UML-ITS Usability Evaluation of Intelligent Tutoring System

Sehrish Abrejo[1†], Amber Baig[2††], Mutee U Rahman[3†††] and Adnan Asghar Ali[4††††]
sehrish-abrejo@hotmail.com amber.baig@isra.edu.pk muteeurahman@gmail.com adnan_asghar_ali@hotmail.com
[1†,2††,3†††,4††††] *Isra University, Hyderabad, Pakistan*

Abstract
The most effective tutoring method is one-on-one, face-to-face in-person human tutoring. However, due to the limited availability of human tutors, computer-based alternatives have been developed. These software based alternatives are called Intelligent Tutoring Systems (ITS) which are used to tutor students in different domains. Although ITS performance is inferior to that of human teachers, the field is growing and has recently become very popular. User interfaces play key role in usability perspective of ITS. Even though ITS research has advanced, the majority of the work has concentrated on learning sciences while mostly disregarding user interfaces. Because of this, the present ITS includes effective learning modules but a less effective interface design. Usability is one approach to gauge a software's performance, while "ease of use" is one way to assess a software's quality. This paper measures the usability effectiveness of an ITS which is designed to teach Object-Oriented (OO) analysis and design concepts using Unified Modeling Language (UML). Computer Supported Usability Questionnaire (CSUQ) survey was conducted for usability evaluation of UML-ITS. According to participants' responses to the system's usability survey, all responses lie between 1 to 3 scale points which indicate that the participants were satisfied and comfortable with most of the system's interface features.

Keywords:
Intelligent Tutoring System, Usability Evaluation, Computer System Usability Questionnaire.

1. Introduction

The field of intelligent tutoring systems (ITS) sits in the intersection of artificial intelligence (AI) and education [1]. There are several ITS that have been developed, each with a strong theoretical foundation based on research in fields including learning, educational technology, cognitive psychology, artificial intelligence, the ITS's particular field, and much more [2]. It delivers knowledge interactively without the use of books and a traditional environment. The primary focus of ITS is to make students *learn by doing*. The students are given a task or problem within a domain (mathematics, Physics, UML diagrams) where they are provided a workspace to carry out the task. The task performed by students is in the form of a solution which is then evaluated by an intelligent tutoring system. The ITS must be intelligent enough to detect and identify errors made by students. During the problem-solving phase, tutoring systems provide hints to students, through which students get ideas to take further steps. The ITS also can detect the current thinking and intellectual level of students to provide tasks in which they need more attention [3].

A multidisciplinary team of professionals from the fields of software engineering, cognitive science, learning sciences and human-computer interaction (HCI) may be involved in an ITS development project. Each software development project requires the creation of user interfaces since ITS is a program meant to be used by humans. An important feature of any system's development is its usability: to make the system in a way which is used by users easily. Better learning results from a user-friendly system [4]. Although, an ITS is a system but its main focus is on learning science, ignoring the importance of interface design. In literature, there is less evidence that an ITS has been evaluated for its usability. Therefore, this paper attempts to measure users satisfaction by evaluating UML-ITS usability through CSUQ survey. The following section describes some of ITSs followed by UML-ITS along with its interface and hints/feedback design. Evaluation method is discussed before the findings. Conclusions are presented in the end.

2. Literature Review

In literature, there are many ITSs implemented to make students learn different domains. Adaptive Peer Tutoring Assistant [5] is a peer tutoring system where one student tutors another student of the same ability. The students have tutor and tutee roles where the tutee solves the problem and the tutor evaluates and marks the correct or incorrect steps performed by the tutee. If the tutor student makes a mistake, i.e. marks the correct step as wrong, then the system prompts and indicates the mistake the tutor has made.
Circuitously Collaborative Learning Environment, or CirCLE [6], is a system launched for mathematics word problems in order to increase students' metacognitive awareness of the learning process. Using the ITS interface, the students provide feedback using pre-defined feedback buttons.

