Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 198 (2022) 203–210

Procedia Computer Science

www.elsevier.com/locate/procedia

The 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH 2021)
November 1-4, 2021, Leuven, Belgium

LAMP: a monitoring framework for mHealth application research

Michael Stach[a,*], Florian Pflüger[a], Manfred Reichert[a], Rüdiger Pryss[b]

[a] *Institute of Databases and Information Systems, Ulm University, Germany*
[b] *Institute of Clinical Epidemiology and Biometry, University of Würzburg, Germany*

Abstract

The usage of mobile applications in healthcare has gained popularity in recent years. In 2018, at least, 10,000 apps related to mental health could be downloaded in the app stores. The popularity of healthcare apps, especially in the field of mental health, is based on in their simplicity in large-scale data collection scenarios used for the improvement of health-related services or research. For these apps, instruments to quantify the quality of an app and repositories for app quality ratings have emerged in recent years. What is rarely considered, however, is the degree of functional correctness of an app, which can have a serious impact on the data collection process and thus on data quality. The increasing restrictions of background services are a challenge for app developers, who need to implement recurring tasks reliably in the background, like the collection of longitudinal data based on questionnaires or sensor measurements. In this paper, we present a monitoring framework to investigate the degree of functional correctness regarding the background service implementation of apps based on notification events. With this framework, we want to enable the large-scale collection of app execution data in the wild to gain more insights into the execution of apps in different execution environments and configurations. The gained knowledge shall help to improve existing applications in the field of mental health and eventually to improve the degree of functional correctness of those apps.

Keywords: mHealth; Digital Phenotyping; Mobile Crowdsensing; Ecological Momentary Assessment; App Monitoring; Notification

1. Introduction

The digitization of healthcare has been advancing for years. Recently, caused by the COVID 19 pandemic, this trend has received a further boost. Every day, over 200 additional apps are being added to the existing basis of over 300,000 apps that are available in the two major app stores (i.e., Google Play Store and Apple App Store) [6, 11]. mHealth apps (i.e., apps for healthcare purposes) make up an ever-increasing share of this app repository, as the

* Corresponding author. Tel.: +49 731 50 24 225 ; fax: +49 731 50 24 134.
E-mail address: michael.stach@uni-ulm.de

10.1016/j.procs.2021.12.229

diverse capabilities of smartphones for data collection has shown to be highly effective [4]. For example, in 2018, there were at least 10,000 apps related to mental health in the app stores [22].

Regarding the increasing number of apps in the stores, the demand for apps with a better quality has emerged. The two major stores make great efforts to review new apps before they are published in the store, with most of the reviews being related to possible violations of the store's terms and conditions. A content review, in turn, would exceed the testing resources of the store operators given the large number of apps in the stores. Since content review of mHealth applications is of general interest to both the app users and researchers as well as practitioners, app quality review instruments (e.g., in the form of standardized questionnaires) have emerged in various areas. Regarding mHealth, instruments like the app evaluation model of the American Psychological Association [1], the Evaluation Tool for Mobile and Web-Based eHealth Interventions (ENLIGHT) [2, 19], or the Mobile App Rating Scale (MARS) [18] and its German version (MARS-G) [12] were therefore developed. Since the latter instruments are mainly used in a research context, platforms like the Mobile Health App Database (*MHAD*) [16] try to collate app quality ratings and present the results to a broader audience. These efforts are intended to assist end users in selecting appropriate mHealth apps based on app quality ratings by domain experts.

Interestingly, much of the research related to app quality has focused on app content ratings. Research related to the implementation quality of the app (i.e., the implementation and execution of the app, including all of its features and services, in line with expectations and requirements) has been the subject of only a few works, although various quality models already exist in the field of software engineering. The Systems and Software Quality Requirements and Evaluation (*SQuaRE*), for example, is part of the ISO/IEC 25010 standard family [7] and provides a product quality model that defines aspects of external and internal quality of software products. With respect to the implementation quality of apps, *SQuaRE* offers the characteristic *functional suitability* with the sub-characteristic *functional correctness*, which is defined "as the degree to which a product or system offers correct results, with the required degree of precision" [14]. When we use the term *software product quality* in the following, we refer to the definition of the *functional correctness* characteristic of the *SQuaRE* product quality model.

