International Conference of Science, Engineering & Environmental Technology (ICONSEET), 2(28): 212-220, 2017
ISSN 0794-9650
www.repcomseet.com

Software Usability Design and Its Implications for Graphical User Interfaces

LONGE O. L., HAMEED M.A., ADEKUNLE A. U.
Department of Computer Science, Federal Polytechnic, Ede, Osun Stat, Nigeria.
longelawrence@gmail.com

Abstract- This paper deals with software usability design for graphical user interfaces (A case study of Stock Control System). Microsoft Visual Basic was used for modeling the interface designs and Visual dBASE was used for the implementation. The interaction style used was menu, and the menu items were grouped to reflect their functions. Usability engineering and user-centered design methodologies were adopted. End-user usability evaluation was conducted on the interface design prototypes. The result shows that meaningful menu command names increase GUI usability. Also, the more relevant the GUI menu commands are to the users' tasks domain, the more user-friendly the GUI. In view of this, it is advisable to coin the menu command names from the users' tasks domain. This is very important since the user-program interaction takes place exclusively through the application interface. Human-factors based design and usability considerations are essential factors in the design of usable application interfaces.

Keywords: GUI, Human-factor* based design*, Interaction style, Usability engineering, User-centered design, Tasks domain.

1. Introduction

Usability is considered an important aspect of a usable software system. Recently, software system usability has made some exciting advances, with more and more establishments starting to take usability seriously. Usability is an aspect of design which is studied in the research field called human computer interaction (HCI) or human factors or ergonomics (Gerard, 2015). This area of research is aimed at building a body of knowledge regarding interactions between humans and their environments (including software products) and methodologies for analysing and designing systems. Poor usability and inefficient design of the end-products are common causes for failed software products (Joseph, 2012). One approach to user-centered design has been the introduction of explicit usability engineering goals into the design process (Laura et al, 2012).

The study of usability design is becoming increasingly important for further development in research and application of graphical user interfaces. The Graphical User Interface (GUI) has gained massive popularity since Apple introduced the first mass-market system with this kind of UI in the 1980s. Many users do not distinct between a system and the accompanied interface since the sophisticated logic that allows the application to do its purpose cannot be seen. A fundamental reality of application development is that the user interface is the system to the users (Raffael, 2015). In computing, a graphical user interface (GUI) is a type of user interface that allows users to interact with electronic devices with images rather than text commands (Fourcan and Utpal, 2014).

GUIs can be used in computers, hand-held devices such as MP3 players, portable media players or gaming devices, household appliances and office equipment (Kuo-Ying, 2009). A GUI represents the information and actions available to a user through graphical icons and visual indicators such as secondary notation, as opposed to text-based interfaces, typed command labels or text navigation (Raffael, 2015). A GUI uses a combination of technologies and devices to provide a platform that the user can interact with, for the tasks of gathering and producing information (Longe et al., 2009).

2. Motivation for the study

Usability issues are important in every context of use, regardless of the area of application. Unfortunately, systems development projects seem to be guided by organizational and technical details instead of usability concerns (Lizano, 2014). Many interface developers, who have no formal training in performing usability evaluations, have difficulty articulating the usability attributes most relevant to the particular user interface they are developing. Many products that require users to interact with them to carry out their tasks (e.g., buying a book online from the web) have not necessarily been designed with the users in mind. Typically, they have been engineered as systems to perform set functions. While they may work effectively from an engineering perspective, it is often at the expense of how the system will be used by real people. There has been a greater need for interaction designers and usability engineers to develop current and next-generation interactive technologies.

The purpose of this study is to design a user-centred interface model for graphical user interface for software applications (a case study of Stock Control System). The authours show usability goals in interaction design process, examine the salient usability design rules and principles in the development and use of software applications and outline good GUI design principles.

