ECONTECHMOD. AN INTERNATIONAL QUARTERLY JOURNAL – 2019. Vol. 8. No. 2. 69–72

The project of the intellectual information system for student knowledge testing

T. Pytlenko, N. Kunanets, N. Mumchur

Information Systems and Networks Department, s, Lviv Polytechnic National University,
14, S. Bandery Str., Lviv, Ukraine
e-mail: ptaraslen@gmail.com, nek.lviv@gmail.com

Received May 17, 2019: accepted June 19, 2019

Abstract. The intellectual information system for student knowledge assessment designed for owners of mobile devices based on Android operating system. The main purpose of the system is to make the process of student knowledge assessment faster and easier through automation of passing the test. The system also provides an additional opportunity for students to test their knowledge by themselves and receive explanations to the questions.

Key words: education, e-learning, knowledge testing, intellectual systems, information systems.

INTRODUCTION

The topicality of this work is due to the fact that various branches of science are rapidly developing, and there is a growing need for education and knowledge testing [9, 10]. The automation of these processes would be useful, both for students and for tutors. New hardware devices, programming languages, frameworks, libraries, as well as their new versions, appear regularly. At the same time, many of the previously popular technologies, lose their relevance. In order to remain competitive, information technology workers need to learn continuously and efficiently. For efficient and fast learning, it is advisable to use a variety of computer information systems that enable acquirement of the necessary knowledge in interactive or gaming mode. An important component of learning is knowledge assessment which is being done by the tutors and self- knowledge assessment which is being done by students. Knowledge testing in the form of different quizzes types helps to identify and fill gaps in student knowledge, to structure knowledge and repeat the necessary topics in a short period of time. Computer systems that automate the process of creating and conducting knowledge testing help to save a lot of time, effort and increase efficiency [1, 2, 8].

THE CONCEPTUAL MODEL OF THE SYSTEM

When creating a conceptual model, we formulate a content and internal representation that combines the concept of the software product user and the developer of the model. The conceptual model includes in the visual form logic, algorithms, assumptions and constraints. In essence, this is a formal description of the modeling object, which reflects the concept (view) of the researcher on the problem. The main elements of the conceptual model are: the conditions of the object's functioning, determined by the nature of the interaction between the object and its environment, as well as between the elements of the object; purpose of the object research and direction of improvement of its functioning; the ability to control the object, determine the composition of the controlled variables of the object. Performing the development of the intellectual information system for students' knowledge assessment, a set of diagrams was created [5, 6].

A. Use case diagram

Use case diagram (UML) are the diagrams that depicts the relationship between actors and precedents in the system. In our case, there are 4 actors, i.e. the administrator, the user (student), the user (teacher) and the guest (Fig. 1).

The administrator checks the teacher's work and, if necessary, communicates with the registered users and guests of the system. The instructor passes authorization on the system, for this he/she inputs a login and a password, in addition, the teacher can enter a password for the recovery. The teacher fills the database with information – questions, and answers to the tests, as well as information containing explanations to the questions. The instructor also assigns a test, i.e. chooses a test, selects a number of questions, indicates the test setup and receives the result. A student and guest can pass the test and receive information about the result, as well as an explanation of the questions for which they gave the wrong answer.

B. The sequence diagram.

Sequence diagrams show messaging (i.e., method calls) between multiple objects in a particular limited time of the situation. Objects are instances of classes. The main emphasis on sequence diagrams is on the order and time points in which messages are sent to objects.

On sequence diagrams, objects are represented by vertical dashed lines with a name over them. The axis of time also has a vertical direction, it is directed downwards, messages sent from one object to another will be indicated by arrows with the names of operation and parameters. A solid arrow indicates requests and a dotted response to them.

