

Interface Features, Program Complexity and Memorability as Indicators of Learnability of Mobile Social Software

Nelson Bogomba Masese^{1,2}, Geoffrey Muchiri Muketha³, Samuel Mungai Mbuguah¹

¹Department of Information Technology, Kibabii University, Kenya

²Department of Computer Science, Kabarak University, Kenya

³Department of Information Technology, Muranga University of Technology, Kenya

Abstract: *Social Computing aims to support the tendency of humans to interact with mobile devices. Technology reinforces this interaction by producing appropriate responses that then lead to improved communication between humans and computational devices. Although latest developments in mobile phone technologies have opened the way for a new generation of mobile social applications that allow users to interact and share information, there is still very limited user support information on how to use different applications. This problem either increases the learning curve of the users, thereby adversely affects their overall efficiency. The main purpose of this paper is to analyze factors that affect the learnability of mobile social software. A sample of 361 respondents was selected, with 345 respondents returning feedback. Primary data was collected through the use of questionnaires and interviews targeting mobile social users in Nakuru County Kenya. Three social networks were used, namely, WhatsApp, Facebook and Twitter. Data analysis was done using descriptive statistics. Findings indicate that interface features affect learnability across the three social networks, with learnability of WhatsApp turning out to be higher than that of compared to Facebook and Twitter. Findings also indicate that more than 60% program supports compatibility with other applications while 59.4% of the respondents agreed that maintaining language is cheap across the three social networks. Other findings indicate that WhatsApp's memorability is easy to execute compared to that of Facebook and Twitter.*

Keywords: Mobile social software, social computing, software learnability

1. Introduction

Learnability is the capability of a software product to facilitate the user to learn its application. A learnability measure should be able to measure the time and effort required by users to use particular interfaces and perform particular functions. It also includes the ease of use of documentation such as manuals, help system and demos. Further the social software should have consistency and conformity to meet user expectations, develop simple applications which serve the purposes and require less learning effort, error robustness or applications and feedback about the success or failure of actions helps users to develop successful strategies for interacting with the application[1]. For example, user documentation and the help system should be thorough, complete, and precise. Besides, it must also solve all of the user's concerns so as to how to accomplish common tasks.

According to [2], he defined learnability is defined as the relation of performance to training and frequency of use, whereby the novice user's learning time with specified training and retention on the part of casual users. Learnability has also been defined as the novices' ability to reach a reasonable level of performance rapidly[3]. While this definition indicates the general idea of learnability, it is unclear what a reasonable level of proficiency would be. Furthermore, it doesn't account for the learning that occurs after such a level has been reached. While [3] considers efficiency of use as a separate aspect of usability, Nielsen defined as the performance level of an expert user, the transition from reasonable to expert performance is not considered. Despite this potential shortcoming, defining learnability based on initial user experiences is common.

[4] defines it similarly as the time it takes members of the user community to learn how to use the commands relevant to a set of tasks. In this paper learnability has been defined as how quickly a new user can begin efficient and error-free interaction with a social software, the new user can execute most of advanced functionalities of the software.

The objective of the paper is to analyze user interface features, program complexity and memorability as learnability factors affecting mobile social software. The purpose of the study is to provide insights to assist software designers, software developers and open source owners to improve their software products in a way that best supports easy learnability of the users.

2. Related Works

This section provides a detailed analysis of related works on learnability.

2.1 Basic Concepts

The following are learnability factors considered in this paper it includes: learnability factors, interface design principles, interface features, program complexity and memorability.

2.1.1 Learnability Factors

2.1.1.1 Predictability

Predictability can be defined as the ease with which users can determine the result of their future interactions with

the interface, based on the past interaction history [6]. There are many degrees to which predictability can be as the knowledge can be restricted to the presently perceived information, so that the user need not remember anything other than what is currently observable views predictability as referring to determinism and visibility of operations [7].

