GSJ: Volume 7, Issue 3, March 2019, Online: ISSN 2320-9186

www.globalscientificjournal.com

Design Thinking for Gesture-based Human Computer Interactions

Kelvin Wambani Siovi [1], Cheruiyot Willison Kipruto [2] , Agnes Mindila [3]

[1] School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology Nairobi, Kenya

[2] School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology Nairobi, Kenya

[3] School of Computing and Information Technology, Jomo Kenyatta University of Agriculture and Technology Nairobi, Kenya

Abstract

Gesture is a way of communication which involves body language and can be defined with or without spoken words. The primary goal of gesture interaction applied to Human-Computer Interaction (HCI) is to create systems, which can identify specific human gestures and use them to convey information or control devices. Gesture interaction is already a promising input modality in modern gaming, augmented and virtual reality. Gestural input makes computing more "natural" by enabling communication with a computer the same way we also communicate with one another. Computer users face challenges when engaging and implementing gesture interactions, that is: complex coordination of all gestures regularly, mastering approachability to different gestures, muscle fatigue and a lack of supporting data input that is seamless with gesture interaction. In combination with a lack of tracking success or failure, end users often struggle to execute gestures correctly. To compensate this, users require a design thinking framework that supports them during execution of gestures.

Keywords: *Design Thinking, Gesture Interaction's, Framework, modalities*

1. Introduction

The maturing of sensing technology over the past few decades has brought gesture-based interaction accessible to everyone. The prevalence of gesture-based interaction has changed how people interact with computing systems in various domains, e.g., gaming, education, e-commerce, and healthcare. Gesture human computer interactions could be more user-friendly and make it quick for manipulations to provide a platform for enjoyable computing [1].

Human computer gesture recognition is a way for computers to understand human body language, thus building a richer bridge between machines and humans than primitive text user interfaces graphical user interfaces, which still limit the majority of input to keyboard and mouse. Gesture recognition is a technology that has the objective of understanding human movement via mathematical procedures [2]

User's experience challenges when engaging and implementing gesture interactions, that is: complex coordination of all gestures regularly, mastering approachability to different gestures, muscle fatigue and a lack of supporting data input that is seamless with gesture interaction. Unlike touch input, users usually have little experience with gesture interaction[3]. With a structure in place for investigating human computer modalities, there is need to organize gesture interactions into a framework to enable a structured method of understanding and designing gestures interaction using design thinking.

Design thinking is a user-centered approach which draws on both the creative and rational thinking of multi-disciplinary teams to develop better solution to complex problems [4]. Design thinking uses the designer's sensibility and methods to match people's needs with what is technologically feasible and viable [5]. There should always be a decrease in the amount of users' effort, how they should interact with technology, in fact, putting users to work is critical in creating products people love [6]. Design thinking provides a human-centered view of technological artifact design and is therefore an appropriate tool for serious game design. Design thinking is an essential tool for simplifying and humanizing by embedding disclosure information in a familiar user interaction [7]. The main objective of this paper was to develop a design thinking framework that can promote a methodological approach to designing gesture interactions. The second objective is to analyze a set of high-level characteristics which can be compared and contrasted in gesture interaction and the third objective is to evaluate the design thinking framework for gesture human computer interaction.

2. Literature review

2.1 Approaches to Gesture Interaction Design

Approach to gesture interaction design is divided into two mutually interdependent fields: the model and fluid gesture design and are largely based on the disciplinary backgrounds from different previous research. Modern academic investigations stem from [8] which laid out the first clear argument against the longstanding belief that speech and gesture are two fundamentally separate (and indeed, hierarchically valued) modes of relaying information which lead to conclusion that gestures share with speech a computational stage thereby becoming parts of the same psychological structure [9]

Model gesture design interaction use depth sensors with the availability of synchronized information that combines movement, color and depth information to obtain a satisfactory segmentation from user. A series of different filters is applied to facilitate desired user gesture input. [10] details how model gesture interaction design combines color and depth information to obtain a satisfactory hand model segmentation for gesture set inputs.