In existing research projects of the authors, which pose an intensive app use (e.g., TrackYourTinnitus [10] or Corona Health [3]), we were able to determine that apps on various devices with different Android versions or configurations led to different behavior. The latter affected the app's functionalities, for example, by not correctly executing background services like GPS tracking or receiving notifications from server applications. Since the mental health research method used in the aforementioned projects requires reliable adherence to data collection schedules, a high degree of *software product quality* is crucial to data quality. Especially when using modern research methods like *Digital Phenotyping* that heavily depend on smartphone-based data collection, various requirements have to be considered.

As there is a plethora of different smartphone configurations (e.g., hardware, OS version, OS vendor, user settings, etc.), creating a generalizable view on this topic requires a monitoring approach that is both versatile and suitable for large-scale use. In this work, we want to present *LAMP*: a configurable monitoring framework for large-scale mHealth application research. With *LAMP*, we want to investigate the behavior of smartphone apps in the wild in order to gain insights into the execution of Android apps. The framework is designed to host multiple large-scale studies (including automated data reporting) and with a focus on low resource consumption and data minimization. Eventually, *LAMP* shall help practitioners and developers to design high quality mHealth apps.

The remainder of this work is structured as follows: In Section 2, background information regarding the app execution on Android as well as related work are described. Subsequently, the proposed framework, including a prototypical implementation, is presented in Section 3. Section 4 gives insights into the evaluation of the prototype. Finally, the work concludes with a summary and outlook in Section 5.

2. Background & Related Work

The use of mobile apps in healthcare is widespread. Often, data are collected in order to subsequently evaluate, interpret, and provide some form of value to the data creators (e.g., providing better treatments). In the area of mental health, for example, apps and sensor technology are often used to perform a *Digital Phenotyping* [8]. Torous et al. define *Digital Phenotyping* as "moment-by-moment quantification of the individual-level human phenotype in-situ using data from smartphones and other personal digital devices." [19]. To achieve this, mobile sensing methods (e.g., *Mobile Crowdsensing* (MCS)) are often used in combination with other research methods. An appropriate research method to achieve both the "moment-by-moment" and the "in situ" aspects of the Torous et al. definition of *Digital Phenotyping* is called *Ecological Momentary Assessment* (EMA). The latter is more common in the research environment of psychology and describes the periodic measurement of individuals' experience, behavior, and psychological and physical well-being [15]. The combination of EMA and MCS [9] has shown its potential for research in healthcare in various studies [10, 17, 3].

According to Smyth et al., measurements using the EMA can increase data quality by reducing the recall bias, increase ecological validity, and track processes that occur within the individual [15]. To achieve this, the user must be prompted to collect data at a time unknown to him/her or, if no user interaction is required, the app must be notified to start collecting data in the background. Since notifications (i.e., messages that are display by the OS to provide reminders or other timely information) are crucial for studies with complex, intertwined notification schedules, it is worth taking a closer look at the limitations of the operating systems and app execution in general.

Operating system features and interfaces are subject to constant changes. Changes to the behavior of an API or operating system services can be error-prone for all those apps that have not yet been tested for a new operating system version. Petter et al. [13] describe in their work the behavioral change of their MCS app with different Android versions. They further describe that the background data collection was affected by the introduction of various energy saving features like the *Doze Mode* and *App Standby* in Android version 6. The *Doze Mode* is activated when an Android smartphone is both unused for a longer period and unplugged from the power supply. While *Doze Mode* defers all background activities (OS services excluded) to a recurring so-called "maintenance window", *App Standby* restricts background network activity of infrequently used apps. Problems with these OS-related restrictions regarding background services where reported in [21]. In the latter, vendor-specific operating system adjustments like the energy-saving mode had such an impact on the data collection that those data sets had to be excluded from the data analysis. Regarding the complexity of modern smartphone OS, app developers must know about these issues to ensure a high degree of *software product quality*. Surprisingly, according to [5], there is a certain amount of unawareness in the open source developer community about operating system internal mechanisms that affect the execution of background services. Since background services are crucial in the light of mHealth apps, this topic should be explored further.

