Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.017

IJCSMC, Vol. 7, Issue. 9, September 2018, pg.1 – 12

Android-Platform Based Determination of Fastest Cross-Platform Framework

Faraz Ur Rehman Quazi[1], Nishant Sinha[2]
[1]P.G. Student, M.E. (IT), Medi-caps Institute of Science & Technology, Indore, India
[2]Assistant Professor, Medi-caps Institute of Science & Technology, Indore, India
[1] contact2farazquazi@gmail.com; [2] nishant.sinha.k@gmail.com

Abstract— Nowadays, mobile application development is very essential for any business organization because of a smartphone is not just a communication device but also an essential part of our daily life. To cover the maximum part of the mobile market, business organization needs to develop the multiple mobile applications for the different kinds of mobile operating systems to achieve the same functionality but that task is costly and time-consuming that is why they have to move on cross-platform mobile application development technologies because of it is cost effective and time-saving approach and allows developers to use same code for multiple mobile operating systems. In this scenario, application developers are facing the challenge to choose the finest cross-platform framework among various frameworks. The objective of this paper is to help application developers and researchers to make the right selection of the finest cross-platform mobile application development framework. This paper presents a task execution time-based comparison of three very popular cross-platform frameworks which are Ionic, React Native, and PhoneGap on the Android platform.

Keywords— Smartphone, Operating System, Mobile Application, Native Application, Cross-platform Application, Cross-platform Frameworks, Ionic Framework, React Native Framework, PhoneGap Framework

I. INTRODUCTION

Today's generation is much more depended on the smartphones. The domain of the smartphones are increasing day by day because smartphones are playing a vital contribution to our daily life, it is progressing in every field of life. It is necessary to deploy your business by using mobile applications which cover the maximum part of the market. There are many operating systems for the mobile devices are available in the marketplace, customers want to deploy the mobile applications over the all operating systems to cover the maximum part of the market in less cost and short time. Because of this scenario, application developers need to develop different native mobile applications for each operating systems to fulfill the customer requirements but that process is time-consuming and costly due to different programming languages used for different operating systems. It is a very hard challenge to develop the native mobile applications to fulfill the expectation of customers in less cost and short time.

© 2018, IJCSMC All Rights Reserved

To overcome the above-listed problems developers are using the cross-platform frameworks for mobile application development, in order to allow using the same code on various operating systems. Cross-platform application development reduces development cost and development time as well as provide a native feel to the end user. There are several cross-platform frameworks are available for cross-platform application development, it is too hard for the developer to make a choice between the different cross-platform frameworks.

This paper presents a comparison of three very popular cross-platform frameworks, which are Ionic, React Native, and PhoneGap. As a result, it provides a reference comparison table for the execution time taken to perform different operations between them. The objective of this paper is to help application developers and researchers to make the best selection of the optimal mobile application development framework. The objective of this paper is to compare the time taken by the cross-platform mobile applications to execute the assigned tasks. Based on this comparative analysis, the developers can select the best finest cross-platform framework for the mobile applications development.

The following sections present a literature survey, problem definition, proposed comparative analysis, comparative analysis result of three cross-platform frameworks respectively. Finally, the last section presents the conclusion of this paper and suggestion for the future work.

II. LITERATURE SURVEY

There are much-related research papers available on comparative analysis between the frameworks for cross-platform application development, these papers presented the cross-platform frameworks comparison based upon the application type, application size, build methodology, used technologies, coding structure, required environment, supported features, and supported platforms.

The paper [1] proposed the comparative analysis of the cross-platform Titanium application and native application. In this paper, compare the applications on the Android and iOS platforms based upon the novelty, attractiveness, efficiency, dependability, and perspicuity but did not include the other core parameters to achieve the accurate comparison results. This comparison was done by the 5 days survey of the 37 participants.

The paper [2] presents the case study of the native applications and hybrid applications with their the advantages and disadvantages. The authors also explain some frameworks based on the native and hybrid technologies but did not summarize any comparative results.

The paper [3] presents the comparative study between native, hybrid, and web applications based upon their features and some useful parameters. The paper [4] presents the comparison between Ionic and OnSenUI. The authors explain the comparison based upon some parameters which are the design of the interface, development cost, development timeline, user experience, maintainability, performance, security, platform independent, tools, and debugging. The paper [5] presents the comparison between the cross-platform frameworks including the Rhodes, PhoneGap, MoSync, and DragonRad. The authors perform the comparison by using the supported application programming interfaces including the calendar, barcode, blue-tooth, notification, camera, geolocation, capture, device, compass, file, connection, contacts, and storage plugins. The paper [6] focuses on the commonly used cross-platform mobile application development tools, and the process which selects the most optimal and suitable cross-platform framework for the particular requirements to depend upon their features. The paper [7] provides the comparison between native and hybrid applications and the comparison is based upon some useful parameters which are coding language, development time, cost, portability, device access, user interface consistency, distribution, and performance. The paper [8] proposed the comparative explication of the cross-platform frameworks by developing the cross-platform applications. The author uses the Titanium framework for developing the applications for the iOS and Android platform development and compare it without including the hardware-based operations.

The paper [9] proposed a method or process to selection of the sublime cross-platform framework to develop mobile applications for the developers and researchers and also evaluate the cross-platform technologies on the basis of time and cost aspects for the mobile applications development. The paper [10] explains the comparison based case study between some cross-platform frameworks for mobile application development on the basis of some parameters which are licensing costs, supported platforms, development environments, user interface, scalability, and maintainability. The paper [11] presents the case study on cross-platform frameworks for application development which are the Ionic framework and Titanium framework. The author did not provide any comparative analysis on the basis of application core functionalities.

