Olympiads in Informatics, 2022, Vol. 16, 23–34

DOI: 10.15388/ioi.2022.03

Common Approaches to Developing Extensible E-learning Systems

Bojan KOSTADINOV[1], Irena STOJMENOVSKA[2]
[1]*Cloud Solutions LLC*
st. Jurij Gagarin 33/27 Skopje, Macedonia
[2]*University American College Skopje*
Treta Makedonska Brigada 60, Skopje, Macedonia
e-mail: bojankostadinov@gmail.com, irena.stojmenovska@uacs.edu.mk

Abstract. Education plays an important role in all of our lives, as it allows people to reach their potential, develop problem-solving skills, and create more opportunities for employment and self-dependency. Many countries around the world are investing heavily in schools and universities, as an educated population is linked to stronger economic growth, happiness, stability, and a reduction in poverty and crime. One of the best ways to provide equal opportunities for good education is through the use of technology and multimedia. In this paper, we will provide a broad overview of e-learning systems and their most common features, and discuss common patterns and approaches for maintaining and extending them, as well as how to improve their performance and stability using the latest available strategies and technologies. Additionally, this paper features several use cases of such systems, and examples of how they were maintained, extended and improved over time, while making sure privacy concerns and usability are not negatively affected.

Keywords: e-learning, extensibility, web-based systems, STEM education.

1. Introduction

The knowledge, skills and motivation of workers is one of the key factors for both business, regional and country growth. Jobs that have a higher barrier to entry for new employees tend to pay higher salaries, offer more worker benefits and other incentives – while, on the other hand, jobs with a lower barrier to entry (such as those that don't require education or training degrees) tend to offer much less benefits and lower salaries, as companies are able to fill those positions much more easily (Janjua *et al.*, 2011). Offering better primary and secondary education tends to allow poor and underdeveloped countries to bring people out of poverty, and leads to stronger and more sustainable economic growth (Appiah, 2017). Similarly, companies are investing heavily in specialized education for workers, in order to stay in front of the competition. With automation and AI, it is expected that education will play an even bigger role in the future, as the skills demanded by the labor market will undoubtably change. To summarize, education is one of the most powerful and proven instruments for increasing economic growth, reducing poverty, and lowering inequality.

E-learning systems allow governments, companies and educational institutions to share knowledge with the help of modern devices and technology – which might include sharing textual learning materials, animations, multimedia, and other content forms. Because e-learning systems might be accessed remotely, and are usually web-based, they are one of the main methods for allowing students of all backgrounds to have similar opportunities to access quality education. Most of the time, e-learning systems are used in addition to traditional on-site learning, but some organizations offer students the ability to complete an entire curriculum online, which might include earning a degree or getting a certificate for completing a course or specialization. Some additional advantages of e-learning are consistency, up-to-date content, quick delivery, personalization, analytics, cost effectiveness and better time management.

One example of a popular e-learning system is Moodle, which is an open-source learning platform that is available in several different languages, and which can be used in both primary schools, secondary schools, universities and other companies and establishments. It can be easily installed and hosted on a web server, and enables different ways of sharing content, as well as offering students various features such as a personalized dashboard, calendars, forums, wikis, quizzes, notifications, file sharing, and much more. One of the main benefits of using Moodle is the rich plugin ecosystem (i.e., its extensibility), which allows anyone to add additional features to Moodle, such as new integrations with other systems, gamification, activities, new types of quiz questions or certificates. Other popular and accessible software, such as the content management system WordPress, which can be used for educational purposes as well, also offers administrators the ability to extend the system with plugins and themes.

In this paper, starting with section 2, we will outline several common approaches to developing extensible e-learning systems such as Moodle, how to maintain and improve them over time, and how to keep libraries and tools up to date in order to make sure the systems are performant and secure. Extensibility is a key feature of any software system, and there are several common theoretical strategies and approaches which apply to all systems (including e-learning systems), as well as several differences which must be taken into account.