In order to monitor notifications on Android devices, Weber et al. presented an approach [20] to record the entire notification history of a device. However, unlike this paper, notifications must be uploaded manually. Another shortcoming is that there is no option to exclude apps. Consequently, sensitive data (e.g., instant messaging content like texts or images) might be included in the data set as well.

3. Large-scale Application Monitoring Platform

In order to explore the impact of operating system constraints due to background service restrictions, a framework was developed to study OS events (e.g., notifications) on mobile devices. This framework shall be used to (1) gain more insights into the *software product quality* of apps in general (with a focus on the investigation of mHealth apps) and (2) serve as platform for notification-related studies. In order to investigate app behavior in the wild, the proposed framework should address the following requirements based on the challenges described in Section 2:

Filtered Notification Listener (RQ1): The framework should be implemented with data minimization and privacy in mind. Sensitive data must not be collected unless this is part of the study.

Additional (Sensor) Information (RQ2): It should be possible to optionally collect (sensor) data, such as location, movement, user settings, and capabilities of the device.

Study Configuration & Agreement (RQ3): Data should not be automatically collected before entering a study. The latter should be configurable. Furthermore, there must be an agreement between the user and the study organizer about the collected data. Both the agreement and the collected data should be visible to the user.

Low Energy Consumption (RQ4): Since the app may be whitelisted to avoid being affected by the OS's energy-saving measures, the app should implement energy-saving functions while maintaining a high quality of services.

Automated Reporting (RQ5): Study organizers should always have insights into the status of their studies. In addition, they should be able to create individual dashboards and data reports.

Scalability (RQ6): To address the plethora of different smartphone configurations, the entire framework and all included components should be able to scale to allow different types of studies at the same level of quality.

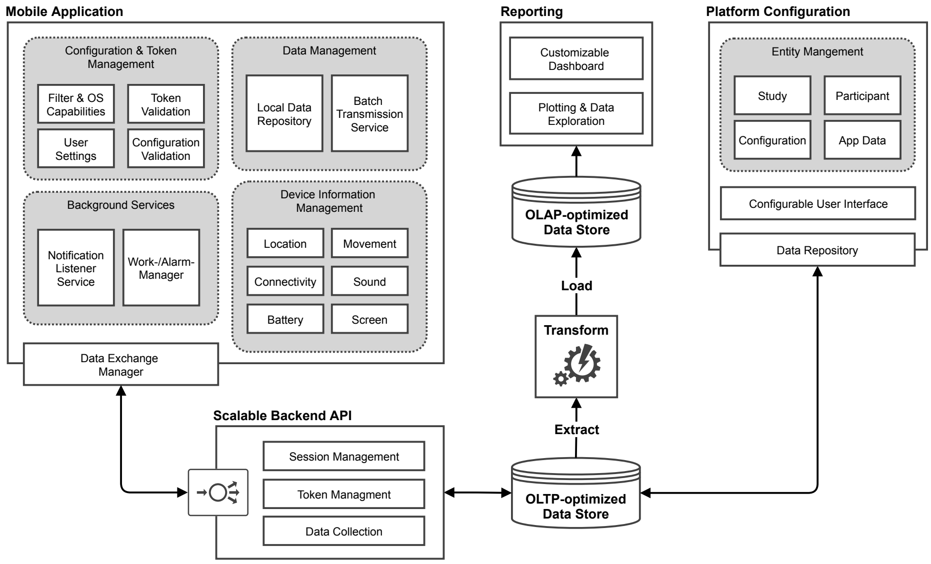

Fig. 1. Overview of the *LAMP* framework components.

