Списание „Математика, компютърни науки и образование“
том 1, брой 1, 2018

Journal “Mathematics, Computer Science and Education”
Vol. 1, No. 1, 2018

Prevention of Cheating in Paper-And-Pencil Tests for the Organizational and Technological Conditions of the Bulgarian Schools with the Help of Web-Based Test Generator Software

Philip Petrov

Abstract: *The current research focuses on using computer software as a help for preparing, printing and evaluating paper-and-pencil tests. The practical realization includes a database design and a web-based frontend for users of three different roles with upgrading permissions. The proposed software can be used as an effort to prevent the common cheating when students pass their tests between each other by implementing a strategy for different problems with shuffled options to different students.*

Keywords: *test, generator, paper-and-pencil, exams.*

THE COMPUTER-BASED TESTING IN BULGARIAN SCHOOLS

In the recent years, there is a trend for increased usage of computer-based testing for assessment of students in schools and universities [5, 9–17]. Many schools in Bulgaria use LCMS platforms like Moodle and Blackboard, which both include great capabilities and plugins for e-assessments; however, paper tests are still far from disappearing. There are still many schools with insufficient computer labs or web access – even in the top schools, the computer laboratories are not easily available for non-IT oriented subjects. This is especially true for Bulgaria, as can be seen from the statistics in [5, 37–39], which clearly shows that Bulgaria is at the top in Europe with close to 90% of the computers located in specialized computer labs when the average for EU is below 60% (primary education) and around 65% (secondary). Therefore, we often see that teachers of non-IT subjects are using the LCMS platforms for homework assignments but are still performing their assessments with “paper-and-pencil”. Moreover, many (usually) older non-IT savvy teachers are accustomed to the traditional form of testing. It is hard to say if the future with technology enhancements will eventually make the paper test to disappear, but it is clear that it is still going to stay for a long time.

THE PROBLEM WITH CHEATING

One of the most common problems with assessments in ongoing evaluations is the cheating when one student prompts the correct answer to a neighbor. This is well known for ages for the paper-and-pencil tests and is increasing issue with the online tests performed from distance [13, 2–12]. If we focus only on tests performed in class under direct supervision of the teacher, we should not expect any difference in the cheating amount between paper-and-pencil and computer tests. Many computer-based tests software already provide technological solutions like shuffling the options in multiple-choice exams and/or choosing different problems from the same topic with the same difficulty for each different student (this is only possible when a large enough problems database is available). It is very easy for the computer to manage such entropy. Shuffling of the options is not expected to negatively affect students score [3, 1–10] but the different problems to different students approach should be performed much more carefully as more issues arise from the scope like if the problems are really with equal difficulty or not.

The first question is how easy it is to implement the described strategies for paper-and-pencil tests. Even with small classes, the preparation of such tests will cost a lot of time for teachers. It is not hard to generate a set of different tests with the LCMS module or specialized exams software once, use the shuffling algorithms and find an efficient way to print; however, after that eventually human (the teacher) must evaluate the submitted papers. It happens that checking multiple-choice exams with shuffled answers and even worse with different problems is much more time consuming because it must be performed with very careful reading of the whole text. In addition, the evaluation process goes much more error-prone. On the other side, when all tests are the same, the teacher can easily memorize the number of the problem and the correct answer

position – that way he can evaluate quick and easy without a big risk for producing an error. This is a lot speedier for both multiple-choice exams and open answer tests.

There are software-based solutions for evaluation of paper-based tests that use OCR software, which can be used even without the need of specialized scanner hardware like shown in [6, 560–571]. Most of the software usually require a special area on the first page on the paper-based test with printed problem numbers and fields of the possible answers (checkboxes for multiple-choice questions or a blank area for open-answer problems). The students must write the correct answers very carefully in that area without scratching outside of the fields. Then teachers must scan or photograph the lists and submit the images to some OCR software, which will "digitalize" the answers and forward them to the computer-based tests software for automatic evaluation. The shuffling of answers and different problems to different students is not widely used with that method but it is not impossible to implement. One of the ways to accomplish this is by printing unique identifier string to every test which can be decoded by the software to show which problem from the database is chosen and how exactly the answers are shuffled in it. The uid string must look like a hash to humans and must not include information about what the correct answer is (even when decoded) – that information should be available for the software from its database. While this approach is a great enhancement, it still has two major caveats for usage in schools for ongoing exams:

- It requires at least one computer and eventually a scanner in the classroom or elsewhere the evaluation cannot be performed directly in front of the students in class after they pass their tests;
- It works stable only for multiple-choice exams. When an open-answer problem is introduced, the evaluation usually is mixed between the computer and the teacher. The big problem here is that the available OCR software products are still not good enough for recognizing hand-written text and evaluating it as a correct or wrong answer, especially when it involves special characters like for example what we often have in mathematic formulas or even graphs of functions [12, 1–7]. The enhancement of technologies in that area is big over the years so we may expect that issue to be resolved in near future but nowadays the usage of OCR for such evaluation is limited.

Another approach to implement "different tests to different students", which due to our experience is much more commonly used in the practice, is to give each test an unique id (1, 2, 3, …) and keep a separate (secret – available only to the teacher) list with correct answers of each test id. That way when a student pass a test for evaluation, the teacher can get the sheet with the correct answers with the same id and quickly compare the correctness of the answers by the corresponding problem numbers. It is questionable if that approach is slower or faster than the OCR one (it depends on the size of the test – larger tests should be faster with OCR and smaller tests will be faster by hand); however it does not have the two issues described above. The major caveat with this approach is the entropy – our practice showed that the teachers prepare relatively small amount of different tests (usually around 30, which covers only one class) and repeat them every year. That was not a big problem (other than students from one class to leak some tests to students from a neighboring class in the time between their exams) until recent years when students started to secretly photograph the tests with their smartphones and then share the information in social networks. It is not hard to imagine how such "leaks" are very helpful for the cheaters later when the rumor is spread that the tests repeat from one year to another. Students are unifying their efforts to cheat with the help of the global network so in response teachers should unify their efforts in the anti-cheating as well.

The current project aimed to solve the entropy problem of the paper-and-pencil tests by implementing the following strategy:

- Centralized expandable database design which can store different variants of problems, classified by different subjects and classes based on the Bulgarian curriculum;
- Web-based interface with "shopping cart alike" adding of problems for generation of tests, which includes an easy print option for both generated tests and a sheet with correct answers;
- No need for installation of specialized software or plugins – just a regular web browser and an internet connection must be sufficient for using the system.

The practical realization of the prototype system is inspired by the theory presented in [1, 344–350].

EXISTING SOFTWARE

It is no surprise that most of the software products for test generation are mainly focused on building computer-based testing. Many of them have printing options or there are easy ways for the teacher to transfer a generated test to a word processing editor; however this highly limits the ease to “randomize” the tests between different students. Still there are some products which can achieve closer results to the proposed target. A comparison of four popular paper-and-pencil test generators was made: EasyTestMaker [4, 1–2], HelpTeaching [8, 1–2], GoConqr [7, 1–2] (which is not only test generator but more like a learning social network) and SchoolHouseTest 4 [11, 1–7].

Table 1.
Comparison of existing paper-and-pencil test generators

Software	License	Shuffling problems order	Shuffling distractors	Database with problems	Software installation	Answers sheet
Easy Test Maker	Paid with very limited free version	Three groups only (paid version only)	Three groups only (paid version only)	No	No	Printable
Help Teaching	Paid with limited free version	No	No	Yes, without problem variants	No	Printable
GoConqr	Free with upgrade option to remove ads and get larger data storage	Only when taking the test online, elsewhere hard to achieve	No	Yes, without problems variants	No	Online only
School House Test 4	Paid	Must be done one test at a time	Must be done one test at a time	No	Yes	Printable

All of the compared programs include some type of license fee. Three of the products are online and only SchoolHouseTest requires installation; though subjectively it has the best design and it was easiest to use. All products except GoConqr support printing of the answers sheet.

HelpTeaching and GoConqr do have databases with ready tests which are either prepared by professionals or community submitted. None of them have the different variants of the same problem option which is highly limiting the “different problems to different students” idea from [1, 344–350].