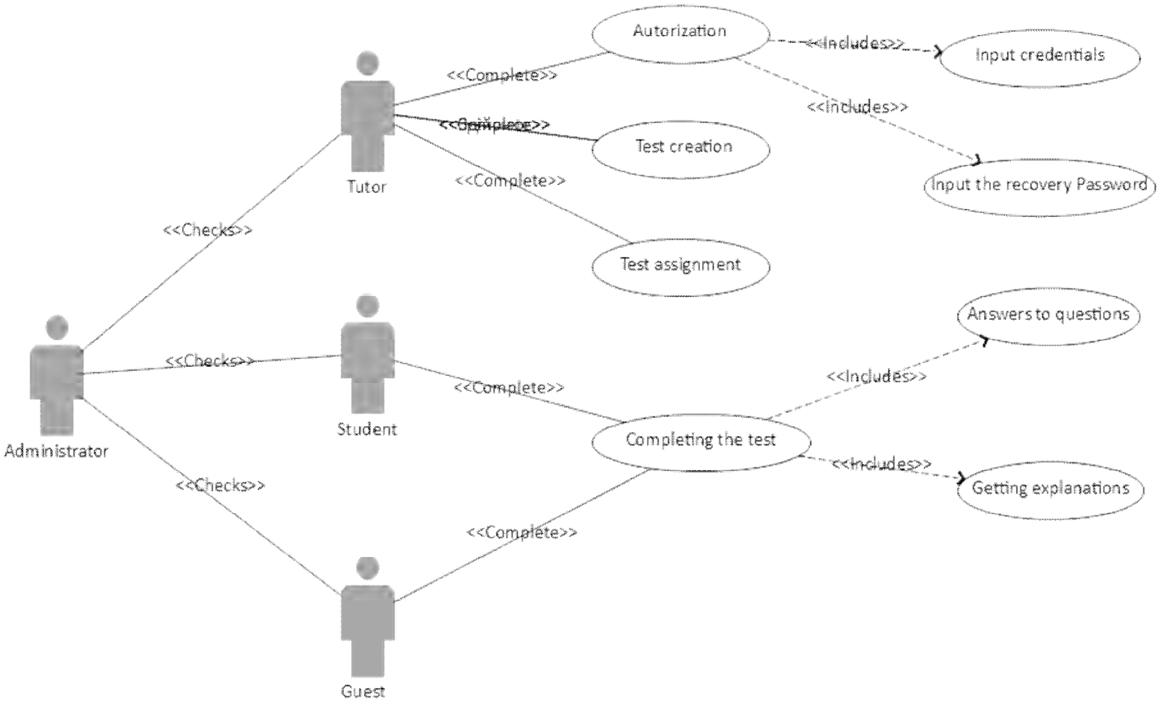

Fig. 1. Use case diagram

Messages can be synchronous – of the usual type of messages when called upon which control is transferred to the called object before the completion of the method, or asynchronous – when called, the control is transmitted back to the direction of the object that made a call. Using asynchronous messages from the side of the called object will show a vertical block that shows the progress of the program.

First, the teacher assigns a knowledge test (quiz), after which the student clicks the "Start test" button, after which a database receives a request for a test question from the system, then the system transforms the data received from the database, the data is displayed on the user's device in the form of questions with answers options; the user answers questions; the system checks the answers for the correctness and generates the final result, which is passed on to the student and teacher (Fig. 2).

C. State Diagram

State diagrams depict various states of an object during its existence and incentives that lead to the transition of an object from one state to another. The state diagram for the intelligent information system for testing students' knowledge is shown in Fig. 3.