Wayang Outpost [7] is an Intelligent Electronic Math Tutoring Software that uses multimedia, animated characters, and animated adventures to help middle and

Manuscript received March 5, 2023
Manuscript revised March 20, 2023
https://doi.org/10.22937/IJCSNS.2023.23.3.12

high school students practice solving math word problems in the format of standardized math tests.
Intelligent Tutoring Supported Collaborative Learning, as proposed by [8], is known as ITSCL. Students may study various computer science concepts through a mix of a collaborative learning environment and an intelligent tutoring system. The ITS analyses student responses using natural language processing (NLP) and compares them to the database's answers based on similarity. The system then provides ITS hints on the chosen group response after selecting the answer that most closely resembles the stored database answer as the group answer.

The tutoring systems explained above provide little or no evidence of users' satisfaction with the system features and interaction methods. HCI was emphasized in early ITS literature [9], but later ITS research tended to be more interested in learning, cognitive, and educational elements of ITS and almost completely disregarded usability, HCI, and user needs. Recently, ITS academics have been concentrating on the usability, HCI, and user interface design of ITS [10,11]. An ITS system is made to be utilized by people, just as other software systems. There is less evidence that the present ITSs have solid usability grounds, despite their strengths in the fields of education and learning methodologies. This paper focuses on usability evaluation of an UML-ITS which is implemented to teach Object-Oriented (OO) analysis and design concepts using UML. For the evaluation purpose, CSUQ is used to measure the users satisfaction with UML-ITS.

3. System Description

UML Class Diagrams were selected as domain for ITS design because they are ill-defined and difficult to understand. To introduce Object-Oriented (OO) analysis and design ideas, the suggested UML-ITS is put into practice. It offers an environment where students may work together to create a UML class diagram based on predetermined specifications. Students are expected to be familiar with UML class diagrams.

A session between two students is created by UML-ITS, and the roles of Tutor (someone who gives feedback and evaluates the responses of the other students) and Tutee (one that develops the solution) are assigned to them. The roles of Tutor and Tutee shall be referred to as such for the remainder of the article. The log file stores the record of every activity the student take. By drawing diagrams on their workspace, the students construct a solution model. They also interact with one another via the chat tool. UML-ITS gives students feedback as they work through a specific task. The final answer can be submitted to ITS once both students have agreed upon the modelled solution. The solution is evaluated against a knowledge base-stored sample or ideal solution. A different scenario is displayed if the solution contains mistakes; otherwise, hints are given.

3.1 Design of Hints and Feedback

Feedback from the system is generated when the tutor and the tutee are stuck during problem-solving. The interaction between tutor and tutee starts when the tutee asks for help, the tutor would use chat not only for an explanation but to provide the tutee with feedback on whether the diagram components drawn are correct or not. To maintain the tutor's role of teaching the tutee and to avoid disturbing the communication between tutor and tutee, computer-generated hints are mediated by the peer tutors. To get help, the tutee asks a peer tutor to provide a hint or suggest the next step. UML-ITS hints with peer feedback evaluator are generated in three ways: First, on the tutor's demand/request, when the tutor is not able to find the mistake/error in the solution or is confused about the next step, then by clicking the Hint button, a tutor can receive feedback from the system which provides information about the next step that tutor should suggest to the tutee.

Second, on tutor's actions, when the tutor finds an error in class/attribute/method/relationship or when the tutor tries to suggest a new step in the solution. All feedback messages are evaluated by the system before they are sent to the tutee, and system hints are displayed to only the tutor guiding them to take the correct step. Moreover, text messages are automatically generated from the tutor's correct actions and are sent to the tutee, eliminating the need for the tutor to type the same error in words and then send it to the tutee. Third, when the tutor submits the solution, if no corrections are required, the next problem is displayed to both students; otherwise, hints are generated depending upon the wrong/missing/extra components in the solution. Hints created by a computer are shown in three levels of detail. The first degree of hint tells whether or not the diagram is correct. The next level specifies the type of construct (class, attribute, relationship, etc.) that includes the error, while the final level specifies the precise location of the error. Table 1 shows the number of hints that UML-ITS generates.

Table 1: Hints levels.

