Notification Log: An Open-Source Framework for Notification Research on Mobile Devices

Dominik Weber
VIS, University of Stuttgart
Stuttgart, Germany
Dominik.Weber@vis.uni-stuttgart.de

Niels Henze
University of Regensburg
Regensburg, Germany
Niels.Henze@ur.de

Alexandra Voit
VIS, University of Stuttgart
Stuttgart, Germany
Alexandra.Voit@vis.uni-stuttgart.de

UbiComp/ISWC '18 Adjunct, October 8–12, 2018, Singapore, Singapore
ACM 978-1-4503-5966-5/18/10.
https://doi.org/10.1145/3267305.3274118

Abstract
In the past decade, the number of always-connected mobile devices exploded. Smartphones are always with the user and host a large number of applications and services that use notifications to gain the user's attention. These notifications and their effects on users have been extensively researched in the context of human-computer interaction. In a body of prior work, numerous small- and large-scale studies were conducted to understand notifications as well as their effects. A common theme in these studies is the need for accessing users' notifications, often for logging purposes. In this paper, we present an open-source framework for notification research on mobile devices. The framework has been used as the foundation of multiple in-the-wild and in-lab studies, and has been downloaded by over 60,000 users. We explain the requirements, the architecture, and past application scenarios of the framework. The scenarios range from enabling reflection on mobile notifications to rich experiences in multi-device environments.

Author Keywords
Notifications; interruptions; mobile devices; digital well-being; data collection; open-source framework.

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction

Notifications are an important feature of current mobile devices. Applications and services on mobile devices use notifications to proactively inform users about various events, such as new messages, emails, appointments, and updates. Prior work investigated what types of notifications users receive on a daily basis and which notifications are valued by users. Regardless of their value, notifications can induce negative effects such as distractions, interruptions, and increased stress. A body of work investigated various approaches to reduce these negative effects such as deferring notifications to breakpoints, using context data to assess the user's interruptibility or filtering unwanted notifications. A common theme in these studies is the need for accessing users' notifications, often for logging purposes.

In this paper, we present *Notification Log*, an open-source framework for logging notifications on mobile devices. We implemented *Notification Log* as a modular Android app that can be easily extended. The framework has been used in multiple in-the-wild and in-lab user studies, and has been downloaded by over 60,000 users. By making the framework publicly available, we hope to accelerate the research of notification experiences that are valued by users while respecting their digital well-being. In the following, we outline related work about notifications on mobile devices. Then, we introduce *Notification Log*'s architecture and provide an overview of prior application scenarios.

Related Work

A body of prior work investigated various aspects of mobile notifications. Sahami Shirazi et al. [15] conducted a large-scale assessment of mobile notifications. The researchers developed an app for Android devices which listens for new notifications and subsequently forwards the notifications to a central server for data analysis. The app was later extended by Weber et al. to support multiple devices [18]. In follow-up work, Weber et al. showed that users perceive notifications differently depending on the type of device [21]. Lee et al. provided further insights about notifications on smartwatches [6]. Dingler et al. investigated the use of notifications for microlearning sessions on the go [4]. The researchers built an Android app that triggered microlearning notifications at random times and depending on a model [14] that also included the time of the last received notification. Pielot et al. [13], Dingler et al. [3], and Mehrotra et al. [9, 10] logged mobile notifications to provide further insights into what kind of notifications users receive on a daily basis and how they are perceived. Mehrotra et al. also presented a system that generates notification rules based on received notifications [8]. Weber et al. investigated the user-defined deferral of notifications [19]. Auda et al. explored user-defined rules for notification deferral and end-of-day summaries [1]. Okoshi et al. presented a middleware to sense users' interruptibility in real time [11], which was used in a large-scale in-the-wild study [12]. Leiva et al. [7] reported on the cost of mobile application interruptions, and Vardhan et al. discussed the balance of convenience and privacy of mobile notifications [16].

A common theme in prior work is the need for accessing or logging users' notifications. The contribution of this paper is an open-source notification logging framework that can act as a starting point for future work.

Notification Log

We implemented the *Notification Log* framework as an Android app with the primary goal of running in the background of personal mobile devices for both in-the-wild and in-lab studies. The requirements for the app are reliable and unobtrusive logging in the background, extensibility, as well as support for most Android versions and devices.