Next, in section 3, we will provide three use-cases and examples of existing e-learning systems, and how they have been maintained and extended over a period of several years – while making sure they remain performant, stable and usable in different usage scenarios (for example, students using modern web browsers from home, and connecting from outdated browsers in schools). All of the examples are e-learning systems which are used in Macedonia, and all of them have a large userbase of students and teachers. At the end of the paper, we will summarize our findings.

2. Developing Extensible E-learning Systems

Extensibility is one of the key software engineering principles, which corresponds to enabling and planning for future growth and updates. It can be thought of as a metric of the difficulty involved with changing existing features or implementing new ones. Extensibility is also an important consideration when discussing system security and reusability – as we would like to be able to easily implement fixes for newly discovered vulnerabilities and requirements.

A software system will go through different phases in its lifetime, including the initial development, evolution, maintenance, phasing out and termination. Different strategies might be employed to extend the life of a software system (Lavelle, 2005), including waiting (requires no changes), wrapping with middleware, renovating into a more modern form (for example, updating the UI), replacing with a new system which is more up-to-date or offers more features, and finally outsourcing the maintenance or development of such a system outside the organization (so resources can be focused elsewhere).

Such approaches are applicable to e-learning systems as well. Because of privacy concerns (which are stricter when it applies to pupils and students), rules around tracking and data collection, or security and network limitations, specific care must be taken when integrating an e-learning system with new software or outsourcing features elsewhere. There are various rules with regards to storing student data, and some organizations or schools might have Internet limitations that only allow certain domains or IP addresses to be available in classrooms. For example, in e-learning systems we often need to store content such as pdf or video files, but deciding to move the storage of such files to a cloud offering such as S3 in Amazon Web Services might lead to a different set of problems. Similarly, some schools might have specific licensing agreements for operating systems or use outdated browsers, so proper planning and testing is required when deciding to switch to a completely different software system. What might work for one organization or school, might be prohibited at another, so it's best to make sure that the system can be used independently (or, alternatively, that it depends on as little outside services as possible), or that all restrictions are known beforehand.

The first major consideration when discussing extensibility is understanding the different forms that exist, and naturally these will also apply to e-learning systems: black-box extensibility, white-box extensibility, and gray-box extensibility. Black-box extensibility refers to systems where knowledge about the internal system implementation is not important when implementing new features or extensions (only a specification regarding interfaces is used). In contrast, white-box extensibility refers to extensibility where we are modifying the source code of the system. Gray-box extensibility is a compromise between the two.

The second major consideration is related to the source code, where we need to make sure that we follow appropriate principles and patterns to make the code understandable, testable, and easily editable. **SOLID** is a set of five design principles associated with the quality of object-oriented programming, which is not only related to writing understandable software, but also one which is flexible, extensible, and maintainable. These principles include:

- The single-responsibility principle.
- The open–closed principle.
- The Liskov substitution principle.
- The interface segregation principle.
- The dependency inversion principle.

In short, the principles indicate that there shouldn't be multiple reasons for a single class to change – i.e. a class should have only one responsibility; that software entities should be open for extension but not modification; that derived classes should be usable in place of base classes; that multiple client-specific interfaces are a better alternative than one general-purpose interface (to decrease coupling between systems); and finally, that we should depend on abstractions vs concretions – i.e. objects should not create their dependencies (i.e. we should pass them).

One common approach to extensibility is using plugins. For example, in WordPress, this is accomplished through hooks (actions or filters), which are a way for a function in one piece of code to interact or trigger actions elsewhere – at pre-defined spots. Actions execute code and return without passing anything back, while filters modify and pass values back to be used later in the code. A good extensible system is one where the internal structure or architecture doesn't change (or is minimally affected) by the addition of new functionality.

The third major consideration is related to following a proper project management structure (Yagel, 2017), which includes using version control, documentation, issue databases and API guidelines.