General Response			
Can you find out errors in the solution? If you think there is no error in the diagram, you can click the next question button.			
Click the Hints button to receive feedback from UML-ITS if you are unable to find errors.			
If Correct error found by tutor – Well Done, Else Try to find out errors in classes, attributes, and relationships.			
Specific Response			
Classes	**Attributes/ Methods**	**Relationships**	
There are some extra/wrong/ missing classes in the diagram	There are some missing attributes/Methods in the diagram	There are some extra/wrong/missing relationships in the diagram	Level 1
Highlight the extra/wrong/ missing classes, in the solution or noun words in the problem text	Highlight the class with missing attributes/Methods in the solution or problem text	Highlight extra/wrong/missing relationships in the solution or problem text and provide hints like "has a ", "is part of", "is a type of"	Level 2
Create/remove/rename highlighted Class Name = "….."	Create/remove/rename highlighted attribute/method with Name = "…."	Create/remove relationship between Class1-Class2, Name = "relationship type"	Level 3

The system is designed to display hints of one error at a time, gradually giving more specific hints, allowing students to concentrate on one mistake at a time rather than a large number of mistakes in one sentence. The system hints begin by evaluating classes, which means that all errors relating to classes (missing classes, incorrect classes, and extra classes) are shown first. After all classes have been described in the diagram, hints are displayed to find errors in each class's attributes and methods. Finally, hints for relationship errors are displayed. Fig 1 shows the interface of UML-ITS displaying different levels of hints.

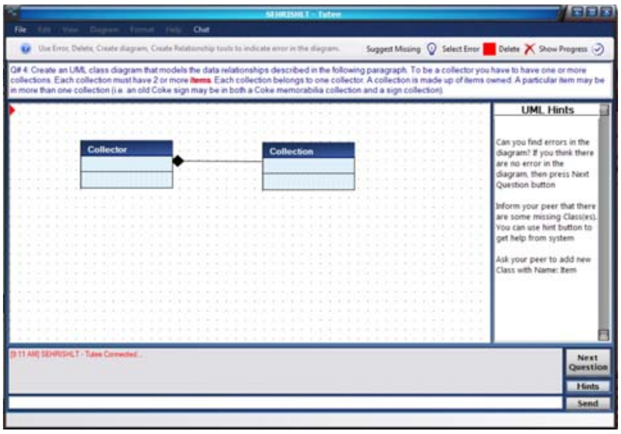

Fig 1: UML-ITS Interface and Hints levels

4. Evaluations

To investigate the effects of UML-ITS on students' learning gains, different assessment studies were carried out, since an ITS' credibility can only be achieved by demonstrating its efficiency in a classroom setting or with typical students.

4.1 Pilot Study

A pilot study as a think-aloud protocol [12,13,14] was conducted, before experimental evaluations, which is a widely used method for computer systems practical evaluation. The main aim of conducting the pilot study was to find out what students perceive about different aspects of the system, particularly, the usability of the interface and the feedback quality that they receive. A total of 30 students participated in the study who were from computer science and software engineering disciplines, at a local University in Hyderabad, and had completed their software engineering course, hence already knew how to model UML class diagrams. The session started with assigning tutor tutee roles randomly to students and it lasted for 60 minutes. Casual conversations during the session and the researcher's findings were used to gather data. Within 30 minutes, most of the students became comfortable with the interface and appreciated the way it was designed. Two groups reported not receiving modeling updates from the tutee, i.e. the changes made in the diagram were not visible on the tutor side. Later, it was found that the issue was due to network connectivity. Most of the students reported that if they click suggest new class button mistakenly, the cancel button was not functioning in the popup window which was corrected in the final version. The feature that the majority of tutor students liked about UML-ITS was automatic message generation. They felt that highlighting and then typing the same mistakes would have taken a lot of their time, while automatic text message generation helped them to concentrate on other errors in the tutee's solution in a shorter amount of time. The automatic text message contains a textual description of the tutee's error, which is sent from tutor to tutee student when the tutor correctly highlights the wrong class diagram component or makes suggestions. Several tutee students found it helpful in terms of localizing the error in their solution when the erroneous component was highlighted in red color along with the tutor's feedback. The majority of the students felt that the UML-ITS feedback messages helped them understand the domain concepts that they found difficult.