Predictability is a user centered concept which refers to the deterministic behavior of the system from the user's perspective [8]. An interactive system possesses predictability if it offers and helps the user to understand and predict the consequences of a certain action with this principle, the user must be able to choose a certain action in the interactive system and be confident that this action will work correctly and will give the expected outcome [9].

2.1.1.2 Synthesizability

Synthesizability, after the user executes an action to satisfy a certain goal, it is critical that the interactive system provides some sort of feedback. Usually there are different ways to represent this feedback, but all of them must provide the user with effective and reliable information about the effect of the action after its execution and the changes that it performed [10].

2.1.1.3 Customizability

Customizability is what many users desire they want the interface to adapt to their needs and customs. This modification can be either done by the user (adaptability). Some of the most common customizations that are required include: providing choice of methods allowing short-cuts and permit users to change features [11].

Software customization helps in data sharing within similar applications, it leads to better software integration and it provides improved time and resource management hence improving efficiency [9].

2.1.1.4 Familiarity

The familiarity principle is concerned with the ability of an interactive system to allow a user to map prior experiences, either real-world or gained from interaction with other systems, onto the features of a new system. This is an externally oriented criterion, which captures the extent to which the user experiences a real-world parallel to the system. Familiarity attempts to measure the correlation of users knowledge with the skills needed for effective interaction [12].

Several interactive systems such as software applications are designed in such a way that their goal is to help the user to create an analogy between an object or situation and the computer application. The users will be more familiar with that application because they will be able to apply the knowledge that they have about the object or situation to the computer applications or computer based softwares [13].

2.1.2 User Interface Design Principles

User interface design principles are intended to improve the quality of user interface design. The design should make simple, common tasks easy, communicating clearly and simply in the user's own language, and providing good shortcuts that are meaningfully related to longer procedures. The paper considered the following design principles and is discussed below.

2.1.2.1 Consistency

As well as matching people's expectations through consistency, layout and interactions the way in which they are used should be consistent throughout the process and between related applications [14,15]. The interface should be consistent in that, wherever possible, comparable operations should be activated in the same way. By maintaining consistency users learn more quickly, this can be achieved by re-applying in one part of the application their prior experiences from another [16]. An added bonus of keeping elements consistent is that you can then use inconsistency to indicate to users where things do not work the way they might expect, breaking consistency is similar to knowing when to be unconventional [17].

2.1.2.2 Error Detection and Recovery

The best way to reduce the amount of errors a user makes is to anticipate possible mistakes and prevent them from happening in the first place. If the errors are unavoidable we need to make them easy to spot and help the user to recover from them quickly and without unnecessary friction [18]. The system should have error detection anticipate possible errors and provide feedback that helps users verify that the user have done what they intended to do and what they intended to do was correct, the system should provide the important to remember that providing feedback by changing the visual state of an object or item is more noticeable than a written message also the system should have error recovery if the error is unavoidable provide clearly marked ways for the user to recover from it for example provide back, undo or cancel commands [19]. If a specific action is irreversible it should be classed as critical and you should make the user confirm first in order to prevent slip ups. Alternatively you can create a system that naturally defaults to a less harmful state. For example if one closes a document without saving it the system should be intelligent enough to know that it is unlikely that the intended action and therefore either auto-save or clearly warn me before closing [15].

2.1.2.3 User Guidance

User guidance refers to the documentation for a product or service provided to the end users. The user documentation is designed to assist end users to use the product or service. This is often referred to as user assistance [20]. The user documentation is a part of the overall product delivered to the customer to enhance learnability of the software product. The interface should provide meaningful feedback when errors occur and provide context-sensitive user help facilities [21].

2.2 Mobile Social Software Interface features

In order to achieve learnability the mobile social software should always aim to achieve the following factors:

2.2.1 Simplicity

Social media applications are rather simple in terms of color scheme and graphics. The color scheme usually consists of a few colors along with slight monochromatic variations, the background is generally white, updates status updated are often highlighted with a light color as well, usually green or yellow, alerts are usually highlighted with a red background color [22].

