iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45965

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Enhancing the Performance of Android

Raunak R Kolle[1], Dr. Rajashekara Murthy S[2]
[1, 2]*Information Science Engineering, RV College of Engineering, Bangalore, India*

Abstract: ***In the present day, with the increasing popularity of the android operating system, applications being built on the Android SDK draw much attention. However, some applications built on android tend to have a cumbersome interface causing inconveniences to users. This can be reduced by eliminating the redundancies during the build of these applications. List view is commonly used in android applications to arrange a list of scrollable items in a sequential order. This consumes a considerable amount of memory and CPU utilization resulting in the degradation of the applications. One proposed solution to this can be the introduction of the RecyclerView. The RecyclerView traditionally replaces the GridView and the ListView which were previously used in android applications. This improvement is achieved by recycling the views which are out of the visibility of the user inturn enhancing both the ListView as well as the GridView. Networking in android was previously a nightmare for developers where they had to retrieve and analyze their own data. Data transfer is performed in Android applications using a Web REST API. Development speed can be enhanced by efficient data analysis by utilizing a library from the entire range of REST client libraries. REST (Representational State Transfer) has become the most commonly used way for creating, publishing, and consuming Web services, exploiting JavaScript Object Notation (JSON) as a data exchange format. The most used libraries for accessing the REST Web APIs today are Android Volley and Retrofit. An android device is hindered by limited power, memory and resources. Several architectures such as MVC(Model-View-Controller), MVP(Model-view-Presenter) and MVVM(Model-View-Viewmodel) are commonly used in the development of android applications. In order to develop applications with efficient CPU utilization, reduced memory usage and a lowered execution time, the most suitable of these architectures is to be utilized.***
Keywords: Android, RecyclerView, Android Volley, Retrofit, MVVM, MVP, ListView

I. INTRODUCTION

The emergence of smartphones have changed the way we lead our lives, as such mobile applications have become an integral part of our day to day life, and any hindrance caused by them is needed to be minimized so that we can have a more pleasurable experience. Lowering the prices of mobile devices with higher memory size and longer battery life has resulted in a surge in the usage of mobile phones and its applications. With the emergence of the android operating system developed by google, it has been one of the most prominent operating systems in mobile phones. With the large number of users using this operating system, it is necessary to keep the applications running on our devices powered by the android operating system efficiently with minimal overheads. Improving the user's experience and bringing in smooth running is needed so that more users can be drawn to and fewer users are drawn away from this operating system.

Certain applications are to be loaded from the server hence requires the need for an internet connection which becomes a major part of the application. Thus the network usage and the performance of the applications must be such that they provide a 100% performance with lesser usage of the internet. The introduction of RecyclerView with the advent of Android Lollipop has made great leaps improving the performances of android devices by great folds.

The rapid advancements of mobile technology, the subsequent innovation in their services and the ease and comfort with which the users can use the applications built on them have had a major impact on the e-commerce sector where the users need to merely open the app, select their required items, add them to their cart and checkout to the payment section. Shopping has been brought to the fingertips of users with e-commerce coming to the mobile front. Any web application is expected to have a counterpart version in the mobile. Inorder to avoid the redundancies between the web application and the mobile application, developers can save development time by creating a storage layer once, such as an API, with a centralized data storage and reuse it with multiple application versions.

There are libraries that support launching and running concurrent programs in the background in separate threads, network caching, and other features that significantly clean up network code. Android Volley and Retrofit are two such libraries. Android Volley was developed by Google while Retrofit was developed by Square.

Architecture plays an important role in developing android applications.

©IJRASET: All Rights are Reserved

Although MVC is the most widely used architecture because of its ease in implementation, it has several disadvantages such as high level of coupling of the Controller and the View. The evaluation of the performance of an application is needed to measure the rate of data loading and the application's connectivity to various operations, a few factors that can be attributed to this measurement are CPU utilization, execution speed and memory usage. The Snapdragon Profiler can be used to measure the real-time usage of the CPU and the Android Profiler can be used to measure the execution speed.

II. RELATED WORK

Parse database as a cloud database for developing Web and mobile applications and the services provided by them such as data integration, parse analytics to maintain the app data and handling of push notifications has been discussed in [1]. Presence of different components in the android framework has been a key finding of [3]. Three varieties of mobile applications such as native, web and hybrid have been elaborated in [2]. These can also be broadly classified as platform specific applications which utilize all the features of a particular device. Layered approach for developing android applications has been described in [4] which can be used for the development of web applications as well. Security measures with respect to android applications have also been discussed in this.