Two of the products provide the ability to not only shuffle distractors but also to shuffle the problems order. EasyTestMaker provides ability for printing up to three “alternate versions” to achieve that. This can work fine when separating the students in groups. Surprisingly the SchoolHouseTest software have that option in the menu; however it does not allow bulk printing, so it must be done one test a time (randomize -> print -> randomize -> print -> etc). The other two software products do not support shuffling of printable tests or it is hard to achieve. The author of the current project considers that the approach for shuffling problems order is much less effective than providing different variants of the same problem to different students.

The full comparison of all products is presented on Table 1. As none of them fully cover the theory in [1, 344–350] and no other product which covers the desired functionality was found, this inspired the creation of a new prototype software.

THE PROTOTYPE DATABASE

The core component of the system is the database with problems. Figure 1 shows the ER Diagram. The users and users_cookie_tokens tables are for the login functionality. The passwords are SHA256 hashed with key stretching using salt and pepper [10, 20–42]. There are three roles in the system:

- *Regular users* do not login, but are able to browse the website with read-only access to the database, add problems to cart, generate and print tests;
- *Authors* can do what the regular users can and additionally can propose new problems for the database;
- *Admins* can do what the authors can and in addition can moderate (approve, disapprove and edit) the new problems which other authors submit.

Even by design the authors are trusted and knowledgeable users, the moderation is a key part for the system design. It is not that much for error correctness but more about styling – the problems must be submitted in similar visual format because elsewhere the prints on paper can look very scrambled and off-position.

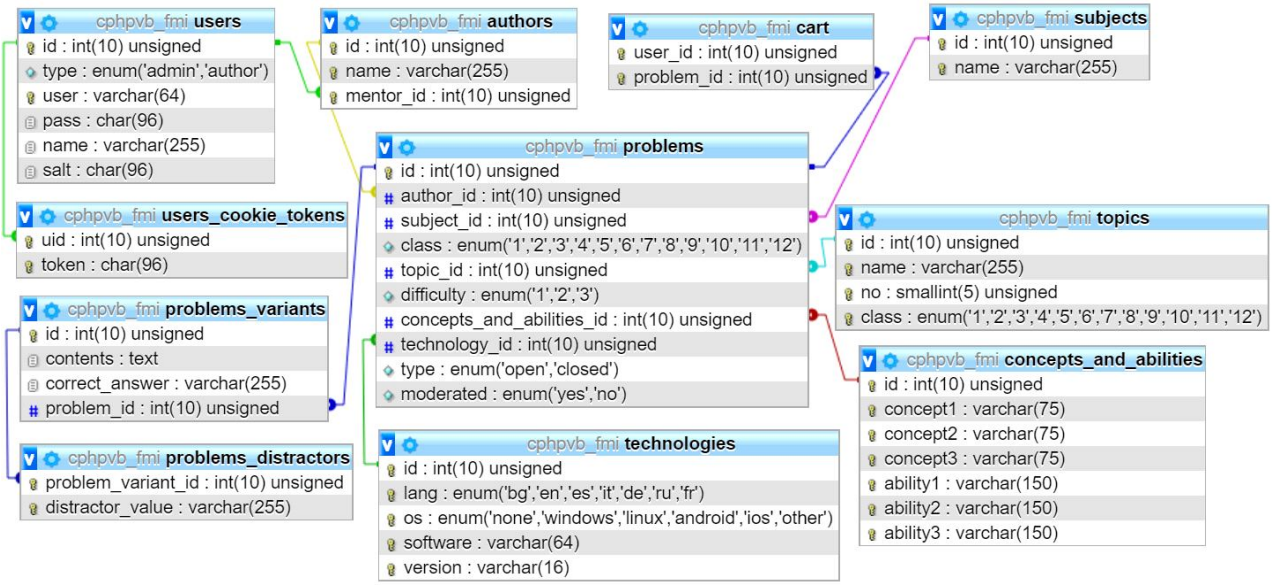

Figure 1. Entity-Relationship Diagram for problems database

The *authors* table is an extension of the users with the meaning that each user can submit problems from different authors, not only by himself. That way he can honor the real authorship of the problems.

The *problems* table is the central relation of the database. It includes the metadata of the problems which is recorded with id and references to the following helper tables:

- *authors* – as explained above;
- *subjects* – describes for which subject the problem is;
- *topics* – what topic from the curriculum the problem covers;
- *technologies* – describes if the problem requires specific technologies like special software, operating system, etc.;
- *concepts_and_abilities* – by design each problem can cover up to three major concepts and up to three major abilities. In future this can be extended further to be more flexible.