The graphics are always very simple and are used very sparingly. The most important reason is the simple fact that vivid visual design isn't really useful on social networking sites [23]. Social applications are supposed to provide a shared environment where the content can be easily produced and where conversations can take place a strong visual design would create unnecessary noise and make it harder for users to focus on their conversations [24].

The colors on social media and social networking sites are always calm and supportive, rather than bright and unbearable, features do not fight for attention, many of them remain invisible most of the time in fact, most social media interfaces are context-sensitive, displaying many features only on demand, with such a large amount of data and functions, bright colors would simply get in the way and distract the user [25].

2.2.2 Prominent and Functional Search

Good search functionality is undoubtedly the pinnacle of good usability and good user interface, in the social media a search functionality is a must simply because of the vast amount of available information [26]. The search, however, has multiple dimensions, apart from the traditional content search the social applications also provide an advanced search of connections in the social graph that can be groups, communities or interests. The upper-right corner of the site is the most suitable location for a search box, Users expect it there a search input area and a clearly visible search button [22].

2.2.3 Prominent Call-To-Action-Buttons

Social applications contain many functions that need to be communicated in some way consequently, buttons and links need to be placed almost on every page except a sign-up form is probably the only reasonable exception [27]. Some links relate to navigation and some let the user adjust specific application function, Buttons are often used to animate users to actions, while links are often more passive and subtle, buttons are also often larger, more vivid and more memorable, despite the task the button performs, it needs to be large and clickable, Often social networking sites have only few call-to-action-buttons that are supposed to motivate users to actions these buttons are usually designed and placed prominently, while other design elements remain very subtle and simple [22].

2.2.4 Calm Separation of Elements

A meaningful organization and presentation of various chunks of information is probably one of the advanced design problems that designers of social user interface have to deal with, in order to make the content readable, scannable and easy to perceive, content blocks need to be visually separated. Each and every element needs to be defined as and presented a separate element in some way. In fact, the separation of elements in a layout is one of the most simple ways to achieve a more clean user interface that the user can easily interact with [28]. However, if many elements are visually separated, the interface contains more chunks of information and consequently the layout becomes more complex [29].

2.2.5 User-Centric User Interface

Since social software is social, it provides users with a user interface that is strongly focused on the personal interests of its users, because social media and social networking sites live by the actions of its users, it's no wonder that social user interfaces are extremely user-centric. Twitter, Facebook and other social applications put the user in the middle of the application, focusing on the little details of users' profiles, suggesting new friends, interests, events and groups. in the attempt to extend their social circles and intensify the engagement of the user [30].

2.3 Program Complexity

Program complexity is a term that encompasses numerous properties of a piece of software, all of which affect internal interactions. There is a distinction between the terms complex and complicated [31]. Complicated implies being difficult to understand but with time and effort, ultimately knowable, complex, on the other hand, describes the interactions between a number of entities as the number of entities increases, the number of interactions between them would increase exponentially, and it would get to a point where it would be impossible to know and understand all of them [32]. Similarly, higher levels of complexity in software increase the risk of unintentionally interfering with interactions and so increases the chance of introducing defects when making changes, but in more extreme cases, it can make modifying the software virtually impossible [33].

2.4 Memorability

Once a user has taken the time to learn how to navigate a mobile social software and find what they are looking for, they need to be able to remember how to do it when they come back. A mobile social software needs to have high memorability in order to execute various tasks that are assigned into [34]. Memorability is a measure of how easy mobile social software is to remember after a substantial time lapse between visits [35].

Setting Memorability evaluates the degree of memorability of user settings. Not all settings are always remembered for example, some software may or may not remember a customized color scheme, If none of a software task are

recalled upon a user's return then the task memorability variable value is 0, If some of a software task are recalled upon a user's return then the task memorability variable value is 0.5 and if all of a software task are recalled upon a user's return then the task memorability variable value is 1[36].