Finally, the fourth major consideration is connected to testing. **Automation testing** is a process that involves an execution of test suites that ensure the quality, security, and performance of a software system. This can be accomplished through different forms of tests, including graphical user interface tests and API tests. Correct use of tests can lead to the development of software which contains less bugs, is more performant, and (of interest to us in this paper) software which is more extensible. Tests enable us to detect issues early in the development cycle (of the software itself, or the development of a new feature), supports easier debugging and enables catching regressions early on – allowing for quicker development and faster releases of new versions. There are different types of tests, including unit tests, integration tests, smoke tests, regression tests, performance tests, etc. End-to-end testing solutions, such as (for example) Cypress – a powerful automation tool for testing any application that runs in a browser, guarantee that changes don't break existing features for users by testing the system in a similar way that a user would. Popular software solutions used in automation testing allow us to execute parallel tests using different platforms, operating systems, and browsers – thus simulating various environments and restrictions that might exist at different organizations (such as, for example, in schools).

3. Examples

In the previous section, we have discussed some important topics that relate to the development of extensible systems – which include proper planning, testing, modularization, architecture, and code structure – all of which are crucial when we are planning to make modifications to any software system, as it might affect its usage, quality, or performance. In this section, we will describe several examples of existing e-learning systems. For all of the outlined systems, the authors of the research paper have been involved either with building or maintaining them. For each system, we will provide a brief overview, describe where it is being used, and discuss the challenges related to maintaining and extending it.

3.1. *MENDO*

MENDO is an online e-learning system that enables anyone to learn programming, study algorithms, and to solve algorithmic tasks. The system is primarily intended for primary and secondary school students, but it has also been actively used by both coaches and universities for conducting online courses and exercises. It offers automated grading – where solutions, which are submitted as source code, are first compiled, and are then fed batches of input data. At the end, their output is tested for correctness and the solutions are awarded an appropriate number of points. MENDO can work with solutions written in several programming languages (and can easily be extended to support more), can work with various types of tasks (including interactive ones), and the system can restrict programs to arbitrary time and memory limitations. MENDO also supports several different ways of comparing the output and grading the solutions – for example, checking for exact equality, comparing number values, ignoring whitespace, writing custom comparators which can (even) award a custom percentage of the points, etc. Aside from learning from interactive tutorials and solving tasks at any time in the day, MENDO acts as a form of gateway for algorithmic programming, offering a news page, forum, wiki and links to books and guides.

The system itself is also used by the Computer Society of Macedonia for conducting both school, regional and national competitions in informatics, as well as organizing the Macedonian Olympiad in Informatics. Through the years, these competitions were organized in different formats (both online and on-site), with different methods of scoring contestants and offering feedback – requiring constant updates and modifications. The organization of all of these events by CSM has resulted in the creation of most of the tasks that are currently available on the system.

Since the organization of the first event in 2010, the system has had more than 16000+ registered users, 1000+ tasks related to algorithmic programming, 530000+ submitted and graded solutions, and more than 40 tutorials for both starting with programming (specifically, learning C++) and getting familiar with algorithms and data structures. The system supports two languages (Macedonian and English).

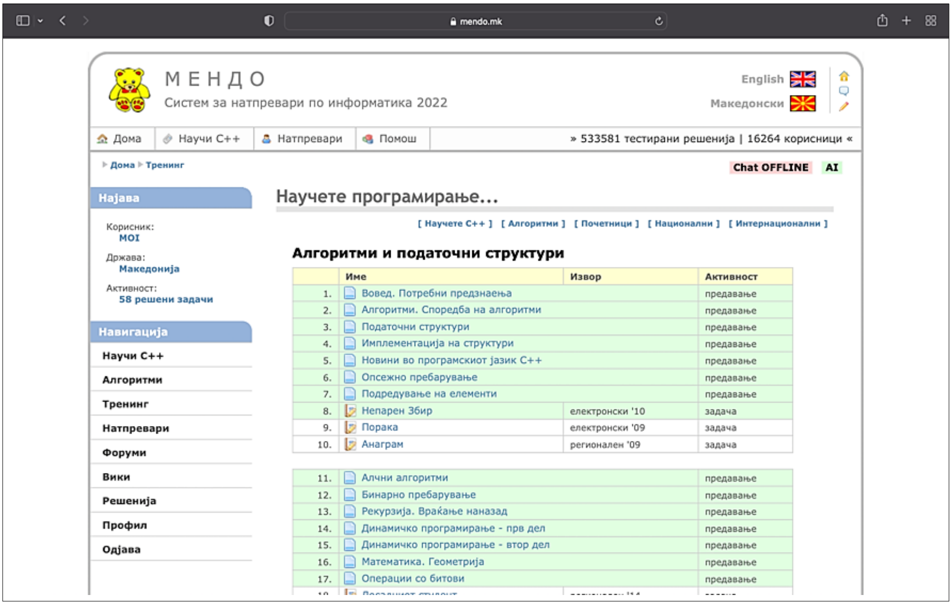


Fig. 1. Screenshot of the MENDO training system – list of tutorials.