3. Literature Review

3.1. Interaction design

The aim of interaction design is to bringing usability into the design process. In essence, it is about developing interactive products that are easy, effective, and enjoyable to use from the users' perspective (Gerard, 2015). Designing usable interactive products thus requires considering who is going to be using them and where they are going to be used. Another key concern is understanding the kind of activities people are doing when interacting with the products. The appropriateness of different kinds of interfaces and arrangements of input and output devices depends on what kinds of activities need to be supported (Kuo-Ying, 2009). What can a user currently do using software application systems? What are the operations to be supported? Are operations diverse? What kind of interface and interactive devices are available? Are they also diverse? The systems can be used to gather information, design documents, control instruments, design programs, draw building plans, and play games.

The interfaces can be multimedia environments, virtual-reality environments, speech-based environments, personal digital assistants and large displays environment. There are also many ways of designing users' interaction with a system (e.g., via the use of menus, commands, forms, icons, etc.). In addition, more innovative forms of interaction are appearing that comprise physical devices with embedded computational power, such as xpen, interactive toys, smart fridges, and networked clothing.

Multitude of choices and decisions confront designers when developing interactive products. Interaction designers are concerned with how to optimize users' interactions with a system, environment or product. This can be done by intuition and hope for the best. Alternatively, it can be done by applying rules, guidelines, and principles based on understanding of the users. This involves:

(a) taking into cognisance what people are good and bad at
(b) considering what might assist people with their current way of doing things
(c) thoughtful of what might provide quality user experiences
(d) getting people involved in the design and listening attentively to what they want
(e) using "tried and tested" user-based methods during the design process

3.2. The makeup of interaction design

It has always been acknowledged that for interaction design to succeed many disciplines need to be involved. The importance of understanding users has led people from a variety of disciplines, such as psychologists and sociologists, to become involved (Joseph, 2012). Also, the growing importance of understanding how to design user-centered devices and end-products has led to different practitioners becoming involved. These include graphic designers, programmers, artists, animators, photographers, system designers, and product designers. However, the down side is the costs involved. The more people that are with different backgrounds in a design team, the more difficult it can be to communicate and progress toward the designs being generated (Gerard, 2015).

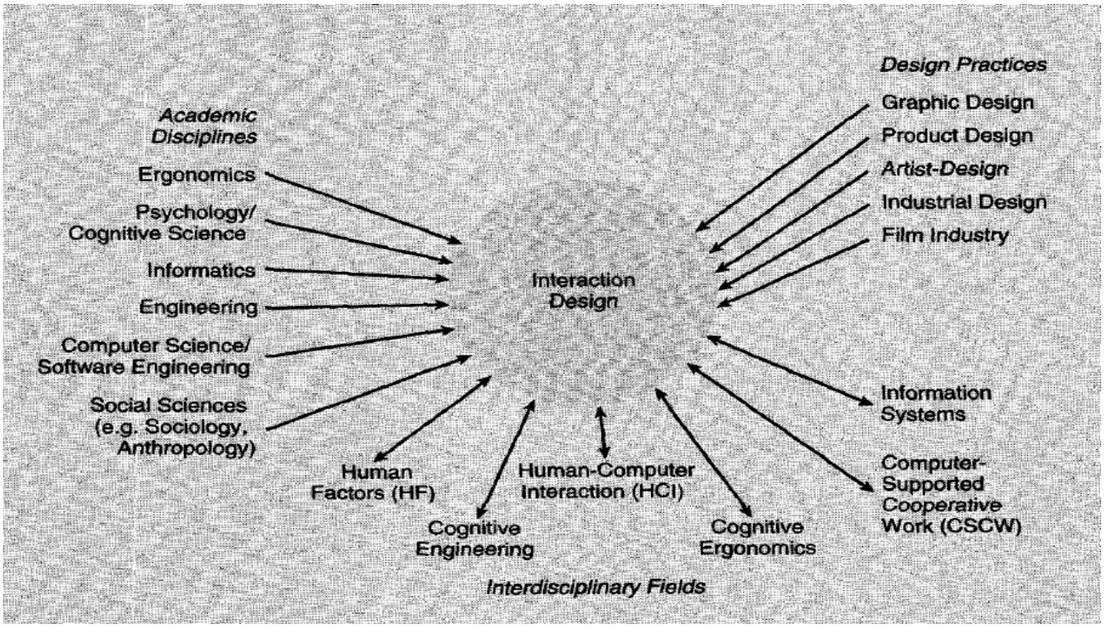

Fig. 1: Academic disciplines, design practices, and interdisciplinary field concerned with interaction design
Source: Preece et al., 2002