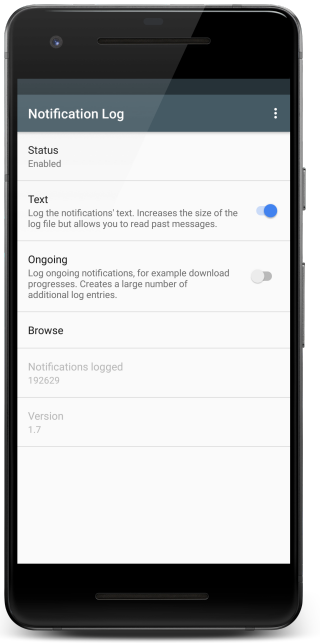

Figure 1: The settings screen allows users to control if and how notifications should be logged. The screen is extensible and allows for a simple integration of additional options.

Data Sources

The *Notification Log* framework consolidates multiple data sources and provides an abstraction layer for a wide range of Android versions (see Figure 2).

Notification Listener Service The central data source of the framework is the Notification Listener Service API [2]. This service is, after granting permission from the user, permanently running in the background of the device and receives callbacks when a notification is added or removed from the system. Recent versions of Android significantly improved the information provided by this API, e.g., by providing information if a notification was removed by the user or the notifying app itself. The API is available since Android 4.3 which runs on 96.40% of Android smartphones and tablets at the time of writing[1]. In particular, this service provides which apps triggered/removed notifications, the text content, priority/importance levels, vibration patterns, and sound amongst a multitude of additional attributes.

Device Meta Data and Context Data *Notification Log* samples device meta data, such as the screen state (on/off), ringer mode (silent, vibration, volume), battery state (current level, if charging), and connectivity state (offline, wifi, mobile). This information is combined with the notification data to provide insights into the device's context when a notification was received or removed.

Additional Data Sources Other data sources can be easily integrated due to the extensible architecture of the framework. In the past, we used the Google Activity Recognition API to extend the notification data. The API reports probabilities for if the device is still or the user is walking, running, cycling, or driving. Another data source is the Google Location API, which reports location updates including longitude, latitude, estimated accuracy in meters, and the age of the location update. Both data sources can be attached to the Notification Listener Service and enhance the logged notification data. Notably, the framework allows the use of these data sources without affecting the device's performance or battery consumption negatively.

Data Consolidation

Notification Log consolidates all data sources in a central Notification Object (highlighted in green in Figure 2). This object acts as an abstraction layer that hides differences in the Android SDK. The result of the data consolidation is serialized to a JSON representation.

Data Persistence

Notification Log stores all unified JSON objects in a private on-device SQLite database (highlighted in yellow in Figure 2). The database is inaccessible by other apps on the device and therefore enables secure logging of notification and context data.

Data Processing

The data stored in the local database is the core of the framework. The *browse* user interface allows users to see recent additions to the log (see Figure 3). It shows a preview of the recorded notification and the JSON representation of the notification and context data (see Figure 4). The database can also be exported manually in a JSON-line format for computation by other applications. Events can be sent to a server either immediately when a notification is added or removed from the system or batched for synchronization in specific intervals when pre-defined conditions are met, e.g., when the device is plugged in and connected to a Wi-Fi network.

[1] Android distribution dashboard https://developer.android.com/about/dashboards/ *(accessed on 2018-07-26)*