The first version of MENDO is described in (Kostadinov *et al.*, 2010), and features several modules: sandbox, grader, controller and auxiliary. Most of the modules were written in Java, and the web application was running on the Apache Tomcat server. It was one of the rare e-learning systems which automatically graded programming solutions on Microsoft Windows. This was done through the sandbox module, which used P/Invoke (Platform Invoke) signatures and Win32 functions to create processes (like the program that a student submits) and group them in jobs. Jobs have the ability of limiting the privileges and resources available to the processes.

In addition, MENDO also controlled other software on the operating system on which it ran, and provided automatic backups for itself and other applications. MySQL was used as an open-source relational database management system to store (most of) the data – like users, task details, submissions, news, etc. The key details that allowed MENDO to support so many features was its modular design which allowed new actions and features to be added by simply implementing appropriate interfaces, as well as the number of quality tests and superb test coverage.

After the release of the system, several updates were made throughout the years, including making design modifications, adding additional features (such as interactive lessons, automated help and hints, or competition statistics – as shown in Fig. 2 below – among others). Some examples of the automated hints include the ability of the system to quickly analyze the source code and, based on the grading results and the source code, to provide feedback to the user regarding simple mistakes such as printing extra data (for example, “The result is: …”), submitting to a wrong task, writing to files, having uninitialized variables, using forbidden imports or functions, etc. Due to the modular nature of the system, and the various test suites created with the first version, most of these

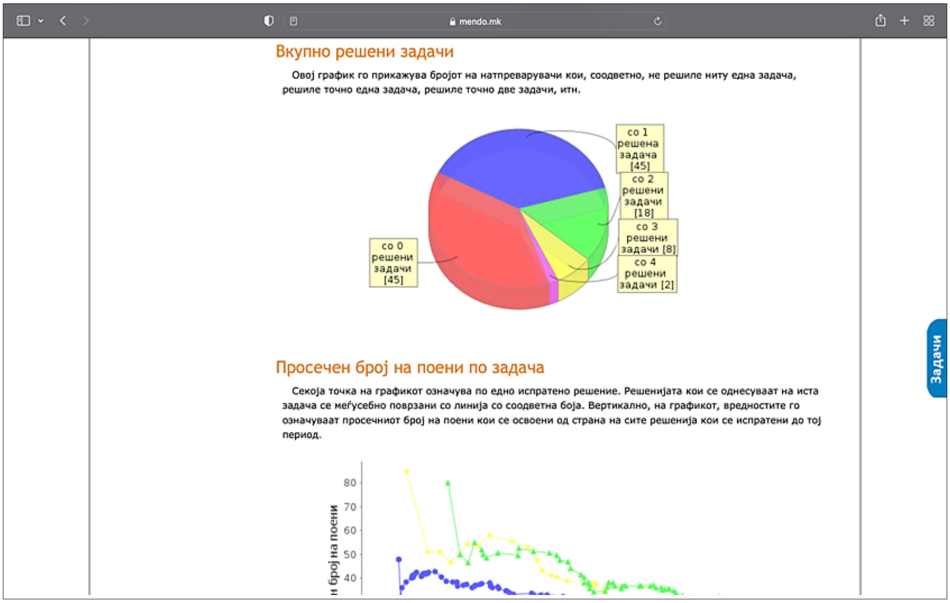


Fig. 2. Screenshot of the MENDO training system – competition statistics.

updates were done smoothly and without any major issues. Because several schools in Macedonia are using browsers which are significantly outdated and don't support many recent features and technologies, we have kept the user interface dependencies fairly consistent. Using virtual machines, we are testing new releases before pushing them to production. Release plans take into consideration firewalls and privacy concerns, and the system itself is visible through a single IP address and doesn't use CDNs.

