTS

Ionic is a cross-platform application development framework based on web technologies. Ionic provides everything you need to create and connect high-quality mobile and web applications on your own using web programming tools and three most common web frameworks Angular, React, Vue. In addition, Ionic provides a wide range of tools for testing mobile applications, such as Ionic CLI, automated testing tools and mobile device simulators[8].

Creating a single application for different platforms in Ionic is possible thanks to the use of the so-called Capacitor. Capacitor is an open source runtime for creating native web applications. It can be used to create cross-platform iOS, Android, and progressive web apps[9].

To develop the user interface, Ionic offers to use its own components instead of standard HTML elements, which allow you to build a native interface for each platform.

Firebase, a popular Backend-as-a-Service platform, provides a powerful real-time database solution that allows developers to perform efficient synchronization in their mobile applications[10].

To support real-time work, you need to use the RxJS library. RxJS is a popular JavaScript library for reactive programming in Web applications. RxJS is a library for composing asynchronous and event-based programs by using observable sequences. It provides one core type, the Observable, satellite types (Observer, Schedulers, Subjects) and operators inspired by Array methods (map, filter, reduce, every, etc.) to allow handling asynchronous events as collections[11].

Let's look at the practical use of the above technologies on the example of an application for a small coffee shop. The coffee shop has dishes, each of which belongs to a certain category, for example, coffee belongs to the category of drink, etc. You need to develop a cross-platform application for the coffee shop administrator that will provide the following features:

1. Working with food categories

- Create a new category of dishes
- Edit a category
- Deleting a category. When you delete a category, all dishes associated with it must also be deleted.

2. Working with dishes

- Create a new dish in a specific category
- Edit a dish
- Deleting a dish.

In addition, you need to develop a web application for visitors to the cafe where they can view dishes and their prices.

Thus, the general architecture of the system under development will look like this (Figure 1).



Figure 1. Coffee shop's architecture

Source: Own elaboration

The data in the database will be stored in two separate structures - categories and dishes (Figure 2).



Figure 2. Coffee shop's Database

Source: Own elaboration

To generate unique record identifiers in the Realtime Database, it was decided to develop our own auto-generation function that will work as follows:

1. Find the maximum value of the identifier among those existing in the database.
2. Generate a new value by adding one to the maximum value.

Thus, the identifier of categories and dishes in the database will always be unique.

One of the most common problems that can arise in NoSQL databases is the problem of data integrity. In a cross-platform application for a coffee shop, this problem arises when a category of dishes is deleted, since all dishes related to it must be deleted along with the category. The authors propose the following algorithm for solving this problem (Figure 3).

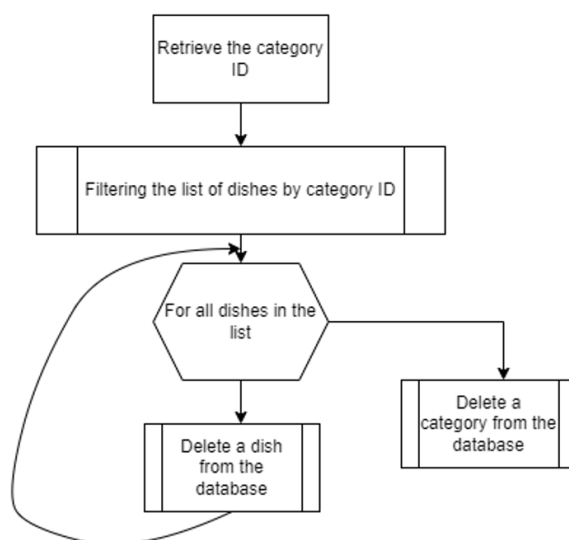


Figure 3. Algorithm for deleting a dish category

Source: Own elaboration

Let's develop the architecture of a cross-platform application for a coffee shop administrator using the TOGAF architectural methodology and the Archimate specification to build a diagram in accordance with the framework (Figure 4).

The TOGAF framework (The Open Group Architecture Framework) is a set of tools for developing architectures for various purposes. With its help, an information system is represented as a set of modules. The ArchiMate specification, a standard of The Open Group, is an open and independent modeling language for enterprise architecture that is supported by various tool providers

and consulting firms. ArchiMate visual modeling helps to describe and understand complex systems[12].