The *difficulty* column is the weight of the problem, which is graded 1, 2 and 3. The idea of that metadata is that when we print tests the easier problems (lower difficulty) should appear before the harder

problems (higher difficulty), so practically speaking it's about sorting. The *class* column is describing the grade in school – in Bulgaria there are 12 grades. The *type* column defines the problem as open answer problem or a multiple-choice exam (closed answer). When the problem is multiple-choice exam, additional data is added in the *distractors* table which is a subclass. The *moderated* column defines if the problem is available for the public database or it is still held under moderation by admins. Overall the described metadata of the problems in the designed database covers the idea of the so called "problems passport" in [1, 344–350].

The *problems_variants* table contains the actual text of each different problem. The foreign key pointing to the problems table groups the similar problems (same subject, topic, class, etc.). The correct answer for each problem variant is also stored in that table and it is essential for the teacher answers print.

The *problems_distractors* table stores the additional data for multiple-choice exams. Here we propose a model with free amount of distractors for every problem which differs from [1, 344–350] where the authors propose strictly fixed limit of 4 distractors.

THE PROTOTYPE FRONTEND

The prototype of the system is using interface with Bulgarian language only. Since it is a presentation-only and not a production use software, the design is not intended to be attractive for end users. It is all online-based and requires only a standard web browser for all user types.

The page available for all users is the "tests generator" (Figure 2). The first page is split in two halves vertically. The left half is a search engine for problems and the right half is the "cart". In the search form if a specific metadata option is not checked, it will include all available entries from that option with the search action. Checking an option will restrict the search results to match the chosen entry. The search results are displayed in a simple table, which dis-plays the metadata of all matched problems. The logged in user have options to preview the problem (in new tab, available for all users), add chosen problems to cart (all users) or to edit and delete problems (admins only and authors only on problems which are submitted by them and are not moderated yet). The preview is in two halves with the metadata on the left and all problem variants on the right with alternating colors (Figure 3).

Going back to the main page, the user can choose how many students will take the exam and hit the "Generate" button. This will lead the users to a page with a randomly generated test (Figure 4). The test is paginated with every next student starting on a new page. There is an additional option to print the correct answers for each test on a separate list. The web form is a simple text area covered with CKEditor [2, 1–2] with some extra plugins enabled (like for adding math formulas in Latex, uploading images, etc.). The "Print" button utilizes the web browser built-in functionality. An additional option for saving generated tests for the archive of the current user is planned for future - it will require a simple extension of the database with an additional "history" table, which will log the contents of the text area, a reference to the logged-in user and a timestamp.

At the top menu, there are additional options available for authors and admins. The "editor" of problems is for adding new problems (authors and admins) or editing/deleting existing problems (admins only). The problems are also submitted using CKEditor.

Lastly, the "moderator" of problems (available for admins only) is a simple table, which presents a queue with problems awaiting moderation and links to preview, edit and then approve or dismiss submitted problems.