34.3. The process of interaction design

Essentially, the process of interaction design involves four basic activities:

(a) identifying needs and establishing requirements
(b) developing alternative designs that meet those requirements
(c) building interactive versions of the designs so that they can be communicated and assessed
(e) evaluating what is being built throughout the process

These activities are intended to inform one another and to be repeated (Shah, 2011). For instance, measuring the usability of what has been built in terms of whether it is easy to learn, provides feedback that certain changes must be made or that certain requirements have not yet been met. Evaluation of what has been built is very much at the heart of interaction design (Soohyung et al., 2011). Evaluation focuses on ensuring that the product is usable. This is usually addressed through a user-centered approach to design, which seeks to involve users throughout the design process.

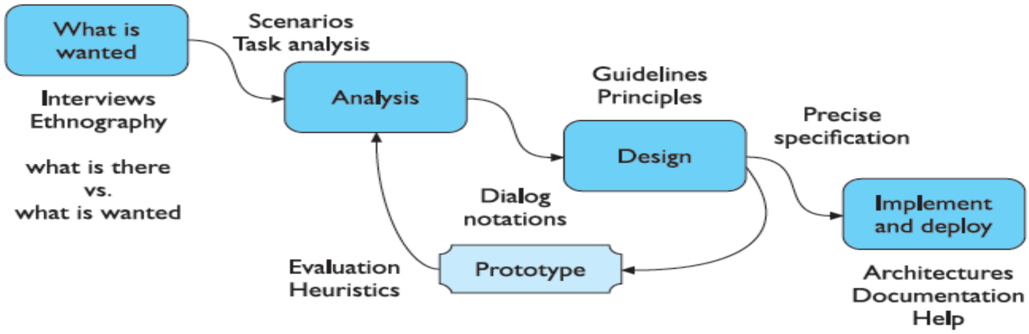

Fig. 2: Interaction design process source: Alan et al., 2004

34.4. Usability goals

According to ISO 9241, Part 11, usability is "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." This definition ties a system's usability to specific conditions, needs, and users (Ahmed et al., 2006). It involves optimizing users' interactions with interactive products to enable them to carry out their activities at work, school, and in their everyday life. More specifically, usability is broken down into the following goals:

(a) effective to use (effectiveness)
(b) efficient to use (efficiency)
(c) easy to learn (learnability)
(d) easy to remember how to use (memorability)
(e) users' subjective impression(Satisfaction)

3.5. Golden rules and principles

One of the central problems that must be solved in a user-centered design process is how to provide designers with the ability to determine the usability consequences of their design decisions. Shneiderman's eight golden rules provide a convenient and succinct summary of the key principles of interface design.

Table 1: Shneiderman's eight golden rules

Rules	Explanation
Strive for consistency	Consistency in action sequences, layout, terminology, command use and so on
Enable frequent users to use shortcuts	Shortcuts, such as abbreviations, special key sequences and macros, to perform regular, familiar actions more quickly
Offer informative feedback for every user action	Action, at a level appropriate to the magnitude of the action
Design dialogs to yield closure	So that users know when they have completed a task
Offer error prevention and simple error handling	So that, ideally, users are prevented from making mistakes and, if they do, they are offered clear and informative instructions to enable them to recover.
Permit easy reversal of actions	In order to relieve anxiety and encourage exploration, since the user knows that he can always return to the previous state
Support internal locus of control	So that the user is in control of the system, which responds to his actions
Reduce short-term memory load	By keeping displays simple, consolidating multiple page displays and providing time for learning action sequences