In this paper, presents the comparison between cross-platform frameworks which are Ionic, React Native, and PhoneGap. This paper compares the execution time taken by cross-platform frameworks to perform the core, hardware, and database functionalities.

© 2018, IJCSMC All Rights Reserved

III. PROBLEM DEFINITION

A current challenge for the business organization, business enterprises, software companies, and individual mobile developers is to choose the target platforms for their native mobile applications. To cover the maximum users and the supreme part of the market, they need to deploy their native mobile applications on the following popular platforms:

1. Android Platform
2. iOS Platform
3. Window Platform
4. BlackBerry Platform

This approach requires the dedicated native mobile applications for each platform and dedicated application developers for each platform based programming language, therefore, this approach is costly and time consuming that is why they have to move from native mobile applications to cross-platform mobile applications. Application developers and software companies can use the following cross-platform frameworks for the implementation of cross-platform apps:

1. Ionic Framework
2. React Native Framework
3. PhoneGap Framework
4. Titanium Framework

Software companies and application developers are facing the problem in order to the selection of fasted and optimal cross-platform framework among the various frameworks but there is no information available which help them to choose the finest cross-platform framework on the basis of execution time to perform the hardware and database related operations. This paper introduces the required comparison which helps them in the selection process of a finest cross-platform framework. paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified.

IV. PROPOSED COMPARATIVE ANALYSIS

This paper proposed the performance based comparative analysis on the Android platform of the three cross-platform mobile applications which are based upon the Ionic framework, React Native framework, and PhoneGap framework respectively. In this paper, compare the cross-platform applications on the basis of execution time taken by the cross-platform frameworks to perform the battery, device, contact, database, network, and hardware specific operations on the Android platform, detailed list of these operations given below:

A. Battery Specific Operations:

- Get Battery Status
- Get Battery Level

B. Device Specific Operations:

- Get Device Manufacturer
- Get Device Model
- Get Device Platform
- Get Device Serial
- Get Device UUID
- Get Device Version

C. Contact Specific Operations:

- Get Single Contact
- Get Contact List

D. Database Specific Operations:

- Fire Database Insertion Query
- Fire Database Selection Query
- Fire Database Update Query
- Fire Database Selection Query

© 2018, IJCSMC All Rights Reserved

E. Network Specific Operations:

- Get Latitude & Longitude
- Get Network Type

F. Hardware Specific Operation:

- Vibrate the Device

This comparative analysis will be done by the developing the three cross-platform applications with the help of Ionic framework, React Native framework, and PhoneGap framework respectively, then perform the all listed operations one by one and noted the execution time for each operation on each mobile application. In this paper, using the Android platform for execution time analysis because of Android platform covers the almost 76% of the whole smartphone market and the choice of Android platform is justified by it is open source and less expensive than iOS and other platforms. On the basis of that comparison, software companies and developers can easily choose the fastest cross-platform framework for developing the cross-compiled and hybrid mobile applications.

V. COMPARATIVE ANALYSIS RESULTS

This portion presents the proposed comparative analysis results of the described parameter. In the table I, compare the execution time to be taken by the cross-platform applications to complete the given operations, also shows the total execution time to complete all operations and the average execution time to complete the single operation. Execution time is noted on the Android platform for all cross-platform applications in milliseconds. Total execution time is 2903.399999884888ms, 1082.699999911712ms, and 2399.100000131875ms taken by the Ionic application, React Native application, and PhoneGap application respectively. Average execution time is 170.7882352873464ms, 063.6882352889242ms, and 141.1235294195220ms taken by the Ionic application, React Native application, and PhoneGap application respectively.

TABLE I
COMPARATIVE ANALYSIS OF CROSS-PLATFORM APPLICATIONS

S. No.	Performed Operation Name	Execution Time Taken by Cross-platform Applications in Millisecond		
		Ionic Application	React Native Application	PhoneGap Application
01	Get Battery Status	146.3999999687075	039.4999999552965	107.6000000350177
02	Get Battery Level	146.5999999782070	015.4000000329688	093.9000000245869
03	Get Device Manufacturer	133.5999999428168	020.2000000281259	093.0000000400468
04	Get Device Model	143.9000000245869	012.1999999973922	090.4000000329688
05	Get Device Platform	147.2000000067055	018.0999999865889	092.9999999236315
06	Get Device Serial	144.5000000530854	016.1000000080093	087.7999999094754
07	Get Device UUID	138.2000000448897	015.6000000424683	094.6999999461695
08	Get Device Version	139.8000000044703	017.0999999390915	089.3000000389292
09	Get Single Contact	142.5000000745058	015.0999999605119	101.7000000458210
10	Get Contact List	430.3000000072643	515.8999999985099	617.6999999908730
11	Fire Insertion Query	126.5999999595806	001.4000000664964	078.2999999821186
12	Fire Selection Query	141.9999999254942	032.1000000694766	097.7000000886619
13	Fire Update Query	126.5000000130385	001.5000000130385	076.5000000130385
14	Fire Deletion Query	125.2999999560415	000.3999999025836	075.5000000819563
15	Get Latitude & Longitude	363.4000000311062	308.3999999798834	384.2999999644235
16	Get Connection Type	155.4999999934807	024.8999999603256	100.6999999983236
17	Vibrate the Device	151.0999999009072	028.7999999709442	117.0000000158324
Total Execution Time		**2903.399999884888**	**1082.699999911712**	**2399.100000131875**
Average Execution Time		**170.7882352873464**	**063.6882352889242**	**141.1235294195220**

© 2018, IJCSMC All Rights Reserved

The following section presents the screenshots which are showing the cross-platform mobile application homepage, operations list, and the execution time evaluation process to perform the various operations on the Ionic application, React Native application, and PhoneGap application respectively.