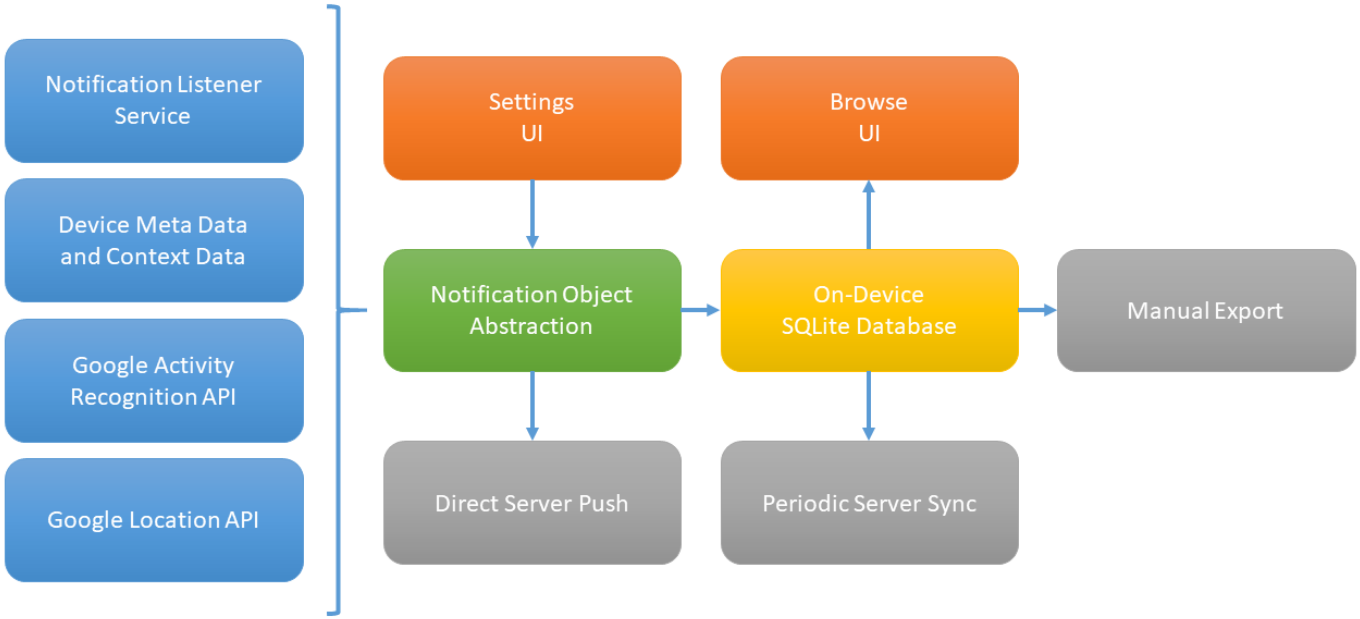

Figure 2: The architecture of *Notification Log*. The framework provides an abstraction layer for unifying notification data with device meta data and context data. The logs can be further extended by including additional data sources, such as the Google Activity Recognition API and the Google Location API. The logged data is stored in an on-device database and can be manually exported, synced periodically or directly pushed to a server. Users can control the logging with an extensible settings screen and browse the locally stored data.

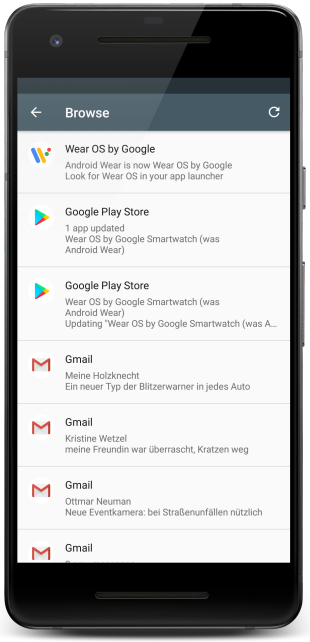

Figure 3: The browse screen allows users to explore the on-device database. It provides a preview of the logged notifications.

Extensibility
The framework can be extended in multiple ways, e.g., by adding additional data sources to the notification object abstraction layer. The existing UIs can be extended by further options that control the logging (see Figure 1) or statistics that provide additional insights about the logged data.

Application Scenarios

The *Notification Log* framework was used in a number of in-the-wild and in-lab studies. For each of these studies, the framework was extended according to the needs of the study. In the following, we report prior applications of the framework and detail the release on the Google Play Store.

Record and Replay of Notifications
Lab studies that involve personal notifications are challenging. When using participants' personal devices, it is not possible to control how many notifications they receive during the study, as this mostly depends on external factors. On the other hand, using fake notifications may be perceived differently by the participants. As a compromise between those approaches, *Notification Log* was used in a lab study setup [17]. The app was installed on the participants' personal smartphones days prior to the lab study. On the day of the study, the recorded notifications were exported and a set of randomly selected notifications were then used in the lab study. By displaying participants' own recorded notifications, it was possible to create a compromise between meaningful notifications while controlling the number of notifications shown during the study.

Reflection on Mobile Notifications
In another study, the framework was used to enable reflection on mobile notifications [22]. In current mobile operating systems, notifications are ephemeral. To provide insights on how many notifications users receive on a daily basis, *Notification Log* was used to record all notifications on the users' personal smartphones for a specific time span. The log data was then exported and loaded in a web-based dashboard. The dashboard breaks down the number of notifications that were created in the time span, along with the apps that created them, the change over time, and the differences between weekdays. This dashboard allows users to reflect on the notifications they receive and enables them to adjust notification settings to improve their digital well-being. In this specific study, the logs were exported manually, but it is easy to imagine a system in which the notifications are periodically synced with a server to provide an always up-to-date dashboard experience.

