Testing Challenges for Mobile Applications: An evaluation and comparative analysis of different testing approaches

Ali Azgar[1], Sohel Rana[2], Saddam Hossain[3], Most. Jannatul Ferdous[4]
[1,4]*Dept. of CSE, Rajshahi University of Engineering and Technology, Bangladesh*
[2]*Dept. of CSE, Dhaka University of Engineering and Technology, Bangladesh*
[3]*Dept. of CSE, Jahangirnagar University, Bangladesh*

***Abstract*: A mobile application, sometimes known as an app, is a form of software designed to operate on a mobile device such as a mobile phone or tablet and is commonly used to deliver user services similar to those accessible on PCs. Mobile applications have become an important part of our lives, allowing us to communicate with people all over the world while sitting at home. However, these applications do more than just connect people; they also provide information, entertainment, and a way to learn new things at any time and from any location. Mobile application testing is very important before releasing any apps. However, since multimode mobile platforms such as handsets, smartphones, tablets, pads, and wearable electronics are now available, various device fragmentations, iOS platforms, and various customer specifications, mobile application testing has recently become incredibly challenging. This article provides an overview of several mobile testing approaches as well as the most recent mobile app testing challenges. In addition, we provide a testing cycle in which mobile apps are sent to clients for feedback. In this article, we also performed a survey of current mobile app users on various bugs in real-world mobile applications, which will be used as a factor in future development.**

***Keywords:* Mobile, Application, App, testing, Smart devices, challenges, Customer, Bugs, Smartphones, App user.**

I. INTRODUCTION

Smartphones have developed from basic communication devices to multi-functional devices. Ordering meals, hailing a cab, or even checking for directions may all be accomplished through a smartphone. It is only feasible since to the wonderful world of smartphone apps. Mobile apps (apps that run on mobile phones or next-generation devices, for example) have become so ubiquitous that they are generating a revolution in the IT industry [1]. The technique of testing software applications for handheld mobile devices for functionality, usability, and consistency is known as mobile application testing [2]. Apps for mobile software distribution are either pre-loaded or may be installed platforms. In recent years, the popularity of mobile phones has skyrocketed. According to a poll conducted by the Yankee Group, revenue production will reach $4.2 billion by 2013, with 7 billion smartphone app downloads in the United States.[3]

Individuals and businesses interested in the development of mobile applications might benefit from mobile app testing services. In terms of smartphones, cellular technology has improved rapidly in recent years. It simply opened up opportunities for software development firms to create various types of mobile applications customized to the demands of end customers. You'll require mobile application testing as a mobile application developer or company to guarantee that your finished app is mature enough to meet the expectations of end users. Experienced and qualified staff will evaluate the developed app to guarantee that it is free of functional and usability faults [2]. Early testing is always the best strategy to design a bug-free software. After you've finished designing a mobile app, you should consult a mobile application testing company to evaluate and repair the app for any possible flaws that a user could face. These testing companies will assist in the development of a bug-free application that supports the newest features and assures maximum compatibility [4]. Regardless, comprehensive mobile app testing will yield the highest return on investment if customers give positive feedback on the published playstore [5].

Mobile application testing varies from typical desktop or online application testing. This implies it comes with its own set of difficulties. The diversity of mobile devices poses a major issue. There are several screen sizes, operating systems, hardware configurations, software versions, and so on available today. There are over sixty thousand Android phones on the marketplace as of 2018, and some of them have already implemented the notch design, which comes with its own set of challenges.

A. Motivation

Nowadays, mobile applications are more than just that! They are in the midst of a golden age of adopting more advanced and hybrid applications. In today's technologically advanced world, mobile app testing has become an essential task. Mobile app testing is becoming increasingly popular in the digital environment. Loading troubles or a lack of performance, sluggish components, reacting to resolves and other challenges are common in mobile native and hybrid applications. Prior to going live, mobile application testing

may be able to resolve these vulnerabilities. In today's technologically evolved culture, the mobile phone has become a crucial part of everyone's life. Everyone is using a mobile app or a mobile-friendly website for their business. Furthermore, various new firms have begun up every day in the e-commerce market, with a profusion of applications covering their online areas. Every day, mobile phone users install new apps based on ratings and reviews, which are closely connected to how well your app works. As a result, mobile application testing has become increasingly important.