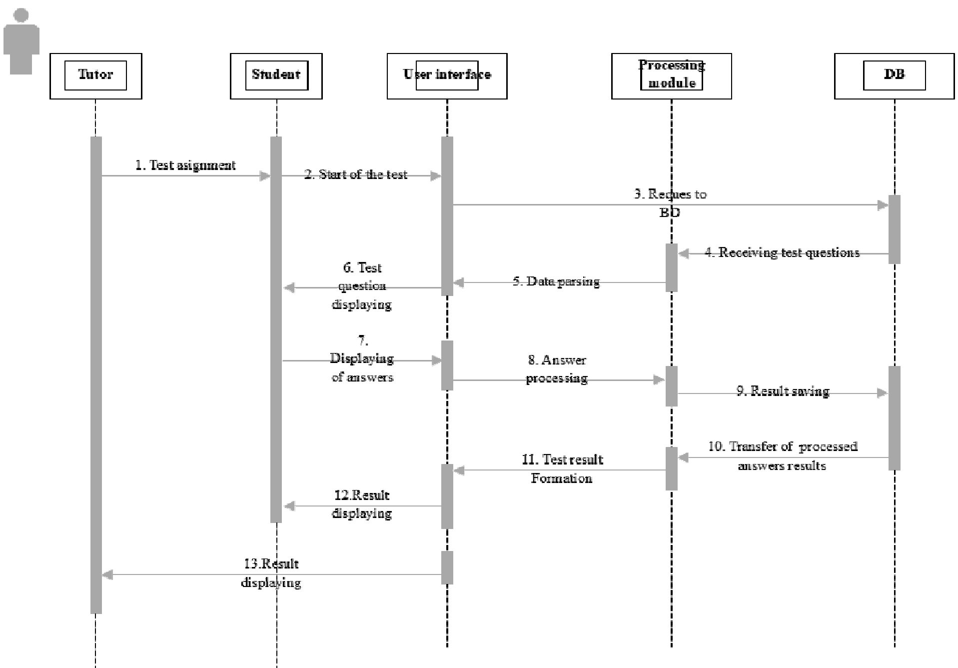

Fig. 2. The sequence diagram

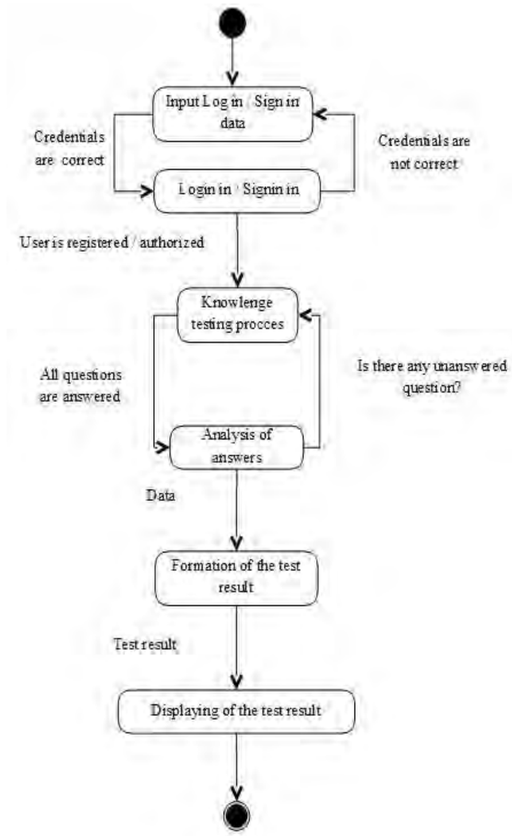

Fig. 3. The state Diagram

In status diagrams, objects are treated as state machines or finite automata that may be in one of the states of a finite state set, and which can change this state through the influence of one of the incentives from a finite set of stimuli. For example, an object of the type Network Server may be in one of the following states for existence: readiness, waiting, stops, and events that may cause a change in the status of an object may be: the creation of an object; the object receives a message "to wait"; the client sends a request for network connection; the client interrupts the request; request completed and interrupted; the object receives a "stop" message. The main blocks of state diagrams are states. The state belongs to only one class and corresponds to the list of attribute values that a class can accept.

In UML, the state describes the internal state of the object of one of the individual classes. There are two special types of states: the beginning and the end. Their feature is that there is no event that can cause the object to return to its original state; in the same way, there is no event that could return the object from the state of the end, only if it reaches it.

D. Deployment Diagram

Deployment Diagram is a UML diagram that displays the computing nodes during the program, components, and objects executed on these nodes, that is, workstations (personal computers, tablets, smartphones) and servers (Web server, database server, and others). The components correspond to the representation of code units working instances.