3.1. Framework

The proposed framework, which is named *LAMP*, is depicted in Fig. 1. All components are directly or indirectly connected (either uni- or bidirectionally) to a central database service, following the single source of truth (SSOT) principle. Since the centralization of the database generates high workload, this component must be OLTP-optimized (*RQ6*). To fulfil *RQ1*, the app must be able to process the study configuration, validate the latter, and compare it with the capabilities of the device (see Fig. 1, ”Configuration & Token Management”). Furthermore, having *RQ4* in mind, all background services are implemented as workers. In case of older OS versions, other scheduling methods are utilized. Choosing batch transmission whenever possible as data exchange method further decreases network traffic (*RQ4*). Additional (sensor) information (*RQ2*) are implemented as a separate module (see Fig. 1, ”Device Information Management”) depending on the devices capabilities.

To create a configurable study environment, data collections can be configured dynamically. Users can participate in a study via a generated token and must subsequently agree to the study configuration (*RQ3*). To respond to changing configurations, the app should periodically check for changes. Once a change is detected, the updated study configuration must be confirmed again.

Automatic report generation has to be implemented using a data warehouse platform (*RQ5*), which allows for a sophisticated data analysis. Having CPU-intensive, analytical queries in mind, an ETL[1] worker is to be executed periodically, which transfers the data into a separate OLAP-optimized data store.

3.2. Prototype

In this section, the prototypical implementation of the framework is presented. This proof-of-concept prototype implements an Android application to enlighten the behavior of apps regarding the notification service as well as constitute a platform for app usage studies. An overview of the implementation is depicted in Fig. 2.

The document-oriented NoSQL database MongoDB was used as OLTP-optimized data store. To take advantage of the JSON-based document model, NodeJS was used to develop a scalable backend API. The NodeJS event loop enables a high number of requests by processing the latter asynchronously and is therefore suitable for processing the app data.

[1] ETL stands for ”Extract”, ”Transform” and ”Load”

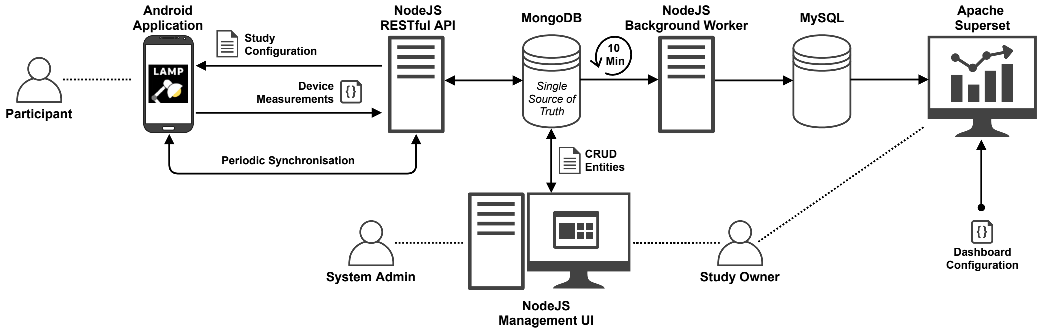

Fig. 2. The *LAMP* platform at a glance.

The Android app was developed using Java. Users can join studies (called ”surveys” in the UI) by entering a token or scan a QR code containing the token. The latter is a UUID[2], automatically generated when creating the participant for a specific study. In the app settings, users can enable or disable location- and movement-based information as well as WiFi information. Consequently, users cannot join studies that require this information. In the notifications tab, a list of recorded notifications is displayed (see Fig. 3). To view detailed information about a notification record, the user must tap on the corresponding list entry.

The management UI, also implemented in NodeJS, allows the configuration of studies, participants, and users in general. In addition, the collected measurements can be displayed. The latter is loaded into the reporting tool in ten-minute intervals. Any updates due to dropouts are also matched. The pre-configured dashboard for each study is depicted in Fig. 4. It consists of seven widgets that give real-time insights into a study.