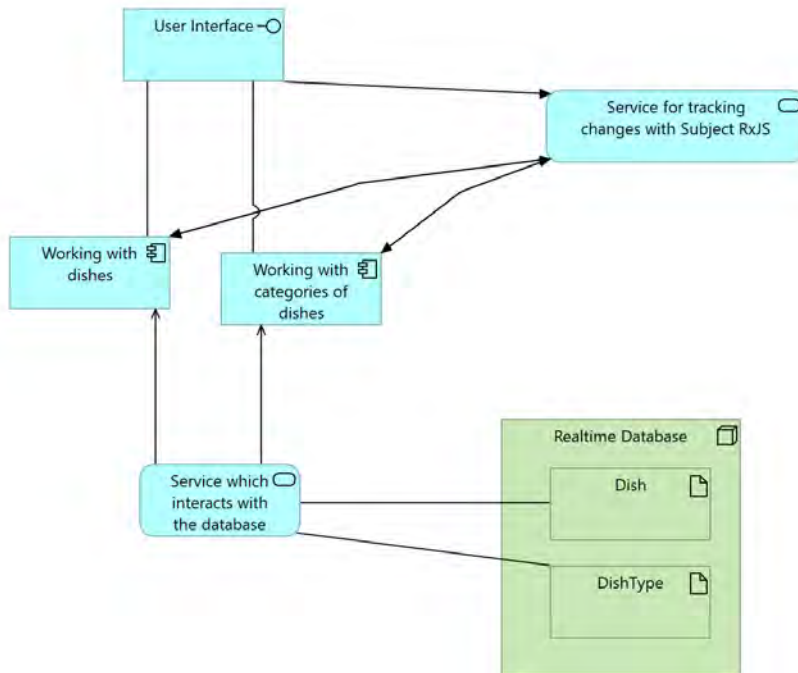


Figure 4. Administrator coffee shop's architecture

Source: Own elaboration

The Angular framework was used to develop a cross-platform application for the administrator using the Ionic framework. When creating the interface, it was decided to change the appearance of standard native user interface elements from the Ionic framework (Figure 5).

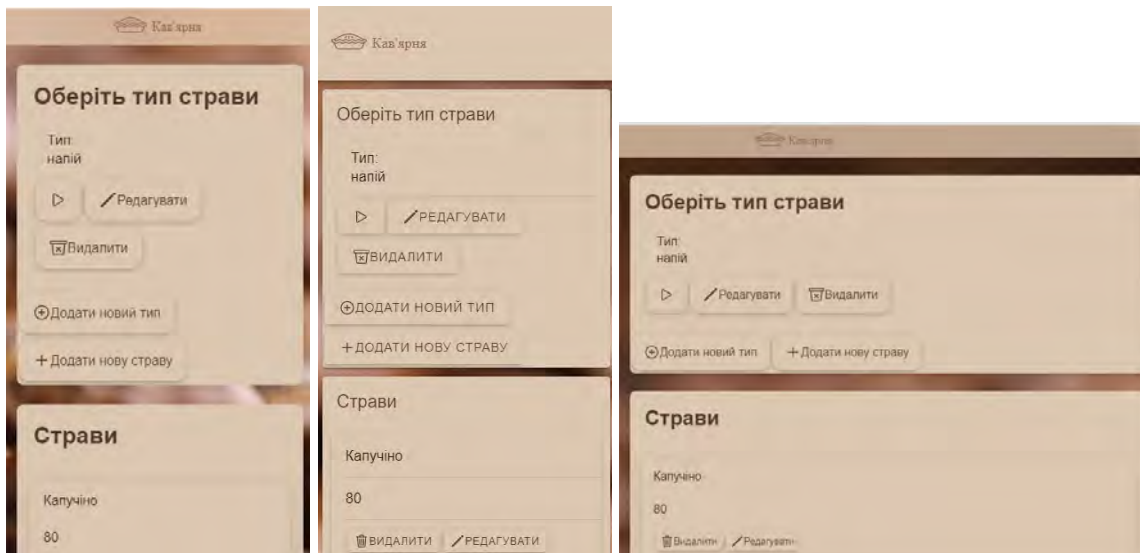


Figure 5. Administrator coffee shop's

Source: Own elaboration

The ReactJS web framework was used to develop the application for the cafe's customers (Figure 6).