Stateful Memorability evaluates the degree of memorability of user state the portions of the experience which, upon user interaction, indicate a preference or customization, but are not directly settable as a user setting[37].

3. methodology

The paper adapted mixed research design. The primary data used in the paper was collected from a survey carried at Nakuru county, targeting mobile social software users. Survey was used as it allows you to measure the significance of the mobile social software on the overall population, the target population was 6,000 and the sample size 361 of respondents was selected.

Experiment were also used to test the memorability of the mobile social software users by assigning the users specific tasks to perform the sample size of 30 respondents was selected. While interviews were used to validate the data that was collected by the questionnaires since domain experts were interviewed the sample size of 20 respondents was selected.

The study achieved 95.3% response rate of the target. This response rate was considered appropriate for analysis and reporting as supported by [38] indicating that response rate of 90% and above is excellent. Descriptive statistics were computed, the results are tabulated in the next section.

4. Results

4.1 Assessment User Interface Features Analysis

Descriptive statistics were computed in order to determine respondent's behavior concerning interface features of the social software. The findings revealed that 43.2 % of the respondents agreed that the interface is clear for the same operations in WhatsApp. It was evident that 21.4% of the respondents agreed that the interface feature was clear in Facebook. However, 18.3% disagreed that the interface feature was clear in Twitter social software. It was evident that 19.4% and 42.0% of the respondents agreed that the user gets minimum surprise by the behavior of the system in Facebook and Whatsapp respectively. it was noted that, only 7.5% of respondents agreed that minimum surprise by the behavior of the system is experienced in Twitter social software.

It was found out that 70.4% of the respondents agreed that the interface includes mechanisms to allow users to recover from errors across the three social networks of Facebook, WhatsApp and Twitter whereas 29.6% has a different opinion. It was realized that the leading social software with an interface that provides the feedback when

errors occur and provide user help facilities was a WhatsApp software (29.9%) followed by Facebook(13.6%).According to Table 1, 50.8% of the respondents affirmed that the icons or commands in the interface are clear and linked to the specific functions across face book, WhatsApp and Twitter. Regarding the fact whether customizing some features is easy in the interface, it was realized that(60.3%) confirmed that the operation existed across the three networks. Conversely (39.7%) of the respondents negated that customization was easy of interface features. This findings therefore points out that interface features may affect learnability across the three social networks. As a matter of fact, learnability of WhatsApp is high as compared to Facebook and Twitter. The findings are displayed in Table 1

Table1: User Interface Feature of social Software

Software Feature	FaceBook Percent (%)		WhatsApp Percent (%)		Twitter Percent (%)		Total (%)
	Yes	No	Yes	No	Yes	No	
The interface is understandable for the same operations	21.4	7.2	43.2	3.8	6.1	18.3	100
The user gets minimum surprise by the behavior of the system	19.4	9.3	42.0	19.4	7.5	2.3	100
The interface has error handling mechanism	19.7	9.0	42.9	18.6	7.8	2.0	100
The interface provides the feedback and provide appropriate user help facilities	13.6	15.1	29.9	31.6	6.7	3.2	100
There navigation in the interface is satisfactory	20.9	7.8	42.9	18.6	6.4	3.5	100
The icons or commands are aligned to specific functions	14.8	13.9	32.5	29.0	3.5	6.4	100
Hierarchical of the interface is simple to understand.	20.0	8.7	42.0	19.4	8.1	1.7	100
There are enough suggestions and prompts towards the right usage.	17.4	11.3	35.9	25.5	7.2	2.6	100
To customization of features is easy.	17.4	11.3	38.6	22.9	4.3	5.5	100
Input method is appropriate	17.4	11.3	35.9	25.5	7.2	2.6	100

4.2 Assessment of Program Complexity

Analysis was computed to determine respondents' assessment of program complexity of social software under investigation. It was observed that 41.2% and 46.4% of respondents affirmed that in Whatsapp, the program in the social software and in the users' guide is clear and easy to understand and that the program is consistent for various operations respectively. Similarly, 3.5% and 2% of the twitter users disagreed with the feature.