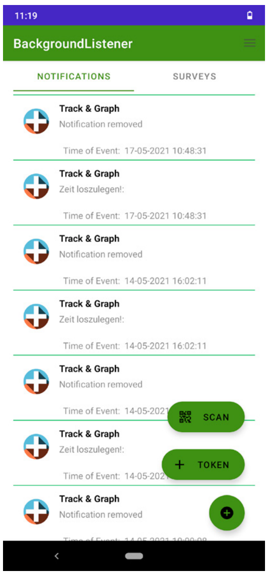

Fig. 3. App: notifications tab

Fig. 4. Pre-configured dashboard (map view was censored due to privacy)

[2] UUID stands for ”Universally Unique Identifier”

4. Evaluation

The proof-of-concept prototype is still in the development phase. Since the resulting framework will be used for scientific measurements, an evaluation of the approach is mandatory. Through a practical evaluation setting, the functionality of the implementation regarding the requirements and the conceptional considerations of the framework is reviewed. The following evaluation is not a complete evaluation of the entire system, but is to be seen as a test run on a small scale. In this evaluation, the reliability of the worker-based implementation of the app background services and the batch transmission of measurements were checked in addition to the general functionality.

For the evaluation, the Nokia 7.2 smartphone with stock Android (i.e., *Android One* in version 10, SDK 29) was used. The app ”Track & Graph”[3] was used to develop a comparable schedule to an longitudinal data collection app (e.g., EMA-based MCS app). ”Track & Graph” is an application to collect longitudinal personal data and to create charts based on the latter. On the backend side, all applications were deployed in a container environment, with 4 CPU cores and 16 GB of RAM. There were to be 3 notifications sent out daily for 5 days at 10am, 12pm, and 2pm (with the exception of May 13, 2021, as this was a public holiday in Germany). Every day at 4pm, the device was powered off and powered on the next day at 8am. The smartphone was used between notification events only to answer questionnaires. Internet access was alternately turned on and off. To perform the measurement under more realistic conditions, energy-saving operating system modes (see Section 2) were activated by removing the power supply and longer periods of no smartphone activity. The measured data points are shown in Table 1.

The salient aspect to evaluate the above framework requirements is the difference between the notification event on the smartphone and the system time in the backend after processing the notification records (*insertion delay*).

[3] GitHub Repository: https://github.com/SamAmco/track-and-graph

Table 1. Data of the practical evaluation setting.

ID	Event Type	System Time (Phone)	UTC Offset	System Time (Backend)*	SDK**	Bat. Status	Bat. Level	Connection
1	posted	2021-05-10 10:00:44	7200000	2021-05-10 12:13:30	29	discharging	70	wifi
2	clicked	2021-05-10 10:13:32	7200000	2021-05-10 14:09:40	29	discharging	70	wifi
3	posted	2021-05-10 12:09:49	7200000	2021-05-11 12:08:07	29	discharging	69	none
4	clicked	2021-05-10 12:09:55	7200000	2021-05-11 12:08:07	29	discharging	69	none
5	posted	2021-05-10 14:04:11	7200000	2021-05-11 12:08:07	29	discharging	68	none
6	clicked	2021-05-10 14:04:15	7200000	2021-05-11 12:08:07	29	discharging	68	none
7	posted	2021-05-11 10:04:20	7200000	2021-05-11 12:08:07	29	discharging	68	none
8	clicked	2021-05-11 10:04:26	7200000	2021-05-11 12:08:07	29	discharging	68	none
9	posted	2021-05-11 12:24:45	7200000	2021-05-11 17:44:04	29	discharging	67	none
10	clicked	2021-05-11 12:24:48	7200000	2021-05-11 17:44:04	29	discharging	67	none
11	posted	2021-05-11 15:43:47	7200000	2021-05-11 17:44:04	29	discharging	66	none
12	clicked	2021-05-11 15:44:00	7200000	2021-05-11 17:44:04	29	discharging	66	none
13	posted	2021-05-12 08:00:12	7200000	2021-05-12 10:18:03	29	discharging	63	wifi
14	clicked	2021-05-12 08:18:05	7200000	2021-05-13 11:28:54	29	discharging	63	wifi
15	posted	2021-05-12 11:07:38	7200000	2021-05-13 11:28:54	29	discharging	63	none
16	clicked	2021-05-12 11:07:40	7200000	2021-05-13 11:28:54	29	discharging	63	none
17	posted	2021-05-12 12:00:41	7200000	2021-05-13 11:28:54	29	discharging	63	none
18	clicked	2021-05-12 12:55:30	7200000	2021-05-13 11:28:54	29	discharging	63	none
19	posted	2021-05-12 14:00:40	7200000	2021-05-13 11:28:54	29	discharging	62	none
20	clicked	2021-05-12 14:00:49	7200000	2021-05-13 11:28:54	29	discharging	62	none
21	posted	2021-05-14 08:09:09	7200000	2021-05-14 10:28:53	29	discharging	57	wifi
22	clicked	2021-05-14 08:09:14	7200000	2021-05-14 10:28:53	29	discharging	57	wifi
23	posted	2021-05-14 10:02:11	7200000	2021-05-14 13:54:34	29	discharging	56	wifi
24	clicked	2021-05-14 12:14:28	7200000	2021-05-14 14:14:33	29	discharging	55	wifi