Source: Alan et al., 2004

Table 2: Norman's seven principles for user-centered design

Principles	Explanation
Use both knowledge in the world and knowledge in the head	Systems should provide the necessary knowledge within the environment and their operation should be transparent to support the user in building an appropriate mental model of what is going on
Simplify the structure of tasks.	There are a number of ways to simplify the structure of tasks. One is to provide mental aids to help the user keep track of stages in a more complex task. Another is to use technology to provide the user with more information about the task and better feedback. A third approach is to automate the task or part of it, as long as this does not detract from the user's experience. The final approach is to change the nature of the task so that it becomes something simpler, provided control is not taken away from the user
Make things visible	The interface should make clear what the system can do and how this is achieved, and should enable the user to see clearly the effect of their actions on the system
Get the mappings right	User intentions should map clearly onto system controls. User actions should map clearly onto system events. So it should be clear what does what and by how much. Controls, sliders and dials should reflect the task, so a small movement has a small effect and a large movement a large effect
Exploit the power of constraints (both natural and artificial)	Constraints are things in the world that make it impossible to do anything but the correct action in the correct way. Here the physical constraints of the design guide the user to complete the task
Design for error	To err is human, so anticipate the errors the user could make and design recovery into the system
When all else fails, standardize	If there are no natural mappings then arbitrary mappings should be standardized so that users only have to learn them once. It is this standardization principle that enables drivers to get into a new car and drive it with very little difficulty

Source: Alan et al., 2004

3.6. Usability design implications for graphical user interfaces

The ultimate test of a product's usability is based on measurements of users' experience with it. Therefore, since a user's direct experience with an interactive system is at the physical interface, focus on the actual user interface is highly desirable. This has cost and design implications for GUIs. The importance of understanding the users has led people from a variety of disciplines to become involved in user-centered designs. However, the down side is the costs involved. The design implication imposes user-centred and proven design principles on GUI designers as shown in Table3:

Table3: Good GUI design principles

GUI design principles	Explanation
Understand People	To understand users fully, developers must first understand people because we all share common characteristics. People learn more easily by recognition than by recall.
Be Careful of Different Perspectives	Designer must use standard icons that can be recognized by the users. The users of the system must have an idea what metaphor the icon was supposed to represent even though it was perfectly intuitive from the designer's perspective.
Design for Clarity	One effective way to increase the clarity of an application is to develop and use a list of reserved words. A common complaint among users is that certain terms are not clear or consistent.
Design for Consistency	Good GUIs use consistent behavior throughout the application and build upon a user's prior knowledge of other successful applications. Each new and exciting experience you

	provide in the software can become an anxiety-inducing experience or an expensive call to your help desk.
Provide Visual Feedback	Users will greatly appreciate knowing how much longer a given operation will take before they can enjoy the fruits of their patience. As a general rule, most users like to have a message dialog box with a progress indicator displayed when operations are going to take longer than seven to ten seconds.
Be Careful With Audible Feedback	Put sound on a few hundred workstations, and a real cacophony emerges in the open-air cubicle environment. However, audible feedback can be useful in cases where you need to warn the user of an impending serious problem, such as one in which proceeding further could cause loss of data or software. Allow users to disable audio feedback, except in cases when an error must be addressed.
Keep Text Clear	Concise wording of text labels, user error messages, and one-line help messages is challenging. Textual feedback can be handled most effectively by assigning these tasks to experienced technical writers.
Provide Traceable Paths	Providing a traceable path is harder than it sounds. It starts with an intuitive menu structure from which to launch your specific features.
Provide Keyboard Support	Using a mouse can become time-consuming and inefficient for the touch typist or frequent users of an application. Keyboard accelerators can provide an efficient way for users to access specific menu items or controls in a window.
Watch the Presentation Model	The look and feel must be consistent. On the basis of users' experiences with one screen or one dialog box, they should have some sense of how to interact with the next screen or control. Identifying the appropriate presentation for the application greatly facilitates the subsequent windows being developed since they will have a common framework in which to reside.
Use Modal vs. Modeless Dialogs Appropriately	Modal dialogs do have many uses in complex applications since most people only work on one window at a time. Try to use modal dialogs when a finite task exists. For tasks with no fixed duration, modeless dialogs are normally the preferable choice with a major caveat.
Use Controls Correctly	Each new control brings with it expected behaviors and characteristics. Choosing the appropriate control for each user task results in higher productivity, lower error rates, and higher overall user satisfaction.