It was also noted that 21.2% and 16.2% of the Facebook users agreed that program in software and users' guide is clear and easy to understand. Regarding whether the program manages well errors during data entry. It was noted that 51.1% of the respondents disagreed with the feature across Facebook, WhatsApp and Twitter. However

it was realized that (48.9%) of the respondents agreed that the program manages well errors during data entry across the three social softwares under investigation. Analyzed data revealed that the program is efficient in execution when using WhatsApp with 41.2% followed by Facebook with 22.0% and finally Twitter with 5.2%. The program supports compatibility with other applications in WhatsApp as posted by 35.4% of the respondents while 20% of the respondents support Facebook. It was observed that 5.5% support in Twitter.

This finding implies that program complexity of a software has an effect on learnability in that software program which is easy and manageable by users will ultimately influence positively its learnability. Table 2 illustrates the findings.

Table 2: Program Complexity of Social Software

Software Feature	FaceBook		WhatsApp		Twitter		Total (%)
	Percent (%)		Percent (%)		Percent (%)		
It requires more system resource	41.2	7.5	6.4	20.3	21.2	3.5	100
Errors are managed well during data entry	13.9	14.8	30.4	31.0	4.6	5.2	100
The program has security features	22.9	5.8	41.7	19.7	6.1	3.8	100
The program provides sufficient documentation	16.5	12.2	29.3	32.2	3.5	6.4	100
The program is efficient in execution	22.0	6.7	41.2	20.3	7.2	2.6	100
The program allows large number of characters from the user.	35.1	10.1	18.6	26.4	5.2	4.6	100
Program maintenance is cheap.	16.5	12.2	37.7	23.8	5.2%	4.6	100

4.3 Assessment of Memorability

Experiment was done to assess memorability level for Facebook, WhatsApp and Twitter the main objective for the experiment was to attempt to find out the relationship between memorability and learnability attributes. The findings revealed that regarding memorability across all the three networks, 33.3% of the respondents found it easy to navigate to a specific function. It was overwhelming to note that 16.7% and 13.3% of the Twitter and Facebook users found it difficult to perform the specified task. Respondents were given a task to block account.

It was observed that nearly all the respondents across the three social networks had difficulty in operating the task provided such that Facebook(13.3%) WhatsApp(20%) and Twitter(16.7%) respectively.

It was observed that Facebook and Twitter respondents found it easy to change settings for security and privacy with equal measure of 13.3% while 16.7% of WhatsApp user demonstrated with ease the task provided. In summary the results indicated that since WhatsApp memorability is easy to execute, its learnability as compared to the other two social software is high. Table 4 shows the findings of the analysis. The finding implies that learnability will increase in a user when he or she is capable of remembering how to operate the icons earlier encountered.

Table 4: Memorability of Social Software

Software Feature	Face Book		WhatsApp		Twitter		Total (%)
	Percent (%)		Percentage (%)		Percentage (%)		
	Easy	Difficult	Easy	Difficult	Easy	Difficult	
Navigate to a specific function	6.7	13.3	33.3	6.7	0.0	16.7	100
Create a new group	3.3	20.0	10.0	9	0.0	17.7	100
Change the colour of notification	10.0	6.7	6.7	10.0	6.7	13.3	100
Search a given account	9.0	0.0	14.3	6.7	7.7	12.3	100
Change the current status	10.0	10.0	10.0	13.3	3.3	16.7	100
Block account	6.7	13.3	13.3	20.0	0.0	16.7	100
Change settings for security and privacy	13.3	13.3	16.7	23.3	13.3	10.0	100

5. Conclusion

It can be concluded that the leading social software with learnable user interface that provides the feedback when errors occur and provide user help facilities influences software learnability, while in program complexity it can

be concluded that the program in the social software and in the user's guide is clear and easy to understand and that the program is consistent for various operations influences learnability lastly Memorability can be concluded that the users who can remember to perform an operation with ease have high learnability. This finding therefore validates memorability as a factor influencing learnability.