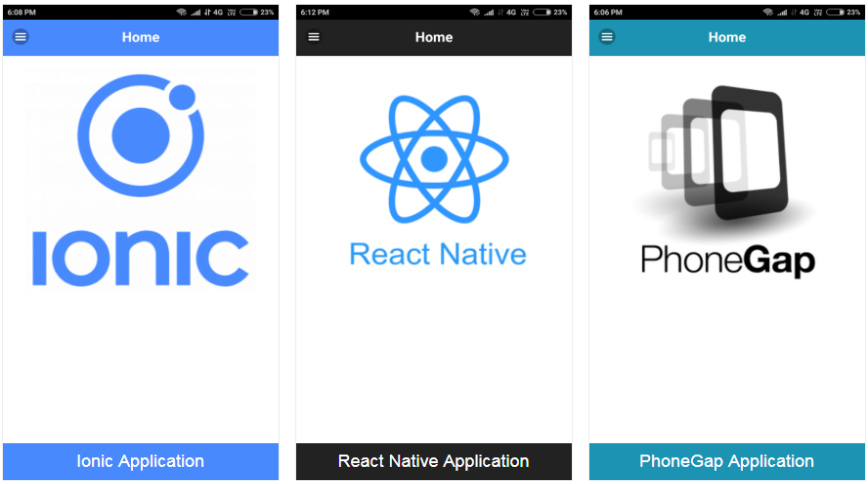

Fig. 1 Screenshot shows the Home page for Ionic application, React Native application, and PhoneGap application respectively

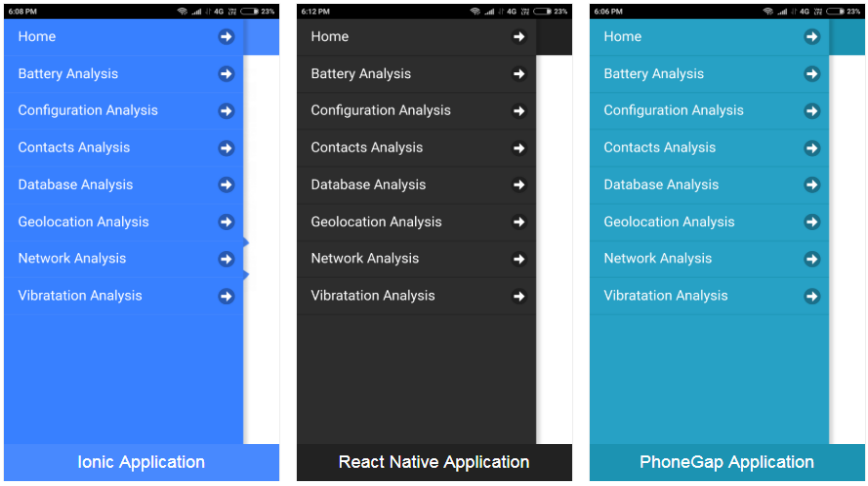

Fig. 2 Screenshot shows the Analysis parameters for Ionic application, React Native application, and PhoneGap application respectively

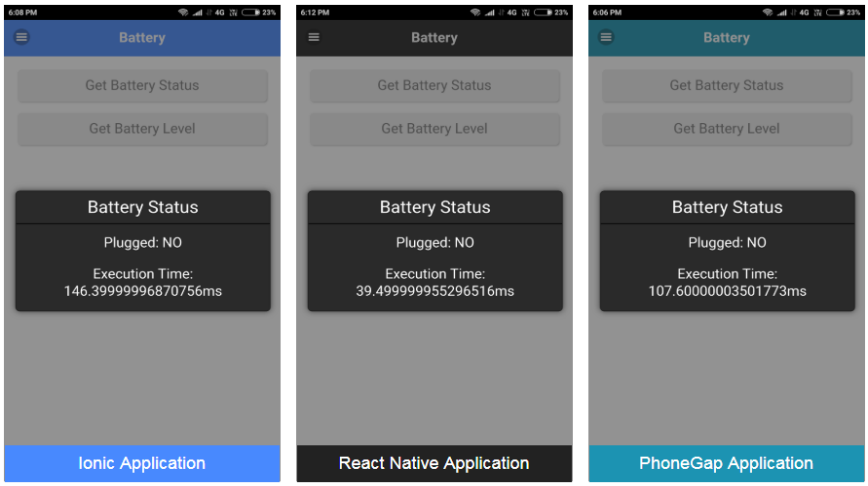

Fig. 3 Screenshot shows the execution time taken in millisecond to getting the Battery Status for Ionic application, React Native application, and PhoneGap application respectively

© 2018, IJCSMC All Rights Reserved

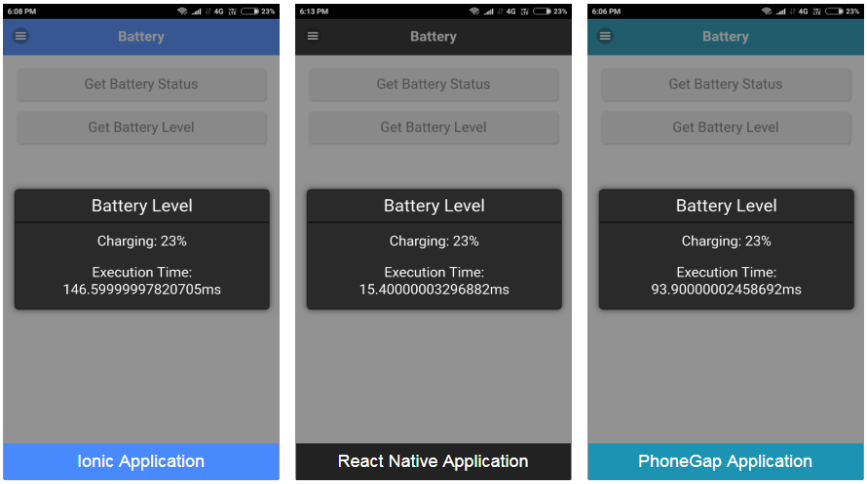

Fig. 4 Screenshot shows the execution time taken in millisecond to getting the Battery Level for Ionic application, React Native application, and PhoneGap application respectively