* Time at which the data record is processed by the backend. The UTC offset of the smartphone (i.e., time zone) is included.
** Refers to the software development kit version of the Android app.

Table 2. Descriptive statistics on the evaluation data (in hours).

Connection	Mean	Median	First Quartile	Third Quartile	Standard Deviation	Range	Minimum	Maximum	Count
All	10,278	3,322	0,276	20,785	10,509	25,179	0,001	25,18	24
WiFi	3,77	0,328	0,276	1,889	8,685	25,179	0,001	25,18	8
None	13,532	19,768	2,507	21,595	10,013	22,353	0,001	22,354	16

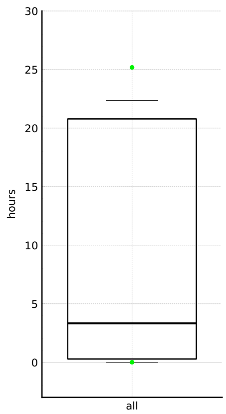

Fig. 5. Insertion delay: entire data set

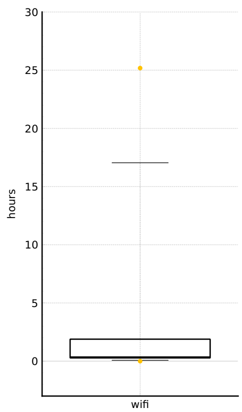

Fig. 6. Insertion delay: WiFi-enabled

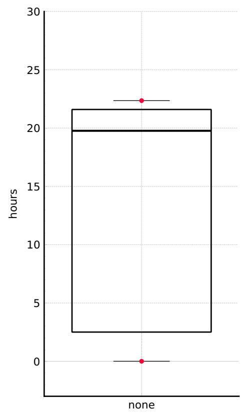

Fig. 7. Insertion delay: offline

The calculated *insertion delays* for the evaluation data, grouped in connection types, are illustrated in the box plots in Fig. 5-7. Descriptive statistics on the data sets are given in Table 2. While the standard deviation is relatively comparable in the data sets, both median and the third quartiles show larger deviations. These can be explained by both the alternating WiFi and the phases in the off state (4pm to 8am). With regard to the entire data set, both the overall system and the aforementioned energy-saving feature appear to be working.

We are currently preparing an evaluation to measure the remaining requirements such as system performance. Multiple devices will be used for this study to provide more meaningful results. Note that the performance aspect was not part of evaluation at hand.

