E-Training Course Test Apps as a Media for Measuring Students' Academic Abilities

Siti Fatimah

Universitas Indo Global Mandiri, Palembang, Indonesia

ARTICLE INFO

ABSTRACT

This study introduces an E-Training application developed to overcome limitations in traditional classroom settings where time constraints and delayed material delivery hinder student development. E-Training, utilizing information technology in education, responds to the evolving landscape of technological advancements and the global need for IT-based teaching concepts. The implementation of this system follows the prototype model, emphasizing iterative development stages, enhancing understanding, reducing risks, and ensuring cost-effective solutions. The research outlines the comprehensive design process involving UML diagrams, including use case, activity, sequence, and class diagrams, to establish a versatile educational platform. It details the system's structure involving three key actors: admin, student, and teacher. Moreover, the study elaborates on the testing phases encompassing black box and white box methodologies. Black box tests validate successful functionalities like logins, data input, and storage across user roles. Meanwhile, white box testing focuses on logic verification, ensuring accurate computations and display of student scores and rankings. Overall, this E-Training application emerges as a solution bridging the limitations of traditional classrooms, offering an adaptable environment for learning and teaching. Its systematic development and successful testing signify a substantial stride towards enhancing educational accessibility and effectiveness in the digital age.

1. Introduction

The time available for teachers and students to meet face-to-face in the classroom is very limited. In addition, the process of delivering teaching materials is almost entirely carried out in the classroom, which causes the delivery of teaching materials to be late or even not delivered if the meeting does not take place. This can hamper student development. To overcome this problem, an E-Learning application was created [1]. E-learning is an educational system or concept that utilizes information technology in the teaching and learning process. Along with the

increasingly rapid development of information technology (IT), the need for an IT-based teaching and learning concept and mechanism has become unavoidable, especially in the world of education. This concept, which became known as E-learning, brought about a change or transformation of conventional education into digital form, both in terms of content and systems [2], [3]. The global era requires the world of education to always and always adapt to technological developments in efforts to improve the quality of education, especially in the learning process. Information technology is the development of information systems by combining computer technology with telecommunications.

As an educational institution, MTs Muhammadiyah 1 Palembang plays a role in the development of science. As a consequence of this, MTs Muhammadiyah 1 Palembang must be able to improve the quality and quality of the education it provides. At MTs Muhammadiyah 1 Palembang, one of the teaching and learning processes between students and teachers is carried out by meetings between students and teachers in class. If meetings between teachers and students do not occur then the learning process cannot be carried out. Apart from this, the process of transferring knowledge is mostly carried out in the classroom, which causes the transfer of knowledge to be hampered. E-training allows students to study and practice questions outside of school hours, providing a different atmosphere because learning does not have to be in the classroom, and with one website facility it will be more fun.

2. Method

First, the research method that the author used to achieve the objectives set at the beginning. The system development stage used is the prototype model [4], [5]. The prototype model in software development is a method where a prototype (an early, incomplete version of the final software system) is built, tested, and then refined in multiple iterations. It's particularly useful when the requirements are not well understood or are likely to change. Following are some of the main phases in the prototype model:

1. Understanding Requirements: The development team works closely with the client or end-users to gather initial requirements. These might not be fully detailed or clear initially.
2. Building a Prototype: Based on the gathered requirements, a basic prototype or mock-up of the software is created. This prototype might not have all the features but serves as a visual representation or demonstration of the key functionalities or user interface.
3. Evaluation: The prototype is demonstrated to stakeholders, including clients and end-users. Feedback is collected and analyzed to understand what works and what needs improvement or changes.
4. Refinement: Using the feedback received, the prototype is refined, and necessary changes are made to improve its functionality, design, or features.

5. Iteration: Steps 3 and 4 are repeated in multiple cycles until the prototype meets the requirements and expectations of the stakeholders.
6. Final Development: Once the prototype is approved and refined adequately, it serves as a blueprint for the final software. The development team uses it as a guide to build the actual software, incorporating the improvements and features identified during the prototype iterations.