The life cycle of mobile applications and the reasons leading the users to uninstall applications from their mobile devices has been presented in [8]. [9] Predicts of data consumption behavioral patterns in mobile applications using data clusters.This done by using Wi-Fi local area network service provider logs to characterize the behavior in mobile applications. A report has been published by the OCAD University in [10] with innovation in service rendered to mobile applications. This report establishes a connection with companies and their growing dependency on mobile applications to increase their revenue. For this purpose they need to acquire real time data from the apps in order to make more sound decisions. [11], [12] & [13] discusses networking libraries used such as Android Volley and Retrofit.

In [18] 2443 commits were made on android applications of which 180 had 547 performance issues which were mostly attributed with GUI, networking, memory management and image loading. The findings of [15] suggests that although MVC is the most widely used architecture because of its ease in implementation, it has several disadvantages such as having a high level of coupling between the Controller and View. It also suggests that the increase in the data volume an application processes, the greater its effect is on the performance. Real-time CPU usage is done using the Snapdragon Profiler in [16] while execution speed is measured using Android Profiler in [17].

III. ARCHITECTURE

ListView contains an adapter which acts as a bridge between the data source and the user interface. The user interface pulls and retrieves the data from the data source which is either a Cursor or an ArrayList.

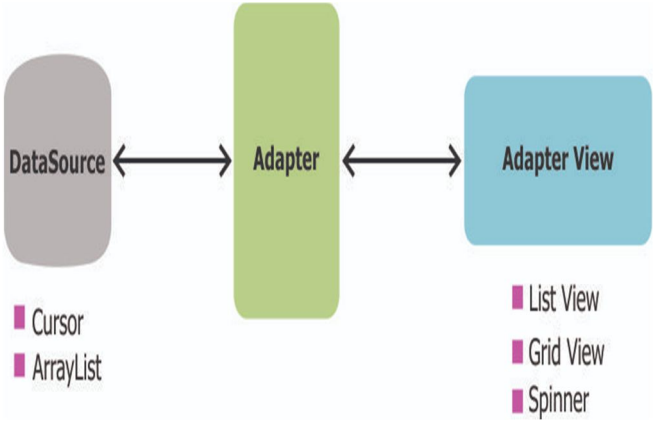


Figure 01: List View Architecture

Recycler View on the other hand is connected to a Layout Manager which tells the Recycler View how the layouts are to be formed for the different views. It is also connected to the adapter with the View Holder which sends the views to the Recycle View. The adapter provides new data to the Recycler View as and when needed which it retrieves from the data source where it binds the data to the views.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

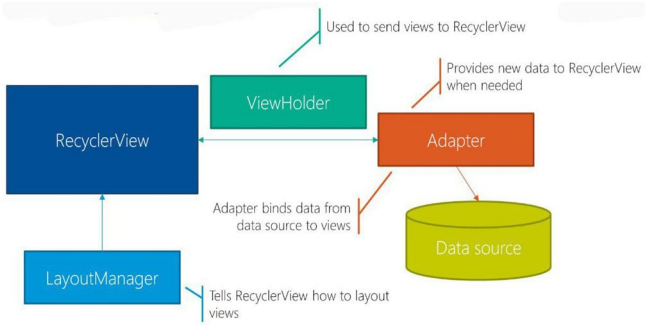


Figure 02: Recycler View Architecture

In Android Volley, network requests are queued and are sent to the cache dispatcher where each request is dequeued. Upon cache hit, the network request is read from the cache and parsed. Upon cache miss, the network dispatcher dequeues the request and network call takes place where HTTP transaction takes place followed by parse response and cache writing. The results from both parses are delivered to the main thread.

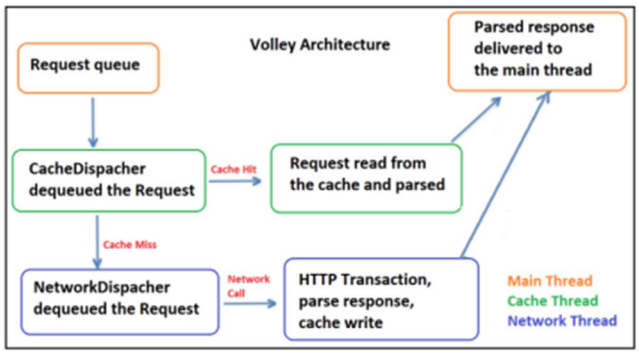


Figure 03: Android Volley Architecture

Retrofit APIs requests for the methods and manipulates the URL before sending it across to the REST adapter which uses GSON and other custom converters to handle the errors. They are then sent to generate HTTP responses which are requested from the web servers.