Source: James, n.d.

4. Methodology

The design methodology used is “Agile”, which is Feature Driven Development (FDD). FDD consists of 5 clearly defined processes. The processes are: (a) build an overall model, (b) build a features list, (c) plan by features, (d) design by features, (e) build by features. The processes (that make up FDD) are structured around defining every element of a project as a feature, then designing and building each feature in an iterative manner (Gerard, 2015).). It means design, coding, and testing in an iterative manner until the whole system emerges.

5.1. Data collection technique

A significant percentage of data and background information needed to successfully design this graphical user interface Model were collected from WAPOG Bookshop, Osogbo through various methods which include the following:

(a) **Interview:** In order to obtain relevant information with regards to the policies, procedures, and situations that might not be apparent from documents the interview method was adopted.

(b) **Direct observation:** The activities of the staff carrying out their various tasks were discreetly observed.

(c) **Document analysis:** The relevant materials and literature were read to get valuable information.

(d) Information is also gathered from various web-sites e.g. www.wikipedia.org.
(e) Various textbooks on usability engineering and GUI design were read.

Qualitative data collection methods provide information useful to understand the processes behind observed results.

5. Results and Discussion

5.1. Interface models

The structure diagrams in Fig. 3 and Fig. 4 show the two proposed models 1and 2 for the GUI. They were used to break down broad categories into finer levels of detail. Developing these models helped to move thinking step by step from generalities to specifics. The first level shows the main menus of the system. The second level shows the submenus of each main menus and this is called pull-down menus.

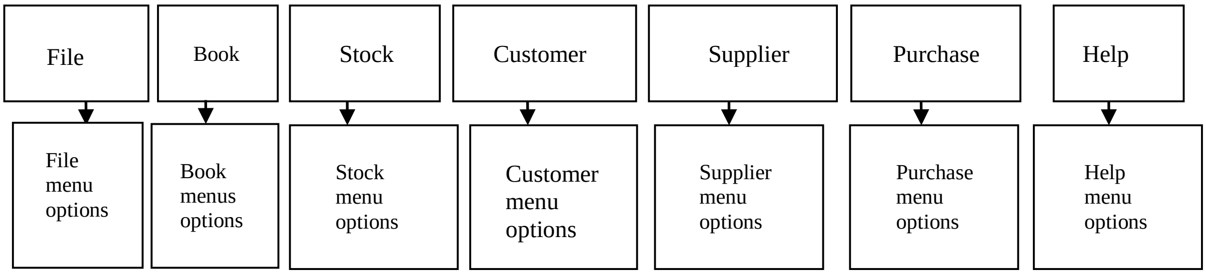

Fig. 3: GUI Model 1

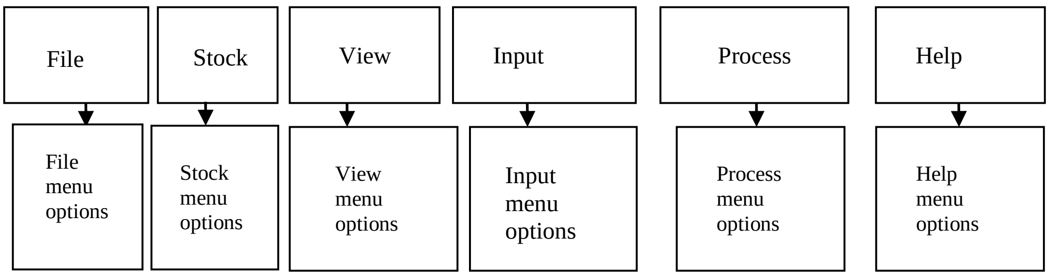

Fig. 4: GUI Model 2

5.2. Interaction styles

The interaction style adopted was menu. This presents a choice of operations or services that can be performed by the system at a given time. Menus in Fig. 5 and Fig. 6 provide information cues in the form of an ordered list of operations that can be scanned. This implies that the names used for the commands in the menus were meaningful and informative. When pointer moved to the position of a menu item, the item was highlighted (by inverse video), indicating that it was the potential candidate for selection. The designs were presented at WAPOG Bookshop for the user acceptance test for the GUIs. All the members of staff at WAPOG Bookshop unanimously accepted the interface design in Fig. 5. This was because the menus of the GUI used command names meaningful to the users and they were coined from the domain of the task. For this research work, Microsoft Visual Basic was used for modeling the interface designs as shown in Fig. 5 and Fig. 6 respectively, and Visual dBASE was used for the implementation.

