Learning Testing Model using test generators and mobile applications

Doru Anastasiu Popescu[a], Cosmin Iulian Gosoiu[b] and Daniel Nijloveanu[c]

[a] *University of Pitesti, Department of Mathematics – Computer Science, Pitesti, Arges, Romania, doru.popescu@upit.ro*

[b] *University of Pitesti, Department of Mathematics – Computer Science, Pitesti, Arges, Romania, cosminiulian1998@icloud.com*

[c] *University of Agronomical Sciences and Veterinary Medicine Bucharest, Faculty of Management and Rural Development, Slatina Branch, Romania, nijloveanu.daniel@managusamv.ro*

Abstract

The knowledge testing method can be done in many ways, being dependent on the content of the course, the characteristics of the student group, the existing technical means for the course, knowledge related to IT. However, testing based on the use of databases with questions and answers selected from several possible choices is a fast, efficient and preferred way for teachers and students. This paper aims to present a test model that uses existing generators and a mobile application created specifically to transmit the test and use the email service which is used for a long time by teachers and students. The mobile application provides the interface for presenting the test and retrieves from the student the answers he sends to the teacher for assessment. The model in this paper has two important stages: generating the test and sending it to the student and retrieving the result through the email service.

Keywords

generator, algorithm, genetic, test, learning, question, answer, mobile, assessment

1. Introduction

Learning is a very important component in people's education. This can be done in different periods of time, with technical means specific to that period. To present a course a teacher can use online platforms, mobile applications, computer networks, communication services such as email, video platforms, social networks or the classic face to face method, a detailed study on this aspect is made in [1] and [4]. Either way, knowledge would be transmitted. Another challenge is assessment. Among the assessment methods is found in a fairly high percentage, the multiple choice test assessment , based on the answer to questions, by choosing the correct answer from several options provided, an analysis of this type of assessment is performed in [2] and [5].

Recent studies, accentuated by the current pandemic caused by the SarsCov 2 virus, highlight researchers' concern to provide teachers with more database-based assessment models that contain questions from the courses they teach.

Models that use such databases should have components that solve at least two problems:

1. How do we generate a test?
2. How do we test students, having prepared the test?

L2D'21: First International Workshop on Enabling Data-Driven Decisions from Learning on the Web, March 12, 2021, Jerusalem, IL

EMAIL: doru.popescu@upit.ro (D. A. Popescu) cosminiulian1998@icloud.com (C. I. Gosoiu) nijloveanu.daniel@managusamv.ro (D. Nijloveanu)

CEUR Workshop Proceedings (CEUR-WS.org)

More or less complex models exist on certain online platforms but they each have their own advantages and disadvantages, according to [3] and [11]. Among the advantages we mention: the use for knowledge transmission and testing of the same platform, standardization of testing, the possibility of transmitting the answers by the student and the return of the result by the teacher, and among the disadvantages we mention: difficult access to platforms by disadvantaged categories of students, teacher's impossibility to intervene in the elaboration of the test, except with certain restrictions given by the platform, the relatively difficult communication with the assessed students, the use of the same platform for all the learning operations.

In this paper we will present an assessment model in which the above two problems are solved differently, in order to obtain a simple, fast and efficient assessment. The test generation will be done separately with a technology based on test generation algorithms that involves only the database with questions and verification/modification of the test by the teacher. This involves implementing a visual application in a programming environment, and the assessment can be done by using a mobile application together with an email service. The methods of generating test are in the attention of e-learning researchers, some of which are presented in [6], [9] and [10]. The novelty brought by this paper is related to the separate implementation of the two components: test generation and test assessment. Using a free email service through a mobile application created with Android Studio in the Java language is another novelty used in sending students' tests and receiving the teacher's response. Specific aspects related to assessment and learning using mobile technologies are presented in [8].