Model gesture interaction design requires allows interaction with a system approach that is definitive of a multi-device user interface through some levels of abstraction that is the concepts and tasks level, the abstract user interface, the concrete user interface, and the final user interface[11].

Fluid gesture design interaction run autonomously performing functions in response to users conveniently due to designed electronic circuits sense responding to human gestures. Automated entry/exit doors that open by themselves when people walk up to them are examples are fluid gesture designs. Fluid gesture design solutions detect only the presence of a body or hand, so the electronics occasionally respond inappropriately. For instance, a door might open unexpectedly when a pedestrian walking by pauses momentarily near the entrance [12].

2.2 Related Work

2.2.1 Sonification of Affordances for Gesture-Based Interfaces

This framework described by [13] approach combines machine learning techniques for understanding user's gestures, with a method for auditory displays of salient features of the underlying inference process in real time thereby providing user with desired gesture inference sets shown in the following figure.

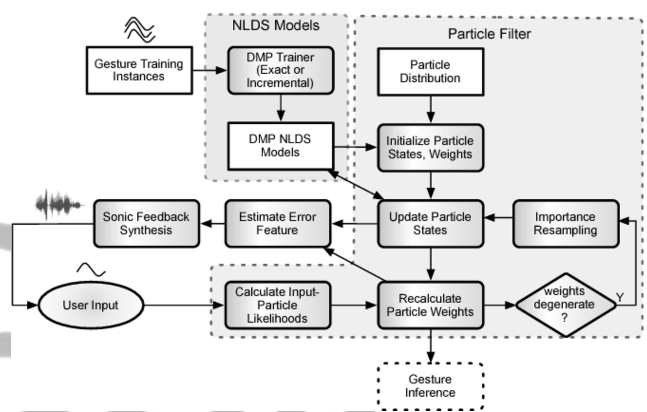

Fig 2.0 A framework of sonification of affordances for gesture-based interfaces

Gesture-based input from the user is acquired by the input device's sensors as an N-dimensional temporal trajectory. A set of particles, given by weighted states of the nonlinear dynamical systems, is evolved in tandem with the gestural input. They maintain a probability distribution that tracks the system's belief as to the correspondence between the user's input and learned models. The error signals and weights that result is then used to derive parameters for a sonification. A key problem with this framework approaches to sonification is the determination of salient parameters for display, which is analogous to the control mapping for gesture interaction input.

2.2.2 A Framework for User-Defined Gestures for Surface Computing

This framework by [14] presents effects of gestures to users and elicits the causes meant to invoke them by using a think-aloud protocol and video analysis, to obtain rich qualitative data that illuminates users' mental models while they are using surface devices like tablets. The effects obtain quantitative measures regarding gesture timing, activity, and preferences resulting in a detailed picture of user-defined gestures and the mental models and performance that accompany them.

The creation of a user-defined gesture sets for surface computing is based on observed user behavior and joins gestures to commands. Referents use reversible gestures, and the same gestures are reused for similar operations. For instance, enlarge, which can be accomplished with four distinct gestures, is performed on an object, but the same four gestures can be used for zoom in if performed on the background, or for open if performed on a container (e.g., a folder). Flexibility exists insofar as the number of fingers rarely matters and the fingers, palms, or edges of the hands can often be used interchangeably.

2.2.3 A Framework for User-Defined Motion Gestures for Mobile Interaction

This framework by [15] demonstrates that uniformity exists among gesture interaction users on parameters of movement and on mappings of gestures onto commands that specify inspired motion gesture set for end-users. To evaluate the degree of consensus among the participants and compare the gesture set, an adoption process of calculating an agreement score for each task.

The problem with the framework is that it requires anything extended while user is not looking at the screen and is then losing feedback so it seems like it's undesirable. Since the tasks selected in the framework required the user to interact with content on the screen after completing the gesture map navigation, navigating previous and next enlists. It ties users to thoroughly to be able to view the screen while performing the gesture.