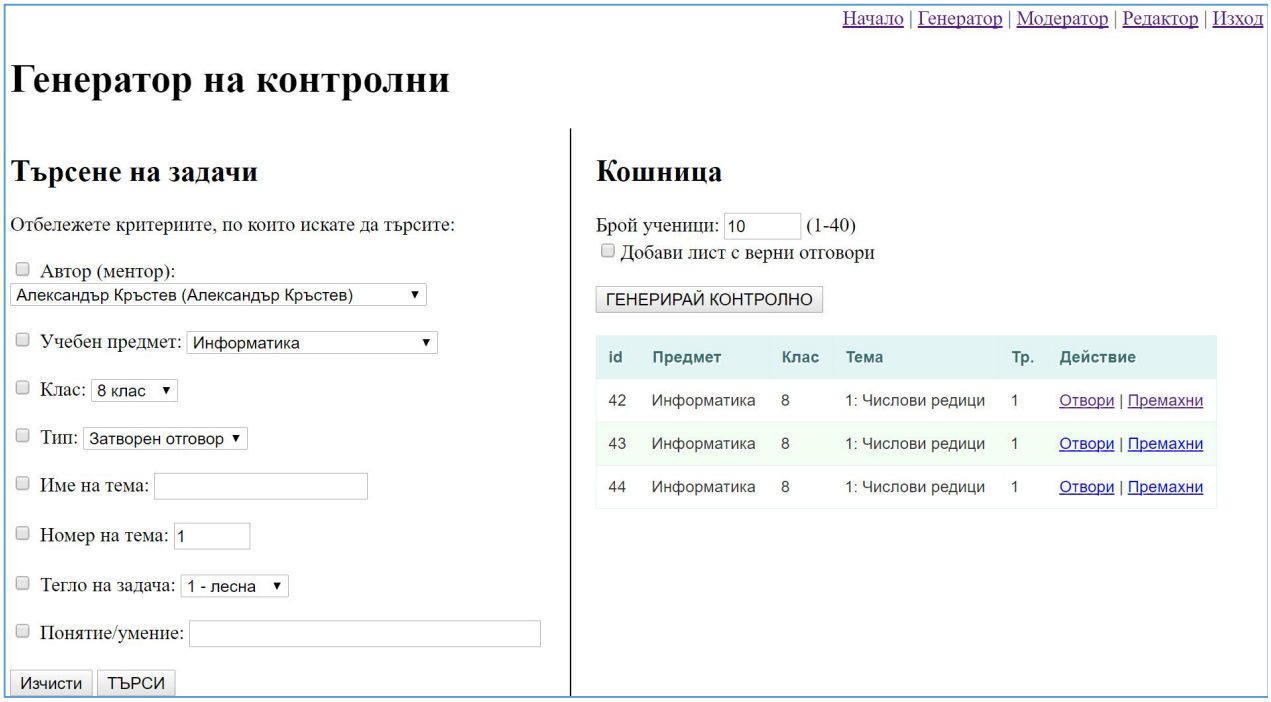

Figure 2. Tests generator first page

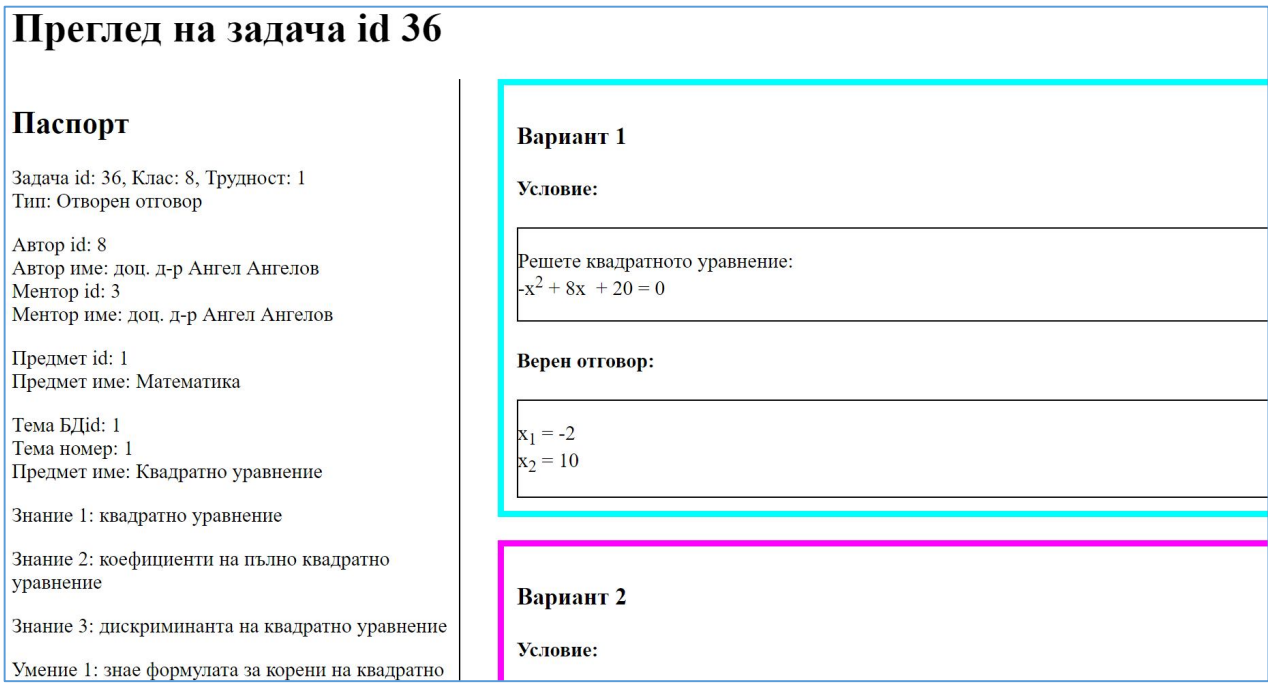

Figure 3. Example preview of a problem

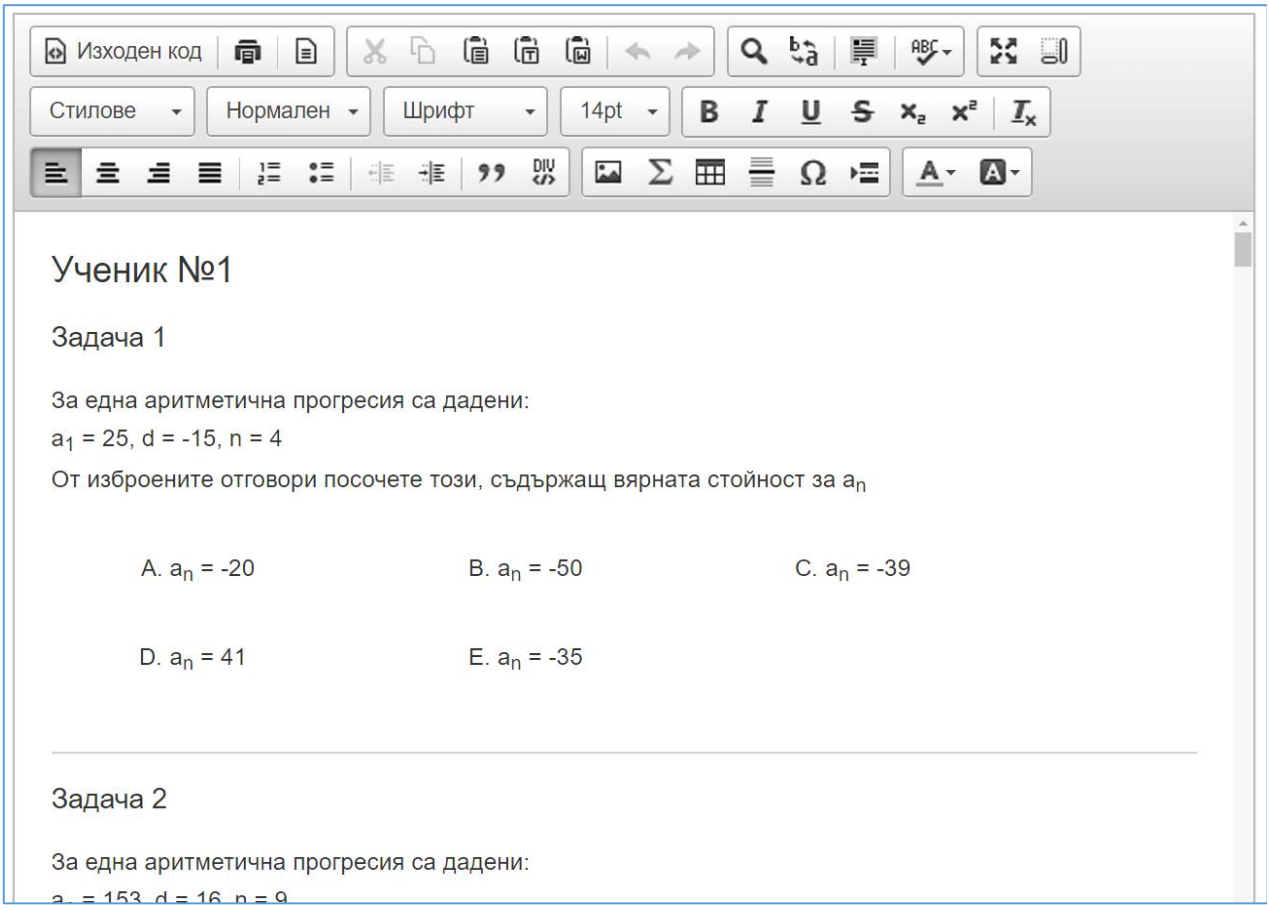

Figure 4. Example test ready for print

From security point of view, the system is protected against XSRF forgeries with a secret token on every form. The protection against eventual XSS attacks is achieved using HTMLawed [9, 1–2].