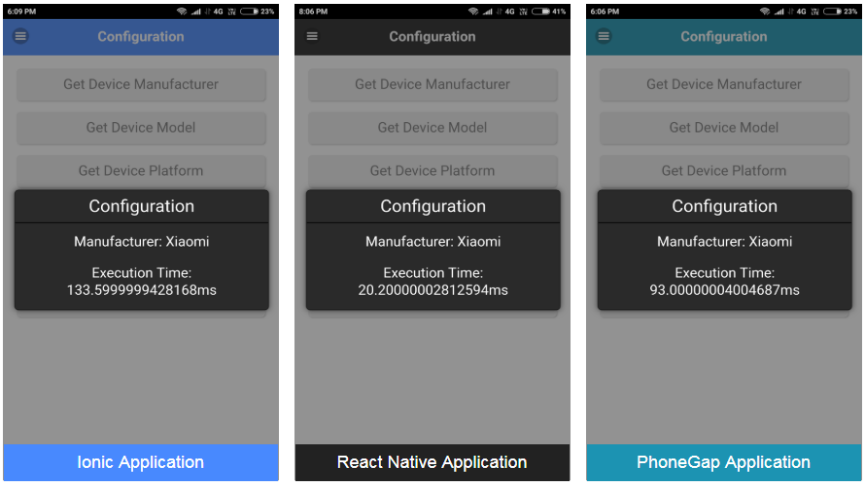

Fig. 5 Screenshot shows the execution time taken in millisecond to getting the Device Manufacturer for Ionic application, React Native application, and PhoneGap application respectively

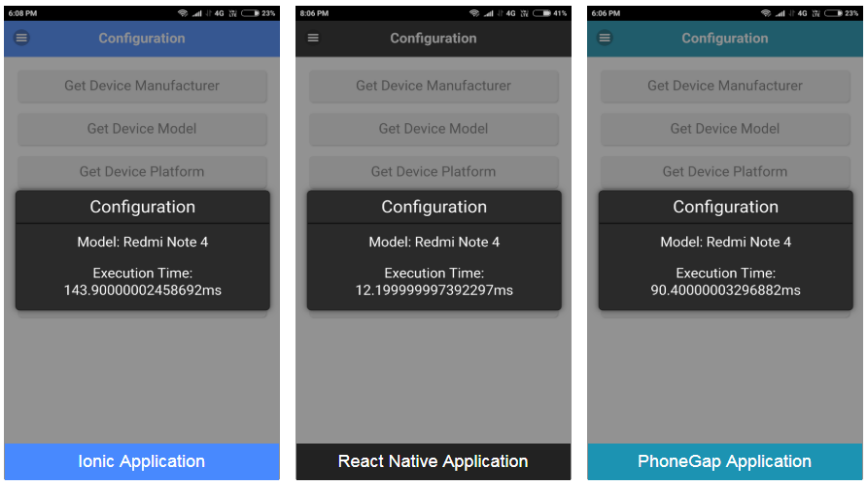

Fig. 6 Screenshot shows the execution time taken in millisecond to getting the Device Model for Ionic application, React Native application, and PhoneGap application respectively

© 2018, IJCSMC All Rights Reserved

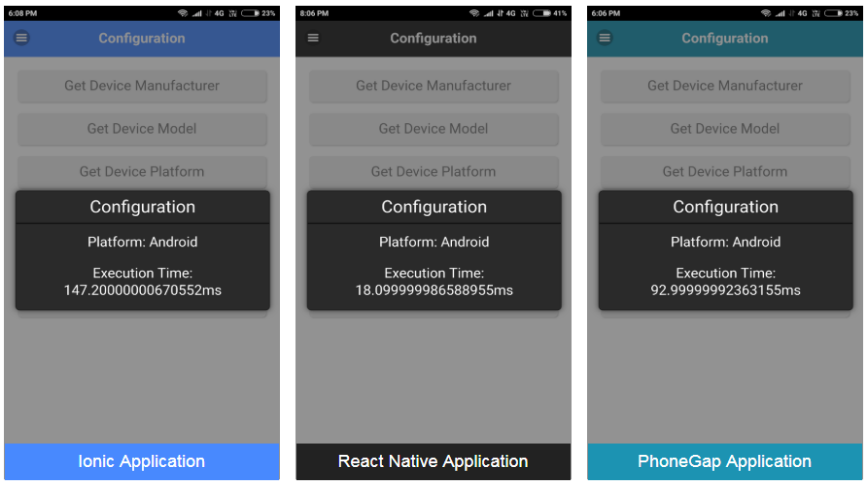

Fig. 7 Screenshot shows the execution time taken in millisecond to getting the Device Platform for Ionic application, React Native application, and PhoneGap application respectively

