End Users' Perspective of Performance Issues in Google Play Store Reviews

Anam Noor[0000−0001−7854−5025]1, Muhammad Daniyal Mehmood[0000−0002−2667−5060]1, and Teerath Das[0000−0003−2024−6545]2

[1] Department of Computer Science, Mohammad Ali Jinnah University, Karachi Pakistan
fa19mscs0018@maju.edu.pk, sp21msse0009@maju.edu.pk
[2] Faculty of Infromation Technology, University of Jyväskylä, Jyväskylä, Finland
teerath.t.das@jyu.fi

Abstract. The success of mobile applications is closely tied to their performance which shapes the user experience and satisfaction. Most users often delete mobile apps from their devices due to poor performance indicating a mobile app's failure in the competitive market. This paper performs a quantitative and qualitative analysis and investigates performance-related issues in Google Play Store reviews. This study has been conducted on 368,704 reviews emphasizing more 1- and 2-star reviews distributed over 55 Android apps. Our research also reports a taxonomy of 8 distinct performance issues obtained using manual inspection. Our findings show that end-users recurrently raised *Updation* (69.11%), *Responsiveness* (25.11%), and *Network* (3.28%) issues among others. These results can be used as preliminary steps towards understanding the key performance concerns from the perspective of end users. Furthermore, Our long-term objective will be to investigate whether developers resolve these performance issues in their apps.

Keywords: Android mobile apps, Google Play reviews, Performance related issues

1 Introduction

Technological breakthroughs have largely influenced modern society, particularly in the field of mobile apps. It is expected that the App Economy will rise to a new peak from $693 Billion in 2021 to $935.2 Billion in 2023[3]. The performance aspect of the apps is a significant indicator of their successful growth. Developers implement new resource-intensive features in the apps to meet the end users' requirements showcasing efficient performance. Thus, performance is a crucial parameter for determining the success or failure of any mobile app, as the user experience highly depends on it: the more flawless performance, the better the user-acceptance ratio.

[3] https://www.statista.com/statistics/269025/worldwide-mobile-app-revenue-forecast/

To date, performance issues have been analyzed in various systems like web applications [1], heterogeneous environments [3], and large scale applications [6]. Further, Liu et al. [4] have explored 70 performance smells in mobile apps and classified them into three broad categories. To the best of our knowledge, one of the studies [2] closely resembles our study. However, the significant difference between the two is that the former [4] analyzed the performance issues in GitHub commits, and the latter explored the reviews of android apps available on the Google Play Store. Our approach is different and novel because we are analyzing performance related issues from end users perspective i.e., exploiting Google reviews of Android apps. Our results shows how the performance reviews vary across different android app categories. Furthermore, we produced a taxonomy of performance issues after manual inspection of 1 and 2-star reviews.

The main **contributions** of this paper are:

- Analysis of performance issues in Google Play Store reviews of 55 Android apps.
- A taxonomy of most common types of performance issue in Android apps.

2 Study Design

The *goal* of this study is to analyze the performance in rich Google reviews, with the *purpose* to comprehend their connectivity with the end-users and attributes of the projects. The *context* of our study is 55 android apps from Google Play Store and examined their 1 and 2- star reviews from the *point of view* of the end users. This research aims to address the following research questions:

RQ1: *To what extent do end users perceive the performance-related issues in Android app reviews?*

RQ2: *What are the performance issues that end users raise in Android apps reviews?*

RQ1 focuses on estimating the prevalence in which the end users consider performance issues in android apps, whereas *RQ2* dedicates to classifying Google reviews with respect to the performance issues as raised by the end users. The design of our study mainly comprises a set of mobile apps distributed across the Google Play Store. The purpose of selecting Google Play Store as our target area of the population lies in its increasingly huge popularity among all the other marketplaces for apps. Figure 1 represents the step-by-step process adopted to get our targeted apps; (i) firstly, we selected the most popular apps from the first quarter of 2021 using different sources[4] (S), which resulted in identifying 55 android apps, (ii) then, we extracted all the reviews of selected apps using a Python script and collected a total of 355,687,535 reviews, (iii) after that, we filtered 368,707 reviews based on their 1 and 2-star reviews. The reason for selecting these reviews is that the end users express their dissatisfaction by giving 1- and 2-star reviews. We inscribed a dedicated script to extract 1- and

[4] https://github.com/anam-noor/Replication-package-

2-star Google reviews from the selected apps. Thus, the final population for this research is 368,707, which is spread over 55 different apps.