In short, the Pilot study helped in identifying the features that needed some improvements and the features that most of the students found useful. From students' interactions, it was observed that the students were satisfied with the interface and appreciated the new features that were added to the UML-ITS design.

4.2 Experimental Design

The UML-ITS was enhanced in the light of pilot study findings before the experimental evaluations. Experimental study was conducted from Computer science and Software Engineering students. UML-ITS was introduced to students by demonstrating its interface along with its features. The students were given 20 minutes to get themselves familiar with the interface of UML-ITS. Students then were randomly assigned roles of tutor and tutee when they logged in to UML-ITS for the first time, the roles were switched whenever they moved to the next problem. The students used UML-ITS for three hours of the laboratory session. According to their roles, students were assigned different seats in the same lab and were only permitted to contact with one another through a chat tool, not in person. After three hours of laboratory experimental sessions, students attempted a post-test along with CSUQ questionnaire to provide their feedback on UML-ITS's interface, its impact on their domain knowledge.

4.3 Usability Evaluation

The objective of usability testing is to see how well users can understand and utilize a product to accomplish their goals. It also relates to the level of satisfaction that users have with that product. The user's perception of UML-ITS was evaluated using Computer System Usability Questionnaire (CSUQ), shown in Fig 2, which is the most widely used usability questionnaire.

	The Computer System Usability Questionnaire Version 3	Strongly agree 1	2	3	4	5	6	Strongly disagree 7	NA
1	Overall, I am satisfied with how easy it is to use this system.	○	○	○	○	○	○	○	○
2	It is simple to use this system.	○	○	○	○	○	○	○	○
3	I am able to complete my work quickly using this system.	○	○	○	○	○	○	○	○
4	I feel comfortable using this system.	○	○	○	○	○	○	○	○
5	It was easy to learn to use this system.	○	○	○	○	○	○	○	○
6	I believe I became productive quickly using this system.	○	○	○	○	○	○	○	○
7	The system gives error messages that clearly tell me how to fix problems.	○	○	○	○	○	○	○	○
8	Whenever I make a mistake using the system, I recover easily and quickly.	○	○	○	○	○	○	○	○
9	The information (such as online help, on-screen messages, and other documentation) provided with this system is clear.	○	○	○	○	○	○	○	○
10	It is easy to find the information I needed.	○	○	○	○	○	○	○	○
11	The information provided with the system is effective in helping me complete my work.	○	○	○	○	○	○	○	○
12	The organization of information on the system screens is clear.	○	○	○	○	○	○	○	○
13	The interface* of this system is pleasant.	○	○	○	○	○	○	○	○
14	I like using the interface of this system.	○	○	○	○	○	○	○	○
15	This system has all the functions and capabilities I expect it to have.	○	○	○	○	○	○	○	○
16	Overall, I am satisfied with this system.	○	○	○	○	○	○	○	○

**The "interface" includes those items that you use to interact with the system. For example, some components of the interface are the keyboard, the mouse, the microphone, and the screens (including their graphics and language).*

Fig 2: CSUQ

CSUQ was developed by James Lewis [15,16] at IBM, which is based on a 7-point scale and uses 16 statements, all positive in tone. Participants rate their agreement or disagreement with each statement, based on a range from strongly agree (1 – best experience) to strongly disagree (7 – worst experience). Furthermore, the CSUQ 16 statements are divided into four subscales which include:

- Overall: the average scores of questions 1 to 16
- System Usefulness (SYSUSE): the average scores of questions 1 to 6
- Information Quality (INFOQUAL): the average scores of questions 7 to 12
- Interface Quality (INTERQUAL): the average scores of questions 13 to 15

5. Results

The increase of students' domain knowledge is the most significant indicator of ITS effectiveness. The details of pretest and posttest along with results and impact of UML-ITS on students' domain learning is discussed in [17]. The main focus of this article is to measure the users satisfaction related to UML-ITS. Table 2 presents the average and standard deviation values of scores for all 16 statements of usability evaluation survey.