2. Learning Testing Model

The assessment model we will present requires the use of a database containing questions in the format:

Table 1
Attributes of questions in the database

Name of the attribute	Short description	Type	Example
ST	The statement of the question	String	In what sea does the Danube flow?
AS	Answer options	String	a) Red Sea; b) Black Sea; c) Mediterranean Sea
CS	Correct answer	char	b
K	Keywords (domain chapter, section, notions)	String	European Geography

The K attribute can be replaced with the question's degree of difficulty or with the estimated solution time.

Depending on the teacher, from the last 3 specifications the most suitable one can be kept. Thus, depending on this choice, the test generator can be built accordingly. In [6], [10] and [12] are presented different ways of representing the questions used in the test.

Different methods can be used for the generator. If the database has few questions, then the backtracking method can be used, but if the number of questions in the database is large (over 30) genetic algorithms can be used, which have very good results, if they are set with appropriate values for population size and number of generations. Details about genetic algorithms in general can be obtained from [11], and for particular cases related to test generations can be obtained from [5] and [6].

After obtaining the tests from the generator, the teacher can intervene in them, through operations of adding questions, modifying the statements or answers. Then, through a web, mobile or e-learning platform, the teacher can send the students the test and receive their answers, after which he sends them back their results.

For our model, the generator uses genetic algorithms (through an application implemented in Java), and the test transmission operation is performed through a mobile application (Android application). Figure 1 shows the model for assessing learning.

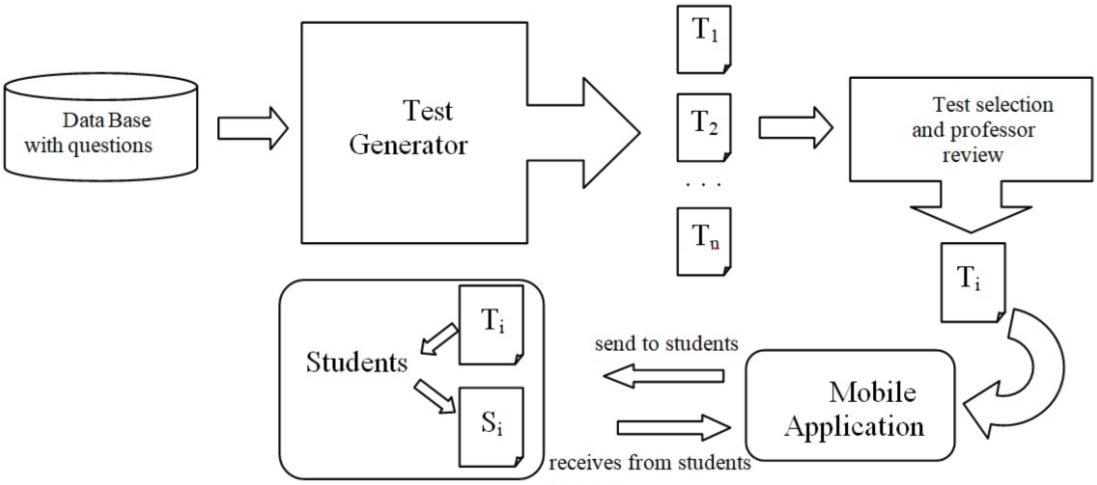


Figure 1: Learning Testing Model

Figure 1 presents de model based on 2 important components: Test Generator, which will be presented in section 3 and Mobile Application which will be presented in section 4.

From the database with questions that have attributes presented in table 1 the test generator selects the questions that will form tests T_1, T_2, …, T_n according with the selected domains by the teacher, when launching the test generator.

From the generated tests the teacher chooses one of them, which can be modified, in the presented model it is marked with T_i then the test is sent to the students through a mobile application. The solution of the test marked in figure 1 with S_i, is sent to the teacher by the student, through the same mobile application and email. The results of the tests are centralized and then sent to the students by the teacher. The mobile phone is a device used in the daily life, by the students as well and that is why we chose it for this testing operation. The implementation of the Mobile Application component was realized by us in Android Studio and is presented in section 4.