Conversely, users having difficulty in executing such operations will have low learnability for the given software.

References

- [1] Al-Qutaish, R. E. (2010). Quality models in software engineering literature: an analytical and comparative study. *Journal of American Science*, 6(3), 166-175.
- [2] Dubey, S. K., Rana, A., & Sharma, A. (2012). Usability Evaluation of Object Oriented software system using fuzzy logic Approach. *International Journal of Computer Applications*, 43(19), 1-6.
- [3] Nielsen, J. (1996). Usability metrics: Tracking interface improvements. *IEEE Software*, 13(6), 12.
- [4] Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., and Diakopoulos, N., Grand (2016) *Challenges in HCI*, ACM Interactions Sixth Edition, Pearson
- [5] Grossman, T., Fitzmaurice, G., & Attar, R. (2009). A survey of software learnability: metrics, methodologies and guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 649-658). ACM.
- [6] Park, J., Han, S. H., Kim, H. K., Cho, Y., & Park, W. (2013). Developing elements of user experience for mobile phones and services: survey, interview, and observation approaches. *Human Factors and Ergonomics in Manufacturing & Service Industries*, 23(4), 279-293.
- [7] Lambe, P. (2014). *Organising knowledge: taxonomies, knowledge and organisational effectiveness*. Elsevier.
- [8] Bourguet, M. L. (2011). *Uncertainty and Error Handling in Pervasive Computing: A User's Perspective*. Intech Open Access Publisher.
- [9] Jameson, A. (2009). Adaptive interfaces and agents. *Human-Computer Interaction: Design Issues, Solutions, and Applications*, 105, 105-130.
- [10] Raman, V., Lignos, C., Finucane, C., Lee, K. C., Marcus, M. P., & Kress-Gazit, H. (2013). Sorry Dave, I'm Afraid I Can't Do That: Explaining Unachievable Robot Tasks Using Natural Language. In *Robotics: Science and Systems*.
- [11] Sundar, S. S., & Marathe, S. S. (2010). Personalization versus customization: The importance of agency, privacy, and power usage. *Human Communication Research*, 36(3), 298-322.
- [12] Chimbo, B., Gelderblom, H., & De Villiers, R. (2011). A comparative analysis of the meaning of 'learnability' for child and adult users.
- [13] Bettini, C., Brdiczka, O., Henriksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2), 161-180.
- [14] Tidwell, J. (2010). *Designing interfaces: Patterns for effective interaction design*. " O'Reilly Media, Inc."
- [15] Shneiderman, B. (2010). *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India
- [16] Scott, B., & Neil, T. (2009). *Designing web interfaces: Principles and patterns for rich interactions*. " O'Reilly Media, Inc."
- [17] Sayago, S., & Blat, J. (2010). Telling the story of older people e-mailing: An ethnographical study. *International Journal of Human-Computer Studies*, 68(1), 105-120.
- [18] Ko, A. J., Abraham, R., Beckwith, L., Blackwell, A., Burnett, M., Erwig, M., & Rosson, M. B. (2011). The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)*, 43(3), 21.
- [19] Traver, V. J. (2010). On compiler error messages: what they say and what they mean. *Advances in Human-Computer Interaction*, 2010.
- [20] Yanez Gómez, R., Cascado Caballero, D., & Sevillano, J. L. (2014). Heuristic evaluation on mobile interfaces: a new checklist. *The Scientific World Journal*, 2014
- [21] Kelkar, S. A. (2015). *Usability And Human-Computer Interaction: A Concise Study*. PHI Learning Pvt. Ltd.
- [22] Tella, A., & Akinboro, E. O. (2015). The impact of social media to library services in digital environment. *Social Media Strategies for Dynamic Library Service Development*, 279-295.
- [23] Rizvanoğlu, K., & Öztürk, Ö. (2010). A close look at the phenomenon: An eye tracking study on the usability of the profile pages in social networking sites. In *Proceedings of International Symposium of Interactive Media Design* (pp. 1-12).
- [24] Murray, J. H. (2011) *Inventing the medium: principles of interaction design as a cultural practice*. MIT Press.
- [25] Akaichi, J. (2014). A medical social network for physicians' annotations posting and summarization. *Social Network Analysis and Mining*, 4(1), 1-8.
- [26] Wimmer, R. D., & Dominick, J. R. (2013). *Mass media research*. Cengage learning.
- [27] Uncel, M. (2011). " Facebook Is Now Friends with the Court": Current Federal Rules and Social Media Evidence. *Jurimetrics*, 43-69.
- [28] Adnan, W. A. W., Hassan, W. N. H., Abdullah, N., & Taslim, J. (2013). Eye tracking analysis of user behavior in online social networks. In *International Conference on Online Communities and Social Computing* (pp. 113-119). Springer Berlin Heidelberg.
- [29] Othman, S. M., Elmasry, R. E. E., & Shaker, S. A. E. A. (2015). The elderly color's preferences to be used in designing Printed Upholstery Fabrics. *International Journal of Innovation and Applied Studies*, 13(2), 430.
- [30] Marenzi, I., Demidova, E., & Nejd, W. (2008). *LearnWeb 2.0. Integrating Social Software for Lifelong Learning*.
- [31] Wolf-Branigin, M. (2013). *Using complexity theory for research and program evaluation*. Oxford University Press
- [32] Mbonimpa, J. C. (2012). *Contribution of Computerised Financial Management Systems In The Functions Of Supreme Court Of Rwanda*
- [33] Kim, Y., Choi, J. S., & Shin, Y. (2014). A decision model for optimizing the service portfolio in SOA governance. In *Information and Communication Technologies (WICT), 2014 Fourth World Congress on* (pp. 57-62). IEEE.
- [34] Garzonis, S., Jones, S., Jay, T., & O'Neill, E. (2009). Auditory icon and earcon mobile service notifications: intuitiveness, learnability, memorability and