The main advantages of the prototype model are:
- Enhanced Understanding: It helps in clarifying and refining requirements as stakeholders interact with the prototype.
- Reduced Risk: Early feedback minimizes the chances of building a product that doesn't meet user needs.
- Cost and Time Savings: Identifying issues early avoids costly changes in later stages of development.

The prototype model is characterized by its iterative nature, allowing for continuous refinement based on stakeholder feedback until the final product meets the required standards.

3. Result and Discussion

The author started designing the proposed system by creating a UML diagram [6], [7]. The proposed system design includes Use case diagrams, Activity diagrams, Sequence diagrams, and Class diagrams.

3.1. Use diagram

Several things need to be described, namely actors and use cases. Actors are users who are connected to the system and can be people (indicated by their role and not their name/personnel). The actor is symbolized by the figure of a stick man with a noun at the bottom that states the role/system. Use cases are depicted with an ellipse symbol with the name of the active verb inside which states the activity from the actor's perspective [8], [9]. The system that the author proposes consists of three actors, namely admin, student, and teacher.

3.2. Activity diagram

An activity diagram is a description of function paths in an information system [10]. In full, the activity diagram defines where the system process starts, where it stops, what activities occur during the system process, and what sequence these activities occur in.

3.3. Sequence diagram

Based on the use case that has been created, a sequence diagram is obtained which describes the behavior of objects in the use case by describing the lifetime of the object and the messages sent and received between objects.

3.4. Class diagram

Class diagrams describe the types of objects in the system and the various static relationships that exist between them [11]. Class diagrams show the properties and operations of a class and the boundaries contained in the object relationships.

3.5. System Interface

The author will only show two examples of system interfaces that were built. First, the user will be shown the home page, namely the login page (Figure 1). The login page displays a form for filling in the Username and Password. The username has been determined by the admin for teachers and students.
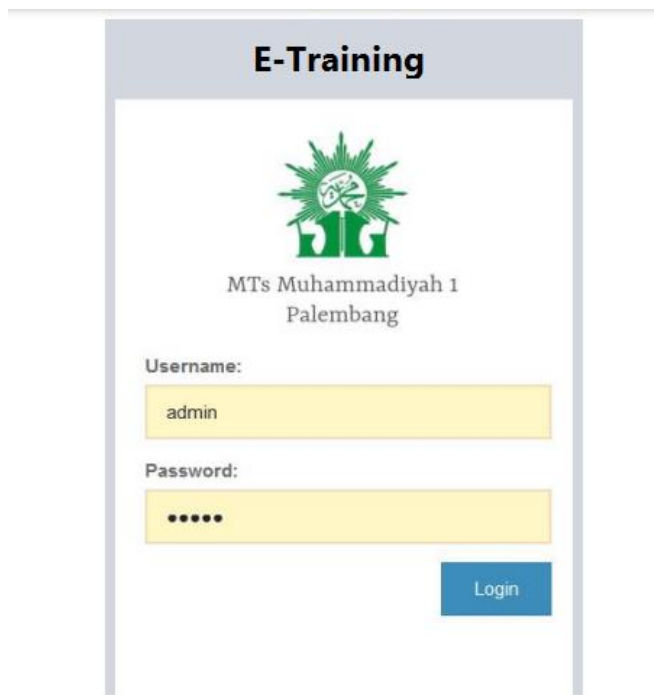

Figure 1 – Login page

Figure 2 – Exam question input page

Figure 2 shows the exam question input page. The question input page is a page that contains questions that will be tested on students. Next, the author carries out testing (black box and white box) on the system that has been built [12], [13].

Black box Testing is testing software in terms of functional specifications without testing the design and program code. Testing is intended to determine whether the functions, input, and output of the software comply with the required specifications. Table 1 shows the results of black box testing.