CONCLUSION

The presented prototype covers the main functionalities of a system for generating and print-ing “paper and pencil” tests theory presented in [1, 344–350]. The article author is the developer of the database schema and the prototype software while the content inside (the problems and their variants) were submitted from various co-workers in Sofia University “St. Kliment Ohridski”, department “Education of mathematics and informatics”. The system was not tested on a large scale yet so it is not possible to show decent feedback from real users; but the author considers that it achieved its main purpose as a working prototype of such system.

FUTURE WORK

A future extension of the prototype must provide a “flood protection” since currently an eventual hacked account of one author can submit unlimited amount of problems to the moderation queue and flood the database with junk entries. Eventually a full production release project must evolve into a CMS-like multiuser system. The database design may be tuned with additional extensions like login history for users, problems versioning control, recycle bin for deleted problems, and other minor updates. A feature for shuffling of the problems order is easily possible to achieve too; however it must be done with respect to the problems difficulty (“the easier problems first” approach must be followed).

The major non-technical work towards production release is the addition of new problems in the database. Currently the prototype system stores only 114 problems (mainly from few topics in mathematics grades 8 and

11 and few examples from IT) with 898 variants; however, this is far from the ambition to provide a database which covers all topics from many different subjects in all 12 grades. The frontend will require a big redesign towards responsiveness and mobile-friendly version.

Using such system only by a single user or by a small group of teachers may be impractical because its nature requires feeding the database with lots of problems. Therefore bringing such system to production use will require a big investment of time and efforts of many problems authors and this can be its main economical obstacle. So further work may require a larger project on national level.

REFERENCES

[1] **Angelov, Dzivev. 2017.** Angelov, A. Dzivev, V. Test generator. *Mathematics and Informatics*. 60 (4), 344–350.

[2] **CKEditor.** Retrieved from https://ckeditor.com/

[3] **Čupić, Šnajder, Bašić. 2009.** Čupić, M. Šnajder, J. Bašić, B. D. Post-test analysis of automatically generated multiple choice exams: a case study. *International Conference on Interactive Computer Aided Learning (ICL 2009)*, Villach, Austria, 1–10.

[4] **Easy Test Maker. 2018.** Retrieved from https://easytestmaker.com.

[5] **European Schoolnet. 2013.** *Survey of schools: ICT in education. Benchmarking access, use and attitudes to technology in European schools*. ISBN 978-92-79-28121-1.

[6] **Fisteus, Pardo, Garcia. 2013.** Fisteus, J. A. Pardo, A. Garcia, N. F. Grading Multiple Choice Exams with Low-Cost and Portable Computer-Vision Techniques. *Journal of Science Education and Technology*, 22(4), 560–571.

[7] **GoConqr. 2018.** Retrieved from https://www.goconqr.com.

[8] **Help Teaching. 2018.** Retrieved from https://www.helpteaching.com.

[9] **htmLawed. 2018.** Retrieved from http://www.bioinformatics.org/phplabware.

[10] **LeBlanc, Messerschmidt. 2016.** LeBlanc, J., Messerschmidt, T. *Identity and Data Security for Web Development: Best Practices*. O'Reilly Media.

[11] **Schoolhouse Test 4. 2018.** Retrieved from https://www.schoolhousetech.com/test.

[12] **Stoliński, Bieniecki. 2010.** Stoliński, S., Bieniecki, W. Evaluation of selected examination tasks with aid of image processing and analysis. *Proc. of the SIS Conference*, Łódź. Retrieved from http://wbieniec.kis.p.lodz.pl/research/files/10_sis_e-exam.pdf.

[13] **Watson, Sottile. 2010.** Watson,G., Sottile, J. Cheating in the Digital Age: Do Students Cheat More in Online Courses?. *Online Journal of Distance Learning Administration 13.1 (2010)*.