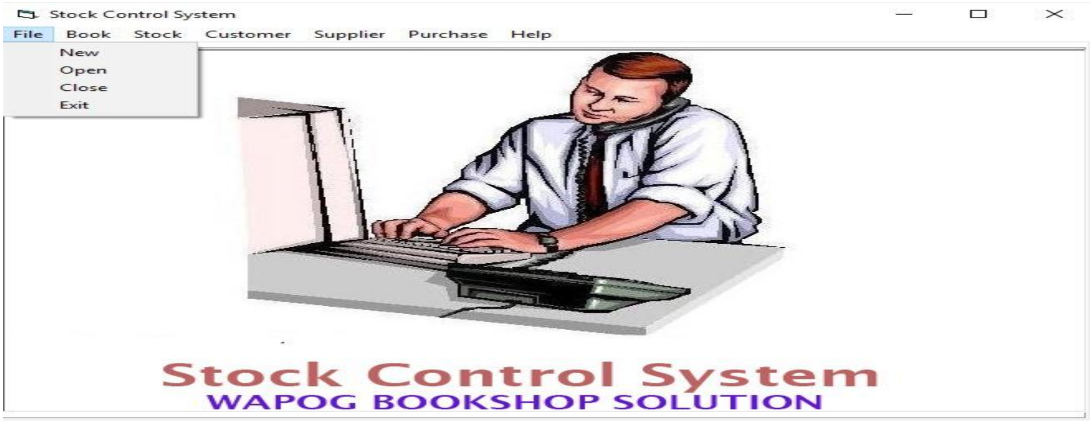

Fig. 5: Screen shot of model 1 for GUI Design

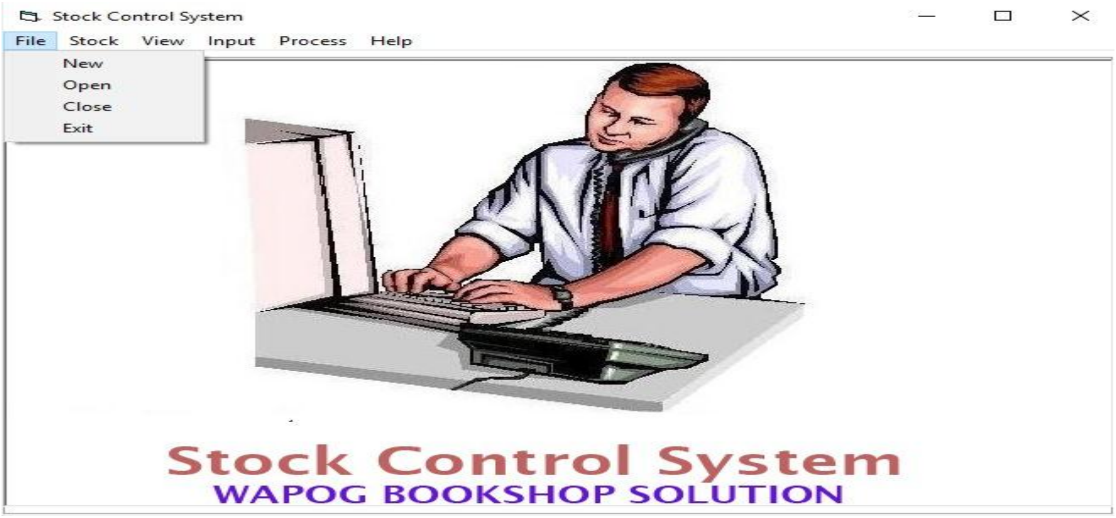

Fig. 6: Screen shot of model 2 for GUI Design

6. Conclusion

Recently, designing user-centered GUIs is a critical skill for application developers, regardless of the GUI platform for which they are designing (Raffael, 2015). Good software usability designs do not happen naturally. They require that the developer learn and apply some basic user-centered design rules and principles for making the design something the user will enjoy working with every day (Fourcan and Utpal, 2014). They also require that the developer get as much experience as possible in working on and being exposed to good GUl designs. Application of user-centered design and good principles of GUI design will definitely offers users the best software products in terms of usability and getting their jobs accomplished.