From figure 1 it is observed that having a database containing information related to questions, the generator obtains the tests T_1, T_2, …, T_n. Configurations related to the test format: the number of questions, the characteristics of the questions (keywords, degree of difficulty, estimated development time, number of generations, population size) are introduced by the teacher in the generator. From the generated tests the teacher chooses one of them, which he can modify, in the presented model it is marked with T_i, then it is sent to the students through a mobile application.

The solution S_i of the test is sent to the teacher by the student, through the same mobile application and through an email service. The test results are centralized and then transmitted to the students of which the teacher. The mobile phone is a device used on a daily basis, including the students and that is why it was chosen by us for this testing operation.

3. Test generator

For the test generator we can use the variants with genetic algorithms presented in [6], [9] or [13]. To understand how the tests are represented in the algorithm we will mark with Nq the number of questions in the database and with N the number of questions in a test. For these generator variants the test is represented by a chromosome, and a question by a gene. For chromosomes we have two variants of representation:

Option 1:

C = (C_1, C_2, …, C_{Nq}), where the C_i gene is *true*, if the question from the database is part of the test, respectively *false* if the question from is not part of the test, i = 1,2,… , Nq..

Option 2:

C = (C_1, C_2, …, C_N), where the genes C_i, i = 1, 2,…, N are the serial numbers of the questions in the database or the IDs of the questions, unique values at the level of the database with questions which will be part of a test.

The fitness function is associated with a chromosome, being specific to the format of the questions (presented at the beginning of section 2) and the restrictions of the tests imposed by the teacher.

In the generator implemented by us the restrictions refer to the fact that the tests' questions must be from a list of domains selected by the professor, as we can see in figure 3. In this case, the fitness function for a chromosome is the number of questions which are found in the selected domains by the professor. The sorting of the chromosomes from the current populations will be made in descending order by the fitness function.

Papers [5], [12] and [13] present some variants of this function. Thus, for example, the fitness function can be the sum of the degrees of difficulty associated with the questions of the genes in the chromosome and in this case the population (chromosomes of a generation) must be ordered ascending by the value | g - f |, where g is the degree of difficulty chosen by the teacher for the test, f is the degree of difficulty of the test associated with the chromosome (i.e. the value of the fitness function). The degree of difficulty of a test, is the sum of the degrees of difficulty of its questions.

Mutation and crossover operations can be performed by the variants presented in [9], [12] or [13]. The diagram of the genetic algorithm used in the test generator is the one presented in figure 2, also used in [9].

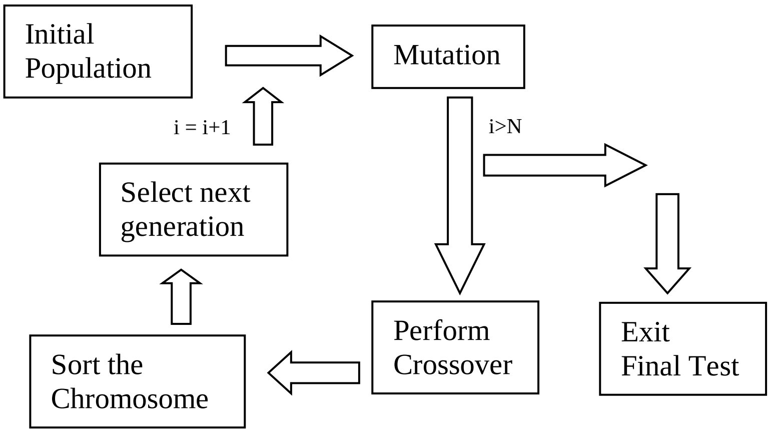
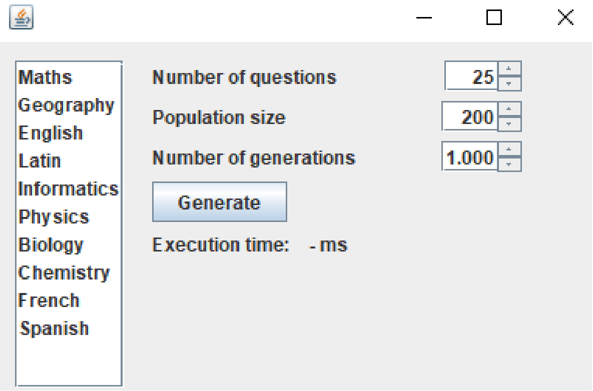