B. Contributions

The study's key contribution of this is the comparative evaluation of various testing techniques and the required test cases for mobile application testing for end users. However, as the number of mobile applications is quickly expanding, the majority of consumers encounter issues during the installation or use of mobile applications. As a consequence, we identified the most likely bugs and their causes as a matter of concern. Finally, based on the number of mobile users and the number of mobile apps, we assess an investigation for various issues that happen to each user.

II. RELATED WORKS

When usability testing is done in a laboratory environment, experiment control and high-quality data gathering are not a problem. However, one of the disadvantages of this environment is its lack of reality. The author of Paper [6] investigates novel ways for assessing the usage of mobile technology in laboratory environment in order to address this problem.

On the basis of three research questions, the author of Paper [1] analyzes new research directions: (RQ1) How different are mobile applications from traditional apps that they require new and specialized testing techniques? , (RQ2) What are the most recent challenges and research directions in mobile app testing? and (RQ3)What function does automated play in mobile application testing?

The practice of evaluating a mobile app for functionality, usability, and performance concerns is known as app testing. The author of paper [2] offers a basic concept for mobile application testing as well as upcoming issues in the field.

Users nowadays demand their phones to have more features, usefulness, and personalization possibilities. Developing mobile apps is time-consuming, expensive, and challenging since there are hundreds of different mobile devices with distinct operating systems, browsers, screen sizes, and native APIs. Cloud - based services for testing process might help and improve this problem. The Autonomous Smartphone Testing as a Service (AM-TaaS) platform, proposed by the author of paper [7], allows automated testing for mobile apps depending on AQuA's test requirements.

Shaik et al. presented a study on software metrics and their increasing relevance in software development and achieving specific software features in article [20]. The authors of article [17] provide an effective technique that assists people and junior staff in first aid institutions in locating available information resources.

The authors of article [18] use a methodology that focuses on discarding redundant information and then applying log transformation to enhance the performance of software metrics for recognizing faults-prone classes of free software and to see a comparative evaluation of the original dataset's metric values with the metric values after the eliminate redundancy, log conversion, and results must be recorded. Because mobile devices are so complex and difficult and key point, they are susceptible to software reliability and performance issues. A series of eight experiments are carried out in order to distinguish SA (Software Aging) in Android mobiles in the authors of article [19]

The authors of article [21] compared the smartphone operating systems Android and apple Operating System (iOS) that are commercially available, with a focus on numerous difficulties. They also spoke about collecting failure data from the IOS platform automatically. Because to its closed-source nature, collecting failure data is difficult.

The author of paper [22] analyzes the topic of software aging in smartphone Operating system, which leads the device's performance to steadily deteriorate and finally fail. They describe an experimental technique for analyzing software aging concerns in the Android operating system using statistical approaches to predict which factors (such as workloads and device settings) increase performance deterioration and resource usage.

The quality of the produced software is determined by its bug-free functioning [23]. Though problems can be introduced at any stage of the software development life cycle, detecting them early in the process can mean a reduction in the cost of verification and validation resources. Using several research questions, the author of paper [23] constructs a defect prediction model based on software modification metrics.

To grasp and know the software metrics, the author of the paper [24] studied a collection of literature reviews from various digital databases that have been accessible since 2008. Finally, it has been determined that software quality is a method of assessing how software is developed and how well the program complies to that design for the criteria such as Correctness, Product quality, Scalability, Completeness, and Absence of defects.

Software quality is a basic necessity for any user, academic, software development firm, or developer. The author of [25] developed a model for an object-oriented Software Bug Prediction System (SBPS) that can anticipate the existence of

defects in a class if they are discovered during software validation using hypothesis based metrics.

MOBILE APPLICATION TESTING APPROACHES

Manual testing and automated testing are two methods for evaluating mobile applications based on how they are performed.