preference. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems(pp. 1513-1522). ACM.

- [35] Coursaris, C. K., Hassanein, K., Head, M. M., & Bontis, N. (2012). The impact of distractions on the usability and intention to use mobile devices for wireless data services. *Computers in Human Behavior*,28(4), 1439-1449.
- [36] Sherman, M., Clark, G., Yang, Y., Sugrim, S., Modig, A., Lindqvist, J., & Roos, T. (2014). User-generated free-form gestures for authentication: Security and memorability. In Proceedings of the 12th annual international conference on Mobile systems, applications, and services (pp. 176-189). ACM.
- [37] McCarney, Daniel. "Password managers: comparative evaluation, design, implementation and empirical analysis." Carleton University(2013).
- [38] Krejcie, R.V., & Morgan, D.W. (1970). Determining Sample Size for Research Activities. *Educational and Psychological Measurement*, 30,607-610

Author Profile

Masese Nelson Holds Bachelor of Science Information Technology from Sikkim Manipal university, masters of computer applications Degree from Periyar university India, currently a PhD candidate at Kibabii university, His research interest are mobile applications and security.

Geoffrey Muchiri Muketha Holds Bachelor of Science Degree in Information Science from Moi University, Kenya, Master of Science degree in Computer Science from Periyar University, India. He completed his PhD in Software Engineering from University Putra Malaysia, Malaysia. His research interests are Business Process modeling and measurement.

Samuel Mungai Mbuguah Holds and BSc honours in Electronics. Engineering (ECE/UK). Masters of Science In Computer based Information Systems from University of Sunderland and PhD in Information Technology of Masinde Muliro University of Science and Technology.