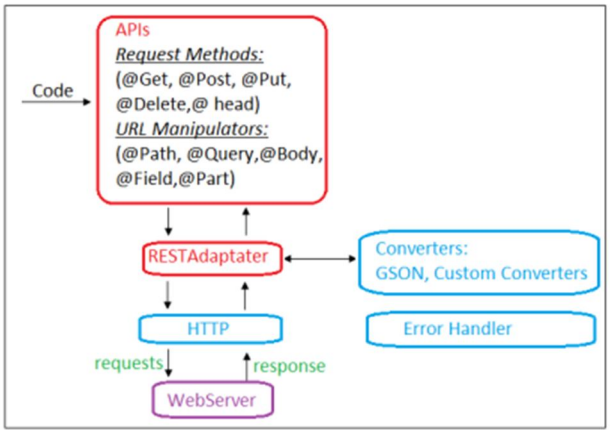


Figure 04: Retrofit Architecture

The View layer is where the user's touchpoint is. The operation performed by the user in the view layer is sent to the presentation layer which fetches the data from the service layer. The service layer performs a callback with the new data and sends the data to the presenter layer which updates the model layer and simultaneously updates the view of the user. This can be visualized in the figure below.

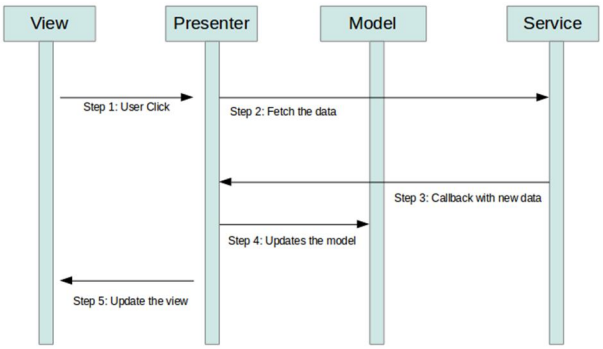


Figure 05: MVP Architecture

In the MVVM architecture, the View layer sends the UI events from the user's touch point to the View Model which is then changed in the Model layer and the model changed events are sent back to the View Model which sends it to the user's touch point, i.e, the View layer as Property Changed events. The View Model sends the update to the Model which reads into the View Model which sends the View Model data to the View.

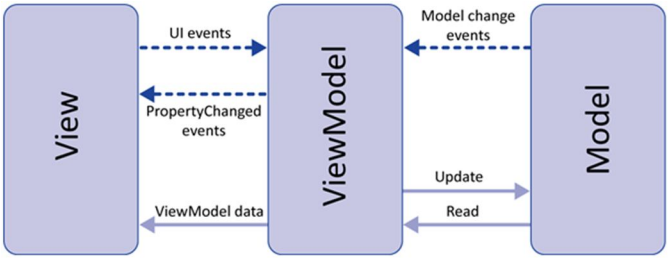


Figure 06: MVVM Architecture

IV. COMPARISON

View Holder pattern was never mandatory in ListView but it is mandatory in the Recycler View which helps speed up the rendering of the view. The view available in the List View is only vertical whereas in the Recycler View, the Linear Layout Manager supports both the vertical and horizontal views, the Staggered Layout Manager and the Grid Layout Manager helps render staggered and grid views as well. All this can be rendered dynamically in the Recycler View. The Recycler View brings a new dimension to the animation making it easy and intuitive. Intercepting items on the List View was comparatively simpler with Adapter View. On Item Click Listener which has become more complex in the Recycler View with the introduction of Recycler View. On Item Touch Listener.

Developed by Square, Retrofit is a clean, simple, lightweight library which acts as a REST client for developing powerful android applications which has a communication channel with remote databases. Retrofit can perform automatic JSON scans through synchronous and asynchronous queries.

Developed by Google, Android Volley is a HTTP library used to make networking operations in android applications quicker and easier. Asynchronous, synchronous, multiple requests can be performed on images, strings and JSONs. The requests are made in order of priority of caching. Volley provides caching, disk transparency, fetching web responses, etc.

Both Android Volley and Retrofit allow easy synchronous and asynchronous network calls with minimal code. For handling API request calls, Retrofit treats them as java method calls and specifies the request and response parameters as java classes. Volley on the other hand requires an extra bit of code for the parsing of the API. Volley has a special request to fetch the images from a network known as *Image Request* which catches the response as a bitmap. Retrofit, unlike Volley, combines *Ok Http Client* for loading the images.