The designed variables for addressing RQ1 include (i) *pReviews*: the performance-related reviews out of the total number of reviews (*tReviews*) of android apps, and (ii) android apps categories in the Google play store. For RQ2, the reviews are categorized into different categories addressing performance issues. We identified a review as a performance-related review (*pReviews*) if it contains one of the following keywords: *update, wait, slow, lag, response, time, speed, graphic, perform, hang, memory, leak, connect, image*, not. These keywords were considered by looking, evaluating, and combining mining methodologies from past empirical studies [7][8][2][5] on software performance (both mobile and not mobile-specific). The script ensures all the possible combinations of the upper and lower case keywords.

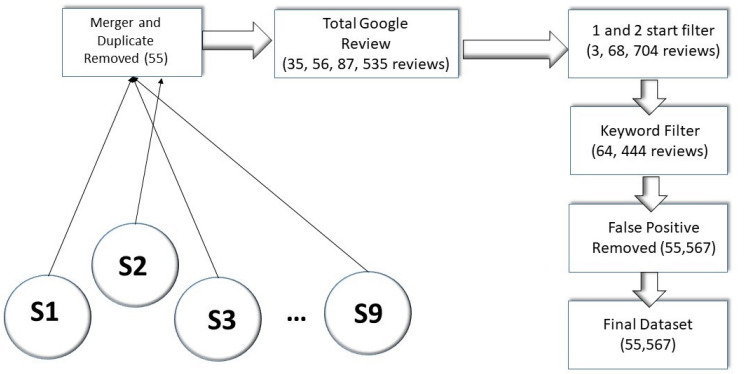

Fig. 1. Google reviews extraction

The matching of the keywords resulted in detecting a set of 64,444 performance-related reviews of 1- and 2-star ratings. After a manual analysis, 8,877 reviews are discarded for being categorized as False Positive. The whole process finally produced a total of 55,567 performance-related reviews.

Furthermore, we investigated the categories of performance concerns by manually labeling the 55,567 reviews. We used manual labeling to categorize performance-related reviews into relevant groups in two phases: the first phase is dedicated to tagging each review with its relevant keywords (e.g., graphics from GUI, slow, and hang). Subsequently, in the second phase, we labeled those tags into more significant groups with an explanatory title (e.g., Memory management issues, Networking issues). The second phase of labeling resulted in 8 different types of categories. The manual inspection of both the phases are conducted by two masters students separately and then supervisor cross check the labels.

3 Results

In this section, we will discuss the results by addressing all research questions of our study, as mentioned in the previous section.

RQ1: *To what extent do end users perceive the performance-related issues in Android app reviews?*

To answer this research question, we primarily compute the ratio of identified number of performance-related reviews (*pReviews*) to the total number of reviews (*tReviews*) present in our dataset. It is interesting to note that all 55 apps in our dataset have at least one performance-related review. A total of 55,567 (15%) performance issues have been identified out of 368,704 reviews.

Table 1 reports the apps categories, the frequency of each app category, total number of reviews (*tReviews*), and performance related reviews (*pReviews*) identified. Performance issues vary across different natures of the applications. For example, the *Entertainment* app category holds the highest percentage of performance-related reviews (22.23%) as shown in Table 1. This is understandable because such apps have a captivating user experience, and users tend to spend more time on these apps due to online sessions. The *Game* app category holds the second highest performance reviews (17.23%) followed by *Shopping* (15.53%) category. The *Game* category mainly relies on the user experience of long sessions, whereas the shopping category is a utility that included in task-based apps with short usage sessions. This shows that performance issues are orthogonal across each category and distributed without the application context.

Table 1. Distribution of performance related issues over various app categories.