A. Manual Testing

As the name implies, manual testing is a human-centered approach that prioritizes user experience. The application's performance, security, and usability are investigated and assessed through the perspective of a user in an exploratory method. This ensures that your program complies with user-friendliness standards. Because problems take time to identify, this type of testing takes a long period. As a rule of thumb, 20% of an application's testing should be performed manually using alpha and beta versions, with the rest being automated.

B. Automated Testing

Automated testing is the second method for evaluating mobile applications. During this procedure, an array of test cases is created that should cover around 80% of the testing process. Automated testing provides several advantages over manual testing, including increased testing efficiency, improved regression test execution, time savings, the ability to reuse test scripts, and the ability to execute test scripts in parallel on multiple platforms.

TYPES OF MOBILE APPLICATION TESTING

A. Usability testing

Usability testing is a great way to see how the software makes it easier for users to accomplish their goals. Individual users are given specific, realistic scenarios of app use during this test. When usability testing is based on direct feedback from the end user, it is also reliable. Usability testing also examines if the design is straightforward, with a focus on convenience of use and customer satisfaction.

B. Performance testing

Performance testing evaluates an application's speed, stability, and responsiveness under various workload situations. The fundamental purpose of a performance test is to ensure that an application is correctly aligned with the performance expectations. Performance evaluations include load testing, volume testing, soak testing, spike testing, and stress testing.

C. Security testing

In today's world, security is a major concern for practically every mobile app owner. According to statistics, 80% of customers are more likely to delete an app due to security issues. As a result, it's vital to prioritize mobile app security testing. For different transactions, certain programs, such as travel apps, require users' personal information. If your app demands anything comparable, you must provide guarantees concerning the app's confidentiality, integrity, and authenticity.

D. Interruption testing

Testing an application's performance in a halted state before returning to the prior state is called interruption testing. Interruptions include incoming phone calls or SMS, alarms, push notifications from mobile apps, battery low or full, network connection loss and recovery, and being plugged in or out while charging.

E. Compatibility testing

Compatibility testing is a type of non-functional testing that validates the functioning of a mobile app across a variety of operating systems, apps, devices, internal hardware requirements, and network circumstances. Compatibility testing is used to see if a mobile app is compatible with different operating systems and versions.

F. Functional testing

Mobile applications functional testing guarantees that the app's functionalities meet the required goals. This type of testing concentrates mostly on the mobile app's main aim and flow. When working on Functional testing services, it ensures that the mobile app's features meet the set requirements and is highly responsive.

G. Memory leakage testing

When a computer application program is unable to manage memory, memory leakage occurs that has been allotted to it, resulting in poor application performance and general system slowness. Because mobile devices have severe memory limits, memory leakage testing is critical for an application's correct operation [5].

H. Installation testing

Some mobile apps are pre-installed on the phone, while others must be downloaded from a google play store. Installation testing guarantees that the installation process goes without a problem and that the user has no troubles. This testing procedure includes installing, updating, and uninstalling a program.

III. CHALLENGES FOR THE MOBILE TESTING

For a long time, it has been clear that mobile phones are the current market players, to the point that some analysts predict they will soon substitute PCs and desktops [2]. Because the mobile application is invisible in nature, its performance is critical [8, 9, 10]. Based on data management, performance testing does stress testing. Application performance may deteriorate while operating on low power and network coverage is poor. Response time, stress, low battery, and network unavailability are the main performance testing issues. User happiness is critical for any product or application [8, 11, 12], and this is especially true for mobile apps. The main challenge in this category is maintaining

response time in any circumstances. Mobile applications are available in various domains such as banking and finance, including sensitive data that might lead to data fraud [13,14,15,16]. In terms of mobile application security, • Identifying device-specific vulnerabilities • Identifying OS-specific vulnerabilities • Unstable encryption and hosting control • Unsafe data storage challenges are predominant.

Others Frequently Encountered Challenge Zones:

A. Multitude Mobile Devices

Over 500 million Android devices have been shipped since Android 1.0, whereas over 220 million iOS devices have been released since 2007. With so many mobile devices to choose from, including handsets, smartphones, tablets, pads, and wearable gadgets, your mobile app will be used in a variety of situations.