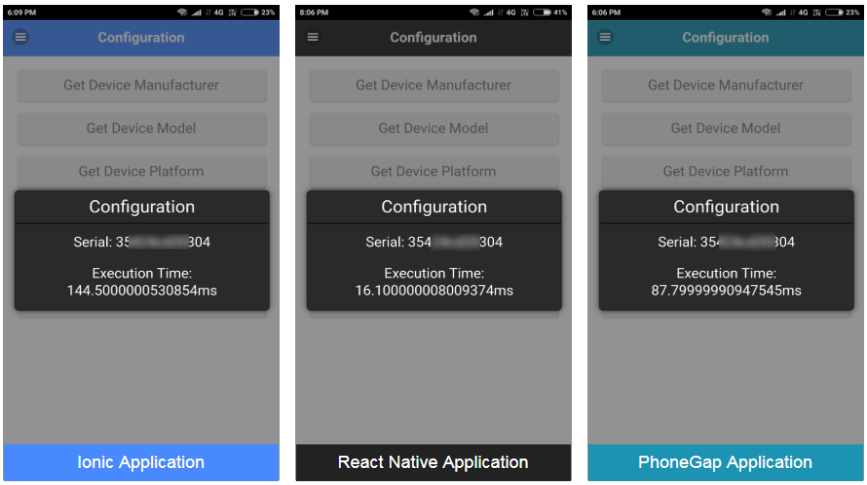

Fig. 8 Screenshot shows the execution time taken in millisecond to getting the Device Serial Number for Ionic application, React Native application, and PhoneGap application respectively

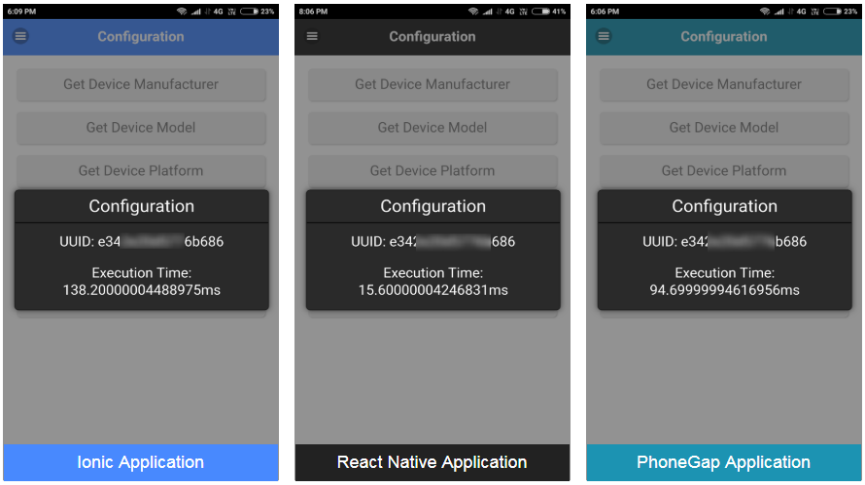

Fig. 9 Screenshot shows the execution time taken in millisecond to getting the Device UUID for Ionic application, React Native application, and PhoneGap application respectively

© 2018, IJCSMC All Rights Reserved

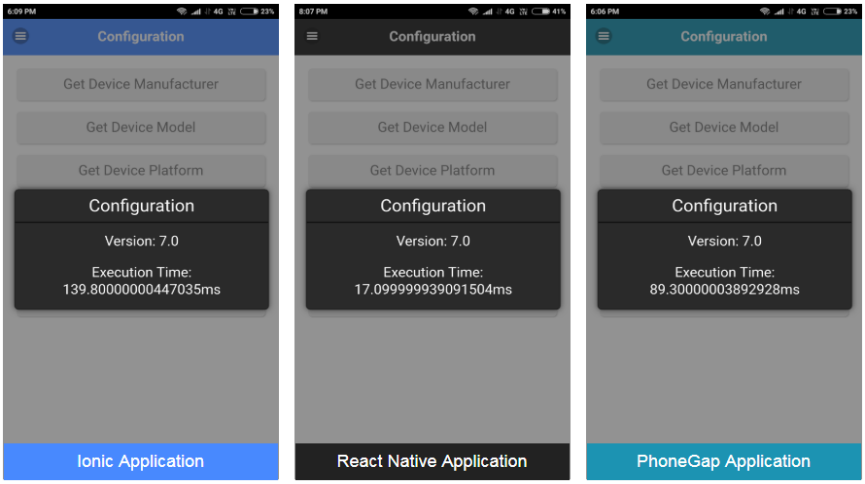

Fig. 10 Screenshot shows the execution time taken in millisecond to getting the Android Platform Version for Ionic application, React Native application, and PhoneGap application respectively

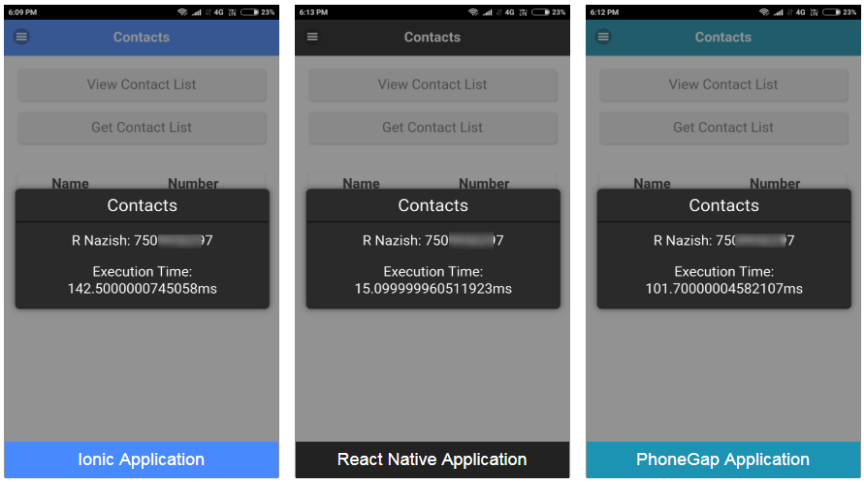

Fig. 11 Screenshot shows the execution time taken in millisecond to getting the Single Contact from List for Ionic application, React Native application, and PhoneGap application respectively