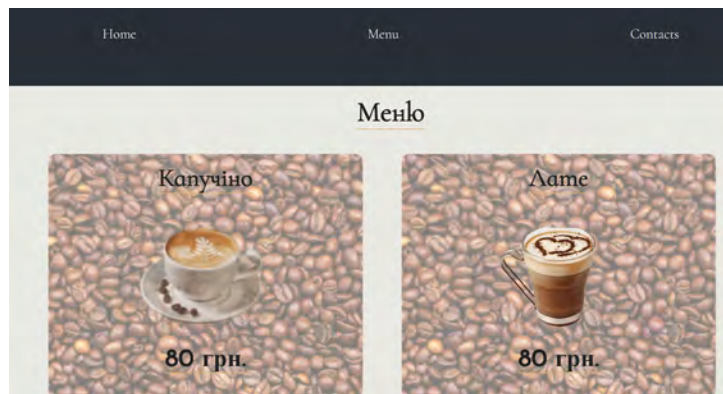


Figure 6. Client's app
Source: Own elaboration

CONCLUSION

The modern market of IT products and the variety of gadgets used at work require developers to focus on developing cross-platform applications. Developing both the client and server side of such applications is quite a challenge.

The authors propose to use the hybrid web framework Ionic for the development of the client side and NoSQL Realtime Database Firebase for the server side. The main problem when using NoSQL Database is the problem of indexing and maintaining data integrity. The authors propose a simple algorithm to solve this problem, which is used in the developed cross-platform application for a small coffee shop.

REFERENCES

1. Operating System Market Share Worldwide. StatCounter [Online]. Available from: <https://gs.statcounter.com/os-market-share> [Accessed 04/11/2024].
2. Biørn-Hansen, A., Rieger, C., Grønli, TM. et al. «An empirical investigation of performance overhead in cross-platform mobile development frameworks» *Empirical Software Engineering*, vol. 25(4), pp. 2997–3040, 2020.
3. Oluwatofunmi Adetunja, Chigozirim Ajaegbub, Nzechukwu Otunemec, Olawale J. Omotosho «Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development», *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, Volume 68, No 1, pp 85-99, 2020.
4. Muhammad Shoaib Farooq, Shamyla Riaz, Atif Alvi, Asghar Ali, Ibtesam U Rehman «Cross-Platform Mobile Development Approaches and Frameworks», *VFAST Transactions on Software Engineering*, Volume 10, Number 2, pp. 79-93, 2022.
5. Oluwatofunmi Adetunji, Chigozirim Ajaegbu, Nzechukwu Otuneme, Olawale J. Omotosho «Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development», *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, Vol. 68, №1, pp. 85-99, 2020.
6. K. Milojković, M. Živković, N. Bačanin Džakula, «Agile Multi-user Android Application Development With Firebase: Authentication, Authorization, and Profile Management», *International Scientific Conference on Information Technology, Computer Science, and Data Science, Belgrade, Singidunum University, Serbia*, pp. 405-412, 2024.

7. Dhananjaysinh Jhala «A Study on Progressive Web Apps As A Unifier for Native Apps and the Web», *International journal of engineering research & technology (ijert)*, Volume 10, Issue 05, pp. 207-210, 2021.
8. Introduction to Ionic [Online]. Available from: <https://ionicframework.com/docs/> [Accessed 04/11/2024].
9. Building cross-platform apps with Capacitor [Online]. Available from: <https://ionic.io/resources/articles/building-cross-platform-apps-with-capacitor> [Accessed 04/11/2024].
10. M. F. Younis and Z. S. Alwan, «Monitoring the performance of cloud real-time databases: A firebase case study», *Al-Sadiq International Conference on Communication and Information Technology (AICCIT)*, Al-Muthana, Iraq, pp. 240-245, 2023. [Accessed 04/11/2024].
11. Introduction to RxJS [Online]. Available from: <https://rxjs.dev/guide/overview> [Accessed 06/11/2024].
12. [TOGAF Library](https://publications.opengroup.org/togaf-library?_ga=2.254190216.1040667565.1608301881-1137755952.1604917148) [Online]. Available from: https://publications.opengroup.org/togaf-library?_ga=2.254190216.1040667565.1608301881-1137755952.1604917148 [Accessed 06/11/2024].