An agreement score reflects single number the degree of consensus among participants. Gesture sets motions that required one hand had high scores than ones requiring two hands. The following figure below illustrates the agreement score for the gesture set developed by case study participants

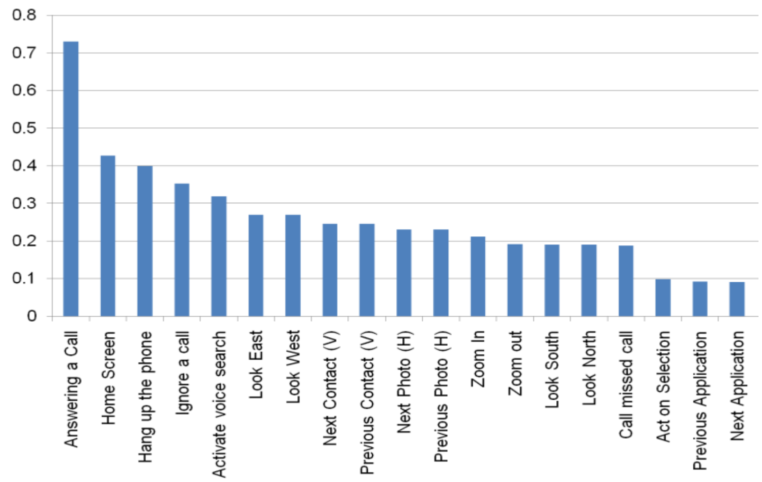

Figure 2.1 An agreement score reflecting the degree of consensus among participants

2.3 Design Thinking

Design thinking is a means of solving complex problems that focuses on the perspective of end users to better determine specific technical requirements. Design thinking builds on the process of empathizing and interacting with the customer from the start of the project a solution that meets the customer's needs and environment [16]

Design thinking converts need into demand. It's a human-centered approach to problem-solving that helps people and organizations become more innovative with the following listed principles [17].

i. Start with needs**user needs not system needs* - Empathy
ii. Doing less – Making it reusable and shareable instead of reinventing the wheel every time.
iii. Design with data - Data drive decision-making from real world behavior, not guesswork.
iv. Do the hard work to make it simple - Making something simple to use is much harder, with underlying systems being complex.

v. Iterate. Then iterate again - Refinements based on feedback. It makes big failures unlikely and turns small failures into lessons.

vi. This is for everyone - Everything built should be as inclusive, legible and readable as possible.

vii. Understand context - Think hard about the context in which they're using our services. Are they in a library? Are they on a phone?

viii. Be consistent, not uniform - Using the same design patterns wherever possible., as it helps people get familiar with your system.

ix. Make things open, it makes things better - With colleagues, with users, with the world. Share code, share designs, share ideas, share intentions, share failures.

2.4 Design Thinking in Practice

Sometimes software is at the center of a product and needs to be integrated with hardware (itself a complex task) and made intuitive and simple from the user's point of view becomes another difficult challenge. In opposition to the traditional design process, design thinking focuses on upfront problem framing before solutions are explored. It is important that the framing be independent of the solutions. Design thinking is an iterative process and relies heavily on prototyping to build knowledge, test and validate concepts [4].

A dozen of other types of complexity that businesses grapple with every day all have one thing in common: People need help making sense of them. Specifically, people need their interactions with technologies and other complex systems to be simple, intuitive, and pleasurable. The role design thinking plays in gesture interaction has been primarily concerned with studies of verbal protocols and very little on addressing consistency concerns [18].

Design thinking builds on the process of empathizing and interacting with the customer from the start of the project a solution that meets the customer's needs and environment. As the name implies, design thinking is a problem-solving framework and not an exclusive project execution framework such as waterfall and agile.