One of the most significant changes to MENDO, was in regards to where the system is hosted. The first version of the system was running on Microsoft Windows, and all submissions were executed and graded there. In contrast, the latest version of the system is running on Linux, which required a different sandbox implementation and various changes to the connections between MENDO and the other systems that it interfaces with (specifically, the forums and the wiki – as their database connectors had certain issues when we moved them to Linux).

Because the sandbox is just another module in the MENDO system, replacing it was a matter of writing another module that can interact with MENDO and that implements the correct API. We selected around 1000+ submissions from the old system, which had various different verdicts, and tested them with the new sandbox to make sure that the results would be the same – indicating everything works as expected. Only minor issues were detected, like some submissions passing a handful of extra test cases because of the speed difference between the original machine where MENDO was running and the new machine where tests were being executed.

Due to automation tests, all of the requirements were tested, and all discovered issues were solved within days – after which we moved all traffic to the new system when it was confirmed that everything was running smoothly.

3.2. *Bebras*

Bebras is an international challenge organized once per year, which aims to promote informatics and computational thinking among primary and secondary school students. In Macedonia, as in most other countries where the challenge takes place, students participate in the annual Bebras challenge from their school, supervised by their teachers (note: in 2021, because of COVID-19, the challenge took place online).

Countries use different systems to organize the challenge – which could be Moodle, a custom contest management system created specifically for the challenge, pen & paper, or something else. In Macedonia, we use a locally developed system which supports various types of tasks (fill-in-the-blank, multiple-choice questions, and interactive challenges), requires minimum resources, works with different types of browsers and operating systems, and one which supports multiple languages. The system also enables organizers to schedule practice and e-learning sessions, without organizing a specific competition. Because of all the specific requirements around organization, communication, task types, scalability, and performance, the Macedonian Bebras system was developed as a separate system that can function independently and which offers different views for teachers, students and organizers. An example screenshot from the system (student view) is shown on Fig. 3.

The three main concerns we had with running an event with tens of thousands of participants were related to 1) communication, 2) the required performance of the system, and 3) the sizeable number of outdated browsers. To successfully tackle these concerns, the system must be simple to use (so that teachers or students can easily participate with-

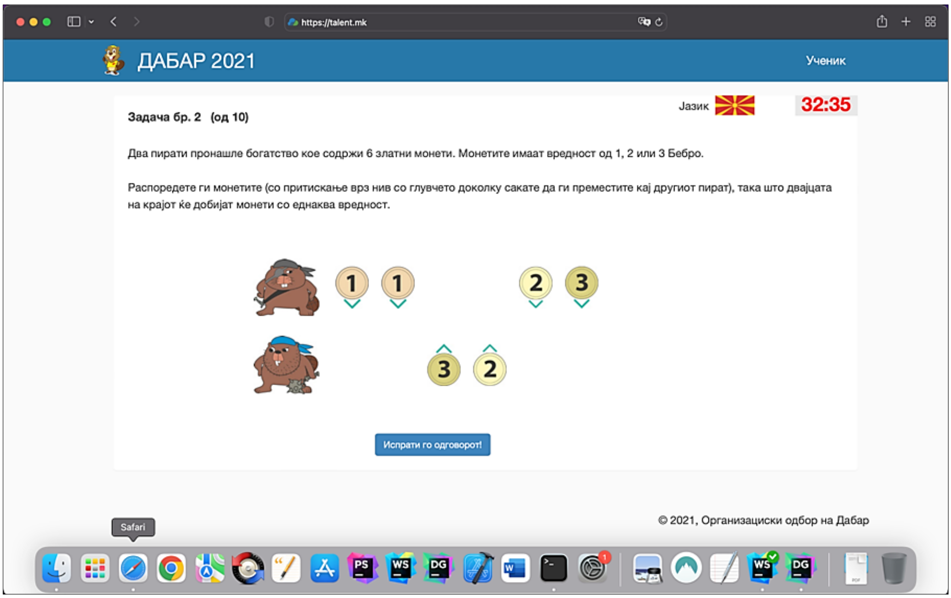


Fig. 3. Screenshot of the Macedonian Bebras system.