App Category	*App Frequency*	*tReviews*	*pReviews*
Entertainment	4	28,558	6350 (22.23%)
Games	11	116,691	20161 (17.27%)
Shopping	9	80219	12294 (15.3%)
Education	6	31054	4310 (13.87%)
Tools	8	34391	3459 (10.3%)
Lifestyle	6	1918	190 (9.90%)
Books and references	1	684	56 (8.18%)
Food and drink	3	14635	886 (6.05%)
Music and audio	8	60538	2653 (4.35%)

RQ2: *What are the performance issues that end users raise in Android apps reviews?*

To answer this research question, we manually analyzed the 55,567 reviews, which resulted in 8 different performance categories. Table 2 depicts eight categories extracted from the manual inspection along with an example of representative review (the representative review is randomly selected from our observed dataset) and *pReviews* (in terms of frequency percentage) of performance-related

reviews for each category. In the following, we will consider each of the listed categories of performance issues obtained from manual analysis.

As mentioned in Table 2, the most frequent concern of end-users while using android apps is the *updation* (69.11%), e.g., "In terms of chat during the games, lags due to update." From the manual inspection of reviews, we derive that the end user's perspective of app performance is of paramount importance for app developers. Some examples of *updation* issues after manual inspection are *"Crashing of apps after update," "App stops working after update" and "Features unavailability"*. Therefore, Performance enhancements constitute a significant part of app updates.

The second most common performance-related complaint expressed by end users' is *Responsiveness* (25.6%), which is determined by evaluating Speed, Lag, and Delay keywords. *"Slow apps", "Waiting a long time for an app to load," "App lagging, app stuck," and "App hang"* are all examples of Responsiveness issues. The *Network* (3.28%) related performance issues are also commonly documented by end-users in their reviews, eg., "The apps seems useless if it does not connect to the server", "Disconnecting from the server" and "Network sharing, and several connection issues". Despite being reported in lesser frequency, *Memory Management* issue ((0.36%)) is still one of the most critical as it can paralyze the app by halting key processes. Examples of such issues are *"Using three times of memory than other apps" and "Immediately ate up over 17MB of memory without ever loading"*.

Loading time of any app plays an important role in the overall app performance for the end user. Loading time makes up for 0.14% of *pReviews*. Some examples are "It takes ages to start the screen" and "Never ending loading time". Moreover, about 0.85% of the *pReviews* are *generic* and do not point to a specific issue category. In these reviews, users do not describe any specific category; instead, they talk in general without explaining the reason. Some examples are *"Improve the app" and "Good game but with performance issues"*.

4 Threats To Validity

The use of keyword matching to detect performance-related reviews poses a threat to *construct Validity*. In our study, we assume that a review containing specified keywords should be considered a performance-related review. This approach may omit a few performance-related reviews because we may miss a few keywords. In order to mitigate this threat, we consider all the keywords in the previous studies [7][8][2][5]. False positives have been reduced by manual inspection of extracted reviews.

External Validity threats primarily affect the generalization of our findings, which are related to the representation of different app categories studied in this study. We were able to mitigate this risk by using a relatively large data set (i.e., 368,704 reviews consisting of 55 apps) and picking apps belonging to diverse app categories and built as part of real-world apps (i.e., all apps are released on the Google Play store and are publicly available)

Table 2. Identified categories of performance-related reviews

Performance Category	*Representative Review*	*pReviews*
Updation	*This is horrible, the update ruined everything, when you are done finding what you want to say the meeting is over*	38,404 (69.11%)
Responsiveness	*Hangs at startup logo. Unusable*	14,247 (25.6%)
Network	*Slow server... Always lost connection when almost win*	1,826 (3.28%)
Generic concern	*As game is growing it is becoming very dull performance*	475 (0.85%)
GUI	*Outdated user interface, not quite intuitive and audio*	319 (0.57%)
Memory management	*Don't you guys have any good devs? Your app is full of memory leaks!*	205 (0.36%)
Loading time	*I either receive an error message or it's stuck on a loading screen*	82 (0.14%)
Image	*Image loading is awful on the app. its really frustrating to have to wait over 20 seconds for images to load*	6 (0.010%)

5 Related Work

Various studies have been done which analyze performance issues in android apps for example Das et. al. [2] takes similar approach, but in the context of performance related commits of the android apps having versioning history hosted on GitHub repositories whereas our study focuses on performance issues as per end user reviews. Another study by Liu et.al [4] investigates performance bugs and categorizes them into categories. They identify 70 performance related bugs and characterize these bugs into 3 categories. In our study we not only consider more performance issue categories in our taxonomy but also relate app categories to these issues forming a pattern. Malavolta et. al. [5]conducts a similar study but into hybrid mobile apps, by mining free apps and reviews from the Google Play Store.