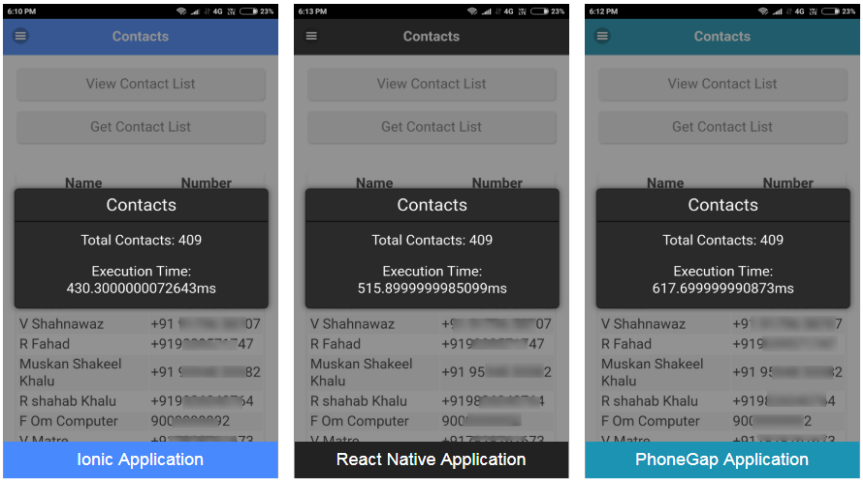

Fig. 12 Screenshot shows the execution time taken in millisecond to getting the Contact List for Ionic application, React Native application, and PhoneGap application respectively

© 2018, IJCSMC All Rights Reserved

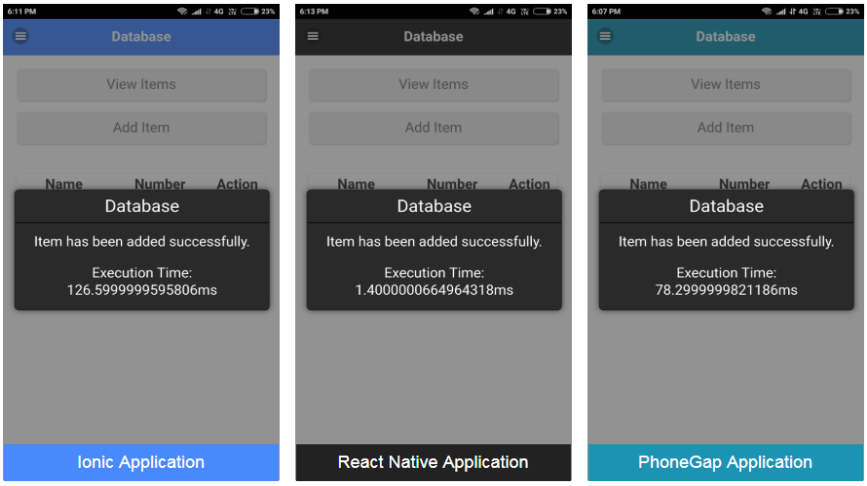

Fig. 13 Screenshot shows the execution time taken in millisecond to fire the Database Insertion Query for Ionic application, React Native application, and PhoneGap application respectively

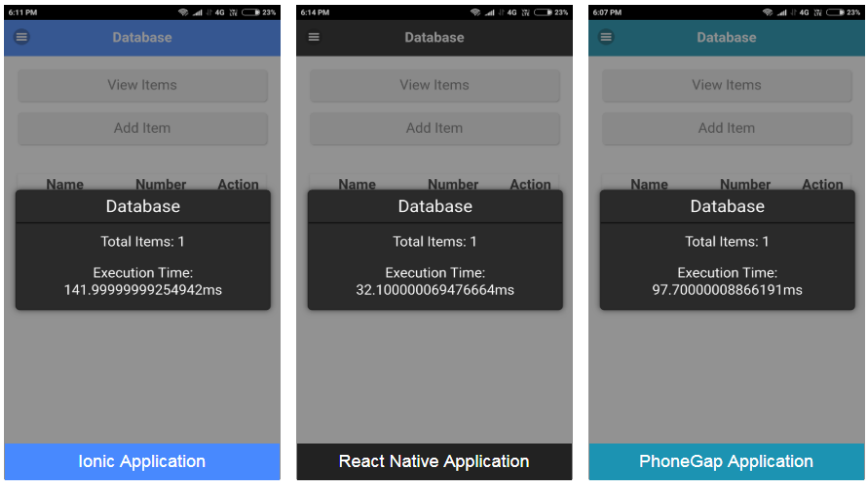

Fig. 14 Screenshot shows the execution time taken in millisecond to fire the Database Selection Query for Ionic application, React Native application, and PhoneGap application respectively