Volley relies on common features such as request priority, bitmap caching, request caching, etc. Retrofit, however, has an intuitive API with serialization and request threading. Caching mechanism is more flexible and elaborate for Volley which supports prioritization unlike Retrofit. Volley uses caching to check for appropriate responses in order to return the parsed otherwise a network hit occurs.

Presenter layer is absent in MVVM, hence unlike MVP, observables are required for the user interface. Any change made to the view will affect the user interface and the presenter for MVP, in MVVM however, all the layers work independently which is useful for unit testing. MVP architecture is heavy with the vast amount of user interfaces. MVVM uses a lesser number of user interfaces making its architecture lighter. Separation of business logic and user interface supports integration testing in MVP while only unit testing can be performed in MVVM.

Due to the complexity of the layering pattern in MVVM, compilation time is much more than that of MVP. New releases can be handled better by MVVM than MVP due to the better maintenance ensuring capability provided by MVVM. Android applications are not the most suitable with the MVP architecture due to the fragments that occur at each stage in the life cycle. MVVM architecture however is extremely android suitaly made with updates in the software and the patches in the system helping the software to be up to date.

V. CONCLUSION

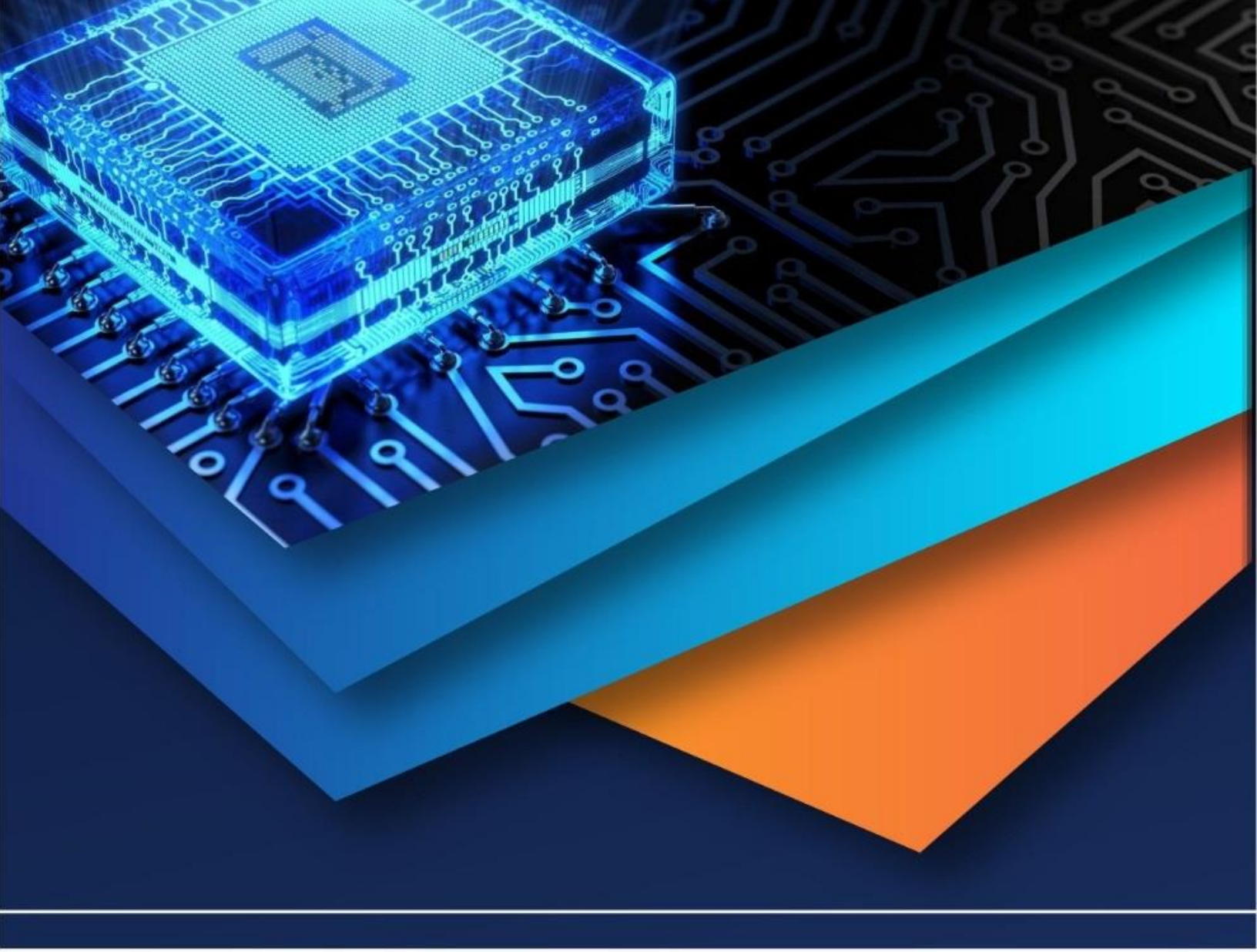
Recycler View is much more powerful, flexible and a major enhancement over List View. The Recycler View, unlike the List View, reuses allocated memory instead of allocating a new memory for each memory element in the view. The addition of the On Item Click Listener method can further enhance the efficiency of the Recycler View.