5. Summary & Outlook

In this paper, we described the problem of lacking *software product quality* of mHealth apps due to OS-related restrictions introduced in recent OS versions (e.g., *Doze Mode* or *App Standby* in Android). In addition, we presented the need for reliable background services and their impact on data collection. We furthermore have elaborated the previously underestimated dimension of *software product quality* of mHealth apps that should be researched more intensively in the future. To tackle this issue, we presented requirements for a platform that is able to investigate the impact of background service limitations based on notification events. We created the *LAMP* framework that meets the analyzed requirements, while putting also a special emphasis on data and energy conservation. An implementation as proof-of-concept prototype, which serves as a research platform for further research was presented as well. *LAMP* was subsequently tested in a practical evaluation. In future work, *LAMP* will be evaluated in more detail. Currently, only an Android version of *LAMP* is available. An iOS version is in development in order to be able to perform data collection on both operating systems.

References

[1] American Psychiatric Association, 2017. App Evaluation Model. https://www.psychiatry.org/psychiatrists/practice/mental-health-apps/app-evaluation-model. [Online; accessed 05-March-2020].

[2] Baumel, Amit and others, 2017. Enlight: A Comprehensive Quality and Therapeutic Potential Evaluation Tool for Mobile and Web-Based eHealth Interventions. J Med Internet Res 19, e82. URL: http://www.jmir.org/2017/3/e82/, doi:10.2196/jmir.7270.

[3] Beierle, F., Schobel, J., Vogel, C., Allgaier, J., Mulansky, L., Haug, F., Haug, J., Schlee, W., Holfelder, M., Stach, M., Schickler, M., Baumeister, H., Cohrdes, C., Deckert, J., Deserno, L., Edler, J.S., Eichner, F.A., Greger, H., Hein, G., Heuschmann, P., John, D., Kestler, H.A., Krefting, D., Langguth, B., Meybohm, P., Probst, T., Reichert, M., Romanos, M., Störk, S., Terhorst, Y., Weiß, M., Pryss, R., 2021. Corona Health—A Study- and Sensor-Based Mobile App Platform Exploring Aspects of the COVID-19 Pandemic. International Journal of Environmental Research and Public Health 18. URL: https://www.mdpi.com/1660-4601/18/14/7395, doi:10.3390/ijerph18147395.

[4] Beierle, F., et al., 2019. What data are smartphone users willing to share with researchers? Journal of Ambient Intelligence and Humanized Computing , 1–13.

[5] C, M., Chimalakonda, S., K, C., 2021. How do open source app developers perceive API changes related to Android battery optimization? An empirical study. Software: Practice and Experience 51, 691–710. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.2928, doi:https://doi.org/10.1002/spe.2928.

[6] IQVIA, 2017. IQVIA Institute for Human Data Science Study: Impact of Digital Health Grows as Innovation, Evidence and Adoption of Mobile Health Apps Accelerate - IQVIA. https://www.iqvia.com/newsroom/2017/11/impact-of-digital-health-grows-as-innovation-evidence-and-adoption-of-mobile-health-apps-accelerate/. [Online; accessed 05-March-2020].

[7] ISO/IEC 25010, 2011. ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Technical Report. 00000.

[8] Jain, S.H., Powers, B.W., Hawkins, J.B., Brownstein, J.S., 2015. The digital phenotype. Nature biotechnology 33, 462–463.

[9] Kraft, R., Schlee, W., Stach, M., Reichert, M., Langguth, B., Baumeister, H., Probst, T., Hannemann, R., Pryss, R., 2020a. Combining Mobile Crowdsensing and Ecological Momentary Assessments in the Healthcare Domain 14, 164. URL: https://www.frontiersin.org/article/10.3389/fnins.2020.00164, doi:10.3389/fnins.2020.00164.

[10] Kraft, R., Stach, M., Reichert, M., Schlee, W., Probst, T., Langguth, B., Schickler, M., Baumeister, H., Pryss, R., 2020b. Comprehensive insights into the TrackYourTinnitus database, in: The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), Elsevier, Leuven, Belgium. pp. 28 – 35. URL: https://www.sciencedirect.com/science/article/pii/S1877050920316872, doi:https://doi.org/10.1016/j.procs.2020.07.008.