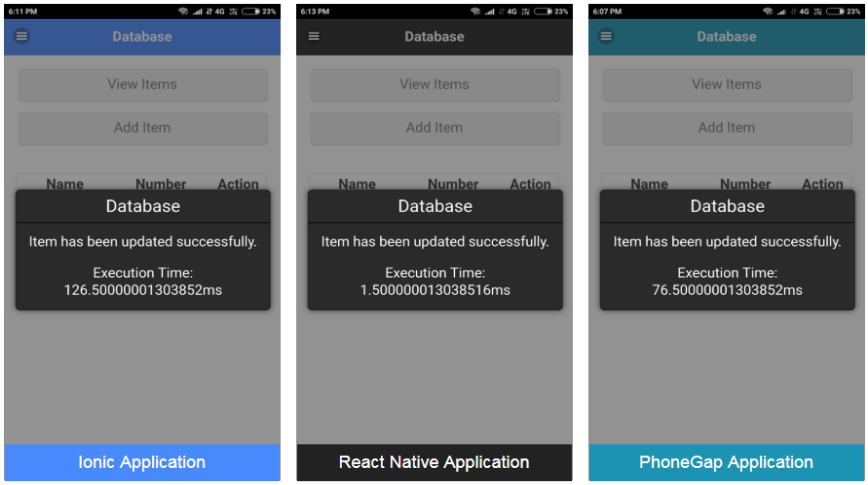

Fig. 15 Screenshot shows the execution time taken in millisecond to fire the Database Update Query for Ionic application, React Native application, and PhoneGap application respectively

© 2018, IJCSMC All Rights Reserved

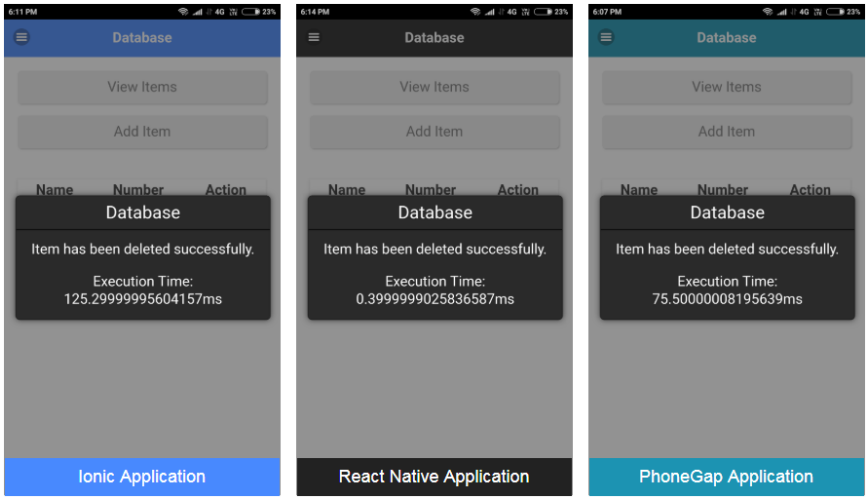

Fig. 16 Screenshot shows the execution time taken in millisecond to fire the Database Deletion Query for Ionic application, React Native application, and PhoneGap application respectively

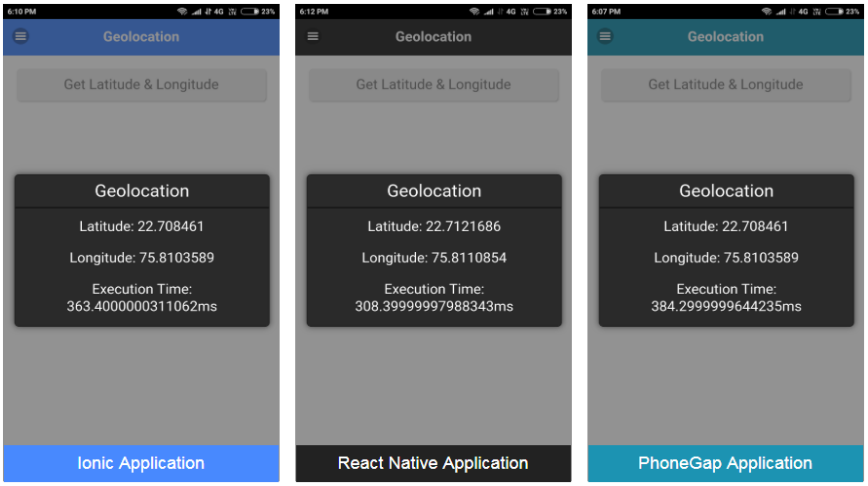

Fig. 17 Screenshot shows the execution time taken in millisecond to getting the Latitude & Longitude for Ionic application, React Native application, and PhoneGap application respectively

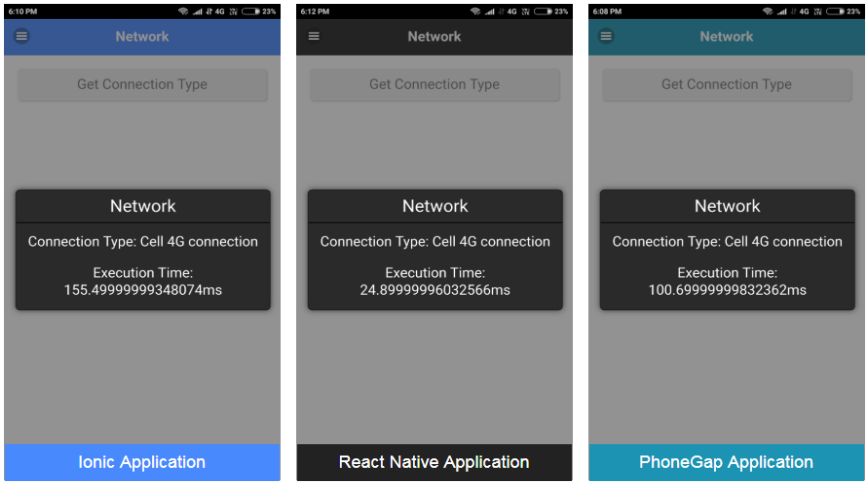

Fig. 18 Screenshot shows the execution time taken in millisecond to getting the Network Connectivity Type for Ionic application, React Native application, and PhoneGap application respectively