8. Recommendation

Indigenous analysts and programmers (designers) should employ user-centered and sound proven GUI design principles such as those stated in Tables 1, 2 and 3 for developing software applications in Nigeria. This will enhance application software usability and thereby salvaging national economy.

References

[1] Ahmed S., Mohammad D., Rex B., Harkirat K. (2006). Usability measurement and metrics: A consolidated model. Springer Science+Business Media, Inc, Software Qual J 2006 14: 159–178.

[2] Alan D., Janet F., Gregory D., Russell B. (2004) Human–Computer Interaction, Pears Education Limited Edinburgh Gate Harlow Essex CM20 2JE England: Interaction, 2004, p.836.

[3] Fourcan K. and Utpal K. (2014). Usability Guidelines for Usable User Interface. International Journal of Research in Engineering and Technology 2014 Volume: 03 Issue: 09.

[4] Gerard J. K. (2015). Human–Computer Interaction: Fundamentals and Practice. [Online] Available: http://www.ittoday.info/Excerpts/HCI.pdf. (Accessed on 10th September, 2016).

[5] James H. (n.d.). Principles of GUI Design. [Online]. Available: *http://www.classicsys.com/css06/cfm/articules.cfm*. (Accessed on 21st October, 2011).

[6] Joseph G. (2012). What is Human Centred Design? [Online]. Available: *http://hcdi.brunel.ac.uk/files/What%20is%20Human%20Centred%20Design.pdf*. (Accessed on 4th September, 2017).

[7] Kuo-Ying H. (2009). Challenges in Human-Computer Interaction Design for Mobile Devices. [Online]. Available: *http://www.iaeng.org/publication/WCECS2009/WCECS2009_pp236-241.pdf.* (Accessed on 3rd August, 2017).

[8] Laura C., Ana M., Maria-Isabel S., Ahmed S. (2012). Usability through Software Design. [Online]. Available: *https://www.researchgate.net/profile/Ana_Moreno6/publication/260570003_Usability_through_Software_Design/links/0f3175319b6b841e87000000/Usability-through-Software-Design.pdf.* (Accessed on 3rd August, 2017).

[9] Lizano F. (2014). Integrating Usability Evaluations into the Software Development Process: Concepts for, and Experiences from, Remote Usability Testing. [Online]. Available: *http://repositorio.conicit.go.cr:8080/xmlui/bitstream/handle/123456789/71/T_Final.pdf?sequence=1&isAllowed=y.* (Accessed on 29th August, 2016).

[10] Longe, O.L., Faleye, E.O. Longe, A.K. (2009). Object-Oriented Programming with Visual dBASE. Ibadan: Aseda Publishing, 2009, p.235.

[11] Preece J., Rogers Y., Sharp H. (2002). Interaction Design: beyond human-computer interaction. John Wiley & Sons, Inc., 2002, p.519.

[12] Raffael P. (2015). Design and Implementation of a Graphical User Interface for Elektra. [Online]. Available: *https://www.libelektra.org/ftp/elektra/pancheri2015gui.pdf.* (Accessed on 2nd June, 2016).

[13] Soohyung J, Suyu L, Kun L. A. (2011). Usability Evaluation Model for Academic Library Websites: Efficiency, Effectiveness and Learnability. *Journal of Library and Information Studies2011:p.11-26.*

[14] Shah R. H. (2011). Incorporating Usability Evaluation in Software Development Environments. [Online] Available: *https://www.dis.uniroma1.it/~dottoratoii/media/students/documents/thesis_humayoun.pdf.* (Accessed on 28th December, 2011).