B. Device Fragmentation & Various OS Platforms

Device fragmentation is possibly the most challenging part of the mobile testing environment. Due to the fact that the iOS device grid is growing at a quicker rate than ever before, Android fragmentation is a major concern.

C. Different Mobile App Types

A mobile app can be a simple app, a web based application, or a hybrid app that combines the two types of information. The testing of each of these app types varies from one another due to the differences in their implementation.

TEST CASE FOR TESTING

The requirements for mobile app testing are heavily influenced by the type of application being tested. Test cases and test scenarios are the base of different mobile applications testing before release of the apps. The performance basically depends on cases and test scenarios. It is very important to set appropriate test cases and create test scenarios. Without test cases, it is hard to track, evaluate, and address quality issues. Scenario testing helps mobile app developers understand how mobile apps are used in the real world, ensuring that the app's end-to-end operation performs as expected. Test cases for functional testing, performance testing, battery usage, usability testing and security testing are predominant to improve QA while setting test cases.

MOBILE APPLICATION TESTING CYCLE

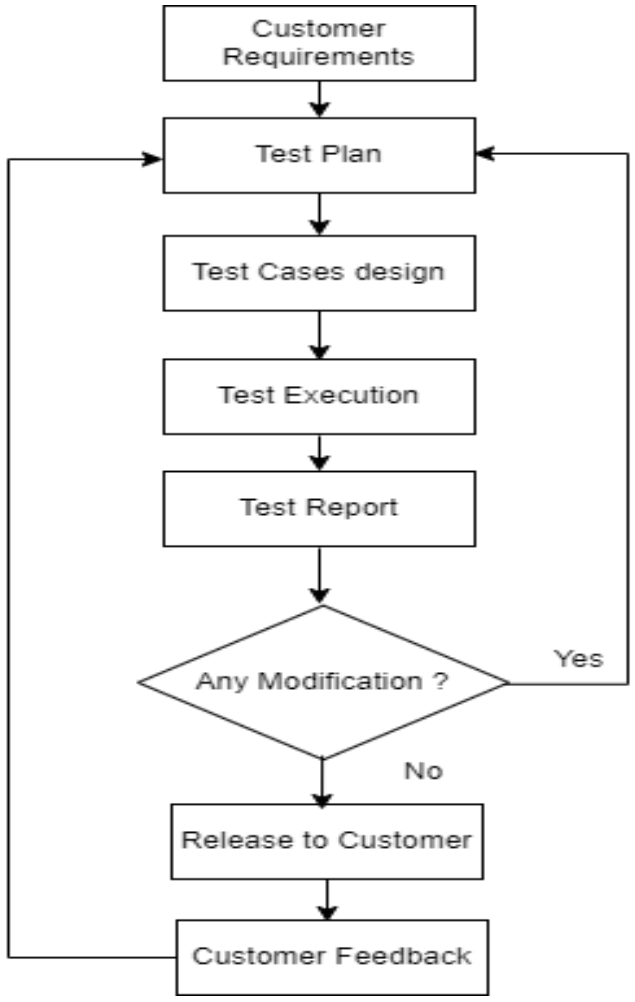

Fig. 1. Testing cycle of mobile apps

COMMON BUGS IN MOBILE APPLICATIONS

Bugs that we encounter on a daily basis may be classified into three types: App-specific bug, Platform-specific bug, and Specific bug.

App-specific bug: They are relevant to the app's business logic. They may be difficult to identify, therefore having a thorough understanding of the app will be really beneficial. It is also critical to document test cases for this sort of problem.

Platform-specific bug: Each mobile platform (Android, iOS) has its own set of issues that are related to how the operating system functions.

Specific bug: Specific problems relating to the app are underlying architecture.

D. Common hampering to use APP

Most of the apps behave in an unwanted way while using for a long time. Some of the common issues are addressed below.

Random crashes: Random crashes can occur for a variety of reasons, including poor memory management, a loss of the software lifecycle, insufficient testing prior to app release, a poor network connection, or simply too much code.