Table 2: Average of students' responses

Q.No.	Avg	S.D
1	1.96	0.92
2	2.1	0.9
3	1.82	0.82
4	1.86	0.8
5	1.9	0.94
6	1.92	0.74
7	2.24	1.18
8	2.18	1.28
9	2.34	1.42
10	2.26	1.16
11	2.1	1.27
12	1.92	1.09
13	2.18	0.95
14	2.22	0.92
15	2.18	1.03
16	2.12	0.93

It can be noted that all responses lie between 1 to 3 scale points which indicates that the students were satisfied and comfortable with most of the system's interface features.

5.1 CSUQ Reliability Measures

[18][19] define reliability as a criterion of a measurement's consistency assessed with coefficient alpha. Only positive values of the coefficient alpha may be interpreted, and it can vary from 0 (totally untrustworthy) to 1 (absolutely reliable). Although coefficient alpha is technically a measure of internal consistency, it is the most often used approach for assessing reliability, both for overall questionnaire assessment and for any subscales supported by factor analysis [16]. Fig 3 shows the overall and subscales' scores mean Cronbach's alpha reliability.

Cronbach's alpha is a metric for determining a collection of scale or test items' dependability (or internal consistency). In other words, a measurement's dependability relates to how consistent it is in measuring a concept, and Cronbach's alpha is one technique to assess that consistency. The mean alpha of the overall scale was 0.843, whereas, the coefficient alpha in all cases was more than or equal to 0.80, suggesting adequate scale dependability. Estimations for the CSUQ sub-scales were 0.81 for SYSUSE, 0.80 for INFOQUAL, 0.87 for INTERQUAL, and 0.86 for the OVERALL scale

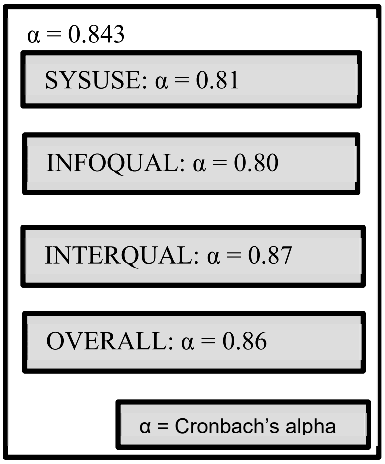

Fig 3: System's reliability measures for CSUQ

5.2 Users Perceived Usability

At this level, the total average scores related to each CSUQ category were evaluated to explore the users' perceived usability of UML-ITS. CSUQ subscales are shown in Table 3.

Table 3: Scores per CSUQ evaluation category

Category	Mean	0-100 scale	Grading Scale
System Usefulness (SYSUSE)	1.9	84.5	A+
Information Quality (INFOQUAL)	2.2	80.1	A-
Interface Quality (INTERQUAL)	2.2	80.1	A-
Overall CSUQ Score	2.1	81.9	A

The average scores estimations for the CSUQ scales were 1.9 for SYSUSE, 2.2 for INFOQUAL, 2.2 for INTERQUAL, and 2.1 for the OVERALL scale. The average scores were then translated to a scale of 0 to 100, with a higher number indicating a better experience. The flowing equations proposed by [16] were used for translation purposes:

$$\textit{CSUQ-SYSUSE} = \mathbf{100 - ((Average(Q1{:}Q6) - 1) * \left(\frac{100}{6}\right))} \quad (1)$$

$$\textit{CSUQ-INFOQUAL} = \mathbf{100 - ((Average(Q7{:}Q12) - 1) * \left(\frac{100}{6}\right))} \quad (2)$$

$$\textit{CSUQ-INTERQUAL} = \mathbf{100 - ((Average(Q13{:}Q15) - 1) * \left(\frac{100}{6}\right))} \quad (3)$$

$$CSUQ\text{-}OVERALL = \mathbf{100} - ((\boldsymbol{Average}(\boldsymbol{Q1}:\boldsymbol{Q16}) - \mathbf{1}) * (\tfrac{\mathbf{100}}{\mathbf{6}})) \quad (4)$$

