Android application development for push notification feature for Indonesian space weather service based on Google Cloud Messaging

Siti Maryam¹, Adi Purwono¹, and Syahril¹
¹Space Science Center of BRIN
Email: siti.maryam@brin.go.id

Abstract. The advances of digital technology bring the era where the communication platform of modern society facilitates the spreading of information which is effortless yet distracting. This paper describes the development of a mobile application on the android platform as a service that provides information on Indonesian space weather conditions where the pushed-notification method is used to deliver information to the service’s stakeholders as the targeted users. The notification which presents the message directly to the users’ device is considered the most effective way to engage the users. A cross-platform message data builder based on GSON and Retrofit interface is implemented to construct information contents in the notification as well as in the application, suitable for use cases that occurred to accommodate the user preference. Furthermore, the application uses mobile device token identification to specifically manage the engagement process served by Firebase, a Google cloud messaging system. The application analytic indicates that the user-application retention of average users is about 80 percent and the success rate of delivered instant messages with push notification is 96 percent respectively during one month of testing from 48 registered users.

1. Introduction
The fact that mobile phones infiltrate almost every citizen and the communication network is growing rapidly which make the shifting of people’s behaviors, furthermore, nearly all the digital activities in recent days are greatly involved with their mobile phones [1]. Nowadays the mobile device is the main media that can penetrate to all users at any background at any time, such that it would be beneficial to develop the mobile application as a tool for the service provider or information supplier to gain attention to their audience [2]. The Indonesian space weather information and forecast service (SWIFtS) is a unique web-based service provider which delivers information of space environment parameters that are not only gained from ground-based observations in Indonesia but also generated by forecasting models [3]. There are three data groups that are updated every single day i.e., solar data, geomagnetic data, and ionospheric data. In addition, there are users and institutions that use specific SWIFtS data for their activities such international space weather community, the radio amateur enthusiasts, remote air-strip operators, to name a few.

The development of SWIFtS mobile application featuring push notification and application analytics is aimed to value the service level, to increase accessibility, and to focus on user engagement. SWIFtS backend server accommodates application on phones to access the updated data through an API (Application Programming Interface). The API consolidates content and data structure
to be delivered according to the request instance. It initiates sending the push notification message on android-based phones as well as reports the application activities and performance.

The push notification process consists of three domains within an android operating system they are worker, server, and initiator. The worker is composed of a message builder namely GSON and an interface called Retrofit. GSON is a Google initiated Java library used to convert Java objects into JSON (JavaScript Object Notation) representation on the android environment [4] and the Retrofit interface is a method of REST (representational state transfer) client where data stream flows inbound-outbound asynchronously between entities and understand to each other and is a Java converter designed for Android and Kotlin system [5]. The Firebase Cloud Messaging (FCM) acts as a server which provides the environment needed for the application to build, target, and send push notification at no cost and reliably. The FCM features the application analytic which is useful for the developer to understand the deployed application behavior and its performance based on cloud monitoring. The initiator is basically a set of use cases that represent the application workflow.

The push notification feature is used to broadcast alert messages of an earthquake event [6], while [7] used the feature as a tool for marketing purposes for their customer. Whereas push notification is proposed for MQTT (Message Queuing Telemetry Transport) protocol for Internet of Things (IoT) devices to solve the problem of instant pushing various messages from connected IoT devices to the server [8].

2. Methodology

The SWIFS space weather information is updated daily and at a certain big event happened such as an impulsive solar storm that generates severe disturbance for satellite operational and high-frequency radio communication. The information should be updated promptly so that the user is aware of the effect of that event, however, as for that case the web-based service is not the right solution. The development of the SWIFS mobile application’s main concern is to attain close engagement to users with a push notification feature, such that a direct and minimum delay of information delivery can be achieved.

2.1 Application server environment

The SWIFS android-based mobile application has three main features which rely on the Firebase cloud service environment as the backend system, they are user authentication/token registration, push notification, and application analytic/crash report presented on Firebase cloud-based dashboard as in figure 1.