Table 1 – Blackbox testing

Tested modules	Testing techniques	Results
Login page	Admin, teacher, student input username and password then click login	Successfully log in and enter the main page according to your respective username access
Admin Main Page	After the Admin has successfully entered the username and password, the application's main page will appear containing a menu of teacher data, student data and subject data which functions to add, delete and edit data.	The system displays the main page menu after the admin successfully logs in
Display of the Add Teacher Data form	Admin inputs data on teachers who teach subjects, then saves the data which will be automatically saved in the teacher data master menu.	Successfully input teacher data and the data is saved in the teacher master data.
Display the Add Student Data Form	Admin inputs student data in class 7 of MTs Muhammadiyah 1 Palembang then saves the data which will be automatically saved in the student data master menu.	Successfully input teacher data and the data is saved in the teacher master data.
Display the Add Subject Data Form	Admin inputs existing subject data in the e-training application then saves the data which will be automatically saved in the subject data master menu.	Successfully input teacher data and the data is saved in the subject master data.
Teacher Main Page View	The teacher inputs the username and password. After completing the input, the system will display the main page containing the Material menu, Questions which contain questions and exercises, then Student Grades which contain the results of the scores from the questions that have	The teacher inputs the username and password then the system displays the main page.
	been worked on by the students.	
Upload Material Menu Display	The teacher inputs material data that will be displayed in the student training application, then saves the data and it will be automatically saved in the student application.	Successfully input material data and the data is automatically saved and appears in the student material menu.
Menu Display Upload practice questions	The teacher inputs practice question data and answers to the questions that students must complete and will automatically appear on the student's application system.	Successfully input the practice questions and it will appear in the student's practice questions menu with the final result showing the final score information for the practice questions that the student has completed.
Student Score Menu Display	The student grade menu display will appear on the teacher's menu if the student has finished working on the practice questions given by the teacher.	The score of students' practice questions will appear according to the number of times students have done the practice questions given by the teacher.
Student Ranking Page	After the student has completed the questions and the student's score has appeared, the student's ranking will be displayed starting from the student's score with the highest score.	Displays student rankings starting from the highest student score.
Student Dashboard Page	The student dashboard display containing material and practice questions appears when the student successfully logs in. Students can study the material when the teacher has uploaded the material. And students can work on	Students can view and study the material and practice questions that have been uploaded by the teacher who is teaching and the final score from working on the practice questions

	practice questions when the teacher has uploaded the practice questions.	will appear when the student has finished working on the practice questions and will be automatically saved in the student's score data in the teacher application.
Student Subject Page	Subject pages contain material and questions input by the teacher and can be read, studied and worked on by students.	When students select the material menu, the option appears to download the material so that students can read and study it. Then students can fill in the choice of questions when the teacher has uploaded the practice questions.
Student grades page	The student grades page is a menu that displays the grades from the practice questions that students have done.	When a student selects the value menu, the value results will appear from the practice questions that the student has worked on several times.

White Box Testing is a way to test an application or software by looking at the module to be able to examine and analyze whether the code of the program created is wrong or not. If the module that has been produced has an output that does not match what was expected, the code will be recompiled and checked again until it matches what was expected. (Mustaqbal M. Sidi, et al: 2015). In white box testing, you will be able to find out which logic does not match the code so that errors in programming can be identified (Table 2).

Table 2 – Whitebox testing

Tested module	Test criteria	Results	Criteria
Get question marks	Calculations to get the value of the questions that have been done	Displays the question score	10-100
Get Student Practice scores	Calculations to get grades from student exercises that have been completed	Displaying Values from Student Results Doing Student Exercises	10-100
Get ranking	Ranking calculations are obtained from the results of students' scores working on questions.	Displays the ranking of student scores from the questions in order starting from the highest score	1-100

4. Conclusion

the implementation of an E-Training application, as detailed in this report, offers significant advantages in modern education. The prototype model utilized in the development process, characterized by its iterative nature and stakeholder engagement, ensures enhanced understanding, reduced risk, and cost-effective development. The system's comprehensive design, encompassing various diagrams like use case, activity, sequence, and class diagrams, reflects a meticulous approach to system development.