Design thinking starts by defining the problem and then developing a solution with a focus on the customer or user of the final product. As you focus on understanding the customer's problem, you can then create prototype solution. This prototype is then tested allowing you to continue to learn and improve upon your solution [19]

3. Research Methodology

3.1 Data Collection

Data collection was across two studies: an expert interview study and participatory case study. Primary data was gathered from different user application domains; from gaming, educational instructors and design thinking experts. The true value in a population was estimated with a sample of persons, to minimize complication. Rather than just the mean or proportion, the standard error was derived for the variable of interest, used to construct a confidence interval. The main reason of equality in theory is sampling. Random sampling is the basic selection process of sampling and is easiest to understand. Random sampling gives equality of the total population [20].

3.1.1 Research Participants' Profile

There were 5 types participants from the sample population selected, including the researcher who took place in the study, pprofessional design thinking experts, tutorial fellows, students and expert gamers.

3.2 High Level Characteristics' Comparisons in Gesture Interaction

Ten volunteers, seven males and three females, between the ages of 18-28 participated in the study. Gestures were recorded using a software called Camtasia tech smith software which records on screen user activities while computing. The software was responsible for logging the data stream of the accelerometer sensor and locking the screen to ensure gesture output feedback was displayed to the participants. At the beginning of each experimental session, the researcher observed participants on different gesture interaction applications' running the custom software. Overall gesture characteristics were grouped into sets of similar tasks. For example, one task set included effects that represented normal use of the hand's gestures: navigating through an application, starting and closing. Another task set involved user jumping and control with gaming application.

3.3 Framework Development

The main objective of this study was to develop a design thinking framework that can promote a methodological approach to designing gesture interactions. The following figure describes the implemented design thinking framework developed by NodeJS programming language.

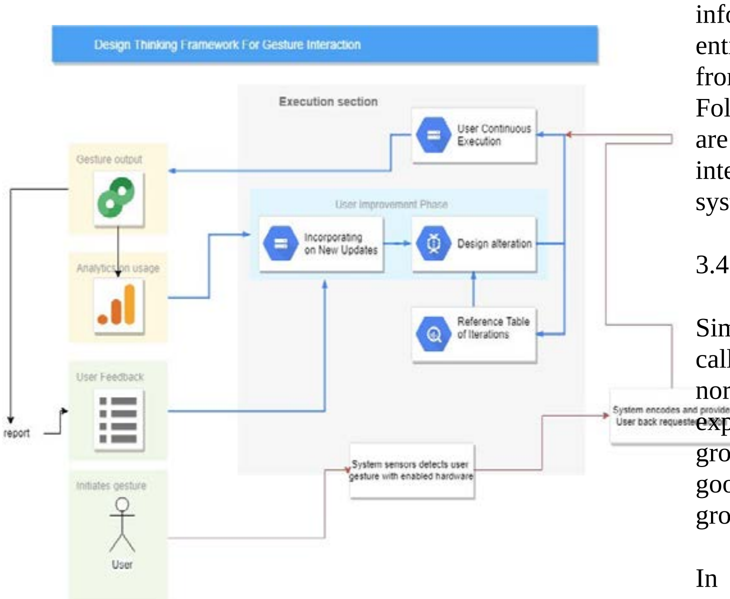

Fig. 3: The Proposed Design Thinking Framework for Gesture Interactions

The design thinking framework has three core entities: gesture output, analytics and feedback which facilitate implementation of critical user gesture input updates and feedback reports displayed in real time to users of the gesture systems to provide alternative gesture inputs. All the three entities share gesture sets identifier in order to consume only gesture having the same identity and ignore the others (without producing errors). This mechanism links together the entire framework to enhance satisfaction for users.

Gesture output entity contains the current state the final states of implementation. When the gesture recognition starts, both the feedback and analytics entities are triggered and the gesture is considered recognized if a gesture output is completed (the final user action that represents has been recognized). The analytics' entity switches to a completed state if the gesture recognition finished correctly or to an error state, if the gesture recognition was not successful and provides an alternative gesture input. The feedback entity optionally updates gesture inputs internal state of the gesture-global state if a gesture output in does not recognize or perform its atomic action, the gesture is reset to its initial state

Each entity in the framework contains an identifier, a point, a type (start, move, end) and a flag representing its state (consumed if its data has been exploited or not consumed otherwise). The entities improve the information provided by the device about user input. Each entity represents an atomic gesture that can be recognized from samples and performs updates on gesture state. Following the lead of the original input language, gestures are efficient to perform, so as to optimize for a maximum interaction performance between user and hardware system.