Figure 2: The scheme of the genetic algorithm for Generator Test

In [7] are described in detail the operations used in the genetic algorithm's scheme from figure 2 and algorithm 1. N is the number of generations. In order to obtain tests that contain all questions in the selected domains we have to run the test generator multiple times varying the dimension of the population.

ALGORITHM 1: TestGenerator Algorithm

```
Step 1.  Read the list of selected Keywords from teacher, number of questions and N number of generations
Step 2.  Generation of the initial population
Step 3.  Sort of chromosomes based on fitness function
Step 4.  for i=1 to N execution
                 - Mutation operation
                 - Perform Crossover operation
                 - Sort of the chromosomes use the fitness function
                 - Select the next generation
         endfor
Step 5.  Write the test from the best chromosom from the last populatiation
```



Maths
Geography
English
Latin
Informatics
Physics
Biology
Chemistry
French
Spanish

Number of questions 25
Population size 200
Number of generations 1.000
Generate
Execution time: - ms

Figure 3: Interface Generator Test

By choosing the right settings related to population size (number of chromosomes) and number of generations, tests are obtained that respect at least 90-100% of the test restrictions related to the question chapters, the difficulty of their questions and the answer time to the questions. A rigorous analysis of these results is presented in [5], [6] and [13]. After choosing the test from the variants given by the Test Generator, the teacher analyzes it and possibly can make changes or additions to it.

4. Mobile application for assessment

Once the assessment test is generated, the teacher must pass it on to the students. To solve this operation we created a mobile application implemented in Android Studio using the Java language. The test we use is the grid, each question having 3 answers, only one of them being true. The answers are taken using radio button elements and sent to the teacher by email. The scheme of using the mobile application is presented in figure 4.

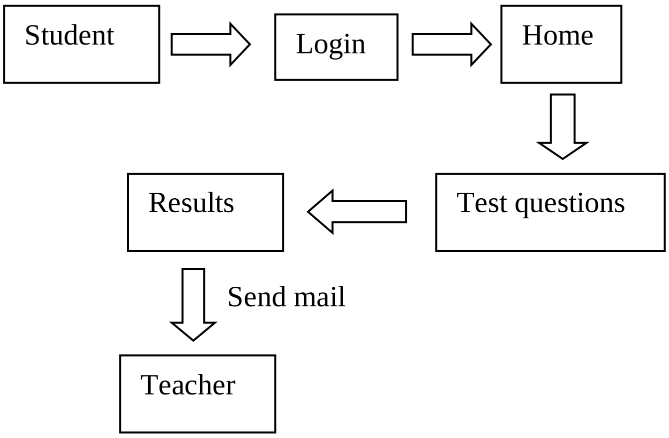
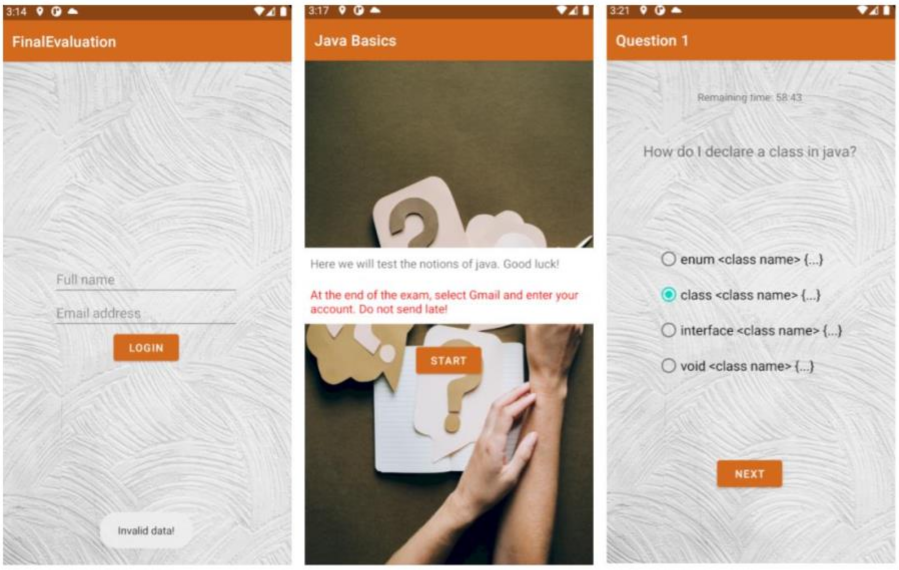