Integration in Existing Infrastructures
Notification Log was integrated into an existing infrastructure for IoT devices in intelligent living environments [5]. Thanks to the loose coupling of the components, it was possible to quickly integrate the Notification Listener Service and the Notification Object Abstraction layer into an existing project. The existing IoT infrastructure was therefore extended by the possibility to react to notifications received on personal smartphones and tablets.

Novel Experiences
In a study to explore the effect of personal content on public displays in work environments, the framework was extended to communicate with nearby public displays [20]. Participants of the study carried Bluetooth beacons. When being near a public display, the modified *Notification Log* app would mirror the pending notifications of the participants' personal smartphones on the public display. For this study, the framework was extended by a component that handles the communication with the public display, and an additional *privacy* screen that allowed the participants to control the level of detail of the mirrored notifications.

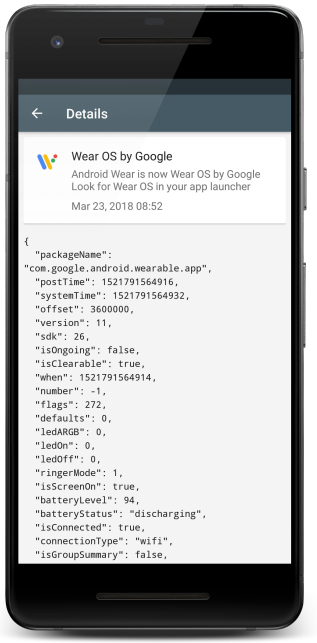

Figure 4: The details screen shows a preview of a logged notification and the corresponding JSON representation.

Availability on the Google Play Store

The base version of *Notification Log* has been available in the Google Play Store since July 2015[2]. At the time of writing, the app has been downloaded over 60,000 times and is installed on over 8,700 Android devices. These downloads originated from 175 countries. Users installed the app on over 360 different Android device models, with Android versions ranging from 4.3 to 8.1.

Open-Source Framework

The *Notification Log* framework was used in a number of in-the-wild and in-lab studies. Due to its extensible architecture, it can be quickly customized for many application scenarios. The base version of the framework has also been used by thousands of users over the span of several years. We now open-sourced the *Notification Log* framework[3] under the MIT license to provide researchers and developers with a flexible framework for notification-related research and projects.

Conclusion

In this paper, we introduced *Notification Log*, an open-source framework for notification research on mobile devices. The framework was implemented as an Android app for a wide range of Android smartphones and tablets. We discussed the architecture of the framework and provided examples of how the framework was used in in-the-wild and in-lab studies. The base version of the framework was made available on the Google Play Store and has been downloaded over 60,000 times on a broad set of different Android devices.

With this publication, we open-source the framework to provide researchers and developers with a foundation for future projects. We hope to accelerate the research of notification experiences that are valued by users while respecting their digital well-being.

[2] *Notification Log* on the Google Play Store https://play.google.com/store/apps/details?id=org.hcilab.projects.nlog

[3] *Notification Log* on GitHub https://github.com/interactionlab/android-notification-log

REFERENCES

1. Jonas Auda, Dominik Weber, Alexandra Voit, and Stefan Schneegass. 2018. Understanding User Preferences Towards Rule-based Notification Deferral. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. ACM, New York, NY, USA, Article LBW584, 6 pages. DOI:http://dx.doi.org/10.1145/3170427.3188688

2. Android Developers. 2018. NotificationListenerService. https://developer.android.com/reference/android/service/notification/NotificationListenerService.html. (2018). [Online; accessed 26-July-2018].

3. Tilman Dingler and Martin Pielot. 2015. I'll Be There for You: Quantifying Attentiveness Towards Mobile Messaging. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 1–5. DOI: http://dx.doi.org/10.1145/2785830.2785840

4. Tilman Dingler, Dominik Weber, Martin Pielot, Jennifer Cooper, Chung-Cheng Chang, and Niels Henze. 2017. Language Learning On-the-go: Opportune Moments and Design of Mobile Microlearning Sessions. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA, Article 28, 12 pages. DOI: http://dx.doi.org/10.1145/3098279.3098565

5. Thomas Kubitza, Alexandra Voit, Dominik Weber, and Albrecht Schmidt. 2016. An IoT Infrastructure for Ubiquitous Notifications in Intelligent Living Environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp '16)*. ACM, New York, NY, USA, 1536–1541. DOI: http://dx.doi.org/10.1145/2968219.2968545