Fig. 4 shows the nodes that are required to use the system. A student user using his Android device launches the program, disrupts the list of available tests, then the program sends a request to the web server to obtain a list of available tests, the web server sends the request to the database server, the database server replies to the web server, the web server sends a response to the program, the user receives a list of available tests. A teacher-user using his Android device can create a test, assign it to a student, and also see the passage.

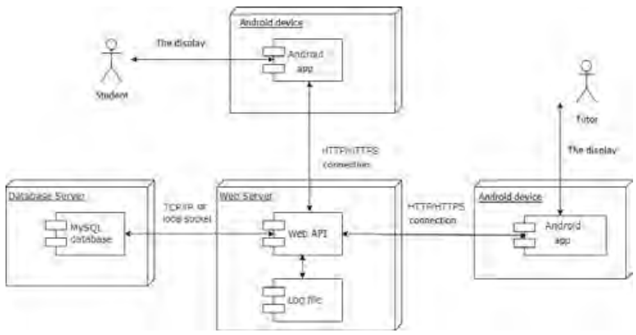

Fig. 4. Deployment Diagram

E. Class Diagram

Class diagrams show the various classes that form the system and their interconnections. Class diagrams are called "static charts" because they show classes along with methods and attributes, as well as the static relationship between them: which classes know about the existence of which classes, and which classes are "part of" others classes – but methods that are called in are not shown. In fig. 5 the classes and links between them for the intelligent student knowledge testing system are shown.

The "Test" class has the following attributes: test name, description, complexity.

The "Test Tasks" class contains the following attributes: question, complexity, several options. The last attribute serves to indicate the presence of several correct answer options.

The "Test Response" class contains the following attributes: option, correct answer.

The Teacher class contains the following attributes: Name, email.

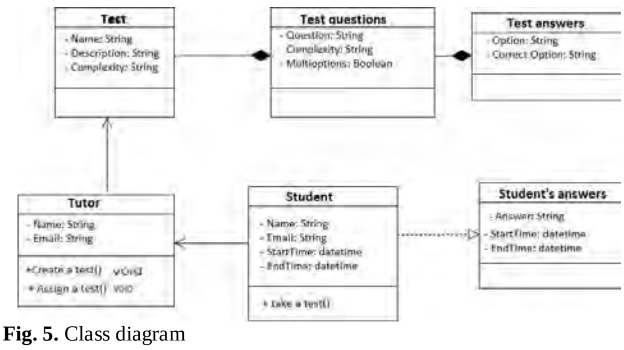

Fig. 5. Class diagram

The "Student" class contains the following attributes: name, email, start time, end time. The last two attributes serve to limit the test time.

The "Student Responses" class contains the following attributes: response, start time, end time.

PROJECT IMPLEMENTATION

A. The system database scheme

The system database scheme of the developed intellectual information system for testing students' knowledge is presented in Fig. 6.

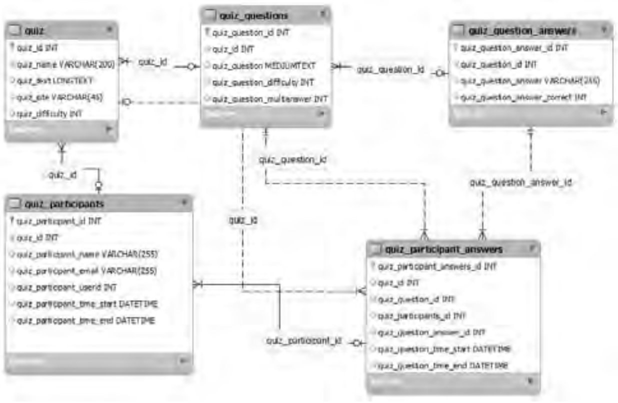

Fig. 6. The system database scheme

Consider the tables of which the database for intelligent information verification system for student knowledge is compiled: quiz; quiz_questions; quiz_question_answers; quiz_participants; quiz_participant_answer; quiz_examiners.