Figure 4: Scheme for using the mobile application

In the login stage the student introduces his/her name and email address, then in the Home section information regarding the test will appear, following that the questions of the test and the answers will be displayed. Throughout the test the remaining time left to finish will be displayed. At the end of the test in the Results section the results will appear. These results are also sent to the professor through email.

The format used in the mobile application for the question file is json, to make it easier to use the question information.

The interface of the application is presented in images 5 and 6. Initially, the name and email address of the student are entered, after which the test is started if the current date and time correspond to those set by the teacher. The questions are developed during the time period set by the teacher. Sending the test result by email must be done within the time allotted to the test.



Invalid data!

Here we will test the notions of java. Good luck!

At the end of the exam, select Gmail and enter your account. Do not send late!

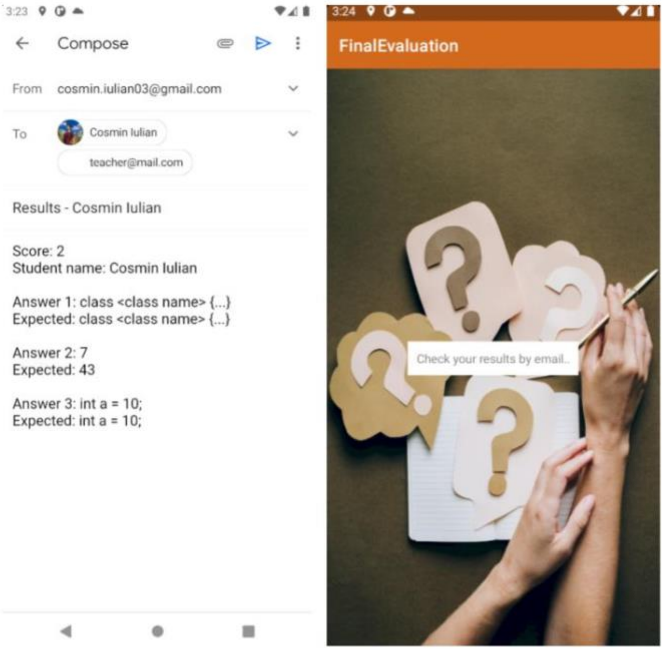
Remaining time: 58:43

How do I declare a class in java?

○ enum <class name> {...}
● class <class name> {...}
○ interface <class name> {...}
○ void <class name> {...}

Figure 5: Interface mobile application for assessment

Using an email service the centralized results obtained by the student are transmitted to the teacher. An image with the application interface is shown in figure 5.