The next step was to utilize the Sauro & Lewis Curved Grading Scale (CGS) [20] to interpret the computed scores for each category of CSUQ. [21] Normalized the distribution using a logarithmic adjustment on reflected scores, then generated percentile ranks for the whole range of System Usability Score (SUS). These percentile ratings were utilized by [22][23] to develop the curved grading scale (CGS). According to the rule-of-thumb for interpreting CSUQ scores on items, items with CSUQ scores less than 50 are undesirable, items with CSUQ scores between 50 and 69 are questionable, and items with CSUQ scores between 70 and 80 are acceptable and items with CSUQ scores above 80 are marginally acceptable. In this study, the average CSUQ scores of each of its subscales were over 80 (Table 3), indicating that the participants were satisfied with the system's use.

6. Conclusion

In absence of human tutor, the purpose of an ITS is to offer learning possibilities. Despite of the recent advancements in technology and educational ideas, contemporary ITS function better. This has narrowed the difference between ITS and human tutoring, yet ITS usability is overlooked. Performance of an ITS can be increased by creating a more usable ITS. The fields of usability and HCI are well-established and have their own ideas, standards, and methods. Integrating usability consideration into an ITS project is recommended in order to gain better understating of user needs and to improve user learning outcomes.

Considering the importance of usability, UML-ITS was evaluated by students through CSUQ usability evaluation survey. The majority of participants appreciated the features and thought that the hints were useful, according to the pilot study. The participants also showed their satisfaction and appreciated the system's user-friendliness and learning aids, according to their responses to the system's usability survey, which were on average scored between 1 – 3 scale points.

Future usability assessments of ITS can be performed to assess the system's strengths and weaknesses. Such assessments aid users' learning experiences as well as the ITS development process. In the end, it can be concluded that proper and better usability features can improve learning experience of the user.

References

[1] Mousavinasab, E., Zarifsanaiey, N., R. Niakan Kalhori, S., Rakhshan, M., Keikha, L., & Ghazi Saeedi, M. (2021). *Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods*. Interactive Learning Environments, 29(1), 142-163 (2021)

[2] Mark, M. A., & Greer, J. E. (1993). *Evaluation methodologies for intelligent tutoring systems*. Journal of Artificial Intelligence in Education, 4, 129-129 (1993).

[3] Paviotti, G., Rossi, P. G., and Zarka, D. *Intelligent tutoring systems: an overview*. Pensa Multimedia, 1-176 (2012).

[4] Miller, J. R. *The role of human-computer interaction in Intelligent Tutoring Systems*. Foundations of intelligent tutoring systems, 143-189 (1988).

[5] Walker, E., Rummel, N., Walker, S. and Koedinger, K. R., *Noticing Relevant Feedback Improves Learning in An Intelligent Tutoring System for Peer Tutoring*, Proceedings of 11th International Conference on Intelligent Tutoring Systems. Springer, Berlin, Heidelberg, pp. 222 – 232 (2012).

[6] Duangnamol T., Suntisrivarporn B., Supnithi T. and Ikeda M., *Circuitously Collaborative Learning Environment to Enhance Metacognition*, Proceedings of International Conference on Computers in Education, Japan: Asia-Pacific Society for Computers in Education, pp.1-4 (2014).

[7] Arroyo, I., Woolf, B. P., and Shanabrook, D. *Casual collaborations while learning mathematics with an intelligent tutoring system*. In Collaborative Learning Environments Workshop at ITS (2012).

[8] Haq, I., Anwar, A., Basharat, I. and Sultan, K., *Intelligent Tutoring Supported Collaborative Learning (ITSCL): A Hybrid Framework*, International Journal of Advanced Computer Science and Applications, 11(8), pp. 523-535 (2020)

[9] Chughtai, R., Zhang, S., & Craig, S. D. *Usability evaluation of intelligent tutoring system: ITS from a usability perspective*. In Proceedings of the human factors and ergonomics society annual meeting Vol. 59, No. 1, pp. 367-371, (2015).

[10] Lin, H. C. K., Wu, C. H., & Hsueh, Y. P. *The influence of using affective tutoring system in accounting remedial instruction on learning performance and usability*. Computers in Human Behavior, 514–522 (2014).