6 Conclusion And Future Work

This paper reports the results of a study by analyzing performance related reviews in Android apps. We investigated a total of 55,567 Google Play Store reviews which were rated 1 and 2-stars out of 368,704 distributed over 55 apps. We proposed a taxonomy for such reviews using manual inspection and identified a total of 8 performance related issue categories. The main findings of our study show performance issues are mostly found in the app due to *Updation*. In addition to that, we also observed numerous reviews for the *responsiveness* of apps and *Network* related issues. This study will help developers understand different performance bottlenecks in their apps from end users perspective.

Future work aims to exploit these labelled performance issue categories of this study to automatically classify using different machine learning algorithms. It is also interesting to analyze the performance-related reviews using natural language processing techniques. Future work also aims to analyze other aspects of non-functional issues in the google reviews.

References

1. Tarek M. Ahmed, Cor-Paul Bezemer, Tse-Hsun Chen, Ahmed E. Hassan, and Weiyi Shang. Studying the effectiveness of application performance management (APM) tools for detecting performance regressions for web applications: an experience report. In Miryung Kim, Romain Robbes, and Christian Bird, editors, *Proceedings of the 13th International Conference on Mining Software Repositories, MSR 2016, Austin, TX, USA, May 14-22, 2016*, pages 1–12. ACM, 2016.
2. Teerath Das, Massimiliano Di Penta, and Ivano Malavolta. A quantitative and qualitative investigation of performance-related commits in android apps. In *2016 IEEE International Conference on Software Maintenance and Evolution, ICSME 2016, Raleigh, NC, USA, October 2-7, 2016*, pages 443–447. IEEE Computer Society, 2016.
3. King Chun Foo, Zhen Ming Jiang, Bram Adams, Ahmed E. Hassan, Ying Zou, and Parminder Flora. An industrial case study on the automated detection of performance regressions in heterogeneous environments. In Antonia Bertolino, Gerardo Canfora, and Sebastian G. Elbaum, editors, *37th IEEE/ACM International Conference on Software Engineering, ICSE 2015, Florence, Italy, May 16-24, 2015, Volume 2*, pages 159–168. IEEE Computer Society, 2015.
4. Yepang Liu, Chang Xu, and Shing-Chi Cheung. Characterizing and detecting performance bugs for smartphone applications. In Pankaj Jalote, Lionel C. Briand, and André van der Hoek, editors, *36th International Conference on Software Engineering, ICSE '14, Hyderabad, India - May 31 - June 07, 2014*, pages 1013–1024. ACM, 2014.
5. Ivano Malavolta, Stefano Ruberto, Tommaso Soru, and Valerio Terragni. End users' perception of hybrid mobile apps in the google play store. In Onur Altintas and Jia Zhang, editors, *2015 IEEE International Conference on Mobile Services, MS 2015, New York City, NY, USA, June 27 - July 2, 2015*, pages 25–32. IEEE Computer Society, 2015.
6. Haroon Malik, Hadi Hemmati, and Ahmed E. Hassan. Automatic detection of performance deviations in the load testing of large scale systems. In David Notkin, Betty H. C. Cheng, and Klaus Pohl, editors, *35th International Conference on Software Engineering, ICSE '13, San Francisco, CA, USA, May 18-26, 2013*, pages 1012–1021. IEEE Computer Society, 2013.
7. Marija Selakovic and Michael Pradel. Performance issues and optimizations in javascript: an empirical study. In Laura K. Dillon, Willem Visser, and Laurie A. Williams, editors, *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, May 14-22, 2016*, pages 61–72. ACM, 2016.
8. Shahed Zaman, Bram Adams, and Ahmed E. Hassan. A qualitative study on performance bugs. In Michele Lanza, Massimiliano Di Penta, and Tao Xie, editors, *9th IEEE Working Conference of Mining Software Repositories, MSR 2012, June 2-3, 2012, Zurich, Switzerland*, pages 199–208. IEEE Computer Society, 2012.