© 2018, IJCSMC All Rights Reserved

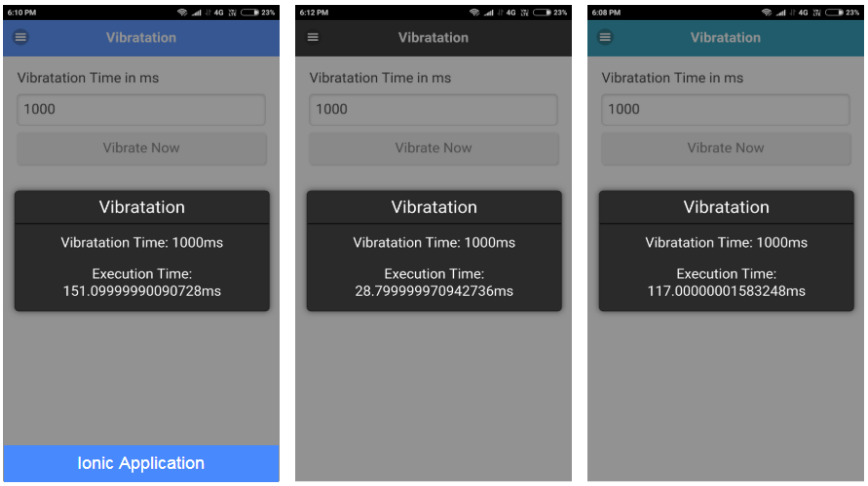

Fig. 19 Screenshot shows the execution time taken in millisecond to Vibrate the Device for Ionic application, React Native application, and PhoneGap application respectively

To get the execution time, all cross-platform applications were installed on the smartphone with the Octa-core Max 2.0 GHz processor, 64GB of ROM, and 4GB of RAM at Android platform version 7.0 and compared the execution time in milliseconds by the Ionic application, React Native application, and PhoneGap application to complete the those operations which are based upon the battery, device, contact, database, network, and hardware specific.

VI. Conclusions and Future work

Finally, on the basis of execution time comparison, it is determined that React Native application performs faster than Ionic application and PhoneGap application, therefore, React Native framework is the finest cross-platform framework for the cross-platform mobile application development. In the comparison table, we can see that the execution time of the React Native application is less than Ionic application and PhoneGap application on the Android platform.

In the future, researchers can use other comparison parameters which are may be based upon the GPS, Camera, Push Notification, etc. Also, we can compare other famous cross-platform frameworks on multiple platforms like iOS, Window, and BlackBerry operating systems for the future work.

Acknowledgement

By this acknowledgment, I express our sincere gratitude to all those people who have been associated with this paper directly or indirectly. I sincerely thank respected guide Mr. Nishant Sinha, who has guided me and cooperated with me at different stages during the preparation of this paper.

References

[1] Esteban Angulo and Xavier Ferre, "A Case Study on Cross-Platform Development Frameworks for Mobile Applications and UX", in Proceedings of the XV International Conference on Human Computer Interaction, ACM, September 2014.

[2] Ketan Anant More, and Ms.Priya Chandran, "Native Vs Hybrid Apps", in International Journal of Current Trends in Engineering & Research (IJCTER), Volume 2, Issue 6, June 2016.

[3] Paulo R.M. de Andrade, Adriano B. Albuquerque, Otávio F. Frota, Robson V. Silveira, and Fátima A. da Silva, "Cross Platform App - A Comparative Study", in Journal of Computer Science and Technology (JCST), February 2015.

[4] Anmol Khandeparkar, Rashmi Gupta, and B. Sindhya, "An Introduction to Hybrid Platform Mobile Application Development", in International Journal of Computer Applications (IJCA), Volume 118, Issue 15, May 2015.

[5] Manuel Palmieri, Inderjeet Singh, and Antonio Cicchetti, "Comparison of Cross-Platform Mobile Development Tools", in 16th International Conference on Intelligence in Next Generation Networks, 2012.

[6] Lachgar Mohamed, and Abdali Abdelmounaïm, "Decision Framework for Mobile Development Methods", in (IJACSA) International Journal of Advanced Computer Science and Applications, Volume 8, Issue 2, 2017.

© 2018, IJCSMC All Rights Reserved

[7] H.T. Palliyagurua, L.I.E.P. Weerathungab, E.A.L.R. Ekanayakec, S.P. Wijesuriyad, and M.P.A.W. Gamagee, "An Approach to Hybrid Technology for Mobile Application Development", in Sri Lanka Institute of Information Technology, Colombo, Sri Lanka, Volume 2, Issue 2, 2016.

[8] Xanthopoulos, Spyros, and Stelios Xinogalos, "A comparative analysis of cross-platform development approaches for mobile applications", in Proceedings of the 6th Balkan Conference in Informatics, ACM, 2013.

[9] P. Smutny, "Mobile development tools and cross-platform solutions", in proceedings of the 13th International Conference on Carpathian Control (ICCC), IEEE Xplore Press, High Tatras, pp. 653-656, 2013.

[10] H. Heitkötter, S. Hanschke, and T. A. Majchrzak, "Evaluating crossplatform development approaches for mobile applications", in Springer, Berlin, pp. 120-138, 2012.

[11] M. Palmieri, I. Singh, and A. Cicchetti, "Comparison of cross-platform mobile development tools", in Intelligence in Next Generation Networks (ICIN), pp. 179 –186, 2012.

© 2018, IJCSMC All Rights Reserved