Logging in / Singing up: Most applications now enable users to create an account, join up for additional capabilities, and log in for exclusive information. This allows the app's owner to delegate what material particular users may access, as well as gather and track data about these users' activity.

Connection and speed: Your user does not want to wait many seconds or even minutes for the next screen or a specific page to load. Users want an app that is quick and easy to use, which is exactly what you have to provide.

Slow interaction: Some applications become slow when installing. When users use their various apps ,some apps slowly respond due to hardware issues, mostly software issues.

Unable to handle Interruptions: When users use your mobile app, they will be subjected to several other external factors. They may get an incoming call, a text message, a notice from another program, or even enter power saver mode on their own. If your program is unable to handle these disruptions, they will begin to have a detrimental impact on the user experience. Even if the user intended to continue using your app, your app may crash when that text message displays.

Permission Issues (camera, mic, etc.): Depending on the functionality and features available in your mobile app, the user may be required to provide permission for it to utilize the device's location, camera, microphone, or anything else. App permissions can improve the user experience, but they can also cause a lot of misery if they aren't working properly. We've heard of programs inadvertently turning on cameras or movies in the background when the user is unaware.

Various Screen Sizes: Users are distinct in a variety of ways, one of which is the device they use. While there are usually just a few operating systems to consider, there are nearly always hundreds of various screen sizes to consider – especially for Android users.

Crash after tapping on button: This is similar to a "time bomb" placed in your software. Typically, this refers to buttons that are "laid" deep inside the program (e.g., inside settings) and are easy to ignore. When you click on such a button, the application crashes.

Too Complex for the Users: In the aim of improving the user experience, app developers frequently make it a complicated affair for the consumers. The main reason Apple was able to generate so much attention so fast was its ease of use. If a user can't find the correct buttons in your software in one go, it's not performing its intended function.

Slow changing of portrait and landscape orientation: The mobile application must perform correctly in both horizontal and vertical orientations. However, you must first ensure that there are no complications or delays or other challenges while moving between these two selections. When moving between programs, all information must be saved.

IV. EVALUATION

This section describes the evaluation and analysis of a survey conducted on current mobile users for recent real-world mobile applications. Based on the assessment and analysis of our study, it is crucial for mobile app owners, including mobile manufacturers, to focus on the issues that customers are facing. They must be aware of these vulnerabilities and ensure that those are resolved before releasing or uploading to the Google Play store.

A. Dataset details

For our evaluation, we collect input from Android mobile phone users as a dataset. In this poll, 500+ Android users participated, with a maximum of 60 apps and a minimum of 5 apps per user. Each user uses 40 or more applications on average. We collect information using the questionnaire approach, with 9 questions supplied through Google form for user feedback. The questions address user experience criteria such as random software crash, logging in / signing in, delayed interaction, inability to handle interruptions, resource access, screen adjustment failure, and hardware crash. Details of the questionnaires: (Q1) whether it crashes (randomly) while using various apps? (Q2) Is it necessary to sign up/log in while installing/using apps? (Q3) Does it get slow to interact after installing/using the app? (Q4) Does any app become slow/shutdown/stop when performing other tasks (phone receiving, texting, other app use, etc.)? (Q5) Is it necessary to have access to your location, camera, or microphone while installing or using any apps? (Q6) Does any app experience delay or failure as a result of changing screen sizes? (Q7) Does it crash after pressing any button? (Q8) Are there any apps that are too complex to use? (Q9) Does it support fast switching between portrait and landscape orientations?

B. Analysis

Based on their input, we examined the user experience. There are around 500+ users that utilize over 18000 apps and offer feedback on problems experienced. Table1 shows an overview of their input.