![Figure 1. The application features.](image)

The cloud service provides a method to register the mobile client application instance succeeding the application installation process which assigned a user registration token ID or FCM Token as a unique identifier, as pictured in figure 2. The FCM works on the Google Cloud ecosystem which guarantees the platform-level transport layer is working flawlessly responsible for routing the message/notification at best yield [9].
2.2 Cloud database and dedicated database co-service
The application development implements two-separate database servers e.g., the cloud database server and dedicated in-house database server for SWIFTS as depicted in figure 3. The cloud database service provided by Firebase is used to keep the token ID data at maximum security to prevent breaches. The FCM user registration token ID is a unique identifier, built from the IMEI (mobile equipment identity) number and the code model of the user’s device where that information is very sensitive related to privacy [10]. The token ID data transaction is made within a trusted and encrypted channel of the Google Cloud Platform. Whereas the dedicated in-house database server supplies not only the space weather data for web-based content but also supplies data for the push notification payload. The query process for composing a push notification instance is divided into two steps.

The first step is to query the tag ID as the primary key of the user’s data from the user database located in the cloud database, this step outputs a list of encrypted token IDs as addresses of notification recipients. The second is to query any information of SWIFTS data to be published according to the SQL query executed. All the query actions are embedded on the Retrofit interface where such interface has customizable properties that make the conversion of the data stream to any data type such as JSON, GSON, and XML format is working seamlessly.
2.3 **Retrofit framework implementation**

The designed mobile application in the Android platform implements the Retrofit type library for the client-side device, it provides a modern yet flexible framework for interacting through API server. The SWIFtS database, Cloud database, and FCM are basically web-based services that most of their API’s operations are relying on the HTTP request method, such that there is a need for a type function interface in the Android environment [11]. Retrofit turns the HTTP request API into a Java/Kotlin interface so that it simplifies the conversion process by using annotations to define how requests are called on the Android operating system. Moreover, Retrofit transforms the REST API on the Android device into a Java interface which is very light and straightforward for any data transaction, it allows to run HTTP requests from the designed android application.

2.4 **Application use cases**

The android application has four actors they are Operator, SWIFtS server, Firebase server, and Users. Each actor initiates its own use cases which consist of one task or more to be performed by the corresponding actor. There are eight use cases which are constructing the system’s behaviors as seen in figure 5. The Operator and the SWIFtS server are identical actors, their duty is to initiate the push notification instance. However, the SWIFtS server is designed to initiate the push notification message periodically as well as automatically after SWIFtS daily forecast activity while the Operator initiates the instance manually by using a graphical user interface (GUI) for specific event or alert.

![Figure 5. Designed application use cases.](image)

3. **Result and discussion**

Figure 6 illustrates the Unified Modelling Language (UML) class diagram of the application derived from the systems use cases, each class contains attributes and methods. The class represents how the use case is executed based on its resource-level relationship and its logical scheme. The notification_builder class has a dependency on both the subscriber_DB class and swiftsparm_DB class while the fcm_instance class is inherited by the notification_builder class. Recipients_ class has on_receive attribute and notif_swiped attribute with both values are Boolean, these values are used as notification flag parameter when logging the application activities.
The Android application is deployed on Google Play environment, downloaded and installed limited to 48 of users/devices which have variety on Android software version from lowest Android 7.0 (Nougat) to the latest of Android 11 as well as a variety of on-device models, the information is then collected after the application installation process. From an operating system (OS) perspective, the designed application should run perfectly on Android 8 and up, since the library dependency is fully compatible on Android 8 and up [12].

**Figure 6.** SWIFTS push notification class diagram.

**Figure 7.** The Application screens.
For a period of four weeks 1909 notifications were broadcasted to all user’s devices which varies in models and OS versions, the analytic results 92 percent of notification was successfully delivered during that period. As the application crashed, unavailable network service and lack of internal device’s memory are the several reasons for being as many as 73 notifications were undelivered.