The testing phase, including black box and white box testing, verified the functionality and logic of the system. Black box testing affirmed successful logins, data input, and data saving across various user roles. White box testing focused on logic verification, ensuring accurate calculation and display of student scores and rankings. Overall, this E-Training application addresses the limitations of traditional classroom settings, offering a versatile platform for students and teachers to engage beyond physical boundaries. Its successful testing and systematic development mark a significant step toward enhancing educational accessibility and effectiveness in the digital era.

REFERENCES

[1] R. E. Matete, A. E. Kimario, and N. P. Behera, "Review on the use of eLearning in teacher education during the coronavirus disease (COVID-19) pandemic in Africa," *Heliyon*, vol. 9, no. 2, p. e13308, 2023, doi: 10.1016/j.heliyon.2023.e13308.

[2] A. H. Obidat, M. Alquraan, and M. H. Obeidat, "Data on factors characterizing the eLearning experience of secondary school teachers and university undergraduate students in Jordan," *Data Br.*, vol. 33, p. 106402, 2020, doi: 10.1016/j.dib.2020.106402.

[3] K. Quimby, N. Spitzer, K. F. Doré, and J. Kawatu, "An eLearning series for staff working in Title X-funded settings: An effort to disseminate national family planning recommendations," *Contraception*, vol. 120, p. 109903, 2023, doi: 10.1016/j.contraception.2022.10.005.

[4] M. Riesener *et al.*, "A model for dependency-oriented prototyping in the agile development of complex technical systems," *Procedia*

CIRP, vol. 84, no. March, pp. 1023–1028, 2019, doi: 10.1016/j.procir.2019.04.196.

[5] S. Hwan Kim, J. Jin, M. Sevinchan, and A. Davies, "How do automated reasoning features impact the usability of a clinical task management system? Development and usability testing of a prototype," *Int. J. Med. Inform.*, vol. 174, no. April, p. 105067, 2023, doi: 10.1016/j.ijmedinf.2023.105067.

[6] G. Bergström *et al.*, "Evaluating the layout quality of UML class diagrams using machine learning," *J. Syst. Softw.*, vol. 192, p. 111413, 2022, doi: 10.1016/j.jss.2022.111413.

[7] H. Wu, "QMaxUSE: A new tool for verifying UML class diagrams and OCL invariants," *Sci. Comput. Program.*, vol. 228, p. 102955, 2023, doi: 10.1016/j.scico.2023.102955.

[8] P. Danenas, T. Skersys, and R. Butleris, "Natural language processing-enhanced extraction of SBVR business vocabularies and business rules from UML use case diagrams," *Data Knowl. Eng.*, vol. 128, no. February, p. 101822, 2020, doi: 10.1016/j.datak.2020.101822.

[9] Meiliana, I. Septian, R. S. Alianto, Daniel, and F. L. Gaol, "Automated Test Case Generation from UML Activity Diagram and Sequence Diagram using Depth First Search Algorithm," *Procedia Comput. Sci.*, vol. 116, pp. 629–637, 2017, doi: 10.1016/j.procs.2017.10.029.

[10] Z. Daw and R. Cleaveland, "Comparing model checkers for timed UML activity diagrams," *Sci. Comput. Program.*, vol. 111, no. P2, pp. 277–299, 2015, doi: 10.1016/j.scico.2015.05.008.

[11] F. Chen, L. Zhang, X. Lian, and N. Niu, "Automatically recognizing the semantic elements from UML class diagram images," *J. Syst. Softw.*, vol. 193, p. 111431, 2022, doi: 10.1016/j.jss.2022.111431.

[12] D. Felicio, J. Simao, and N. Datia, "Rapitest: Continuous black-box testing of restful web apis," *Procedia Comput. Sci.*, vol. 219, no. 2022, pp. 537–545, 2023, doi: 10.1016/j.procs.2023.01.322.

[13] P. L. Fung *et al.*, "Evaluation of white-box versus black-box machine learning models in estimating ambient black carbon concentration," *J. Aerosol Sci.*, vol. 152, no. September 2020, 2021, doi: 10.1016/j.jaerosci.2020.105694.