out much issues and direct communication with organizers), should be performant so it can support many concurrent users, and the system must support (and be tested) on various browsers. In Macedonia, for example, some computers are limited to using the now (extremely) outdated Firefox 3 (first released in 2008), because of a project organized around that time by the government, aimed at purchasing and delivering new computers to schools. To organize a successful event, plans include testing the system with virtual machines which can run outdated operating systems and browsers, creating benchmarks for stress testing the system, and creating an architecture and content which can be easily extended, tested, and maintained (this is because Bebras usually contains a lot of interactive tasks, which must be properly coded and tested on several browser environments). Cypress was used for testing the final version of the system, before starting the event each year, by automating the flows that students would take on the system and randomly entering the possible answers.

The created system uses an ORM for storing data in two databases – the data is written immediately in one database, while the second one (asynchronously) gets and writes the same data to a ledger in order to make sure all teacher and student information is safely stored – this is used, for example, after the event in order to distribute results and certificates to participants.

Besides structuring the source code properly so it can be extended and modified easily during multiple iterations of the event (for which we followed guidelines such as those outlined in Section 2); having automated backups, proper planning, extensive testing and strong cooperation with teachers was crucial in creating a useful competition management system, and organizing an event with such a large number of participating schools, teachers and students.

3.3. *New E-learning System Based on Item Response Theory*

In (Kostadinov & Stojmenovska, 2022), we describe a distributed e-learning system based on item response theory, that adapts to students, and offers them test questions and learning materials which correspond to their level of knowledge. The system is composed of a main server component where the main data is located, and several separate web agents which should be run at schools (optimally) or other accessible web servers – these agents are the web locations where students connect to. Agents synchronize data with the main server component at configurable regular intervals.

The system has a collection of tasks and explanations, and functions in such a way that it adapts to students based on their determined level of knowledge and skills. It is originally designed to be used by primary and secondary school students, but it might be extended with more content in the future to support more groups and use cases. After a student answers a certain question, the system will show them the correct solution and will also include a brief tutorial regarding the idea and theory behind it. Afterwards, if the student solved the task correctly, the system would generally proceed with a harder task – and vice versa, if the student answers incorrectly, the system will continue with a simpler task. This process is repeated continuously multiple times,

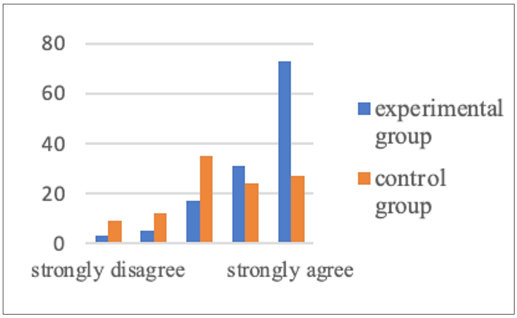


Fig. 4. Students' willingness to use the web applications.

with different tasks, while a student is studying. A more detailed description of the system, and how it uses item response theory to choose the appropriate tasks, can be found in (Kostadinov & Stojmenovska, 2022).

We tested the system with many students (in mathematics), who were split in two categories – one category was using the system as intended, while the other one received questions at random and had access to learning materials in pdf format (commonly shared that way by some schools in our country) instead of the prepared explanations. The gathered results were extremely positive, indicating strong support by students for wanting to continue to use the system.

Because this is a distributed e-learning system, and there are various privacy rules related to storing student data, the web agents that are run at schools are the only ones that store student information. The data synchronized with the main server component is anonymized and is mostly related to task data, answers and content, as defined in the referenced research paper. To guarantee that schools with different machines and operating systems can run the e-learning system on their infrastructure, we use Docker containers and an SQLite relational database for each agent. A well-defined, versioned API is used for communication between the components, which guarantees that future updates will not break existing systems. This is very important when designing distributed systems that are expected to be modified in the future, and especially ones which need to be installed in different locations and environments. (Note that, for the gathered data presented above, the web agents were running on our infrastructure, instead of in schools – this was done in order to simulate a real working environment where we can test the newly created e-learning system, without asking teachers to setup web servers and networks themselves, as they might not have the necessary time, privileges or skills.)