TABLE I. DETAILS OF USER RESPONSES

Total attendee in survey: 501				
Evaluation Criteria	**No. of user faced bugs**	**% of bug happened per user**	**No. of user faced no bugs**	**No. of bugs (%)**
Software Crash	340	67.86	161	32.13
Logging in / Singing up	229	45.71	270	53.89
Slow interaction	266	53.09	233	46.50
Unable to handle Interruptions	279	55.69	222	44.31
Resources access	334	66.67	167	33.33
Screen adjustment failure	234	46.71	267	53.29
Hardware Crash	174	34.73	327	65.26
Unfriendly features	211	42.12	286	57.08
Slow screen rotation	222	44.31	278	55.48

In this case, 340 people out of 501 suffered a random application crash. Almost 67.86 percent of users have reported that their apps fail at any time for no apparent cause. Similarly, 229 users out of 501 faced login/sign up issues with their related applications. While installing or using their apps, 53% of users noted that they interacted slowly. According to 55% of users, certain apps are unable to handle other activities such as call reception, texting, or other difficulties when using the apps. While installation, certain apps require resources (such as a location, camera, or other identification). This is a problem that is reported by 66% of users. 46% of users claim that some apps fail to adjust various screen sizes of phones. Hardware-related crashes (specifically, crashes upon touching any button) are reported by 34% of users. User demand is for user-friendly features, yet 42 percent of users have recognized that certain applications are not user pleasant. They are far too complicated. Screen rotation is a function of any Android phone, yet 44 percent of users have reported that certain applications are not familiar with screen rotation. Overall user experience shown in fig. 2.

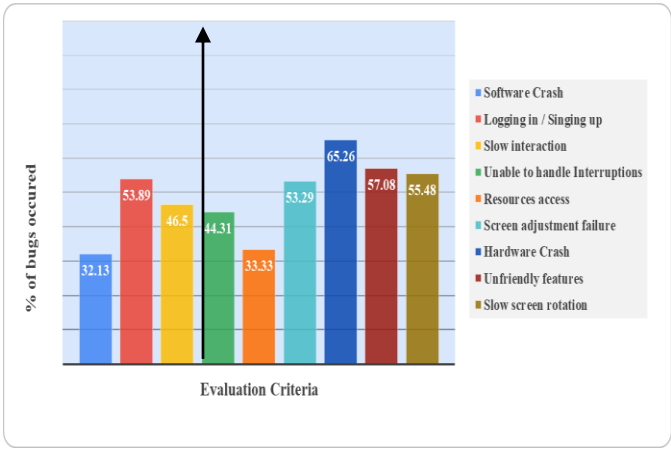

Fig. 2. Percentage of bugs faced by user for different evaluation criteria

End customers usually expect user-friendly, bug-free apps, however the majority of survey respondents reported experiencing random software failures and resource access difficulties that were extremely concerning. Other issues such as login/sign up, delayed interaction, inability to interrupt, and screen adjustment are also encountered by around 50% of users. The other concerns, such as hardware and rotation issues, were encountered by a smaller number of users. As a result, it is up to the owners of mobile apps to resolve these difficulties so that customer satisfaction does not decrease.

V. CONCLUSION

Recently, the number of Android smartphones is increasing rapidly, and their popularity over computers and laptops is expanding by the day. Due to the huge number of apps for various types of mobile devices, it is quite difficult to develop any Android applications that are bug free and provide complete customer pleasure. In this article, we introduce fundamental testing approaches for mobile applications before entering into real-world challenges with mobile apps. Some bugs may happen when a user utilizes applications in a real-world context. We conduct a survey on experienced users for current real-life bugs of mobile applications to explore usability problems and other software and hardware concerns raised by real users. In this case, four problems out of nine were reported by more than half of the users, with the most common issues being software crashes and resource access concerns. hardware related crashes have experienced less than others problems. Although our survey is insufficient for analyzing total user level defeats, it is an estimate for app owners to assess real-world defeats. It is suggested that app owners resolve any bugs that may arise before publishing their apps. In the future, we will conduct a perfect survey for better analysis at the user, software, and hardware levels.

REFERENCES

[1] Muccini, Henry, Antonio Di Francesco, and Patrizio Esposito. "Software testing of mobile applications: Challenges and future research directions." 2012 7th International Workshop on Automation of Software Test (AST). IEEE, 2012.

[2] Nimbalkar, Ravi Ramchandra. "Mobile application testing and challenges." International Journal of Science and Research 2.7 (2013): 56-58.

[3] http://en.wikipedia.org/wiki/Mobile_application_testing.