Compose

From cosmin.iulian03@gmail.com

To Cosmin Iulian

teacher@mail.com

Results - Cosmin Iulian

Score: 2
Student name: Cosmin Iulian

Answer 1: class <class name> {...}
Expected: class <class name> {...}

Answer 2: 7
Expected: 43

Answer 3: int a = 10;
Expected: int a = 10;

Figure 6: Transmission of test results through the mobile application

Respecting the teacher's settings regarding the time and date of starting the test, the application worked with very good results for the tests used in the assessment. The application can be easily improved, to use different formats of questions, answers, because the implementation was done in Android Studio using Java. The application implementation mechanisms can also be used for other operating systems used by mobile phones or other gadgets. Another option that can be used for the testing application is the one that uses web applications, with an interface similar to the one on the mobile phone.

5. Conclusions

The student testing model presented in this paper is based on simple and cost-effective means of communication. The effort of students and teachers in terms of technology and technical support is not great. The results obtained so far are very good. It remains that in the next period to use this model on a larger scale and to make a detailed study of the obtained results. An important concern is the integration of the generator and the mobile application in a single application and ensuring an interface suitable for the teacher who performs the assessment and the student who is evaluated.

In the upcoming period we will try to realize a mobile application for other operating systems, as well as a web application that integrates both modules: test generating and students testing.

6. References

[1] Becheru A., Popescu E., Design of a conceptual knowledge extraction framework for a social learning environment based on social network analysis methods, Proceedings ICCC 2017, pp. 177-182(2017).

[2] Boopathiraj, C., Chellamani, K.”Analysis of Test Items on Difficulty Level and Discrimination Index in the Test for Research in Education” International Journal Of Social Sciences & Interdisciplinary Research, pp. 189-193, (2013).

[3] Brenda W. Yang, Juan Razoand Adam M. Persky, Using Testing as a Learning Tool, Am J Pharm Educ. Nov; vol. 83(9): 7274 (2019)

[4] O'NeilH. F.,PerezR. S.. ”Web-Based Learning: Theory, Research, and Practice”, Routledge, pp. 220-230, (2013).

[5] Guang C., Yuxiao D., Wanlin G., Lina Y., Simon S., Qing W., Ying Y., Hongbiao J.. ”A implementation of an automatic examination paper generation system”, Mathematical and Computer Modelling,Volume 51, Issues 11–12, pp. 1339-1342, (2010).

[6] Li Y., Li S., Li X. ”Test Paper Generating Method Based on Genetic Algorithm”, AASRI, Procedia, Volume 1, pp. 549-553, (2012).

[7] Melanie Mitchell, An Introduction to Genetic Algorithms, MIT Press, (1996)

[8] PopescuE., StefanC., IlieS., IvanovicM., ”A Mobile Learning Application for Collaborative Note-Taking in Lecture Settings”, Proceedings ICWL, Lecture Notes in Computer Science, Vol. 10013, Springer, pp. 131-140, (2016).

[9] PopescuD. A., BoldN., DomsaO. ”Generating assessment tests with restrictions using genetic algorithms” ICCA,pp. 696-700, (2016).

[10] PopescuD. A., BoldN., DomsaO., A generator of sequences of hierarchical tests which contain specified keywords. 11-th IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2016, Timisoara, Romania, May 12-14(2016)

[11] Wang F., Wang W., Yang H., Pan, Q.”A novel discrete differential evolution algorithm for computer-aided test-sheet composition problems”, Information Engineering and Computer Science ICIECS, Wuhan, pp. 1 – 4, (2009).

[12] Popescu D. A., Bold N. ”Forming Tests from Questions with Different Theoretical and Practical Degree”. SOCIALEDU, CARE. Communications in Computer and Information Science, vol 677. Springer, Cham, (2016).

[13] Popescu D. A., Bold N., Nijloveanu D., A Method Based on Genetic Algorithms for Generating Assessment Tests Used for Learning, Journal Polibits, vol. 54, pp.53-60, (2016)