Although Android Volley and Retrofit have similar performances providing a callback function with an interface having two methods: success and failure. Android Volley specified the endpoint dynamically while calling the API returning a JSONArray/ JSON Object respectively. Retrofit on the other hand allows the user to set the endpoint of the URL when calling the API. When performed with easy API and complex APIs, Android Volley proved to be faster offering more features than Retrofit.

MVVM architecture has proven to show better performance than MVP with respect to the usage and execution speed. However, MVP had better memory utilization. The improvement in CPU utilization and execution speed in MVVM has come at the cost of higher memory utilization.

REFERENCES

[1] Lakshmi, B., Sai Durga, V., Anji Reddy, K.: Mobile quiz through wi-fi on android platform. Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET) 4(4) (2015)
[2] Kathuria, A., Gupta, A.: Challenges in android application development: a case study. Int. J. Comput. Sci. Mob. Comput. IJCSMC 4(5), 294–299 (2015)
[3] Gupta, P., Kumar, M., Sharma, M.: Architecture for mobile quiz application using android application framework. IPASJ Int. J. Comput. Sci. (IIJCS) 2(11) (2014)
[4] Holla, S., Katti, M.M.: Android based mobile application development and its security. Int. J. Comput. Trends Technol. 3(3), 486–490 (2012)
[5] G. McLean, K. Al-Nabhani et A. Wilson, " Developing a Mobile Applications Customer Experience Model (MACE) - Implications for Retailers", 2018, Journal of Business Research, Vol. 85, pp. 325–336
[6] A. Tarute, S. Nikou et R. Gatautis, "Mobile application driven consumer engagemen", 2017, Telematics and Informatics, Vol. 34, pp. 145–156.
[7] J. Song, et al , "Platform adoption by mobile application developers: A multimethodological approach",2018, Decision Support Systems, Vol. 107, pp. 26–39.
[8] A. Vagrani, N. Kumar et P. V. Ilavarasan, "Decline in Mobile Application Life Cycle", 2017, Procedia Computer Science, Vol. 22, pp. 957–964.
[9] K. W. Lima, et al. "Characterizing and predicting mobile application usage", 2016, Computer Communications, Vol. 95, pp. 82–94.
[10] L. Andrew, P. Sam et W. Kathleen," Mobile Innovation: Ontario's Growing Mobile Content, Services, and Applications. Toronto : Mobile Experience Innovation Centre", 2012.
[11] S. Yang, and H. Jieping. "Research and implementation of Web Services in Android network communication framework Volley." Service Systems and Service Management (ICSSSM), 2014 11th International Conference on. IEEE, 2014.
[12] F. Kirkpatrick, "Volley, easy, fast networking for android", Google.
[13] Square.github.io. (n.d.). "Retrofit". [online] at: http://square.github.io/retrofit/ [Accessed 4 Jan. 2018].
[14] D. Sillars, High Performance Android Apps: improving ratings with speed, optimizations, and testing. 2015.
[15] L. Tian, "A Comparison of Android Native App Architecture – MVC , MVP and MVVM," 2016.
[16] Qualcomm, Qualcomm ® Snapdragon TM Profiler, G. San Diego, U.S.A.: Qualcomm Technologies, Inc., 2020.
[17] developers.android.com, "Measure app performance with Android Profiler," 2020.
[18] T. Das, M. Di Penta, and I. Malavolta, "A quantitative and qualitative investigation of performance-related commits in android apps," Proc. - 2016 IEEE Int. Conf. Softw. Maint. Evol. ICSME 2016, no. January 2018, pp. 443–447, 2017, doi: 10.1109/ICSME.2016.49.
[19] N. S. Sibarani, G. Munawar, and B. Wisnuadhi, "Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin," 9th Ind. Res. Work. Natl. Semin., no. Juli, pp. 319–324, 2018.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)