[11] Mathews, S.C., et al., 2019. Digital health: a path to validation. NPJ digital medicine 2, 1–9.

[12] Messner, E.M., Terhorst, Y., Barke, A., Baumeister, H., Stoyanov, S., Hides, L., Kavanagh, D., Pryss, R., Sander, L., Probst, T., 2020. The German Version of the Mobile App Rating Scale (MARS-G): Development and Validation Study. JMIR Mhealth Uhealth 8, e14479. URL: http://mhealth.jmir.org/2020/3/e14479/, doi:10.2196/14479.

[13] Petter, O., Hirsch, M., Mushtaq, E., Hevesi, P., Lukowicz, P., 2019. Crowdsensing under Recent Mobile Platform Background Service Restrictions: A Practical Approach, in: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, Association for Computing Machinery, New York, NY, USA. p. 793–797. URL: https://doi.org/10.1145/3341162.3344867, doi:10.1145/3341162.3344867.

[14] Rodríguez, M., Oviedo, J.R., Piattini, M., 2016. Evaluation of Software Product Functional Suitability: A Case Study. Software Quality Professional 18. 00022.

[15] Smyth, J.M., Heron, K.E., 2014. Ecological Momentary Assessment (EMA) in Family Research, Springer International Publishing, Cham, pp. 145–161. URL: https://doi.org/10.1007/978-3-319-01562-0_9, doi:10.1007/978-3-319-01562-0_9.

[16] Stach, M., Kraft, R., Probst, T., Messner, E.M., Terhorst, Y., Baumeister, H., Schickler, M., Reichert, M., Sander, L.B., Pryss, R., 2020a. Mobile Health App Database - A Repository for Quality Ratings of mHealth Apps, in: 33rd IEEE CBMS International Symposium on Computer-Based Medical Systems (CBMS), IEEE, Rochester, MN, USA. pp. 427–432. URL: https://ieeexplore.ieee.org/abstract/document/9183243/, doi:10.1109/CBMS49503.2020.00087.

[17] Stach, M., Vogel, C., Gablonski, T.C., Andreas, S., Probst, T., Reichert, M., Schickler, M., Pryss, R., 2020b. Technical Challenges of a Mobile Application Supporting Intersession Processes in Psychotherapy, in: The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC), Elsevier, Leuven, Belgium. pp. 261 – 268. URL: https://www.sciencedirect.com/science/article/pii/S187705092031718X, doi:https://doi.org/10.1016/j.procs.2020.07.038.

[18] Stoyanov, S., et al., 2015. Mobile App Rating Scale: A New Tool for Assessing the Quality of Health Mobile Apps. JMIR mHealth and uHealth 3, e27.

[19] Torous, J.B., et al., 2018. A Hierarchical Framework for Evaluation and Informed Decision Making Regarding Smartphone Apps for Clinical Care. Psychiatric Services 69, 498–500. URL: https://doi.org/10.1176/appi.ps.201700423, doi:10.1176/appi.ps.201700423. pMID: 29446337.

[20] Weber, D., Voit, A., Henze, N., 2018. Notification Log: An Open-Source Framework for Notification Research on Mobile Devices, in: Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, Association for Computing Machinery, New York, NY, USA. pp. 1271–1278. URL: https://doi.org/10.1145/3267305.3274118, doi:10.1145/3267305.3274118.

[21] Weber, D., Voit, A., Henze, N., 2019. Clear All: A Large-Scale Observational Study on Mobile Notification Drawers, in: Proceedings of Mensch Und Computer 2019, Association for Computing Machinery, New York, NY, USA. pp. 361–372. URL: https://doi.org/10.1145/3340764.3340765, doi:10.1145/3340764.3340765.

[22] Wisniewski, H., Liu, G., Henson, P., Vaidyam, A., Hajratalli, N.K., Onnela, J.P., Torous, J., 2019. Understanding the quality, effectiveness and attributes of top-rated smartphone health apps. Evidence-based mental health 22, 4–9.