The quiz table contains test information: ID, test name, site name, description, the complexity of the test.

The quiz_questions table contains test questions: question identifier, question text, and information about them: complexity, number of correct answers.

The quiz_question_answers table contains variants of answers, among which is correct.

The quiz_ examiners table contains information about the teacher who creates and assigns a test to the students. This is the teacher's identifier, his name, email.

The quiz_participants table contains information about the users who are in the system and can pass the test. This is the user ID, his name, email, the start time of the test, and the end time.

The quiz_participant_answer table contains information about the user's replies during the test.

B. Description of queries

SQL queries are used to analyze information contained in the system database and its processing. Let's consider some of them:

Display the names and e-mail addresses of all users: (students) registered in the system:

SELECT quiz_participant_name, quiz_participant_email
FROM quiz_participants;

Display the names and complexity of all tests available on the system:

SELECT quiz_name, quiz_difficulty
FROM quiz_participants;

Display the number of tests of different levels of difficulty:

SELECT COUNT(quiz_id), quiz_difficulty
FROM quiz
GROUP BY quiz_difficulty;

Display the names of the items available to the user with the login - student1:

SELECT quiz_participant_name, quiz_name
FROM quiz_participants
INNER JOIN quiz ON quiz_participants.quiz_id = quiz.quiz_id
WHERE quiz_participant_id = 'student1';

Display the names of the teachers and students recorded to them:

SELECT quiz_examiner_name, quiz_participant_name
FROM quiz_examiners
INNER JOIN quiz_participants ON quiz_participants.examiner_id = quiz_ examiners.examiner_id;

CONCLUSIONS

As a result of scientific work, an intellectual information system for checking students' knowledge was developed.

The results of the work are as follows:

– An analysis of literary and other sources has been carried out, which made it possible to identify the shortcomings of known systems and to formulate requirements for a projected intellectual information system for testing students' knowledge.

– A conceptual model of the system for testing students' knowledge in the form of a series of diagrams was designed, namely, a use-case diagram, a sequence diagram, a state diagram, a deployment diagram, and a class diagram.

– Practically implemented student verification system as an application for mobile devices running under the Android operating system.

As a result of the work, an intelligent information system for verifying students' knowledge is ready to be used, which with minor improvement can be practically implemented in educational institutions as an additional means for facilitating the work of teachers and preparing students for classes.

REFERENCES

1. **Finger, G.** Online Learning and Lifelong Learning. Online Education for Lifelong Learning. doi:10.4018/9781599043197.ch003
2. What is eLearning? The Importance & Meaning for Education in 2018. (n.d.). Retrieved from https://www.talentlms.com/elearning/what-is-elearning
3. **Kisak, P. F**. 2015. *Pedagogy:*. United States: Create Space Independant Publishing Platform.
4. **Daniela, L., & Rūdolfa, A.** 2018. Learning Platforms: How to Make the Right Choice. *Didactics of Smart Pedagogy*, 191–209.
5. Basic Notions of System Analysis. (n.d.). System Analysis: Theory and Applications Data and Knowledge in a Changing World, 35–83.
6. **Rond M. D.** 2005. UML. Culemborg: Van Duuren Media.
7. **Saleh, M., & Abel, M.** 2015. Information Systems: Towards a System of Information Systems. In: Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management.
8. **Kamsa I., Elouahbi R., El Khoukhi F. 2018.** STUDY SMART NOT HARD. Turkish Online Journal of Distance Education – TOJDE January 2018, Vol.: 19 Number: 1: 62–74.
9. **Zhezhnych P., Shilinh A. 2018.** Linguistic approaches the planning of educational services in higher education institution. Econtechmod. Vol. 7, No 4: 13–20.
10. **Nardi A. 2018.** Evaluating online learning: a review of studies on e-testing in the university context. Form@re - Open Journal per la formazione in rete. Vol. 18, No 1: 179–191.