4. Conclusion

Education is a powerful agent of change, and there is a very strong link between education and quality of life. A good education enables people to prepare for the job market, empowers them to understand essential facts about the world around them, and drives both short- and long-term economic growth. Research has indicated that countries with higher education and literacy rates, also tend to have a stronger GDP and lower unemployment rate (Appiah, 2017). Establishments and governments throughout the world have recognized education as an important tool to transform their organization (on a smaller scale) or society (on a larger scale) to be more productive, stable, independent, and to promote equality among different groups.

The use of educational technology and e-learning is one of the best methods that allows pupils to have similar access to learning opportunities in life. There are different types of e-learning systems, but some of the advantages that are common between them include the ability to learn at one's own pace, to cover material at a time that suits people, and to digest up-to-date content.

Extensibility is one of the main software engineering principles, and it corresponds to enabling, designing, and planning for future growth and updates. There are various strategies that might be employed to extend the life of a software system, which might include wrapping with middleware, renovating/updating, outsourcing, or replacing with another system.

Designing and writing quality source code that follows design principles is also very important, and the same applies to testing, project management and documentation. E-learning systems are sometimes installed on schools' hardware, and these systems might need to follow certain strict privacy guidelines, or to be updated less frequently due to licensing restrictions. Similarly, the school or organization using the system might have network firewalls and/or outdated operating systems and browsers. All of this must be taken into consideration when creating such systems, as it influences the number of options that can be utilized, the required support of multiple API versions, the use of tech, and the ability to outsource services like hosting static files or using Content Delivery Networks (CDNs).

References

Appiah, E.N. (2017). The effect of education expenditure on per capita GDP in developing countries. *International Journal of Economics and Finance*, 9(10), 136–144.

AWS S3 – Amazon Web Services – Cloud Object Storage. (accessed 13/5/2022): `https://aws.amazon.com/s3/`

Janjua, P.Z., Kamal, U.A. (2011). The role of education and income in poverty alleviation: A cross-country analysis. *The Lahore Journal of Economics*, 16(1), 143–172.

Kostadinov, B., Jovanov, M., Stankov, E. (2010). A new design of a system for contest management and grading in informatics competitions. In: *ICT Innovations Conference 2010*, *Web Proceedings*, 87–96.

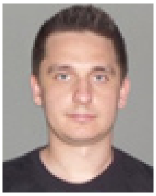
Kostadinov, B., Stojmenovska I. (2022). Creating a distributed, privacy-aware e-learning system based on item response theory. In: *19th International Conference on Informatics and Information Technologies (CIIT 2022)*.

Lavelle, G. (2005). Practical strategies for dealing with legacy systems. *Risk Management*, 52(2), 45–46.

MENDO – The Macedonian contest management system. (accessed 12/5/2022): `https://mendo.mk/`

Moodle – Open-source learning platform. (accessed 11/5/2022). `https://moodle.org/`

Yagel, R. (2017). Extending Software Systems While Keeping Conceptual Integrity. *In: SEKE* (pp. 432–435).



B. Kostadinov is the founder of Cloud Solutions, an author, and a former competitive programmer. In 2014, he defended his MSc thesis in Intelligent information systems at the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, in Skopje. He is currently a PhD student at UACS, and is one of the organizers of the national competitions in informatics in Macedonia.

I. Stojmenovska is a Dean of the School of Computer Science and Information Technology at the University American College – Skopje. She is a PhD holder in Theoretical Mathematics. Her main research interests are in abstract algebra, in particular including vector valued algebraic structures and combinatorial theory within. She also has research interests related to mathematical education and information systems and management. Prof. Stojmenovska is a member of the European Mathematical Society and appears as an author/coauthor of 40 peer-reviewed research papers published so far.