[11] Koscianski, A., & Zanotto, D. D. C. F. *A Design Model for Educational Multimedia Software*. Creative Education, 5(23), 2003, (2014).

[12] ERICSSON, K. A. and SIMON, H. A, *Protocol Analysis: Verbal Reports as Data*, Cambridge, MA: The MIT Press (1984).

[13] Matarasso, A. K., Rieke, J. D., White, K., Yusufali, M. M., and Daly, J. J. *Combined real-time fMRI and real time fNIRS brain computer interface (BCI). Training of volitional wrist extension after stroke, a case series pilot study*. PloS one, 16(5), e0250431 (2021).
[14] Siyam, N., & Abdallah, S. *A Pilot Study Investigating the Use of Mobile Technology for Coordinating Educational Plans in Inclusive Settings*. Journal of Special Education Technology, 01626434211033581 (2021).
[15] Lewis, J., *Measuring Perceived Usability: The CSUQ, SUS, and UMUX*, International Journal of Human–Computer Interaction, 34(12), pp.1148-1156 (2018).
[16] Lewis, J. R. *Measuring perceived usability: SUS, UMUX, and CSUQ ratings for four everyday products*. International Journal of Human–Computer Interaction, 35(15), 1404-1419 (2019).
[17] Abrejo, S., Kazi, H., Rahman, M. U., Baloch, A., & Baig, A. *Learning from Peer Mistakes: Collaborative UML-Based ITS with Peer Feedback Evaluation*. Computers, 11(3), 30 (2022).
[18] Cortina, J. M, *What is Coefficient Alpha? An Examination of Theory and Applications*, Journal of Applied psychology, 78(1), pp. 98 (1993).
[19] Schmitt, N., *Uses And Abuses of Coefficient Alpha*, Psychological Assessment, 8(4), pp. 350-360 (1996).
[20] Lewis, J. R., Utesch, B. S. and Maher, D. E., *UMUX-LITE—When there's no time for the SUS,* In Proceedings of CHI 2013, Paris, France: Association for Computing Machinery, pp. 2099–2102 (2013).
[21] Sauro, J. and Lewis, J. R., *When designing usability questionnaires, does it hurt to be positive?* Proceedings of CHI 2011 Vancouver, Canada: ACM, pp. 2215–2223 (2011).
[22] Sauro, J. and Lewis, J. R., *Quantifying The User Experience: Practical Statistics for User Research (2nd ed.).* Cambridge, MA: Morgan Kaufmann. (2016).
[23] Lewis, J. R. and Sauro, J., *Revisiting the Factor Structure of the System Usability Scale*, Journal of Usability Studies, 12(4), 183 (2017).

Sehrish Abrejo received her Bachelor's of Computer Science degree from Isra University, Hyderabad. She continued her further studies in the same discipline and received MSCS, M.Phil and PhD degrees from same institute. She is currently working as Assistant Professor in the department of Computer Science at Isra University, Hyderabad, Pakistan. Her research areas are Mobile ad hoc Networks, Artificial Intelligence and Natural Language Processing.

Amber Baig received the BS and MS degrees in Computer Science from IMCS, University of Sindh, Jamshoro and the M.Phil and PhD degrees in Computer Science from DCS, Isra University, Hyderabad. She is currently working as Associate Professor in the Department of Computer Science, Isra University, Hyderabad, Pakistan. Her research interest includes Artificial Intelligence, Natural Language Processing and Human Computer Interaction.

Mutee U Rahman received the B.Sc. and M.Sc. degrees in Computer Science, from University of Sindh in 1997 and 1999, respectively. He received the PhD. In Computer Science degree from Isra Univ. in 2017. He is working as a Professor in Computer Science at Isra University, Hyderabad Pakistan. His research interests include: Natural Language Processing, Computational Linguistics and Artificial Intelligence.

Adnan Asghar is currently pursuing his Ph.D. degree in Information Technology at University of Sindh, Jamshoro, Pakistan. Presently working as Senior Lecturer in the Department of Computer Science at Isra University, Hyderabad, Pakistan. His research interests include: Internet of Things (IoT), Network Science, Technology and Innovation Management