The Firebase server logs and records the application analytics for each Android device as part of the Google Cloud service. The application activities such as user-application interaction (retention) are important parameters as an indication of user involvement over application features. As the notification is successfully delivered to users, there are two kinds of user response, the user taps the notification and the opposite user dismisses it. After the user taps the notification message, the application’s main activity initiates an instant view of the application dashboard (figure 7) which displays the space weather information, at this state the user is considered active. The states are logged at the period of one month, the chart of user-application retention can be seen in figure 9. About 80 percent of active users was recorded during 31 days of the testing campaign however on the weekend period (Saturday and Sunday, day 4 and 5 for the examples) the number of active users tend to be lower than on the weekdays.

4. Conclusion
A vast variety of Android device models and OS versions is one of the big challenges when deploying an Android application. A hundred percent successful application that can run on every device is difficult since the implementation of the software is specifically dependent on the OS and the device specification. The analytic data in figure 9 reveals that the application usage is in accordance with the user behaviors, the SWIFTS information was less needed at the weekend. However, the developed
push notification functionality is worked as expected yet effective in gaining users’ attention, it was at nearly 80 percent of the average active user daily on the first month of testing.

Acknowledgments
The authors thank the editor and reviewer for the constructive comments and suggestions on the manuscript. All of the authors that contribute in this paper, Siti Maryam provides the list of application requirements and the use cases, Adi Purwono implements the use cases on android application, prepares the web-based API and writing the article, and Syahril provides the testing data results.

References
[1] Ravipati K. (2020), Mobiles Have Changed the Way We Communicate, International Journal of English Research. Volume 6, Issue 6, 2020, Page No. 40-43, ISSN: 2455-2186
[2] Rashidul I., Rofiqul I. and Tohidul A. (2010), Mobile Application and Its Global Impact International Journal of Engineering & Technology IJET-IJENS Vol: 10 No: 06
[3] Y. Andrian, E. Maryam, K. DaeYong, S. H. Maeng and H. Ju (2019), Space Weather Data Management System and Monitoring in Decentralized Storage Environment, 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), 2019, pp. 1-4, doi: 10.23919/APNOMS.2019.8892999.
[4] Ji, Zhanlin, Ganchev, Ivan, O’Droma, Mairtin and Zhao, Q. (2014). A Push-Notification Service for Use in the UCWW. 10.1109/CyberC.2014.88.
[5] R. T. Fielding, R. N. Taylor, J. R. Erenkrantz, M. M. Gorlick, J. Whitehead, R. Khare, and P. Oreizy (2017), Reflections on the REST Architectural Style and Principled Design of the Modern Web Architecture. ESEC/FSE August 2017 Pages 4–14, doi:10.1145/3106237.3121282
[6] Rahmad Maulidi, R., Kristanto, B.K. and Listio, Y.D. (2020), Earthquake Information Push Notification System in Android Application Using Google Firebase. International Journal Information System and Computer Science (IJICS)
[7] Firdaus, D., Priambodo, B. and Jumarya, Y. (2019), Implementation of Push Notification for business incubator. International Journal of Online and Biomedical Engineering (iJOE) 15(14):42
[8] Tang, K., Wang, Y., Liu, H., Sheng, Y., Wang, X., and Wei, Z. (2013), Design and Implementation of Push Notification System Based on the MQTT Protocol. ISCA 2013
[9] Guido Albertengo, Fikru G.Debele, Waqar Hassan and Dario Stramandino (2020), On the performance of web services, google cloud messaging and firebase cloud messaging. Digital Communications and Networks Volume 6, Issue 1, February 2020, Pages 31-37
[10] Li, Wanpeng and Mitchell, Chris. (2016), Analysing the Security of Google’s Implementation of OpenID Connect. 357-376. 10.1007/978-3-319-40667-1_18
[11] Siduzzaman, Hossan Alom, Rakibul Sarwar and Talha Miah (2020), Performance comparison of HTTP/2 for Common E-Commerce Web Frameworks with Traditional HTTP. Journal of Physics: Conference Series. 1529. 052023. 10.1088/1742-6596/1529/5/052023.
[12] J. Liu and J. Yu (2011), Research on Development of Android Applications, 4th International Conference on Intelligent Networks and Intelligent Systems, 2011, pp. 69-72, doi: 10.1109/ICINIS.2011.40