[4] Satyanarayanan, Mahadev. "Fundamental challenges in mobile computing." Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing. 1996.

[5] http://developer.android.com/reference/android/test/ AndroidTest-Case.html

[6] Beck, E., et al. "Experimental evaluation of techniques for usability testing of mobile systems in a laboratory setting." Proceedings of OzCHI 2003, Brisbane, Australia. CHISIG, 2003.

[7] Villanes, Isabel Karina, Erick Alexandre Bezerra Costa, and Arilo Claudio Dias-Neto. "Automated mobile testing as a service (AM-TaaS)." 2015 IEEE World Congress on Services. IEEE, 2015.

[8] Huang, Junxian, et al. "Anatomizing application performance differences on smartphones." Proceedings of the 8th international conference on Mobile systems, applications, and services. 2010.

[9] Froehlich, Jon, et al. "MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones." Proceedings of the 5th international conference on Mobile systems, applications and services. 2007.

[10] Wac, Katarzyna, et al. "Studying the experience of mobile applications used in different contexts of daily life." Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack. 2011.
[11] Hornbæk, Kasper. "Current practice in measuring usability: Challenges to usability studies and research." International journal of human-computer studies 64.2 (2006): 79-102.
[12] P. Dinda and et al. “The user in experimental computer systems research”. In Proc. of the Workshop,on Experimental Computer Science, (2007)
[13] Nie, Jin, and Xianling Hu. "Mobile banking information security and protection methods." 2008 International Conference on Computer Science and Software Engineering. Vol. 3. IEEE, 2008.
[14] Pousttchi, Key, and Martin Schurig. "Assessment of today's mobile banking applications from the view of customer requirements." 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the. IEEE, 2004.
[15] Li, Ying, and Can Zhang. "Customer's adoption decision analysis of mobile banking service." 2010 International Conference on Management and Service Science. IEEE, 2010.
[16] Enck, William, et al. "A study of android application security." USENIX security symposium. Vol. 2. No. 2. 2011.
[17] Ahmed, A.S., Salah, H.A. and Jameel, J.Q. 2020. Software Development for First Aid Decision Support System. Iraqi Journal of Science. 61, 1 (Jan. 2020), 202-214. DOI:https://doi.org/10.24996/ijs.2020.61.1.23.
[18] Altaie, A.M., Hamo, A.Y. and Alsarraj, R.G. 2021. Software Fault Estimation Tool Based on Object-Oriented Metrics . Iraqi Journal of Science. (May 2021), 63-69. DOI:https://doi.org/10.24996/ijs.2021.SI.2.7.
[19] Noori, K.S. and Fahad, A.A. 2020. Monitoring and Enhancement of Mobile System Performance. Iraqi Journal of Science. 61, 9 (Sep. 2020), 2418-2425. DOI:https://doi.org/10.24996/ijs.2020.61.9.28.
[20] Shaik A., Reddy C. R. K., Manda B., Prakashini. C and Deepthi. K. 2010. “Metrics for Object Oriented Design Software Systems: A Survey”, Journal of Emerging Trends in Engineering and Applied Sciences (JETEAS), pp.190-198.
[21] Wukkadada, Bharati, Ramith Nambiar, and Amala Nair. "Mobile operating system: Analysis and comparison of Android and iOS." International Journal of Computing and Technology 2.7 (2015): 273-276.
[22] Cotroneo, Domenico, et al. "Software aging analysis of the android mobile os." 2016 IEEE 27th international symposium on software reliability engineering (ISSRE). IEEE, 2016.
[23] Rhmann, Wasiur, et al. "Software fault prediction based on change metrics using hybrid algorithms: An empirical study." Journal of King Saud University-Computer and Information Sciences 32.4 (2020): 419-424.
[24] Belachew, Ermiyas Birihanu, Feidu Akmel Gobena, and Shumet Tadesse Nigatu. "Analysis of software quality using software metrics." International Journal on Computational Science & Applications (IJCSA) 8.4/5 (2018).
[25] Gupta, Dharmendra Lal, and Kavita Saxena. "Software bug prediction using object-oriented metrics." Sādhanā 42.5 (2017): 655-669.