3.4 Data processing

Simulations were conducted using an online simulator called LYMB including dummy users experience test of normal users with gesture interactions and developer experts. Ten dummy participants were divided into two groups to have two qualitative tests: the first group has 3 good gamers and 4 normal users took part in the second group for the interaction comfort test.

In order to validate, the design thinking framework, repetitive qualitative simulations by LYMB recorded timestamps of the three main core entities of the framework which are real-time gesture output sets, analytics and feedback. The simulations were carried out with dummy users in line with different gesture input set activities including running, jumping, walking and a combination of both.

4. Result's and Evaluation

4.1 Confusion Matrix Results for User Gesture activities

Experimental results from the simulations of general user gesture input activities were presented in a confusion matrix. Gesture input sets from low intense activities provided the best performance displaying a result of up to 90%. Major gesture input actions were a slight body movement action (Hands, waist, legs or entire body). Gesture input sets from high intense user activities presented very low output of 77% clearly showing that high intense user activity does not produce consistent gesture input sets. The following figure 4 displays simulation experimental results for the design thinking framework of gesture interaction.

GSJ: Volume 7, Issue 3, March 2019
ISSN 2320-9186

	Sitting	Standing	Jumping	Walking	Running	Accuracy
Sitting	85%	0	0	0	0	
Standing	0	90%	0	0	0	
Jumping	0	0	82%	0	0	
Walking	0	0	0	77%	0	
Running	0	0	0	0	83%	85%

Fig 4 experimental results for the design thinking framework

4.2 Framework Core Entities Results

4.2.1 Output Entity

Gesture output entity contains the current state the final states of implementation. When the gesture recognition starts, both the feedback and analytics entities are triggered and the gesture is considered recognized if a gesture output is completed (the final user action that represents has been recognized). Gesture output can be used with any gesture sign made from single user command. This allows defining gesture output sets based on user commands either using hands, legs or full body motion, which can be optimized for the framework. Output entity results are shown as follows in figure 4.1

Gesture Interactions	Output Tests
User hands and legs	Gesture output was very slow to all users
User full body Motions	Gesture output was accurate to all users
User hands alone	Gesture output was relatively slow or fast
User legs alone	Gesture output was relatively slow or fast
User head	Gesture output was accurate to all users

Fig 4.1 Gesture Output Results

4.2.2 Analytics Entity

The analytics entity accounts for continuous recognition of both positive and negative users' responses from gesture output during a time period. In the tests performed, the analytics successfully achieved positive responses of 95% while the user was standing, jumping and running interaction states in all of the test cases. Gesture interaction state while user was seated showed minimal analytics due to delay of input sets coordination. The delay introduced by input set conditions is perfectly negligible for the response of the design thinking framework.

Gesture Interaction state	Recognized Analytics Behavior	Effect on the Framework
Standing	95%	Detailed gesture movement analysis
Seated	10%	Minimal gesture analysis
Walking	65%	Moderate gesture movement analysis
Jumping	95%	Detailed gesture movement analysis
Running	95%	Detailed gesture analysis

Fig 4.2 Analytics Entity Results

4.2.3 Feedback Entity

When an intermediate output completes, the gesture-defined output entity connections are exploited in order to add elements to the feedback and analytics section. If a gesture output in the final entity does not recognize or perform its atomic action, the gesture is reset to its initial state. The information contained in the feedback entity optionally updates gesture inputs internal state and/or the gesture-global state.

When the feedback entity consumes a sample motion from user gesture input, it display results which can predetermine alternative gesture input set. User alertness displayed 75% which is a good result since the gesture input variables are dummy characters. Alertness and excitement received major score showing easiness of updating alternative gesture input sets.