6. Jemin Lee, Jinse Kwon, and Hyungshin Kim. 2016. Reducing Distraction of Smartwatch Users with Deep Learning. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. ACM, New York, NY, USA, 948–953. DOI: http://dx.doi.org/10.1145/2957265.2962662

7. Luis Leiva, Matthias Böhmer, Sven Gehring, and Antonio Krüger. 2012. Back to the App: The Costs of Mobile Application Interruptions. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 291–294. DOI:http://dx.doi.org/10.1145/2371574.2371617

8. Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. 2016. PrefMiner: Mining User's Preferences for Intelligent Mobile Notification Management. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 1223–1234. DOI:http://dx.doi.org/10.1145/2971648.2971747

9. Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 813–824. DOI: http://dx.doi.org/10.1145/2750858.2807544

10. Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. 2016. My Phone and Me: Understanding People's Receptivity to Mobile Notifications. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1021–1032. DOI: http://dx.doi.org/10.1145/2858036.2858566

11. Tadashi Okoshi, Jin Nakazawa, and Hideyuki Tokuda. 2014. Attelia: Sensing User's Attention Status on Smart Phones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 139–142. DOI: http://dx.doi.org/10.1145/2638728.2638802

12. T. Okoshi, K. Tsubouchi, M. Taji, T. Ichikawa, and H. Tokuda. 2017. Attention and engagement-awareness in the wild: A large-scale study with adaptive notifications. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 100–110. DOI: http://dx.doi.org/10.1109/PERCOM.2017.7917856

13. Martin Pielot, Karen Church, and Rodrigo de Oliveira. 2014. An In-situ Study of Mobile Phone Notifications. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 233–242. DOI: http://dx.doi.org/10.1145/2628363.2628364

14. Martin Pielot, Tilman Dingler, Jose San Pedro, and Nuria Oliver. 2015. When Attention is Not Scarce - Detecting Boredom from Mobile Phone Usage. In

Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15). ACM, New York, NY, USA, 825–836. DOI:http://dx.doi.org/10.1145/2750858.2804252

15. Alireza Sahami Shirazi, Niels Henze, Tilman Dingler, Martin Pielot, Dominik Weber, and Albrecht Schmidt. 2014. Large-scale Assessment of Mobile Notifications. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 3055–3064. DOI: http://dx.doi.org/10.1145/2556288.2557189

16. Raj Vardhan, Ameya Sanzgiri, Dattatraya Kulkarni, Piyush Joshi, and Srikanth Nalluri. 2017. Notify Assist: Balancing Privacy and Convenience in Delivery of Notifications on Android Smartphones. In *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society (WPES '17)*. ACM, New York, NY, USA, 17–20. DOI:http://dx.doi.org/10.1145/3139550.3139561

17. Dominik Weber, Sven Mayer, Alexandra Voit, Rodrigo Ventura Fierro, and Niels Henze. 2016. Design Guidelines for Notifications on Smart TVs. In *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video (TVX '16)*. ACM, New York, NY, USA, 13–24. DOI: http://dx.doi.org/10.1145/2932206.2932212

18. Dominik Weber, Alireza Sahami Shirazi, and Niels Henze. 2015. Towards Smart Notifications Using Research in the Large. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '15)*. ACM, New York, NY, USA, 1117–1122. DOI: http://dx.doi.org/10.1145/2786567.2794334

19. Dominik Weber, Alexandra Voit, Jonas Auda, Stefan Schneegass, and Niels Henze. 2018a. Snooze! Investigating the User-defined Deferral of Mobile Notifications. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18)*. ACM, New York, NY, USA, Article 2, 13 pages. DOI: http://dx.doi.org/10.1145/3229434.3229436

20. Dominik Weber, Alexandra Voit, Gisela Kollotzek, Lucas van der Vekens, Marcus Hepting, Florian Alt, and Niels Henze. 2018b. PD Notify: Investigating Personal Content on Public Displays. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. ACM, New York, NY, USA, Article LBW014, 6 pages. DOI: http://dx.doi.org/10.1145/3170427.3188475

21. Dominik Weber, Alexandra Voit, Philipp Kratzer, and Niels Henze. 2016a. In-situ Investigation of Notifications in Multi-device Environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 1259–1264. DOI:http://dx.doi.org/10.1145/2971648.2971732

22. Dominik Weber, Alexandra Voit, Huy Viet Le, and Niels Henze. 2016b. Notification Dashboard: Enabling Reflection on Mobile Notifications. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. ACM, New York, NY, USA, 936–941. DOI:http://dx.doi.org/10.1145/2957265.2962660