Different User Emotions	Feedback Tests
Alertness in execution	75% to all users
Excitement during execution	65.5% to all users
Sadness during execution	13% to all users
Anxiety during execution	35% to all users
Anger during execution	15% to all users

Fig 4.3 Output Entity Results

4.3 Evaluation Results

An evaluation for test users with the design thinking framework for gesture interaction outlined factors consideration of average gesture input sets, degree of freedom and significance of gesture interaction based on different scenarios. The evaluation results were recorded five times both simulated with similar user input gestures sets without the framework and with the framework. The average user time turn around reduced after the framework by a half of the initial time showing ease to use by the users. The user's average gesture input sets increased up to 75% with average execution time reducing by 82% with the interaction. Overall significance of the framework was a success of 88% as shown in table below.

	Average User Time	Average Gesture Input Sets	Average Execution time	Frequency	Significance
Eva 1 Before framework With Framework	16.333 10.55	2 5	3.166 2.100	.9.05	.493
Eva2 Before framework With Framework	16.333 7.000	2 5	8.111 2.100	1.357	.380
Eva3 Before framework With Framework	14.333 5.555	2 5	7.111 2.100	1.444	.364
Eva4 Before framework With Framework	16.333 8.444	2 5	8.111 2.100	1.357	.380
Eva5 Before framework With Framework	13.22 6.222	2 5	6.555 2.100	2.385	.240

Figure 4.4: Evaluation results

4. Conclusions

The solution modelled, implemented and tested in a prototype design thinking framework that can provide a methodological approach for gesture interaction design.

The design thinking framework for gesture interaction does not address the very specific need of design thinking experts and developers, but can help them pointing out to more specific guidelines (addition of three core entities – Analytics, Feedback and output) from the developed design thinking framework, whenever there exists need for a specific type of technological approach for gesture recognition. This study would be valuable if design thinking experts and system developers worked together to produce a unified approach to designing gesture input interaction for users from the developed framework.

Although the proposed design thinking framework for gesture interaction made changes to a number of user activities, the study did not go further to combine another low-level user's characteristics. It would be interesting to extend the low-level user characteristics guidelines and success criteria to cover higher levels of gesture interaction usability.

Acknowledgments

I would like to thank Sharon Wangari from the Nairobi Design Institute (NDI) for her in-depth works on various design thinking concepts and insights which motivated my interest in the field with the extra materials she provided.
Credit to my supervisors Dr. Agnes Mindila and Prof. Cheruiyot. Kipruto for their guidance and support in assisting me with the entire research. They have been with me through high and low moments I experienced during research. I am very grateful to them and may God bless you.
Special acknowledgement to Hall Kirkham, who gave everything up when I started this degree and has never left my side, if only in spirit. Finally, to my parents; late Prof. Elphas and Juliet for their support and encouragement throughout my studies.

References

[1] S. Muser, "Gestures in Human-Computer-Interaction," 2015.

[2] R. P. Sharma and G. K. Verma, "Human Computer Interaction using Hand Gesture," *Procedia Comput. Sci.*, vol. 54, pp. 721–727, 2015.

[3] P. Dalka and A. Czyzewski, "Human-computer interface based on visual lip movement and gesture recognition," vol. 7, no. 3, pp. 124–139, 2010.

[4] A. G. Butler and M. A. Roberto, "When Cognition Interferes with Innovation: Overcoming Cognitive Obstacles to Design Thinking: Design thinking can fail when cognitive obstacles interfere; appropriate cognitive countermeasures can help disarm the traps.," *Res. Technol. Manag.*, vol. 61, no. 4, pp. 45–51, 2018.

[5] R. Razzouk and V. Shute, "What Is Design Thinking and Why Is It Important?," *Rev. Educ. Res.*, vol. 82, no. 3, pp. 330–348, 2012.

[6] J. Zimmerman, J. Forlizzi, and S. Evenson, "Research Through Design as a Method for Interaction Design Research in HCI design research in HCI," 2007.

[7] C. L. Owen, "Design Thinking : Driving Innovation," *BPM Strateg. Mag.*, pp. 1–5, 2006.

[8] C. L. Nehaniv, "Classifying types of gesture and inferring intent," *Companions Hard Probl. Open Challenges Robot. Interact.*, p. 74, 2005.

[9] K. Murphy, "Building Meaning in Interaction: Rethinging Gesture Classifications," *Crossroads Lang. Interact. Cult. 2003 UC Regents CA Vol 5. p. 27-47 2005*, vol. 5, no. 1972, pp. 1–10, 2005.

[10] C. S. Ingulkar and A. N. Gaikwad, "Hand Data Glove: A wearable real time device for human computer Interaction," *Int. J. Sci. Eng.*, vol. 1, no. 2, pp. 99–104, 2013.

[11] T. Winograd, "From Computing Machinery to Interaction Design," pp. 1–10, 1997.

[12] J. M. Palacios, C. Sagués, E. Montijano, and S. Llorente, "Human-computer interaction based on hand gestures using RGB-D sensors," *Sensors (Switzerland)*, vol. 13, no. 9, pp. 11842–11860, 2013.

[13] Y. Visell and J. Cooperstock, "Modeling and Continuous Sonification of Affordances for Gesture-Based Interfaces," *Proc. ICAD 2007*, no. January 2007, pp. 423–429, 2007.

[14] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "User-Defined Gestures for Surface Computing," 2009.

[15] J. Ruiz, "Motiongestures-Chi2011.Pdf," 2011.

[16] R. Razzouk and V. Shute, "What Is Design Thinking and Why Is It Important?," *Rev. Educ. Res.*, vol. 82, no. 3, pp. 330–348, 2012.

[17] Plattner, "Design thinking," p. 240, 2009.

[18] P. Baert, "the Role of Design Thinking Making People Want Things People," 2015.

[19] K. D. Elsbach and I. Stigliani, "Design Thinking and Organizational Culture: A Review and Framework for Future Research," *J. Manage.*, vol. 44, no. 6, pp. 2274–2306, 2018.

[20] N. O. T. For and C. Distribution, "Simple Random Sampling 3.1," 2008.

Kevin Wambani is a passionate tech developer with experience leading in Information Security, technology consultancy and Design Thinking. He is undertaking Masters in Information technology from Jomo Kenyatta University of Agriculture and Technology. He holds a Bachelors of Business Information technology from JKUAT (2014). He is a Microsoft Certified Professional. He is currently a Teaching Assistant at Masinde Muliro University of Science and Technology.

Prof. Wilson Cheruiyot has acquired the following degrees: Bachelor of Science in Mathematics and computer science (1994), Masters of Science in Computer Application Technology (2002) and PhD in Computer Science Applications Technology (2012). He is also certified with Microsoft association in the following: Microsoft Certified Database Administrator (MCDBA) and Microsoft Certified Professional (MCP). Prof. Cheruiyot is currently a Senior Lecturer and Deputy

Director (Postgraduate Programs) at the Jomo Kenyatta University of Agriculture and Technology. Previous, he worked as an auditor with the Kenya National Audit Office (KENAO) (1994 - 2003) and the Teachers Service Commission of Kenya (1994 – 1997).

Dr. Agnes Mindila is a Lecturer and ICT for Development researcher at the Department of Computing and COD at the Department of Information Technology at JKUAT. She holds a BSc. Electrical and Electronics Engineering (JKUAT), MSc. Management of Information Technology (University of Sunderland, UK) and PhD Information Technology (JKUAT). Her research interests are Wireless Sensor Networks, Environmental safety Monitoring, and Biocomputing. Her best research award was Blockchain Technology-Based Computer Application for Vaccine Supply Chain Management. grant, an initiative funded by the Bill & Melinda Gates Foundation, to foster groundbreaking research in global health and development.